OSE course projects

Prof. Dr. Philipp Eisenhauer

Dec 08, 2021

CONTENTS

Our two courses OSE data science and OSE scientific computing require students to work on independent projects. This documentation includes some basic instructions and example projects.

CHAPTER

ONE

PROJECTS

Grading for OSE data science and OSE scientific computing is based on a group project due at the end of the semester, which should be presented in the form of a Jupyter Notebook. We encourage you to code your projects in Python, you may also use R or Julia. You can submit the project in the form of a GitHub repository or pull request on an existing repository (depending on your project).

You are free to select a topic of your choice related to the contents of the respective class. For example, you can either replicate the core results of a computational publication or apply for the chance to work on a collaboration project with one of our private sector partners. Other project ideas include running a benchmarking exercise for an algorithm, contributing to one of our group's software packages of your choice, or creating a notebook similar to the ones presented in the lectures on a computational topic that interests you. Note that several textbooks explore the implementation of involved computational economic models, porting their implementation to Python can serve as a valuable starting point for your project.

Note for students taking EPP:

Participants of the course "Effective Programming Practices for Economists" by Professor Hans-Martin von Gaudecker are welcome to submit their project for grading in OSE data science or OSE scientific computing. Note that the project still has to fulfill the topic and submission requirements listed above in addition to any requirements stated by the EPP course. Please reach out in the course Zulip chat for any questions about the project.

1.1 Kaggle Competitions

Kaggle hosts numerous (causal) machine learning tasks often sponsored by companies. In this context, a typical course project describes the competition you participated in and implements a version of your solution strategy. You can then, for example, explore the impact of alternative numerical components of your solution and investigate the effect of tuning parameters on its performance.

1.2 Replication Projects

You can replicate and extend a research article related to the topics of the course. A typical replication notebook starts with presenting the baseline article, reproducing selected key results, critical assessment of quality, and an independent contribution such as robustness check and visualizations. As a starting point for the introductory part of your notebook, please consider the recent article by Berk & al. (2017) on How to Write an Effective Referee Report.

1.3 Collaboration Projects

Collaboration projects with our partners from the private sector allow students to directly put the skills acquired during class into action, gain hands-on experience in a professional data science setting, and receive feedback and mentoring from seasoned data scientists. Collaboration projects are announced in class, where we also provide further details about the application process.

1.4 Example Projects

1.4.1 Replication Projects

Here are some examples of replication projects from earlier iterations of the OSE data science course.

Angrist (1990)

The randomly assigned risk of induction generated by the draft lottery is used to construct estimates of the effect of veteran status on civilian earnings. These estimates are not biased by the fact that certain types of men are more likely than others to service in the military. Social Security administrative records indicate that in the early 1980s, long after their service in Vietnam was ended, the earnings of white veterans were approximately 15 percent less than the earnings of comparable nonveterans.

Project by Pascal Heid

Replication of Angrist (1990): Lifetime earnings and the Vietnam era draft lottery: Evidence from social security administrative records

Project by Pascal Heid, Summer 2020.

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from auxiliary.auxiliary_figures import get_figure1, get_figure2, get_figure3
from auxiliary.auxiliary_tables import (
    get_table1,
    get_table2,
    get_table3,
    get_table4,
)
from auxiliary.auxiliary_data import process_data
from auxiliary.auxiliary_visuals import background_negative_green, p_value_star
from auxiliary.auxiliary_extensions import (
    get_flexible_table4,
    get_figure1_extension1,
    get_figure2_extension1,
    get_bias,
    get_figure1_extension2,
    get_figure2_extension2,
)
import warnings
```

(continues on next page)

(continued from previous page)

```
warnings.filterwarnings("ignore")
plt.rcParams["figure.figsize"] = [12, 6]
```

This notebook replicates the core results of the following paper:

> Angrist, Joshua. (1990). Lifetime Earnings and the Vietnam Era Draft Lottery: Evidence from Social Security Administrative Records. *American Economic Review*. 80. 313-36.

In the following just a few notes on how to read the remainder:

- In this excerpt I replicate the Figures 1 to 3 and the Tables 1 to 4 (in some extended form) while I do not consider Table 5 to be a core result of the paper which is why it cannot be found in this notebook.
- I follow the example of Angrist keeping his structure throughout the replication part of this notebook.
- The naming and order of appearance of the figures does not follow the original paper but the published correction.
- The replication material including the partially processed data as well as some replication do-files can be found here.

1. Introduction

For a soft introduction to the topic, let us have a look at the goal of Angrist's article. Already in the first few lines Angrist states a clear-cut aim for his paper by making the remark that "yet, academic research has not shown conclusively that Vietnam (or other) veterans are worse off economically than nonveterans". He further elaborates on why research had yet been so inconclusive. He traces it back to the flaw that previous research had solely tried to estimate the effect of veteran status on subsequent earnings by comparing the latter across individuals differing in veteran status. He argues that this naive estimate might likely be biased as it is easily imaginable that specific types of men choose to enlist in the army whose unobserved characteristics imply low civilian earnings (self-selcetion on unobservables).

Angrist avoids this pitfall by employing an instrumental variable strategy to obtain unbiased estimates of the effect of veteran status on earnings. For that he exploits the random nature of the Vietnam draft lottery. This lottery randomly groups people into those that are eligible to be forced to join the army and those that are not. The idea is that this randomly affects the veteran status without being linked to any unobserved characteristics that cause earnings. This allows Angrist to obtain an estimate of the treatment effect that does not suffer from the same shortcomings as the ones of previous studies.

He finds that Vietnam era veterans are worse off when it comes to long term annual real earnings as opposed to those that have not served in the army. In a secondary point he traces this back to the loss of working experience for veterans due to their service by estimating a simple structural model.

In the following sections I first walk you through the identification idea and empirical strategy. Secondly, I replicate and explain the core findings of the paper with a rather extensive elaboration on the different data sources used and some additional visualizations. Thirdly, I critically assess the paper followed by my own two extensions concluding with some overall remarks right after.

2. Identification and Empirical Approach

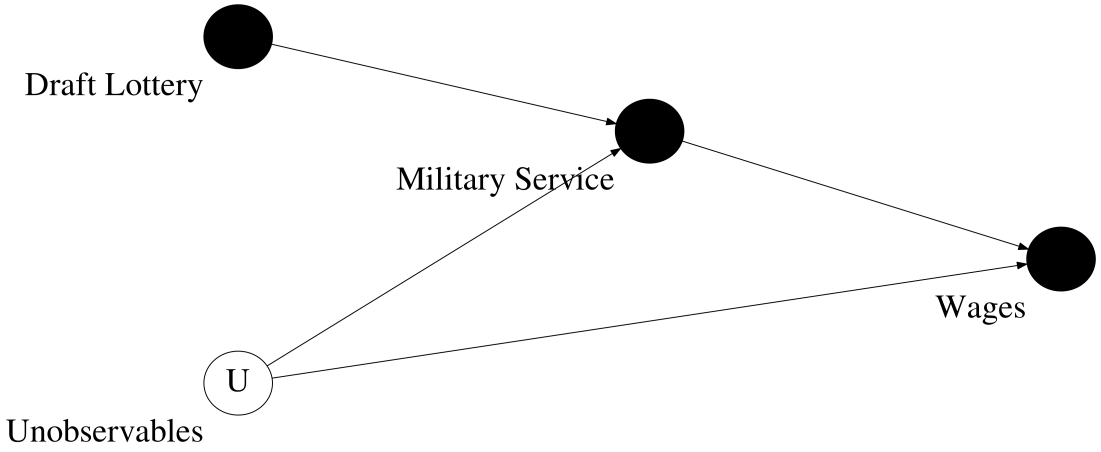
As already mentioned above the main goal of Angrist's paper is to determine the causal effect of veteran status on subsequent earnings. He believes for several reasons that conventional estimates that only compare earnings by veteran status are biased due to unobservables that affect both the probability of serving in the military as well as earnings over lifetime. This is conveniently shown in the causal graph below. Angrist names two potential reasons why this might be likely. First of all, he makes the point that probably people with few civilian opportunities (lower expected earnings) are more likely to register for the army. Without a measure for civilian opportunities at hand a naive estimate of the effect of military service on earnings would not be capable of capturing the causal effect. Hence, he believes that there is probably some self-selection into treatment on unobservables by individuals. In a second point, Angrist states that the selection criteria of the army might be correlated with unobserved characteristics of individuals that makes them more prone to receiving future earnings pointing into a certain direction.

Econometrically spoken, Angrist argues with the following linear regression equation representing a version of the right triangle in the causal graph:

$$y_{cti} = \beta_c + \delta_t + s_i\alpha + u_{it}.$$

He argues that estimating the above model with the real earnings y_{cti} for an individual i in cohort c at time t being determined by cohort and time fixed effects (β_c and δ_t) as well an individual effect for veteran status is biased. This is for the above given reasons that the indicator for veteran status s_i is likely to be correlated with the error term u_{it}.

Angrist's approach to avoid bias is now to employ an instrumental variable approach which is based on the accuracy of the causal graph below.



The validity of this causal graph rests on the crucial reasoning that there is no common cause of the instrument (Draft Lottery) and the unobserved variables (U). Angrist provides the main argument that the draft lottery was essentially random in nature and hence is not correlated with any personal characteristics and therefore not linked to any unobservables that might determine military service and earnings. As will be later explained in more detail, the Vietnam draft lottery determined randomly on the basis of the birth dates whether a person is eligible to be drafted by the army in the year following the lottery. The directed graph from Draft Lottery to Military Service is therefore warranted as the fact of having a lottery number rendering a person draft-eligible increases the probability of joining the military as opposed to a person that has an excluded lottery number.

This argumentation leads Angrist to use the probability of being a veteran conditional on being draft-eligible in the

lottery as an instrument for the effect of veteran status on earnings. In essence this is the Wald estimate which is equal to the following formula:

$$\hat{\alpha}_{IV,WALD} = \frac{E[earnings \mid eligible = 1] - E[earnings \mid eligible = 0]}{E[veteran \mid eligible = 1] - E[veteran \mid eligible = 0]}$$

The nominator equals to the estimated α from equation (1) while the denominator can be obtained by a first stage regression which regresses veteran status on draft-eligibility. It reduces to estimating the difference in conditional probabilities of being a veteran $prob(veteran \mid eligible = 1) - prob(veteran \mid eligible = 0)$. Estimates for this are obtained by Angrist through weighted least squares (WLS). This is done as Angrist does not have micro data but just grouped data (for more details see the data section in the replication). In order to obtain the estimates of the underlying micro level data it is necessary to adjust OLS by the size of the respective groups as weights. The above formula is also equivalent to a Two Stage Least Squares (2SLS) procedure in which earnings are regressed on the fitted values from a first stage regression of veteran status on eligibility.

In a last step, Angrist generalizes the Wald grouping method to more than just one group as instrument. There are 365 lottery numbers that were split up into two groups (eligible and non-eligible) for the previous Wald estimate. Those lottery numbers can also be split up even further into many more subgroups than just two, resulting in many more dummy variables as instruments. Angrist splits the lottery numbers into intervals of five which determine a group j. By cohort c he estimates for each group j the conditional probability of being a veteran p_{cj}. This first stage is again run by WLS. The resulting estimate $\hat{p}_{cj}$ is then used to conduct the second stage regression below.

$$\bar{y}_{ctj} = \beta_c + \delta_t + \hat{p}_{cj}\alpha + \bar{u}_{ctj}$$

The details and estimation technique will be further explained when presenting the results in the replication section below.

3. Replication

3.1 Background and Data

The Vietnam Era Draft Lottery

Before discussing how the data looks like it is worthwhile to understand how the Vietnam era draft lottery was working in order to determine to which extent it might actually serve as a valid instrument. During the Vietnam war there were several draft lotteries. They were held in the years from 1970 to 1975. The first one took place at the end of 1969 determining which men might be drafted in the following year. This procedure of determining the lottery numbers for the following year continued until 1975. The table below shows for which years there were lotteries drawn and which birth years were affected by them in the respective year. For more details have a look here.

Year	Cohorts	Draft-Eligibility Ceiling
1970	1944-50	195
1971	1951	125
1972	1952	95
1973	1953	95
1974	1954	95
1975	1955	95
1976	1956	95

The authority of drafting men for the army through the lottery expired on June 30, 1973 and already before no one was drafted anymore. The last draft call took place on December 7, 1972.

The general functioning of those seven lotteries was that every possible birthday (365 days) was randomly assigned a number between 1 and 365 without replacement. Taking the 1969 lottery this meant that the birthdate that had the number 1 assigned to, it caused every man born on that day in the years 1944 to 1950 to be drafted first if it came to a draft call in the year 1970. In practice, later in the same year of the draft lottery, the army announced draft-eligibility ceilings determining up to which draft lottery number was called in the following year. In 1970, this means that every man having a lottery number of below 195 was called to join the army. As from 1973 on nobody was called anymore, the numbers for the ceiling are imputed from the last observed one which was 95 in the year 1972. Men with lottery numbers below the ceiling for their respective year are from here on called "draft-eligible".

Being drafted did not mean that one actually had to serve in the army, though. Those drafted had to pass mental and physical tests which in the end decided who had to join. Further it should be mentioned that Angrist decides to only use data on those that turned 19 when being at risk of induction which includes men born between 1950 and 1953.

The Data

Continuous Work History Sample (CWHS)

This administrative data set constitutes a random one percent sample draw of all possible social security numbers in the US. For the years from 1964 until 1984 it includes the **FICA** (social security) earnings history censored to the Social Security maximum taxable amount. It further includes FICA taxable earnings from self-employment. For the years from 1978 on it also has a series on total earnings (**Total W-2**) including for instance cash payments but excluding earnings from self-employment. This data set has some confidentiality restrictions which means that only group averages and variances were available. This means that Angrist cannot rely on micro data but has to work with sample moment which is a crucial factor for the exact implementation of the IV method. A group is made of by year of earnings, year of birth, ethnicity and five consecutive lottery numbers. The statistics collected for those also include the number of people in the group, the fraction of them having taxable earnings equal and above the taxable maximum and the fraction having zero earnings.

Regarding the actual data sets available for replication we have the data set `cwhsa` which consists of the above data for the years from 1964 to 1977 and then `cwhsb` which consists of the CWHS for the years after.

Above that Angrist provides the data set `cwhsc_new` which includes the **adjusted FICA** earnings. For those Angrist employed a strategy to approximate the underlying uncensored FICA earnings from the reported censored ones. All of those three different earnings variables are used repeatedly throughout the replication.

[3]:
```
process_data("cwhsa")
```

[3]:
```
                                                earnings  earnings variance  \
ethnicity birth year year lottery interval
1         44         64   1                 1691.030029        1480.599976
                          2                 1535.430054        1359.020020
                          3                 1818.010010        1604.420044
                          4                 1636.380005        1626.270020
                          5                 1889.800049        1639.609985
...                                                 ...                ...
2         53         77   69                3643.739990        4273.600098
                          70                4127.490234        5623.089844
                          71                4712.459961        4588.279785
                          72                4676.939941        5321.140137
                          73                4651.870117        4989.020020

                                                sample size  \
ethnicity birth year year lottery interval
1         44         64   1                       182.0
```

(continues on next page)

(continued from previous page)

```
                          2                          187.0
                          3                          210.0
                          4                          208.0
                          5                          207.0
...                                                    ...
2          53          77   69                         53.0
                          70                          55.0
                          71                          76.0
                          72                          85.0
                          73                          83.0

                                                fraction zero earnings
ethnicity birth year year lottery interval
1          44          64   1                                   0.170
                          2                                   0.187
                          3                                   0.171
                          4                                   0.231
                          5                                   0.184
...                                                             ...
2          53          77   69                                  0.415
                          70                                  0.473
                          71                                  0.316
                          72                                  0.353
                          73                                  0.241

[20440 rows x 4 columns]
```

The above earnings data only consists of FICA earnings. The lottery intervals from 1 to 73 are equivalent to intervals of five consecutive lottery numbers. Consequently, the variable lottery interval equals to one for the lottery numbers 1 to 5 and so on. The ethnicity variable is encoded as 1 for a white person and 2 for a nonwhite person.

[4]:
```
process_data("cwhsb")
```

[4]:
```
                                                        earnings  \
data source ethnicity birth year year lottery interval
TAXAB       1          44          78   1                10625.58
                                        2                11546.46
                                        3                11401.16
                                        4                10899.99
                                        5                11667.14
...                                                           ...
TOTAL       2          53          84   69                6846.43
                                        70               11357.89
                                        71                8695.86
                                        72               14013.24
                                        73               10742.71

                                                        earnings variance  \
data source ethnicity birth year year lottery interval
TAXAB       1          44          78   1                          7052.47
                                        2                          8032.55
                                        3                          7508.27
                                        4                          7342.60
```

(continues on next page)

(continued from previous page)

```
                                          5                              7507.56
...                                                                          ...
TOTAL       2          53          84     69                             9117.49
                                          70                            14734.47
                                          71                             9613.24
                                          72                            14182.30
                                          73                            18095.78

                                                                  sample size  \
data source ethnicity birth year year lottery interval
TAXAB       1          44          78     1                              179
                                          2                              182
                                          3                              209
                                          4                              206
                                          5                              207
...                                                                      ...
TOTAL       2          53          84     69                              53
                                          70                              55
                                          71                              76
                                          72                              84
                                          73                              83

                                                                  fraction zero earnings
data source ethnicity birth year year lottery interval
TAXAB       1          44          78     1                                       0.179
                                          2                                       0.198
                                          3                                       0.196
                                          4                                       0.189
                                          5                                       0.159
...                                                                                 ...
TOTAL       2          53          84     69                                      0.396
                                          70                                      0.455
                                          71                                      0.368
                                          72                                      0.274
                                          73                                      0.506

[20440 rows x 4 columns]
```

As stated above this data now consists of earnings from 1978 to 1984 for FICA (here encoded as "TAXAB") and Total W-2 (encoded as "TOTAL").

Survey of Income and Program Participation (SIPP) and the Defense Manpower Data Center (DMDC)

Throughout the paper it is necessary to have a measure of the fraction of people serving in the military. For this purpose the above two data sources are used.

The **SIPP** is a longitudinal survey of around 20,000 households in the year 1984 for which is determined whether the persons in the household are Vietnam war veterans. The survey also collected data on ethnicity and birth data which made it possible to match the data to lottery numbers. The **DMDC** on the other hand is an administrative record which shows the total number of new entries into the army by ethnicity, cohort and lottery number per year from mid 1970 until the end of 1973. Those sources are needed for the results in Table 3 and 4. A combination of those two are matched to the earnings data of the CWHS which constitutes the data set `chwsc_new` below.

[5]:
```
data_cwhsc_new = process_data("cwhsc_new")
data_cwhsc_new
```

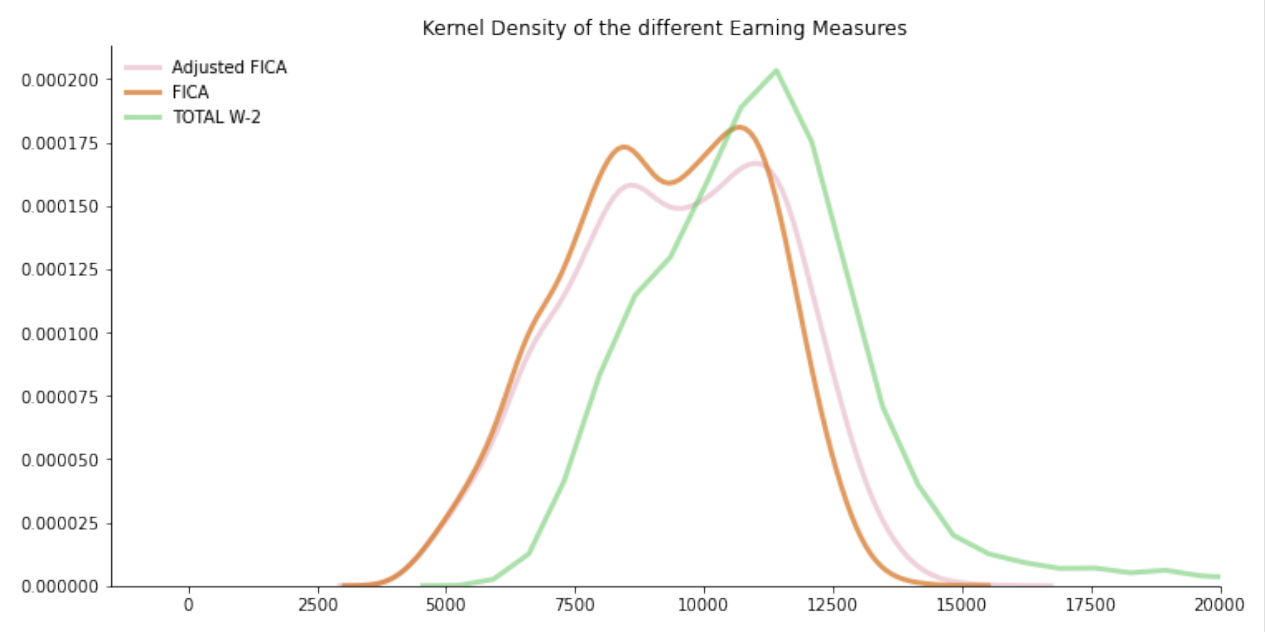
[5]:
```
                                                          earnings  \
data source ethnicity birth year year lottery interval
ADJ         1         50         74   1                 8853.940430
                                  75   1                 9062.639648
                                  76   1                10096.055664
                                  77   1                10916.072266
                                  78   1                11738.444336
...                                                             ...
TOTAL       2         53         84   37                10562.357422
                                      57                 8988.295898
                                      40                 9857.195312
                                      11                 8690.839844
                                      23                 9709.985352

                                                     probability of serving
data source ethnicity birth year year lottery interval
ADJ         1         50         74   1                             0.352700
                                  75   1                             0.352700
                                  76   1                             0.352700
                                  77   1                             0.352700
                                  78   1                             0.352700
...                                                                      ...
TOTAL       2         53         84   37                            0.111818
                                      57                            0.082410
                                      40                            0.111429
                                      11                            0.088025
                                      23                            0.073750

[12818 rows x 2 columns]
```

This data set now also includes the adjusted FICA earnings which are marked by "ADJ" as well as the probability of serving in the military conditional on being in a group made up by ethnicity, birth cohort and lottery interval.

Below we have a short look at how the distribution of the different earnings measures look like. In the table you see the real earnings in 1978 dollar terms for the years from 1974 to 1984 for FICA and adjusted FICA as well as the years 1978 until 1984 for Total W-2.

[6]:
```
for data in ["ADJ", "TAXAB", "TOTAL"]:
    ax = sns.kdeplot(
        data_cwhsc_new.loc[data, "earnings"],
        color=np.random.choice(np.array([sns.color_palette()]).flatten(), 4),
    )
ax.set_xlim(xmax=20000)
ax.legend(["Adjusted FICA", "FICA", "TOTAL W-2"], loc="upper left")
ax.set_title("Kernel Density of the different Earning Measures")
```

[6]:
```
Text(0.5, 1.0, 'Kernel Density of the different Earning Measures')
```



For a more detailed description of the somewhat confusing original variable names in the data sets please refer to the appendix at the very bottom of the notebook.

3.2 Establishing the Validity of the Instrument

In order to convincingly pursue the identification strategy outlined above it is necessary to establish an effect of draft eligibility (the draft lottery) on veteran status and to argue that draft eligibility is exogenous to any unobserved factor affecting both veteran status and subsequent earnings. As argued before one could easily construct reasonable patterns of unobservables that both cause veteran status and earnings rendering a naive regression of earnings on veteran status as biased.

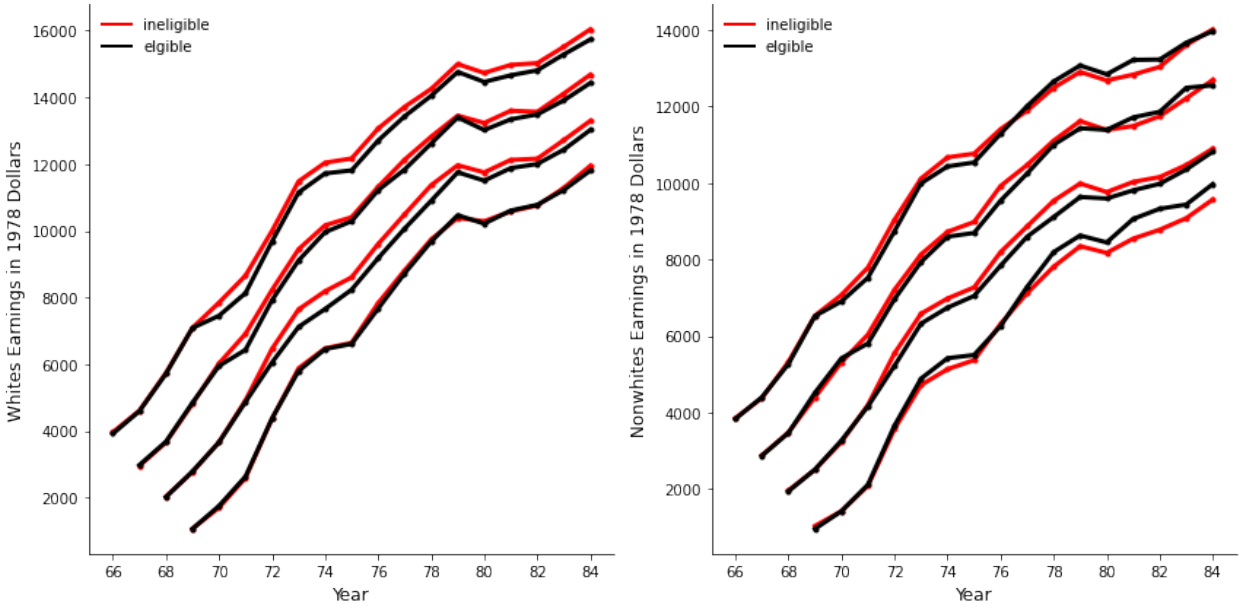
The first requirement for IV to be valid holds as it is clearly observable that draft-eligibility has an effect on veteran status. The instrument is hence **relevant**. For the second part Angrist argues that the draft lottery itself is random in nature and hence not correlated with any unobserved characteristics (**exogenous**) a man might have that causes him to enroll in the army while at the same time making his earnings likely to go into a certain direction irrespective of veteran status.

On the basis of this, Angrist now shows that subsequent earnings are affected by draft eligibility. This is the foundation to find a nonzero effect of veteran status on earnings. Going back to the causal diagram from before, Angrist argued so far that there is no directed graph from Draft Lottery to the unobservables U but only to Military Service. Now he further establishes the point that there is an effect of draft-eligibility (Draft Lottery) that propagates through Military Service onto earnings (Wages).

In order to see this clearly let us have a look at **Figure 1** of the paper below. For white and nonwhite men separately the history of average FICA earnings in 1978 dollar terms is plotted. This is done by year within cohort across those that were draft-eligible and those that were not. The highest two lines represent the 1950 cohort going down to the cohort of men born in 1953. There is a clearly observable pattern among white men in the cohorts from 1950 to 52 which shows persistently lower earnings for those draft-eligible starting the year in which they could be drafted. This cannot be seen for those born in 1953 which is likely due to the fact that nobody was actually drafted in 1973 which would have otherwise been "their" year. For nonwhite men the picture is less clear. It seems that for cohorts 50 to 52 there is slightly higher earnings for those ineligible but this does not seem to be persistent over time. The cohort 1953 again does not present a conclusive image. Observable in all lines, though, is that before the year of conscription risk there is no difference in earnings among the group which is due to the random nature of the draft lottery.

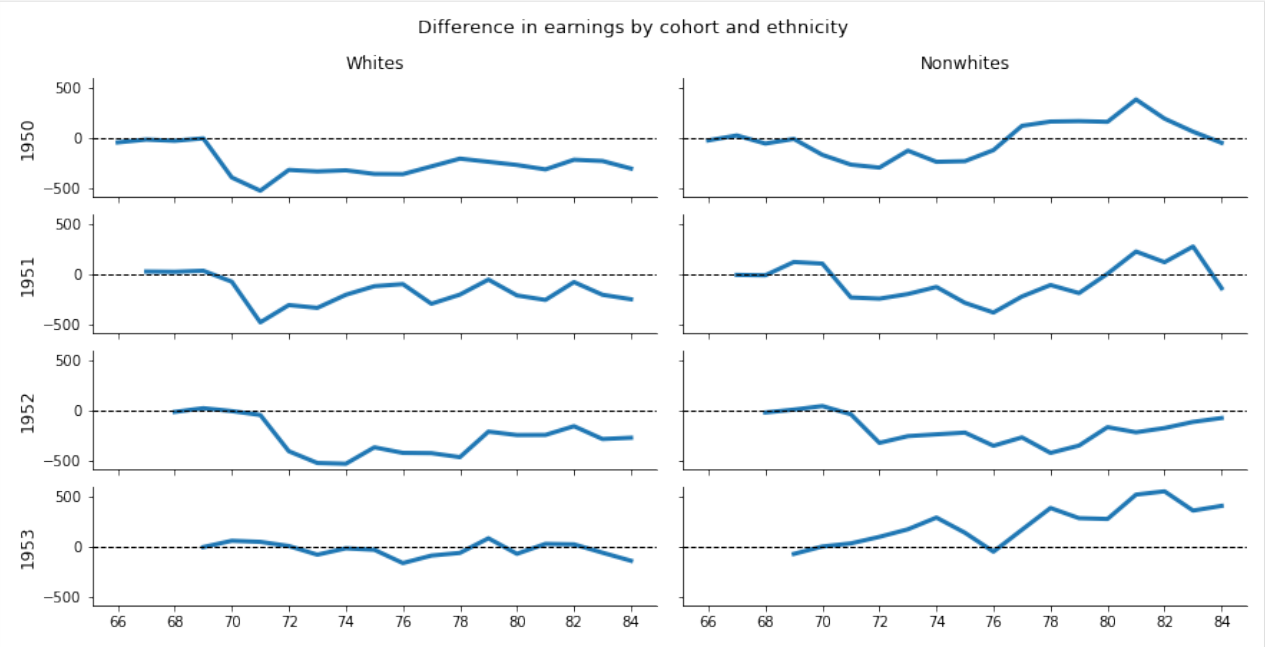
[7]:
```
# read in the original data sets
data_cwhsa = pd.read_stata("data/cwhsa.dta")
data_cwhsb = pd.read_stata("data/cwhsb.dta")
data_cwhsc_new = pd.read_stata("data/cwhsc_new.dta")
data_dmdc = pd.read_stata("data/dmdcdat.dta")
data_sipp = pd.read_stata("data/sipp2.dta")
```

[8]:
```
get_figure1(data_cwhsa, data_cwhsb)
```



A more condensed view of the results in Figure 1 is given in **Figure 2**. It depicts the differences in earnings between the red and the black line in Figure 1 by cohort and ethnicity. This is just included for completeness as it does not provide any further insight in comparison to Figure 1.

[9]:
```
get_figure2(data_cwhsa, data_cwhsb)
```



A further continuation of this line of argument is resulting in **Table 1**. Angrist makes the observations from the figures before even further fine-grained and explicit. In Table 1 Angrist estimates the expected difference in average FICA and Total W-2 earnings by ethnicity within cohort and year of earnings. In the table below for white men we can observe that there is no significant difference to the five percent level for the years before the year in which they might be drafted. This changes for the cohorts from 1950 to 52 in the years 1970 to 72, respectively. There we can observe a significantly lower income for those eligible in comparison to those ineligible. This seems to be persistent for the cohorts 1950 and 52 while less so for those born in 1951 and 1953. It should further be noted that Angrist reports that the quality of the Total W-2 earnings data was low in the first years (it was launched in 1972) explaining the inconlusive estimations in the periods at the beginning.

To focus the attention on the crucial points I mark all the negative estimates in different shades of green with more negative ones being darker. This clearly emphasizes the verbal arguments brought up before.

[10]:
```
table1 = get_table1(data_cwhsa, data_cwhsb)
table1["white"].style.applymap(background_negative_green)
```

[10]:
```
<pandas.io.formats.style.Styler at 0x7f03c498e940>
```

For the nonwhite males there is no clear cut pattern. Only few cells show significant results which is why Angrist in the following focuses on white males when constructing IV estimates. For completeness I present Table 1 for nonwhite males below although it is somewhat less important for the remainder of the paper.

[11]:
```
table1["nonwhite"].style.applymap(background_negative_green)
```

[11]:
```
<pandas.io.formats.style.Styler at 0x7f03c456d0d0>
```

3.3 Measuring the Effect of Military Service on Earnings

3.3.1 Wald-estimates

As discussed in the identification section a simple OLS regression estimating the model in equation (1) might suffer from bias due to elements of s_i that are correlated with the error term u_{it}. This problem can be to a certain extent circumvented by the grouping method proposed by Abraham Wald (1940). Grouping the data by the instrument which is draft eligibility status makes it possible to uncover the effect of veteran status on earnings. An unbiased estimate of α can therefore be found by adjusting the difference in mean earnings across eligibility status by the difference in probability of becoming a veteran conditional on being either draft eligible or not. This verbal explanation is translated in the following formula:

$$\hat{\alpha} = \frac{\bar{y}^e - \bar{y}^n}{\hat{p}(V|e) - \hat{p}(V|n)}$$

The variable $\bar{y}$ captures the mean earnings within a certain cohort and year further defined by the superscript e or n which indicates draft-eligibility status. The above formula poses the problem that the conditional probabilities of being a veteran cannot be obtained from the CWHS data set alone. Therefore in **Table 2** Angrist attempts to estimate them from two other sources. First from the SIPP which has the problem that it is a quite small sample. And secondly, he matches the CWHS data to the DMDC. Here it is problematic, though, that the amount of people entering the army in 1970 (which is the year when those born 1950 were drafted) is only collected for the second half of the year. This is the reason why Angrist has to go with the estimates from the SIPP for the cohort of 1950 while taking the bigger sample of the matched DMDC/CWHS for the birth years 1951 to 53. The crucial estimates needed for the denominator of equation (3) are presented in the last column of Table 2 below. It can already be seen that the differences in earnings by eligibility that we found in Table 1 will be scaled up quite a bit to obtain the estimates for $\hat{\alpha}$. We will come back to that in Table 3.

Note: The cohort 1950 for the DMDC/CWHS could not be replicated as the data for cohort 1950 from the DMDC set is missing in the replication data. Above that the standard errors for the estimates coming form SIPP differ slightly from the published results but are equal to the results from the replication code.

[12]:
```
table2 = get_table2(data_cwhsa, data_dmdc, data_sipp)
table2["white"]
```

[12]:

Data Set	Cohort	Statistic	Sample	P(Veteran)	P(Veteran|eligible)	P(Veteran|ineligible)
SIPP (84)	1950	Value	351.0	0.2673	0.3527	0.1934
		Standard Error		0.0136	0.0215	
	1951	Value	359.0	0.1973	0.2831	
		Standard Error		0.0124	0.0230	
	1952	Value	336.0	0.1554	0.2310	
		Standard Error		0.0111	0.0245	
	1953	Value	390.0	0.1298	0.2192	
		Standard Error		0.0102	0.0313	
DMDC/CWHS	1951	Value	16768.0	0.1176	0.2071	
		Standard Error		0.0025	0.0053	
	1952	Value	17703.0	0.1515	0.2683	
		Standard Error		0.0027	0.0065	
	1953	Value	17749.0	0.1343	0.1548	
		Standard Error		0.0026	0.0053	

(continues on next page)

(continued from previous page)

```
                  Standard Error                0.0166
          1951    Value                         0.1469
                  Standard Error                0.0139
          1952    Value                         0.1257
                  Standard Error                0.0119
          1953    Value                         0.1126
                  Standard Error                0.0104
DMDC/CWHS 1951    Value                         0.0708
                  Standard Error                0.0024
          1952    Value                         0.1102
                  Standard Error                0.0027
          1953    Value                         0.1268
                  Standard Error                0.0029

                                  P(V|eligible) - P(V|ineligible)
Data Set  Cohort Statistic
SIPP (84) 1950   Value                                    0.1594
                 Standard Error                           0.0272
          1951   Value                                    0.1362
                 Standard Error                           0.0269
          1952   Value                                    0.1053
                 Standard Error                           0.0273
          1953   Value                                    0.1066
                 Standard Error                           0.0330
DMDC/CWHS 1951   Value                                    0.1362
                 Standard Error                           0.0059
          1952   Value                                    0.1581
                 Standard Error                           0.0071
          1953   Value                                    0.0280
                 Standard Error                           0.0060
```

[13]:
```
table2["nonwhite"]
```

[13]:
```
                                  Sample  P(Veteran)  P(Veteran|eligible)  \
Data Set  Cohort Statistic
SIPP (84) 1950   Value              70.0      0.1625               0.1957
                 Standard Error                0.0281               0.0449
          1951   Value              63.0      0.1703               0.2014
                 Standard Error                0.0283               0.0497
          1952   Value              52.0      0.1332               0.1449
                 Standard Error                0.0265               0.0525
          1953   Value              55.0      0.1749               0.2247
                 Standard Error                0.0297               0.0762
DMDC/CWHS 1951   Value            5258.0      0.0794               0.1173
                 Standard Error                0.0037               0.0076
          1952   Value            5493.0      0.0953               0.1439
                 Standard Error                0.0040               0.0095
          1953   Value            5303.0      0.0925               0.0984
                 Standard Error                0.0040               0.0079

                                  P(Veteran|ineligible)  \
Data Set  Cohort Statistic
```

(continues on next page)

(continued from previous page)

SIPP (84)	1950	Value	0.1355
		Standard Error	0.0353
	1951	Value	0.1514
		Standard Error	0.0340
	1952	Value	0.1288
		Standard Error	0.0308
	1953	Value	0.1642
		Standard Error	0.0321
DMDC/CWHS	1951	Value	0.0599
		Standard Error	0.0040
	1952	Value	0.0794
		Standard Error	0.0042
	1953	Value	0.0904
		Standard Error	0.0046

			P(V|eligible) - P(V|ineligible)
Data Set	Cohort	Statistic	
SIPP (84)	1950	Value	0.0603
		Standard Error	0.0571
	1951	Value	0.0500
		Standard Error	0.0603
	1952	Value	0.0161
		Standard Error	0.0609
	1953	Value	0.0605
		Standard Error	0.0827
DMDC/CWHS	1951	Value	0.0574
		Standard Error	0.0086
	1952	Value	0.0644
		Standard Error	0.0104
	1953	Value	0.0080
		Standard Error	0.0092

In the next step Angrist brings together the insights gained so far from his analysis. **Table 3** presents again differences in mean earnings across eligibility status for different earnings measures and within cohort and year. The values in column 1 and 3 are directly taken from Table 1. In column 2 we now encounter the adjusted FICA measure for the first time. As a reminder, it consists of the scaled up FICA earnings as FICA earnings are only reported to a certain maximum amount. The true average earnings are likely to be higher and Angrist transformed the data to account for this. We can see that the difference in mean earnings is most often in between the one of pure FICA earnings and Total W-2 compensation. In column three there is again the probability difference from the last column of Table 2. As mentioned before the measure is taken from the SIPP sample for the cohort of 1950 and the DMDC/CWHS sample for the other cohorts. Angrist decides to exclude cohort 1953 and nonwhite males as for those draft eligibility does not seem to be an efficient instrument (see Table 1 and Figure 1 and 2). Although Angrist does not, in this replication I also present Table 3 for nonwhites to give the reader a broader picture. Further Angrist focuses his derivations only on the years 1981 to 1984 as those are the latest after the Vietnam war for which there was data avalaible. Effects in those years are most likely to represent long term effects.

Let us now look at the most crucial column of Table 3 which is the last one. It captures the Wald estimate for the effect of veteran status on adjusted FICA earnings in 1978 dollar terms per year and cohort from equation (3). So this is our $\hat{\alpha}$ per year and cohort. For white males the point estimates indicate that the annual loss in real earnings due to serving in the military was around 2000 dollars. Looking at the high standard errors, though, only few of the estimates are actually statistically significant. In order to see this more clearly I added a star to those values in the last column that are statistically significant to the five percent level.

Note: In the last column I obtain slightly different standard errors than in the paper. The same is the case, though, in

the replication code my replication is building up on.

[14]:
```
table3 = get_table3(data_cwhsa, data_cwhsb, data_dmdc, data_sipp, data_cwhsc_new)
p_value_star(table3["white"], slice(None), ("", "Service Effect in 1978 $"))
```

[14]:
```
First Level               Draft Eligibility Effects in Current $  \
Second Level                                          FICA Earnings
Cohort Year Statistic
1950   1981 Value                                          -435.8
            Standard Error                                 210.6
       1982 Value                                          -320.2
            Standard Error                                 235.9
       1983 Value                                          -349.6
            Standard Error                                 261.7
       1984 Value                                          -484.4
            Standard Error                                 286.8
1951   1981 Value                                          -358.3
            Standard Error                                 203.7
       1982 Value                                          -117.3
            Standard Error                                 229.1
       1983 Value                                          -314.1
            Standard Error                                 253.3
       1984 Value                                          -398.5
            Standard Error                                 279.3
1952   1981 Value                                          -342.9
            Standard Error                                 206.9
       1982 Value                                          -235.1
            Standard Error                                 232.4
       1983 Value                                          -437.7
            Standard Error                                 257.6
       1984 Value                                          -436.1
            Standard Error                                 281.9

First Level                                                        \
Second Level              Adjusted FICA Earnings Total W-2 Earnings
Cohort Year Statistic
1950   1981 Value                         -487.8             -589.7
            Standard Error                 237.6              299.4
       1982 Value                         -396.1             -305.5
            Standard Error                 281.7              345.5
       1983 Value                         -450.1             -512.9
            Standard Error                 302.0              441.2
       1984 Value                         -638.8            -1143.3
            Standard Error                 336.6              492.3
1951   1981 Value                         -428.8              -71.6
            Standard Error                 216.7              423.4
       1982 Value                         -278.6              -72.8
            Standard Error                 251.5              372.2
       1983 Value                         -452.2             -896.6
            Standard Error                 277.7              426.4
       1984 Value                         -573.4             -809.2
            Standard Error                 308.0              381.0
1952   1981 Value                         -392.7             -440.5
            Standard Error                 220.3              265.1
```

(continues on next page)

(continued from previous page)

```
       1982 Value                          -255.3             -514.7
            Standard Error                  254.0              296.6
       1983 Value                          -500.1             -915.7
            Standard Error                  283.3              395.3
       1984 Value                          -560.1             -767.2
            Standard Error                  310.8              376.1

First Level                                                        \
Second Level               P(V|eligible) - P(V|ineligible)
Cohort Year Statistic
1950   1981 Value                                      0.159
            Standard Error                             0.027
       1982 Value
            Standard Error
       1983 Value
            Standard Error
       1984 Value
            Standard Error
1951   1981 Value                                      0.136
            Standard Error                             0.027
       1982 Value
            Standard Error
       1983 Value
            Standard Error
       1984 Value
            Standard Error
1952   1981 Value                                      0.105
            Standard Error                             0.027
       1982 Value
            Standard Error
       1983 Value
            Standard Error
       1984 Value
            Standard Error

First Level
Second Level               Service Effect in 1978 $
Cohort Year Statistic
1950   1981 Value                          -2195.3*
            Standard Error                   1069.5
       1982 Value                           -1679.0
            Standard Error                   1194.1
       1983 Value                           -1849.3
            Standard Error                   1240.7
       1984 Value                           -2517.1
            Standard Error                   1326.3
1951   1981 Value                          -2258.3*
            Standard Error                   1141.2
       1982 Value                           -1382.1
            Standard Error                   1247.5
       1983 Value                           -2174.4
            Standard Error                   1335.3
```

(continues on next page)

(continued from previous page)

```
       1984 Value                          -2644.3
            Standard Error                  1420.3
1952   1981 Value                          -2675.1
            Standard Error                  1500.6
       1982 Value                          -1638.2
            Standard Error                  1630.1
       1983 Value                          -3110.0
            Standard Error                  1761.9
       1984 Value                          -3340.9
            Standard Error                  1853.8
```

Looking at nonwhite males now, we observe what we already expected. All of the Wald estimates are actually far away from being statistically significant.

[15]:
```
p_value_star(table3["nonwhite"], slice(None), ("", "Service Effect in 1978 $"))
```

[15]:
```
First Level               Draft Eligibility Effects in Current $  \
Second Level                                          FICA Earnings
Cohort Year Statistic
1950   1981 Value                                            534.5
            Standard Error                                   413.6
       1982 Value                                            285.2
            Standard Error                                   461.3
       1983 Value                                             96.1
            Standard Error                                   512.6
       1984 Value                                            -76.9
            Standard Error                                   548.2
1951   1981 Value                                            313.2
            Standard Error                                   419.2
       1982 Value                                            175.5
            Standard Error                                   471.6
       1983 Value                                            419.6
            Standard Error                                   538.2
       1984 Value                                           -223.2
            Standard Error                                   562.9
1952   1981 Value                                           -305.9
            Standard Error                                   429.1
       1982 Value                                           -262.6
            Standard Error                                   476.8
       1983 Value                                           -177.3
            Standard Error                                   531.5
       1984 Value                                           -123.4
            Standard Error                                   568.6

First Level                                                      \
Second Level              Adjusted FICA Earnings Total W-2 Earnings
Cohort Year Statistic
1950   1981 Value                          654.0              802.6
            Standard Error                 495.2              524.6
       1982 Value                          335.4              326.0
            Standard Error                 529.8              609.0
       1983 Value                          169.1              315.5
            Standard Error                 551.6              720.0
```

(continues on next page)

(continued from previous page)

```
       1984 Value                            -65.1             -287.4
            Standard Error                   601.9              804.1
1951   1981 Value                            401.5              416.0
            Standard Error                   446.6              745.2
       1982 Value                            228.1             -244.3
            Standard Error                   524.4              647.8
       1983 Value                            398.9              254.3
            Standard Error                   558.8              767.6
       1984 Value                           -293.5             -718.6
            Standard Error                   598.1              771.6
1952   1981 Value                           -316.5             -272.4
            Standard Error                   454.8              492.9
       1982 Value                           -502.6             -160.2
            Standard Error                   524.1              590.0
       1983 Value                           -275.9              -53.6
            Standard Error                   546.6              643.5
       1984 Value                            -99.8             -288.1
            Standard Error                   600.3              721.0

First Level                                                        \
Second Level               P(V|eligible) - P(V|ineligible)
Cohort Year Statistic
1950   1981 Value                                    0.06
            Standard Error                          0.057
       1982 Value
            Standard Error
       1983 Value
            Standard Error
       1984 Value
            Standard Error
1951   1981 Value                                    0.05
            Standard Error                          0.06
       1982 Value
            Standard Error
       1983 Value
            Standard Error
       1984 Value
            Standard Error
1952   1981 Value                                   0.016
            Standard Error                          0.061
       1982 Value
            Standard Error
       1983 Value
            Standard Error
       1984 Value
            Standard Error

First Level
Second Level               Service Effect in 1978 $
Cohort Year Statistic
1950   1981 Value                             7780.5
            Standard Error                    5891.3
```

(continues on next page)

(continued from previous page)

```
       1982 Value                          3758.5
            Standard Error                 5937.0
       1983 Value                          1836.3
            Standard Error                 5990.4
       1984 Value                          -677.8
            Standard Error                 6269.8
1951   1981 Value                          5760.5
            Standard Error                 6407.4
       1982 Value                          3081.9
            Standard Error                 7087.0
       1983 Value                          5224.8
            Standard Error                 7318.6
       1984 Value                         -3687.0
            Standard Error                 7513.4
1952   1981 Value                        -14104.0
            Standard Error                20262.8
       1982 Value                        -21092.7
            Standard Error                21993.9
       1983 Value                        -11221.1
            Standard Error                22235.2
       1984 Value                         -3892.0
            Standard Error                23420.2
```

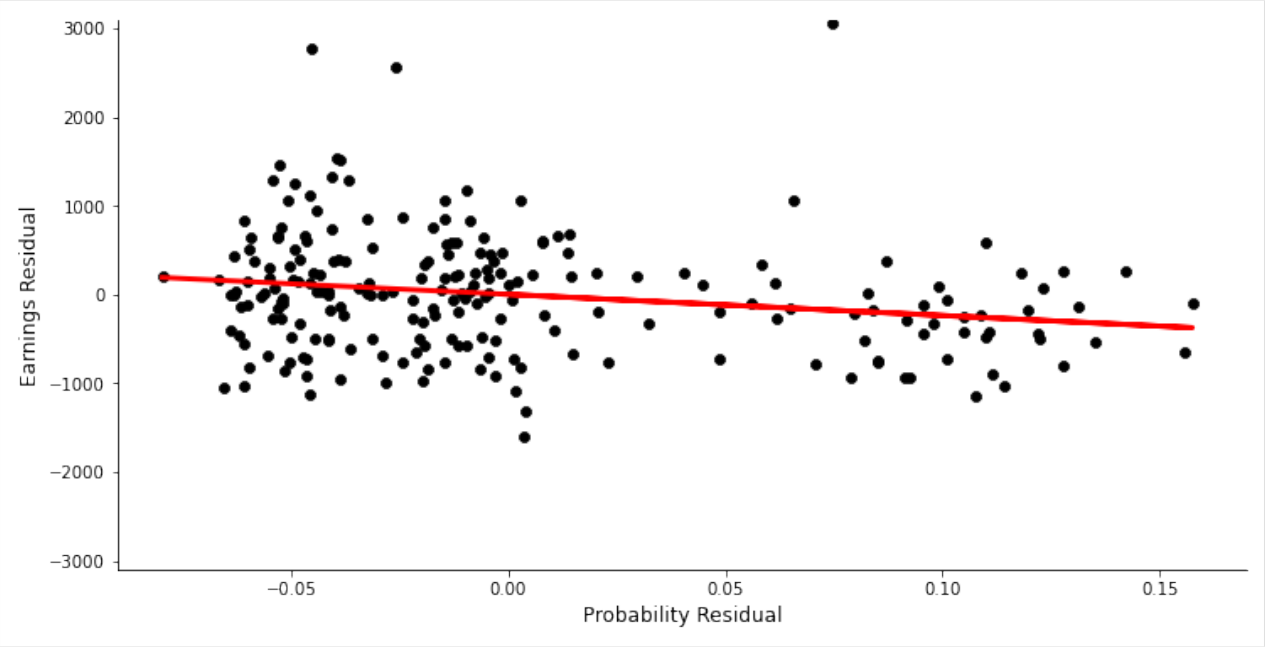
3.3.2 More complex IV estimates

In the next step Angrist uses a more generalized version of the Wald estimate for the given data. While in the previous analysis the mean earnings were compared solely on the basis of two groups (eligibles and ineligibles, which were determined by the lottery numbers), in the following this is extended to more complex subgroups. The grouping is now based on intervals of five consecutive lottery numbers. As explained in the section on idenficication this boils down to estimating the model described in equation (2).

$$\bar{y}_{ctj} = \beta_c + \delta_t + \hat{p}_{cj}\alpha + \bar{u}_{ctj}$$

$\bar{y}_{ctj}$ captures the mean earnings by cohort c, in year t for group j. $\hat{p}_{cj}$ depicts the estimated probability of being a veteran conditional on being in cohort c and group j. We are now interested in obtaining an estimate of α. In our current set up α corresponds to a linear combination of the many different possible Wald estimates when comparing each of the subgroups in pairs. With this view in mind Angrist restricts the treatment effect to be same (i.e. equal to α) for each comparison of subgroups. The above equation is equivalent to the second stage of the 2SLS estimation. Angrist estimates the above model using the mean real earnings averaged over the years 1981 to 84 and the cohorts from 1950 to 53. In the first stage Angrist has to estimate $\hat{p}_{cj}$ which is done again by using a combination of the SIPP sample and the matched DWDC/CWHS data set. With this at hand Angrist shows how the equation (2) looks like if it was estimated by OLS. The following **Figure 3** is also called Visual Instrumental Variables (VIV). In order to arrive there he takes the residuals from an OLS regression of $\bar{y}_{ctj}$ and $\hat{p}_{cj}$ on cohort and time dummies, respectively. Then he performs another OLS regression of the earnings residuals on the probability residuals. This is depicted in Figure 3 below. The slope of the regression line corresponds to an IV estimate of α. The slope amounts to an estimate of -2384 dollars which serves as a reference for the treatment effect measured by another, more efficient method described below the Figure.

[16]:
```
get_figure3(data_cwhsc_new)
```



We now shortly turn back to a remark from before. Angrist is forced to only work with sample means due to confidentiality restrictions on the underlying micro data. For the Wald estimates it is somewhat easily imaginable that this does not pose any problem. For the above estimation of α using 2SLS this is less obvious. Angrist argues, though, that there is a Generalized Method of Moments (GMM) interpretation of the 2SLS approach which allows him to work with sample moments alone. Another important implication thereof is that he is not restricted to using only one sample to obtain the sample moments. In our concrete case here, it is therefore unproblematic that the earnings data is coming from another sample than the conditional probabilities of being a veteran as both of the samples are drawn from the same population. This is a characteristic of the GMM.

In the following, Angrist estimates equation (2) by using the more efficient approach of Generalized Least Squares (GLS) as opposed to OLS. The GLS is more efficient if there is correlation between the residuals in a regression model. Angrist argues that this is the case in the above model equation and that this correlation can be estimated. GLS works such that coming from the estimated covariance matrix $\hat{\Omega}$ of the residuals, the regressors as well as the dependent variable are transformed using the upper triangle of the Cholesky decomposition of $\hat{\Omega}^{-1}$. Those transformed variables are then used to run a regular OLS model with nonrobust standard errors. The resulting estimate $\hat{\alpha}$ then is the most efficient one (if it is true that there is correlation between the residuals).

Angrist states that the optimal weigthing matrix Ω resulting in the most efficient estimate of $\hat{\alpha}$ looks the following:

$$\Omega = V(\bar{y}_{ctj}) + \alpha^2 V(\hat{p}_{cj}).$$

All of the three elements on the right hand side can be estimated from the data at hand.

Now we have all the ingredients to have a look at the results in **Table 4**. In practice, Angrist estimates two models in the above manner based on the general form of the above regression equation. Model 1 allows the treatment effect to vary by cohort while Model 2 collapses them into a scalar estimate of α. The results for white men in Model 1 show that for each of the three earnings measures as dependent variable only few are statistically significant to the five percent level (indicated by a star added by me again). A look at Model 2 reveals, though, that the combined treatment effect is significant and it amounts to a minus of 2000 dollar (we look again at real earnings in 1978 dollar terms) annualy for those having served in the army. For cohort 1953 we obtain insignificant estimates which was to be expected given that actually nobody was drafted in that year.

Note: The results are again a bit different to those in the paper. The same is the case, though, in the replication code my replication is building up on.

[17]:
```
table4 = get_table4(data_cwhsc_new)
p_value_star(
    table4["white"], (slice(None), slice(None), ["Value", "Standard Error"]),␣
↪(slice(None)),
)
```

[17]:

Model	Cohort	Statistic	FICA Taxable Earnings
Model 1	1950	Value	-1709.2
		Standard Error	946.8
	1951	Value	-1457.1
		Standard Error	954.7
	1952	Value	-1724.0
		Standard Error	863.3
	1953	Value	1223.8
		Standard Error	3232.5*
	Chi Squared		578.3
Model 2	1950-53	Value	-1562.9
		Standard Error	521.7
	Chi Squared		579.1

Model	Cohort	Statistic	Adjusted FICA Earnings
Model 1	1950	Value	-2093.7
		Standard Error	1109.2
	1951	Value	-1983.7
		Standard Error	1036.5
	1952	Value	-1943.0*
		Standard Error	927.5
	1953	Value	900.7
		Standard Error	3506.6*
	Chi Squared		630.3
Model 2	1950-53	Value	-1920.4
		Standard Error	576.8
	Chi Squared		631.0

Model	Cohort	Statistic	Total W-2 Compensation
Model 1	1950	Value	-1895.0
		Standard Error	1336.9
	1951	Value	-2431.4
		Standard Error	1155.4
	1952	Value	-2058.7*
		Standard Error	1004.8
	1953	Value	-488.6
		Standard Error	3947.4*
	Chi Squared		569.5
Model 2	1950-53	Value	-2094.5
		Standard Error	649.1
	Chi Squared		569.7

Angrist also reports those estimates for nonwhite men which are not significant. This was already expected as the the instrument was not clearly correlated with the endogenous variable of veteran status.

[18]:
```
p_value_star(
    table4["nonwhite"], (slice(None), slice(None), ["Value", "Standard Error"]),␣
↪(slice(None)),
)
```

[18]:

Model	Cohort	Statistic	FICA Taxable Earnings	Adjusted FICA Earnings	Total W-2 Compensation
Model 1	1950	Value	3893.7*	3871.9*	5711.8*
		Standard Error	5355.1	6246.9	7206.5
	1951	Value	-891.3	-333.4	2609.0
		Standard Error	4399.6	4667.1	4887.1
	1952	Value	-3182.9	-3457.7	-3068.0
		Standard Error	3994.9	4194.9	4222.7
	1953	Value	-5928.3	-8571.5	-6325.8
		Standard Error	10302.3*	10652.6*	11393.0*
	Chi Squared		616.7	681.7	693.6
Model 2	1950-53	Value	-643.3	-999.7	367.8
		Standard Error	2406.8	2602.6	2733.8
	Chi Squared		618.4	683.4	695.6

This table concludes the replication of the core results of the paper. Summing up, Angrist constructed a causal graph for which he employs a plausible estimation strategy. Using his approach he concludes with the main result of having found a negative effect of serving in the military during the Vietnam era on subsequent earnings for white male in the United States.

Angrist provides some interpretation of the found effect and some concerns that might arise when reading his paper. I will discuss some of his points in the following critical assessment.

4. Critical Assessment

Considering the time back then and the consequently different state of research, the paper was a major contribution to instrumental variable estimation of treatment effects. More broadly, the paper is very conclusive and well written. Angrist discusses caveats quite thoroughly which makes the whole argumentation at first glance very concise. Methodologically, the paper is quite complex as due to the kind of data available. Angrist is quite innovative in that regard as he comes up with the two sample IV method in this paper which allows him to pratically follow his identification strategy. The attempt to explain the mechanisms behind the negative treatment effect found by him makes the paper comprehensive and shows the great sense of detail Angrist put into this paper.

While keeping in mind the positive sides of his paper, in hindsight, Angrist is a bit too vocal about the relevance and accuracy of his findings. Given our knowledge about the local average treatment effect (**LATE**) we encountered in our lecture, Angrist only identifies the average treatment effect of the compliers (those that enroll for the army if they are draft-eligible but do not if they are not) if there is individual level treatment heterogeneity and if the causal graph from before is accurate. Hence, the interpretation of the results gives only limited policy implications. For the discussion of veteran compensation the group of those who were induced by the lottery to join the military are not crucial. As there is no draft lottery anymore, what we are interested in is how to compensate veterans for their service who "voluntarily" decided to serve in the military. This question cannot be answered by Angrist's approach given the realistic assumption that there is treatment effect heterogeneity (which also Angrist argues might be warranted).

A related difficulty of interpretation arises because in the second part, Angrist uses an overidentified model. As already discussed before this amounts to a linear combination of the average treatment effects of subgroups. This mixes the LATEs of several subgroups making the policy implications even more blurred as it is not clear what the individual contributions of the different subgroups are. In this example here this might not make a big difference but should be kept in mind when using entirely different instrumental variables to identify the LATE.

In a last step, there are several possible scenarios to argue why **the given causal graph might be violated**. Angrist himself delivers one of them. After the lottery numbers were drawn, there was some time in between the drawing and the announcement of the draft-eligibility ceiling. This provoked behavioral responses of some individuals with low numbers to volunteer for the army in order to get better terms of service as well as enrolling in university which rendered them ineligible for the army. In our data, it is unobservable to see the fraction of individuals in each group to join university. If there was actually some avoidance behavior for those with low lottery numbers, then the instrument would be questionable as there would be a path from the Draft Lottery to unobservables (University) which affects earnings. At the same time there is also clearly a relation between University and Military Service.

Rosenzweig and Wolpin (2000) provide a causal graph that draws the general interpretability of the results in Angrist (1990) further into question. Let us look at the causal graph below now imagining that there was no directed graph from Draft Lottery to Civilian Experience. Their argument is that Military Service reduces Schooling and Civilian Experience which lowers Wages while affecting Wages directly and increasing them indirectly by reducing Schooling and increasing work experience. Those subtle mechanism are all collapsed into one measure by Angrist which gives an only insufficiently shallow answer to potentially more complex policy questions. Building up on this causal graph, Heckman (1997) challenges the validity of the instrument in general by making the point that there might be a directed graph from Draft Lottery to Civilian Experience. The argument goes as follows: Employers, after learning about their employees' lottery numbers, decrease the training on the job for those with a high risk of being drafted. If this is actually warranted the instrument Draft Lottery cannot produce unbiased estimates anymore.

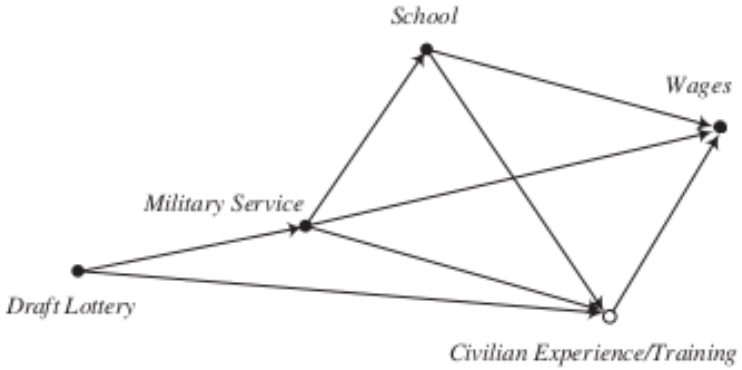


Figure 10.2 A directed graph for compliers with the Vietnam draft lottery as an IV for military service.

Morgan and Winship (2014) add to this that the bias introduced by this is further affected by how strongly Draft Lottery affects Military Service. Given the factor that the lottery alone does not determine military service but that there are tests, might cause the instrument to be rather weak and therefore a potential bias to be rather strong.

5. Extensions

5.1 Treatment effect with different years of earning

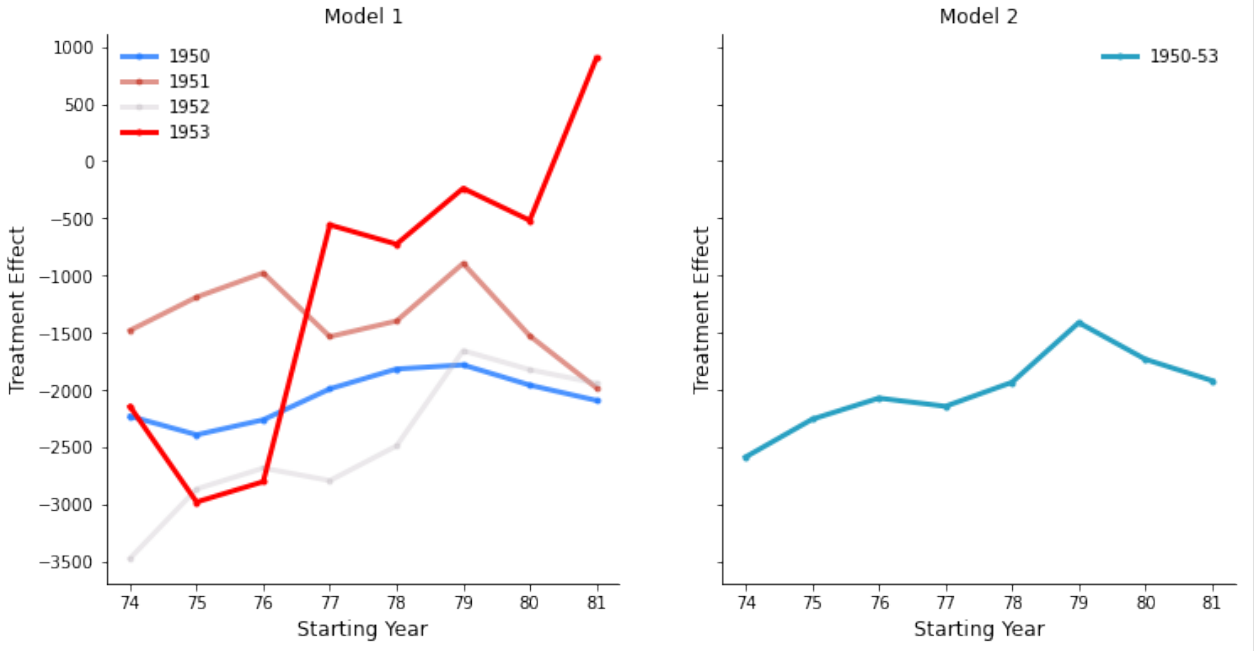
In the calculation of the average treatment in Table 4 Angrist chooses to calculate it for earnings in the years from 1981 to 84. While he plausibly argues that this most likely constitutes a long term effect (as those are the last years for which he has data) in comparison to earlier years, it does not give a complete picture. Looking at Table 1 again we can see that for the earnings differences in the years 81 to 84 quite big estimates are calculated. Assuming that the difference in probability of serving given eligibility versus noneligibility stays somewhat stable across the years, we would expect some heterogeneity in average treatment effects depending on which years we use the earnings data of. Angrist, though, does not investigate this although he has the data for it at hand. For example from a policy perspective one could easily argue that a look at the average treatment effect for earlier years (close to the years in which treatment happens) might be more relevant than the one for years after. This is because given the long time between the actual service and the earnings data of 1981 to 84 it is likely that second round effects are driving some of the results. These might be initially caused by veteran status but for later years the effect of veteran status might mainly act by means of other variables. For instance veterans after the war might be forced to take simple jobs due to their lack of work experience and from then on their path is determined by the lower quality of the job that they had to take right after war. For policy makers it might be of interest to see what happens to veterans right after service to see what needs to be done in order to stop second round effects from happening in the first place.

To give a more wholesome image, I estimate the results for Table 4 for different years of earnings of white men. As mentioned before the quality of the Total W-2 data set is rather low and the adjusted FICA is more plausible than the FICA data. This is why I only use the adjusted FICA data in the following. For the adjusted FICA I have data for Table 4 for the years from 1974 to 1984. For each possible four year range within those ten years I estimate Model 1 and 2 from Table 4 again.

Below I plot the average treatment effects obtained. On the x-axis I present the starting year of the range of the adjusted FICA data used. For starting value 74 it means that the average treatment effect is calculated for earnings data of the years 1974 to 77. The results at the starting year 81 are equivalent to the ones found by Angrist in Table 4 for white men.

[19]:
```
# get the average treatment effects of Model 1 and 2 with adjusted FICA earnings for
# several different ranges of four years
results_model1 = np.empty((8, 4))
results_model2 = np.array([])
for number, start_year in enumerate(np.arange(74, 82)):
    years = np.arange(start_year, start_year + 4)
    flex_table4 = get_flexible_table4(data_cwhsc_new, years, ["ADJ"], [50, 51, 52, 53])
    results_model1[number, :] = (
        flex_table4["white"].loc[("Model 1", slice(None), "Value"), :].values.flatten()
    )
    results_model2 = np.append(
        results_model2, flex_table4["white"].loc[("Model 2", slice(None), "Value"), :].
↪values,
    )
```

[20]:
```
# Plot the effects for white men in Model 1 and 2
# (colors apart from Cohort 1950 are random, execute again to
# change them)
get_figure1_extension1(results_model1, results_model2)
```



The pattern is more complex than what we can see in the glimpse of Table 4 in the paper. We can see that there is quite some heterogeneity in average treatment effects across cohorts when looking at the data for early years. This changes when using data of later years. Further the fact of being a veteran does seem to play a role for the cohort 1953 right after the war but the treatment effect becomes insignificant when looking at later years. This is interesting as the cohort of 1953 was the one for which no one was drafted (remember that in 1973 no one was drafted as the last call was in December 1972).

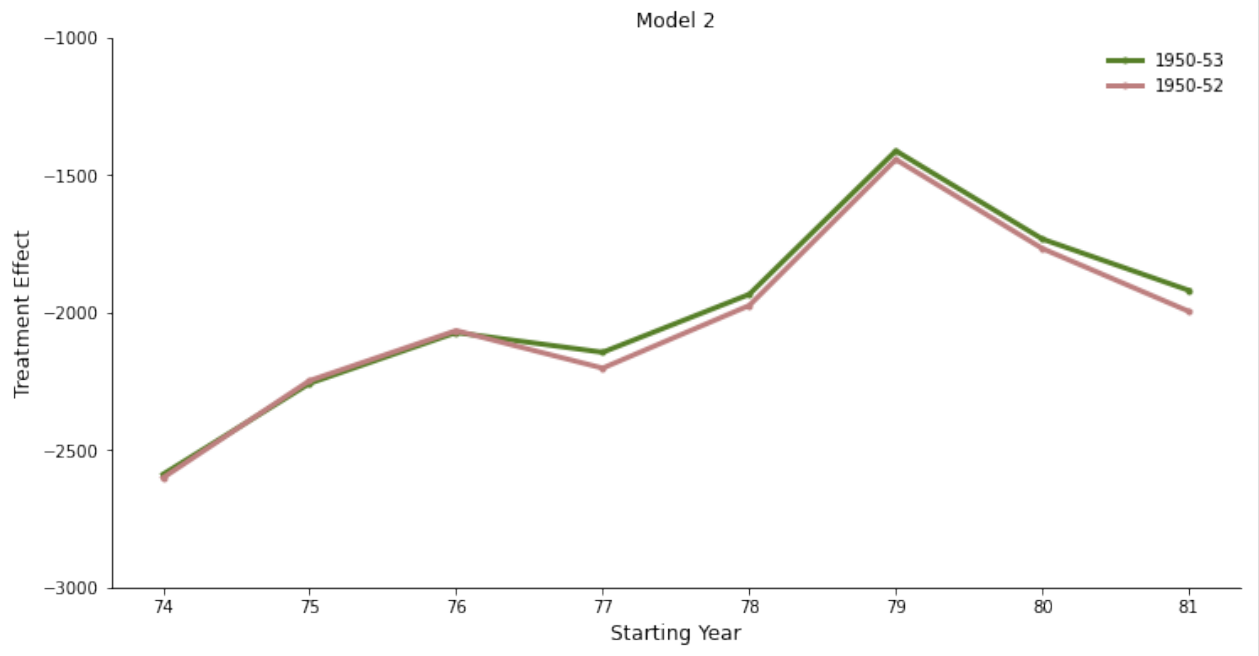
Another observation is linked to the fact that draft eligibility does not matter for those born in 1953. These people appear to have voluntarily joined the army as no one of them could have possibly been drafted. This cannot be said for the cohorts before. Employers can only observe whether a person is a veteran and when they are born (and not if they are compliers or not). A theory could be that employers act on the loss of experience for initial wage setting for every army veteran right after the war. The fact that the cohort of 1953 could only be volunteers but not draftees could

give them a boost in social status to catch up again in the long run, though. This mechanism might explain to a certain extent why we observe the upward sloping line for the cohort of 1953 (but not for the other groups).

As discussed in the critical assessment, we actually only capture the local average treatment effect of the compliers. Those are the ones who join the army when they are draft-eligible but do not when they are not. The identifying assumption for the LATE requires that everyone is a complier. This is probably not warranted for the cohort of 1953. In that year it is easily imaginable that there are both defiers and compliers which means that we do not capture the LATE for cohort 1953 in Model 1 and for cohort 1950-53 in Model 2 but something else we do not really know how to interpret. This might be another reason why we observe this peculiar pattern for the cohort of 1953. Following up on this remark I estimate the Model 2 again excluding the cohort of 1953 to focus on the cohorts for which the assumptions for LATE are likely to hold.

[21]:
```
results_model2_53 = np.array([])
for number, start_year in enumerate(np.arange(74, 82)):
    years = np.arange(start_year, start_year + 4)
    flex_table4 = get_flexible_table4(data_cwhsc_new, years, ["ADJ"], [50, 51, 52])
    results_model2_53 = np.append(
        results_model2_53, flex_table4["white"].loc[("Model 2", slice(None), "Value"), :
↪].values,
    )
```

[22]:
```
get_figure2_extension1(results_model2, results_model2_53)
```



We can see that for later years the treatment effect is a bit lower when excluding the cohort of 1953. It confirms the findings of Angrist with the advantage of making it possible to attach a clearer interpretation to it.

Following the above path, it would also be interesting to vary the amount of instruments used by more than just the two ways Angrist has shown. It would be interesting to break down the interval size of lottery numbers further. Unfortunately I could no find a way to do that with the already pre-processed data I have at hand.

5.2 Bias Quantification

In the critical assessment I argued that the simple Wald estimate might be biased because employers know their employees' birth date and hence their draft eligibility. The argument was that employers invest less into the human capital of those that might be drafted. This would cause the instrument of draft eligibility to not be valid and hence suffer from bias. This bias can be calculated in the following way for a binary instrument:

$$\frac{E[Y|Z=1]-E[Y|Z=0]}{E[D|Z=1]-E[D|Z=0]} = \delta + \frac{E[\epsilon|Z=1]-E[\epsilon|Z=0]}{E[D|Z=1]-E[D|Z=0]}$$

What has been done in the last column of Table 3 (the Wald estimate) is that Angrist calculated the left hand side of this equation. This calculation yields an unbiased estimate of the treatment effect of D (veteran status) on Y (earnings) δ if there is no effect of the instrument Z (draft eligibility) on Y through means of unobservables ϵ. In our argumentation this assumption does not hold which means that $E[\epsilon|Z=1]-E[\epsilon|Z=0]$ is not equal to zero as draft eligibility affects Y by the behavioral change of employers to make investing into human capital dependent on draft eligibility. Therefore the left hand side calculation is not equal to the true treatment effect δ but has to be adjusted by the bias $\frac{E[\epsilon|Z=1]-E[\epsilon|Z=0]}{E[D|Z=1]-E[D|Z=0]}$.

In this section I run a thought experiment in which I quantify this bias. The argumentation here is rather heuristic because I lack the resources to really find a robust estimate of the bias but it gives a rough idea of whether the bias might matter economically. My idea is the following. In order to get a measure of $E[\epsilon|Z=1]-E[\epsilon|Z=0]$ I have a look at estimates for the effect of work experience on earnings. Remember that the expected difference in earnings due to a difference in draft eligibility is caused by a loss in human capital for those draft eligible because they might miss out on on-the-job-training. This loss in on-the-job-training could be approximated by a general loss in working experience. For an estimate of that effect I rely on Keane and Wolpin (1997) who work with a sample for young men between 14 and 21 years old from the year 1979. The effect of working experience on real earnings could be at least not far off of the possible effect in our sample of adjusted FICA real earnings of 19 year old men for the years 1981 to 1984. Remember that lottery participants find out about whether they are draft eligible or not at the end of the year before they might be drafted. I assume that draft dates are spread evenly over the draft year. One could then argue that on average a draft eligible person stays in his job for another half a year after having found out about the eligibility and before being drafted. Hence, for on average half a year an employer might invest less into the human capital of this draft eligible man. I assume now that employers show a quite moderate behavioral response. During the six months of time, the employees only receive a five month equivalent of human capital gain (or work experience gain) as opposed to the six months they stay in the company. This means they loose one month of work experience on average in comparison to those that are not draft eligible.

To quantify this one month loss of work experience I take estimates from Keane and Wolpin (1997). For blue collar workers they roughly estimate the gain in real earnings in percent from an increase in a year of blue collar work experience to be 4.6 percent (actually their found effect depends on the years of work experience but I simplify this for my rough calculations). For white collar workers the equivalent estimate amounts to roughly 2.7 percent. I now take those as upper and lower bounds, calculate their one month counterparts and quantify the bias in the Wald estimates of the last column of Table 3. The bias $\frac{E[\epsilon|Z=1]-E[\epsilon|Z=0]}{E[D|Z=1]-E[D|Z=0]}$ is then roughly equal to the loss in annual real earnings due to one month less of work experience divided by the difference in probability of being a veteran conditional on draft eligibility.

The first table below depicts how the bias changes by cohort across the different years of real earnings with increasing estimates of how a loss in experience affects real earnings. Clearly with increasing estimates of how strong work experience contributes to real earnings, the bias gets stronger. This is logical as it is equivalent to an absolute increase in the nominator. Above that the bias is stronger for later years of earnings as the real earnings increase by year. Further the slope is steeper for later cohorts as the denominator is smaller for later cohorts. Given the still moderate assumption of a loss of one month of work experience we can see that the bias does not seem to be negligible economically especially when taking the blue collar percentage estimate.

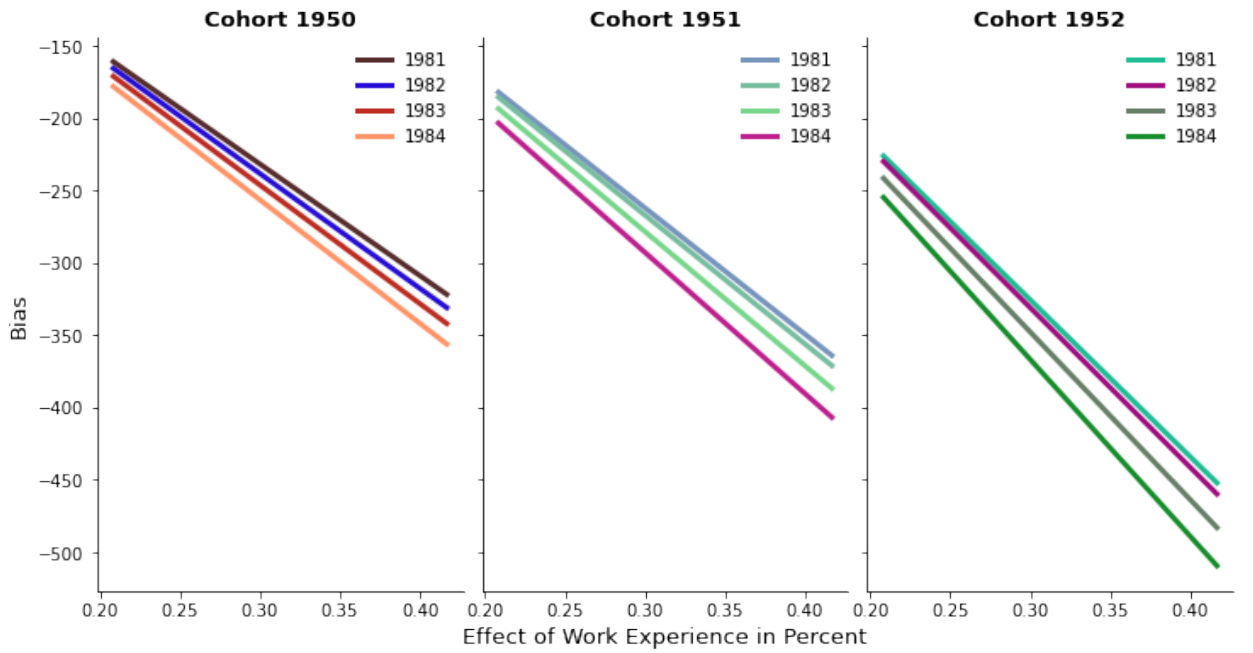
[23]:
```
# Calculate the bias, the true delta and the orginal Wald estimate for a
# ceratain interval of working experience effect
```

(continues on next page)

(continued from previous page)

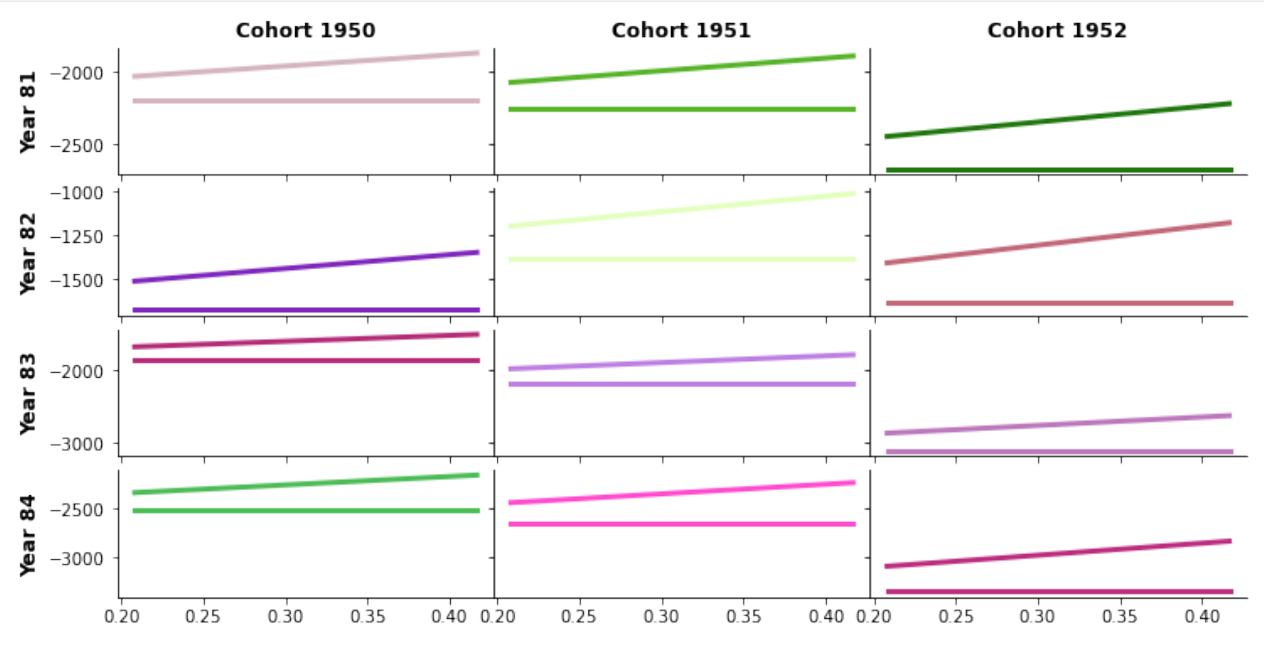
```
interval = np.linspace(0.025, 0.05, 50) / 12
bias, true_delta, wald = get_bias(
    data_cwhsa, data_cwhsb, data_dmdc, data_sipp, data_cwhsc_new, interval
)
```

[24]:
```
# plot the bias by cohort
get_figure1_extension2(bias, interval)
```



To get a sense of how the size of the bias relates to the size of the previously estimated Wald coefficients, let us have look at the figure below. It shows for each cell consisting of a cohort and year combination, the Wald estimate from Table 3 as the horizontal line and the true δ depending on the weight of the loss in work experience as the upward sloping line. Given that our initial estimates of the Wald coefficients are in a range of only a few thousands, an estimated bias of roughly between 200 and 500 dollars cannot be characterized as incosiderable. Further given Angrist's policy question concerning Veteran compensation, even an estimate that is higher by 200 dollars makes a big difference when it is about compensating thousands of veterans.

[25]:
```
# plot the the true delta (accounted for the bias) compared to the original Wald estimate
get_figure2_extension2(true_delta, wald, interval)
```



6. Conclusion

Regarding the overall quality and structure of Angrist (1990), reading it is a real treat. The controversy after its publication and the fact that it is highly cited clearly show how important its contribution was and still is. It is a great piece of discussion when it comes to the interpretability and policy relevance of instrumental variable approaches. As already reiterated in the critical assessment, one has to acknowledge the care Angrist put into this work. Although his results do not seem to prove reliable, it opened a whole discussion on how to use instrumental variables to get the most out of them. Another contribution that should not go unnoticed is that Angrist shows that instruments can be used even though they might not come from the same sample as the dependent and the endogenous variable. Practically, this is very useful as it widens possible areas of application for instrumental variables.

Overall, it has to be stated that the paper has some shortcomings but the care put into this paper and the good readibility allowed other researchers (and Angrist himself) to swoop in giving helpful remarks that improved the understanding of instrumental variable approaches for treatment effect evaluation.

References

Angrist, J. (1990). Lifetime Earnings and the Vietnam Era Draft Lottery: Evidence from Social Security Administrative Records. *American Economic Review*. 80. 313-36.

Angrist, J. D., & Pischke, J.-S. (2009). Mostly harmless econometrics: An empiricist's companion.

Heckman, J. (1997). Instrumental Variables: A Study of Implicit Behavioral Assumptions Used in Making Program Evaluations. *The Journal of Human Resources*, 32(3), 441-462. doi:10.2307/146178

Keane, M., & Wolpin, K. (1997). The Career Decisions of Young Men. *Journal of Political Economy*, 105(3), 473-522. doi:10.1086/262080

Morgan, S., and Winship, C. (2014). Counterfactuals and Causal Inference: Methods and Principles for Social Research (Analytical Methods for Social Research). Cambridge: Cambridge University Press. doi:10.1017/CBO9781107587991

Rosenzweig, M. R. and Wolpin, K. I.. (2000). "Natural 'Natural Experiments' in Economics." *Journal of Economic Literature* 38:827–74.

Wald, A. (1940). The Fitting of Straight Lines if Both Variables are Subject to Error. *Ann. Math. Statist.* 11 , no. 3, 284–300.

Appendix

Key Variables in the Data Sets

data_cwhsa

Name	Description
index	
byr	birth year
race	ethnicity, 1 for white and 2 for nonwhite
interval	interval of draft lottery numbers, 73 intervals with the size of five consecutive numbers
year	year for which earnings are collected
variables	
vmn1	nominal earnings
vfin1	fraction of people with zero earnings
vnu1	sample size
vsd1	standard deviation of earnings

data_cwhsb

Name	Description
index	
byr	birth year
race	ethnicity, 1 for white and 2 for nonwhite
interval	interval of draft lottery numbers, 73 intervals with the size of five consecutive numbers
year	year for which earnings are collected
type	source of the earnings data, "TAXAB" for FICA and "TOTAL" for Total W-2
variables	
vmn1	nominal earnings
vfin1	fraction of people with zero earnings
vnu1	sample size
vsd1	standard deviation of earnings

data_cwhsc_new

Name	Description
index	
byr	birth year
race	ethnicity, 1 for white and 2 for nonwhite
interval	interval of draft lottery numbers, 73 intervals with the size of five consecutive numbers
year	year for which earnings are collected
type	source of the earnings data, “ADJ” for adjusted FICA, “TAXAB” for FICA and “TOTAL” for Total W-2
variables	
earnings	real earnings in 1978 dollars
nj	sample size
nj0	number of persons in the sample with zero earnings
iweight_old	weight for weighted least squares
ps_r	fraction of people having served in the army
ern74 to ern84	unweighted covariance matrix of the real earnings

data_dmdc

Name	Description
index	
byr	birth year
race	ethnicity, 1 for white and 2 for nonwhite
interval	interval of draft lottery numbers, 73 intervals with the size of five consecutive numbers
variables	
nsrvd	number of people having served
ps_r	fraction of people having served

data_sipp (this is the only micro data set)

Name	Description
index	
u_brthyr	birth year
nrace	ethnicity, 0 for white and 1 for nonwhite
variables	
nvstat	0 if man is not a veteran, 1 if he is
fnlwgt_5	fraction of people with this index among overall sample
rsncode	1 if person was draft eligible, else if not

Lindo et al. (2010)

Lindo et al. (2010) examine the effects of academic probation on student outcomes using a regression discontinuity design. The analysis is based on data from a large Canadian university and evaluates whether academic probation is successful in improving the performance of low scoring students. Consistent with a model of performance standards, the authors find that being placed on probation in the first year of university induces some students to drop out of school while it improves the grades of students who continue their studies. In a more general sense, academic probation can offer insights into how agents respond to negative incentives and the threat of punishment in a real-world context.

Project by Annica Gehlen

Replication of Lindo et al. (2010): Ability, gender, and performance standards: Evidence from academic probation

Project by Annica Gehlen, Summer 2019

This notebook contains my replication of the results from the following paper:

> Lindo, J. M., Sanders, N. J., & Oreopoulos, P. (2010). Ability, gender, and performance standards: Evidence from academic probation. American Economic Journal: Applied Economics, 2(2), 95-117.

Downloading and viewing this notebook:

- The best way to view this notebook is by downloading it and the repository it is located in from GitHub. Other viewing options like *MyBinder* or *NBViewer* may have issues with displaying images or coloring of certain parts (missing images can be viewed in the folder files on GitHub).
- The original paper, as well as the data and code provided by the authors can be accessed here.

Information about replication and individual contributions:

- For the replication, I try to remain true to the original structure of the paper so readers can easily follow along and compare. All tables and figures are named and labeled as they appear in Lindo et al. (2010).
- The tables in my replication appear transposed compared to the original tables to suit my workflow in Python.
- For transparency, all sections in the replication that constitute independent contributions by me and are not part of results presented (or include deviations from the methods used) in the paper are marked as *extensions*.

Table of Contents

[1]:
```
import pandas as pd
import pandas.io.formats.style
```

[2]:
```
from auxiliary.example_project_auxiliary_predictions import (
    prepare_data,
    calculate_bin_frequency,
    create_predictions,
    create_bin_frequency_predictions,
    bootstrap_predictions,
    create_fig3_predictions,
    get_confidence_interval,
    bandwidth_sensitivity_summary,
    trim_data,
    create_groups_dict
)
from auxiliary.example_project_auxiliary_plots import (
    plot_hist_GPA,
    plot_covariates,
    plot_figure1,
    plot_figure2,
    plot_figure3,
    plot_figure4,
    plot_figure5,
    plot_figure4_with_CI,
    plot_figure_credits_year2,
    plot_nextCGPA
)
from auxiliary.example_project_auxiliary_tables import (
    describe_covariates_at_cutoff,
    create_table6,
    create_table1,
    estimate_RDD_multiple_datasets,
    estimate_RDD_multiple_outcomes,
```

(continues on next page)

(continued from previous page)

```
    color_pvalues
)
```

1. Introduction

Lindo et al. (2010) examine the effects of academic probation on student outcomes using data from Canada. Academic probation is a university policy that aims to improve the performance of the lowest- scoring students. If a student's Grade Point Average (GPA) drops below a certain threshold, the student is placed on academic probation. The probation status serves as a warning and does not entail immediate consequences, however, if students fail to improve their grades during the following term, they face the threat of being suspended from the university. In a more general sense, academic probation may offer insights into how agents respond to negative incentives and the threat of punishment in a real-world context with high stakes.

To estimate the causal impact of being placed on probation, Lindo et al. (2010) apply a **regression discontinuity design (RDD)** to data retrieved from three campuses at a large Canadian university. The RDD is motivated by the idea that the students who score just above the threshold for being put on academic probation provide a good counterfactual to the 'treatment group' that scores just below the threshold. In line with the performance standard model that serves as the theoretical framework for the paper, Lindo et al. (2010) find that being placed on probation induces students to drop out but increases the grades of the students who remain in school. The authors also find large heterogeneities in the way different groups of students react to academic probation.

Main variables

Treatment	Main outcomes	Main Covariates
Academic probation	Drop-out rates	Gender
.	Subsequent performance	HS grades
.	Graduation rates	Native language

In this notebook, I replicate the results presented in the paper by Lindo et al. (2010). Furthermore, I discuss in detail the identification strategy used by the authors and evaluate the results using multiple robustness checks. My analysis offers general support for the findings of Lindo et al. (2010) and points out some factors which may enable a deeper understanding of the causal relationship explored in the paper.

This notebook is structured as follows. In the next section, I present the performance standard model that lays down the theoretical framework for the paper (Section 2). In Section 3, I analyze the identification strategy that Lindo et al. (2010) use to unravel the causal effects of academic probation on student outcomes and Section 4 briefly discusses the empirical strategy the authors use for estimation. Section 5 and Section 6 constitute the core of this notebook. Section 5 shows my replication of the results in the paper and discussion thereof. In Section 6 I conduct various robustness checks and discuss some limitations of the paper. Section 7 offers some concluding remarks.

2. Theoretical Background

The underlying framework used for the analysis is a model developed by Bénabou and Tirole (2000) which models agent's responses to a performance standard. While Bénabou and Tirole (2000) model a game between a principal and an agent, Lindo et al. (2010) focus only on the agent to relate the model to the example of academic probation.

In the performance standard model, the agents face a choice between three options:

1. **Option 1**: Incurs cost c_1 and grants benefit V_1 if successful.
2. **Option 2**: Incurs cost c_2 and grants benefit V_2 if successful.
3. **Neither** option: Incurs 0 cost and 0 benefit.

Option 1 has a lower cost and a lower benefit than option 2 such that:

$$0 < c_1 < c_2, 0 < V_1 < V_2.$$

Ability, denoted by θ, translates to the probability of successfully completing either option. Assuming agents have perfect information about their ability, they solve the maximizing problem

$$max\{0, \theta V_1 - c_1, \theta V_2 - c_2\}. \tag{1.1}$$

Let $\underline{\theta}$ be the ability level where the agent is indifferent between neither and option two and let $\bar{\theta}$ be the ability level at which the agent is indifferent between option 1 and option 2. Assuming that

$$\underline{\theta} \equiv \frac{c_1}{V_1} < \bar{\theta} \equiv \frac{c_2 - c_1}{V_2 - V1} < 1 \tag{1.2}$$

ensures that both options are optimal for at least some θ.

It can be shown that:

- the lowest ability types ($\theta < \underline{\theta}$) choose neither option,
- the highest ability types ($\bar{\theta} < \theta$) choose the difficult option,
- the individuals in between the high and low type ($\underline{\theta} < \theta < \bar{\theta}$) choose the easier option.

If the principal now removes option 1 or makes choosing this option much more costly, then the agent will choose option 2 if and only if

$$\theta \geq \frac{c_2}{V_2} \equiv \theta^* \tag{1.3}$$

and choose neither option otherwise. The agents who would have chosen option 1 now split according to ability. Agents with high ability (specifically those with $\theta \in [\theta^*, \bar{\theta}]$) work harder, thereby choosing option 2, while low ability types (those with $\theta \in [\underline{\theta}, \theta^*]$) do not pursue option 2 (and thus choose neither option).

In the context of academic probation students face a similar decision and possible courses of action. Students whose GPA is just above the probation cutoff face the full set of options for the next year:

1. **Option 1**: Return to school and exhibit low effort and leading to a low GPA
2. **Option 2**: Return to school and exhibit high effort with the intent of achieving a high GPA
3. **Neither** option: Drop out of university

Students who score below the probation cutoff face a restricted set of options as the university administration essentially eliminates option 1 by suspending students if they do not improve their grades.

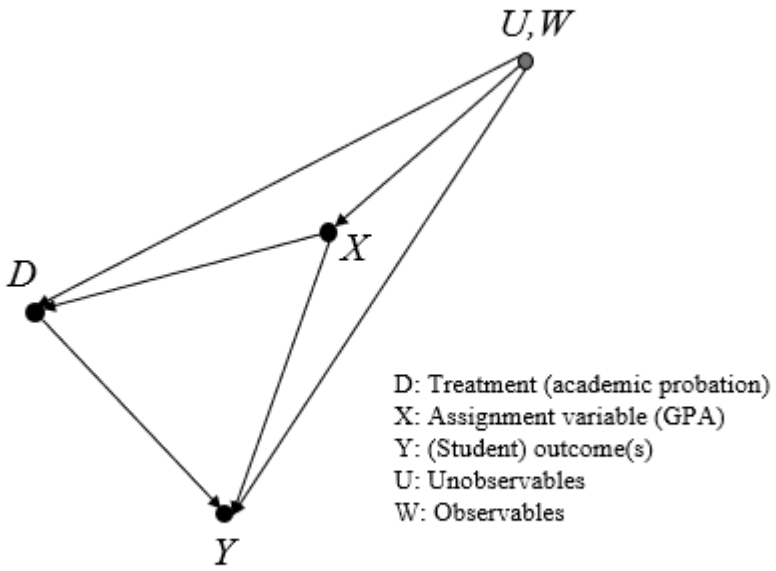
Lindo et al. (2010) formulate three testable implications of this theoretical framework:

- *Forbidding option 1 will***increase the overall probability of students dropping out**.
- *Forbidding option 1 will***increase the performance of those who return**.
- *Forbidding option 1 will cause***relatively low-ability students to drop out***and***relatively high-ability students to return and work harder**.

3. Identification

Lindo et al. (2010) in their paper aim to evaluate how academic probation affects students, specifically their probability of dropping out of university and whether it motivates those who remain to improve their grades. Students are placed on probation if their Grade Point Average (GPA) drops below a certain threshold and face the threat of suspension if they fail to improve their GPA in the next term. Students are thus clearly separated into a treated group (who is put on probation) and an untreated group based on their GPA.

The causal graph below illustrates the relationship between the assignment variable X, treatment D and outcome Y. While X (the GPA) directly assigns students to treatment, it may also be linked to student outcomes. Additionally, there may be observables W and unobservables U also affecting X,D, and Y. There are thus multiple backdoor paths that need to be closed in order to isolate the effect of academic probation. Simply controlling for the variables in question, in this case, does not suffice since there are unobservables that we cannot condition on. A randomized experiment, on the other hand, could eliminate selection bias in treatment by randomly assigning probation to students. The research question evaluated in the paper constitutes a classic policy evaluation problem in economics where we try to understand the causal implications of a policy without being able to observe the counterfactual world where the policy is not administered. However, as with many questions in economics, implementing a randomize experiment directly is not a feasible option, especially since we are examing the effect of a penalty whose consequences may affect students for the rest of their lives.



Since it is not possible to randomize assignment to treatment, another method is needed to isolate the effects of academic probation on student outcomes. Lindo et al. (2010) apply a regression discontinuity design (RDD) to the problem at hand, a method pioneered by Thistlethwaite and Campbell (1960) in their analysis of the effects of scholarships on student outcomes. In fact, the identification problem in Lindo et al. (2010) is quite similar to that of Thistlethwaite and Campbell (1960) as both papers evaluate the causal effects of an academic policy on student outcomes. However, while the scholarship administered to high performing students in Thistlethwaite and Campbell (1960) constitutes a positive reinforcement for these students, Lindo et al. (2010) examine the effects of a negative reinforcement or penalty on low performing students. This means that, in contrast to Thistlethwaite and Campbell (1960) and many other applications

of RD, our treatment group lies *below* the cutoff and not above it. This does not change the causal inference of this model but it might be confusing to readers familiar with RD designs and should thus be kept in mind.

The regression discontinuity design relies on the assumption of local randomization, i.e. the idea that students who score just above the cutoff do not systematically differ from those who score below the cutoff and thus pose an appropriate control group for the students who are placed on probation. This identification strategy relies on the assumption that students are unable to precisely manipulate their grades to score just above or below the probation threshold. Within the neighborhood around the discontinuity threshold, the RDD thus in a sense mimics a randomized experiment.

To explain how the use of regression discontinuity allows Lindo et al. (2010) to identify treatment effects, I draw on material provided in Lee and Lemieux (2010) and their discussion on the RDD in the potential outcomes framework. As mentioned above, for each student i we can image a potential outcome where they are placed on probation $Y_i(1)$ and where they are not $Y_i(0)$ but we can never simultaneously observe both outcomes for each student. Since it is impossible to observe treatment effects at the individual level, researchers thus estimate average effects using treatment and control groups.

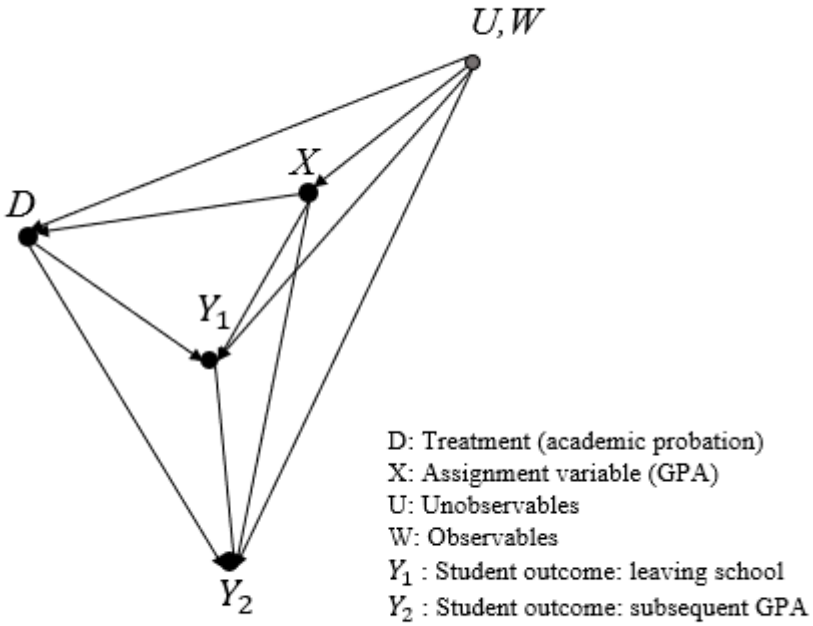
For the RDD this potential outcomes framework translates by imagining there are two underlying relationships between the average student outcome and the assignment variable X (the students' GPA), which are represented by $E[Y_i(1)|X]$ and $E[Y_i(0)|X]$. Since all students who score below the cutoff c are placed on probation, we only observe $E[Y_i(1)|X]$ for those below the cutoff and $E[Y_i(0)|X]$ for those above the cutoff.

We can estimate the average treatment effects by taking the difference of the conditional expectations at the cutoff if these underlying functions are continuous throughout the cutoff:

$$lim_{\epsilon \downarrow 0} E[Y_i|X_i = c + \epsilon] - lim_{\epsilon \uparrow 0} E[Y_i|X_i = c + \epsilon] = E[Y_i(1) - Y_i(0)|X = c]. \tag{1.4}$$

As explained above, this *continuity assumption* is fulfilled by the RDD because we can assume that students have *imprecise control* over the assignment variable, their GPA. We can clearly identify the average treatment effects because there is a natural sharp cutoff at the threshold. The treatment administered to students is being confronted with the information that they are placed on probation and the subsequent threat of suspension. Being put on probation does not involve any actions by the students, in fact being assigned to the treatment group already constitutes the treatment in itself. Non-compliers thus do not pose a concern for this research design.

As the theoretical framework discussed in the prior section illustrates, students on probation face the decision of dropping out or trying to improve their performance in the next term. While the estimation on effects on dropping out using the regression discontinuity design is relatively straight forward, the estimation of effects for subsequent performance adds additional challenges.



The extended causal graph above illustrates how the subsequent performance of students is also affected by whether a student drops out or not. This factor adds additional complexity to the estimation problem because we cannot observe the subsequent GPA for students who drop out after being placed on probation. This factor puts into question the comparability of the treatment and control group in subsequent periods. I address these concerns and possible solutions in later sections of this notebook.

Aside from the two main outcomes, Lindo et al. (2010) also examine the effects of academic probation on graduation rates of students. However, since information about student's academic progress over the whole course of their studies is limited in the available data, only very simple analysis is possible.

4. Empirical Strategy

The authors examine the impact of being put on probation after the first year in university. The probation status after the first year is a deterministic function of student's GPA, formally

$$PROB_{IC}^{year1} = 1(GPANORM_{IC}^{year1} < 0), \tag{1.5}$$

where $PROB_{IC}^{year1}$ represents the probation status of student i at campus c and $GPANORM_{IC}^{year1}$ is the distance between student i's first-year GPA and the probationary cutoff at their respective campus. The distance of first-year GPA from the threshold thus constitutes the *running variable* in this RD design. Normalizing the running variable in this way makes sense because the three campuses have different GPA thresholds for putting students on probation (the threshold at campus 1 and 2 is 1.5, at campus 3 the threshold is 1.6), using the distance from the cutoff as the running variable instead allows Lindo et al. (2010) to pool the data from all three campuses.

Applying the regression discontinuity design, the treatment effect for students near the threshold is obtained by comparing the outcomes of students just below the threshold to those just above the threshold.

The following equation can be used to estimate the effects of academic probation on subsequent student outcomes:

$$Y_{ic} = m(GPANORM_{ic}^{year1}) + \delta 1(GPANORM_{ic}^{year1} < 0) + u_{ic} \tag{1.6}$$

- Y_{ic} denotes the outcome for student i at campus c,
- $m(GPANORM_{ic}^{year1})$ is a continuous function of students' standardized first year GPAs,
- $1(GPANORM_{ic}^{year1} < 0)$ is an indicator function equal to 1 if the student's GPA is below the probation cutoff,
- u_{ic} is the error term,
- δ is the coefficient for the estimated impact of being placed on academic probation after the first year.

For the regression analysis, Lindo et al. (2010) extend the above equation by an interaction term and a constant:

$$Y_{ic} = \alpha + \delta 1(GPANORM_{ic}^{year1} < 0) + \beta(GPANORM_{ic}^{year1}) + \gamma(GPANORM_{ic}^{year1}) \times 1(GPANORM_{ic}^{year1} < 0) + u_{ic}. \tag{1.7}$$

This regression equation does not include covariates because Lindo et al. (2010) implement a split sample analysis for the covariates in the analysis.

5. Replication of Lindo et al. (2010)

5.1. Data & Descriptive Statistics

Lindo et al. (2010) filter the data to meet the following requirements:

- Students high school grade measure is not missing,
- Students entered university before the year 2004 (to ensure they can be observed over a 2-year period),
- Students are between 17 and 21 years of age at time of entry.
- Distance from cutoff is maximally 0.6 (or 1.2).

The first three requirements are already fulfilled in the provided data. It should be noted that the high school measure is a student's average GPA in courses that are universally taken by high school students in the province. Thus all students that remain in the sample (84 % of the original data) attended high school in the province. This has the advantage that the high school measurement for all students is very comparable. An additional implication that should be taken note of for later interpretations is that this also implies that all students assessed in the study attended high school in the province. The group of 'nonnative' English speakers thus, for example, does not include students that moved to Canada after completing high school.

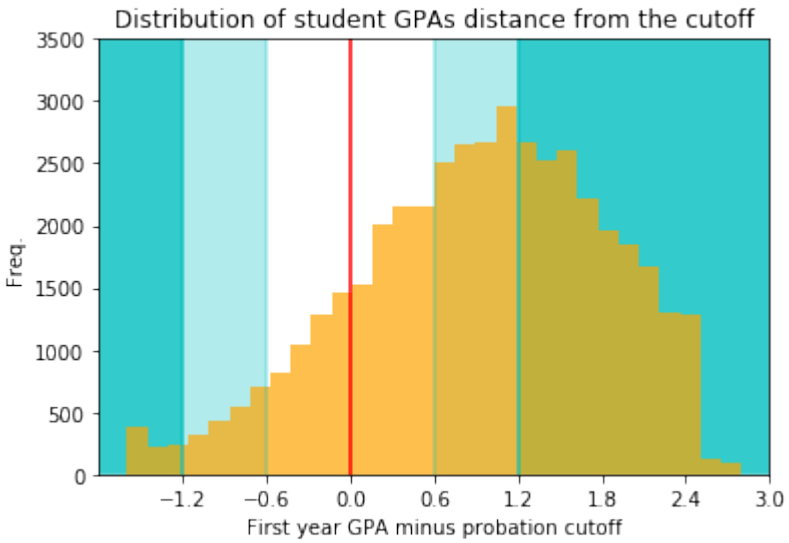
[3]:
```
data_1 = pd.read_stata("data/data-performance-standards-1.dta")
data_2 = pd.read_stata("data/data-performance-standards-2.dta")
```

[4]:
```
data = pd.concat([data_1, data_2], axis=1)
data = prepare_data(data)
```

NOTE: The original data provided by the authors can be found here. For this replication the data is split into two .dta-files due to size constraints.

As shown in the graph below, the distance from the cutoff for university GPA in the provided dataset still spans from values of -1.6 to 2.8 as can be seen below. Lindo et al. (2010) use a bandwidth of *(-0.6, 0.6)* for regression results and a bandwidth of *(-1.2, 1.2)* for graphical analysis.

[5]:
```
plot_hist_GPA(data)
```



[6]:
```
# Reduce sample to students within 1.2 points from cutoff.
sample12 = data[abs(data["dist_from_cut"]) < 1.2]
sample12.reset_index(inplace=True)
print(
    "A sample of students within 1.2 points from the cuttoff consits of",
    len(sample12),
    "observations.",
)
```

```
A sample of students within 1.2 points from the cuttoff consits of 25389 observations.
```

[7]:
```
# Reduce sample to students within 0.6 points from cutoff.
sample06 = data[abs(data["dist_from_cut"]) < 0.6]
sample06.reset_index(inplace=True)
print("The final sample includes", len(sample06), "observations.")
```

```
The final sample includes 12530 observations.
```

Table 1 shows the descriptive statistics of the main student characteristics and outcomes in the restricted sample with a bandwidth of 0.6 from the cutoff. The majority of students are female (62%) and native English speakers (72%). Students in the reduced sample on average placed in the 33rd percentile in high school. It should also be noted that quite a large number of students (35%) are placed on probation after the fist year. An additional 11% are placed on probation after the first year.

Table 1- Summary statistics

[8]:
```
create_table1(sample06)
```

[8]:

	Mean	Standard Deviation	Description	Type
hsgrade_pct	33.33	23.29	High School Grade Percentile	Characteristics
totcredits_year1	4.43	0.53	Credits attempted first year	Characteristics
age_at_entry	18.72	0.74	Age at entry	Characteristics
male	0.38	0.48	Male	Characteristics
english	0.72	0.45	English is first language	Characteristics
bpl_north_america	0.87	0.34	Born in North America	Characteristics
loc_campus1	0.48	0.50	At Campus 1	Characteristics
loc_campus2	0.21	0.41	At Campus 2	Characteristics
loc_campus3	0.31	0.46	At Campus 3	Characteristics
dist_from_cut	0.11	0.33	Distance from cutoff in first year	Outcomes
probation_year1	0.35	0.48	On probation after first year	Outcomes
probation_ever	0.46	0.50	Ever on acad. probation	Outcomes
left_school	0.05	0.22	Left Uni after 1st evaluation	Outcomes
nextGPA	0.47	0.81	Distance from cutoff at next evaluation	Outcomes
suspended_ever	0.16	0.36	Ever suspended	Outcomes
gradin4	0.29	0.45	Graduated by year 4	Outcomes
gradin5	0.56	0.50	Graduated by year 5	Outcomes
gradin6	0.66	0.47	Graduated by year 6	Outcomes

5.2. Results

5.2.1. Tests of the Validity of the RD Approach

The core motivation in the application of RD approaches is the idea, that the variation in treatment near the cutoff is random if subjects are unable to control the selection into treatment (Lee & Lemieux, 2010). This condition, if fulfilled, means the RDD can closely emulate a randomized experiment and allows researchers to identify the causal effects of treatment.

For evaluating the effects of academic probation on subsequent student outcomes, the RDD is thus a valid approach only if students are not able to precisely manipulate whether they score above or below the cutoff. Lindo et al. (2010) offer multiple arguments to address concerns about nonrandom sorting:

1. The study focuses on first-year students, assuming this group of students is likely to be less familiar with the probation policy on campus. To verify their conjecture, the authors also conducted a survey in an introductory economics course which revealed that around 50 % of students were unsure of the probation cutoff at their campus. They also claim that this analysis showed no relationship between knowledge of probation cutoffs and students' grades.
2. The authors also point out that most first-year courses span the entire year and most of the evaluation takes place at the end of the term which would make it difficult for students to purposely aim for performances slightly above the cutoff for academic probation.
3. Finally, and most importantly, the implication of local randomization is testable. If nonrandom sorting were to be a problem, there should be a discontinuity in the distribution of grades at the cutoff with a disproportionate number of students scoring just above the cutoff. Additionally, all the covariates should be continuous throughout the cutoff to ensure that the group above the probation cutoff constitutes a realistic counterfactual for the treated group.

In the following section, I first conduct a brief visual and descriptive check of validity before presenting my replication of the validity checks conducted in Lindo et al. (2010).

i. Extension: Visual Validity Check

To check for discontinuities in the covariates and the distribution of students around the cutoff Lindo et al. (2010) use local linear regression analysis. Before implementing the rather extensive validity check conducted by Lindo et al. (2010) I show in this section that a rather simple descriptive and graphical analysis of the distribution of covariates already supports the assumption they are continuous throughout the threshold.

Extension | Table - Descriptive Statistics of Treated and Untreated Group Close to the Cutoff

The table below shows the means of the different covariates at the limits of the cutoff from both sides (here within a bandwidth of 0.1 grade points). We can see that the means of the groups below and above the probation cutoff are very similar, even equal for some of the variables.

[9]:
```
cov_descriptives = describe_covariates_at_cutoff(sample06, bandwidth=0.1)
cov_descriptives
```

[9]:

	Below cutoff		Above cutoff	
	Mean	Std.	Mean	Std.
hsgrade_pct	30.94	22.61	31.76	22.65
totcredits_year1	4.42	0.55	4.37	0.54
age_at_entry	18.73	0.76	18.72	0.75
male	0.38	0.49	0.38	0.49

(continues on next page)

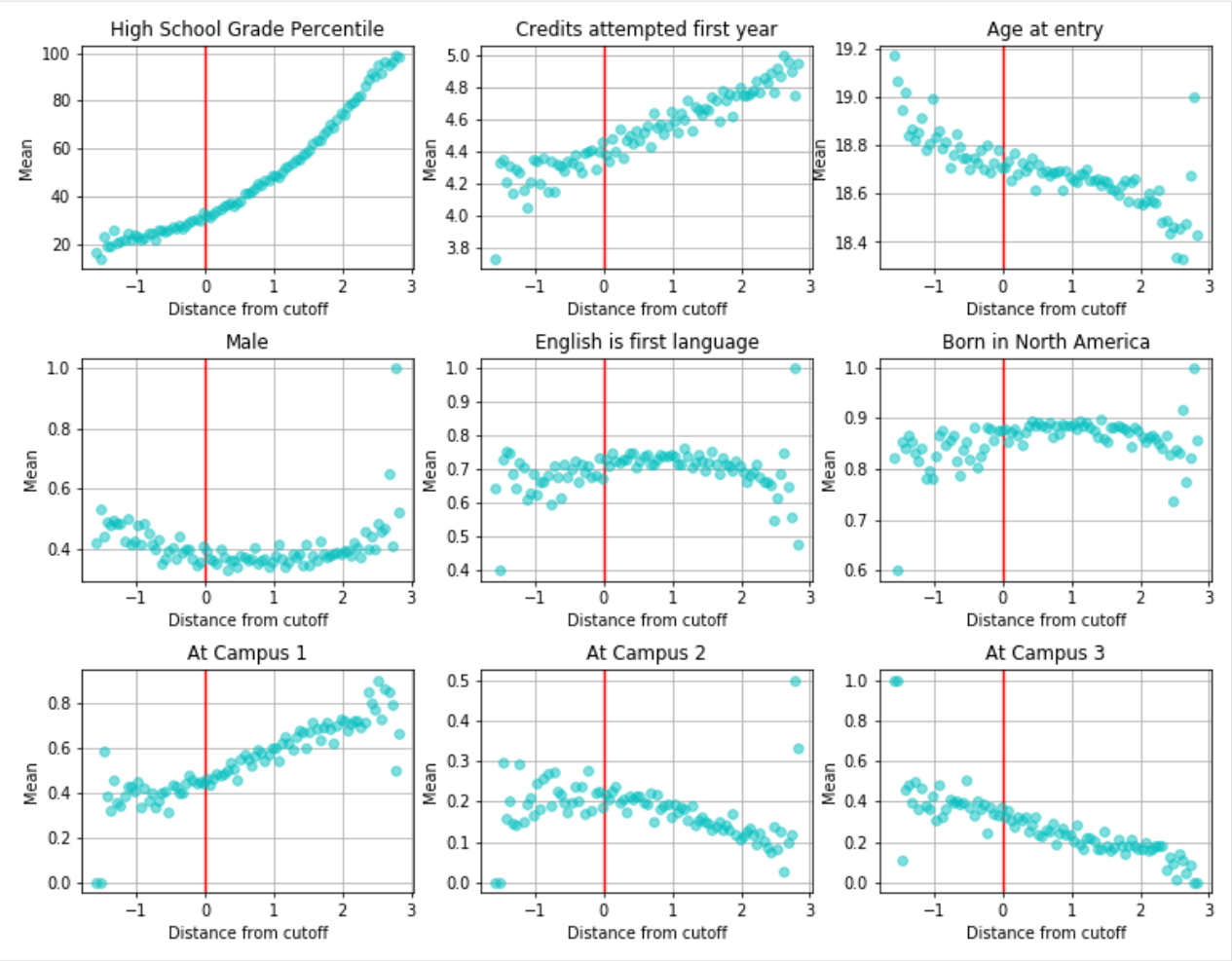
(continued from previous page)

```
english                  0.70   0.46         0.72   0.45
bpl_north_america        0.88   0.33         0.87   0.34
loc_campus1              0.44   0.50         0.45   0.50
loc_campus2              0.22   0.42         0.21   0.41
loc_campus3              0.34   0.47         0.34   0.47

                                  Description

hsgrade_pct        High School Grade Percentile
totcredits_year1   Credits attempted first year
age_at_entry                    Age at entry
male                                    Male
english               English is first language
bpl_north_america         Born in North America
loc_campus1                        At Campus 1
loc_campus2                        At Campus 2
loc_campus3                        At Campus 3
```

Extension | Figure - Distribution of Covariates throughout the Probation Cutoff

The figure below shows the means of the nine covariates in bins of size 0.5 (grade points). Similar to the descriptive table shown above, this visualization shows that there seem to be no apparent discontinuities in the distribution of students for any of the observable characteristics (graphs with bins of size 0.1 or 0.025 suggest the same).

[10]:
```
plot_covariates(
    data=data, descriptive_table=cov_descriptives, bins="dist_from_cut_med05"
)
```



ii. Advanced Validity Check

(as conducted by Lindo et al. (2010))

Figure 1 | Distribution of Student Grades Relative to their Cutoff

To test the assumption of local randomization, Lindo et al. (2010) run a local linear regression on the distribution of students throughout the cutoff. As mentioned above, these should be continuous as a jump in the distribution of students around the cutoff would indicate that students can in some way manipulate their GPA to place above the cutoff.

For the analysis, the data (containing all observations within 1.2 GPA points from the cutoff) is sorted into bins of size 0.1. The bins contain their lower limit but not their upper limit. To replicate the result from Lindo et al. (2010), I calculate the frequency of each bin and then run a local linear regression with a bandwidth of 0.6 on the size of the bins. Figure 1 shows the bins and the predicted frequency for each bin. The results show that the distribution of grades seems to be continuous around the cutoff, suggesting that we can assume local randomization.

This method of testing the validity is especially useful because it could capture the effects of unobservables, whose influence we cannot otherwise test like we test for discontinuities in observable characteristics in the parts above and below. If all observable characteristics would show to be continuous throughout the cutoff but we could still observe a jump in the distribution of students above the cutoff, this would suggest that some unobservable characteristic distin-

guishes students above and below the probation threshold. Fortunately, the results shown below indicate that this is not the case supporting the RDD as a valid identification strategy.

[11]:
```
bin_frequency_fig1 = calculate_bin_frequency(sample12, "dist_from_cut_med10")
predictions_fig1 = create_bin_frequency_predictions(
    bin_frequency_fig1, bin_frequency_fig1.bins.unique().round(4), 0.6
)
plot_figure1(
    bin_frequency_fig1, bin_frequency_fig1.bins.unique().round(4), predictions_fig1
)
```

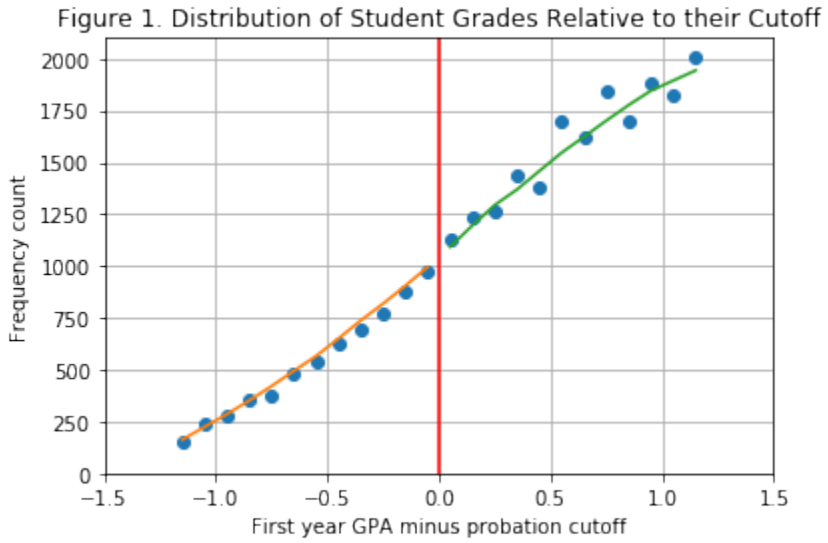


Table 2 - Estimated Discontinuities in Observable Characteristics

Table 2 shows the results of local linear regression (using a bandwidth of 0.6) for a range of observable characteristics that are related to student outcomes. Significant discontinuities would indicate that students with certain characteristics might be able to manipulate their grades to score above the probation cutoff. Similar to the descriptive validity checks on covariates in the section, these results additionally support the validity of the RDD. Table 2 shows that the coefficient for scoring below the cutoff is insignificant at the 10% level for all covariates.

[12]:
```
table2_variables = (
    "hsgrade_pct",
    "totcredits_year1",
    "age_at_entry",
    "male",
    "english",
    "bpl_north_america",
    "loc_campus1",
    "loc_campus2",
)
regressors = ["const", "gpalscutoff", "gpaXgpalscutoff", "gpaXgpagrcutoff"]
```

[13]:
```
table2 = estimate_RDD_multiple_outcomes(sample06, table2_variables, regressors)
table2.style.applymap(color_pvalues, subset=["P-Value (1)", "P-Value (0)"])
```

[13]:
```
<pandas.io.formats.style.Styler at 0x7fe748c013d0>
```

NOTE: My results for 'Male' and 'Age at entry' are switched compared to the table presented in Lindo et al. (2010). Since the results are identical otherwise, I assume this difference stems from an error in the table formatting of the published paper.

NOTE: The p-values in all regression tables are color-coded to enhance readability:

- P-values at the 10% level are magenta,
- P-values at the 5 % level are red,
- P-values at the 1 % level are orange.

The color-coding may not be visible in all viewing options for Jupyter Notebooks (e.g. MyBinder).

5.2.2. First Year GPAs and Academic Probation

Figure 2 and Table 3 show the estimated discontinuity in probation status. Figure 2 and the first part of Table 3 show the estimated discontinuity for the probation status after the *first year*. The second part of Table 3 presents the results for the estimated effects of scoring below the cutoff on the probability of *ever* being placed on academic probation.

Figure 2 and part 1 of Table 3 verify that the discontinuity at the cutoff is **sharp**, i.e. all students whose GPA falls below the cutoff are placed on probation. For students below the cutoff, the probability of being placed on probation is 1, for students above the cutoff it is 0.

It should be noted that the estimated discontinuity at the cutoff is only approximately equal to 1 for all of the different subgroups, as the results in Part 1 of Table 3 show. The authors attribute this fact to administrative errors in the data reportage.

Figure 2 - Porbation Status at the End of First Year

[14]:
```
predictions_fig2 = create_predictions(sample12, "probation_year1", regressors, 0.6)
plot_figure2(sample12, predictions_fig2)
```

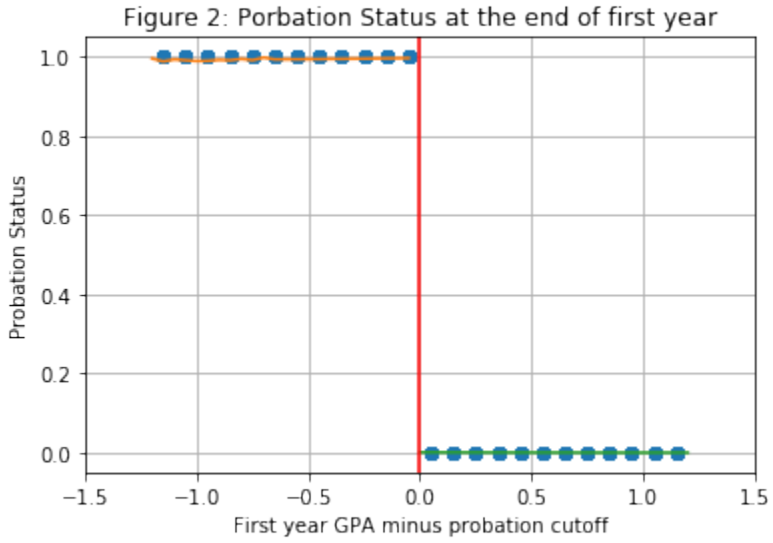


Table 3 - Estimated Discontinuity in Probation Status

To estimate the discontinuity in probation status, the authors again use a bandwidth of 0.6 from the cutoff. In addition to the whole sample, they also estimate the discontinuities for certain subgroups within the selected bandwidth:

- **high school grades below** and **above the median** (here, median refers to the median of the entire dataset (median: *50*) and not the median of the subset of students with a GPA within 0.6 grade points of the probation cutoff (the median for this set would be *28*))
- **male** and **female** students
- **english** native speakers and students with a different native language (**nonenglish**)

[15]:
```
sample_treat06 = sample06[sample06["dist_from_cut"] < 0]
sample_untreat06 = sample06[sample06["dist_from_cut"] >= 0]
sample06 = pd.concat([sample_untreat06, sample_treat06])
groups_dict_keys = [
    "All",
    "HS Grades < median",
    "HS Grades > median",
    "Male",
    "Female",
    "Native English",
    "Nonnative English",
]
groups_dict_columns = [
    "const",
    "lowHS",
    "highHS",
    "male",
    "female",
    "english",
    "noenglish",
]
groups_dict_06 = create_groups_dict(sample06, groups_dict_keys, groups_dict_columns)
```

Table 3 | Part 1 - Estimated Discontinuity in Probation Status for Year 1

[16]:
```
table3_1 = estimate_RDD_multiple_datasets(
    groups_dict_06, groups_dict_keys, "probation_year1", regressors
)
table3_1.style.applymap(color_pvalues, subset=["P-Value (1)", "P-Value (0)"])
```

[16]:
```
<pandas.io.formats.style.Styler at 0x7fe748c130d0>
```

Table 3 | Part 2 - Estimated Discontinuity in Probabtion Status Ever

Part 2 of Table 3 presents the estimated effect of scoring below the cutoff in the first year for *ever* being placed on probation. The results show that even of those who score slightly above the probation cutoff in year 1, 33 % are placed on probation at some other point in time during their studies.

For the different subgroups of students this value varies from 29% (for students with high school grades above the median) up to 36.7% (for the group of males). These results already indicate that we can expect heterogeneities in the way different students react to being placed on academic probation.

The fact that it is not unlikely for low performing students just slightly above the cutoff to fall below it later on also underlines these student's fitness as a control group for the purpose of the analysis. Lindo et al. (2010) argue that the

controls can be thought of as receiving a much weaker form of treatment than the group that is placed on probation, as scoring just above the cutoff in year 1 does not save students from falling below the cutoff and being placed on probation in subsequent terms.

[17]:
```
table3_1 = estimate_RDD_multiple_datasets(
    groups_dict_06, groups_dict_keys, "probation_ever", regressors
)
table3_1.style.applymap(color_pvalues, subset=["P-Value (1)", "P-Value (0)"])
```

[17]:
```
<pandas.io.formats.style.Styler at 0x7fe748b1a410>
```

5.2.3. The Immediate Response to Academic Probation

Students who have been placed on academic probation enter their next term at university with the threat of suspension in case they fail to improve their grades. Recalling the theoretical framework presented in prior sections, students face the following set of options after each term:

1. **Option 1**: Return to school, exhibit low effort and achieving a low GPA,
2. **Option 2**: Return to school, exhibit high effort with the intent of achieving a high GPA,
3. **Neither** option: Drop out of university.

Students on probation face a different set of choices than the students that were not placed on probation as the threat of suspension essentially eliminates option 1. Of course, students could enter the next term, exhibit low effort, and receive low grades, but this would result in suspension. Since both option 1 and option 3 result in the student not continuing school (at least for a certain period of time), students who cannot meet the performance standard (thus leading to suspension) are much better off dropping out and saving themselves the cost of attending university for another term.

Table 4 - Estimated Effect on the Decision to Leave after the First Evaluation

[18]:
```
table4 = estimate_RDD_multiple_datasets(
    groups_dict_06, groups_dict_keys, "left_school", regressors
)
table4.style.applymap(color_pvalues, subset=["P-Value (1)", "P-Value (0)"])
```

[18]:
```
<pandas.io.formats.style.Styler at 0x7fe748b08cd0>
```

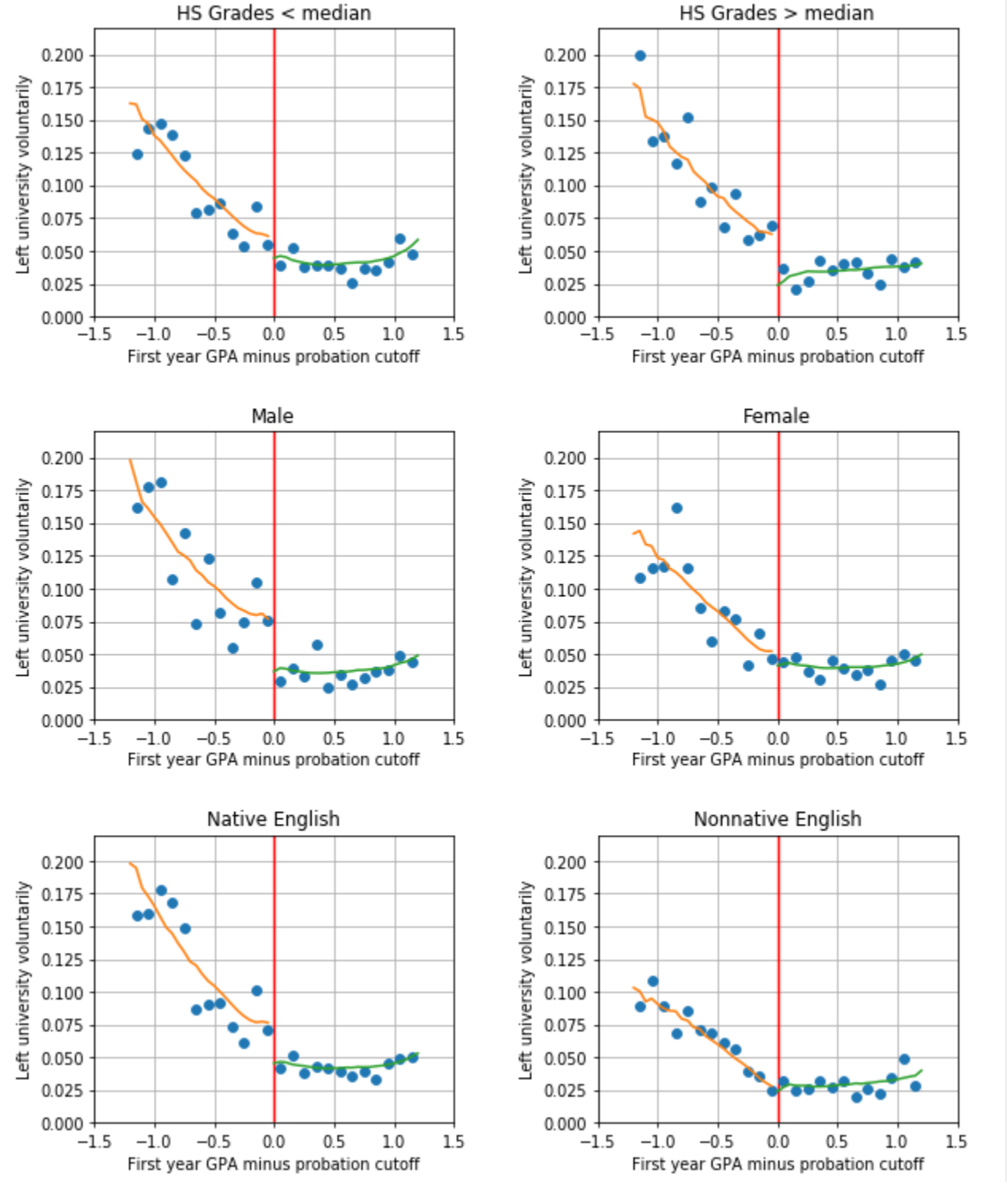
The results presented in Table 4 and and Figure 3 show the effects of being placed on probation on the probability to drop out of school after the first evaluation. The first row of Table 4 shows the average effect of academic probation on this outcome. The results indicate that, on average, being placed on probation increases the probability of leaving university by 1.8 percentage points. A student on academic probation is thus 44% more likely to drop out than their control group counterpart.

The results presented in the rest of Table 4 and and Figure 3 show that the average effect of being placed on probation is also characterized by large heterogeneities between the different subgroups of students. For males and native English speakers, the results, which are significant at the 5% level, show an increase of 3.7 and 2.8 percentage points respectively in the probability of leaving university after being placed on probation after the first evaluation. The results show no significant effects for these group's counterparts, the subgroups of females and nonnative English speakers.

Aside from gender and native language, the results also indicate that high school performance seems to play a role in how students react on being placed on probation. For the group of students who scored above the median in high school academic probation roughly doubles the probability of leaving school compared to the control group while there is no such effect for students who scored below the median in high school. Lindo et al. (2010) contribute this finding to a discouragement effect for those students who are placed on probation, which seems to be larger for students who did well in high school.

Figure 3 - Stratified Results for Voluntarily Leaving School at the End of the First year

[19]:
```
groups_dict_12 = create_groups_dict(sample12, groups_dict_keys, groups_dict_columns)
predictions_groups_dict = create_fig3_predictions(groups_dict_12, regressors, 0.6)
plot_figure3(groups_dict_12, predictions_groups_dict, groups_dict_keys)
```



5.2.4. The Impact onSubsequent Performance

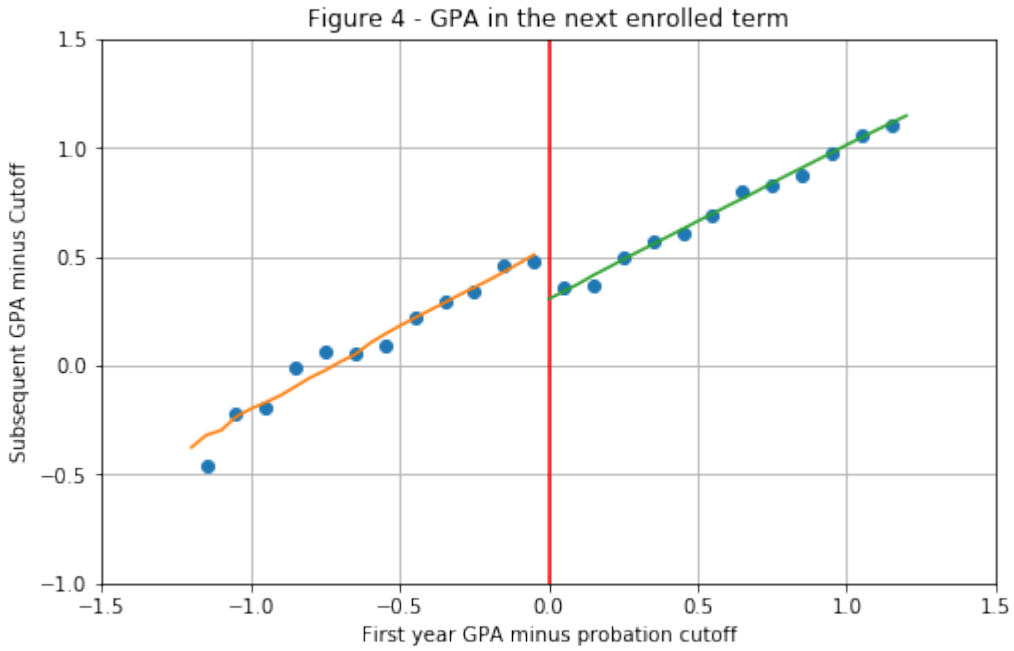
i. Main Results for Impact on GPA & Probability of Placing Above Cutoff in the Next Term

The next outcome Lindo et al. (2010) analyze is the performance of students who stayed at university for the next term. The theoretical framework presented in Section 2 predicts that those students on probation who stay at university will try to improve their GPA. Indeed, if they do not manage to improve, they will be suspended and could have saved themselves the effort by dropping out.

The results presented in Figure 4 and Table 5 show the estimated discontinuity in subsequent GPA. Lindo et al. (2010) find significant results (at the 5% level) for all subgroups, which is an even bigger effect than that of probation on drop out rates, where only some subgroups were affected.

Figure 4 - GPA in the Next Enrolled Term

[20]:
```
predictions_fig4 = create_predictions(sample12, "nextGPA", regressors, 0.6)
plot_figure4(sample12, predictions_fig4)
```



As part A of Table 5 shows, the average treatment effect on the GPA in the next term is positive for all groups of students. The average student on probation has a GPA increase of 0.23 grade points which is 74% of the control group.

The increase is greatest for students who have high school grades below the median. These students increase their GPA by 0.25 grade points on average, 90% more than their control group. This is an interesting finding because the counterpart, students who scored above the median in high school, are especially likely to drop out. Thus high school grades seem to have a large effect on whether students perceive academic probation as discouragement or as an incentive to improve their performance.

It should be noted here, that the '*next term*' may not be the next year for all students because some students take summer classes. If students fail to improve their grades during summer classes, they are already suspended after summer and will not enter the second year. Only using grades from the second year would thus omit students who were suspended before even entering the second year. The existence of summer classes may complicate the comparability of students after being put on probation. However, in a footnote Lindo et al. (2010) mention that they find no statistically

significant impact of academic probation on the probability that a student enrolls in summer classes and the estimates for subsequent GPA are nearly identical when controlling for whether a student's next term was attending a summer class.

NOTE: Lindo et al. (2010) in this call this the '*improvement*' of students' GPA, however, this phrasing in my opinion could be misleading, as the dependent variable in this analysis is the distance from cutoff in the next term. The results thus capture the increase in subsequent GPA in general and not relative to the GPA in the prior term.

Table 5 - Estimated Discontinuites in Subsequent GPA | Part A - Next Term GPA

[21]:
```
table5 = estimate_RDD_multiple_datasets(
    groups_dict_06, groups_dict_keys, "nextGPA", regressors
)
table5.style.applymap(color_pvalues, subset=["P-Value (1)", "P-Value (0)"])
```

[21]:
```
<pandas.io.formats.style.Styler at 0x7fe742572fd0>
```

Table 5 - Estimated Discontinuites in Subsequent GPA | Part B - Probability of Placing Above the Cutoff in Next Term

Panel B of Table 5 shows the probability of scoring above the cutoff in the next term. This statistic is very important because it decides whether students on academic probation are suspended after the subsequent term. It is therefore important for students who scored below the cutoff in the first year to not only improve their GPA, but improve it enough to score above the cutoff in the next term. Again academic probation increases the probability of students scoring above the cutoff in the next term for all subgroups.

[22]:
```
table5 = estimate_RDD_multiple_datasets(
    groups_dict_06, groups_dict_keys, "nextGPA_above_cutoff", regressors
)
table5.style.applymap(color_pvalues, subset=["P-Value (1)", "P-Value (0)"])
```

[22]:
```
<pandas.io.formats.style.Styler at 0x7fe7480b9990>
```

ii. Formal Bound Analysis on Subsequent GPA (partial extension)

As already mentioned in the section on the identification strategy, analyzing outcomes that occur after the immediate reaction to probation (the decision whether to drop out or not) becomes more challenging if we find that students are significantly more or less likely to drop out if they have been placed on academic probation. As discussed in the preceding section, this is the case because some groups of students indeed are more likely to drop out if they have been placed on probation.

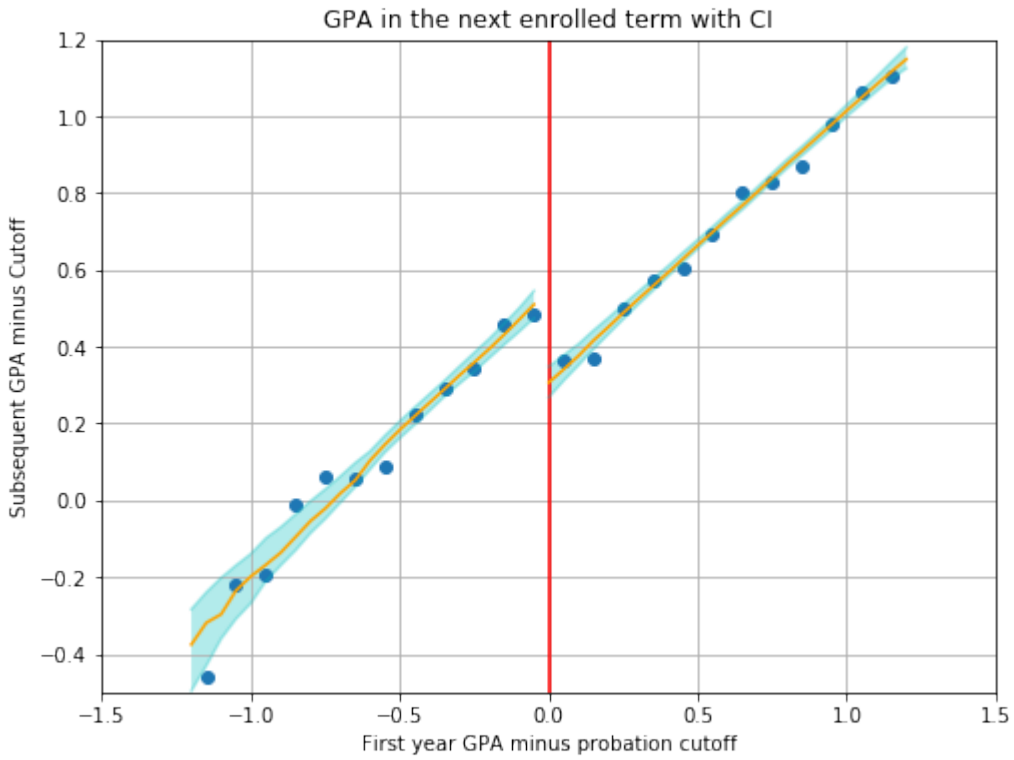
For the analysis of subsequent GPA, this means that the results become less reliable because there is a group of students (those who dropped out) whose subsequent performance cannot be observed. This can cause the results to be biased. For example, if academic probation causes students with relatively low ability to drop out (which the performance model would predict) then we would find a positive impact on subsequent GPA being solely driven by the fact that the low performers in the treatment group dropped out. If, on the other hand, high ability students were more likely to drop out, the estimates for the impact on subsequent performance would be downward biased.

In short, the control group might not be comparable anymore. To test whether the results on subsequent GPA are robust to these concerns, Lindo et al. (2010) use formal bound analysis for the results on subsequent GPA which I present below.

In addition to this formal bound analysis, I plot confidence intervals for the results on subsequent GPA. Confidence intervals are a useful way to support the graphical analysis of RDDs and ensure the discontinuity at the threshold does not disappear when new population samples are drawn. The graph below shows the estimates from before including a bootstrap 95% percent confidence interval. The confidence interval around the cutoff shows to be quite small, and the fall in subsequent GPA between the treatment and control group persists even at the borders of the confidence interval.

Subsequent Performance with 95% Confidence Interval

[23]:
```
bootstrap_pred = bootstrap_predictions(
    n=100, data=sample12, outcome="nextGPA", regressors=regressors, bandwidth=0.6
)
CI = get_confidence_interval(
    data=bootstrap_pred, lbound=2.5, ubound=97.5, index_var="dist_from_cut"
)
predictions_fig4_CI = pd.concat(
    [predictions_fig4, CI[["upper_bound", "lower_bound"]]], axis=1
)
plot_figure4_with_CI(data=sample12, pred=predictions_fig4_CI)
```



NOTE: The confidence intervals presented here are the product of only 100 resampling iterations of the bootstrap because increasing the number of times the data is resampled significantly increases the runtime of this notebook. However, I have tested the bootstrap for up to 1000 iterations and the results do not diverge very much from the version shown here.

This type of confidence interval, however, does not correct for potential biases in the treatment or control group discussed above because the bootstrap only resamples the original data and therefore can at best achieve the estimate resulting from the original sample.

To test the sensitivity to possible nonrandom attrition through specific students dropping out of university, Lindo et al. (2010) perform a formal bound analysis using a trimming procedure proposed by Lee (2009)*. The reasoning for this approach is based on the concerns described above. To find a lower bound of the estimate, Lindo et al. (2010) assume that academic probation causes students who would have performed worse in the next term to drop out. The control group is thus made comparable by dropping the lowest-performing students (in the next term) from the sample, assuming these students would have dropped out had they been placed on probation. To calculate the upper bound estimate, the same share of students is dropped from the upper part of the grade distribution instead.

The share of students who need to be dropped is given by the estimated impact of probation on leaving school. For example, in the entire sample students on probation are 1.8 percentage points more likely to drop out, which is 44% of the control mean. Thus, to make the groups comparable again we presumably need to drop 44% more students from the control group than actually dropped out.

For groups of students where the estimated impact of probation on leaving school is negative, students from the control group need to be dropped instead (i.e. here the lower bound is given by dropping the top students in the treatment group and the upper bound is given by dropping the bottom students).

While all results I have presented in this replication so far are exactly identical to the results from Lindo et al. (2010), I, unfortunately, cannot replicate the results from the formal bound analysis precisely. The description in the paper is brief and the provided STATA code from the authors does not include the formal bound analysis. While referring to methods presented in Lee (2009) has been helpful to understand the trimming procedure, I am unable to replicate the exact numbers presented in Lindo et al. (2010).

The table pictured below shows the results of the formal bound analysis presented in Lindo et al. (2010). The authors conclude that the positive effects of academic probation on students' subsequent GPA are too great to be explained by the attrition caused by dropouts.

NOTE: In their paper Lindo et al. (2010) quote *'Lee (2008)'* which could also refer to a different paper by Lee and Card from 2008 listed in the references. However, since this paper in contrast to the 2009 paper by Lee does not mention formal bound analysis and since Lee (2009) is not mentioned anywhere else in the paper, I am certain this is a citation error.

Formal Bound Analysis from Lindo et al. (2010) (p.110)

Relevant group	All (1)	HS grades < median (2)	HS grades > median (3)	Male (4)	Female (5)	Native English (6)	Nonnative English (7)
Lower bound estimate	0.186	0.207	0.083	0.111	0.224	0.142	0.228
	(0.031)	(0.039)	(0.099)	(0.063)	(0.041)	(0.054)	(0.057)
Upper bound estimate	0.259	0.270	0.212	0.258	0.258	0.262	0.247
	(0.028)	(0.031)	(0.084)	(0.042)	(0.038)	(0.041)	(0.054)

The table below shows my results using the proposed trimming procedure (table is again transposed compared to the original). The overall results are quite similar to the ones presented in Lindo et al. (2010), all estimates presented in Table 5 still lie between the lower and upper bound. It should be noted that in my replication the lower bound estimate for students with high school grades above the median was not significant at the 10% level while the results for all other groups were.

Replication of Formal Bound Analysis

[24]:
```
table4["add_leavers"] = round(
    table4["GPA below cutoff (1)"] / table4["Intercept (0)"], 2
)
add_leavers = table4["add_leavers"]
```

[25]:
```
lb_trimmed_dict_06 = trim_data(groups_dict_06, add_leavers, True, False)
lower_bound = estimate_RDD_multiple_datasets(
    lb_trimmed_dict_06, groups_dict_keys, "nextGPA", regressors
)
```

[26]:
```
ub_trimmed_dict_06 = trim_data(groups_dict_06, add_leavers, False, True)
upper_bound = estimate_RDD_multiple_datasets(
    ub_trimmed_dict_06, groups_dict_keys, "nextGPA", regressors
)
```

[27]:
```
bounds = pd.concat([lower_bound.iloc[:, [0, 2]], upper_bound.iloc[:, [0, 2]]], axis=1)
bounds.columns = pd.MultiIndex.from_product(
    [
        [
            "Lower Bound Estimate",
            "Upper Bound Estimate",
        ],
        ["GPA below cutoff (1)", "Std.err (1)"],
    ]
)
bounds
```

[27]:

	Lower Bound Estimate		Upper Bound Estimate	\
	GPA below cutoff (1)	Std.err (1)	GPA below cutoff (1)	
groups				
All	0.175	0.025	0.259	
HS Grades < median	0.211	0.028	0.267	
HS Grades > median	0.036	0.082	0.215	
Male	0.108	0.043	0.261	
Female	0.229	0.036	0.248	
Native English	0.140	0.035	0.261	
Nonnative English	0.215	0.053	0.256	

	Std.err (1)
groups	
All	0.026
HS Grades < median	0.029
HS Grades > median	0.079
Male	0.042
Female	0.036
Native English	0.035
Nonnative English	0.056

5.2.5. The Impacts on Graduation

As a third outcome, Lindo et al. (2010) examine the effects of academic probation on students' graduation rates. As already discussed in the previous section, the outcomes that are realized later in time are more complex to examine because of all the different choices a student has made until she or he reaches that outcome. Graduation rates are the product of a dynamic decision-making process that spans throughout the students' time at university. While the study focuses mainly on the effects of being put on probation after the first year, the decision problem described in the theoretical framework can be faced by students at different points during their academic career as students can be placed on probation each term or for multiple terms in a row. There are different ways in which academic probation could affect graduation rates. On the one hand, it could reduce the probability of graduating because probation increases the probability of dropping out and some students who fail to increase their grades are suspended. On the other hand, these students might have graduated either way and thus do not have an effect. Additionally, probation could increase graduation rates because those students who remain improve their performance.

Figure 5 - Graduation Rates

Figure 5 and Table 6 show the estimated impacts of academic probation after the first year on whether a student has graduated in four, five or six years. The effects are negative for all three options, suggesting that the negative effects discussed above overweigh potential positive effects on graduation rates.

[28]:
```
plot_figure5(
    sample12,
    create_predictions(sample12, "gradin4", regressors, 0.6),
    create_predictions(sample12, "gradin5", regressors, 0.6),
    create_predictions(sample12, "gradin6", regressors, 0.6),
)
```

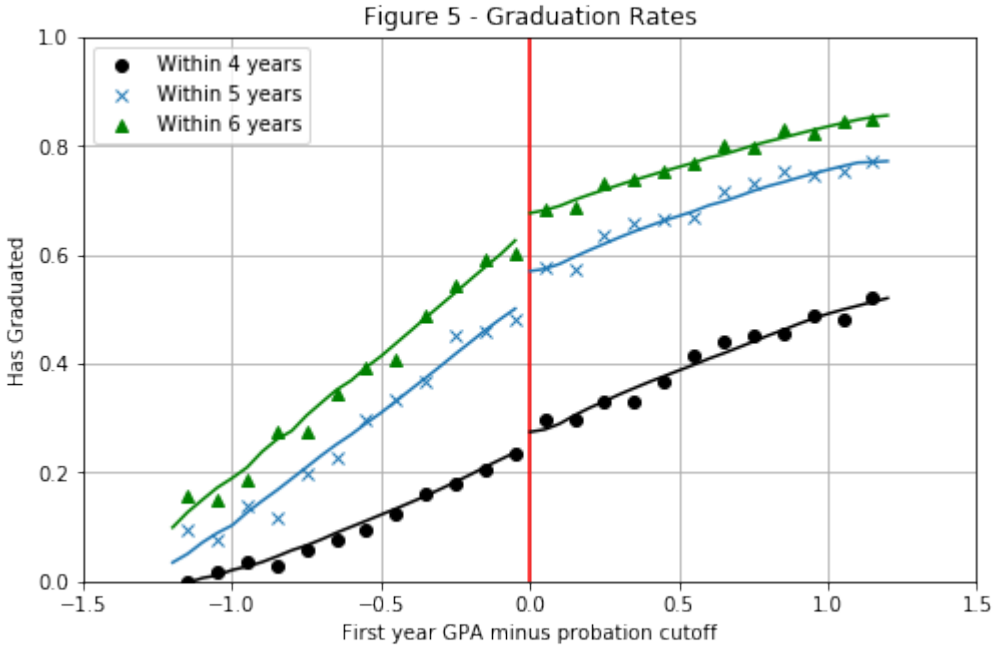


Table 6 - Estimated Effects on Graduation

The effects on graduation rates are insignificant for most subgroups, the group of students with high school grades above the median stands out as being especially negatively affected by being placed on probation in the first year. This group of students sees their probability of graduation within six years reduced by 14.5 percent. Lindo et al. (2010) attribute these results to the fact that this group of students is especially likely to drop out after being put on probation and also on average does not do much better than their counterpart if they continue to attend university.

Overall the results on graduation rates are rather limited. This likely stems from the more complex nature in which probation in the first year can affect this outcome later down the line. Unfortunately, most of the data in the provided dataset focus on the first two years of students' time at university (e.g. we only now the GPA of the first two years). Much more information would be needed to uncover the mechanisms in which probation may affect students' probability of graduating within specific timeframes.

NOTE: Below I only show the sections of Table 6 that are discussed above as the entire table is quite extensive. The other results presented in Table 6 of the paper can be viewed by uncommenting the code at the end of this section.

Graduated after 6 years

[29]:
```
table6 = create_table6(groups_dict_06, groups_dict_keys, regressors)
table6.loc[["All", "HS Grades > median"], "Graduated after 6 years"].style.applymap(
    color_pvalues, subset=["P-Value (1)", "P-Value (0)"]
)
```

[29]:
```
<pandas.io.formats.style.Styler at 0x7fe748a850d0>
```

Code for complete Table 6:

[30]:
```
# table6.loc[:, 'Graduated after 4 years'].style.applymap(
# color_pvalues, subset=['P-Value (1)', 'P-Value (0)']
# )
```

[31]:
```
# table6.loc[:, 'Graduated after 5 years'].style.applymap(
# color_pvalues, subset=['P-Value (1)', 'P-Value (0)']
# )
```

[32]:
```
# table6.loc[:, 'Graduated after 6 years'].style.applymap(
# color_pvalues, subset=['P-Value (1)', 'P-Value (0)']
# )
```

6. Extension: Robustness Checks

As discussed in my replication of Lindo et al. (2010) above, the authors use a variety of validity and robustness checks to analyze the reliability of their results. Aside from some smaller independent contributions that I already discuss in the replication part for better context, I in this section further analyze subsequent performance and check the bandwidth sensitivity of the results in drop out rates and subsequent GPA.

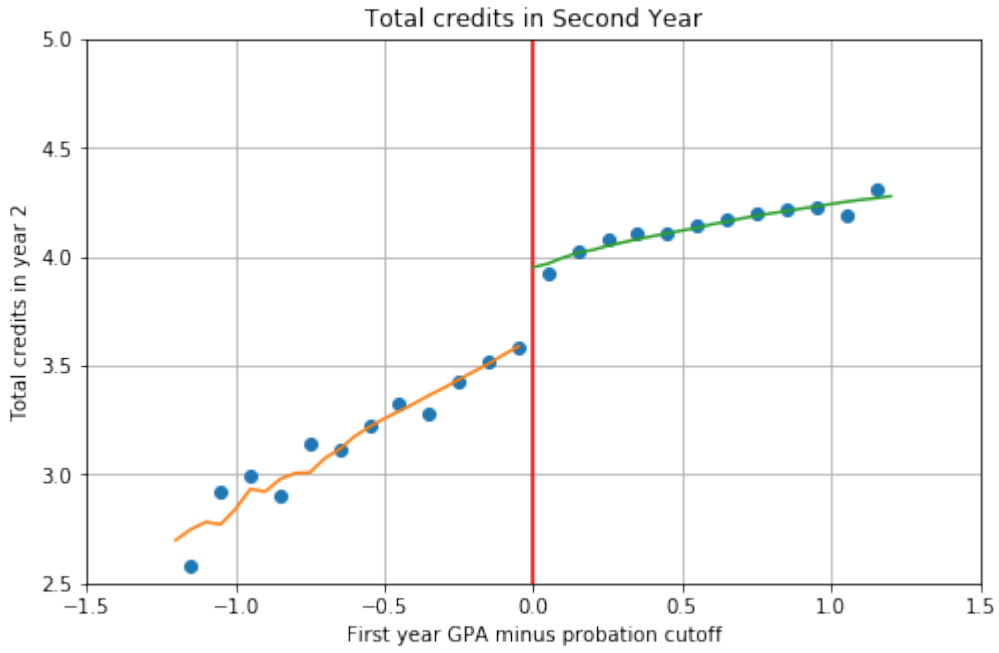
6.1. A Closer Look at Students' Subsequent Performance.

6.1.1. Subsequent Performance and Total Credits in Year 2

The results from Lindo et al. (2010) presented above show that students are more likely to drop out after being placed on academic probation but those who remain in school tend to improve their GPA above the cutoff in the next term. These results are generally in line with the theoretical framework presented in the paper which predicts that students either drop out or improve their GPA if the cost of not improving in the next term increases. The performance standard model explains these results through students self-selecting between increasing effort and dropping out based on their abilities (which are defined as the probability of meeting the performance standard). Students who are less likely to improve their GPA should thus be more likely to drop out. Unfortunately, it is not possible to test this prediction, as Lindo et al. (2010) emphasize in the paper because the probability of meeting the performance standard is not observed for students who leave school.

However, examining the students who remain in school may give some further insights. While Lindo et al. (2010) observe that students have been placed on probation on average improve their performance, it is not clear under which circumstances this is happening. A look at the amount of credits students are taking in their second year may give some insights. The results presented below show that being placed on probation after the first year has a negative effect on the amount of credits students take in the second year for all of the examined subgroups except the group of nonnative English speakers. This is a stark contrast to the first year where both the treatment and control group take almost the same amount of credits (as shown in the section on the validity of the RD Approach).

[33]:
```
predictions_credits_year2 = create_predictions(
    sample12, "total_credits_year2", regressors, 0.6
)
plot_figure_credits_year2(sample12, predictions_credits_year2)
```



The results indicate that being placed on probation decreases the total credits taken by the average student in year two by 0.33, around 8% of the control mean. As the table below shows, the results are most prominent for males, native English speakers, and students with high school grades above the median. Interestingly, these are the same groups of students that are most likely to drop out, suggesting that the discouragement effect persists throughout these groups and even those who re-enroll for the next term proceed with caution by taking fewer credits.

[34]:
```
table_total_credits_year2 = estimate_RDD_multiple_datasets(
    groups_dict_06, groups_dict_keys, "total_credits_year2", regressors
)
table_total_credits_year2.style.applymap(
    color_pvalues, subset=["P-Value (1)", "P-Value (0)"]
)
```

[34]:
```
<pandas.io.formats.style.Styler at 0x7fe748d0c2d0>
```

When interpreting these results it should be kept in mind that some students' next evaluation takes place during summer classes. Students who have taken summer classes enter their second year already having either passed the next evaluation or not. Those who fell below the cutoff will have been suspended and thus are missing from the data for the second year and those who have passed the threshold in the summer classes are likely not on probation anymore. Estimating the effects of probation on credits taken in the second year separately for both groups shows that those who did not take classes in the summer are more affected than those who did. For the students who took summer classes, the results are only significant for males, students with high school grades above the median and native English speakers.

No summer classes

[35]:
```
sample06_nosummer = sample06[sample06.summerreg_year1 == 0]
groups_dict_06_nosummer = create_groups_dict(
    data=sample06_nosummer, keys=groups_dict_keys, columns=groups_dict_columns
)
table_totcred_y2_nosummer = estimate_RDD_multiple_datasets(
    groups_dict_06_nosummer, groups_dict_keys, "total_credits_year2", regressors
)
table_totcred_y2_nosummer.style.applymap(
    color_pvalues, subset=["P-Value (1)", "P-Value (0)"]
)
```

[35]:
```
<pandas.io.formats.style.Styler at 0x7fe748d1d910>
```

Summer classes

[36]:
```
sample06_summer = sample06[sample06.summerreg_year1 == 1]
groups_dict_06_summer = create_groups_dict(
    sample06_summer, groups_dict_keys, groups_dict_columns
)
table_totcred_y2_summer = estimate_RDD_multiple_datasets(
    groups_dict_06_summer, groups_dict_keys, "total_credits_year2", regressors
)
table_totcred_y2_summer.style.applymap(
    color_pvalues, subset=["P-Value (1)", "P-Value (0)"]
)
```

[36]:
```
<pandas.io.formats.style.Styler at 0x7fe748cc0810>
```

These findings are useful for interpreting the subsequent performance of students because more credits likely signify a larger workload for the student. Instead of increasing their effort, students may just decrease their workload by completing fewer credits in the next term. Unfortunately, we cannot test this in detail because the data doesn't show how many credits students completed in which term.

Reducing the sample for the analysis of the subsequent GPA to students who did not attend summer classes and completed 4 credits in the second year (the most frequent amount of credits takeen by this group of students) shows that the effect of scoring below the cutoff in year 1 becomes insignificant for the students who have above-median high school grades and nonnative English speakers. The improvement decreases a bit for some groups like females or students with high school grades below the median but increases for others like males and native english speakers. Overall the results are still highly significant though considering the small window of observations to which the data is reduced in this case. This suggests that while students on probation do seem to take fewer credits in the next year, the improvements to subsequent performance is too great to just be attributed to students decreasing their workload.

[37]:
```
sample06_many_credits = sample06_nosummer[(sample06_nosummer.total_credits_year2 == 4)]
groups_dict_06_manycredits = create_groups_dict(
    sample06_many_credits, groups_dict_keys, groups_dict_columns
)
table_manycredits = estimate_RDD_multiple_datasets(
    groups_dict_06_manycredits, groups_dict_keys, "nextGPA", regressors
)
table_manycredits.style.applymap(color_pvalues, subset=["P-Value (1)", "P-Value (0)"])
```

[37]:
```
<pandas.io.formats.style.Styler at 0x7fe748e341d0>
```

6.1.2. Subsequent Cumulative Grade Point Average (CGPA)

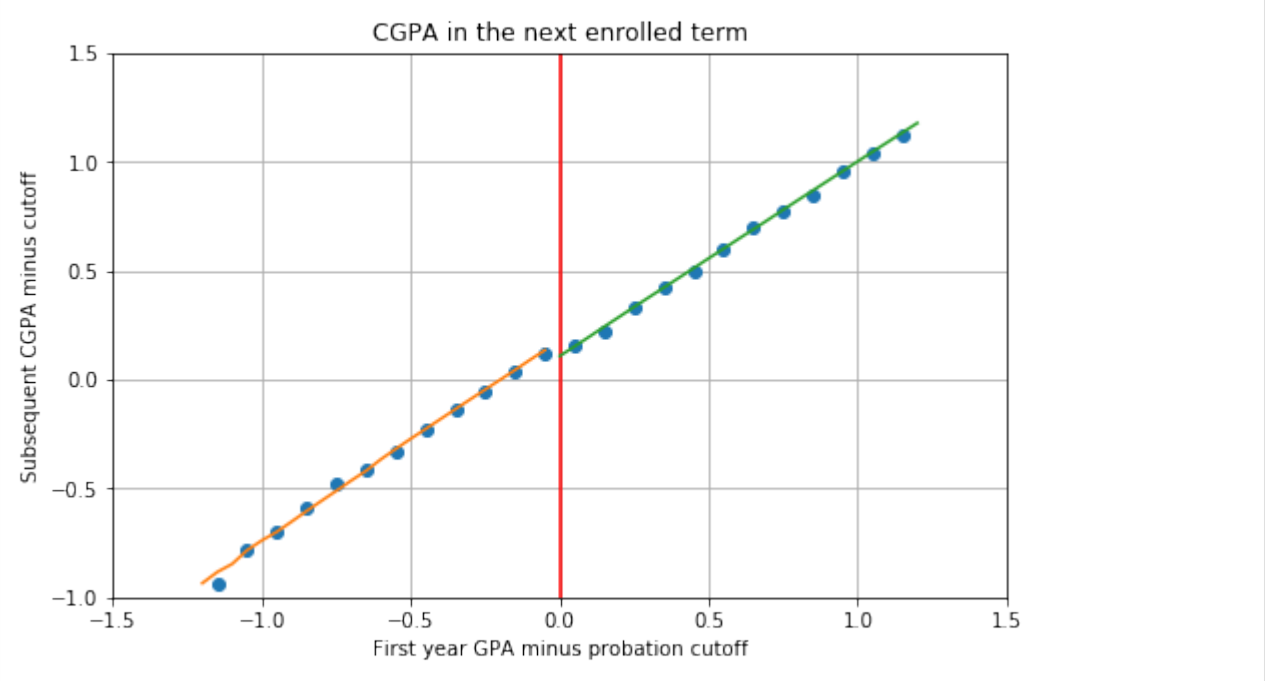
An additional factor that might be important for the analysis of subsequent performance is the Cumulative Grade Point Average (CGPA). Lindo et al. (2010) focus their analysis of subsequent performance solely on the grades achieved in the next term. However, in the section on the institutional background in the paper the authors write:

> *At all campuses, students on probation can avoid suspension and return to good academic standing by bringing their cumulative GPA up to the cutoff.* (Lindo et al., 2010, p.98).

To avoid suspension in the long term, students on probation thus are required to not only score above the cutoff in the next term but to score high enough to bring their CGPA above the probation threshold. Students who score above the threshold in the next term but still have a CGPA below the cutoff remain on probation. Students who fail to bring their GPA above the cutoff (and thus also their CGPA since their first-year GPA and first-year CGPA are the same) are suspended.

As the figure and table below show, the positive effects of probation on subsequent performance carry over to students' CGPA as well. Being placed on probation on average increases students' CGPA by 0.07 grade points or 63% of the control mean although the difference is rather difficult to spot visually.

[38]:
```
predictions_nextCGPA = create_predictions(sample12, "nextCGPA", regressors, 0.6)
plot_nextCGPA(sample12, predictions_nextCGPA)
```



Effect of Academic Probation on Subsequent CGPA

[39]:
```
table_nextCGPA = estimate_RDD_multiple_datasets(
    groups_dict_06, groups_dict_keys, "nextCGPA", regressors
)
table_nextCGPA.style.applymap(color_pvalues, subset=["P-Value (1)", "P-Value (0)"])
```

[39]:
```
<pandas.io.formats.style.Styler at 0x7fe748e27650>
```

However, in contrast to the probability of improving the next term GPA above the cutoff, academic probation has no significant effect on the probability of improving the CGPA above the cutoff in the next term except for the group of nonnative English speakers where the probability is actually negative. Indeed, out of all students on probation (within 0.6 grade points of the cutoff), only around 37% improve their next term CGPA above the cutoff. Around 23% improve their GPA above the cutoff but not their CGPA and remain on probation. The other students dropped out or are suspended after the next term. This suggests that the effects of probation span much longer than just the subsequent term for many students, not only indirectly because they have had the experience of being placed on probation but also directly because many of them remain on probation for multiple subsequent terms. These factors underline the points made in previous sections about the complexity of the way academic probation can affect a student's academic career. After being placed on probation a student can take a multitude of different paths, many more than the theoretical framework introduced in Section 2 leads on. A more dynamic approach to estimating the effects of academic probation could likely offer more insights into how students react to this university policy.

Effect of Academic Probation on the Probability of Achieving a CGPA Above the Cutoff in the Next Term

[40]:
```
table_nextCGPA_above_cutoff = estimate_RDD_multiple_datasets(
    groups_dict_06, groups_dict_keys, "nextCGPA_above_cutoff", regressors
)
table_nextCGPA_above_cutoff.style.applymap(
    color_pvalues, subset=["P-Value (1)", "P-Value (0)"]
)
```

[40]:
```
<pandas.io.formats.style.Styler at 0x7fe748deeb10>
```

6.2. Bandwidth Sensitivity

As a final robustness check, I evaluate the model at different bandwidths to ensure that results are not limited to one specific sample of students within a particular bandwidth. Lindo et al. (2010) use a distance from the threshold of 0.6 for the main regression analysis and 1.2 for graphical analysis (although the estimated curve at each point relies on a local linear regression with a bandwidth of 0.6 as well). The chosen bandwidth around the cutoff thus captures around 25% of the total range of grades (the GPA values observed in the first year span from 0 to 4.3).

Lindo et al. (2010) do not discuss the reasoning behind their choice of bandwidth in detail and do not apply optimal bandwidth selection methods like some other applications of regression discontinuity (Imbens & Lemieux, 2008; Lee & Lemieux, 2010). However, from a heuristic standpoint, this bandwidth choice seems reasonable. Since the cutoff lies at a GPA of 1.5 (1.6 at Campus 3), this bandwidth includes students whose GPA falls roughly between 0.9 and 2.1 grade points, so a range of around one average grade point including the edges. A much larger bandwidth would not make sense because it would include students that are failing every class and students who are achieving passable grades and are thus not very comparable to students who pass or fall below the threshold by a small margin.

I evaluate bandwidths of length 0.2 (0.1 distance from cutoff on each side) up to 2.4 (1.2 distance from cutoff on both sides). As Lindo et al. (2010), I choose a maximum bandwidth of 1.2 the reasons explained in the paragraph above.

Bandwidth sensitivity of the effect of probation on the probability of leaving school

The table below shows the estimated effect of probation on the probability to leave school after the first year using local linear regression (same specification as before) for all bandwidths between 0.1 and 1.2. The bandwidths are on the vertical axis, and the different subgroups are on the horizontal axis of the table. An *x* in the table indicates that the estimate was insignificant at the 10% level and is thus not shown for readability.

The table shows that the results for the effects on leaving school are relatively sensitive to bandwidth selection. Estimates of students within only 0.2 grade points of the probation threshold are not significant for any of the groups considered. Results for students with high school grades below the median are only significant for bandwidths between 0.3 and 0.5 while estimates for students with high school grades above the median are only significant between values of 0.5 and 0.7. The results for the other subgroups, on the other hand, seem to be quite robust to bandwidth selection.

The findings reported in this table suggest that some results presented in the previous sections should be interpreted carefully. Especially the estimates of students based on high school grades might be driven by some underlying factors that are not observed in this study. These could explain the sensitivity of the results to bandwidth selection.

[41]:
```
bandwidths = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2]
summary_left_school = bandwidth_sensitivity_summary(
    data, "left_school", groups_dict_keys, groups_dict_columns, regressors
)
```

(continues on next page)

(continued from previous page)

```
summary_left_school.loc[(bandwidths, "probation"), :]
# summary_left_school  #<- uncommenting this code will reveal the table including pvalues
```

[41]:

	All	HS Grades < median	HS Grades > median	Male	Female	\
0.1 probation	x	x	x	x	x	
0.2 probation	x	x	x	x	x	
0.3 probation	0.021	0.019	x	0.046	x	
0.4 probation	0.024	0.022	x	0.063	x	
0.5 probation	0.021	0.016	0.038	0.05	x	
0.6 probation	0.018	x	0.032	0.037	x	
0.7 probation	0.017	x	0.032	0.038	x	
0.8 probation	0.014	x	x	0.036	x	
0.9 probation	x	x	x	0.037	x	
1.0 probation	x	x	x	0.03	x	
1.1 probation	0.01	x	x	0.028	x	
1.2 probation	0.013	x	x	0.029	x	

	Native English	Nonnative English
0.1 probation	x	x
0.2 probation	x	x
0.3 probation	0.031	x
0.4 probation	0.037	x
0.5 probation	0.032	x
0.6 probation	0.028	x
0.7 probation	0.028	x
0.8 probation	0.022	x
0.9 probation	x	x
1.0 probation	x	x
1.1 probation	0.015	x
1.2 probation	0.018	x

Bandwidth sensitivity of the effect of probation on subsequent GPA

The results for the effects of academic probation on subsequent performance, on the other hand, seem to be quite robust to bandwidth selection. The estimated effects are the highest for most subgroups around the threshold of 0.6 chosen by Lindo et al. (2010) but the effects do not change sign for any subgroup and still remain quite similar.

Again, the group of students with high school grades above the median does not show significant results for bandwidths between 0.1 and 0.4 and thus seems to be the most sensitive to bandwidth selection.

[42]:

```
summary_nextGPA = bandwidth_sensitivity_summary(
    data, "nextGPA", groups_dict_keys, groups_dict_columns, regressors
)
summary_nextGPA.loc[(bandwidths, "probation"), :]
# summary_nextGPA    #<- uncommenting this code will reveal the table including pvalues
```

[42]:

	All	HS Grades < median	HS Grades > median	Male	Female	\
0.1 probation	0.191	0.19	x	0.163	0.208	
0.2 probation	0.172	0.222	x	0.131	0.201	
0.3 probation	0.233	0.258	x	0.203	0.247	
0.4 probation	0.241	0.257	x	0.21	0.25	
0.5 probation	0.227	0.247	0.141	0.183	0.246	

(continues on next page)

(continued from previous page)

```
0.6 probation  0.233              0.247              0.179  0.207  0.246
0.7 probation  0.246              0.255              0.216  0.197  0.273
0.8 probation  0.232              0.237              0.198  0.181   0.26
0.9 probation  0.215              0.221              0.178  0.164  0.242
1.0 probation  0.218               0.22              0.189  0.157  0.251
1.1 probation  0.224              0.226              0.197  0.183  0.247
1.2 probation  0.233              0.233              0.203  0.201  0.249

               Native English Nonnative English
0.1 probation           0.134              0.311
0.2 probation           0.154              0.207
0.3 probation           0.221              0.258
0.4 probation           0.237              0.247
0.5 probation           0.221              0.237
0.6 probation           0.229               0.24
0.7 probation           0.238              0.264
0.8 probation           0.229              0.237
0.9 probation           0.213              0.219
1.0 probation           0.211              0.234
1.1 probation           0.214              0.249
1.2 probation            0.23              0.243
```

7. Conclusion

Overall, the results in this notebook support the findings reported by Lindo et al. (2010) in their paper. The transparent research methods and STATA code provided by the authors allowed me to reproduce the results precisely for almost all tables and figures except for the formal bound analysis presented in Section 5.2.4. for which I could only produce similar results. In addition to the replication of the main results from Lindo et al. (2010), I discuss the identification strategy used in the paper and evaluate the robustness of the results, especially in the context of the performance standard model used in the paper. The results presented in Lindo et al. (2010) and my additional evaluation offer overall support for the performance standard model by Bénabou and Tirole (2000) which predicts that students who are put on probation will be more likely to drop out of university or improve their performance if they remain in school. However, one core feature of the model, the idea that students make their choices based on their ability to meet the performance standard, could not be tested due to the fact that the subsequent performance of students who left school cannot be observed.

Lindo et al. (2010) find large heterogeneities in the way students react to probation based on a set of covariates, however, the underlying sources of these heterogeneities are not evaluated. Further analysis of performance standards like academic probation using a larger set of information on student characteristics like personality traits, patience or socioeconomic background may thus be helpful to reveal the reasons why different students react to this negative incentive in certain ways.

Additionally, the study focused only on the effects of academic probation in the first year and relatively short term outcomes while long term outcomes were not assessed in detail. As already discussed in the section on the effects of academic probation on graduation rates, analyzing long term outcomes is much more difficult because of the multitude of different choices a student can make before reaching a specific outcome. Being placed on probation in the first year already expands the types of paths students may follow greatly. However, students can be placed on probation, suspended or leave school each term. To analyze the long term effects of academic probation in detail, there are too many questions that the data used in this study cannot answer.

Overall the findings from Lindo et al. (2010) offer quite robust results on the effects of academic probation on low performing students. They contribute important insights into how students or individuals in general may react to negative incentives with the threat of severe real-world penalties if they fail to adjust their behavior.

8. References

- **Bénabou, R., & Tirole, J. (2000)**. *Self-Confidence and Social Interactions* (No. w7585). National bureau of economic research.
- **Imbens, G. W., & Lemieux, T. (2008)**. Regression discontinuity designs: A guide to practice. *Journal of Econometrics*, 142(2), 615-635.
- **Lee, D. S. (2009)**. Training, Wages, and Sample Selection: Estimating Sharp Bounds on Treatment Effects. *The Review of Economic Studies*, 76(3), 1071-1102.
- **Lee, D. S., & Lemieux, T. (2010)**. Regression Discontinuity Designs in Economics. *Journal of Economic Literature*, 48(2), 281-355.
- **Lindo, J. M., Sanders, N. J., & Oreopoulos, P. (2010)**. Ability, Gender, and Performance Standards: Evidence from Academic Probation. *American Economic Journal: Applied Economics*, 2(2), 95-117.
- **Thistlethwaite, D. L., & Campbell, D. T. (1960)**. Regression-discontinuity analysis: An alternative to the ex post facto experiment. *Journal of Educational Psychology*, 51(6), 309.

Notebook by Annica Gehlen | Find me on GitHub at https://github.com/amageh.

1.4.2 Collaboration Projects

Examples of collaboration projects with Daimler AG and Deutsch Bank organized in a previous iteration of the OSE data science course are available below.

Daimler AG

The main focus of this project is to develop a procedure to identify abnormal observations in sensor data collected during production processes. Each data point can be interpreted as a set of points where a function was observed. Therefore, the whole problem can be approached from a standpoint of outlier classification in functional data.

Project by Jakob Juergens

Outlier Detection in Sensor Data using Functional Depth Measures

Notebook by **Jakob R. Jürgens** - Final project for the course **OSE - Data Science** in the summer semester 2021 - Find me at jakobjuergens.com

---

The best way to access this project is to clone this repository and execute the jupyter notebook and the shiny app locally. Alternatively, on the main site of this repository, there are nbviewer and binder badges set up to directly look at them in the browser.

The following packages and their dependencies are necessary to execute the notebook and the shiny app. If you are executing the code locally, make sure that these packages are provided and that an R Kernel (like irkernel) is activated in the Jupyter notebook.

[1]:
```
suppressMessages(library(MASS))
suppressMessages(library(tidyverse))
suppressMessages(library(shiny))
suppressMessages(library(shinydashboard))
```

(continues on next page)

(continued from previous page)

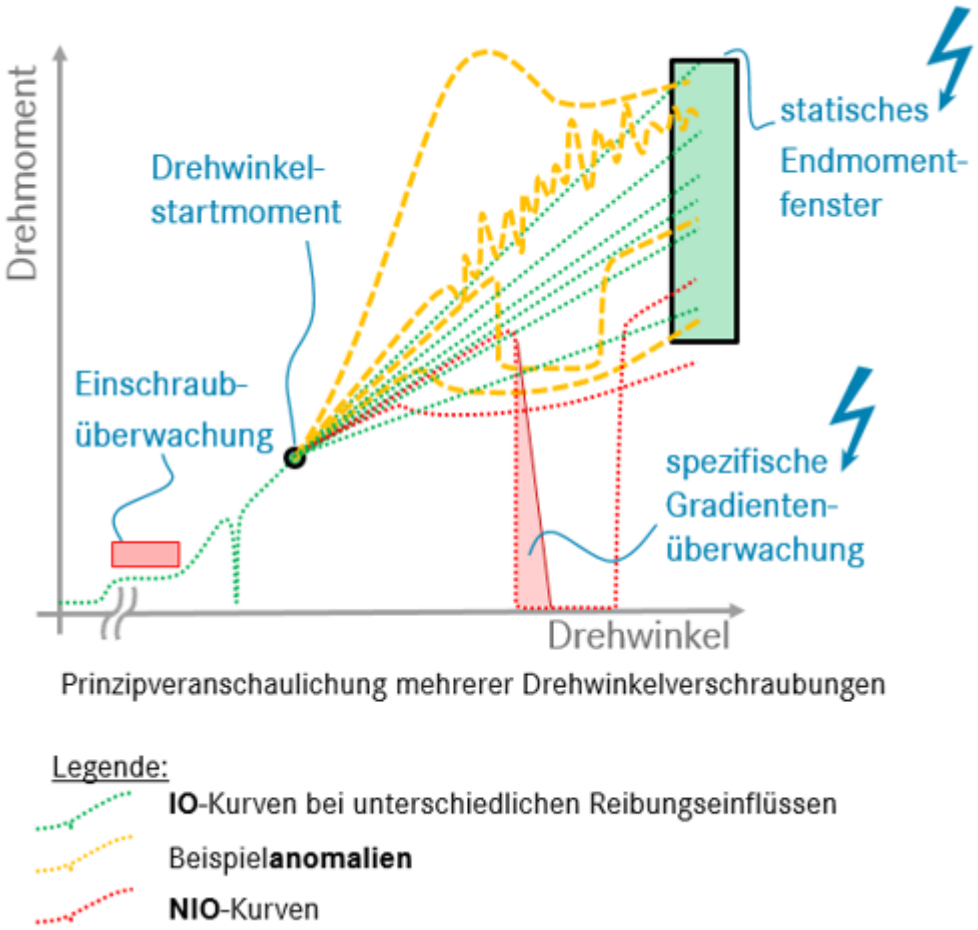
```
suppressMessages(library(largeList))
suppressMessages(library(parallel))
suppressMessages(library(Rcpp))
suppressMessages(library(repr))

options(repr.plot.width=30, repr.plot.height=8)

# suppressMessages(library(gganimate)) #needed only to generate the gifs

source("auxiliary/observation_vis.R")
source("auxiliary/distribution_vis.R")
source("auxiliary/updating_vis.R")
source("auxiliary/generate_set_1.R")
source("auxiliary/generate_set_2.R")
source("auxiliary/generate_set_3.R")

sourceCpp('auxiliary/rcpp_functions.cpp')
```

Table of Contents

Introduction

This project is part of a cooperation with **Daimler AG** and deals with outlier detection in sensor data from production processes. Since this project will take place over the courses **OSE - Data Science** (summer semester 2021) and **OSE - Scientific Computing for Economists** (winter semester 2021/2022), the current state of the project can be seen as a description of the progress at the half time point. Much of the following will be developed further and is therefore subject to change in future revisions.

The main data set that is used in this project deals with the relation of angle and torque during the process of tightening a bolt in a screwing connection. The corresponding data set will be called "Endanzugsproblem" in the following notebook and contains ~350000 observations of what can be imagined as a function that maps angles to torque. The following schematic will give an idea of what the data set represents and what the problem is:



To clarify some things about this simplified schematic:

- The so called Endanzug is only part of the tightening process, but the parts of the observation happening before it are not subject of this analysis
- The focus of this project lies on curves that are "In Ordnung", so observations that do not immediately disqualify themselves in some way by for example not reaching the fixed window of acceptable final values
- The observations typically have a high frequency of measuring torque, but the measuring points are not equidistant

- The angles where torque is measured are not shared between observations, but the measuring interval of angles might overlap between observations
- The Endanzug does not start at the same angle for every observation and also does not necessarily start at the same torque
- Outliers can be very general, so methods based on detecting only specific types of outliers may not be able to effectively filter out other suspicious observations. So the optimum would be to have some kind of Omnibus test for outliers

Especially due to the high frequency of measurement and the non-identical points where torque is measured, the idea of interpreting each observation as a function and therefore approaching the problem from a standpoint of functional data analysis comes to mind. One method that is used in functional data analysis to identify outliers is based on what is called a "functional depth measure". Gijbels and Nagy (2017) introduces the idea of depth as follows and then elaborate on the theoretical properties a depth function for functional data should possess:

> For univariate data the sample median is well known to be appropriately describing the centre of a data cloud. An extension of this concept for multivariate data (say p-dimensional) is the notion of a point (in $\mathbb{R}^p$) for which a statistical depth function is maximized.

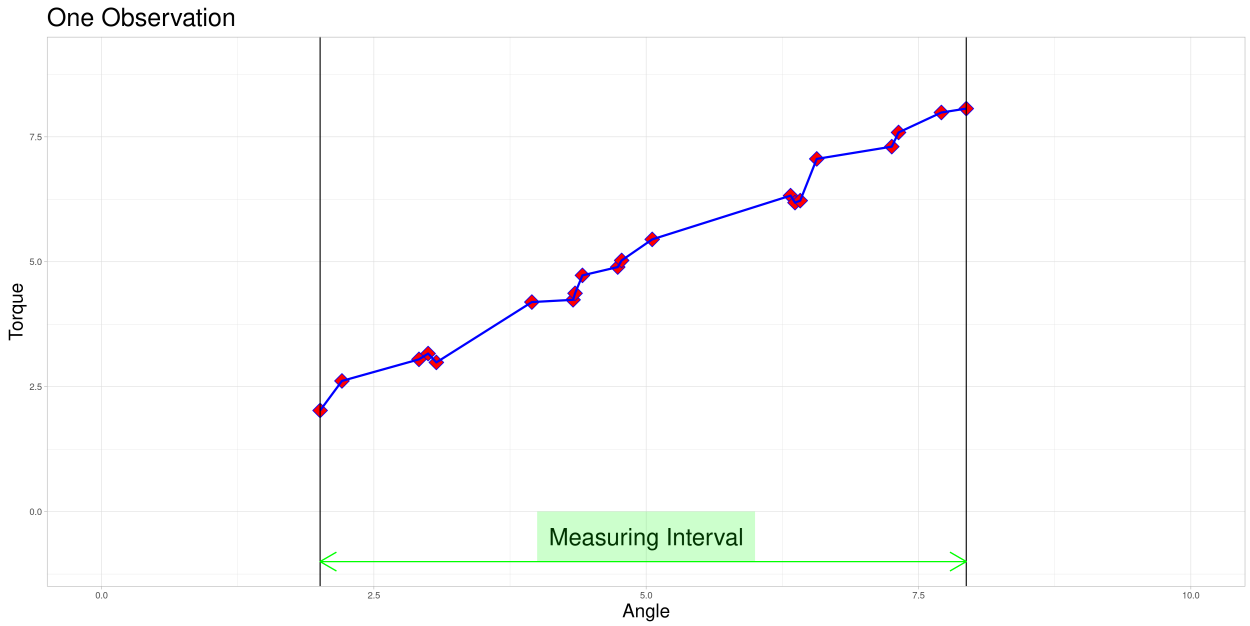
The idea is to define an analogous concept to centrality measures (such as the distance from some central tendency like the median) in a scalar setting for functional data and then use those to determine which functions in a set are typical for the whole population. Due to the more applied nature of this project, I will not go into detail on the theoretical properties of the methods used, but focus on giving intuition why the chosen methods make sense in this context.

The main inspiration for my approach to the problem of detecting outliers in a data set such as the one described above is the paper "**Outlier detection in functional data by depth measures, with application to identify abnormal NOx levels**" by Febrero-Bande, Galeano, and Gonzàlez-Manteiga (2008). I am going to first describe their algorithm, then present my extensions and my implementation and finally apply it to three simulated data sets mimicking the "Endanzugsproblem" as the original data is property of Daimler AG, which I cannot make public.

Observation Structure

For the sake of clarity, I will show the typical structure of one observation and define a couple of objects I will refer to later. To give context for the later choices of data generating processes, it is useful to know that the physical process of tightening a bolt is typically associated with a linear relationship between angle and torque (at least in the relevant parts of the tightening process) - this approximation is good for the parts of the tightening process that are part of the "Endanzug". Therefore, simulations in the later parts of this notebook typically assume an approximately linear process as the data generating process for non-outliers.

One observation in a data set might look as follows.



Observation	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Angle	2.01	2.21	2.91	3.00	3.07	3.95	4.33	4.35	4.41	4.74	4.77	5.06	6.33	6.37	6.41	6.57	7.25	7.32	7.71	7.94
Torque	2.02	2.61	3.05	3.16	2.99	4.19	4.24	4.37	4.73	4.89	5.03	5.45	6.32	6.18	6.22	7.06	7.30	7.59	7.99	8.06

- The red diamonds represent measurements of torque that were taken at a recorded angle.
- The blue lines are an example of **linear interpolation** between the points that were actually measured. This will become important later on.
- The **measuring interval** marked in green is the convex hull of angles where measurements were taken for this observation.

The data set contains many of these objects, that do not necessarily share these characteristics.

[2]:
```
# Generated using function from auxiliary/observation_vis.R
# source("auxiliary/observation_vis.R")
# obs_vis()
```

The Algorithm

The idea of Febrero-Bande, Galeano, and Gonzàlez-Manteiga (2008) is an iterative process that classifies observations as outliers if their functional depth lies below a threshold C, which is determined using a bootstrapping procedure in each iteration. The algorithm can be decomposed into two parts:

1. **The iterative process**: (quoted from Febrero-Bande, Galeano, and Gonzàlez-Manteiga (2008))
    1. Obtain the functional depths $D_n(x_i), \ldots, D_n(x_n)$ for one of the functional depths [...]
    2. Let $x_{i_1}, \ldots, x_{i_k}$ be the k curves such that $D_n(x_{i_k}) \leq C$, for a given cutoff C. Then, assume that $x_{i_1}, \ldots, x_{i_k}$ are outliers and delete them from the sample.

3. Then, come back to step 1 with the new dataset after deleting the outliers found in step 2. Repeat this until no more outliers are found.

The underlying idea is that observations that are more central in the sense of the chosen depth will have higher depth assigned to them. So choosing the least deep observations from a data set is a reasonable way to search for atypical observations. The question of where to draw the border for observations to be classified as abnormal is done using a bootstrapping procedure described next.

2. **Determining C**: (quoted from Febrero-Bande, Galeano, and Gonzàlez-Manteiga (2008))
    1. Obtain the functional depths $D_n(x_i),\ \ldots,\ D_n(x_n)$ for one of the functional depths [...]
    2. Obtain B standard bootstrap samples of size n from the dataset of curves obtained after deleting the $\alpha\%$ less deepest curves. The bootstrap samples are denoted by x_i^b for $i = 1,\ \ldots,\ n$ and $b = 1,\ \ldots,\ B$.
    3. Obtain smoothed bootstrap samples $y_i^b = x_i^b + z_i^b$, where z_i^b is such that $(z_i^b(t_1), \ldots, z_i^b(t_m))$ is normally distributed with mean 0 and covariance matrix $\gamma\Sigma_x$ where Σ_x is the covariance matrix of $x(t_1), \ldots, x(t_m)$ and γ is a smoothing parameter. Let y_i^b, $i = 1,\ \ldots,\ n$ and $b = 1,\ \ldots,\ B$ be these samples. *
    4. For each bootstrap set $b = 1,\ \ldots,\ B$, obtain C^b as the empirical 1% percentile of the distribution of the depths $D(y_i^b)$, $i = 1,\ \ldots,\ n$.
    5. Take C as the median of the values of C^b, $b = 1,\ \ldots,\ B$.

This is done, as a theoretical derivation of the distribution of depth values for data generated by a specific data generating process is often infeasible. Instead, one drops a fraction of observations α and uses a smoothed bootstrapping algorithm to approximate the corresponding threshold value to remove approximately that fraction of observations in the procedure listed under 1.

*At this point in the algorithm the assumption that the functional observations are observed at a set of discrete points $t_1,\ \ldots,\ t_m$ has already been made.

Some important points:

- I decided to deviate from the original procedure proposed by the authors by allowing the user to decide whether to reestimate C in each iteration of the process. The authors present good arguments, why keeping C fixed is the better approach and my testing confirms those. But to keep the possibilities for experimentation open, I opted to implement it in this way nevertheless. In later stages of this project I will go into detail on why keeping C constant also has some problems and try to introduce corrections to improve the method developped below.
- The authors recommend a choice of $\gamma = 0.05$ as a smoothing parameter for the bootstrap which I adopted in my applications.
- For the choice of α preexisting information on the data should be used if available. A good way to choose this is the expected fraction of outliers in the data. However, my testing showed that in some cases the choice of this parameter had to be lower than the actual fraction of observations generated by abnormal processes when using a sampling procedure to get better results.

The authors propose three functional depth measures and benchmark them in a simulation setting. Because of their results and the computational cost which are comparatively small, I chose to use **h-modal depth** for my implementation, which I will introduce in the following.

h-modal depth

Introduced by Cuevas, Febrero-Bande, and Fraiman (2006) h-modal depth is one of three depth measures covered in Febrero-Bande, Galeano, and Gonzàlez-Manteiga (2008). I will follow the summary in the latter paper for my overview. The idea behind this depth is that a curve is central in a set of curves if it is closely surrounded by other curves.

In mathematical terms, the h-modal depth of a curve x_i in relation to a set of curves $x_1, \ldots, x_n$ is defined as follows:

$$MD_m(x_i, h) = \sum_{k=1}^{n} K(\frac{||x_i - x_k||}{h}) \quad (1.8)$$

where $K : \mathbb{R}^+ \to \mathbb{R}^+$ is a kernel function and h is a bandwidth. The authors recommend using the truncated Gaussian kernel, which is defined as follows:

$$K(t) = \frac{2}{\sqrt{2\pi}} \exp(-\frac{t^2}{2}) \quad \text{for} \quad t > 0 \quad \text{and} \quad 0 \quad \text{otherwise} \quad (1.9)$$

and to choose h as the 15th percentile of the empirical distribution of $\{||x_i - x_k|| \,|\, i, k = 1, \ldots, n\}$

I chose to implement the L^2 norm - one of the norms recommended by the authors - as it performed better than the L^∞ norm (which was also recommended) in my preliminary tests. In a functional setting L^2 is defined by:

$$||x_i - x_k||_2 = \sqrt{\int_a^b (x_i(t) - x_k(t))^2 dt} \quad (1.10)$$

where a and b are the boundaries of the measurement interval. This can be replaced by its empirical version

$$||x_i - x_k||_2 = \sqrt{\sum_{j=2}^{m} \Delta_j (x_i(t_j) - x_k(t_j))^2} = \sqrt{\sum_{j=2}^{m} (t_j - t_{j-1})(x_i(t_j) - x_k(t_j))^2} \quad (1.11)$$

in case of a discrete set of m observation points shared between observations.

The choice of the functional norm could be adjusted to deal with data resembling different functional forms. This could be part of a possible extension, where different norms are implemented and compared. This might become part of future iteratons of this project.

Difficulties due to the Data

The Endanzug does not start at the same angle for every observation.

In this setting the fact that the first measurement is taken at different angles is not of a problem, since the property of interest is the shape of the curve after the Endanzug has started. In real world terms, the beginning of the "Endanzug" is determined by the first angle where a specific torque is exceeded and the change in torque during the "Endanzug" is of interest and not the position of the "Endanzug" in the whole tightening process. Therefore, all observations can be modified by subtracting the first angle of the "Endanzug" from all angles, effectively **zeroing** the observations.

At this time, I am going to focus on problems where zeroing is admissible and elaborate on how to possibly extend the method to scenarios where it is not. Those generalizations will be part of further work on this project.

After zeroing, the measurement intervals might still not be identical due to differing lengths.

I decided to try to remedy this remaining problem in the zeroed data set by stretching the measuring intervals to a shared interval while leaving the observed torques identical. Excessive stretching however is problematic, as it can lead to similar observations ending up very different. Putting a conservative limit on stretching should however conserve the quality of relationships between observations. This should lead to limited influence on the calculated depths of the functional observations. To do so, I define a parameter $\lambda \geq 1$ called **acceptable stretching** and make observations comparable by stretching their measuring intervals by a factor $\psi_i \in [1/\lambda, \lambda]$ before approximating them using linear interpolation. In an optimal setting stretching would not be necessary and employing it leads to trade offs, so this parameter should be chosen conservatively.

If zeroing as described in 1. is not appropriate, a combination of **acceptable stretching** and **acceptable shifting** could be implemented to increase the size of sets of pairwise comparable functions. This will be part of future extensions of this project.

As you can already see in this animation, the acceptable stretching introduces inaccuracies even in an approximately linear setting. Outlier classifications could be quite sensitive to this parameter.

The angles where torque is measured are not shared between observations.

Assuming that the measuring intervals are identical, I decided to use **linear interpolation** to approximate the observations. This is done to treat them as if they had been observed at a **shared grid of angles** to make them compatible with the simplification of the **h-modal norm** described above. This is only an approximation, but choosing an appropriately fine grid to approximate the observations should limit the influence of this procedure on the calculated functional depths. Espacially in case of the dataset under consideration in this project, performing this approximation by linear interpolation should not result in large distortions due to the linearity of the described physical process. In other settings this approximation could lead to bad performance. One example that came to my mind is if most observations are zero, the frequency of taking measurements is comparatively low and the relevant parts of the observations are instantaneous deviations from zero (or spikes that have infinitesimally small duration). In cases like the one described, it would probably be a better idea to choose a different method or at least choose a different functional norm more appropriate for the dgp due to the necessity of very fine grids to achieve a appropriate approximation of the data. Benchmarking of the method for different data generating processes will be part of future revisions to explore the applicability of the developed method.

For sufficiently smooth processes with a high measuring frequency this approximation should however not result in huge distortions.

Another possibility to approach this third problem would be to use a different version of the norm for discretized points I described above. Instead of calculating

$$||x_i - x_k||_2 = \sqrt{\sum_{j=2}^{m} \Delta_j (x_i(t_j) - x_k(t_j))^2} (1.12)$$

for a set of points on a grid approximated by linear interpolation, one could instead define functions $\tilde{x}_i$ which are just the piece wise linear functions defined by connecting the observed points of x_i. A norm based on this could be constructed as:

$$||x_i - x_k||_{\tilde{2}} = \sqrt{\int_a^b (\tilde{x}_i(t) - \tilde{x}_k(t))^2 dt} (1.13)$$

For very fine grid approximations these criteria should result in similar depths, as under some mild conditions, the first approach will converge to the second for increasingly fine grids. In a sense the first approach is similar to a Riemann sum which for increasingly fine grids converges to the corresponding Riemann integral under the necessary assumptions.

Runtime Complexity

The runtime complexity of this algorithm is at least $O(n^2)$ and I concluded that using my implementation it is infeasible to use it on a very large data set such as the "Endanzugsproblem" (assuming that all observations are comparable at once). Even when splitting up the observations as proposed above into comparable subsets, some of them will be too large to directly approach with this method. To solve this problem, I instead opted to use a **sampling approach** which I explain in the next section.

[3]:
```
### The gifs have been rendered using function from auxiliary/observation_vis.R (needs␣
↪more packages, than are available in this environment)
# source("auxiliary/observation_vis.R")
# stretching_vis()
# zeroing_vis()
# lin_approx_vis()
```

Sampling Approach

Since it is infeasible to use this method on very large data sets at once, I chose to pursue a sampling-based approach instead. Intuitively this is reasonable as in many cases observations that are atypical in the full data set will be classified as atypical in its subsamples more often than "typical" observations. If this principle does not apply to the data set, a sampling-based approach is difficult to justify. In a case like that it is more reasonable to choose differnet methods to identify abnormal observations.

As the assumption seems reasonable in case of the "Endanzugsproblem", instead of performing the algorithm described above on the whole data set (or its comparable subsets), I devise a procedure as described in the following:

Let $\{x_1, \ldots, x_L\}$ be a set of observations that are comparable using the algorithm but too large to perform this procedure in reality.

1. Define the following objects:
   - Let *num_samples*

     $$= (a_1, \ldots, a_L) \in \mathbb{N}_0^L$$

     where a_i is the number of samples x_i was part of $\quad \forall i \in \{1, \ldots, L\}$. (Initialize all entries as 0)
   - Let *num_outliers*

     $$= (b_1, \ldots, b_L) \in \mathbb{N}_0^L$$

     where b_i is the number of samples x_i was classified as an outlier in

     $$\forall i \in \{1, \ldots, L\}$$

     . (Initialize all as entries 0)
   - Let *frac_outliers*

     $$= (c_1, \ldots, c_L) \in \mathbb{R}_{\geq 0}^L$$

     where $c_i = \begin{cases} 1 & a_i = 0 \\ \frac{b_i}{a_i} & a_i > 0 \end{cases} \quad \forall i \in \{1, \ldots, L\}$ (Initialize all entries as 1)
2. Draw a sample of size K from $\{x_1, \ldots, x_L\}$ without replacement
3. Perform the outlier detection procedure on this sample and update the vectors according to your results.
4. Go back to two and iterate this process until some condition is fulfilled.

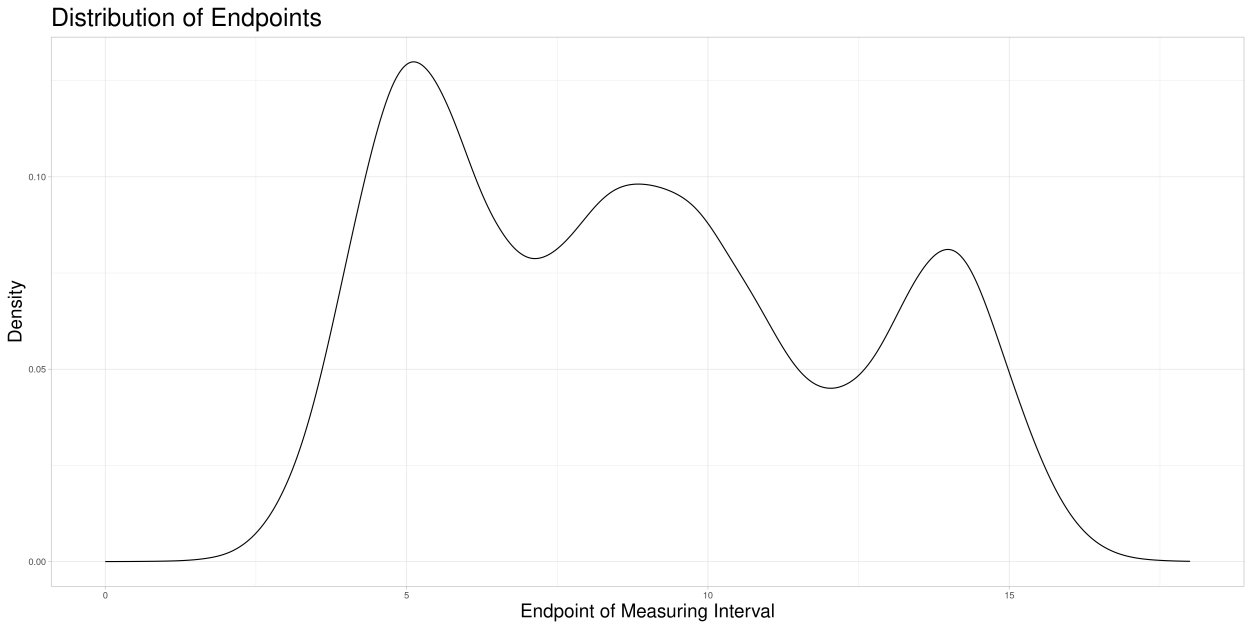
Typical conditions could be:

- A specified number of iterations was reached
- Every observation was part of more than a specified number of samples
- The vector of certainties did not change enough according to some criterion over a specified number of iterations

In the end, the entries of *frac_outliers* can be used as a metric for the outlyingness of an observation. If a binary decision rule is needed, every observation with an entry over some specified threshold could be classified as an outlier.

Finding Comparable Sets of Observations

Assume for the sake of simplicity that **zeroing** is reasonable so that the minima of the measurement intervals of the observations are all zero. Therefore, the measurement intervals only differ in their end points or lengths which is identical in this case. As an example to visualize possible ways to find comparable subsets assume that the empirical distribution of endpoints looks as follows:



In the following I describe three methods to find comparable subsets to perform the sampling procedure on and explain why I chose the method that I ultimately implemented.

Static Splitting

For **Static Splitting** the idea is to partition the whole data set into pairwise disjunct subsets of pairwise comparable observations. This partition depends on the acceptable stretching parameter and is not necessarily unique for a choice of the acceptable stretching parameter. Therefore, a choice procedure would have to be introduced if this approach were to be taken. Some possible partitions of the set above (not necessarily consistent with the same acceptable stretching parameter) could look like this:

However, this idea has some problems:

- The choice of subsets could introduce a new source of distortions in addition to the acceptable stretching parameter.
- Adding new observations could change the chosen subsets, making an updating procedure difficult to realize.
- In each individual subset, the observations that are changed due to stretching are identical over samples. This could lead to distortions since for some observations not the original but only the stretched observations are taken into account in the classification.

Dynamic Splitting

In **Dynamic Splitting** the allocation of comparable subsets takes place dynamically for each realization of the observation interval. The procedure is as follows:
For each realization of the endpoint, determine the subset of comparable observations and perform the sampling approach on this subset but keep the parameter for acceptable stretching constant for all end points.

This approach has some advantages over the first one:

- The choice of subsets becomes only a question of the acceptable stretching parameter and not of the choice of algorithm that chooses the partition of the data set.
- Adding new observations is unproblematic, as new observations do not influence the allocation of comparable subsets. Therefore additional samples containing the new observation can be drawn without creating problems in the comparability to previous sampling-based results. A procedure like this is described in the next section.
- Each observation can enter the classification procedure undistorted in at least the comparable subset corresponding to its own endpoint. Additionally, it can enter the classification in samples, where it is comparable due to acceptable stretching. The latter can realize for different degrees of stretching - increasing or decreasing the length of the measuring interval. This solves the problem of observations being used for classification only in a specific distorted state.

Dynamic Splitting with varying acceptable stretching parameter

As shown above, the difference in length of these comparable subsets changes quite substantially and does not react to the density of observations having similar measurement intervals. A deviation from this appraoch might be desirable for multiple reasons:

- If there are many observations with a very similar measuring interval, including observations with a more dissimilar interval (in terms of the animation above, a midpoint farther away) might be detrimental to the procedure's performance. If there are fewer observations close by one might want to allow for a bigger acceptable stretching parameter to allow for more comparisons.
- In tandem with the advantage above one could also change the sample size to suit specific needs of the procedure.

It would be possible to use a **local acceptable stretching parameter** that changes as a function of the estimated density of end points (since zeroing was admissible in this example). Using a rather simple function determining the local acceptable stretching parameter to serve as an example leads to the following choice of comparable subsets.

In this example the effect is quite subtle, but in comparison to the previous animation, one can see that the expansion of the interval of end points of comparable observations is slower in regions, where the estimated density of end points is higher. There are two reasons why I decided against making acceptable stretching a varying parameter in my implementation:

1. It introduces another complication as the function to determine the local acceptable stretching has to be chosen.
2. It makes the updating procedure described later more difficult, as adding more observations will change the estimated density of the end points and thereby change the comparable subsets.

Choice for Implementation

Due to the advantages and disadvantages described above, I decided to implement Dynamic Splitting with a global acceptable stretching parameter. This makes it easier to later justify an updating procedure and still does not create distorted results as described for Static Splitting.

[4]:
```
### The graphics and gifs have been rendered using function from auxiliary/observation_
↪vis.R (needs more packages, than are available in this environment)
# source("auxiliary/distribution_vis.R")
# dist_vis()
# static_splits_vis()
# dynamic_splits_vis()
# dynamic_splits2_vis()
```

Description of Full Procedure for Existing Data sets

Having explained

- The algorithm by Febrero-Bande, Galeano, and Gonzàlez-Manteiga (2008)
- The sampling approach
- The procedure for selecting comparable subsets with dynamic splitting of the data set

I am going to explain the full procedure applied to an existing data set assuming that zeroing is admissible.

Definition of Objects

- Let $x_1, \ldots, x_n$ be the full data set of observations as described in the beginning
- Let $I_1 = [s_1, e_1], \ldots, I_n = [s_n, e_n]$ be the measuring intervals of $x_1, \ldots, x_n$

Assuming that zeroing is admissible:

- Let $\bar{x}_1, \ldots, \bar{x}_n$ be the corresponding zeroed observations
- Let $\bar{I}_1 = [0, \bar{e}_1], \ldots, \bar{I}_n = [0, \bar{e}_n]$ be the corresponding zeroed measuring intervals
- Let $\bar{E} = \{\bar{e} \mid \exists k \in \{1, \ldots, n\} \text{ s.t. } \bar{e} = \max(\bar{I}_k)\}$ be the set of measuring interval end points occurring in $\bar{I}_1, \ldots, \bar{I}_n$
- Let $\bar{\Lambda}(\bar{x}, \bar{e})$ be the resulting object when zeroed observation $\bar{x}$ is stretched to fit zeroed measuring interval $\bar{I} = [0, \bar{e}]$

Again assuming that zeroing is admissible, one can reasonably define the following objects.

- Let $\bar{Z}(\bar{e}, \lambda) = \{\bar{x}_k \mid \frac{\bar{e}_k}{\bar{e}} \in [\frac{1}{\lambda}, \lambda]\}$ be the set of zeroed observations that can be compared to a zeroed observation with measuring interval $\bar{I} = [0, \bar{e}]$ with an acceptable stretching factor of λ.
- Let $\bar{S}(\bar{e}, \lambda) = \{\bar{\Lambda}(\bar{e}, \bar{x}) \mid \bar{x} \in \bar{Z}(\bar{e}, \lambda)\}$ be the set of zeroed observations that have been stretched with an admissible stretching factor to be comparable to a zeroed observation with zeroed measuring interval $\bar{I} = [0, \bar{e}]$.

If zeroing is not a valid approach, corresponding objects dependent on acceptable shifting and acceptable stretching have to be defined, which will be one challenge in generalizing this method.

Procedure

1. Choose parameters:
   - λ acceptable stretching
   - L sample size in sampling procedure (may also be varying depending on chosen approach for sampling)*
   - K number of equidistant points in grid for approximation by linear interpolation*
   - α fraction of observations to drop in approximation of cutoff value C in outlier classification algorithm*
   - B sample size in approximation of cutoff value C in outlier classification algorithm*
   - γ tuning parameter in approximation of cutoff value C in outlier classification algorithm*
2. Initialize the following vectors:

- *num_samples*

$$= (a_1, \ldots, a_n) \in \mathbb{N}_0^n$$

  with all entries being 0

- *num_outliers*

$$= (b_1, \ldots, b_n) \in \mathbb{N}_0^n$$

  with all entries being 0

- *frac_outliers*

$$= (c_1, \ldots, c_n) \in \mathbb{R}_{\geq 0}^n$$

  with all entries being 1

3. Iterate through the elements of $\bar{E}$ doing the following:
   - Let $\hat{e}$ be the element of $\bar{E}$ currently looked at
   - Determine $\bar{S}(\hat{e}, \lambda)$ and perform the sampling based outlier identification procedure on this set for some predetermined stopping condition
   - Update *num_samples*, *num_outliers*, and *frac_outliers* for both the stretched and non-stretched observations in $\bar{S}(\hat{e}, \lambda)$ as described in the section about sampling
   - Go to the next element of $\bar{E}$ and repeat until all elements have been reached.
4. Report *frac_outliers* as a measure of outlyingness for the observations.

*not explicitly mentioned in the procedure below

Updating

The procedure described above is constructed to work for a full data set. In a day-to-day setting the data set will not be static. Instead, new observations will be added and it would be a problem, if all calculations had to be done all over again only to incorporate a comparatively small number of new observations.

Instead, I devise mechanisms to assign comparable values of *frac_outliers* to the newly added observations and to possibly also update the pre-existing observations due to the presence of the newly added ones. In the following assume that new observations are added sequentially. Two ways came to my mind to approach this updating procedure, one more true to the original process (1) and the other one less computationally expensive (2).

Version 1:

This procedure involves additional samples from all sets the new observation could have been part of. So both in a stretched form or in its original form

- Let x' be the new observation and $\bar{x}'$ its zeroed counterpart. Define I', $\bar{I}'$ and $\bar{e}'$ accordingly.
- Determine all elements $\bar{e} \in \bar{E}$ such that $\bar{x}' \in \bar{Z}(\bar{e}, \lambda)$. Define $\bar{U}(\bar{x}', \lambda) = \{\bar{e} \in \bar{E} \mid \bar{x}' \in \bar{Z}(\bar{e}, \lambda)\}$ as the subset of $\bar{E}$ called the Updating Window. In this setting where zeroing is admissible, this can be simplified to $\bar{U}(\bar{x}', \lambda) = \{\bar{e} \in \bar{E} \mid \bar{e} \in [\frac{1}{\lambda}\bar{e}', \lambda\bar{e}']\} = \bar{E} \cap [\frac{1}{\lambda}\bar{e}', \lambda\bar{e}']$

For all $\tilde{e} \in \bar{U}(\bar{x}', \lambda)$ perform a sampling procedure as follows:

- $\bar{\Lambda}(\bar{x}', \tilde{e})$ is part of each sample
- The size of each sample is identical to the one used in the original procedure
- The number of samples for each $\tilde{e}$ is the expected value of the number of samples the new observation would have been part of, if it had been in the original data set
- The updating procedure of the vectors works as before. Not only the entry of the new observation in each vector is added and updated, also the entries for the original observations included in the new samples are updated.

So the set of zeroed observations that is potentially updated during this procedure is the following:

$$\bigcup_{\tilde{e} \in \bar{U}(\bar{x}', \lambda)} \bar{Z}(\tilde{e}, \lambda) = \left\{ \bar{x} \mid \bar{e} \in [\frac{1}{\lambda} \min\{\bar{U}(\bar{x}', \lambda)\}, \lambda \max\{\bar{U}(\bar{x}', \lambda)\}] \right\}$$

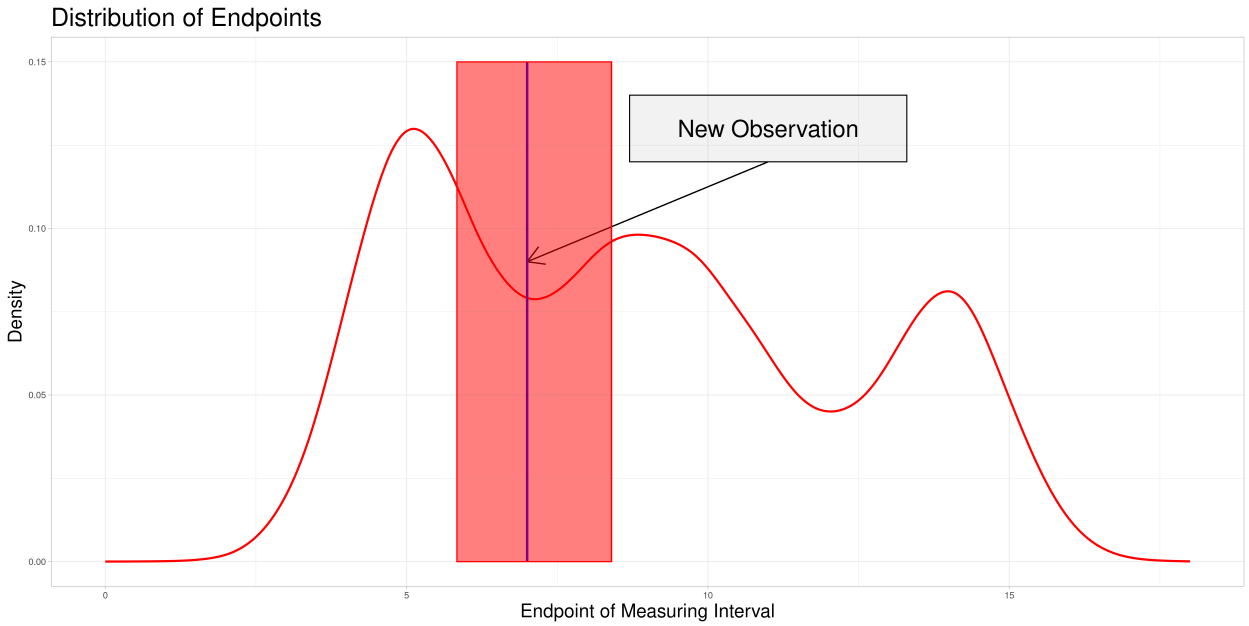
Which is visualized in the following animation. (The endpoints of the potentially updated zeroed observations are marked by the red rectangle.)

Version 2:

In comparison the other procedure involves only additional sampling from the set where the new observation is non-stretched.

- Let x' be the new observation and $\bar{x}'$ its zeroed counterpart. Define I', $\bar{I}'$ and $\bar{e}'$ accordingly.
- Determine $\bar{S}(\bar{e}', \lambda)$ and perform additional sampling as follows:
  - $\bar{x}'$ is part of each sample
  - The number of samples drawn is the expected value of samples the new observation would have been part of, if it had been in the original data set
  - The updating procedure works as before. Not only the entry of the new observation in each vector is added and updated, also the entries for the original observations are updated.

The following graphic shows the equivalent objects of version 1:



This procedure is less true to the values calculated in the original data set, as the new observation is only taken into consideration in its non-stretched form. Additionally, pre-existing observations could be taken into consideration in their stretched form more frequently depending on the structure of new data being added which could lead to additional distortions. Therefore, I decided to start by implementing the first method and possibly include comparisons of both methods in future revisions.

[5]:
```
### The graphics and gifs have been rendered using functions from auxiliary/updating_vis.
↪R (needs more packages, than are available in this environment)
# source("auxiliary/updating_vis.R")
# upd_1_vis()
# pot_upd_obs_vis()
# upd_2_vis()
```

Implementation

I decided to implement these methods in **R** and **C++** and to employ parallelization where possible, to strike a balance between speed and ease of use. **C++** functions referenced in the following can be found in */auxiliary/rcpp_functions.cpp*. The following section will follow a similar structure as the description of the procedure above ordered as follows:

1. *Grid Approximation*
2. *Febrero-Bande, Galeano, and Gonzàlez-Manteiga (2008) for Observations on a Shared Grid*
3. *Sampling approach*
4. *Dynamic Splitting and Finding Comparable Subsets*
5. *Full Procedure*
6. *Updating*

All of these functions can also be found in */auxiliary/R_functions.R* and will in the future be available as an Rcpp-package the tar-ball of which will be included in the repo. In future revisions there will be a secondary notebook with the details of the implementation. This notebook will instead focus on explaining the method and evaluating its perfomance.

Grid Approximation

Finding a grid for Approximation

Assuming that the observations already share the measuring intervals. (Same grid length still acts as a placeholder for a more sophisticated procedure.)

[6]:
```
# func_dat:

grid_finder <- function(func_dat){
  measuring_interval <- c(min(func_dat[[1]]$args), max(func_dat[[1]]$args))
  return(seq(measuring_interval[1], measuring_interval[2], length.out = 100))
}
```

Grid approximation by Linear Interpolation

This function acts as a wrapper for a **C++** function, that given a vector of arguments, a vector of values and a vector that represents the grid to be used for approximation, performs the desired approximation and returns the vector of approximated values taken at the grid points. This wrapper then combines those values to a matrix, where each row represents the approximated values of an observation at the given grid points.

[7]:
```
# func_dat: list that contains the observations
# each observation is a list, that conatins two vectors of identical length: args and␣
↪vals
# grid: grid to use for approximation

grid_approx_set_obs <- function(func_dat, grid) {
  res_mat <- matrix(data = unlist(
      map(.x = func_dat,
          .f = function(obs) grid_approx_obs(obs$args, obs$vals, grid))
    ), nrow = length(func_dat), byrow = TRUE)

  return(res_mat)
}
```

Febrero-Bande, Galeano, and Gonzàlez-Manteiga (2008) for Observations on a Shared Grid

The following functions implement the algorithm described above for functional observations that are observed at a common set of discrete points. The grid approximation above serves as a preparation to make these functions applicable.

approx_C

The function *approx_C* implements the approximation of the cutoff value C by bootstrapping described in Febrero-Bande, Galeano, and Gonzàlez-Manteiga (2008).

[8]:
```
# matr_dat: data in matrix form - each row contains the grid approximations of one␣
↪observation
# fdepths: corresponding depths for the observations
# alpha: quantile of least deep observations to drop before bootstrapping
```

(continues on next page)

(continued from previous page)

```
# B: number of smoothed bootstrap samples to use
# gamma: tuning parameter for smoothed bootstrap
# grid: grid used in approximation of matr_dat

approx_C <- function(matr_dat, fdepths, alpha, B, gamma, grid) {

  # infer number of observations from length of depth vector
  n <- length(fdepths)
  # Get number of elements in grid
  grid_length <- length(grid)
  # determine threshold to drop observations with lowest depth values
  depth_thr <- quantile(x = fdepths, probs = alpha)
  # drop observations for bootstrapping
  matr_dat_red <- matr_dat[fdepths >= depth_thr, ]
  n_red <- dim(matr_dat_red)[1]

  # Determine vcov-matrix for smoothed bootstrapping
  Sigma_x <- cov(matr_dat_red)
  my_vcov <- gamma*Sigma_x

  # Draw bootstrap samples from data set
  fsamples <- map(.x = 1:B,
                  .f = function(inds) matr_dat_red[sample(x = 1:n_red, size = n, replace␣
↪= TRUE), ])

  # Create smoothing components for bootstrapping
  smoothing_components <- map(.x = 1:B,
                              .f = function(x) mvrnorm(n = n, mu = rep(0, times = grid_
↪length), Sigma = my_vcov))

  # Obtain smoothed bootstrap samples
  smoothed_BS_samples <- map(.x = 1:B,
                             .f = function(b) fsamples[[b]] + smoothing_components[[b]])

  # Calculate depths for each smoothed bootstrap sample
  bootstrap_depths <- map(.x = 1:B,
                          .f = function(b) hM_depth(smoothed_BS_samples[[b]], grid))

  # Calculate first percentile from depths of smoothed bootstrap samples
  one_perc_quantiles <- unlist(map(.x = bootstrap_depths,
                                   .f = function(sample) quantile(sample, probs = 0.01)))

  # return median of first percentiles
  return(median(one_perc_quantiles))
}
```

outlier_iteration

This function performs one iteration of the algorithm, including the calculation of functional depths, the approximation of C and the flagging of observations with depths lower than C

[9]:
```
# matr_dat: data in matrix form - each row contains the grid approximations of one␣
↪observation
# alpha: quantile of least deep observations to drop before bootstrapping (in␣
↪approximation of C - optional if C is specified)
# B: number of smoothed bootstrap samples to use (in approximation of C - optional if C␣
↪is specified)
# gamma: tuning parameter for smoothed bootstrap
# ids: identifiers of individual observations
# grid: grid used in approximation of matr_dat
# C: should be provided. Otherwise C will be approximated in each step of the iteration

outlier_iteration <- function(matr_dat, alpha = 0.05, B = 50, gamma, ids, grid, C = NULL)
↪{

  # Calculating functional depths using a function from ./auxiliary/Rcpp_functions.cpp
  fdepths <- hM_depth(matr_dat, grid)

  if(missing(C)){
      # Approximating C
      C <- approx_C(matr_dat = matr_dat, fdepths = fdepths, alpha = alpha,
                    B = B, gamma = gamma, grid = grid)
  }

  # Flagging observations with depths lower than the cutoff value C
  outliers <- which(fdepths < C)

  return(list(matr_dat = matr_dat[-outliers, ],
              ids = ids[-outliers],
              outlier_ids = ids[outliers]))
}
```

outlier_detection This function serves as a wrapper for the previous function and iterates the process until no new observations are flagged.

[10]:
```
# matr_dat: data in matrix form - each row contains the grid approximations of one␣
↪observation
# alpha: quantile of least deep observations to drop before bootstrapping (in␣
↪approximation of C - optional if C is specified)
# B: number of smoothed bootstrap samples to use (in approximation of C - optional if C␣
↪is specified)
# gamma: tuning parameter for smoothed bootstrap
# ids: identifiers of individual observations
# grid: grid used for the approximation
# C: should be provided. Otherwise C will be approximated in each step of the iteration

outlier_detection <- function(matr_dat, alpha = 0.05, B = 100, gamma = 0.05, ids, grid,␣
↪C = NULL){
```

(continues on next page)

(continued from previous page)

```
    tmp_ids <- ids
    # Initialize empty vectors for position of flagged observations in func_dat and ids␣
↪of flagged observations
    outlier_ids <- c()

    # loop that continues until an iteration does not flag any new observations
    condition <- TRUE
    while(condition){

        # perform iteration
        iter_res <- outlier_iteration(matr_dat = matr_dat, alpha = alpha, B = B, gamma =␣
↪gamma, ids = tmp_ids, grid = grid, C = C)
        new_outliers <- iter_res$outlier_ids

        # if there are no new flagged observations stop loop
        if(length(new_outliers) == 0){condition <- FALSE}
        else{
          #otherwise: add flagged observations to vector
          outlier_ids <- c(outlier_ids, new_outliers)
          # reduce data to non-flagged observations
          matr_dat <- iter_res$matr_dat
          # reduce ids to non-flagged observations
          tmp_ids <- iter_res$ids
        }
    }

    # return identifiers of flagged observations and position of these flagged␣
↪observations in the data set
    return(list(outlier_ids = outlier_ids,
                outlier_ind = which(is.element(ids, outlier_ids))))
}
```

Outlier Detection - Wrapper

The following function acts as a wrapper to the previous one in case, C should not be recalculated in each iteration. My recommendation would be to use this function as recalculation of C in each iteration can lead to classifying unreasonably many observations as outliers.

[11]:
```
# func_dat: list that contains the observations
# each observation is a list, that contains two vectors of identical length: args and␣
↪vals
# ids: identifiers of individual observations
# alpha: quantile of least deep observations to drop before bootstrapping (in␣
↪approximation of C - optional if C is specified)
# B: number of smoothed bootstrap samples to use (in approximation of C - optional if C␣
↪is specified)
# gamma: tuning parameter for smoothed bootstrap

detection_wrap <- function(func_dat, ids, alpha, B, gamma = 0.05){

    # determine the grid for approximation
```

(continues on next page)

(continued from previous page)

```
    grid <- grid_finder(func_dat)

    # Approximate by linear interpolation
    matr_dat <- grid_approx_set_obs(func_dat, grid)

    # calculate h-modal depths
    fdepths <- hM_depth(matr_dat, grid)

    # Approximate a value of C
    C_appr <- approx_C(matr_dat = matr_dat, fdepths = fdepths, alpha = alpha, B = B,␣
↪gamma = gamma, grid = grid)

    # Perform the outlier classification procedure for the approximated value of C
    flagged <- outlier_detection(matr_dat = matr_dat, ids = ids, grid = grid, C = C_appr)

    # Return the list of outlier ids and outlier indices - these are useful in different␣
↪cases
    return(flagged)
}
```

Sampling Approach

Helper function for parallelization

Since data sets can get large very quickly, it is useful to perform parallelization in this sampling approach in a less memory intensive way. Therefore I decided to write the data set in its prepared form to the disk and use a format that supports random access in lists, to only read in the observations that are part of the sample. After reading in those observations, it performs the outlier detection procedure implemented above.

[12]:
```
# list_path: path to the random access list of the data set (generated by package␣
↪largeList)
# index: index of observations to use in the procedure
# alpha: quantile of least deep observations to drop before bootstrapping (in␣
↪approximation of C)
# B: number of smoothed bootstrap samples to use (in approximation of C)
# gamma: tuning parameter for smoothed bootstrap

random_access_par_helper <- function(list_path, ids, index, alpha, B, gamma){

    # read in the observations identified by the variable index
    func_dat <- readList(file = list_path, index = index)

    # perform the outlier detection procedure on the sample
    # in a tryCatch statement as the procedure creates notamatrix errors in random cases
    sample_flagged <- tryCatch(
        {detection_wrap(func_dat = func_dat, ids = ids, alpha = alpha, B = B, gamma =␣
↪gamma)},
         error=function(cond){
             return(list(outlier_ids = c(), outlier_ind = c()))}
    )
```

(continues on next page)

(continued from previous page)

```
    # return the object generated by the outlier detection procedure
    return(sample_flagged$outlier_ids)
}
```

[13]:
```
# cl: cluster object generated by parallel package
# n_obs: number of observations in data set
# n_samples: number of samples to use
# sample_size: number of observations to use in each sample
# alpha: quantile of least deep observations to drop before bootstrapping (in␣
↪approximation of C)
# B: number of smoothed bootstrap samples to use (in approximation of C)
# gamma: tuning parameter for smoothed bootstrap
# list_path: path to the random access list of the data set (generated by package␣
↪largeList)

sampling_wrap <- function(cl, n_obs, n_samples, sample_size, alpha, B, gamma, list_path){

    ids <- 1:n_obs

    # Initialize vectors described in the theoretical section
    num_samples <- rep(x = 0, times = n_obs)
    num_outliers <- rep(x = 0, times = n_obs)
    frac_outliers <- rep(x = 1, times = n_obs)

    # Draw indexes for sampling from functional data without replacement
    sample_inds <- map(.x = 1:n_samples,
                       .f = function(i) sample(x = ids, size = sample_size, replace =␣
↪FALSE))

    # Determine how often each observation appeared in the samples and update the vector
    freq_samples <- tabulate(unlist(sample_inds))
    num_samples[1:length(freq_samples)] <- num_samples[1:length(freq_samples)] + freq_
↪samples

    # Perform the outlier classification procedure on the chosen samples parallelized
    # with the function clusterApplyLB() from the parallel package
    sample_flagged_par <- clusterApplyLB(cl = cl,
                                         x = sample_inds,
                                         fun = function(smpl){
                                             random_access_par_helper(list_path = list_
↪path, ids = ids[smpl],
                                                                                index = smpl,␣
↪alpha = alpha,
                                                                                B = B, gamma =␣
↪gamma)})

    # Determine how often each observation were flagged in the samples and update the␣
↪vector
    freq_outliers <- tabulate(unlist(sample_flagged_par))
    num_outliers[1:length(freq_outliers)] <- num_outliers[1:length(freq_outliers)] +␣
↪freq_outliers
```

(continues on next page)

(continued from previous page)

```
    # termine fraction of samples each observation was flagged as an outlier in
    certainties <- unlist(map(.x = 1:n_obs,
                              .f = function(i) ifelse(num_samples[i] != 0, num_
↪outliers[i]/num_samples[i], 1)))

    # Return list containing the three central vectors: num_samples, num_outliers,␣
↪certainties
    return(list(num_samples = num_samples,
                num_outliers = num_outliers,
                certainties = certainties))
}
```

How to use this function?

Using the function *sampling_wrap()* is not as straight-forward as using the previous as multiple steps have to be performed before and after using this function to ensure a problem free experience. I will explain the following code fragments in detail, as depending on which operating system a user employs modifications have to be made. For this reason, the code is commented out in these parts, in order not to cause technical problems that are difficult to replicate. The following code segments were written on a Linux machine and will work on UNIX systems. Since forking is not supported under windows, alternatives would have to be used, like using PSOCK-Clusters instead of ForkClusters.

[14]:
```
# num_cores <- detectCores()
# cl <- makeForkCluster(num_cores)
#
#
# invisible(clusterCall(cl, fun = function() library('largeList')))
# invisible(clusterCall(cl, fun = function() library('Rcpp')))
# invisible(clusterCall(cl, fun = function() library('purrr')))
# invisible(clusterCall(cl, fun = function() library('MASS')))
# invisible(clusterCall(cl, fun = function() sourceCpp('auxiliary/rcpp_functions.cpp')))
#
# clusterExport(cl, varlist = list("grid_approx_set_obs",
#                                  "approx_C", "grid_finder",
#                                  "outlier_iteration", "outlier_detection",
#                                  "detection_wrap", "random_access_par_helper"),
#                                  envir = .GlobalEnv)
#
# sampling_wrap(cl = cl, n_obs = n_obs, n_samples = n_samples,
#               sample_size = sample_size, alpha = alpha, B = B, gamma = gamma,
#               list_path = list_path)
#
# stopCluster(cl)
```

- *num_cores()* detects the number of logical cores
- *makeForkCluster()* creates a virtual cluster object that can serve as an argument to functions from the parallel package to perform parallelized computations
- the *clusterCall()* calls execute the command inside on each individual node of the virtual cluster to ensure that all necessary packages are loaded in each instance
- *clusterExport* this exports objects from an environment to each of the nodes. These can be functions or objects.

(These last to steps are technically not necessary in case of a fork cluster, but will navigate around some hard to troubleshoot problems that can occur.)

- *sampling_wrap()* performs the actions implemented above on the virtual cluster *cl*
- *stopCluster()* stops the cluster, to ensure that it does not clog up the system

Dynamic Splitting

Zeroing

To use the dynamic splitting procedure in the previously described settings, it is first necessary to zero all observations. This is implemented for the functional observations and its results should be saved as a separate object for future use.

[15]:
```
# func_dat: list that contains the observations
# each observation is a list, that contains two vectors of identical length: args and␣
↪vals

zero_observations <- function(func_dat){
    zeroed_func_dat <- map(.x = func_dat,
                           .f = function(fnc){
                               args = fnc$args - fnc$args[1]
                               return(args = args, vals = fnc$vals)
                           })
    return(zeroed_func_dat)
}
```

Determine measuring intervals

This function returns a matrix containing in each row the beginning and end point of the measuring interval of the corresponding observation.

[16]:
```
# func_dat: list that contains the observations
# each observation is a list, that conatins two vectors of identical length: args and␣
↪vals

measuring_int <- function(func_dat){
    intervals <- matrix(data = unlist(map(.x = func_dat,
                                          .f = function(obs) c(min(obs$args), max(obs
↪$args)))),
                        nrow = length(func_dat), byrow = TRUE)

    return(intervals)
}
```

Create a list of measuring intervals that occur in the data set

This set is needed later on for iterating through all possible realizations of the measuring interval to determine the comparable sets for each one.

[17]:
```
# measuring_intervals: use output from measuring_int()

unique_intervals <- function(measuring_intervals){

    # for finding unique entries transforming to a list is easier
    interval_list <- map(.x = seq_len(nrow(measuring_intervals)),
                         .f = function(i) measuring_intervals[i,])

    # find unique entries
    unique_intervals <- unique(interval_list)

    # combine into matrix again
    unique_matrix <- matrix(data = unlist(unique_intervals),
                            nrow = length(unique_intervals),
                            byrow = TRUE)

    # return matrix where each row contains the beginning and end points of a unique␣
↪measuring interval
    # from the data set
    return(unique_matrix)
}
```

Find comparable observations for one measuring interval

Given a measuring interval that is currently under consideration and the matrix of all measuring intervals, determine the indices of the observations that are comparable given an acceptable stretching factor λ. Here zeroing is assumed, such that the condition simplifies to a condition on the endpoint of the intervals.

[18]:
```
# main_interval: vector of two elements: starting and end point of measuring interval
# measuring_intervals: use output from measuring_int()
# lambda: acceptable stretching parameter
# ids: identifiers of individual observations

comparable_obs_finder <- function(main_interval, measuring_intervals, lambda, ids){

    # Determine comparable observations by checking interval endpoints
    comparable <- which(measuring_intervals[,2] >= main_interval[2]/lambda
                        & measuring_intervals[,2] <= main_interval[2]*lambda)

    # Return the correspoding indices and the ids of the comparable observations
    return(list(ind = comparable,
                ids = ids[comparable]))
}
```

This function can then be used for determining which sets to sample from using the sampling procedure implemented above while iterating through the measuring intervals that occur in the data set.

Full Procedure

Stretching an observation

The first function needed for the full procedure is the ability to stretch an observation to be comparable to another.

[19]:
```
# obs: a list that conatins two vectors of identical length: args and vals
# measuring_interval: a vector with 2 elements, the start and end points of the desired␣
↪measuring interval

stretch_obs <- function(obs, measuring_interval){

    # calculate stretching factor
    phi <- (measuring_interval[2] - measuring_interval[1]) / (max(obs$args) - min(obs
↪$args))

    # stretch arguments by appropriate factor
    args_stretched <- obs$args * phi

    # return in the format for functional observations
    return(list(args = args_stretched,
                vals = obs$vals))
}
```

Stretching a set of observations to prepare sampling procedure

This function acts as a wrapper for the previous function and applies it to a set of observations that are stretched to the same measuring interval.

[20]:
```
# func_dat: list that contains the observations
# each observation is a list, that conatins two vectors of identical length: args and␣
↪vals
# measuring_interval: a vector with 2 elements, the start and end points of the desired␣
↪measuring interval

stretch_data <- function(func_dat, measuring_interval){

    # apply function stretch_obs() to each observation in the data set
    stretch_dat <- map(.x = func_dat,
                       .f = function(obs) stretch_obs(obs = obs, measuring_interval =␣
↪measuring_interval))

    # return list of stretched observations
    return(stretch_dat)
}
```

Performing stretching and sampling procedure on set of observations

This is nearly identical to the functions for the sampling procedure itself and could easily be included as a subcase. For the sake of clarity, I nevertheless decided to make these separate functions. Understanding the sampling procedure and the stretching functions will make these functions easy to understand.

[21]:
```
# list_path: path to the random access list of the data set (generated by package␣
↪largeList)
# index: index of observations to use in the procedure
# alpha: quantile of least deep observations to drop before bootstrapping (in␣
↪approximation of C)
# B: number of smoothed bootstrap samples to use (in approximation of C)
# gamma: tuning parameter for smoothed bootstrap
# measuring_interval: a vector with 2 elements, the start and end points of the desired␣
↪measuring interval

random_access_par_stretch_helper <- function(list_path, ids, index, alpha, B, gamma,␣
↪measuring_interval){

    # read in the observations identified by the variable index
    func_dat <- stretch_data(func_dat = readList(file = list_path, index = index),
                             measuring_interval = measuring_interval)

    # perform the outlier detection procedure on the sample
    # in a tryCatch statement as the procedure creates notamatrix errors in random cases
    sample_flagged <- tryCatch(
        {detection_wrap(func_dat = func_dat, ids = ids, alpha = alpha, B = B, gamma =␣
↪gamma)},
         error=function(cond){
             return(list(outlier_ids = c(0), outlier_ind = c(0)))}
    )

    # return the object generated by the outlier detection procedure
    return(sample_flagged$outlier_ids)
}
```

This function performing the parallelized sampling is again very similar. It only differs in four points:

- argument measuring_interval: interval the observations are stretched to
- argument comparable: ids of the observations to include in the process as now not all observations are to be sampled in the same process
- no argument n_obs: this can be replaced by comparable
- use of the appropriately changed helper function for the parallelization

Be careful when using the following function. The vectors that are returned only relate to the set of comparable observations currently under consideration. A value of 3 in the first entry of *num_samples* means that the first observation of the whole data set occurred in 3 samples, not that the first observation from the set of comparable observations occurred in 3 samples. This has to be kept in mind while later using the output of this function.

[22]:
```
# cl: cluster object generated by parallel package
# n_samples: number of samples to use
# sample_size: number of observations to use in each sample
# alpha: quantile of least deep observations to drop before bootstrapping (in␣
↪approximation of C)
```

(continues on next page)

(continued from previous page)

```
# B: number of smoothed bootstrap samples to use (in approximation of C)
# gamma: tuning parameter for smoothed bootstrap (in approximation of C)
# list_path: path to the random access list of the data set (generated by package␣
↪largeList)
# measuring_interval: a vector with 2 elements, the start and end points of the desired␣
↪measuring interval
# comparable: vector with the indices of comparable observations in the largelist

stretch_and_sample <- function(cl, n_samples, sample_size, alpha, B, gamma, list_path,␣
↪measuring_interval, comparable, n_obs){

    ids <- comparable

    # Initialize vectors described in the theoretical section
    num_samples <- rep(x = 0, times = n_obs)
    num_outliers <- rep(x = 0, times = n_obs)

    # Draw indexes for sampling from functional data without replacement
    sample_inds <- map(.x = 1:n_samples,
                       .f = function(i) sample(x = ids, size = sample_size, replace =␣
↪FALSE))

    # Determine how often each observation appeared in the samples and update the vector
    freq_samples <- tabulate(unlist(sample_inds))
    num_samples[1:length(freq_samples)] <- num_samples[1:length(freq_samples)] + freq_
↪samples

    # Perform the outlier classification procedure on the chosen samples parallelized
    # with the function clusterApplyLB() from the parallel package
    sample_flagged_par <- clusterApplyLB(cl = cl,
                                         x = sample_inds,
                                         fun = function(smpl){
                                             random_access_par_stretch_helper(list_path␣
↪= list_path, ids = smpl,
                                                                              index =␣
↪smpl, alpha = alpha, B = B, gamma = gamma,
                                                                              measuring_
↪interval = measuring_interval)})

    # Determine how often each observation were flagged in the samples and update the␣
↪vector
    freq_outliers <- tabulate(unlist(sample_flagged_par))
    num_outliers[1:length(freq_outliers)] <- num_outliers[1:length(freq_outliers)] +␣
↪freq_outliers

    # Return list containing the three central vectors: num_samples, num_outliers,␣
↪certainties
    return(list(num_samples = num_samples,
                num_outliers = num_outliers))
}
```

Putting the pieces together

Now that there are functions that

- Find the set of measuring intervals that occur in the data set
- Find comparable observations given a specific measuring interval and an acceptable stretching factor
- Perform the sampling procedure with acceptable stretching on comparable subsets of the data set

it is possible to assemble the pieces into one cohesive function that performs the outlier classification procedure described above on a data set where zeroing is admissible.

[23]:
```
# cl: cluster object generated by parallel package
# list_path: path to the random access list of the data set (generated by package␣
↪largeList)
# measuring_intervals: matrix of measuring intervals
# n_obs: number of observations in the data set
# lambda: acceptable stretching parameter
# n_samples: number of samples to use in each iteration (NULL for procedure determining␣
↪value)
# sample_size: number of observations to use in each sample in each iteration (NULL for␣
↪procedure determining value)
# alpha: quantile of least deep observations to drop before bootstrapping (in␣
↪approximation of C) (NULL for procedure determining value)
# B: number of smoothed bootstrap samples to use (in approximation of C) (NULL for␣
↪procedure determining value)
# gamma: tuning parameter for smoothed bootstrap (in approximation of C)

dectection_zr_smpl <- function(cl, list_path, measuring_intervals, n_obs, lambda, n_
↪samples = NULL, sample_size = NULL, alpha = NULL, B = NULL, gamma = 0.05){

    # generate useful identifies for vectors
    ids <- 1:n_obs

    # create vectors as described in the description part
    num_samples <- rep(x = 0, times = n_obs)
    num_outliers <- rep(x = 0, times = n_obs)
    frac_outliers <- rep(x = 1, times = n_obs)

    # determine unique intervals to iterate through
    unique_intervals <- unique_intervals(measuring_intervals)
    n_intervals <- dim(unique_intervals)[1]

    # iteration process
    for(i in 1:n_intervals){

        # Possible output
        # print(paste0(i, " out of ", n_intervals))

        # find comparable observations
        comparable <- comparable_obs_finder(main_interval = unique_intervals[i,],
                                            measuring_intervals = measuring_intervals,
                                            lambda = lambda, ids = ids)$ids
```

(continues on next page)

(continued from previous page)

```
        # do stretching and sampling procedure on current comparable observations
        intv_res <- stretch_and_sample(cl = cl, n_samples = n_samples, sample_size =␣
↪sample_size,
                                       alpha = alpha, B = B, gamma = gamma, list_path =␣
↪list_path,
                                       measuring_interval = measuring_intervals[i,],␣
↪comparable = comparable,
                                       n_obs = n_obs)

        # update the vectors
        num_samples <- num_samples + intv_res$num_samples
        num_outliers <- num_outliers + intv_res$num_outliers
    }

    # calculate the relative frequency of outliers
    frac_outliers <- unlist(map(.x = 1:n_obs,
                                .f = function(i) ifelse(num_samples[i] != 0, num_
↪outliers[i]/num_samples[i], 1)))

    # Return the three vectors
    return(list(num_samples = num_samples,
                num_outliers = num_outliers,
                certainties = frac_outliers))
}
```

Updating

The updating procedure is currently incomplete but will be finalized in future revisions of this project.

Find measuring intervals with possible occurences of new observation

[24]:
```
# new_func_obs: list with two elements: vectors of equal length called args and vals
# new_id: identifier for the new observation (be careful not to create duplicates - for␣
↪example just consecutive intergers would be fine)
# list_path: path to the largeList object, where data set is saved
# id_path: path to the RDS file with the identifiers

obs_append <- function(new_func_obs, new_id, list_path, id_path){

    # read in previous ids
    ids <- readRDS(file = id_path)
    # append new ID
    new_ids <- c(ids, new_id)
    # overwrite old ID file
    saveRDS(object = new_ids, file = id_path)

    # largeLists support appending
    saveList(object = list(new_func_obs), file = list_path, append = TRUE)
}
```

Appending observation to list of functional data and ids

[25]:
```
# new_func_obs: list with two elements: vectors of equal length called args and vals
# unique_intervals: matrix containing measuring intervals that occur in the data set␣
↪(one in each row)
# lambda: acceptable stretching parameter

possible_occurences <- function(new_func_obs, unique_intervals, lambda){

    # determine measuring interval of new observation
    new_measuring_interval <- c(min(new_func_obs$args),  max(new_func_obs$args))

    # determine for all previously used measuring intervals if the new
    # observation could have been part of the stretching and sampling procedure
    occurs <- map(.x = 1:(dim(unique_intervals)[1]),
                  .f = function(i) ifelse(new_measuring_interval[2] >= unique_
↪intervals[i,2]/lambda
                                          & new_measuring_interval[2] <= unique_
↪intervals[i,2]*lambda,
                                          TRUE, FALSE))

    # return the indices and intervals that the new observation could have been a part of
    return(list(occurs = occurs,
                occurs_intervals = unique_intervals[occurs, ]))
}
```

Determine expected number of occurences in this measuring interval sampling run

[26]:
```
# orig_n_obs: number of observations in the original comparable set (use output from␣
↪comparable_obs_finder to determine this)
# orig_n_samples: number of samples drawn in the original procedure (this can be updated␣
↪by adding those drawn now for future updates)
# orig_sample_size: sample size used in the original prrocedure for this comparable␣
↪subset

exp_num_samples <- function(orig_n_obs, orig_n_samples, orig_sample_size){

    # determine the probability that the observation would have been part of any␣
↪original sample,
    # if it had been part of the data set
    # exp_per_sample <- choose(orig_n_obs, orig_sample_size - 1) / choose(orig_n_obs + 1,
↪ orig_sample_size)
    # simplifies to:
    exp_per_sample <- orig_sample_size / orig_n_obs

    # multiplay with the number of samples and return the number rounded to the next␣
↪whole integer
    return(ceiling(orig_n_samples * exp_per_sample))
}
```

Update one measuring interval

work in progress

[ ]:

Updating for one new observation

work in progress

[27]:

```
obs_update <- function(cl, new_func_obs, list_path, measuring_intervals, lambda, n_
↪samples = NULL,
                       prev_num_samples, prev_num_outliers, sample_size = NULL, alpha =␣
↪NULL, B = NULL,
                       gamma = 0.05){

    main_interval <- c(min(new_func_obs$args),  max(new_func_obs$args))

    comparable <- comparable_obs_finder

}
```

Simulated Data

To show possible usecases of the implementation I create three datasets:

1. **No sampling** This data set shows the outlier classification procedure of Febrero-Bande, Galeano, and González-Manteiga (2008) in action. It is characterized by:
    - Observations with identical measuring intervals
    - Relatively few observations meaning that the algorithm can be applied without sampling
2. **Sampling** A data set where observations share the measuring interval but which is large enough that sampling is necessary to use the algorithm
3. **Full procedure**
    - A dataset where observations share the starting point of the measuring interval but have different endpoints.
    - This is meant to show the full procedure on a data set that is effectively already zeroed.

The functional form of the non-outliers in these data sets is approximately **linear** which has two reasons:

1. Simplicity
2. The approximate linearity of the relationship that is fundamental to the main data set explored in this project (angle and torque when tightening a bolt).

All of these data sets and the results of the outlier classification procedures applied to them can be seen in the **shiny app** provided in the repository.

No sampling:

This data set is based on a simple data generating process. Each of the 500 observations is generated as follows:

- Determine if the observation is generated as an outlier by drawing from a Bernoulli distributed random variable with parameter 0.05.
- Determine the number of points k where the function is observed. (In the terms of the method above the number of points where measurements are taken) This is drawn from a discrete uniform with elements $10, \ldots, 100$

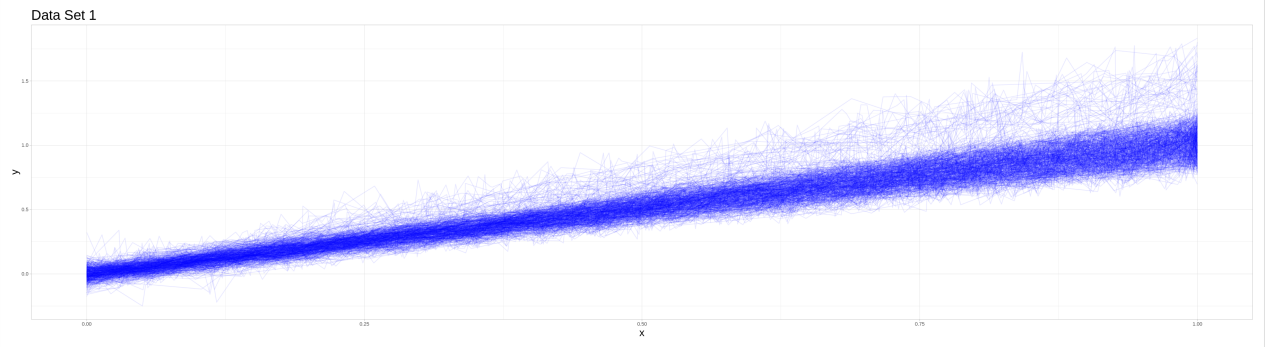
The generation of dependent variables depends on whether the observation is generated as an outlier. The general process is as follows:

- Draw k-2 realizations of $p \sim U[0,1]$ s.t. $p_1, \ldots, p_k$ i.i.d. Let $p_{(1)}, \ldots, p_{(k-2)}$ be the sorted realizations and define $(0,\ p_{(1)},\ \ldots,\ p_{(k-2)},\ 1)^T = \vec{p}$ (This is equivalent to our grid of measuring points.)
- Draw k realizations of $s \sim U[\underline{s}, \bar{s}]$ s.t. $s_1, \ldots, s_k$ i.i.d. and define $(s_1,\ \ldots,\ s_k)^T = \vec{s}$
- Draw k realizations of $\epsilon \sim \mathcal{N}[0, \sigma]$ s.t. $\epsilon_1, \ldots, \epsilon_k$ i.i.d. and define $(\epsilon_1,\ \ldots,\ \epsilon_k)^T = \vec{\epsilon}$
- Let $\vec{y} = m\vec{s} \odot \vec{p} + \vec{\epsilon}$ be the vector of realizations of the dependent variable, where $\odot$ is the component-wise (or Hadamard) product.

The parameters are different for non-outliers and outliers: For non-outliers: * $\underline{s} = 0.8, \quad \bar{s} = 1.2$ * $\sigma = 0.05$ * $m = 1.02$

and for outliers: * $\underline{s} = 1, \quad \bar{s} = 1.4$ * $\sigma = 0.1$ * $m = 1.2 * 1.02$

[28]:
```
# Data generated, transformed and visualized with functions from auxiliary/generate_set_
↪1.R
# Data set is also saved to ./data/Set_1 and if existent read from there instead
# source("auxiliary/generate_set_1.R")

if(file.exists("./data/Set_1/functional.llo") &&
   file.exists("./data/Set_1/ids.RDS") &&
   file.exists("./data/Set_1/outliers.RDS")){
    data_set_1 <- list(data = readList(file = "./data/Set_1/functional.llo"),
                       ids = readRDS(file = "./data/Set_1/ids.RDS"),
                       outliers = readRDS(file = "./data/Set_1/outliers.RDS"))

} else{
    data_set_1 <- generate_set_1()
}

if(file.exists("./data/Set_1/tibble.RDS")){
   tidy_set_1 <- readRDS("./data/Set_1/tibble.RDS")
} else{
   tidy_set_1 <- tidify_1(data_set_1$data, data_set_1$ids)
   saveRDS(object = tidy_set_1, file = "./data/Set_1/tibble.RDS")
}

vis_1(tidy_set_1)
```



Prepare data for visualization in the shiny app.

[29]:
```
if(file.exists("./data/Set_1/detection.RDS")){
    set_1_detection <- readRDS("./data/Set_1/detection.RDS")
} else{
    set_1_detection <- detection_wrap(func_dat = data_set_1$data, alpha = 0.05, B = 50,␣
↪gamma = 0.05, ids = data_set_1$ids)
    saveRDS(object = set_1_detection, file = "./data/Set_1/detection.RDS")
}

if(file.exists("./data/Set_1/summary.RDS")){
    set_1_summary <- readRDS("./data/Set_1/summary.RDS")
} else{

    missed_outliers = setdiff(data_set_1$outliers, set_1_detection$outlier_ind)
    false_outliers = setdiff(set_1_detection$outlier_ind, data_set_1$outliers)

    set_1_summary <- list(flagged = set_1_detection$outlier_ind,
                          original = data_set_1$outliers,
                          missed = missed_outliers,
                          false = false_outliers)

    saveRDS(object = set_1_summary, file = "./data/Set_1/summary.RDS")
}

if(file.exists("./data/Set_1/shiny_tibble.RDS")){
    shiny_tibble_1 <- readRDS("./data/Set_1/shiny_tibble.RDS")
} else{
    shiny_tibble_1 <- tidy_set_1 %>%
        mutate(outlier = ifelse(ids %in% data_set_1$outliers, TRUE, FALSE),
               flagged = ifelse(ids %in% set_1_detection$outlier_ind, TRUE, FALSE))
    saveRDS(object = shiny_tibble_1, file = "./data/Set_1/shiny_tibble.RDS")
}
```

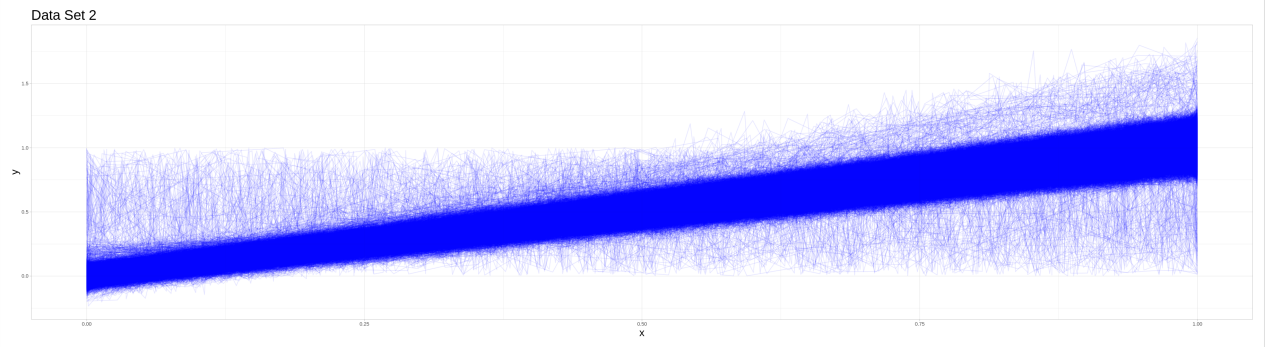
Sampling:

This second data set consists of 10000 observations where 95% are generated by the same process as the non-outlier variant in the previous data set. So a vector of arguments is drawn from the continuous uniform on $[0, 1]$ of random length. Call this vector *args* in the following description.

But the outliers take 5 different forms each appearing in about one percent of cases. I will describe their data generating processes in mathematical form in the following:

1. Exactly as the outliers in data set 1.
2. Sigmoid function: $\frac{1}{1+\exp(-3*\text{args})} + \epsilon$ where ϵ is a vector of appropriate length containing i.i.d. elements drawn from $\mathcal{N}(0, 0.05^2)$
3. Half sigmoid function: $\frac{2}{1+\exp(-3*\text{args})} - 1 + \epsilon$ where ϵ is a vector of appropriate length containing i.i.d. elements drawn from $\mathcal{N}(0, 0.05^2)$
4. Exponential function: $\frac{\exp(args)-1}{e-1} + \epsilon$ where ϵ is a vector of appropriate length containing i.i.d. elements drawn from $\mathcal{N}(0, 0.05^2)$
5. Noise: i.i.d. draws from $U[0, 1]$

[30]:
```
# Data generated, transformed and visualized with functions from auxiliary/generate_set_
↪2.R
# Data set is also saved to ./data/Set_2 and if existent read from there instead
# source("auxiliary/generate_set_2.R")

if(file.exists("./data/Set_2/functional.llo") &&
   file.exists("./data/Set_2/ids.RDS") &&
   file.exists("./data/Set_2/outliers.RDS")){
    data_set_2 <- list(data = readList(file = "./data/Set_2/functional.llo"),
                       ids = readRDS(file = "./data/Set_2/ids.RDS"),
                       outliers = readRDS(file = "./data/Set_2/outliers.RDS"))

} else{
    data_set_2 <- generate_set_2()
}

if(file.exists("./data/Set_2/tibble.RDS")){
   tidy_set_2 <- readRDS("./data/Set_2/tibble.RDS")
} else{
   tidy_set_2 <- tidify_2(data_set_2$data, data_set_2$ids)
   saveRDS(object = tidy_set_2, file = "./data/Set_2/tibble.RDS")
}

vis_2(tidy_set_2)
```



Due to the computational cost of this classifying procedure I saved its results and only perform the classification, if those results could not be found. In the case of 10000 observations can still be done on the full set and I opted to use this lower number of observations not due to computational problems with larger data sets but due to overplotting in the visualizations, that has to be addressed, before large data sets can be appropriately visualized.

[31]:
```
if(file.exists("./data/Set_2/results.RDS")){
    set_2_results <- readRDS(file = "./data/Set_2/results.RDS")

} else{
    num_cores <- detectCores()
    cl <- makeForkCluster(nnodes = num_cores)

    clusterExport(cl, varlist = list("grid_approx_set_obs",
                                     "approx_C",
                                     "grid_finder",
                                     "outlier_iteration",
                                     "outlier_detection",
                                     "detection_wrap",
                                     "random_access_par_helper"),
                                      envir = .GlobalEnv)

    set_2_results <- sampling_wrap(cl = cl, n_obs = 10000, n_samples = 150,
                                sample_size = 500, alpha = 0.05, B = 100, gamma = 0.05,
                                list_path = "./data/Set_2/functional.llo")

    stopCluster(cl)

    saveRDS(object = set_2_results, file = "./data/Set_2/results.RDS")
}

if(file.exists("./data/Set_2/shiny_tibble.RDS")){
    shiny_tibble_2 <- readRDS("./data/Set_2/shiny_tibble.RDS")
} else{
    shiny_tibble_2 <- tidy_set_2 %>%
        mutate(outlier = ifelse(ids %in% data_set_2$outliers, TRUE, FALSE))

    lengths <- unlist(map(.x = 1:10000,
                          .f = function(i) length(data_set_2$data[[i]]$vals)))

    shiny_tibble_2$cert <- unlist(map(.x = 1:10000,
                                      .f = function(i) rep(set_2_results$certainties[i],␣
↪times = lengths[i])))
```

(continues on next page)

(continued from previous page)

```
    saveRDS(object = shiny_tibble_2, file = "./data/Set_2/shiny_tibble.RDS")
}
```

To see the results of this classification procedure for different values of the certainty threshold, you can use the shiny app provided in the repository.

Full Procedure:

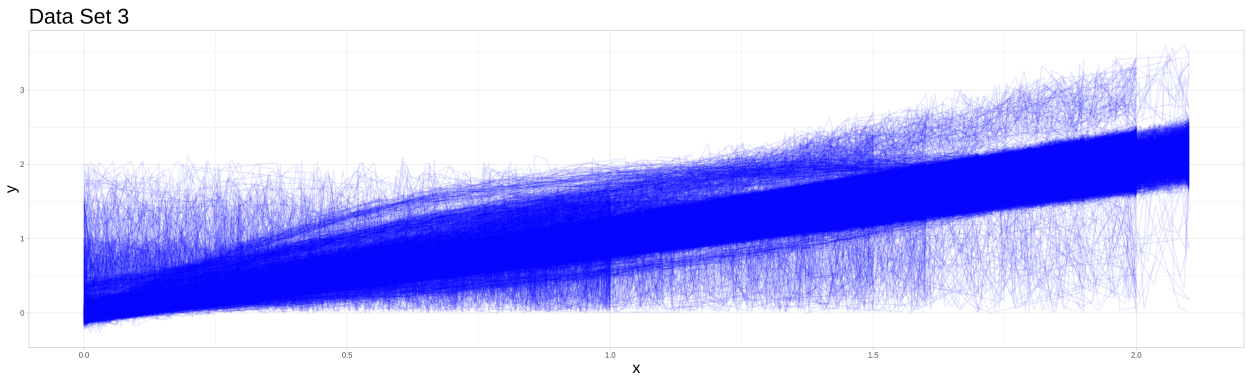
This data set is similar to the sampling data set in its data generating process. There are 30000 observations each generated by a similar process as in data set 2.

The main difference is, that the measuring intervals are not identical and are drawn from the following possibilities:

Endpoint of Measuring Interval	0.9	1	1.1	1.5	1.6	1.7	1.9	2	2.1
Probability	0.05	0.2	0.05	0.07	0.15	0.08	0.1	0.25	0.05

Also the data generating processes for both the outliers and the non-outliers are scaled such that the expected realizations at the beginning and end of the data set are identical. (Especially important for the exponential and sigmoidal outliers.) An exception from this are the type 5 outliers, which stay uniformly distributed - but the upper border of the support is made to fit the expected realization of the non-outlier process at the end point of the measuring interval.

[32]:
```
# Data generated, transformed and visualized with functions from auxiliary/generate_set_
↪3.R
# Data set is also saved to ./data/Set_3 and if existent read from there instead
source("auxiliary/generate_set_3.R")

if(file.exists("./data/Set_3/functional.llo") &&
   file.exists("./data/Set_3/ids.RDS") &&
   file.exists("./data/Set_3/outliers.RDS")){
    data_set_3 <- list(data = readList(file = "./data/Set_3/functional.llo"),
                       ids = readRDS(file = "./data/Set_3/ids.RDS"),
                       outliers = readRDS(file = "./data/Set_3/outliers.RDS"))

} else{
    data_set_3 <- generate_set_3()
}

if(file.exists("./data/Set_3/tibble.RDS")){
   tidy_set_3 <- readRDS("./data/Set_3/tibble.RDS")
} else{
   tidy_set_3 <- tidify_3(data_set_3$data, data_set_3$ids)
   saveRDS(object = tidy_set_3, file = "./data/Set_3/tibble.RDS")
}

### Due to a lengthy plotting time and overplotting, this plot is saved an inserted as a␣
↪png
# vis_3(tidy_set_3)
```



Full procedurre still does not worrk. Produces certainties greater one.

[33]:
```
if(file.exists("./data/Set_3/results.RDS")){
   set_3_results <- readRDS(file = "./data/Set_3/results.RDS")
} else{
    num_cores <- detectCores()
    cl <- makeForkCluster(nnodes = num_cores)

    clusterExport(cl, varlist = list("grid_approx_set_obs",
                                     "approx_C",
                                     "grid_finder",
                                     "outlier_iteration",
                                     "outlier_detection",
                                     "detection_wrap",
                                     "random_access_par_helper"),
                                      envir = .GlobalEnv)

    measuring_intervals <- measuring_int(func_dat = data_set_3$data)

    comp_test <- comparable_obs_finder(main_interval = c(0,1), measuring_intervals =␣
↪measuring_intervals,  lambda = 1.2, ids = 1:30000)$ids

    #test <- stretch_and_sample(cl = cl, n_samples = 100, sample_size = 100, alpha = 0.
↪02, B = 100, gamma = 0.05, n_obs = 30000,
    #                           list_path = "./data/Set_3/functional.llo", measuring_
↪interval = c(0,1),
    #                           comparable = comp_test)

    set_3_results <- dectection_zr_smpl(cl = cl, list_path = "./data/Set_3/functional.llo
↪", measuring_intervals = measuring_intervals,
                                        n_obs = 30000, lambda = 1.2, n_samples = 300,␣
↪sample_size = 300, alpha = 0.02, B = 100, gamma = 0.05)

    stopCluster(cl)

   saveRDS(object = set_3_results, file = "./data/Set_3/results.RDS")
}

if(file.exists("./data/Set_3/shiny_tibble.RDS")){
    shiny_tibble_3 <- readRDS("./data/Set_3/shiny_tibble.RDS")
} else{
```

(continues on next page)

(continued from previous page)

```
    shiny_tibble_3 <- tidy_set_3 %>%
        mutate(outlier = ifelse(ids %in% data_set_3$outliers, TRUE, FALSE))

    lengths <- unlist(map(.x = 1:30000,
                          .f = function(i) length(data_set_3$data[[i]]$vals)))

    shiny_tibble_3$cert <- unlist(map(.x = 1:30000,
                                      .f = function(i) rep(set_3_results$certainties[i],␣
↪times = lengths[i])))

    saveRDS(object = shiny_tibble_3, file = "./data/Set_3/shiny_tibble.RDS")
}
```

Shiny App

Instead of presenting the visualizations for the classification procedure in static graphics, I decided to use a shiny web app instead. This has multiple advantages in the setting of this final project, but the main motivation behind it was the use case described to me: **An engineer wants to get a preselection of suspicious observations they should have a look at.**

In this context, having a raw R file as output without an easy way to interact with it, would not be particularly useful. Instead, being able to run this shiny app on a local server with the precalculated values stored in a database, would be more in line with the idea of the job. This also motivates the features I implemented for the app (and plan to implement in future updates) like:

- setting the focus to single observations
- changing the plotting window
- changing the centrainty threshold for observations to be classified as atypical

etc.

To start the shiny app, I recommend cloning this repository and executing the file **app.R** locally. This will start the shiny app on a local server. Instead, one can choose to use the **binder** button on the repo site to start it. Due to a lengthy build process, this is not the recommended way to look at it.

The shiny app serves as the visualization for all results of the previously explained simulation studies and shows what a possible deployment of the method in a real world scenario could look like.

Outlook

Since this project will be continued in the next semester as part of the lecture "OSE - Scientific Computing for Economists" I want to give an outlook on what is to come as part of future revisions of this project.

Finalize Implementation of Updating Procedure

In the current state of the project, the implementation of the updating procedure is incomplete. Future revisions will fill this incompleteness and add simulations to show the process of updating and compare cases where observations were added in an updating procedure to data sets containing them from the beginning. This can serve as a device to check the validity of the updating procedure.

Improvements of Implementation

The implementation above still has some other problems. The most striking is the tendency of the sampling procedure to mark a full sample as atypical, if it contains a lower than expected fraction of atypical observations. I plan to address this by introducing a rescaling factor to the cutoff threshold C that adjusts for the removal of observations from the sample currently under consideration. This seems reasonable, as adding a new observation to a data set will always increase calculated depths for all observations. Therefore, removing observations can lead to overall lower depths and incrementally lower all observations under the cutoff that determines if they are classified as atypical in a sample. This is not described in the paper this project is based on - so it will make some theoretical work necessary to develop an appropritae correction.

Generalizations

As described in a previous section, the method described in this project for data sets which allow for zeroing of the observations could be generalized by introducing another parameter **acceptable shifting** in addition to **acceptable stretching**. This could then be used to define a method that finds comparable subsets in data sets that do not allow zeroing. As this introduces more variables and makes a more sophisticated splitting algorithm necessary, it was out of the scope of this project, but will be addressed in the future.

Parameter Choice

An important part of this procedure will be the choice of its tuning parameters: * In its purest form the algorithm needs a choice for α, γ, B and a grid to use for approximation purposes * Adding sampling adds the choice of sample size and number of samples * The full procedure then additionally needs λ

Some of these like sample size and number of samples could be made dependent on the structure of comparable subsets and even change when switching from one comparable subset to the next. One goal of this could be to make each observation appear in a similar number of samples overall. But other reasonable procedures are possible. Others like α and λ could be chosen by a simulation method. Constructing a similar but smaller data set with intentionally added outliers to perform cross validation or a similar procedure could be an approach for this case. A more sophisticated and detailed description of this method will be part of future revisions.

Making Results reproducible

Currently, the results of the method are mostly non-reproducible when taking about exact numbers. Qualitative results of the procedure will be similar, but due to randomization in the method the replication of exact numbers is currently not possible. This can be addressed with some work to allow for manually setting seeds in the classification algorithm without sampling and in the choice of samples. Because of parallelization this cannot be done be setting the seed once in the beginning.

Performance Measures for the Algorithm & Benchmarking

Tightly connected to the point of parameter choice is the question of how to measure the performance of the outlier detection procedure in different settings.

- First, I am going to look for existing data sets that are commonly used to benchmark outlier classification procedures. The performance of this algorithm in these preclassified settings can serve as grounds for determining in which cases and using which parameter choices the method performs well and compare it to existing methods that are applicable in comparable scenarios.
- Second, as can be seen by the previously generated data, an ex ante classification of realizations created by different dgps may not appropriately cover the idea of outlyingness. Some realizations in previous data sets look very typical for the non-outlier dgp and a classification as atypical due to the different dgp could lead to an underestimation of the procedures performance. Therefore, a comparison to established methods can serve as a better tool to judge the effectiveness.

Identifying what contributes to outlyingness of an observation

Once atypical observations in a data set are identified, it is very interesting to see what contributes to their outlyingness in the eyes of the algorithm. To create some ex-post explanation for why a classification decision was made would be a useful tool to inform future real-world decisions or improve the procedure itself by incorporating that information into the mechanism. Some interesting approaches to create an ex-post explanation are the following:

1. Create slightly altered realizations of an observation that has been marked as an outlier and see what effect different alterations have on its classification (or certainty in case of the sampling-based methods).
2. Compare locally similar observations that have different outcomes in the overall procedure.

There are more ways to gain information on what features contribute to outlyingness, but due to the current scope of this project, more in-depth considerations of this approach will only be part of future revisions.

Creation of an Rcpp-Package to improve usability

Further iterations of this project will be different in the way the functions are made available. Instead of implementing the functions directly in the main notebook, there will be an additional notebook explaining the implementation and an R package that can be installed directly that contains all functionality provided above. This will make it easier to use the functions in the future.

Improvements to the Shiny App

The visualizations of the shiny app will also undergo considerable overhauls, starting with changes to which observations are plotted when focus is set to a single observation. In this case for example plotting more observations in the close vicinity of the observation might be of greater interest, whereas observations farther away may provide little information of use and could be plotted less frequently to avoid overplotting. There are further improvements planned but due to the interactive process of designing an interface that is meant for direct interaction with the user, it is difficult to predict the exact nature of the changes.

Sources

- Cuevas, A. & Febrero-Bande, M. & Fraiman, R. (2006). On the use of bootstrap for estimating functions with functional data. Computational Statistics & Data Analysis. 51. 1063-1074.
- Febrero-Bande, M. & Galeano, P. & Gonzàlez-Manteiga, W. (2008). Outlier detection in functional data by depth measures, with application to identify abnormal NOx levels. Environmetrics. 19. 331 - 345.
- Gijbels, I. & Nagy, S. (2017). On a General Definition of Depth for Functional Data. Statistical Science. 32. 630-639.

Notebook by **Jakob R. Jürgens** Final project for the course **OSE - data science** in the summer semester 2021 Find me at jakobjuergens.com

Deutsche Bank

The Deutsche Bank project uses different modern Machine Learning (ML) methods to detect so-called regret credits, which can be considered as credits that become deficient at some point in time. The project's scope is to develop AI-related models that can determine the regret-probability of credits and assign it to certain characteristics. One challenge is to shed light on the so-called black-box, where even the AI developers do not know why their AIs make certain decisions. Another challenge is to design the model such that it provides precise results although the data set is imbalanced, that is, it contains many more non-regret credits than regret credits. Moreover, data visualizations are performed to support the result's content to finally present them to the stakeholders.

Project by Arbi Kodraj

Deutsche Bank Collaboration Project

[ ]:

CHAPTER

TWO

PARTNERS

Our courses equip students with the required skills in statistics, technology, and communication to use data for decision-making. Our partnerships with the private and public sector connect students directly with employment opportunities that match their interests and skill set.

2.1 Private sector

2.1.1 Ernst & Young

Alexander Sommer, Manager Advanced Analytics, provided us with an overview of the statistical methodologies used for his consulting services. His talk focused on two case studies. First, he and his team applied outlier detection methods, cluster analysis, and simulation methods to improve the quality assurance for a major manufacturer in the automobile sector. Second, he reported on his experience working with the public sector regarding the development of an online recommendation engine. Throughout his presentation, Alexander emphasized the need for data visualization and a structured workflow to facilitate communication with clients.

2.1.2 McKinsey & Company

Nils Wittmann, Analytics Expert, explained how his company uses the latest analytical tools to improve the internal processes of his clients and their interaction with customers. In addition, Nils shared his experiences from two recent projects. First, he reported on the use of regression models and machine learning techniques to tackle customer churn. Second, he presented an ensemble approach to forecast customer demand. Both projects illustrated the need for visualization and communication skills, in addition to technical knowledge, to bridge the gap between data insights and business value.

2.1.3 Deutsche Bank

Susanne Scholten and Martin Slowik, discussed the practice of data analytics in the banking sector with us. As an example, they focused their presentation on default prediction and discussed the bank's internal model to predict the likelihood of default and the associated losses. They also introduced us to the agile software development environment used at the Deutsche Bank. In that context, Martin stressed the importance of project monitoring and the availability of continuous model validation.

2.2 Public sector

2.2.1 Bundesrechnungshof

Sebastian Garmann, Peter Koß, and Gregor Teischler, data scientists from the Bundesrechnungshof, explained how data analytics support their auditing of public institutions. For example, they presented a project assessing the effectiveness of communication with the public. In doing so, they combined quantitative analysis and qualitative analysis using approaches such as text mining, word clouds, network graphs, and sentiment scores. Throughout, they emphasized the efforts of the German government to increase access and use of government data by citizens and researchers through numerous open data initiatives.

CHAPTER

THREE

REPOSITORY TEMPLATE

We provide a repository template for your course project. As we use GitHub Classroom to administrate the student projects, you need to sign up for a GitHub Account.

CHAPTER

FOUR

REPRODUCIBILITY

Reproducibility is a cornerstone of sound computational work, so please ensure full reproducibility of your project by setting up a GitHub Actions CI as your continuous integration service. We provide an introductory tutorial for conda and GitHub Actions here.

If, for example, the computation of results takes multiple hours, you might not be able to run parts of your code on GitHub Actions CI. In such cases, you can add the result in a file to your repository and load it in the notebook. See below for an example code.

```
# If we are running on GitHub Actions CI we will load a file with existing results.

if os.environ.get("CI") == "true":
    rslt = pkl.load(open("stored_results.pkl", "br"))
else:
    rslt = compute_results()

# Now, we are ready for further processing.
```

However, if you decide to do so, please be sure to provide an explanation in your notebook explaining why exactly this is required in your case.

CHAPTER

FIVE

FAQS

5.1 Why do we need to pitch our project in class?

We want to ensure that the whole group knows which topics each of us works on. Often, many projects share the same challenges, and this allows to reach out to other groups working on related issues directly.

5.2 Do we need to report on the progress of our project?

Yes. At the end of a lecture, I will frequently select a student at random to report on the current state of their project.

5.3 Why are the projects public?

Transparency and reproducibility are core values in research. Also, we want to learn from each other.

5.4 Why do we work in groups?

The need for collaboration are ubiquitous in business and research. This projects allows you to practice collaborative work and the supporting tools.

5.5 Where can I look for publications that provide the data behind their research?

Some journals provide the data for their published articles as data supplements directly on their website. In addition, the Replication Wiki and the Harvard Dataverse compile a lot of such information.

5.6 What are other useful resources for research data?

There is a tremendous amount of data available online. For example, MDRC provides a host of data files for public use here from the evaluation of public policy initiatives. The UC Irvine machine learning repository also maintains several hundred datasets. More generally, Google Dataset Search allows you to look for all kinds of online data.

A primer on finding data is available here on the personal website of Prof. Sebastian Tello-Trillo. In general, textbooks often provide an impressive amount of data from research articles.

5.7 Do we get to present our projects at the end of the course?

Yes, at the end of the lecture period we will host a "Demo Day", where selected projects will be presented to the whole class.

Make sure you subscribe to the OSE course project stream in the bonn-econ-teaching Zulip chat, where you can post any questions you may have regarding your course project.

CHAPTER

SIX

POWERED BY

We gratefully acknowledge funding by the Federal Ministry of Education and Research (BMBF) and the Ministry of Culture and Science of the State of North Rhine-Westphalia (MKW) as part of the Excellence Strategy of the federal and state governments.