Assignment 3

Your Name

Intro

This R day is going to cover the regression theory we learned in topic #3. For this assignment, I have created simulated data on a set of workers. For explanatory variables, we have whether they went to college and their age. Our key outcome variable is the (hourly) wage earned by the worker.

We will load the data using the `read.csv` function. This function accepts a string containing the file path of the wage data. You might have to change your working directory in RStudio to the folder where this file and the csv are located.

```
df <- read.csv("wage_data_sim.csv")
head(df)
```

```
  age college     wage
1  31       0 30.17237
2  22       0 17.64049
3  22       0 20.90050
4  31       0 30.83628
5  42       0 35.34770
6  55       0 35.54634
```

Running a regression by-hand

In this section, we are going to run a regression completely by hand. Later, we will use a package called `fixest` which will make this much more simple. First, we need to create our matrix `W` and our outcome variable `y`.

To do this, use the function `W <- cbind(...)` which accepts a comma separated list of vectors and will construct a matrix for us. For now, use `1` and `df$age` as the two covariates we want to include. Second, grab the outcome with `y <- df$wage` to match the notation we used in the slides.

```
#
```

Estimating the OLS coefficients

Recall the formula for our OLS coefficients are given by

$$\hat{\beta} = (W'W)^{-1}W'y$$

The three things you need to know to estimate this are: 1. To transpose a matrix, we use `t(W)` (`t` = transpose) 2. To multiply matrices, we use the syntax `A %*% B`. Note that `*` is not correct and will error 3. To invert a matrix, we use `solve(A)` to get A^{-1}

Let's use these to solve for the OLS coefficients

```
#
```

Great! We have estimated our first regression by hand. The result is a vector of length 2 containing out estimated intercept and slope coefficient. Now, we want to estimate standard errors on these coefficients so we can form confidence intervals.

Grabbing the residuals

The first step is to calculate the fitted values $W\beta$ and the residuals $y - W\beta$. We can do this with a simple matrix multiply of `W` and `beta_hat`. Note that `W %*% beta_hat` returns a $N \times 1$ matrix, but we want it to be a vector. We can use `yhat <- as.numeric(W %*% beta_hat)` to convert to a vector.

```
#
```

Estimating Standard Errors

First, form the estimate of Σ consisting of a diagonal matrix with $\hat{u}_i^2$ on the diagonal. You can create this matrix using `diag(resid)`. Store this in a variable called `Sigma_hat`.

```
#
```

Now, we can calculate our variance-covariance matrix as

$$(W'W)^{-1}W'\hat{\Sigma}W(W'W)^{-1}$$

Do that below and store in a variable called `vcov_hat`

```
#
```

To get the standard errors, we want to grab the diagonal elements of $Var(\hat{\beta})$. To do so, we can pass the matrix `vcov_hat` to `diag()`. The standard errors are the square-root of the diagonal elements. Save these to a variable called `se`.

```
#
```

Technically, the `HC1` standard error does a degree-of-freedome correction where we multiply `vcov_hat` by $N/(N-K)$. Modify your code above to do that. You can get N and K via `N <- nrow(W)` and `K <- ncol(W)`. Note this is scalar multiply so you should use the regular `*`.

Form confidence intervals

To form 95% confidence intervals, we will add and subtract 1.96 times the standard error to the point estimate. Do that now to create `ci_lower` and `ci_upper` for the slope coefficient on `age`

```
#
```

Checking for remaining explainable variation

Remember that our goal of forecasting is to explain as much variation as possible in y using our underlying explanatory variables X. One way to visually assess this is to create a plot of a key X variable on the x-axis and your residuals, $\hat{u}$, on the y-axis. Do that using `plot(x = age, y = resid)`below and see if you notice any systematic pattern between `age` and `resid`

```
#
```

If you notice a pattern, go above and add an additional term to `W` to better model the CEF $\mathbb{E}$(wage|age). Rerun the code and see if you have any more systematic variation in `resid` that you can see. Explain what term you added to `W` below. Try not to overfit the sample, though!

Answer:

Form a forecast and conduct inference

Let's predict a person's wage at age 40 using our model. To do so, let's create a vector, W_i, that contains the correct value. You can do `W_i <- c(...)` where `...` are the variables needed to match how you define `W` above.

```
#
```

Then, forecast $W_i'\hat{\beta}$. One again, use `as.numeric()` to create convert this to a scalar. Call this `yhat_i`

```
#
```

Last, let's estimate the variance of our forecast by using

$$W_i' V W_i,$$

where V is our `vcov_hat`. Take the square-root of this to get the standard error on our forecasted `yhat_i`.

```
#
```

Form a 95% confidence interval on your forecasted wage.

```
#
```

Using packages to make life easier

We will use the `fixest` package to do all of these much more simply. Note we already loaded the package above with `library(fixest)`.

Running OLS with `feols`

The main call we will use is `feols()` which estimates a regression model. The syntax will be `feols(formula, data = df, vcov = "hc1")`, where `df` is the name of your data.frame and `formula` specifies the regression you want to run. The form is of `y ~ w1 + w2 + ...` where `y` is the name of your outcome variable and `w1`, `w2`, … are the covariates you want to include.

For example, our formula is `wage ~ 1 + age + age^2`. Since intercepts are so ubiquitous, you do not need the `1 +`, but I do it for clarity. Estimate this regression and store the results in `est`. The easiest way to view the results is to call `print(est)`.

```
#
```

Do the results match what you found by hand? What happens if you delete `vcov = "hc1"`?

Answer:

Forming confidence intervals

Let's redo our exercise of forming a confidence interval around the coefficient on `age`. We can access the coefficients from our estimate using `coef(est)` and the standard errors using `se(est)`.

```
#
```

Or, more easily, we can use `confint(est)`:

```
#
```

Forecasting using `predict`

Last, we can form our forecast by running `predict()` with the argument `newdata`. To do so, we need to create a data.frame holding our `W_i`. Note we don't need to create the intercept of `age^2` anymore, just `df_i <- data.frame(age = 40)` works. Then calling `predict(est, newdata = df_i)` will give us our prediction. Add `se.fit = TRUE` to the `predict` call to get standard error as well.

```
#
```

Do the results match what you found by hand?

Answer:

Review Practice

Let's add some review questions for you to practice running regressions in R.

Try running a regression of `wage` on a cubic polyonmial in `age` and add an indicator for whether a person attended college. You can create an indicator (or set of indicators) using the `i()` function in a formula. Print out the results using `print()`

```
#
```

Review questions: 1. What is the coefficient and standard error on `college`?

Answer:

2. Is the coefficient on `age^3` statistically significant?

Answer:

3. What do we predict the average earnings for a 35 year old worker with a college degree to be?

```
#
```

Answer:

4. How much higher/lower do we expect a 35 year old with a college degree to earn relative to a 30 year old person without a college degree?

```
#
```

Answer: