R HW4

Your name goes here

Due date: September 20, 2018

Please email your completed problem sets (both the .pdf and the R Markdown files) to Zuhad and Jesse (zuhadhai@stanford.edu and yoderj@stanford.edu) by 1:30 PM Thursday, September 20. Name each file using the convention Firstname_Lastname_rhw4 with the appropriate suffix (either .pdf or .Rmd).

Problem 1: Optimizing OLS Regression

In Lab 4, we have already used optim to minimize the sum of squared errors for a linear regression with an intercept and one or two variable(s). While this approach is different from the matrix-based estimator we used to estimate OLS in the Lab 3 Homework, both approaches should (asymptotically) give us the same answers. However, thus far we have used a function that is specific to the trade dataset inasmuch as we specified the names of the variables that should be included in the regression model inside the function.

In this problem, you will write a more generic function that can be applied to any dataset and any dependent and independent variables, and use optim to minimize the sum of squared errors.

a) Write a function called "sum.squared.errors" that takes four inputs—(1) a vector of parameter values you want to optimize; (2) the data frame you want to work with; (3) the name of the dependent variable; and (4) a list of names of the independent variable(s)—and returns one number: the sum of squared errors. Here are some *hints*:
   - Hint 1: The dataset you will work with contains missing values. Thus, tell your function to omit rows with missing values in at least one of the variables on either side of the regression equation. The "complete.cases()" function will be useful here.
   - Hint 2: Recall that you can express a regression equation by $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$.
   - Hint 3: The errors of a regression model can be computed by $\hat{\epsilon} = \boldsymbol{Y} - \hat{\boldsymbol{Y}} = \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}$, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ need to be matrices.

```
sum.squared.errors <- function(params, data, dv.name, iv.names){
  betas <- params
  d <- data[complete.cases(data[,c(dv.name, iv.names)]),]
  Y <- as.matrix(d[,dv.name])
  X <- as.matrix(cbind(rep(1, nrow(d)), d[,iv.names]))
  Y.hat <- X %*% betas
  model.error <- sum((Y - Y.hat)^2)
    return(model.error)
}
```

b) Now use your "sum.squared.errors" function on a bigger version of the data set we worked with in section. In particular, use optim to find the parameters that minimize the sum of squared errors for a linear regression predicting "free.trade.support" with an intercept term and three variables: "income", "education", and "democrat". That is, we are saying that we can predict whether or not someone supports free trade based on their income, level of education, and their partisanship (specifically, whether they are a democrat or not). To do so, load the "trade2.Rdata" data frame and use optim and your "sum.squared.errors" function to estimate the parameters of the following regression:

$$\text{trade support}_i = \beta_0 + \beta_1 * \text{income}_i + \beta_2 * \text{education}_i + \beta_3 * \text{democrat}_i + \epsilon_i$$

```
## Loading data
load("trade2.Rdata")

## Using optim
optim.estimates <- optim(par=c(0,0,0,0), fn=sum.squared.errors, data=trade,
                         dv.name="free.trade.support",
                         iv.names=c("income", "education", "democrat"))$par
optim.estimates
```

```
## [1]  3.155124e-01  1.613207e-06  7.386189e-02 -5.063435e-02
```

c) Use R's canned "lm()" function to estimate the same model. The "summary()" or "coef()" commands will be useful to pull out the coefficients from an lm object (Hint: We can add multiple right hand side variables in lm() using the plus sign. For example, lm(y ~ x1 + x2, data = data)).

```
lm <- lm(free.trade.support ~ income + education + democrat , data=trade)
summary(lm)
```

```
##
## Call:
## lm(formula = free.trade.support ~ income + education + democrat,
##     data = trade)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5103 -0.4539 -0.3398  0.5404  0.7110
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.156e-01  5.047e-02   6.253  6.2e-10 ***
## income       1.613e-06  8.408e-07   1.919   0.0553 .
## education    7.376e-02  3.918e-02   1.882   0.0601 .
## democrat    -5.074e-02  3.472e-02  -1.461   0.1442
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4936 on 904 degrees of freedom
##   (202 observations deleted due to missingness)
## Multiple R-squared:  0.01533,    Adjusted R-squared:  0.01206
## F-statistic:  4.69 on 3 and 904 DF,  p-value: 0.002947
```

d) Summarize your optim estimates and the lm results. Are the results the same?

```
lm.estimates <- coef(lm)
out <- cbind(optim.estimates, lm.estimates, round(optim.estimates - lm.estimates, digits=5))
colnames(out) <- c("optim", "lm", "diff")
out
```

```
##                     optim            lm     diff
## (Intercept)  3.155124e-01  3.155698e-01 -0.00006
## income       1.613207e-06  1.613414e-06  0.00000
## education    7.386189e-02  7.375901e-02  0.00010
## democrat    -5.063435e-02 -5.074427e-02  0.00011
```