PS3 Solutions

Teaching Staff

February 2023

Q1

First a regression table helper function to clean up the summary tables.

```r
regTable = function(model){
  m = summary(model)
  tidydf = data.frame(terms = names(m[["coefficients"]][,1]),
                      estimates = round(m[["coefficients"]][,1],4),
                      std.error = m[["coefficients"]][,2],
                      statistic = m[["coefficients"]][,3],
                      p.value = m[["coefficients"]][,4],
                      row.names = NULL)

  ## Round the numeric columns to 4 digits
  tidydf[, 2:ncol(tidydf)] = apply(tidydf[, 2:ncol(tidydf)],
                                   2,
                                   function(x) round(x, 4)
                                   )
  return(tidydf)
}
```

```r
set.seed(123)
x = rexp(1500, rate = 2)
```

1

```r
boot_univariate = function(datvec, statint, B, alpha){
  out = vector(mode = "logical", length = B)
  for(i in 1:B){
    out[i] = statint(sample(datvec, replace = T))
  }
  conf.out = quantile(out, probs = c(alpha/2, 1-alpha/2))
  return(conf.out)
}
```

2

```r
boot_univariate(x, median, 10000, 0.05)
```

```
##      2.5%     97.5%
## 0.3313953 0.3856648
```

The produces a 95% bootstrap confidence interval for the median of the variable represented by x.

Bonus

```
iqr = function(x){
  return(quantile(x, probs = .75) - quantile(x, probs = .25))
}
boot_univariate(x, iqr, 10000, 0.05)
```

```
##      2.5%     97.5%
## 0.5170515 0.6014724
```

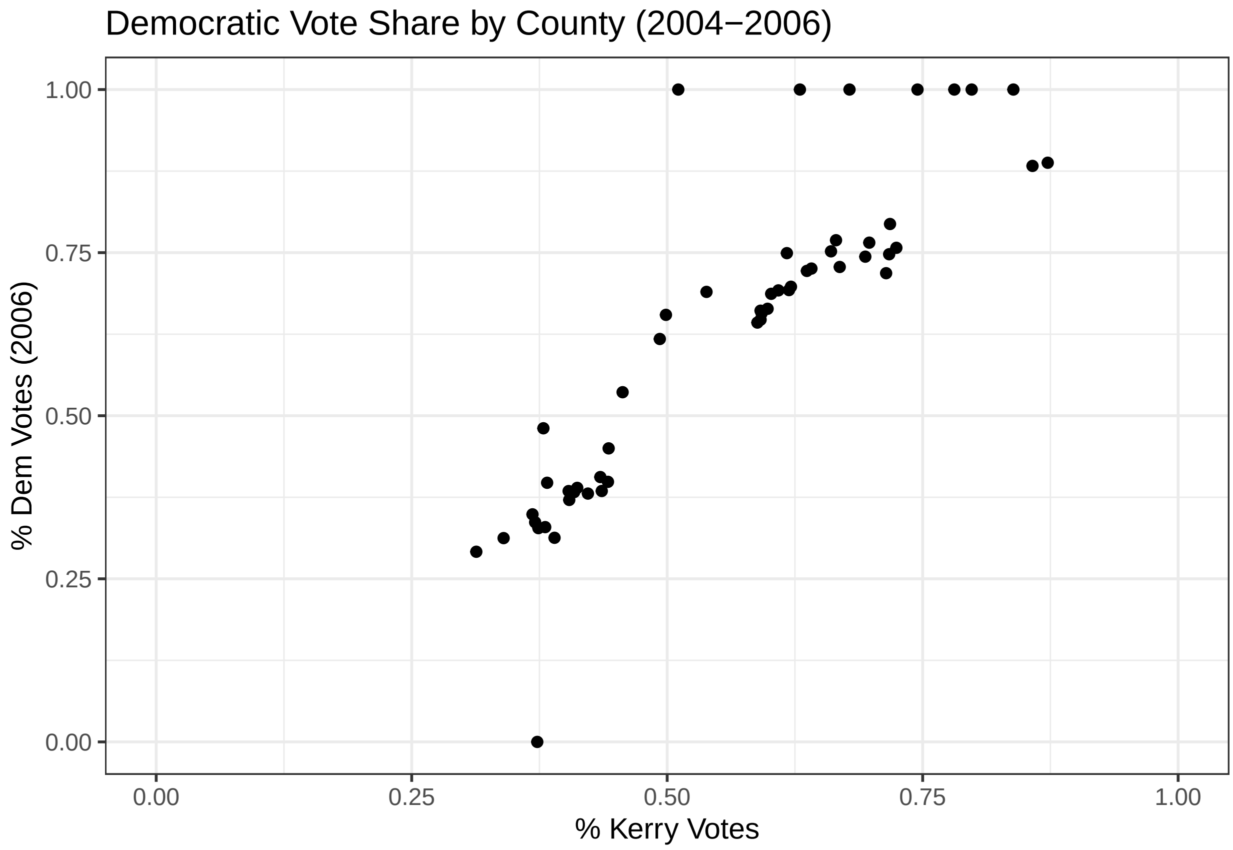
Q2

1

```
ca = read.csv("../data/ca2006.csv")
```

2

```
plot = ca |>
  ggplot(aes(dem_pres_2004, prop_d))+
  geom_point()+
  labs(x = "% Kerry Votes",
       y = "% Dem Votes (2006)",
       title = "Democratic Vote Share by County (2004-2006)")+
  theme_bw()+
  ylim(0,1)+
  xlim(0,1)
plot
```

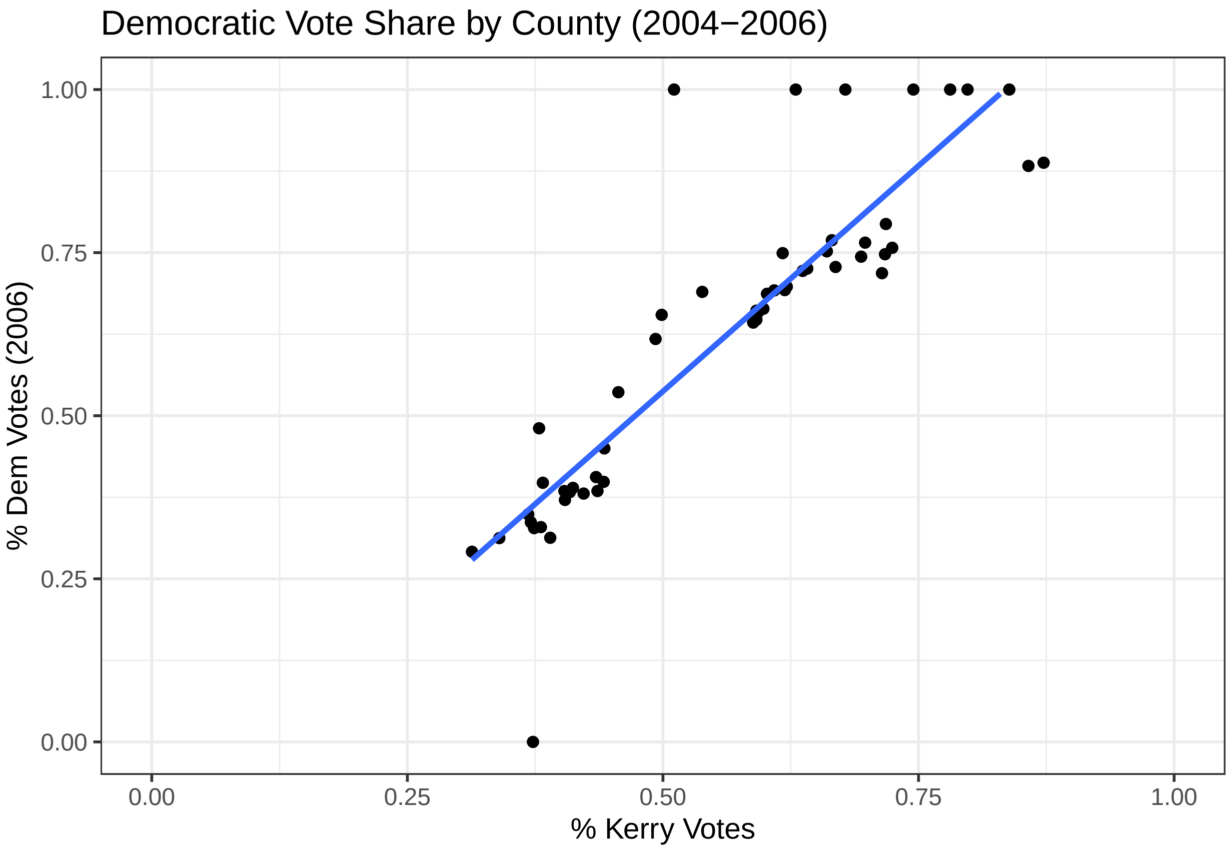


3

```
reg = lm(prop_d ~ dem_pres_2004, data = ca)
regTable(reg)
```

```
##            terms estimates std.error statistic p.value
## 1   (Intercept)   -0.1539    0.0598   -2.5744   0.013
## 2 dem_pres_2004    1.3827    0.1029   13.4363   0.000
```

```
plot + geom_smooth(method = "lm", se = F)
```



4

```
my_predict = function(coefs, newdata, ols = TRUE){
  if(ols == TRUE){
    ## Linear Model prediction
    prediction = coefs%*%newdata
    return(unname(prediction))
  }else{
    ## Logit Model prediction
    betas = unname(coefs) %*% newdata
    odds = 1/ (1 + exp(-betas))
    return(odds)
  }
}

my_predict(reg$coefficients, newdata = c(1, 0.5))
```

```
##           [,1]
## [1,] 0.5374445
```

5 and 6

```
mreg = lm(prop_d ~ dem_pres_2004 + dem_pres_2000 + dem_inc,
          data = ca)

my_predict(mreg$coefficients,
           newdata = c(1,0.5, 0.5, 1),
           ols = TRUE)
```

```
##             [,1]
## [1,] 0.6147444
```

7

```
boot_reg = function(df, N = 53, B = 10000, alpha = 0.05){
  set.seed(pi)
  simple = vector(mode = "logical", length = B)
  multi = vector(mode = "logical", length = B)
  for(i in 1:B){
    dat = df[sample.int(nrow(df), 53, replace = T),]
    simple[i] = my_predict(lm(prop_d ~ dem_pres_2004,
                              data = dat)$coefficients,
                           newdata = c(1,0.5))
    multi[i] = my_predict(lm(prop_d ~ dem_pres_2004 +
                               dem_pres_2000 + dem_inc,
                             data = dat)$coefficients,
                          newdata = c(1,0.5,0.5,1))
  }
  sci = quantile(simple, probs = c(alpha/2, 1-alpha/2))
  mci = quantile(multi, probs = c(alpha/2, 1-alpha/2))
  return(list(simple = simple, multi = multi,
              simple_ci = sci, multi_ci = mci))
}
```

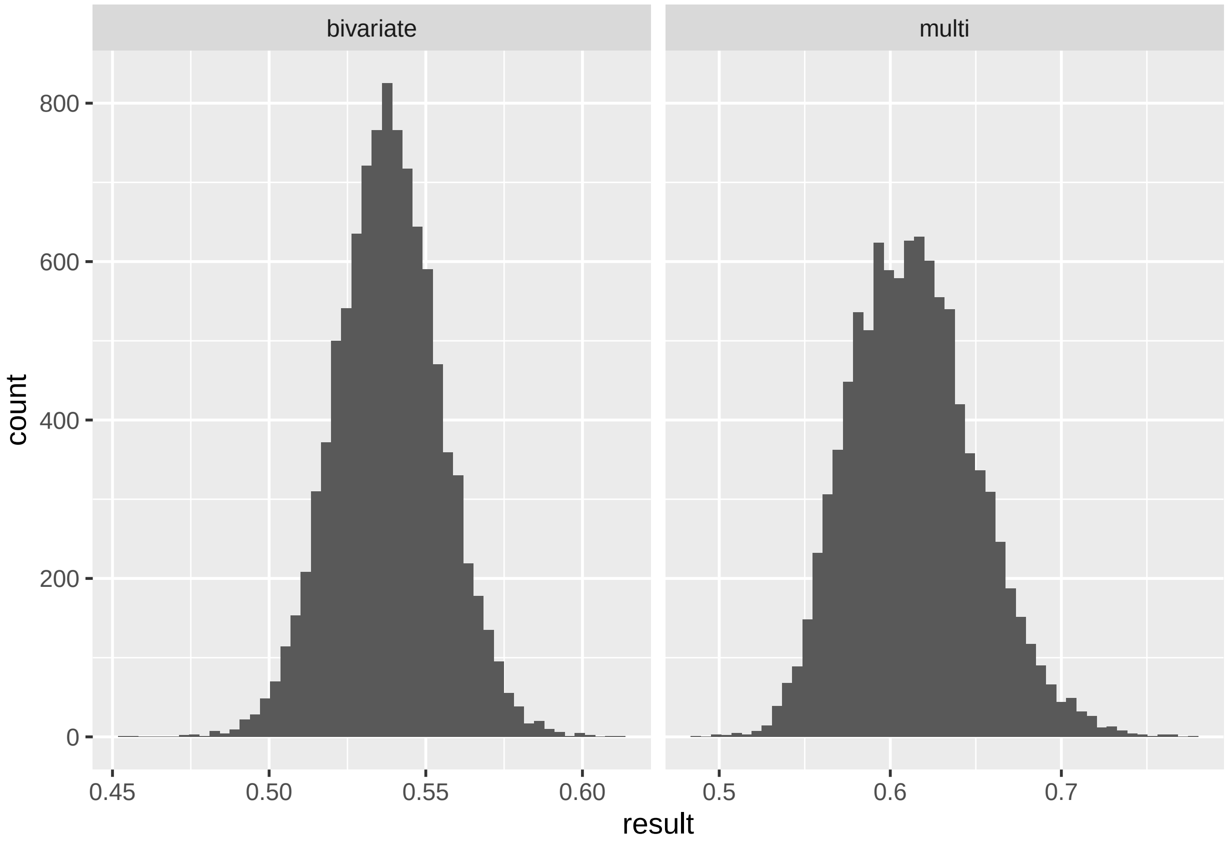
8

```
results = boot_reg(df = ca)
```

2.5%	97.5%
0.5050168	0.5716776
0.5496060	0.6924033

```
out = data.frame(id = c(rep("bivariate", 10000),
                          rep("multi", 10000)),
                   result = c(results$simple,
                               results$multi))
```

```
out |>
  ggplot(aes(result))+
  geom_histogram(bins = 50)+
  facet_wrap(~id, scales = "free_x")
```



9

```
mean(results$simple > .5)
```

```
## [1] 0.988
```

```
mean(results$multi > .5)
```

```
## [1] 0.9996
```

Q3

1 and 2

```
clinton = read.csv("../data/vote92.csv")
mean(clinton$clintonvote)
```

```
## [1] 0.4576458
```

3

```
logit = glm(clintonvote ~ dem + female + clintondist, data = clinton,
            family = "binomial")
regTable(logit)
```

```
##           terms estimates std.error statistic p.value
## 1 (Intercept)    -1.4069    0.1876   -7.5003  0.0000
## 2           dem     3.0565    0.1869   16.3566  0.0000
## 3      female    0.1742    0.1841    0.9459  0.3442
## 4 clintondist   -0.1448    0.0278   -5.2149  0.0000
```

4 and 5

```
## see my_predict() function definition
my_predict(logit$coefficients, newdata = c(1,1,1,1), ols = FALSE)
```

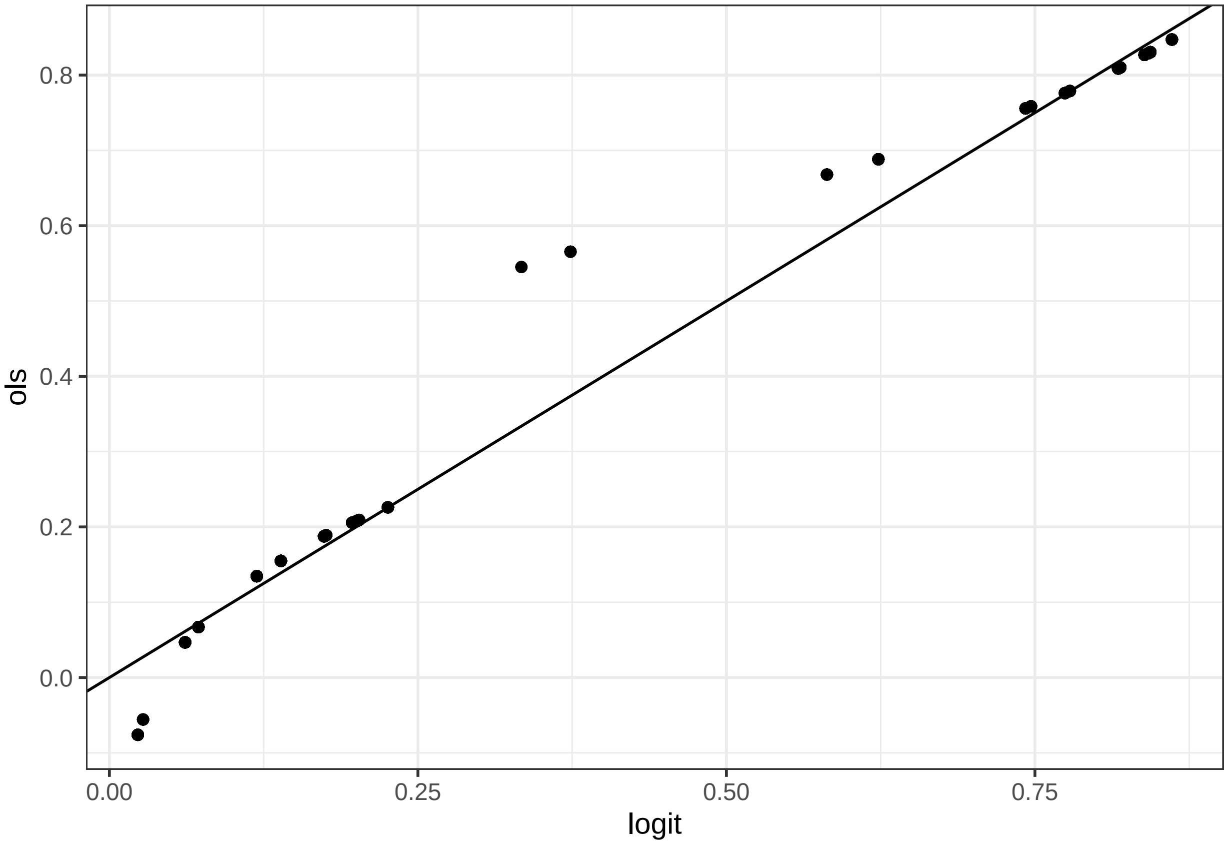
```
##           [,1]
## [1,] 0.8427606
```

6

```
ols = lm(clintonvote ~ dem + female + clintondist, data = clinton)
ols.preds = vector(mode = "logical", nrow(clinton))
logit.preds = vector(mode = "logical", nrow(clinton))

for(i in 1:nrow(clinton)){
  newdata = newdata = c(1, as.numeric(clinton[i,c(2:4)]))
  ols.preds[i] = my_predict(ols$coefficients, newdata, ols = TRUE)
  logit.preds[i] = my_predict(logit$coefficients, newdata, ols = FALSE)
}
```

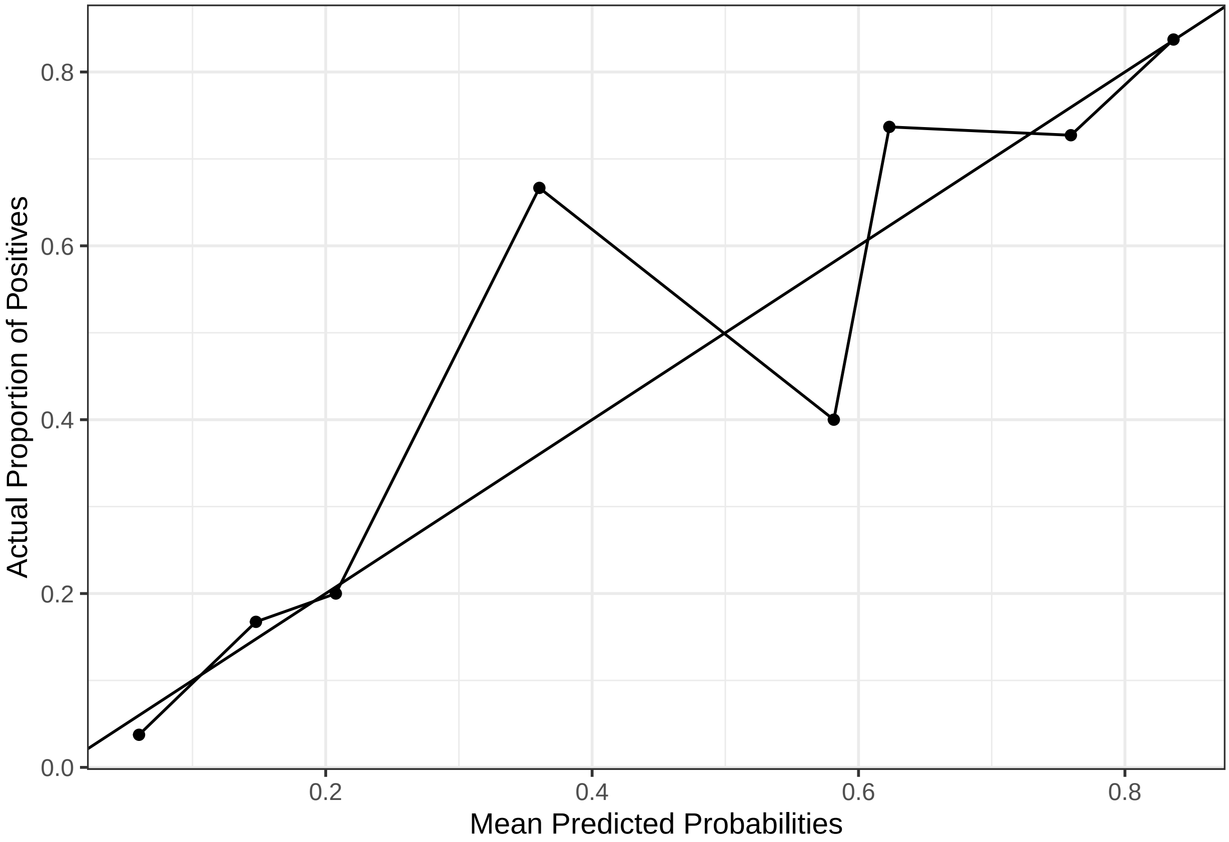
```
data.frame(ols = ols.preds, logit = logit.preds) |>
  ggplot(aes(logit, ols))+
  geom_point()+
  geom_abline(intercept = 0,slope = 1)+
  theme_bw()
```



Bonus

```
bins = cut(logit.preds, breaks = seq(0,1,.1), right = FALSE,
           labels = c(1:10))
bonusDat = data.frame(preds = logit.preds, bins = bins)
mean_prob = aggregate(bonusDat$preds, by = list(bins), FUN=mean)
posi_prob = aggregate(clinton$clintonvote, by = list(bins), FUN=mean)

data.frame(mean_prob = mean_prob$x, posi_prob = posi_prob$x) |>
  ggplot(aes(mean_prob, posi_prob))+
  geom_point()+
  geom_line()+
  geom_abline(intercept = 0, slope = 1)+
  theme_bw()+
  labs(x = "Mean Predicted Probabilities",
       y = "Actual Proportion of Positives")
```


Actual Proportion of Positives
0.0
0.2
0.4
0.6
0.8
0.2
0.4
0.6
0.8
Mean Predicted Probabilities