InstagramFollowers\_FinalProject

Soma Shekar Vayuvegula

08/07/2022

# Importing and Cleaning Data

## Instagram Categories Dataset importing

insta\_categories <- read.csv('/Users/somashekarvayuvegula/Documents/Workspace/dsc520/completed/Final\_Project/data/top\_200\_instagrammers\_categories.csv')  
head(insta\_categories)

## Username Main.Category Main.video.category  
## 1 cristiano Sports Sports  
## 2 kyliejenner Fashion & Beauty Fashion & Beauty  
## 3 leomessi Sports Sports  
## 4 selenagomez Entertainment Music  
## 5 therock   
## 6 kimkardashian Fashion & Beauty Fashion & Beauty

head(insta\_categories$Main.Category)

## [1] "Sports" "Fashion & Beauty" "Sports" "Entertainment"   
## [5] "" "Fashion & Beauty"

## Instagrammers Details Dataset importing

insta\_details <- read.csv('/Users/somashekarvayuvegula/Documents/Workspace/dsc520/completed/Final\_Project/data/top\_200\_instagrammers\_details.csv')  
head(insta\_details)

## Username Channel.Name Country Url  
## 1 cristiano cristiano ES https://www.instagram.com/cristiano  
## 2 kyliejenner kyliejenner US https://www.instagram.com/kyliejenner  
## 3 leomessi leomessi AR https://www.instagram.com/leomessi  
## 4 selenagomez selenagomez US https://www.instagram.com/selenagomez  
## 5 therock therock US https://www.instagram.com/therock  
## 6 kimkardashian kimkardashian US https://www.instagram.com/kimkardashian

## Instagram Followers and Likes Dataset importing

insta\_followers <- read.csv('/Users/somashekarvayuvegula/Documents/Workspace/dsc520/completed/Final\_Project/data/top\_200\_instagrammers\_likes\_followers\_Jul2022.csv')  
  
head(insta\_followers)

## Username Likes Posts Followers Boost.Index Comments.Avg.  
## 1 cristiano 22876451727 3328 465027234 92 51758.331  
## 2 kyliejenner 43048545079 6921 356687629 91 47534.121  
## 3 leomessi 4670492197 875 347032978 90 47044.540  
## 4 selenagomez 8442642603 1835 334551681 93 39167.116  
## 5 therock 9562231242 6660 327064138 91 8529.747  
## 6 kimkardashian 14920061391 5603 323090977 91 16964.807  
## Views.Avg. Avg..1.Day Avg..3.Day Avg..7.Day Avg..14.Day Avg..30.Day  
## 1 17009494 NA NA 3321113 5327340 6948659  
## 2 22875473 NA NA 1223002 2196528 4692459  
## 3 11761596 NA 4810554 3199807 5359469 5668454  
## 4 10723973 NA NA NA 2340219 2340219  
## 5 5413831 NA NA 713970 1101339 1165227  
## 6 9642516 NA 2583151 2699978 2704005 2586789  
## Engagement.Rate Engagement.Rate..60.Days.  
## 1 0.014915592 0.015903093  
## 2 0.017617215 0.016188635  
## 3 0.015533562 0.019045021  
## 4 0.013912687 0.007719662  
## 5 0.004425938 0.003722545  
## 6 0.008303645 0.009688863

## Removing unwanted columns

insta\_followers <- subset (insta\_followers, select = -c(Comments.Avg.,Views.Avg.,Avg..1.Day,Avg..3.Day,Avg..7.Day,Avg..14.Day,Avg..30.Day,Engagement.Rate,Engagement.Rate..60.Days.))  
  
head(insta\_followers)

## Username Likes Posts Followers Boost.Index  
## 1 cristiano 22876451727 3328 465027234 92  
## 2 kyliejenner 43048545079 6921 356687629 91  
## 3 leomessi 4670492197 875 347032978 90  
## 4 selenagomez 8442642603 1835 334551681 93  
## 5 therock 9562231242 6660 327064138 91  
## 6 kimkardashian 14920061391 5603 323090977 91

## Instagram Followers and Likes Dataset importing

library("readxl")  
country\_names <- read.csv('/Users/somashekarvayuvegula/Documents/Workspace/dsc520/completed/Final\_Project/data/iso-country-codes.csv', na = "\\N", quote = '')  
head(country\_names)

## Alpha.2.code Alpha.3.code English.short.name.lower.case Numeric.code  
## 1 AD ASM Andorra 16  
## 2 AE UAE United Arab Emirates 804  
## 3 AF ALA Afghanistan 248  
## 4 AG ATA Antigua and Barbuda 10  
## 5 AI AGO Anguilla 24  
## 6 AL AFG Albania 4  
## ISO.3166.2  
## 1 ISO 3166-2:AS  
## 2 ISO 3166-2:UA  
## 3 ISO 3166-2:AX  
## 4 ISO 3166-2:AQ  
## 5 ISO 3166-2:AO  
## 6 ISO 3166-2:AF

# Final Dataset

## Merging all the datasets

df\_details\_combined <- merge(insta\_categories,insta\_details,by.x="Username",by.y="Username")  
df\_followers\_combined <-merge(df\_details\_combined,insta\_followers,by.x="Username",by.y="Username")  
df\_final <-merge(df\_followers\_combined,country\_names[ , c("Alpha.2.code", "English.short.name.lower.case")],by.x="Country",by.y="Alpha.2.code")  
  
names(df\_final)[names(df\_final)=="English.short.name.lower.case"] <- "Country.name"  
names(df\_final)[names(df\_final)=="Main.video.category"] <- "Sub.category"  
head(df\_final)

## Country Username Main.Category Sub.category Channel.Name  
## 1 AE nusr\_et nusr\_et  
## 2 AI norafatehi Entertainment Movies norafatehi  
## 3 AR georginagio Fashion & Beauty Fashion & Beauty georginagio  
## 4 AR leomessi Sports Sports leomessi  
## 5 AR paulodybala Sports Sports paulodybala  
## 6 AU chrishemsworth Entertainment Movies chrishemsworth  
## Url Likes Posts Followers  
## 1 https://www.instagram.com/nusr\_et 1358263112 2302 46891641  
## 2 https://www.instagram.com/norafatehi 1660332211 1682 41161527  
## 3 https://www.instagram.com/georginagio 1323180384 726 39025459  
## 4 https://www.instagram.com/leomessi 4670492197 875 347032978  
## 5 https://www.instagram.com/paulodybala 1843671992 1263 47720068  
## 6 https://www.instagram.com/chrishemsworth 1731131414 859 55165178  
## Boost.Index Country.name  
## 1 81 United Arab Emirates  
## 2 83 Anguilla  
## 3 74 Argentina  
## 4 90 Argentina  
## 5 85 Argentina  
## 6 86 Australia

## Removing the rows for which followers, likes,username, main category, sub category country name are blank

df\_final <- df\_final[!(df\_final$Username == "" | df\_final$Main.Category == ""| df\_final$Sub.category == ""| df\_final$Likes == ""| df\_final$Followers == ""| df\_final$Country.name == ""), ]  
  
head(df\_final)

## Country Username Main.Category Sub.category Channel.Name  
## 2 AI norafatehi Entertainment Movies norafatehi  
## 3 AR georginagio Fashion & Beauty Fashion & Beauty georginagio  
## 4 AR leomessi Sports Sports leomessi  
## 5 AR paulodybala Sports Sports paulodybala  
## 6 AU chrishemsworth Entertainment Movies chrishemsworth  
## 7 BB badgalriri Entertainment Music badgalriri  
## Url Likes Posts Followers  
## 2 https://www.instagram.com/norafatehi 1660332211 1682 41161527  
## 3 https://www.instagram.com/georginagio 1323180384 726 39025459  
## 4 https://www.instagram.com/leomessi 4670492197 875 347032978  
## 5 https://www.instagram.com/paulodybala 1843671992 1263 47720068  
## 6 https://www.instagram.com/chrishemsworth 1731131414 859 55165178  
## 7 https://www.instagram.com/badgalriri 13027355720 4837 133436105  
## Boost.Index Country.name  
## 2 83 Anguilla  
## 3 74 Argentina  
## 4 90 Argentina  
## 5 85 Argentina  
## 6 86 Australia  
## 7 88 Barbados

# Questions for frture steps

## What kind of plot are required to show the optimal output

## What is the optimal form to represent the result

# What information is not self-evident?

## After eliminating the missing data, combining based on the Username & country and removing unwanted columns fom the final dataset, final dataset becomes very less comparing to the initial one. The result is going to be based on the available dataset which is relatively very small.

# What are different ways you could look at this data?

## We can make the prediction based on the country name, main category, followers and likes but these are not just enough data to predict which category is more successful in a particular country. Please suggest some other way which will be opt in different ways.

# How do you plan to slice and dice the data?

## Slicing and dicing the data is happened in the final dataset by merging and eliminating unwanted cloumns.

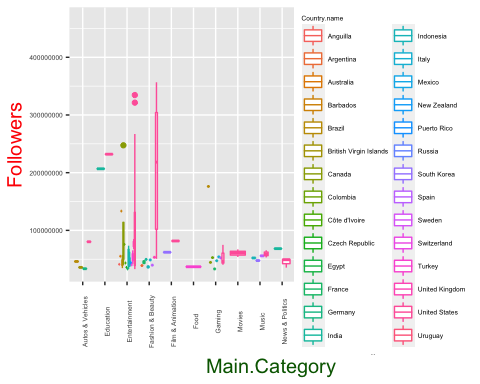
# How could you summarize your data to answer key questions?

## Data has the usenrname, url, country, country name, posts, likes, followers, main category and sub category. Based on these column we can able to answer our questions.

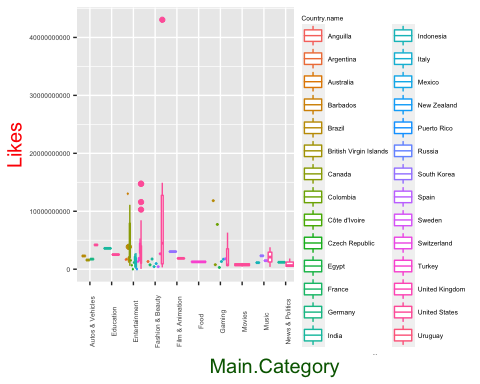
# What types of plots and tables will help you to illustrate the findings to your questions?

# Boxplot

library(ggplot2)  
options(scipen = 999)  
boxplot<-ggplot(data=df\_final,aes(x=Main.Category,y=Followers))+geom\_boxplot(aes(colour=Country.name))  
  
boxplot + theme(axis.title.x=element\_text(colour="DarkGreen",size = 15),  
 axis.title.y = element\_text(colour = "Red",size = 15),  
 axis.text.x = element\_text(size = 5, angle = 90),  
 axis.text.y = element\_text(size=5),  
 legend.title = element\_text(size=5),  
 legend.text=element\_text(size=5),  
 legend.position = c(1,1),  
 legend.justification = c(1,1))



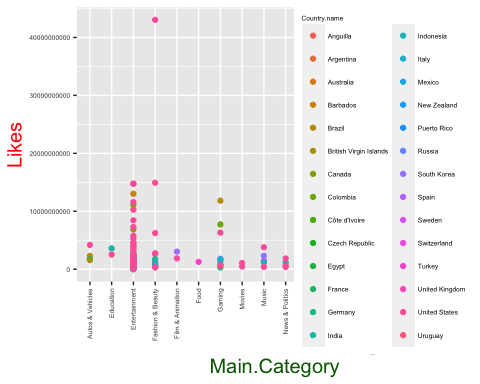
options(scipen = 999)  
boxplot<-ggplot(data=df\_final,aes(x=Main.Category,y=Likes))+geom\_boxplot(aes(colour=Country.name))  
  
boxplot + theme(axis.title.x=element\_text(colour="DarkGreen",size = 15),  
 axis.title.y = element\_text(colour = "Red",size = 15),  
 axis.text.x = element\_text(size = 5, angle = 90),  
 axis.text.y = element\_text(size=5),  
 legend.title = element\_text(size=5),  
 legend.text=element\_text(size=5),  
 legend.position = c(1,1),  
 legend.justification = c(1,1))

 # Scatter plot

library(ggplot2)  
  
options(scipen = 999)  
scatter\_plot <- ggplot(data=df\_final,aes(x=Main.Category,y=Followers))+geom\_point(aes(colour=Country.name))+  
 theme(axis.title.x=element\_text(colour="DarkGreen",size = 15),  
 axis.title.y = element\_text(colour = "Red",size = 15),  
 axis.text.x = element\_text(size=5,angle = 90, vjust = 0.5, hjust=1),  
 axis.text.y = element\_text(size=5),  
 legend.title = element\_text(size=5),  
 legend.text=element\_text(size=5),  
 legend.position = c(1,1),  
 legend.justification = c(1,1))  
scatter\_plot

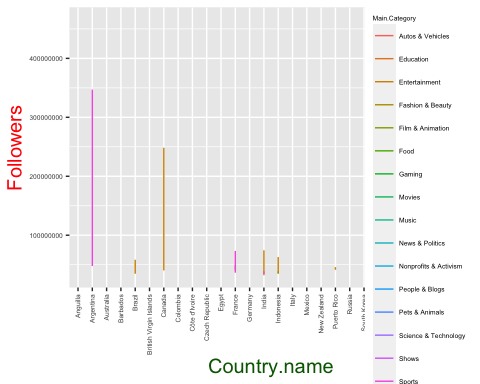


options(scipen = 999)  
scatter\_plot <- ggplot(data=df\_final,aes(x=Main.Category,y=Likes))+geom\_point(aes(colour=Country.name))+  
 theme(axis.title.x=element\_text(colour="DarkGreen",size = 15),  
 axis.title.y = element\_text(colour = "Red",size = 15),  
 axis.text.x = element\_text(size=5,angle = 90, vjust = 0.5, hjust=1),  
 axis.text.y = element\_text(size=5),  
 legend.title = element\_text(size=5),  
 legend.text=element\_text(size=5),  
 legend.position = c(1,1),  
 legend.justification = c(1,1))  
scatter\_plot

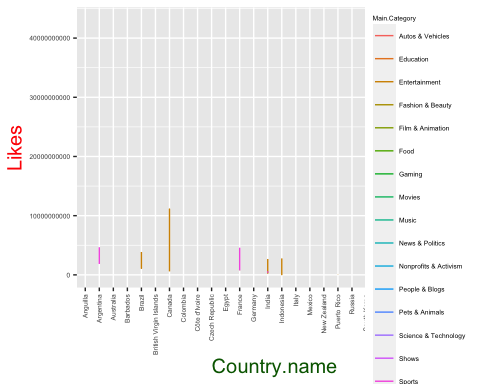


# Trend lines

options(scipen = 999)  
trend\_line<-ggplot(data=df\_final,aes(x=Country.name,y=Followers,colour=Main.Category))+geom\_line()  
  
trend\_line+theme(axis.title.x=element\_text(colour="DarkGreen",size = 15),  
 axis.title.y = element\_text(colour = "Red",size = 15),  
 axis.text.x = element\_text(size=5,angle = 90, vjust = 0.5, hjust=1),  
 axis.text.y = element\_text(size=5),  
 legend.title = element\_text(size=5),  
 legend.text=element\_text(size=5),  
 legend.position = c(1,1),  
 legend.justification = c(1,1))



options(scipen = 999)  
trend\_line<-ggplot(data=df\_final,aes(x=Country.name,y=Likes,colour=Main.Category))+geom\_line()  
  
trend\_line+theme(axis.title.x=element\_text(colour="DarkGreen",size = 15),  
 axis.title.y = element\_text(colour = "Red",size = 15),  
 axis.text.x = element\_text(size=5,angle = 90, vjust = 0.5, hjust=1),  
 axis.text.y = element\_text(size=5),  
 legend.title = element\_text(size=5),  
 legend.text=element\_text(size=5),  
 legend.position = c(1,1),  
 legend.justification = c(1,1))



# Table with sum and mean of followers based on country and main category

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

summarise(group\_by(df\_final, Country.name, Main.Category), sum(Followers), mean(Followers))

## `summarise()` has grouped output by 'Country.name'. You can override using the  
## `.groups` argument.

## # A tibble: 76 × 4  
## # Groups: Country.name [28]  
## Country.name Main.Category `sum(Followers)` `mean(Followers)`  
## <chr> <chr> <dbl> <dbl>  
## 1 Anguilla Entertainment 41161527 41161527   
## 2 Argentina Fashion & Beauty 39025459 39025459   
## 3 Argentina Sports 394753046 197376523   
## 4 Australia Entertainment 55165178 55165178   
## 5 Barbados Entertainment 133436105 133436105   
## 6 Brazil Autos & Vehicles 46091767 46091767   
## 7 Brazil Entertainment 311186018 44455145.  
## 8 Brazil Gaming 176162107 176162107   
## 9 Brazil Nonprofits & Activism 43950253 43950253   
## 10 Brazil Shows 34714162 34714162   
## # … with 66 more rows

# Table with sum and mean of likes based on country and main category

library(dplyr)  
  
summarise(group\_by(df\_final, Country.name, Main.Category), sum(Likes), mean(Likes))

## `summarise()` has grouped output by 'Country.name'. You can override using the  
## `.groups` argument.

## # A tibble: 76 × 4  
## # Groups: Country.name [28]  
## Country.name Main.Category `sum(Likes)` `mean(Likes)`  
## <chr> <chr> <dbl> <dbl>  
## 1 Anguilla Entertainment 1660332211 1660332211   
## 2 Argentina Fashion & Beauty 1323180384 1323180384   
## 3 Argentina Sports 6514164189 3257082094.  
## 4 Australia Entertainment 1731131414 1731131414   
## 5 Barbados Entertainment 13027355720 13027355720   
## 6 Brazil Autos & Vehicles 2282083715 2282083715   
## 7 Brazil Entertainment 13609352481 1944193212.  
## 8 Brazil Gaming 11825612530 11825612530   
## 9 Brazil Nonprofits & Activism 620329319. 620329319.  
## 10 Brazil Shows 1440246511 1440246511   
## # … with 66 more rows

# Table with sum and mean of followers and likes based on country and main category

library(dplyr)  
  
df\_final %>%  
 group\_by(Country.name, Main.Category) %>%  
 summarise(sum\_followers=format(sum(Followers), scientific=FALSE),  
 sum\_likes=format(sum(Likes), scientific=FALSE),  
 mean\_followers=format((mean(Followers)), scientific=FALSE),  
 mean\_likes=format((mean(Likes)), scientific=FALSE))

## `summarise()` has grouped output by 'Country.name'. You can override using the  
## `.groups` argument.

## # A tibble: 76 × 6  
## # Groups: Country.name [28]  
## Country.name Main.Category sum\_followers sum\_likes mean\_followers mean\_likes  
## <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 Anguilla Entertainment 41161527 16603322… 41161527 1660332211  
## 2 Argentina Fashion & Bea… 39025459 13231803… 39025459 1323180384  
## 3 Argentina Sports 394753046 65141641… 197376523 3257082094  
## 4 Australia Entertainment 55165178 17311314… 55165178 1731131414  
## 5 Barbados Entertainment 133436105 13027355… 133436105 130273557…  
## 6 Brazil Autos & Vehic… 46091767 22820837… 46091767 2282083715  
## 7 Brazil Entertainment 311186018 13609352… 44455145 1944193212  
## 8 Brazil Gaming 176162107 11825612… 176162107 118256125…  
## 9 Brazil Nonprofits & … 43950253 620329319 43950253 620329319   
## 10 Brazil Shows 34714162 14402465… 34714162 1440246511  
## # … with 66 more rows

## Filter data based on country to see which category tops the list

## Filter the country name based on prediction to be done.

## Example: I want to predict and see which category of instagram is successful in India

library(dplyr)  
df\_final %>%  
 group\_by(Country.name, Main.Category) %>%  
 summarise(sum\_followers=format(sum(Followers), scientific=FALSE),  
 sum\_likes=format(sum(Likes), scientific=FALSE),  
 mean\_followers=format((mean(Followers)), scientific=FALSE),  
 mean\_likes=format((mean(Likes)), scientific=FALSE)) %>%  
 filter(any(Country.name == 'India'))

## `summarise()` has grouped output by 'Country.name'. You can override using the  
## `.groups` argument.

## # A tibble: 7 × 6  
## # Groups: Country.name [1]  
## Country.name Main.Category sum\_followers sum\_likes mean\_followers mean\_likes  
## <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 India Education 206743723 35924902… 206743723 3592490225  
## 2 India Entertainment 764462619 21124091… 54604473 1508863665  
## 3 India Fashion & Beau… 49721095 17572135… 49721095 1757213598  
## 4 India News & Politics 68330604 11909760… 68330604 1190976040  
## 5 India People & Blogs 54422099 77674850… 54422099 7767485056  
## 6 India Sports 73780559 958397403 36890280 479198702   
## 7 India Trailers 43215034 16087819… 43215034 1608781966

# Do you plan on incorporating any machine learning techniques to answer your research questions? Explain.

## With the help the plot and summarized tables, we can answer our question and there won’t be any requirement for machine learning.