

Reference:

GitHub link: https://github.com/saurabh-dixit-ds/Saurabh_Dixit_DA301_Assignment

Background:

Turtle Games, a game manufacturer and retailer sell their own products, as well as products manufactured by other companies. Their product range includes Lego board games, video games and toys. They have a global customer base and have a business objective of improving overall sales performance.

Business Case 1:

What price should be set for the Lego sets that have 8,000 Lego pieces?

Approach:

Data gathered in **lego.csv** file was **loaded, cleaned, prepared, and analyzed** to derive insights and predict price using **Simple Linear Regression**.

Data was **loaded** into a data frame named **df_lego** and viewed to understand the dimensions (rows and columns), datatypes and the general values it contained.

```
df_lego.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12261 entries, 0 to 12260
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   ages                  12261 non-null  int64
 1   list_price            12261 non-null  float64
 2   num_reviews           12261 non-null  int64
 3   piece_count           12261 non-null  int64
 4   play_star_rating      12261 non-null  float64
 5   review_difficulty     12261 non-null  int64
 6   country                12261 non-null  int64
dtypes: float64(2), int64(5)
memory usage: 670.6 KB
```

```
df_lego.describe()
```

	ages	list_price	num_reviews	piece_count	play_star_rating	review_difficulty	country
count	12261.00000	12261.000000	12261.000000	12261.000000	12261.000000	12261.000000	12261.000000
mean	16.68828	65.141998	14.603050	493.405921	3.709689	1.988826	10.015333
std	8.21868	91.980429	34.356847	825.364580	1.641130	1.787565	6.185450
min	0.00000	2.272400	0.000000	1.000000	0.000000	0.000000	0.000000
25%	11.00000	19.990000	1.000000	97.000000	3.600000	0.000000	4.000000
50%	19.00000	36.587800	4.000000	216.000000	4.400000	2.000000	10.000000
75%	23.00000	70.192200	11.000000	544.000000	4.700000	4.000000	15.000000
max	30.00000	1104.870000	367.000000	7541.000000	5.000000	5.000000	20.000000

```
df_lego.shape
```

```
(12261, 7)
```

Describing the data set

- Num of Rows : 12261
- Num of Columns : 7

As part of **data cleaning** process, records with missing values were deleted.

```
# Check for missing values  
df_lego.isna().sum()
```

```
ages                0  
list_price          0  
num_reviews         0  
piece_count         0  
play_star_rating    0  
review_difficulty   0  
country             0  
dtype: int64
```

```
# Check for null values  
df_lego.isnull().sum()
```

```
ages                0  
list_price          0  
num_reviews         0  
piece_count         0  
play_star_rating    0  
review_difficulty   0  
country             0  
dtype: int64
```

Sense check the data

- There are No missing values in any of the columns in the Lego data set

Min and Max values in lego data set

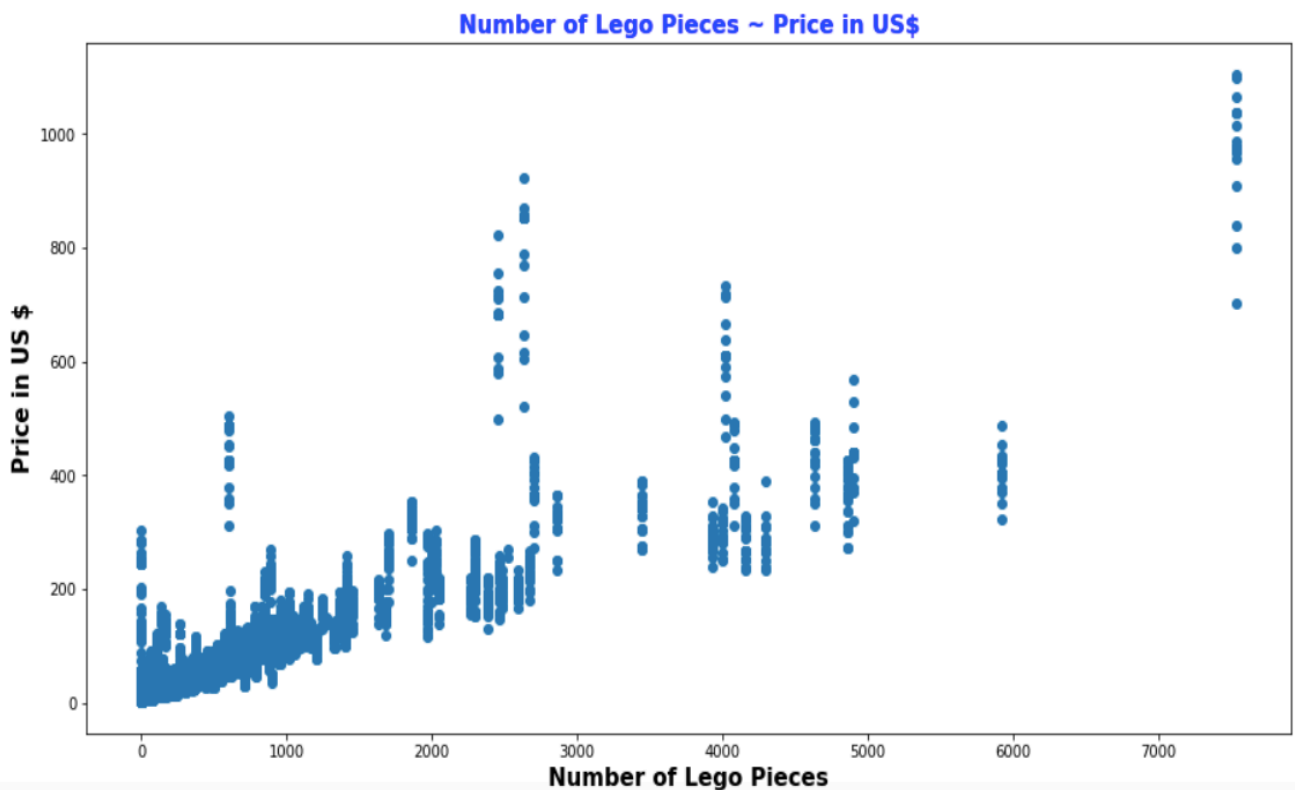
```
df_lego.min()
```

```
ages          0.0000
list_price    2.2724
num_reviews   0.0000
piece_count    1.0000
play_star_rating 0.0000
review_difficulty 0.0000
country       0.0000
dtype: float64
```

```
df_lego.max()
```

```
ages          30.00
list_price    1104.87
num_reviews   367.00
piece_count   7541.00
play_star_rating 5.00
review_difficulty 5.00
country       20.00
dtype: float64
```

Scatter plot below shows how data is distributed across different price ranges based on the number of pieces in the Lego set.



Key Observations

1. **Minimum Price** of Lego product is **2.2742 US\$**
2. **Maximum Price** is **1104.87 US\$**
3. **Majority (12075)** of the available Lego products are in the price range of 3 to 400 US\$
4. **Very few** Lego products with **list price > 400 US\$**
5. Price of Lego products goes **beyond 600 US\$** for the first time when the No. of Lego pieces is **between 2500 and 2700 pieces**

```
f = 'y ~ x'
ols_test = ols(f, data = df_lego).fit()

ols_test.summary()
```

OLS Regression Results

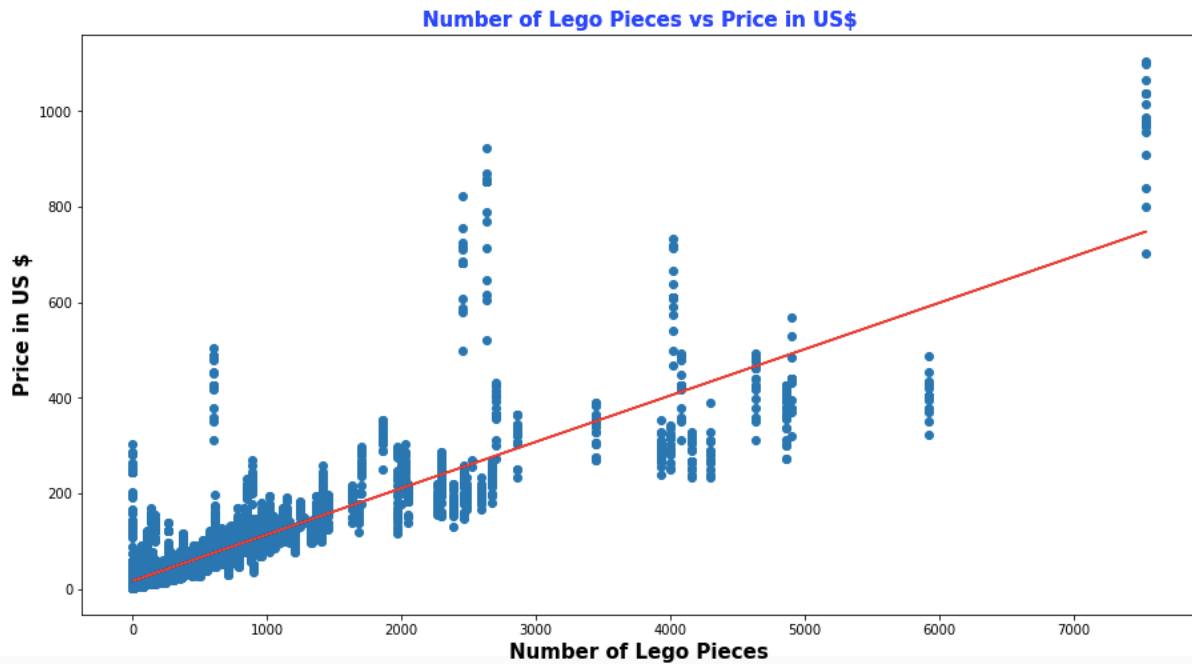
Dep. Variable:	y	R-squared:	0.756
Model:	OLS	Adj. R-squared:	0.756
Method:	Least Squares	F-statistic:	3.804e+04
Date:	Mon, 04 Jul 2022	Prob (F-statistic):	0.00
Time:	01:28:15	Log-Likelihood:	-64182.
No. Observations:	12261	AIC:	1.284e+05
Df Residuals:	12259	BIC:	1.284e+05
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
Intercept	17.3243	0.478	36.256	0.000	16.388	18.261
X	0.0969	0.000	195.027	0.000	0.096	0.098

Observations based on Simple Linear Regression Analysis

1. **R-squared (the coefficient of determination)** is **0.756**.
2. We have a **High** value of the **coefficient of determination**:
 - **Strong relationship** between the model and dependent variable **Price**
 - **75.6%** of the **Variation in the Price** is **explained by** the variation in the **Number of Pieces**
3. **Standard Error** is **0.478**

```
Text(0.5, 1.0, 'Number of Lego Pieces vs Price in US$')
```



Conclusion

Business Question 1 : What price should be set for the Lego sets that have 8,000 Lego pieces?

Answer : Based on the Regression Equation, **Predicted Price** for the Lego set having **8000 pieces** is : **792.5243 US \$**

Business Case 2:

What price should be set for all the Lego sets that have 8,000 Lego pieces and are most likely to be purchased by customers who are 30 years old?

Approach:

Since there are **two independent factors**, Lego pieces and the age group that influence the price, a **Multiple Linear Regression** model was built and checked for its effectiveness.

```
# create train and test data sets
# training dataset is 70% of the total dataset
# testing dataset is 30% of the total dataset
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.3, train_size=0.7)
```

```
multi = LinearRegression()
multi.fit(x_train.values, y_train.values)
```

```
▼ LinearRegression
LinearRegression()
```

```
multi.predict(x_train.values)
```

```
array([ 70.53380938,  53.74695744, 101.34393831, ...,  33.50291614,
        38.02457438,  36.76498915])
```

```
# Checking the value of R-squared, intercept and coefficients
print("R-squared: ", multi.score(x_train.values, y_train.values))
print("Intercept: ", multi.intercept_)
print("Coefficients:")
list(zip(x_train.values, multi.coef_))
```

```
R-squared:  0.7554519274419179
Intercept:  16.570832777690825
Coefficients:
```

```
[(array([544,  26]), 0.09594332369728356),
 (array([374,  19]), 0.06806955807883854)]
```

Key Observations

High positive value of **R-squared 0.7554** indicates a **strong relationship** and a good model for prediction

```
# make predictions
New_Value1 = 8000
New_Value2 = 30
print ('Predicted Value: \n', multi.predict([[New_Value1 ,New_Value2]]))
```

```
Predicted Value:
[ 786.1595091]
```

Question 2 : What price should be set for all the Lego sets that have 8,000 Lego pieces and are most likely to be purchased by customers who are 30 years old?

Answer : Predicted Price : 786.1595 US Dollars

Business Case 3:

What is the general sentiment of customers across all products?

Approach:

Feedback collected in games_reviews.csv file, was systematically loaded, sense checked, tokenized, and analyzed for sentiments and polarity.

Load & Sense check:

```
# import data into Python
df_game_rev = pd.read_csv('game_reviews.csv')
df_game_rev.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   overall               15000 non-null  int64
1   verified              15000 non-null  bool
2   reviewTime            15000 non-null  object
3   reviewerID            15000 non-null  object
4   reviewerName          15000 non-null  object
5   reviewText            14990 non-null  object
6   summary               14998 non-null  object
7   unixReviewTime        15000 non-null  int64
8   image                 160 non-null   object
dtypes: bool(1), int64(2), object(6)
memory usage: 952.3+ KB
```

```
df_game_rev['reviewText']
```

```
0      When it comes to a DM's screen, the space on t...
1      An Open Letter to GaleForce9*:\n\nYour unpaint...
2      Nice art, nice printing.  Why two panels are f...
3      Amazing buy! Bought it as a gift for our new d...
4      As my review of GF9's previous screens these w...
...
14995  Garbage.  Broke after 1 use.  Absolutely ridic...
14996  Our granddaughter loves these as part of her b...
14997  Got water in it after the first use. Shorted o...
14998  I like print vs digital scheduling.
```

Read > Pre-process > and Tokenize Text into Words:

```
# Look at one raw game review text and it's type
print(results_list_values[0])
type(results_list_values[0])
```

When it comes to a DM's screen, the space on the screen itself is at an absolute premium. The fact that 50% of this space is wasted on art (and not terribly informative or needed art as well) makes it completely useless. The only reason that I gave it 2 stars and not 1 was that, technically speaking, it can at least still stand up to block your notes and dice rolls. Other than that, it drops the ball completely.

str

```
# Split up each review into individual words
results_list_values_token = [word_tokenize(str(_)) for _ in results_list_values]
```

```
# Get a list of all english words so we can exclude anything that doesnt appear on the list
all_english_words = set(words.words())
```

```
# Some pre-processing:
#-- lets get every word
#-- lets convert it to lowercase
#-- only include if the word is alphanumeric and if it is in the list of English words.

results_list_values_token_nostop = \
[[y.lower() for y in x if y.lower() not in stop_words and y.isalpha() and y.lower() in all_english_words]\
 for x in results_list_values_token]
```

```
# Let's have a look at the same review as above
results_list_values_token_nostop[0]
```

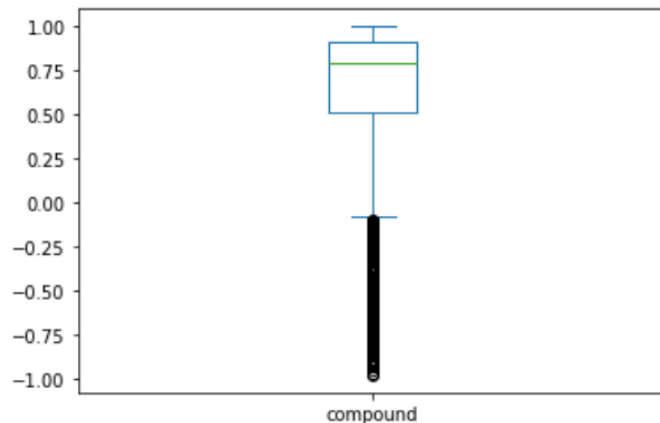
```
# Let's have a look at the same review as above  
results_list_values_token_nostop[0]
```

```
['comes',  
 'screen',  
 'space',  
 'screen',  
 'absolute',  
 'premium',  
 'fact',  
 'space',  
 'wasted',  
 'art',  
 'terribly',  
 'informative',  
 'art',  
 'well',  
 'completely',  
 'useless',  
 'reason',  
 'gave',  
 'technically',  
 'speaking',  
 'least',  
 'still',  
 'stand',  
 'block',  
 'dice',  
 'ball',  
 'completely']
```

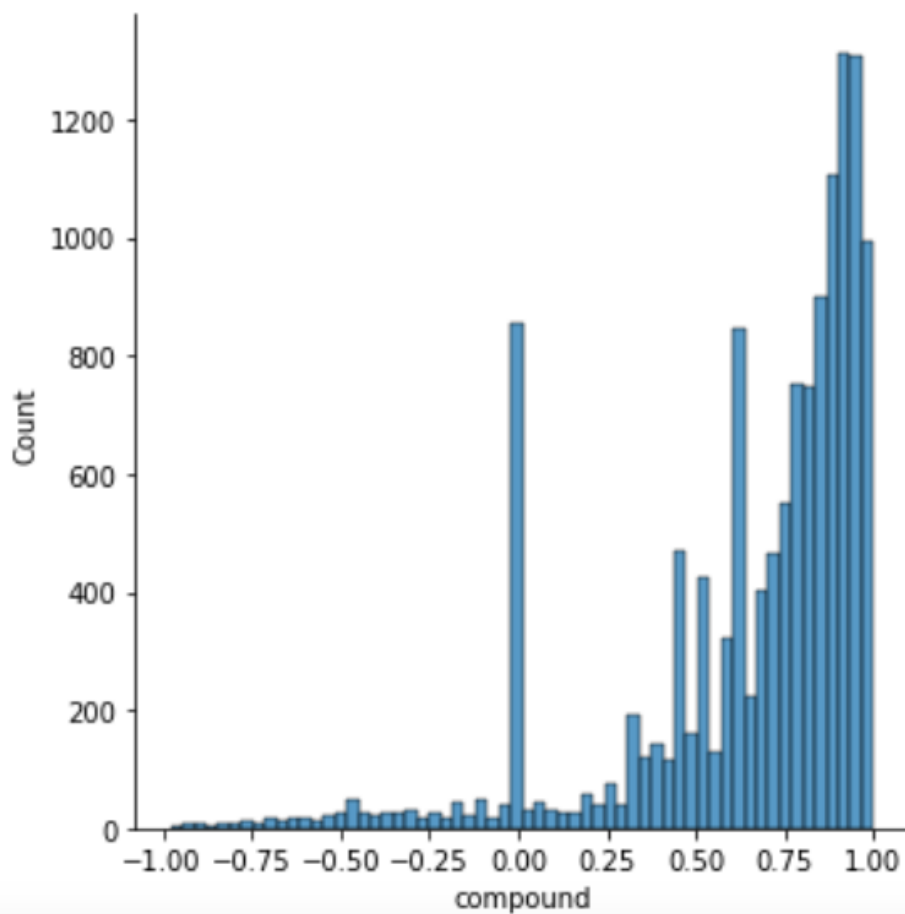
Perform Sentiment Analysis:

```
# The boxplot is a nice way to see how many values sit on the edges as outliers.  
_plot = polarity_pd.reset_index()['compound'].sort_values()  
_plot.plot(kind='box')
```

<AxesSubplot:>

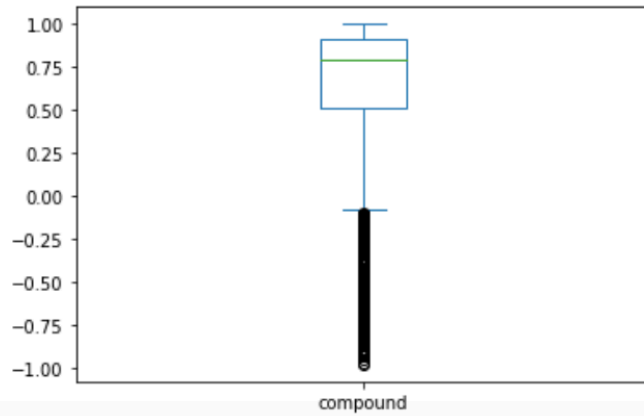


<seaborn.axisgrid.FacetGrid at 0x7faa750f3c40>

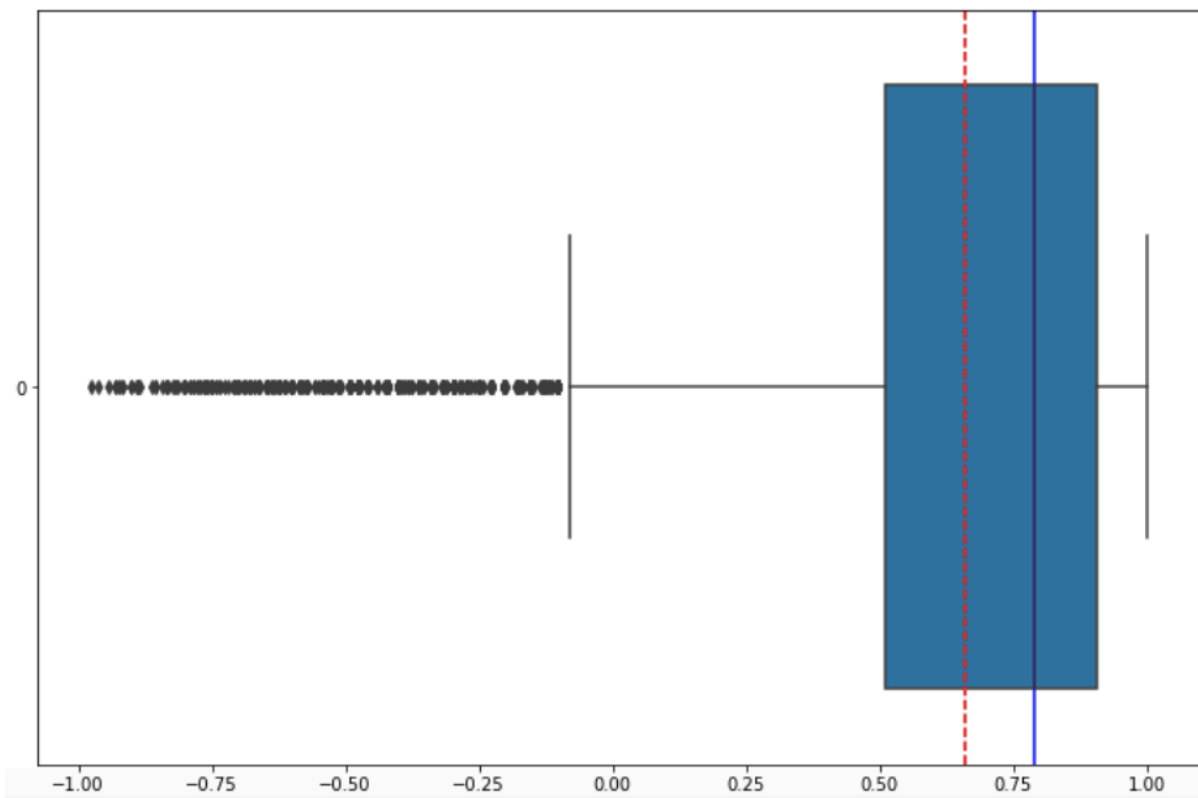


```
# The boxplot is a nice way to see how many values sit on the edges as outliers.
_plot = polarity_pd.reset_index()['compound'].sort_values()
_plot.plot(kind='box')
```

<AxesSubplot:>



<matplotlib.lines.Line2D at 0x7faa754339d0>



Key observations from above distribution plot

1. **Vast Majority** of the Reviews are **Positive** with the **Compound polarity scores** greater than **0.5**
2. **Left Skewed** distribution
3. **Lower/Negative Polarity score** values **on the left** and more **positive polarity scores** mostly on the **right end**

```
2]: mean=np.mean(_plot)
    print(mean)

0.659056308809365

3]: median=np.median(_plot)
    print(median)

0.7906
```

Key Observations

General Sentiment of customers across all Products is: **Positive** with the mean polarity score of **+0.6590**

Methodology and Analysis :

The Polarity scores of customer reviews across all products was obtained using vaderSentiment Analysis. These scores were normalized to obtain compound scores for all the reviews.

- The average(mean) polarity score is: **0.6590**
- The median(central value) polarity score is : **0.7906**

Business Case 4:

Based on the polarity of the sentiment, what are the top 20 positive and top 20 negative reviews?

Top 20 Positive reviews (based on sentiment polarity)

```
pd.set_option('display.max_colwidth', 200)
df_text_pol.sort_values(by='comp_polarity', ascending=False).head(20)
```

	comp_polarity	reviewtxt
12032	0.9999	Eminent Domain is a game by Seth Jaffee, published by Tasty Minstrel Games. It is for 2-4 players. In this game, players will be building an empire in space. They will be surveying new planets ...
3619	0.9998	When I first heard about Days of Wonder's newest game, Ticket to Ride (Days of Wonder, 2004 - Alan Moon), I was excited. But how could I not be - for all of Days of Wonders games so far have been ...
1121	0.9995	Disclaimer: Bought this from a local store. Paid list value, but supporting local game stores helps keep them in business, and it's a rough market to keep a gaming store running. Also, I do not ow...
7643	0.9995	The USA version of Ticket To Ride is fun but frustratingly cutthroat; I would not recommend getting the base set without the <a data-hook="product-link-linked" class="a-link-normal" href="/Da...
879	0.9994	Whenever I see this game on my shelf, I get a disturbing visual of Quark's big head from Star Trek: Deep Space Nine. I then picture him playing Tongo with a bunch of other Ferengi...a game that de...
12040	0.9994	So I went camping as kind of a chaperone with a youth group and learned to play this game at a picnic table. The occasional breeze made so many stacks of card perilous, but it would have been too...
1666	0.9993	If you only employ one creativity-enhancing resource for the rest of your life, make that resource the Ball of Whacks!\n\nBreakthroughs in effective creativity-inspiring methods seldom occur. Mos...
13990	0.9992	This is an in-depth review of the product "Robotech RPG Tactics Starter Board Game."\\n\\n1) OPENING THE BOX\\n\\nThe first thing you'll notice about Robotech: RPG Tactics is the sheer size of the box...
11188	0.9992	The game is easy to learn and explain. One person draws a card and selects a question from it to read out loud. Everyone else writes down an answer that they think the reader will think is the bes...
6425	0.9992	I needed that route! You just cut me off, now instead of destination cards giving me gobs of points, now they all count against me. Theres no way around! My nerves have never felt more vexed th...
3571	0.9992	Newest update: My daughter is now 6.5, and I still think this is one of the best things I've bought on Amazon. We recently saw one of those horrible ASPCA commercials, and she asked if we could s...
358	0.9991	This kit is AWESOME! My 5-year old daughter and I made the chihuahua dog first, and it came out looking exactly like the picture. Although I love crafts I'm not that great at them, so I was deligh...
1570	0.9991	As a dad of two boys Im always on the lookout for activities for us to do together. Something we can all enjoy and equally get into. We built a Da Vinci catapult, a siege tower, did some explori...
6461	0.9991	Originally posted at [...], a new idea everyday!\nGame- Ticket to Ride\nProducer-Days of Wonder\nPrice- \$35-45\nTL;DR- Theme doesn't stop this train! 92.5%\n\nBasics- Around America in seven days!...
7402	0.999	This game is extremely elegant, making it easy for those new to gaming to get into, but it also provides great opportunities for strategy that game lovers can appreciate. Everyone I have played it...
857	0.9989	I grew up playing Monopoly. Lots of people did. It's unfortunate, because there was this gem just sitting there, begging to be played, but passed over. Here's what was missed:\n\nTHEME\nThe gam...
2634	0.9988	Publisher: Set Enterprises Inc.\n\nPlayers: 2 6 players\n\nAges: 5 + adult\n\nPlaying Time: 40 minutes\n\nGame Mechanics: Hand management and set collection\n\nContents: 103 unique playing c...
12043	0.9987	Originally posted at [...], a new idea everyday!\n\nGame-Eminent Domain\nPrice-\$40\nProducer-Tasty Minstrel Games\nSet-Up/Play/Clean-Up-1 Hour\nTL;DR-Only a few faults make this game great instead...
1295	0.9987	Lords of Waterdeep was awesome, and Scoundrels of Skullport makes it even bigger, better, and crazier, like the original game on steroids. You get two expansions in one: Undermountain and Skullpor...
5368	0.9986	... not because it's not a great gateway game, but because there's an even better version out there.\n\nThe beauty of vanilla Ticket to Ride is its simplicity and accessibility for players of all ...

Top 20 Negative reviews (based on sentiment polarity)

```
In [67]: pd.set_option('display.max_colwidth', 200)
df_text_pol.sort_values(by='comp_polarity', ascending=False).tail(20)
```

	comp_polarity	reviewtxt
1015	-0.93	I wish I'd watched some of the gameplay videos of Ashardalon before buying. As others have mentioned, it's not an RPG, it's a board game. But as a board game, it's a particularly bad boardgame. It...
901	-0.93	Acquire is a great game of luck, strategy, and (like monopoly) can teach people about capitalism. This new updated version from Hasbro does just that. It is a sadly cheap and flimsy copy of the or...
14977	-0.9311	This bath toy needs to taken off the market! Its worthless in every way! It never fit together. It began leaking immediately. Within 5 minutes it was a dead toy. My two little ones were so disappo...
12203	-0.9331	For those who are fascinated by the Ray Kurzweil future of superintelligent machines and transhumanism, this RPG is the one for you. In the untold future, mankind has expanded out into the solar s...
3628	-0.9337	Really excited to receive my game. But unfortunately the outside of the box was damaged when I received it. A corner is basically ripped off. Im sort of a board game freak so this made me sad.
363	-0.9349	I found that this card game does the opposite of what it was intended for. It actually has the kids focusing on ways to get angry, etc. instead of teaching how to be calm and act better. It really...
10755	-0.9357	We heard about Elf on a Shelf from some friends and loved this concept! I went to the Hallmark store and purchased one. We loved the book but were so disappointed with the quality of the Elf. I...
3964	-0.9366	We opened this gift for Christmas today but had bought on Prime Day. The box was damaged to where it barely holds things together. They also failed to include all of the color cards, so we cant pl...
11163	-0.9441	I have now use this deck a few times, and while I was saddened that all the pieces were card board and SMALLER the the regular sets, I love playing with Pooky. I have struggled finding the right c...
10640	-0.9484	Okay, here is the real deal on this product... The story is cute, the website is also cute, the doll is not the cutest by today's standards but is very reminiscent of vintage Christmas ornaments. T...

12173	-0.9493	This is the book that introduced me to several futurist concepts, including transhumanism, sousveillance, and the idea of death itself being an inconvenience rather than a tragedy. For those alone...
281	-0.952	I bought this thinking it would be really fun , but I was disappointed . It's really messy and it isn't nearly as easy as it seems. Also, the glue is useless.\nFor a 9 year old the instructions ar...
12172	-0.9587	The quality of this game's mechanics vary wildly.\nIn a sci-fi game where people are programs uploaded into bio-mechanical brains, hacking is garbage. You would think you could hack a person's bra...
14097	-0.9606	I was also disappointed with this puzzle, which my husband gave me for Christmas. While the puzzle-building itself was enjoyably challenging, I was fortunate to be finishing it in a bright room l...
10591	-0.9661	Worst ever idea. I have no problem telling my kids that St. Nick visits every kid's house in just one night using slower-than-light even-toed ungulate technology, but the idea of Claus Inc. outsou...
10428	-0.9701	Trust me, you can't win with this product. My son loved it at 6 & 7, but the last few days last year he started to get scared. Thankfully Christmas came around and Albert went home. Fast forwar...
1559	-0.973	The One Ring is a very innovative RPG set in Middle Earth between the time of The Hobbit and The Lord of the Rings.\n\nIn order to play it you need occasional reference to die rolls. Specifically,...
13637	-0.9823	If i could give this less than one star, i would. knowing that it was used, i had expected some wear and use, but this is ridiculous. some of the pages are missing, there is scribbling in crayon a...
1119	-0.9877	Here is my review, cross-posted from boardgamegeek.com:\n\nI have fond memories of D&D from my youth, that I occasionally attempt to recapture. I remember the sense of vague foreboding conjured by...
6613	-0.9889	It needs changes in four of its rules, and reminds me of Risk, where I could potentially lose friends. The game takes about 1 hour to play. The only good thing I can say is the artwork is beauti...

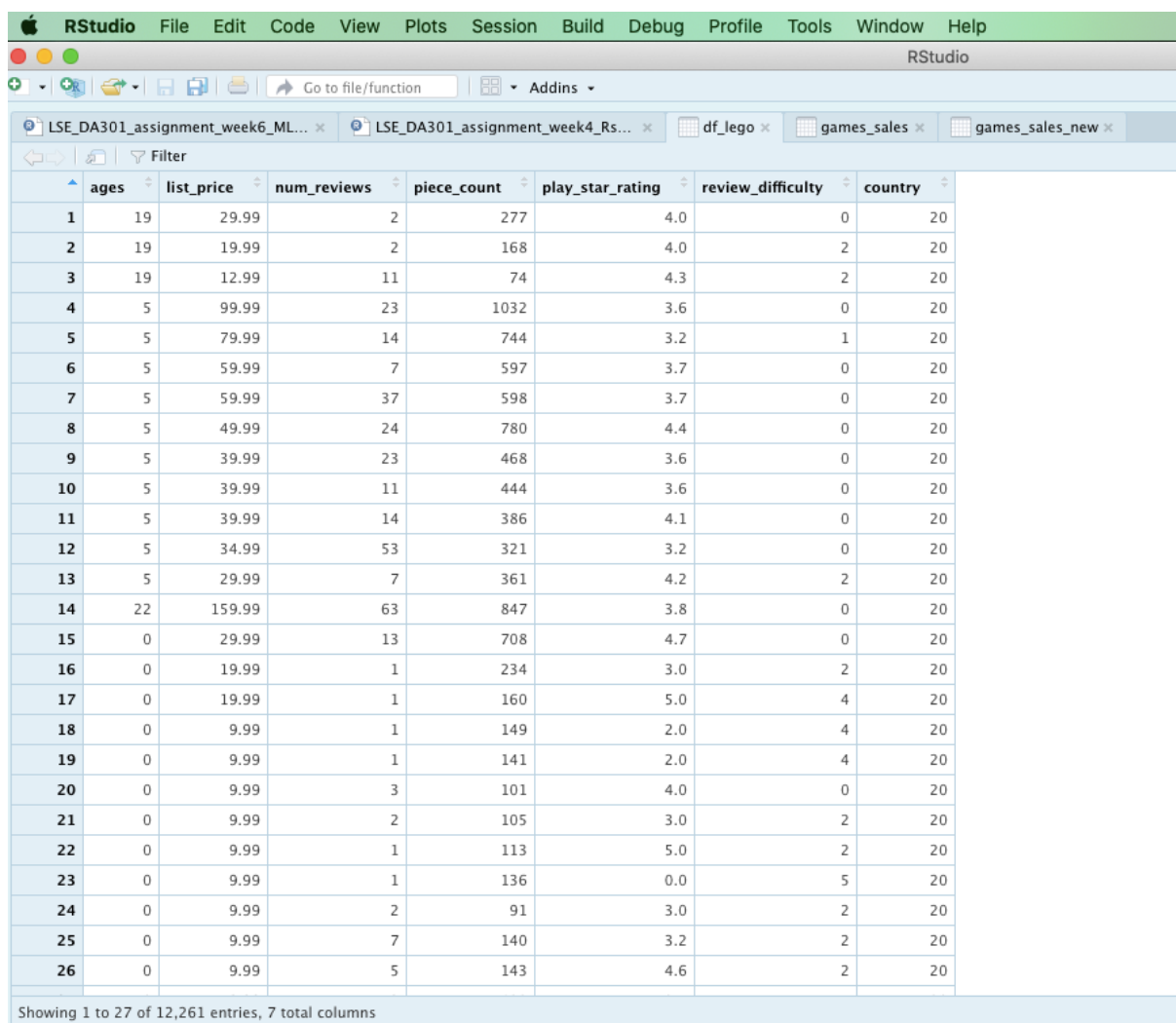
Business Case 5:

Which age group submits the most reviews?

Approach:

Loaded, cleaned, and analyzed the data to determine the age group that submits the most review and is most likely to submit the reviews.

Load and View:



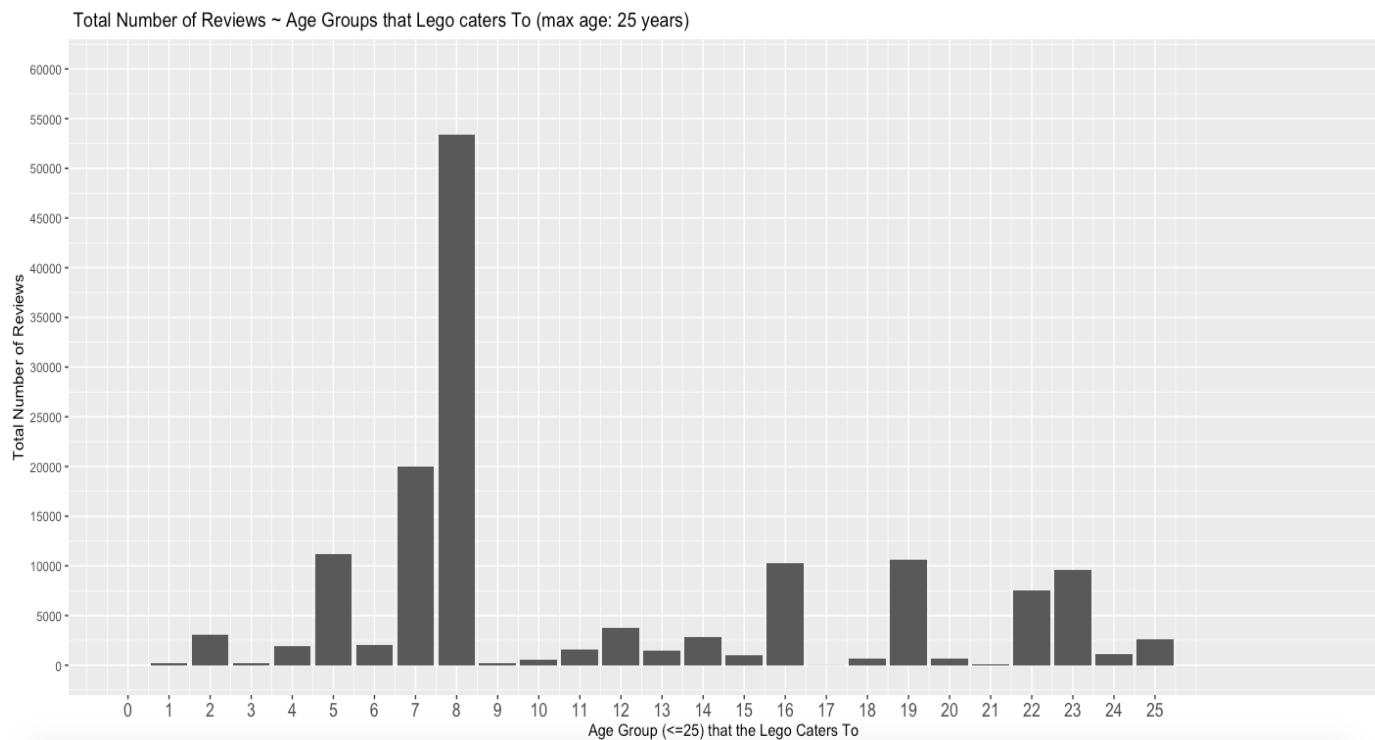
	ages	list_price	num_reviews	piece_count	play_star_rating	review_difficulty	country
1	19	29.99	2	277	4.0	0	20
2	19	19.99	2	168	4.0	2	20
3	19	12.99	11	74	4.3	2	20
4	5	99.99	23	1032	3.6	0	20
5	5	79.99	14	744	3.2	1	20
6	5	59.99	7	597	3.7	0	20
7	5	59.99	37	598	3.7	0	20
8	5	49.99	24	780	4.4	0	20
9	5	39.99	23	468	3.6	0	20
10	5	39.99	11	444	3.6	0	20
11	5	39.99	14	386	4.1	0	20
12	5	34.99	53	321	3.2	0	20
13	5	29.99	7	361	4.2	2	20
14	22	159.99	63	847	3.8	0	20
15	0	29.99	13	708	4.7	0	20
16	0	19.99	1	234	3.0	2	20
17	0	19.99	1	160	5.0	4	20
18	0	9.99	1	149	2.0	4	20
19	0	9.99	1	141	2.0	4	20
20	0	9.99	3	101	4.0	0	20
21	0	9.99	2	105	3.0	2	20
22	0	9.99	1	113	5.0	2	20
23	0	9.99	1	136	0.0	5	20
24	0	9.99	2	91	3.0	2	20
25	0	9.99	7	140	3.2	2	20
26	0	9.99	5	143	4.6	2	20

Showing 1 to 27 of 12,261 entries, 7 total columns

```
Console Terminal Jobs
R 4.1.2 ~ /
> #View the dataset in a tabular format
> View(df_lego)
> #Look up the overall Structure of the dataset
> str(df_lego)
'data.frame': 12261 obs. of 7 variables:
 $ ages      : int  19 19 19 5 5 5 5 5 5 ...
 $ list_price : num  30 20 13 100 80 ...
 $ num_reviews : int  2 2 11 23 14 7 37 24 23 11 ...
 $ piece_count : int  277 168 74 1032 744 597 598 780 468 444 ...
 $ play_star_rating : num  4 4 4.3 3.6 3.2 3.7 3.7 4.4 3.6 3.6 ...
 $ review_difficulty: int  0 2 2 0 1 0 0 0 0 0 ...
 $ country     : int  20 20 20 20 20 20 20 20 20 20 ...
> # sum of missing values
> sum(is.na(df_lego))
[1] 0
> # delete all the records with missing values
> df_lego <- na.omit(df_lego)
> head(df_lego)
  ages list_price num_reviews piece_count play_star_rating review_difficulty country
1  19    29.99         2         277           4.0              0          20
2  19    19.99         2         168           4.0              2          20
3  19    12.99        11          74           4.3              2          20
4   5     99.99        23        1032           3.6              0          20
5   5     79.99        14         744           3.2              1          20
6   5     59.99         7         597           3.7              0          20
> |
```

Business case 6:

Which are the most popular (i.e. with the most number of reviews) Lego sets purchased by customers who are at the most 25 years old (≤ 25 years)



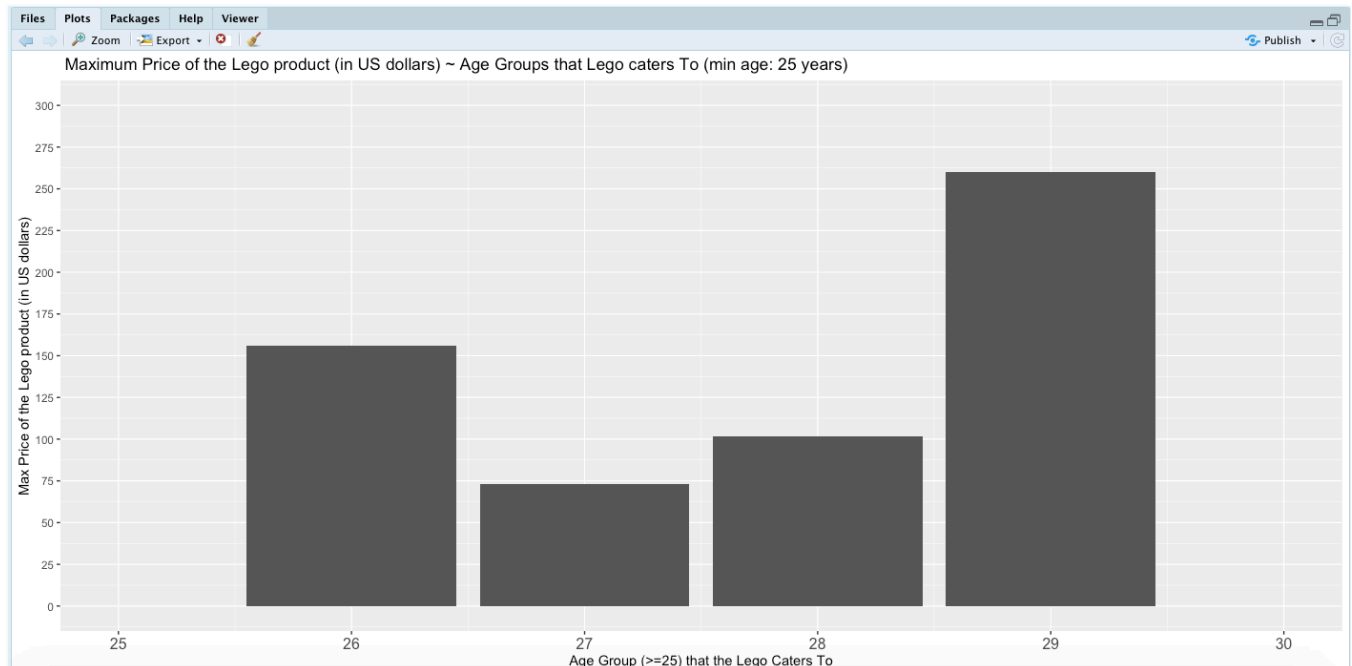
The customer group that will most likely leave a review on the products is in the Age Group:
**** Age: 8 ****

Additional facts, this specific customer group

- **420** records of reviews for Age group 8
- Has mostly given play star rating of 4.0 or above
- Has review difficulty 0 or 1 (mostly 1)

Business Case 7:

What is the most expensive Lego set purchased by customers who are at least 25 years old (≥ 25 years)?



```
# ANS : The most expensive Lego set purchased by customers who are at least 25 years old ( $\geq 25$  years)
#       has the List price : 259.87 US$
#       Further details are as below:
#       ages list_price num_reviews piece_count play_star_rating review_difficulty country
# -----
#       29      259.87         6      1413         4.3             0      16
```

Business case 8:

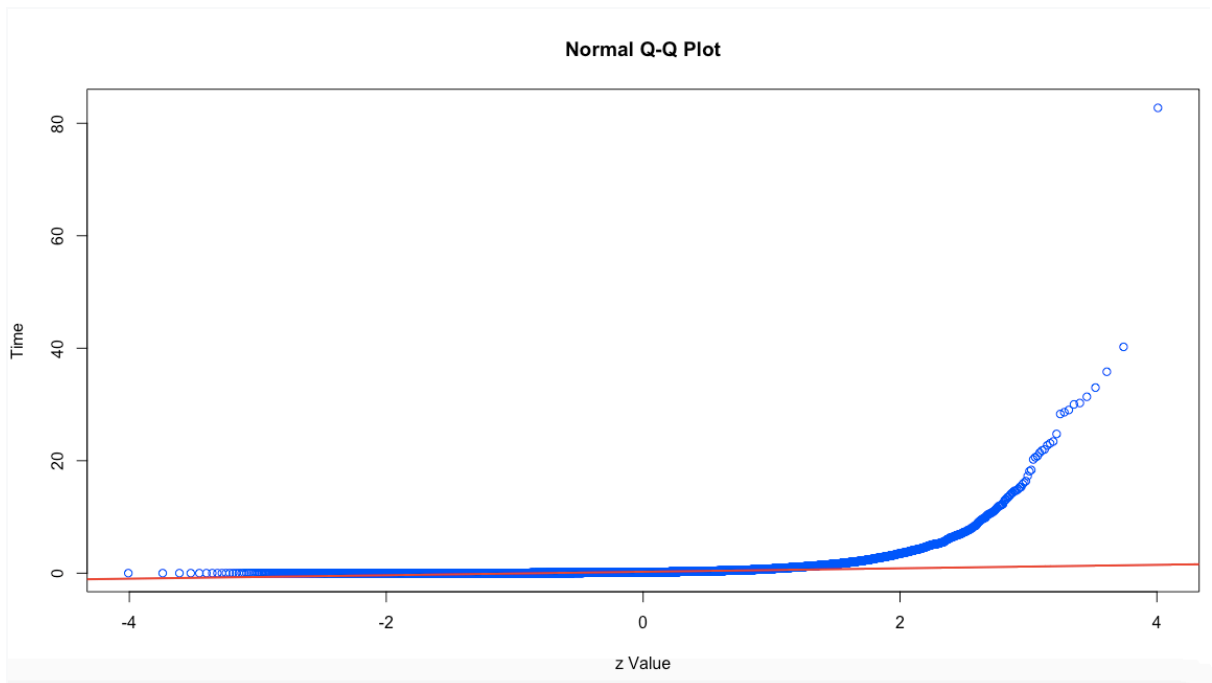
Perform exploratory data analysis (EDA) of Games Sales Data set to derive Insights

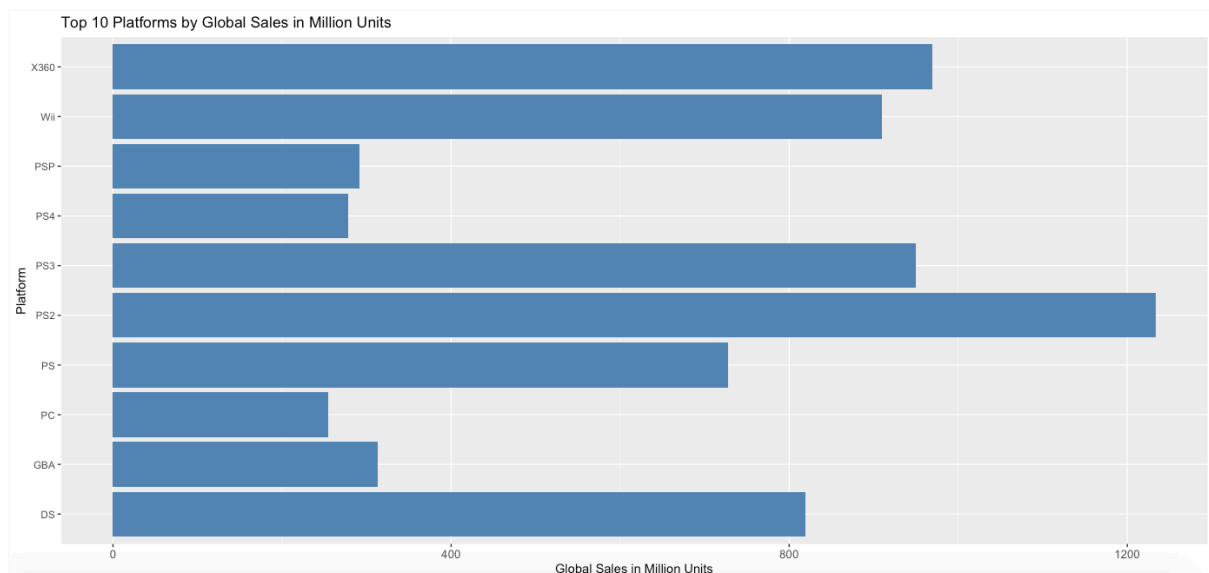
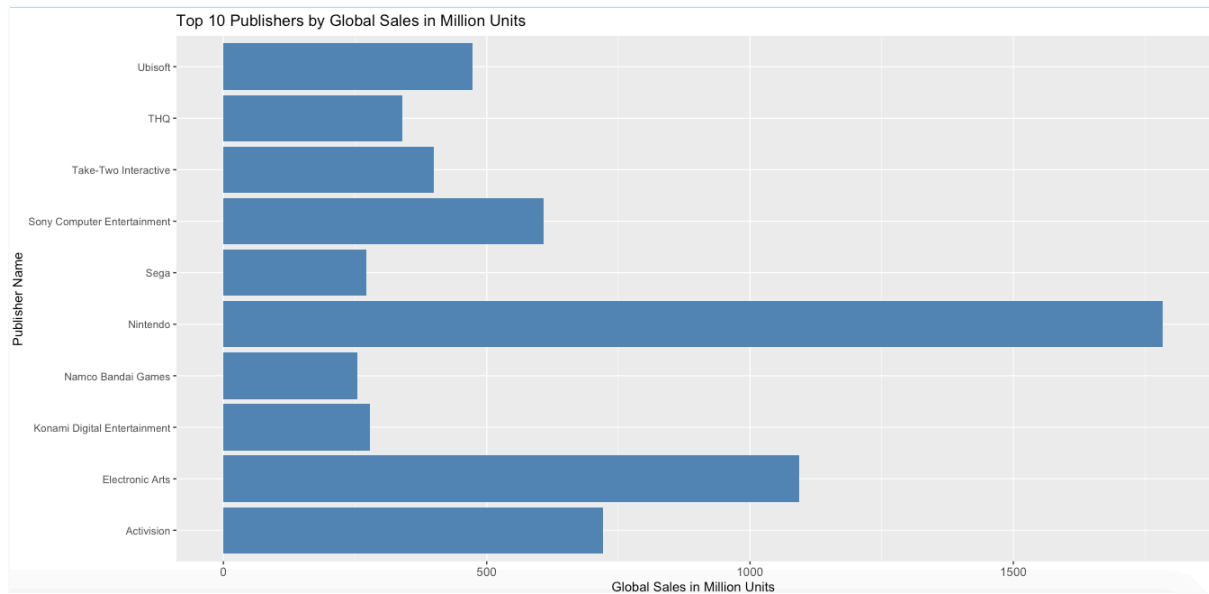
Approach:

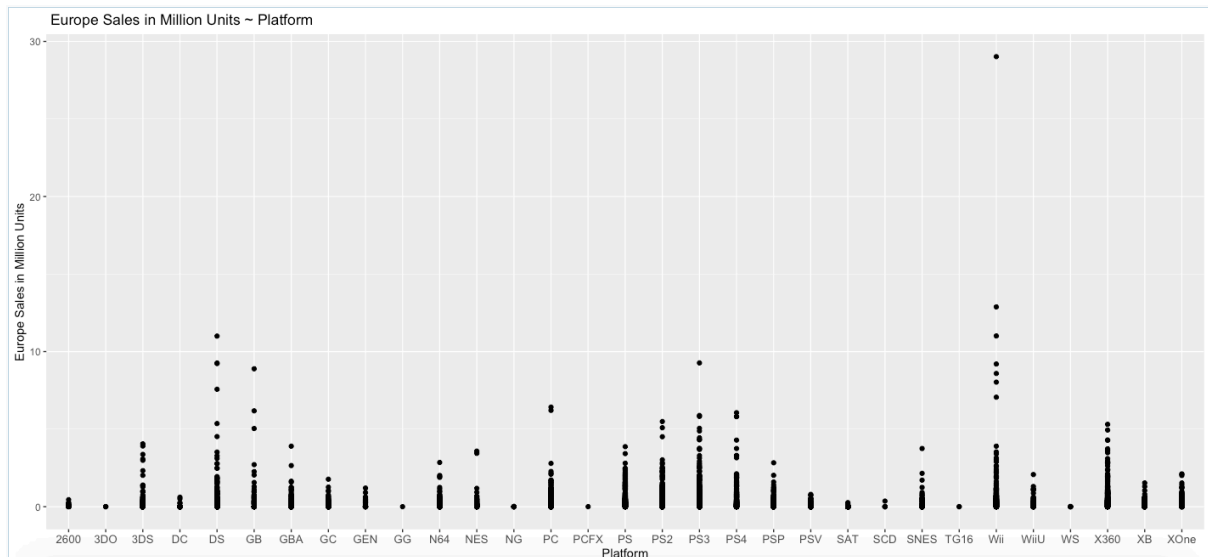
Loaded/Cleaned and analysed the data as below

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Global_Sales
1	1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	82.74
2	2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	40.24
3	3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	35.82
4	4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	33.00
5	5	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	31.37
6	6	Tetris	GB	1989	Puzzle	Nintendo	23.20	2.26	30.26
7	7	New Super Mario Bros.	DS	2006	Platform	Nintendo	11.38	9.23	30.01
8	8	Wii Play	Wii	2006	Misc	Nintendo	14.03	9.20	29.02
9	9	New Super Mario Bros. Wii	Wii	2009	Platform	Nintendo	14.59	7.06	28.62
10	10	Duck Hunt	NES	1984	Shooter	Nintendo	26.93	0.63	28.31
11	11	Nintendogs	DS	2005	Simulation	Nintendo	9.07	11.00	24.76
12	12	Mario Kart DS	DS	2005	Racing	Nintendo	9.81	7.57	23.42
13	13	Pokemon Gold/Pokemon Silver	GB	1999	Role-Playing	Nintendo	9.00	6.18	23.10
14	14	Wii Fit	Wii	2007	Sports	Nintendo	8.94	8.03	22.72
15	15	Wii Fit Plus	Wii	2009	Sports	Nintendo	9.09	8.59	22.00

Showing 1 to 15 of 16,598 entries, 9 total columns



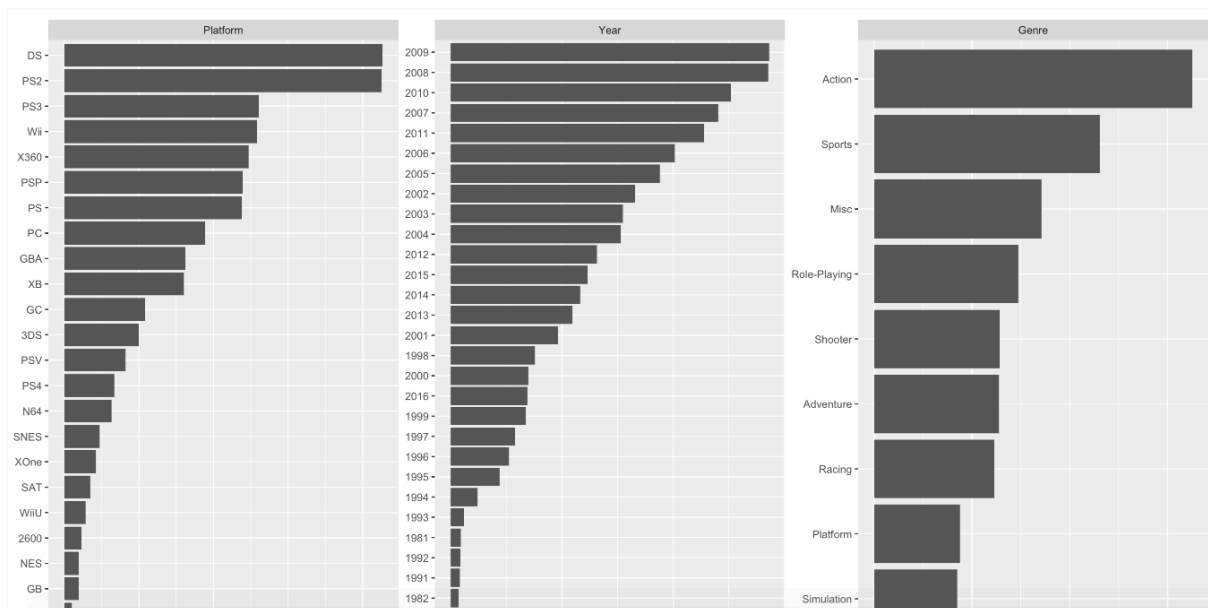




Basic Statistics

Raw Counts

Name	Value
Rows	16,327
Columns	9
Discrete columns	5
Continuous columns	4
All missing columns	0
Missing observations	0
Complete Rows	16,327
Total observations	146,943
Memory allocation	2 Mb



Key Insights

- **577** : Unique Publishers
- **11360**: Unique Game Titles
- Some Publishers have produced and launched more than one game from their studio
- **DS**: is the most frequently seen platform. It is a handheld game console from Nintendo
- **PlayStation 2** and **PlayStation 3** come next in popularity after DS
- **Action**: is the most popular Genre
- **Year 2009** : saw the maximum number of Game launches

ADDITIONAL Insights gained from EDA

- Total 577 Publishers
- Top 5 Publishers in terms of Global Sales are:
 1. Nintendo
 2. Electronic Arts
 3. Activision
 4. Sony Computer Entertainment
 5. Ubisoft
- Top 5 Platforms in terms of Global Sales in million units are :

Platform	Global Sales (in million units)
-----	-----
1. PS2	1233.
2. X360	970.
3. PS3	949.
4. Wii	910.
5. DS	819.

- Distinct Gaming Platforms : 31

Business case 9:

Determine the optimal global sales for all the video games based on EU Sales and North America Sales

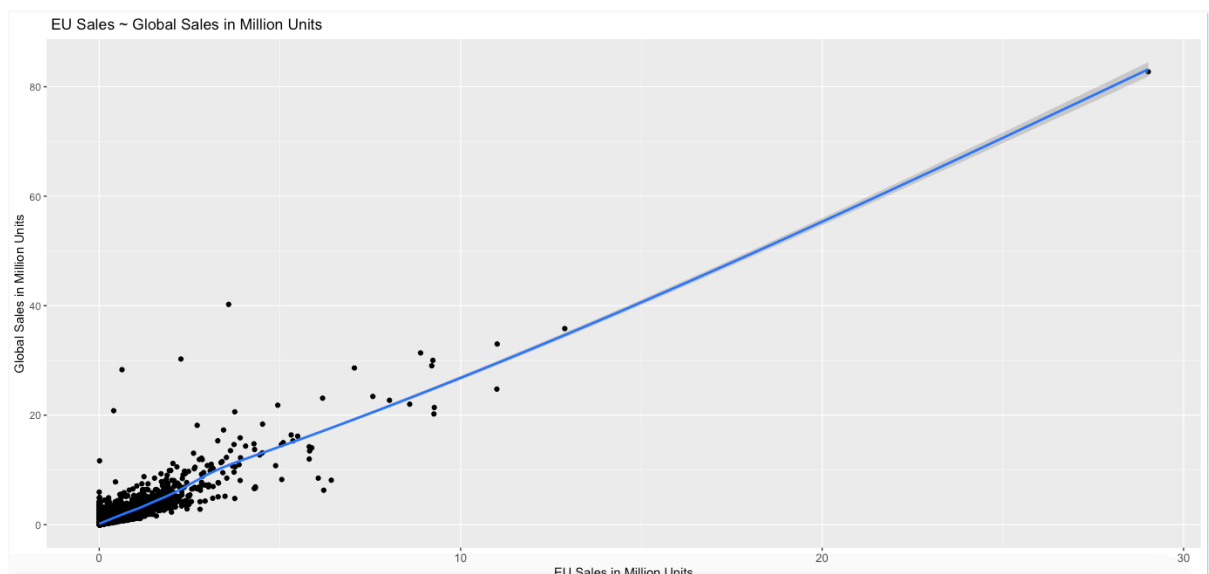
Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Global_Sales
1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	82.74
2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	40.24
3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	35.82
4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	33.00
5	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	31.37
6	Tetris	GB	1989	Puzzle	Nintendo	23.20	2.26	30.26
7	New Super Mario Bros.	DS	2006	Platform	Nintendo	11.38	9.23	30.01
8	Wii Play	Wii	2006	Misc	Nintendo	14.03	9.20	29.02
9	New Super Mario Bros. Wii	Wii	2009	Platform	Nintendo	14.59	7.06	28.62
10	Duck Hunt	NES	1984	Shooter	Nintendo	26.93	0.63	28.31
11	Nintendogs	DS	2005	Simulation	Nintendo	9.07	11.00	24.76
12	Mario Kart DS	DS	2005	Racing	Nintendo	9.81	7.57	23.42
13	Pokemon Gold/Pokemon Silver	GB	1999	Role-Playing	Nintendo	9.00	6.18	23.10
14	Wii Fit	Wii	2007	Sports	Nintendo	8.94	8.03	22.72
15	Wii Fit Plus	Wii	2009	Sports	Nintendo	9.09	8.59	22.00

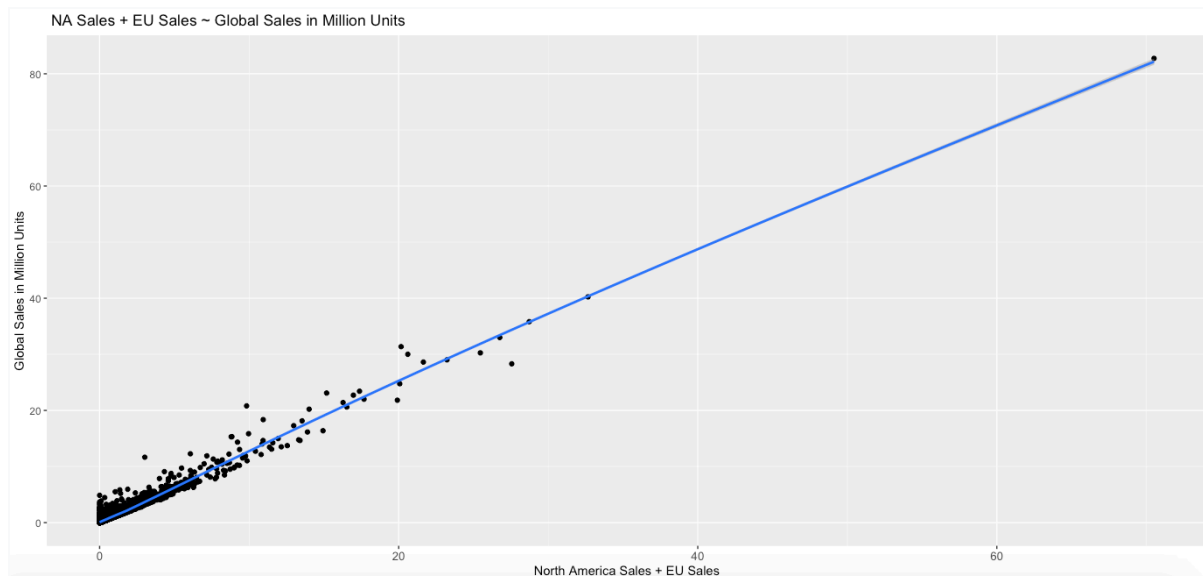
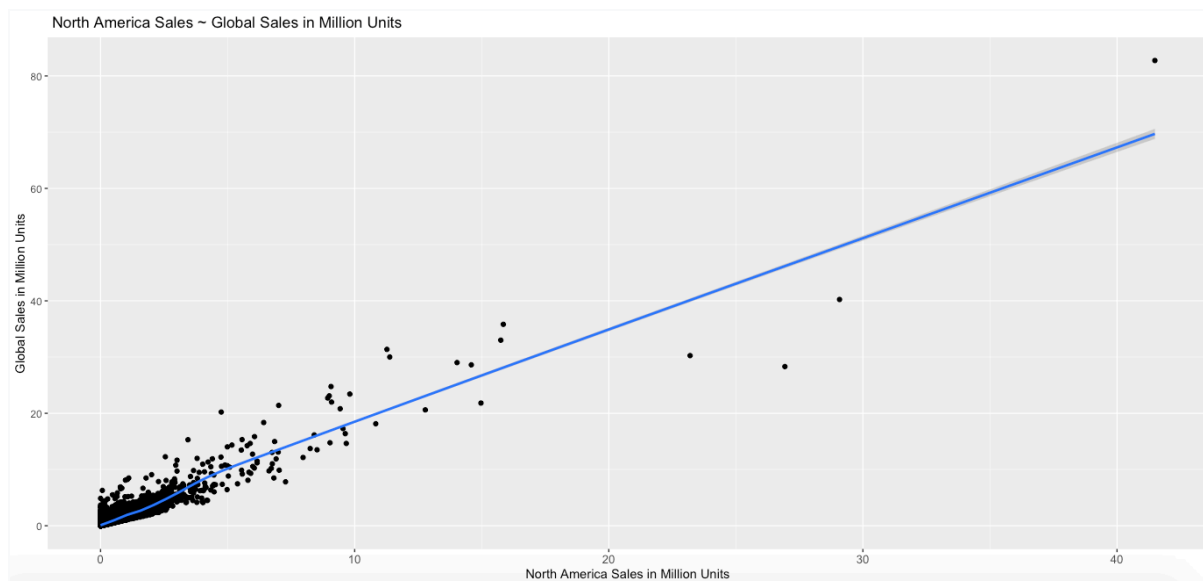
Showing 1 to 15 of 16,598 entries, 9 total columns

```

> # delete all the records with missing values
> games_sales_new <- na.omit(games_sales)
> head(games_sales_new)
  Rank Name Platform Year Genre Publisher NA_Sales EU_Sales Global_Sales
1    1   Wii Sports   Wii 2006   Sports  Nintendo   41.49   29.02    82.74
2    2 Super Mario Bros. NES 1985 Platform  Nintendo   29.08    3.58    40.24
3    3   Mario Kart Wii   Wii 2008   Racing  Nintendo   15.85   12.88    35.82
4    4   Wii Sports Resort Wii 2009   Sports  Nintendo   15.75   11.01    33.00
5    5 Pokemon Red/Pokemon Blue GB 1996 Role-Playing Nintendo   11.27    8.89    31.37
6    6      Tetris      GB 1989   Puzzle  Nintendo   23.20    2.26    30.26
> dim(games_sales_new)
[1] 16598    9
> sum(is.na (games_sales_new))
[1] 0
> # Still some N/A values in Year, filter those out and store only the records that Do NOT HAVE "N/A" in Year column
> games_sales_new <- games_sales_new[games_sales_new["Year"]!="N/A",]
> dim(games_sales_new)
[1] 16327    9

```





```
> cor(games_sales_new$NA_Sales, games_sales_new$Global_Sales)
[1] 0.9412677
> # Check correlation coefficient between Global Sales and European Sales
> cor(games_sales_new$EU_Sales, games_sales_new$Global_Sales)
[1] 0.903271
> # Check correlation coefficient between Global Sales and European Sales + N-America Sales
> cor(games_sales_new$NA_Sales + games_sales_new$EU_Sales, games_sales_new$Global_Sales)
[1] 0.9818667
```

```
summary(model1) # Print the summary statistics.
print(coefficients(model1)) # Look up coefficients from model 1
## Plot RESIDUALS for model1
hist(residuals(model1), col = 'steel blue')

|### STATISTICAL OBSERVATIONS FROM MODEL-1 SUMMARY
## Coefficients:
#Estimate Std. Error t value Pr(>|t|)
#(Intercept) 0.035512 0.002420 14.67 <2e-16 ***
# NA_Sales 1.150282 0.004377 262.83 <2e-16 ***
# EU_Sales 1.351483 0.007068 191.22 <2e-16 ***
# Multiple R-squared: 0.9648, Adjusted R-squared: 0.9648
# F-statistic: 2.238e+05 on 2 and 16324 DF, p-value: < 2.2e-16

# OBSERVATIONS
# Multiple R-square: 0.9648 *** indicates a STRONG LINEAR RELATIONSHIP between NA_Sales, EU_Sales AND Global_Sales
# 96.48% of the variations in y (Global_Sales) can be explained by the predictor / independent variables NA_Sales and EU_Sales
# p-value < 0.05, which means the model is statistically significant
```

Key INSIGHTS

- **Strong and Positive Correlation** between
 1. North America Sales and Global Sales, with **correlation coefficient: 0.9412**
 2. Europe Sales and Global Sales, with **correlation coefficient: 0.9032**
- **Very strong and Positive Correlation** between North America Sales + EU Sales AND Global Sales, with the **correlation coefficient: 0.9818**

Model Building:

Consider strong positive correlations, a model was built with Global sales as y (Dependent variable) and NA + EU sales as two Independent variables (x1 and x2)

Accuracy of the model was assessed by reviewing model summary

Model1 Summary

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.035512 0.002420 14.67 <2e-16 ***

NA_Sales 1.150282 0.004377 262.83 <2e-16 ***

EU_Sales 1.351483 0.007068 191.22 <2e-16 ***

Multiple R-squared: 0.9648, Adjusted R-squared: 0.9648

F-statistic: 2.238e+05 on 2 and 16324 DF, p-value: < 2.2e-16

Model Assessment summary

- Multiple R-square: 0.9648: indicates a strong linear relationship between NA_Sales, EU_Sales AND Global_Sales
- 96.48% of the variations in y (Global_Sales) can be explained by the predictor / independent variables NA_Sales and EU_Sales
- p-value < 0.05, which means the model is statistically significant

Using the coefficients obtained from the model, a Multiple Linear Regression equation was formed with the formula

MLR equation

```
predicted_global_sales <- intercept +  
  NA_sales_coef * games_sales_new$NA_Sales +  
  EU_sales_coef * games_sales_new$EU_Sales
```

Using above MLR equation, the **predicted global sales** for all the products (Games) were obtained and are shown in below table.

Rank	Game Title	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Global_Sales	Predicted_Global_Sales
1	1 Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	82.74	86.9808
2	2 Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	40.24	38.3240
3	3 Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	35.82	35.6746
4	4 Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	33.00	33.0323
5	5 Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	31.37	25.0139
6	6 Tetris	GB	1989	Puzzle	Nintendo	23.20	2.26	30.26	29.7764
7	7 New Super Mario Bros.	DS	2006	Platform	Nintendo	11.38	9.23	30.01	25.5999
8	8 Wii Play	Wii	2006	Misc	Nintendo	14.03	9.20	29.02	28.6076
9	9 New Super Mario Bros. Wii	Wii	2009	Platform	Nintendo	14.59	7.06	28.62	26.3596
10	10 Duck Hunt	NES	1984	Shooter	Nintendo	26.93	0.63	28.31	31.8640
11	11 Nintendogs	DS	2005	Simulation	Nintendo	9.07	11.00	24.76	25.3349
12	12 Mario Kart DS	DS	2005	Racing	Nintendo	9.81	7.57	23.42	21.5505
13	13 Pokemon Gold/Pokemon Silver	GB	1999	Role-Playing	Nintendo	9.00	6.18	23.10	18.7402
14	14 Wii Fit	Wii	2007	Sports	Nintendo	8.94	8.03	22.72	21.1714
15	15 Wii Fit Plus	Wii	2009	Sports	Nintendo	9.09	8.59	22.00	22.1008
16	16 Kinect Adventures!	X360	2010	Misc	Microsoft Game Studios	14.97	4.94	21.82	23.9316
17	17 Grand Theft Auto V	PS3	2013	Action	Take-Two Interactive	7.01	9.27	21.40	20.6272
18	18 Grand Theft Auto: San Andreas	PS2	2004	Action	Take-Two Interactive	9.43	0.40	20.81	11.4233
19	19 Super Mario World	SNES	1990	Platform	Nintendo	12.78	3.75	20.61	19.8042
20	20 Brain Age: Train Your Brain in Minutes a Day	DS	2005	Misc	Nintendo	4.75	9.26	20.22	18.0141
21	21 Pokemon Diamond/Pokemon Pearl	DS	2006	Role-Playing	Nintendo	6.42	4.52	18.36	13.5290
22	22 Super Mario Land	GB	1989	Platform	Nintendo	10.83	2.71	18.14	16.1556
23	23 Super Mario Bros. 3	NES	1988	Platform	Nintendo	9.54	3.44	17.28	15.6583
24	24 Grand Theft Auto V	X360	2013	Action	Take-Two Interactive	9.63	5.31	16.38	18.2891
25	25 Grand Theft Auto: Vice City	PS2	2002	Action	Take-Two Interactive	8.41	5.49	16.15	17.1290
26	26 Pokemon Ruby/Pokemon Sapphire	GBA	2002	Role-Playing	Nintendo	6.06	3.90	15.85	12.2770