CCT College Dublin

Assessment Cover Page

Module Title:	Statistical techniques for Data Analysis
Assessment Title:	Descriptive Stats – CA1
Lecturer Name:	Aldana Louzan
Student Full Name:	Raphael Fernandes Gomes
Student Number:	2022091
Assessment Due Date:	20/03/2022
Date of Submission:	20/03/2022

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Index

Introduction

This report gives a small overview of descriptive analysis through research, practical questions, and application of descriptive analysis with a data set in JupyterNotebook.

The purpose of the three sections of this analysis is to better understand Descriptive Analysis and measures of central tendency and measures of variability (spread) and apply them to a data set to perform analysis.

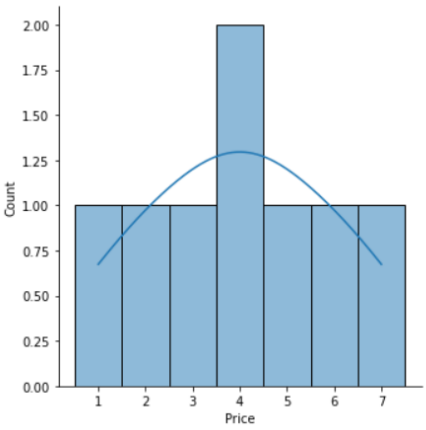
First Section: Research

- Is the following statement true? Justify carrying out research and use appropriate references to support your answer:

"If a dataset has mean=median=mode, we can ensure that the dataset follows a Symmetric Distribution. In addition, every time that a dataset follows a Symmetric Distribution, that means that mean=median=mode".

If a dataset has the same mean, median, and mode values, we can conclude that it will have a perfectly normal symmetrical distribution (Unimodal), represented visually by the peak of the curve and with the tails on both sides also symmetrically. [1, 2, 3, 4, 5, 6, 7]

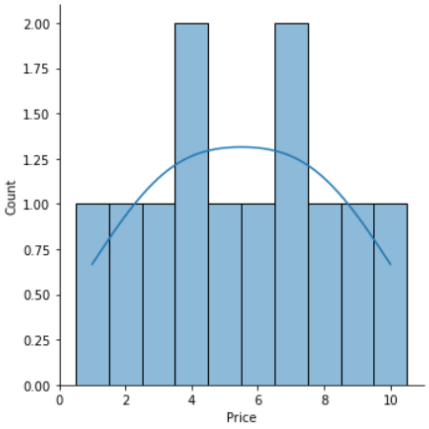
```
In [101]: sns.displot(Products_data['Price'], kde=True, discrete=True )
Out[101]: <seaborn.axisgrid.FacetGrid at 0x1be5dd37550>
```



(Figure 1: The normal distribution shown in this plot has a mean of 4, mode of 4 and median 4).

But in the other hand, when a dataset follows a Symmetric Distribution doesn't mean that the Dataset has the mean, median and mode equally because the dataset can be two modes (bimodal) or multimodal sample where the two modes or the multi modes would be different from the mean and median and when that happens you might report the mean or median as appropriate. [3, 6]

```
In [106]: sns.displot(Products_databi['Price'], kde=True, discrete=True )
Out[106]: <seaborn.axisgrid.FacetGrid at 0x1be5ed22850>
```

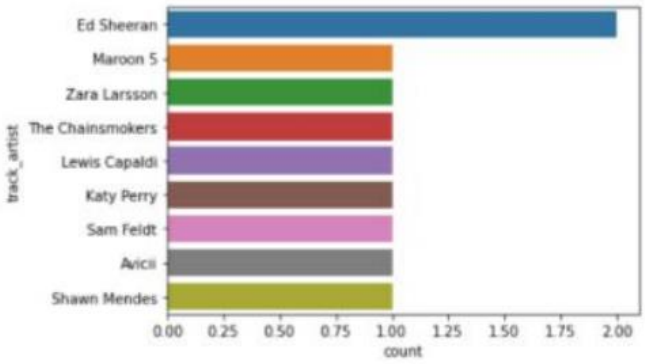


(**Figure 2:** The normal distribution shown in this plot has a mean of 5.5, mode of 4 and 7 and median 5.5).

Second Section: Practical Questions

Answer the following questions based on this bar chart which plots the variable “track_artist” from the dataframe = data_spotify. You must justify all your answers with appropriate statistical concepts and if you carried out research, you must provide appropriate references.

```
In [14]: sns.countplot(y='track_artist', data=data_spotify.iloc[0:10])
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x177a40a0>
```



A. Can we deduce that there are only 8 different track artists in our entire dataset?

No. As the .iloc[ ] indexer was used to select the entire column and the first 10 rows of the dataset is not possible deduce that there are only 8 different track artists.

B. In the range analysed, can we deduce there is a unique mode?

Yes. In the range analysed and represented on the bar chart we clearly can see that we have only one mode as “Ed Sheeran” is the most frequent in the range that was selected to be analysed.

C. Is it true that in the range analysed the mode has a value of 2?

Yes. The mode has a value of 2 as represented by the horizontal axis where it shows the frequency that the elements appear in the range selected to be analysed.

D. How many data elements there are in the range analysed?

The range picked to be analysed with the function .iloc[] contains 10 elements.

E. In this plot, the horizontal axis represents the mode.

No, the horizontal axis represents the counts of observation in the categorical variable, in other words, the horizontal axis is showing how many times the elements appear.

Third Section: Descriptive stats & Dataset

Introduction of the Data set

The data set that I choose to deal with is a data set containing thousands of games with their names, the number of Sales in North America (NA) Europe (EU), Japan (JP), Others (Rest of the World), and Global Sales (Total worldwide), Year of release, Genre, Platforms were the games released and Publisher of the Game.

I chose this data set as a fan of games and because the game industry is growing enormously.

According to the latest Accenture (NYSE: ACN) report, it is estimated that the total value of the games industry now exceeds $300 billion.[2]

Data Dictionary

Col	Full Name of Variables	Definition of Variables	Type of Variables	
			Qualitative/Quantitative	Categorical Disc/Contin
A	Rank	Rank of overall sales	Qualitative	Categorical
B	Name	The games name	Qualitative	Categorical
C	Platform	Platform of the games release (i.e. PC, PS4, Xbox, etc.)	Qualitative	Categorical
D	Year	Year of the game's release	Quantitative	Continuous
E	Genre	Genre of the games (i.e. Racing, Sports, Action, etc.)	Qualitative	Categorical
F	Publisher	Publisher of the game	Qualitative	Categorical
G	NA_Sales	Sales in North America	Quantitative	Continuous
H	EU_Sales	Sales in Europe	Quantitative	Continuous
I	JP_Sales	Sales in Japan	Quantitative	Continuous
J	Other_Sales	Sales in the rest of the world	Quantitative	Continuous
K	Global_Sales	Total Worldwide sales	Quantitative	Continuous

Importing Data and Data Review

Importing and checking all the information about the content in my dataset, (Data type) of each of columns, memory that the Data frame occupies, number of rows, columns, memory usage.

In [1]:
```
import pandas as pd
import numpy as np
import random
import time
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
%matplotlib inline
```

In [2]:
```
games_df = pd.read_csv("data/games.csv")
```

In [3]:
```
games_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 11 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Rank                     16598 non-null  int64
 1   Game Name                16598 non-null  object
 2   Platform                 16598 non-null  object
 3   Release Year             16327 non-null  float64
 4   Genre                    16598 non-null  object
 5   Publisher                16540 non-null  object
 6   NorthAmerica_Sales       16598 non-null  float64
 7   Europe_Sales             16598 non-null  float64
 8   Japan_Sales              16598 non-null  float64
 9   Rest _of_the_World_Sales  16598 non-null  float64
 10  Global_Sales             16598 non-null  float64
dtypes: float64(6), int64(1), object(4)
memory usage: 1.4+ MB
```

In [4]:
```
games_df
```

Out[4]:

	Rank	Game Name	Platform	Release Year	Genre	Publisher	NorthAmerica_Sales	Europe_Sales	Japan_Sales	Rest _of_the_World_Sales	Global_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37
...	...	...	...	...	...	...	...	...	...	...	...
16593	16596	Woody Woodpecker in Crazy Castle 5	GBA	2002.0	Platform	Kemco	0.01	0.00	0.00	0.00	0.01
16594	16597	Men in Black II: Alien Escape	GC	2003.0	Shooter	Infogrames	0.01	0.00	0.00	0.00	0.01
16595	16598	SCORE International Baja 1000: The Official Game	PS2	2008.0	Racing	Activision	0.00	0.00	0.00	0.00	0.01
16596	16599	Know How 2	DS	2010.0	Puzzle	7G//AMES	0.00	0.01	0.00	0.00	0.01
16597	16600	Spirits & Spells	GBA	2003.0	Platform	Wanadoo	0.01	0.00	0.00	0.00	0.01

16598 rows × 11 columns

In [5]:
```
games_df.nunique()
```

Out[5]:
```
Rank                        16597
Game_Name                   11492
Platform                       30
Release_Year                   41
Genre                          12
Publisher                     577
NorthAmerica_Sales            409
Europe_Sales                  305
Japan_Sales                   244
Rest _of_the_World_Sales      157
Global_Sales                  623
dtype: int64
```

In [6]:
```
games_df.describe()
```

Out[6]:

	Rank	Release_Year	NorthAmerica_Sales	Europe_Sales	Japan_Sales	Rest _of_the_World_Sales	Global_Sales
count	16597.000000	16338.000000	16597.000000	16597.000000	16597.000000	16597.000000	16597.000000
mean	8300.149184	2006.402252	0.264683	0.146661	0.077785	0.048066	0.537472
std	4791.638040	5.837302	0.816705	0.505365	0.309300	0.188594	1.555070
min	1.000000	1977.000000	0.000000	0.000000	0.000000	0.000000	0.010000
25%	4151.000000	2003.000000	0.000000	0.000000	0.000000	0.000000	0.060000
50%	8300.000000	2007.000000	0.080000	0.020000	0.000000	0.010000	0.170000
75%	12449.000000	2010.000000	0.240000	0.110000	0.040000	0.040000	0.470000
max	16600.000000	2020.000000	41.490000	29.020000	10.220000	10.570000	82.740000

Preparing the Data Set to be Analyse

Checking if my data set is containing any missing values or duplicate that can affect my analyse and fixing possible errors.

In [17]:
```
games_df.isnull().sum()
```

Out[17]:
```
Rank                          0
Game_Name                     0
Platform                      0
Release_Year                259
Genre                         0
Publisher                     0
NorthAmerica_Sales            0
Europe_Sales                  0
Japan_Sales                   0
Rest _of_the_World_Sales      0
Global_Sales                  0
dtype: int64
```

In [18]:
```
games_df.dropna(inplace=True)
games_df.drop(games_df[games_df['Release_Year']>2016].index, inplace=True)
games_df.isnull().sum()
#Taking out the missing values, once they will not impact my analise
```

Out[18]:
```
Rank                        0
Game_Name                   0
Platform                    0
Release_Year                0
Genre                       0
Publisher                   0
NorthAmerica_Sales          0
Europe_Sales                0
Japan_Sales                 0
Rest _of_the_World_Sales    0
Global_Sales                0
dtype: int64
```

In [21]:
```
games_df["Platform"].replace("DS", "Nintendo DS", inplace=True)
games_df["Platform"].replace("NES", "Nintendo NES", inplace=True)
games_df["Platform"].replace("SNES", "Super Nintendo", inplace=True)
games_df["Platform"].replace("GB", "GameBoy Color", inplace=True)
games_df["Platform"].replace("3DS", "Nintendo 3DS", inplace=True)
games_df["Platform"].replace("DC", "Dreamcast", inplace=True)
games_df["Platform"].replace("GEN", "SEGA Genesis", inplace=True)
games_df["Platform"].replace("GG", "Game Gear", inplace=True)
games_df["Platform"].replace("NG", "Neo Geo", inplace=True)
games_df["Platform"].replace("SAT", "Sega Saturn", inplace=True)
games_df["Platform"].replace("SCD", "Sega CD", inplace=True)
games_df["Platform"].replace("TG16", "Turbo Grafx", inplace=True)
games_df["Platform"].replace("WS", "WanderSwan", inplace=True)
games_df["Platform"].replace("2600", "Atari", inplace=True)
games_df.drop(columns="Rank",inplace=True)

#Changing the Name to be more Clear, and remove the column that I not gonna use.
```

In [22]:
```
games_df.head()
```

Out[22]:

	Game_Name	Platform	Release_Year	Genre	Publisher	NorthAmerica_Sales	Europe_Sales	Japan_Sales	Rest _of_the_World_Sales	Global_Sales
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41	29	4	8	83
1	Super Mario Bros.	Nintendo NES	1985.0	Platform	Nintendo	29	4	7	1	40
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	16	13	4	3	36
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	16	11	3	3	33
4	Pokemon Red/Pokemon Blue	GameBoy Color	1996.0	Role-Playing	Nintendo	11	9	10	1	31

Central Tendency, Dispersion and 5 Number Summary

In [11]:
```
meanYear = games_df["Year"].mean()
print(meanYear)
modeYear = games_df["Year"].mode()
print(modeYear)
medianYear = games_df["Year"].median()
print(medianYear)
```

```
2006.3994734908779
0    2009.0
dtype: float64
2007.0
```

In [34]:
```
meanGlob = games_df["Global_Sales"].mean()
print(meanGlob)
modeGlob = games_df["Global_Sales"].mode()
print(modeGlob)
medianGlob = games_df["Global_Sales"].median()
print(medianGlob)
```

```
0.5417980898739431
0    0.02
dtype: float64
0.17
```

In [14]:
```
games_df['Year'].std()
```

Out[14]: 5.835278628418993

In [15]:
```
games_df['Year'].var()
```

Out[15]: 34.05047667128345

In [39]:
```
games_df['Global_Sales'].std()
```

Out[39]: 1.5664910453602057

In [38]:
```
games_df['Global_Sales'].var()
```

Out[38]: 2.45389419519371

In [16]:
```
quantileYear1 = games_df["Year"].quantile(0.25)
print(quantileYear1)
quantileYear2 = games_df["Year"].quantile(0.50)
print(quantileYear2)
quantileYear3 = games_df["Year"].quantile(0.75)
print(quantileYear3)
```

```
2003.0
2007.0
2010.0
```

In [41]:
```
quantileGlob1 = games_df["Global_Sales"].quantile(0.25)
print(quantileGlob1)
quantileGlob2 = games_df["Global_Sales"].quantile(0.50)
print(quantileGlob2)
quantileGlob3 = games_df["Global_Sales"].quantile(0.75)
print(quantileGlob3)
```

```
0.06
0.17
0.48
```

In [17]:
```
quantileYear1, quantileYear3 = np.percentile(games_df['Year'], [75 ,25])
iqr = quantileYear1 - quantileYear3
iqr
```

Out[17]: 7.0

In [42]:
```
quantileGlob1, quantileGlob3 = np.percentile(games_df['Global_Sales'], [75 ,25])
iqr = quantileGlob1 - quantileGlob3
iqr
```

Out[42]: 0.42

In [18]:
```
print("rangeYear", games_df.Year.max()-games_df.Year.min())
```

```
rangeYear 39.0
```

In [43]:
```
print("rangeGlobal_Sales", games_df.Global_Sales.max()-games_df.Global_Sales.min())
```

```
rangeGlobal_Sales 82.72999999999999
```

In [28]:
```
games_df["Year"].skew()
```

Out[28]: -1.0147905622588844

In [44]:
```
games_df["Global_Sales"].skew()
```

Out[44]: 17.288035948097253

In [19]:
```
games_df['Year'].describe()
```

Out[19]:
```
count    16334.000000
mean      2006.399473
std          5.835279
min       1977.000000
25%       2003.000000
50%       2007.000000
75%       2010.000000
max       2016.000000
Name: Year, dtype: float64
```

In [45]:
```
games_df['Global_Sales'].describe()
```

Out[45]:
```
count    16334.000000
mean         0.541798
std          1.566491
min          0.010000
25%          0.060000
50%          0.170000
75%          0.480000
max         82.740000
Name: Global_Sales, dtype: float64
```

First Analise:

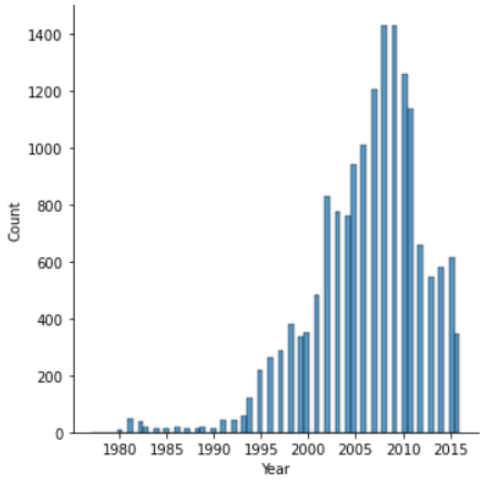
In my variable Year, the mean is 2006, the mode is 2009 and the median is 2007.

And in the Variable Global_Sales, the mean is 0.54, the mode 0.02 and the median is 0.17.

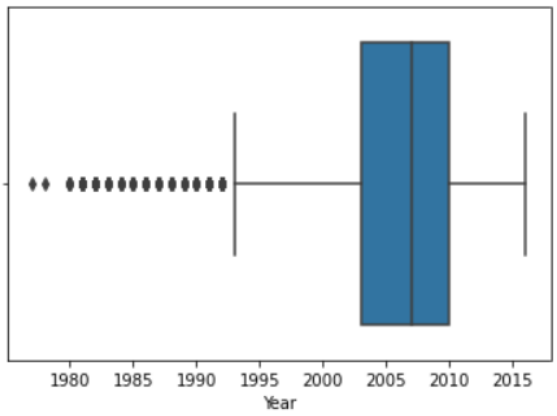
Analysing only the mean and median in the variable Year and Global_Sales of my dataset, I already can deduce in the Variable Year as mean<median and in the Variable Global_Sales as the mean>median I might have outliers.

Plots to show the dispersion in the variables

```
In [81]: sns.displot(games_df['Year'])
Out[81]: <seaborn.axisgrid.FacetGrid at 0x1f941b8c0d0>
```

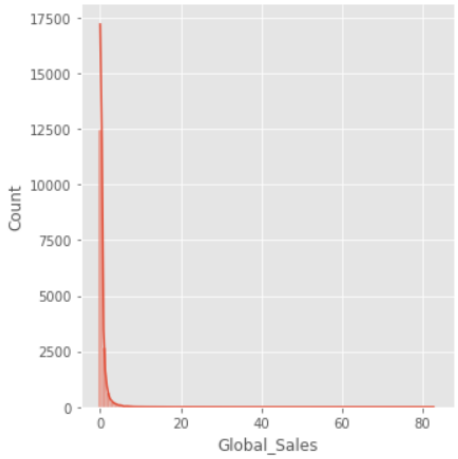



```
In [82]: sns.boxplot(x=games_df["Year"])
Out[82]: <AxesSubplot:xlabel='Year'>
```



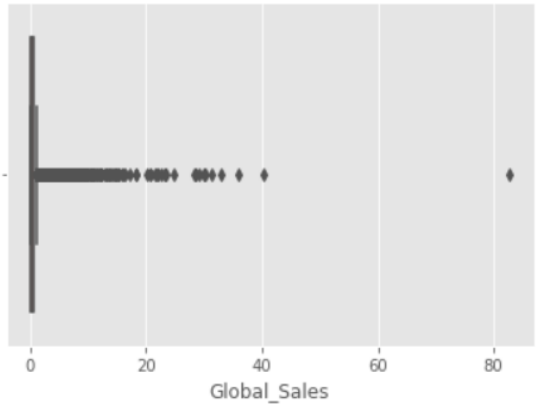
In [46]:
```
sns.displot(games_df['Global_Sales'], kde=True, discrete=True)
```

Out[46]: <seaborn.axisgrid.FacetGrid at 0x2127b14d3a0>



In [47]:
```
sns.boxplot(x=games_df["Global_Sales"])
```

Out[47]: <AxesSubplot:xlabel='Global_Sales'>



Analysis of Variables Picked

No, the variable “Year” as previously calculated has a Negative Skew distribution as the central tendency measures are different and mean<median.

And the Variable “Global_Sales” as previously calculated has a Positive Skew distribution, as the central tendency measures are different and mean>median.

```
In [11]: meanYear = games_df["Year"].mean()
         print(meanYear)
         modeYear = games_df["Year"].mode()
         print(modeYear)
         medianYear = games_df["Year"].median()
         print(medianYear)
```

```
2006.3994734908779
0    2009.0
dtype: float64
2007.0
```

```
In [28]: games_df["Year"].skew()
```

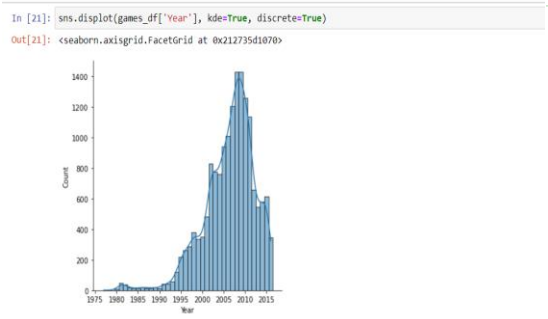
```
Out[28]: -1.0147905622588844
```

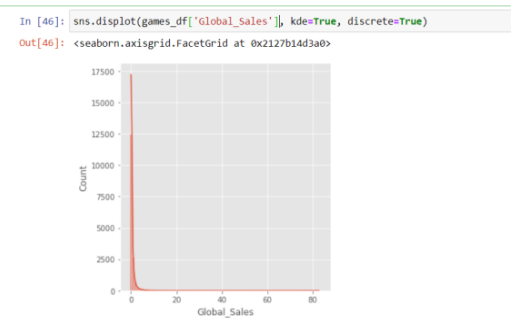
```
In [37]: meanGlob = games_df["Global_Sales"].mean()
         print(meanGlob)
         modeGlob = games_df["Global_Sales"].mode()
         print(modeGlob)
         medianGlob = games_df["Global_Sales"].median()
         print(medianGlob)
```

```
0.5417980898739431
0    0.02
dtype: float64
0.17
```

```
In [44]: games_df["Global_Sales"].skew()
```

```
Out[44]: 17.288035948097253
```



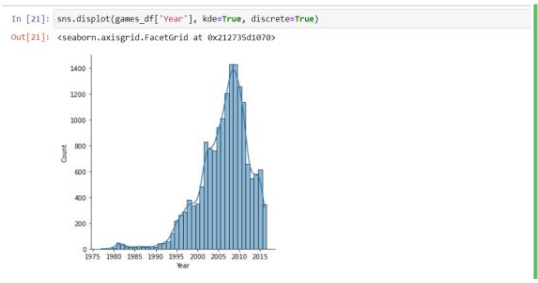


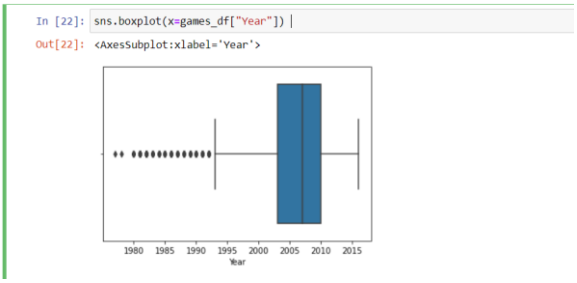
No, in the variables picked has not missing value, because I cleaned the missing Values before starting my analyse.

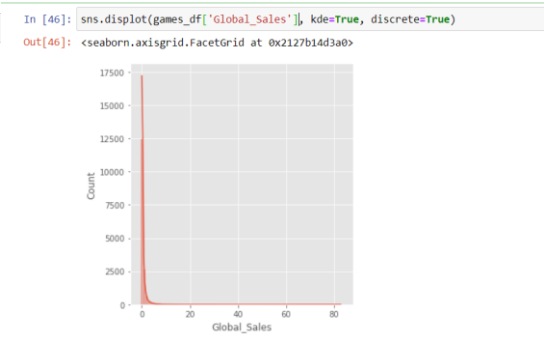
```
In [7]: games_df.isnull().sum()

Out[7]: Rank            0
        Name            0
        Platform        0
        Year          259
        Genre           0
        Publisher       0
        NA_Sales        0
        EU_Sales        0
        JP_Sales        0
        Other_Sales     0
        Global_Sales    0
        dtype: int64
```

```
In [8]: games_df.dropna(inplace=True)
        games_df.drop(games_df[games_df['Year']>2016].index, inplace=True)
        games_df.isnull().sum()

Out[8]: Rank            0
        Name            0
        Platform        0
        Year            0
        Genre           0
        Publisher       0
        NA_Sales        0
        EU_Sales        0
        JP_Sales        0
        Other_Sales     0
        Global_Sales    0
        dtype: int64
```

As both variables do not follow a symmetrical distribution but follows a Positive and Negative Skew, we can conclude that is consequences of outliers in the Variables picked.







Conclusion

As a result of this report, we were able to conclude that descriptive statistics is a summary statistic that quantitatively describes or summarizes characteristics of a collection of information, while descriptive statistics is the process of using and analysing these statistics through Measures of Frequency, Measures of Central Tendency, Measures of Dispersion or Variation, Measures of Position.

References

Works Cited

[1] Academy, Khan. "Shapes of Distributions (Video)." *Khan Academy*, 12 Feb. 2015, www.khanacademy.org/math/cc-sixth-grade-math/cc-6th-data-statistics/cc-6-shape-of-data/v/shapes-of-distributions. Accessed 16 Mar. 2022.

[2] Kaur P, Stoltzfus J, Yellapu V. "Descriptive Statistics. Int J Acad Med." *IJAM - International Journal of Academic Medicine*, 2018, www.ijam-web.org/text.asp?2018/4/1/60/230853. Accessed 8 Mar. 2022.

[3] Manikandan, S. "Measures of Central Tendency: Median and Mode." *Journal of Pharmacology and Pharmacotherapeutics*, vol. 2, no. 3, 2011, p. 214, www.ncbi.nlm.nih.gov/pmc/articles/PMC3157145/, 10.4103/0976-500x.83300. Accessed 19 Nov. 2019.

[4] McCluskey, Anthony, and Abdul Ghaaliq Lalkhen. "Statistics II: Central Tendency and Spread of Data." *Continuing Education in Anaesthesia Critical Care & Pain*, vol. 7, no. 4, Aug. 2007, pp. 127–130, www.bjaed.org/action/showPdf?pii=S1743-1816%2817%2930353-0, 10.1093/bjaceaccp/mkm020. Accessed 8 Mar. 2022.

[5] Mcleod, Saul. "What Is a Normal Distribution in Statistics?" *Simplypsychology.org*, Simply Psychology, 28 May 2019, www.simplypsychology.org/normal-distribution.html.

[6] Statistics intro: Mean, median, & mode. "Statistics Intro: Mean, Median, & Mode." *Khan Academy*, 2019, www.khanacademy.org/math/ap-statistics/summarizing-quantitative-data-ap/measuring-center-quantitative/v/statistics-intro-mean-median-and-mode. Accessed 8 Mar. 2022.

[7] Stephanie. “Bimodal Distribution.” *Statistics How To.*, 29 July 2013, www.statisticshowto.com/what-is-a-bimodal-distribution/#:~:text=The%20%E2%80%9Cmode%E2%80%9D%20in%20bimodal%20distribution. Accessed 8 Mar. 2022.