

NETFLIX DATA ANALYSIS

Neil de la Fuente Rodríguez, neildlf@gmail.com (mailto:neildlf@gmail.com)

01/03/2022 || source: <https://www.kaggle.com/prasertk/netflix-subscription-price-in-different-countries> (<https://www.kaggle.com/prasertk/netflix-subscription-price-in-different-countries>)

- **Welcome to Task 1!**
- UNIVARIATE DESCRIPTIVE STATISTICS
- BIVARIATE DESCRIPTIVE STATISTICS
- CONCLUSIONS

```
library(psych)
```

Welcome to Task 1!

Today we'll be working on univariate and bivariate descriptive statistics.

UNIVARIATE DESCRIPTIVE STATISTICS

In order to make this work I used a dataset about Netflix, its amount of films, TV shows and prices all over the world. The data contained is about the Country_code, Country, Total Library Size, No. of TV Shows, No. of Movies, Cost Per Month - Basic, Cost Per Month - Standard and Cost Per Month - Premium. This dataset has qualitative and quantitative data.

So let's start by opening our datafile as a dataframe, I'll call it netflix in order to simplify the title.

```
netflix <- read.csv("netflix_data.csv")  
head(netflix,10) #preview the dataframe, just the first 5 elements.
```

```
## Country_code Country Total.Library.Size No..of.TV.Shows No..of.Movies
## 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
## 7 bg Bulgaria 6797 4819 1978
## 8 ca Canada 6239 4311 1928
## 9 cl Chile 4994 3156 1838
## 10 co Colombia 4991 3156 1835
## Cost.Per.Month...Basic.... Cost.Per.Month...Standard....
## 1 3.74 6.30
## 2 7.84 12.12
## 3 9.03 14.67
## 4 10.16 15.24
## 5 7.99 10.99
## 6 4.61 7.11
## 7 9.03 11.29
## 8 7.91 11.87
## 9 7.07 9.91
## 10 4.31 6.86
## Cost.Per.Month...Premium....
## 1 9.26
## 2 16.39
## 3 20.32
## 4 20.32
## 5 13.99
## 6 9.96
## 7 13.54
## 8 15.03
## 9 12.74
## 10 9.93
```

next step will be to calculate the tables of absolute, relative, and cumulative frequencies.

```
abs_freq <- table(netflix$No..of.TV.Shows)
abs_freq
```

```
##
## 1675 1712 1937 2449 2473 2638 2883 2930 2955 2973 2977 2978 3134 3152 3154 3155
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 5 4
## 3156 3158 3162 3261 3334 3344 3374 3419 3512 3536 3545 3565 3604 3619 3624 3650
## 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## 3654 3686 3718 3779 3806 3814 3826 3832 4003 4050 4079 4109 4154 4311 4426 4479
## 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
## 4486 4490 4515 4551 4802 4819 5055 5234
## 1 1 1 1 1 1 1 1
```

```
rel_freq <- prop.table(abs_freq)
rel_freq
```

```
##
##      1675      1712      1937      2449      2473      2638      2883
## 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462
##      2930      2955      2973      2977      2978      3134      3152
## 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462
##      3154      3155      3156      3158      3162      3261      3334
## 0.07692308 0.06153846 0.03076923 0.01538462 0.01538462 0.01538462 0.01538462
##      3344      3374      3419      3512      3536      3545      3565
## 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462
##      3604      3619      3624      3650      3654      3686      3718
## 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462
##      3779      3806      3814      3826      3832      4003      4050
## 0.03076923 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462
##      4079      4109      4154      4311      4426      4479      4486
## 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462
##      4490      4515      4551      4802      4819      5055      5234
## 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462 0.01538462
```

```
cumsum(abs_freq)
```

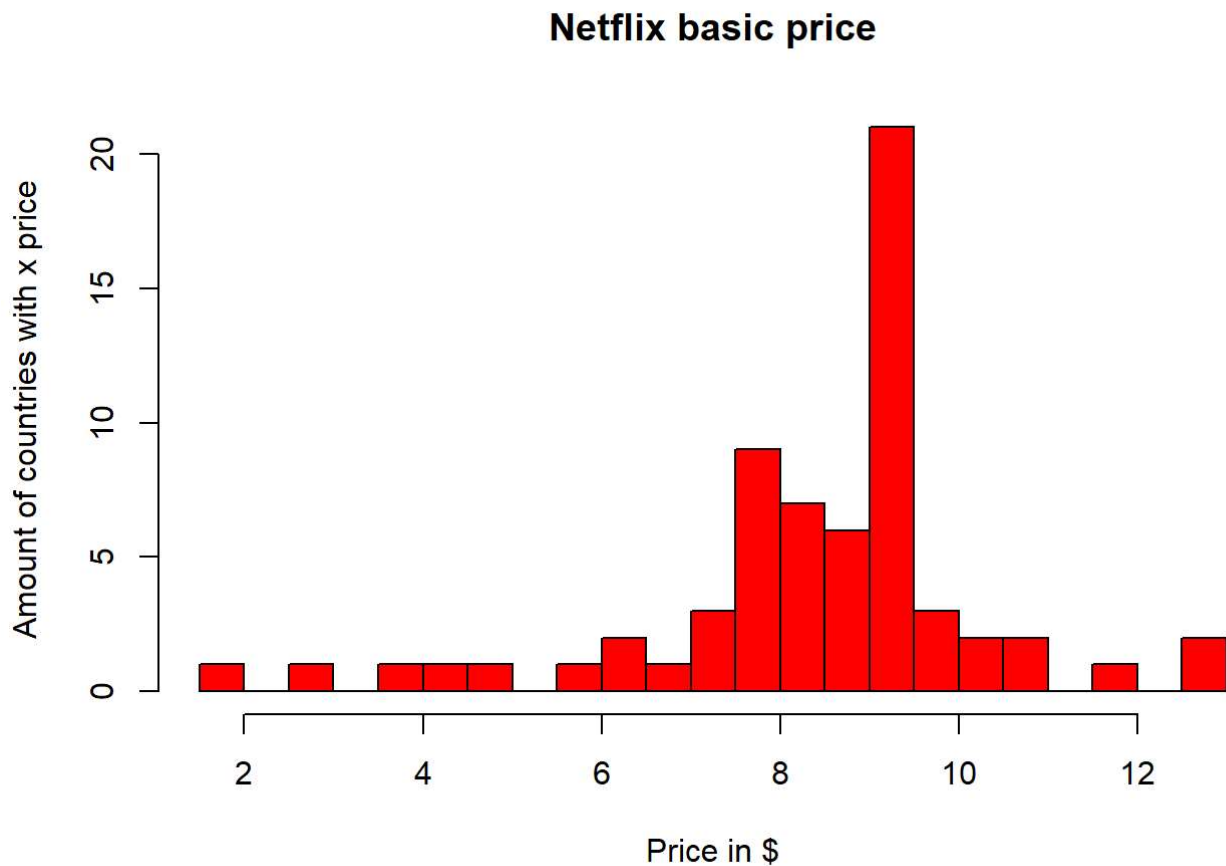
```
## 1675 1712 1937 2449 2473 2638 2883 2930 2955 2973 2977 2978 3134 3152 3154 3155
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   19   23
## 3156 3158 3162 3261 3334 3344 3374 3419 3512 3536 3545 3565 3604 3619 3624 3650
##   25   26   27   28   29   30   31   32   33   34   35   36   37   38   39   40
## 3654 3686 3718 3779 3806 3814 3826 3832 4003 4050 4079 4109 4154 4311 4426 4479
##   41   42   43   45   46   47   48   49   50   51   52   53   54   55   56   57
## 4486 4490 4515 4551 4802 4819 5055 5234
##   58   59   60   61   62   63   64   65
```

```
cumsum(rel_freq)
```

```
##      1675      1712      1937      2449      2473      2638      2883
## 0.01538462 0.03076923 0.04615385 0.06153846 0.07692308 0.09230769 0.10769231
##      2930      2955      2973      2977      2978      3134      3152
## 0.12307692 0.13846154 0.15384615 0.16923077 0.18461538 0.20000000 0.21538462
##      3154      3155      3156      3158      3162      3261      3334
## 0.29230769 0.35384615 0.38461538 0.40000000 0.41538462 0.43076923 0.44615385
##      3344      3374      3419      3512      3536      3545      3565
## 0.46153846 0.47692308 0.49230769 0.50769231 0.52307692 0.53846154 0.55384615
##      3604      3619      3624      3650      3654      3686      3718
## 0.56923077 0.58461538 0.60000000 0.61538462 0.63076923 0.64615385 0.66153846
##      3779      3806      3814      3826      3832      4003      4050
## 0.69230769 0.70769231 0.72307692 0.73846154 0.75384615 0.76923077 0.78461538
##      4079      4109      4154      4311      4426      4479      4486
## 0.80000000 0.81538462 0.83076923 0.84615385 0.86153846 0.87692308 0.89230769
##      4490      4515      4551      4802      4819      5055      5234
## 0.90769231 0.92307692 0.93846154 0.95384615 0.96923077 0.98461538 1.00000000
```

Once having the tables done, I'll go for step 2 on Univariate Descriptive Statistics, which will be drawing a histogram. In this case the histogram will be about the netflix basic subscription price in chosen countries. Also the aim is answering to this question: "What is the default number of classes?"

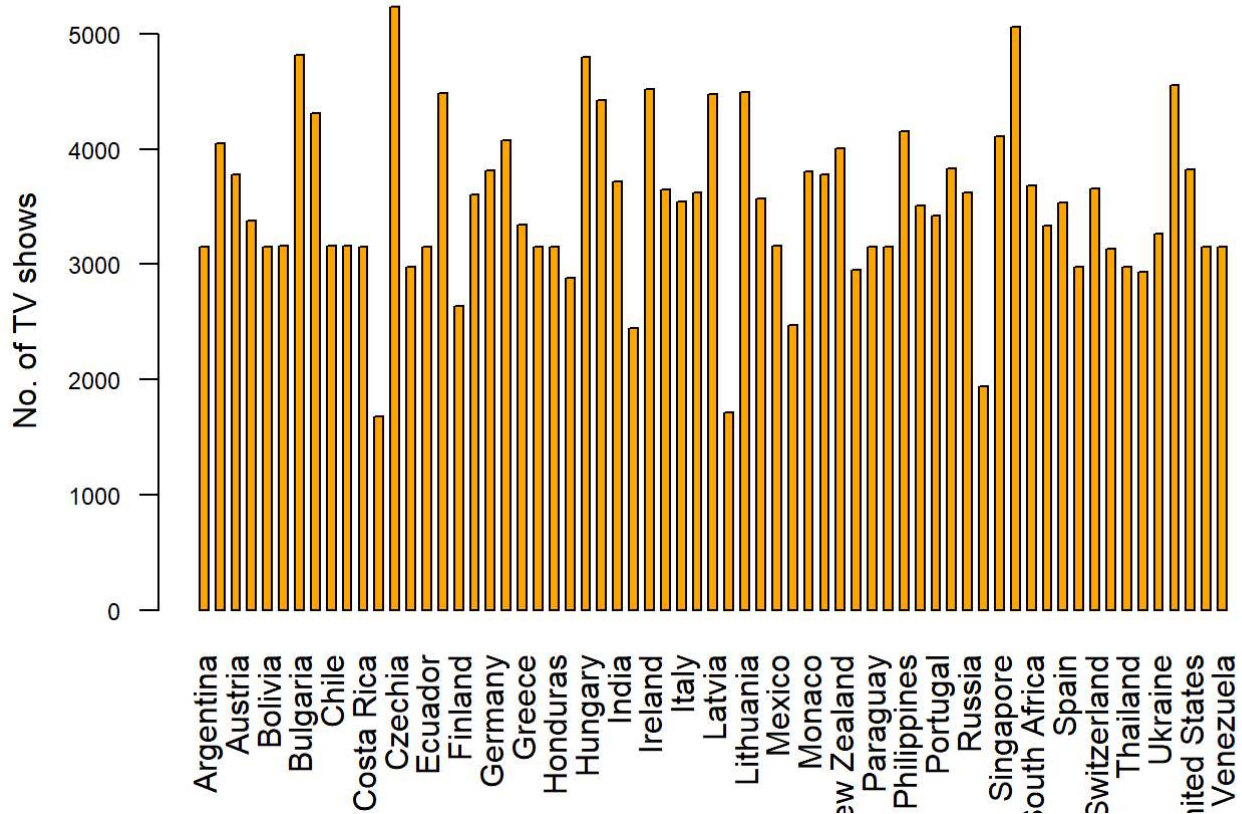
```
hist(netflix$Cost.Per.Month...Basic.... , breaks = 20 , col = "red", border = "black", main =
"Netflix basic price", xlab = "Price in $", ylab = "Amount of countries with x price" )
```



The default number of classes is 8, when we don't specify the amount of breaks.

Now let's draw a **barplot** using the normal **barplot** from R, for you to see how useful it is. In this time we'll choose the amount of movies:

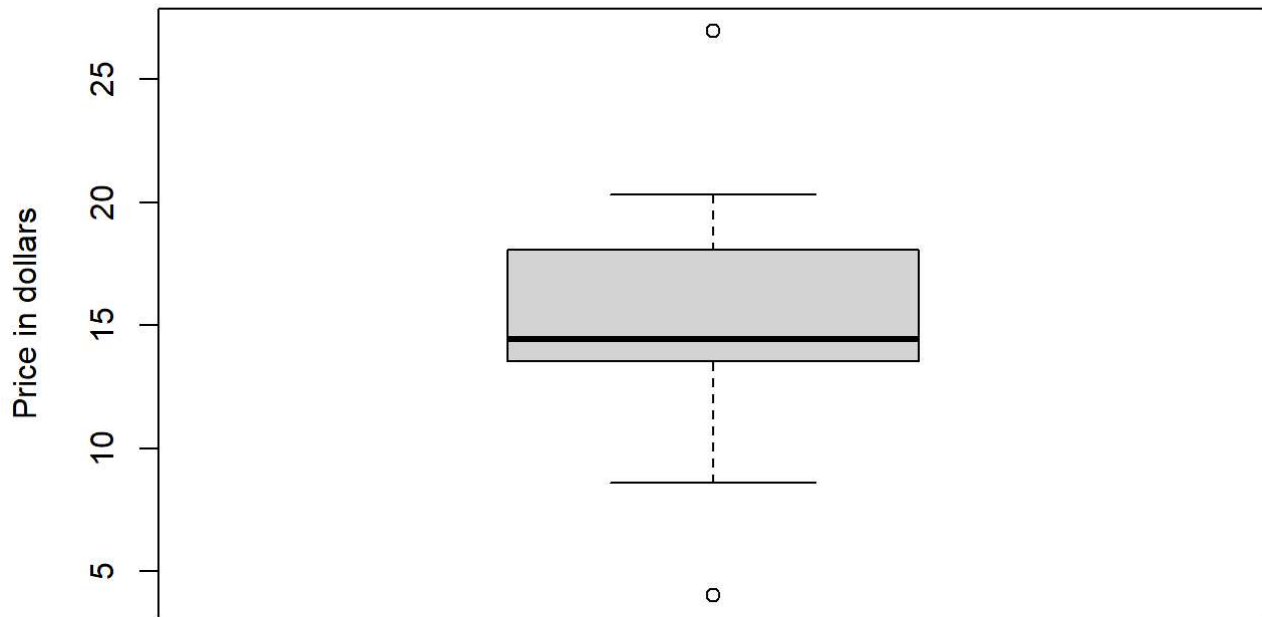
```
barplot(
  height = netflix$No..of.TV.Shows, names = netflix$Country, srt = 90, las = 2, col = "orange"
, ylab = "No. of TV shows", space = 0.8, cex.axis = 0.75)
```



As you can see the barplot measures both quantitative and qualitative data, countries(qualitative) and No. of TV shows (quantitative).

The next step consists in drawing a **boxplot**, which will also show the median, the quartiles and the outliers.

```
boxplot(netflix$Cost.Per.Month...Premium..., ylab = "Price in dollars")
```



For the next step (5), we must calculate the center measurements: **mean, median, mode**. We'll do it out of the total library sizes.

```
library(lsr)
mean(netflix$Total.Library.Size)
```

```
## [1] 5314.415
```

```
median(netflix$Total.Library.Size)
```

```
## [1] 5195
```

```
modeOf(netflix$Total.Library.Size)
```

```
## [1] 4991 4989 4797
```

For the next step (6), we must calculate the dispersion measurements: **variance, standard deviation**, we'll use the amount of TV shows.

```
var(netflix$No..of.TV.Shows)
```

```
## [1] 522744.3
```

```
sd(netflix$No..of.TV.Shows)
```

```
## [1] 723.0106
```

For the next step (7), we must calculate the position measurements: **percentiles, quartiles**, we'll do it with the price of the Premium subscription.

```
quartiles = quantile(netflix$Cost.Per.Month...Premium.... )
quartiles
```

```
##      0%    25%    50%    75%   100%
##  4.02 13.54 14.45 18.06 26.96
```

```
percentiles = quantile(netflix$Cost.Per.Month...Premium.... , prob = seq(0, 1, by = 0.1))
percentiles
```

```
##      0%    10%    20%    30%    40%    50%    60%    70%    80%    90%   100%
##  4.020 11.122 13.008 13.540 13.990 14.450 16.038 18.046 19.620 20.320 26.960
```

For the next step (8), we must calculate the position measurements: **Quartiles and interquartile range**.

```
quantile(netflix$No..of.Movies, c(.25,.5,.75))
```

```
##   25%   50%   75%
## 1628 1841 1980
```

```
IQR(netflix$No..of.Movies)
```

```
## [1] 352
```

For the next step (9), we must Calculate the shape measurements: Skewness Coefficient, kurtosis coefficient.

```
library(e1071)
skewness(netflix$Cost.Per.Month...Basic....)
```

```
## [1] -0.9239394
```

```
kurtosis(netflix$Cost.Per.Month...Basic....)
```

```
## [1] 2.349559
```

BIVARIATE DESCRIPTIVE STATISTICS

Let's calculate the **covariance** and the **linear correlation coefficient** between two variables (amount of content and price of the standard subscription) of the netflix dataset.

```
cov(netflix$Total.Library.Size, netflix$Cost.Per.Month...Standard....)
```

```
## [1] -154.9805
```

```
cor(netflix$Total.Library.Size, netflix$Cost.Per.Month...Standard....)
```

```
## [1] -0.05519988
```

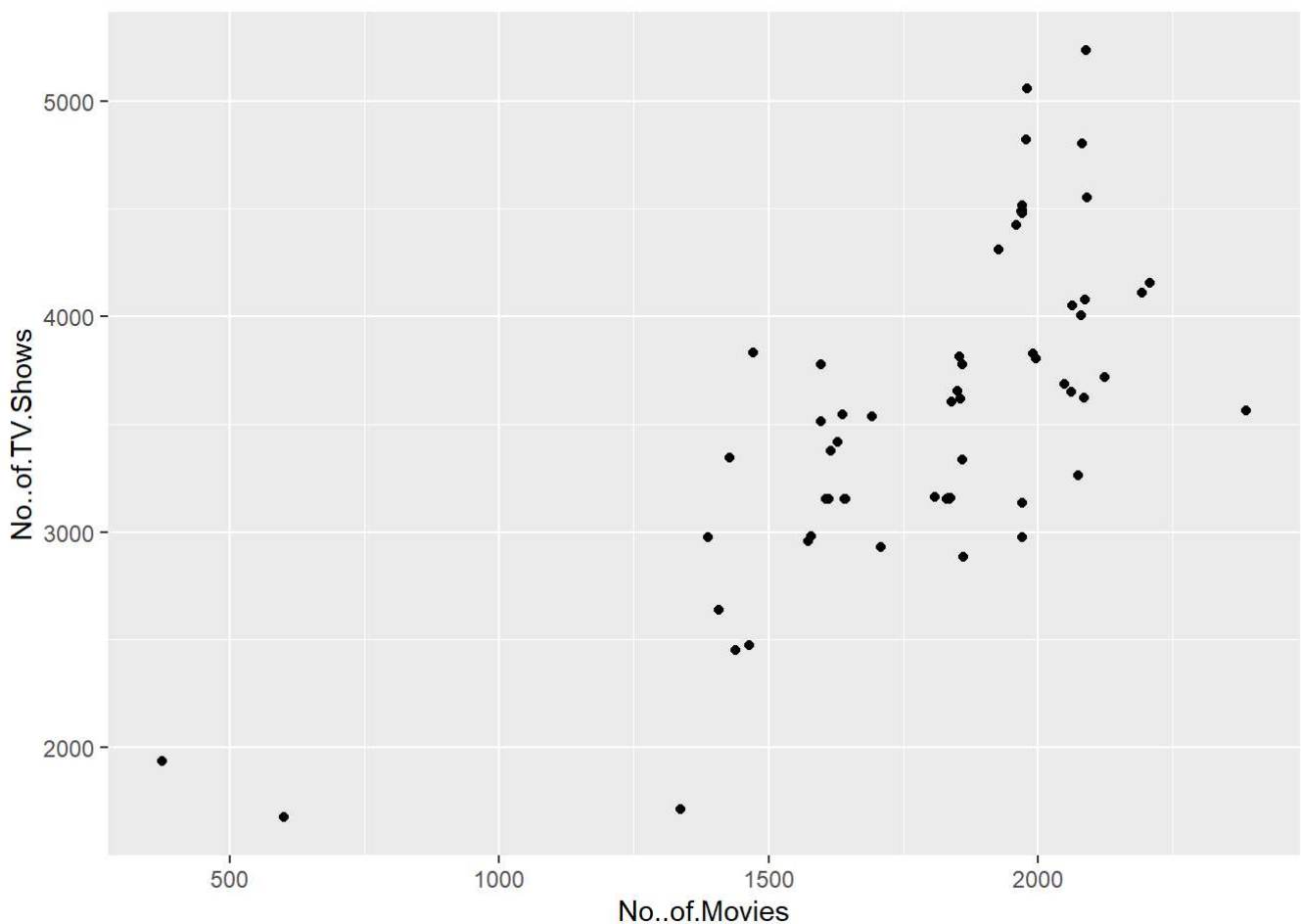
Now I'll represent the data in a **scatter plot**.

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
## %+%, alpha
```

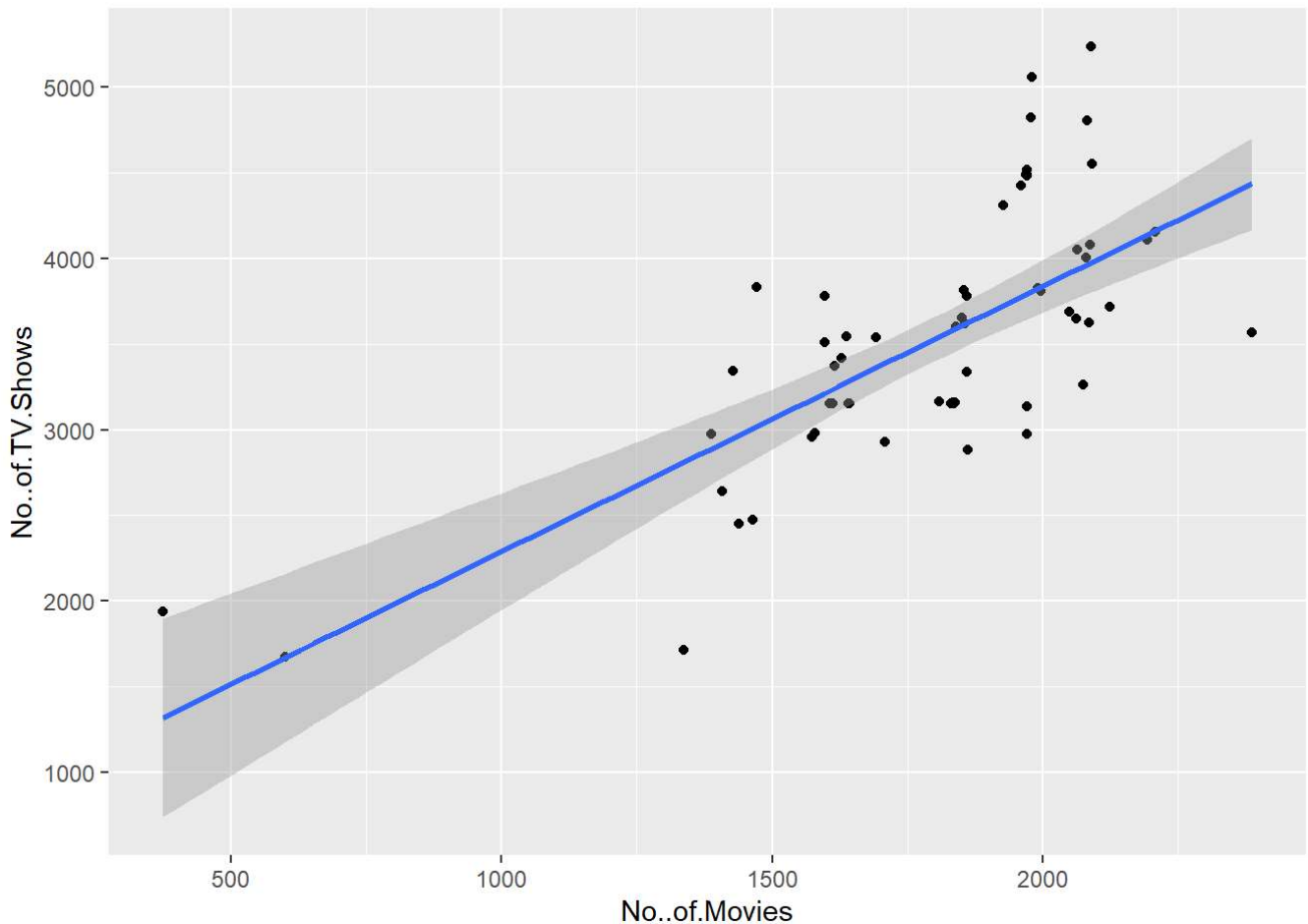
```
ggplot(data = netflix, mapping = aes(x = No..of.Movies, y =No..of.TV.Shows))+
  geom_point()
```



Next step, Calculate the **regression line** of two variables, also add it to the graph found in the previous section and say what the **variability of the data** is explained by the regression model.


```
ggplot(data = netflix, mapping = aes(x = No..of.Movies, y = No..of.TV.Shows)) +
  geom_point() +
  stat_smooth(method = lm)
```

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



```
summary(lm(netflix$No..of.TV.Shows~netflix$No..of.Movies))
```

```
##
## Call:
## lm(formula = netflix$No..of.TV.Shows ~ netflix$No..of.Movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1096.66  -422.08   -26.06   269.47  1258.16
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    743.2897    362.7724     2.049   0.0446 *
## netflix$No..of.Movies  1.5459     0.1988    7.775 8.82e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 520.6 on 63 degrees of freedom
## Multiple R-squared:  0.4897, Adjusted R-squared:  0.4816
## F-statistic: 60.46 on 1 and 63 DF, p-value: 8.818e-11
```

The variability of this regression model is explained by the Multiple R-squared: 0.4897, which means that we would not use this because is not representative as a correlation, normally it should have at least a 70%

Now, I'm going to calculate the **adjusted values and residuals** from the first 5 data of the regression model.

```
df <- data.frame(x = head(netflix$No..of.Movies,5), y = head(netflix$No..of.TVShows,5))
df$adjusted <- 743.2897 + 1.5459* df$x
df$residuals <- df$adjusted - df$y
```

```
df
```

```
##      x    y adjusted residuals
## 1 1606 3154 3226.005    72.0051
## 2 2064 4050 3934.027   -115.9727
## 3 1861 3779 3620.210   -158.7904
## 4 1616 3374 3241.464   -132.5359
## 5 1836 3155 3581.562    426.5621
```

Let's calculate a **contingency table** over the categorical variables.

As the only two categorical variables I have are Countries names and countries codes, it may look a bit strange.

```
table(netflix$Country_code, netflix$Country)
```

##		Argentina	Australia	Austria	Belgium	Bolivia	Brazil	Bulgaria	Canada	Chile
##	ar	1	0	0	0	0	0	0	0	0
##	at	0	0	1	0	0	0	0	0	0
##	au	0	1	0	0	0	0	0	0	0
##	be	0	0	0	1	0	0	0	0	0
##	bg	0	0	0	0	0	0	1	0	0
##	bo	0	0	0	0	1	0	0	0	0
##	br	0	0	0	0	0	1	0	0	0
##	ca	0	0	0	0	0	0	0	1	0
##	ch	0	0	0	0	0	0	0	0	0
##	cl	0	0	0	0	0	0	0	0	1
##	co	0	0	0	0	0	0	0	0	0
##	cr	0	0	0	0	0	0	0	0	0
##	cz	0	0	0	0	0	0	0	0	0
##	de	0	0	0	0	0	0	0	0	0
##	dk	0	0	0	0	0	0	0	0	0
##	ec	0	0	0	0	0	0	0	0	0
##	ee	0	0	0	0	0	0	0	0	0
##	es	0	0	0	0	0	0	0	0	0
##	fi	0	0	0	0	0	0	0	0	0
##	fr	0	0	0	0	0	0	0	0	0
##	gb	0	0	0	0	0	0	0	0	0
##	gi	0	0	0	0	0	0	0	0	0
##	gr	0	0	0	0	0	0	0	0	0
##	gt	0	0	0	0	0	0	0	0	0
##	hk	0	0	0	0	0	0	0	0	0
##	hn	0	0	0	0	0	0	0	0	0
##	hr	0	0	0	0	0	0	0	0	0
##	hu	0	0	0	0	0	0	0	0	0
##	id	0	0	0	0	0	0	0	0	0
##	ie	0	0	0	0	0	0	0	0	0
##	il	0	0	0	0	0	0	0	0	0
##	in	0	0	0	0	0	0	0	0	0
##	is	0	0	0	0	0	0	0	0	0
##	it	0	0	0	0	0	0	0	0	0
##	jp	0	0	0	0	0	0	0	0	0
##	kr	0	0	0	0	0	0	0	0	0
##	li	0	0	0	0	0	0	0	0	0
##	lt	0	0	0	0	0	0	0	0	0
##	lv	0	0	0	0	0	0	0	0	0
##	mc	0	0	0	0	0	0	0	0	0
##	md	0	0	0	0	0	0	0	0	0
##	mx	0	0	0	0	0	0	0	0	0
##	my	0	0	0	0	0	0	0	0	0
##	nl	0	0	0	0	0	0	0	0	0
##	no	0	0	0	0	0	0	0	0	0
##	nz	0	0	0	0	0	0	0	0	0
##	pe	0	0	0	0	0	0	0	0	0
##	ph	0	0	0	0	0	0	0	0	0
##	pl	0	0	0	0	0	0	0	0	0
##	pt	0	0	0	0	0	0	0	0	0
##	py	0	0	0	0	0	0	0	0	0
##	ro	0	0	0	0	0	0	0	0	0
##	ru	0	0	0	0	0	0	0	0	0

##	se	0	0	0	0	0	0	0	0	0
##	sg	0	0	0	0	0	0	0	0	0
##	sk	0	0	0	0	0	0	0	0	0
##	sm	0	0	0	0	0	0	0	0	0
##	th	0	0	0	0	0	0	0	0	0
##	tr	0	0	0	0	0	0	0	0	0
##	tw	0	0	0	0	0	0	0	0	0
##	ua	0	0	0	0	0	0	0	0	0
##	us	0	0	0	0	0	0	0	0	0
##	uy	0	0	0	0	0	0	0	0	0
##	ve	0	0	0	0	0	0	0	0	0
##	za	0	0	0	0	0	0	0	0	0
##										
##		Colombia	Costa Rica	Croatia	Czechia	Denmark	Ecuador	Estonia	Finland	France
##	ar	0	0	0	0	0	0	0	0	0
##	at	0	0	0	0	0	0	0	0	0
##	au	0	0	0	0	0	0	0	0	0
##	be	0	0	0	0	0	0	0	0	0
##	bg	0	0	0	0	0	0	0	0	0
##	bo	0	0	0	0	0	0	0	0	0
##	br	0	0	0	0	0	0	0	0	0
##	ca	0	0	0	0	0	0	0	0	0
##	ch	0	0	0	0	0	0	0	0	0
##	cl	0	0	0	0	0	0	0	0	0
##	co	1	0	0	0	0	0	0	0	0
##	cr	0	1	0	0	0	0	0	0	0
##	cz	0	0	0	1	0	0	0	0	0
##	de	0	0	0	0	0	0	0	0	0
##	dk	0	0	0	0	1	0	0	0	0
##	ec	0	0	0	0	0	1	0	0	0
##	ee	0	0	0	0	0	0	1	0	0
##	es	0	0	0	0	0	0	0	0	0
##	fi	0	0	0	0	0	0	0	1	0
##	fr	0	0	0	0	0	0	0	0	1
##	gb	0	0	0	0	0	0	0	0	0
##	gi	0	0	0	0	0	0	0	0	0
##	gr	0	0	0	0	0	0	0	0	0
##	gt	0	0	0	0	0	0	0	0	0
##	hk	0	0	0	0	0	0	0	0	0
##	hn	0	0	0	0	0	0	0	0	0
##	hr	0	0	1	0	0	0	0	0	0
##	hu	0	0	0	0	0	0	0	0	0
##	id	0	0	0	0	0	0	0	0	0
##	ie	0	0	0	0	0	0	0	0	0
##	il	0	0	0	0	0	0	0	0	0
##	in	0	0	0	0	0	0	0	0	0
##	is	0	0	0	0	0	0	0	0	0
##	it	0	0	0	0	0	0	0	0	0
##	jp	0	0	0	0	0	0	0	0	0
##	kr	0	0	0	0	0	0	0	0	0
##	li	0	0	0	0	0	0	0	0	0
##	lt	0	0	0	0	0	0	0	0	0
##	lv	0	0	0	0	0	0	0	0	0
##	mc	0	0	0	0	0	0	0	0	0
##	md	0	0	0	0	0	0	0	0	0
##	mx	0	0	0	0	0	0	0	0	0

##	my	0	0	0	0	0	0	0	0
##	nl	0	0	0	0	0	0	0	0
##	no	0	0	0	0	0	0	0	0
##	nz	0	0	0	0	0	0	0	0
##	pe	0	0	0	0	0	0	0	0
##	ph	0	0	0	0	0	0	0	0
##	pl	0	0	0	0	0	0	0	0
##	pt	0	0	0	0	0	0	0	0
##	py	0	0	0	0	0	0	0	0
##	ro	0	0	0	0	0	0	0	0
##	ru	0	0	0	0	0	0	0	0
##	se	0	0	0	0	0	0	0	0
##	sg	0	0	0	0	0	0	0	0
##	sk	0	0	0	0	0	0	0	0
##	sm	0	0	0	0	0	0	0	0
##	th	0	0	0	0	0	0	0	0
##	tr	0	0	0	0	0	0	0	0
##	tw	0	0	0	0	0	0	0	0
##	ua	0	0	0	0	0	0	0	0
##	us	0	0	0	0	0	0	0	0
##	uy	0	0	0	0	0	0	0	0
##	ve	0	0	0	0	0	0	0	0
##	za	0	0	0	0	0	0	0	0

##

	Germany	Gibraltar	Greece	Guatemala	Honduras	Hong Kong	Hungary	Iceland
--	---------	-----------	--------	-----------	----------	-----------	---------	---------

##	ar	0	0	0	0	0	0	0
##	at	0	0	0	0	0	0	0
##	au	0	0	0	0	0	0	0
##	be	0	0	0	0	0	0	0
##	bg	0	0	0	0	0	0	0
##	bo	0	0	0	0	0	0	0
##	br	0	0	0	0	0	0	0
##	ca	0	0	0	0	0	0	0
##	ch	0	0	0	0	0	0	0
##	cl	0	0	0	0	0	0	0
##	co	0	0	0	0	0	0	0
##	cr	0	0	0	0	0	0	0
##	cz	0	0	0	0	0	0	0
##	de	1	0	0	0	0	0	0
##	dk	0	0	0	0	0	0	0
##	ec	0	0	0	0	0	0	0
##	ee	0	0	0	0	0	0	0
##	es	0	0	0	0	0	0	0
##	fi	0	0	0	0	0	0	0
##	fr	0	0	0	0	0	0	0
##	gb	0	0	0	0	0	0	0
##	gi	0	1	0	0	0	0	0
##	gr	0	0	1	0	0	0	0
##	gt	0	0	0	1	0	0	0
##	hk	0	0	0	0	1	0	0
##	hn	0	0	0	0	1	0	0
##	hr	0	0	0	0	0	0	0
##	hu	0	0	0	0	0	1	0
##	id	0	0	0	0	0	0	0
##	ie	0	0	0	0	0	0	0
##	il	0	0	0	0	0	0	0

##	in	0	0	0	0	0	0	0	0	
##	is	0	0	0	0	0	0	0	1	
##	it	0	0	0	0	0	0	0	0	
##	jp	0	0	0	0	0	0	0	0	
##	kr	0	0	0	0	0	0	0	0	
##	li	0	0	0	0	0	0	0	0	
##	lt	0	0	0	0	0	0	0	0	
##	lv	0	0	0	0	0	0	0	0	
##	mc	0	0	0	0	0	0	0	0	
##	md	0	0	0	0	0	0	0	0	
##	mx	0	0	0	0	0	0	0	0	
##	my	0	0	0	0	0	0	0	0	
##	nl	0	0	0	0	0	0	0	0	
##	no	0	0	0	0	0	0	0	0	
##	nz	0	0	0	0	0	0	0	0	
##	pe	0	0	0	0	0	0	0	0	
##	ph	0	0	0	0	0	0	0	0	
##	pl	0	0	0	0	0	0	0	0	
##	pt	0	0	0	0	0	0	0	0	
##	py	0	0	0	0	0	0	0	0	
##	ro	0	0	0	0	0	0	0	0	
##	ru	0	0	0	0	0	0	0	0	
##	se	0	0	0	0	0	0	0	0	
##	sg	0	0	0	0	0	0	0	0	
##	sk	0	0	0	0	0	0	0	0	
##	sm	0	0	0	0	0	0	0	0	
##	th	0	0	0	0	0	0	0	0	
##	tr	0	0	0	0	0	0	0	0	
##	tw	0	0	0	0	0	0	0	0	
##	ua	0	0	0	0	0	0	0	0	
##	us	0	0	0	0	0	0	0	0	
##	uy	0	0	0	0	0	0	0	0	
##	ve	0	0	0	0	0	0	0	0	
##	za	0	0	0	0	0	0	0	0	
##										
##		India	Indonesia	Ireland	Israel	Italy	Japan	Latvia	Liechtenstein	Lithuania
##	ar	0	0	0	0	0	0	0	0	0
##	at	0	0	0	0	0	0	0	0	0
##	au	0	0	0	0	0	0	0	0	0
##	be	0	0	0	0	0	0	0	0	0
##	bg	0	0	0	0	0	0	0	0	0
##	bo	0	0	0	0	0	0	0	0	0
##	br	0	0	0	0	0	0	0	0	0
##	ca	0	0	0	0	0	0	0	0	0
##	ch	0	0	0	0	0	0	0	0	0
##	cl	0	0	0	0	0	0	0	0	0
##	co	0	0	0	0	0	0	0	0	0
##	cr	0	0	0	0	0	0	0	0	0
##	cz	0	0	0	0	0	0	0	0	0
##	de	0	0	0	0	0	0	0	0	0
##	dk	0	0	0	0	0	0	0	0	0
##	ec	0	0	0	0	0	0	0	0	0
##	ee	0	0	0	0	0	0	0	0	0
##	es	0	0	0	0	0	0	0	0	0
##	fi	0	0	0	0	0	0	0	0	0
##	fr	0	0	0	0	0	0	0	0	0

##	gb	0	0	0	0	0	0	0	0	0
##	gi	0	0	0	0	0	0	0	0	0
##	gr	0	0	0	0	0	0	0	0	0
##	gt	0	0	0	0	0	0	0	0	0
##	hk	0	0	0	0	0	0	0	0	0
##	hn	0	0	0	0	0	0	0	0	0
##	hr	0	0	0	0	0	0	0	0	0
##	hu	0	0	0	0	0	0	0	0	0
##	id	0	1	0	0	0	0	0	0	0
##	ie	0	0	1	0	0	0	0	0	0
##	il	0	0	0	1	0	0	0	0	0
##	in	1	0	0	0	0	0	0	0	0
##	is	0	0	0	0	0	0	0	0	0
##	it	0	0	0	0	1	0	0	0	0
##	jp	0	0	0	0	0	1	0	0	0
##	kr	0	0	0	0	0	0	0	0	0
##	li	0	0	0	0	0	0	0	1	0
##	lt	0	0	0	0	0	0	0	0	1
##	lv	0	0	0	0	0	0	1	0	0
##	mc	0	0	0	0	0	0	0	0	0
##	md	0	0	0	0	0	0	0	0	0
##	mx	0	0	0	0	0	0	0	0	0
##	my	0	0	0	0	0	0	0	0	0
##	nl	0	0	0	0	0	0	0	0	0
##	no	0	0	0	0	0	0	0	0	0
##	nz	0	0	0	0	0	0	0	0	0
##	pe	0	0	0	0	0	0	0	0	0
##	ph	0	0	0	0	0	0	0	0	0
##	pl	0	0	0	0	0	0	0	0	0
##	pt	0	0	0	0	0	0	0	0	0
##	py	0	0	0	0	0	0	0	0	0
##	ro	0	0	0	0	0	0	0	0	0
##	ru	0	0	0	0	0	0	0	0	0
##	se	0	0	0	0	0	0	0	0	0
##	sg	0	0	0	0	0	0	0	0	0
##	sk	0	0	0	0	0	0	0	0	0
##	sm	0	0	0	0	0	0	0	0	0
##	th	0	0	0	0	0	0	0	0	0
##	tr	0	0	0	0	0	0	0	0	0
##	tw	0	0	0	0	0	0	0	0	0
##	ua	0	0	0	0	0	0	0	0	0
##	us	0	0	0	0	0	0	0	0	0
##	uy	0	0	0	0	0	0	0	0	0
##	ve	0	0	0	0	0	0	0	0	0
##	za	0	0	0	0	0	0	0	0	0
##										
##		Malaysia	Mexico	Moldova	Monaco	Netherlands	New Zealand	Norway	Paraguay	
##	ar	0	0	0	0	0	0	0	0	
##	at	0	0	0	0	0	0	0	0	
##	au	0	0	0	0	0	0	0	0	
##	be	0	0	0	0	0	0	0	0	
##	bg	0	0	0	0	0	0	0	0	
##	bo	0	0	0	0	0	0	0	0	
##	br	0	0	0	0	0	0	0	0	
##	ca	0	0	0	0	0	0	0	0	
##	ch	0	0	0	0	0	0	0	0	

##	cl	0	0	0	0	0	0	0	0
##	co	0	0	0	0	0	0	0	0
##	cr	0	0	0	0	0	0	0	0
##	cz	0	0	0	0	0	0	0	0
##	de	0	0	0	0	0	0	0	0
##	dk	0	0	0	0	0	0	0	0
##	ec	0	0	0	0	0	0	0	0
##	ee	0	0	0	0	0	0	0	0
##	es	0	0	0	0	0	0	0	0
##	fi	0	0	0	0	0	0	0	0
##	fr	0	0	0	0	0	0	0	0
##	gb	0	0	0	0	0	0	0	0
##	gi	0	0	0	0	0	0	0	0
##	gr	0	0	0	0	0	0	0	0
##	gt	0	0	0	0	0	0	0	0
##	hk	0	0	0	0	0	0	0	0
##	hn	0	0	0	0	0	0	0	0
##	hr	0	0	0	0	0	0	0	0
##	hu	0	0	0	0	0	0	0	0
##	id	0	0	0	0	0	0	0	0
##	ie	0	0	0	0	0	0	0	0
##	il	0	0	0	0	0	0	0	0
##	in	0	0	0	0	0	0	0	0
##	is	0	0	0	0	0	0	0	0
##	it	0	0	0	0	0	0	0	0
##	jp	0	0	0	0	0	0	0	0
##	kr	0	0	0	0	0	0	0	0
##	li	0	0	0	0	0	0	0	0
##	lt	0	0	0	0	0	0	0	0
##	lv	0	0	0	0	0	0	0	0
##	mc	0	0	0	1	0	0	0	0
##	md	0	0	1	0	0	0	0	0
##	mx	0	1	0	0	0	0	0	0
##	my	1	0	0	0	0	0	0	0
##	nl	0	0	0	0	1	0	0	0
##	no	0	0	0	0	0	0	1	0
##	nz	0	0	0	0	0	1	0	0
##	pe	0	0	0	0	0	0	0	0
##	ph	0	0	0	0	0	0	0	0
##	pl	0	0	0	0	0	0	0	0
##	pt	0	0	0	0	0	0	0	0
##	py	0	0	0	0	0	0	0	1
##	ro	0	0	0	0	0	0	0	0
##	ru	0	0	0	0	0	0	0	0
##	se	0	0	0	0	0	0	0	0
##	sg	0	0	0	0	0	0	0	0
##	sk	0	0	0	0	0	0	0	0
##	sm	0	0	0	0	0	0	0	0
##	th	0	0	0	0	0	0	0	0
##	tr	0	0	0	0	0	0	0	0
##	tw	0	0	0	0	0	0	0	0
##	ua	0	0	0	0	0	0	0	0
##	us	0	0	0	0	0	0	0	0
##	uy	0	0	0	0	0	0	0	0
##	ve	0	0	0	0	0	0	0	0
##	za	0	0	0	0	0	0	0	0

##		Peru	Philippines	Poland	Portugal	Romania	Russia	San Marino	Singapore
##	ar	0	0	0	0	0	0	0	0
##	at	0	0	0	0	0	0	0	0
##	au	0	0	0	0	0	0	0	0
##	be	0	0	0	0	0	0	0	0
##	bg	0	0	0	0	0	0	0	0
##	bo	0	0	0	0	0	0	0	0
##	br	0	0	0	0	0	0	0	0
##	ca	0	0	0	0	0	0	0	0
##	ch	0	0	0	0	0	0	0	0
##	cl	0	0	0	0	0	0	0	0
##	co	0	0	0	0	0	0	0	0
##	cr	0	0	0	0	0	0	0	0
##	cz	0	0	0	0	0	0	0	0
##	de	0	0	0	0	0	0	0	0
##	dk	0	0	0	0	0	0	0	0
##	ec	0	0	0	0	0	0	0	0
##	ee	0	0	0	0	0	0	0	0
##	es	0	0	0	0	0	0	0	0
##	fi	0	0	0	0	0	0	0	0
##	fr	0	0	0	0	0	0	0	0
##	gb	0	0	0	0	0	0	0	0
##	gi	0	0	0	0	0	0	0	0
##	gr	0	0	0	0	0	0	0	0
##	gt	0	0	0	0	0	0	0	0
##	hk	0	0	0	0	0	0	0	0
##	hn	0	0	0	0	0	0	0	0
##	hr	0	0	0	0	0	0	0	0
##	hu	0	0	0	0	0	0	0	0
##	id	0	0	0	0	0	0	0	0
##	ie	0	0	0	0	0	0	0	0
##	il	0	0	0	0	0	0	0	0
##	in	0	0	0	0	0	0	0	0
##	is	0	0	0	0	0	0	0	0
##	it	0	0	0	0	0	0	0	0
##	jp	0	0	0	0	0	0	0	0
##	kr	0	0	0	0	0	0	0	0
##	li	0	0	0	0	0	0	0	0
##	lt	0	0	0	0	0	0	0	0
##	lv	0	0	0	0	0	0	0	0
##	mc	0	0	0	0	0	0	0	0
##	md	0	0	0	0	0	0	0	0
##	mx	0	0	0	0	0	0	0	0
##	my	0	0	0	0	0	0	0	0
##	nl	0	0	0	0	0	0	0	0
##	no	0	0	0	0	0	0	0	0
##	nz	0	0	0	0	0	0	0	0
##	pe	1	0	0	0	0	0	0	0
##	ph	0	1	0	0	0	0	0	0
##	pl	0	0	1	0	0	0	0	0
##	pt	0	0	0	1	0	0	0	0
##	py	0	0	0	0	0	0	0	0
##	ro	0	0	0	0	1	0	0	0
##	ru	0	0	0	0	0	1	0	0
##	se	0	0	0	0	0	0	0	0

##	sg	0	0	0	0	0	0	0	1
##	sk	0	0	0	0	0	0	0	0
##	sm	0	0	0	0	0	0	1	0
##	th	0	0	0	0	0	0	0	0
##	tr	0	0	0	0	0	0	0	0
##	tw	0	0	0	0	0	0	0	0
##	ua	0	0	0	0	0	0	0	0
##	us	0	0	0	0	0	0	0	0
##	uy	0	0	0	0	0	0	0	0
##	ve	0	0	0	0	0	0	0	0
##	za	0	0	0	0	0	0	0	0
##									
##		Slovakia	South Africa	South Korea	Spain	Sweden	Switzerland	Taiwan	Thailand
##	ar	0	0	0	0	0	0	0	0
##	at	0	0	0	0	0	0	0	0
##	au	0	0	0	0	0	0	0	0
##	be	0	0	0	0	0	0	0	0
##	bg	0	0	0	0	0	0	0	0
##	bo	0	0	0	0	0	0	0	0
##	br	0	0	0	0	0	0	0	0
##	ca	0	0	0	0	0	0	0	0
##	ch	0	0	0	0	0	1	0	0
##	cl	0	0	0	0	0	0	0	0
##	co	0	0	0	0	0	0	0	0
##	cr	0	0	0	0	0	0	0	0
##	cz	0	0	0	0	0	0	0	0
##	de	0	0	0	0	0	0	0	0
##	dk	0	0	0	0	0	0	0	0
##	ec	0	0	0	0	0	0	0	0
##	ee	0	0	0	0	0	0	0	0
##	es	0	0	0	1	0	0	0	0
##	fi	0	0	0	0	0	0	0	0
##	fr	0	0	0	0	0	0	0	0
##	gb	0	0	0	0	0	0	0	0
##	gi	0	0	0	0	0	0	0	0
##	gr	0	0	0	0	0	0	0	0
##	gt	0	0	0	0	0	0	0	0
##	hk	0	0	0	0	0	0	0	0
##	hn	0	0	0	0	0	0	0	0
##	hr	0	0	0	0	0	0	0	0
##	hu	0	0	0	0	0	0	0	0
##	id	0	0	0	0	0	0	0	0
##	ie	0	0	0	0	0	0	0	0
##	il	0	0	0	0	0	0	0	0
##	in	0	0	0	0	0	0	0	0
##	is	0	0	0	0	0	0	0	0
##	it	0	0	0	0	0	0	0	0
##	jp	0	0	0	0	0	0	0	0
##	kr	0	0	1	0	0	0	0	0
##	li	0	0	0	0	0	0	0	0
##	lt	0	0	0	0	0	0	0	0
##	lv	0	0	0	0	0	0	0	0
##	mc	0	0	0	0	0	0	0	0
##	md	0	0	0	0	0	0	0	0
##	mx	0	0	0	0	0	0	0	0
##	my	0	0	0	0	0	0	0	0

##	nl	0	0	0	0	0	0	0	0
##	no	0	0	0	0	0	0	0	0
##	nz	0	0	0	0	0	0	0	0
##	pe	0	0	0	0	0	0	0	0
##	ph	0	0	0	0	0	0	0	0
##	pl	0	0	0	0	0	0	0	0
##	pt	0	0	0	0	0	0	0	0
##	py	0	0	0	0	0	0	0	0
##	ro	0	0	0	0	0	0	0	0
##	ru	0	0	0	0	0	0	0	0
##	se	0	0	0	0	1	0	0	0
##	sg	0	0	0	0	0	0	0	0
##	sk	1	0	0	0	0	0	0	0
##	sm	0	0	0	0	0	0	0	0
##	th	0	0	0	0	0	0	0	1
##	tr	0	0	0	0	0	0	0	0
##	tw	0	0	0	0	0	0	1	0
##	ua	0	0	0	0	0	0	0	0
##	us	0	0	0	0	0	0	0	0
##	uy	0	0	0	0	0	0	0	0
##	ve	0	0	0	0	0	0	0	0
##	za	0	1	0	0	0	0	0	0
##									
##	Turkey Ukraine United Kingdom United States Uruguay Venezuela								
##	ar	0	0	0	0	0	0	0	
##	at	0	0	0	0	0	0	0	
##	au	0	0	0	0	0	0	0	
##	be	0	0	0	0	0	0	0	
##	bg	0	0	0	0	0	0	0	
##	bo	0	0	0	0	0	0	0	
##	br	0	0	0	0	0	0	0	
##	ca	0	0	0	0	0	0	0	
##	ch	0	0	0	0	0	0	0	
##	cl	0	0	0	0	0	0	0	
##	co	0	0	0	0	0	0	0	
##	cr	0	0	0	0	0	0	0	
##	cz	0	0	0	0	0	0	0	
##	de	0	0	0	0	0	0	0	
##	dk	0	0	0	0	0	0	0	
##	ec	0	0	0	0	0	0	0	
##	ee	0	0	0	0	0	0	0	
##	es	0	0	0	0	0	0	0	
##	fi	0	0	0	0	0	0	0	
##	fr	0	0	0	0	0	0	0	
##	gb	0	0	1	0	0	0	0	
##	gi	0	0	0	0	0	0	0	
##	gr	0	0	0	0	0	0	0	
##	gt	0	0	0	0	0	0	0	
##	hk	0	0	0	0	0	0	0	
##	hn	0	0	0	0	0	0	0	
##	hr	0	0	0	0	0	0	0	
##	hu	0	0	0	0	0	0	0	
##	id	0	0	0	0	0	0	0	
##	ie	0	0	0	0	0	0	0	
##	il	0	0	0	0	0	0	0	
##	in	0	0	0	0	0	0	0	

##	is	0	0	0	0	0	0
##	it	0	0	0	0	0	0
##	jp	0	0	0	0	0	0
##	kr	0	0	0	0	0	0
##	li	0	0	0	0	0	0
##	lt	0	0	0	0	0	0
##	lv	0	0	0	0	0	0
##	mc	0	0	0	0	0	0
##	md	0	0	0	0	0	0
##	mx	0	0	0	0	0	0
##	my	0	0	0	0	0	0
##	nl	0	0	0	0	0	0
##	no	0	0	0	0	0	0
##	nz	0	0	0	0	0	0
##	pe	0	0	0	0	0	0
##	ph	0	0	0	0	0	0
##	pl	0	0	0	0	0	0
##	pt	0	0	0	0	0	0
##	py	0	0	0	0	0	0
##	ro	0	0	0	0	0	0
##	ru	0	0	0	0	0	0
##	se	0	0	0	0	0	0
##	sg	0	0	0	0	0	0
##	sk	0	0	0	0	0	0
##	sm	0	0	0	0	0	0
##	th	0	0	0	0	0	0
##	tr	1	0	0	0	0	0
##	tw	0	0	0	0	0	0
##	ua	0	1	0	0	0	0
##	us	0	0	0	1	0	0
##	uy	0	0	0	0	1	0
##	ve	0	0	0	0	0	1
##	za	0	0	0	0	0	0

CONCLUSIONS

As a conclusion taken by this analysis I would declare that netflix could look expensive but the wole library is huge and at the end of the day it could make sense if you love TV shows and films. Nevertheless, I see too much similar prices in regions with too much different purchasing power, such as Switzerland vs Argentina.

With this said, I finish, I hope you have enjoyed it and learned a little more about the prices and catalogue of the hugest streaming platform. ***Thanks and see you soon!!!***