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

Lab 4: 1/28 and 1/29

**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.

\*\* You may have to re-type quotation marks if copying-pasting the code into R as the quotation character for MS word and R varies.

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) # necessary if using .Rmd to import data

 library(modelsummary)

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

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

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**

desc\_stats <- your\_data %>%

select(variable1, variable2, variable3, variable4, variable5, variable6…)

#include the variables whose summary you want. Notice the new object name for summary variables **desc\_stats**

**Rename variables using names() so the names are meaningful.**

names(desc\_stats) <- c(”Per-pupil expenditure ($)" , “ Intuitive Name” , “Another Name” ...)

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

datasummary\_skim(desc\_stats ,

type = “all”,

fun\_numeric = list(Mean = Mean, SD =

 SD, Min = Min, Median = Median, Max = Max),

title=“Give a title to the table”, # often doesn’t work, do by hand

notes = “Write a note”,

output = “tables/descriptive\_stats.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",

title = “Your Title Here”) +

 theme\_minimal()

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

 mod1 <- lm(outcome\_variable ~ predictor\_variable, data = your\_data)

summary(mod1)

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

modelsummary(mod1,

stars= T,

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

 coef\_rename = c("variable1" = "New Name",

“variable2” = “New Name”), #renames labels that will appear in table

 notes = "Write a note",

title= "Give a title") # can output to tables folder if using R script

# You can edit modelsummary() code to change which statistics and estimates appear (such as p-value and confidence intervals). If you want to change the things appearing on the table, ask. Or use ?modelsummary() in R

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 (new variables)**

 your\_data$predict **<- predict(mod1)**

your\_data$raw\_resid **<- resid(mod1)**

your\_data$std\_resid **<- rstandard(mod1)**

your\_data$stu\_resid **<- rstudent(mod1)**

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

**Histogram**

ggplot(your\_data, aes(x = std\_resid)) +

geom\_histogram() +

labs(title = “Give a title”,

x = “Title of x-axis”,

y = “Title of y-axis")

**# standardized residuals** are more suitable for this diagnostic because they provide a normalized measure of residuals to check for deviations from normality

**QQ plot**

ggplot(your\_data) +

 stat\_qq(aes(sample = stu\_resid)) +

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

 theme\_minimal()

**#** it’s better to use **studentized residuals** for this diagnostic because they account for leverage and are more accurate for determining if residuals deviate from a normal distribution.

**Plotting fitted vs. residuals**

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

 geom\_point() +

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

 labs( x = “Fitted Values” ,

y = “Raw Residualts" ,

title = “ Raw Vs. Fitted Residuals Title” ) +

theme\_minimal()