ARE 212 Midterm

Kendra Marcoux

3/10/2021

```r
format_it <- function(plot, title, subtitle, xname, yname, xlabel, ylabel){
  plot +
    scale_y_continuous(expand = c(0, 0), labels = ylabel) +
    scale_x_continuous(labels = xlabel) +
    theme_classic() +
    labs(title = title,
         subtitle = subtitle,
         x = xname,
         y = yname) +
    theme(plot.title = element_text(size = 14, face = "bold", hjust = .5),
          plot.subtitle = element_text(size = 14, hjust = .5))
}

scatter_brain <- function(df, xvar, yvar, title, subtitle, xname, yname, xlabel, ylabel){
  format_it(df %>%
              ggplot(aes(x = !!sym(xvar), y = !!sym(yvar))) +
              geom_point(),
            title, subtitle, xname, yname, xlabel, ylabel)
}

regressive_habits <- function(df, yvar, constant, xvar){

  if(constant == "constant"){
    x <- rep(1, NROW(df))
    m0 <- diag(1, NROW(df)) - 1/NROW(df)*rep(1, NROW(df))%*%t(rep(1, NROW(df)))
    xnames <- append("(Intercept)", xvar)
  } else {
    x <- c()
    m0 <- diag(1, NROW(df))
    xnames <- xvar
  }

  if (NROW(xvar)>0){
    for (vars in 1:NROW(xvar)){
        x <- x %>%
          bind_cols(pull(df, !!sym(xvar[[vars]])) %>%
                      na.omit()) %>%
          as.matrix()
    }
  }

  y <- pull(df, !!sym(yvar)) %>%
    na.omit()

  b_coeff <- solve(t(x)%*%x)%*%t(x)%*%y
  yhat <- x%*%b_coeff
  e <- y-yhat
  se <- sqrt(diag(solve(t(x)%*%x)*as.numeric(t(e)%*%e)/(NROW(x) - NCOL(x))))
  white_var <- solve(t(x)%*%x)%*%t(x)%*%diag(diag(e%*%t(e)))%*%x%*%solve(t(x)%*%x)
  white_se <- sqrt(diag(white_var))
  t_stat <- b_coeff/se
  sst <- t(y%*%m0%*%y)
  ssr <- t(e)%*%e
  sse <- t(b_coeff)%*%t(x)%*%m0%*%x%*%b_coeff
  r_sqr <- 1-ssr/sst
  coeff_mat <- as.matrix(bind_cols(b_coeff, se, t_stat))
  colnames(coeff_mat) <- c("coefficient","std. error", "t-stat")
  rownames(coeff_mat) <- xnames
  white_coeff_mat <- as.matrix(bind_cols(b_coeff, se, white_se))
  colnames(white_coeff_mat) <- c("coefficient","ols std. error","white std. error")
  rownames(white_coeff_mat) <- xnames

  list("coeff_mat" = coeff_mat,
       "white_coeff_mat" = white_coeff_mat,
       "xmat" = x,
       "white_var" = white_var,
       "n" = NROW(x),
       "n_minus_k" = NROW(x) - NCOL(x),
       "sst" = sst,
       "ssr" = ssr,
       "sse" = sse,
       "yhat" = yhat,
       "e" = e,
       "r_sqr" = r_sqr)
}

take_it_two_the_stage <- function(df, yvar, constant, x_k, instrument, xvar) {

  red_form_xvars1 <- append(instrument, xvar)
  red_form_xk <- regressive_habits(df, x_k, constant, red_form_xvars1)

  x_k_vec <- pull(df, !!sym(x_k))
  df <- df %>%
    mutate(!!x_k := red_form_xk$yhat)

  red_form_xvars2 <- append(x_k, xvar)
  red_form_y <- regressive_habits(df, yvar, constant, red_form_xvars2)

  z <- as.matrix(bind_cols(1, red_form_y$xmat[,-c(1,2)], pull(df, !!sym(instrument[[1]]))))
  if(NROW(instrument)>1){
    for (vars in 2:NROW(instrument)){
      z <- as.matrix(z %>%
                       bind_cols(pull(df, !!sym(instrument[[vars]])) %>% na.omit()))
      }
  }
```

```r
  x <- as.matrix(bind_cols(1, x_k_vec, red_form_y$xmat[,-c(1,2)]))
  e <- pull(df, !!sym(yvar)) - x %*% red_form_y$coeff_mat[,1]
  iv_se <- sqrt(diag(solve(t(x)%*%z%*%solve(t(z)%*%z)%*%t(z)%*%x)*
                       as.numeric(t(e)%*%e)/red_form_y$n_minus_k))
  white_var <- solve(t(x)%*%z%*%solve(t(z)%*%z)%*%t(z)%*%x)%*%
    t(x)%*%z%*%solve(t(z)%*%z)%*%
    as.matrix(t(z)%*%diag(diag(e%*%t(e))))%*%z%*%
    t(t(x)%*%z%*%solve(t(z)%*%z))%*%
    solve(t(x)%*%z%*%solve(t(z)%*%z)%*%t(z)%*%x)
  white_se <- sqrt(diag(white_var))
  coeff_mat <- as.matrix(bind_cols(red_form_y$coeff_mat[,1],iv_se, white_se))
  colnames(coeff_mat) <- c("coefficient","std. error", "white std. error")
  rownames(coeff_mat) <- append("(Intercept)",red_form_xvars2)
  list("coeff_mat" = coeff_mat,
       "white_var" = white_var)
}

sim_city <- function(n){
  x_i <- rnorm(n,0,1)
  epsilon_i <- (rchisq(n, 1)-1)/sqrt(2)
  true_theta <- c(1,0.5)
  y <- true_theta[1] +true_theta[2]*x_i+epsilon_i

  x <- as.matrix(bind_cols(1, x_i))
  b_coeff <- solve(t(x)%*%x)%*%t(x)%*%y
  bias <- b_coeff[2] - true_theta[2]
  tibble(n= n, bias = bias)
}
```

```r
#Load libraries
library(pacman)
p_load(dplyr, haven, readr, knitr, psych, ggplot2,stats4, stargazer, lmSupport, magrittr,
       qwraps2, Jmisc, qwraps2, rlang, dataCompareR, purrr, tinytex, sandwich, lmtest,
       cowplot, ivreg)
```

Exercise 1

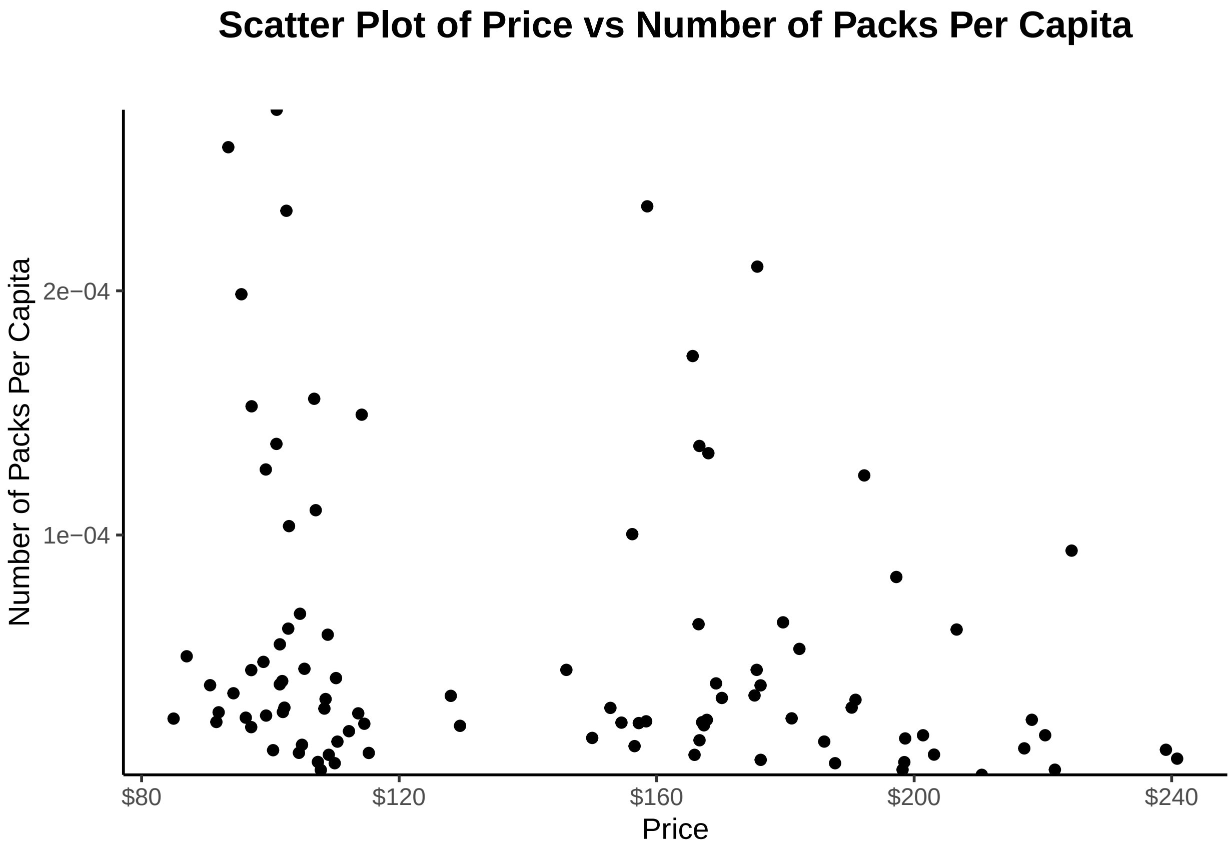
1.) Below I load the data and create a variable q=packs/population=number packs per capita:

```r
raw_data <- read_dta("../Data/cigarros.dta")

cigarros_data <- raw_data %>%
  mutate(q = packs/population)
```

2.) Below I make a scatter plot of q and price:

```r
scatter_brain(cigarros_data,
              "price",
              "q",
              "Scatter Plot of Price vs Number of Packs Per Capita",
              "",
              "Price",
              "Number of Packs Per Capita",
              scales::dollar,
              waiver())
```

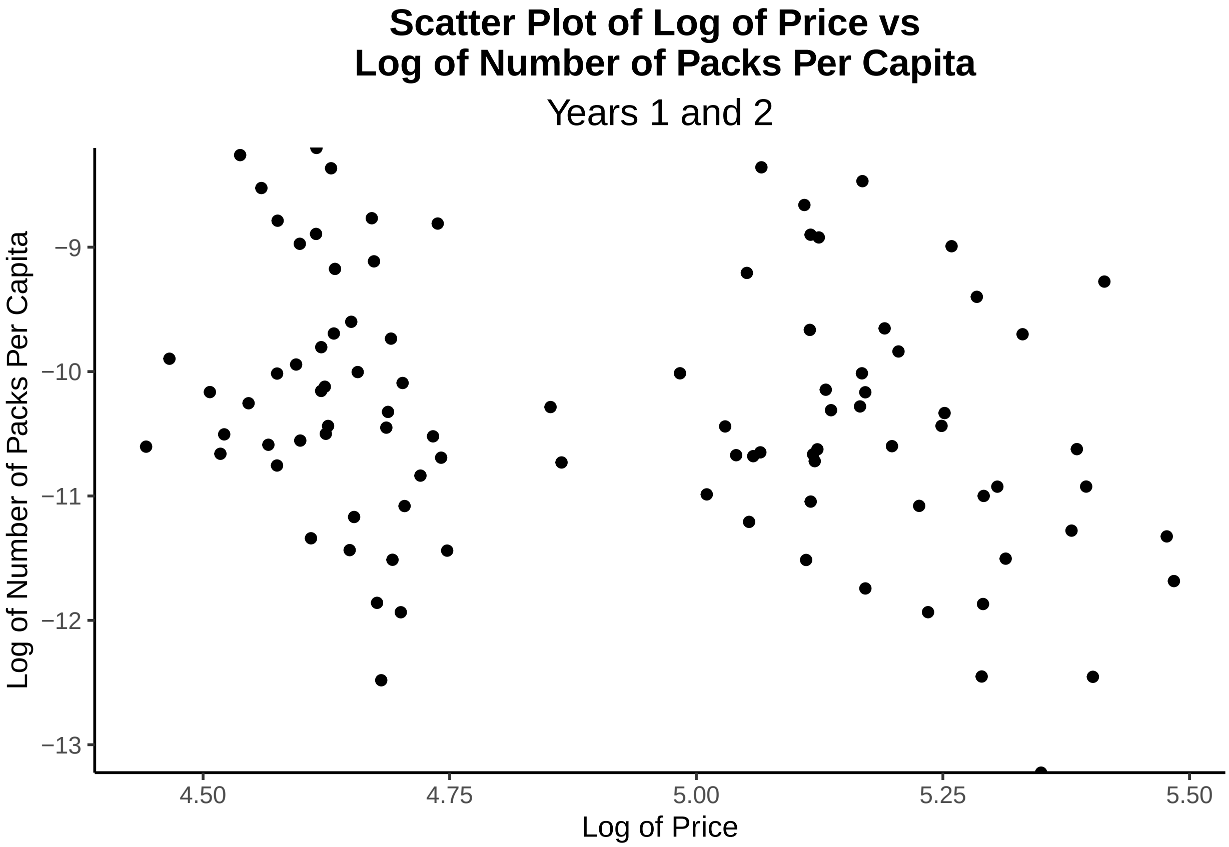


3.) Here I create variables for log price and log q:

```r
cigarros_data <- cigarros_data %>%
  mutate(log_q = log(q),
         log_price = log(price))
```

4.) Below I make a scatter plot of log(q) and log(price):

```
log_scatter <- scatter_brain(cigarros_data,
                             "log_price",
                             "log_q",
                             "Scatter Plot of Log of Price vs \n Log of Number of Packs Per Capita",
                             "Years 1 and 2",
                             "Log of Price",
                             "Log of Number of Packs Per Capita",
                             waiver(),
                             waiver()); log_scatter
```

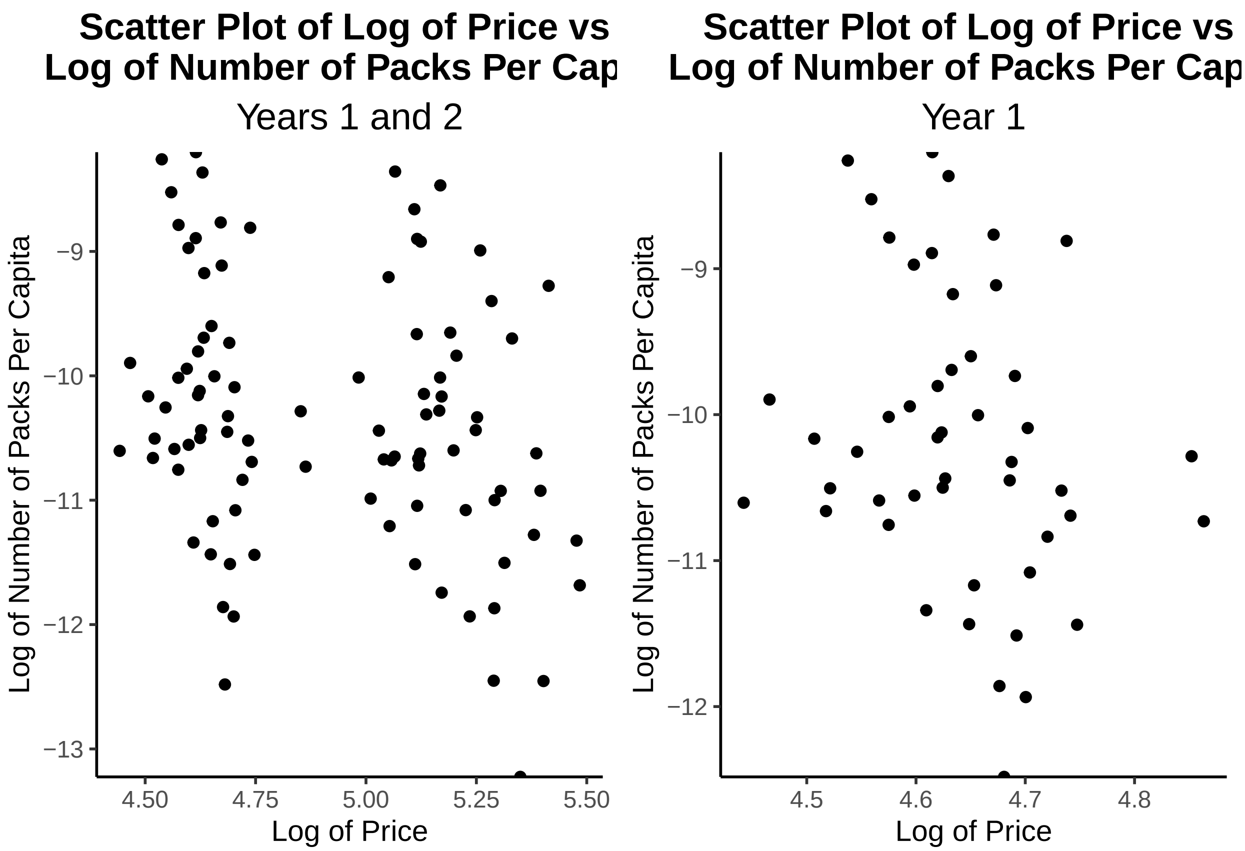


5.) Below I filter to year==1:

```
mydata1 <- cigarros_data %>%
  filter(year==1)
```

6.) Below I plot log(price) and log(q) for year 1 only and compare to the plot of both years. Looking at the two plots we can see that the prices in the plot of both years are grouped into two groups, one with lower prices which are the year one prices, and one with higher prices, which are the year 2 prices. The range of prices is much smaller in the year 1 only graph.

```r
log_scatter_y1 <- scatter_brain(mydata1,
                             "log_price",
                             "log_q",
                             "Scatter Plot of Log of Price vs \n Log of Number of Packs Per Capita",
                             "Year 1",
                             "Log of Price",
                             "Log of Number of Packs Per Capita",
                             waiver(),
                             waiver())
plot_grid(log_scatter, log_scatter_y1)
```



Exercise 2

1.) Below we use the matrix algebra established in the function regressive_habits defined above to estimate the linear model:

$$\log(Q_i) = \beta_0 + \log(P_i)\alpha + \epsilon_i \quad (1)$$

```r
reg1 <- regressive_habits(mydata1,"log_q","constant",c("log_price"))
reg1$coeff_mat
```

```
##             coefficient std. error    t-stat
## (Intercept)    2.809519   7.895234  0.355850
## log_price     -2.798347   1.702217 -1.643942
```

Our estimate $\hat{\alpha}$ from the linear model defined above is -2.7983472. The interpretation of this coefficient is that a one percent increase in the price per pack is associated with a 2.80% decrease in the number of packs consumed per capita.

2.) Our standard errors assuming homoskesdasticity (OLS std. error) and assuming heteroskedasticity (white std. error) are given below:

```
reg1$white_coeff_mat
```

```
##             coefficient ols std. error white std. error
## (Intercept)    2.809519       7.895234         6.124138
## log_price     -2.798347       1.702217         1.320472
```

We can now compare the t-stats for the null of zero assuming homoskedasticity against the t-stat for the same null assuming heteroskedasticity:

```
reg1_tstat_homosk <- reg1$coeff_mat["log_price",3]
reg1_tstat_heterosk <- reg1$white_coeff_mat["log_price",1]/reg1$white_coeff_mat["log_price",3]
```

We have that our t-stat for the null of $\beta_{\log(price)} = 0$ assuming homoskedasticity is -1.6439422 and our t-stat for the null of $\beta_{\log(price)} = 0$ assuming heteroskedasticity is -2.1192015. Clearly the t-stat assuming homoskedasticity is lower in absolute value and at the 95% confidence level we cannot reject the null. Using the white heteroskedastic t-stat, however, we can reject the null hypothesis that $\beta_{\log(price)} = 0$ with 95% confidence.

3.) Below we show the estimated OLS Covariance $\hat{Cov}(b_{constant}, b_{\log(price)})$ assuming heteroskedasticity:

```
reg_1_white_covar <- reg1$white_var[2,1]; reg_1_white_covar
```

```
## [1] -8.08465
```

Then we can see that $\hat{Cov}(b_{constant}, b_{\log(price)})$ assuming heteroskedasticity is equal to -8.0846497.

4.) The variable BB_i is positively correlated with both $\log(price)$ and $\log(Q)$, then our regression in equation (1) would suffer from omitted variable bias. The omission of BB_i would lead to a negative bias in the magnitude of our estimate of $\hat{\alpha}$, and the true size of α, that is the true effect of an increase in price would have on demand, would be higher (less negative) than our estimate in equation 1. This is due to the fact that prices are negatively correlated with demand, but demand and billboards are positively correlated.

Exercise 3

1.) We can use the matrix algebra set up in the function take_it_two_the_stage above:

```
reg2 <- take_it_two_the_stage(mydata1, "log_q","constant","log_price",c("taxs"),c())
reg2$coeff_mat
```

```
##             coefficient std. error white std. error
## (Intercept)    1.695959   8.991968         7.607487
## log_price     -2.558224   1.938747         1.634699
```

We can see both the homoskedasticity consistent and white heteroskedasticity consistent standard errors in the matrix above.

2.) Our null hypothesis is $H_0 : \beta_{\log(price)} = -1$ with the alternative hypothesis that: $H_A : \beta_{\log(price)} \neq -1$. We can use the white, heteroskedasticity robust standard errors to create a test statistic to test this hypothesis. Our test statistic will be given by:

$$t = \frac{\hat{\beta}_{\log(price)} - (-1)}{se^W_{\log(price)}}$$

```
reg2_tstat <- (reg2$coeff_mat["log_price",1]+1)/reg2$coeff_mat["log_price",3]; reg2_tstat
```

```
## [1] -0.9532177
```

```
crit_val_reg2 <- qt(.025, 47, lower.tail=F)
```

Our test statistic is has a t distribution with 47 degrees of freedom so our critical value is 2.0117405 and due to the fact that |-0.9532177|<2.0117405 we cannot reject the null hypothesis that $\beta_{\log(price)} = -1$ at the 5 percent significance level.

Exercise 4

1.) We can run a fit based test on the instruments. We need to run both restricted and unrestricted versions of the first stage of our model:

Our unrestricted model is:

$$log(price) = \beta_0 + taxs\beta_1 + log(taxs)\beta_2 + \epsilon$$

while our restricted model where $\beta_1 = \beta_2 = 0$ is:

$$log(price) = \beta_0 + \epsilon$$

our statistic for the fit-based test is given by:

$$F = \frac{(SSR_r - SSR_u)/J}{s^2}$$

where $s^2 = \frac{e'e}{n-k}$ from the unrestricted model.

```
mydata1 <- mydata1 %>%
  mutate(log_taxs=log(taxs))

unrestricted_reg3 <- regressive_habits(mydata1, "log_price", "constant", c("taxs","log_taxs"))
restricted_reg3 <- regressive_habits(mydata1, "log_price", "constant", c())

f_stat_reg3 <- abs((unrestricted_reg3$ssr - restricted_reg3$ssr)/2/
  (t(unrestricted_reg3$e)%*%unrestricted_reg3$e/unrestricted_reg3$n_minus_k))
```

We have that our F-statistic is 76.1339019 which is greater than 10 so we do not have to worry about weak instruments.

2.) We can use the matrix algebra set up in the function take_it_two_the_stage above to compute the estimates and heteroskedasticity consistent standard errors:

```
reg4 <- take_it_two_the_stage(mydata1,"log_q","constant","log_price",
                              c("taxs","log_taxs"),c())
reg4$coeff_mat
```

```
##               coefficient std. error white std. error
## (Intercept)      1.860137   8.987547          7.54074
## log_price       -2.593626   1.937793          1.62007
```

3.) Our original OLS estimate for α was -2.7983472 while our 2SLS estimate is -2.5936263. Clearly our 2SLS estimate is higher (less negative) than our OLS estimate, which makes sense considering the fact that the OVB mentioned in Exercise 2 Question 4 gave the OLS estimate $\hat{\alpha}$ a negative bias.

4.) Our heteroskedatic consistent standard error in the 2SLS regression is 1.6200705 while our IV heteroskedatic consistent standard error is 1.6346988. We can see that our 2SLS standard error is lower than our IV standard error. This makes sense due to the fact that we often see efficiency gains with added instruments.

Exercise 5

1.) We can use our 2SLS estimate of elasticity from Exercise 4 and mean(price) to compute the implied marginal cost:

```r
marginal_cost <- mean(mydata1$price)+1/log_p_coeff_reg4*mean(mydata1$price)
marginal_cost_pretty <- scales::dollar(marginal_cost)
```

Then we have that our estimated marginal cost is $63.68.

2.) We can use the delta method to construct a 95% confidence interval of estimated marginal cost. Our function $g(\alpha) = mean(price) + \frac{1}{\alpha}mean(price)$ such that $g'(\alpha) = -\frac{mean(price)}{\alpha^2}$. We can then use the delta method to derive the robust (heteroskedastic consistent) variance of our marginal cost estimate such that:

$$V_{MC}^{W} = g'(\alpha) * V_{\hat{\alpha}}^{W} * g'(\alpha)$$

```r
marginal_cost_var <- (-mean(mydata1$price)/log_p_coeff_reg4^2)^2*reg4$white_var
marginal_cost_se <- sqrt(diag(marginal_cost_var))[[2]]

mc_lower_bound <- marginal_cost-qt(1-.05/2, df=nrow(mydata1)-1)*marginal_cost_se
mc_upper_bound <- marginal_cost+qt(1-.05/2, df=nrow(mydata1)-1)*marginal_cost_se
mc_lower_bound_pretty <- scales::dollar(mc_lower_bound)
mc_upper_bound_pretty <- scales::dollar(mc_upper_bound)
```

Then we have that our 95% confidence interval for our estimate of marginal cost is ($13.47, $113.90). We can be 95% confident that the true marginal cost lies within this interval.

3.) We found our standard error for our estimate of marginal cost in question 2 so we can use in our hypothesis tests below:

Our null hypothesis is $H_0 : MC = 10$ and our alternative hypothesis is $H_A : MC \neq 10$. We can derive the following test statistic:

$$t = \frac{\hat{MC} - 10}{se_{MC}^{W}}$$

```r
mc_t <- (marginal_cost -10)/marginal_cost_se
mc_p_val <- pt(mc_t, df=nrow(mydata1)-1, lower.tail = F)
```

We have that our test statistic is given by 2.1506773 and the p value is 0.0183378. This means that we can reject the null hypothesis at the 95% confidence level but not at the 99% percent confidence level.

Exercise 6

1.) You need to include one less than the total number of categories when you are using fixed effects to avoid multicollinearity (or the linear combination of all state level fixed effects would give you the constant vector).

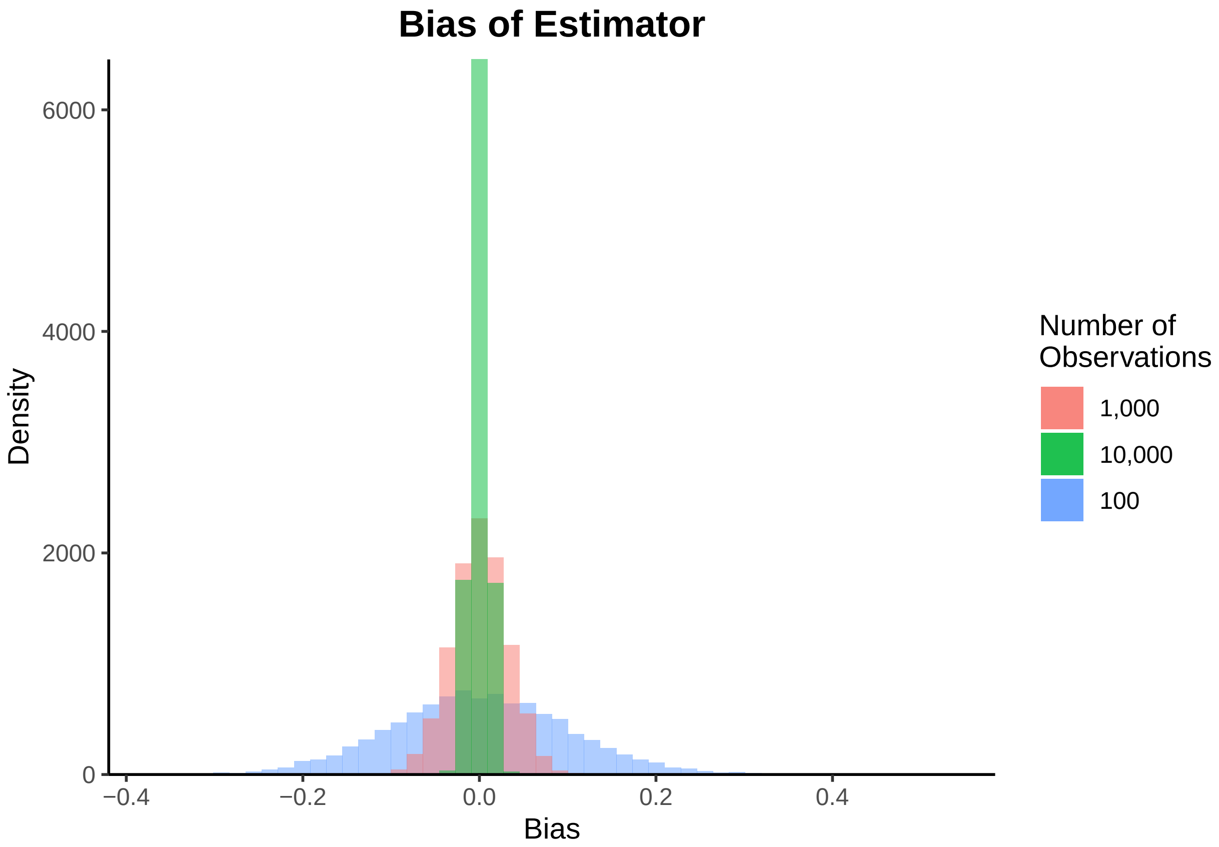
2.) The estimates in columns 1 and 2 are equivalent because fixed effects act to demean the data by state, so the two approaches are numerically equivalent despite the fact that they are computationally different.

Exercise 7

We can now run the simulation sim_city that we defined above:

```
n <- 10000
simulation_results <- c(rep(100, n),rep(1000, n),rep(10000, n)) %>%
  map_dfr(sim_city)

ggplot(simulation_results, aes(x=bias))+
  geom_histogram(data=subset(simulation_results, n==100), aes(fill = "100"), alpha=.5, bins = 50)+
  geom_histogram(data=subset(simulation_results, n==1000), aes(fill = "1,000"), alpha=.5,  bins = 50)+
  geom_histogram(data=subset(simulation_results, n==10000), aes(fill = "10,000"), alpha=.5, bins = 50)+
  scale_fill_discrete(name = "Number of\nObservations") +
  scale_y_continuous(expand = c(0, 0)) +
  theme_classic()+
  labs(title = "Bias of Estimator",
       x = "Bias",
       y = "Density") +
  theme(plot.title = element_text(size = 14, face = "bold", hjust = .5))
```



We can see that the bias of the estimator converges to 0 as the number of observations increases.