EDUC 643 Lab: Applied Statistics in Education and Human Services II

Lab 4: 1/30 and 1/31

**Helpsheet for Assignment 1**

**Don’t forget to load packages: library(**tidyverse**)** and use **read.csv()** to read inthe data**.** Use **modelsummary()** to format regression tables.

1. **Use the library() function to load necessary libraries. Two ways to do this:**

**library(**pacman**)**

**p\_load(**here, tidyverse, modelsummary**)**

**OR**

**library(**tidyverse**)**

**library(**here**)** # only necessary for those that are using ` here ` to import data

**library(**modelsummary**)**

1. **Import data using read.csv()**

**If using R script -**

**your\_data** <- **read.csv(“**data/your\_data.csv”**)**

**If using Rmd -**

If this isn’t working, use the `here` function

**your\_data** <- **read.csv(here(“**data/your\_data.csv”**))**

**From the weblink -**

**your\_data** <- **read.csv(“**paste\_web\_link.csv”**))**

1. **Change variable type to match their measurement scale, especially nominal/ordinal variables using the factor() function.**

your\_data$column\_name <- **factor(your\_data$column\_names**,  
 levels = c(...), #levels from codebook  
 labels = c(...)**)**  #labels from codebook

1. **Select variables for summary statistics creating new object with selected variables**

**your\_data\_desc** <- **select(your\_data**, **c(**variable1, variable2, variable3, variable4,

variable5, variable6**))** #include the variables whose summary you want. Notice the new object name for summary variables **your\_data\_desc**

**Rename variables using rename() so the names are meaningful. Alternatively, you can export the descriptive table to word and change names there.**

**your\_data\_desc <-** your\_data\_desc **%>%**

**rename(**"Per-pupil expenditure ($)" **= ppe,** # example from the assignment

"Variable 2 name" **= variable2,** # continue with variables from data

"variable three name” **= variable3)** # continue with variables from data

**Creating summary statistics table using datasummary()**

**datasummary(**formula = **All(your\_data\_desc) ~ Mean + SD,**

data = **your\_data\_desc,**

title= **“Give a title to the table”,**

notes = **“Write a note”,**

output = “table/descriptive\_cont.docx”**)** # If using R script, you can save this file and include it in the memo. If using Rmd, you don’t need this argument and should remove it.

1. **Visualize the bivariate relationship using ggplot()**

**ggplot(**data = **your\_data**, **aes(**x = predictor\_variable, y = outcome\_variable**))** +

**geom\_point(**color = "your\_color"**)** +

**geom\_smooth(**method = 'lm', se = F, color = "other\_color"**)** +

**labs(**x = "Your X Axis Title Here",

y = "Your Y Axis Title Here"**)** +

**theme\_minimal()**

1. **OLS fit using lm() function**

**your\_model\_name** <- **lm(**outcome\_variable **~** predictor\_variable**,** data =your\_data**)**

**summary(**your\_model\_name**)**

1. **Regression table using modelsummary()**

**modelsummary(your\_model\_name,**

stars= T,

gof\_omit = "Adj.|AIC|BIC|Log",

coef\_rename = c("name\_in\_dataset" = "new\_name"), #renames labels that will appear in table

notes = "Write a note",

title= "Write a title",

output = "Give a path"**)** #refer note in D. abour datasummary. Same applies here

1. **Look at one specific observation in your dataset**

**subset\_data <- filter**(your\_data**,** variable3 **==** "variable3 condition")

1. **Regression Assumptions**

**Extract fitted values and residuals**

your\_data$predict **<- predict(your\_model\_name)**

your\_data$raw\_resid **<- resid(your\_model\_name)**

your\_data$std\_resid **<- rstandard(your\_model\_name)**

your\_data$stu\_resid **<- rstudent(your\_model\_name)**

**Graph residuals (std\_resid and stu\_resid) to examine their distribution**

**Histogram**

**ggplot(**your\_data**, aes(x = your\_variable)) +**

**geom\_histogram() +**

**labs(**title = “Give a title”,

x = “Title of x-axis”,

y = “Title of y-axis"**)**

**QQ plot**

**ggplot(**your\_data**) +**

**stat\_qq(aes(**sample = **your\_variable**)**) +**

**geom\_abline(**color = **'your\_color') +**

**theme\_minimal()**

**Plotting fitted vs. residuals**

**ggplot(**your\_data**, aes(**x = **predict,** y = **raw\_resid)) +**

**geom\_point() +**

**geom\_hline(**yintercept = 0, color = "red", linetype="dashed"**) +**

**ylab(**"Raw Residualts"**) +**

**xlab(**"Fitted values"**) +**

**theme\_minimal()**