Defining the Effect of Percentage of Household Internet Access and Recent GDP on the Pricing System of Netflix Subscriptions in Various Countries

Karolina Sydor cms7634

Daniel Almeraz da32543

Introduction

Netflix is a popular streaming service that has been globally adopted as it provides a variety of quality film media to consume. Ever since the growing demand for this streaming service, Netflix subscription types and prices have emerged and been modified. However, the subscription prices are not the same for every country. In this project, we will be analyzing the various variables and their effects on the pricing system of Netflix subscriptions for each country. The variables that we will be particularly investigating are countries' population percentage of internet access and GDP. There are three datasets we will be using: the Netflix subscription price dataset, the population percentage of internet access dataset, and the global GDP dataset. The Netflix data set [1] was obtained from the Kaggle website and it possesses eight variables: country code, country, library size, number of TV shows, number of movies, basic subscription cost, standard subscription cost, and premium subscription cost. In this dataset, we will be primarily looking at the variable's country, library size, basic subscription cost, standard subscription cost, and premium subscription cost. The second dataset obtained was the population percentage of internet access dataset [2] which was found from the World Bank organization with 65 variables: country name, country code, indicator name, indicator code, and the years 1965 through 2020. In this dataset, we will be investigating the variables country name and the year 2019. The final dataset we will be looking at is the GDP dataset [3] and this dataset was also collected from the World Bank organization with 65 variables: country name, country code, indicator name, indicator code, and the years 1965 through 2020. For this dataset, we will be looking at the country names and 2020 variables. We are interested in these datasets as it is very plausible the economic stance of various countries can affect the pricing system of media services and we decided that analyzing the effects of countries' GDP and population of internet access are feasible variables to cause an impact on subscription cost of Netflix. The potential trend line we expect to see is the higher the GDP and percent of internet access are for a country, the higher the cost of each subscription type will be.

Work Cited

[1] https://www.kaggle.com/datasets/prasertk/netflix-subscription-price-in-different-countries (https://www.kaggle.com/datasets/prasertk/netflix-subscription-price-in-different-countries)
[2] https://data.worldbank.org/indicator/IT.NET.USER.ZS (https://data.worldbank.org/indicator/IT.NET.USER.ZS)
[3] https://data.worldbank.org/indicator/NY.GDP.MKTP.CD (https://data.worldbank.org/indicator/NY.GDP.MKTP.CD)

Importing proper packages and the data sets

In order to facilitate recreation of our code, we downloaded the datasets from the previously given sources and uploaded them to a GitHub repository from which we can load the data directly.

```
#Setting up our libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)

#Importing the Netflix subscription fee dataset from a HTML
netflix_subscription_fee = read_csv(url("https://raw.githubusercontent.com/dalmeraz/netflix_pric
e_influences/main/Netflix%20subscription%20fee%20Dec-2021.csv"))
```

```
## Rows: 65 Columns: 8
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Country_code, Country
## dbl (6): Total Library Size, No. of TV Shows, No. of Movies, Cost Per Month ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#Importing the household percentage internet access data set from a HTML
internet_access = read_csv(url("https://raw.githubusercontent.com/dalmeraz/netflix_price_influen
ces/main/internet_percentage.csv")) %>% select(-'...66')
```

```
## New names:
## * `` -> ...66
## Rows: 266 Columns: 66-- Column specification ------------------------------------------------
--------
## Delimiter: ","
## chr  (4): Country Name, Country Code, Indicator Name, Indicator Code
## dbl (49): 1960, 1965, 1970, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, ...
## lgl (13): 1961, 1962, 1963, 1964, 1966, 1967, 1968, 1969, 1971, 1972, 1973, ...
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#Importing the household percentage internet access data set from a HTML
gdp_values = read_csv(url("https://raw.githubusercontent.com/dalmeraz/netflix_price_influences/m
ain/gdp_values.csv")) %>% select(-'...66')
```

```
## New names:
## * `` -> ...66
## Rows: 266 Columns: 66-- Column specification ------------------------------------------------
--------
## Delimiter: ","
## chr  (4): Country Name, Country Code, Indicator Name, Indicator Code
## dbl (61): 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ...
## lgl  (1): ...66
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

To provide a general picture for the structure of our datasets, we print the first 6 rows of each using the `head()` function.

```
#Taking a general and quick look at out imported datasets
head(netflix_subscription_fee)
```

```
## # A tibble: 6 x 8
##   Country_code Country   `Total Library Size` `No. of TV Shows` `No. of Movies`
##   <chr>        <chr>                    <dbl>             <dbl>           <dbl>
## 1 ar           Argentina                 4760              3154            1606
## 2 au           Australia                 6114              4050            2064
## 3 at           Austria                   5640              3779            1861
## 4 be           Belgium                   4990              3374            1616
## 5 bo           Bolivia                   4991              3155            1836
## 6 br           Brazil                    4972              3162            1810
## # ... with 3 more variables: `Cost Per Month - Basic ($)` <dbl>,
## #   `Cost Per Month - Standard ($)` <dbl>, `Cost Per Month - Premium ($)` <dbl>
```

```
head(internet_access)
```

```
## # A tibble: 6 x 65
##   `Country Name`  `Country Code` `Indicator Name` `Indicator Code` `1960` `1961`
##   <chr>           <chr>          <chr>            <chr>             <dbl> <lgl>
## 1 Aruba           ABW            Individuals usi~ IT.NET.USER.ZS       NA NA
## 2 Africa Eastern~ AFE            Individuals usi~ IT.NET.USER.ZS       NA NA
## 3 Afghanistan     AFG            Individuals usi~ IT.NET.USER.ZS       NA NA
## 4 Africa Western~ AFW            Individuals usi~ IT.NET.USER.ZS       NA NA
## 5 Angola          AGO            Individuals usi~ IT.NET.USER.ZS       NA NA
## 6 Albania         ALB            Individuals usi~ IT.NET.USER.ZS       NA NA
## # ... with 59 more variables: `1962` <lgl>, `1963` <lgl>, `1964` <lgl>,
## #   `1965` <dbl>, `1966` <lgl>, `1967` <lgl>, `1968` <lgl>, `1969` <lgl>,
## #   `1970` <dbl>, `1971` <lgl>, `1972` <lgl>, `1973` <lgl>, `1974` <lgl>,
## #   `1975` <dbl>, `1976` <dbl>, `1977` <dbl>, `1978` <dbl>, `1979` <dbl>,
## #   `1980` <dbl>, `1981` <dbl>, `1982` <dbl>, `1983` <dbl>, `1984` <dbl>,
## #   `1985` <dbl>, `1986` <dbl>, `1987` <dbl>, `1988` <dbl>, `1989` <dbl>,
## #   `1990` <dbl>, `1991` <dbl>, `1992` <dbl>, `1993` <dbl>, `1994` <dbl>, ...
```

```
head(gdp_values)
```

```
## # A tibble: 6 x 65
##   `Country Name`  `Country Code` `Indicator Name` `Indicator Code` `1960` `1961`
##   <chr>           <chr>          <chr>            <chr>             <dbl>  <dbl>
## 1 Aruba           ABW            GDP per capita ~ NY.GDP.PCAP.CD     NA     NA
## 2 Africa Eastern~ AFE            GDP per capita ~ NY.GDP.PCAP.CD    148.   147.
## 3 Afghanistan     AFG            GDP per capita ~ NY.GDP.PCAP.CD     59.8   59.9
## 4 Africa Western~ AFW            GDP per capita ~ NY.GDP.PCAP.CD    108.   113.
## 5 Angola          AGO            GDP per capita ~ NY.GDP.PCAP.CD     NA     NA
## 6 Albania         ALB            GDP per capita ~ NY.GDP.PCAP.CD     NA     NA
## # ... with 59 more variables: `1962` <dbl>, `1963` <dbl>, `1964` <dbl>,
## #   `1965` <dbl>, `1966` <dbl>, `1967` <dbl>, `1968` <dbl>, `1969` <dbl>,
## #   `1970` <dbl>, `1971` <dbl>, `1972` <dbl>, `1973` <dbl>, `1974` <dbl>,
## #   `1975` <dbl>, `1976` <dbl>, `1977` <dbl>, `1978` <dbl>, `1979` <dbl>,
## #   `1980` <dbl>, `1981` <dbl>, `1982` <dbl>, `1983` <dbl>, `1984` <dbl>,
## #   `1985` <dbl>, `1986` <dbl>, `1987` <dbl>, `1988` <dbl>, `1989` <dbl>,
## #   `1990` <dbl>, `1991` <dbl>, `1992` <dbl>, `1993` <dbl>, `1994` <dbl>, ...
```

Here we see that the GDP and internet datasets are similar, which is logical given that they come from the same source. Additionally, we see that the Netflix data is well organized but still untidy.

Tidying and Wrangling Netflix Data

We begin tiding the Netflix data by renaming some of the columns. Then, given that there is 3 separate subscription prices, we decided to mutate a new variable that holds the average price form each point. Additionally, we tidy by using pivot longer to combine Basic, Standard and Premium columns to two columns, one that contains the subscription info and one that includes the price. After that, we remove columns that are unnecessary for us.

```
# Rename columns
netflix_subscription_fee %>%
  rename(Basic = `Cost Per Month - Basic ($)`) %>%
  rename(Standard = `Cost Per Month - Standard ($)`) %>%
  rename(Premium = `Cost Per Month - Premium ($)`) %>%
  rename(Library_Size = "Total Library Size") -> tidy_netflix

# Tidying by creating "Subscription_Type" and "Price"
tidy_netflix %>%
  pivot_longer(cols = c(Basic:Premium), names_to = "Subscription_Type", values_to = "Price") ->
 tidy_netflix

# Create new "Average_Price" column
tidy_netflix %>% group_by(Country) %>% mutate(Average_Price = mean(Price)) -> tidy_netflix

# Remove unnecessary columns
tidy_netflix %>%
  select(Country, Library_Size, Subscription_Type, Price, Average_Price) %>%
  as.data.frame() -> tidy_netflix
```

Now, let’s see the outcome of the tidying.

```
# Summary of the tidy Netflix dataset
summary(tidy_netflix)
```

```
##    Country           Library_Size  Subscription_Type      Price      
##  Length:195         Min.   :2274   Length:195         Min.   : 1.97  
##  Class :character   1st Qu.:4948   Class :character   1st Qu.: 9.03  
##  Mode  :character   Median :5195   Mode  :character   Median :11.29  
##                     Mean   :5314                      Mean   :11.99  
##                     3rd Qu.:5952                      3rd Qu.:14.18  
##                     Max.   :7325                      Max.   :26.96  
##  Average_Price  
##  Min.   : 2.997  
##  1st Qu.:10.823  
##  Median :11.490  
##  Mean   :11.990  
##  3rd Qu.:13.543  
##  Max.   :20.100  
```

In order to begin to familiarize ourselves with the countries that pay the most for Netflix, let’s print out the 10 countries with the highest `Price` for the `Subscription_Type` of `Basic`.

```
# Wrangling the dataset to see the top 10 countries with highest basic subscription price
tidy_netflix %>%
  filter(Subscription_Type=="Basic") %>%
  arrange(desc(Price)) %>%
  head(10) %>%
  knitr::kable("pipe", align=c("l", "c", "c", "c"))
```

Country	Library_Size	Subscription_Type	Price	Average_Price
Liechtenstein	3048	Basic	12.88	20.10000
Switzerland	5506	Basic	12.88	20.10000
Denmark	4558	Basic	12.00	15.54667
Sweden	4361	Basic	10.90	14.93333
Israel	5713	Basic	10.56	15.05000
Belgium	4990	Basic	10.16	15.24000
France	5445	Basic	10.16	15.24000
Norway	4528	Basic	9.94	13.28667
Taiwan	5105	Basic	9.74	11.90333
Singapore	6303	Basic	9.51	12.81000

Looking at this data, we see that a lot of these countries are European. Further more, an interesting finding here is that pricing between `Subscription_Types` doesn't increase in the same proportion between all the countries. This is evident by examples such as Sweden and Israel where although we see that the `Basic` `Subscription_Type` is higher in Sweden, but the `Average_Price` is higher in Israel indicating that for at least one of the other `Subscription_Types` (but possibly both), Israel `Price` is higher.

Tidying and Wrangling Internet Access Data

Before tidying internet access we noticed that one issue we would have is dealing with all the NA values it contains. We decided to begin by investigating the percentage of NA values per year. We do this before tiding our data since we found it easier to analyze the percentage of missing values per column with the initial format.

```
# Calculate mean of NA values per column
colMeans(is.na(internet_access))
```

```
##   Country Name   Country Code Indicator Name Indicator Code           1960 
##     0.00000000     0.00000000     0.00000000     0.00000000     0.97368421 
##           1961           1962           1963           1964           1965 
##     1.00000000     1.00000000     1.00000000     1.00000000     0.97368421 
##           1966           1967           1968           1969           1970 
##     1.00000000     1.00000000     1.00000000     1.00000000     0.97368421 
##           1971           1972           1973           1974           1975 
##     1.00000000     1.00000000     1.00000000     1.00000000     0.97368421 
##           1976           1977           1978           1979           1980 
##     0.97368421     0.97368421     0.97368421     0.97368421     0.97368421 
##           1981           1982           1983           1984           1985 
##     0.97368421     0.97368421     0.97368421     0.97368421     0.97368421 
##           1986           1987           1988           1989           1990 
##     0.97368421     0.97368421     0.97368421     0.96992481     0.03759398 
##           1991           1992           1993           1994           1995 
##     0.03759398     0.03759398     0.03759398     0.03759398     0.03759398 
##           1996           1997           1998           1999           2000 
##     0.18421053     0.13909774     0.11654135     0.09022556     0.08646617 
##           2001           2002           2003           2004           2005 
##     0.07518797     0.06766917     0.09022556     0.07894737     0.07518797 
##           2006           2007           2008           2009           2010 
##     0.07894737     0.05263158     0.06390977     0.06390977     0.06390977 
##           2011           2012           2013           2014           2015 
##     0.05263158     0.07518797     0.07518797     0.08270677     0.09398496 
##           2016           2017           2018           2019           2020 
##     0.08646617     0.09022556     0.34962406     0.34586466     0.71804511
```

As can be see a lot of the earlier years are missing data, which makes sense due to the lack of popularity of the internet before the 1990s. Additionally though, a big worry point here is the amount of NA values for the year 2020. Ideally we’d have data that lines up in time as closely as possible however, given the much higher availability of observations for 2019, we decided that it would be more informative and useful to continue with the observations strictly from 2019 in this dataset.

We next move on to tidying the internet data. We first merge the individual year columns into one large `Year` column and then set all the values of the percent of population that has access to internet to a separate column `IA_Pop_percent`.

```
# Tidying the internet access data set by creating a 'year' and 'population percentage' column
internet_access %>%
  pivot_longer(cols = c('1960':'2020'), names_to = "Year", values_to = "IA_Pop_Percent") -> tidy
_internet

# Taking observations from this dataset from only the year 2019 and renaming the variable 'Count
ry Name' to 'Country.' This is done to make it easier for when we join of all three datasets tog
ether
tidy_internet %>%
  filter(Year==2019) %>%
  select(`Country Name`, `IA_Pop_Percent`)%>%
  rename(Country= 'Country Name')%>%
  na.omit() -> tidy_internet
```

To familiarize with countries with highest internet access, let’s look at the 10 countries with the highest internet access.

```
#Take a quick look at the top 10 countries with the highest percent of population with internet
 access in 2019
tidy_internet %>%
  arrange(desc(IA_Pop_Percent)) %>%
  head(10) %>%
  knitr::kable("pipe", align=c("l", "c"))
```

Country	IA_Pop_Percent
Bahrain	99.70149
Qatar	99.65280
Kuwait	99.54268
United Arab Emirates	99.15000
Iceland	99.00000
Denmark	98.04643
Norway	98.00000
Luxembourg	97.12064
Canada	96.50000
Korea, Rep.	96.15758

Similarly to the Netflix data, we see a lot of European countries here but to a smaller degree.

Tidying and Wrangling GDP Data

Once again, before tidyng the dataset, we begin by looking at the percentage of NA values per year for the same reasons we did in the Internet Percentage dataset.

```
# Finding the percent of NA values for each year in the GDP dataset
colMeans(is.na(gdp_values))
```

```
##   Country Name   Country Code Indicator Name Indicator Code           1960
##     0.00000000     0.00000000     0.00000000     0.00000000     0.51879699
##           1961           1962           1963           1964           1965
##     0.49624060     0.48496241     0.48496241     0.48496241     0.44360902
##           1966           1967           1968           1969           1970
##     0.43233083     0.42105263     0.40225564     0.40225564     0.36842105
##           1971           1972           1973           1974           1975
##     0.35714286     0.35714286     0.35714286     0.35338346     0.34586466
##           1976           1977           1978           1979           1980
##     0.34210526     0.33082707     0.33458647     0.33082707     0.28571429
##           1981           1982           1983           1984           1985
##     0.27067669     0.26691729     0.26315789     0.25939850     0.25187970
##           1986           1987           1988           1989           1990
##     0.24436090     0.22932331     0.21052632     0.21052632     0.15413534
##           1991           1992           1993           1994           1995
##     0.17293233     0.16165414     0.15037594     0.13533835     0.09774436
##           1996           1997           1998           1999           2000
##     0.09774436     0.09774436     0.09022556     0.08646617     0.07142857
##           2001           2002           2003           2004           2005
##     0.06766917     0.04887218     0.04887218     0.04887218     0.04887218
##           2006           2007           2008           2009           2010
##     0.04511278     0.04511278     0.04135338     0.04135338     0.03759398
##           2011           2012           2013           2014           2015
##     0.03007519     0.03383459     0.03007519     0.03007519     0.03383459
##           2016           2017           2018           2019           2020
##     0.03759398     0.03759398     0.03759398     0.04887218     0.09022556
```

As shown above, the GDP dataset has a very small percentage of NA values in the year 2020. We decided that it would be appropriate to use 2020 in this instance as there is an abundance of observations to draw from this particular year.

```
# Tidying the GDP data set by creating a 'year' and 'GDP' column
gdp_values %>%
  pivot_longer(cols = c('1960':'2020'), names_to = "Year", values_to = "GDP") -> tidy_GDP

# Taking observations from this dataset from only the year 2020 and renaming the variable 'Count
ry Name' to 'Country.' This is done to make it easier for when we join of all three datasets tog
ether
tidy_GDP %>%
  filter(Year==2020) %>%
  select(`Country Name`, `GDP`)%>%
  rename(Country= 'Country Name')%>%
  na.omit() -> tidy_GDP
```

Similarly to the internet dataset, we continue to familiarize ourselves with the data by printing the 10 countries with highest GDPs.

```
#Taking a quick look at the top 10 countries with the highest GDP in 2020
tidy_GDP %>%
  arrange(desc(GDP)) %>%
  head(10) %>%
  knitr::kable("pipe", align=c("l", "c"))
```

Country	GDP
Monaco	173688.19
Luxembourg	116014.60
Bermuda	107079.48
Switzerland	87097.04
Ireland	85267.76
Cayman Islands	85082.53
Norway	67329.68
United States	63593.44
North America	61502.10
Denmark	61063.32

We see similar results here as when viewing the internet dataset and even see a bit of overlap with a couple of countries.

Joining/Merging

Prior to merging we decided to take a look at each of our newly tidy dataset and see how many observations and how many countries are in each dataset. First, we find the number of rows by using the ‘nrow’ function, and then the number of unique ID’s which in this instance are countries.

```
#finding number of observations and number of countries for our tidy GDP dataset
nrow(tidy_GDP)
```

```
## [1] 242
```

```
length(unique(tidy_GDP$Country))
```

```
## [1] 242
```

```
#finding number of observations and number of countries for our tidy internet access dataset
nrow(tidy_internet)
```

```
## [1] 174
```

```
length(unique(tidy_internet$Country))
```

```
## [1] 174
```

```
#finding number of observations and number of countries for our tidy Netflix dataset
nrow(tidy_netflix)
```

```
## [1] 195
```

```
length(unique(tidy_netflix$Country))
```

```
## [1] 65
```

Here we see that every observation in the GDP and both the internet access datasets contain a unique country. Where as for the Netflix dataset, we have three times the rows as countries. This is reasonable as each country has three different plans: Basic, Standard and Premium. 3 (Netflix plans) multiplied by 65 (countries) is a total of 195 (observations). When it come down to the number of countries represented, we see the GDP dataset containing the most with 242 countries, followed by the internet access dataset with 174 countries, and the lowest being the Netflix dataset with 65 countries.

Next, we join our data through the commonly shared variable `Country` . First we join the GDP and internet access datasets to create `world_info` . Then we will merge this newly created dataset with the Netflix dataset to create our main dataset called `netflix_prices_with_context` .

```
# Combining the GDP and internet access data sets into one
world_info <- inner_join(tidy_GDP, tidy_internet)
```

```
## Joining, by = "Country"
```

```
# Combining the world info data set with the Netflix data set as one
netflix_prices_with_context <- inner_join(world_info, tidy_netflix)
```

```
## Joining, by = "Country"
```

Now, lets start looking into the data that we lost while joining. Let’s begin by checking the `world_info` join which combined our GDP and internet datasets.

```
nrow(world_info)
```

```
## [1] 174
```

Looking at the number of rows in the new dataset world_info, we see that there is a total of 174 observations. This means that most likely all the countries within the internet access dataset found a match within the GDP data. Next, let’s see the 10 countries with the highest GDP that existed in the GDP dataset but not the internet dataset.

```
# Top 10 countries with highest GDP that was not found in the internet access dataset
anti_join(tidy_GDP, tidy_internet) %>%
  arrange(desc(GDP)) %>%
  head(10)
```

```
## Joining, by = "Country"
```

```
## # A tibble: 10 x 2
##    Country                       GDP
##    <chr>                       <dbl>
##  1 Monaco                    173688.
##  2 Bermuda                   107079.
##  3 Cayman Islands             85083.
##  4 Australia                  51693.
##  5 New Zealand                41441.
##  6 Guam                       34624.
##  7 Bahamas, The               25194.
##  8 Turks and Caicos Islands  23880.
##  9 St. Kitts and Nevis       18438.
## 10 Curacao                    16746.
```

Among the list we find a lot of recognizable countries that could have been valuable data points.

Next, we’ll look at the data lost when joining the world_info to the Netflix dataset

```
# Investigate data lost with netflix_prices_with_context join
nrow(netflix_prices_with_context)
```

```
## [1] 141
```

```
length(unique(netflix_prices_with_context$Country))
```

```
## [1] 47
```

Here we see a larger decrease in countries which largely makes sense since in the Netflix data we only had 65 datapoints. This means that here we lost 33 world_info observations (174-141) which is 18% of the original data and 54 Netflix observation (195-174) which is 28% of the data.

Let’s look at the 10 countries with the largest GDPs that were lost with the join again.

```
# Finding top 10 countries with highest GDP that was lost merging the world info dataset and tod
y Netflix dataset
anti_join(world_info, tidy_netflix) %>%
  arrange(desc(GDP)) %>% head(10)
```

```
## Joining, by = "Country"
## # A tibble: 10 x 3
##    Country                      GDP IA_Pop_Percent
##    <chr>                      <dbl>          <dbl>
##  1 Luxembourg               116015.           97.1
##  2 North America             61502.           90.2
##  3 Qatar                     50124.           99.7
##  4 Hong Kong SAR, China      46324.           91.7
##  5 Post-demographic dividend  44475.          87.9
##  6 High income               44003.           89.1
##  7 Macao SAR, China          39403.           86.5
##  8 OECD members              38219.           85.1
##  9 Euro area                 37968.           84.8
## 10 United Arab Emirates      36285.           99.1
```

In this anti_join we see a lot of useful information. It seems a lot of the data points for `Country` that were lost weren't in fact countries, thus most of the data filtered through the join makes sense.

For thoroughness we decided to also look at the data overlap between GDP and Netflix data, to see how many of the keys lost with the world_info join could have found matches with the Netflix data.

```
# Anti join the dataset tidy GDP and tidy Netflix to find the number of missing values lost when merging the datasets together
nrow(anti_join(tidy_GDP, tidy_netflix))
```

```
## Joining, by = "Country"
```

```
## [1] 187
```

Here we see that we lost 187 observation when combining the two tidy datasets together.

Note: We don't compare the internet access dataset overlap with the Netflix one because the results would be the same as in our world_info comparison to the Netflix data. Due to none of internet access datasets unique ID's were lost when creating world_info, it would be repetitive to join tidy Netflix and tidy internet.

Let's go ahead and take a look into the summary statistics from a few categorical and numeric variables. We decided to create a new variable called 'interesting_case' where we set the conditions of if the countries GDP is lower then the global average but the average subscription cost of Netflix is higher then the global average it will read 'TRUE' and otherwise 'FALSE'. We looked at the proportion of 'TRUE' values in this categorical variable. The next variable we looked at is the summarized number of countries present this dataset. As for the numeric variables we looked at specifically GDP, population percent of internet access and library size and found the global average for each variable.

Wranging the Merged Datasets

```
# Setting values for the global average GDP and global average Netflix price found in the merged datasets
avg_gdp <- mean(netflix_prices_with_context$GDP)
avg_netflix_price <- mean(netflix_prices_with_context$Average_Price)

# Finding the summary statistics for two categorical variables: interesting cases and Country.
netflix_prices_with_context %>%
  mutate(intresting_case = GDP < avg_gdp & Average_Price > avg_netflix_price)%>%
  summarize(mean(intresting_case))
```

```
## # A tibble: 1 x 1
##   `mean(intresting_case)`
##                     <dbl>
## 1                   0.106
```

```
netflix_prices_with_context %>%
  mutate(intresting_case = GDP < avg_gdp & Average_Price > avg_netflix_price)%>%
  filter(intresting_case == TRUE) %>%
  group_by(Country) %>%
  summarize(mean(Average_Price))
```

```
## # A tibble: 5 x 2
##   Country    `mean(Average_Price)`
##   <chr>                      <dbl>
## 1 Costa Rica                  12.7
## 2 Greece                      12.4
## 3 Portugal                    13.5
## 4 Spain                       14.7
## 5 Uruguay                     12.7
```

```
netflix_prices_with_context %>%
  summarize(n_distinct(Country))
```

```
## # A tibble: 1 x 1
##   `n_distinct(Country)`
##                   <int>
## 1                    47
```

```
# Finding the summary statistics for three numeric variables: GDP, internet access percentage and library size.
netflix_prices_with_context %>%
  summarize(mean(GDP))
```

```
## # A tibble: 1 x 1
##   `mean(GDP)`
##         <dbl>
## 1      28179.
```

```
netflix_prices_with_context %>%
  summarize(mean(IA_Pop_Percent))
```

```
## # A tibble: 1 x 1
##   `mean(IA_Pop_Percent)`
##                    <dbl>
## 1                   78.7
```

```
netflix_prices_with_context %>%
  summarize(mean(Library_Size))
```

```
## # A tibble: 1 x 1
##   `mean(Library_Size)`
##                  <dbl>
## 1                5369.
```

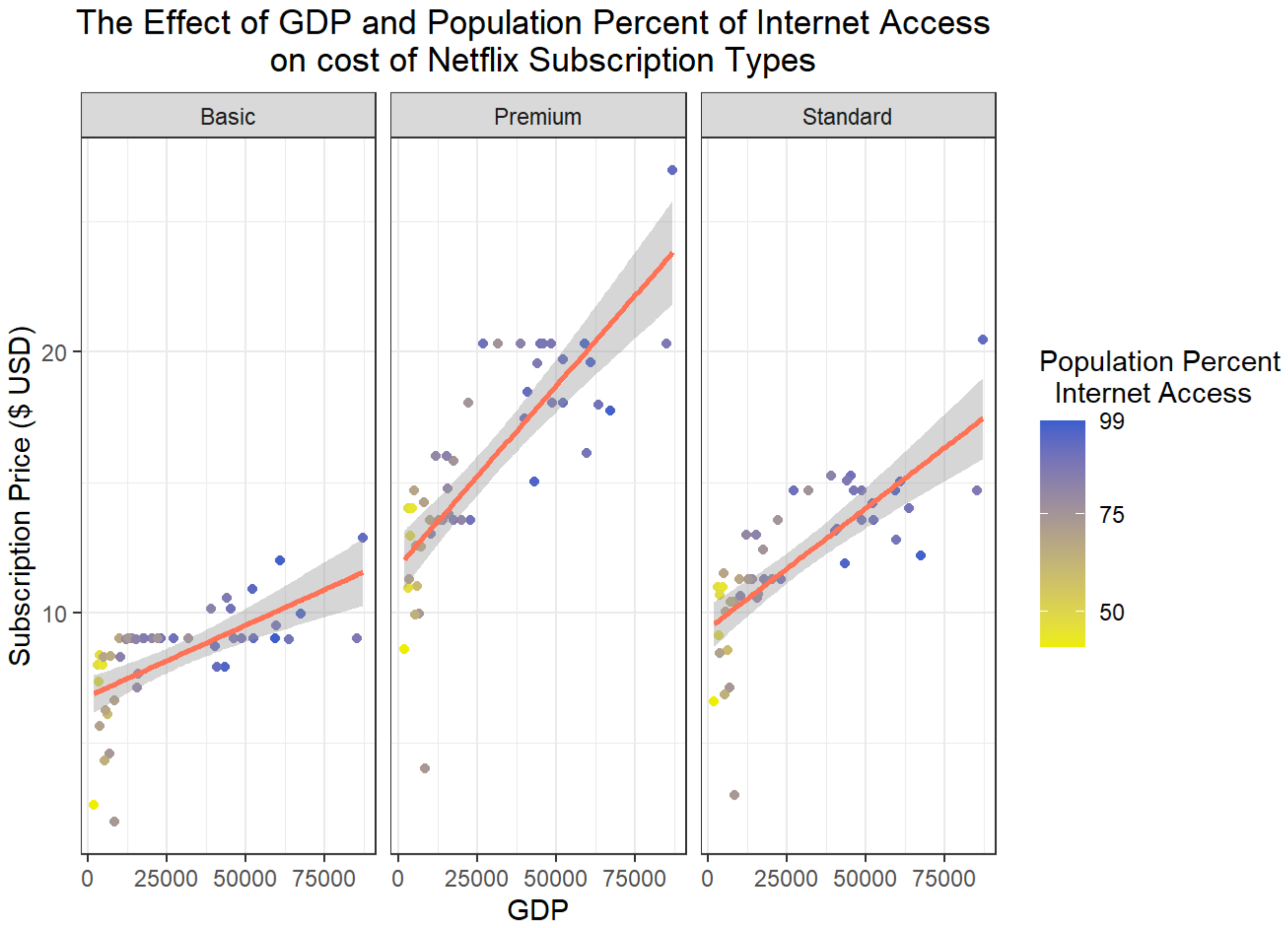
Regarding if the countries fit the interesting case only 10.64% fit this criteria which in total is 5 countries: Costa Rica, Greece, Portugal, Spain, Uruguay. There are a total of 47 different countries in the Netflix prices with context dataset. The global average GDP for this dataset is 28179.32 USD. The global average for population with internet access in this dataset is 78.7%. Finally the last summary statistic we have is the dataset’s global average library size which is 5368 films and TV shows. The reason why we did not report the global library size average being 5368.6 films and TV shows is because in this case we do not count for proportions of films or TV shows as this is a discrete value.

Visualization

Now, we will create visualizations in order to explore trends within the dataset and in order to evaluate our hypothesis.

```
# Creating scatterplot separated by type of Netflix subscription to analyze the relationship of subscription price, GDP and population percent with internet access.
ggplot(netflix_prices_with_context, aes(x = GDP, y = Price)) +
  geom_point(aes(color=IA_Pop_Percent)) +
  geom_smooth(method=lm, color= "coral1")+
  scale_color_gradient(low = "yellow2", high = "royalblue3", breaks= c(50,75,99)) +
  facet_wrap(~Subscription_Type)+
  theme_bw()+
  labs(
  title = "The Effect of GDP and Population Percent of Internet Access
  on cost of Netflix Subscription Types ",
  x = "GDP",
  y = "Subscription Price ($ USD)",
  color= "Population Percent
  Internet Access")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```

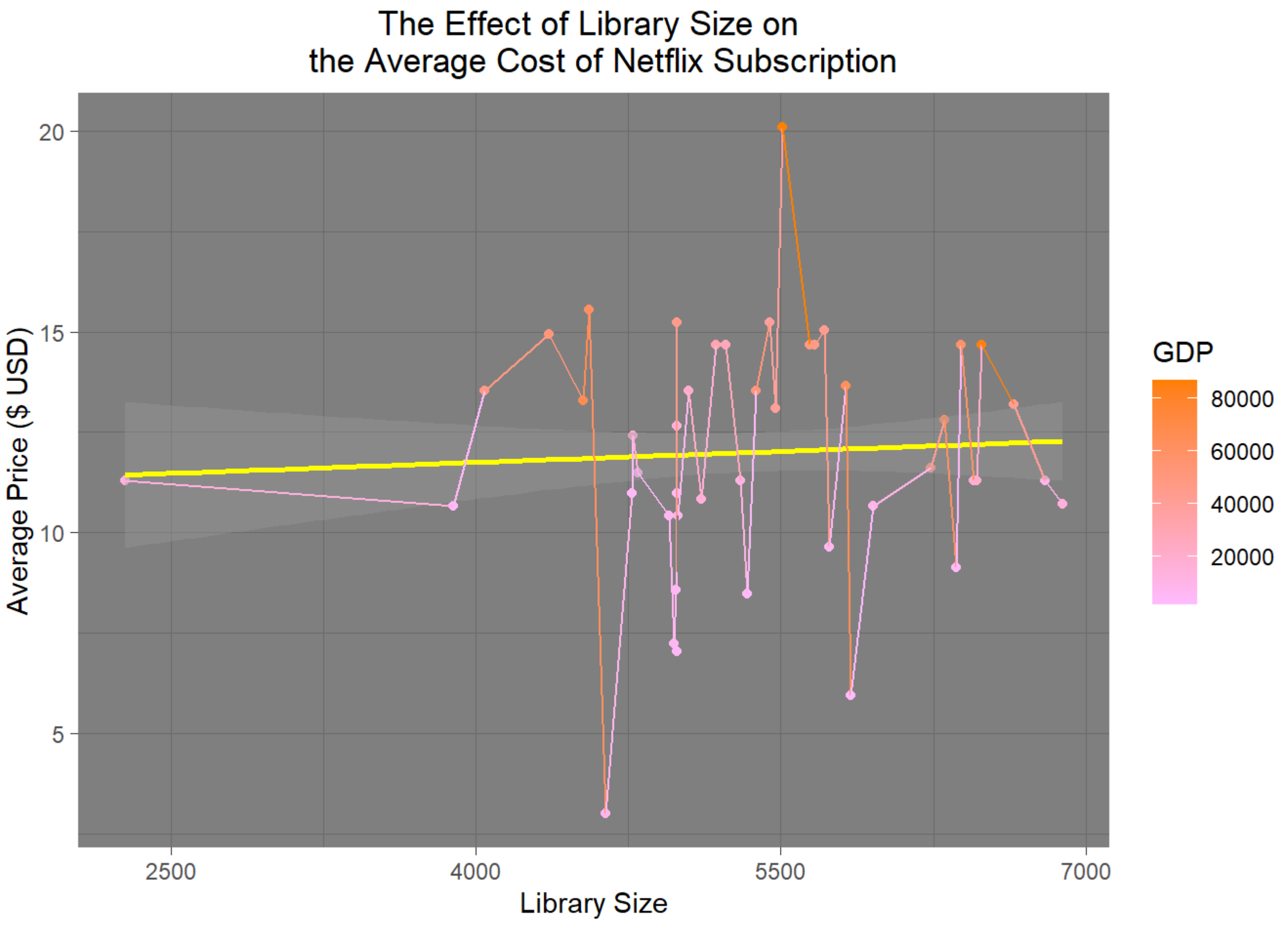


In this figure we take a look into the relationship between GDP and it’s effects on subscription prices and the population percent of internet access effect on subscription prices. Let us analyze the relationship between GDP and subscription price! As the GDP increases so does the price showing a positive correlation between the two variables. This can be expected as higher GDP values indicate the country is doing economically well, so consequently prices for a popular streaming services are increased. It can also be noted that as GDP increases, the color gradient of internet access continues to darken indicating countries with higher GDPs also have populations more inclined to having internet access. This also indicates that countries with higher percent of population to internet access will also have higher prices for Netflix subscriptions.

In the following figure we are going to be looking at the effect of library size on the average subscription price of Netflix.

```
# Creating a scatterplot to analyze the relationship between average subscription cost, GDP and
 library size
ggplot(netflix_prices_with_context, aes(x = Library_Size, y = Average_Price, color=GDP)) +
  geom_point() +
  geom_smooth(method=lm, color= "yellow1")+
  theme_dark()+
  scale_x_continuous(breaks=c(2500,4000,5500,7000))+
  scale_color_gradient(low="plum1", high="darkorange1")+
  geom_line(stat = "summary", fun = "mean")+
  labs(
  title = "The Effect of Library Size on
  the Average Cost of Netflix Subscription",
  x = "Library Size",
  y = "Average Price ($ USD)",
  color= "GDP")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



The regression line displayed has a very small slope indicating that the library size does not necessarily effect the average subscription price. We predicted that the larger the library size it would increase the average subscription price. However another variable, GDP, was included in the legend and as the price increases one can see that the color follows a trend of going from a purple (a lower GDP) to an orange (a higher GDP) and this trend is logical as the better the economy a country has, the more expensive the cost of subscription is expected to be.

The following plot examines the data available to us by mapping each original source to a world map.

```
# Use rworldmap for world plotting, command to install:
# install.packages('rworldmap',dependencies=TRUE)
library(rworldmap)
```

```
## Warning: package 'rworldmap' was built under R version 4.1.3
```

```
## Loading required package: sp
```

```
## Warning: package 'sp' was built under R version 4.1.3
```

```
## ### Welcome to rworldmap ###
```

```
## For a short introduction type :   vignette('rworldmap')
```

```
# Set up R so that the next for plots get concatenated to a single 2 by 2 plot
par(mfrow=c(2,2))

# Prepare GDP data and create map
gdp_data <- joinCountryData2Map(tidy_GDP, joinCode = "NAME", nameJoinColumn = "Country")
```

```
## 188 codes from your data successfully matched countries in the map
## 54 codes from your data failed to match with a country code in the map
## 55 codes from the map weren't represented in your data
```

```
gdp_map <- mapCountryData(gdp_data, nameColumnToPlot="GDP", colourPalette = "terrain", addLegend
=TRUE, mapTitle="Gross Domestic Product")

# Prepare internet data and create map
internet_data <- joinCountryData2Map(tidy_internet, joinCode = "NAME", nameJoinColumn = "Countr
y")
```

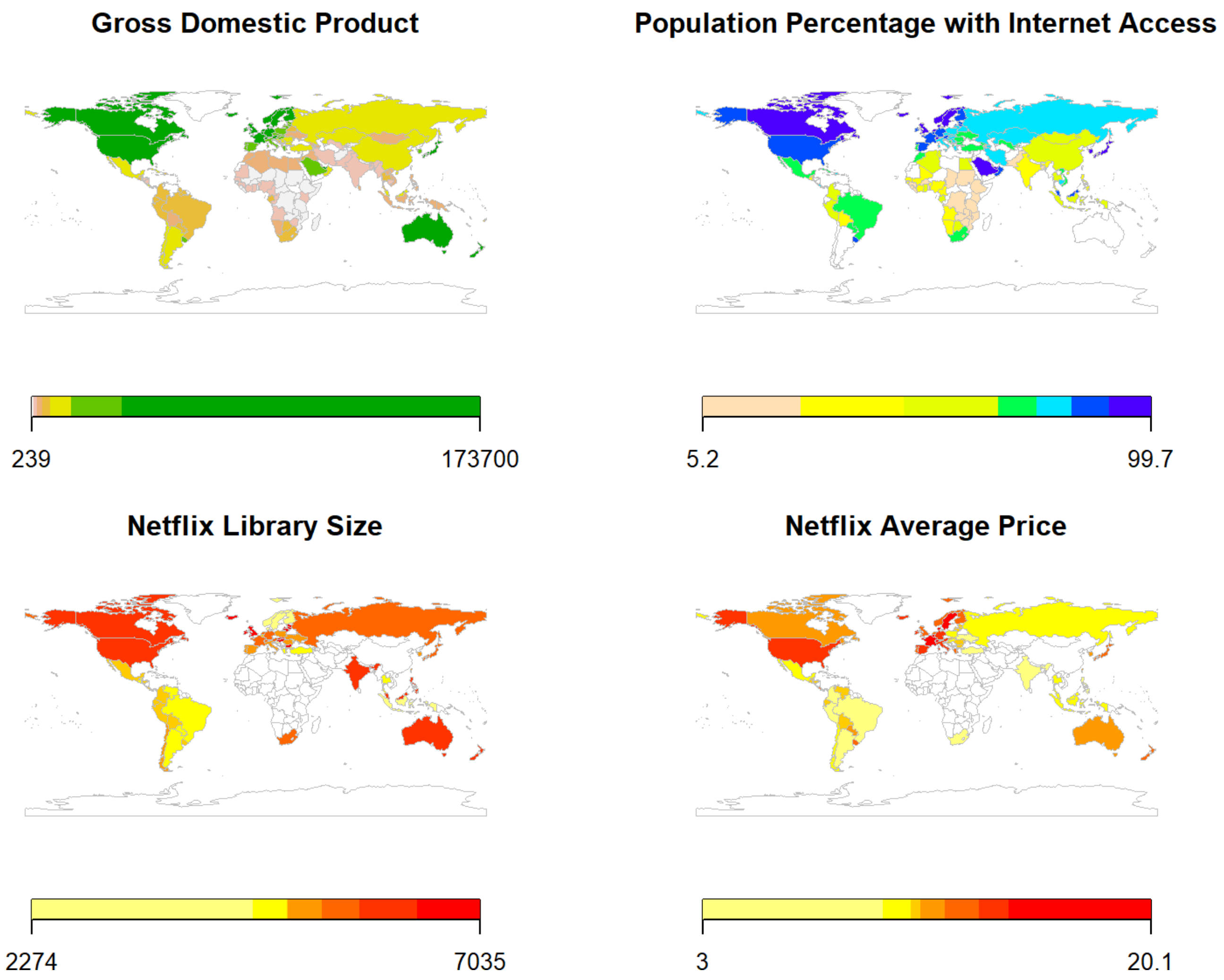
```
## 126 codes from your data successfully matched countries in the map
## 48 codes from your data failed to match with a country code in the map
## 117 codes from the map weren't represented in your data
```

```
internet_map <- mapCountryData(internet_data, nameColumnToPlot="IA_Pop_Percent", colourPalette = "topo", addLegend=TRUE, mapTitle="Population Percentage with Internet Access" )

# Prepare Netflix data and create Average Price and Library Size map
netflix_data <- joinCountryData2Map(tidy_netflix, joinCode = "NAME", nameJoinColumn = "Country")
```

```
## 189 codes from your data successfully matched countries in the map
## 6 codes from your data failed to match with a country code in the map
## 180 codes from the map weren't represented in your data
```

```
netflix_map_library <- mapCountryData(netflix_data, nameColumnToPlot="Library_Size", addLegend=TRUE, mapTitle="Netflix Library Size")
netflix_map_avg_price <- mapCountryData(netflix_data, nameColumnToPlot="Average_Price", addLegend=TRUE, mapTitle="Netflix Average Price" )
```



This plot provides a lot of intresting insights. Firstly, we see a lack of availabilty for data in Africa for netflix, which could be due to the lack of involvement in Netflix’s part or missing data. Additionally, we see a lot situations where countries with high GDP and Internet Access are usually surrounded by other countries with high GDP and internet access as well.

Acknowledgements

Each member contributed equally to the project. Karolina wrote the title and introduction, contributed to tidying, contributed to the wrangling and created the two scatterplots for the visualization. Daniel did all the joining and merging, contributed to the tidying, contributed to the wrangling and created the cool maps in the visualization. Special shout out to Daniel as he found super creative ways to help this project be easier done and understood!

```
##           sysname           release           version          nodename
##         "Windows"          "10 x64"     "build 19043" "DESKTOP-FBRIDEV"
##           machine             login              user    effective_user
##          "x86-64"          "Daniel"          "Daniel"          "Daniel"
```