Econ 675 Assignment 6

Nathan Mather[*]

March 16, 2020

Contents

1 Q1 continuity-Based Identification in SRD Designs

1.1 Q1.1

$$\tau_{SRD} = \lim_{\epsilon \to 0^+} \mathrm{E}[Y_i | \tilde{X}_i = \epsilon] - \lim_{\epsilon \to 0^+} \mathrm{E}[Y_i | \tilde{X}_i = -\epsilon]$$

Using definition of x tilde and the given assumption we get:

$$\tau_{SRD} = \lim_{\epsilon \to 0^+} \mathrm{E}[Y_i | X_i = c+\epsilon] - \lim_{\epsilon \to 0^+} \mathrm{E}[Y_i | X_i = c-\epsilon] = \lim_{\epsilon \to 0^+} \mathrm{E}[Y_{1i}(c) | X_i = c+\epsilon] - \lim_{\epsilon \to 0^+} \mathrm{E}[Y_{0i}(c) | X_i = c-\epsilon]$$

$$= \mathrm{E}[[Y_{1i}(c) - [Y_{0i}(c) | X_i = c]$$

[*]Shouts out to Ani for the help with question 1 and some help on Q2, Thank you to Tyler for help with STATA. All credit goes to Tyler for STATA because my brain does not understand STATA, and the Thanks to R for being infinitely better than STATA

1.2 Q1.2

$$\lim_{\epsilon \to 0^+} \mathrm{E}[Y_i | \tilde{X}_i = \epsilon] = \mathrm{E}[Y_{1i}(C_i) | X_i = C_i]$$

Where the last equality follows similar calculation to part 1.

$$= \sum_{c \in C} \mathrm{E}[Y_{1i}(C_i) | X_i = c, C_i = c] \mathrm{P}[X_i = c, C_i = c] = \sum_{c \in C} \mathrm{E}[Y_{1i}(C_i) | X_i = c, C_i = c] \frac{f_{X|C}(C|C) \mathrm{P}[C_i = c]}{\sum_{c \in C} f_{X|C}(C|C) \mathrm{P}[C_i = c]}$$

A similar calculation gives us the other term and combining them gives us the desired result. This is just the weighted average of the treatment effects and each cutoff which makes a lot of sense.

1.3 Q1.3

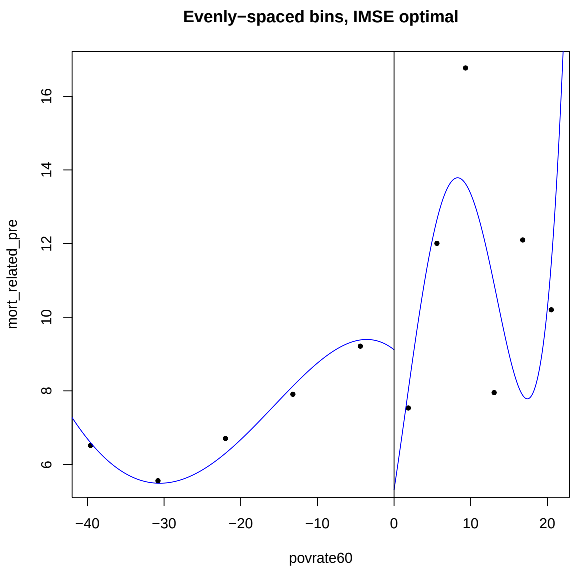
If I understand this set up correctly it is just that now the treatment effect not only varies across cutoffs but now varies depending on my individual characteristics like race or gender. I can't quite get the math figured out but I think the result is intuitive. We are now averaging over the different cutoffs and integrating over the individual characteristics. So we are essentially getting the weighted average treatment across cutoffs and characteristics.

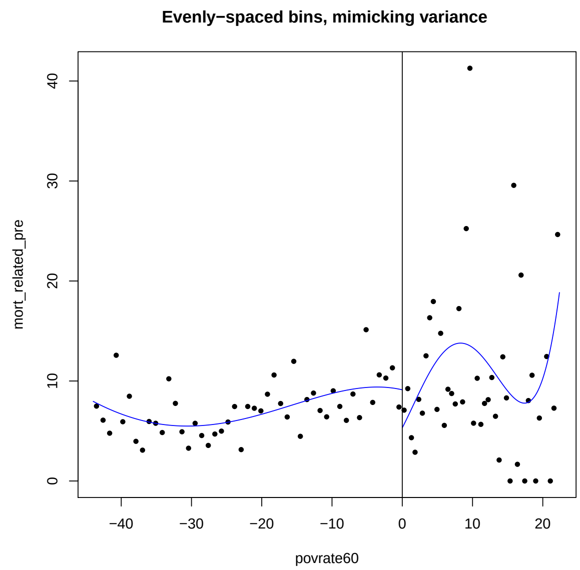
2 Question 2: The Effect of Head Start on Child Mortality

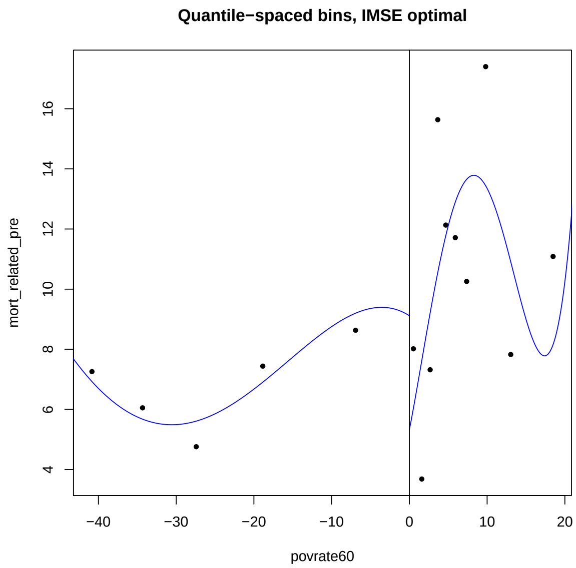
I didn't include the duplicate STATA plots and tables because there is already a lot going on, but the code is in the appendix. Results were generally the same except where I used robust standard errors in Stata and not in R.

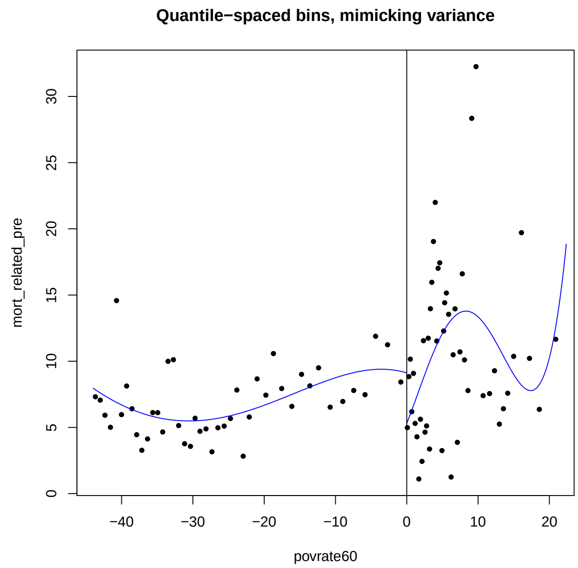
2.1 Q2.1 RD Plots and Falsification Tests

2.1.1 Q2.1.1



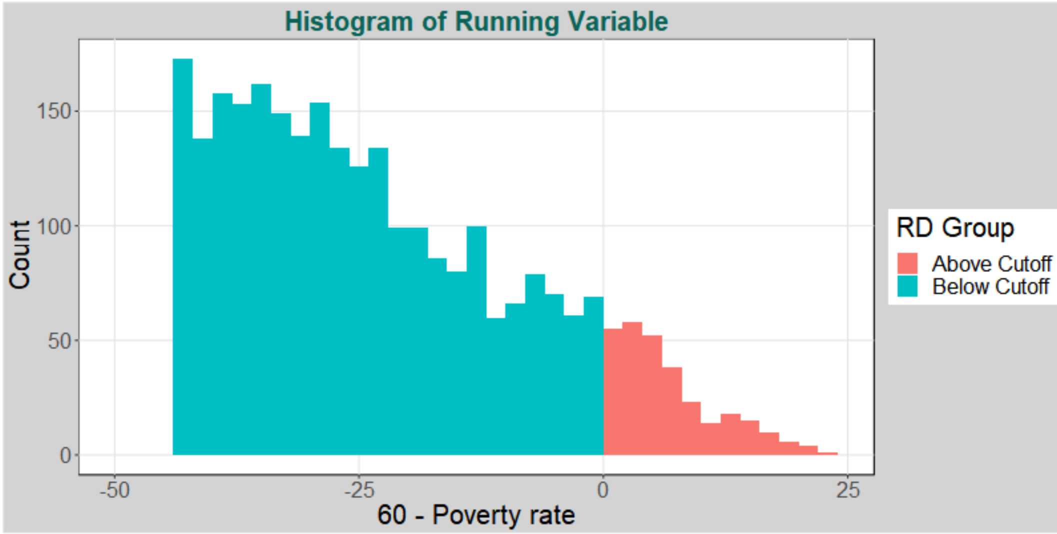






Since we are using pre treatment variables we shouldn't see much of a jump at the cutoff and we generally don't. The global polynomial is giving some illusion of a jump but this could mostly be because of what Matias talked about in class. How Polynomials tend to get less accurate towards the edges.

2.1.2 Q2.1.2



local binomial test

Bandwidth	Number Below Cutoff	Number Above Cutoff	binomial P Valu
0.40	6	8	0.79
0.60	9	10	1.00
0.80	12	12	1.00
1.00	18	16	0.86
1.20	20	20	1.00
1.40	24	22	0.88
1.60	28	24	0.68
1.80	32	27	0.60
2.00	35	29	0.53
2.20	43	33	0.30
2.40	44	35	0.37
2.60	51	38	0.20
2.80	53	40	0.21
3.00	53	40	0.21
3.20	54	45	0.42
3.40	58	47	0.33
3.60	62	49	0.25
3.80	64	51	0.26
4.00	69	55	0.24

The histogram suggests people were not selecting accross the cutoff. If they were we would expect to see bunching or large spike in the number of people on one side of the cutoff. The local binomial tests continue to support this idea.

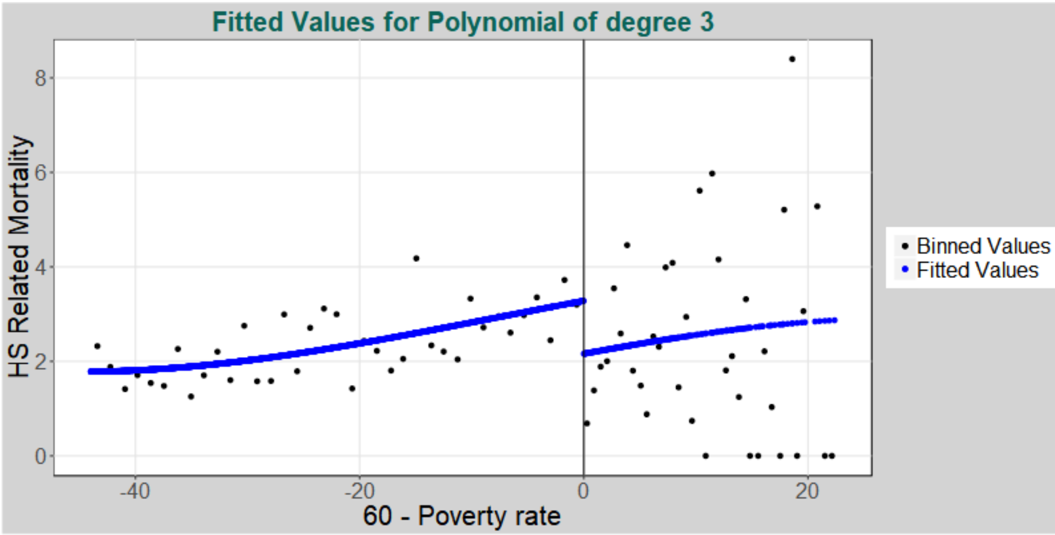
2.2 Q2.2 Global and Flexible Parametric Methods

2.2.1 Q2.2.1

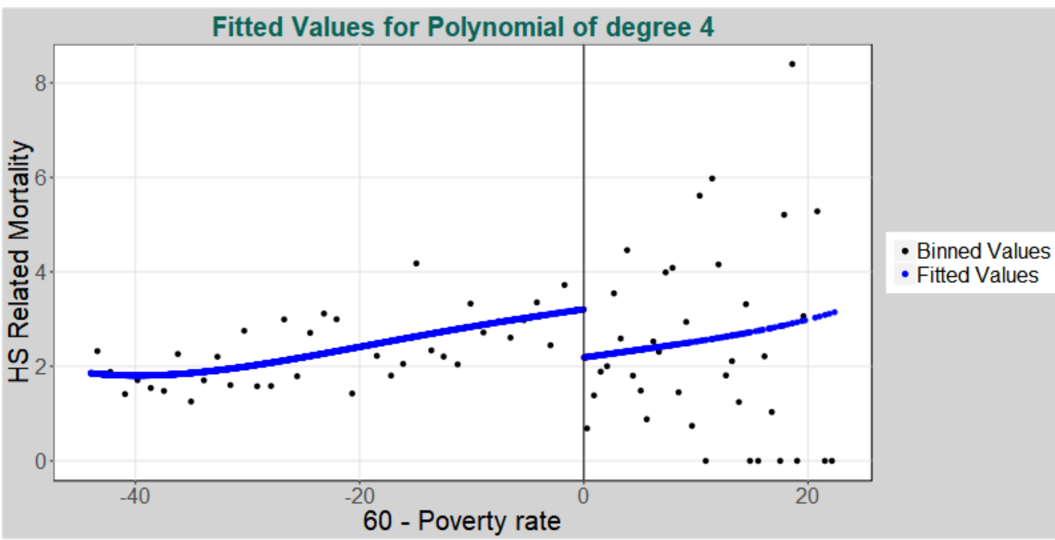
global polynomial fit

value	Polynomial 3	Polynomial 4	Polynomial 5	Polynomial 6
Estimate	-1.12	-1.02	-1.66	-1.75
Standard Error	0.67	0.76	0.86	0.87

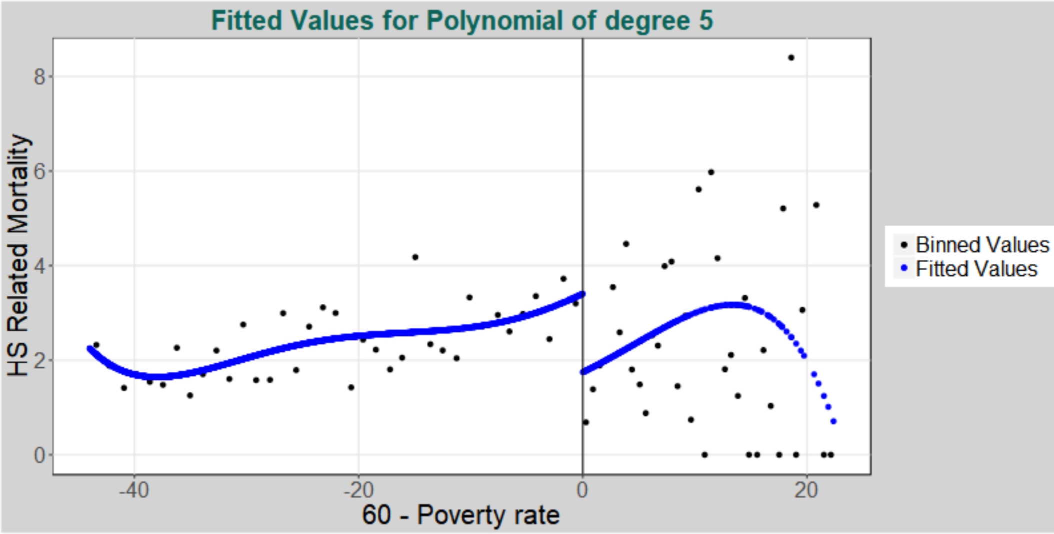
.

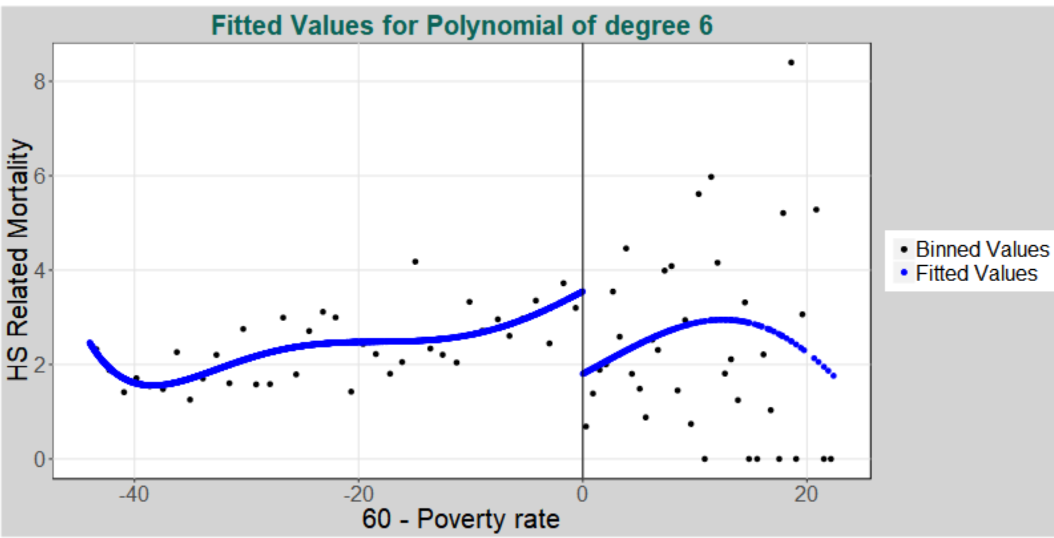


Fitted Values for Polynomial of degree 3
HS Related Mortality
60 - Poverty rate
Binned Values
Fitted Values



Fitted Values for Polynomial of degree 4
HS Related Mortality
60 - Poverty rate
Binned Values
Fitted Values





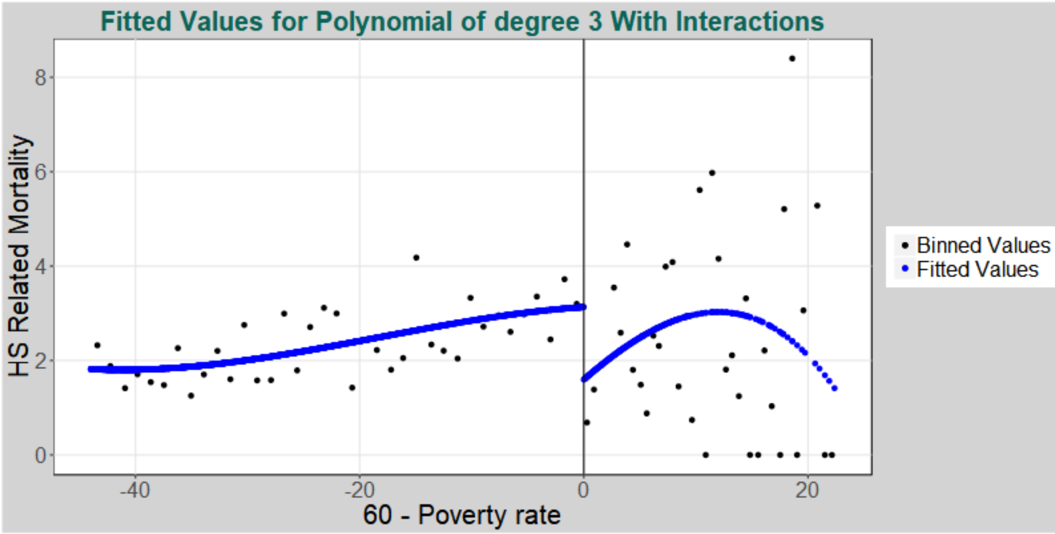
The results suggest there is an effect, i.e. a drop in mortality. The results should not be trusted though because global estimation in general is not reliable. We are stretching the assumptions of the RD design be assuming here that people on either side of the cutoff are roughly identical along the entire support of the data. It is likely true just around the cutoff but becomes less plausible as the bandwidth expands to the full support.

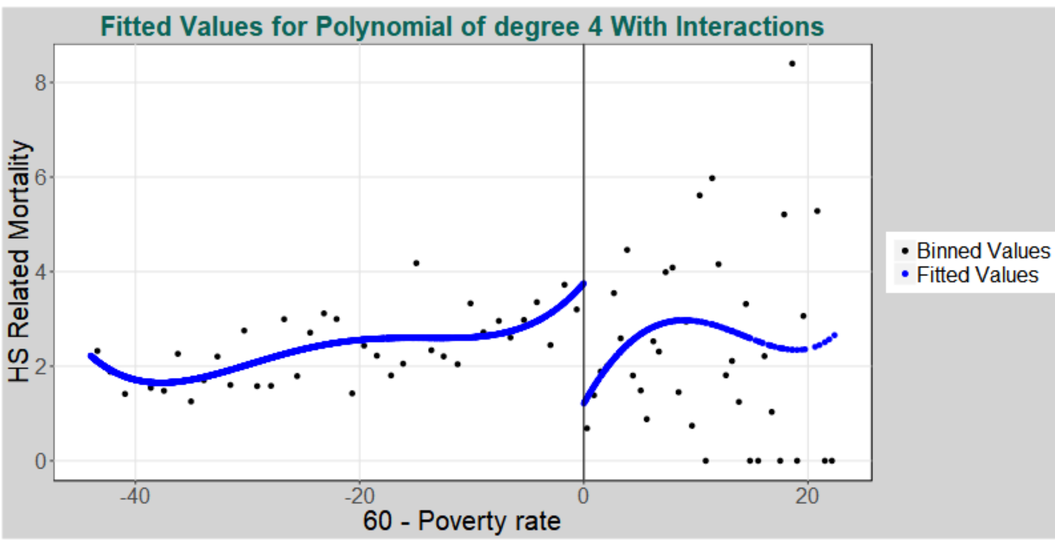
2.2.2 Q2.2.2

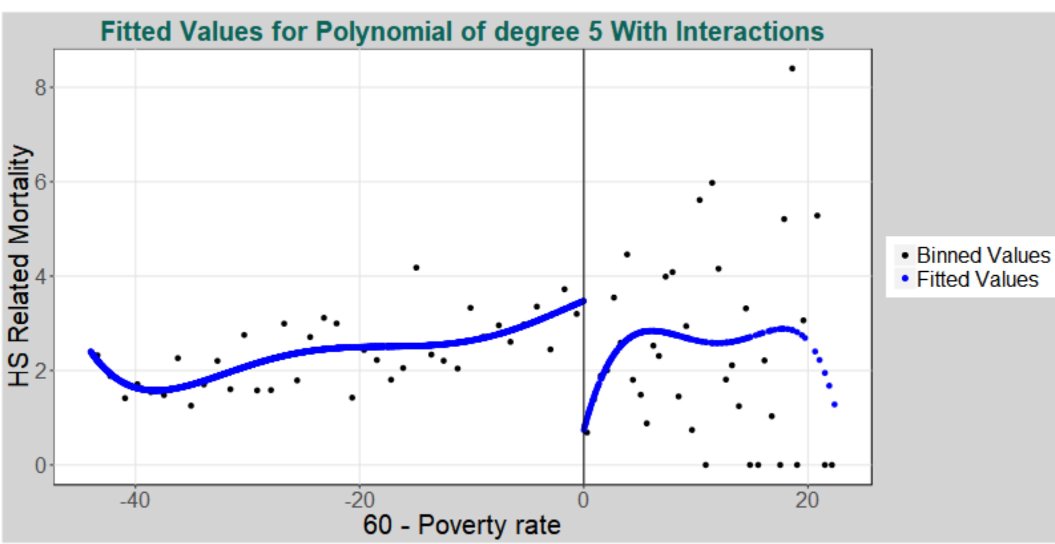
global polynomial fit, Fully Interacted

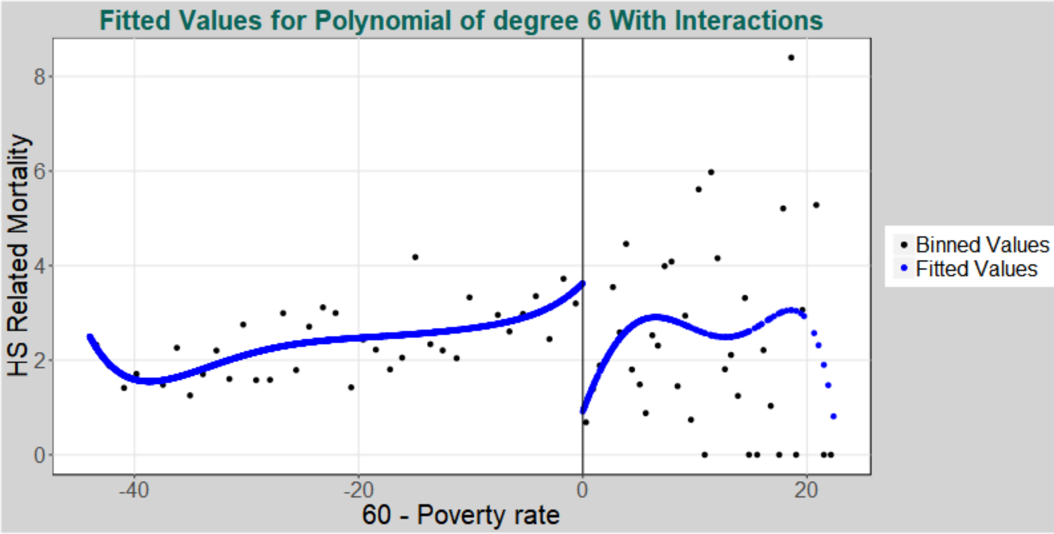
value	Polynomial 3	Polynomial 4	Polynomial 5	Polynomial 6
Estimate	-2.94	-10.70	-45.72	28.77
Standard Error	2.14	9.39	53.29	320.06

.


Fitted Values for Polynomial of degree 3 With Interactions
HS Related Mortality
60 - Poverty rate
Binned Values
Fitted Values


Fitted Values for Polynomial of degree 4 With Interactions
HS Related Mortality
60 - Poverty rate
Binned Values
Fitted Values


Fitted Values for Polynomial of degree 5 With Interactions
HS Related Mortality
60 - Poverty rate
Binned Values
Fitted Values



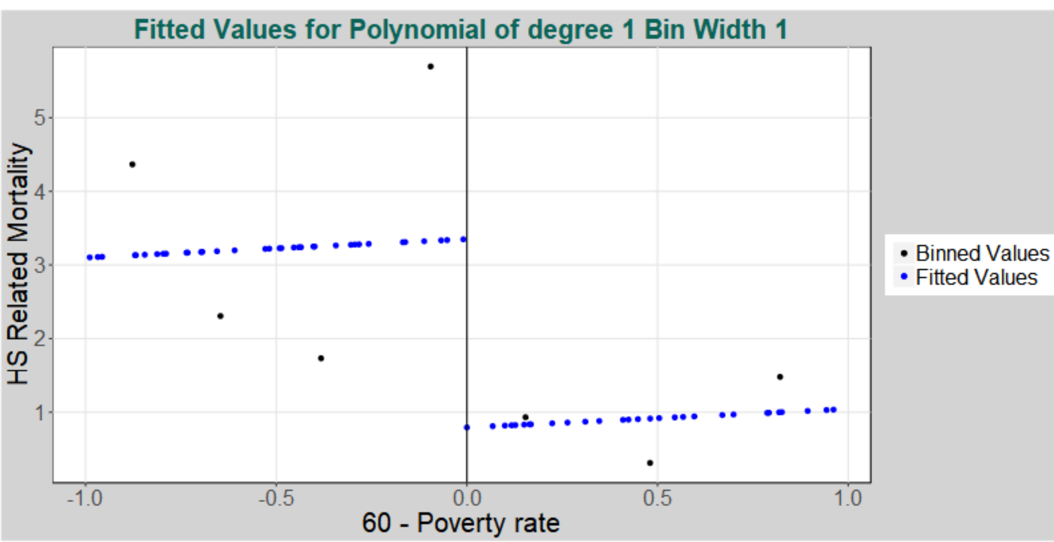
Again these global estimates are generally not reliable. The interaction terms make the predicted values pretty crazy and does not seem plausible.

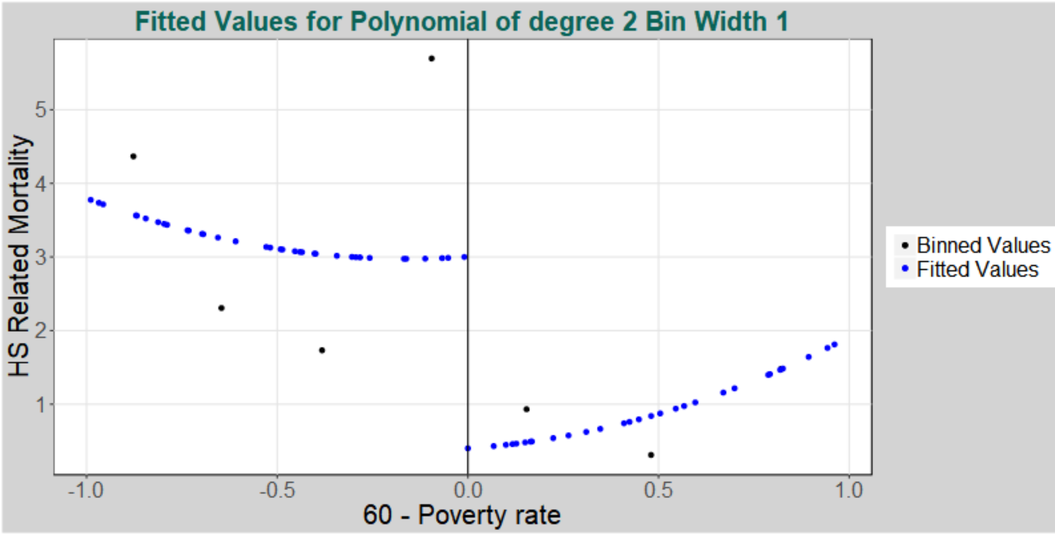
2.2.3 Q2.2.3

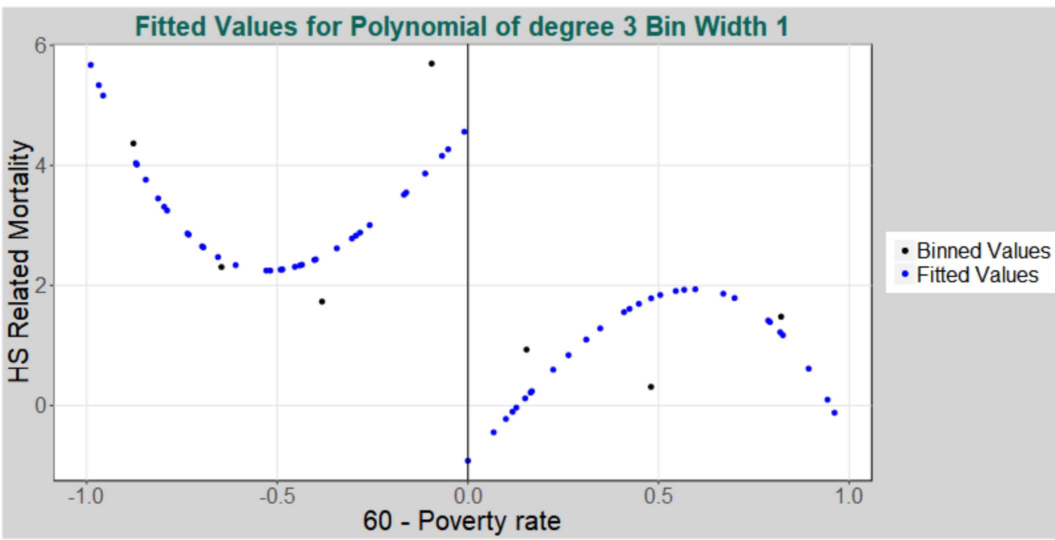
Local Parametric Model Bandwidth of 1

value	Polynomial 1	Polynomial 2	Polynomial 3
Estimate	-2.56	-2.60	-5.55
Standard Error	1.92	1.93	2.55

Graphs for Bandwidth of 1



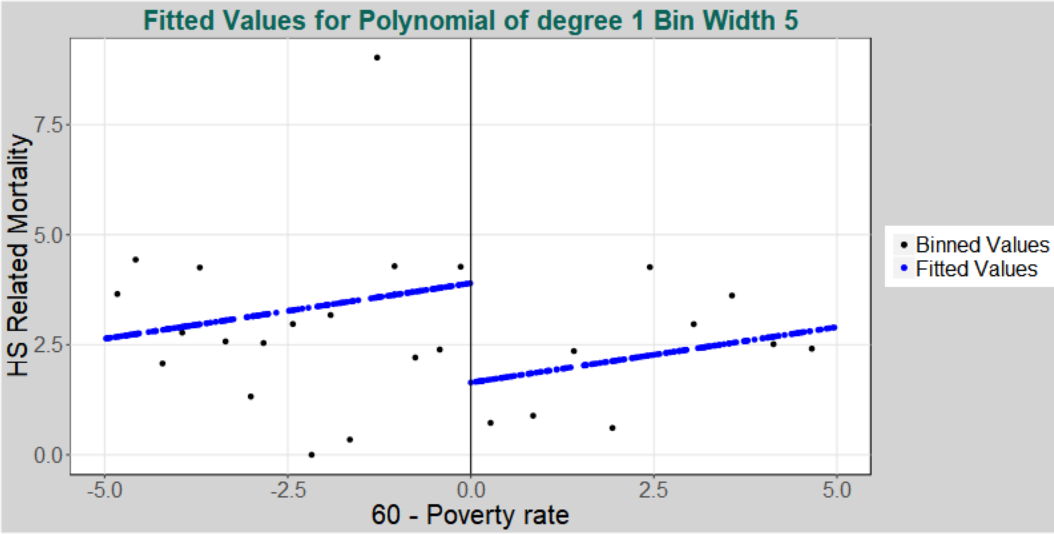


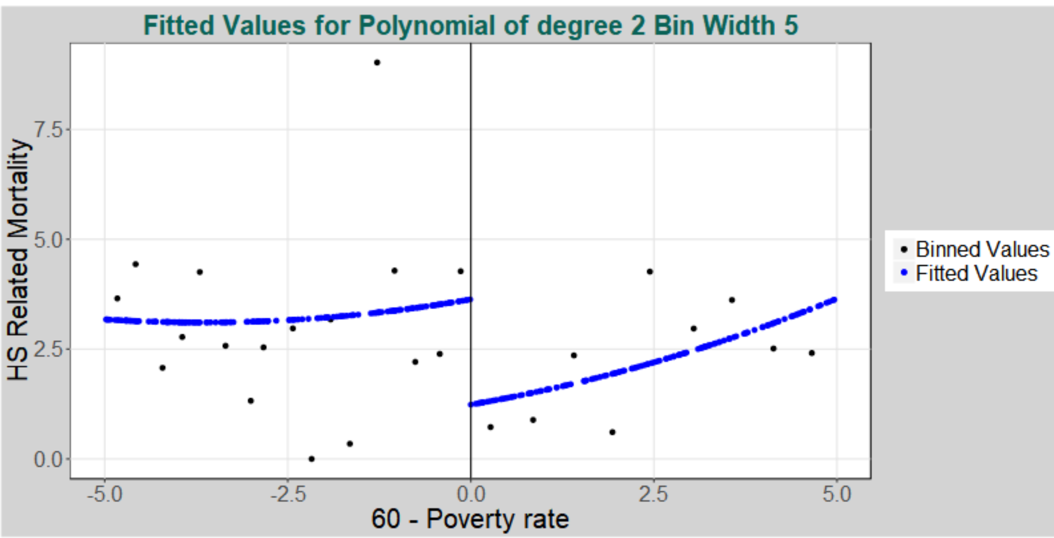


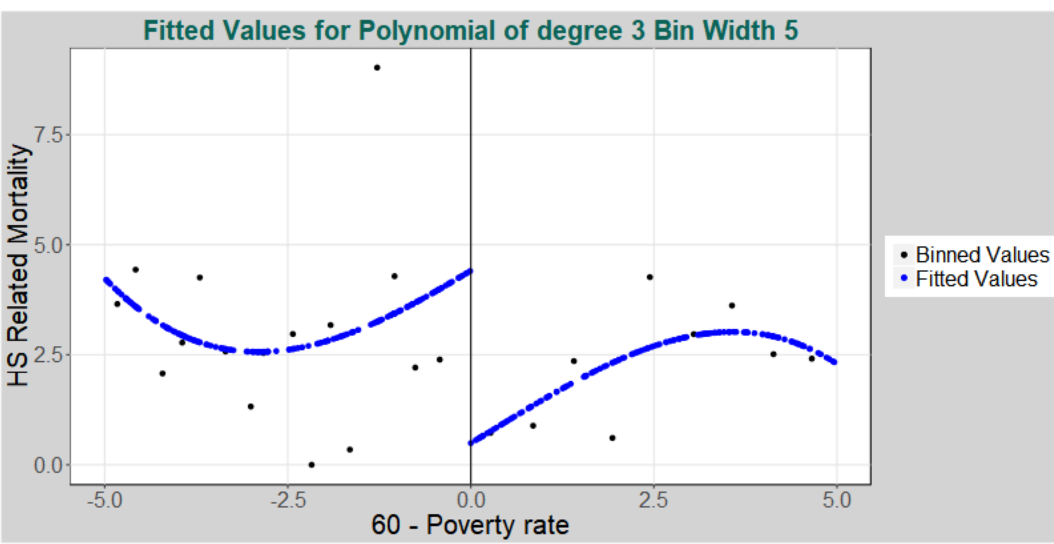
Local Parametric Model Bandwidth of 5

value	Polynomial 1	Polynomial 2	Polynomial 3
Estimate	-2.26	-2.40	-3.92
Standard Error	1.31	1.32	1.72

Graphs for Bandwidth of 5



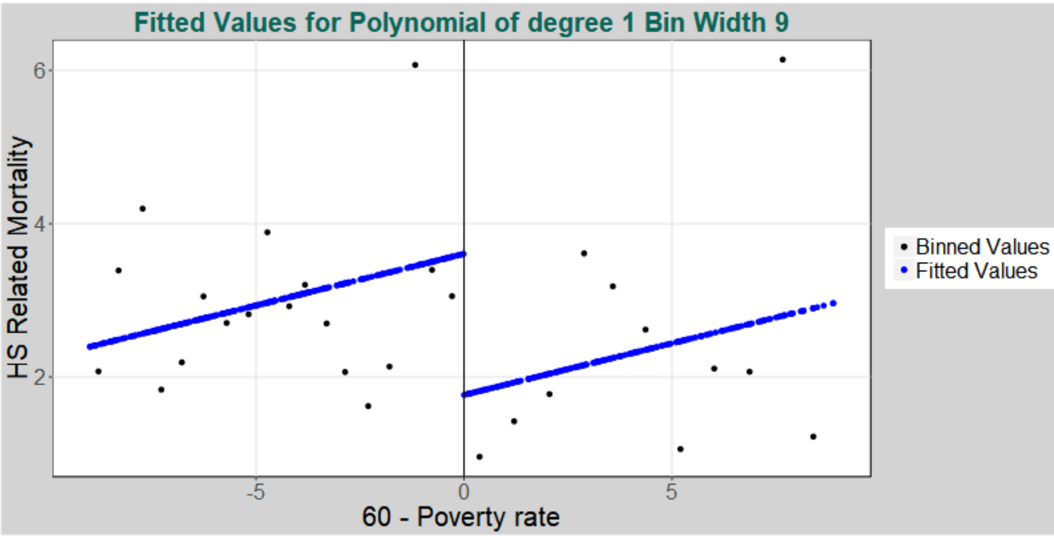


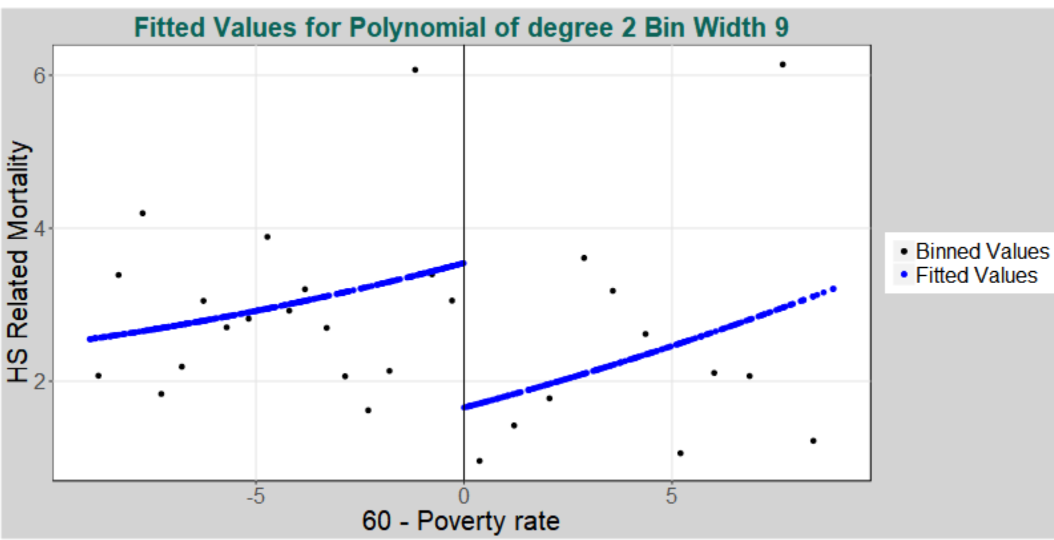


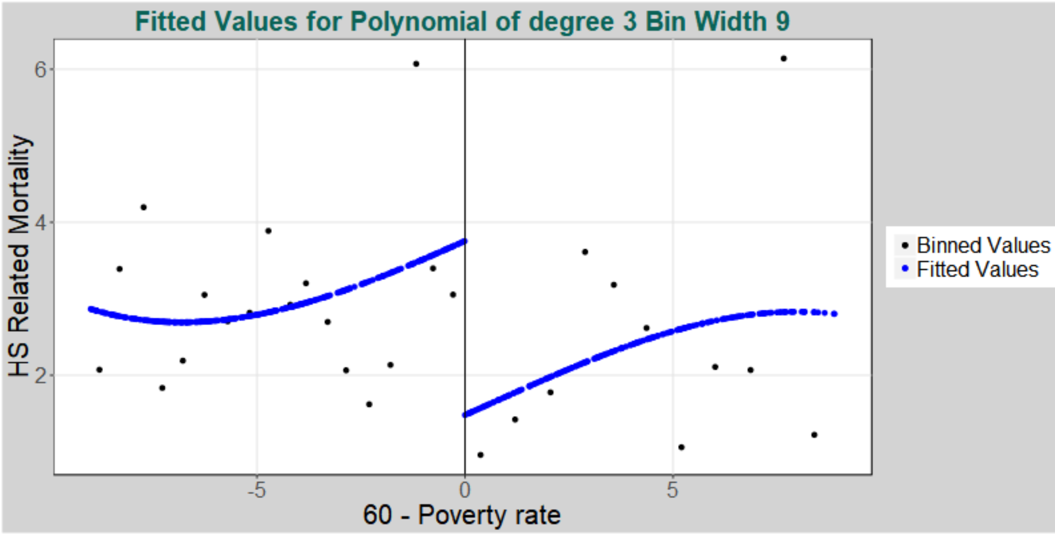
Local Parametric Model Bandwidth of 9

value	Polynomial 1	Polynomial 2	Polynomial 3
Estimate	-1.84	-1.89	-2.28
Standard Error	0.93	0.94	1.22

Graphs for Bandwidth of 9



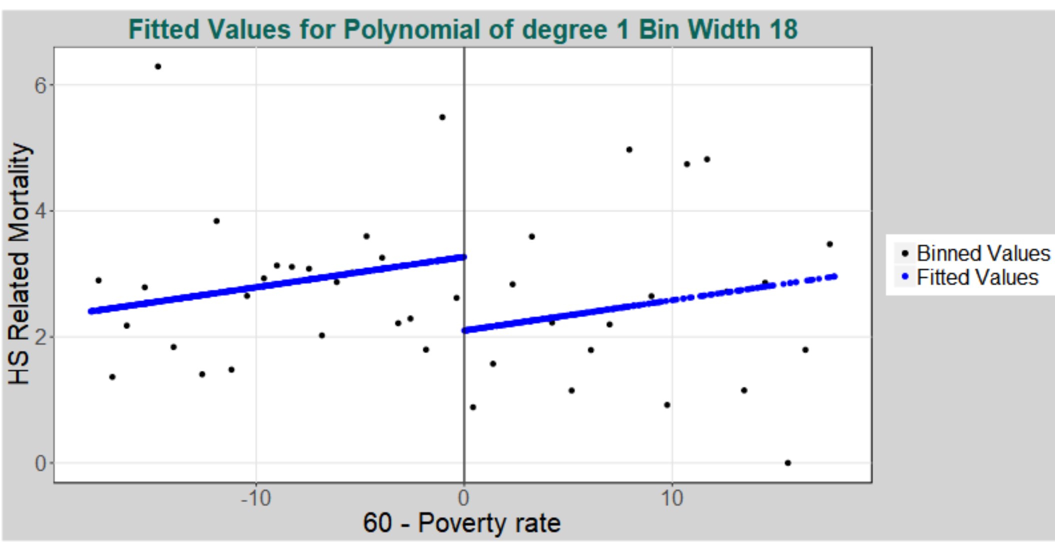


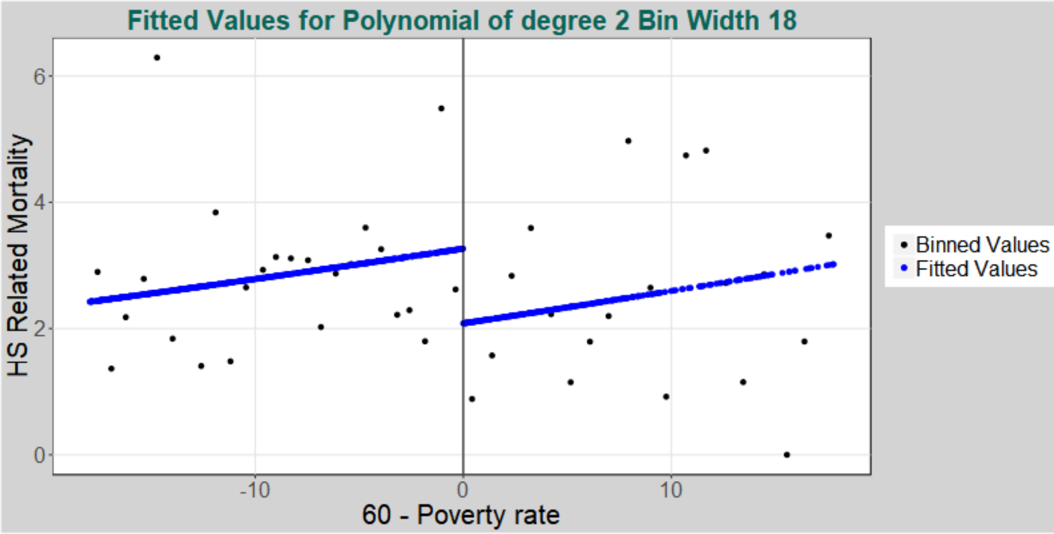


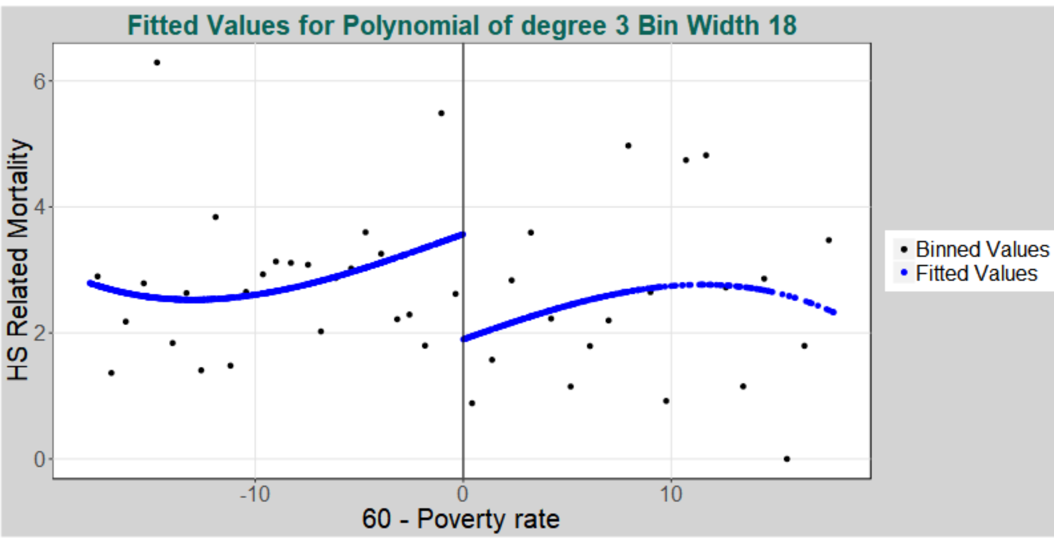
Local Parametric Model Bandwidth of 18

value	Polynomial 1	Polynomial 2	Polynomial 3
Estimate	-1.17	-1.18	-1.67
Standard Error	0.76	0.80	1.02

Graphs for Bandwidth of 18







While one of these binwidths and polynomials could be a reasonable estimate there and ad-hoc estimate will not reliably pick the correct one. Graphing them all first and picking the one that fits the best is essentially p hacking and isn't a good idea either. So, while one of these may be legitimate, we need a way to systematically determine which one that is. A similar problem exists with picking what degree of polynomial to use, not just the bandwidth.

2.2.4 Q2.2.4

2.3 Q2.3 Robust Local Polynomial Methods

2.3.1 Q2.3.1

Robust local polynomial degree 0

Estimator	Coeff	Std. Err.	CI Lower	CI Upper	polynomial
Conventional	-2.11	0.99	-4.05	-0.17	0
Bias-Corrected	-2.56	0.99	-4.50	-0.62	0
Robust	-2.56	1.23	-4.96	-0.15	0

Robust local polynomial degree 1

Estimator	Coeff	Std. Err.	CI Lower	CI Upper	polynomial
Conventional	-2.41	1.21	-4.77	-0.05	1
Bias-Corrected	-2.78	1.21	-5.14	-0.42	1
Robust	-2.78	1.37	-5.46	-0.10	1

Robust local polynomial degree 2

Estimator	Coeff	Std. Err.	CI Lower	CI Upper	polynomial
Conventional	-3.47	1.37	-6.16	-0.79	2
Bias-Corrected	-3.78	1.37	-6.46	-1.10	2
Robust	-3.78	1.45	-6.62	-0.94	2

These results also suggest a negative relationship but the data driven approach to selecting a bandwidth and the nonparametric approach have solved the problem I outlined above. We have picked a bandwidth and a model in a systematic way so we are conducting one test that best fits the data and not p-hacking.

2.3.2 Q2.3.2

(a)

Placebo Test with Mortality Related Pre-Treatment

	Coeff	Std. Err.	CI Lower	CI Upper
Conventional	-2.38	2.25	-6.78	2.03
Bias-Corrected	-1.77	2.25	-6.17	2.64
Robust	-1.77	2.68	-7.01	3.48

Placebo Test with Mortality Injury Post-Treatment

	Coeff	Std. Err.	CI Lower	CI Upper
Conventional	1.13	3.77	-6.26	8.53
Bias-Corrected	1.52	3.77	-5.87	8.92
Robust	1.52	4.39	-7.07	10.12

(b)

Bandwidth and Kernal Robustness Check

	kernal	Bw = 1	Bw = 2	Bw = 3	Bw = 4	Bw = 5	Bw = 6	Bw = 7	Bw = 8	Bw = 9	Bw = 10
1	epanechnikov	-4.97	-2.61	-1.86	-3.03	-3.85	-4.04	-3.69	-3.17	-2.87	-2.78
2	triangular	-4.72	-3.07	-2.04	-2.85	-3.59	-3.86	-3.67	-3.29	-3.04	-2.92
3	uniform	-5.79	-1.82	-2.28	-3.83	-4.21	-3.94	-3.10	-2.33	-2.62	-2.75

(c)

Donut Hole Robustness Check

	# obs dropped	1	2	3	4	5	6	7	8	9	10
1	estimate	-2.73	-2.86	-2.58	-3.02	-2.92	-2.73	-2.56	-2.73	-2.73	-2.73

(d)

Placebo Cutoff Robustness Check

	Statistic	c = -10	c = -8	c = -6	c = -4	c = -2	c = 0	c = 2	c = 4	c = 6	c = 8	c = 10
1	Estimate	0.55	-0.26	0.40	-0.09	2.24	-2.78	3.13	-1.53	1.64	-5.80	4.18
2	p Value	0.49	0.78	0.64	0.93	0.17	0.02	0.02	0.27	0.13	0.06	0.28

2.3.3 Q2.3.3

The placebo tests all return statistically insignificant results. The robustness checks all for different bandwidths and kernaals return negative coefficients which supports our data driven model. Similarily the donut hole approach reports negative estimates. Finally, the placebo cutoff test is not consistently reporting significant negative effects for other cut points. The cutpoint of 8 might be of some concern if the others weren't so supportive.
Overall the results suggest a robust and significant negative relationship at the cut point.

2.4 Q 2.4 Local Randomization Methods

I am choosing 1.8 as my hypothesized bandwidth. It is reported in the table below along with other bandwidths as a robustness check. RD plots for this are done above.

Neynman's approach

	Statistic	w = 0.8	w = 1	w = 1.2	w = 1.4	w = 1.6	w = 1.8	w = 2	w = 2.2	w = 2.4	w = 2.6
1	Estimate	-0.63	-1.69	-1.28	-2.01	-1.65	-1.08	-0.91	-0.81	-0.45	-0.30
2	P-Value	0.59	0.05	0.05	0.04	0.05	0.11	0.09	0.09	0.29	0.42
3	Std Error	1.15	0.86	0.64	0.96	0.84	0.67	0.54	0.47	0.42	0.37

Similarly to the RD methods this is supporting the finding of a negative relationship. We find this for most of the bandwidths I have selected above in addition to the single bandwidth I chose as My hypothesis.

3 Appendix

3.1 R Code

pset 6 675

```r
#======================#
# ==== PS 6 Metrics ====
#======================#

#=========================================#
# ==== Load packages clear workspace ====
#=========================================#

# clear workspace
rm(list = ls(pos = ".GlobalEnv"), pos = ".GlobalEnv")
options(scipen = 999)
cat("\f")

# laod packages
library(data.table)          # helps do everything faster and better
library(ggplot2)             # for pretty plots
library(xtable)              # for latex tables
library(rdrobust)            # for RD plots and other stuff
library(rddensity)           # for RD density continuity tests
library(rdlocrand)           # for RD randomization inference
library(grid)
library(gridGraphics)
library(ggplotify)           # use this to fix the wierd graphs in rdrobust
library(broom)
library(sandwich)
# output folder
f_out <- "c:/Users/Nmath_000/Documents/Code/courses/econ 675/ps_6_tex/"

# plot attributes
plot_attributes <- theme( plot.background = element_rect(fill = "lightgrey"),
                          panel.grid.major.x = element_line(color = "gray90"),
                          panel.grid.minor  = element_blank(),
                          panel.background = element_rect(fill = "white", colour = "black") ,
                          panel.grid.major.y = element_line(color = "gray90"),
                          text = element_text(size= 20),
                          plot.title = element_text(vjust=0, hjust = 0.5, colour = "#0B6357",face = "bol


#====================#
# ==== Question 2 ====
#====================#

# load data
hs <- fread("c:/Users/Nmath_000/Documents/MI_school/Second Year/675 Applied Econometrics/hw/hw6/HeadSta
hs

#================#
# ==== Q 2.1 ====
#================#
```

```r
#==================#
# ==== Q2.1.1 ====
#==================#

  # Evenly-spaced bins, IMSE optimal
  rdplot(hs[,mort_related_pre],
         hs[,povrate60],
         c =0,
         p=1,
         binselect = "es",
        x.label="povrate60",
        y.label="mort_related_pre",
        title="Evenly-spaced bins, IMSE optimal")

  # I guess do this because this since I can't make ggplots with this function
  dev.copy(pdf, paste0(f_out, 'plot_211ia.pdf'))
  dev.off()

  # Evenly-spaced bins, mimicking variance
  rdplot(hs[,mort_related_pre],
         hs[,povrate60],
         p=1,
         binselect = "esmv",
         x.label="povrate60",
         y.label="mort_related_pre",
         title="Evenly-spaced bins, mimicking variance ")

  dev.copy(pdf, paste0(f_out, 'plot_211ib.pdf'))
  dev.off()

  # Quantile-spaced bins, IMSE optimal
  rdplot(hs[,mort_related_pre],
         hs[,povrate60],
         p=1,
         binselect = "qs",
         x.label="povrate60",
         y.label="mort_related_pre",
         title="Quantile-spaced bins, IMSE optimal")

  dev.copy(pdf, paste0(f_out, 'plot_211iia.pdf'))
  dev.off()

  # Quantile-spaced bins, mimicking variance
  rdplot(hs[,mort_related_pre],
         hs[,povrate60],
         p=1,
         binselect = "qsmv",
         x.label="povrate60",
         y.label="mort_related_pre",
         title="Quantile-spaced bins, mimicking variance")

  dev.copy(pdf, paste0(f_out, 'plot_211iib.pdf'))
  dev.off()
```

```r
#==================#
# ==== Q 2.1.2 ====
#==================#

##### i Histogram ###
   # add above below zero flag
  hs[povrate60 >= 0, f_cut := "Above Cutoff"]
  hs[povrate60 < 0, f_cut := "Below Cutoff"]

  # make a histogram before and after cutoff
  plot_2.1.2.i <- ggplot(hs, aes(povrate60)) +
                  geom_histogram(aes(fill = f_cut), breaks = seq(-50,25,2)) +
                  xlab("60 - Poverty rate") +
                  ylab("Count") + ggtitle("Histogram of Running Variable") +
                  scale_fill_discrete(name = "RD Group")
  # check it out
  plot_2.1.2.i

  # then add atrribute that make it look good once save
  plot_2.1.2.i <- plot_2.1.2.i + plot_attributes

## ii local binomial test ###
# make a grid of bandwidiths to test
bi_test <- data.table(bandwidth = seq(.4,4,.2))

# get number above and below cutoff for each bandwidth
bi_test[,below_c := nrow(hs[abs(povrate60) <= bandwidth/2 & f_cut == "Below Cutoff"]), bandwidth]
bi_test[,above_c := nrow(hs[abs(povrate60) <= bandwidth/2 & f_cut == "Above Cutoff"]), bandwidth]
bi_test[,total := below_c + above_c]

# do binomial tests
bi_test[, bin_test := binom.test(below_c, total, .5)$p.value, bandwidth]

# make this table for hw
table_2.1.2.ii <- bi_test[, -c("total")]
colnames(table_2.1.2.ii) <- c("Bandwidth",
                              "Number Below Cutoff",
                              "Number Above Cutoff",
                              "binomial P Valu")

## iii continuity in design tests
## Continuity in density tests (defaults are triangular kernel, jackknife SEs)
cdt <- rddensity(hs$povrate60)

# save plot
png(paste0(f_out, "plot_212i.png"),
    height = 400,
    width = 800,
    type = "cairo")
print(plot_2.1.2.i)
dev.off()

# save table
```

```r
  print(xtable(table_2.1.2.ii, type = "latex"),
        file = paste0(f_out, "table_212ii.tex"),
        include.rownames = FALSE,
        floating = FALSE)

#=================#
# ==== Q 2.2 ====
#=================#

  #==================#
  # ==== Q 2.2.1 ====
  #==================#

  # create polynomials
  p_dt <- as.data.table(poly(hs$povrate60, 6))
  colnames(p_dt) <- paste0("poly_pov60_", colnames(p_dt))
  p_dt[,poly_pov60_1 := NULL ]
  hs <- cbind(hs, p_dt)
  hs[, treatment := as.numeric(f_cut == "Above Cutoff")]

  # initialize data
  table_2.2.1 <- data.table(value = c("Estimate", "Standard Error"))

  # write loop  to make everything we need for polynomial of order N
  poly_n <- 3
  for(poly_n in 3:6){

    # get x variables we need, there is a better way to do this out this works fine
    x_vars <- c("povrate60",
                "treatment",
                grep(paste(as.character(c(1:poly_n)), collapse = "|"),
                     colnames(hs),
                     value = TRUE))

    # make formula
    reg_form <- as.formula(paste0("mort_related_post ~", paste(x_vars, collapse = " + ")))

    # run regressin
    reg_o1 <- lm(reg_form, data = hs)
    reg_o <- data.table(tidy(reg_o1))
    tab_col <- reg_o[term == "treatment", c(estimate, std.error)]

    # put stuff in table
    table_2.2.1[, temp := tab_col]
    setnames(table_2.2.1, "temp", paste0("Polynomial ", poly_n))

    # get fitted values and data
    temp.rd = rdplot(hs[,mort_related_post], hs[,povrate60] ,hide=TRUE)

    temp.rd_dt <- data.table(rdplot_mean_x = temp.rd$vars_bins$rdplot_mean_x,
                             rdplot_mean_y =  temp.rd$vars_bins$rdplot_mean_y)

    fitted_dt <- data.table(x = hs$povrate60, y = reg_o1$fitted.values)
```

```r
  # make plot
  t_plot <- ggplot() + geom_point(data = temp.rd_dt,
                                   aes(x = rdplot_mean_x, y = rdplot_mean_y, color = "Binned Values"))

  t_plot <- t_plot + geom_point(data = fitted_dt,
                                 aes(x = x, y = y, color = "Fitted Values")) +
            geom_vline(xintercept = 0)

  t_plot <- t_plot + xlab("60 - Poverty rate") +
            ylab("HS Related Mortality") +
            scale_color_manual(values = c("black", "blue")) +
            theme(legend.title=element_blank())

  t_plot <- t_plot + ggtitle(paste0("Fitted Values for Polynomial of degree ", poly_n)) +
            plot_attributes
  t_plot

  # save plot
  png(paste0(f_out,
             "plot_221_poly_",
             poly_n,
             ".png"),
      height = 400,
      width = 800,
      type = "cairo")

  print(t_plot)
  dev.off()

}

#==================#
# ==== Q 2.2.2 ====
#==================#

  # Make interaction terms
  int_dt <- as.data.table(poly(hs$povrate60, 6))
  colnames(int_dt) <- paste0("treat_poly_pov60_", colnames(int_dt))
  int_dt[,treat_poly_pov60_1 := NULL ]
  hs <- cbind(hs, int_dt)
  cols <- grep("treat_poly", colnames(hs), value = TRUE)
  hs[, (cols) := lapply(.SD, function(x) x*treatment), .SDcols = cols]

  # initialize data
  table_2.2.2 <- data.table(value = c("Estimate", "Standard Error"))

  # write loop  to make everything we need for polynomial of order N
  poly_n <- 3
  for(poly_n in 3:6){

    # get x variables we need, there is a better way to do this out this works fine
    x_vars <- c("povrate60",
                "treatment",
```

```r
            grep(paste(as.character(c(1:poly_n)), collapse = "|"),
                 colnames(hs),
                 value = TRUE))

# make formula
reg_form <- as.formula(paste0("mort_related_post ~", paste(x_vars, collapse = " + ")))

# run regressin
reg_o1 <- lm(reg_form, data = hs)
reg_o <- data.table(tidy(reg_o1))
tab_col <- reg_o[term == "treatment", c(estimate, std.error)]

# put stuff in table
table_2.2.2[, temp := tab_col]
setnames(table_2.2.2, "temp", paste0("Polynomial ", poly_n))

# get fitted values and data
temp.rd = rdplot(hs[,mort_related_post], hs[,povrate60] ,hide=TRUE)
temp.rd_dt <- data.table(rdplot_mean_x = temp.rd$vars_bins$rdplot_mean_x,
                         rdplot_mean_y =  temp.rd$vars_bins$rdplot_mean_y)
fitted_dt <- data.table(x = hs$povrate60, y = reg_o1$fitted.values)

# make plot
t_plot <- ggplot() + geom_point(data = temp.rd_dt,
                                aes(x = rdplot_mean_x,
                                    y = rdplot_mean_y,
                                    color = "Binned Values"))

t_plot <- t_plot + geom_point(data = fitted_dt,
                              aes(x = x,
                                  y = y,
                                  color = "Fitted Values")) + geom_vline(xintercept = 0)

t_plot <- t_plot + xlab("60 - Poverty rate") +
  ylab("HS Related Mortality") + scale_color_manual(values = c("black", "blue")) +
  theme(legend.title=element_blank())

t_plot <- t_plot +
  ggtitle(paste0("Fitted Values for Polynomial of degree ", poly_n, " With Interactions")) +
  plot_attributes

t_plot

# save plot
png(paste0(f_out,
           "plot_222_poly_",
           poly_n,
           ".png"),
    height = 400,
    width = 800,
    type = "cairo")

print(t_plot)
```

```
    dev.off()

  }

#=================#
# ==== q2.2.3 ====
#=================#

  # drop the interaction terms
  hs <- hs[,
           grep( "treat_poly",
                 colnames(hs),
                 invert = TRUE,
                 value = TRUE),
           with = FALSE]


  in_dt <- hs
  F_223 <- function(p, bw, in_dt = hs){

    # get x variables we need, there is a better way to do this out this works fine
    x_vars <- c("povrate60",
                "treatment",
                grep(paste(as.character(c(1:p)), collapse = "|"),
                     colnames(hs), value = TRUE))

    # subset data down to appropriate binwidth
    w_dt <- in_dt[abs(povrate60) <= bw]

    # make formula
    reg_form <- as.formula(paste0("mort_related_post ~", paste(x_vars, collapse = " + ")))

    # run regressin
    reg_o1 <- lm(reg_form, data = w_dt)
    reg_o <- data.table(tidy(reg_o1))
    tab_col <- reg_o[term == "treatment", c(estimate, std.error)]

    # put in table
    out_tab <-  data.table(value = c("Estimate", "Standard Error"), temp = tab_col)
    setnames(out_tab, "temp", paste0("Polynomial ", p))

    # get fitted values and data
    temp.rd = rdplot(w_dt[,mort_related_post], w_dt[,povrate60] ,hide=TRUE)
    temp.rd_dt <- data.table(rdplot_mean_x = temp.rd$vars_bins$rdplot_mean_x,
                             rdplot_mean_y =  temp.rd$vars_bins$rdplot_mean_y)
    fitted_dt <- data.table(x = w_dt$povrate60, y = reg_o1$fitted.values)

    # make plot
    t_plot <- ggplot() + geom_point(data = temp.rd_dt, aes(x = rdplot_mean_x, y = rdplot_mean_y, colo
    t_plot <- t_plot + geom_point(data = fitted_dt, aes(x = x, y = y, color = "Fitted Values"))  + ge
    t_plot <- t_plot + xlab("60 - Poverty rate") + ylab("HS Related Mortality") + scale_color_manual(
    t_plot <- t_plot + ggtitle(paste0("Fitted Values for Polynomial of degree ", p, " Bin Width ", bw
    t_plot
```

```r
    # save plot
    png(paste0(f_out, "plot_223_poly_", p, "_bw_", bw, ".png"), height = 400, width = 800, type = "ca
    print(t_plot)
    dev.off()

    # return table
    return(out_tab)

    }

    # lapply over different polynomials for each bw
    bw1 <- Reduce( function(x,y) merge(x, y, by = "value"), lapply(1:3, F_223, bw = 1))
    bw5 <- Reduce( function(x,y) merge(x, y, by = "value"), lapply(1:3, F_223, bw = 5))
    bw9 <- Reduce( function(x,y) merge(x, y, by = "value"), lapply(1:3, F_223, bw = 9))
    bw18 <- Reduce( function(x,y) merge(x, y, by = "value"), lapply(1:3, F_223, bw = 18))

    # save tables
    tabs <- grep("bw", ls(), value = TRUE)
    for(tab_i in tabs){

      print(xtable(get(tab_i), type = "latex"),
            file = paste0(f_out, "table_223_", tab_i, ".tex"),
            include.rownames = FALSE,
            floating = FALSE)

    }

#==============#
# ==== 2.3 ====
#==============#

  #================#
  # ==== 2.3.1 ====
  #================#

   # run this thing for 3 polynomials
   rd_regs <- lapply(0:2, function(i) rdrobust(y = hs[,mort_related_post], x = hs[,povrate60], p = i))

   # grab out results we need
   tables_231 <- lapply(1:3,
                        function(i) data.table(Reduce( function(x,y) cbind(x,y) ,
                                                        list(rd_regs[[i]]$coef,
                                                             rd_regs[[i]]$se,
                                                             rd_regs[[i]]$ci)),
                                               keep.rownames = TRUE))

   # rename rn and add in polynomial
   fun_231 <- function(i, in_dt){
     setnames(in_dt, "rn", "Estimator")
     in_dt[, polynomial := i]
     return(in_dt)
   }
   tables_231 <-  mapply(fun_231, 0:2, tables_231, SIMPLIFY = FALSE)
```

```
  # save them
  for(i in 1:3){

    print(xtable(tables_231[[i]], type = "latex"),
          file = paste0(f_out, "table_231_poly_", paste0(i-1), ".tex"),
          include.rownames = FALSE,
          floating = FALSE)

  }

#==================#
# ==== Q 2.3.2 ====
#==================#

  #============#
  # ==== a ====
  #============#

  # run placebo test with other variables
  pl_1 <- rdrobust(hs[, mort_related_pre], hs[,povrate60], p = 1)
  pl_2 <- rdrobust(hs[, mort_injury_post], hs[,povrate60], p = 1)

  table_232ai <- cbind(pl_1$coef, pl_1$se, pl_1$ci)
  table_232aii <- cbind(pl_2$coef, pl_2$se, pl_2$ci)


    print(xtable(table_232ai, type = "latex"),
      file = paste0(f_out, "table_232ai.tex"),
      include.rownames = TRUE,
      floating = FALSE)

    print(xtable(table_232aii, type = "latex"),
          file = paste0(f_out, "table_232aii.tex"),
          include.rownames = TRUE,
          floating = FALSE)


  #============#
  # ==== b ====
  #============#

  h_l <- data.table(h= c(1:10), m =1)

  kern_l <- data.table(kern = c("triangular", "uniform", "epanechnikov"), m =1)

  xwalk <- merge(h_l, kern_l, by = "m", allow.cartesian = TRUE)

  # function to run on all these
  f_232b <- function(h_i, kern_i){

    rdrobust(hs[, mort_related_post], hs[,povrate60], p = 1, h = h_i, kernel = kern_i)$coef[[2]]
  }
```

```r
# DO IT. JUST DO IT!!!! DOOOO IT (labeouf 2015)
xwalk[, estimate := mapply(f_232b, xwalk[, h], xwalk[, kern], SIMPLIFY = FALSE)]
xwalk[, m := NULL]

table_2.3.2b <- dcast.data.table(xwalk, kern ~ h, value.var = "estimate")

setnames(table_2.3.2b, "kern", "kernal")

cols_change <- grep("kern", colnames(table_2.3.2b), value = TRUE, invert = TRUE)
setnames(table_2.3.2b, cols_change, paste0("Bw = ", cols_change))

# Save it
print(xtable(table_2.3.2b, type = "latex"),
      file = paste0(f_out, "table_232b.tex"),
      include.rownames = TRUE,
      floating = FALSE)

#============#
# ==== c ====
#============#

# sort data
hs_sort <- copy(hs)
hs_sort[, abs_pov60 := abs(povrate60)]
hs_sort <- setorder(hs_sort, abs_pov60)

# donut hole function
hole <- 1
do_hole <- function(hole = NULL){

  # remove hole
  w_dt <- hs_sort[-hole]

  # run the thing
  res <- rdrobust(w_dt[, mort_related_post], w_dt[,povrate60], p = 1)$coef[[2]]
  return(res)

  }

# run it on 1-10
table_2.3.2c <- data.table( l = c(1:10),
                            est = unlist(lapply(1:10, do_hole)),
                            "# obs dropped" = "estimate")
table_2.3.2c <- dcast.data.table(table_2.3.2c,
                                 `# obs dropped`~l,
                                 value.var = "est")

# Save it
print(xtable(table_2.3.2c, type = "latex"),
      file = paste0(f_out, "table_232c.tex"),
      include.rownames = TRUE,
      floating = FALSE)
```

```
    #============#
    # ==== d ====
    #============#

    cutoffs = seq(-10,10,2)

    # funciton
    c_fun <-  function(c_i){

    res <-  rdrobust(hs[, mort_related_post], hs[,povrate60], p = 1, c= c_i)
    out_t <- data.table(Statistic = c("Estimate", "p Value"),
                        temp = c(res$coef[[2]], res$pv[[2]]))
    setnames(out_t, "temp", paste0("c = ", c_i))

    }

    rd.cutoffs = lapply(cutoffs, c_fun)

    table_2.3.2d <- Reduce( function(x,y) merge(x, y, by = "Statistic"),rd.cutoffs)

    # Save it
    print(xtable(table_2.3.2d, type = "latex"),
          file = paste0(f_out, "table_232d.tex"),
          include.rownames = TRUE,
          floating = FALSE)


#================#
# ==== Q 2.4 ====
#================#

    # windows
    windows <- seq(.8, 2.6, .2)

    # function to do the stuff I need for a given window
    wind_i <- 1
    f_2.4 <- function(wind_i){

       # subset data to window
        w_dt <- hs[ abs(povrate60) <= wind_i, ]

      # run regression
        reg <- lm( mort_related_post~povrate60, data = w_dt)
        reg_t <- data.table(tidy(reg))

      # put in table
        tab_i <- data.table(Statistic = c("Estimate", "Std Error", "P-Value"),
                            w = as.numeric(reg_t[term == "povrate60",
                                                 c("estimate", "std.error", "p.value")]))
        setnames(tab_i, "w", paste0("w = ", wind_i))

      # return it
        return(tab_i)
```

```
  }

  # run it on all the windows
  table_2.4 <- lapply(windows, f_2.4)
  table_2.4 <- Reduce( function(x,y) merge(x, y, by = "Statistic"),table_2.4)

  #save it
  print(xtable(table_2.4, type = "latex"),
        file = paste0(f_out, "table_24.tex"),
        include.rownames = TRUE,
        floating = FALSE)


#=============================#
# ==== save other tables ====
#=============================#

print(xtable(table_2.2.1, type = "latex"),
      file = paste0(f_out, "table_221.tex"),
      include.rownames = FALSE,
      floating = FALSE)

print(xtable(table_2.2.2, type = "latex"),
      file = paste0(f_out, "table_222.tex"),
      include.rownames = FALSE,
      floating = FALSE)
```

3.2 STATA Code

```
clear all
set more off, perm

cap log close

log using "C:\Users\Nmath_000\Documents\Code\courses\econ 675\ps_6_tex\pset6_stata.smcl", replace

*******************************************************
*** Q2: The Effect of Head Start on Child Mortality ***
*******************************************************
use "C:\Users\Nmath_000\Documents\MI_school\Second Year\675 Applied Econometrics\hw\hw6\HeadStart.dta", clear
cd "C:\Users\Nmath_000\Documents\Code\courses\econ 675\ps_6_tex\"

global y mort_related_post
global z mort_injury_post
global yf mort_related_pre
global x povrate60
gen treat = ($x > 0)
forvalues p = 0/6 {
gen p`p' = $x^`p'
gen tp`p' = $x^`p'*treat
gen up`p' = $x^`p'*(1-treat)
}
order povrate60 mort* treat* p* t* u*

* Q2.1.1 RD Plots

* Evenly spaced bins, IMSE-optimal
rdplot $yf $x, c(0) binselect(es) ///
    graph_options(title("Evenly-spaced binning, IMSE-optimal"))

graph save temp1.gph, replace

* Quantile-spaced bins, IMSE-optimal
rdplot $yf $x, c(0) binselect(qs) ///
    graph_options(title("Quantile-spaced binning, IMSE-optimal"))

graph save temp2.gph, replace


* Evenly spaced bins, IMSE-optimal
rdplot $yf $x, c(0) binselect(esmv) ///
    graph_options(title("Evenly-spaced binning, Minimum-variance"))

graph save temp3.gph, replace

* Quantile-spaced bins, IMSE-optimal
rdplot $yf $x, c(0) binselect(qsmv) ///
    graph_options(title("Quantile-spaced binning, Minimum-variance"))

graph save temp4.gph, replace

* Now combine all graphs

gr combine temp1.gph temp2.gph ///
            temp3.gph temp4.gph, col(2) iscale(.5)

graph export $resdir/q211a_stata.png, replace

* Q2.1.2 Falsification Tests
* Histograms
twoway (hist $x if treat, freq width(2) bcolor("0 100 0 0")) ///
       (hist $x if !treat, freq width(2) bcolor("100 0 0 0") xline(0)), ///
       legend(label(1 "Treated") label(2 "Untreated"))

graph export $resdir/q211b_stata.png, replace

* Local Randomization
```

```
rdwinselect $x

* Continuity in Density
rddensity $x

*/
* Q2.2 Global and Flexible Parametric Methods

* 2.2.1
eststo clear
* Run regressions, save beta and se, graph residuals
forvalues pol = 3/6 {
eststo: reg $y treat p1-p`pol', vce(hc2)
capture drop pred
predict pred
twoway scatter pred $x, title("Order `pol'")
graph save temp`pol'.gph, replace
}

* Export graph
graph combine temp3.gph temp4.gph ///
    temp5.gph temp6.gph, col(2) iscale(.5)

graph export pset6_q221_stata.png, replace

* Export table
esttab using table_q221_stata.tex, b se keep(treat) ///
        noobs nostar nonote mtitles("p:3" "p:4" "p:5" "p:6") nonumbers replace


******************
* 2.2.2
******************
eststo clear

* Run regressions, save beta and se, graph residuals
forvalues pol = 3/6 {
eststo: reg $y treat tp1-tp`pol' up1-up`pol', vce(hc2)
capture drop pred
predict pred
twoway scatter pred $x, title("Order `pol'")
graph save temp`pol'.gph, replace
}

* Export graph
graph combine temp3.gph temp4.gph ///
    temp5.gph temp6.gph, col(2) iscale(.5)

graph export pset6_q222_stata.png, replace

* Export table
esttab using pset6_q222_stata.tex,  se keep(treat) ///
        noobs nostar nonote mtitles("p:3" "p:4" "p:5" "p:6") nonumbers replace

******************
* 2.2.3
******************

* Run regressions, save beta and se, graph residuals
foreach h of numlist 1 5 9 18 {
eststo clear
forvalues pol = 0/2 {
eststo: reg $y treat p0-p`pol' if abs($x) < `h'
capture drop pred
predict pred
twoway scatter pred $x, title("Order `pol', h = `h'")
graph save temp`h'`pol'.gph, replace
}
* Export table
esttab using pset6_q223h`h'_stata.tex, b se keep(treat) ///
```

```
        noobs nostar nonote mtitles("p:0" "p:1" "p:2") nonumbers replace
}

* Export graph
graph combine temp10.gph temp11.gph temp12.gph ///
                temp50.gph temp51.gph temp52.gph ///
                temp90.gph temp91.gph temp92.gph ///
                temp180.gph temp181.gph temp182.gph, ///
               col(3) iscale(.5)

graph export pset6_q223_stata.png, replace


* Q2.3.1
eststo clear
eststo: rdrobust $y $x, p(0) q(1) all
eststo: rdrobust $y $x, p(1) q(2) all
eststo: rdrobust $y $x, p(2) q(3) all

esttab using pset6_q231_stata.tex, b ci ///
        noobs nostar nonote nonumbers replace mtitles("p:0" "p:1" "p:2")

* Q2.3.2a
rdrobust $yf $x, p(0) q(1) all
rdrobust mort_injury_post $x, p(0) q(1) all
di "Looks like there is no effect on the placebo outcomes"


* Q2.3.2b
foreach k in tri uni epa {
    eststo clear

    foreach h of numlist 1/10 {
        eststo: rdrobust $y $x, p(1) q(2) h(`h') kernel(`k') all
    }

        esttab using pset6_q232b`k'_stata.tex, b ///
            noobs nostar nonote nomtitles replace
}
* Q2.3.2c
sort order

eststo clear
forvalues l = 1/10 {
    eststo: rdrobust $y $x if _n > `l', p(1) q(2) all
}

    esttab using pset6_q232c_stata.tex, b ///
        noobs nostar nonote nomtitles replace
* Q2.3.2d
eststo clear
forvalues c = -10(2)10 {
    eststo: rdrobust $y $x, p(1) q(2) c(`c') all
}
    esttab using pset6_q232d_stata.tex, ///
            noobs nostar nonote nomtitles replace

******************
* 2.4
******************

* Q2.4.3
eststo clear
forvalues w = .8(.2)2.6 {
    rdrandinf $y $x, wl(-`w') wr(`w') seed(123)
//  estadd scalar beta_bc = e(tau_bc)
//  estimates store m`l'
}
    esttab using $resdir/pset6_q243_stata.tex, ///
            nonote replace
```

```
log close

translate pset6_stata.smcl pset6_stata.pdf
```