PS4 vs Xbox One Sales Analysis for Marketing Strategy

Data Wrangling Project

View on website: https://rpubs.com/cthemin/676984

Required packages

```
# R Packages used for this report
library(readr)
library(dplyr)
library(tidyr)
library(outliers)
library(ggplot2)
library(lattice)
```

Executive Summary

The purpose of this report is to apply Data Wrangling method using the real-world data. The process includes data preprocessing steps as below:
1. Collect 2 datasets from an open source data platform such as Kaggle and read in in R Studio using appropriate functions and library packages.
2. Merge/combine two datasets into one and select useful variables to work with.
3. Provide description of each variable used on the data.
4. Understand the data such as the structure, the data type, discover the unique values in factor variable, label the name and put in order where necessary, check the class and the levels.
5. Perform Tidy and Manipulate technique such as mutate a new variable with calculation, reshape the data from untidy to tidy format for better analysis.
6. Scan the data to find any missing values, errors, or special values, then find a suitable methodology to deal with it.
7. Scan for any outliers using summary statistics, apply appropriate plots on the data, explain the methodology and transform it for better insight.

Provide conclusion in overall.

Data

The first dataset is XboxOne Games Sales, collected from Kaggle Open Data which can be downloaded from the link: https://www.kaggle.com/sidtwr/videogames-sales-dataset?select=XboxOne_GameSales.csv (https://www.kaggle.com/sidtwr/videogames-sales-dataset?select=XboxOne_GameSales.csv).

The original dataset has 613 observations and 10 variables.

And the second dataset is Playstation 4 Games Sales, collected also from Kaggle Open Data which can be downloaded from the link: https://www.kaggle.com/sidtwr/videogames-sales-dataset?select=PS4_GamesSales.csv (https://www.kaggle.com/sidtwr/videogames-sales-dataset?select=PS4_GamesSales.csv).

The original dataset has 1034 observations and 9 variables.

Both datasets have 9 main variables that can be used for further analysis and are described as follow:

1. Game: name/title of the game
2. Year: year when the game was published
3. Genre: genre of the game
4. Publisher: name of the company that publish the game
5. North America: sales in North America (in Million)
6. Europe: sales in Europe (in Million)
7. Japan: sales in Japan (in Million)
8. Rest of World: sales in the Rest of the world (in Million)
9. Global: sales made globally

The two datasets are merged into one using the full_join() function by Game, Year, Publisher, and Genre. We are interested to analyse the total games sales.

After joined the datasets, we ended up with only 12 variables where: 4 of them are Game, Year, Genre, and Publisher. 4 variables are the XboxOne Sales in each continent and, 4 variables are the Playstation 4 sales in each continent.

```
##### DATASET 1: XBOX ONE #####
# Import and read the dataset
xbox <- read_csv("XboxOne_GameSales.csv")
```

```
## Parsed with column specification:
## cols(
##   Pos = col_double(),
##   Game = col_character(),
##   Year = col_character(),
##   Genre = col_character(),
##   Publisher = col_character(),
##   `North America` = col_double(),
##   Europe = col_double(),
##   Japan = col_double(),
##   `Rest of World` = col_double(),
##   Global = col_double()
## )
```

```
# Select the columns for analysis
xbox <- xbox %>% select("Year", "Game", "Publisher", "Genre", "North America", "Europe", "Japan", "Rest of World")

# Check the first 5 rows
head(xbox)
```

Y…	Game	Publisher	Genre	North America	Euro…	Jap…
<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
2014	Grand Theft Auto V	Rockstar Games	Action	4.70	3.25	0.01
2015	Call of Duty: Black Ops 3	Activision	Shooter	4.63	2.04	0.02
2017	Call of Duty: WWII	Activision	Shooter	3.75	1.91	0.00
2018	Red Dead Redemption 2	Rockstar Games	Action-Adventure	3.76	1.47	0.00
2014	MineCraft	Microsoft Studios	Misc	3.23	1.71	0.00
2014	Call of Duty: Advanced Warfare	Activision	Shooter	3.25	1.49	0.01

6 rows | 1-7 of 8 columns

```
##### DATASET 2: PLAYSTATION 4 #####
# Import and read the dataset
ps4 <- read_csv("PS4_GamesSales.csv")
```

```
## Parsed with column specification:
## cols(
##   Game = col_character(),
##   Year = col_character(),
##   Genre = col_character(),
##   Publisher = col_character(),
##   `North America` = col_double(),
##   Europe = col_double(),
##   Japan = col_double(),
##   `Rest of World` = col_double(),
##   Global = col_double()
## )
```

```
# Select the columns for analysis
ps4 <- ps4 %>% select("Year", "Game", "Publisher", "Genre", "North America", "Europe", "Japan", "Rest of World")

# Check the first 5 rows
head(ps4)
```

Year <chr>	Game <chr>	Publisher <chr>	Genre <chr>	North America <dbl>	Europe <dbl>	Japan <dbl>
2014	Grand Theft Auto V	Rockstar Games	Action	6.06	9.71	0.60
2015	Call of Duty: Black Ops 3	Activision	Shooter	6.18	6.05	0.41
2018	Red Dead Redemption 2	Rockstar Games	Action-Adventure	5.26	6.21	0.21
2017	Call of Duty: WWII	Activision	Shooter	4.67	6.21	0.40
2017	FIFA 18	EA Sports	Sports	1.27	8.64	0.15
2016	FIFA 17	Electronic Arts	Sports	1.26	7.95	0.12

6 rows | 1-7 of 8 columns

```
# Merging the two datasets using full_join by Game, Year, Publisher, Genre
joinxp <- full_join(xbox, ps4, by = c("Game", "Year", "Publisher", "Genre"))

# Check the head of the dataset
head(joinxp)
```

Y… <chr>	Game <chr>	Publisher <chr>	Genre <chr>	North America.x <dbl>	Europe.x <dbl>
2014	Grand Theft Auto V	Rockstar Games	Action	4.70	3.25
2015	Call of Duty: Black Ops 3	Activision	Shooter	4.63	2.04
2017	Call of Duty: WWII	Activision	Shooter	3.75	1.91
2018	Red Dead Redemption 2	Rockstar Games	Action-Adventure	3.76	1.47
2014	MineCraft	Microsoft Studios	Misc	3.23	1.71
2014	Call of Duty: Advanced Warfare	Activision	Shooter	3.25	1.49

6 rows | 1-6 of 12 columns

Understand

First, we check the data structure of the joined table using str() function. There are combination of numeric and characters datatypes. Some of them need to be converted such as: 1. Year: convert from Char to Factor, relabelled and ordered as True 2. Genre: convert from Char to Factor, the order is not important

During data conversion, the value is checked using unique() function to find out what are the labels.

After the conversion, we check the class() and the levels() again to ensure the datatype are converted correctly and ordered.

```
# Check data structure
str(joinxp)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 1177 obs. of  12 variables:
##  $ Year           : chr  "2014" "2015" "2017" "2018" ...
##  $ Game           : chr  "Grand Theft Auto V" "Call of Duty: Black Ops 3" "Call of Duty: WWII" "Red Dead Redemption 2"
...
##  $ Publisher      : chr  "Rockstar Games" "Activision" "Activision" "Rockstar Games" ...
##  $ Genre          : chr  "Action" "Shooter" "Shooter" "Action-Adventure" ...
##  $ North America.x: num  4.7 4.63 3.75 3.76 3.23 3.25 3.37 2.94 2.94 2.91 ...
##  $ Europe.x       : num  3.25 2.04 1.91 1.47 1.71 1.49 1.26 1.62 1.49 1.44 ...
##  $ Japan.x        : num  0.01 0.02 0 0 0 0.01 0.02 0.02 0.03 0 ...
##  $ Rest of World.x: num  0.76 0.68 0.57 0.54 0.49 0.48 0.48 0.45 0.45 0.44 ...
##  $ North America.y: num  6.06 6.18 4.67 5.26 NA 2.84 2.2 2.91 NA 3.11 ...
##  $ Europe.y       : num  9.71 6.05 6.21 6.21 NA 3.34 3.65 3.97 NA 3.83 ...
##  $ Japan.y        : num  0.6 0.41 0.4 0.21 NA 0.14 0.29 0.27 NA 0.19 ...
##  $ Rest of World.y: num  3.02 2.44 2.12 2.26 NA 1.22 1.12 1.34 NA 1.36 ...
```

```
##### Data conversion Steps #####
# 1. Check for any unique value in the Year and Genre
unique(xbox$Year)
```

```
## [1] "2014" "2015" "2017" "2018" "2016" "2013" "N/A"  "2019" "2020"
```

```
unique(xbox$Genre)
```

```
##  [1] "Action"           "Shooter"          "Action-Adventure" "Misc"
##  [5] "Role-Playing"     "Racing"           "Sports"           "Fighting"
##  [9] "Adventure"        "MMO"              "Music"            "Simulation"
## [13] "Strategy"         "Platform"         "Puzzle"           "Visual Novel"
```

```
# 2a. Convert Year datatype to factor and put it in order
joinxp$Year <- factor(joinxp$Year, levels = c("2013", "2014", "2015", "2016", "2017", "2018", "2019", "2020"), ordered = T)

# 2b. Convert Genre datatype to factor
joinxp$Genre <- factor(joinxp$Genre, levels = c("Action", "Shooter", "Action-Adventure", "Misc", "Role-Playing", "Racing",
"Sports", "Fighting", "Adventure", "MMO", "Music", "Simulation", "Strategy", "Platform", "Puzzle", "Visual Novel"))

# 3. Check the class and levels
class(joinxp$Year)
```

```
## [1] "ordered" "factor"
```

```
levels(joinxp$Year)
```

```
## [1] "2013" "2014" "2015" "2016" "2017" "2018" "2019" "2020"
```

```
class(joinxp$Genre)
```

```
## [1] "factor"
```

```
levels(joinxp$Genre)
```

```
##  [1] "Action"           "Shooter"          "Action-Adventure" "Misc"
##  [5] "Role-Playing"     "Racing"           "Sports"           "Fighting"
##  [9] "Adventure"        "MMO"              "Music"            "Simulation"
## [13] "Strategy"         "Platform"         "Puzzle"           "Visual Novel"
```

Tidy & Manipulate Data II (Create/Mutate)

The joined table ended up with 2 similar continent sales (1 for XboxOne and the other for Playstation 4). We must combine/add the 2 sales into one continent using mutate() function and call it as total of the continent sales (eg. North America total sales, Europe total sales, Japan total sales, and Rest of World total sales).

After having the total sales of each continent, we will drop the individual continent sales and focus only on the total sales.

```
# Mutate new variables by adding the two same continents
joinxp <- joinxp %>% mutate(North_America_Total_Sales = `North America.x` + `North America.y`,
                            Europe_Total_Sales = Europe.x + Europe.y,
                            Japan_Total_Sales = Japan.x + Japan.y,
                            Rest_Of_World_Total_Sales = `Rest of World.x` + `Rest of World.y`)

# Select only the continents that are already combined
joinxp_filter <- joinxp %>% select(-c("North America.x", "North America.y",
                                      "Europe.x", "Europe.y", "Japan.x", "Japan.y",
                                      "Rest of World.x", "Rest of World.y"))

# Check the head of the new filtered dataset
joinxp_filter %>% head(3)
```

Year	Game	Publisher	Genre	North_America_Total_Sales
<ord>	<chr>	<chr>	<fctr>	<dbl>
2014	Grand Theft Auto V	Rockstar Games	Action	10.76
2015	Call of Duty: Black Ops 3	Activision	Shooter	10.81
2017	Call of Duty: WWII	Activision	Shooter	8.42

3 rows | 1-5 of 8 columns

Tidy & Manipulate Data I (Reshape Data to Tidy Format)

Considering we are interested with the total sales only regardless of where the games were sold, having four continents are difficult to see. It would be easier to undertand the data by reshaping it to only 2 variables; Continents and Sales.

After gathered the continents, the datatype is automatically saved as Character. It will be converted to Factor and relabelled.

```
# Gather the sales data from separate continents into single value
joinxp_tidy <- gather(joinxp_filter, key = "Continents", value = "Sales", 5:8)

# Convert datatype to Factor and relabelled it
joinxp_tidy$Continents <- factor(joinxp_tidy$Continents, levels = c("North_America_Total_Sales", "Europe_Total_Sales", "Jap
an_Total_Sales", "Rest_Of_World_Total_Sales"), labels = c("North America", "Europe", "Japan", "Rest of the World" ))

# Check the head of the new reshaped dataset
head(joinxp_tidy, 3)
```

Year	Game	Publisher	Genre	Continents	Sales
<ord>	<chr>	<chr>	<fctr>	<fctr>	<dbl>
2014	Grand Theft Auto V	Rockstar Games	Action	North America	10.76
2015	Call of Duty: Black Ops 3	Activision	Shooter	North America	10.81
2017	Call of Duty: WWII	Activision	Shooter	North America	8.42

3 rows

Scan I (Missing values, special values, or obvious errors)

There are 4 variables with missing values detected using colsums() function. Below are the detail and solution to handle it:

1. Year: 1068 missing values Solution: Since Year is a Factor datatype, a new level name “Unknown” is addded to replace the missing value. Assuming that the publisher did not provide it since the beginning.
2. Publisher: 1068 missing values Solution: Replacing with string: “Unknown”. Assuming that the publisher did not provide it since the beginning.

3. Genre: 8 missing values Solution: Replacing with one of the factor values: “Misc”. Miscellaneous can be for any genres.
4. Sales: 2828 missing values Solution: The missing values occurance are due to the combined datasets from XboxOne and Playstation 4 where both continents are duplicated. In this report, we will omit the value.

There are no special values or obious errors found.

```
# Check for infinite or NaN or NA values
# Create a function to detect any infinite or nan or na values
is.specialorNA <- function(x){
if (is.numeric(x)) (is.infinite(x) | is.nan(x) | is.na(x))
}

# Apply the function to the dataset
sapply(joinxp_tidy, function(x) sum(is.na(x)))
```

```
##       Year       Game  Publisher      Genre Continents      Sales
##       1068          0       1068          8          0       2828
```

```
# Create a copy for scanning purposes
joinxp_scan <- joinxp_tidy

### Year ###
# Observe the Missing values
Empty_Year <- joinxp_scan %>% filter(is.na(Year))
Empty_Year %>% head(3)
```

Year <ord>	Game <chr>	Publisher <chr>	Genre <fctr>	Continents <fctr>	Sales <dbl>
NA	Dance Central: Spotlight	*NA*	Music	North America	*NA*
NA	A Boy and His Blob	*NA*	Platform	North America	0
NA	Another World	*NA*	Adventure	North America	0

3 rows

```
# Adding a new level to a factor
levels(joinxp_scan$Year) <- c(levels(joinxp_scan$Year), "Unknown")
levels(joinxp_scan$Year)
```

```
## [1] "2013"    "2014"    "2015"    "2016"    "2017"    "2018"    "2019"
## [8] "2020"    "Unknown"
```

```
# Replace with Unknown through labels
joinxp_scan <- joinxp_scan %>% mutate(Year = replace(Year, is.na(Year), "Unknown"))


### Publisher ###
# Observe the Missing values
Empty_Publisher <- joinxp_scan %>% filter(is.na(Publisher))
Empty_Publisher %>% head(3)
```

Year <ord>	Game <chr>	Publisher <chr>	Genre <fctr>	Continents <fctr>	Sales <dbl>
Unknown	Dance Central: Spotlight	*NA*	Music	North America	*NA*
Unknown	A Boy and His Blob	*NA*	Platform	North America	0
Unknown	Another World	*NA*	Adventure	North America	0

3 rows

```
# Replace with Unknown
joinxp_scan <- joinxp_scan %>% mutate(Publisher = replace(Publisher, is.na(Publisher), "Unknown"))


### Genre ###
# Observe the Missing values
Empty_Genre <- joinxp_scan %>% filter(is.na(Genre))
Empty_Genre %>% head(3)
```

Year <ord>	Game <chr>	Publisher <chr>	Genre <fctr>	Continents <fctr>	Sales <dbl>
2017	Knowledge is Power	Sony Interactive Entertainment	*NA*	North America	*NA*
2017	That's You	Sony Interactive Entertainment	*NA*	North America	*NA*
2017	Knowledge is Power	Sony Interactive Entertainment	*NA*	Europe	*NA*

3 rows

```
# Replacing the missing values with "Misc"
joinxp_scan <- joinxp_scan %>% mutate(Genre = replace(Genre, is.na(Genre), "Misc"))

joinxp_scan %>% filter(is.na(Genre))
```

0 rows

```
### Sales ###
# Dealing with Missing values in Sales
Empty_sales <- joinxp_scan %>% filter(is.na(Sales))
Empty_sales %>% head(3)
```

Year <ord>	Game <chr>	Publisher <chr>	Genre <fctr>	Continents <fctr>	Sales <dbl>
2014	MineCraft	Microsoft Studios	Misc	North America	*NA*
2015	Halo 5: Guardians	Microsoft Studios	Shooter	North America	*NA*
2015	Star Wars Battlefront (2015)	Electronic Arts	Shooter	North America	*NA*

3 rows

```
# Remove all missing values due to redundant data
joinxp_scan <- na.omit(joinxp_scan)

# Recheck again if still any missing values
colSums(is.na(joinxp_scan))
```

```
##       Year       Game  Publisher      Genre Continents      Sales
##          0          0          0          0          0          0
```

Scan II (Outliers)

We are interested to observe the total games sales. For any non-sale games or 0 sales, we will remove it from the data.

Methodology: 1. The statistic summary is performed and grouped by Continents. As can be seen, the Standard Deviation for Japan and Rest of the World are much lower than North America and Europe. This mean the market in North America and Europe performed much better.

2. In histogram plot, the distribution is right-skewed and when applying a Normal Distribution, it is hardly fitted.
3. Upon checking the outliers using boxplot, there seems to be so many outliers.
4. When performing the Z-score method, it has detected 33 outliers in the Sales data.

```
#Subset the sales with amount greater than 0
joinxp_scan <- subset(joinxp_scan, Sales > 0)
joinxp_scan
```

Year <ord>	Game <chr>	Publisher <chr>
2014	Grand Theft Auto V	Rockstar Games
2015	Call of Duty: Black Ops 3	Activision
2017	Call of Duty: WWII	Activision
2018	Red Dead Redemption 2	Rockstar Games
2014	Call of Duty: Advanced Warfare	Activision
2016	Battlefield 1	Electronic Arts
2015	Fallout 4	Bethesda Softworks
2016	Call of Duty: Infinite Warfare	Activision
2016	FIFA 17	Electronic Arts
2014	Assassin's Creed: Unity	Ubisoft

1-10 of 1,176 rows | 1-3 of 6 columns

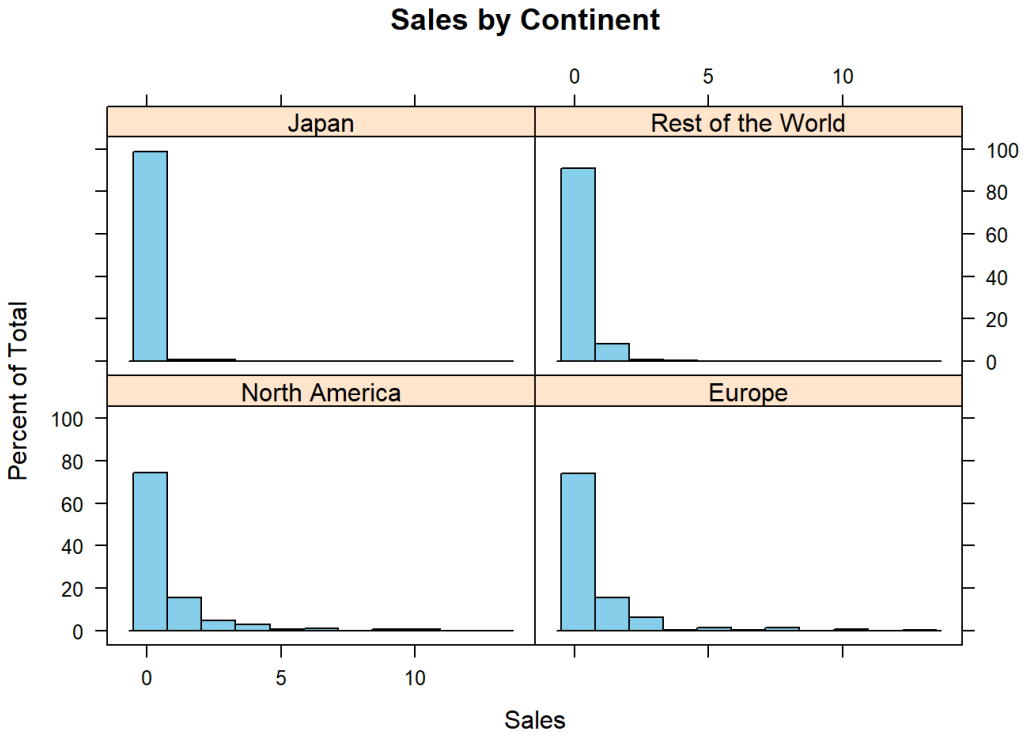
Previous **1** 2 3 4 5 6 … 118 Next

```
# Summary Statistics of Sales grouped by Continents
joinxp_scan %>% group_by(Continents) %>% summarise(
  Mean=mean(Sales),
  Median=median(Sales),
  SD=sd(Sales),
  Q1=quantile(Sales, probs=.25),
  Q3=quantile(Sales,prob=.75),
  IQR=IQR(Sales),
  Min=min(Sales),
  Max=max(Sales),
)
```

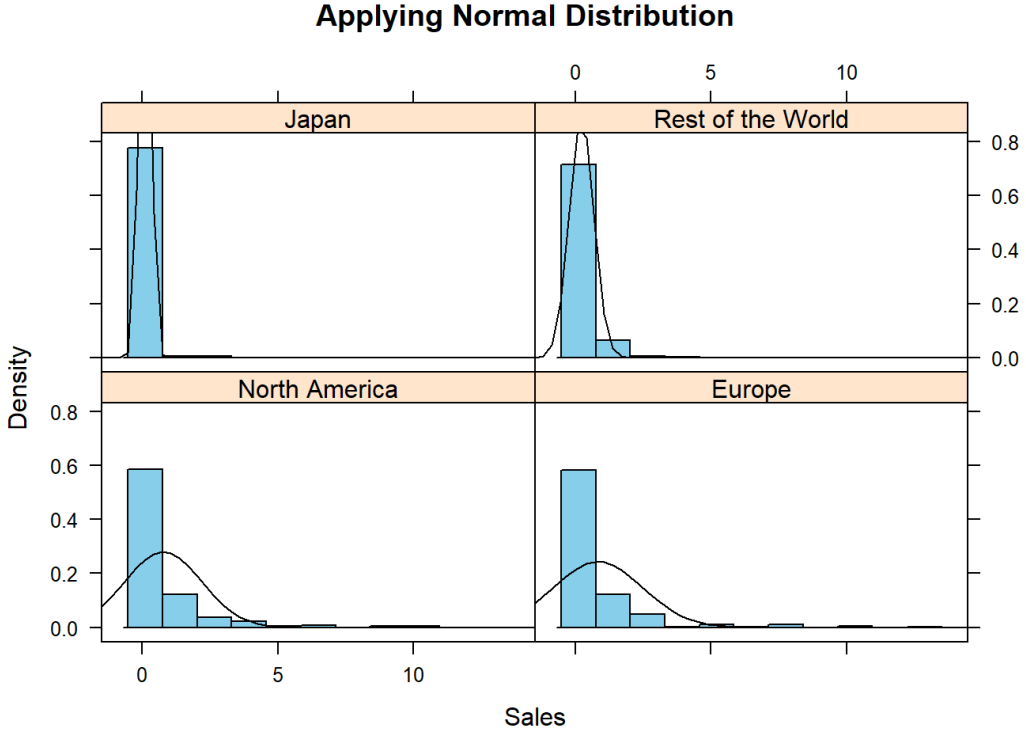
Continents <fctr>	Mean <dbl>	Median <dbl>	SD <dbl>	Q1 <dbl>	Q3 <dbl>	IQR <dbl>	Min <dbl>	Max <dbl>
North America	0.77317280	0.23	1.4219905	0.10	0.78	0.68	0.01	10.81
Europe	0.84187291	0.30	1.6423430	0.08	0.80	0.72	0.01	12.96
Japan	0.09785714	0.04	0.2034850	0.02	0.11	0.09	0.01	2.17
Rest of the World	0.23814607	0.07	0.4474981	0.02	0.24	0.22	0.01	3.78

4 rows

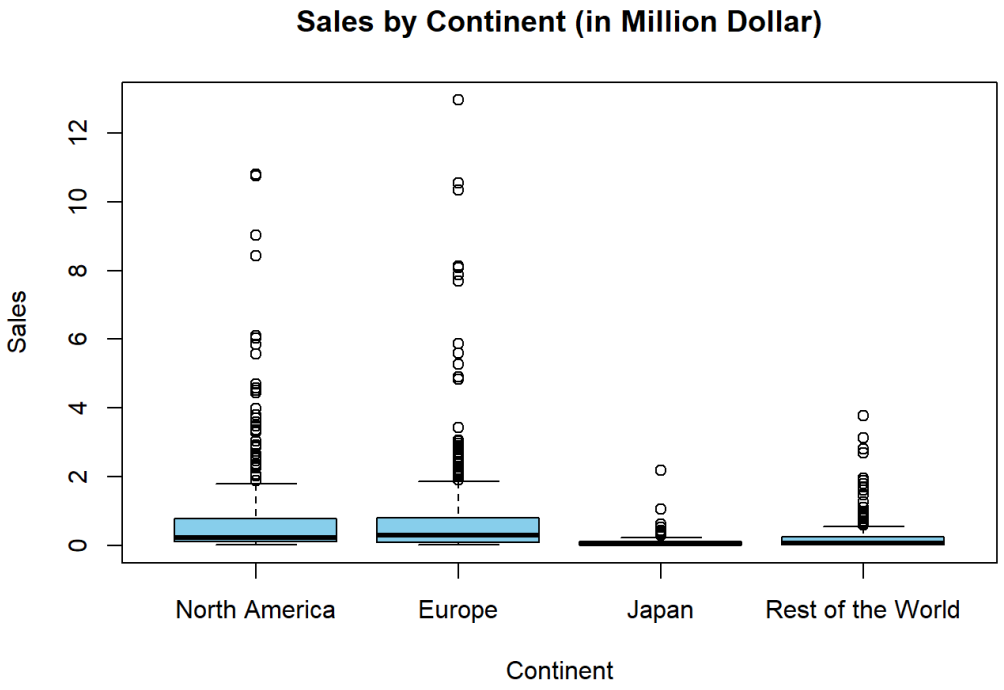
```
# Emprirical Distribution
h <- joinxp_scan %>% histogram(~ Sales|Continents,
                    col="skyblue",
                    layout=c(2,2),
                    data=.,
                    freq=TRUE,
                    main=("Sales by Continent"),
                    )
h
```




```
# Fitting normal distribution
joinxp_scan %>% histogram(~ Sales|Continents, data=.,
          type = "density", col="skyblue",
          main=("Applying Normal Distribution"),
          panel = function(x, ...) {
              panel.histogram(x, ...)
              panel.mathdensity(dmath = dnorm, col = "black",
                                args = list(mean=mean(x),sd=sd(x)))
          } )
```



```
# Boxplot for Sales by Continents
boxplot(joinxp_scan$Sales ~ joinxp_scan$Continents, main = "Sales by Continent (in Million Dollar)", ylab = "Sales", xlab =
"Continent", col="skyblue")
```




```
# Checking outliers using Z-score
z.scores <- joinxp_scan$Sales %>% scores(type = "z")
summary(z.scores)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.43348 -0.40858 -0.33387  0.00000 -0.03503 10.31631
```

```
length(which(abs(z.scores) > 3))
```

```
## [1] 24
```

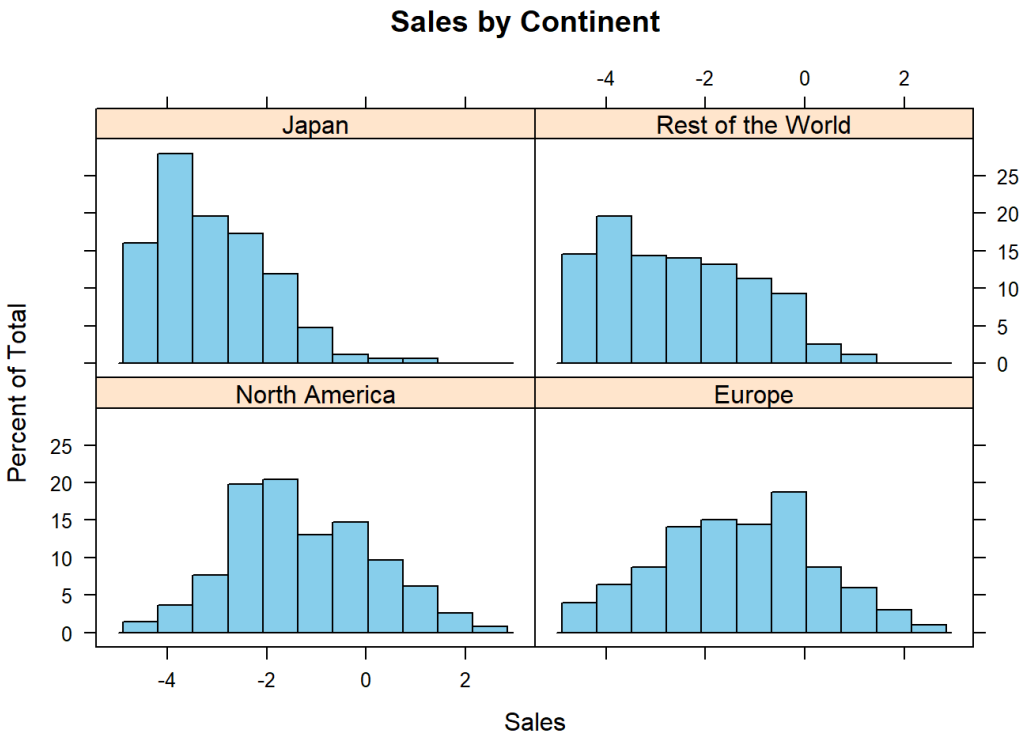
Transform

Applying log transformation is the best way to reduce outliers and decrease the skewness of the empirical distribution.

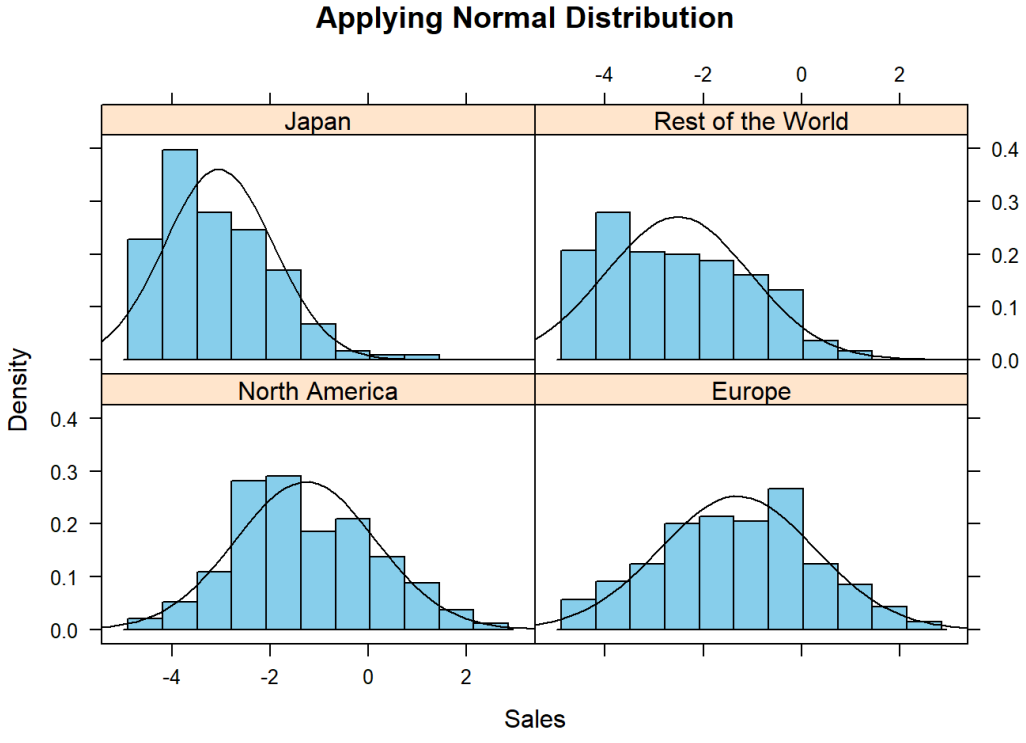
When applying log transformation to the boxplot, the outliers are reduced and only 1 outlier is detected which is normal as it is not fall far away from the Maximum value.

We can also perform comparison of Sales by Genre using the same method.

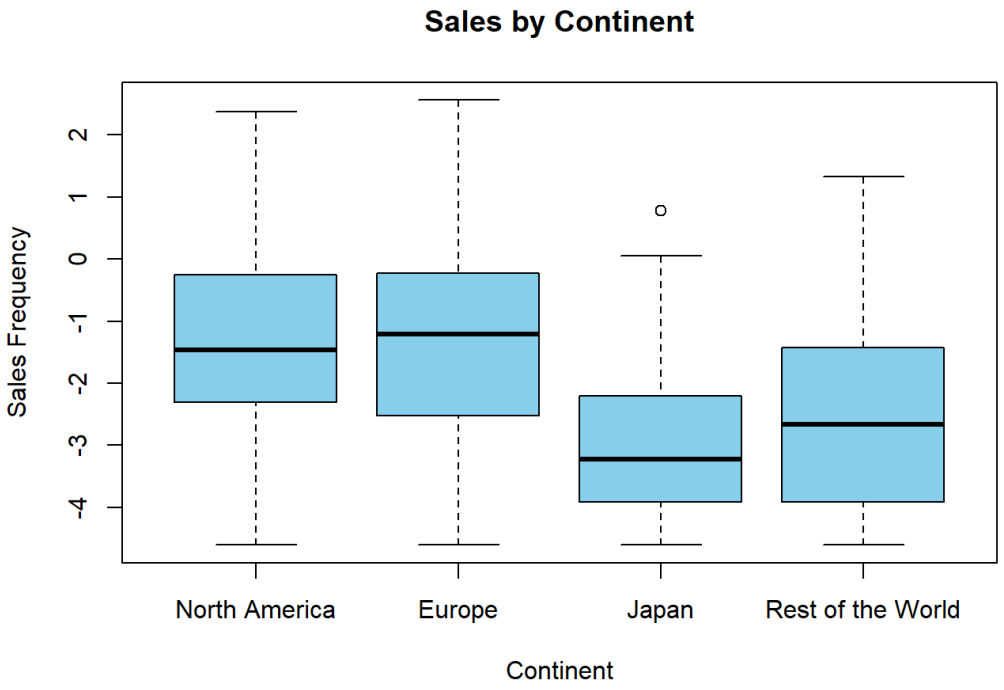
```
# Empirical Distribution with Log(Sales)
h2 <- joinxp_scan %>%  histogram(~log(Sales)|Continents,
                    col="skyblue",
                    data=.,
                    freq=TRUE, xlab = "Sales",
                    main=("Sales by Continent"),
                    )
h2
```



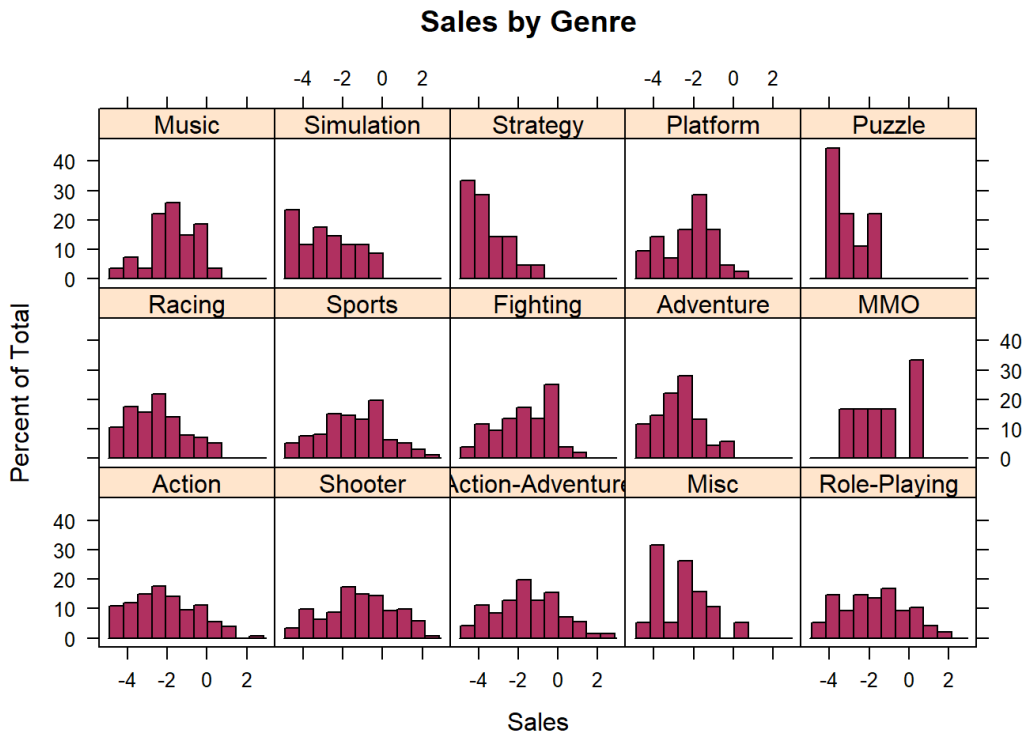

```
# Fitting normal distributions on log(Sales)
joinxp_scan %>% histogram(~ log(Sales)|Continents, data=.,
          type = "density", col="skyblue",
          main=("Applying Normal Distribution"), xlab = "Sales",
          panel = function(x, ...) {
              panel.histogram(x, ...)
              panel.mathdensity(dmath = dnorm, col = "black",
                                args = list(mean=mean(x),sd=sd(x)))
          } )
```



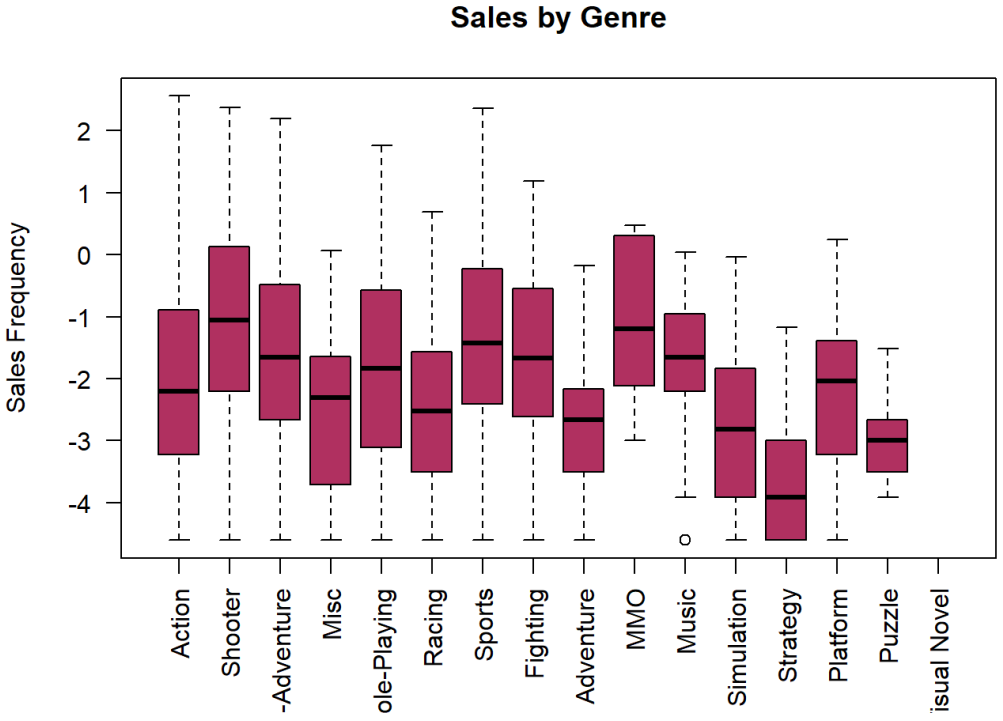
```
# Check for Outliers using boxplot after log
boxplot(log(joinxp_scan$Sales) ~ joinxp_scan$Continents, main = "Sales by Continent", ylab = "Sales Frequency", xlab = "Con
tinent", col = "skyblue")
```



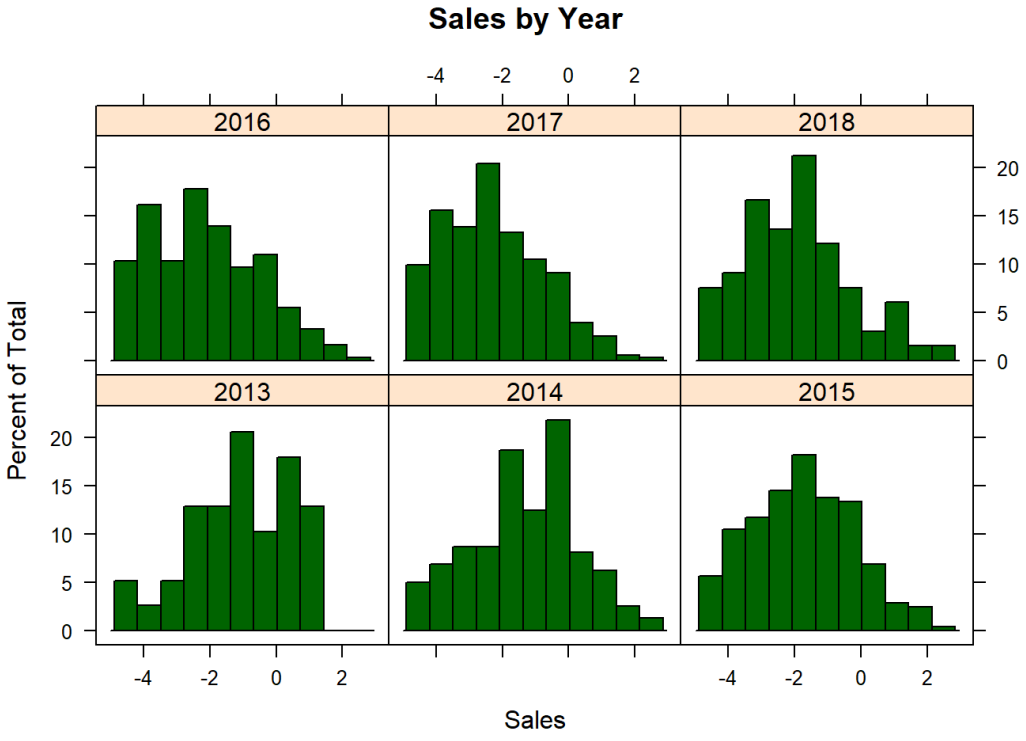

```
### Additional work: Sales comparison by Genre
# Histogram
h3 <- joinxp_scan %>%  histogram(~log(Sales)|Genre,
                    col="maroon",
                    data=.,
                    freq=TRUE,
                    xlab = "Sales",
                    main=("Sales by Genre")
                    )
h3
```



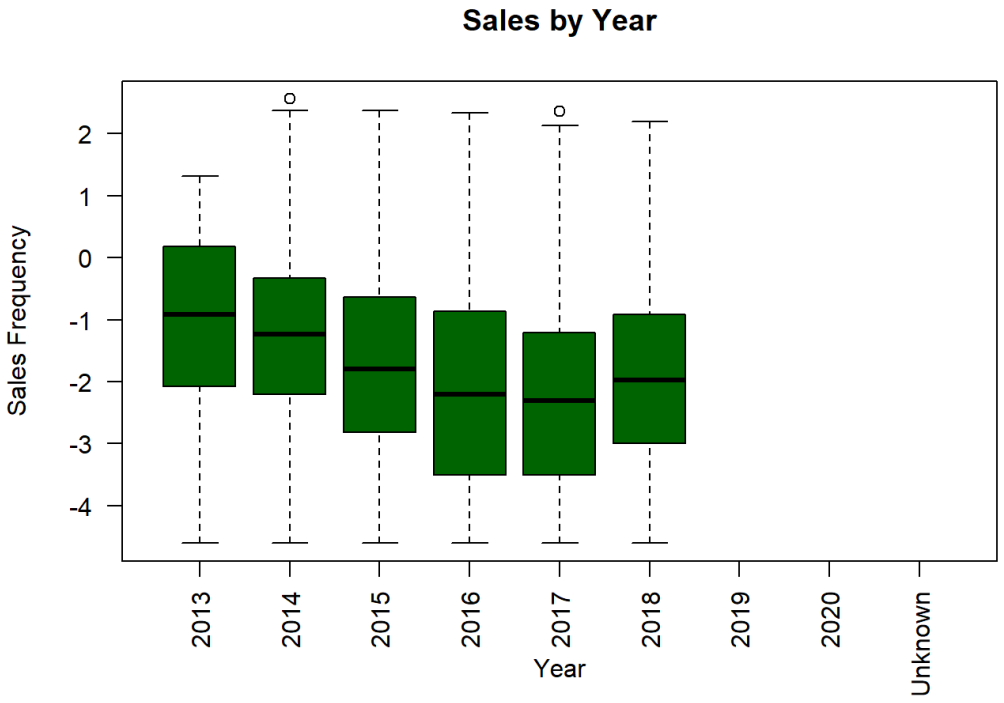

```
# Boxplot
boxplot(log(joinxp_scan$Sales) ~ joinxp_scan$Genre, main = "Sales by Genre", ylab = "Sales Frequency", xlab = " ", col = "m
aroon", par(las=2))
```



```
### Additional work: Sales comparison by Year
# Histogram
h4 <- joinxp_scan %>%  histogram(~log(Sales)|Year,
                    col="darkgreen",
                    data=.,
                    freq=TRUE,
                    xlab = "Sales",
                    main=("Sales by Year")
                    )
h4
```




```
# Boxplot
boxplot(log(joinxp_scan$Sales) ~ joinxp_scan$Year, main = "Sales by Year", ylab = "Sales Frequency", xlab = "Year", col =
"darkgreen")
```



Conclusion

European countries perform the best sales among the other continents followed by North America, Japan and the rest of world. We can assume that many gamers are based in Europe, or possibly because there are more countries in Europe than the rest of the world.

For marketing strategy, it is best to hold more events in European and North American countries.