D214 Data Analytics Capstone Performance Task 3

Sean Simmons

WDU Data Analytics

MSDA D214

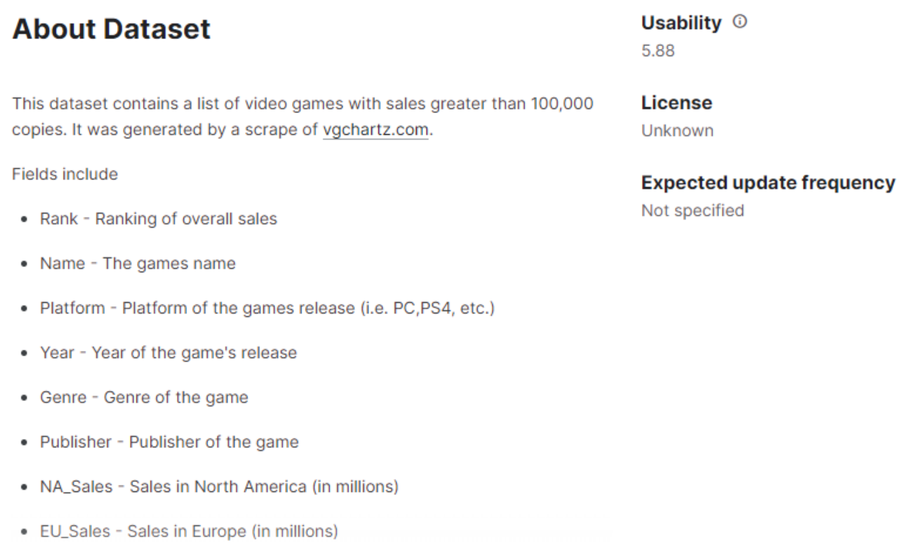
March 2023

Executive Summary and Implications

A. Develop an executive summary using your data and results from task 2. The summary should be written for a *technical audience* of your data-analytics peers or members of your team and should include *each* of the following:

- **a statement of the problem and the hypothesis**
- **a summary of the data-analysis process**
- **an outline of the findings**
- **an explanation of the limitations of the techniques and tools used**
- **a summary of proposed actions**
- **expected benefits of the study (be as specific and quantitative as possible)**

The Data



About Dataset

This dataset contains a list of video games with sales greater than 100,000 copies. It was generated by a scrape of vgchartz.com.

Fields include

- Rank - Ranking of overall sales
- Name - The games name
- Platform - Platform of the games release (i.e. PC,PS4, etc.)
- Year - Year of the game's release
- Genre - Genre of the game
- Publisher - Publisher of the game
- NA_Sales - Sales in North America (in millions)
- EU_Sales - Sales in Europe (in millions)

Usability
5.88

License
Unknown

Expected update frequency
Not specified

We have a dataset that shows global sales for video games and several characteristics, explained by the original author of the dataset, Smith, in the above screenshot (*Video Game Sales)*. This dataset was downloaded and then uploaded as a dataframe in my python environment.

Statement of the problem and hypothesis

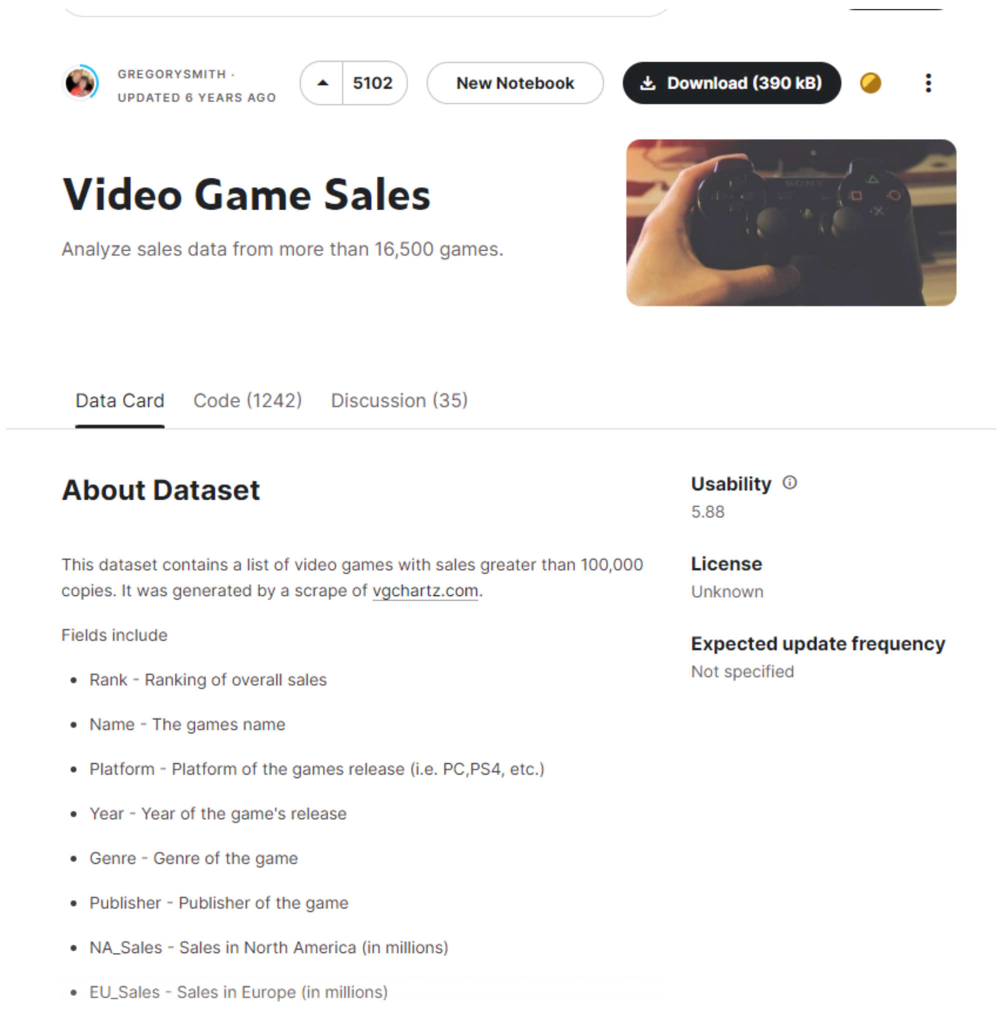
The business needs a multiple linear regression (MLR) model to predict future global sales. The question becomes, can we create a MLR model to predict future sales for our organization? Which brings us to our two hypotheses:

Null hypothesis-. We can not create a model to statistically significantly predict video game global sales.

Alternate Hypothesis- We can create a model to statistically significantly predict video game global sales.

Summary of the data analysis process

In order to extract the data, you need to go to kaggle.com and create an account. Then, I searched for viable datasets and found the video game sales dataset pictured on the next page. I clicked the download button and downloaded the dataset to my device.



Video Game Sales

Analyze sales data from more than 16,500 games.

About Dataset

This dataset contains a list of video games with sales greater than 100,000 copies. It was generated by a scrape of vgchartz.com.

Fields include

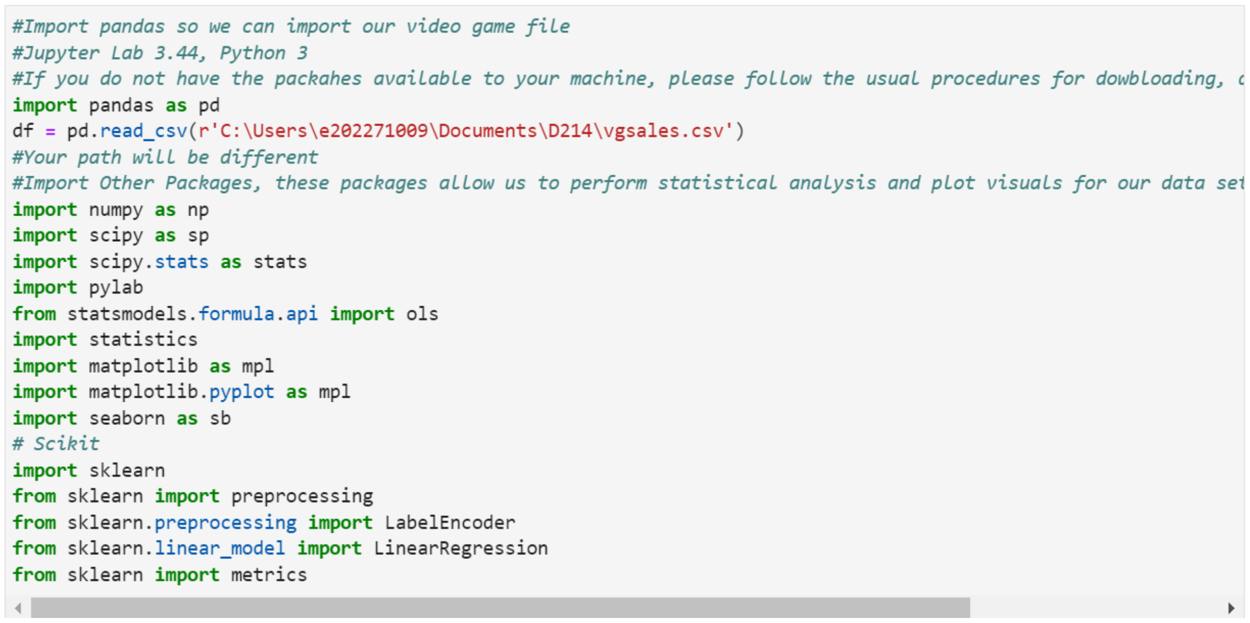
- Rank - Ranking of overall sales
- Name - The games name
- Platform - Platform of the games release (i.e. PC,PS4, etc.)
- Year - Year of the game's release
- Genre - Genre of the game
- Publisher - Publisher of the game
- NA_Sales - Sales in North America (in millions)
- EU_Sales - Sales in Europe (in millions)

Usability
5.88

License
Unknown

Expected update frequency
Not specified

I used kaggle because it is a widely known and available trusted source of datasets. This specific dataset is also well explained and easy to understand. Smith, the original author, listed no necessary citations for this work, but I have included an APA citation of the dataset, nonetheless. The descriptions of each column can be found in the kaggle site to help in our data exploration.

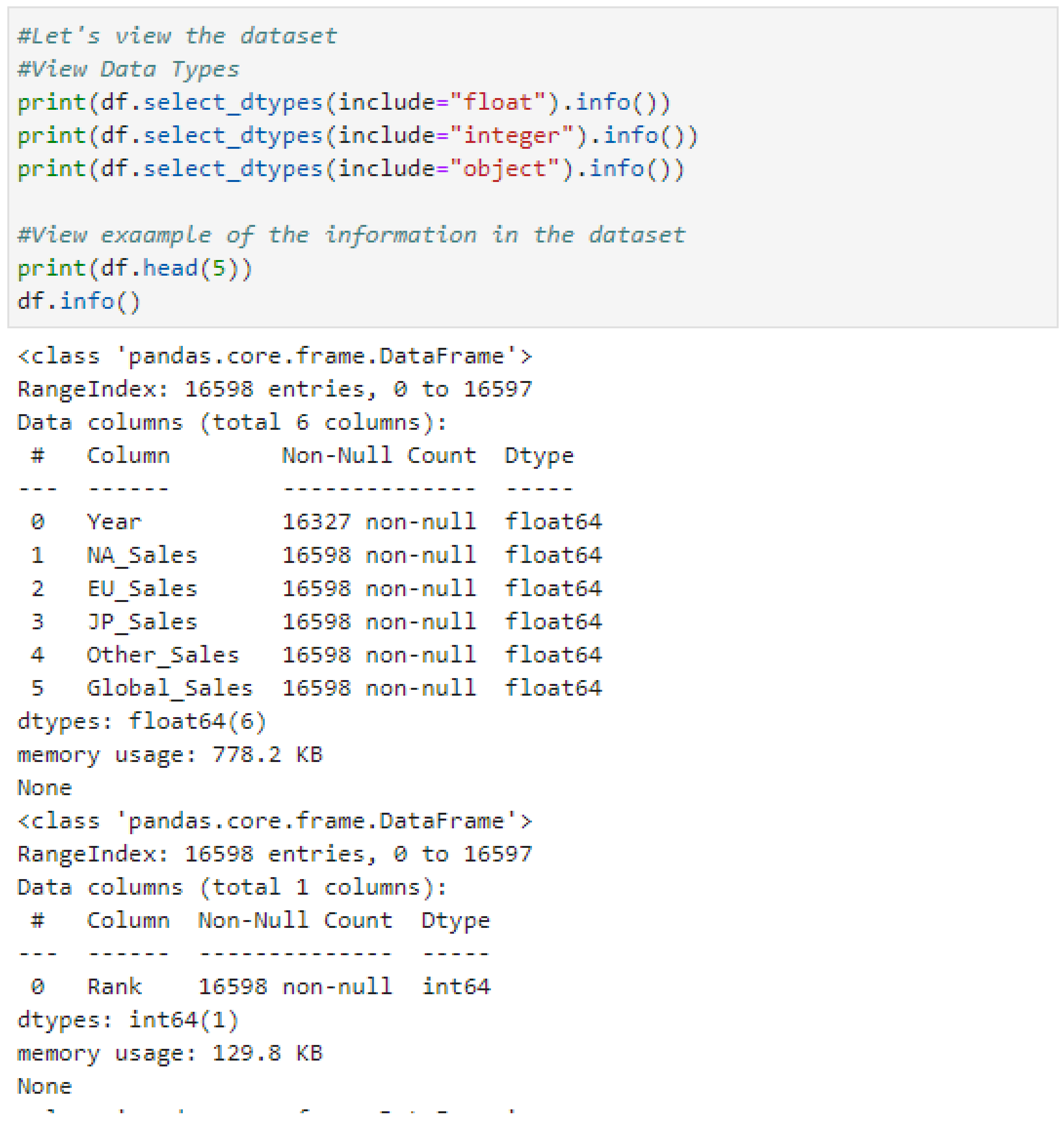


```
#Import pandas so we can import our video game file
#Jupyter Lab 3.44, Python 3
#If you do not have the packahes available to your machine, please follow the usual procedures for dowbloading, c
import pandas as pd
df = pd.read_csv(r'C:\Users\e202271009\Documents\D214\vgsales.csv')
#Your path will be different
#Import Other Packages, these packages allow us to perform statistical analysis and plot visuals for our data set
import numpy as np
import scipy as sp
import scipy.stats as stats
import pylab
from statsmodels.formula.api import ols
import statistics
import matplotlib as mpl
import matplotlib.pyplot as mpl
import seaborn as sb
# Scikit
import sklearn
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn import metrics
```

I am using python version 3 in Jupyter labs version 3.44. To import the data pandas had to be imported. I chose to use this coding environment and pandas for the same reason- they are widely accepted as an easy and straightforward tool to perform a wide range of data analysis tasks (Prasanna 2021).

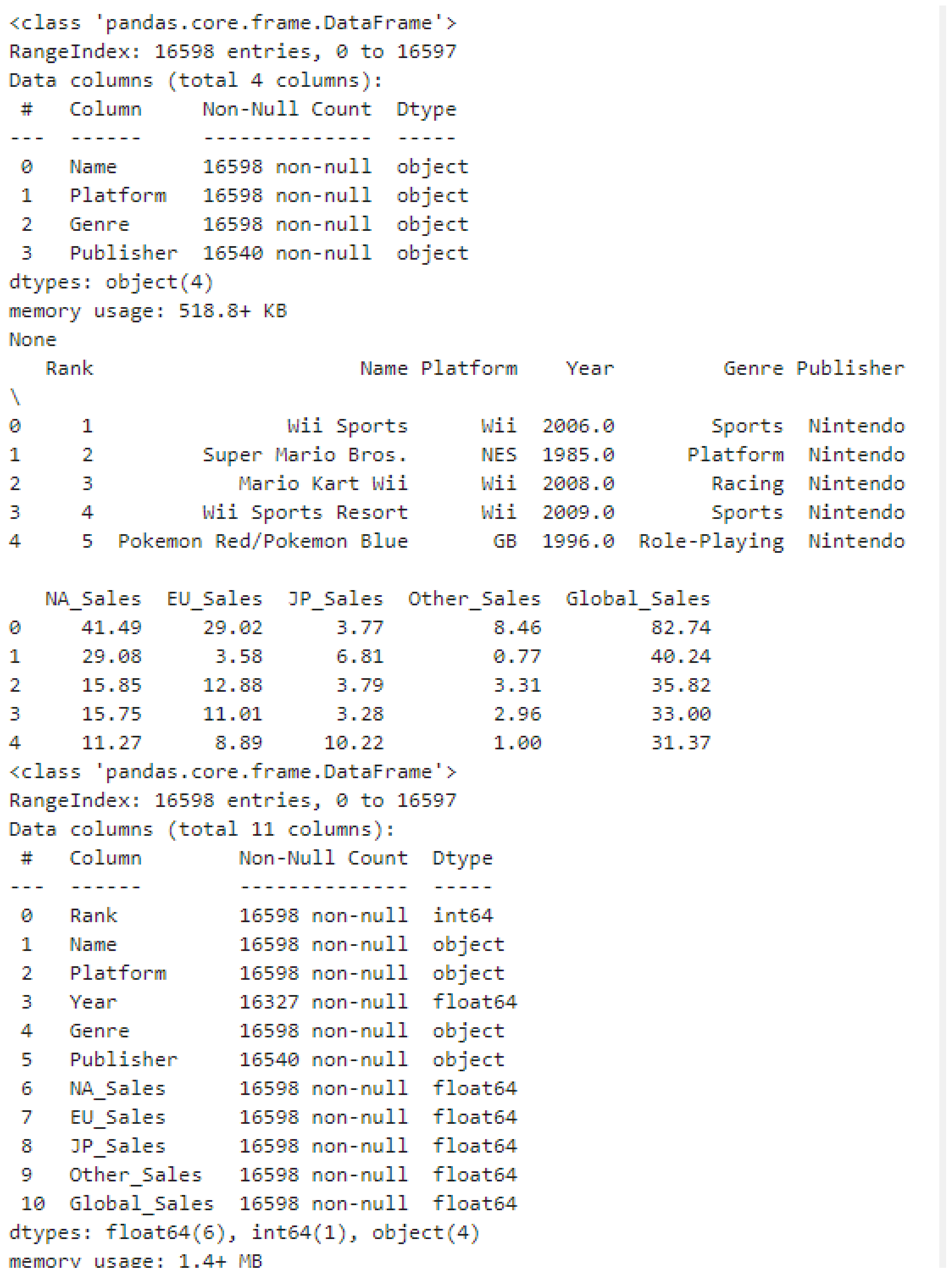
Data Preparation

Now that the data is uploaded into my coding environment, we need to view the raw data for exploratory and information gathering needs.We will gather the data types, first 5 rows and counts. This is a beginning step in our exploratory data analysis (EDA).



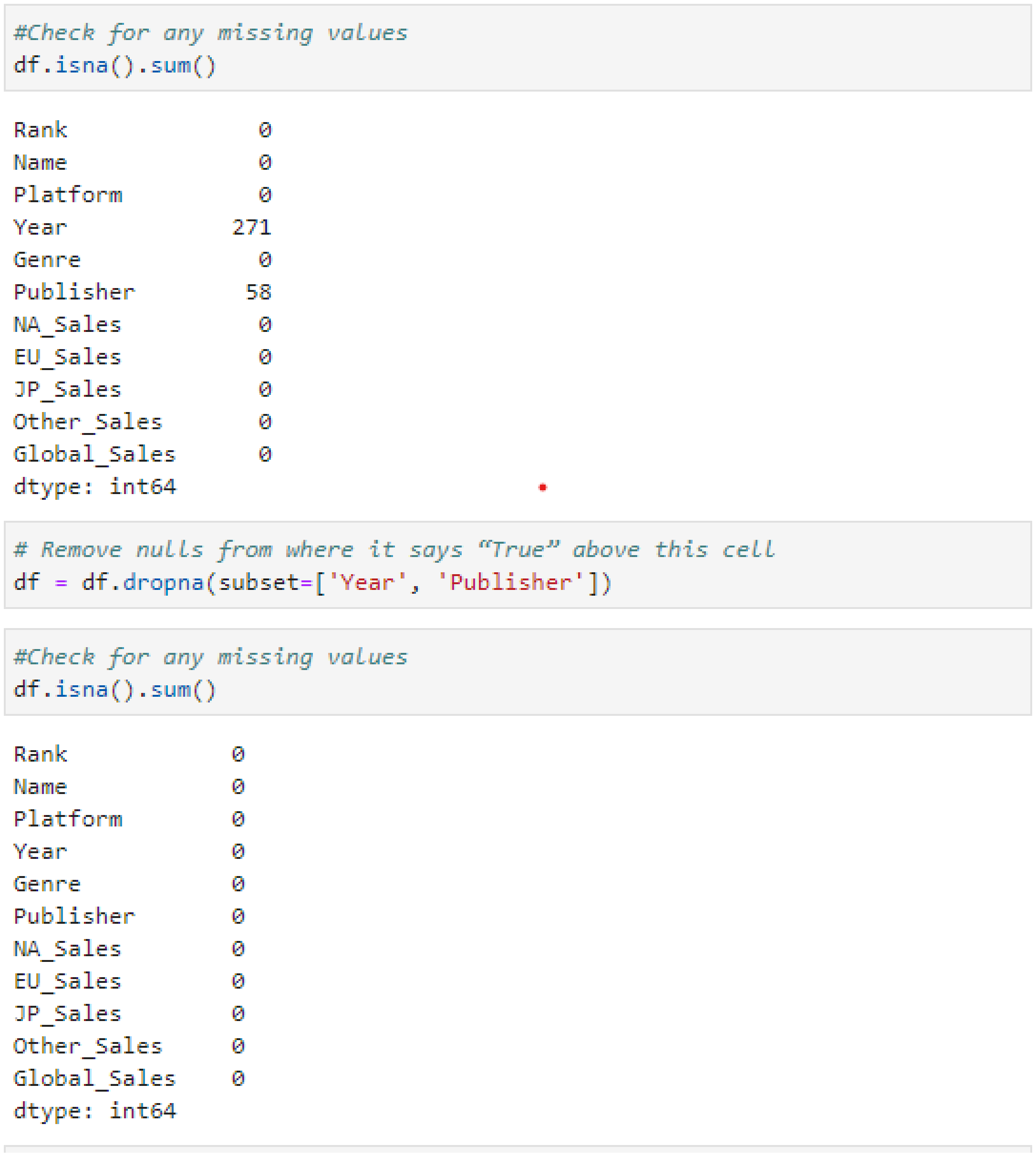
```python
#Let's view the dataset
#View Data Types
print(df.select_dtypes(include="float").info())
print(df.select_dtypes(include="integer").info())
print(df.select_dtypes(include="object").info())

#View exaample of the information in the dataset
print(df.head(5))
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Year          16327 non-null  float64
 1   NA_Sales      16598 non-null  float64
 2   EU_Sales      16598 non-null  float64
 3   JP_Sales      16598 non-null  float64
 4   Other_Sales   16598 non-null  float64
 5   Global_Sales  16598 non-null  float64
dtypes: float64(6)
memory usage: 778.2 KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Rank    16598 non-null  int64
dtypes: int64(1)
memory usage: 129.8 KB
None
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   Name       16598 non-null  object
 1   Platform   16598 non-null  object
 2   Genre      16598 non-null  object
 3   Publisher  16540 non-null  object
dtypes: object(4)
memory usage: 518.8+ KB
None
   Rank                      Name Platform    Year         Genre Publisher  \
0     1                Wii Sports      Wii  2006.0        Sports  Nintendo   
1     2         Super Mario Bros.      NES  1985.0      Platform  Nintendo   
2     3            Mario Kart Wii      Wii  2008.0        Racing  Nintendo   
3     4         Wii Sports Resort      Wii  2009.0        Sports  Nintendo   
4     5  Pokemon Red/Pokemon Blue       GB  1996.0  Role-Playing  Nintendo   

   NA_Sales  EU_Sales  JP_Sales  Other_Sales  Global_Sales  
0     41.49     29.02      3.77         8.46         82.74  
1     29.08      3.58      6.81         0.77         40.24  
2     15.85     12.88      3.79         3.31         35.82  
3     15.75     11.01      3.28         2.96         33.00  
4     11.27      8.89     10.22         1.00         31.37  
<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   Name          16598 non-null  object 
 2   Platform      16598 non-null  object 
 3   Year          16327 non-null  float64
 4   Genre         16598 non-null  object 
 5   Publisher     16540 non-null  object 
 6   NA_Sales      16598 non-null  float64
 7   EU_Sales      16598 non-null  float64
 8   JP_Sales      16598 non-null  float64
 9   Other_Sales   16598 non-null  float64
 10  Global_Sales  16598 non-null  float64
dtypes: float64(6), int64(1), object(4)
memory usage: 1.4+ MB
```

Now, we must explore the data for missing information. Based on the nature of the data a 0 is an acceptable value and we should not look to mitigate outliers, since we want all of the sales information represented in our multiple linear regression analysis. Later we will see any outliers and continue our decision of leaving them alone in our histograms, but if we find any null values we must fill them and then check for nulls again to verify (Prasanna 2021):



```
#Check for any missing values
df.isna().sum()
```

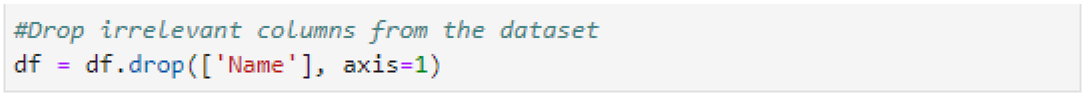
```
Rank              0
Name              0
Platform          0
Year            271
Genre             0
Publisher        58
NA_Sales          0
EU_Sales          0
JP_Sales          0
Other_Sales       0
Global_Sales      0
dtype: int64
```

```
# Remove nulls from where it says "True" above this cell
df = df.dropna(subset=['Year', 'Publisher'])
```

```
#Check for any missing values
df.isna().sum()
```

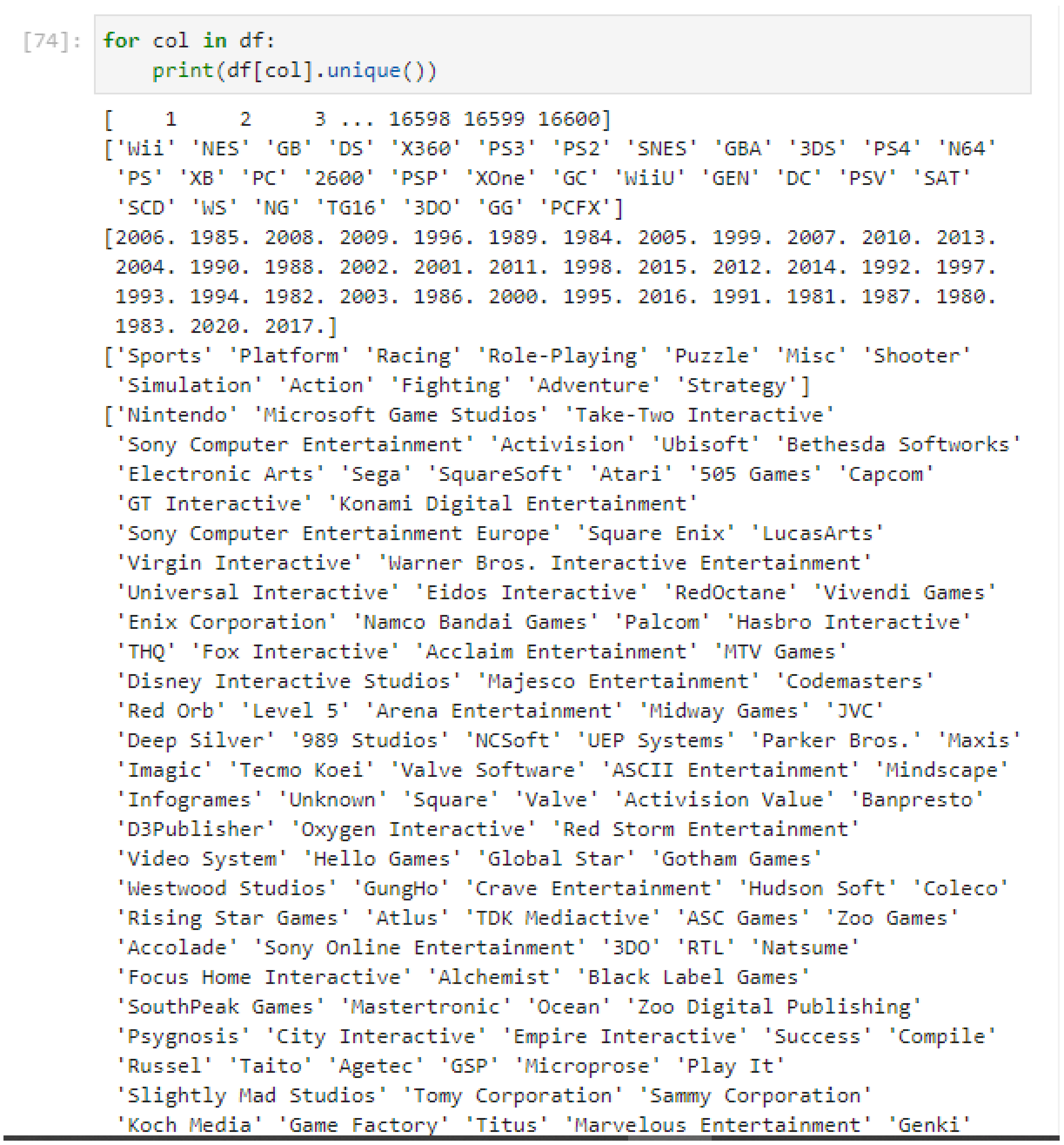
```
Rank            0
Name            0
Platform        0
Year            0
Genre           0
Publisher       0
NA_Sales        0
EU_Sales        0
JP_Sales        0
Other_Sales     0
Global_Sales    0
dtype: int64
```

Now I want to drop the only column that will not aid us in any way, the name column.



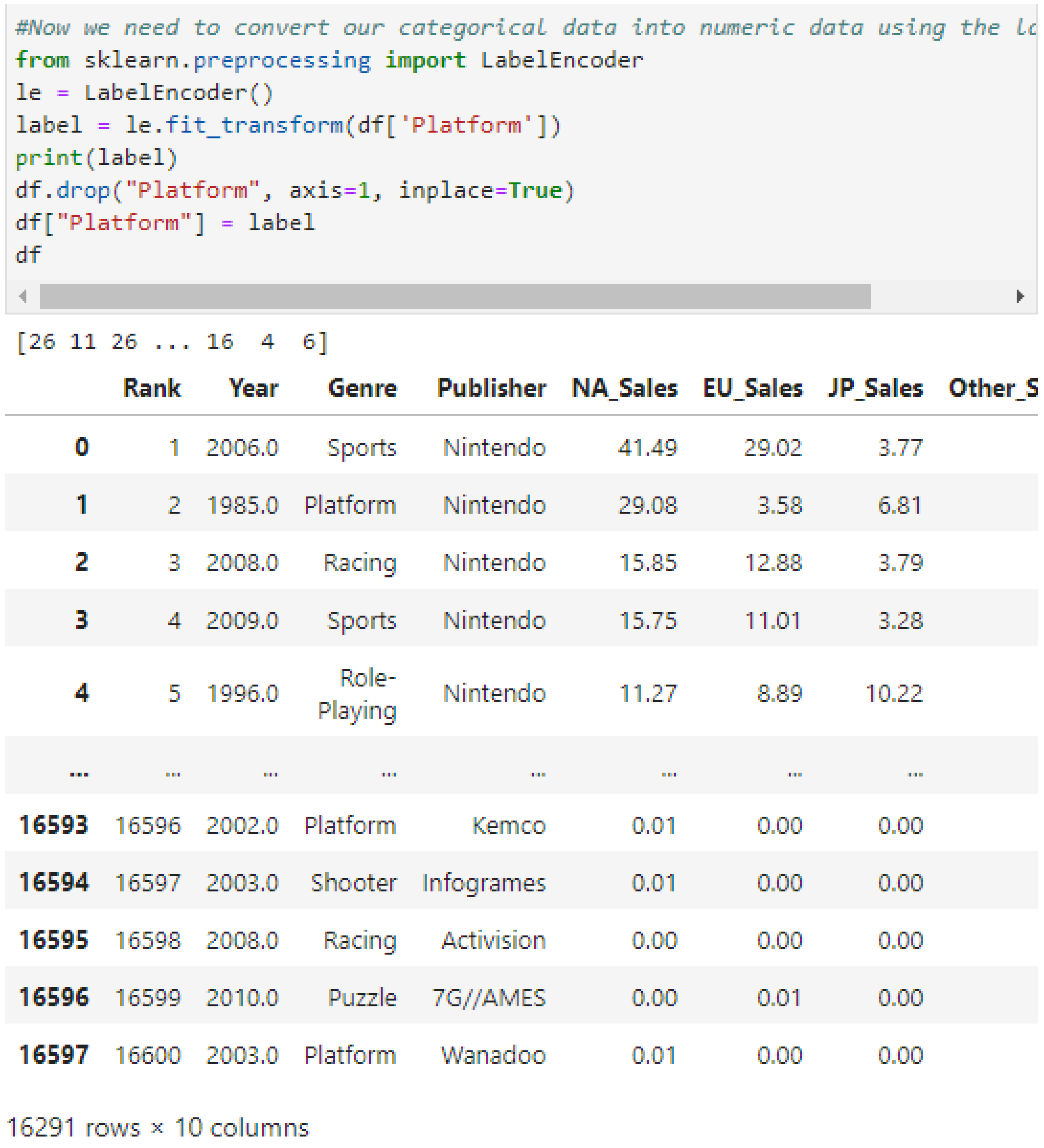
```
#Drop irrelevant columns from the dataset
df = df.drop(['Name'], axis=1)
```

Afterwards, we now need to change our data types. From the initial EDA I know we have a lot of string data. For our hypothesis and multiple linear regression (MLR) we need numerical data. We need to use labelencoder to convert each of our strings into numeric data.



```
[74]: for col in df:
          print(df[col].unique())
```

```
[    1     2     3 ... 16598 16599 16600]
['Wii' 'NES' 'GB' 'DS' 'X360' 'PS3' 'PS2' 'SNES' 'GBA' '3DS' 'PS4' 'N64'
 'PS' 'XB' 'PC' '2600' 'PSP' 'XOne' 'GC' 'WiiU' 'GEN' 'DC' 'PSV' 'SAT'
 'SCD' 'WS' 'NG' 'TG16' '3DO' 'GG' 'PCFX']
[2006. 1985. 2008. 2009. 1996. 1989. 1984. 2005. 1999. 2007. 2010. 2013.
 2004. 1990. 1988. 2002. 2001. 2011. 1998. 2015. 2012. 2014. 1992. 1997.
 1993. 1994. 1982. 2003. 1986. 2000. 1995. 2016. 1991. 1981. 1987. 1980.
 1983. 2020. 2017.]
['Sports' 'Platform' 'Racing' 'Role-Playing' 'Puzzle' 'Misc' 'Shooter'
 'Simulation' 'Action' 'Fighting' 'Adventure' 'Strategy']
['Nintendo' 'Microsoft Game Studios' 'Take-Two Interactive'
 'Sony Computer Entertainment' 'Activision' 'Ubisoft' 'Bethesda Softworks'
 'Electronic Arts' 'Sega' 'SquareSoft' 'Atari' '505 Games' 'Capcom'
 'GT Interactive' 'Konami Digital Entertainment'
 'Sony Computer Entertainment Europe' 'Square Enix' 'LucasArts'
 'Virgin Interactive' 'Warner Bros. Interactive Entertainment'
 'Universal Interactive' 'Eidos Interactive' 'RedOctane' 'Vivendi Games'
 'Enix Corporation' 'Namco Bandai Games' 'Palcom' 'Hasbro Interactive'
 'THQ' 'Fox Interactive' 'Acclaim Entertainment' 'MTV Games'
 'Disney Interactive Studios' 'Majesco Entertainment' 'Codemasters'
 'Red Orb' 'Level 5' 'Arena Entertainment' 'Midway Games' 'JVC'
 'Deep Silver' '989 Studios' 'NCSoft' 'UEP Systems' 'Parker Bros.' 'Maxis'
 'Imagic' 'Tecmo Koei' 'Valve Software' 'ASCII Entertainment' 'Mindscape'
 'Infogrames' 'Unknown' 'Square' 'Valve' 'Activision Value' 'Banpresto'
 'D3Publisher' 'Oxygen Interactive' 'Red Storm Entertainment'
 'Video System' 'Hello Games' 'Global Star' 'Gotham Games'
 'Westwood Studios' 'GungHo' 'Crave Entertainment' 'Hudson Soft' 'Coleco'
 'Rising Star Games' 'Atlus' 'TDK Mediactive' 'ASC Games' 'Zoo Games'
 'Accolade' 'Sony Online Entertainment' '3DO' 'RTL' 'Natsume'
 'Focus Home Interactive' 'Alchemist' 'Black Label Games'
 'SouthPeak Games' 'Mastertronic' 'Ocean' 'Zoo Digital Publishing'
 'Psygnosis' 'City Interactive' 'Empire Interactive' 'Success' 'Compile'
 'Russel' 'Taito' 'Agetec' 'GSP' 'Microprose' 'Play It'
 'Slightly Mad Studios' 'Tomy Corporation' 'Sammy Corporation'
 'Koch Media' 'Game Factory' 'Titus' 'Marvelous Entertainment' 'Genki'
```

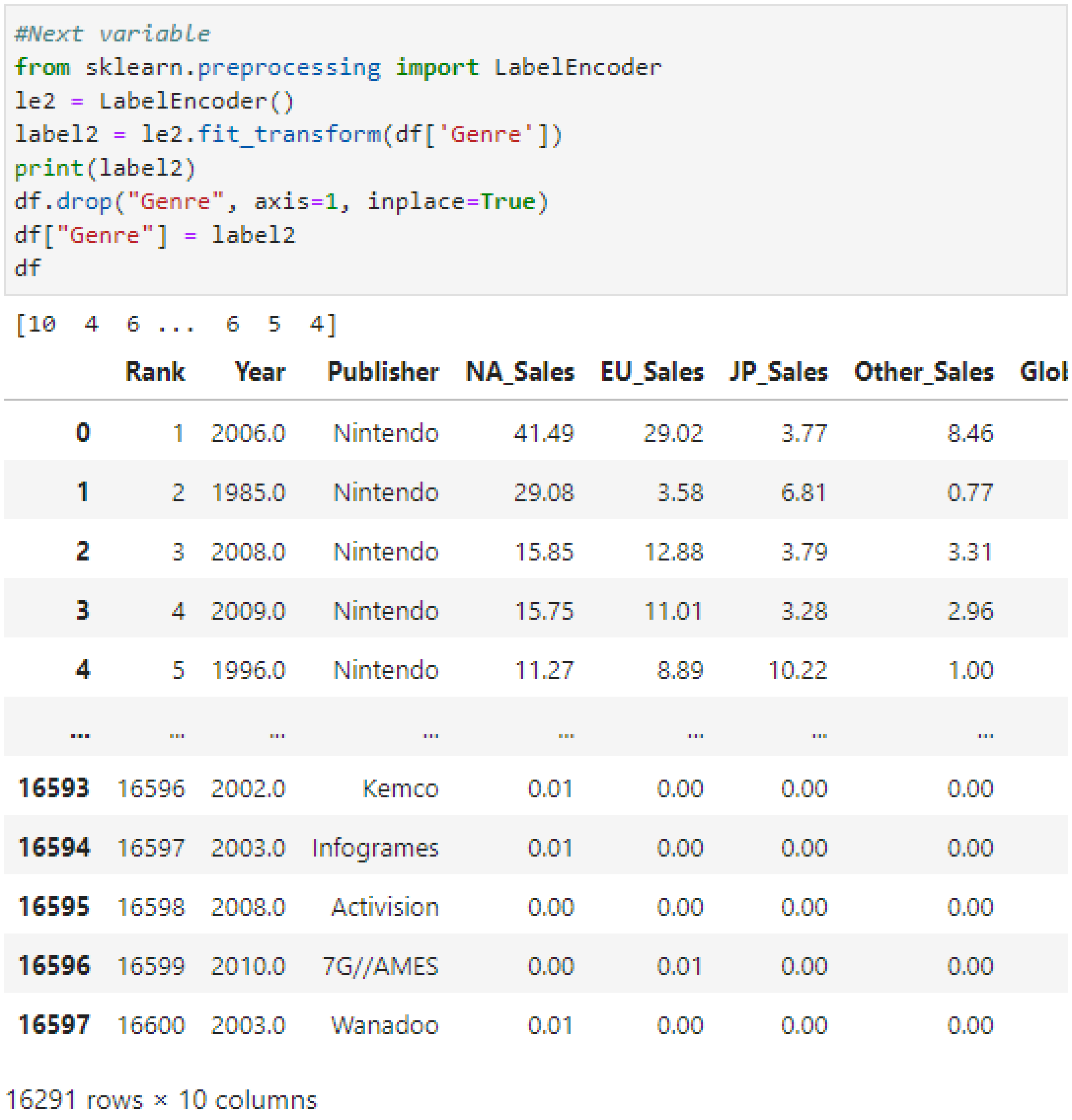


```
#Now we need to convert our categorical data into numeric data using the Lc
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
label = le.fit_transform(df['Platform'])
print(label)
df.drop("Platform", axis=1, inplace=True)
df["Platform"] = label
df
```

```
[26 11 26 ... 16  4  6]
```

	Rank	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_S
0	1	2006.0	Sports	Nintendo	41.49	29.02	3.77	
1	2	1985.0	Platform	Nintendo	29.08	3.58	6.81	
2	3	2008.0	Racing	Nintendo	15.85	12.88	3.79	
3	4	2009.0	Sports	Nintendo	15.75	11.01	3.28	
4	5	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	
...	...	...	...	...	...	...	...	
16593	16596	2002.0	Platform	Kemco	0.01	0.00	0.00	
16594	16597	2003.0	Shooter	Infogrames	0.01	0.00	0.00	
16595	16598	2008.0	Racing	Activision	0.00	0.00	0.00	
16596	16599	2010.0	Puzzle	7G//AMES	0.00	0.01	0.00	
16597	16600	2003.0	Platform	Wanadoo	0.01	0.00	0.00	

16291 rows × 10 columns

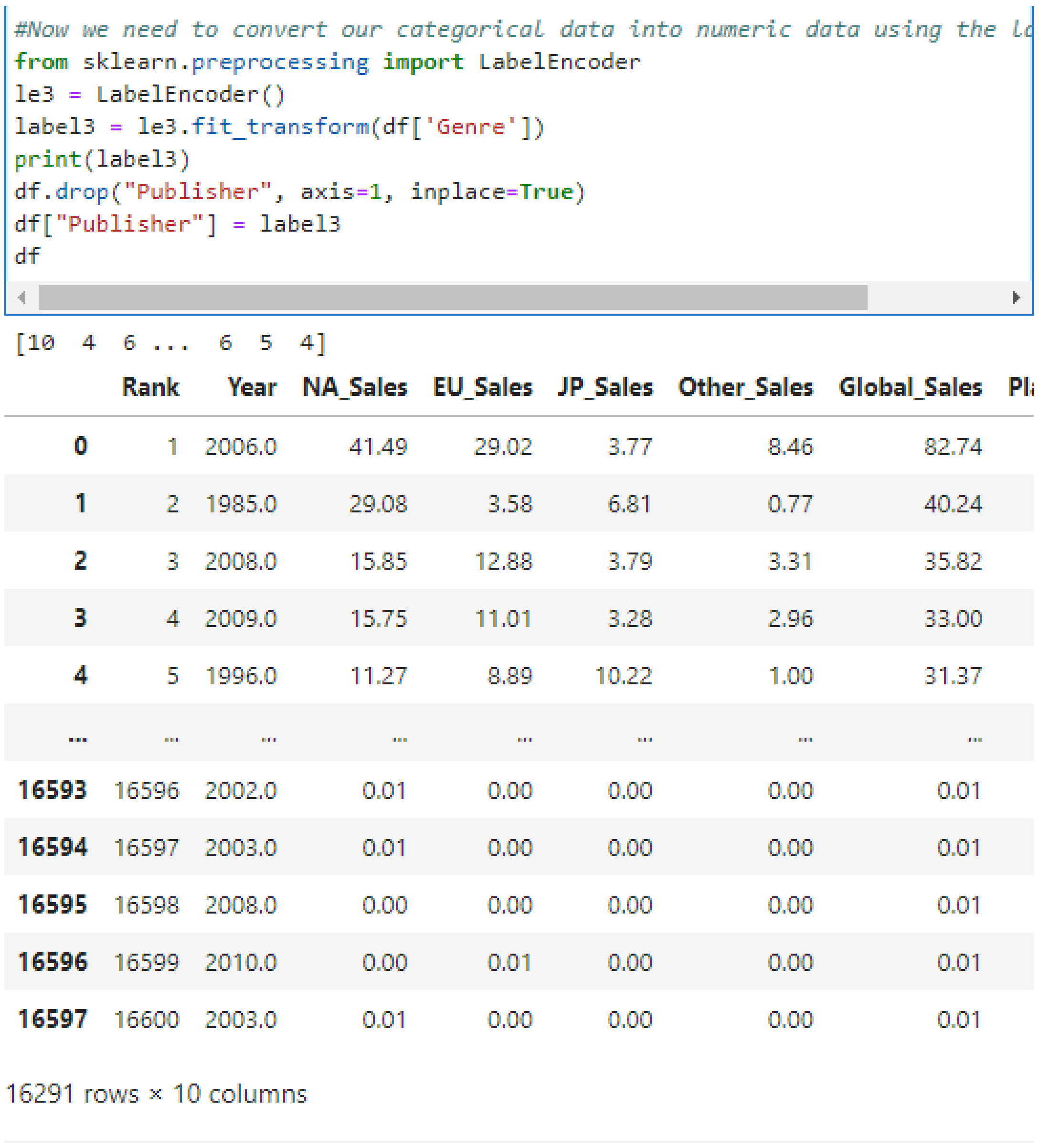


```python
#Next variable
from sklearn.preprocessing import LabelEncoder
le2 = LabelEncoder()
label2 = le2.fit_transform(df['Genre'])
print(label2)
df.drop("Genre", axis=1, inplace=True)
df["Genre"] = label2
df
```

```
[10  4  6 ...  6  5  4]
```

	Rank	Year	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Glol
0	1	2006.0	Nintendo	41.49	29.02	3.77	8.46	
1	2	1985.0	Nintendo	29.08	3.58	6.81	0.77	
2	3	2008.0	Nintendo	15.85	12.88	3.79	3.31	
3	4	2009.0	Nintendo	15.75	11.01	3.28	2.96	
4	5	1996.0	Nintendo	11.27	8.89	10.22	1.00	
...	...	...	...	...	...	...	...	
16593	16596	2002.0	Kemco	0.01	0.00	0.00	0.00	
16594	16597	2003.0	Infogrames	0.01	0.00	0.00	0.00	
16595	16598	2008.0	Activision	0.00	0.00	0.00	0.00	
16596	16599	2010.0	7G//AMES	0.00	0.01	0.00	0.00	
16597	16600	2003.0	Wanadoo	0.01	0.00	0.00	0.00	

16291 rows × 10 columns



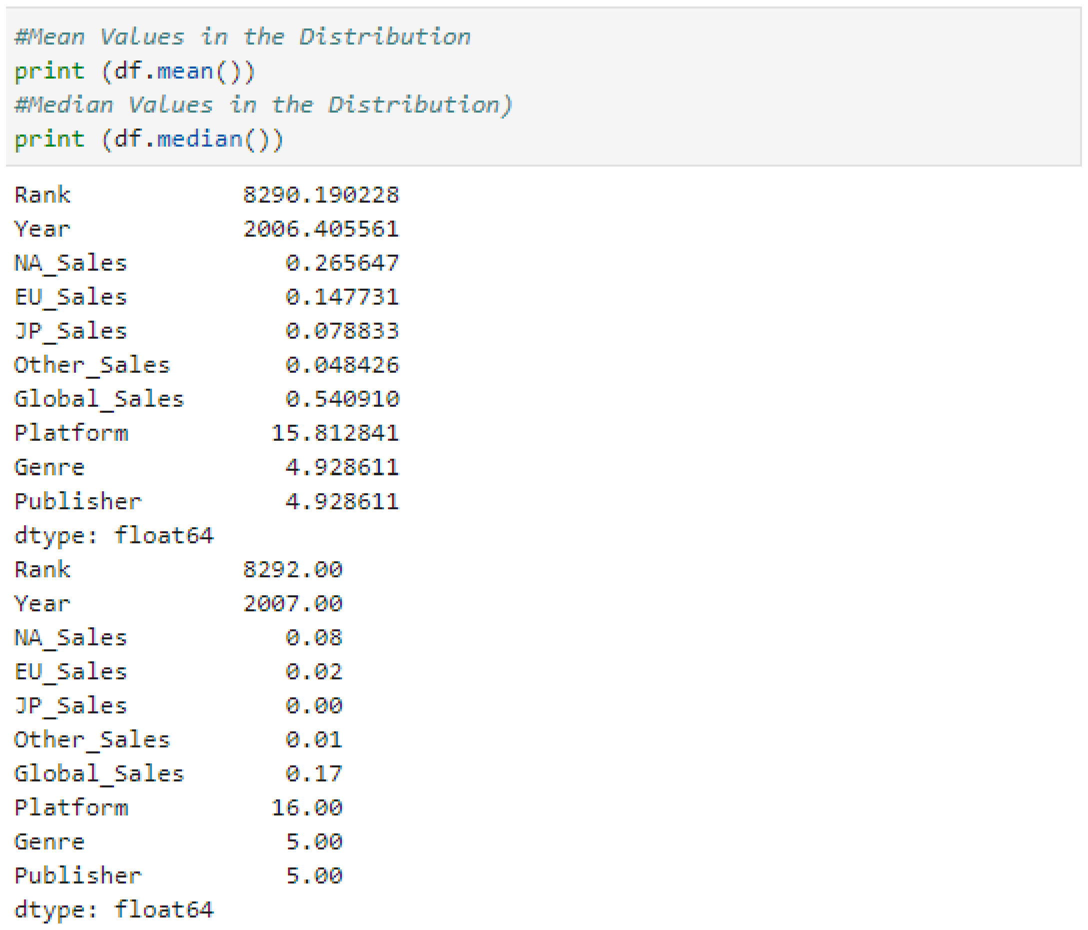
```
#Now we need to convert our categorical data into numeric data using the Lc
from sklearn.preprocessing import LabelEncoder
le3 = LabelEncoder()
label3 = le3.fit_transform(df['Genre'])
print(label3)
df.drop("Publisher", axis=1, inplace=True)
df["Publisher"] = label3
df
```

```
[10  4  6 ...  6  5  4]
```

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Pl
0	1	2006.0	41.49	29.02	3.77	8.46	82.74	
1	2	1985.0	29.08	3.58	6.81	0.77	40.24	
2	3	2008.0	15.85	12.88	3.79	3.31	35.82	
3	4	2009.0	15.75	11.01	3.28	2.96	33.00	
4	5	1996.0	11.27	8.89	10.22	1.00	31.37	
...	...	...	...	...	...	...	...	
16593	16596	2002.0	0.01	0.00	0.00	0.00	0.01	
16594	16597	2003.0	0.01	0.00	0.00	0.00	0.01	
16595	16598	2008.0	0.00	0.00	0.00	0.00	0.01	
16596	16599	2010.0	0.00	0.01	0.00	0.00	0.01	
16597	16600	2003.0	0.01	0.00	0.00	0.00	0.01	

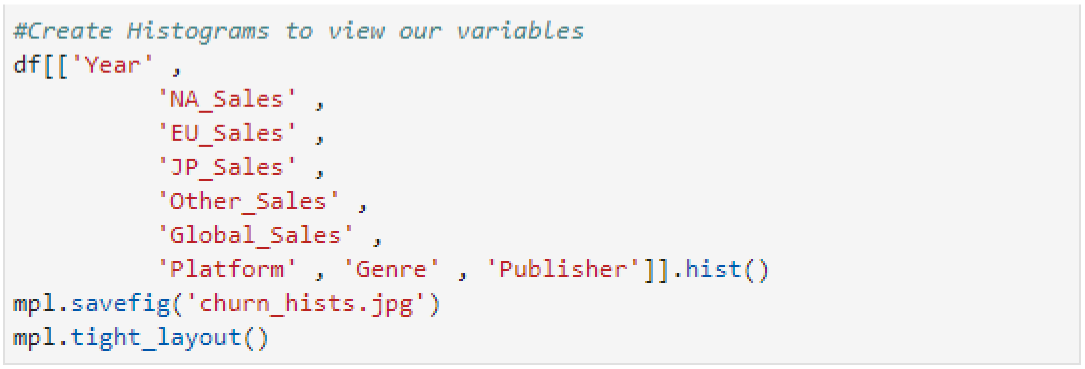
16291 rows × 10 columns

Then we performed EDA through visualizing our data in histograms, boxplots, and scatter plots. We also pulled the means and medians of our dataset. One disadvantage of this is that it can be too much information and it can mislead our ideas of what the final MLR model should look like. The trends and relationships identified here may not present itself in the model in the same way we see it. For example, the boxplots show several outliers for many variables, but we have to remember we are choosing to keep all data points in our dataset to preserve the integrity of the data. Although, this is a very small mental disadvantage and can be ignored by remaining unbiased:

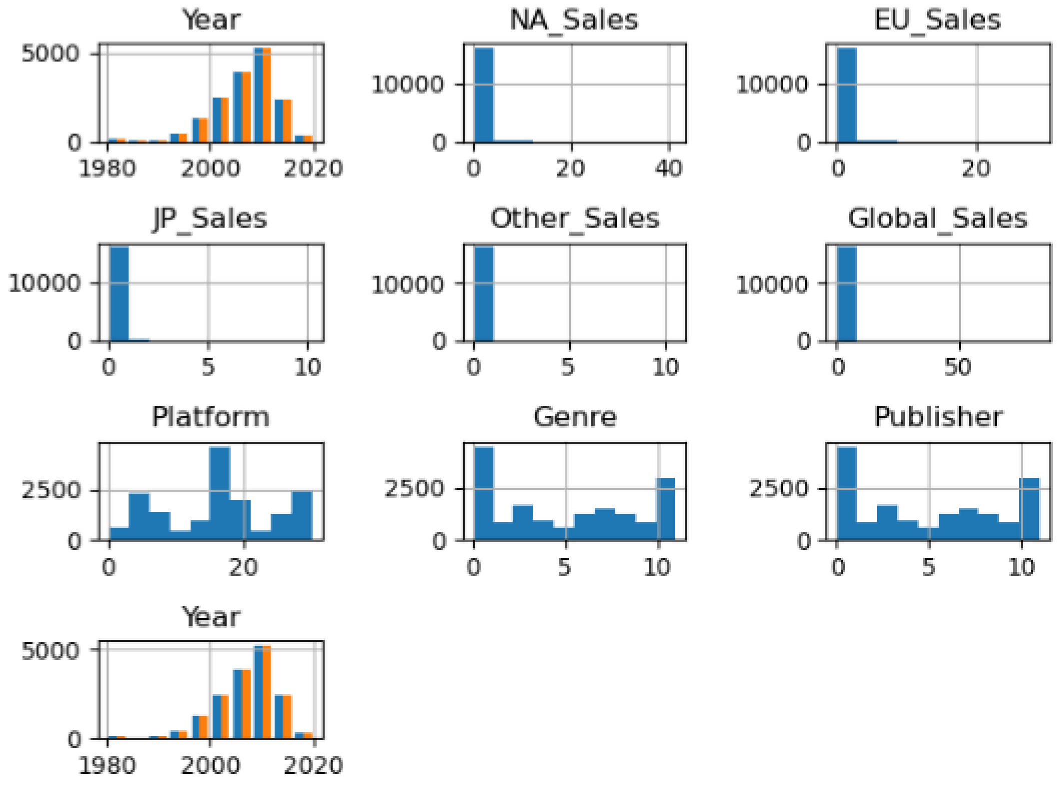


```
#Mean Values in the Distribution
print (df.mean())
#Median Values in the Distribution)
print (df.median())
```

```
Rank            8290.190228
Year            2006.405561
NA_Sales           0.265647
EU_Sales           0.147731
JP_Sales           0.078833
Other_Sales        0.048426
Global_Sales       0.540910
Platform          15.812841
Genre              4.928611
Publisher          4.928611
dtype: float64
Rank            8292.00
Year            2007.00
NA_Sales           0.08
EU_Sales           0.02
JP_Sales           0.00
Other_Sales        0.01
Global_Sales       0.17
Platform          16.00
Genre              5.00
Publisher          5.00
dtype: float64
```

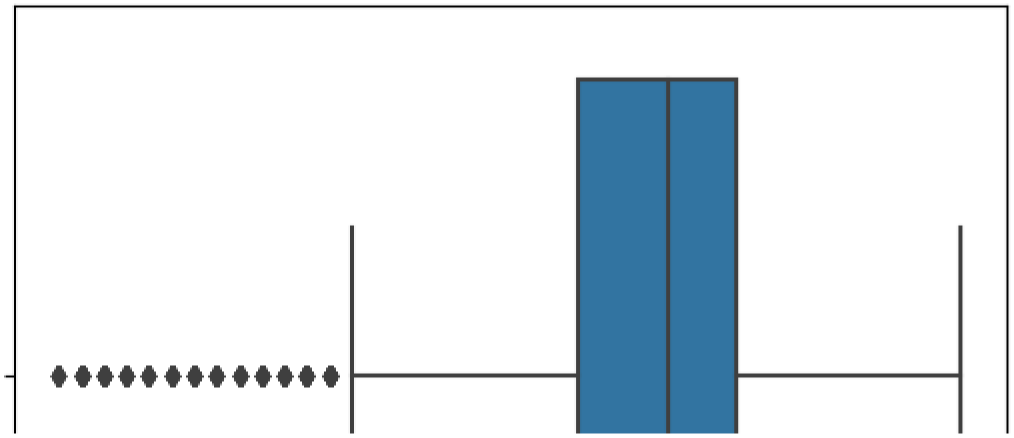


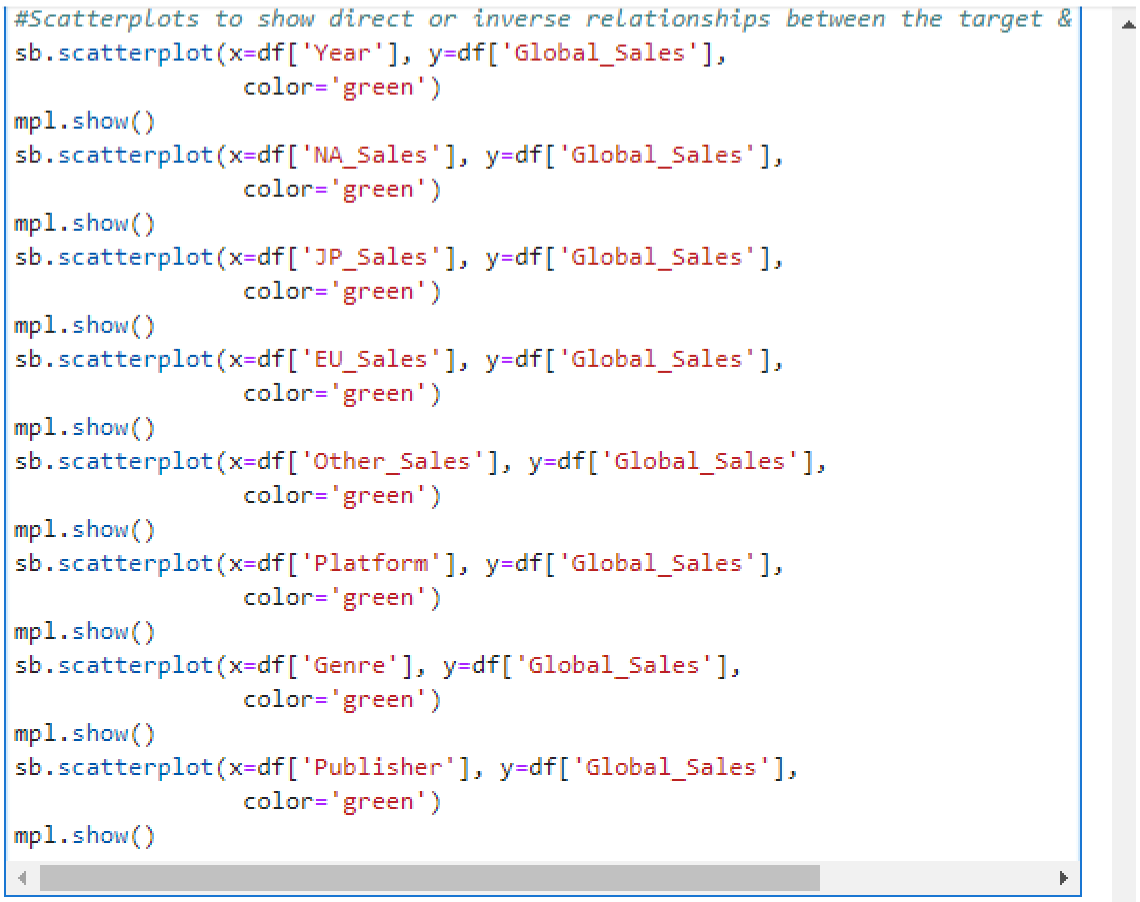
```
#Create Histograms to view our variables
df[['Year' ,
          'NA_Sales' ,
          'EU_Sales' ,
          'JP_Sales' ,
          'Other_Sales' ,
          'Global_Sales' ,
          'Platform' , 'Genre' , 'Publisher']].hist()
mpl.savefig('churn_hists.jpg')
mpl.tight_layout()
```



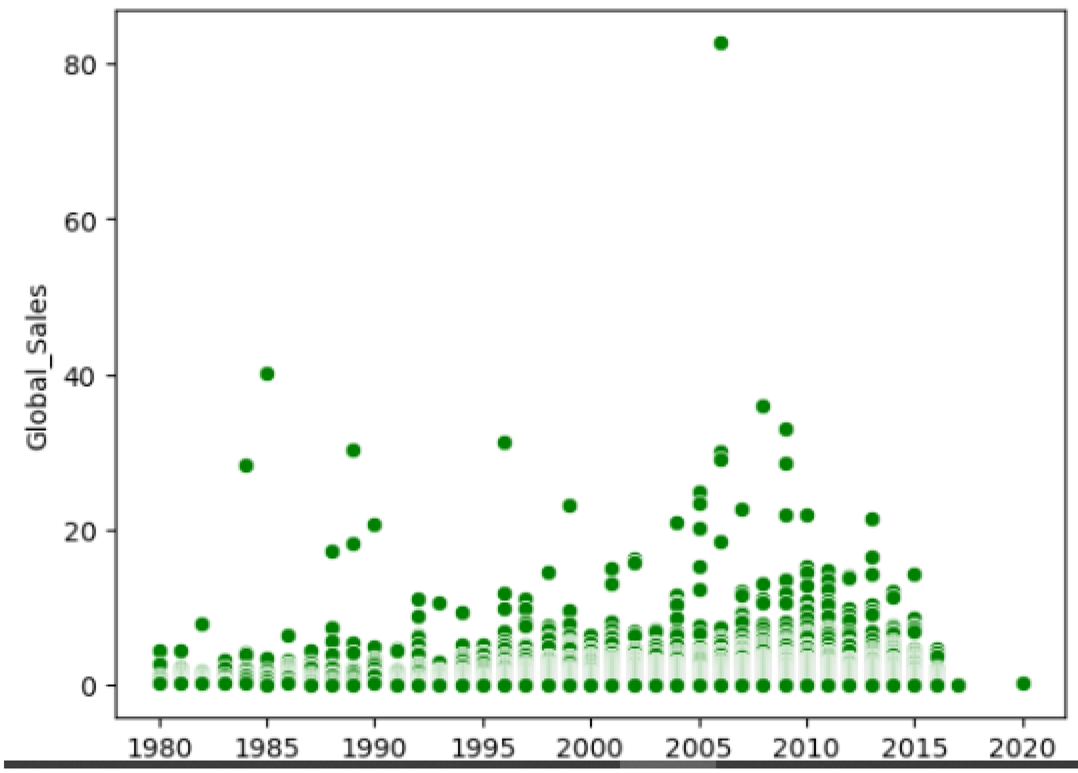
```
#Create Boxplots for our continuous variables
sb.boxplot('Year' , data = df)
mpl.show()

sb.boxplot('NA_Sales' , data = df)
mpl.show()

sb.boxplot('EU_Sales' , data = df)
mpl.show()

sb.boxplot('JP_Sales' , data = df)
mpl.show()

sb.boxplot('Other_Sales' , data = df)
mpl.show()

sb.boxplot('Global_Sales' , data = df)
mpl.show()

sb.boxplot('Platform' , data = df)
mpl.show()

sb.boxplot('Genre' , data = df)
mpl.show()

sb.boxplot('Publisher' , data = df)
mpl.show()
```

```
C:\Users\e202271009\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From versi
on 0.12, the only valid positional argument will be `data`, and passing ot
her arguments without an explicit keyword will result in an error or misin
terpretation.
  warnings.warn(
```





```
#Scatterplots to show direct or inverse relationships between the target &
sb.scatterplot(x=df['Year'], y=df['Global_Sales'],
                color='green')
mpl.show()
sb.scatterplot(x=df['NA_Sales'], y=df['Global_Sales'],
                color='green')
mpl.show()
sb.scatterplot(x=df['JP_Sales'], y=df['Global_Sales'],
                color='green')
mpl.show()
sb.scatterplot(x=df['EU_Sales'], y=df['Global_Sales'],
                color='green')
mpl.show()
sb.scatterplot(x=df['Other_Sales'], y=df['Global_Sales'],
                color='green')
mpl.show()
sb.scatterplot(x=df['Platform'], y=df['Global_Sales'],
                color='green')
mpl.show()
sb.scatterplot(x=df['Genre'], y=df['Global_Sales'],
                color='green')
mpl.show()
sb.scatterplot(x=df['Publisher'], y=df['Global_Sales'],
                color='green')
mpl.show()
```

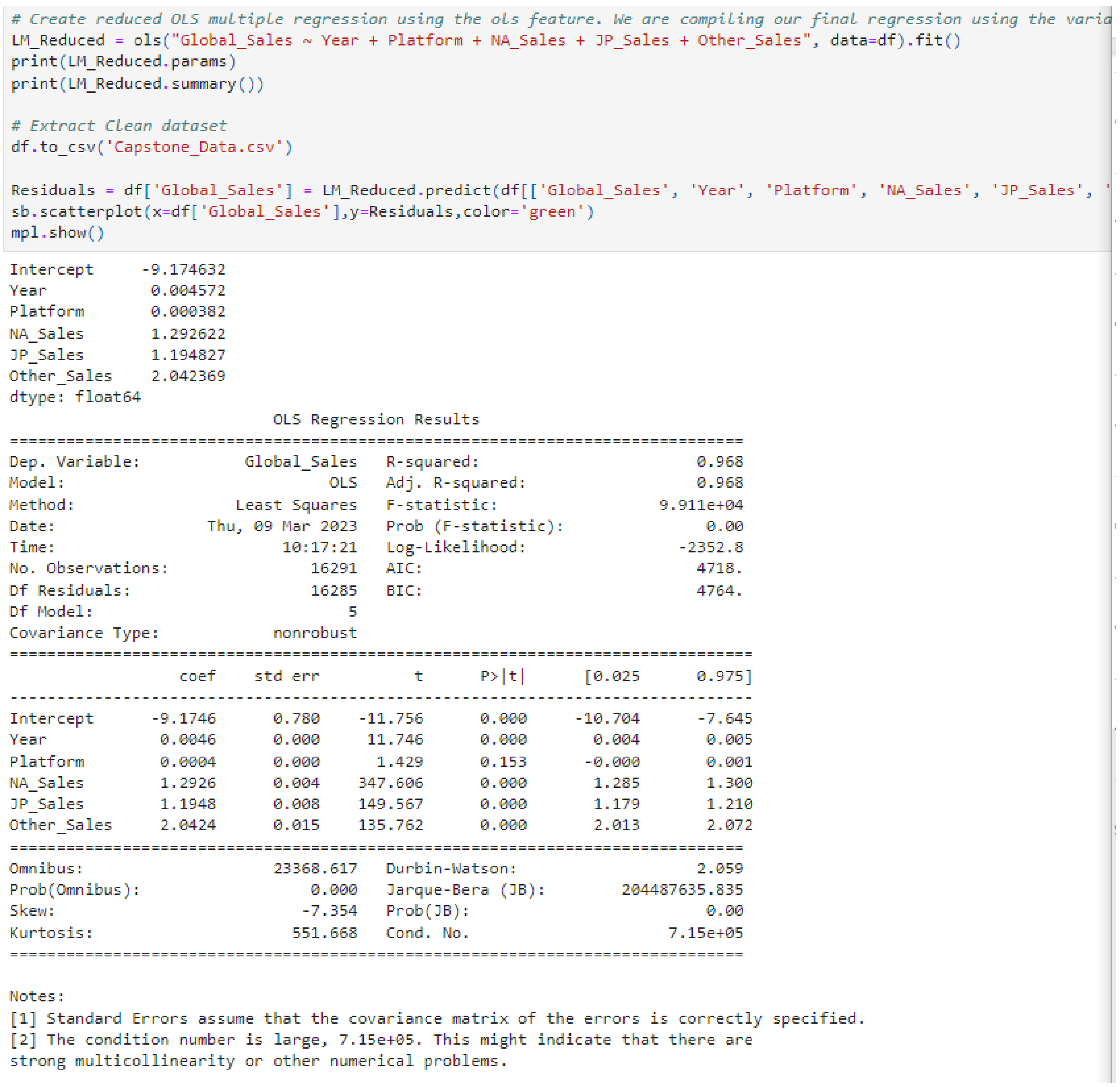


The python file and html have been uploaded to follow along with this technical report. Now that the data is fully prepared, we ran a MLR and assessment using the sklearn package, residual plot, and residual error analysis seen below in the report.

Outline of findings

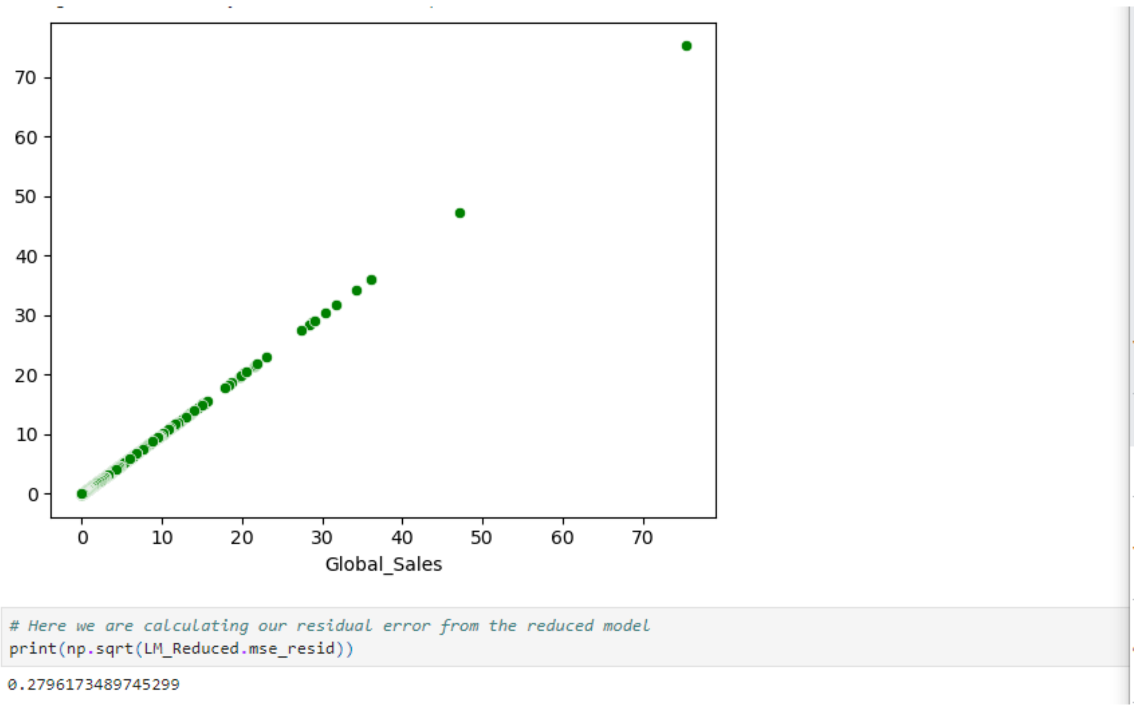
There were two initial models we ran, one with and one without the sales data, and then they were both reduced using the following methods: p value stepwise backwards reduction, Variance Inflation Factor (VIF) reduction. The final reduced models were compiled and then evaluated with their residual error and R squared value as the strength of prediction. The model without the sales value was statistically insignificant with a strength of prediction of 0.7%.

The final model was statistically significant and we can reject the null hypothesis. We can create a MLR model that can predict future global sales. The model output and information are as follows:



```python
# Create reduced OLS multiple regression using the ols feature. We are compiling our final regression using the varia
LM_Reduced = ols("Global_Sales ~ Year + Platform + NA_Sales + JP_Sales + Other_Sales", data=df).fit()
print(LM_Reduced.params)
print(LM_Reduced.summary())

# Extract Clean dataset
df.to_csv('Capstone_Data.csv')

Residuals = df['Global_Sales'] - LM_Reduced.predict(df[['Global_Sales', 'Year', 'Platform', 'NA_Sales', 'JP_Sales', '
sb.scatterplot(x=df['Global_Sales'],y=Residuals,color='green')
mpl.show()
```

```
Intercept     -9.174632
Year           0.004572
Platform       0.000382
NA_Sales       1.292622
JP_Sales       1.194827
Other_Sales    2.042369
dtype: float64
                            OLS Regression Results
==============================================================================
Dep. Variable:           Global_Sales   R-squared:                       0.968
Model:                            OLS   Adj. R-squared:                  0.968
Method:                 Least Squares   F-statistic:                 9.911e+04
Date:                Thu, 09 Mar 2023   Prob (F-statistic):               0.00
Time:                        10:17:21   Log-Likelihood:                -2352.8
No. Observations:               16291   AIC:                             4718.
Df Residuals:                   16285   BIC:                             4764.
Df Model:                           5
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept      -9.1746      0.780    -11.756      0.000     -10.704      -7.645
Year            0.0046      0.000     11.746      0.000       0.004       0.005
Platform        0.0004      0.000      1.429      0.153      -0.000       0.001
NA_Sales        1.2926      0.004    347.606      0.000       1.285       1.300
JP_Sales        1.1948      0.008    149.567      0.000       1.179       1.210
Other_Sales     2.0424      0.015    135.762      0.000       2.013       2.072
==============================================================================
Omnibus:                    23368.617   Durbin-Watson:                   2.059
Prob(Omnibus):                  0.000   Jarque-Bera (JB):       204487635.835
Skew:                          -7.354   Prob(JB):                         0.00
Kurtosis:                     551.668   Cond. No.                     7.15e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.15e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```



```
In [22]: # Here we are calculating our residual error from the reduced model
         print(np.sqrt(LM_Reduced.mse_resid))

0.2796173489745299
```

y = -9.174632 + (Year * 0.004572) + (Platform * 0.000382) + (NA_Sales * 1.292622) + (JP_Sales * 1.194827) + (Other_Sales * 2.042369)

Residual error: .28

R-squared: .968, 96.8% strength of prediction

The coefficients suggest that per each unit of the following, Global Sales will increase or decrease by the following per one unit of the variable:

Year, Global Sales will increase by 0.0046

Platform, Global Sales will increase by .0004

NA_Sales, Global Sales will increase by 1.12926

JP_Sales, Global Sales will increase by 1.1948

Other_Sales, Global Sales will increase by 2.0424

The condition number is large, indicating we have multicollinearity or other numerical problems with this model. However, we can reject the null hypothesis and this is a valid, statistically significant model to predict future global sales.

Explanation of the limitations of the techniques and tools

The major limitations from this analysis are MLR model's do not show causality and only show linear relationships. Both of these can be mitigated with further investigation. The following limitations and techniques are pulled from the full report on the analysis:

Python- v3

Jupyter labs v3.44

I am using python version 3 in Jupyter labs version 3.44. To import the data pandas had to be imported. I chose to use this coding environment and pandas for the same reason- they are widely accepted as an easy and straightforward tool to perform a wide range of data analysis tasks. One disadvantage of python and jupyter labs is that it is considered a complicated language when installing the numerous packages needed, whereas other languages like R have many built in capabilities. Despite this, I am more familiar with and wanted to perform this analysis in python (Prasanna 2021).

To analyze the model, we will look at p values with stepwise backwards reduction, variance inflation factor reduction (VIF), residual error, and r square. Stepwise backwards reduction works by removing all variables with a p value greater than .05, which is an output of our model and a disadvantage of it is that it quickly eliminates variables from one metric and allows for bias to be introduced into the model. VIF measures how much a variable is influenced by another variable i.e. the higher the number the greater the need to eliminate the variable. We used a cutoff of 3 for a strict elimination and therefore shows the disadvantage of possibly allowing bias and definitely allowing for an arbitrary reduction method (Yadav 2021).

Beyond the reduction metrics, I chose to use residual error to measure our model because it is the difference between the actual and predicted values. This will allow us to measure how well our prediction method works and should show small residual error and a tightly plotted line. One disadvantage of this is that any inconsistencies with residuals will show up with an inconsistent model. Finally, the R squared value shows strength of prediction and this can be looked at to show percentage of strength of prediction, to measure the accuracy of the model. One disadvantage of R squared is that it does not decrease as our model changes.

Summary of recommended course of actions

The full report contains the detailed course of action. Here is a summary:

I recommend the hypothetical business to open up the analysis to non-linear relationships and to go back and greatly increase the size of the dataset and the characteristics listed within the dataset. Doing both of these will build on this investigation.

We need to gather a larger dataset with more games and characteristics of games to run another model with the same questions. Further information gathering could be where the game was released, which time of year, if other major titles released during the same time, and the overall inflation/economy at the time since the sales span many years. I believe gathering a much larger dataset and introducing more possible characteristics would present this hypothetical business problem with a clearer picture.

A second direction for future study to build a model to predict future global sales would be to dive into the relationships found here, despite the strong multicollinearity and non-conclusive results. This would mean diving into the relationship between year and sales and see if there's an inflation factor or if there’s a pattern we can investigate for the year and sales information. Seeing the overall performance of platforms would also be beneficial in this path and opening up our analysis to non-linear relationships will provide more actionable insight.

In summary, we need to fine tune our model and increase the accuracy while finding a way to decrease the condition number. This will take several iterations and a larger dataset.

Expected benefits of the study (quantitative).

We can use the regression equation to predict future sales:

y = -9.174632 + (Year * 0.004572) + (Platform * 0.000382) + (NA_Sales * 1.292622) +

(JP_Sales * 1.194827) + (Other_Sales * 2.042369)

We can manipulate our coefficients to change our global sales

Per one unit…

Year, Global Sales will increase by 0.0046

Platform, Global Sales will increase by .0004

NA_Sales, Global Sales will increase by 1.12926

JP_Sales, Global Sales will increase by 1.1948

Other_Sales, Global Sales will increase by 2.0424

We have a low residual error at .28 and a high strength of prediction at 96.8%, indicating our reduction factors are adequate and the model needs more fine tuning. We have a working model that needs fine tuning. In summary, we now have a multiple linear regression model that can be used to predict future global sales. We can also identify characteristics of games with high global sales if needed and have a launching point for further investigations.

Presentation of Findings

B. Present your Capstone project to a *non-technical audience*. Use Panopto to record yourself delivering a presentation that includes slides displayed by a multimedia tool (e.g., PowerPoint, Keynote, Prezi). Your recording should capture your multimedia presentation. Your presentation should also demonstrate appropriate communication skills for the audience.

Note: appearing on camera is optional.

1. **Cover in your presentation *each* of the following points:**
   - **a brief introduction of yourself and your background**
   - **a statement of the problem and the hypothesis**
   - **a summary of the data-analysis process**
   - **an outline of the findings**
   - **an explanation of the limitations of the techniques and tools used**
   - **a summary of proposed actions**
   - **expected benefits of the study (be as specific and quantitative as possible)**

Panopto Video Link:

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=a9220d12-0ef5-4c36-928c-afc00177aa3d

The presentation has been included as "Capstone Presentation"

All of the required points are included and named in the titles of the slides

C. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

Sources

Prasanna, M. (2021, October 21). *How to Efficiently Handle Large Datasets for Machine Learning and Data Analysis Using Python*. Medium. https://python.plainenglish.io/working-with-large-datasets-for-machine-learning-d8da0dd802fb

Video Game Sales. (n.d.). Www.kaggle.com. https://www.kaggle.com/datasets/gregorut/videogamesales

Yadav, H. (2021, May 8). *Multiple Linear Regression Implementation in Python*. Machine Learning with Python. https://medium.com/machine-learning-with-python/multiple-linear-regression-implementation-in-python-2de9b303fc0c

D. Demonstrate professional communication in the content and presentation of your submission.

This aspect cannot be summarized, however, I hope it has shown through in all aspects of this report. Thank you for reading this.