Problem Set 1

Owen Jetton

04/06/2022

Question 1

```r
data = read.csv("data-001.csv")
```

Question 2

```r
reg1 = lm(data = data,
          formula = income_black_2010 ~ pop_enslaved_1860 + pop_total_1860 + pop_total_2010)

q2_coef = reg1$coefficients[["pop_enslaved_1860"]]

summary(reg1)
```

```
##
## Call:
## lm(formula = income_black_2010 ~ pop_enslaved_1860 + pop_total_1860 +
##     pop_total_2010, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -38016  -7204  -2791   4140  57433
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.895e+04  6.887e+02  42.036  < 2e-16 ***
## pop_enslaved_1860 -2.670e-01  1.292e-01  -2.067   0.0391 *
## pop_total_1860     5.593e-02  6.349e-02   0.881   0.3787
## pop_total_2010     1.108e-02  1.804e-03   6.142 1.36e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11790 on 706 degrees of freedom
## Multiple R-squared:  0.06072,    Adjusted R-squared:  0.05673
## F-statistic: 15.21 on 3 and 706 DF,  p-value: 1.33e-09
```

The coefficient on "pop_enslaved_1860" is -0.2670247

Question 3

```r
# endogenous variable
Y = as.matrix(data$income_black_2010)
```

```
# exogenous variables (with intercept)
X = matrix(c(rep(1, 710), data$pop_enslaved_1860, data$pop_total_1860, data$pop_total_2010),
                    ncol = 4)

reg_q3 = solve(t(X) %*% X) %*% t(X) %*% Y

reg_q3[2,1]
```

```
## [1] -0.2670247
```

The coefficient on "pop_enslaved_1980" is -0.2670247 which is the same as in question 2.

Question 4

```
reg_fun = function(y, x) {
  coef = solve(t(x) %*% x) %*% t(x) %*% y

  return(coef)
}
```

Results:

```
reg_fun(Y, X)
```

```
##               [,1]
## [1,]  2.895156e+04
## [2,] -2.670247e-01
## [3,]  5.592848e-02
## [4,]  1.107846e-02
```

Question 5

```
reg_fun2 = function(y, x) {

  # coefficient equation
  coef = solve(t(x) %*% x) %*% t(x) %*% y

  # standard errors
    # error (residuals)
  e = (y - x %*% coef)
    # variance sigma estimate calculation
  s_sq = (1/(dim(x)[1] - dim(x)[2]-1))*sum(e^2)

  # calculate variance matrix
  variance_matrix = s_sq * solve(t(x) %*% x)

  # arrange the results
  stnd_errors = sqrt(diag(variance_matrix))

  results = cbind(coef, stnd_errors)

  return(results)

}
```

Results:

```
reg_fun2(Y, X)
```

```
##                      stnd_errors
## [1,]  2.895156e+04 6.892133e+02
## [2,] -2.670247e-01 1.292888e-01
## [3,]  5.592848e-02 6.353467e-02
## [4,]  1.107846e-02 1.804909e-03
```

My function reports the coefficients and standard errors correctly.

Question 6

To be approximately correct, the standard errors reported from my function rely on the assumptions of homoskedasticity, nonautocorrelation, and normally distributed errors:

$$\epsilon|X \sim N(0, \sigma^2 I)$$

Question 7

In order for my coefficients to be interpretable as causal, one needs to assume that the model we're estimating is the true model, that there are no omitted relevant variables, that the exogenous variables are in fact *exogenous*.

Extra Credit

```
reg_fun3 = function(data, var_y, var_x) {

  y = as.matrix(data %>% select(all_of(var_y)))

  x = as.matrix(cbind(intercept = c(rep(1, length(y))),
                      data %>% select(all_of(var_x))))

  # coefficient equation
  coef = solve(t(x) %*% x) %*% t(x) %*% y

  # standard errors
    # error (residuals)
  e = (y - x %*% coef)
    # standard error calculation
  s_sq = (1/(dim(x)[1] - dim(x)[2]-1))*sum(e^2)

  # calculate variance matrix
  variance_matrix = s_sq * solve(t(x) %*% x)

  # arrange the results
  stnd_errors = sqrt(diag(variance_matrix))

  results = cbind(coef, stnd_errors)

  return(results)

}
```

Results:

```r
reg_fun3(data = data,
         var_y = c("income_black_2010"),
         var_x = c("pop_enslaved_1860", "pop_total_1860", "pop_total_2010")
)
```

```
##                    income_black_2010  stnd_errors
## intercept              2.895156e+04 6.892133e+02
## pop_enslaved_1860     -2.670247e-01 1.292888e-01
## pop_total_1860         5.592848e-02 6.353467e-02
## pop_total_2010         1.107846e-02 1.804909e-03
```