PES
UNIVERSITY

Comprehensive Report
on

“Data Analytics and Data Visualization with Tableau”
(Summer Course)

Submitted by:

Name:	**Ravi Aishwarya**
SRN:	**PES2UG19CS322**
SEM:	**7**

Faculty Handling the Course:

Dr.Sudeepa Roy Dey
Dr.Prajwala TR
Prof.Ruby Dinakar

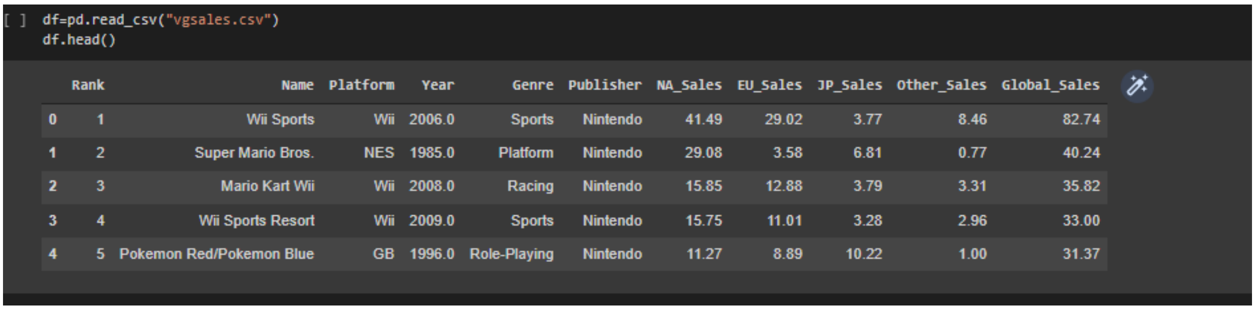
4th -9th, July 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
PES UNIVERSITY, EC campus
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, Ind

Table of Contents:

SOME OF THE SNIPPETS OF CODE ON EXPLORATION OF DATA:

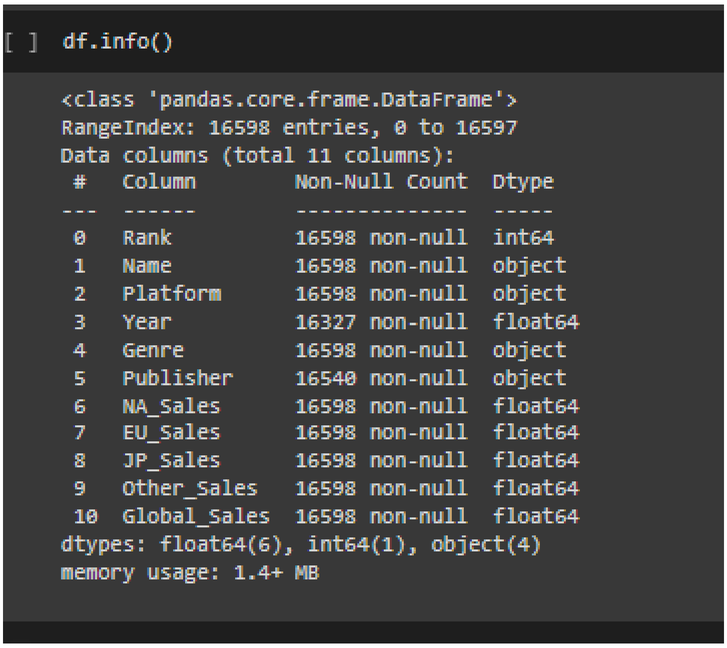
DATASET:



```
[ ]  df=pd.read_csv("vgsales.csv")
     df.head()
```

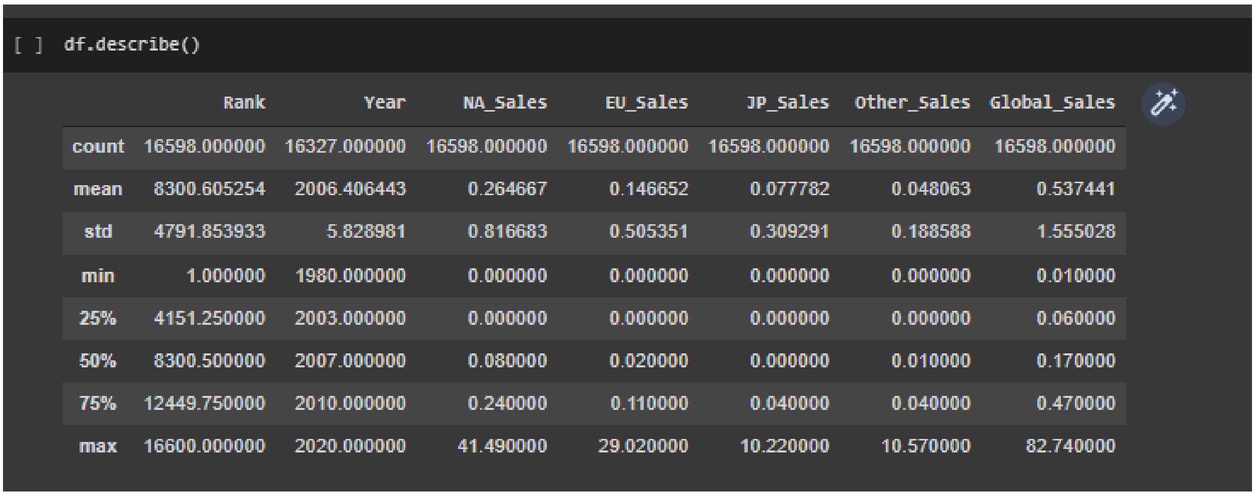
	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	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

Information of the data:



```
[ ]  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   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
```

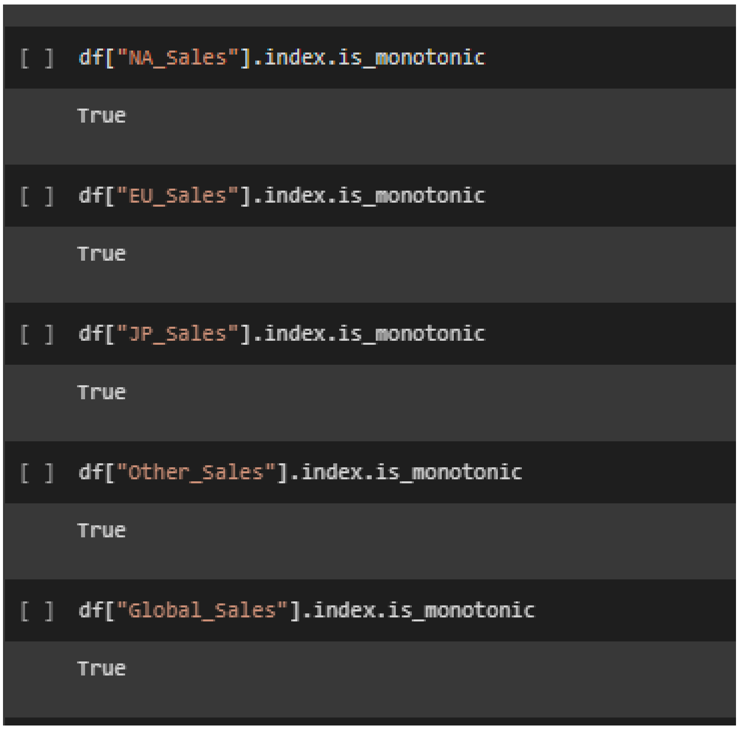
Description of the data:



```
[ ] df.describe()
```

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
count	16598.000000	16327.000000	16598.000000	16598.000000	16598.000000	16598.000000	16598.000000
mean	8300.605254	2006.406443	0.264667	0.146652	0.077782	0.048063	0.537441
std	4791.853933	5.828981	0.816683	0.505351	0.309291	0.188588	1.555028
min	1.000000	1980.000000	0.000000	0.000000	0.000000	0.000000	0.010000
25%	4151.250000	2003.000000	0.000000	0.000000	0.000000	0.000000	0.060000
50%	8300.500000	2007.000000	0.080000	0.020000	0.000000	0.010000	0.170000
75%	12449.750000	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

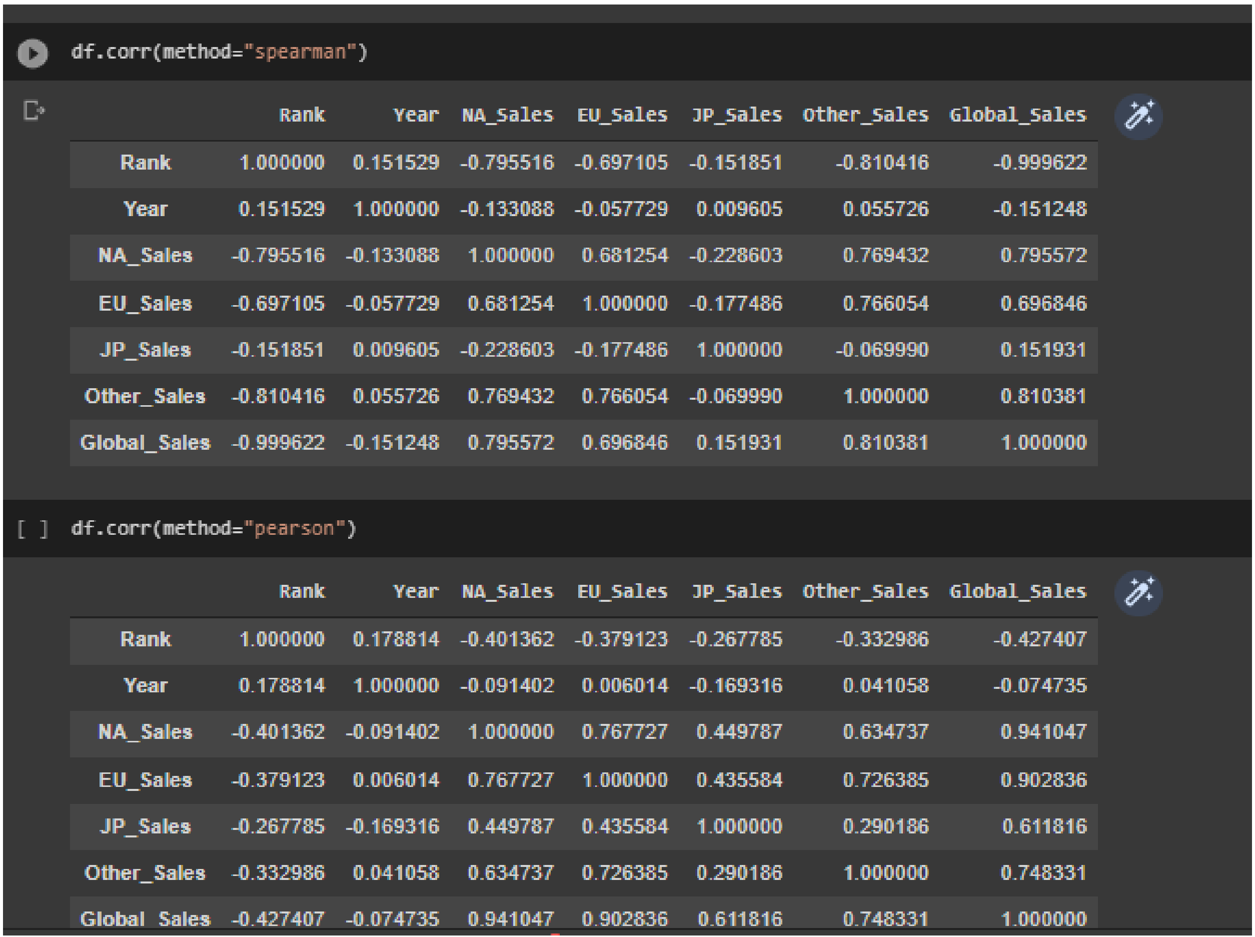
Nature of data:



```
[ ] df["NA_Sales"].index.is_monotonic

    True

[ ] df["EU_Sales"].index.is_monotonic

    True

[ ] df["JP_Sales"].index.is_monotonic

    True

[ ] df["Other_Sales"].index.is_monotonic

    True

[ ] df["Global_Sales"].index.is_monotonic

    True
```

We observe that data is monotonic

Correlation between features:

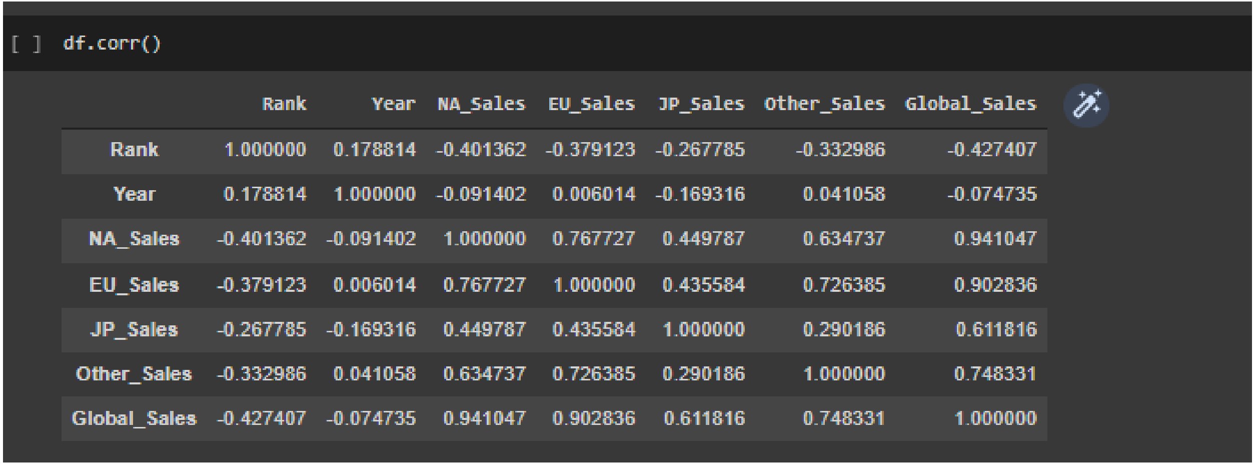


```
df.corr(method="spearman")
```

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
Rank	1.000000	0.151529	-0.795516	-0.697105	-0.151851	-0.810416	-0.999622
Year	0.151529	1.000000	-0.133088	-0.057729	0.009605	0.055726	-0.151248
NA_Sales	-0.795516	-0.133088	1.000000	0.681254	-0.228603	0.769432	0.795572
EU_Sales	-0.697105	-0.057729	0.681254	1.000000	-0.177486	0.766054	0.696846
JP_Sales	-0.151851	0.009605	-0.228603	-0.177486	1.000000	-0.069990	0.151931
Other_Sales	-0.810416	0.055726	0.769432	0.766054	-0.069990	1.000000	0.810381
Global_Sales	-0.999622	-0.151248	0.795572	0.696846	0.151931	0.810381	1.000000

```
[ ] df.corr(method="pearson")
```

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
Rank	1.000000	0.178814	-0.401362	-0.379123	-0.267785	-0.332986	-0.427407
Year	0.178814	1.000000	-0.091402	0.006014	-0.169316	0.041058	-0.074735
NA_Sales	-0.401362	-0.091402	1.000000	0.767727	0.449787	0.634737	0.941047
EU_Sales	-0.379123	0.006014	0.767727	1.000000	0.435584	0.726385	0.902836
JP_Sales	-0.267785	-0.169316	0.449787	0.435584	1.000000	0.290186	0.611816
Other_Sales	-0.332986	0.041058	0.634737	0.726385	0.290186	1.000000	0.748331
Global_Sales	-0.427407	-0.074735	0.941047	0.902836	0.611816	0.748331	1.000000

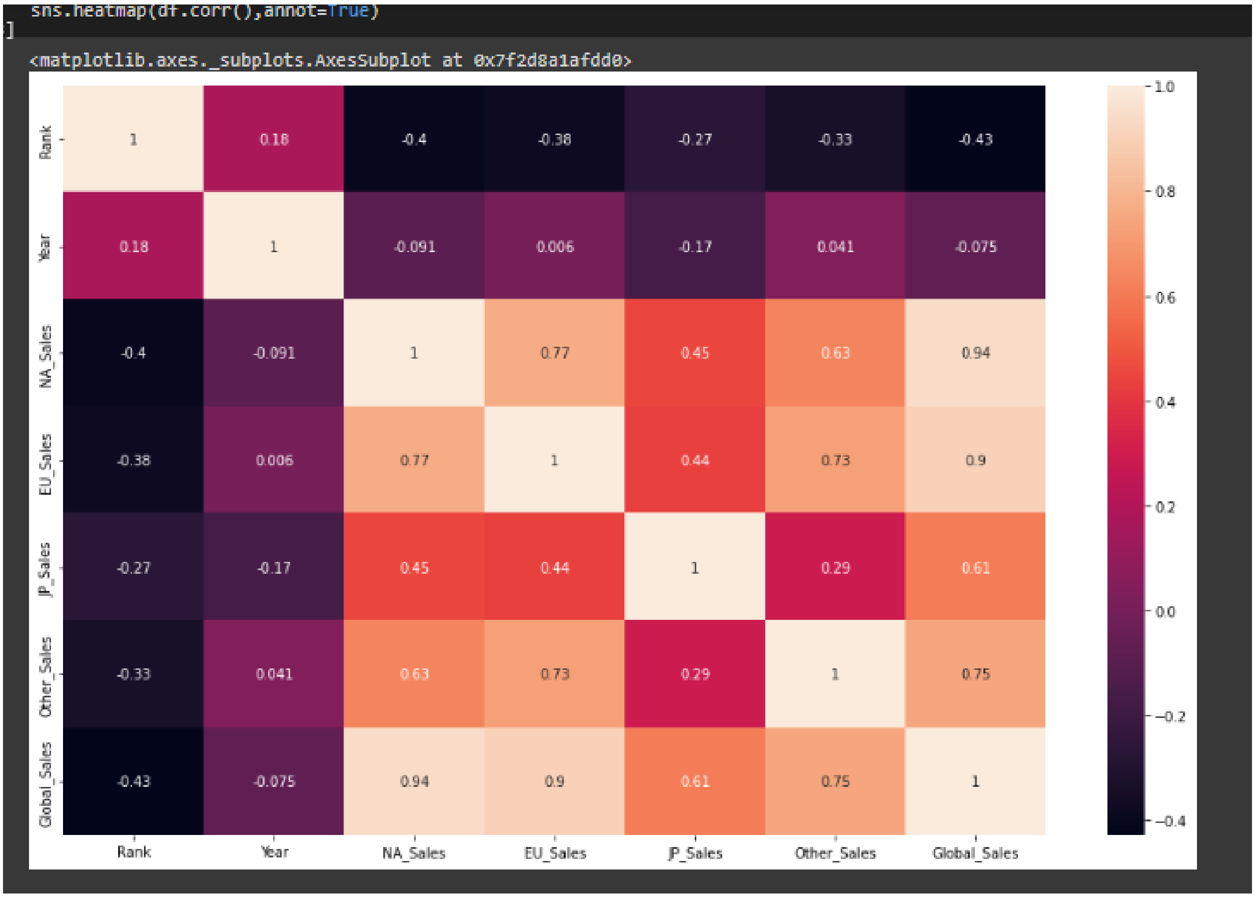


```
[ ] df.corr()
```

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
Rank	1.000000	0.178814	-0.401362	-0.379123	-0.267785	-0.332986	-0.427407
Year	0.178814	1.000000	-0.091402	0.006014	-0.169316	0.041058	-0.074735
NA_Sales	-0.401362	-0.091402	1.000000	0.767727	0.449787	0.634737	0.941047
EU_Sales	-0.379123	0.006014	0.767727	1.000000	0.435584	0.726385	0.902836
JP_Sales	-0.267785	-0.169316	0.449787	0.435584	1.000000	0.290186	0.611816
Other_Sales	-0.332986	0.041058	0.634737	0.726385	0.290186	1.000000	0.748331
Global_Sales	-0.427407	-0.074735	0.941047	0.902836	0.611816	0.748331	1.000000

We see that, by default it is considering pearson.
As it it monotonic we can consider both spearman and pearson.

Heatmap of correlation:



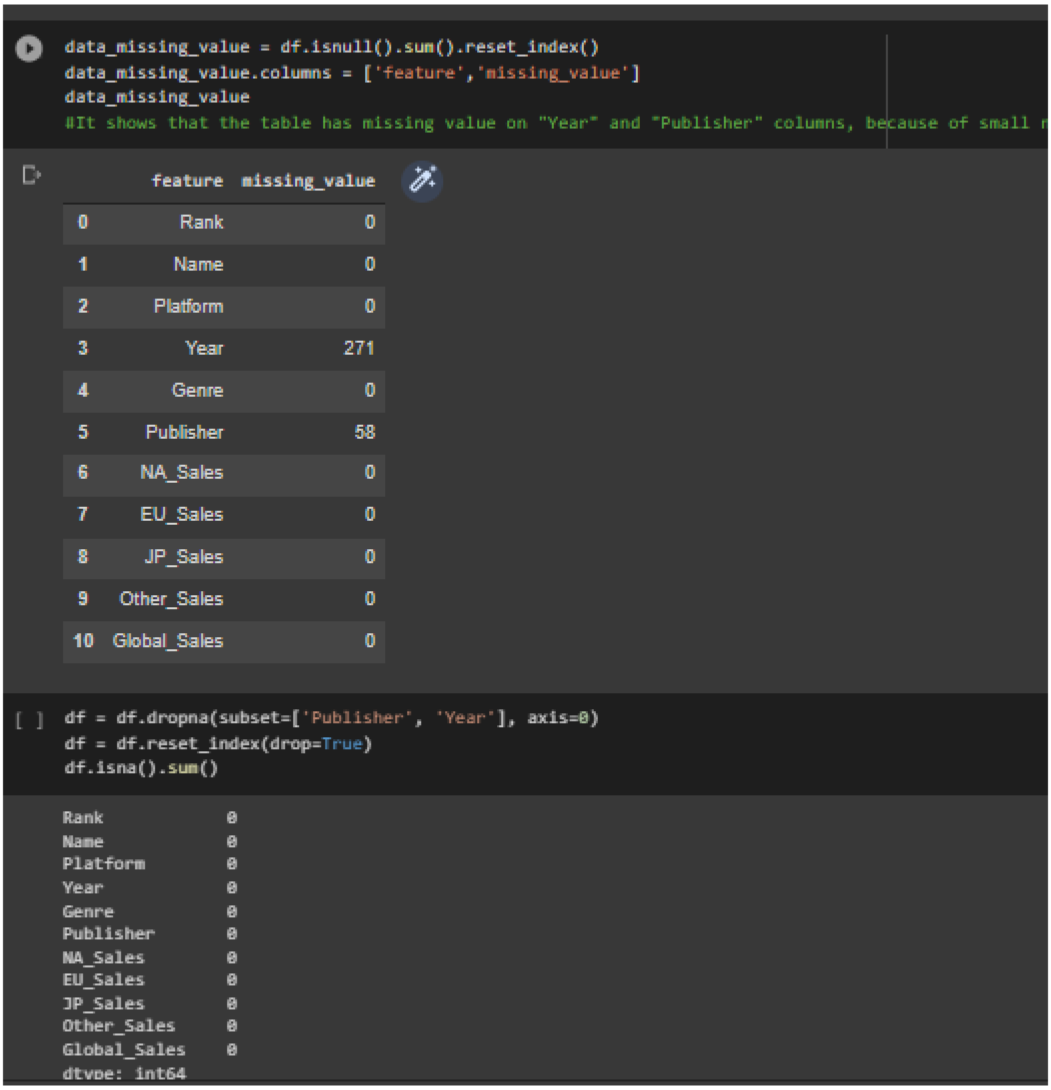
```
 sns.heatmap(df.corr(),annot=True)
]
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2d8a1afdd0>
```

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
Rank	1	0.18	-0.4	-0.38	-0.27	-0.33	-0.43
Year	0.18	1	-0.091	0.006	-0.17	0.041	-0.075
NA_Sales	-0.4	-0.091	1	0.77	0.45	0.63	0.94
EU_Sales	-0.38	0.006	0.77	1	0.44	0.73	0.9
JP_Sales	-0.27	-0.17	0.45	0.44	1	0.29	0.61
Other_Sales	-0.33	0.041	0.63	0.73	0.29	1	0.75
Global_Sales	-0.43	-0.075	0.94	0.9	0.61	0.75	1

We observe that:
‘Global sales’ and ‘NA sales’ are having highest correlation .
‘Global sales’ and ‘Rank’ are having lowest correlation.

Data Cleaning:



```python
data_missing_value = df.isnull().sum().reset_index()
data_missing_value.columns = ['feature','missing_value']
data_missing_value
#It shows that the table has missing value on "Year" and "Publisher" columns, because of small r
```

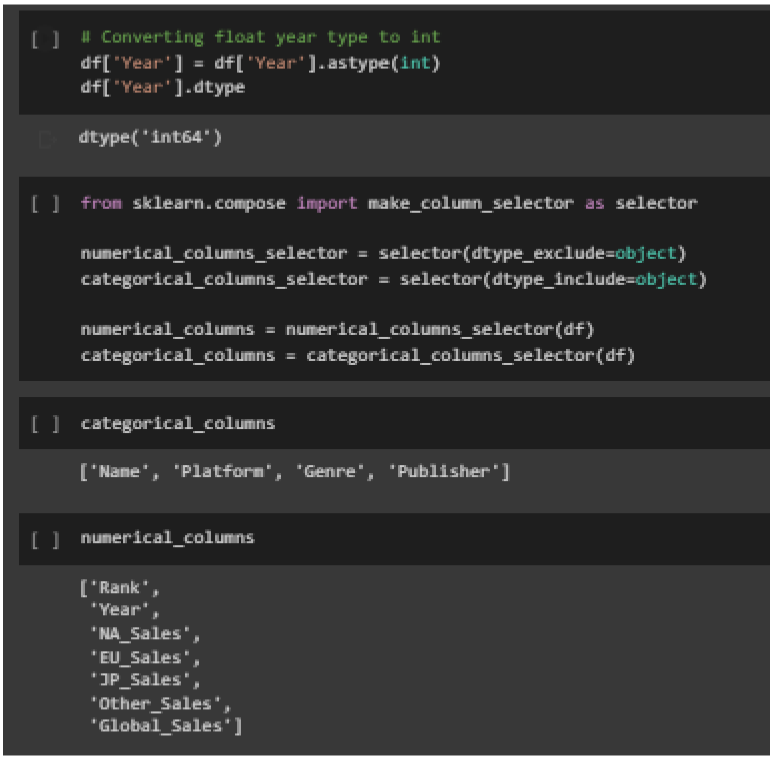
	feature	missing_value
0	Rank	0
1	Name	0
2	Platform	0
3	Year	271
4	Genre	0
5	Publisher	58
6	NA_Sales	0
7	EU_Sales	0
8	JP_Sales	0
9	Other_Sales	0
10	Global_Sales	0

```python
df = df.dropna(subset=['Publisher', 'Year'], axis=0)
df = df.reset_index(drop=True)
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
```

We observe there are missing values in ‘year’ and ‘publisher’ which we will remove it.

Data Pre-processing:



```
# Converting float year type to int
df['Year'] = df['Year'].astype(int)
df['Year'].dtype
```

```
dtype('int64')
```

```
from sklearn.compose import make_column_selector as selector

numerical_columns_selector = selector(dtype_exclude=object)
categorical_columns_selector = selector(dtype_include=object)

numerical_columns = numerical_columns_selector(df)
categorical_columns = categorical_columns_selector(df)
```

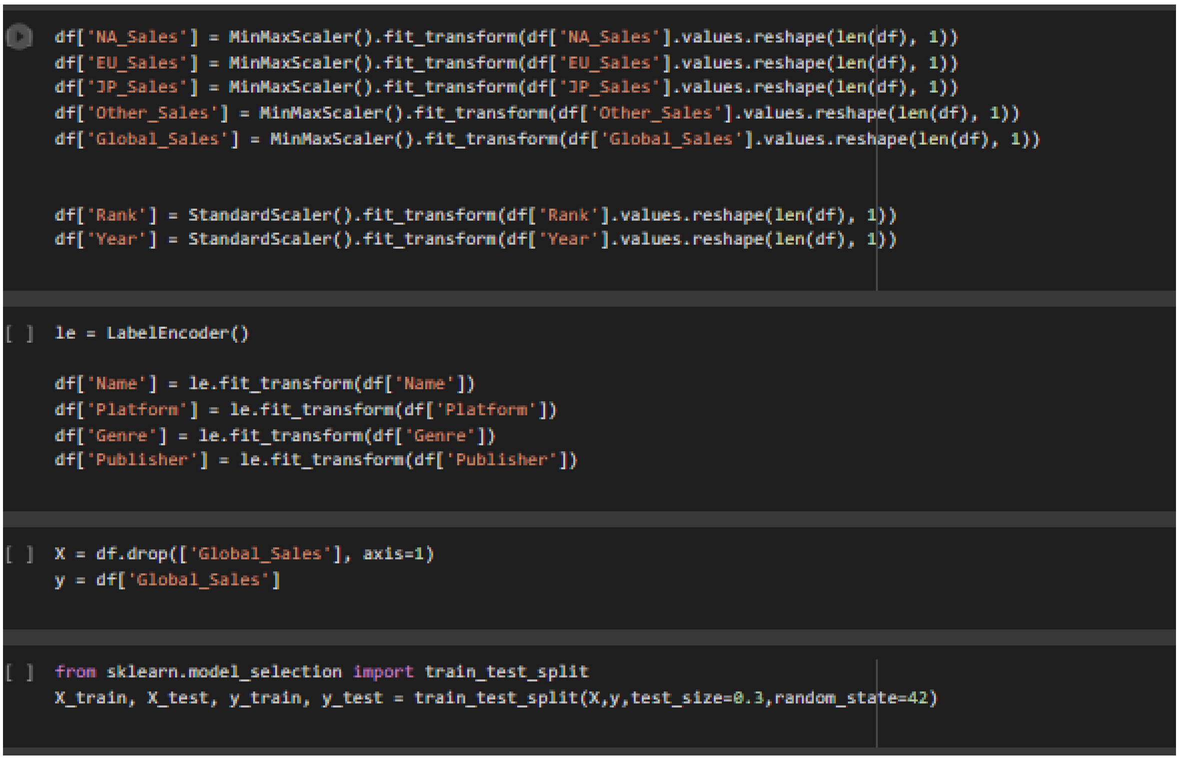
```
categorical_columns
```

```
['Name', 'Platform', 'Genre', 'Publisher']
```

```
numerical_columns
```

```
['Rank',
 'Year',
 'NA_Sales',
 'EU_Sales',
 'JP_Sales',
 'Other_Sales',
 'Global_Sales']
```

We see there are few categorical and numerical columns



```
df['NA_Sales'] = MinMaxScaler().fit_transform(df['NA_Sales'].values.reshape(len(df), 1))
df['EU_Sales'] = MinMaxScaler().fit_transform(df['EU_Sales'].values.reshape(len(df), 1))
df['JP_Sales'] = MinMaxScaler().fit_transform(df['JP_Sales'].values.reshape(len(df), 1))
df['Other_Sales'] = MinMaxScaler().fit_transform(df['Other_Sales'].values.reshape(len(df), 1))
df['Global_Sales'] = MinMaxScaler().fit_transform(df['Global_Sales'].values.reshape(len(df), 1))


df['Rank'] = StandardScaler().fit_transform(df['Rank'].values.reshape(len(df), 1))
df['Year'] = StandardScaler().fit_transform(df['Year'].values.reshape(len(df), 1))
```

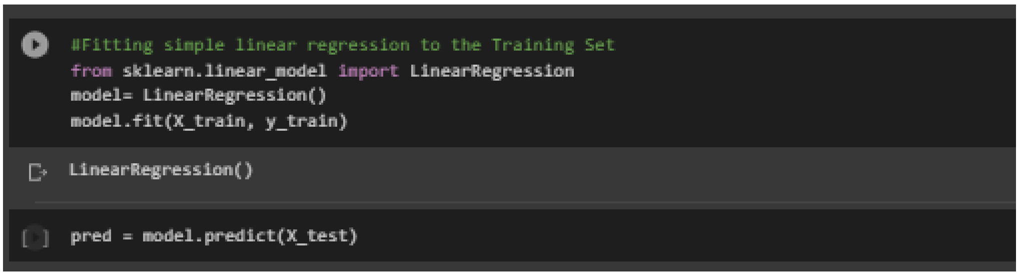
```
le = LabelEncoder()

df['Name'] = le.fit_transform(df['Name'])
df['Platform'] = le.fit_transform(df['Platform'])
df['Genre'] = le.fit_transform(df['Genre'])
df['Publisher'] = le.fit_transform(df['Publisher'])
```

```
X = df.drop(['Global_Sales'], axis=1)
y = df['Global_Sales']
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_state=42)
```

Using normalization we are going to rescale the numerical data
Using Label Encoder we are going to convert the labels into a numeric form so as to convert them into the machine-readable form.

Model Selection and prediction:

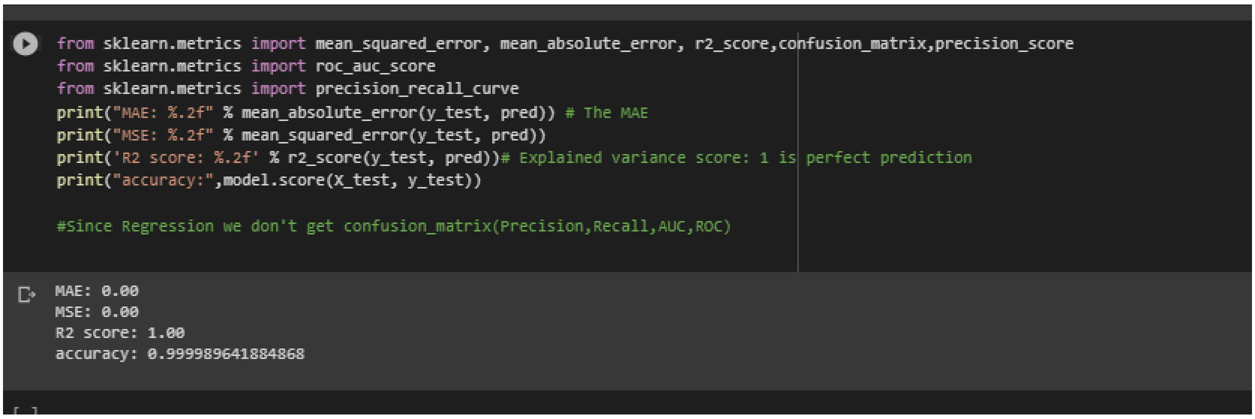


```
#Fitting simple linear regression to the Training Set
from sklearn.linear_model import LinearRegression
model= LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression()
```

```
pred = model.predict(X_test)
```

‘Linear Regression’ is used as our target variable is ‘Global sales’ which is a numeric data



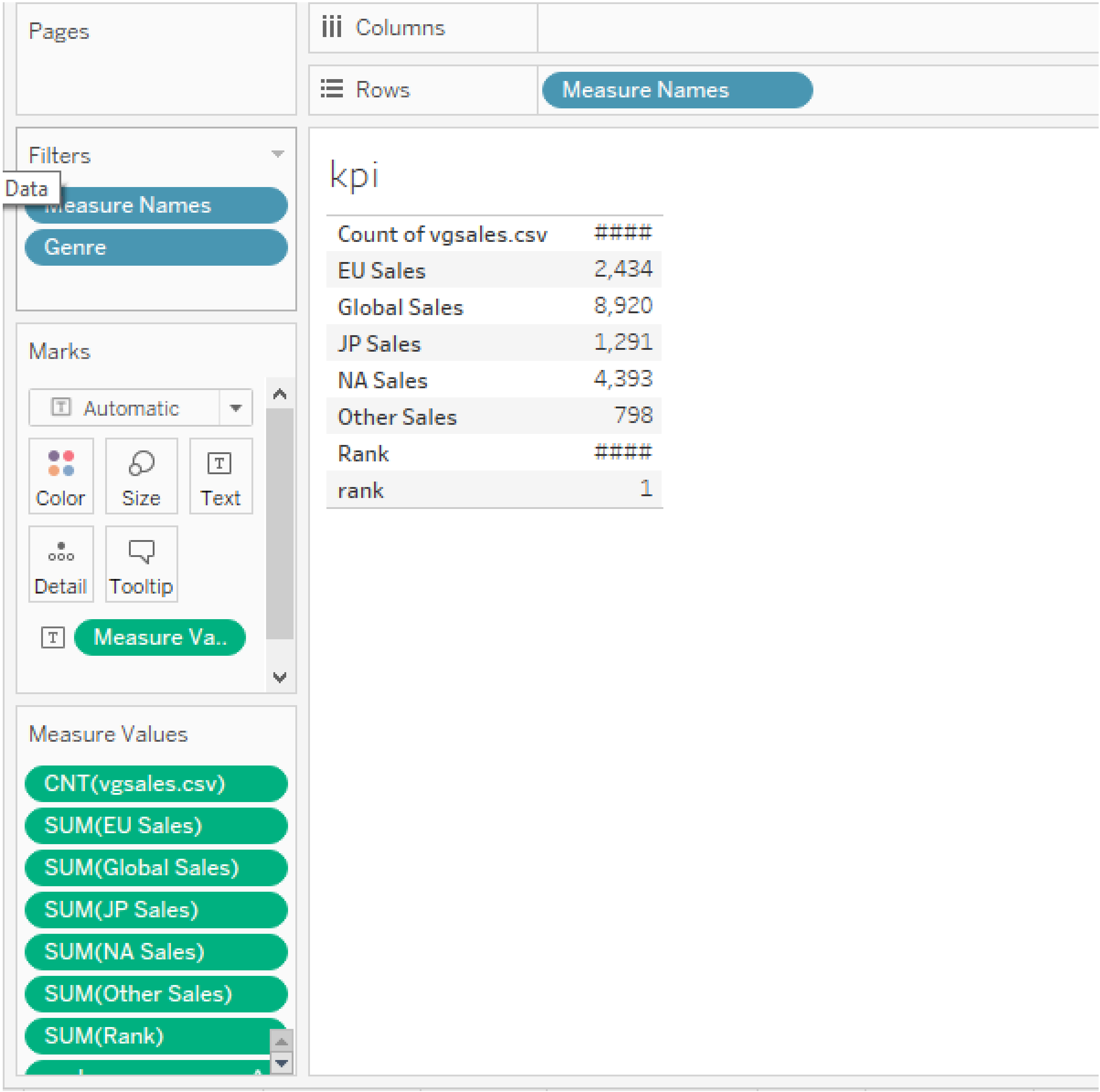
```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score,confusion_matrix,precision_score
from sklearn.metrics import roc_auc_score
from sklearn.metrics import precision_recall_curve
print("MAE: %.2f" % mean_absolute_error(y_test, pred)) # The MAE
print("MSE: %.2f" % mean_squared_error(y_test, pred))
print('R2 score: %.2f' % r2_score(y_test, pred))# Explained variance score: 1 is perfect prediction
print("accuracy:",model.score(X_test, y_test))

#Since Regression we don't get confusion_matrix(Precision,Recall,AUC,ROC)
```

```
MAE: 0.00
MSE: 0.00
R2 score: 1.00
accuracy: 0.999989641884868
```

Hence we got the accuracy of 99.99% which is a perfect prediction.

TABLEAU VISUALIZATION OF VIDEO GAMES:

- **kpi:**



Rows: Measure Names

Filters: Measure Names, Genre

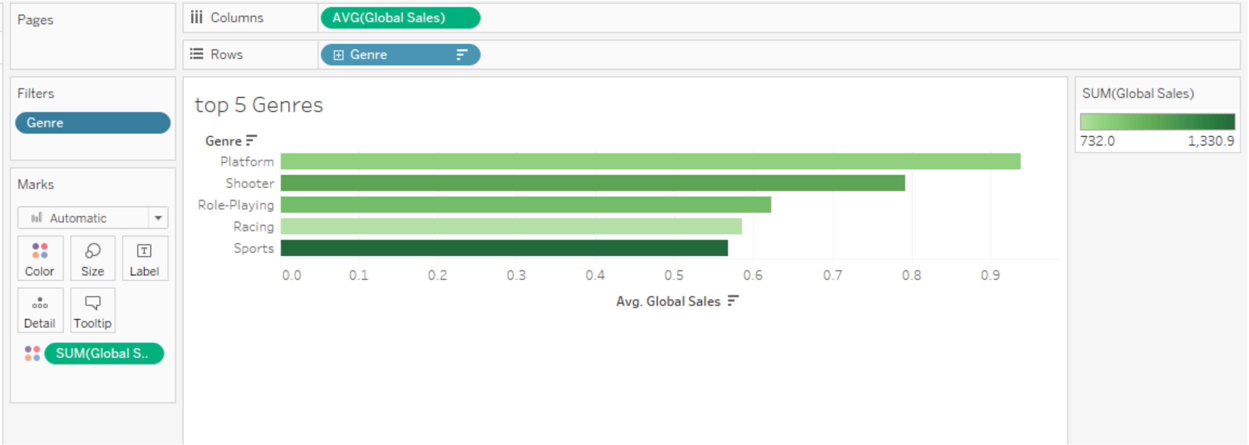
Marks: Measure Va..

Measure Values: CNT(vgsales.csv), SUM(EU Sales), SUM(Global Sales), SUM(JP Sales), SUM(NA Sales), SUM(Other Sales), SUM(Rank)

kpi

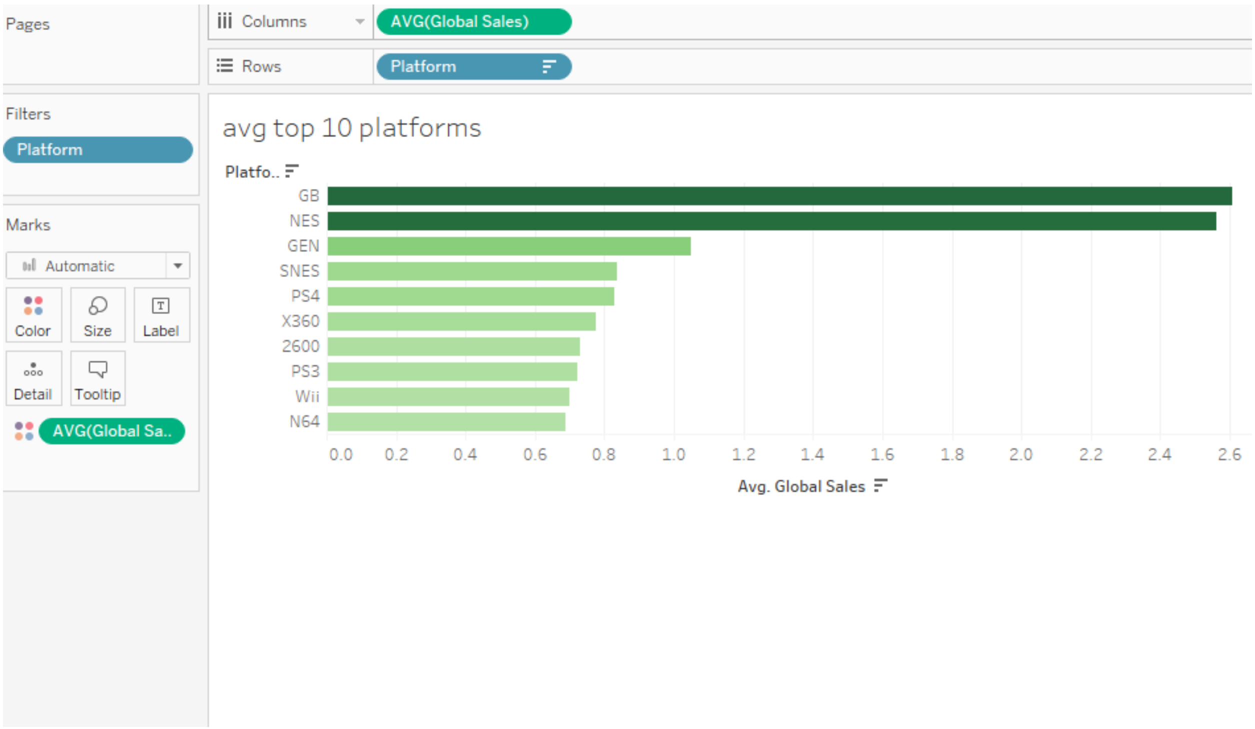
Count of vgsales.csv	####
EU Sales	2,434
Global Sales	8,920
JP Sales	1,291
NA Sales	4,393
Other Sales	798
Rank	####
rank	1

- **Top 5 Genres for average highest global sales:**



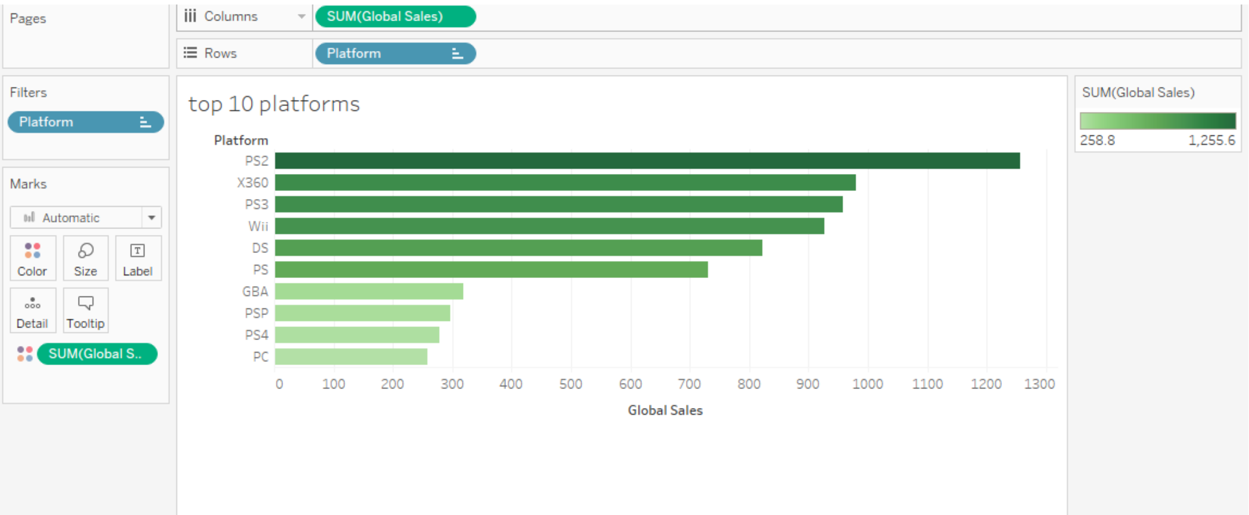
‘Platform’ is having the average highest Global sales in top 5 Genres.

- **Top 10 platforms for average highest global sales:**



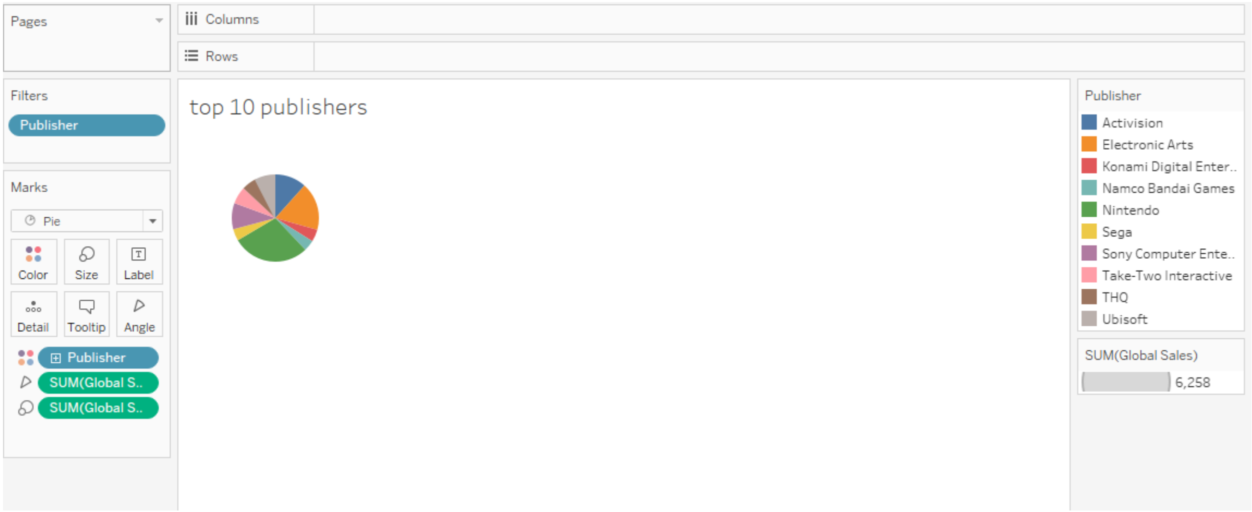
‘GB’ is having the average highest Global sales and lowest is ‘N64’ in top 10 platforms.

- **Top 10 platforms for highest global sales: (SUM)**



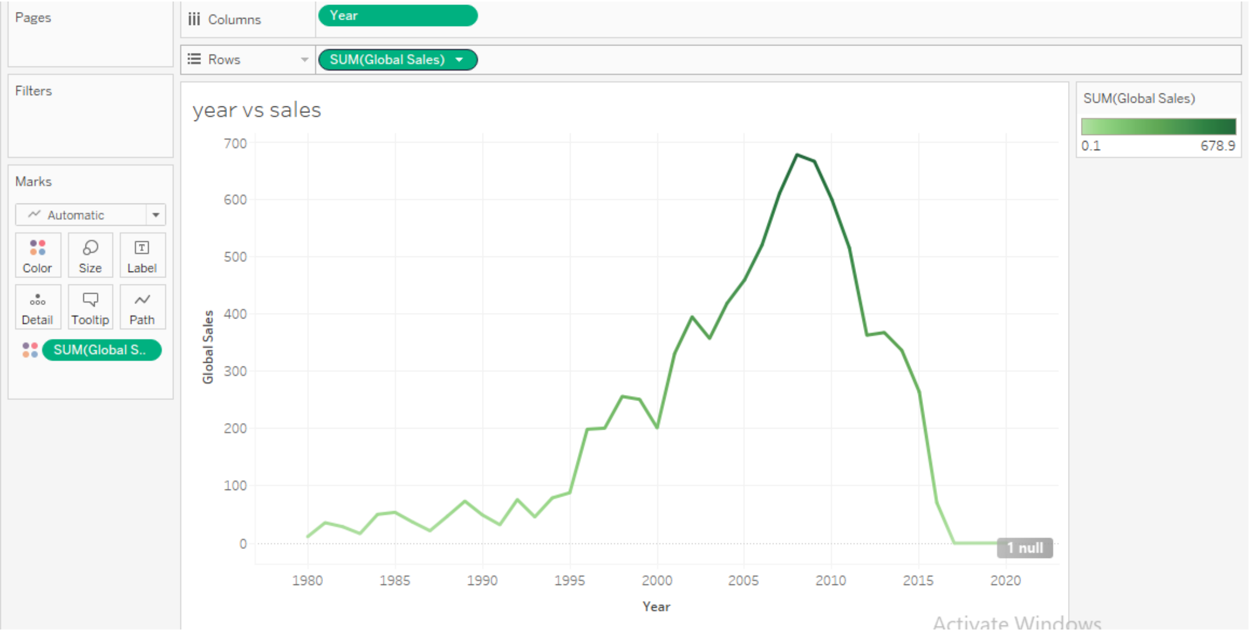
'PS2' is having the highest Global sales and 'PC' the lowest in top 10 platforms.

Top 10 publishers :



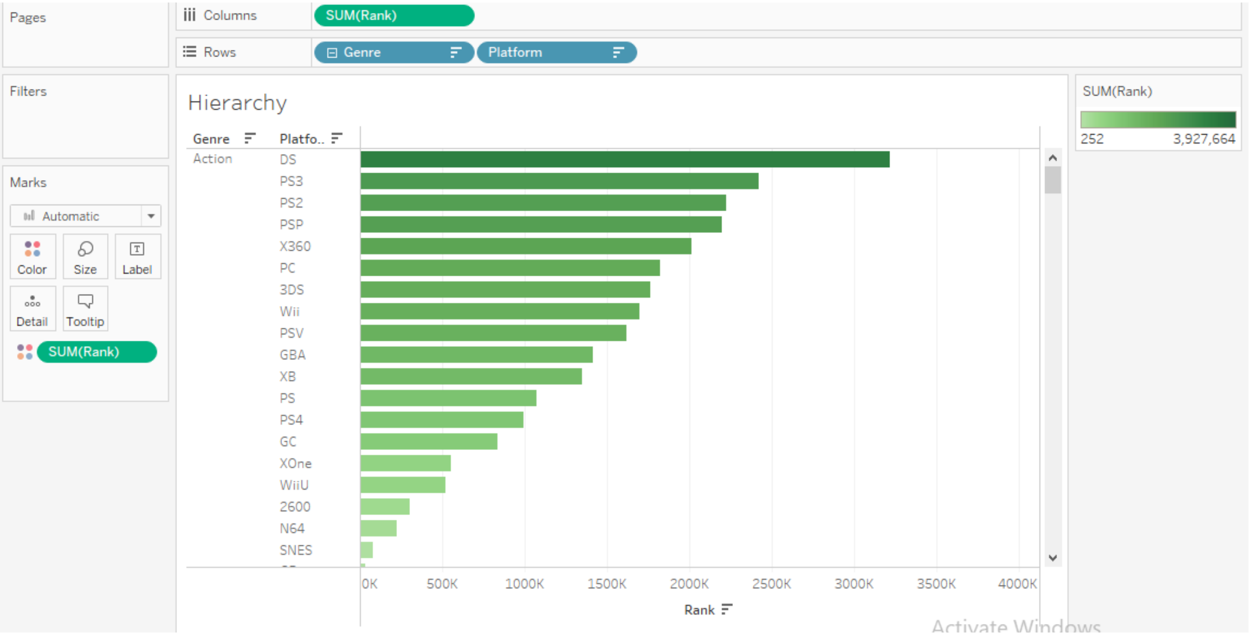
'Nintendo' is the highest top 10 publishers having Global Sales and 'Namco Bandai Games' is having the lowest.

- **Year vs Sales:**



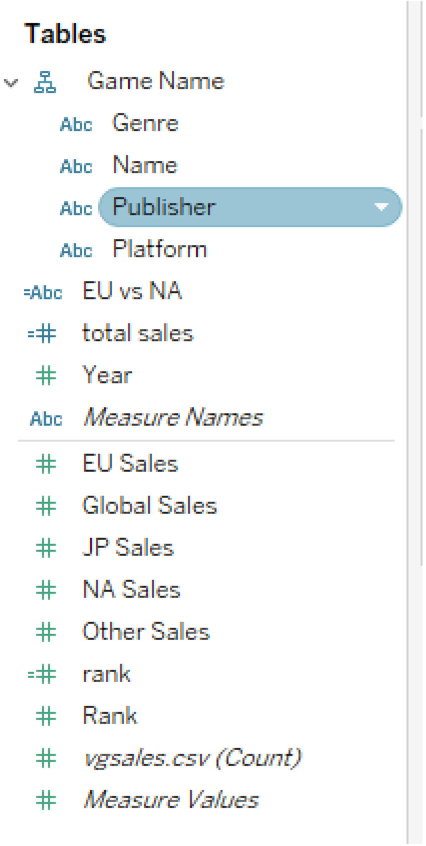
2008 is having the highest sales of 678.9 and lowest at 2020 of 0.1.

- **Hierarchy:**



We have considered ‘Game_Name’ as a hierarchy.

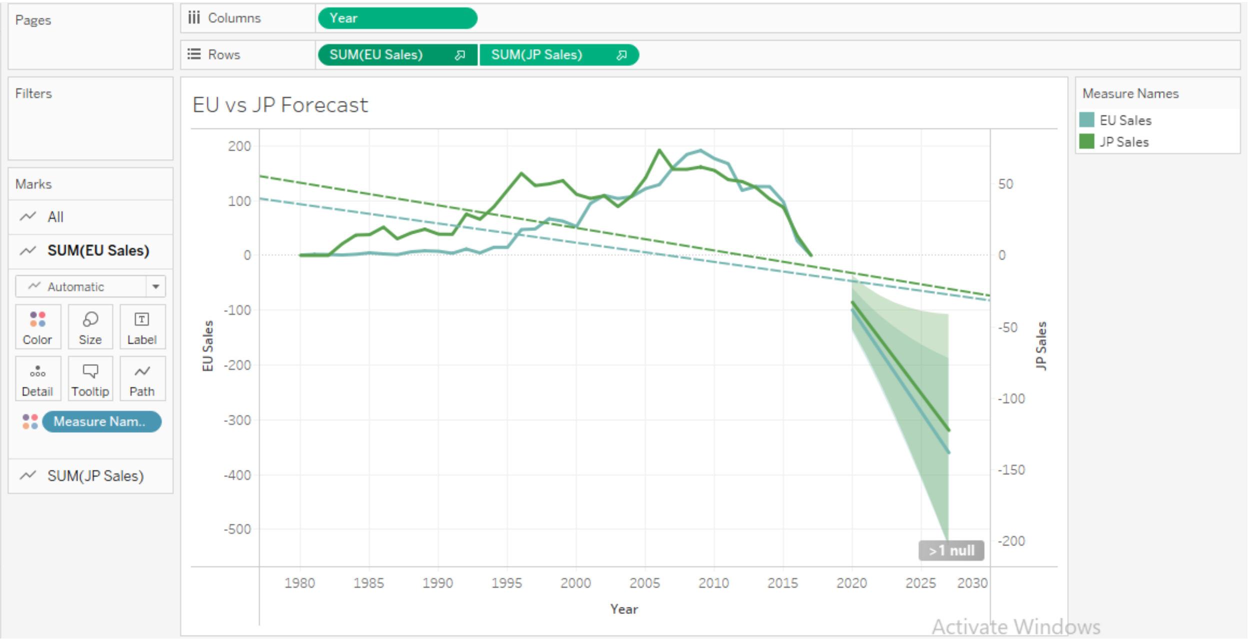
Under that Name, Genre, Publisher, Platform shown as below:



Tables

- Game Name
  - Abc Genre
  - Abc Name
  - Abc Publisher
  - Abc Platform
- =Abc EU vs NA
- =# total sales
- # Year
- Abc *Measure Names*
- # EU Sales
- # Global Sales
- # JP Sales
- # NA Sales
- # Other Sales
- =# rank
- # Rank
- # *vgsales.csv (Count)*
- # *Measure Values*

- **EU vs JP Forecast:**



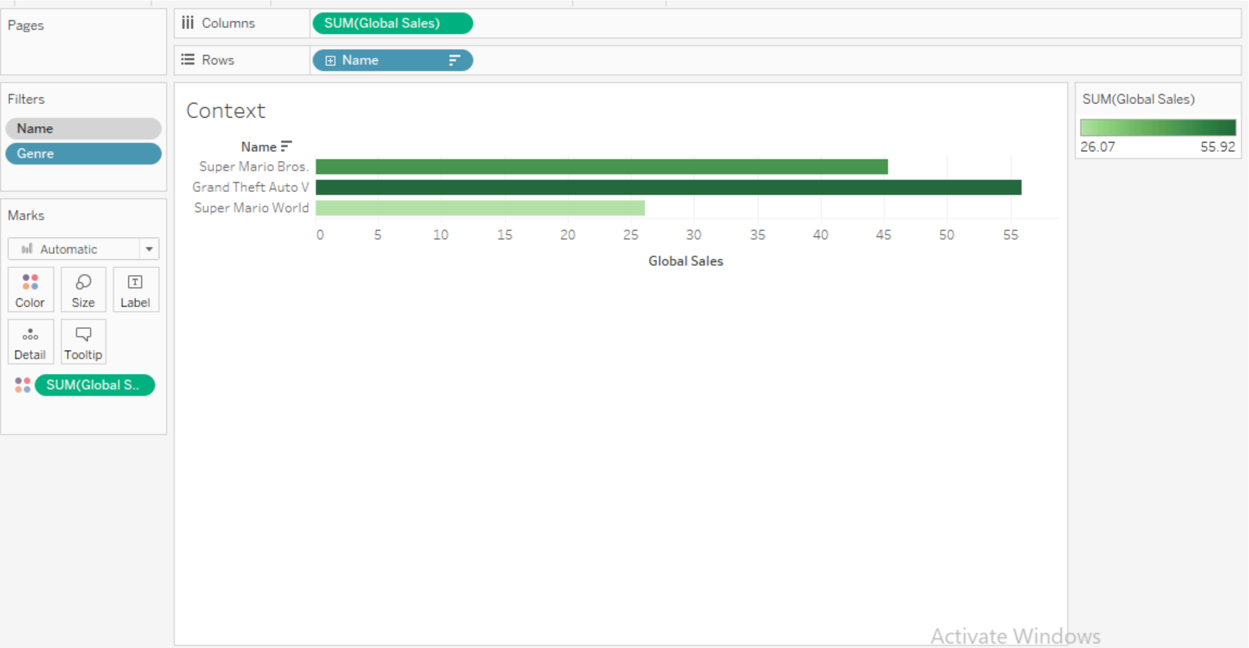
Columns: Year

Rows: SUM(EU Sales) SUM(JP Sales)

Marks: All, SUM(EU Sales), Automatic, Measure Names, SUM(JP Sales)

We observe that there is a decline over the upcoming years.
So the forecast is showing declining trend which might be because of the genre,platform and other factors .

- **<u>Context filter:</u>**



Columns: SUM(Global Sales)

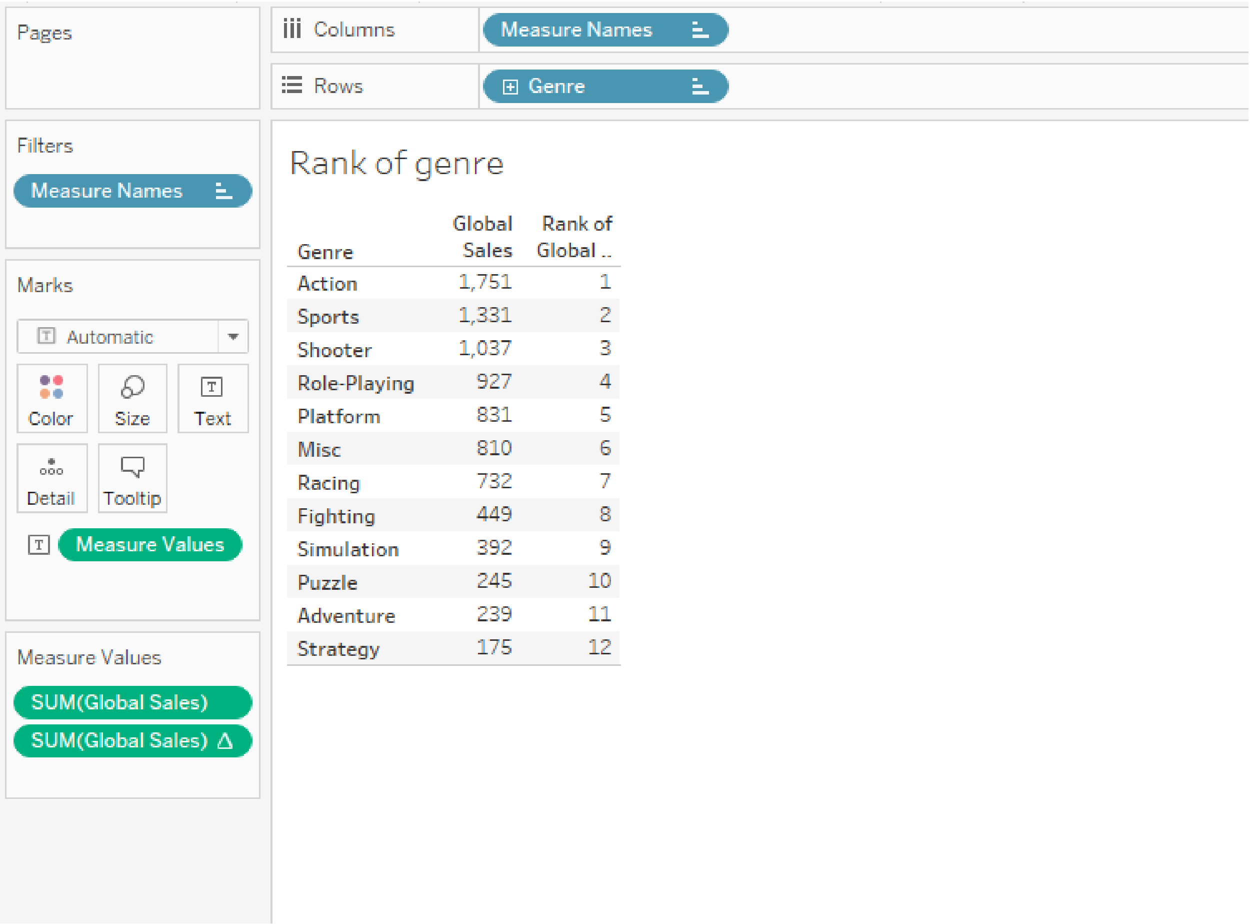
Rows: Name

Filters: Name, Genre

Marks: Automatic

Considering both normal filter and context filter, it considers the context filter first which is ‘Name’ of the top 10 Genre and the Genre filter (Normal filter) is considering only ‘Action’ and ‘Platform’ as it is having the highest number of sales.

- **Rank by Genre:**

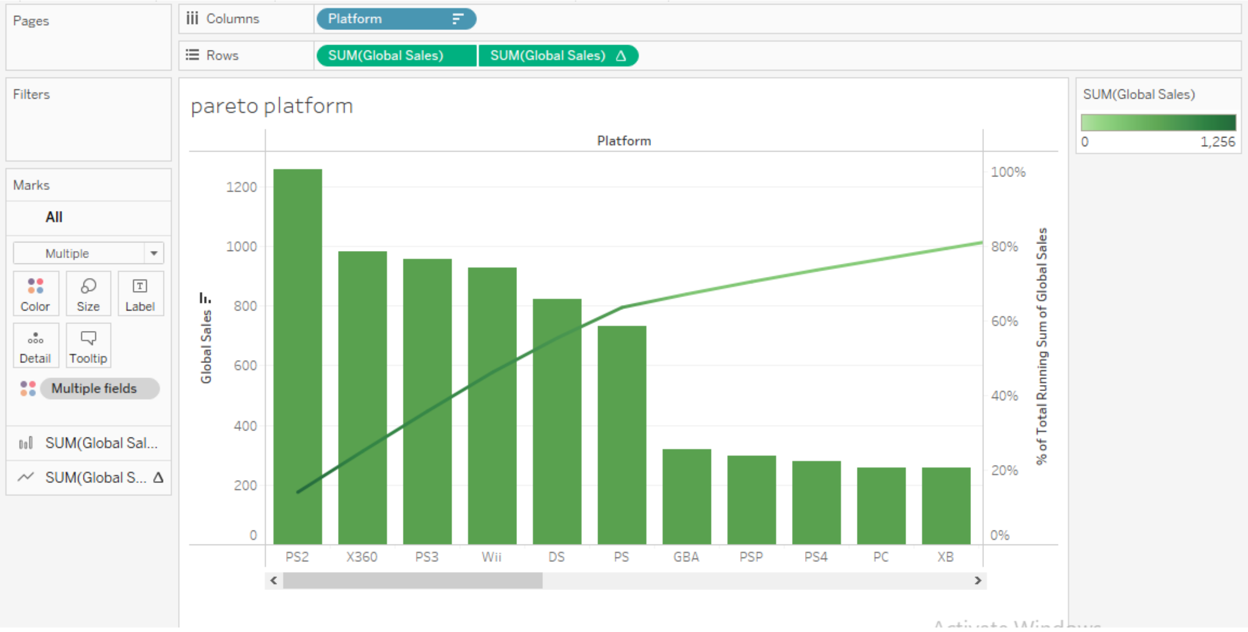


Rank of genre

Genre	Global Sales	Rank of Global ..
Action	1,751	1
Sports	1,331	2
Shooter	1,037	3
Role-Playing	927	4
Platform	831	5
Misc	810	6
Racing	732	7
Fighting	449	8
Simulation	392	9
Puzzle	245	10
Adventure	239	11
Strategy	175	12

Rank is calculated for the Genre based on Global sales
we can see ‘Action’ stands highest and ‘Strategy’ stands lowest.

- **Pareto for platform:**



Columns: Platform

Rows: SUM(Global Sales) SUM(Global Sales) △

Marks: All — Multiple

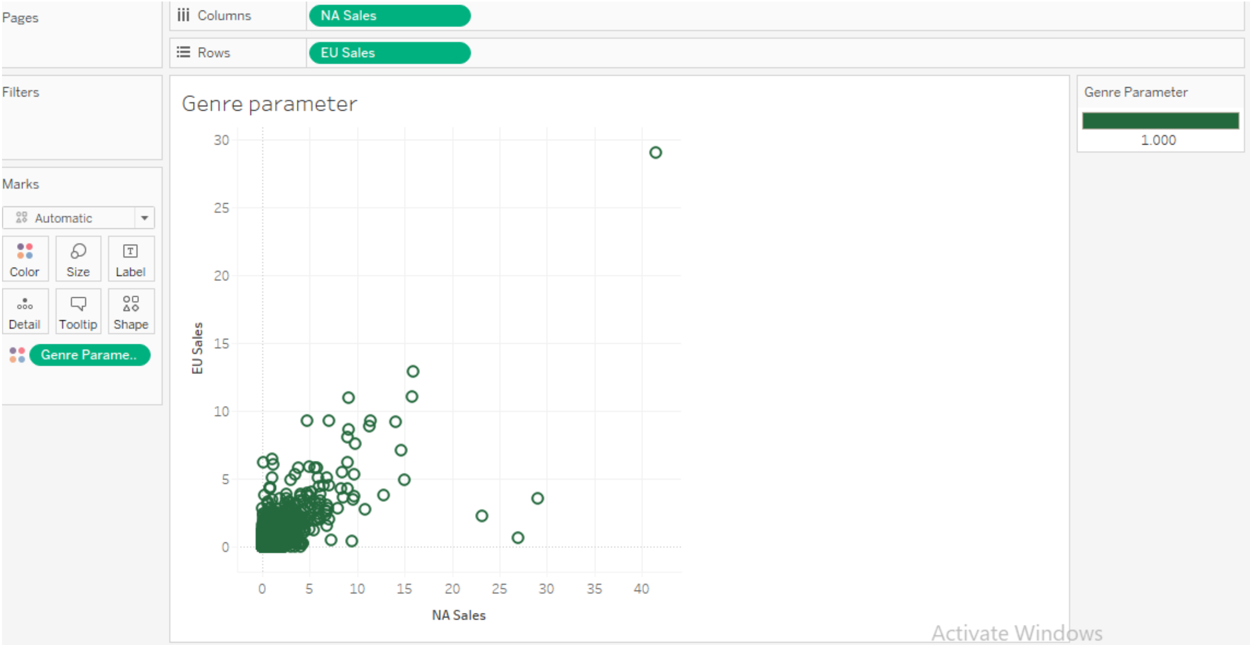
Multiple fields

SUM(Global Sal...

SUM(Global S... △

It is a 80-20% rule which calculates the cumulative total of the Global Sales across table in descending order for platforms.

- **Genre parameter:**



Columns: NA Sales

Rows: EU Sales

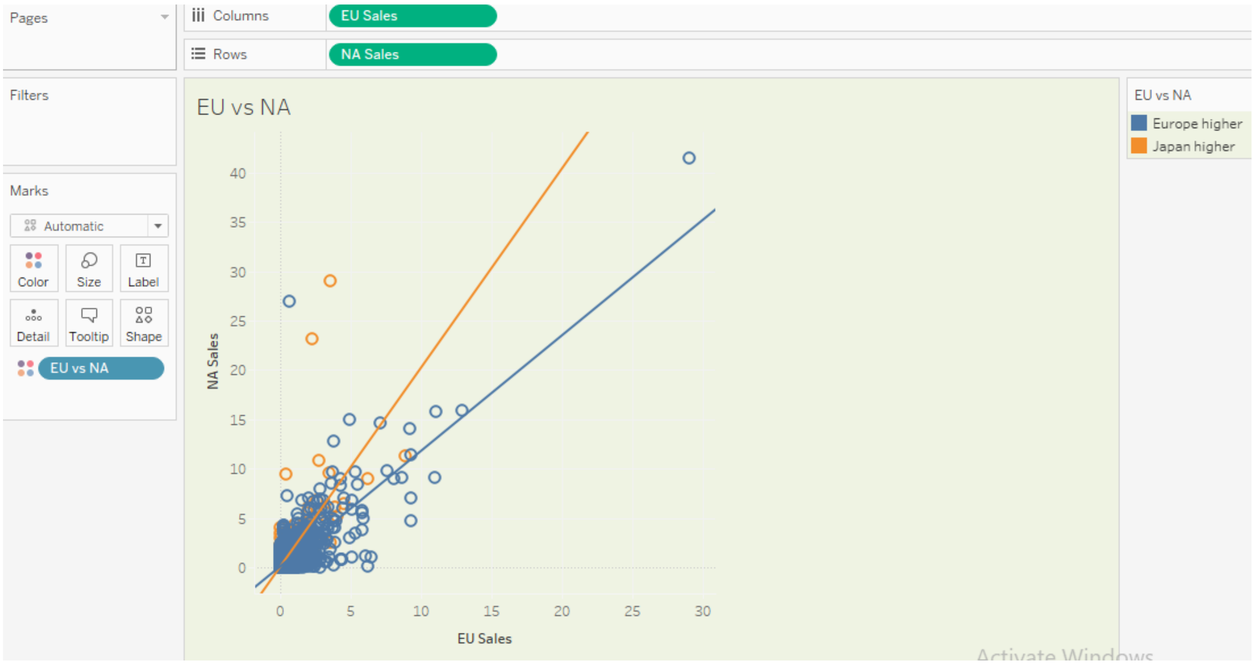
Marks: Automatic

Genre Parame..

Genre Parameter: 1.000

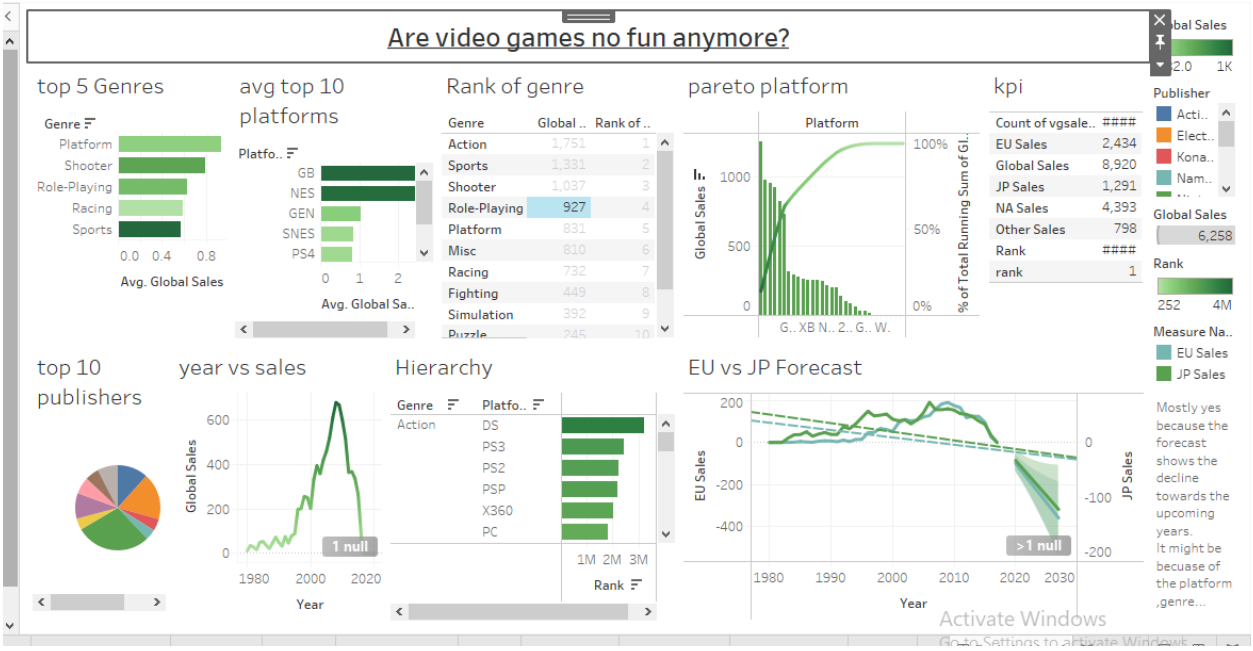
Created a 'Genre parameter' but it does not impact the data points as it is considering the default value itself.

- **EU vs NA:**



It is the comparision of Europe and North America .
We can observe that EU is having higher sales than NA

- ## Final Dashboard:



Are video games no fun anymore?

Rank of genre

Genre	Global ..	Rank of ..
Action	1,751	1
Sports	1,331	2
Shooter	1,037	3
Role-Playing	927	4
Platform	831	5
Misc	810	6
Racing	732	7
Fighting	449	8
Simulation	392	9
Puzzle	245	10

kpi

Count of vgsale..	####
EU Sales	2,434
Global Sales	8,920
JP Sales	1,291
NA Sales	4,393
Other Sales	798
Rank	####
rank	1

Mostly yes because the forecast shows the decline towards the upcoming years. It might be becuase of the platform ,genre...

FEW USEFUL INSIGHTS

The real question is 'Are video games no fun anymore'?

- Mostly yes, because the forecast shows the decline towards the upcoming years.It might be becuase of the platform,genre.
- We observe that even though 'Action' is ranked No. 1 in the highest sales, lowest is 'Platform'
- But the average of the global sales are highest in 'Platform' which means there was a time where 'Action' has shown its peak sales i-e; in 2009 and then it kept declining over years.
- For overall sales, year of sales is peak in 2008.
- Global sales kept decresing since the and it has fallen to 0.1 in 2020's. [started decreasing from 2013)
- So when we forecast 'Europe' and 'Japan' sales, forecast shows sales might decrease over time.

THANKYOU