HW5

Erik Andersen

2023-11-20

```
here::i_am("HW5/HW5.Rmd")
```

```
## here() starts at /Users/erikandersen/Documents/Classes/ECON 587
```

```
# Load packages
pacman::p_load(tidyverse, magrittr, rdd, ivreg)
```

```
# Load data
oz_df = haven::read_dta(here::here('HW5', "data", "Ozier_JHR_Econ587.dta"))
```

Question 1

```
# Generate indicator for treatment status. Treated if test > 0
oz_df = oz_df |>  mutate(treatment = if_else(test > 0, 1, 0))

# Regress secondary on test score, treatment, and interaction
oz_df %>% lm(secondary ~ test*treatment,.) |> broom::tidy()
```

a)

```
## # A tibble: 4 x 5
##   term           estimate std.error statistic  p.value
##   <chr>             <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)      0.330     0.0263    12.5   1.02e-34
## 2 test             0.267     0.0688     3.88  1.07e- 4
## 3 treatment        0.165     0.0377     4.38  1.24e- 5
## 4 test:treatment   0.0151    0.0967     0.157 8.76e- 1
```

```
# Define restrictions
restrict = c(0.8, 0.4, 0.2, 0.1)

lapply(restrict, function(x){
  oz_df |> filter(abs(test) < x) %>%
    lm(secondary ~ test*treatment,.) |>
    broom::tidy()
})
```

c)

```
## [[1]]
## # A tibble: 4 x 5
##   term           estimate std.error statistic  p.value
##   <chr>             <dbl>     <dbl>     <dbl>    <dbl>
```

```
## 1 (Intercept)      0.330     0.0263    12.5   1.02e-34
## 2 test             0.267     0.0688     3.88  1.07e- 4
## 3 treatment        0.165     0.0377     4.38  1.24e- 5
## 4 test:treatment   0.0151    0.0967     0.157 8.76e- 1
##
## [[2]]
## # A tibble: 4 x 5
##   term           estimate std.error statistic  p.value
##   <chr>             <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)       0.306    0.0358     8.56  3.28e-17
## 2 test              0.117    0.168      0.699 4.85e- 1
## 3 treatment         0.186    0.0521     3.57  3.71e- 4
## 4 test:treatment    0.186    0.237      0.783 4.34e- 1
##
## [[3]]
## # A tibble: 4 x 5
##   term           estimate std.error statistic      p.value
##   <chr>             <dbl>     <dbl>     <dbl>        <dbl>
## 1 (Intercept)       0.283    0.0510     5.55  0.0000000409
## 2 test             -0.125    0.462     -0.271 0.786
## 3 treatment         0.237    0.0752     3.15  0.00170
## 4 test:treatment    0.196    0.663      0.295 0.768
##
## [[4]]
## # A tibble: 4 x 5
##   term           estimate std.error statistic  p.value
##   <chr>             <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)       0.277    0.0724     3.83  0.000153
## 2 test             -0.281    1.26      -0.223 0.824
## 3 treatment         0.224    0.114      1.97  0.0498
## 4 test:treatment    0.506    1.99       0.254 0.799
```

```
# Estimate rdd. Note default standard errors are heteroskedastic robust
RDestimate(secondary ~ test, oz_df, cutpoint = 0)
```

d)

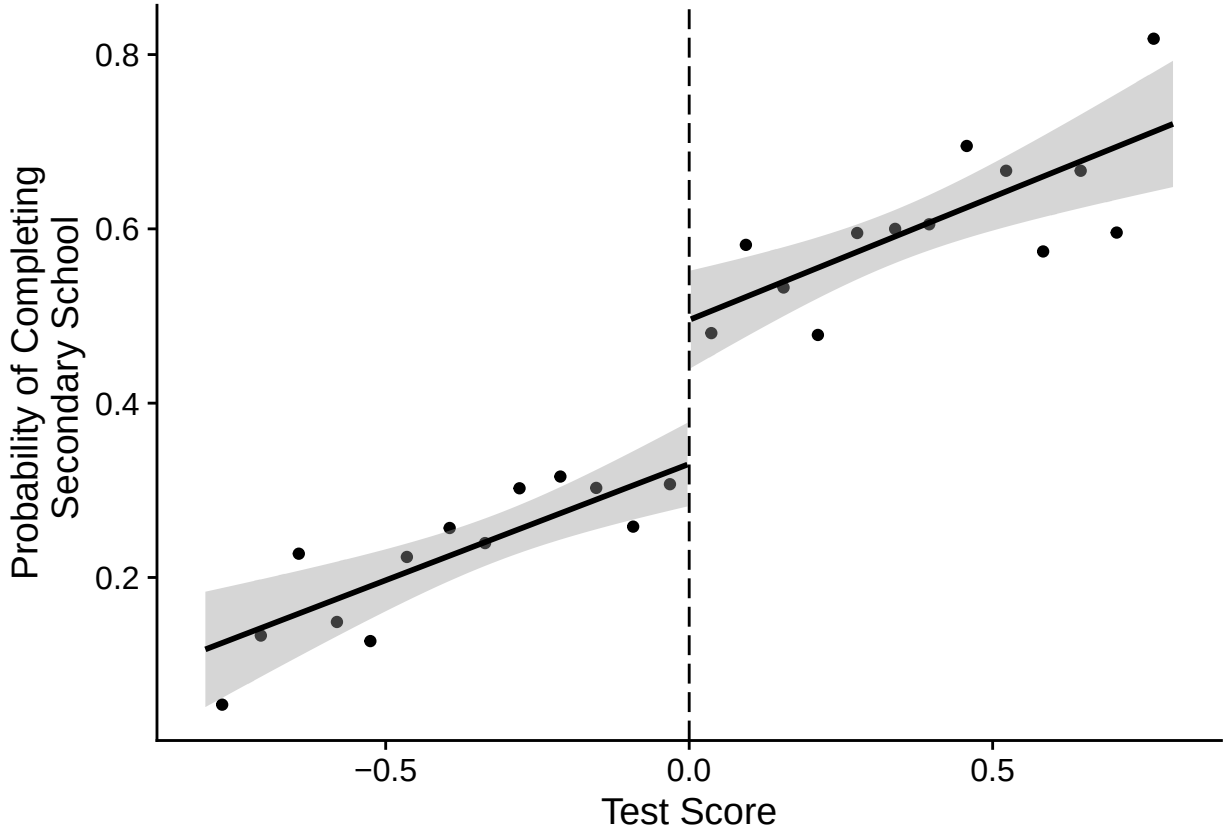
```
##
## Call:
## RDestimate(formula = secondary ~ test, data = oz_df, cutpoint = 0)
##
## Coefficients:
##      LATE    Half-BW  Double-BW
##     0.181      0.227      0.171
```

```
# Plot rdd graph
oz_df |> group_by(a = cut(test, 26)) |>
  summarise(secondary=mean(secondary), test = mean(test)) |>
  ggplot(aes(test, secondary)) +
    geom_point() +
    geom_smooth(data = oz_df |> filter(test > 0), method = 'lm', col = 'black') +
    geom_smooth(data = oz_df |> filter(test < 0), method = 'lm', col = 'black') +
```

```
    geom_vline(xintercept = 0, lty = 5) +
    xlab("Test Score") + ylab("Probability of Completing\n Secondary School") +
    cowplot::theme_cowplot()
```

e)

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggsave(here::here("HW5", "rdd_plot.pdf"))
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

```
# Rerun esimation of rdd, but override default kernal and bandwidths
RDestimate(secondary ~ test, oz_df, cutpoint = 0, kernel = 'rectangular', bw = 0.8)
```

f)

```
##
## Call:
## RDestimate(formula = secondary ~ test, data = oz_df, cutpoint = 0,
##     bw = 0.8, kernel = "rectangular")
##
## Coefficients:
##      LATE    Half-BW  Double-BW
##     0.164      0.184      0.164
```

```
ivreg(rv ~ secondary + test + female + test:treatment |
        treatment + test + female + test:treatment, data = oz_df) |> broom::tidy()
```

g)

```
## # A tibble: 5 x 5
##   term           estimate std.error statistic  p.value
##   <chr>             <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)       2.93     0.150      19.5  1.64e-77
## 2 secondary         0.726    0.311       2.33 1.98e- 2
## 3 test              0.612    0.159       3.85 1.22e- 4
## 4 female           -0.176    0.0483     -3.64 2.80e- 4
## 5 test:treatment   -0.311    0.133      -2.34 1.95e- 2
```

```
# Run estimate again, but use controls. Note the regressors are in the opposite order than the homework
RDestimate(rv ~ test + secondary, oz_df, cutpoint = 0)
```

h)

```
##
## Call:
## RDestimate(formula = rv ~ test + secondary, data = oz_df, cutpoint = 0)
##
## Coefficients:
##      LATE    Half-BW  Double-BW
##      1.47       1.32       1.14
```

```
# Replicate g result. I don't include test interacted with treatment because it gives insane results li
RDestimate(rv ~ test + secondary | female , data = oz_df, cutpoint = 0,
           kernel = 'rectangular', bw = 0.6)
```

i)

```
##
## Call:
## RDestimate(formula = rv ~ test + secondary | female, data = oz_df,
##     cutpoint = 0, bw = 0.6, kernel = "rectangular")
##
## Coefficients:
##      LATE    Half-BW  Double-BW
##     0.744      1.266      0.670
```

Question 2

```
# Load data
sim_df = haven::read_dta(here::here("HW5", "data", "RD_Manip_Econ587.dta"))
```

```
# Plot kernel densities for all four measures of wealth for bandwidths 0.1, 0.05, 0.01
lapply(c(0.1, 0.05, 0.01),
       function(x) {
         sim_df |> ggplot(aes(reportwealth1)) +
```

```
            geom_density(bw = x) +
            geom_density(aes(reportwealth2), bw = x, col = 'red') +
            geom_density(aes(reportwealth3), bw = x, col = 'blue') +
            geom_density(aes(reportwealth4), bw = x, col = 'green') +
            xlab("Wealth Kernel Density") + ylab("") +
            ggtitle(paste("Wealth Kernel Density with Bandwidth =", x)) +
            labs(caption = "Black: Reported Wealth 1\n Red: Reported Wealth 2\n
                 Blue: Reported Wealth 3\n Green: Reported Wealth 4") +
            cowplot::theme_cowplot()
          ggsave(here::here("HW5", paste(
            "Kernel_wealth_bandwidth_", x, ".pdf", sep = ""
          )))
        }
)
```

a)

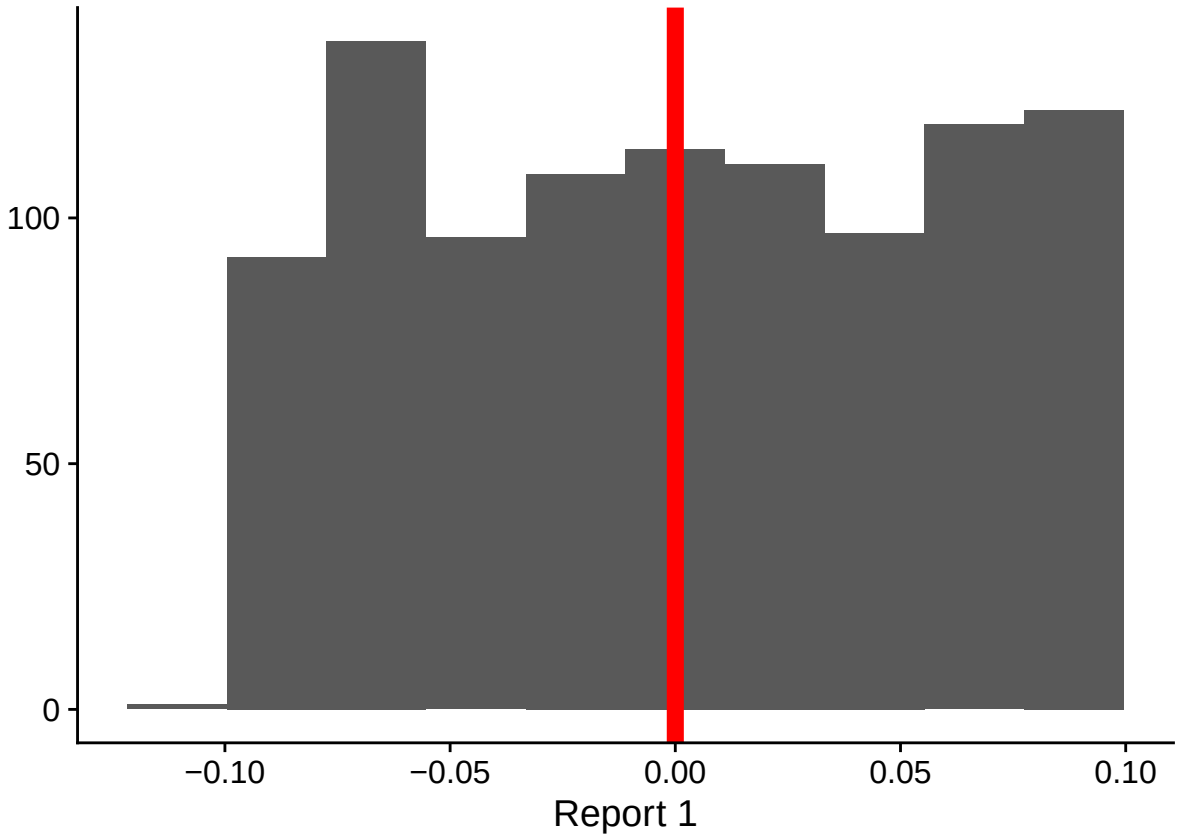
```
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
```

```
## [[1]]
## [1] "/Users/erikandersen/Documents/Classes/ECON 587/HW5/Kernel_wealth_bandwidth_0.1.pdf"
##
## [[2]]
## [1] "/Users/erikandersen/Documents/Classes/ECON 587/HW5/Kernel_wealth_bandwidth_0.05.pdf"
##
## [[3]]
## [1] "/Users/erikandersen/Documents/Classes/ECON 587/HW5/Kernel_wealth_bandwidth_0.01.pdf"
```

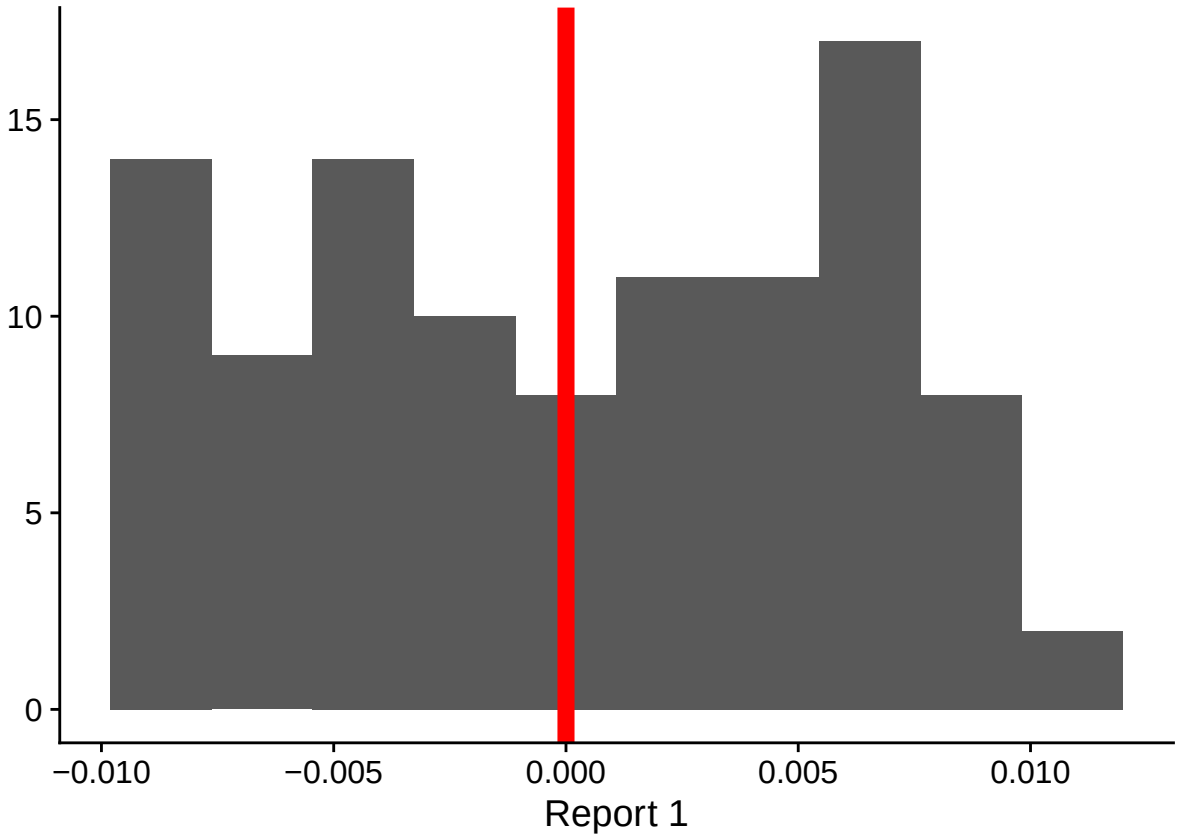
```
# Test smoothness around the cuttoff using ocular methods. Unfortunately to loop this over the four sce
bandwidths = c(0.1, 0.01, 0.05)

lapply(bandwidths, function(x){
  sim_df |> filter(abs(reportwealth1) < x) |>
    select(reportwealth1)|>
    ggplot(aes(reportwealth1)) +
        geom_histogram(bins = 10) +
        geom_vline(xintercept = 0, color = 'red', lwd = 3) +
        xlab("Report 1") + ylab("") + cowplot::theme_cowplot()
})
```

b)

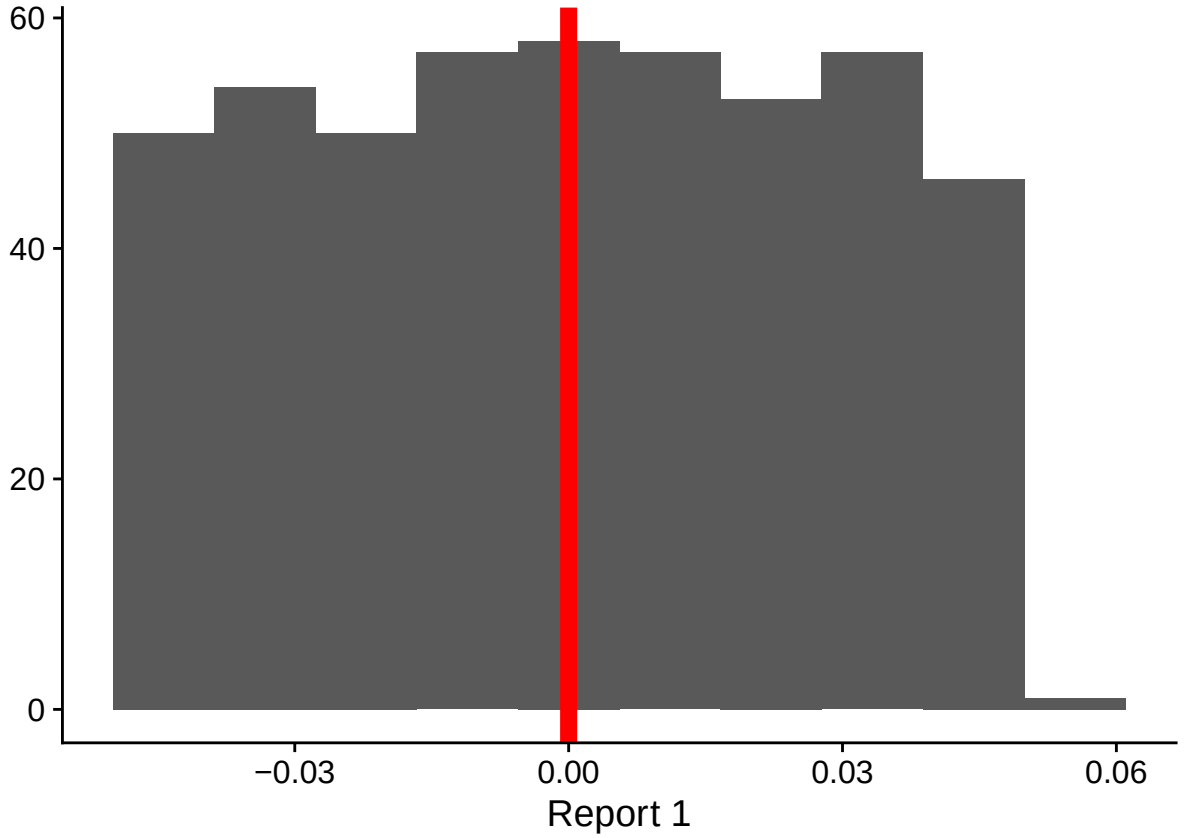
```
## [[1]]
```




```
##
## [[2]]
```

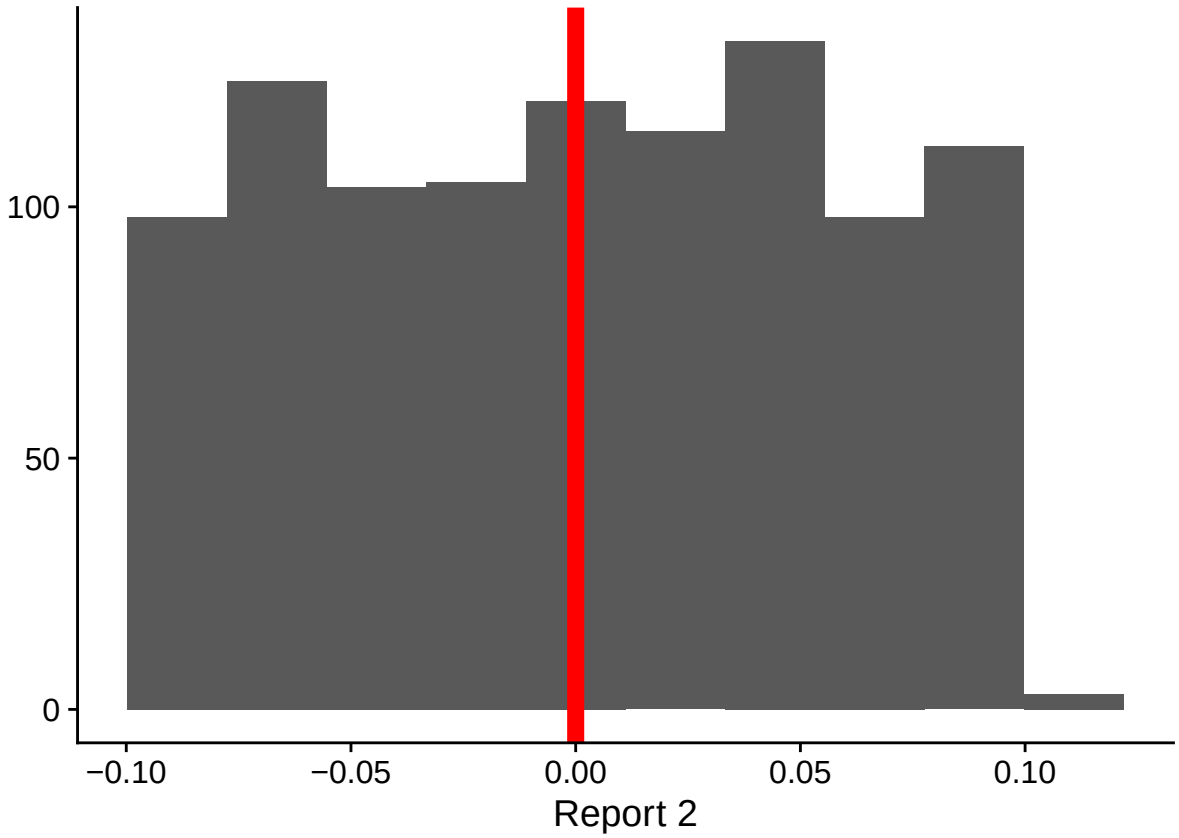



```
##
## [[3]]
```

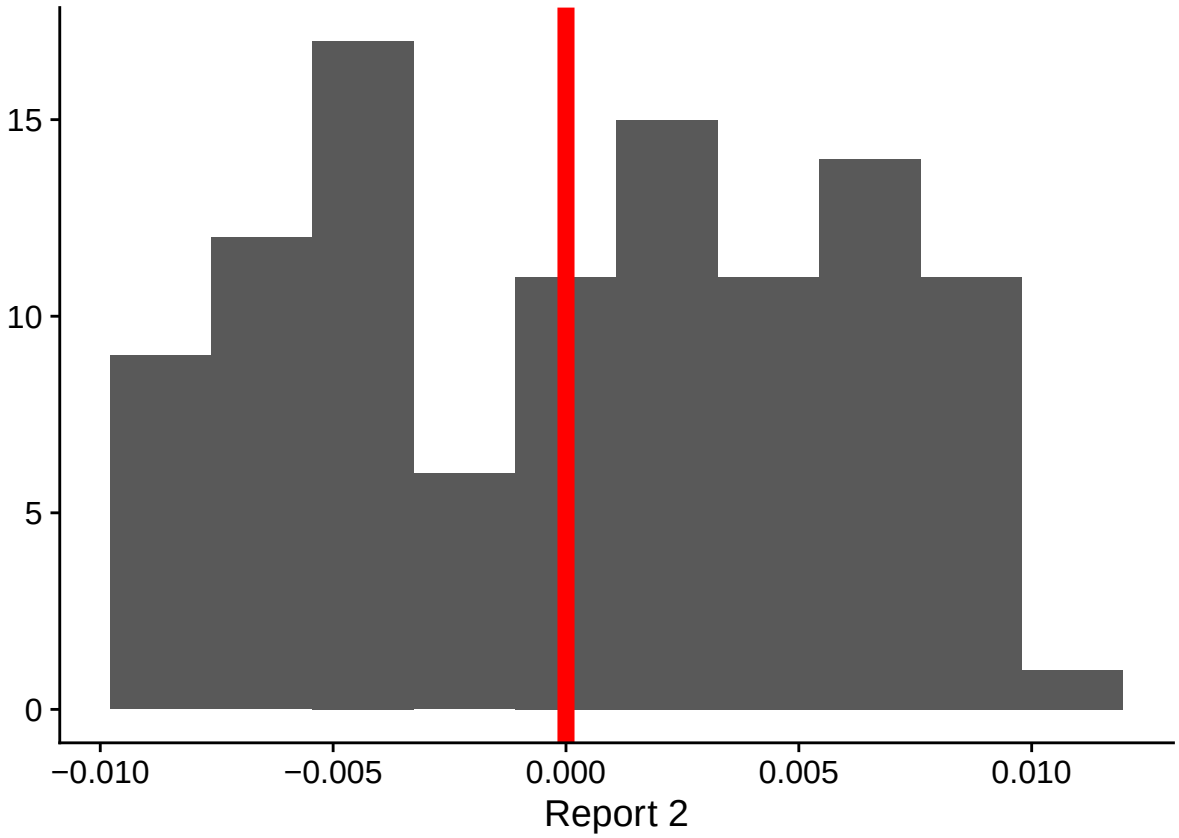



```
lapply(bandwidths, function(x){
  sim_df |> filter(abs(reportwealth2) < x) |>
    select(reportwealth2)|>
    ggplot(aes(reportwealth2)) +
        geom_histogram(bins = 10) +
        geom_vline(xintercept = 0, color = 'red', lwd = 3)+
        xlab("Report 2") + ylab("")+ cowplot::theme_cowplot()
})
```

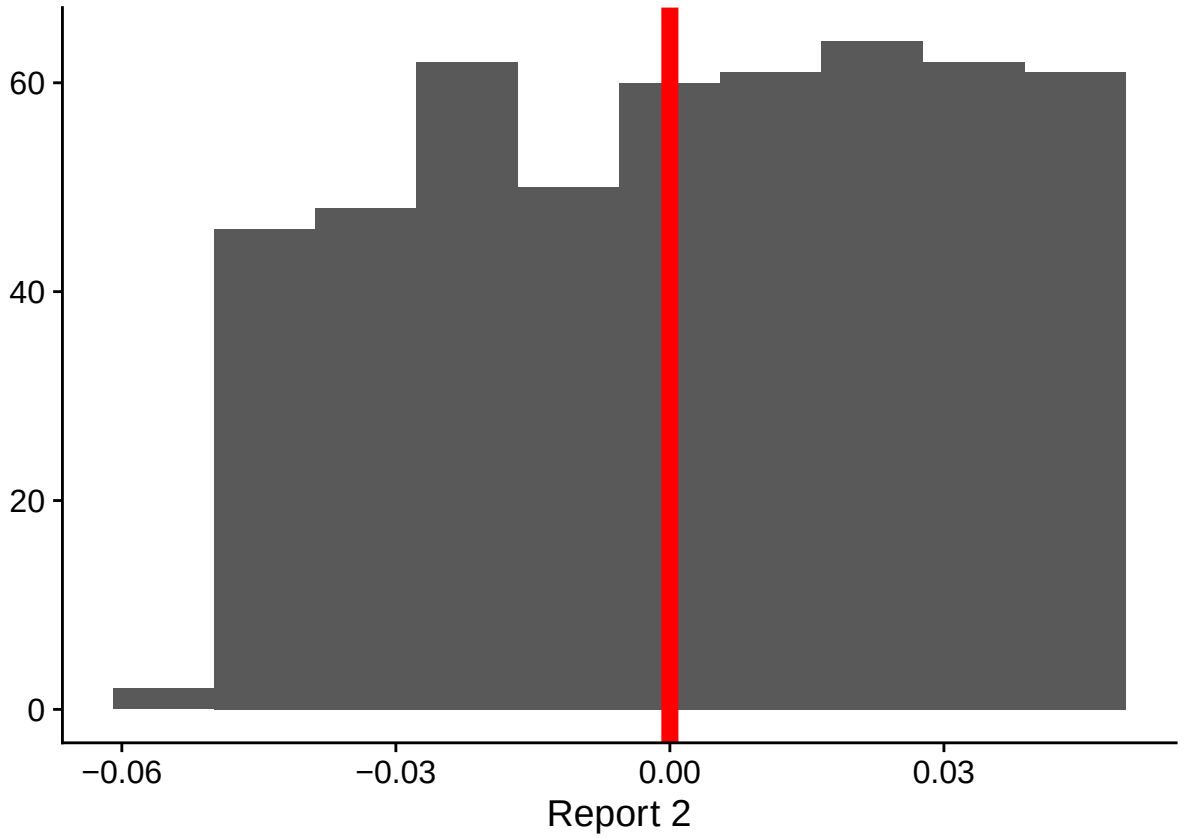
```
## [[1]]
```




```
##
## [[2]]
```

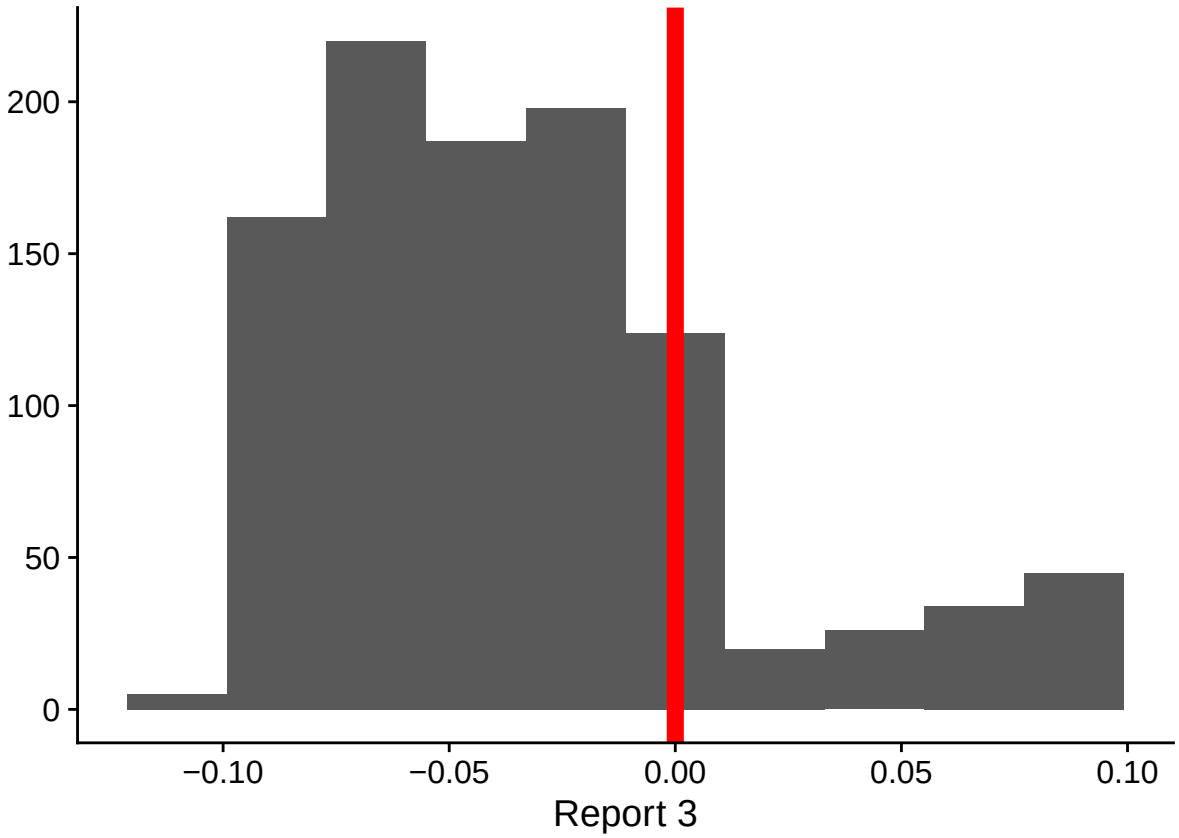



```
##
## [[3]]
```

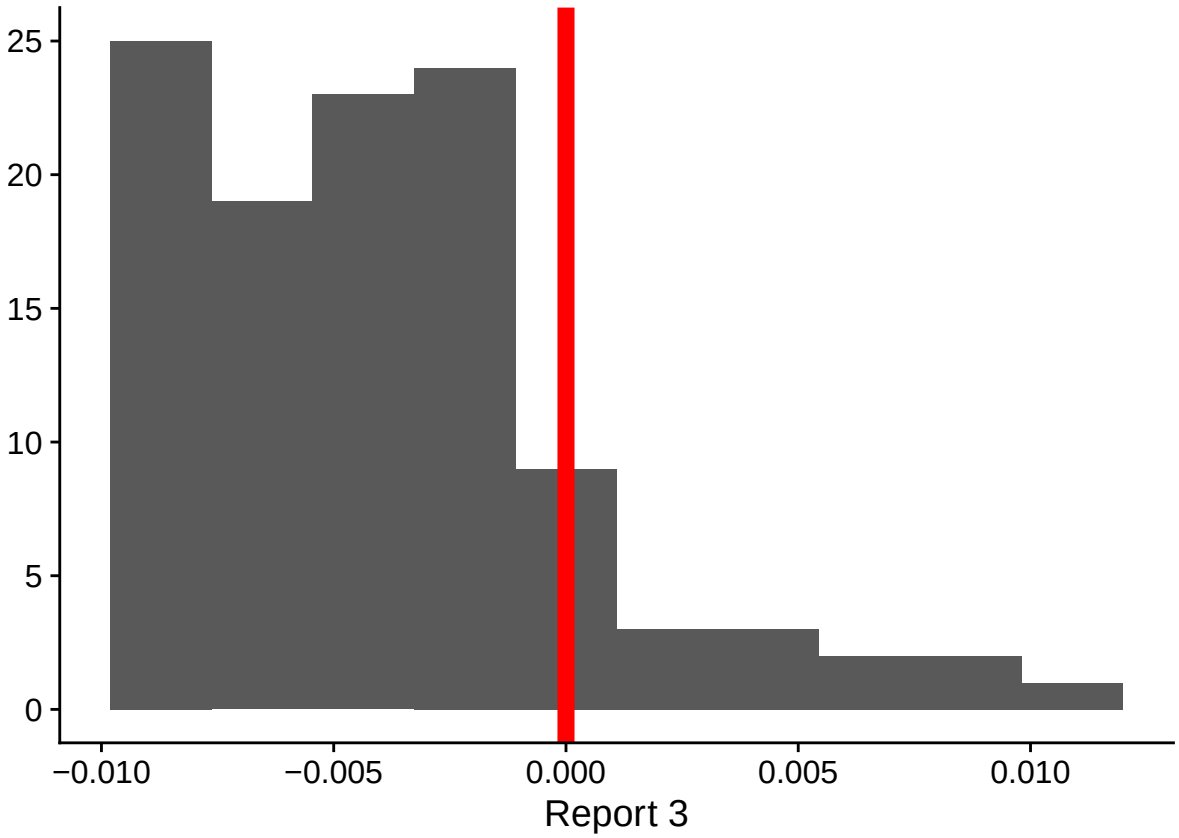



```
lapply(bandwidths, function(x){
  sim_df |> filter(abs(reportwealth3) < x) |>
    select(reportwealth3)|>
    ggplot(aes(reportwealth3)) +
        geom_histogram(bins = 10) +
        geom_vline(xintercept = 0, color = 'red', lwd = 3)+
        xlab("Report 3") + ylab("")+ cowplot::theme_cowplot()
})
```

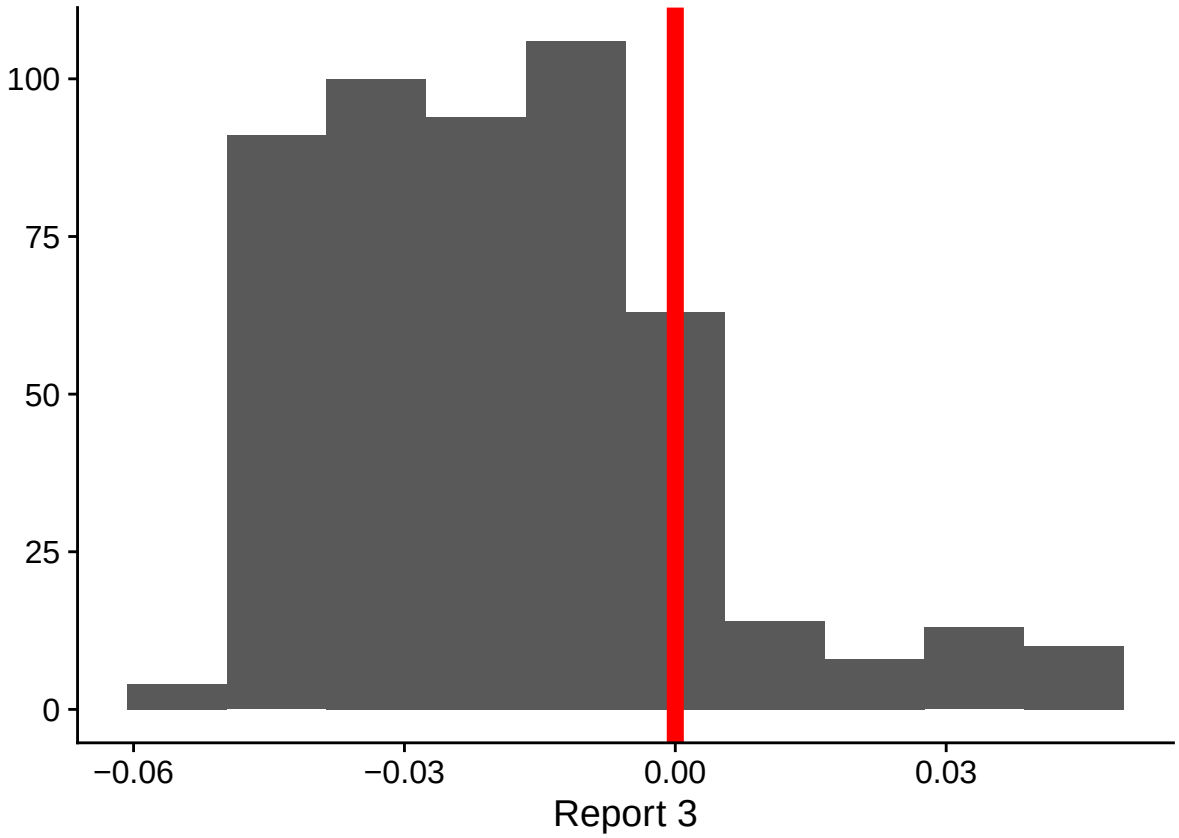
```
## [[1]]
```




```
##
## [[2]]
```

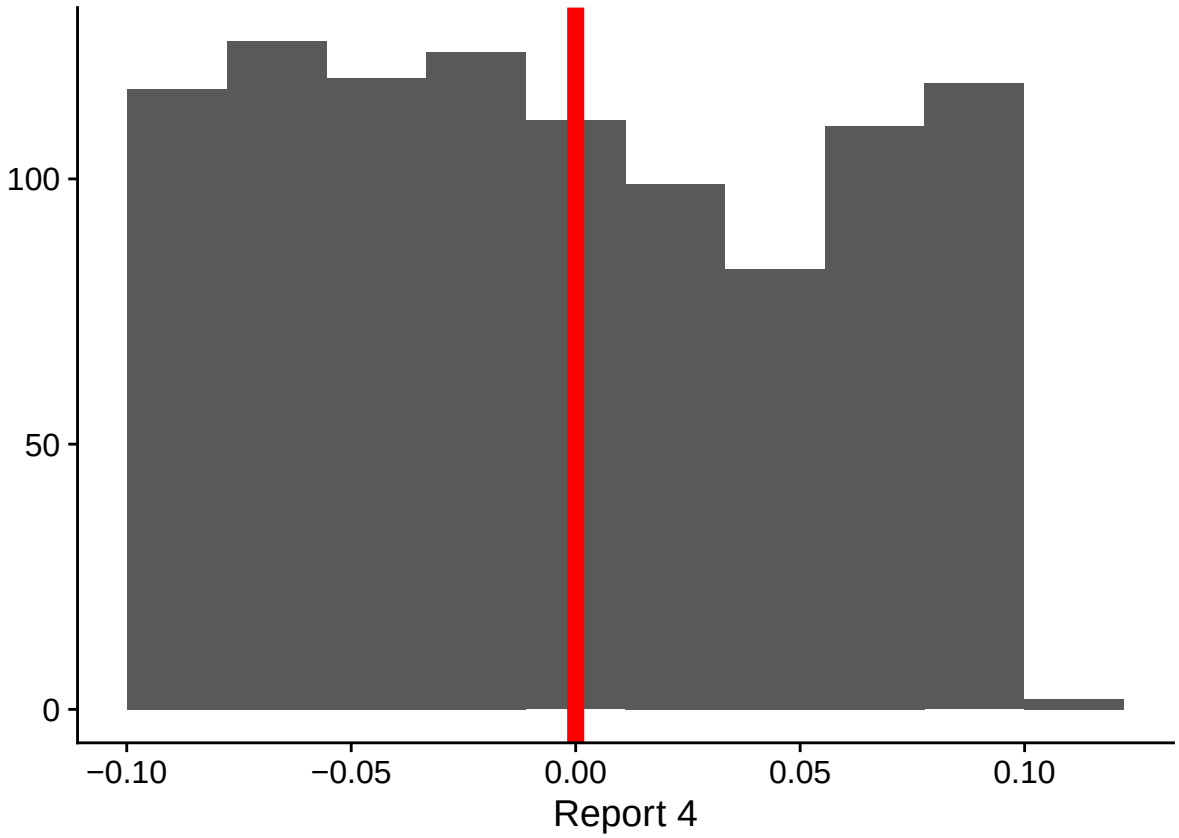



```
##
## [[3]]
```

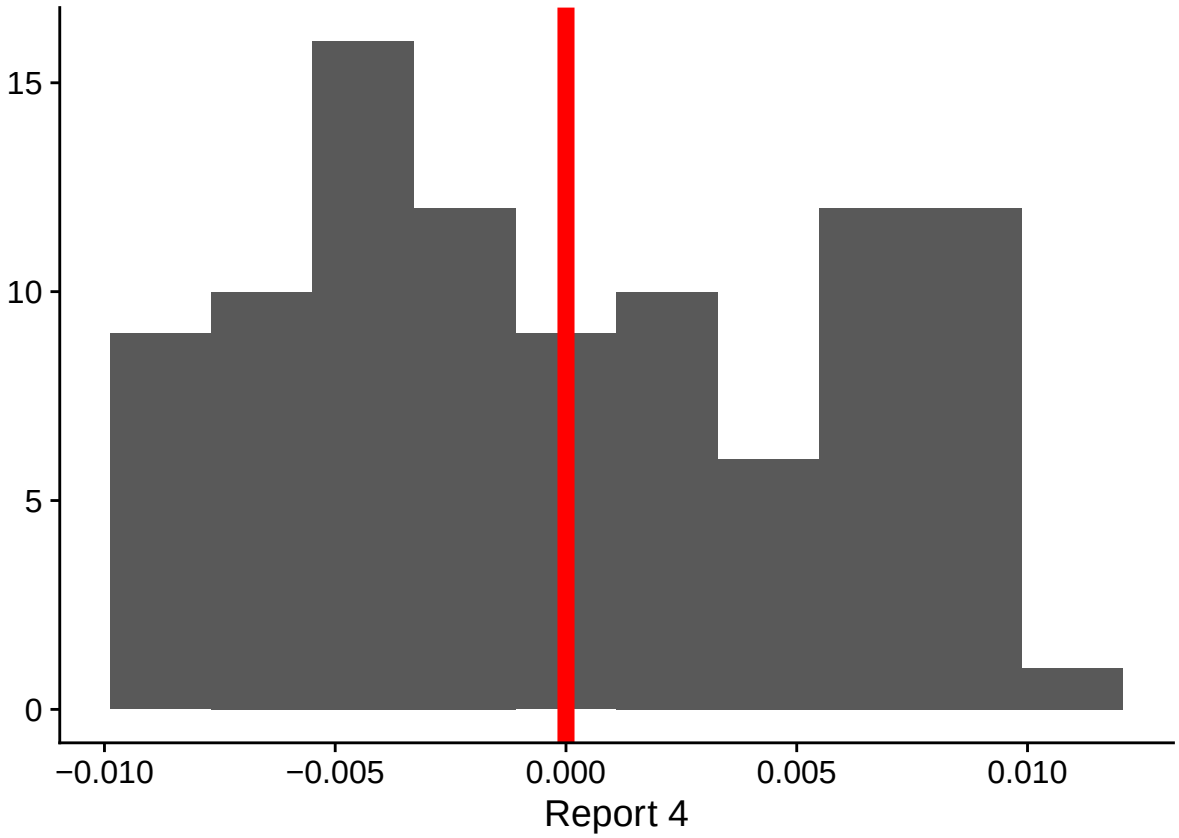



```
lapply(bandwidths, function(x){
  sim_df |> filter(abs(reportwealth4) < x) |>
    select(reportwealth4)|>
    ggplot(aes(reportwealth4)) +
        geom_histogram(bins = 10) +
        geom_vline(xintercept = 0, color = 'red', lwd = 3)+
        xlab("Report 4") + ylab("")+ cowplot::theme_cowplot()
})
```

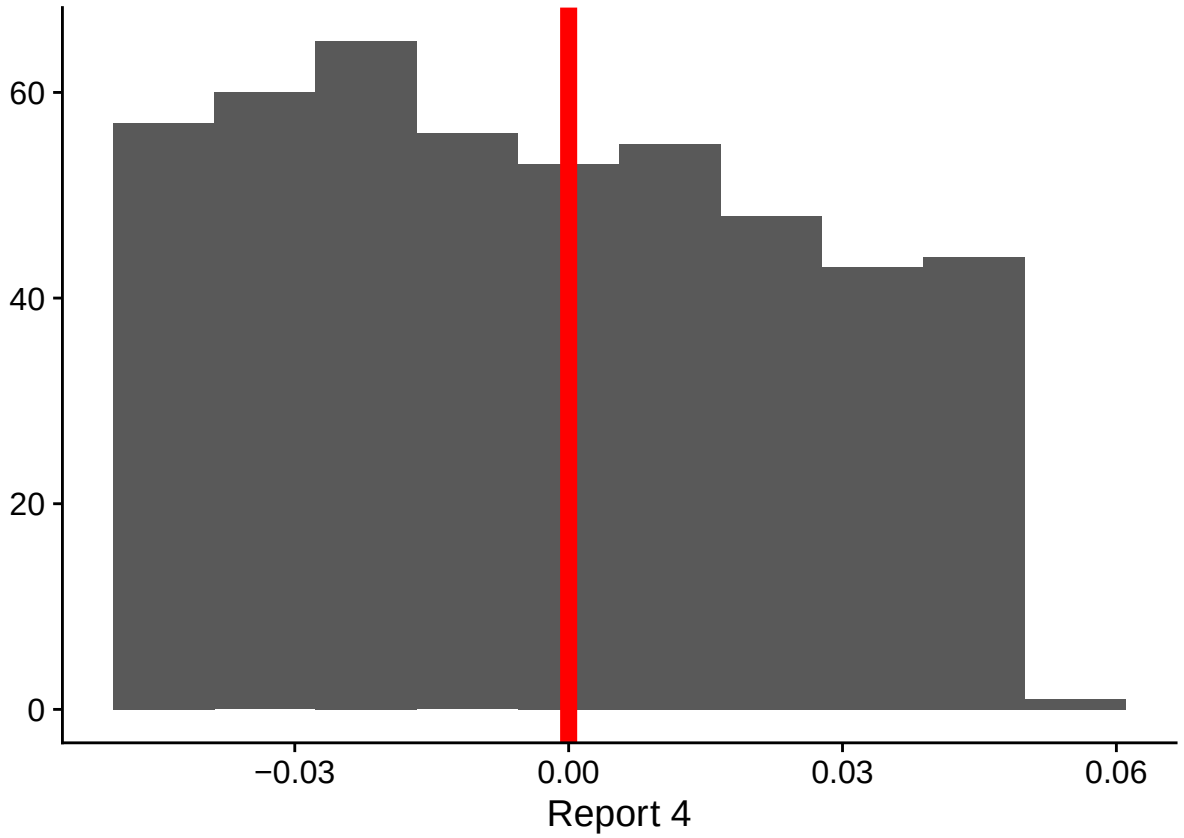
```
## [[1]]
```



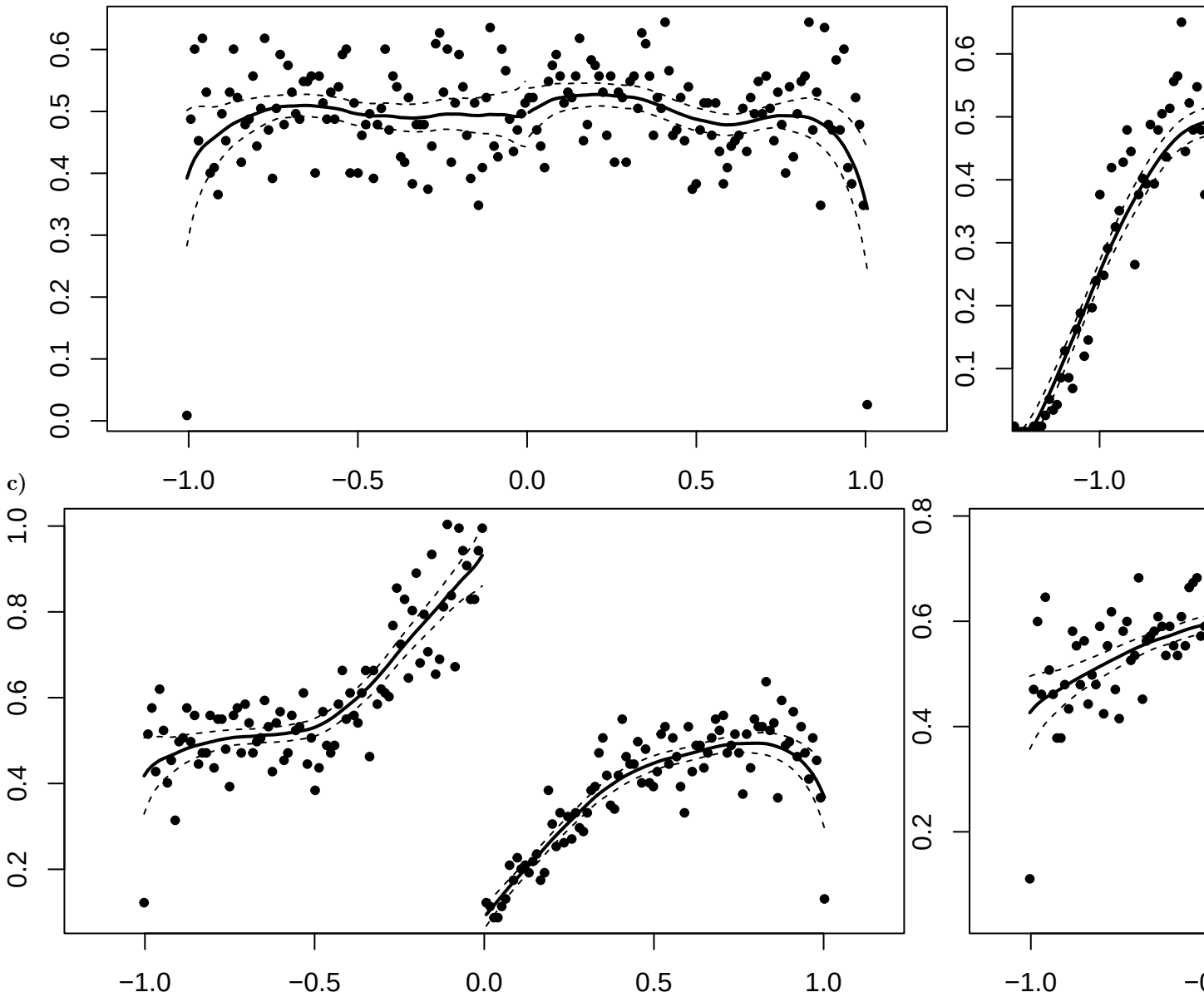

```
##
## [[2]]
```




```
##
## [[3]]
```




```
# McCrary kovac test
vars = paste0("reportwealth", 1:4)

# P-values are reported
sapply(vars, function(x){
  DCdensity(sim_df[[x]], cutpoint = 0)
})
```




```
## reportwealth1 reportwealth2 reportwealth3 reportwealth4
##     9.792e-01     3.191e-01     8.155e-56     4.972e-02
```

```r
# Generate treatment variables equal to one if repored wealth is below 0
sim_df %<>% mutate(treat1 = if_else(reportwealth1 < 0, 1, 0),
                   treat2 = if_else(reportwealth2 < 0, 1, 0),
                   treat3 = if_else(reportwealth3 < 0, 1, 0),
                   treat4 = if_else(reportwealth4 < 0, 1, 0))

# For each reported wealth generate the following
# new-wealth = true-wealth + 0.2*treat + e    e ~ N(0, 0.01)
sim_df %<>% mutate(new_wealth1 = truewealth + 0.2 * treat1 + rnorm(nrow(sim_df), 0, sqrt(0.01)),
                   new_wealth2 = truewealth + 0.2 * treat2 + rnorm(nrow(sim_df), 0, sqrt(0.01)),
                   new_wealth3 = truewealth + 0.2 * treat3 + rnorm(nrow(sim_df), 0, sqrt(0.01)),
                   new_wealth4 = truewealth + 0.2 * treat4 + rnorm(nrow(sim_df), 0, sqrt(0.01)))
```

```r
# Summarize the new wealths
lapply(1:4, function(i){
  pdf(here::here("HW5", paste0("new-wealth", i, ".pdf")))
  plot(c(sim_df[paste0("reportwealth", i)]) |> unlist(),
       c(sim_df[paste0('new_wealth', i)]) |> unlist(),
       xlab = paste0("Report Wealth", i),
       ylab = paste0("New Wealth", i))
  dev.off()
})
```

d)

```
## [[1]]
## pdf
##   2
##
## [[2]]
## pdf
##   2
##
## [[3]]
## pdf
##   2
##
## [[4]]
## pdf
##   2
```

```r
lapply(1:4, function(i){

  # Generate formula
  y = paste0("new_wealth", i)
  treat = paste0("treat", i)
  report = paste0("reportwealth", i)
  x = paste(treat, report, sep = "*")
  formula = paste(y, x, sep = "~")

  # Run regression
  lm(formula, sim_df) |> summary()
})
```

e)

```
## [[1]]
##
## Call:
## lm(formula = formula, data = sim_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3850 -0.0685  0.0008  0.0684  0.3467
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)          0.002187   0.002767    0.79     0.43
## treat1               0.197881   0.003957   50.01   <2e-16 ***
## reportwealth1        1.000628   0.004853  206.20   <2e-16 ***
## treat1:reportwealth1 0.000624   0.006888    0.09     0.93
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0994 on 9996 degrees of freedom
## Multiple R-squared:  0.961,  Adjusted R-squared:  0.961
## F-statistic: 8.16e+04 on 3 and 9996 DF,  p-value: <2e-16
##
##
## [[2]]
##
## Call:
## lm(formula = formula, data = sim_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5107 -0.0953  0.0025  0.0964  0.4747
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.01287    0.00385    3.35  0.00083 ***
## treat2                0.16416    0.00552   29.76  < 2e-16 ***
## reportwealth2         0.95537    0.00665  143.73  < 2e-16 ***
## treat2:reportwealth2 -0.02087    0.00944   -2.21  0.02701 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.14 on 9996 degrees of freedom
## Multiple R-squared:  0.922,  Adjusted R-squared:  0.922
## F-statistic: 3.95e+04 on 3 and 9996 DF,  p-value: <2e-16
##
##
## [[3]]
##
## Call:
## lm(formula = formula, data = sim_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4970 -0.0923 -0.0135  0.0697  0.9731
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -0.00345    0.00572    -0.6     0.55
## treat3                0.33676    0.00665    50.7   <2e-16 ***
## reportwealth3         1.00270    0.00903   111.1   <2e-16 ***
## treat3:reportwealth3  0.16784    0.01104    15.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.146 on 9996 degrees of freedom
```

```
## Multiple R-squared:  0.917,  Adjusted R-squared:  0.917
## F-statistic: 3.69e+04 on 3 and 9996 DF,  p-value: <2e-16
##
##
## [[4]]
##
## Call:
## lm(formula = formula, data = sim_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5054 -0.1138 -0.0254  0.0828  0.7491
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.15421    0.00489    31.5   <2e-16 ***
## treat4                0.18184    0.00661    27.5   <2e-16 ***
## reportwealth4         0.86137    0.00926    93.0   <2e-16 ***
## treat4:reportwealth4  0.28481    0.01217    23.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.167 on 9996 degrees of freedom
## Multiple R-squared:  0.891,  Adjusted R-squared:  0.891
## F-statistic: 2.72e+04 on 3 and 9996 DF,  p-value: <2e-16
```

```r
lapply(1:4, function(i){

  # Generate formula
  y = paste0("new_wealth", i)
  x = paste0("reportwealth", i)
  formula = paste(y, x, sep = "~")

  # Run rdd
  RDestimate(formula, sim_df)
})
```

f)

```
## [[1]]
##
## Call:
## RDestimate(formula = formula, data = sim_df)
##
## Coefficients:
##      LATE    Half-BW  Double-BW
##    -0.192     -0.186     -0.197
##
##
## [[2]]
##
## Call:
## RDestimate(formula = formula, data = sim_df)
##
```

```
## Coefficients:
##      LATE    Half-BW  Double-BW
##    -0.206     -0.192     -0.203
##
##
## [[3]]
##
## Call:
## RDestimate(formula = formula, data = sim_df)
##
## Coefficients:
##      LATE    Half-BW  Double-BW
##    -0.302     -0.288     -0.316
##
##
## [[4]]
##
## Call:
## RDestimate(formula = formula, data = sim_df)
##
## Coefficients:
##      LATE    Half-BW  Double-BW
##    -0.200     -0.202     -0.193
```