Workshop Exercise

In this exercise, we will examine some data from the World Happiness Report, a landmark survey of the state of global happiness. Our goal is to better understand how some variables may be related to global happiness! Here are some of the main variables in our dataset:

* Country or region
* Happiness score
* GDP per capita
* Social support
* Healthy life expectancy
* Freedom to make life choices
* Generosity
* Perceptions of corruption

## 1.1 General Guidelines

The main goal of this exercise is to have some hands-on experience with R. Don’t worry if we do not complete all of the exercise tasks, the idea for now is to get started and see how far we can get! If you get stuck along the way do not hesitate to ask for help. You can use the ? command followed by a function to get some help on how to use that function.

?mean

## 1.2 Getting ready

In R, many functions can be found inside ‘packages’. Think of packages as a collection of useful functions that R users create to make your data analysis life easier and more exciting. During this exercise we will try to use some functions from several different packages.

To install a package you can use install.packages("package.name"), and then load the package with library(package.name).

install.packages(name.of.package)  
library(package)  
  
  
#Run this part of the code to install all required packages for this exercise  
if(!require(dplyr)){install.packages("dplyr")}   
if(!require(ggplot2)){install.packages("ggplot2")}   
if(!require(Hmisc)){install.packages("Hmisc")}   
if(!require(pastecs)){install.packages("pastecs")}   
if(!require(psych)){install.packages("psych")}   
if(!require(corrplot)){install.packages("corrplot")}   
if(!require(ggExtra)){install.packages("ggExtra")}   
if(!require(plotly)){install.packages("plotly")}   
if(!require(purrr)){install.packages("purrr")}   
if(!require(tidyr)){install.packages("tidyr")}   
if(!require(PerformanceAnalytics)){install.packages("PerformanceAnalytics")}

Done! Now you have access to a multitude of functions that can help your analysis.

## 1.3 First step: Importing the data

First of all, we need to import our data into R! You can find all datasets in the “Dataset.world.happiness” folder. There are a total of 5 datasets in csv format (2015, 2016, 2017, 2018, 2019). Try using the read.csv function to import the datasets

data.2015 <- read.csv("2015.csv")

## 1.4 Adding a new variable

Now that we have the datasets imported in R you can have a look at them by using the functions View, head, or str. Try adding a new variable called “Year” to the datasets.

View(data)  
head(data)  
str(data)  
  
#Creating a new variable  
data.2015$name.of.new.variable <- 2015

## 1.5 Merging the different datasets

We now have five datasets in R, but to continue working on our questions it may be easier to merge all the datasets in a single dataframe. Try using the full\_join function and the piping term %>% to merge the datasets.

data.raw <- data.2015 %>%   
 full\_join(data.2016) %>%  
 full\_join(data.2017) %>%  
 full\_join(data.2018) %>%  
 full\_join(data.2019)

## 1.6 Cleaning the data

Let’s have a closer look at the data and plot a histogram for the Happiness.score variable. You can try to use the hist function or the qplot function of the ggplot2 package.

qplot(data.raw$Happiness.score)

There seems to be something off, the distribution of Happiness.score seems very odd with some very extreme scores. Let’s have a closer look by ordering the Happiness.score values from highest to lowest

data.raw[order(data.raw$Happiness.score,decreasing = T),]

It seems our dataset contains some data from Venus, Mars and Pluto, which does not fit within our research question. Let’s remove these observations. As usual with R, there are several different solutions for any given issue. For removing rows from your dataset you could try using the filter function as follows:

data.raw2 <- data.raw %>% filter(Country != c("Pluto","Venus","Mars"))  
  
qplot(data.raw2$Happiness.score)

Now our data is clean, let’s save this dataset in a different object

data.clean <- data.raw2

One last modification: Change the Year variable to factor

data.clean$Year <- as.factor(data.clean$Year)

## 1.7 Descriptive statistics

Now we are ready to start examining our data with some descriptive statistics! R has many different functions for running descriptive statistics. Try using, for example, the summary function from base R. Or perhaps the describe function of the Hmisc package, the describe function of the psych package, or the stat.desc function of the pastecs package.

You can also run descriptive statistics by group with the describe.by function from the psych package.

summary(data.clean)  
  
library(Hmisc)  
Hmisc::describe(data.clean)  
  
library(psych)  
psych::describe(data.clean)  
  
library(pastecs)  
pastecs::stat.desc(data.clean)  
  
#Descriptive statistics by group  
  
psych::describe.by(data.clean, "Year")

## 1.8 Visualizing the data

Having tables for descriptive statistics is useful, but in many cases we will want to create some graphs for better visualising our data. Let’s try to run some plots with the following functions:

library(ggExtra)  
library(plotly)  
library(purrr)  
library(tidyr)  
  
#Let's plot some histograms  
  
qplot(data.clean$Happiness.score)  
  
#We can also plot many variables at the same time  
data.clean %>%  
 keep(is.numeric) %>%   
 gather() %>%   
 ggplot(aes(value)) +  
 facet\_wrap(~ key, scales = "free") +  
 geom\_histogram()  
  
#Some initial exploratory plots  
  
theme\_set(theme\_bw()) # pre-set the bw theme.  
  
ggplot(data.clean, aes(GDP, Happiness.score)) +   
 geom\_jitter(aes(col=Year)) +   
 geom\_smooth(aes(col=Year), method="lm", se=F) +  
 labs(title ="Happiness vs GDP by year")  
  
ggplot(data.clean, aes(Freedom, Happiness.score)) +   
 geom\_jitter(aes(col=Year)) +   
 geom\_smooth(aes(col=Year), method="lm", se=F) +  
 labs(title ="Happiness vs Freedom by year")  
  
g <- ggplot(data.clean, aes(Social.support, Happiness.score)) +   
 geom\_jitter(aes(col=Year)) +   
 geom\_smooth(aes(col=Year), method="lm", se=F) +  
 labs(title ="Happiness vs Social Support by year")  
  
ggMarginal(g, type = "histogram", fill="transparent")  
  
ggplot(data.clean, aes(Social.support, Happiness.score)) +   
 geom\_jitter () +   
 geom\_smooth(method="lm", se=F) +  
 labs(title ="Happiness vs Social Support by year") +   
 facet\_wrap(facets = vars(Year))

As a bonus, you can try to create some interactive plots with the plotly package!

fig <- plot\_ly(  
 data.clean, x = ~GDP, y = ~Happiness.score,  
 color = ~Year, size = ~GDP,  
 #Hover text:  
 text = ~paste("GDP: ", GDP, '$<br>Country:', Country)  
)

## 1.9 Inferential statistics

Time to answer some of our research questions. First, let’s test some correlations between our main variables of interest.

library(corrplot)  
  
variables <- data.clean[,c("Happiness.score","Corruption","GDP","Social.support","Healthy.life.expectancy","Freedom","Generosity")]  
  
correlation.matrix <- cor(variables,use="complete.obs")  
  
corrplot(correlation.matrix, type = "upper", order = "hclust",   
 tl.col = "black", tl.srt = 45)  
  
library("PerformanceAnalytics")  
chart.Correlation(variables, histogram=TRUE, pch=19)

Finally, let’s try fitting a regression model and check some of the main assumptions for this analysis.

model <- lm(Happiness.score ~ Corruption + Freedom+ GDP + Generosity + Social.support + Healthy.life.expectancy, data = data.clean)  
  
par(mfrow = c(2, 2))  
plot(model)  
  
summary(model)