LSE Course 3: Predictive Analytics

ASSIGNMENT 3 REPORT

Hazel Chan

July 3, 2022

Table of Contents

Background

Turtle Games is a game manufacturer and retailer that sells own products and products manufactured by other companies. They offer three product categories: Lego, various toys and games, and video games. As a global company, they have an objective of improving overall sales performance based on data analysis of price, customer sentiment and global sales forecast. This report will detail the methodology used to conduct analysis, insights, and predictions of sales.

Approach & Visualisation

Github link: https://github.com/hazz292/LSE_DA_Assignment_3_Turtle_Games

There are three aspects to analyse the dataset and improve sales performance. Firstly, simple and multiple linear regression functions in python are used to build a pricing model for lego products based on pieces and customer age. R tidyverse package is used to analyse the age group most likely to leave reviews and highest price point customers age 25 or above are willing to purchase. Secondly, R Natural Language Toolkit is used to conduct sentiment analysis and understand feedback from customers who purchased various toys and games. Thirdly, multiple linear regression in R is used to predict total global sales of video games based on Europe and North America sales.

A3_Week_1.ipynb

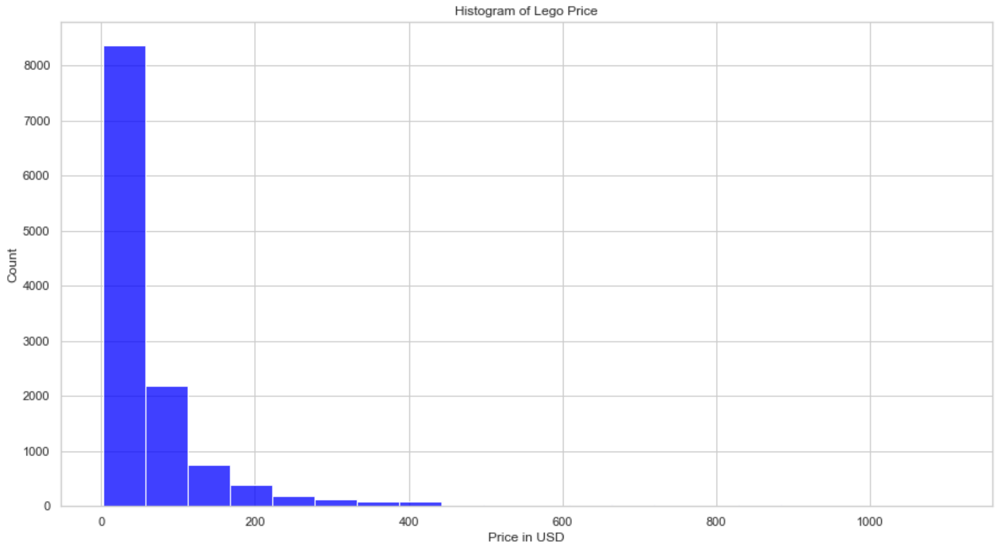
To predict the optimal price for lego products with 8000 pieces, pricing trend needs to be identified before creating simple and multiple linear regression models for prediction.

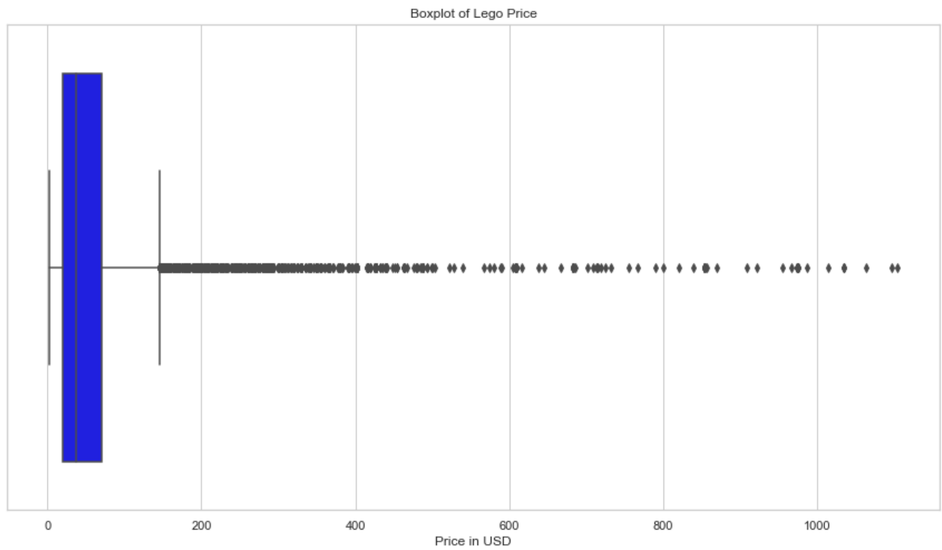
Using describe function in python, the price ranges from USD 2.27 to max USD1104 with a mean of USD65.

]:

	ages	list_price	num_reviews	piece_count	play_star_rating	review_difficulty	country
count	12261.00	12261.00	12261.00	12261.00	12261.00	12261.00	12261.00
mean	16.69	65.14	14.60	493.41	3.71	1.99	10.02
std	8.22	91.98	34.36	825.36	1.64	1.79	6.19
min	0.00	2.27	0.00	1.00	0.00	0.00	0.00
25%	11.00	19.99	1.00	97.00	3.60	0.00	4.00
50%	19.00	36.59	4.00	216.00	4.40	2.00	10.00
75%	23.00	70.19	11.00	544.00	4.70	4.00	15.00
max	30.00	1104.87	367.00	7541.00	5.00	5.00	20.00

Using seaborn package to create histogram and boxplot, price distribution is strongly skewed to the right with a long tail on the positive side. Majority of the lego is priced between USD 20 to 70, while outliers represent expensive products from USD 180 to USD 1100.





A subset with price as x and pieces as y is created to build simple linear regression model and split into train (70%) and test (30%) sets to validate accuracy of the model.

```
[24]: # Independent variable
      X = slr_data[['piece_count']]

      # Dependent variable
      y = slr_data['list_price']
```

```
[25]: # Create  the subset (70/30);
      # Control the shuffling/avoid variation in values between variable.

      X_train,X_test,y_train,y_test = train_test_split(X,y,train_size=0.7,
                                                        random_state=100)
```

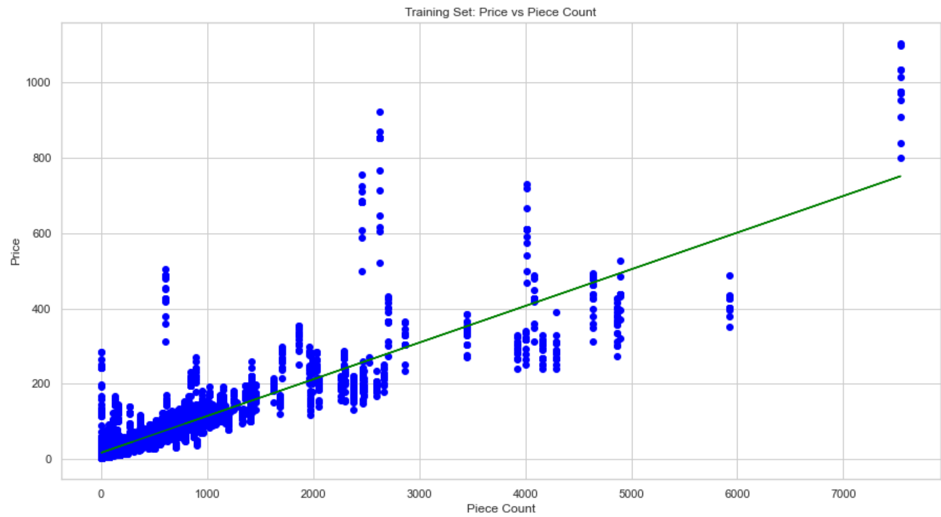
Train data set is fitted with linear regression.

```
[26]: # Fit linear regression model
      lm = LinearRegression()

      # Fit the model.
      lm.fit(X_train, y_train)

[26]: ▾ LinearRegression
      LinearRegression()
```

A scatter plot is created to visualize relationship the positive relationship between price and pieces of train set. A strong R-squared value signifies an increase in pieces explains 76% variation, increase, in price.



With the intercept and coefficient value, the predict function is used to predict price of 8000 pieces lego. According to the training model, the optimal price is USD797.

```
[30]: # Print R-squared value of the training data.
      print("R-Squared:", lm.score(X_train,y_train))

      R-Squared: 0.7644351150518354
```

Quite strong R-squared value and explains 76% of the dependent variable.

```
[31]: # Print the intercept value of training set
      print("Intercept value: ", lm.intercept_)
      # Print the coefficient value.
      print("Coefficient value: ", lm.coef_)

      Intercept value:  17.02159750575565
      Coefficient value:  [0.09744347]
```

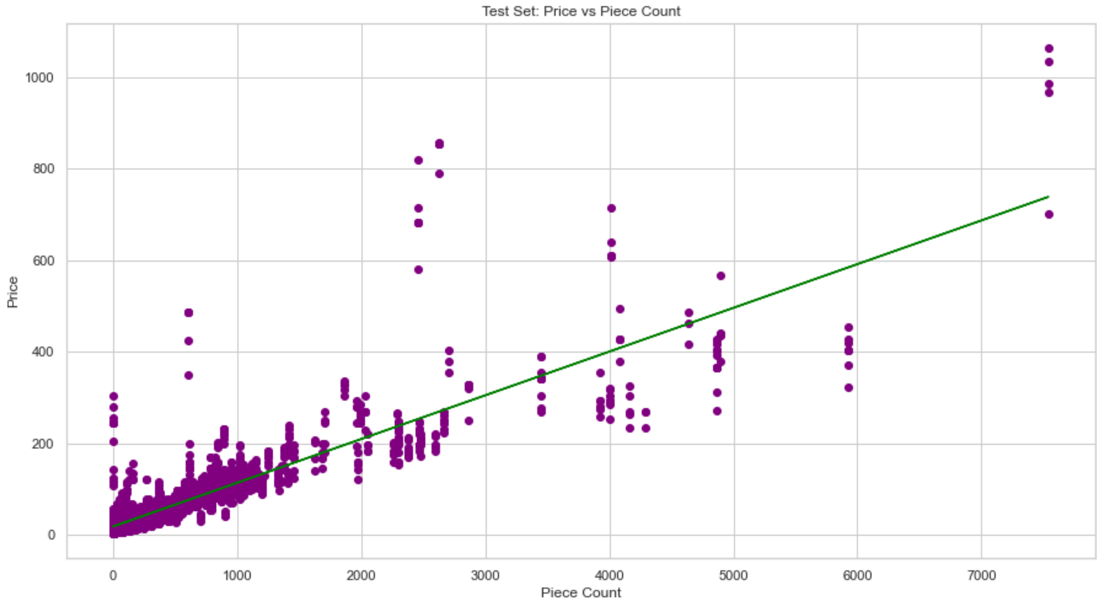
The coefficient value tells us that each additional lego piece in the product is associated with an increase of 0.097 USD in product price.

```
[32]: # Use the predict() method with an array to call the
      # Predict the price for a lego product with 8000 pieces.
      predictedPrice = lm.predict([[8000]])

      # Print the results.
      print(predictedPrice)

      [796.56932178]
```

To validate the model, the test set is used to predict the price, resulting in USD 783 which is similar to USD 797.



[38]:
```
# Print R-squared value of the test data.
print("R-squared:", lm.score(X_test,y_test))
```

```
R-squared: 0.736283806932537
```

As a rule of thumb, values greater than 0.60 are typically considered acceptable. Quite strong R-squared value and explains 74% of the dependent variable.

[39]:
```
# Print the intercept value.
print("Intercept value: ", lm.intercept_)
# Print the coefficient value.
print("Coefficient value: ", lm.coef_)
```

```
Intercept value:  18.042636129589262
Coefficient value:  [0.09561197]
```

Intercept and coefficient values are quite close to that obtained from training data set.

[40]:
```
# Predict the price for a lego product with 8000 pieces with test data set.
predictedPrice = lm.predict([[8000]])

# Print the results.
print(predictedPrice)
```

```
[782.93836492]
```

Subset is created to include three variables, price, pieces, and age. Multiple linear regression model is built based on train set with a strong R-square of 0.76 meaning pieces and age explains 76% of price variation. Predicted price for 8000 pieces of lego purchased by 30 years-old is USD783.

[46]:
```
# Fit linear regression model
mlr = LinearRegression()

# Fit the model
mlr.fit(X_train, y_train)
```

[46]:
```
▾ LinearRegression
LinearRegression()
```

[47]:
```
# Call predictions for x array
mlr.predict(X_train)
```

[47]:
```
array([105.31946916,  38.79803594,  25.21506289, ...,  28.52852519,
        26.36951271,  44.92267922])
```

[48]:
```
len(mlr.predict(X_train))
```

[48]:
```
8582
```

[49]:
```
# Checking the value of R-squared, intercept and coefficients
print("R-squared: ", mlr.score(X_train, y_train)) # Print the R-squared value
print("Intercept: ", mlr.intercept_) # Print the intercepts
print("Coefficients:") # Print coefficients
list(zip(X_train, mlr.coef_)) # Map similar index of multiple containers
```

```
R-squared:  0.7681985466459664
Intercept:  16.985596749203417
Coefficients:
```

[49]:
```
[('piece_count', 0.09569755116044504), ('ages', 0.029872780947O2156)]
```

[50]:
```
# Make predictions
# Set a variable as 8000 pieces
New_pieces = 8000

# Set a variable as 30 year old
New_age = 30

# Print the predicted price value
print ('Predicted Value: \n', mlr.predict([[New_pieces, New_age]]))
```

```
Predicted Value:
 [783.46218946]
```

Test set is fitted into the model predicting optimal price at USD 814 which is similar USD783.

[54]:
```
# Checking the value of R-squared, intercept and coefficients
print("R-squared: ", mlr.score(X_test, y_test)) # Print the R-squared value
print("Intercept: ", mlr.intercept_) # Print the intercepts
print("Coefficients:") # Print coefficients
list(zip(X_test, mlr.coef_)) # Map similar index of multiple containers
```

```
R-squared:  0.7344069243325273
Intercept:  16.516853988947737
Coefficients:
```

[54]:
```
[('piece_count', 0.09961851206458482), ('ages', 0.029226634168314652)]
```

R-squared of 0.73 signifies 73% of the variation in price (y dependent variable) can be explained by age of customer and pieces in lego products (x independent variables).

[55]:
```
# Make predictions
# Set a variable as 8000 pieces
New_pieces2 = 8000

# Set a variable as 30 year old
New_age2 = 30

# Print the predicted price value
print ('Predicted Value: \n', mlr.predict([[New_pieces2, New_age2]]))
```

```
Predicted Value:
 [814.34174953]
```

A3_Week_3.ipynb

Natural language processing is applied to analyse customer reviews of various toys and games and identify areas for improvement to satisfy customer and maximise sales.

Using python, game_reviews dataset is imported and viewed. Subset with only full reviewText is created, since summary is too short and cleaned by removing missing values, convert all text to lowercase, remove punctuation marks and duplicates.

[17]:
```python
# Reset index and preview data
reviews_clean.reset_index(inplace=True)

# View clean dataframe
reviews_clean.head()
```

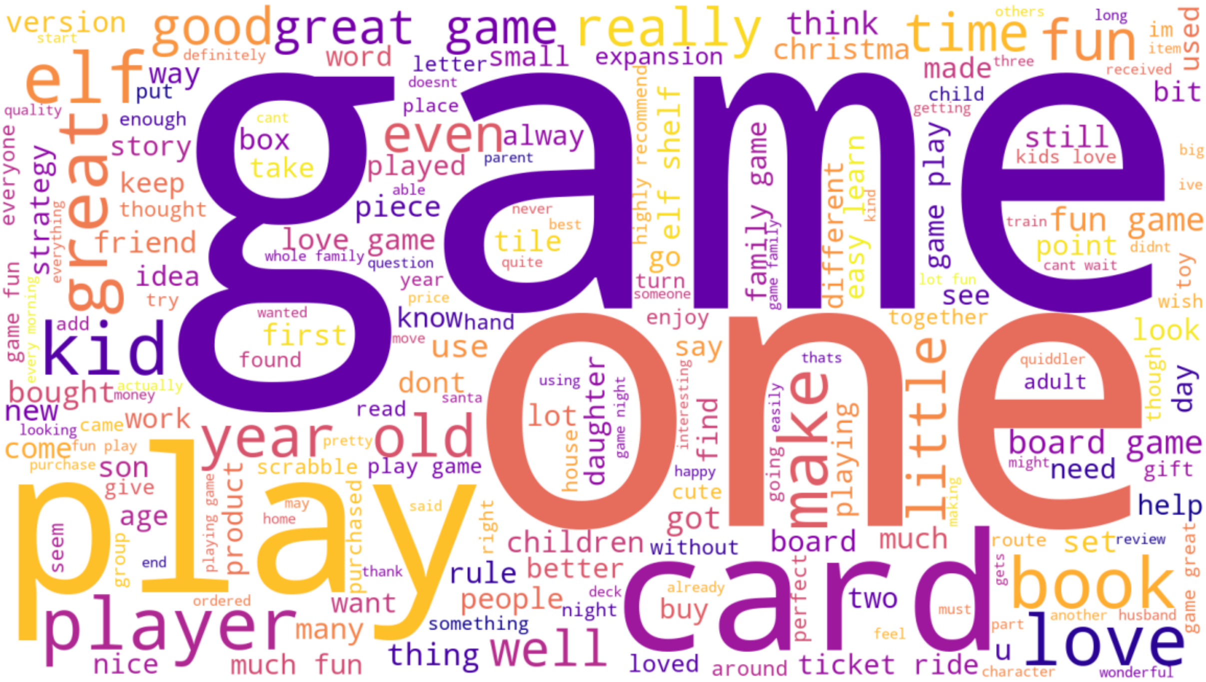
[17]:

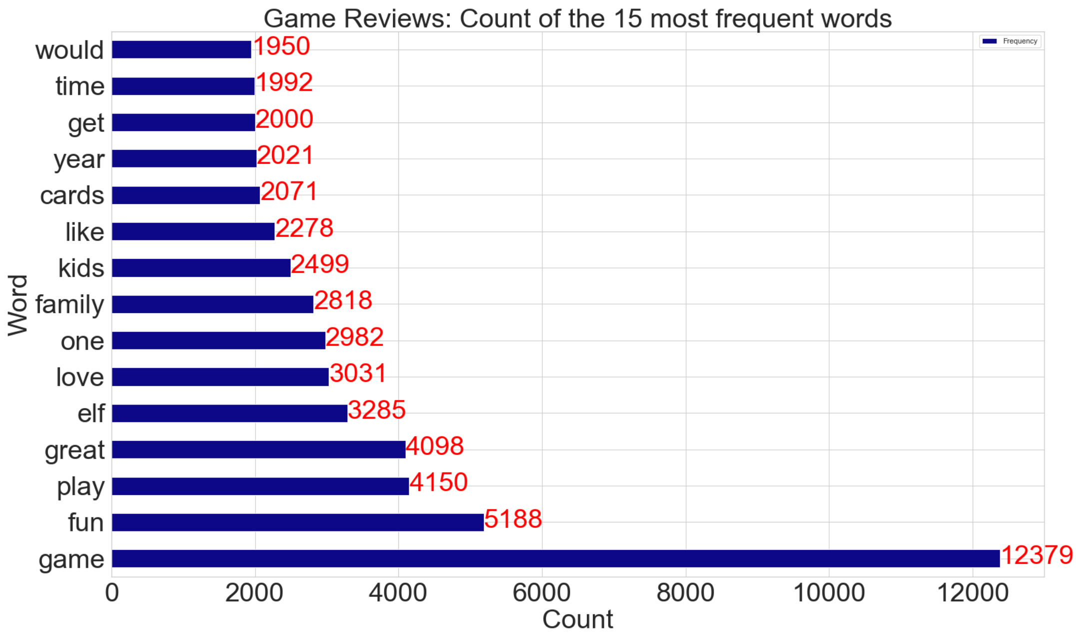
	index	reviewText
0	0	when it comes to a dms screen the space on the screen itself is at an absolute premium the fact that 50 of this space is wasted on art and not terribly informative or needed art as well makes it completely useless the only reason that i gave it 2 stars and not 1 was that technically speaking it can at least still stand up to block your notes and dice rolls other than that it drops the ball completely
1	1	an open letter to galeforce9 your unpainted miniatures are very not bad your spell cards are great your board games are meh your dm screens however are freaking terrible im still waiting for a single screen that isnt polluted with pointless artwork where useful referenceable tables should be once again youve created a single use screen that is only useful when running the storm kings thunder adventure even despite the fact that its geared to that adventure path its usefulness negligible at best i massive swath of the inner panel is wasted on artwork and a bloated overland map which could have been easily reduced to a single panel in size and the few table you have are nighuseless themselves in short stop making crap dm screens
2	2	nice art nice printing why two panels are filled with a general forgotten realms map is beyond me most of one of them is all blue ocean such a waste i dont understand why they cant make these dm screens more useful for these kinds of adventures rather than solely the specific adventure youre supposed to be able to transpose this adventure to other lands outside the forgotten realms so even just a list of new monsters or npcs would at least be useful than the map even more would just be stuff related to running the game but broaduse stuff related to giants same thing with curse of strahd why not make it useful for raven loft undead or horror campaigns in general instead a huge amount of screen space is solely mapping out castle ravenloft which is only useful during a small fraction of the time even for the curse of strahd adventure let alone various other ravenloft adventuring they really kill the extended use of these screens by not thinking about their potential use both for the adventure in question as well as use in a broader sense the rage of demons screen is far more useful for broad under dark adventuring covering a lot of rules for the various conditions you may suffer and the map is only one panel this storm giants one is decent for a few tables it includes but really misses the mark maybe they should ask a few dms what they would use
3	3	amazing buy bought it as a gift for our new dm and its perfect
4	4	as my review of gf9s previous screens these were completely unnecessary and nearly useless skip them this is the definition of a waste of money

[18]:
```python
# View shape of clean dataframe
reviews_clean.shape
```

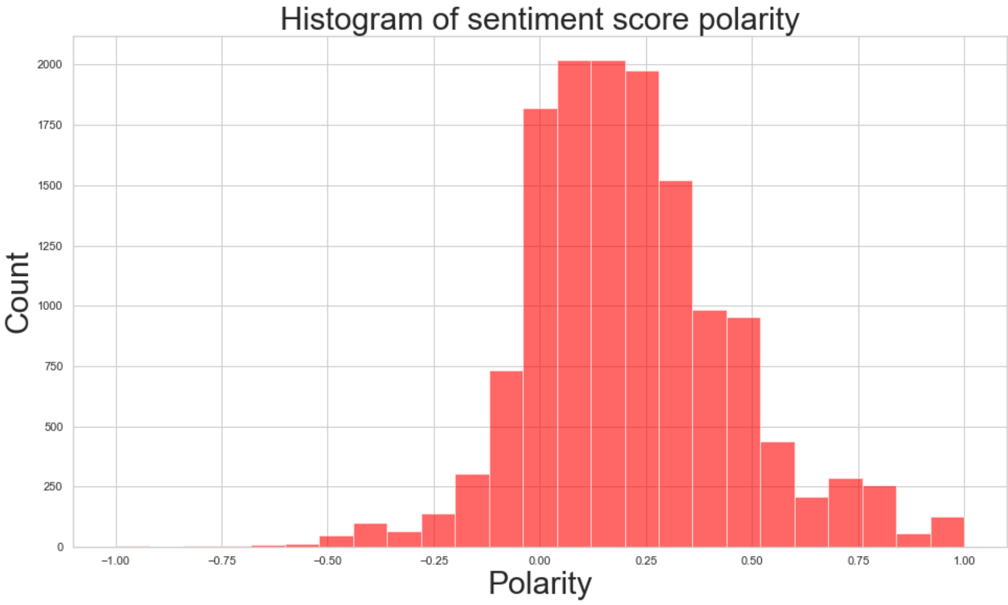
[18]: (14081, 2)

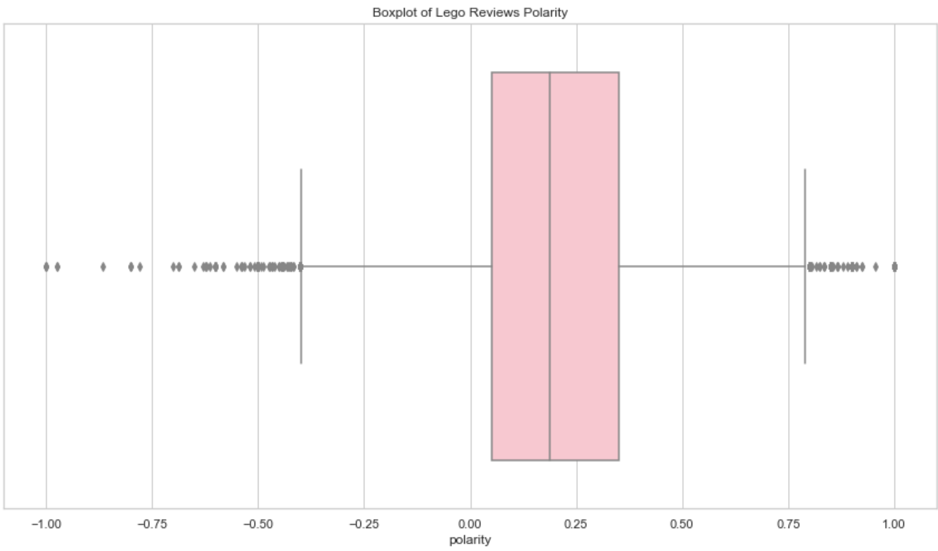
Next, text is pre-processed using tokenisation to calculate the frequency distribution of words. Each sentence is split into individual tokens, added into one list using for loop and plotted. As there are many stopwords such as “and”, “the” affecting results, they are removed and clean tokens are visualised with WordCloud. Words with higher frequency are larger such as “game”, “play”, “great”.





Polarity score is generated using textblob library and plotted onto histogram and boxplot. Overall, majority of reviews are slightly positive with polarity score mainly between 0 to 0.25.

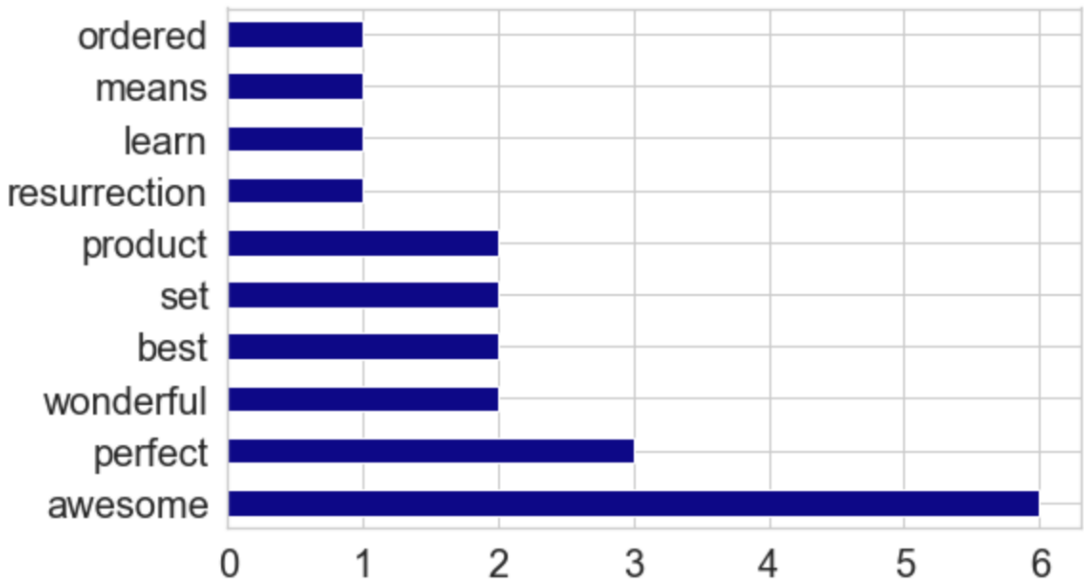




To understand reason behind positive sentiment, the top 20 positive reviews are extracted. Using document-term matrix to extract the positive features, most top reviewers thought games are "awesome", "perfect" and "wonderful".

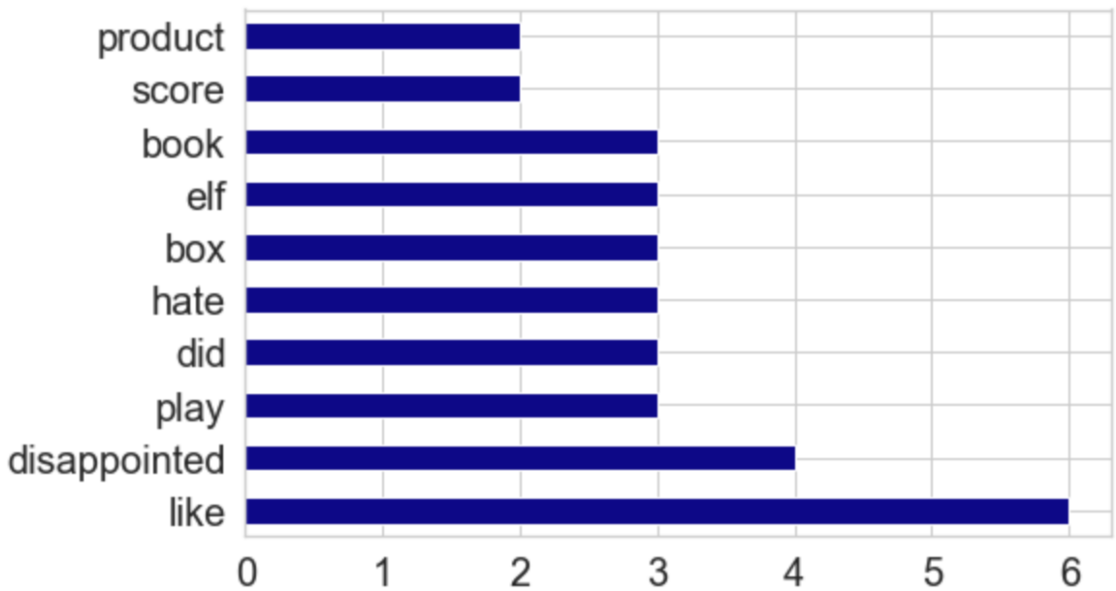
[42]:

	reviewText	polarity	subjectivity
7	came in perfect condition	1.000000	1.000000
164	awesome book	1.000000	1.000000
193	awesome gift	1.000000	1.000000
489	excellent activity for teaching selfmanagement skills	1.000000	1.000000
517	perfect just what i ordered	1.000000	1.000000
583	wonderful product	1.000000	1.000000
601	delightful product	1.000000	1.000000
613	wonderful for my grandson to learn the resurrection story	1.000000	1.000000
782	perfect	1.000000	1.000000
922	awesome	1.000000	1.000000
1118	awesome set	1.000000	1.000000
1149	best set buy 2 if you have the means	1.000000	0.300000
1158	awesome addition to my rpg gm system	1.000000	1.000000
1279	its awesome	1.000000	1.000000
1376	one of the best board games i played in along time	1.000000	0.300000
1516	my daughter loves her stickers awesome seller thank you	1.000000	1.000000
1573	this was perfect to go with the 7 bean bags i just wish they were not separate orders	1.000000	1.000000
1677	awesome toy	1.000000	1.000000
1682	it is the best thing to play with and also mind blowing in some ways	1.000000	0.300000
1688	excellent toy to simulate thought	1.000000	1.000000



Top 20 negative reviews are extracted based on negative polarity score and most reviewers were “disappointed” and mentioned “box”, “book”, “elf”.

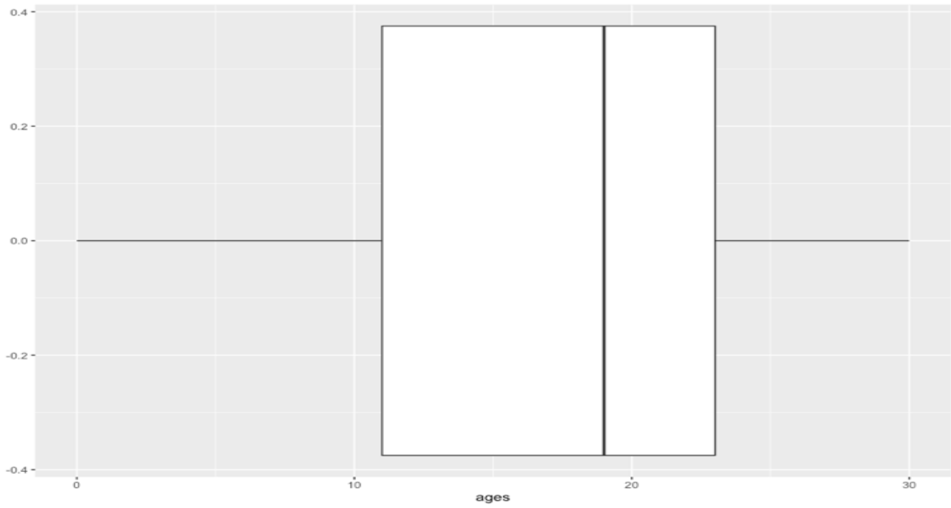
	reviewText	polarity	subjectivity
207	booo unles you are patient know how to measure i didnt have the patience neither did my daughter boring unless you are a craft person which i am not	-1.000000	1.000000
1987	kids did not like it thought it was boring	-1.000000	1.000000
3218	some of the suggestions are disgusting	-1.000000	1.000000
7812	awful we did not receive what was advertised we paid 30 for the boxes set with book we got the elf in a bag without the book	-1.000000	1.000000
7515	was the elf on the shelf but it didnt have the dvd i was very disappointed	-0.975000	0.975000
8861	i havent even taken it out of the box yet but its already falling apart i contacted customer service and never even got a response i am very disappointed in this product	-0.975000	0.975000
8198	i hate the holidays bcuz of the elf he was disgusting i hate him with my life he doesnot leave the shelf alone	-0.866667	0.933333
12386	i do not under stand how you keep score or read the scoring i i do not like that at all i can never play score with anyone at all i hate that i cant play points	-0.800000	0.900000
8531	cliche and stupid i should not drink and amazon	-0.800000	1.000000
8638	just stupid	-0.800000	1.000000
181	incomplete kit very disappointing	-0.780000	0.910000
13413	i like this product for my daughter she is into the bad kitty book collection so it was an added bonus	-0.700000	0.666667
4060	ordered for my sons birthday opened it up today to play and the board is damaged before we even take it out of the box the game is already falling apart very disappointed	-0.687500	0.687500
4090	id like to upload a photo of the condition of the game boxit looks like its been used as a soccer ball 2 corners of the box are smashed in and on is even ripped how am i supposed to give this as a gift without it looking like i bought this on clearance very disappointed	-0.687500	0.687500
11263	horrible and incomplete flash cardsdo not buy not helpful i was too late to return them	-0.650000	0.800000
2082	this was a bit disappointing my students find it boring and the letters are hard to understand	-0.630556	0.747222
10768	boring did i mention boring well its boring pass on this one there are a lot better games out there	-0.625000	0.875000
13122	had no idea the extent you have to go through to put this together hundreds and i mean hundreds of pieces that dont snap together it will take my teen age son and i months to put this stupid thing together horrible plan horrible	-0.622500	0.737500
7744	i received a small paperback bookfor 3000 the picture shows an elf hardcover book and box that it all comes in very disappointed for the student we bought this for	-0.612500	0.687500
4691	want to hate your friends and family get this game	-0.600000	0.650000



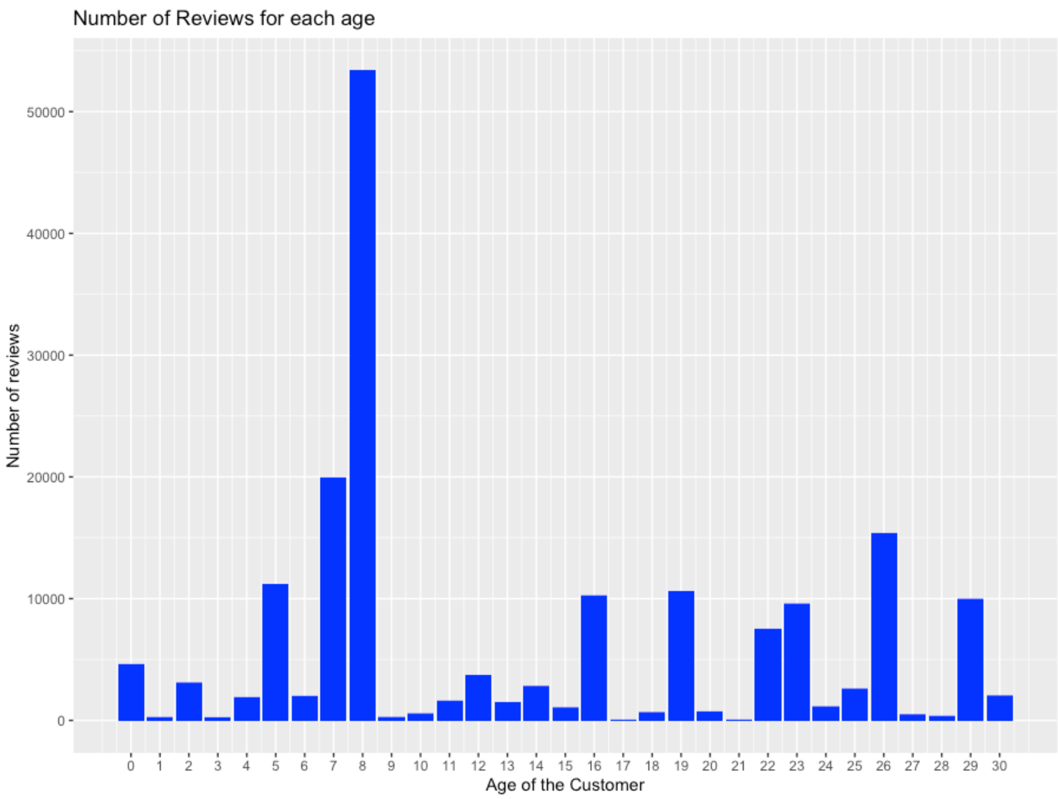
A3_Week_4.R

R Tidyverse package is used to wrangle and visualise lego data set to understand which age group is most likely to leave reviews and which most expensive price point is purchased by customer above age 25.

The lego data set is imported, viewed using as_tibble function, and check for missing values. Then, qplot function is used to view distribution of age variable using boxplot. Majority of customers are aged between 10 to 25.



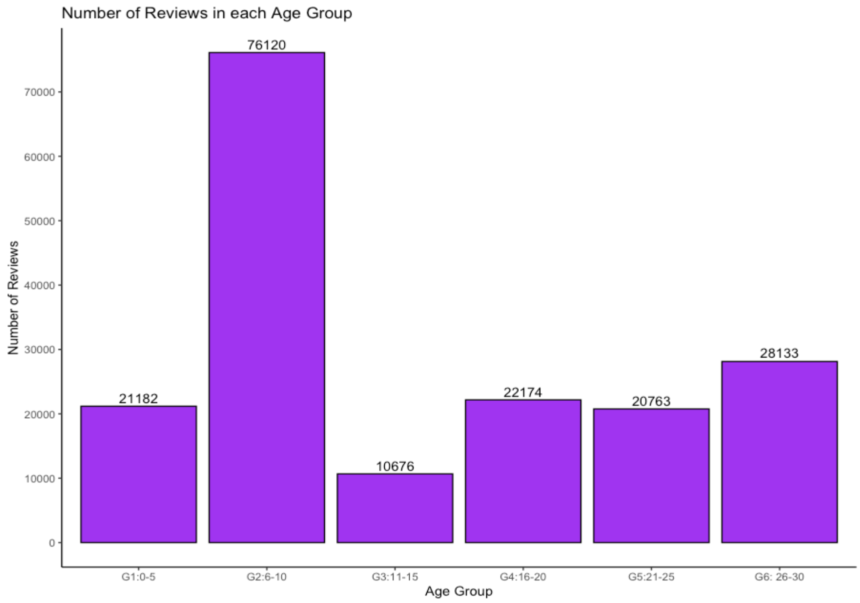
Column graph identified that age 8 customer left the most reviews.



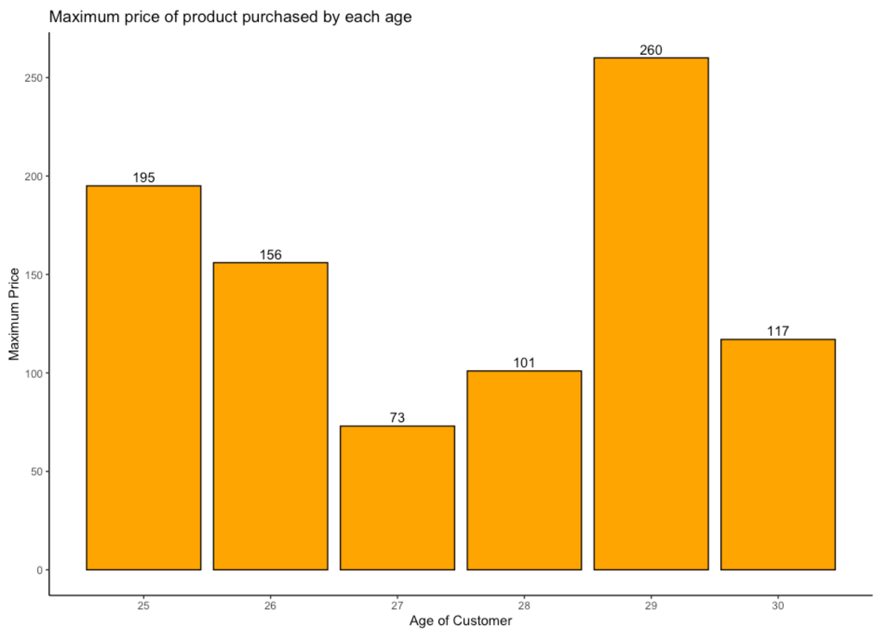
A new column is added to age_df to map the corresponding customer age into 6 groups and aggregate by age_group.

```
# Group customers into corresponding age group
age_df <- age_df %>%
    mutate(age_group = case_when(ages <= 5 ~ "G1:0-5",
                                 ages >= 6 & ages <= 10 ~ "G2:6-10",
                                 ages >= 11 & ages <= 15 ~ "G3:11-15",
                                 ages >= 16 & ages <= 20 ~ "G4:16-20",
                                 ages >= 21 & ages <= 25 ~ "G5:21-25",
                                 ages >= 26 & ages <= 30 ~ "G6: 26-30"))
```

Column graph identifies customers age 6-10 are most likely to leave reviews, then followed by age 26-30.



Subset with customer older than age 25 is created to identify highest price point accepted. Dataframe is aggregated by age and summarised by maximum price. Column graph indicates age 29 customers purchased the most expensive lego product at USD260, then age 25 customers at USD195.



A3_Week_5.R

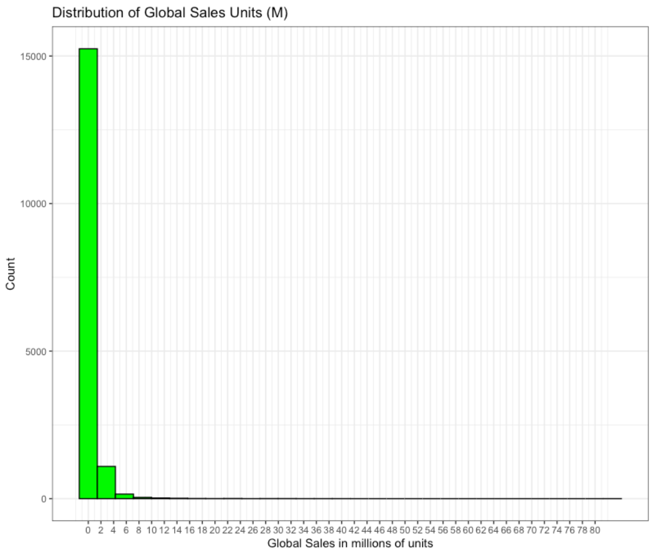
Game_sales subset is created with relevant variables = and cleaned by identifying missing values.

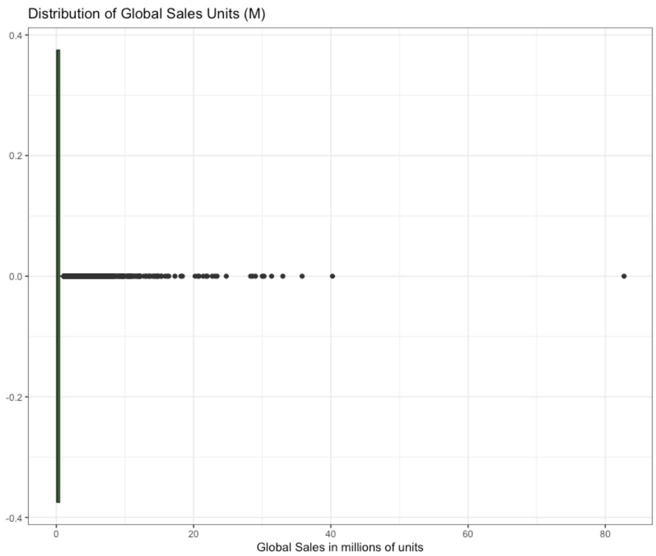
Global sales, North America and Europe sales distribution are visualised using histogram and boxplot of ggplot. All sales distribution is extremely skewed to the right due to extreme outliers as identified by upperbound calculated using interquartile range. Outliers are included as they represent the best-selling products.

Majority of global sales is between 0.06M to 1M units with outlier of 82M, high skewness of 17.34 and heavy kurtosis of 606.75.

```
> # Summary Statistics
> summary(sales$Global_Sales)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0100  0.0600  0.1700  0.5374  0.4700 82.7400 
> 
```

```
> # Skewness
> skewness(sales$Global_Sales)
[1] 17.39907
> # Kurtosis
> kurtosis(sales$Global_Sales)
[1] 606.7501
> 
```

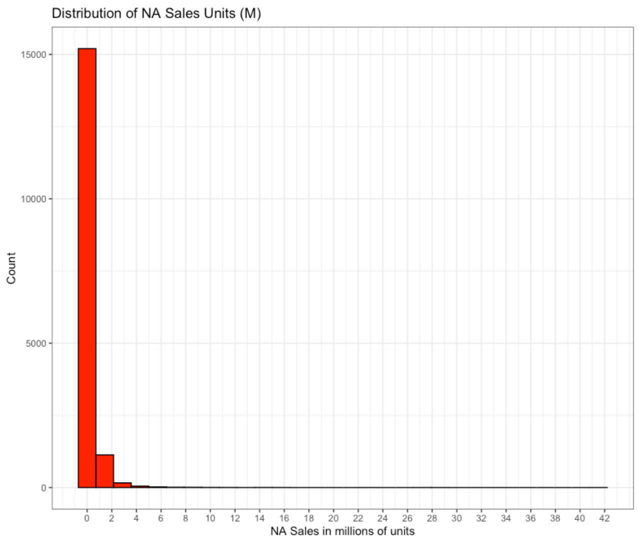


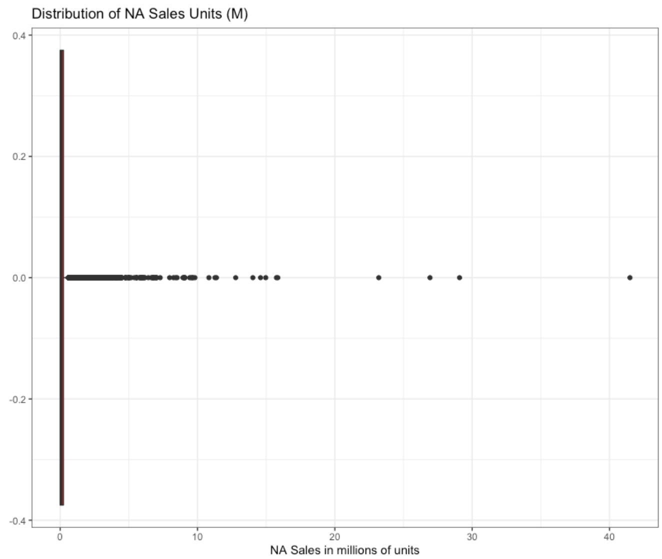


Majority of NA sales is between 0.01M to 0.6M units with outlier of 82M, high skewness of 18.79 and heavy kurtosis of 651.

```
> # Summary Statistics
> summary(sales$NA_Sales)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.0000  0.0800  0.2647  0.2400 41.4900 
> 
```

```
> # Skewness
> skewness(sales$NA_Sales)
[1] 18.79793
> # Kurtosis
> kurtosis(sales$NA_Sales)
[1] 651.9344
```

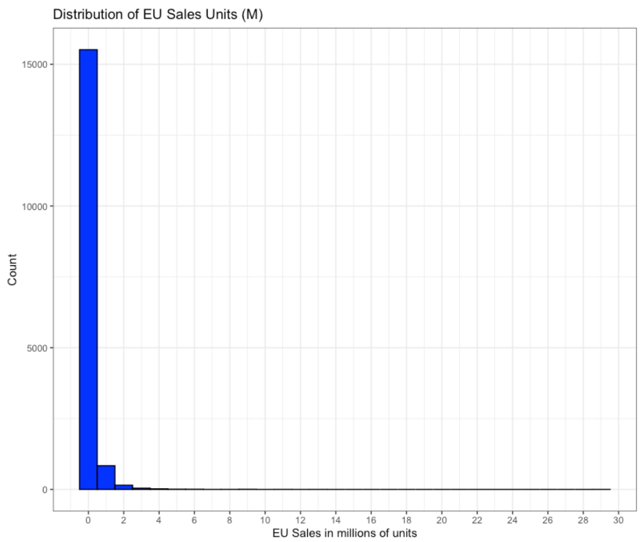


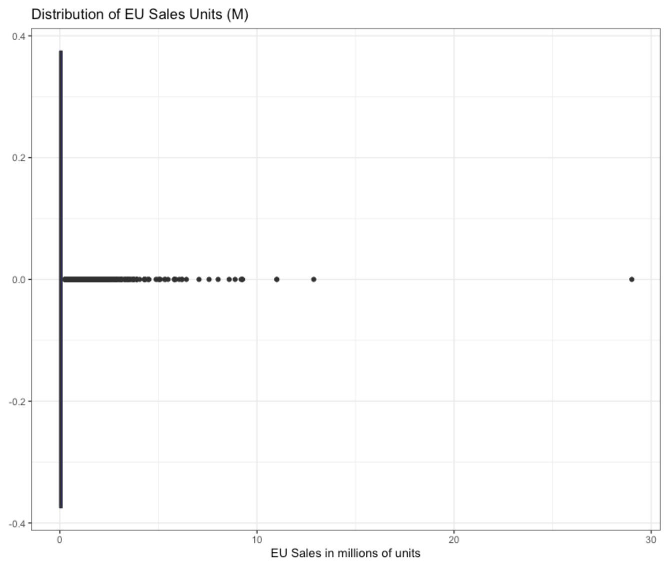


Majority of EU sales is between 0.01M to 0.27M units with outlier of 29M, high skewness of 18.87 and heavy kurtosis of 758.80.

```
> # Summary Statistics
> summary(sales$EU_Sales)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.0000  0.0200  0.1467  0.1100 29.0200 
> 
```

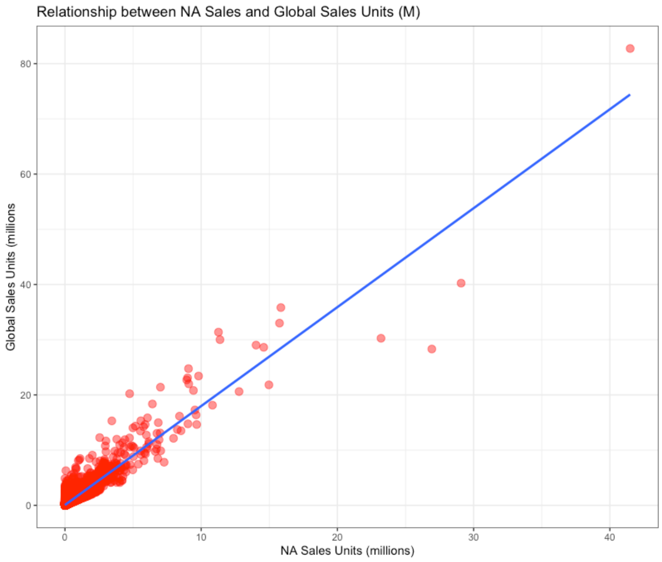
```
> # Skewness
> skewness(sales$EU_Sales)
[1] 18.87383
> # Kurtosis
> kurtosis(sales$EU_Sales)
[1] 758.7997
```

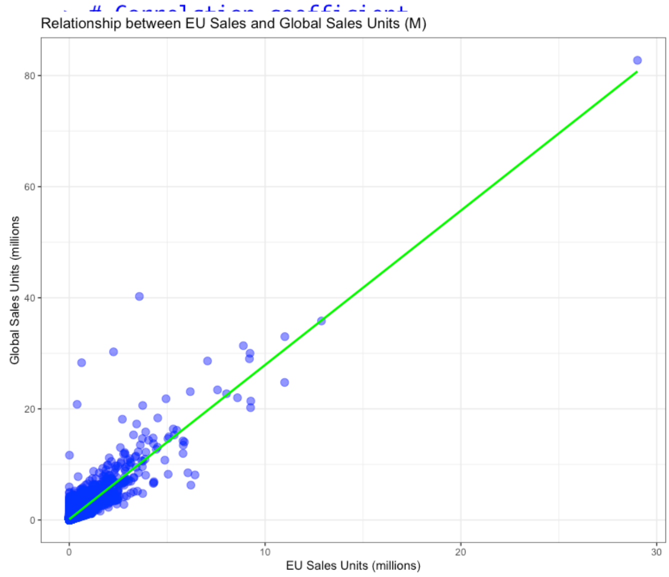




Scatterplot shows a strong positive relationship between North America, Europe and Global sales with a correlation coefficient of 0.9 which is very close to 1.

```
> # Correlation coefficient
> cor (sales$NA_Sales, sales$Global_Sales)
[1] 0.9410474
>
```





A3_Week_6.R

Multiple linear regression model is built using lm function to predict global sales, dependent variable, based on North America and Europe sales, independent variable, to minimise production waste.

```
##############################################################################################################
# Build model
##############################################################################################################

# R syntax for MLR:
  # myModel <- lm(y ~ x1 + x2 + x3, data=mydata)
  # y dependent variable = Global Sales
  # x independent variable = EU_Sales, Global_Sales

# View correlation between EU/NA sales and Global sales
cor(sales)
  # Both EU and NA sales are highly correlated with global sales
  # with 0.9 correlation coefficient, very close to 1.

# Create a new regression model with lm function
model1 = lm(Global_Sales ~ NA_Sales + EU_Sales, data=sales)

# Print the summary statistics.
summary(model1)
  # In this model, the Multiple R-squared is very strong at 0.96, very close to 1.
  # This means that North America and Europe sales explains 96% of the variability of the Global Sales variable.
  # The three stars next to the variables indicates that they have high significance in the model.

```

Multiple R-squared 0.96 is very strong and close to 1, meaning North America and Europe sales explains 96% of Global sales variation. Residual standard error is quite small meaning regression model fit dataset closely.

```
> # Print the summary statistics.
> summary(model1)

Call:
lm(formula = Global_Sales ~ NA_Sales + EU_Sales, data = sales)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.2223 -0.0634 -0.0415 -0.0049  9.3929 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.034924   0.002385   14.64   <2e-16 ***
NA_Sales    1.149675   0.004328  265.67   <2e-16 ***
EU_Sales    1.351735   0.006994  193.28   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2918 on 16595 degrees of freedom
Multiple R-squared:  0.9648,	Adjusted R-squared:  0.9648 
F-statistic: 2.274e+05 on 2 and 16595 DF,  p-value: < 2.2e-16
```

Total dataframe is created by aggregating sum of global, NA and EU sales of all games. Predict function generates outcome 8340M which is close to actual value 8920M.

```

# Create new dataframe with sum of all EU, NA and Global sales
total <- sales %>%
  summarise(across(where(is.numeric), ~ sum(.x, na.rm = TRUE)))

# Print Total Sales dataframe
total

# Make prediction based on new dataframe
predictTotal = round(predict(model1, newdata = total, interval = 'confidence'),2)

# Print prediction of global sales for next year
predictTotal

# Print the object.
View(predictTotal)

# Convert predicted values to dataframe
total_df <- data.frame(predictTotal)

# Create final dataframe with genre name and predicted global sales
final_total <- cbind(total[("Global_Sales")], total_df)

# Rename column name
final_total <- final_total %>% rename(Predicted_Global_Sales = fit,
                                      Actual_Global_Sales = Global_Sales)

# Print final dataframe
final_total
```

Actual_Global_Sales	Predicted_Global_Sales	lwr	upr
8920.44	8340.8	8316.45	8365.14

Predictions

As predicted using linear regression model, lego price should increase as pieces and age increase assuming positive linear relationship. Business should price lego with 8000 pieces at USD783 and if customer is age 30, price at USD 814. Identifying optimal price points will enable business to align with customers' perceived value of products, increase likelihood of purchase and maximise sales.

Business should improve product features based on customer reviews to satisfy demands with attractive product offerings. Overall review sentiment is slightly positive with polarity score between 0 to 0.25. Based on top negative review analysis, business should improve quality of box packaging and books materials to improve sales performance.

Customers age between 6-10 left most reviews of 76,120. Further analysis can be conducted to understand if they are frequent customers. Age 29 customers purchased most expensive lego product at USD260, then age 25 customers at USD195 signifying higher purchasing power. Business should target expensive product range for older customers.

Lastly, based on EU and NA historic sales, global sales of all video games are predicted to reach 8340M units in next financial year. Business should take forecasted sales into consideration when sourcing and manufacturing units to ensure enough inventory to capture potential sales.