PSET3_ParthDesai

Parth Desai

2023-02-16

R Markdown

Question 1

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

Part 1.1

```
boot_univariate <- function(datvec, statint, B, alpha){
  samp <- c()
  for (i in 1:B) {
    samp[i] <- statint(sample((datvec), length(datvec), replace = TRUE))
  }
  sd(samp)
  return(quantile(samp, probs = c(alpha/2, (alpha/2) + (1-alpha))))
}
```

Part 1.2

```
boot_univariate(datvec = x, statint = median, B = 10000, alpha = 0.05)
```

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

The boot_univariate function passed with the specific parameters as shown above represent that for dataset x, when we are interested in the mean, with 10,000 resamples of x, there is 95% confidence that the values lie between 33.13953% and 38.56648%.

Bonus 1

```r
summary(x)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000413 0.147888 0.358380 0.510851 0.706927 3.605504
```

```r
boot_univariate(datvec = x, statint = median, B = 10000, alpha = 0.5)
```

```
##       25%       75%
## 0.3467349 0.3656483
```

Question 2

```r
library(ggplot2)
```

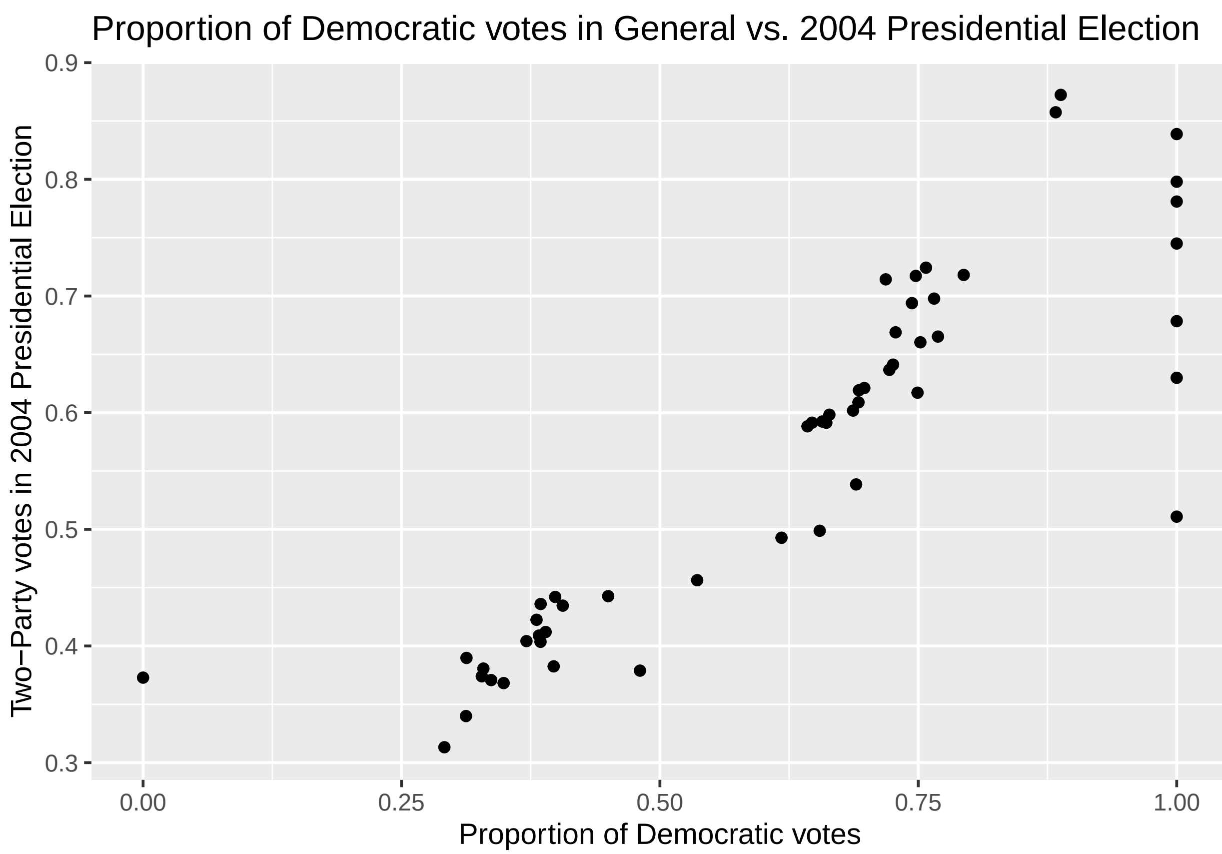
```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

Part 2.1

```r
ca2006 <- read.csv('ca2006.csv')
```

Part 2.2

```r
plot_a <- ggplot(data = ca2006, aes(x=prop_d, y= dem_pres_2004))
plot_a + geom_point() +
  ggtitle("Proportion of Democratic votes in General vs. 2004 Presidential Election") +
  xlab("Proportion of Democratic votes") +
  ylab("Two-Party votes in 2004 Presidential Election")
```

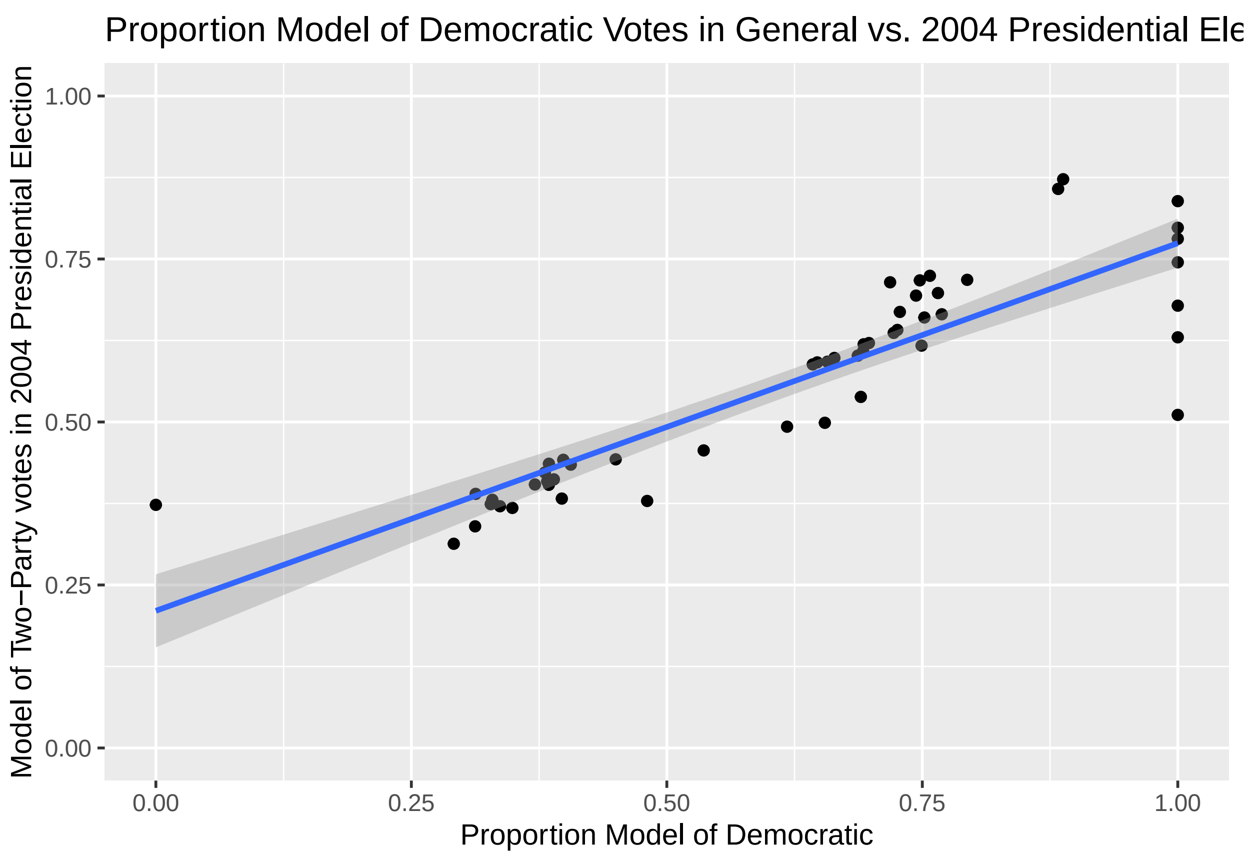


Part 2.3

```r
mod1 <- lm(prop_d ~ dem_pres_2004, data = ca2006)
summary(mod1)
```

```
##
## Call:
## lm(formula = prop_d ~ dem_pres_2004, data = ca2006)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36168 -0.04314 -0.00830  0.01233  0.44754
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.15390    0.05978  -2.574    0.013 *
## dem_pres_2004  1.38268    0.10291  13.436   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1125 on 51 degrees of freedom
## Multiple R-squared:  0.7797, Adjusted R-squared:  0.7754
## F-statistic: 180.5 on 1 and 51 DF,  p-value: < 2.2e-16
```

```r
plot_b <- ggplot(data = mod1, aes(x = prop_d, y = dem_pres_2004))
plot_b + geom_point() + geom_smooth(method = "lm", formula = y ~ x) +
  coord_cartesian(ylim = c(0,1), xlim = c(0,1)) +
  ggtitle('Proportion Model of Democratic Votes in General vs. 2004 Presidential Election') +
  xlab('Proportion Model of Democratic') +
  ylab('Model of Two-Party votes in 2004 Presidential Election')
```



Part 2.4

```r
let_predict <- function(model, x.star){
  a <- model$coefficients
  return(a %*% x.star)
}

newdata1 <- c(1, dem_pres_2004 = 0.5)
let_predict(mod1, newdata1)
```

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

Part 2.5

```
mod2 <- lm(prop_d ~ dem_pres_2004 + dem_pres_2000 + dem_inc, data = ca2006)
```

Part 2.6

```
newdata2 <- c(1, dem_pres_2004 = 0.5, dem_pres_2000 = 0.5, dem_inc = 1)
let_predict(mod2, newdata2)
```

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

Part 2.7

```
set.seed(pi)
B = 10000
bivariate = c()
multivariate = c()
boot_samp <- c()
for (x in 1:B) {
  boot_samp <- sample(nrow(ca2006), length(ca2006$district), replace = TRUE)
  new_df <- ca2006[boot_samp, ]

  mod3 <- lm(prop_d ~ dem_pres_2004, data = new_df)
  mod4 <- lm(prop_d ~ dem_pres_2004 + dem_pres_2000 + dem_inc, data = new_df)

  bivariate[x] = let_predict(mod3, newdata1)
  multivariate[x] = let_predict(mod4, newdata2)
}
```

Part 2.8

```
bivariate_ci <- c(quantile(bivariate, probs = 0.025), quantile(bivariate, probs = 0.975))
bivariate_ci
```

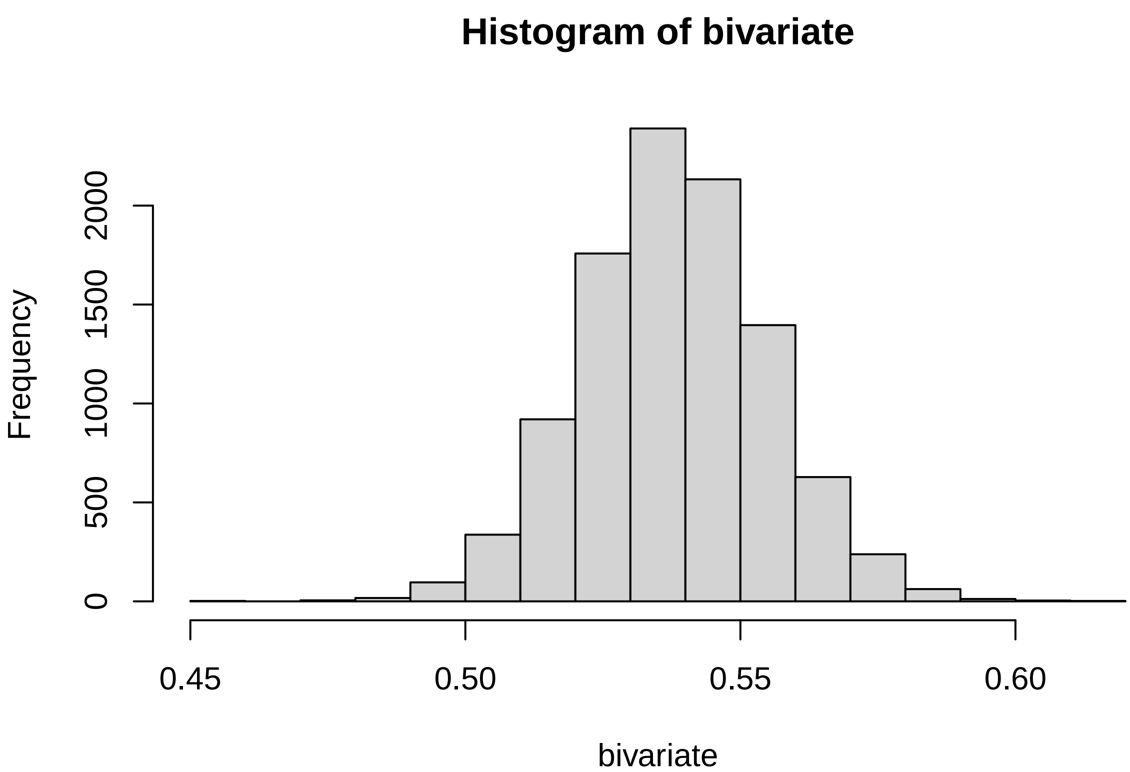
```
##      2.5%     97.5%
## 0.5050168 0.5716776
```

```
multivariate_ci <- c(quantile(multivariate, probs = 0.025), quantile(multivariate, probs = 0.975))
multivariate_ci
```

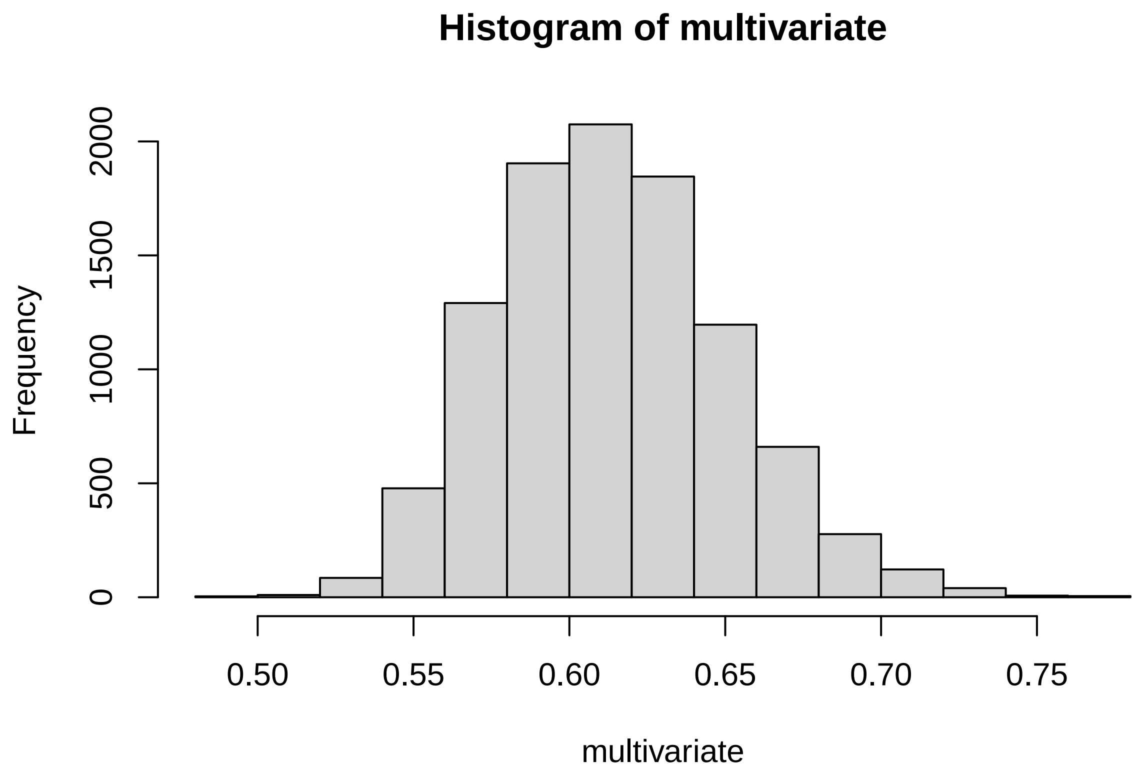
```
##      2.5%     97.5%
## 0.5496060 0.6924033
```

The bivariate confidence interval is from 50.50168% to 57.16776% and the multivariate confidence interval is 54.9606% to 69.24033%

```
hist(bivariate)
```




```
hist(multivariate)
```



Part 2.9

```
bivariate_correct <- c()
for(x in 1:B){
  if(bivariate[x] > 0.5){
    bivariate_correct[x] <- as.numeric(bivariate[x] > 0.5)
  }
  else{
    bivariate_correct[x] <- 0
  }
}
bivariate_only_correct <- subset(bivariate_correct, bivariate_correct[] == 1)

(length(bivariate_only_correct)/ length(bivariate_correct)) * 100
```

```
## [1] 98.8
```

```
multivariate_correct <- c()
for (x in 1:B) {
  if(multivariate[x] > 0.5){
```

```
    multivariate_correct[x] <- as.numeric(multivariate[x] > 0.5)
  }
  else{
    multivariate_correct[x] <- 0
  }
}

multivariate_only_correct <- subset(multivariate_correct, multivariate_correct[] == 1)

(length(multivariate_only_correct)/length(multivariate_correct)) * 100
```

Around 98.8% the bivariate regression reports the Democrat winning

```
## [1] 99.96
```

Around 99.96% the multivariate regression reports the Democrat winning

Question 3

Part 3.1

```
vote92 <- read.csv('vote92.csv')
```

Part 3.2

```
(length(subset(vote92, clintonvote == 1)[,1])/length(vote92[,1])) * 100
```

```
## [1] 45.76458
```

The percentage of voters for Clinton was approximately 45.764% of respondents.

Part 3.3

```
mod5 <- glm(clintonvote ~ dem + female + clintondist, family = binomial(link = "logit"),
            data = vote92)
```

Part 3.4

```
probvot <- function(data_set, model_input, factor1, factor2, factor3,
                    regression_type,x.star){
  model <- list()
  model <- glm(model_input ~ factor1 + factor2 + factor3, family = regression_type,
```

```
                data = data_set)
  return(let_predict(model, x.star))
}
```

Part 3.5

```
probvot(vote92, vote92$clintonvote, vote92$dem, vote92$female, vote92$clintondist,
        binomial(link = "logit"), x.star <- c(1, female = 1, dem = 1, clintondist = 1))
```

```
##          [,1]
## [1,] 1.678914
```

Part 3.6

```
mod6 <- lm(clintonvote ~ dem + female + clintondist, data = vote92)
func <- function(model, datfrm){
  values <- c()
  for (x in 1:length(datfrm[,1])) {
    values[x] <- let_predict(model, c(1, dem = datfrm$dem[x],
                                      female = datfrm$female[x], datfrm$clintondist[x]))
  }
  return(values)
}


lin_regress <- func(mod6, vote92)
head(lin_regress)
```

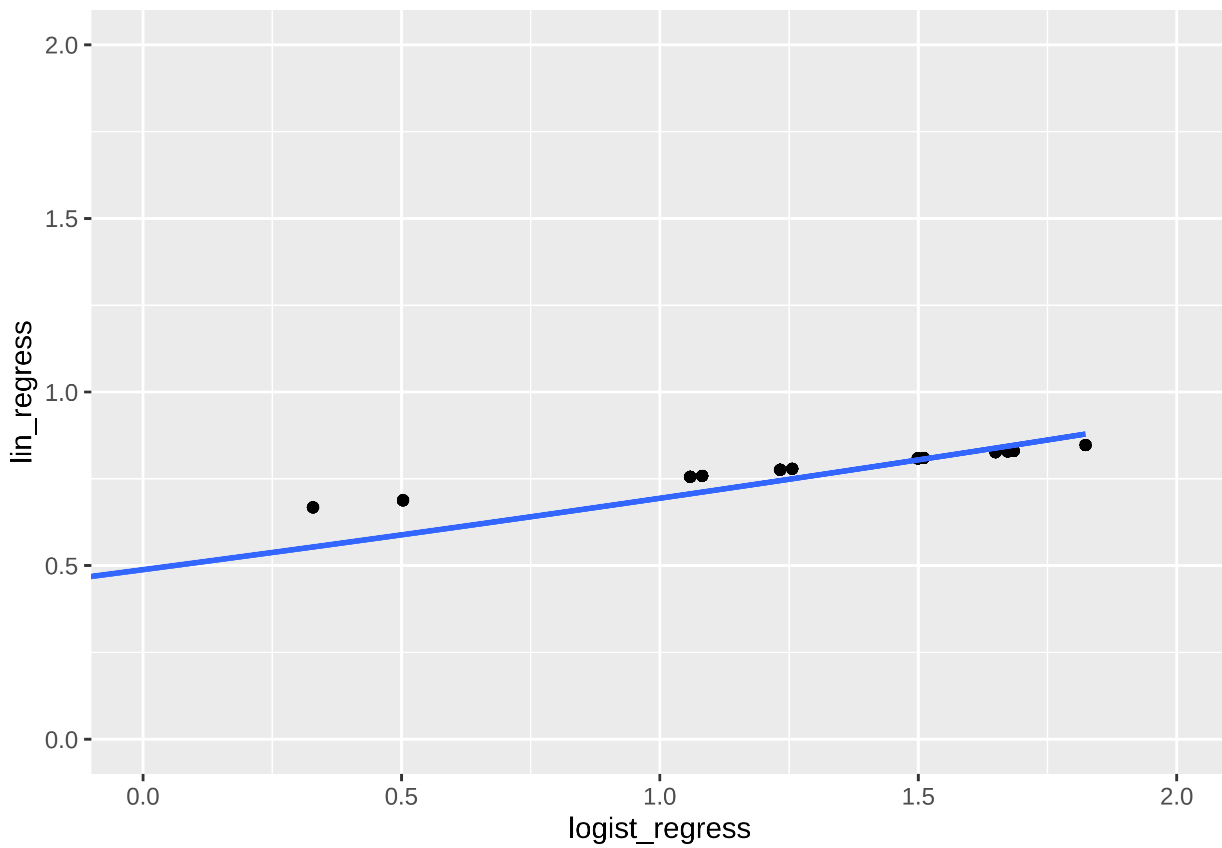
```
## [1] 0.1548988 0.1548988 0.8290671 0.2056878 0.2092911 0.7788589
```

```
mod7 <- glm(clintonvote ~ dem + female + clintondist,
            family = binomial(link = "logit"), data = vote92)
logist_regress <- func(mod7, vote92)
head(logist_regress)
```

```
## [1] -1.823665 -1.823665  1.673063 -1.406974 -1.371828  1.255983
```

```
finale <- data.frame(lin_regress, logist_regress)

h <- ggplot(data = finale, aes(x = logist_regress, y = lin_regress))
h + geom_point() + geom_smooth(method = "lm", formula = y ~ x + I(x^2)) +
  coord_cartesian(ylim = c(0,2), xlim = c(0,2))
```


2.0
1.5
1.0
0.5
0.0
lin_regress
0.0
0.5
1.0
1.5
2.0
logist_regress