R Code Appendix for Gill and Zorn, 'Debunking the Erroneous Claims of Election Fraud'*

Rebecca Gill, Ph.D.[1] and Christopher Zorn, Ph.D.[2]

[1]University of Nevada Las Vegas [†]
[2]Pennsylvania State University [‡]

October 16, 2024

What follows is the R code used to generate the analyses in the original paper.

```
library(pandoc)
pandoc_activate("3.1.6.2", rmarkdown = TRUE)

library(knitr)
library(cowplot)
library(olsrr)
library(car)
library(texreg)
library(vtable)
library(data.table)
library(ggeffects)
library(GGally)
library(datawizard)
library(broom)
library(tidyverse)
```

```
cc <- fread(
  "https://elections.clarkcountynv.gov/electionresultsTV/sov/20G/PRESIDENT.txt")

# wrangle data

candidate_columns <- c(
  "biden" = "Biden, Joseph R.",
  "trump" = "Trump, Donald J.",
  "jorg"  = "Jorgensen, Jo",
  "blank" ="Blankenship, Don",
  "none"  = "None of These Candidates"
)
```

*The authors write on behalf of themselves. Nothing in this report should be read as speaking for any institution with which Professor Gill or Professor Zorn are associated. This manuscript is currently under development. Please do not cite this working paper without express permission from the authors.

†rebecca.gill@unlv.edu

‡zorn@psu.edu

```r
name_mapping <- c(candidate_columns,
  "type"     = "Tally Type",
  "precinct" = "Precinct",
  "reg"      = "Registration",
  "turnout"  = "Turnout",
  "total"    = "Total Votes"
)

cc <- rename(cc, all_of(name_mapping))

candidates <- c("biden", "trump", "jorg", "blank", "none")

cc <- cc %>%
  mutate(across(all_of(candidates), as.numeric))

cc <- cc %>%
  mutate(
    type = factor(type,
                  levels = c("Totals", "Election Day", "Early Vote", "Mail"),
                  labels = c("totals", "dayof", "early", "mail"))
  )

cc2 <- cc %>%
  filter(type %in% c("early", "mail")) %>%
  select(precinct, type, trump, biden) %>%
  pivot_wider(names_from = type, values_from = c(trump, biden), names_sep = "_") %>%
  rename(
    t_e = trump_early,
    t_m = trump_mail,
    b_e = biden_early,
    b_m = biden_mail
  )

cc2 <- na.omit(cc2)
row.names(cc2) <- as.character(cc2$precinct)
sumtable(cc2,
         vars = c('t_e', 't_m', 'b_e', 'b_m'),
         out = "latex")
```

Table 1. Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
t_e	1065	220	159	0	102	314	1127
t_m	1065	128	97	0	66	173	844
b_e	1065	161	107	0	93	223	666
b_m	1065	285	175	0	179	386	1069

```r
cc2 <- cc2 %>%
  mutate(x = t_e / (t_e + b_e)) %>%
  mutate(y = t_m / (t_m + b_m)) %>%
  mutate(g = t_e / (t_e + b_m)) %>%
  mutate(h = t_m / (t_m + b_e)) %>%
  mutate(n = t_e + b_e + t_m + b_m) %>%
  mutate(omega = (t_e + b_e) / n) %>%
  mutate(lambda = (t_e + b_m) / n) %>%
  mutate(alpha = (t_e + t_m) / n)

sumtable(cc2,
         vars = c('x', 'y', 'g', 'h', 'n', 'omega', 'lambda', 'alpha'),
         out = 'latex')
```

Table 2. Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
x	985	0.56	0.15	0.047	0.48	0.66	1
y	985	0.31	0.11	0.034	0.24	0.37	0.93
g	985	0.43	0.14	0.027	0.34	0.51	0.95
h	985	0.45	0.15	0.043	0.35	0.52	1
n	1065	795	485	0	501	1069	3369
omega	985	0.48	0.072	0.21	0.44	0.52	0.73
lambda	985	0.64	0.039	0.41	0.61	0.66	0.82
alpha	985	0.43	0.13	0.052	0.35	0.51	0.96

```r
std.fit <- lm(alpha ~ y + x, data = cc2)
bst.fit <- lm(alpha ~ g + h, data = cc2)

texreg(list(std.fit, bst.fit),
       custom.model.names = c("Standard", "Bastard"),
       caption.above = TRUE)
```

```r
std.prd <- ggpredict(std.fit, terms = c("y", "x"))
std.plt <- plot(std.prd)

bst.prd <- ggpredict(bst.fit, terms = c("g", "h"))
bst.plt <- plot(bst.prd)

plot_grid(std.plt, bst.plt)
```

```r
wc <- read_csv("rawdata/Washoe.csv")

# wrangle
```

Table 3. Statistical models

	Standard	Bastard
(Intercept)	-0.02^{***}	-0.00^{**}
	(0.00)	(0.00)
y	0.49^{***}	
	(0.01)	
x	0.53^{***}	
	(0.01)	
g		0.64^{***}
		(0.00)
h		0.37^{***}
		(0.00)
R^2	0.98	1.00
Adj. R^2	0.98	1.00
Num. obs.	985	985

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

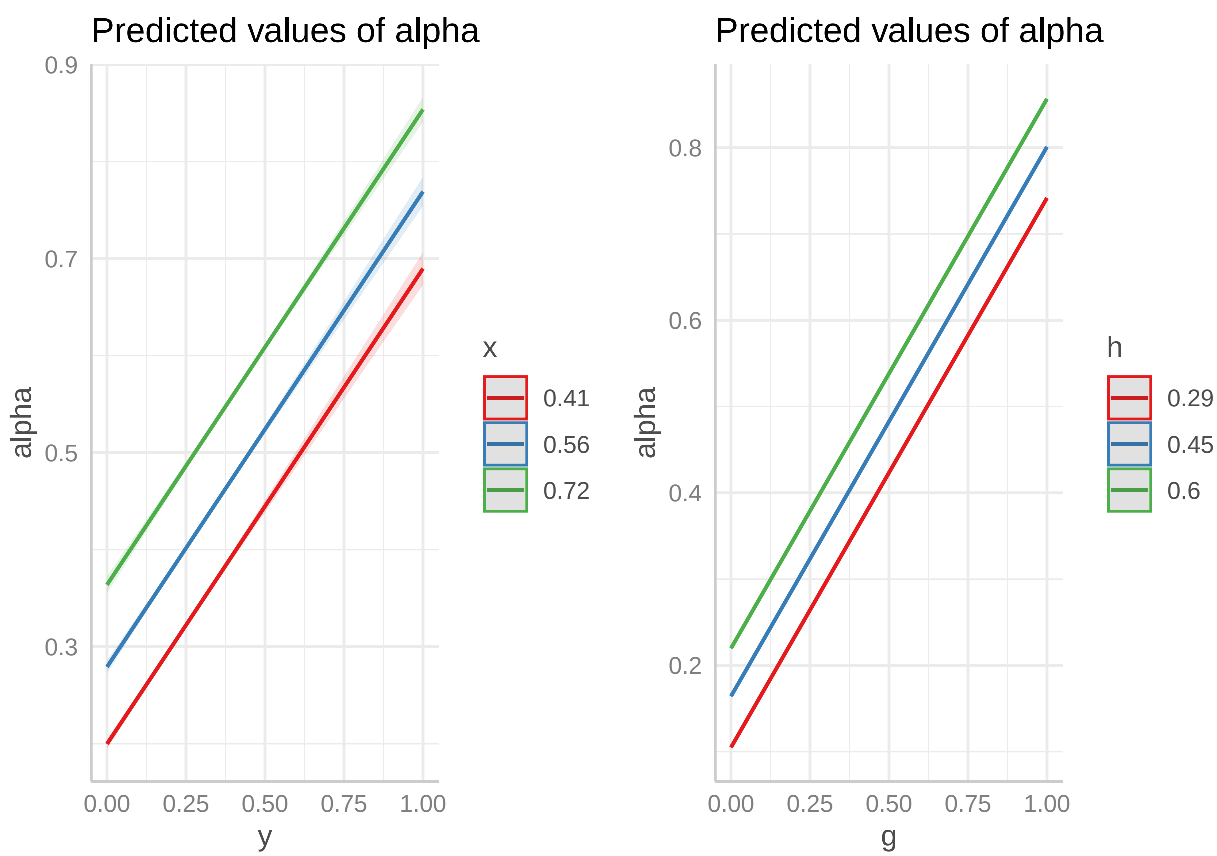


Figure 1. Predicted Values of Trump Vote Share

```r
reno5 <- wc %>% # select variables
  select(PrecinctPortion, CountingGroup,
         starts_with("RENO.CITY.COUNCIL..WARD.5"))

candidate_columns <- c( # candidate vars to rename
  "sbp" = "RENO.CITY.COUNCIL..WARD.5..Vote.For.1._BROWNING.PEUCHAUD..SHEILA_NP",
  "bmc" = "RENO.CITY.COUNCIL..WARD.5..Vote.For.1._CASSIDY..BRIAN.M._NP",
  "dtr" = "RENO.CITY.COUNCIL..WARD.5..Vote.For.1._REESE..DEVON.T._NP",
  "tcw" = "RENO.CITY.COUNCIL..WARD.5..Vote.For.1._WEBSTER..TARA.C._NP"
)

name_mapping <- c(candidate_columns, # all vars to rename
  "precinct" = "PrecinctPortion",
  "type"     = "CountingGroup"
)

reno5 <- rename(reno5, all_of(name_mapping)) # rename them

candidates <- c("sbp", "bmc", "dtr", "tcw") # new list of candidates

reno5 <- reno5 %>% # fix the precinct var and drop obs from other wards
  mutate(across(all_of(candidates), as.numeric)) %>%
  drop_na(all_of(candidates)) %>%
  mutate(precinct = str_extract(precinct, "\\((\\d+)\\)")) %>%
  mutate(precinct = factor(str_replace_all(precinct, "[()]", "")))

reno5 <- reno5 %>% # fix the type variable
  mutate(
    type = factor(type,
                  levels = c("Early Voting", "Election Day", "Mail"),
                  labels = c("early", "dayof", "mail"))
  )

reno5.sum <- reno5 %>% # make summary
  group_by(precinct, type) %>%
  summarize(across(c(sbp, bmc, dtr, tcw),
                   sum,
                   na.rm = TRUE),
            .groups = 'drop')

reno5.wide <- reno5.sum %>%
  pivot_wider(names_from = type,
              values_from = c(sbp, bmc, dtr, tcw),
              names_sep = "_")

labz <- c('SBP Early Vote', 'SBP Election Day', 'SBP Mail',
          'BMC Early Vote', 'BMC Election Day', 'BMC Mail',
          'DTR Early Vote', 'DTR Election Day', 'DTR Mail',
          'TCW Early Vote', 'TCW Election Day', 'TCW Mail')

reno5.sm <- na.omit(reno5.wide) # drop the NAs
reno5.sm <- reno5.sm[-2,] # drop the NAN
```

```r
sumtable(reno5.sm,
         vars = c('sbp_early', 'sbp_dayof', 'sbp_mail',
                  'bmc_early', 'bmc_dayof', 'bmc_mail',
                  'dtr_early', 'dtr_dayof', 'dtr_mail',
                  'tcw_early', 'tcw_dayof', 'tcw_mail'),
         labels = labz,
         out = "latex")
```

Table 4. Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
SBP Early Vote	16	9.9	6.8	0	5	15	21
SBP Election Day	16	12	6.9	1	8.5	15	24
SBP Mail	16	79	68	1	42	92	267
BMC Early Vote	16	22	15	1	11	35	44
BMC Election Day	16	31	20	0	21	41	79
BMC Mail	16	99	102	1	43	107	390
DTR Early Vote	16	24	17	0	12	36	54
DTR Election Day	16	26	16	1	16	38	48
DTR Mail	16	156	106	1	86	208	402
TCW Early Vote	16	8.9	7	0	2.5	14	21
TCW Election Day	16	13	9	0	7	19	30
TCW Mail	16	74	48	0	38	108	149

```r
reno5.sm <- reno5.sm %>%
  mutate(A = bmc_early + bmc_dayof + sbp_dayof) %>%
  mutate(B = dtr_early + dtr_dayof + bmc_mail) %>%
  mutate(C = tcw_mail) %>%
  mutate(D = dtr_mail + sbp_mail) %>%
  mutate(n = A + B + C + D) %>%
  mutate(g = A / (A + D)) %>%
  mutate(h = C / (C + B)) %>%
  mutate(lambda = (A + D) / n) %>%
  mutate(alpha = (A + C) / n)

sumtable(reno5.sm,
         vars = c('A', 'B', 'C', 'D', 'n', 'g', 'h','lambda', 'alpha'),
         out = 'latex')
```

```r
reno.fit <- lm(alpha ~ g + h, data = reno5.sm)
reno3.fit <- lm(alpha ~ g + poly(h, 3), data = reno5.sm)
texreg(list(reno.fit, reno3.fit),
       custom.model.names = c("Bastard Model", "Bastard Cubic Model"),
       caption.above = TRUE)
```

Table 5. Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
A	16	65	39	2	42	88	141
B	16	149	119	5	83	190	451
C	16	74	48	0	38	108	149
D	16	235	169	3	128	292	669
n	16	523	353	12	311	646	1368
g	16	0.26	0.14	0.13	0.21	0.24	0.75
h	16	0.3	0.18	0	0.17	0.42	0.56
lambda	16	0.57	0.027	0.52	0.56	0.59	0.63
alpha	16	0.28	0.087	0.17	0.19	0.34	0.47

Table 6. Statistical models

	Bastard Model	Bastard Cubic Model
(Intercept)	-0.02^{**}	0.11^{***}
	(0.01)	(0.00)
g	0.66^{***}	0.65^{***}
	(0.01)	(0.02)
h	0.44^{***}	
	(0.01)	
poly(h, 3)1		0.31^{***}
		(0.01)
poly(h, 3)2		0.01
		(0.01)
poly(h, 3)3		0.01
		(0.01)
R^2	1.00	1.00
Adj. R^2	0.99	1.00
Num. obs.	16	16

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

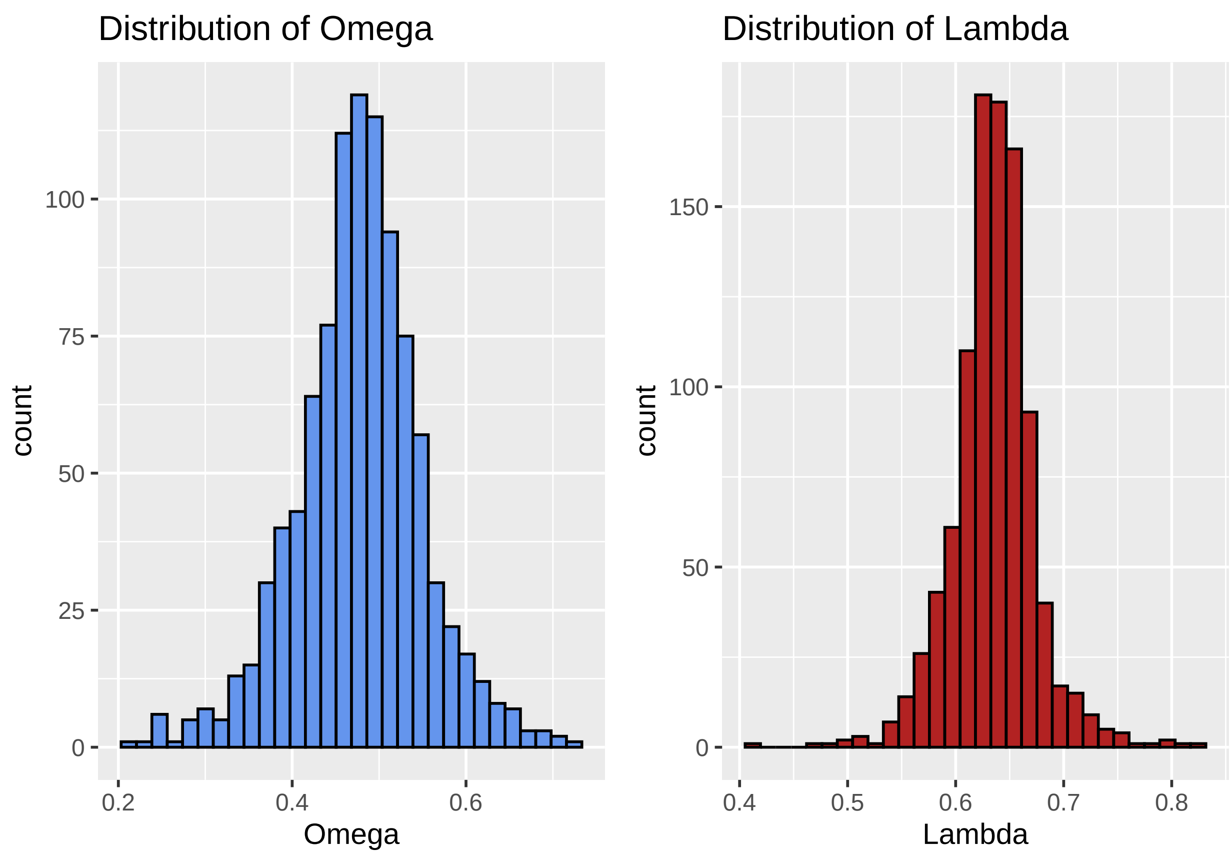
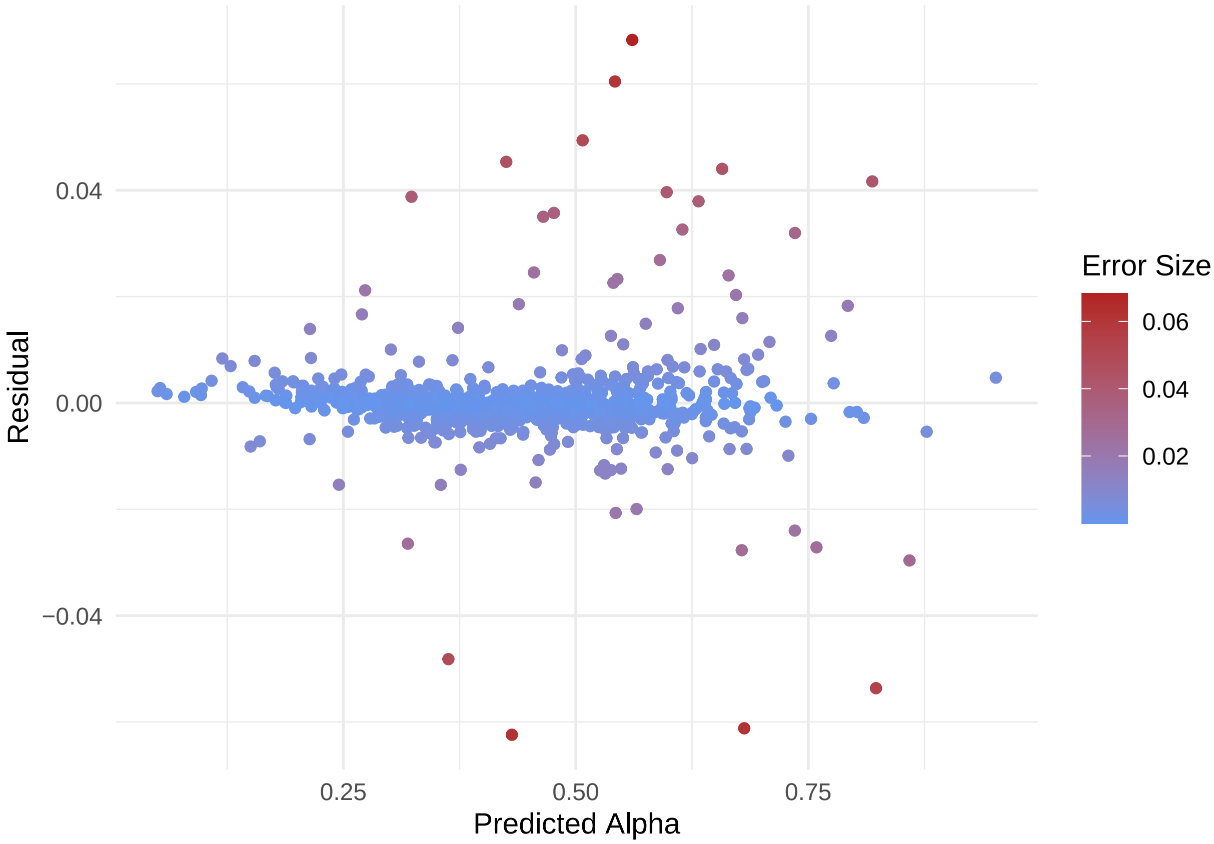


Figure 2. Distribution of Omega and Lambda (Calculated)

```
omega.hist <- ggplot(cc2, aes(x = omega)) +
  geom_histogram(fill = "cornflowerblue",
                 color = "black") +
  labs(title="Distribution of Omega",
       x = "Omega")

lambda.hist <- ggplot(cc2, aes(x = lambda)) +
  geom_histogram(fill = "firebrick",
                 color = "black") +
  labs(title="Distribution of Lambda",
       x = "Lambda")

plot_grid(omega.hist, lambda.hist)
```

```
cc2 <- augment(bst.fit) # put resids into data frame
cc2$dist0 <- abs(cc2$.resid) # calculate error magnitude

ggplot(cc2, aes(x = .fitted, y = .resid, color = dist0)) +
  geom_point() +
  scale_color_gradient(low = "cornflowerblue", high = "firebrick") +
  labs(color = "Error Size") +
  xlab("Predicted Alpha") +
```

```r
  ylab("Residual") +
  theme_minimal() +
  theme(legend.position = "right")
```




```r
influencePlot(bst.fit)
```

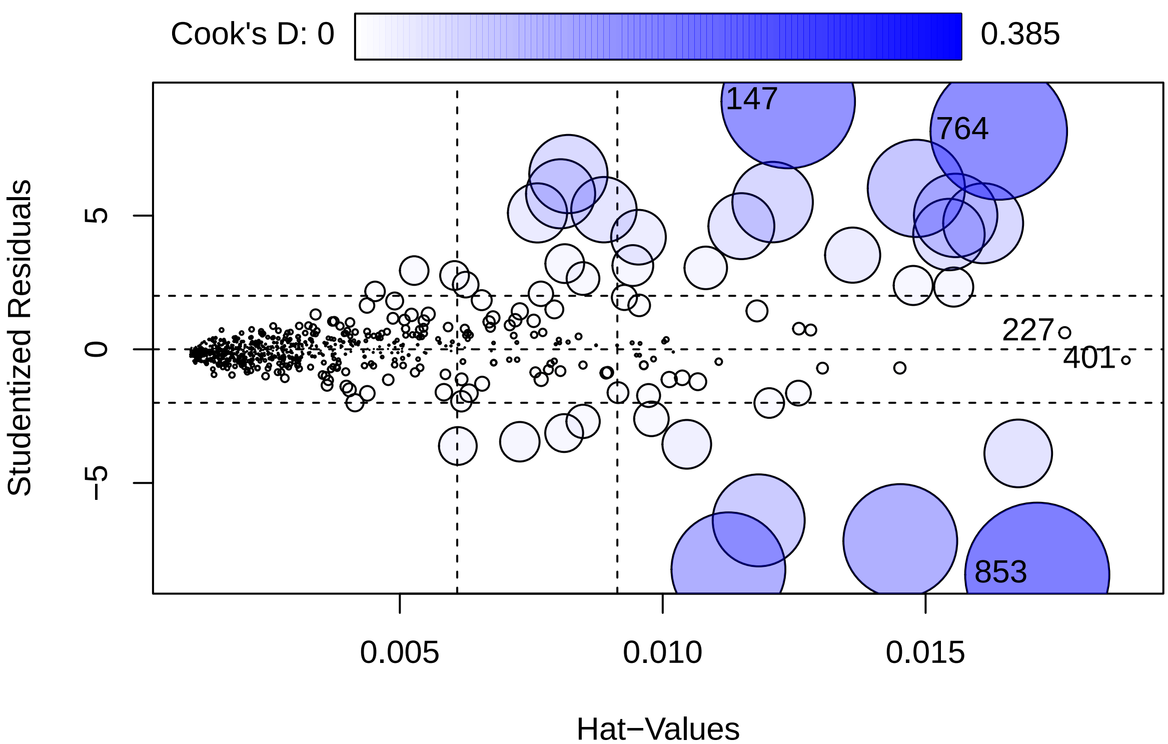


Figure 3. Influence of Outliers on Lambda Estimate