# Final V3 - No Help Comments

Here we set our working directory and call it

setwd("~/Desktop/PSCI 390 Final Project")

getwd()

We make sure our libraries for this code are running, such as plotly, tidyverse, and ggpubr. If these are not installed, you can install them by using “ install.packages(plotly)” and so on, the other easier way is to go to your top menu and look for package installer, and search for the package and install it that way.

library(plotly)

library(tidyverse)

library(ggpubr)

Here is where we read our file, in this case it is a csv file

data.final<- read.csv("Refugee Final.csv")

I need to use gridExtra and Scales in order to make the scaling work, without it I could not turn scientific notation to commas as you will see in the code below

require(gridExtra)

require(scales)

# Side by Side Comparison of Greece and Turkey: Refugees and Cases #4

* Here I put all of my code to a single variable called “refugee.plot.1”.
* ggplot is another way to plot things and has more flexibility.
* Data = data.final will pull from our csv which we named data.final.
* aes(x = years) , I am making sure that the colum with the Years on them (2010-2019) are used in the x axis.
* Geom\_point(aes(y = Greece.Refugee),color = “blue”) Is to make my plot to use a point instead of a line, and make the y axis the Greece.refugee colum from the csv file and color it blue.
* The same as above is used for turkey, it is also layed out on the y axis with a different color.
* We use scale\_x\_continuous, and scale\_y\_continuous in order to… make those axis continuous, for example, x axis will be from 2010-2019 in breaks of 1, like 2010,2011 and so on.
* Scale\_y\_continuous is not broken into a sequence and so it is its own self., and here we turn our scientific notation, into commas by using labels = comma

refugee.plot.1 <- ggplot(data = data.final, aes( x = Years)) + geom\_point(aes(y = Greece.Refugee),

color = "blue") + geom\_point(aes(y = Turkey.Refugee), color = "red") + scale\_x\_continuous(breaks=seq(2010,2019,1), name = "Year") + scale\_y\_continuous(name = " # of Refugees Emigrationg to Greece & Turkey", labels = comma)

cases.plot.2 <- ggplot(data = data.final, aes( x = Years)) + geom\_point(aes(y = Greece.Case), color= "blue") + geom\_point(aes(y= Turkey.Case), color = "red") + scale\_x\_continuous(breaks=seq(2010,2019,1), name = "Year") + scale\_y\_continuous(name= " Number of Human Rights Violations against Greece & Turkey")

grid.arrange(refugee.plot.1, cases.plot.2, ncol=2)

* Grid.arrange puts our two plots together side by side and prints it out ☺

# Side by Side Comparison of Greece self, and Turkey self in regards to refuges and cases. #5

refugee.plot.3 <- ggplot(data = data.final, aes(x = Years)) + geom\_point(aes(y = Greece.Refugee), color = "blue") + geom\_point(aes(y= Greece.Case), color = "steelblue3") + scale\_x\_continuous(breaks=seq(2010,2019,1), name = "Year") + scale\_y\_continuous(name = " Greece Cases & Refugee Comparison")

refugee.plot.4 <- ggplot(data = data.final, aes(x = Years)) + geom\_point(aes(y = Turkey.Refugee), color = "red") + geom\_point(aes(y = Turkey.Case), color = "pink1") + scale\_x\_continuous(breaks=seq(2010,2019,1), name = "Year") + scale\_y\_continuous(name = "Turkey Cases & Refugee Comparison", labels = comma)

grid.arrange(refugee.plot.3, refugee.plot.4, ncol=2)

# Interactive Visualization Slide #4 & Slide #5 -> Slide 6

first.slide <- subplot (refugee.plot.1, case.plot.2)

ggplotly(first.slide)

* We once again create a new variable and it is called, first.slide and we attach a subplot which holds two plots.
* Now we use ggplotly to call first.slide and turn it into an interactive visualization!

second.slide <- subplot (refugee.plot.3, refugee.plot.4)

ggplotly(second.slide)

# Side by Side Comparison of Greece and Turkey Refugees and Cases in regards to correlation ; #7

correlation.plot.1 <- ggscatter(data.final, x = "Greece.Case", y = "Turkey.Case", add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson", xlab = "Greece Cases", ylab = "Turkey Cases")

correlation.plot.2 <- ggscatter(data.final, x = "Greece.Refugee", y = "Turkey.Refugee", add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson", xlab = "Greece Refugees", ylab = "Turkey Refugees") + scale\_y\_continuous( labels = comma)

* Here we make a variable, correlation.plot.1, take in ggscatter and all of its data.
* So, as ggscatter sounds, it is a scatterplot used for correlations. Here we use pearson correlation to find r for the x and y axis.

grid.arrange(correlation.plot.1, correlation.plot.2, ncol=2)

# Side by Side of Greece Self, and Turkey Self in regards to correlation in Refugees and Cases ; #8

correlation.plot.3 <- ggscatter(data.final, x = "Greece.Case", y = "Greece.Refugee", add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson", xlab = "Greece Cases", ylab = "Greece Refugees")

correlation.plot.4 <- ggscatter(data.final, x = "Turkey.Case", y = "Turkey.Refugee", add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson", xlab = "Turkey Cases", ylab = "Turkey Refugees") + scale\_y\_continuous(labels = comma)

grid.arrange(correlation.plot.3, correlation.plot.4, ncol=2)

# Visualization of Slide #7 & #8 -> #9

fourth.slide <- subplot(correlation.plot.1, correlation.plot.2)

ggplotly(fourth.slide)

fifth.slide <- subplot(correlation.plot.3, correlation.plot.4)

ggplotly(fifth.slide)

* I hope this helped, if you have questions please ask the professor or you are wealcome to email me ☺
* Castr299@cougars.csusm.edu

# End of R

# R with comments will be uploaded to github