

Department of Psychological and Behavioural Science

**COURSEWORK SUBMISSION FORM AND
PLAGIARISM/ACADEMIC HONESTY DECLARATION**

Please ensure that a completed copy of this form is uploaded as part of your coursework submission.

Candidate Number: 49045

Course code: PB4A7

Word-count: 2739

Date: 12/01/2023

The Department wishes to draw your attention to the School Calendar Regulations on Assessment Offences and Plagiarism:

<https://info.lse.ac.uk/Staff/Divisions/Academic-Registrars-Division/Teaching-Quality-Assurance-and-Review-Office/Assets/Documents/Calendar/RegulationsAssessmentOffences-Plagiarism.pdf>

All work submitted as part of the requirements for any assessment of the School (e.g., examinations, essays, dissertations, and any other work, including computer programs), whether submitted for formative or summative assessment, must be expressed in your own words and incorporate your own ideas and judgments. Plagiarism must be avoided in all such work. Plagiarism can involve the presentation of another person's thoughts or words as if they were your own. However, please note that plagiarism also includes self-plagiarism, which is where you as the author re-use your own previously submitted work or data in a "new" written piece of work without letting the reader know that this material has appeared elsewhere. The definition of "your own work" also includes work produced by collaboration or group-work expressly permitted by the Department.

Please also note that plagiarism as defined by the School above, need not be deliberate or intentional for it to constitute an assessment offence.

Declaration (without signature, to preserve anonymity): Having read and understood LSE's guidelines on plagiarism/academic honesty, I hereby confirm by completing and attaching this form that the work submitted is my own. By submitting this form I hereby confirm I understand the Department's policy on summative assessment, and that I have read the relevant parts of the MSc programme handbook.

Candidate number: 49045

**Replication of Hansen (2015) Punishment and Deterrence: Evidence from
Drunk Driving**

Word count: 2739

Submitted as the summative assessment for
PB4A7: Quantitative Applications for Behavioural Science 2022

In the study *Punishment and Deterrence: Evidence from Drunk Driving* (Hansen, 2015), Hansen investigated the effect of punishments and sanctions on reducing repeat drunk driving. The study used a regression discontinuity design (RDD), utilising the administrative data from 1995 to 2011 of the state of Washington, U.S., where two thresholds of blood alcohol content (BAC) are used to determine the status of driving under the influence (DUI). Specifically, a driver with a BAC over 0.08 is considered a case of DUI and will be punished via measures such as fines, jail time, and driving license suspension. One with a BAC over 0.15 is considered a case of aggravated DUI, to which more severe punishments and sanctions are applied.

To address the above research question, using the RDD with the BAC as the running variable is justified for several reasons. First, the BAC has clear-cut numeric thresholds for determining the severity of punishments. Second, whether a driver has a BAC level just above or below the thresholds is justifiably assumed to be completely random. Third, neither drivers nor police can manipulate the BAC measure. Applying RDD, Hansen hypothesised that receiving harsher punishments and sanctions at both thresholds would reduce offenders' future recidivism of DUI. This hypothesis was supported by the findings of the study.

The present study aims to replicate the findings from Hansen (2015) using the RDD applied to the data provided by the course instructor. It will only be focused on the effect of receiving punishments on recidivism at the 0.08 BAC threshold. The paper proceeds as follows. Section 1 describes the data and the econometrics method, and discusses the assumptions of applying the RDD. Section 2 presents the main results, and Section 3 discusses the results as well as critiques and extensions to the original study. Section 4 concludes.

1 Data, Methods, and Assumptions

1.1 Data

The data on which the present study is based is provided by the class instructor. It documents the BAC level of each driver, the time when it was measured, and whether it was measured at a traffic accident. The data also contains information on drivers' recidivism, gender, race, and age.

1.2 Assumptions of the RDD

Several assumptions are required for the RDD to give accurate estimates (See Appendix B for a discussion on manipulation). One of these is that the running variable should not contain non-random heaping (Barreca et al., 2016). This means that BAC should not be much more likely to take certain values than others, which can bias the estimates if occurring at the threshold. Based on the histogram of the BAC as a continuous variable (Figure 1), this assumption seems to be violated. Curiously, when the BAC is plotted as a discrete variable, non-random heaping disappears (Figure 2).

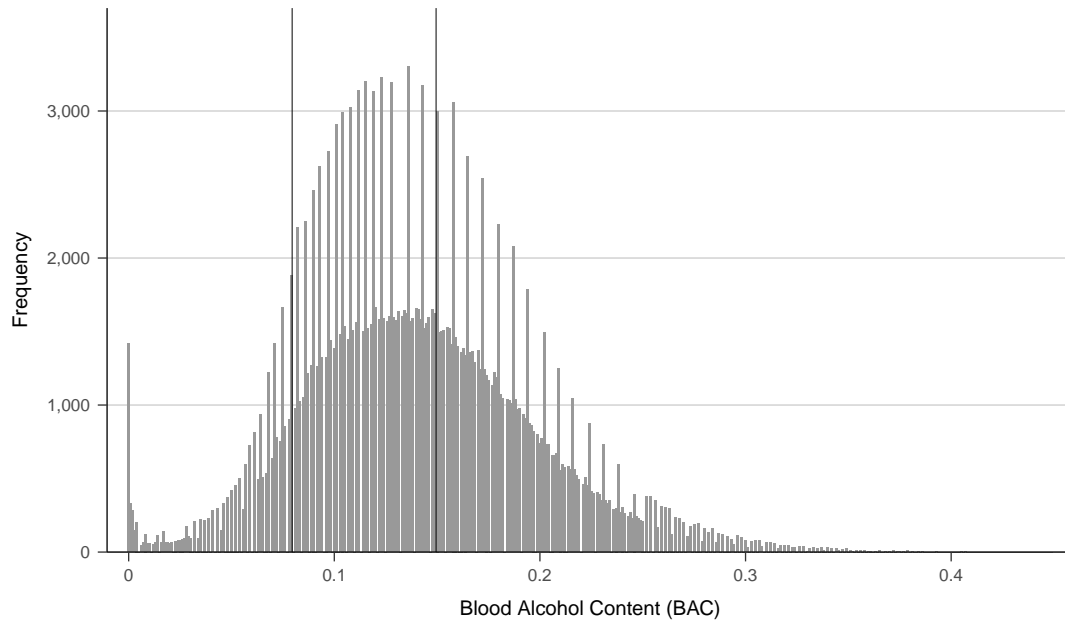


Figure 1: Histogram of blood alcohol content as a continuous variable. The blood alcohol content is plotted as a continuous variable with a bin width of 0.001, the precision used on the breathalysers. The y-axis represents the frequency of observations in each bin. The vertical black lines represent the two thresholds at 0.08 and 0.15.

A closer inspection of the data suggests that this is probably due to the lack of precision in the BAC in the current data. Many BAC values deviate by a very small amount from the values that are supposed to be given by the breathalysers (i.e., precise to three digits after the decimal point). For some bins, the measured BAC value at the left boundary deviate upwards and the value at the right boundary deviating downwards. Such bins will contain a larger number of observations than their neighbours and thus lead to heaping (See Appendix C for a detailed discussion based on simulations).

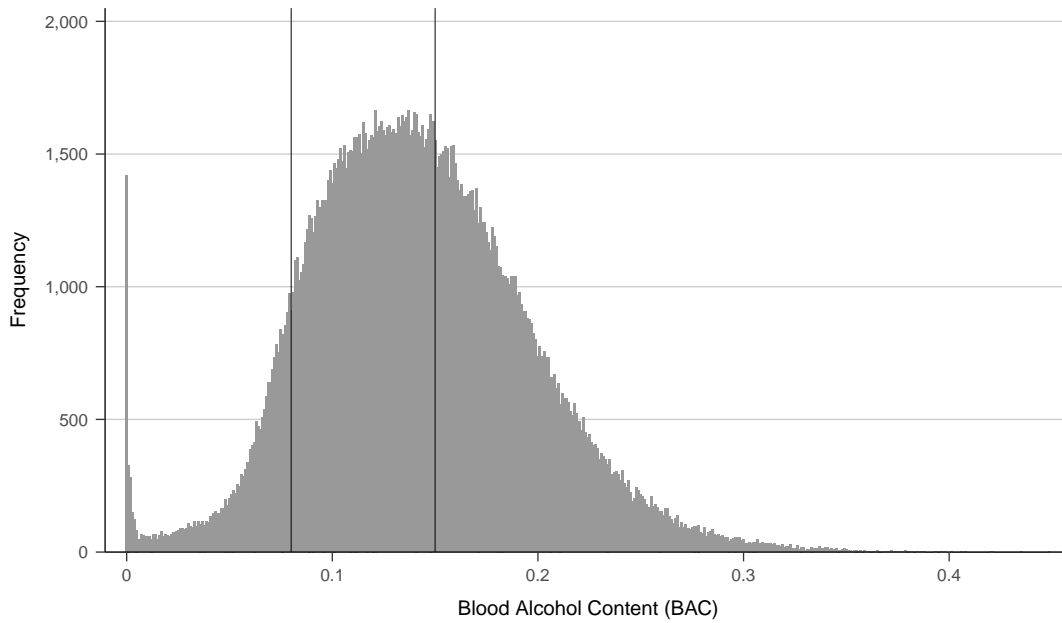


Figure 2: Histogram of blood alcohol content as a discrete variable. The blood alcohol content is plotted as a discrete variable with a bin width of 0.001, the precision used on the breathalysers. The y-axis represents the frequency of observations in each bin. The vertical black lines represent the two thresholds at 0.08 and 0.15.

1.3 The model and control variables

Similar to the original study (Hansen, 2015), the present study utilises a local linear regression model with a rectangular kernel weight function to estimate the effect of receiving punishments at the threshold on recidivism. For sensitivity analyses, the models are re-estimated using second-order polynomials or a triangle kernel function (See Appendix B for model details).

To determine the control variables, I run preliminary analyses on the effect of receiving punishments at the threshold on four predetermined characteristics, including three demographic variables (gender, race, and age) and the BAC test being conducted in a traffic accident. The analyses use the same local linear regression model as the main analyses, with a bandwidth of 0.05 and a rectangular kernel weight function. As Table 1 shows, I fail to reject any of the null hypotheses that predetermined characteristics remained the same at the threshold. This indicates that gender, race, age, and the BAC test being conducted in a traffic accident, on average, were not different between people who did not receive punishments and those who received punishments at the threshold. According to Calonico et al. (2019), I will include the four predetermined characteristics in regression discontinuity models to improve the precision of the estimation.

Table 1: Regression Discontinuity Estimates of the Effect of Receiving Punishments at the 0.08 BAC Threshold on Predetermined Characteristics

	Male	White	Age	Accident
<i>DUI</i>	0.006 (0.006)	0.006 (0.005)	−0.141 (0.164)	−0.003 (0.004)
Mean	0.784	0.846	0.085	33.99
Num. of Obs.	89,967	89,967	89,967	89,967

Note. Regression discontinuity based estimates of the effect of receiving punishments at the 0.08 BAC threshold on four predetermined characteristics. All models use a bandwidth of 0.05 and a rectangular kernel weight function. Counterfactual predictions of mean recidivism is calculated at the 0.079 BAC threshold. Heteroscedasticity-robust standard errors are in parentheses.

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

2 Results

2.1 The Effect of Receiving Punishments on Recidivism

Table 2 reports the estimated effect of receiving punishments at the threshold on recidivism. The local linear regression model with a rectangular kernel function gives the estimate that receiving punishments at the 0.08 BAC threshold decreases recidivism by 2.4 percentage points, which is statistically significant at the level of 0.01. The local second-order polynomial with a rectangular kernel function gives an estimate of a decrease by 1.4 percentage points, which is statistically significant at the level of 0.05. These estimates are consistent across both bandwidths and across models with different types of kernel functions.

Figure 3 plots means of observed recidivism in bins and predicted recidivism based on linear regression models or second-order polynomials within the interval $BAC \in [0.06, 0.11]$. Panel A and B and Panel C and D use different methods for binning observations and representing confidence intervals to ensure the robustness of the visual evidence (See the figure caption) (Calonico, Cattaneo, et al., 2015a). All panels show an apparent drop in recidivism at the BAC threshold of 0.08. This visually indicates that there is an effect of receiving punishments at the threshold in decreasing recidivism.

Table 2: Regression Discontinuity Estimates of the Effect of Receiving Punishments at the 0.08 BAC Threshold on Recidivism

	Rectangular kernel		Triangular kernel	
	Linear	Quadratic	Linear	Quadratic
<i>A. $DUI \in [0.03, 0.13]$</i>				
<i>DUI</i>	−0.024*** (0.004)	−0.014** (0.006)	−0.020*** (0.005)	−0.014** (0.006)
Mean	0.104	0.099	0.100	0.099
Controls	Yes	Yes	Yes	Yes
Num. of Obs.	89,967	89,967	89,967	89,967
<i>B. $DUI \in [0.055, 0.105]$</i>				
<i>DUI</i>	−0.021*** (0.006)	−0.014* (0.008)	−0.018*** (0.006)	−0.016* (0.009)
Mean	0.101	0.098	0.101	0.100
Controls	Yes	Yes	Yes	Yes
Num. of Obs.	46,957	46,957	46,957	46,957

Note. Regression discontinuity based estimates of the effect of receiving punishments at the 0.08 BAC threshold on recidivism. Panel A presents estimates based on a 0.05 bandwidth, and Panel B presents estimates based on a 0.025 bandwidth. The table includes results from both linear and quadratic models, with either a rectangular or a triangular kernel weight function. Controls include individuals' gender, race, age, and an indicator of whether the BAC test was conducted in a traffic accident. Counterfactual predictions of mean recidivism are calculated at the 0.079 BAC threshold and mean age of the respective populations, averaging over individuals' gender and race, as well as over whether the BAC testing was conducted in an accident. Heteroscedasticity-robust standard errors are in parentheses.

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

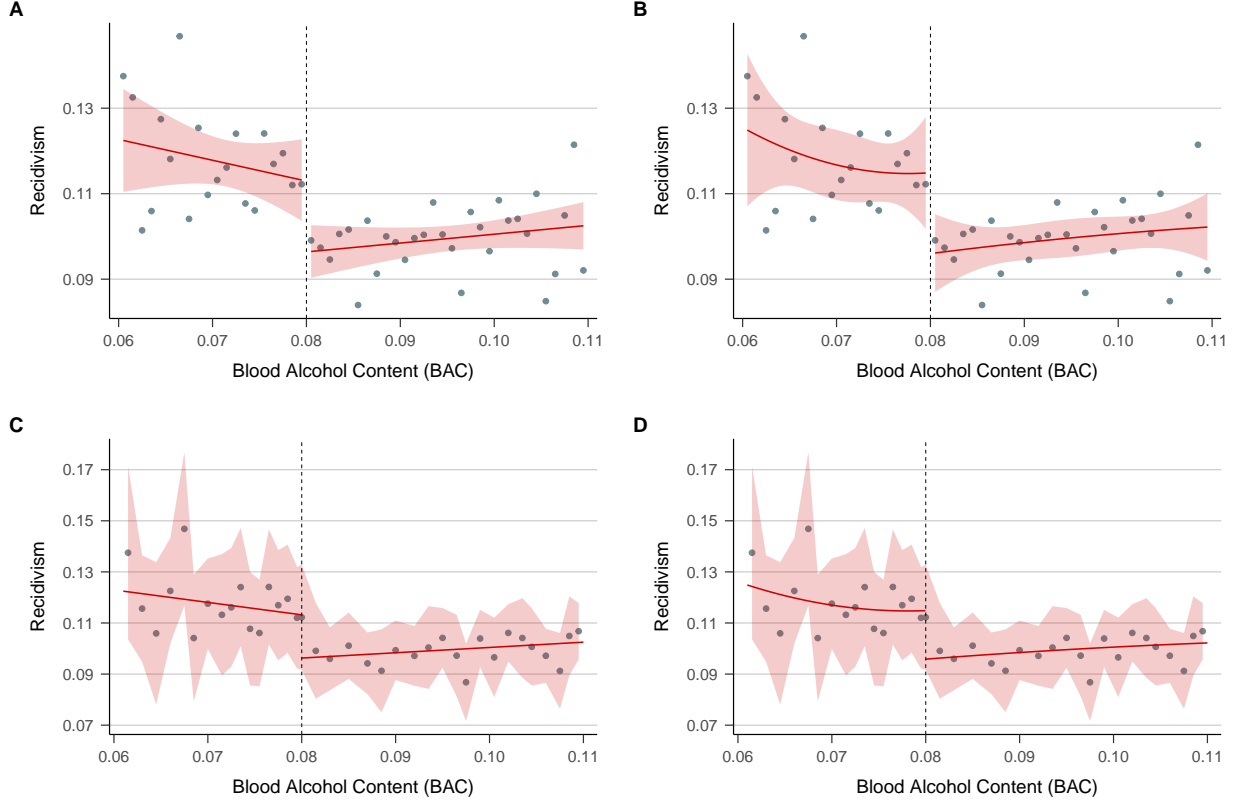


Figure 3: Regression Discontinuity Plot of the Effect of Receiving Punishments at the 0.08 BAC Threshold on Recidivism. Plot of means of recidivism in bins and predicted recidivism using simple regression models. In plotting the binned means (the grey points), Panel A and B choose the number of bins on each side of the threshold using the formula $\min\{\sqrt{n}, 10 \times \frac{\ln n}{\ln 10}\}$, while Panel C and D choose the number using quantile spaced binning (Calonico, Cattaneo, et al., 2015a). For the confidence intervals (the shaded areas in red), Panel A and B plot the 95% confidence interval of the regression models, while Panel C and D plot the 95% confidence interval of each bin. For the regression models (the red lines), Panel A and C plot the best fitted linear models, while Panel B and D plot the best fitted second-order polynomials. All panels plot the data within the interval $BAC \in [0.06, 0.11]$. The vertical dashed lines represent the BAC threshold at 0.08.

2.2 Robustness

Since there seems to be non-random heaping in the current data (Figure 1), this section will use donut hole regression discontinuity models to investigate the robustness of the results. The donut hole approach entirely drops the observations near the threshold before fitting an RD model. Under appropriate assumptions, it is effective in preventing a heap just at the threshold from biasing the estimates (Barreca et al., 2016).

For the present study, I drop the observations in the interval $BAC \in [0.079, 0.081]$, and re-estimate local linear and quadratic models with either a rectangular or a triangular kernel weight function. It is worth stressing that local polynomials are especially important for donut hole models. This is because the donut hole approach gets rid of the very observations on which the estimates of local average effects rely upon. The estimates given by donut hole models are essentially based on models' extrapolation over the region of the donut hole (Dowd, 2021) and may be especially sensitive to the assumptions about the underlying func-

Table 3: Donut Regression Discontinuity Estimates of the Effect of Receiving Punishments at the 0.08 BAC Threshold on Recidivism

	Rectangular kernel		Triangular kernel	
	Linear	Quadratic	Linear	Quadratic
<i>A. $DUI \in [0.03, 0.13]$</i>				
<i>DUI</i>	−0.026*** (0.005)	−0.014* (0.007)	−0.022*** (0.005)	−0.018** (0.006)
Mean	0.105	0.099	0.102	0.101
Controls	Yes	Yes	Yes	Yes
Num. of Obs.	88,085	88,085	88,085	88,085
<i>B. $DUI \in [0.055, 0.105]$</i>				
<i>DUI</i>	−0.022*** (0.007)	−0.015 (0.011)	−0.020*** (0.007)	−0.020 (0.012)
Mean	0.103	0.099	0.103	0.104
Controls	Yes	Yes	Yes	Yes
Num. of Obs.	45,075	45,075	45,075	45,075

Note. Regression discontinuity based estimates of the effect of receiving punishments at the 0.08 BAC threshold on recidivism after dropping observations in the interval $[0.079, 0.081]$. Panel A presents estimates based on a 0.05 bandwidth, and Panel B presents estimates based on a 0.025 bandwidth. The table includes results from both linear and quadratic models, with either a rectangular or a triangular kernel weight function. Controls include individuals' gender, race, age, and an indicator of whether the BAC test was conducted in a traffic accident. Counterfactual predictions of mean recidivism are calculated at the 0.079 BAC threshold and mean age of the respective populations, averaging over individuals' gender and race, as well as over whether the BAC testing was conducted in an accident. Heteroscedasticity-robust standard errors are in parentheses.

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

tional form. Using polynomials with different orders can ensure the robustness of the results over different functional assumptions (See Appendix C for a detailed discussion based on simulations).

The results from donut hole models are presented in Table 3. According to the local linear regression model with a rectangular kernel function, receiving punishments at the threshold decreases recidivism by 2.6 percentage points and is statistically significant at the level of 0.01. According to the local second-order polynomial with a rectangular kernel function, the decrease is 1.4 percentage points and is statistically significant at the level of 0.1. These estimates are consistent across both bandwidths and across models with different types of kernel functions, except for not being statistically significant in quadratic models estimated with a narrower bandwidth. Therefore, the results given by donut hole models are essentially similar with those given by ordinary RDD models.

3 Discussion, Critique and Extension

Using a regression discontinuity design, the present study aims to replicate the findings of Hansen (2015) about the effect of punishments and sanctions on reducing repeat drunk driving. Specifically, the present

study focuses on testing whether receiving punishments at the 0.08 Blood Alcohol Content (BAC) threshold reduces future recidivism of drunk driving. The findings based on local linear regression models show that receiving punishments at the threshold reduces recidivism by up to 2.4 percentage points ¹. Sensitivity analyses based on local second-order polynomials show that receiving punishments at the threshold reduces recidivism by up to 1.6 percentage points ². These estimates are consistent with those given by the models that take into account non-random heaping in the data, establishing the robustness of the results.

3.1 Strengths of the original study

The findings of the present study successfully replicate the main results of Hansen (2015), which gives an estimated decrease caused by receiving punishments of up to 2 percentage points at the 0.08 BAC threshold. The success of the replication demonstrates the validity of the methodology and results of the original study. Hansen (2015) also presents related results that the present study, due to its scope and the limitation of the data available, does not replicate. As a part of the strengths of the original study, these findings either establish the robustness of the main results under different assumptions or reveal the heterogeneity of the effect of receiving punishments.

In terms of the robustness of the main results, the original study estimates regression discontinuity models using polynomials up to the third order, three types of kernel functions, and based on a series of bandwidths that extend beyond 0.05 and 0.025. The results show that the point estimates are relatively consistent, only except for those given by the local second-order polynomials and those estimated based on very small bandwidths.

The heterogeneity of the effects of punishments include the effects among different populations and the effect on different types of recidivism. First of all, the original study separates analyses for drivers with and without prior drunk driving experience, in addition to running regression models on the entire population of drivers. This allows the original study to find that the effect of receiving punishments at the threshold is more pronounced among drivers with previous drunk driving experience. Second, the original study conducts a detailed analyses of the effect of receiving punishments on three types of recidivism, refusal of taking a BAC test, and the likelihood of being stopped by a police. It finds that receiving punishments decreases the likelihood of being stopped as well as recidivism of all kinds. Both results reveal the heterogeneous nature of the effect of receiving punishments on recidivism.

3.2 Suggested improvements and extensions

Despite multiple strengths of the original study, it can still be improved in several aspects. First, the primary bandwidths used in the original study (0.05 and 0.025) seem to be arbitrary choices of the researcher. Bandwidth choice can be improved by using data-driven methods with a predetermined principle. For example, Mean Squared Error (MSE) optimal bandwidth is an available method and has recently been improved for

¹The average estimated effect is 2.1 percentage points. The average estimate is calculated over models based on different bandwidths and with different kernel functions, assuming equal weights of estimates.

²The average estimated effect is 1.5 percentage points.

better inference with a bias correction (Cattaneo & Vazquez-Bare, 2017; Imbens & Kalyanaraman, 2012). Alternatively, one can choose to use a method that minimises coverage error (CE) probability when the goal is to construct optimal confidence intervals for inference (Calonico et al., 2022). Both MSE-optimal and CE-optimal bandwidths are based on the data and constructed according to an explicit desideratum, which makes the bandwidth choice more objective and transparent. Future research can and should make full use of these methods as they are readily available in packages of mainstream statistical software (Calonico, Cattaneo, et al., 2015b; Calonico et al., 2017).

Second, in light of the present study, local second-order polynomials consistently give lower estimates than linear models. In fact, this is also the finding of the original study (Appendix Table 5 of Hansen, 2015). This suggests that the estimates of the effect of receiving punishments at the threshold somewhat depend on the assumptions about the effect's functional form. Although it is difficult to pin down one reasonable functional assumption due to the complex nature of the effect, future research should give equal weights in presenting the results given by different models. This is especially important for such an estimand that is closely associated with policy making, since even a small difference in the estimate can lead to big differences when scaled up.

Third, the use of simple linear regression models is inadequate when recidivism, the outcome variable, is a binary variable. In simple words, this is because of a mathematical inconsistency between the range of the outcome variable (which can only take the value of 0 and 1) and the range of a linear regression model (which can take the value of any real number). Future research can utilise logistic regressions in combination with the RDD to estimate the effect of punishments on recidivism. This should not be a difficult task because recent research in econometrics has developed and programmed such methods based on the multinomial logit model, together with optimal bandwidth selection and bias correction techniques (Xu, 2017).

Fourth, the robustness checks using donut hole regression discontinuity models are limited. Although the original study excluded the possibility of non-random heaping based on a histogram, the treatment of the variable (i.e., continuous vs. discrete) of which is unclear, non-random heaping may exist in the original data in light of the present study. If this were the case, the donut hole approach could potentially still give biased estimates. This is because non-random heaping outside the dropping window can bias the estimation (Barreca et al., 2016). Future studies can adopt a solution suggested by previous research, that is, to separate the analyses for non-heaped data and heap points (if any). Unbiased estimates can be derived by taking a population-weighted average of the two estimates based on the respective sample sizes (Barreca et al., 2016).

Finally, the external and internal validity of the study can be improved in future extensions. To improve the external validity of the estimation, one can use data from regions other than the state of Washington. This is to establish that punishments are universally effective in reducing repeat drunk driving across regions and the effect is not caused by some other factors specific to the state of Washington. The internal validity of the causal relation can be improved by conducting a longitudinal field experiment. One can choose a region where laws criminalizing DUI are not yet enforced, and carry out an experiment by manipulating the magnitude of the punishments received by drivers if found driving under the influence. Drivers with a BAC level just below a chosen threshold are randomly assigned to receive lighter punishments than those

with a BAC level just above the threshold. This method can theoretically provide stronger evidence for the causal relation between receiving punishments and recidivism. However, there are also challenges to such designs. For example, it is hard to prevent spillover effects. Drivers may find out punishments vary among individuals, and the consequence of such an effect is essentially unpredictable. Ethically speaking, it also invites debates on how such an experiment can adhere to the principle of transparency. Due to the scope of this paper, I will leave these challenges to be addressed by future research.

4 Conclusion

The present study successfully replicates the main results of the study *Punishment and Deterrence: Evidence from Drunk Driving* (Hansen, 2015) about the effect of punishments and sanctions on reducing repeat drunk driving using a regression discontinuity design. Specifically, I find that receiving punishments at the 0.08 BAC threshold reduces recidivism by up to 2.4 percentage points according to local linear regression models, and the decrease is up to 1.6 percentage points according to local second-order polynomials. These results provide evidence for the validity of the original study. The original study also conducts additional analyses to establish the robustness of the results and reveal heterogeneity of the effects. Despite these strengths, the original study can be improved in aspects such as bandwidth selection, the selection of appropriate models and robustness checks, the presentation of the results, as well as internal and external validity.

References

- Barreca, A. I., Lindo, J. M., & Waddell, G. R. (2016). Heaping-induced bias in regression-discontinuity designs. *Economic Inquiry*, 54(1), 268–293. <https://doi.org/10.1111/ecin.12225>
- Calonico, S., Cattaneo, M. D., & Farrell, M. H. (2022). Coverage error optimal confidence intervals for local polynomial regression. *Bernoulli*, 28(4). <https://doi.org/10.3150/21-BEJ1445>
- Calonico, S., Cattaneo, M. D., Farrell, M. H., & Titiunik, R. (2019). Regression discontinuity designs using covariates. *The Review of Economics and Statistics*, 101(3), 442–451. https://doi.org/10.1162/rest_a_00760
- Calonico, S., Cattaneo, M. D., Farrell, M. H., & Titiunik, R. (2017). rdrobust: Software for regression-discontinuity designs. *The Stata Journal*, 17(2), 372–404. <https://doi.org/https://doi.org/10.1177/1536867X1701700208>
- Calonico, S., Cattaneo, M. D., & Titiunik, R. (2015a). Optimal data-driven regression discontinuity plots. *Journal of the American Statistical Association*, 110(512), 1753–1769. <https://doi.org/10.1080/01621459.2015.1017578>
- Calonico, S., Cattaneo, M., D., & Titiunik, R. (2015b). rdrobust: An R package for robust nonparametric inference in regression-discontinuity designs. *The R Journal*, 7(1), 38. <https://doi.org/10.32614/RJ-2015-004>
- Cattaneo, M. D., Jansson, M., & Ma, X. (2020). Simple local polynomial density estimators. *Journal of the American Statistical Association*, 115(531), 1449–1455. <https://doi.org/10.1080/01621459.2019.1635480>
- Cattaneo, M. D., & Vazquez-Bare, G. (2017). The choice of neighborhood in regression discontinuity designs. *Observational Studies*, 3(2), 134–146. <https://doi.org/https://doi.org/10.1353/obs.2017.0002>
- Dowd, C. (2021). *Donuts and distant CATEs: Derivative bounds for RD extrapolation* (No. 3641913). <https://doi.org/10.2139/ssrn.3641913>
- Hansen, B. (2015). Punishment and deterrence: Evidence from drunk driving. *American Economic Review*, 105(4), 1581–1617. <https://doi.org/10.1257/aer.20130189>
- Imbens, G., & Kalyanaraman, K. (2012). Optimal bandwidth choice for the regression discontinuity estimator. *The Review of Economic Studies*, 79(3), 933–959. <https://doi.org/https://doi.org/10.1093/restud/rdr043>
- Xu, K.-L. (2017). Regression discontinuity with categorical outcomes. *Journal of Econometrics*, 201(1), 1–18. <https://doi.org/10.1016/j.jeconom.2017.07.004>

Appendix A: Analysis Code

This appendix presents the code associated with the assignment questions. A full list of analysis code is stored in the Github repository (link masked for the anonymity of the report).

1. Create the variables for Driving Under the Influence, and quadratic term for Blood alcohol level and produce two histograms of BAC1 (one as a discrete variable, one as a continuous). Explain the differences of the histograms.

```
# load libraries
library(tidyverse);library(haven);library(here)
library(rdrobust);library(wesanderson)

# input data
bac_data <-
  here("summative/data/hansen_dwi.dta") %>% # locate the data
  read_dta() %>% # read the dta file
  # code variables
  mutate(
    # code into drive under influence (dui) = 1 if bac1 >= 0.08
    dui = ifelse(bac1 >= 0.08, 1, 0),
    # create a centered bac1 variable
    bac1_ctd = bac1 - 0.08,
    # create a quadratic term for both bac1 and bac1_ctd
    bac1_sq = bac1^2,
    bac1_ctd_sq = bac1_ctd^2,
    # create donut hole for later use
    donut = ifelse(abs(bac1_ctd) <= 0.001, 1, 0)
  )

# create the histogram treating BAC as a discrete variable
# equivalent to stata function histogram bac1, discrete width(0.001)
bac_data %>%
  ggplot(aes(x = bac1)) +
  geom_histogram(binwidth = 0.001, fill = "grey60", color = NA) +
  geom_vline(xintercept = 0.08, size = 0.3) +
  geom_vline(xintercept = 0.15, size = 0.3) +
  scale_x_continuous(breaks = seq(0, 0.5, by = 0.1),
    labels = c("0", "0.1", "0.2", "0.3", "0.4", "0.5"),
    expand = c(0, 0.01)) +
  scale_y_continuous(breaks = seq(0, 2000, by = 500),
```

```

        labels = c("0", "500", "1,000", "1,500", "2,000"),
        limits = c(0, 2050),
        expand = c(0, 0)) +
labs(x = "Blood Alcohol Content (BAC)", y = "Frequency") +
theme_classic() +
theme(
  panel.grid = element_blank(),
  panel.grid.major.y = element_line(color = "grey80", size = 0.4),
  axis.line = element_line(size = 0.4),
  axis.text.y = element_text(size = 11, margin = margin(0, 5, 0, 0)),
  axis.text.x = element_text(size = 11, margin = margin(5, 0, 0, 0)),
  axis.ticks = element_line(size = 0.4),
  axis.ticks.length = unit(6, units = "pt"),
  axis.title.y = element_text(size = 13, margin = margin(0, 10, 0, 0)),
  axis.title.x = element_text(size = 13, margin = margin(10, 0, 0, 0)),
  plot.margin = margin(20, 20, 20, 20)
)

```

```

# create the histogram treating BAC as a continuous variable
# equivalent to stata function histogram bac1, width(0.001)
bac_data %>%
  mutate(bac1 = bac1 - 0.0005) %>%
  ggplot(aes(x = bac1)) +
  geom_histogram(binwidth = 0.001, fill = "grey60", color = NA) +
  geom_vline(xintercept = 0.08 - 0.0005, size = 0.3) +
  geom_vline(xintercept = 0.15 - 0.0005, size = 0.3) +
  scale_x_continuous(breaks = seq(0, 0.5, by = 0.1),
    labels = c("0", "0.1", "0.2", "0.3", "0.4", "0.5"),
    expand = c(0, 0.01)) +
  scale_y_continuous(breaks = seq(0, 3000, by = 1000),
    labels = c("0", "1,000", "2,000", "3,000"),
    limits = c(0, 3700),
    expand = c(0, 0)) +
labs(x = "Blood Alcohol Content (BAC)", y = "Frequency") +
theme_classic() +
theme(
  panel.grid = element_blank(),
  panel.grid.major.y = element_line(color = "grey80", size = 0.4),

```

```

axis.line = element_line(size = 0.4),
axis.text.y = element_text(size = 11, margin = margin(0, 5, 0, 0)),
axis.text.x = element_text(size = 11, margin = margin(5, 0, 0, 0)),
axis.ticks = element_line(size = 0.4),
axis.ticks.length = unit(6, units = "pt"),
axis.title.y = element_text(size = 13, margin = margin(0, 10, 0, 0)),
axis.title.x = element_text(size = 13, margin = margin(10, 0, 0, 0)),
plot.margin = margin(20, 20, 20, 20)

)

```

2. Running regressions on covariates (white, male, age and accident) to see if there is a jump in average values for each of these at the cutoff and explain the results.

```

# load the function to display summary statistics with robust standard errors
source(here("summative/code/summaryR.R"))

# get the data within the bandwidth
dat_bandwidth_0.05 <- bac_data %>% filter(abs(bac1_ctd) <= 0.05)

# check being white
white_rdd <- dat_bandwidth_0.05 %>% lm(white ~ dui*bac1_ctd, data = .)
# display results with robust SEs
summaryR.lm(white_rdd, type = "hc1")

# check being male
male_rdd <- dat_bandwidth_0.05 %>% lm(male ~ dui*bac1_ctd, data = .)
summaryR.lm(male_rdd, type = "hc1")

# check accident
accident_rdd <- dat_bandwidth_0.05 %>% lm(acc ~ dui*bac1_ctd, data = .)
summaryR.lm(accident_rdd, type = "hc1")

# check age
age_rdd <- dat_bandwidth_0.05 %>% lm(aged ~ dui*bac1_ctd, data = .)
summaryR.lm(age_rdd, type = "hc1")

```

3. Produce main recidivism results of the paper (with our dataset) using recidivism as dependent variable as well as with a changed bandwidth of the RDD to 0.055 to 0.105.


```

# linear regression
linear_a <-
  dat_bandwidth_0.05 %>%
  lm(recidivism ~ dui*bac1_ctd + white + male + acc + aged,
     data = .)
summaryR.lm(linear_a, type = "hc1")

# quadratic model
qua_a <-
  dat_bandwidth_0.05 %>%
  lm(recidivism ~ dui*(bac1_ctd + bac1_ctd_sq) + white + male + acc + aged,
     data = .)
summaryR.lm(qua_a, type = "hc1")

# get the data within the 0.025 bandwidth
dat_bandwidth_0.025 <- bac_data %>% filter(abs(bac1_ctd) <= 0.025)

# linear regression
linear_b <-
  dat_bandwidth_0.025 %>%
  lm(recidivism ~ dui*bac1_ctd + white + male + acc + aged,
     data = .)
summaryR.lm(linear_b, type = "hc1")

# quadratic model
qua_b <-
  dat_bandwidth_0.025 %>%
  lm(recidivism ~ dui*(bac1_ctd + bac1_ctd_sq) + white + male + acc + aged,
     data = .)
summaryR.lm(qua_b, type = "hc1")

```

4. Replicate Qe by running “donut hole regressions”. Explain why one might need a donut hole regression? How do I run a donut hole regression?
5. Run local polynomials and explain why local polynomials might be needed after a donut hole.

```

# linear regression
linear_a_donut <-
  dat_bandwidth_0.05 %>%
  filter(donut == 0) %>%

```

```

lm(recidivism ~ dui*bac1_ctd + white + male + acc + aged,
  data = .)
summaryR.lm(linear_a_donut, type = "hc1")

# narrower bandwidth
linear_b_donut <-
  dat_bandwidth_0.025 %>%
  filter(donut == 0) %>%
  lm(recidivism ~ dui*bac1_ctd + white + male + acc + aged,
    data = .)
summaryR.lm(linear_b_donut, type = "hc1")

# quadratic model
qua_a_donut <-
  dat_bandwidth_0.05 %>%
  filter(donut == 0) %>%
  lm(recidivism ~ dui*(bac1_ctd + bac1_ctd_sq) + white + male + acc + aged,
    data = .)
summaryR.lm(qua_a_donut, type = "hc1")

# narrower bandwidth
qua_b_donut <-
  dat_bandwidth_0.025 %>%
  filter(donut == 0) %>%
  lm(recidivism ~ dui*(bac1_ctd + bac1_ctd_sq) + white + male + acc + aged,
    data = .)
summaryR.lm(qua_b_donut, type = "hc1")

```

6. Produce an RDplot and a cmogram with bandwidths of 0.060 & 0.11. Explain what the respective graphs show.

```

# replicate the cmogram output given by stata
# get the trimmed data
dat_tirmmed <- bac_data %>% filter(bac1 > 0.06 & bac1 < 0.11)

# number of breaks left to the cutoff
n_left <-
  filter(dat_tirmmed, bac1_ctd < 0) %>%
  nrow() %>%
  min(sqrt(.), 10*log(.) / log(10)) %>%

```

```

    floor() + 1
# number of breaks right to the cutoff
n_right <-
  filter(dat_tirmmed, bac1_ctd >= 0) %>%
  nrow() %>%
  min(sqrt(.), 10*log(.) / log(10)) %>%
  floor() + 1

# get the breaks based on bin numbers
min_bac <- min(dat_tirmmed$bac1)
max_bac_lower <- filter(dat_tirmmed, bac1_ctd < 0)$bac1 %>% max()
max_bac <- max(dat_tirmmed$bac1)
breaks <- c(seq(min_bac, max_bac_lower, length.out = n_left),
             seq(0.08, max_bac, length.out = n_right))

# plotting
dat_tirmmed %>%
  # create the bins
  mutate(bins = cut(bac1, breaks = breaks)) %>%
  group_by(bins) %>%
  summarize(
    recidivism = mean(recidivism), # calculate the mean recidivism in each group
    bac = mean(bac1), # roughly the midpoint of each bin for plotting the x axis
    dui = ifelse(bac >= 0.08, 1, 0) %>% factor() # indicator for DUI
  ) %>%
  ggplot(aes(x = bac, y = recidivism, color = dui)) +
  geom_point(show.legend = FALSE) +
  # superimpose a fitted line model and confidence interval using the raw data
  geom_smooth(data = dat_tirmmed,
              aes(x = bac1, y = recidivism, color = dui),
              method = "lm", formula = y ~ x,
              inherit.aes = FALSE, show.legend = FALSE) +
  # a quadratic model with CI
  # geom_smooth(data = dat_tirmmed,
  #
      aes(x = bac1, y = recidivism, color = dui),
  #
      method = "lm", formula = y ~ poly(x, 2),
  #
      inherit.aes = FALSE, show.legend = FALSE) +
  geom_vline(xintercept = 0.08, color = "red", linetype = 2) +
  scale_x_continuous(breaks = seq(0.02, 0.14, by = 0.02)) +

```

```

scale_y_continuous(breaks = seq(0.07, 0.15, by = 0.02)) +
scale_color_discrete(type = wes_palette("Royal1", 2, type = "discrete")) +
labs(x = "Blood Alcohol Content (BAC)", y = "Recidivism") +
theme_classic() +
theme(
  panel.grid = element_blank(),
  panel.grid.major.y = element_line(color = "grey80", size = 0.4),
  axis.line = element_line(size = 0.4),
  axis.text.y = element_text(size = 11, margin = margin(0, 5, 0, 0)),
  axis.text.x = element_text(size = 11, margin = margin(5, 0, 0, 0)),
  axis.ticks = element_line(size = 0.4),
  axis.ticks.length = unit(6, units = "pt"),
  axis.title.y = element_text(size = 13, margin = margin(0, 10, 0, 0)),
  axis.title.x = element_text(size = 13, margin = margin(10, 0, 0, 0)),
  plot.margin = margin(20, 20, 20, 20)
)

```

```

# RD plot
# run the rdrobust model
rd <- with(dat_tirmmed,
  rdplot(recidivism, bac1,
    c = 0.08, # cutoff point
    p = 1, # order of the polynomial
    binselect = "qs", # method for select number of bins
    masspoints = FALSE,
    ci = 95,
    hide = TRUE)
)

# get the statistics for plotting
c = 0.08 # threshold
rdplot_mean_bin = rd$vars_bins[, "rdplot_mean_bin"]
rdplot_mean_y = rd$vars_bins[, "rdplot_mean_y"]
y_hat = rd$vars_poly[, "rdplot_y"]
x_plot = rd$vars_poly[, "rdplot_x"]
rdplot_cil_bin = rd$vars_bins[, "rdplot_ci_l"]
rdplot_cir_bin = rd$vars_bins[, "rdplot_ci_r"]
rdplot_mean_bin = rd$vars_bins[, "rdplot_mean_bin"]
y_hat_r = y_hat[x_plot >= c][-1]

```

```

y_hat_l=y_hat[x_plot<c]
x_plot_r=x_plot[x_plot>=c][-1]
x_plot_l=x_plot[x_plot<c]
# collect dots for plotting lines in one tibble
line <- tibble(x = c(x_plot_l, x_plot_r),
               y = c(y_hat_l, y_hat_r),
               dui = c(rep(0, 499), rep(1, 500)) %>% factor())

# plotting
ggplot() +
  geom_point(aes(x = rdplot_mean_bin, y = rdplot_mean_y),
             col = "#728C94", na.rm = TRUE) +
  geom_line(data = line, aes(x = x, y = y, color = dui),
            na.rm = TRUE, inherit.aes = FALSE, show.legend = FALSE) +
  geom_ribbon(aes(x = rdplot_mean_bin, ymin = rdplot_cil_bin, ymax = rdplot_cir_bin),
             fill = "#CE0000", alpha = 0.2) +
  scale_x_continuous(breaks = seq(0.06, 0.11, by = 0.01), expand = c(0.03, 0)) +
  scale_y_continuous(breaks = seq(0.07, 0.17, by = 0.02)) +
  scale_color_discrete(type = c("#CE0000", "#CE0000")) +
  labs(x = "Blood Alcohol Content (BAC)", y = "Recidivism") +
  geom_vline(xintercept = 0.08, size = 0.3, linetype = 2) +
  theme_classic() +
  theme(
    panel.grid = element_blank(),
    panel.grid.major.y = element_line(color = "grey80", size = 0.4),
    axis.line = element_line(size = 0.4),
    axis.text.y = element_text(size = 11, margin = margin(0, 5, 0, 0)),
    axis.text.x = element_text(size = 11, margin = margin(5, 0, 0, 0)),
    axis.ticks = element_line(size = 0.4),
    axis.ticks.length = unit(6, units = "pt"),
    axis.title.y = element_text(size = 13, margin = margin(0, 10, 0, 0)),
    axis.title.x = element_text(size = 13, margin = margin(10, 0, 0, 0)),
    plot.margin = margin(20, 20, 20, 20)
  )

```

Appendix B: Notes on the Econometrics Methods

The RDD assumption about manipulation

In addition to the absence of non-random heaping, another important assumption of the RDD is that people need to be randomly assigned to receiving punishment at the BAC threshold. In other words, neither the drivers nor the police can manipulate the BAC level and thus whether one will receive punishments. Hansen (2015) has already given a plausible theoretical justification for this assumption. Empirically, I plot the distribution of BAC and test for discontinuity in its distribution at the threshold (Figure 1). Based on a recently developed local polynomial density estimator (Cattaneo et al., 2020), the hypothesis that the distribution is continuous at 0.08 cannot be rejected at the level of 0.05 ($p = 0.890$). Therefore, there is no evidence for the existence of manipulation on the BAC level.

Models

The local linear regression model used in the main analyses is specified by the following function:

$$R_i = \alpha + \beta_1 DUI_i + \beta_2 BAC_i + \beta_3 DUI_i \times BAC_i + \tau Z_i + \epsilon_i \quad (1)$$

The second-order polynomial used in the sensitivity analyses by the following function:

$$R_i = \alpha + \beta_1 DUI_i + \beta_2 BAC_i + \beta_3 BAC_i^2 + \beta_4 DUI_i \times BAC_i + \beta_5 DUI_i \times BAC_i^2 + \tau Z_i + \epsilon_i \quad (2)$$

In both models, the variable R_i is an indicator of recidivism, DUI_i is an indicator of whether BAC is above the 0.08 threshold, BAC_i is the measure of BAC level, and Z_i is a vector of control variables. The variable BAC_i in the model is centred around the threshold value of 0.08 so that the coefficients β_1 can be directly interpreted as the estimates of the effect.

Appendix C: Simulations

Simulating non-random heaping

This simulation explains how measurement error in the breathalysers can give rise to non-random heaping. It will also show how this particular kind of non-random heaping is invisible when the data is treated as discrete.

We first simulate a raw dataset. To keep the simulation simple but also similar to the original case, the data here is uniformly distributed from 0 to 1, with a precision to 3 digits after the decimal point.

```
# set seed
set.seed(32876)

## simulate the original data ##
# a sequence of possible values given by a breathalyser
# a vector of integers from 0 to 1000
x_0 <- seq(1, 1e3, by = 1)

# sample from x_0 with equal probability to simulate
# a sample of data collected by the breathalyser
n <- 1e5
x_index <- sample(x_0, size = n, replace = TRUE,
                  prob = rep(1/length(x_0), length(x_0)))
# rescale to x
# x is precise to 3 digits after the decimal point and uniformly distributed
x <- x_index/1000
```

Now we want to simulate the measurement error made by the breathalyser. Assume our breathalyser makes errors in a periodic manner: it can measure the first 18 values accurately, and then measure the next value slightly higher, followed by the next value measured slightly lower. In other words, for the sequence of the values the breathalyser is able to measure, it makes 2 errors at the end of every 20 values.

```
## simulate measurement error ##
# for the sequence of value a breathalyser is able to measure
# every 18 values that it is able to correctly measure is followed by
# a value it makes measurement error upwards
# and then a value with an error downwards
error_0 <- c(rep(0, 18), 1, -1)
error <- rep(error_0, n/length(error_0))

## simulate heaping ##
```

```

# create an empty vector
x_error <- rep(0, length(x))
# for each value of x, the measured value is determined by both
# its original value and whether the breathalyser is able to
# correctly measure x
for (i in 1:length(x)) x_error[i] <- x[i] + error[x_index[i]]*2e-6

# treat x_error as a discrete variable
x_discrete <- as.factor(x_error)

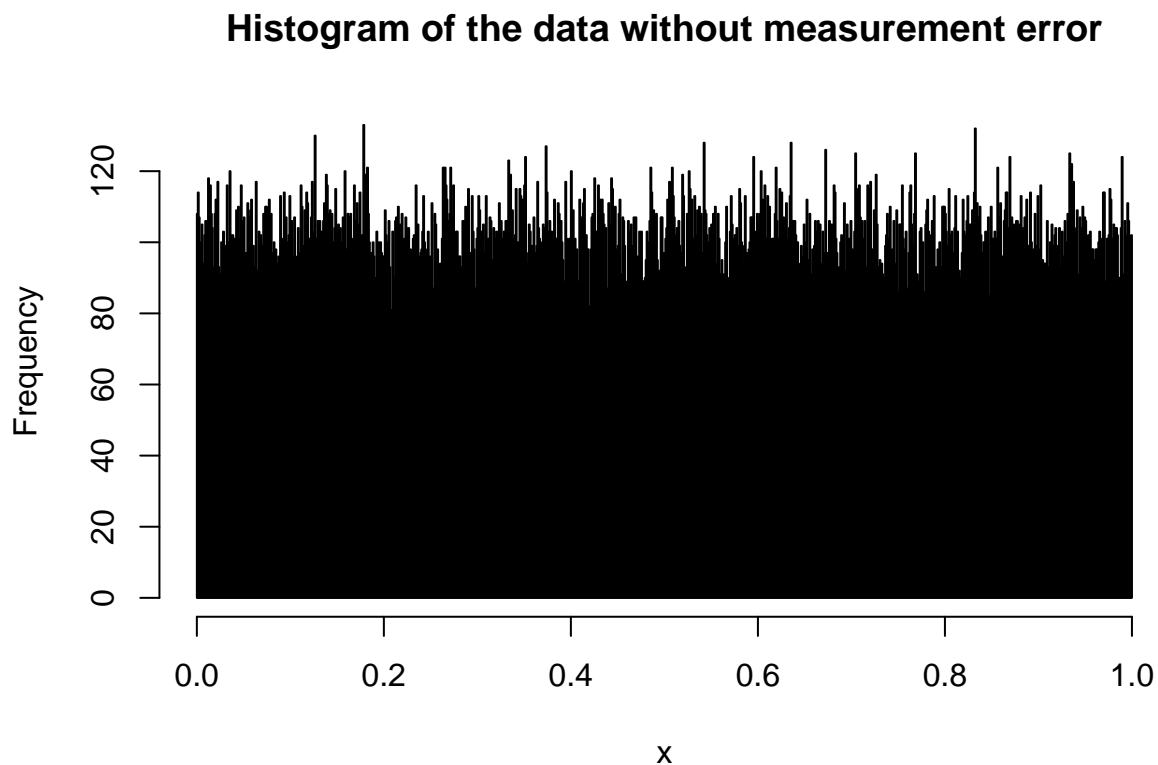
```

Let's check the results. First we want to see what the data looks like if the breathalyser does not make any error. We can see from the plot below that the data is uniformly distributed nicely.

```

hist(x, breaks = seq(0, 1, by = 0.001), col = "white",
     main = "Histogram of the data without measurement error")

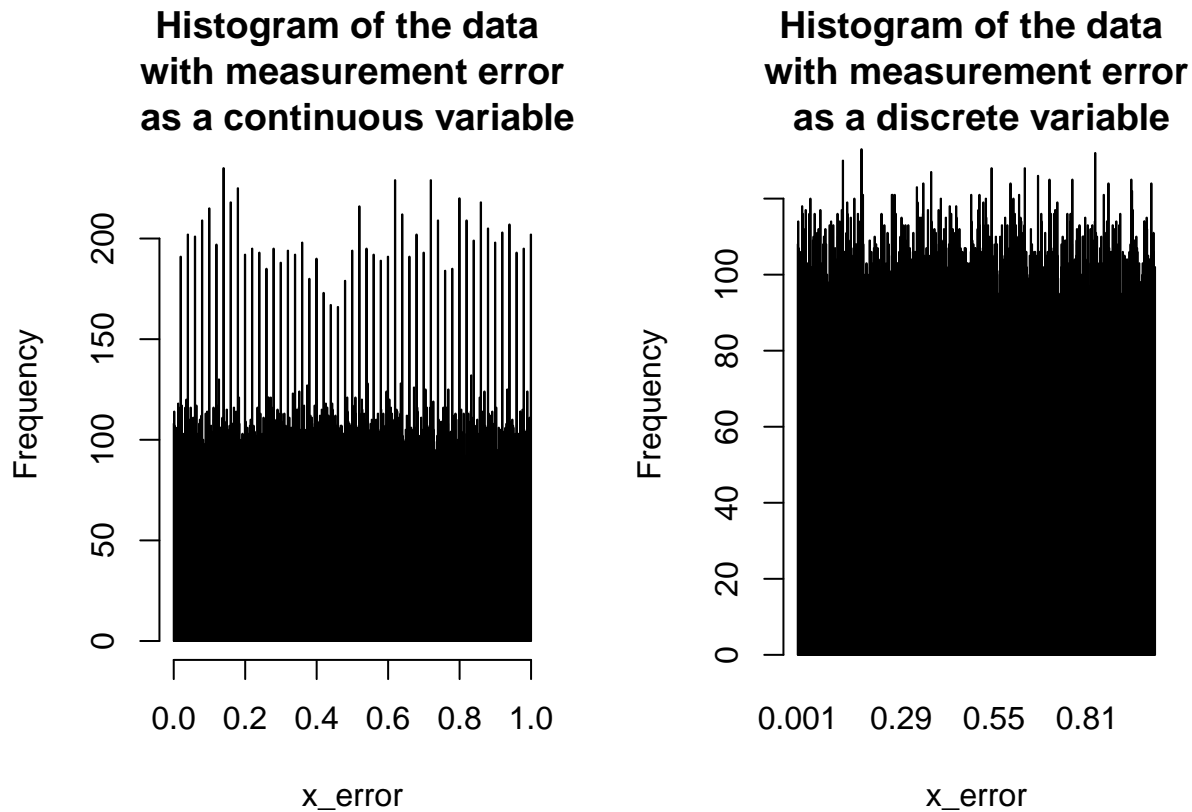
```



Here are the histograms of the data with measurement error. The plot on the left is when the data is treated as a continuous variable. The measurement error makes the last bin at the end of every 20 bins contain more values than others. The plot on the right is when the data is treated as a discrete variable. Non-random heaping is invisible since each value in this histogram has its own bin. The values measured with error that

would be binned together in the continuous histogram are separated here. This makes the heaping invisible in this plot.

```
par(mfrow = c(1, 2))
hist(x_error, breaks = seq(0, 1, by = 0.001),
     main = "Histogram of the data \n with measurement error \n as a continuous variable")
barplot(table(x_discrete), xlab = "x_error", ylab = "Frequency",
        main = "Histogram of the data \n with measurement error \n as a discrete variable")
```



The necessity of local polynomials in a donut hole model

This section gives a simulation example to illustrate why local polynomials are needed in a donut hole design. Particularly, under some data generating process, getting rid of the data near the threshold can lead to a drastic difference between the estimates given by a linear and a quadratic model.

First generating the data we want.

```
# set seed
set.seed(34827)

## simulate the data ##
```

```

# generate the running variable
n <- 100
x <- seq(-0.5, 0.5, length.out = n)
x_sq <- x^2
# generate the indicator variable, setting 0 as the cutoff point
ind <- ifelse(x < 0, 0, 1)
# mean value of the outcome
mu_left <- 1 + 0.05*x[which(ind == 0)]
mu_right <- 1.1 - 0.02*x[which(ind == 1)] + x[which(ind == 1)]^2
mu <- c(mu_left, mu_right)
# simulate outcome variable with random error ~ norm(0, 0.02)
y <- rnorm(n, mu, 0.02)

```

Here are the estimates given by the regular RDD models. We can see the linear and the quadratic models are already giving different estimates.

```

## regular RDD ##
# linear model
model_linear <- lm(y ~ ind*x)
summary(model_linear)

##
## Call:
## lm(formula = y ~ ind * x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.066646 -0.015409 -0.002732  0.012278  0.056368
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.003208   0.006792  147.713 < 2e-16 ***
## ind          0.046232   0.009605   4.813 5.52e-06 ***
## x            0.056133   0.023293   2.410  0.0179 *
## ind:x        0.456895   0.032941  13.870 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02401 on 96 degrees of freedom
## Multiple R-squared:  0.9554, Adjusted R-squared:  0.954

```

```
## F-statistic: 685.3 on 3 and 96 DF, p-value: < 2.2e-16
```

```
# quadratic model
```

```
model_qua <- lm(y ~ ind*(x + x_sq))
```

```
summary(model_qua)
```

```
##
```

```
## Call:
```

```
## lm(formula = y ~ ind * (x + x_sq))
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -0.049210 -0.013100 -0.000029  0.012250  0.049643
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  1.002073   0.008416 119.061 < 2e-16 ***
```

```
## ind          0.090294   0.011903   7.586 2.34e-11 ***
```

```
## x            0.042659   0.076987   0.554  0.581
```

```
## x_sq        -0.026679   0.147601  -0.181  0.857
```

```
## ind:x       -0.039505   0.108876  -0.363  0.718
```

```
## ind:x_sq     1.036230   0.208739   4.964 3.07e-06 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Residual standard error: 0.01982 on 94 degrees of freedom
```

```
## Multiple R-squared:  0.9702, Adjusted R-squared:  0.9686
```

```
## F-statistic: 612.5 on 5 and 94 DF, p-value: < 2.2e-16
```

We now run the donut hole models with the data points in the window $[-0.1, 0.1]$ dropped. We can see the difference between the two models gets even bigger.

```
## donut whole RDD ##
```

```
# create an index for the data to be removed
```

```
# here the data within the window [-0.1, 0.1] are dropped
```

```
index <- which(abs(x) <= 0.1)
```

```
x_donut <- x[-index]
```

```
x_sq_donut <- x_sq[-index]
```

```
ind_donut <- ind[-index]
```

```
y_donut <- y[-index]
```

```

# linear model
donut_linear <- lm(y_donut ~ ind_donut*x_donut)
summary(donut_linear)

##
## Call:
## lm(formula = y_donut ~ ind_donut * x_donut)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.059963 -0.012245  0.000161  0.013934  0.052182
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.997372   0.009385  106.277  <2e-16 ***
## ind_donut      0.010642   0.013272   0.802    0.425
## x_donut        0.040351   0.028904   1.396    0.167
## ind_donut:x_donut 0.589272   0.040876  14.416  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02131 on 76 degrees of freedom
## Multiple R-squared:  0.9703, Adjusted R-squared:  0.9692
## F-statistic: 828.7 on 3 and 76 DF,  p-value: < 2.2e-16

# quadratic model
donut_qua <- lm(y_donut ~ ind_donut*(x_donut + x_sq_donut))
summary(donut_qua)

##
## Call:
## lm(formula = y_donut ~ ind_donut * (x_donut + x_sq_donut))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.051596 -0.012810  0.000166  0.009101  0.052266
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.97677   0.02237  43.666  < 2e-16 ***

```

```

## ind_donut          0.09586    0.03163    3.030    0.00336 **
## x_donut            -0.11927    0.16139   -0.739    0.46224
## x_sq_donut         -0.26337    0.26244   -1.004    0.31888
## ind_donut:x_donut   0.24830    0.22824    1.088    0.28017
## ind_donut:x_sq_donut 1.08934    0.37115    2.935    0.00444 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02017 on 74 degrees of freedom
## Multiple R-squared:  0.9741, Adjusted R-squared:  0.9724
## F-statistic: 557.7 on 5 and 74 DF,  p-value: < 2.2e-16

```

The moral here is essentially that removing data points may influence the conclusions we draw when comparing models with different functional forms. Even if the results may be robust to different functional forms under the regular RDD design, they can still be sensitive to the functional assumption under the donut hole design. It is thus a good idea to use both linear models and polynomials in the donut hole design.