DSCC 462 Final Project

Brynn (Yein) Lee, Medhini Sridharr, Wonha Shin

2023-12-14

Index

#0. Introduction {#tag0}

In this project, we embark on a comprehensive analysis of the gaming industry, leveraging the extensive 'Video Games Sales' dataset sourced from Kaggle. Our goal is to delve deep into this rich source of gaming data, aiming to unveil key trends and glean insightful perspectives within this dynamic and ever-evolving sector.

We will employ a range of statistical methods and tests, each carefully selected to address specific questions and hypotheses. Through this thorough exploration, we aim to unearth valuable insights that will not only deepen our understanding of the gaming industry but also provide actionable intelligence within this dynamic sector.

#1. Finding and Cleaning Data {#tag1}

Getting and Cleaning Data

```
# Loading required libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to bec
```

```
library(ggplot2)
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(dplyr)
library(e1071)
library(readr)
library(tidyr)
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(stats)
# library(qqplotr)

# Reading data
video <- read.csv("Video_Games_Sales.csv")

# Check data
print(head(video))
```

```
##                       Name Platform Year_of_Release        Genre Publisher
## 1               Wii Sports      Wii            2006       Sports  Nintendo
## 2        Super Mario Bros.      NES            1985     Platform  Nintendo
## 3           Mario Kart Wii      Wii            2008       Racing  Nintendo
## 4        Wii Sports Resort      Wii            2009       Sports  Nintendo
## 5 Pokemon Red/Pokemon Blue       GB            1996 Role-Playing  Nintendo
## 6                   Tetris       GB            1989       Puzzle  Nintendo
##   NA_Sales EU_Sales JP_Sales Other_Sales Global_Sales Critic_Score Critic_Count
## 1    41.36    28.96     3.77        8.45        82.53           76           51
## 2    29.08     3.58     6.81        0.77        40.24           NA           NA
## 3    15.68    12.76     3.79        3.29        35.52           82           73
## 4    15.61    10.93     3.28        2.95        32.77           80           73
```

```
## 5    11.27     8.89    10.22        1.00        31.37           NA           NA
## 6    23.20     2.26     4.22        0.58        30.26           NA           NA
##   User_Score User_Count Developer Rating
## 1        8.0        322  Nintendo      E
## 2         NA         NA
## 3        8.3        709  Nintendo      E
## 4        8.0        192  Nintendo      E
## 5         NA         NA
## 6         NA         NA
```

```
print(tail(video))
```

```
##                                                 Name Platform Year_of_Release
## 16714 SCORE International Baja 1000: The Official Game      PS2            2008
## 16715                     Samurai Warriors: Sanada Maru      PS3            2016
## 16716                                LMA Manager 2007     X360            2006
## 16717                          Haitaka no Psychedelica      PSV            2016
## 16718                                 Spirits & Spells      GBA            2003
## 16719                               Winning Post 8 2016      PSV            2016
##            Genre    Publisher NA_Sales EU_Sales JP_Sales Other_Sales
## 16714     Racing   Activision     0.00     0.00     0.00           0
## 16715     Action   Tecmo Koei     0.00     0.00     0.01           0
## 16716     Sports  Codemasters     0.00     0.01     0.00           0
## 16717  Adventure Idea Factory     0.00     0.00     0.01           0
## 16718   Platform      Wanadoo     0.01     0.00     0.00           0
## 16719 Simulation   Tecmo Koei     0.00     0.00     0.01           0
##       Global_Sales Critic_Score Critic_Count User_Score User_Count Developer
## 16714         0.01           NA           NA         NA         NA
## 16715         0.01           NA           NA         NA         NA
## 16716         0.01           NA           NA         NA         NA
## 16717         0.01           NA           NA         NA         NA
## 16718         0.01           NA           NA         NA         NA
## 16719         0.01           NA           NA         NA         NA
##       Rating
## 16714
## 16715
## 16716
## 16717
## 16718
## 16719
```

By looking at the first and last few rows of the data, we see that the data has missing values, some redundant columns, some columns that are correlated and some columns that have many values close to 0. This shows that the data requires some cleaning. We will also be checking the descriptive statistics of the data to decide if further data preprocessing is required.

```r
## Data Cleaning Checking null values
nulls <- data.frame(Column_Name = names(video), Null_Count = sapply(video,
    function(x) sum(is.na(x) | x == "")))
nulls$Null_Percentage <- (nulls$Null_Count/nrow(video)) * 100
print(nulls)
```

```
##                     Column_Name Null_Count Null_Percentage
## Name                        Name          2      0.01196244
## Platform                Platform          0      0.00000000
## Year_of_Release Year_of_Release          0      0.00000000
## Genre                       Genre          2      0.01196244
## Publisher               Publisher          0      0.00000000
## NA_Sales                 NA_Sales          0      0.00000000
## EU_Sales                 EU_Sales          0      0.00000000
## JP_Sales                 JP_Sales          0      0.00000000
## Other_Sales           Other_Sales          0      0.00000000
## Global_Sales         Global_Sales          0      0.00000000
## Critic_Score         Critic_Score       8582     51.33082122
## Critic_Count         Critic_Count       8582     51.33082122
## User_Score             User_Score       9129     54.60254800
## User_Count             User_Count       9129     54.60254800
## Developer               Developer       6623     39.61361325
## Rating                     Rating       6769     40.48687122
```

There are 8 columns with NULL values. These will be handled by dropping some rows that contain NULL values and then handling the remaining missing values for each of these columns. Some more EDA and visualization is done before this to make handling of missing values easier and more accurate. Irrelevant and redundant columns will also be dropped.

```r
# Drop rows where Rating = NULL
df <- video[!(is.na(video$Rating) | video$Rating == ""), ]

# Drop Developer column as this has info that is very
# similar to Publisher column (which has no NULLs)
df <- df[, !(names(df) %in% c("Developer"))]

# Dropping User_Count and Critic_Count as we have the
# average score columns:
df <- df[, !(names(df) %in% c("Critic_Count", "User_Count"))]
```

We first look at the relative frequency tables and barplots to further understand the data.

#2. Descriptive Statistics {#tag2}

2.1 Relative Frequency Table

Relative Frequency Table

```r
# Relative frequency table for 'Platform'
platform_table <- table(df$Platform)
platform_relative_freq <- prop.table(platform_table) * 100

# Print the relative frequency table
print("Relative Frequency Table for Platform:")
```

```
## [1] "Relative Frequency Table for Platform:"
```

```r
platform_relative_freq
```

```
##
##        3DS         DC         DS        GBA         GC         PC         PS
##  2.2914573  0.1407035 12.8040201  5.2462312  4.7135678  7.7788945  2.0904523
##        PS2        PS3        PS4        PSP        PSV        Wii       WiiU
## 14.8844221  9.5678392  2.5628141  5.4673367  1.5175879 10.0703518  1.0552764
##       X360         XB       XOne
## 10.5728643  7.3668342  1.8693467
```

```r
# Relative frequency table for 'Genre'
genre_table <- table(df$Genre)
genre_relative_freq <- prop.table(genre_table) * 100

# Print the relative frequency table
print("Relative Frequency Table for Genre:")
```

```
## [1] "Relative Frequency Table for Genre:"
```

```r
genre_relative_freq
```

```
##
##       Action    Adventure     Fighting         Misc     Platform       Puzzle
##    21.989950     4.482412     4.402010     8.864322     5.718593     3.437186
##       Racing Role-Playing      Shooter   Simulation       Sports     Strategy
##     8.763819     7.809045    10.241206     5.718593    15.165829     3.407035
```

```r
# Relative frequency table for 'Publisher'
publisher_table <- table(df$Publisher)
publisher_relative_freq <- prop.table(publisher_table) * 100

# Print the relative frequency table (top 10)
print("Relative Frequency Table for Publisher (Top 10):")
```

```
## [1] "Relative Frequency Table for Publisher (Top 10):"
```

```
head(publisher_relative_freq[order(publisher_relative_freq, decreasing = TRUE)],
    10)
```

```
##
##              Electronic Arts                       Ubisoft
##                    11.447236                      7.879397
##                    Activision                           THQ
##                     7.849246                      5.346734
## Konami Digital Entertainment  Sony Computer Entertainment
##                     3.839196                      3.688442
##         Take-Two Interactive                          Sega
##                     3.648241                      3.246231
##           Namco Bandai Games                      Nintendo
##                     3.105528                      3.105528
```

```
# Relative frequency table for 'Rating'
rating_table <- table(df$Rating)
rating_relative_freq <- prop.table(rating_table) * 100

# Print the relative frequency table
print("Relative Frequency Table for Rating:")
```

```
## [1] "Relative Frequency Table for Rating:"
```

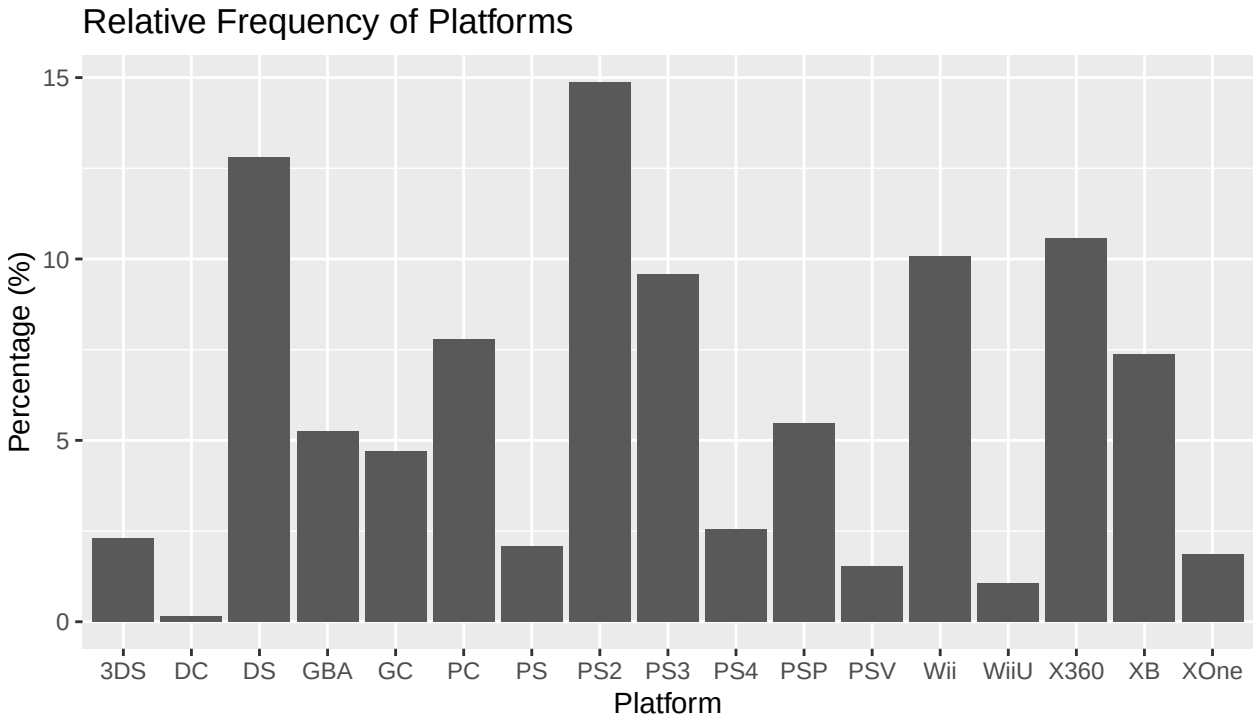
```
rating_relative_freq
```

```
##
##          AO           E        E10+          EC         K-A           M
##  0.01005025 40.11055276 14.27135678  0.08040201  0.03015075 15.70854271
##          RP           T
##  0.03015075 29.75879397
```

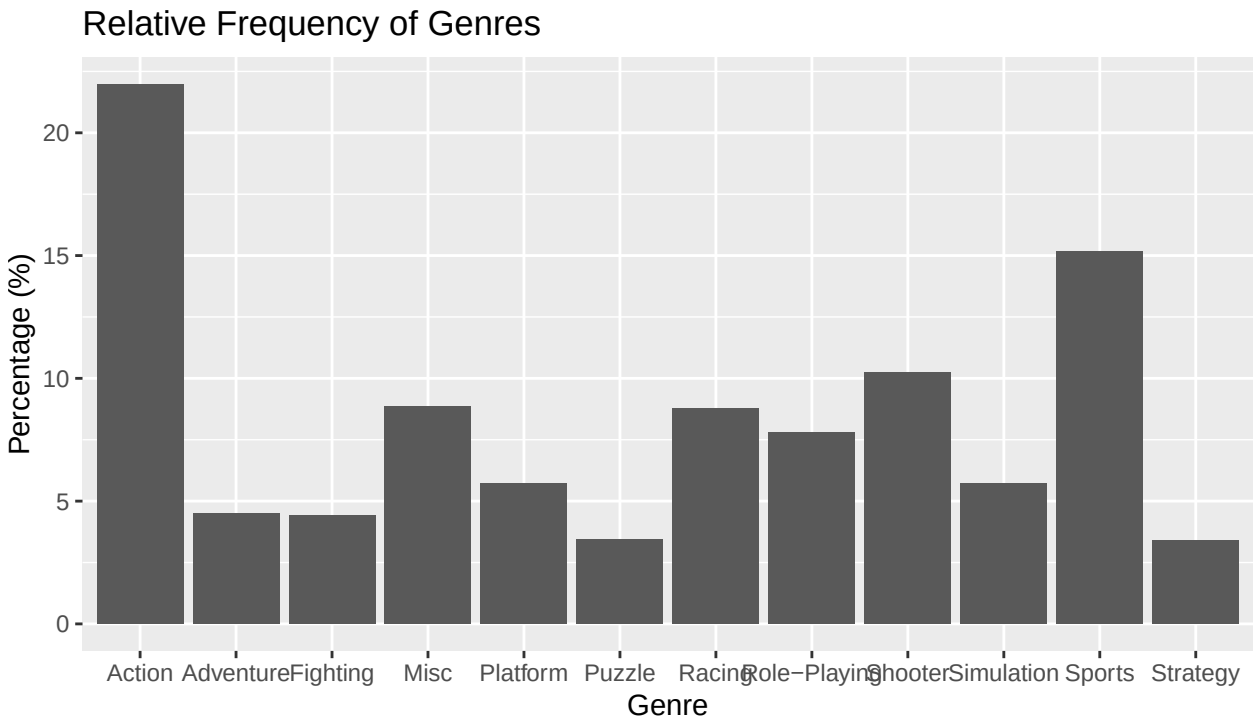
2.2 Relative frequency barplot

Relative or absolute frequency barplot

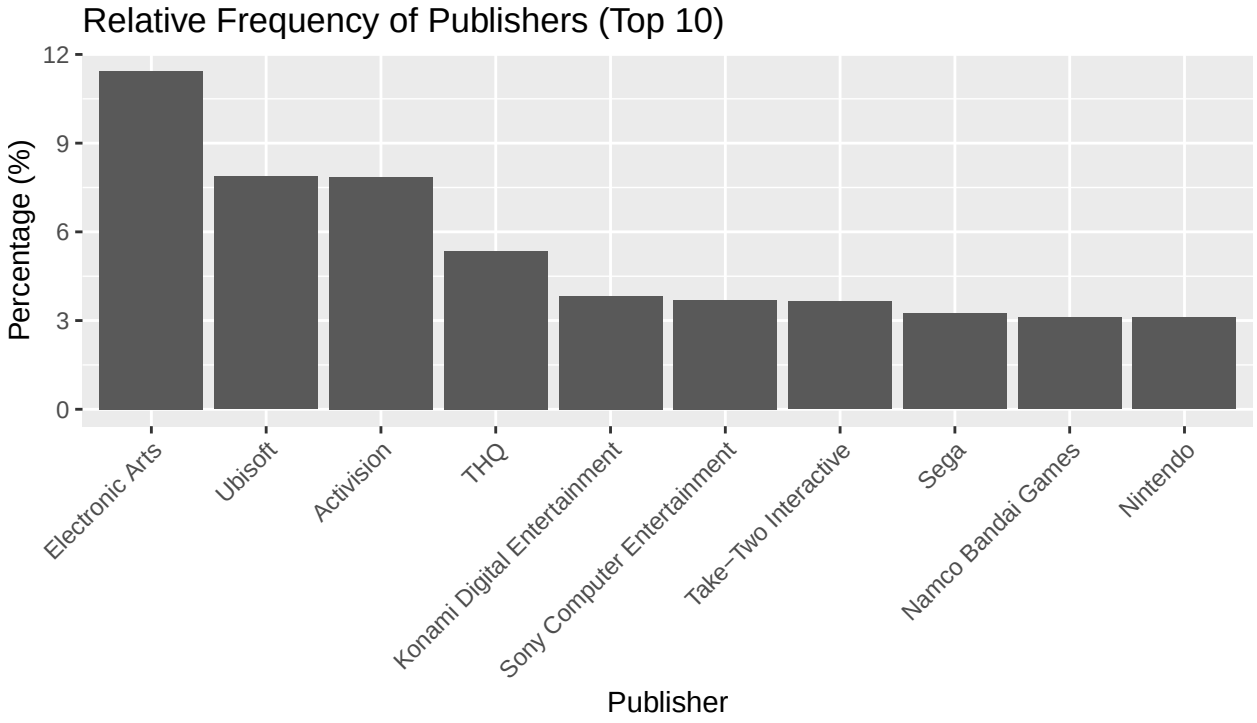
```r
# Convert tables to data frames for ggplot
df_platform <- as.data.frame(platform_relative_freq)
df_genre <- as.data.frame(genre_relative_freq)
df_publisher <- as.data.frame(head(publisher_relative_freq[order(publisher_relative_freq,
    decreasing = TRUE)], 10))
df_rating <- as.data.frame(rating_relative_freq)

names(df_platform) <- c("Platform", "Frequency")
names(df_genre) <- c("Genre", "Frequency")
names(df_publisher) <- c("Publisher", "Frequency")
names(df_rating) <- c("Rating", "Frequency")

# Plot for Platforms
p_platform <- ggplot(df_platform, aes(x = Platform, y = Frequency)) +
    geom_bar(stat = "identity") + labs(title = "Relative Frequency of Platforms",
    x = "Platform", y = "Percentage (%)") + theme(plot.margin = unit(c(1,
    0, 1, 0), "cm"))
print(p_platform)
```



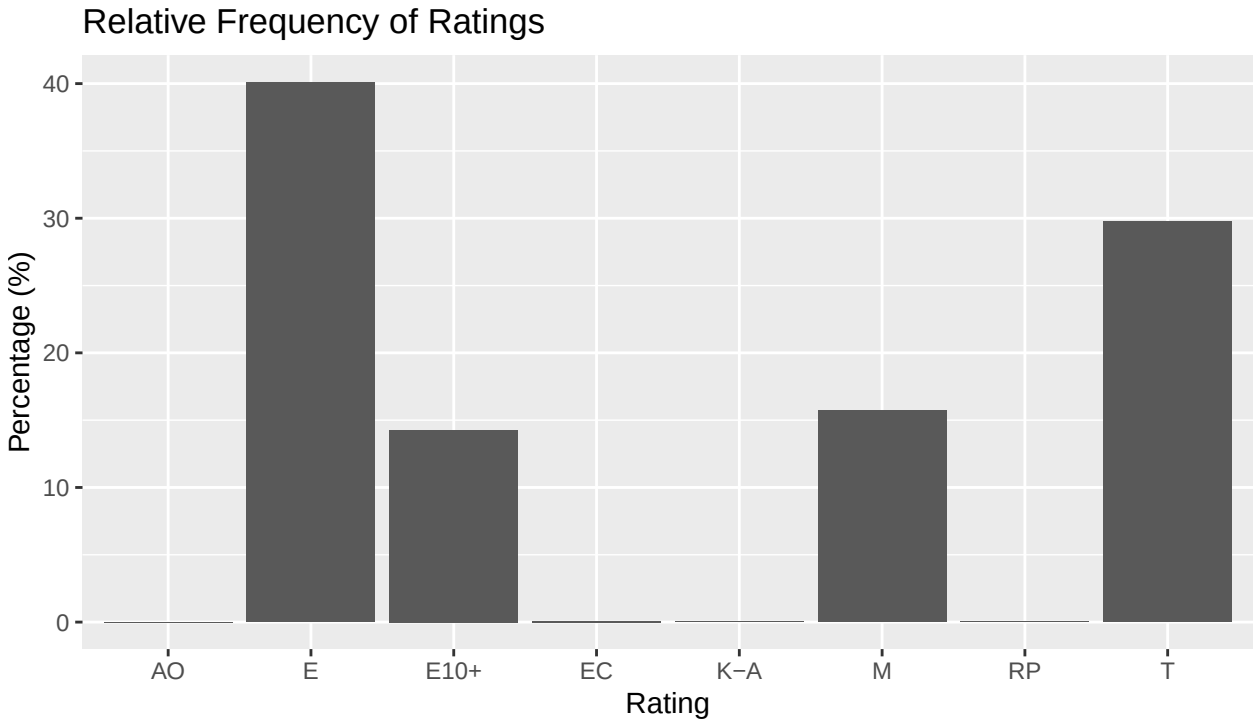

```r
# Plot for Genres
p_genre <- ggplot(df_genre, aes(x = Genre, y = Frequency)) +
    geom_bar(stat = "identity") + labs(title = "Relative Frequency of Genres",
    x = "Genre", y = "Percentage (%)") + theme(plot.margin = unit(c(1,
    0, 1, 0), "cm"))
print(p_genre)
```




```r
# Plot for Publishers
p_publisher <- ggplot(df_publisher, aes(x = Publisher, y = Frequency)) +
    geom_bar(stat = "identity") + labs(title = "Relative Frequency of Publishers (Top 10)",
    x = "Publisher", y = "Percentage (%)") + theme(axis.text.x = element_text(angle = 45,
    hjust = 1)) + theme(plot.margin = unit(c(1, 0, 1, 0), "cm"))
print(p_publisher)
```



```r
# Plot for Ratings
p_rating <- ggplot(df_rating, aes(x = Rating, y = Frequency)) +
    geom_bar(stat = "identity") + labs(title = "Relative Frequency of Ratings",
    x = "Rating", y = "Percentage (%)") + theme(plot.margin = unit(c(1,
    0, 1, 0), "cm"))
print(p_rating)
```



The frequency tables and barplots tell us that the dataset predominantly features PS2 as the most common gaming platform and 'Action' and 'Sports' as the leading game genres, with Electronic Arts being the top publisher. Most games are rated as suitable for 'Everyone' (E) or 'Teen' (T), highlighting a focus on general and teen audiences in the video game industry.
Now, we look at the summary statistics of the numerical columns to discern what kind of pre-processing they might need and see if any rows need to be dropped before we handle missing values of Critic Score and User Score (these are now the only columns that require missing value imputation).

2.3 Summaries of Center and Dispersion

Summaries of center and dispersion

Sales

```r
# List of sales region columns
sales_regions <- c("NA_Sales", "EU_Sales", "JP_Sales", "Other_Sales",
    "Global_Sales")
```

```r
# Create a function to calculate and print summaries
calculate_and_print_summaries <- function(region_column) {
    result <- paste("Region:", region_column, "\n", "Mean:",
        mean(df[, region_column]), "\n", "Median:", median(df[,
            region_column]), "\n", "Mode:", as.numeric(names(sort(table(df[,
            region_column]), decreasing = TRUE)[1])), "\n", "Trimmed Mean (10%):",
        mean(df[, region_column], trim = 0.1), "\n", "IQR:",
        IQR(df[, region_column]), "\n", "Variance:", var(df[,
            region_column]), "\n", "Coefficient of Variation:",
        (sd(df[, region_column])/mean(df[, region_column])) *
            100, "%\n", "Skewness:", skewness(df[, region_column]),
        "\n")
    cat(result)
    cat("\n")
}

# Loop through sales regions and calculate/print summaries
for (region in sales_regions) {
    calculate_and_print_summaries(region)
}
```

```
## Region: NA_Sales
##  Mean: 0.317604020100507
##  Median: 0.12
##  Mode: 0
##  Trimmed Mean (10%): 0.177045226130654
##  IQR: 0.25
##  Variance: 0.673729608989711
##  Coefficient of Variation: 258.438270565771 %
##  Skewness: 18.4748244697473
##
## Region: EU_Sales
##  Mean: 0.181993969849243
##  Median: 0.04
##  Mode: 0
##  Trimmed Mean (10%): 0.0803002512562827
##  IQR: 0.14
##  Variance: 0.339900858341369
##  Coefficient of Variation: 320.345872172759 %
##  Skewness: 18.428204060119
##
## Region: JP_Sales
##  Mean: 0.0457698492462308
##  Median: 0
##  Mode: 0
##  Trimmed Mean (10%): 0.00469346733668375
##  IQR: 0
```

```
##  Variance: 0.0579920045185156
##  Coefficient of Variation: 526.143946844317 %
##  Skewness: 12.575536978153
##
## Region: Other_Sales
##  Mean: 0.0643477386934691
##  Median: 0.02
##  Mode: 0.01
##  Trimmed Mean (10%): 0.0286633165829143
##  IQR: 0.04
##  Variance: 0.0520918601692693
##  Coefficient of Variation: 354.692204458128 %
##  Skewness: 21.8318074348466
##
## Region: Global_Sales
##  Mean: 0.609958793969855
##  Median: 0.21
##  Mode: 0.02
##  Trimmed Mean (10%): 0.328801507537688
##  IQR: 0.48
##  Variance: 2.76159665123186
##  Coefficient of Variation: 272.445492234404 %
##  Skewness: 18.5046722375061
```

Critic Score

```r
# List of columns for which we want to calculate summaries
score_columns <- c("Critic_Score")

# Create a function to calculate and print summaries
calculate_and_print_summaries <- function(score_column) {
    result <- paste("Summary Statistics for", score_column, "\n",
        "Mean:", mean(df[, score_column], na.rm = TRUE), "\n",
        "Median:", median(df[, score_column], na.rm = TRUE),
        "\n")

    mode_val <- as.numeric(names(sort(table(df[, score_column]),
        decreasing = TRUE)[1]))
    if (!is.na(mode_val)) {
        result <- paste(result, "Mode:", mode_val, "\n")
    } else {
        result <- paste(result, "Mode: No unique mode (multiple values with the same highest fr
    }
```

```
    result <- paste(result, "Trimmed Mean (10%):", mean(df[,
        score_column], trim = 0.1, na.rm = TRUE), "\n", "IQR:",
        IQR(df[, score_column], na.rm = TRUE), "\n", "Variance:",
        var(df[, score_column], na.rm = TRUE), "\n", "Coefficient of Variation:",
        (sd(df[, score_column], na.rm = TRUE)/mean(df[, score_column],
            na.rm = TRUE)) * 100, "%\n", "Skewness:", skewness(df[,
            score_column], na.rm = TRUE), "\n")

    cat(result)
    cat("\n")
}

# Loop through score columns and calculate/print summaries
for (score_col in score_columns) {
    calculate_and_print_summaries(score_col)
}
```

```
## Summary Statistics for Critic_Score
##  Mean: 68.9713185994537
##  Median: 71
##  Mode: 70
##  Trimmed Mean (10%): 69.8846989447548
##  IQR: 19
##  Variance: 194.648252154039
##  Coefficient of Variation: 20.2281760724356 %
##  Skewness: -0.61385582566675
```

User Score

```
score_columns_2 <- c("User_Score")

for (score_col in score_columns_2) {
    calculate_and_print_summaries(score_col)
}
```

```
## Summary Statistics for User_Score
##  Mean: 7.1268789978678
##  Median: 7.5
##  Mode: 7.8
##  Trimmed Mean (10%): 7.31693870752832
##  IQR: 1.8
##  Variance: 2.25224956765693
##  Coefficient of Variation: 21.0575999536448 %
##  Skewness: -1.25870392719268
```

The values (mean and median) are also quite low for JP_Sales and Other_Sales, i.e., they are quite close to 0. Other_Sales and JP_Sales also have very low mean and median values, indicating lower sales overall, but also the highest variability and widest range of data.

```
# Combine Japan and Other Sales columns as they have very
# low numbers and can be combined
df$Other_Sales <- df$Other_Sales + df$JP_Sales
df <- df[, !(names(df) %in% c("JP_Sales"))]
```

The skewness values of sales for all regions are considerably high, indicating that the data is highly skewed. This skewness is further emphasized by the large difference between the mean and median values.
Upon inspection of the data, we see that there are quite a few rows where Sales = 0 (for all sales columns). These rows are dropped as they do not provide information that is useful for analysis and they severely skew the data. We then handle remaining missing values.

```
# Remove rows where Sales = 0
sales_cols <- grep("Sales", names(df), value = TRUE)
for (i in sales_cols) {
    print(paste(i, ":", sum(df[, i] == 0)))
}
```

```
## [1] "NA_Sales : 776"
## [1] "EU_Sales : 2134"
## [1] "Other_Sales : 2163"
## [1] "Global_Sales : 0"
```

```
# Get list of indices where Sales = 0
na <- which(df$NA_Sales == 0)
eu <- which(df$EU_Sales == 0)
ot <- which(df$Other_Sales == 0)
all_idx <- c(na, eu, ot)

# Drop rows where Sales = 0
df <- df[-all_idx, ]
print(nrow(df))
```

```
## [1] 6130
```

```
# Imputation for Critic Score and User Score - Replacing
# with median
platforms_critic_NaN <- unique(df$Platform[is.na(df$Critic_Score)])
platforms_user_NaN <- unique(df$Platform[is.na(df$User_Score)])

for (platform in platforms_critic_NaN) {
```

```
    median_score <- median(df$Critic_Score[df$Platform == platform],
        na.rm = TRUE)
    df$Critic_Score[is.na(df$Critic_Score) & df$Platform == platform] <- median_score
}

for (platform in platforms_user_NaN) {
    median_score <- median(df$User_Score[df$Platform == platform],
        na.rm = TRUE)
    df$User_Score[is.na(df$User_Score) & df$Platform == platform] <- median_score
}

# Confirming if NULLs have been accounted for:
null_counts <- data.frame(Column_Name = names(df), Null_Count = sapply(df,
    function(x) sum(is.na(x) | x == "")))
print(null_counts)
```

```
##                     Column_Name Null_Count
## Name                        Name          0
## Platform                Platform          0
## Year_of_Release Year_of_Release          0
## Genre                      Genre          0
## Publisher              Publisher          0
## NA_Sales                NA_Sales          0
## EU_Sales                EU_Sales          0
## Other_Sales          Other_Sales          0
## Global_Sales        Global_Sales          0
## Critic_Score        Critic_Score          0
## User_Score            User_Score          0
## Rating                    Rating          0
```

2.4 Side-by-Side Boxplots

Side By Side Boxplots

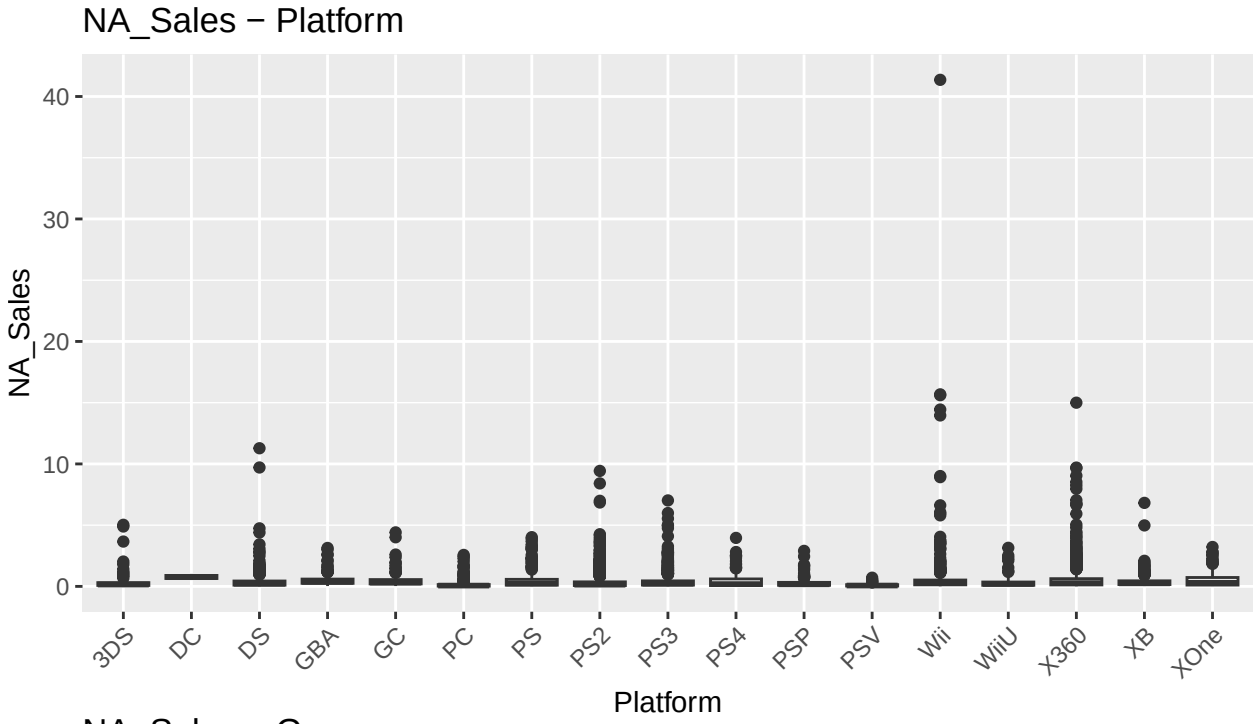
We now check the distrubtion of Sales data within each category of the categorical variables using side-by-side boxplots.

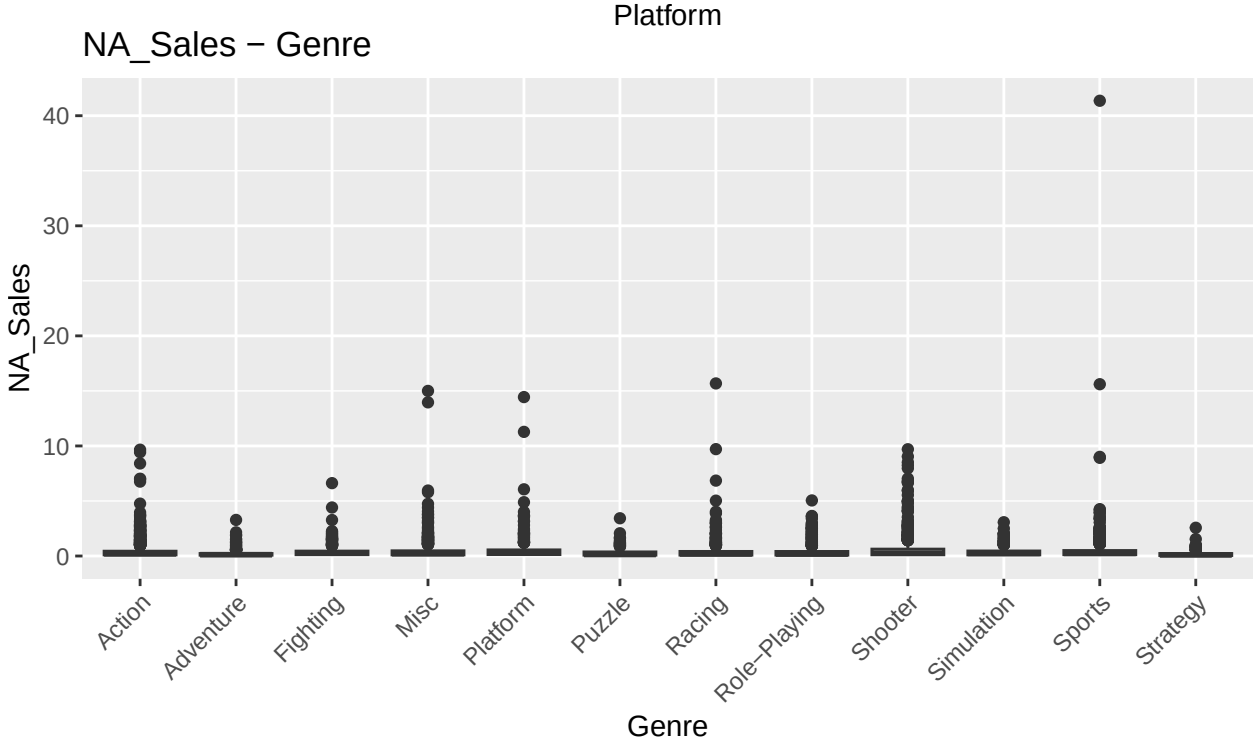
```
# Descriptive Stats - Side By Side Boxplots
categorical_cols <- c("Platform", "Genre", "Rating")

# cat_plots - function for plotting side-by-side boxplots
# with modified titles
cat_plots <- function(target, df = df, cat_cols = categorical_cols,
```

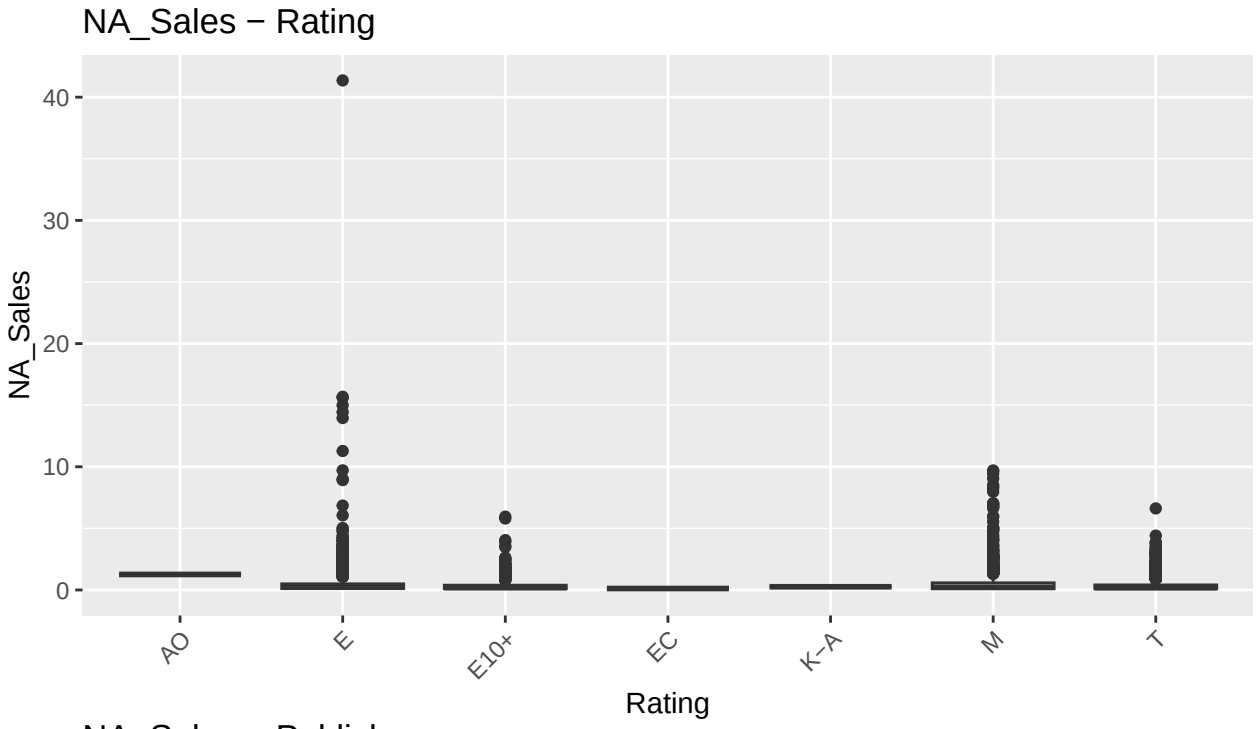
```
    limit_yax = FALSE, l = 20, b = 10) {
    par(mfrow = c(length(cat_cols), 1), mar = c(4, 4, 2, 1))

    for (col in cat_cols) {
        p <- ggplot(df, aes_string(x = col, y = target)) + geom_boxplot() +
            theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
            ggtitle(paste(target, "-", col)) + theme(plot.margin = unit(c(1,
            0, 1, 0), "cm"))

        if (limit_yax) {
            p <- p + ylim(0, max(df[[target]]))
        }

        # Print the boxplot
        print(p)
    }

    par(mfrow = c(1, 1))  # Reset the plotting layout to default
}

# Preprocessing for Publisher - To improve interpretability
# of boxplot, we look only at the top values
publisher_count <- as.data.frame(table(df$Publisher))
names(publisher_count) <- c("Publisher", "Count")

publisher_count <- publisher_count[publisher_count$Count >= 149,
    , drop = FALSE]
video_publisher <- df[df$Publisher %in% publisher_count$Publisher,
    , drop = FALSE]

num_cols <- c("NA_Sales", "EU_Sales", "Other_Sales", "Global_Sales",
    "Critic_Score", "User_Score")

par(mfrow = c(2, 1))

# Calling function for each sales column
for (col in num_cols) {
    cat_plots(col, df = df, cat_cols = c("Platform", "Genre",
        "Rating", "Publisher"), limit_yax = TRUE, l = 30, b = 8)

    # For Publisher
    cat_plots(col, df = video_publisher, cat_cols = c("Publisher",
        "Rating"), limit_yax = TRUE, l = 30, b = 8)
}
```

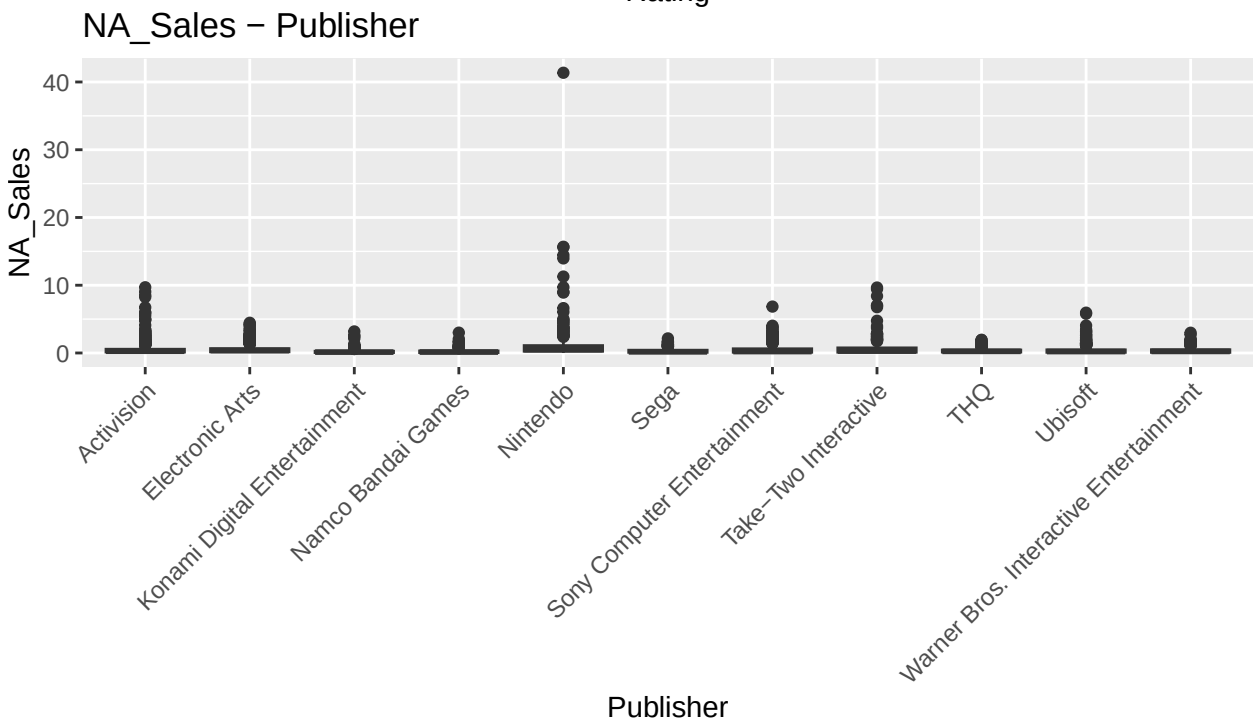
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
```

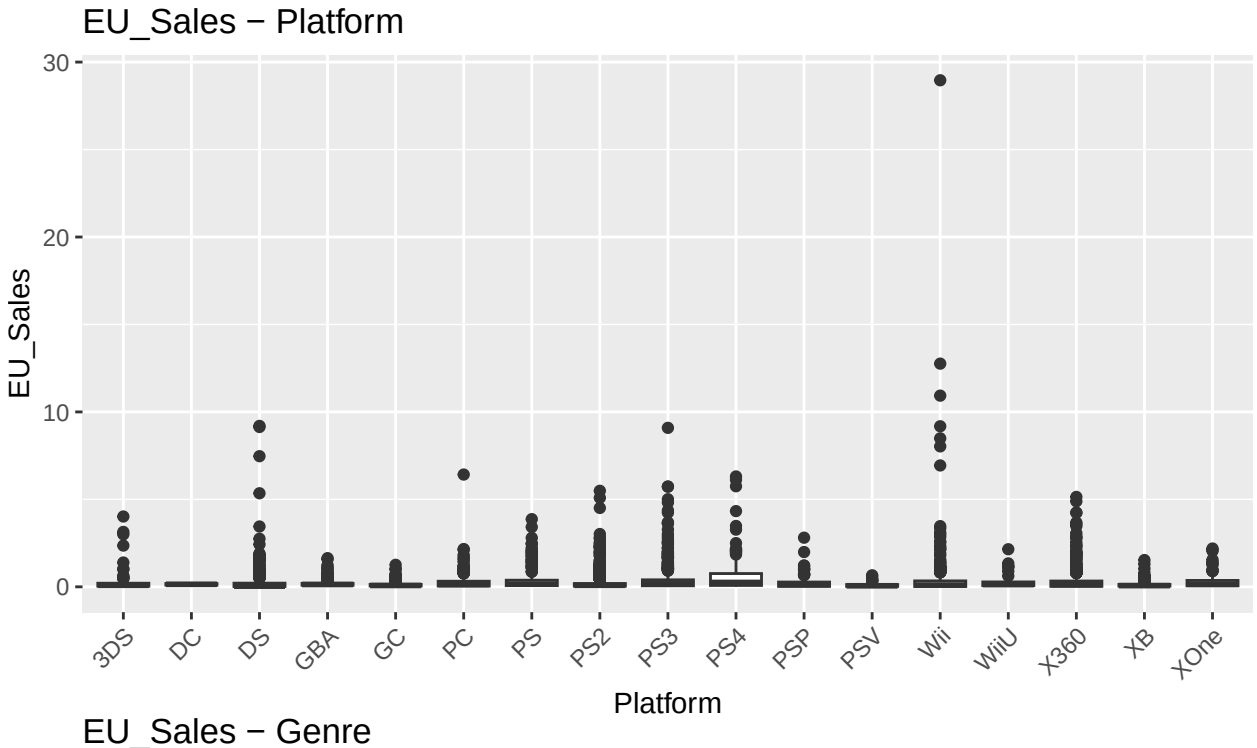
```
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

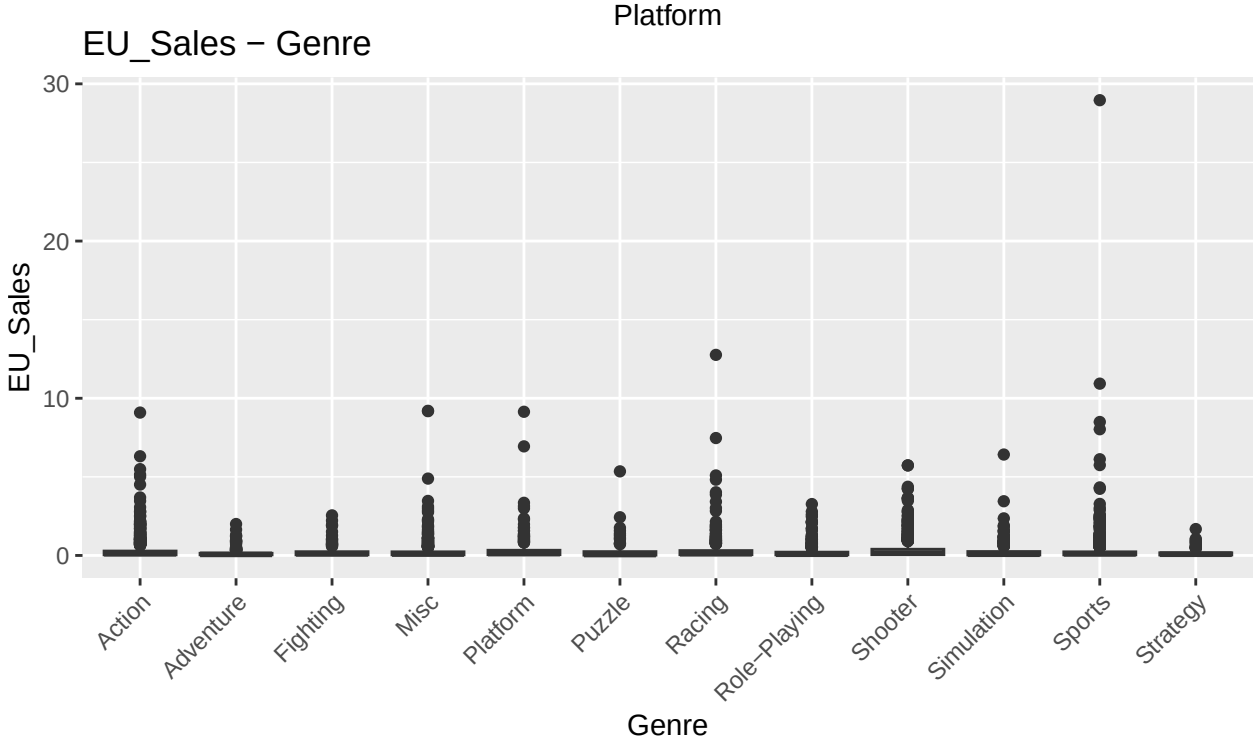


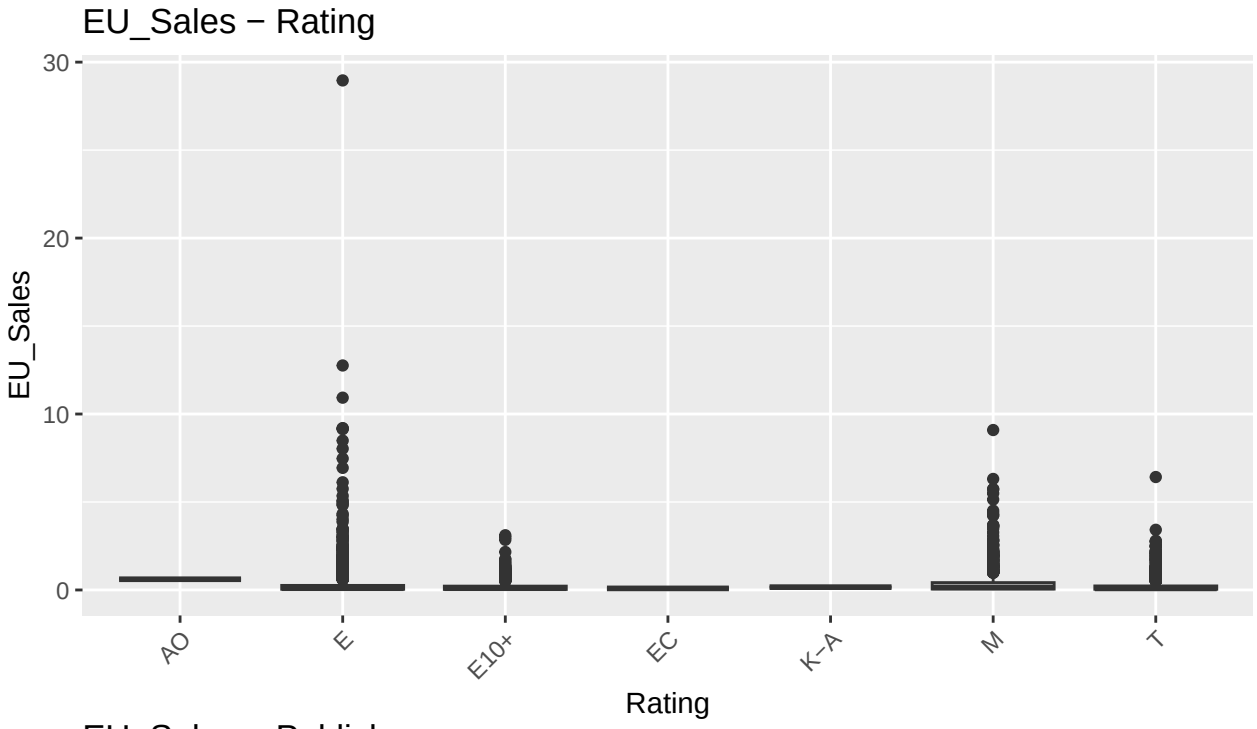


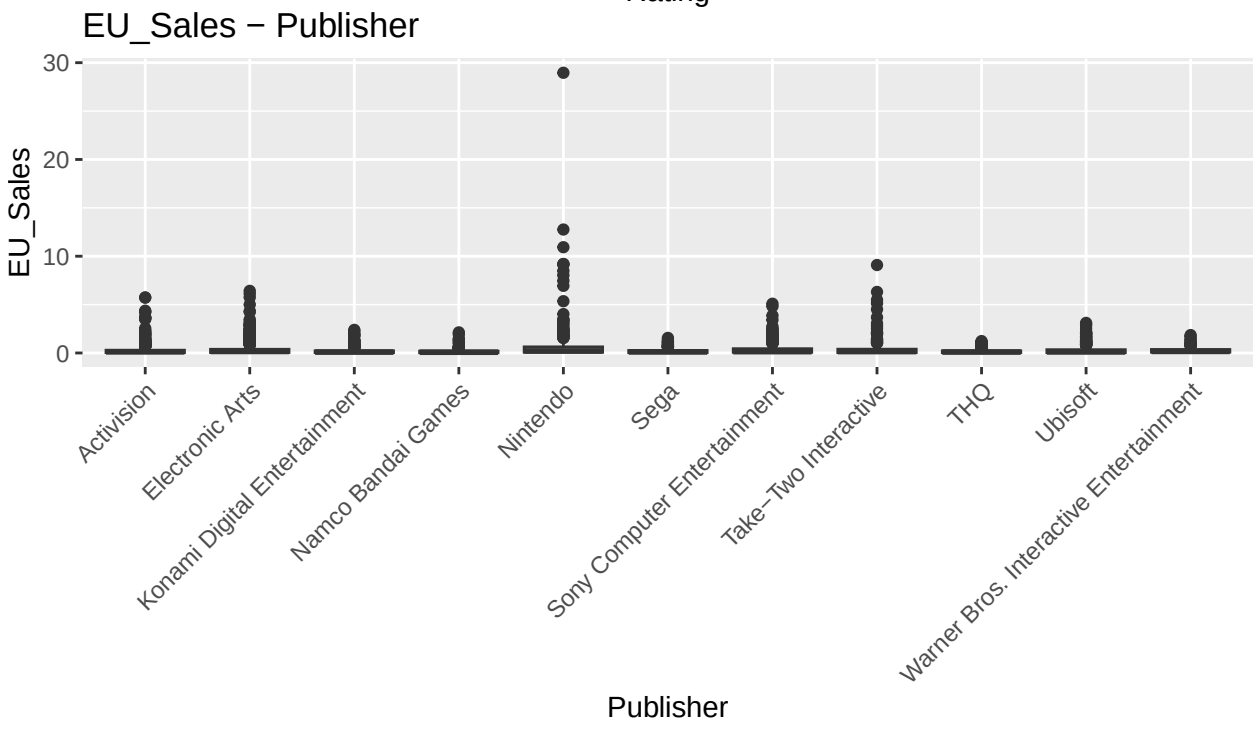

NA_Sales − Rating
NA_Sales
40
30
20
10
0
AO
E
E10+
EC
K−A
M
T
Rating

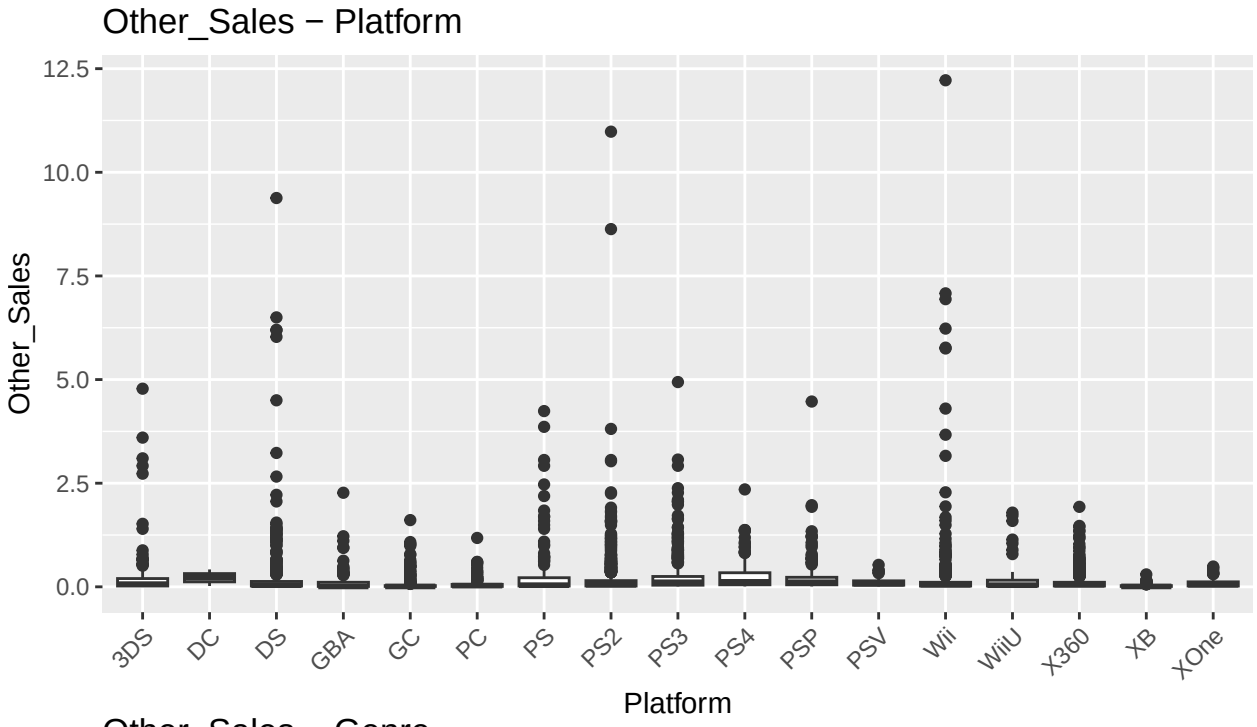

NA_Sales − Publisher
NA_Sales
40
30
20
10
0
Activision
Electronic Arts
Konami Digital Entertainment
Namco Bandai Games
Nintendo
Sega
Sony Computer Entertainment
Take−Two Interactive
THQ
Ubisoft
Warner Bros. Interactive Entertainment
Publisher

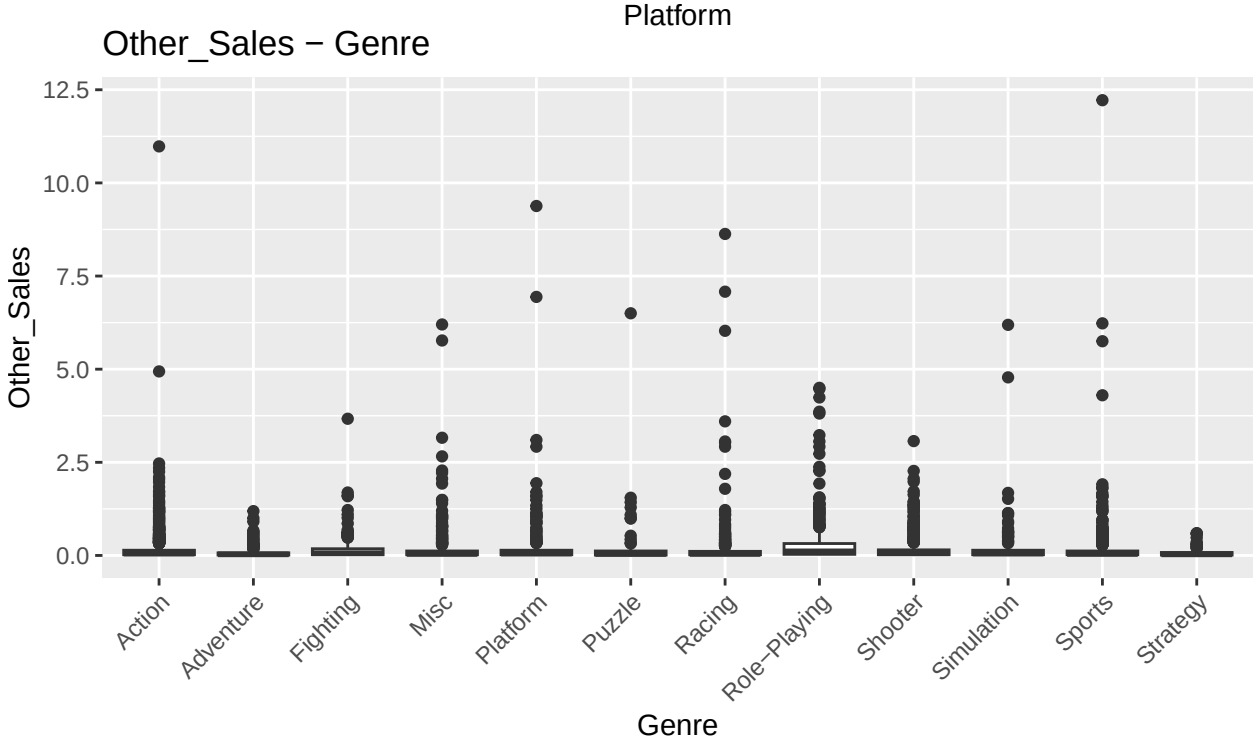

EU_Sales − Platform
EU_Sales
30
20
10
0
3DS
DC
DS
GBA
GC
PC
PS
PS2
PS3
PS4
PSP
PSV
Wii
WiiU
X360
XB
XOne
Platform

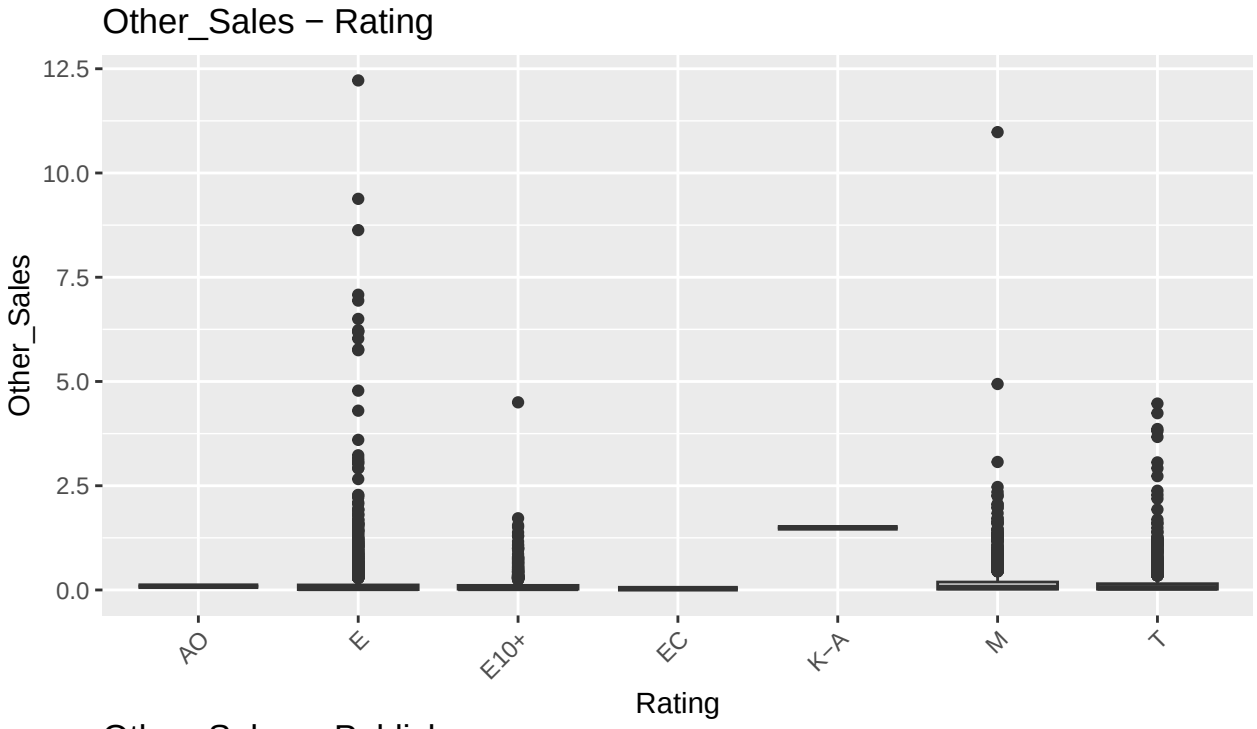

EU_Sales − Genre
EU_Sales
30
20
10
0
Action
Adventure
Fighting
Misc
Platform
Puzzle
Racing
Role−Playing
Shooter
Simulation
Sports
Strategy
Genre

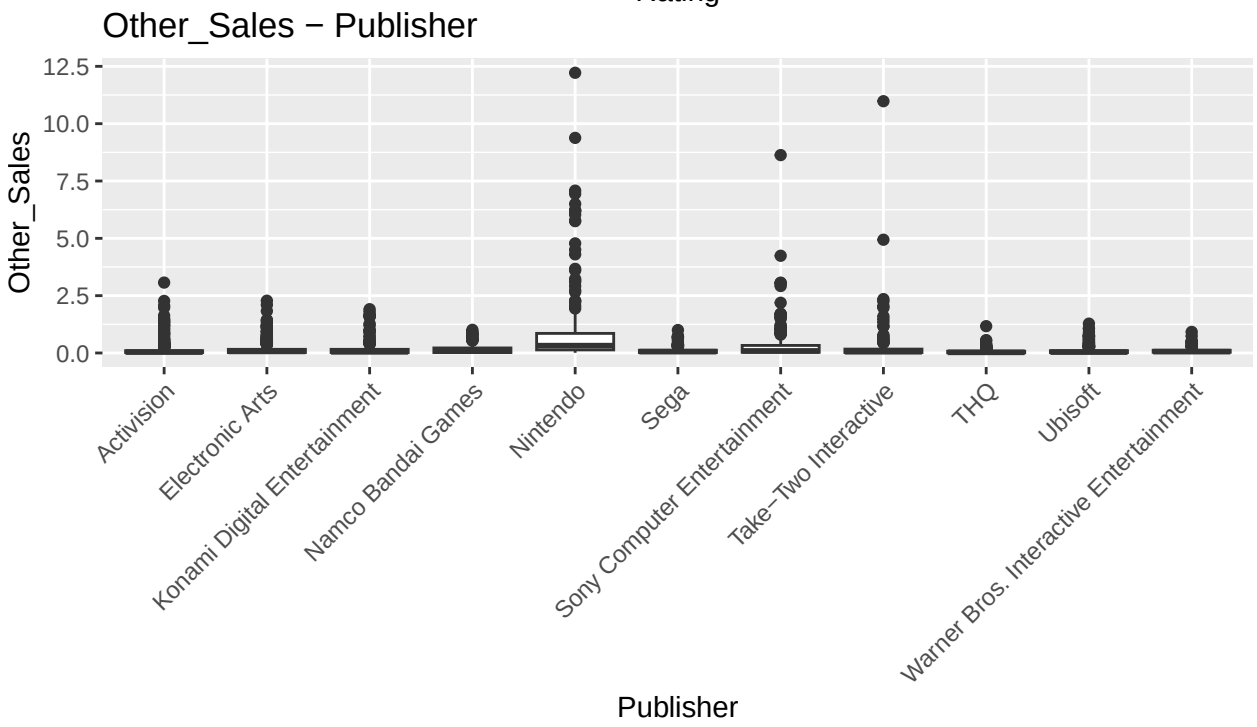

EU_Sales − Rating
EU_Sales
30
20
10
0
AO
E
E10+
EC
K−A
M
T
Rating

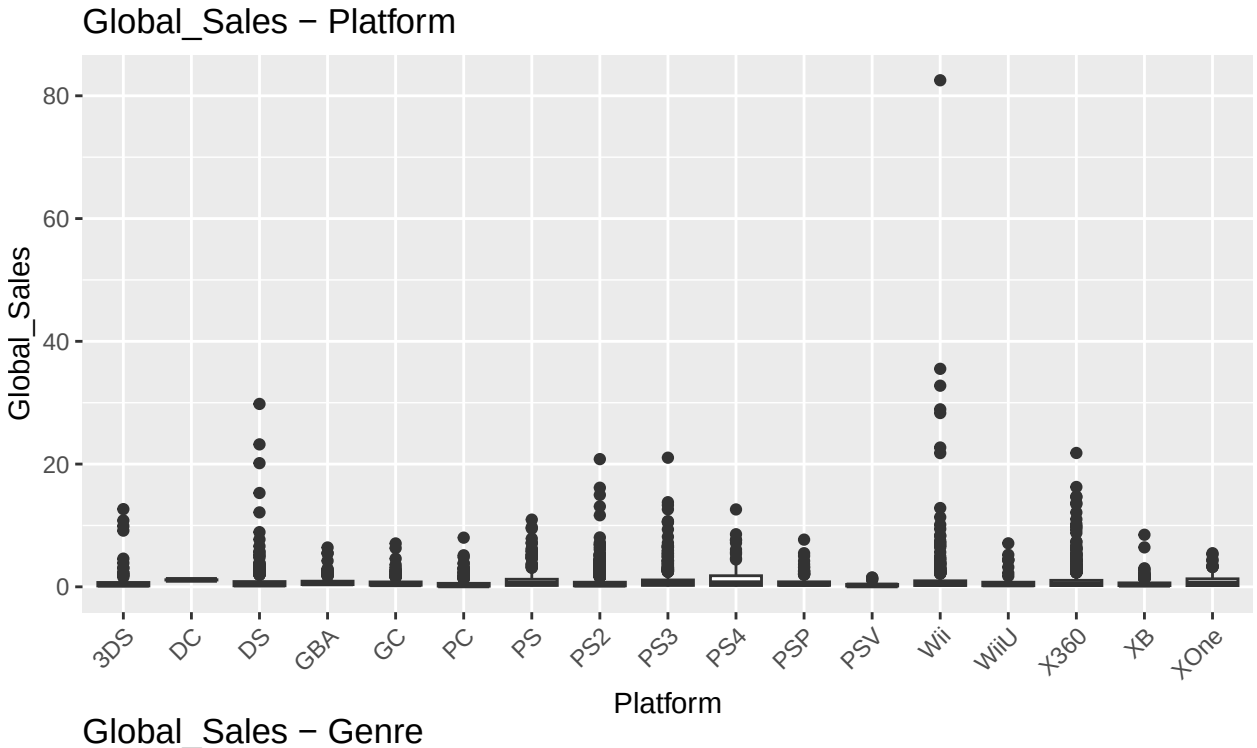

EU_Sales − Publisher
EU_Sales
30
20
10
0
Activision
Electronic Arts
Konami Digital Entertainment
Namco Bandai Games
Nintendo
Sega
Sony Computer Entertainment
Take−Two Interactive
THQ
Ubisoft
Warner Bros. Interactive Entertainment
Publisher

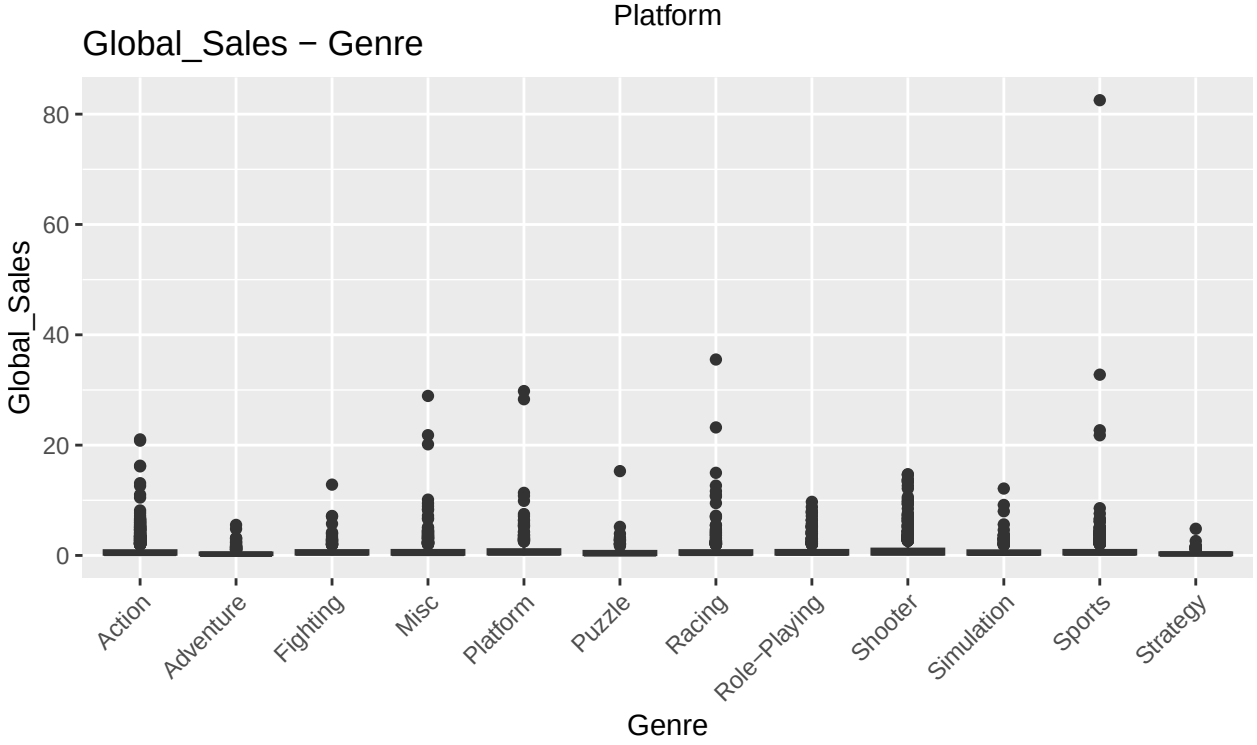

Other_Sales − Platform
Other_Sales
12.5
10.0
7.5
5.0
2.5
0.0
3DS
DC
DS
GBA
GC
PC
PS
PS2
PS3
PS4
PSP
PSV
Wii
WiiU
X360
XB
XOne
Platform

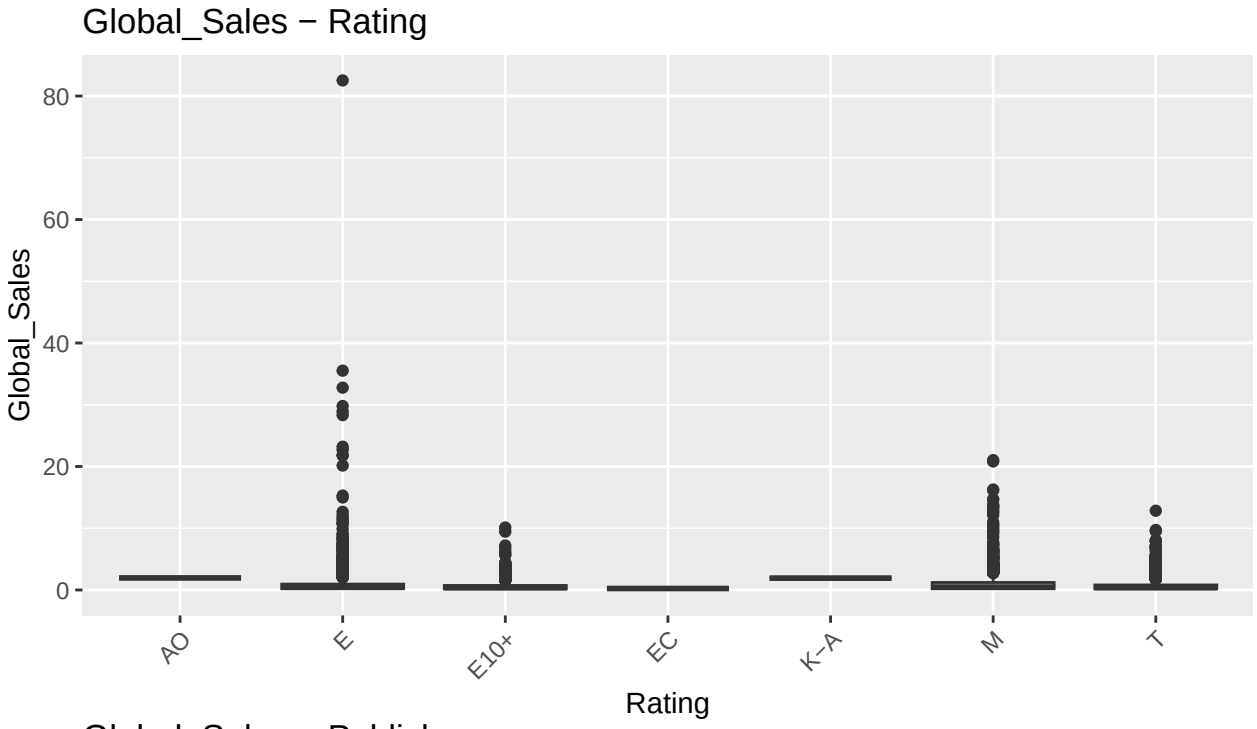

Other_Sales − Genre
Other_Sales
12.5
10.0
7.5
5.0
2.5
0.0
Action
Adventure
Fighting
Misc
Platform
Puzzle
Racing
Role−Playing
Shooter
Simulation
Sports
Strategy
Genre

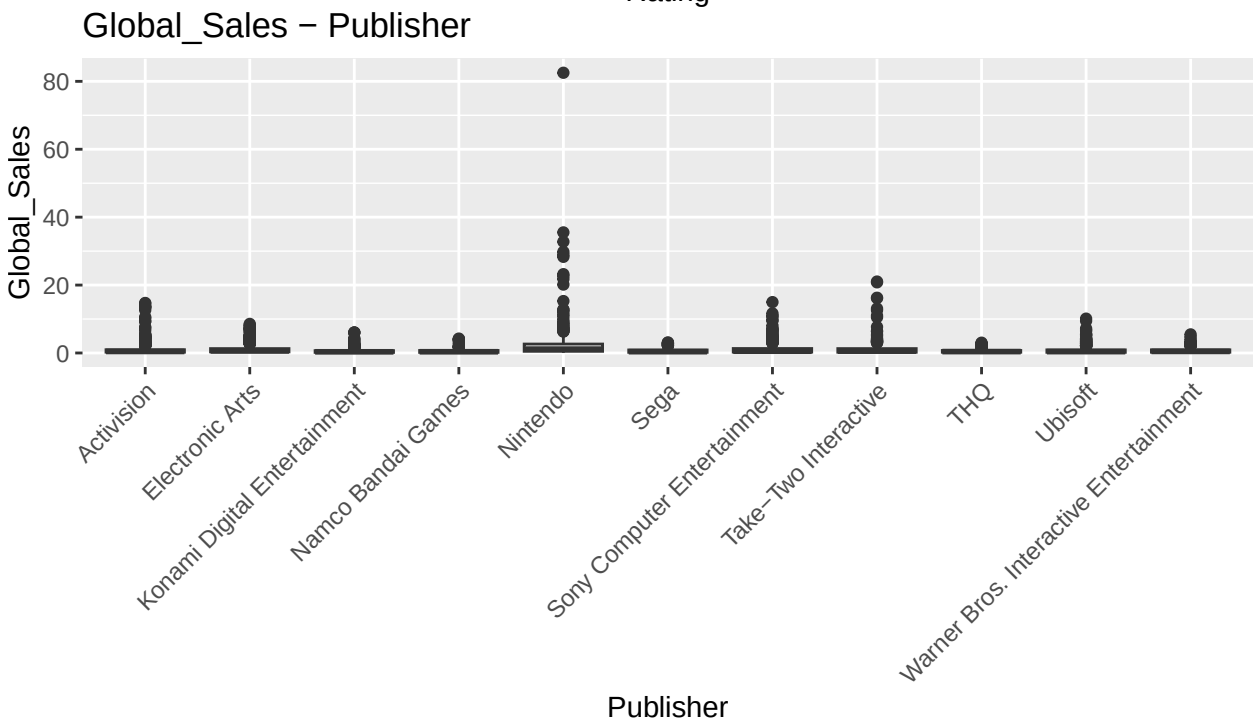

Other_Sales − Rating
Other_Sales
12.5
10.0
7.5
5.0
2.5
0.0
AO
E
E10+
EC
K−A
M
T
Rating

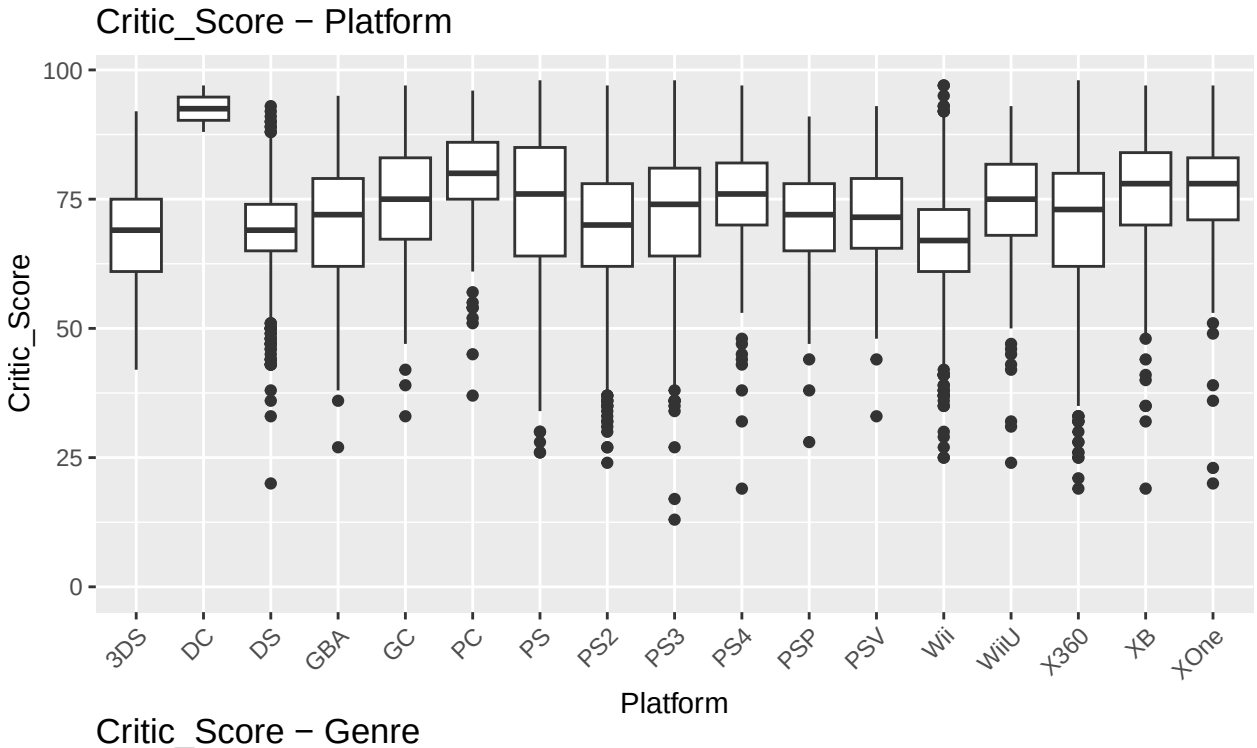

Other_Sales − Publisher
Other_Sales
12.5
10.0
7.5
5.0
2.5
0.0
Activision
Electronic Arts
Konami Digital Entertainment
Namco Bandai Games
Nintendo
Sega
Sony Computer Entertainment
Take−Two Interactive
THQ
Ubisoft
Warner Bros. Interactive Entertainment
Publisher

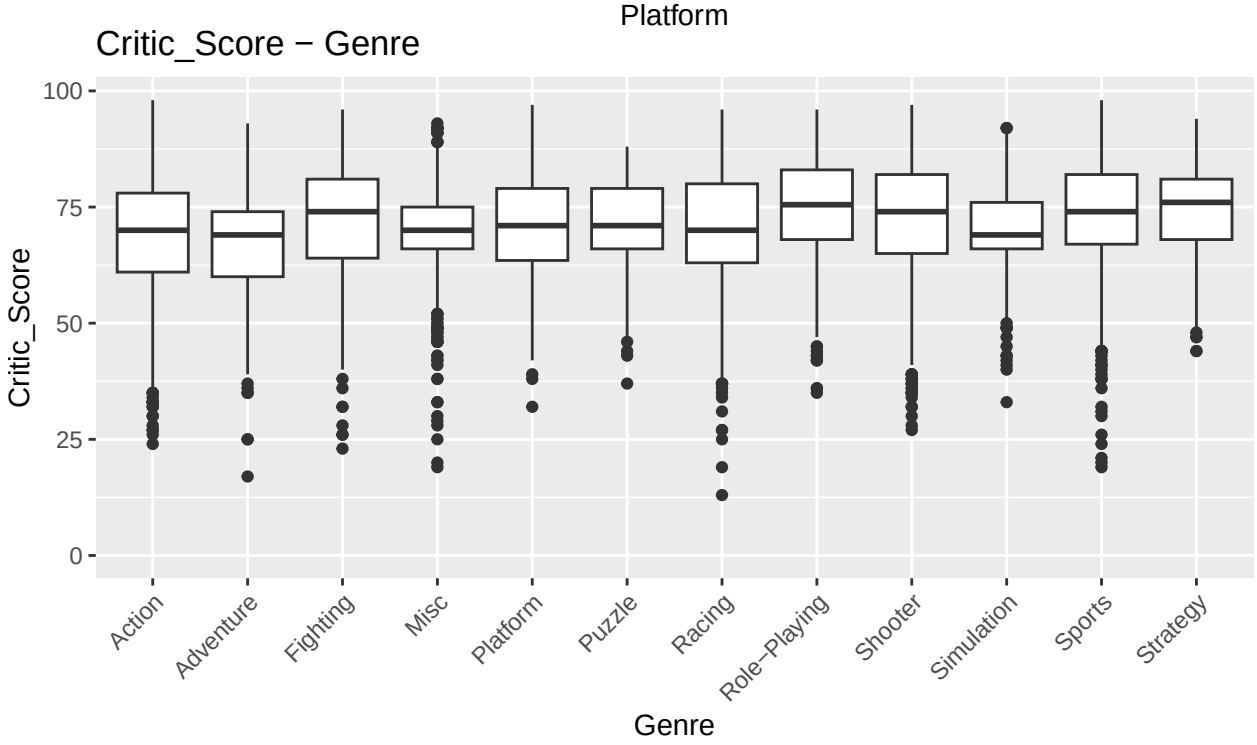

Global_Sales − Platform
Global_Sales
0
20
40
60
80
3DS
DC
DS
GBA
GC
PC
PS
PS2
PS3
PS4
PSP
PSV
Wii
WiiU
X360
XB
XOne
Platform

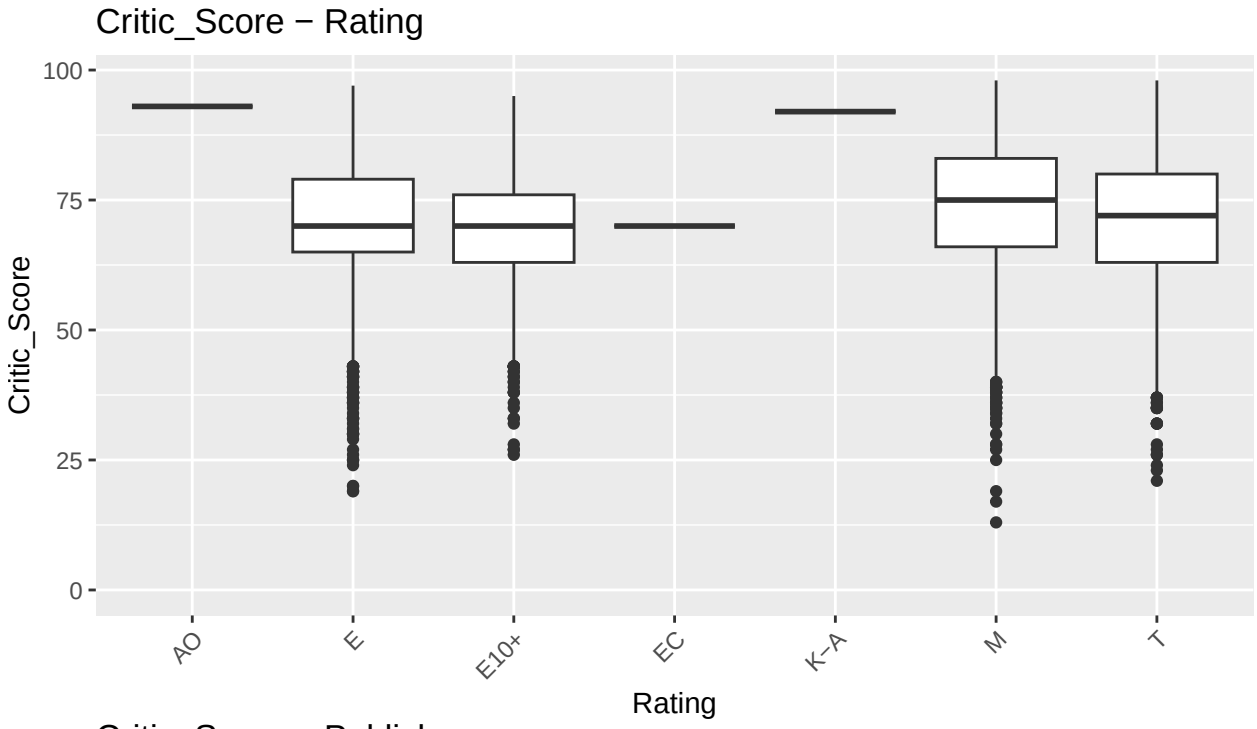

Global_Sales − Genre
Global_Sales
0
20
40
60
80
Action
Adventure
Fighting
Misc
Platform
Puzzle
Racing
Role−Playing
Shooter
Simulation
Sports
Strategy
Genre

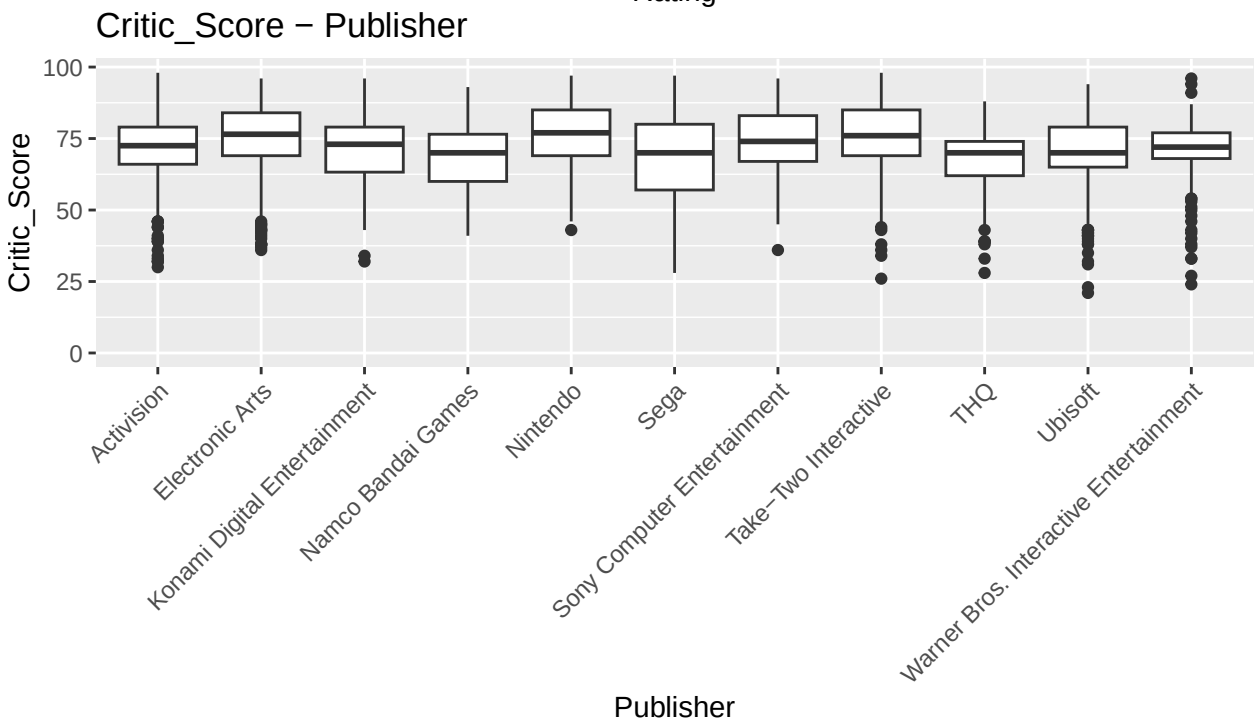

Global_Sales − Rating
Global_Sales
80
60
40
20
0
AO
E
E10+
EC
K−A
M
T
Rating

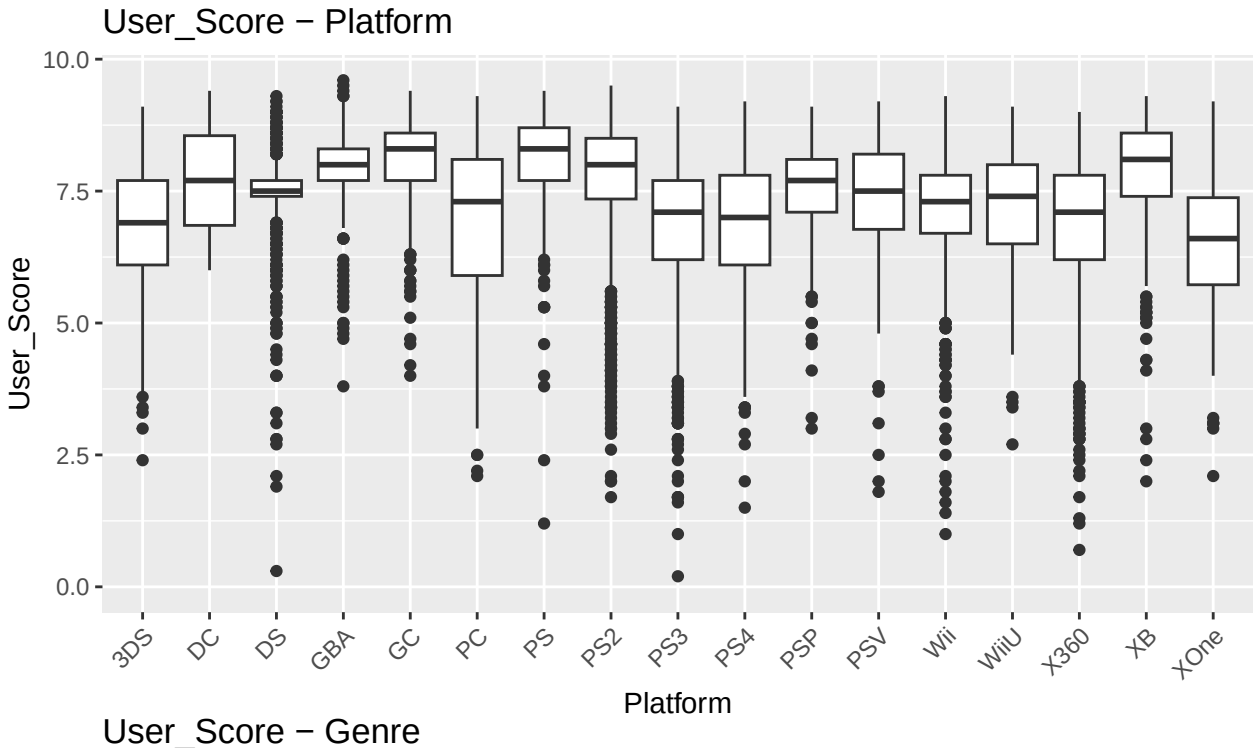

Global_Sales − Publisher
Global_Sales
80
60
40
20
0
Activision
Electronic Arts
Konami Digital Entertainment
Namco Bandai Games
Nintendo
Sega
Sony Computer Entertainment
Take−Two Interactive
THQ
Ubisoft
Warner Bros. Interactive Entertainment
Publisher

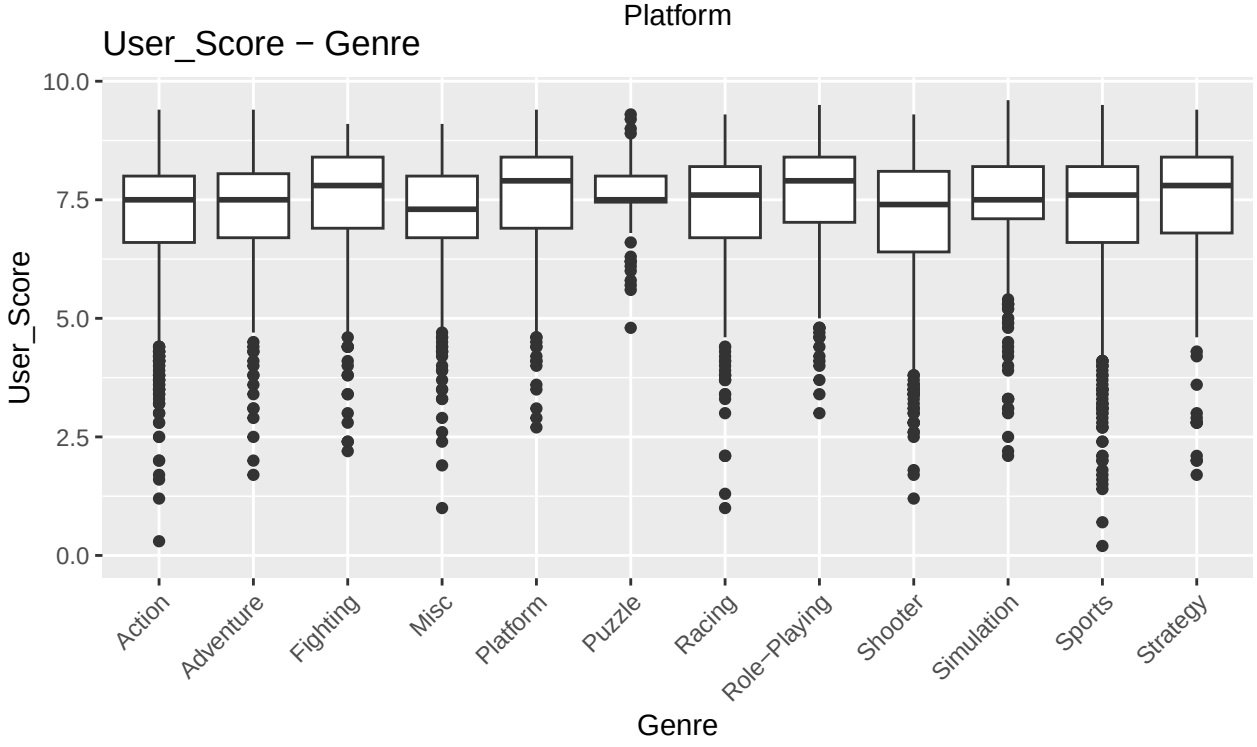

Critic_Score − Platform
Critic_Score
100
75
50
25
0
3DS
DC
DS
GBA
GC
PC
PS
PS2
PS3
PS4
PSP
PSV
Wii
WiiU
X360
XB
XOne
Platform

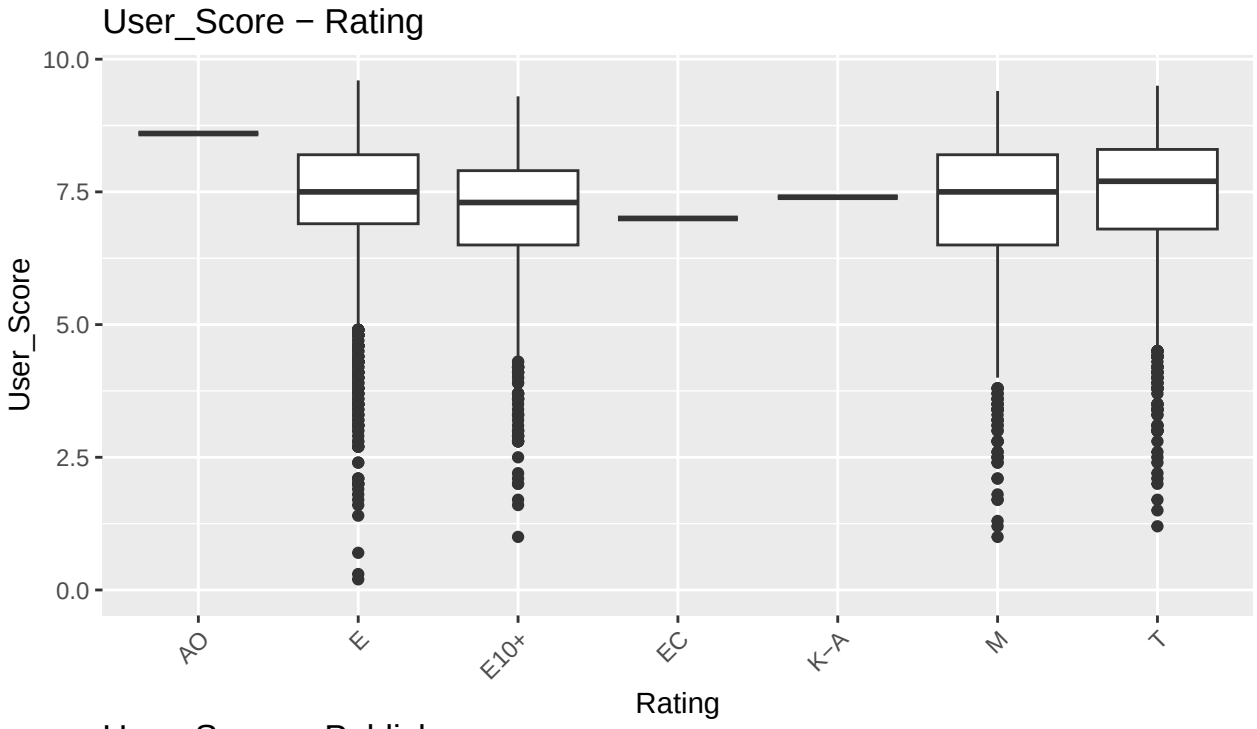

Critic_Score − Genre
Critic_Score
100
75
50
25
0
Action
Adventure
Fighting
Misc
Platform
Puzzle
Racing
Role−Playing
Shooter
Simulation
Sports
Strategy
Genre

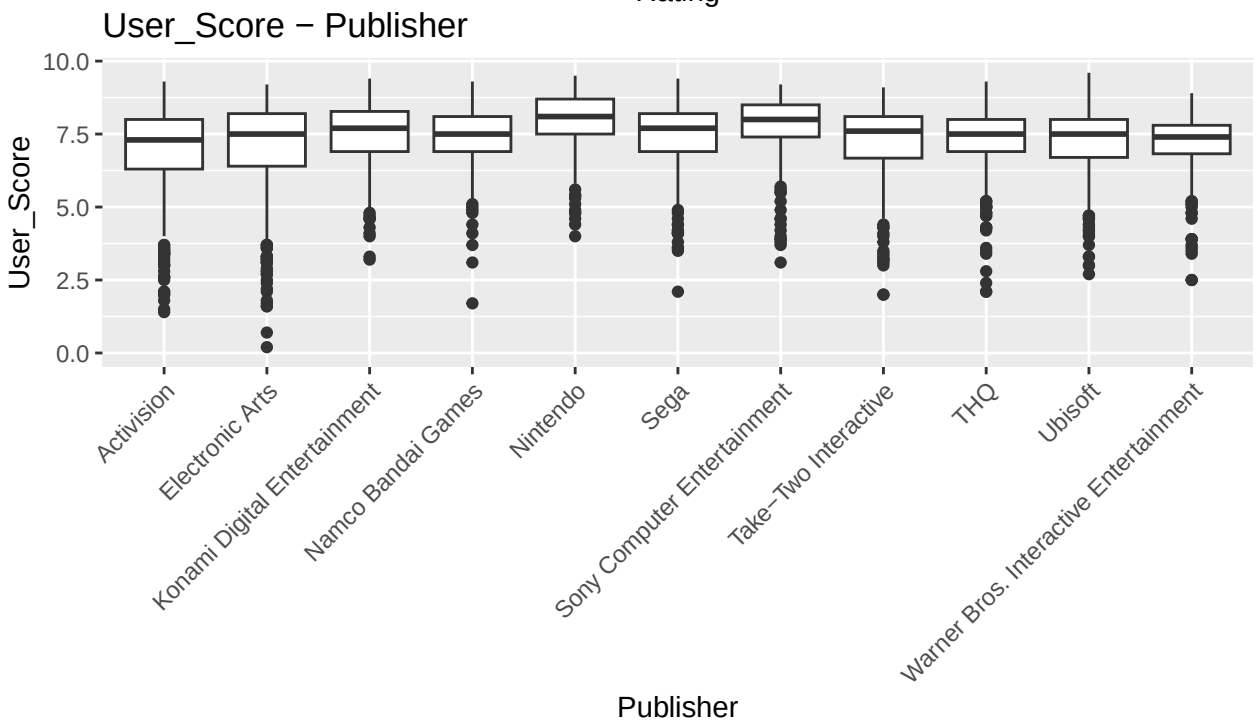

Critic_Score − Rating
Critic_Score
100
75
50
25
0
AO
E
E10+
EC
K−A
M
T
Rating


Critic_Score − Publisher
Critic_Score
100
75
50
25
0
Activision
Electronic Arts
Konami Digital Entertainment
Namco Bandai Games
Nintendo
Sega
Sony Computer Entertainment
Take−Two Interactive
THQ
Ubisoft
Warner Bros. Interactive Entertainment
Publisher


User_Score − Platform
User_Score
10.0
7.5
5.0
2.5
0.0
3DS
DC
DS
GBA
GC
PC
PS
PS2
PS3
PS4
PSP
PSV
Wii
WiiU
X360
XB
XOne
Platform


User_Score − Genre
User_Score
10.0
7.5
5.0
2.5
0.0
Action
Adventure
Fighting
Misc
Platform
Puzzle
Racing
Role−Playing
Shooter
Simulation
Sports
Strategy
Genre





The boxplots indicate presence of numerous outliers. We also see that the Rating column requires further investigation:

```
# Checking distribution of Rating column
print(table(df$Rating))
```

```
##
##    AO     E E10+    EC  K-A     M     T
```

```
##    1 2149  901    1    1 1201 1876
```

```r
# Dropping rows where Rating is AO, K-A, or EC as these
# have only one record each and it is not possible to
# perform any kind of analysis
df <- df[!(df$Rating %in% c("AO", "K-A", "EC")), ]
print(nrow(df))
```

```
## [1] 6127
```

2.5 Histograms with the appropriate number of bins and vertical lines

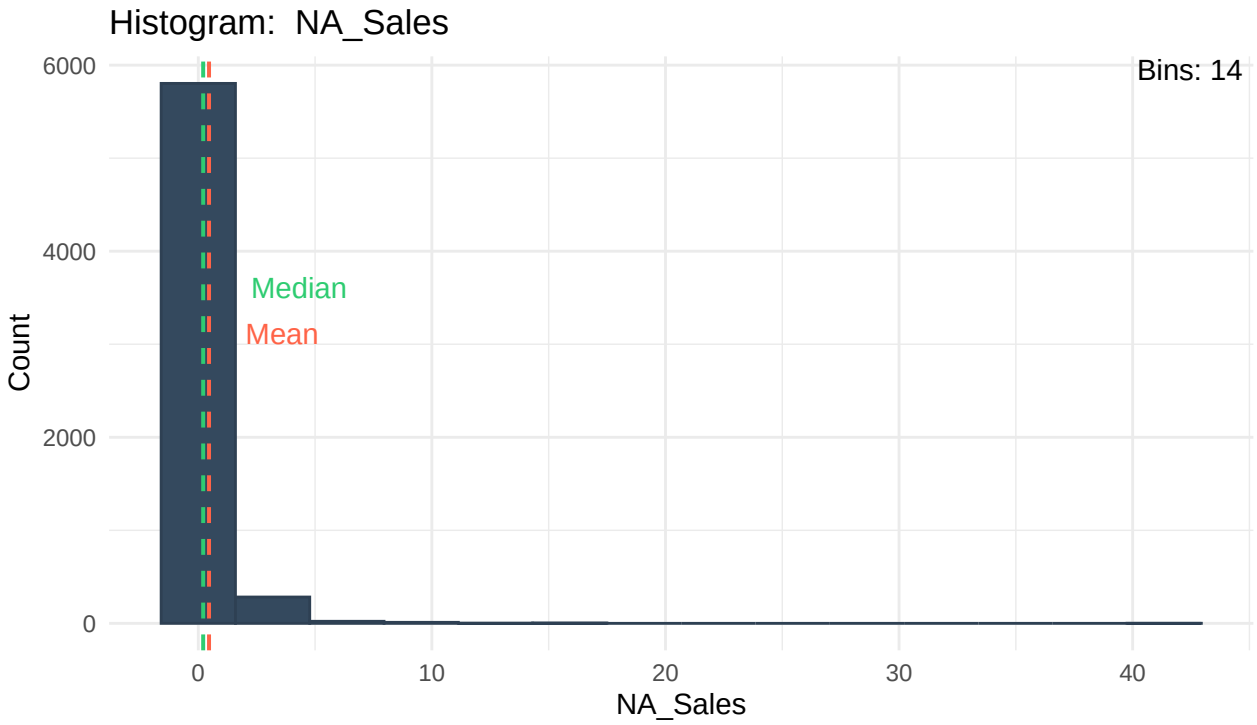
Histograms:

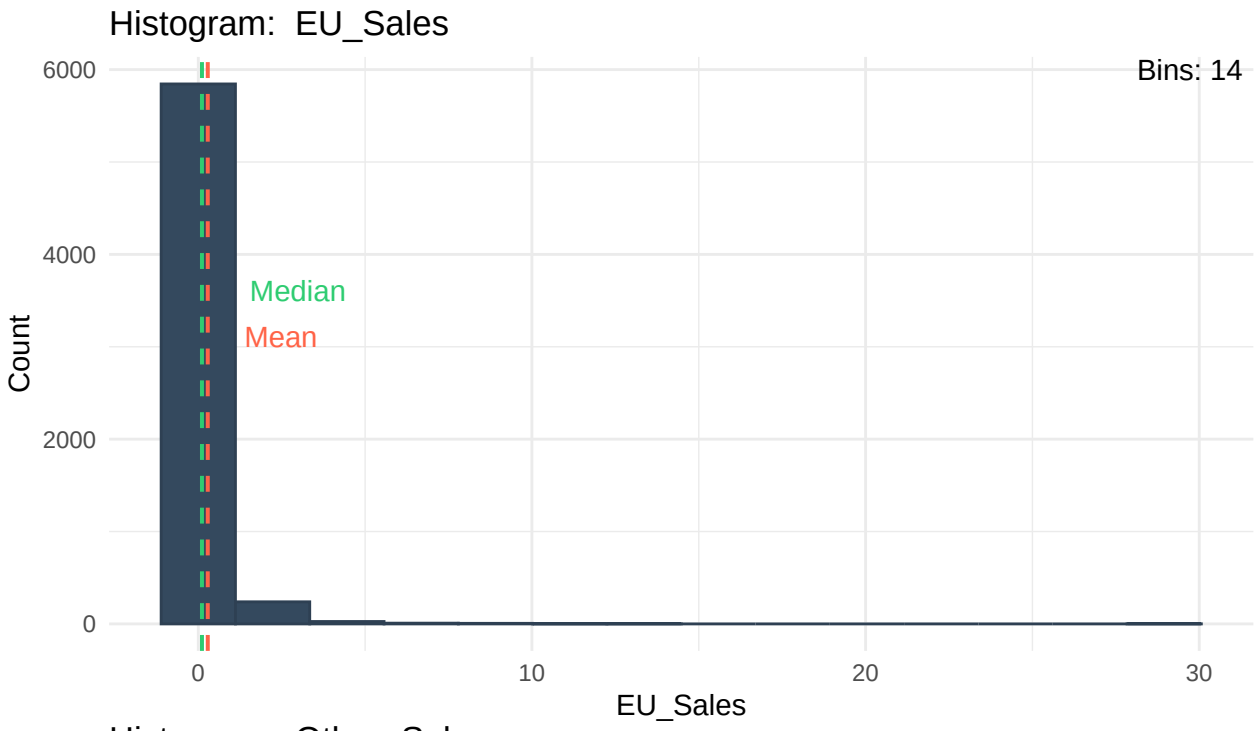
Since the boxplots indicate presence of outliers, we also plot histograms to check distribution of numerical columns, the data might be severely skewed:

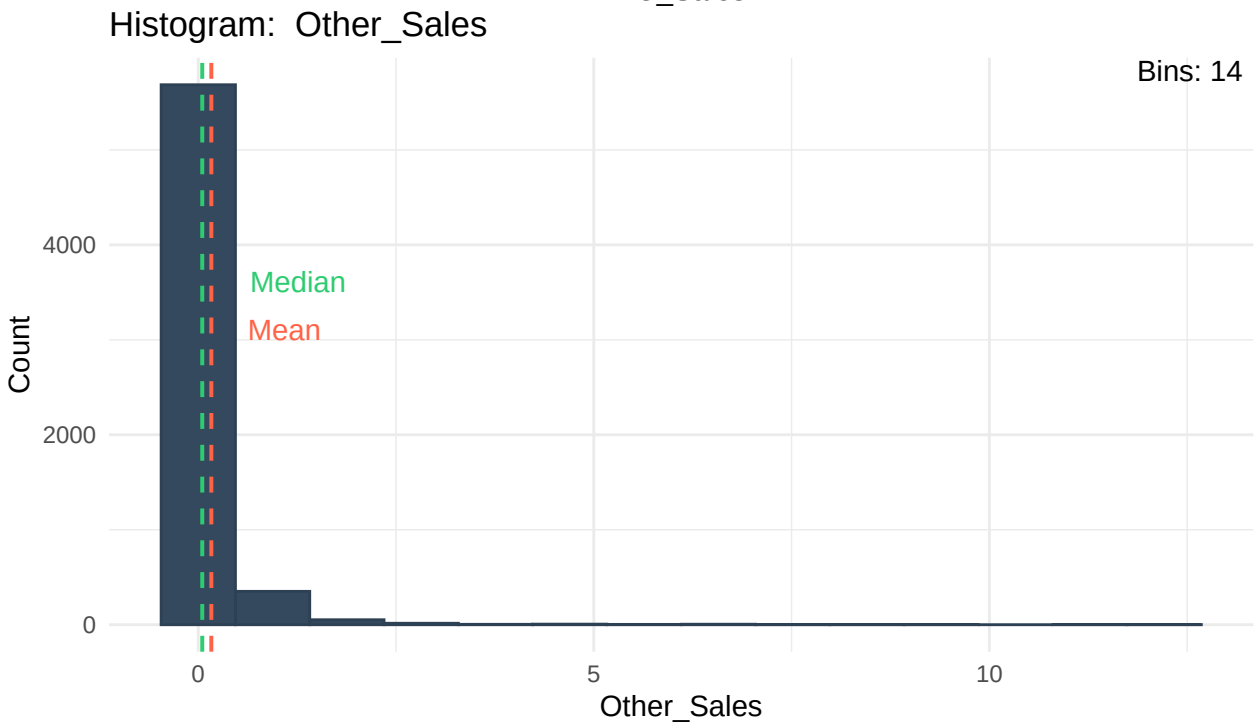
```r
numerical_columns <- c("NA_Sales", "EU_Sales", "Other_Sales",
    "Global_Sales", "Critic_Score", "User_Score")

# Function to calculate optimal bin number using Sturges'
# formula
sturges_bins <- function(n) {
    return(ceiling(log2(n) + 1))
}

# Histograms for each numerical column
for (column in numerical_columns) {
    # Calculate the optimal number of bins
    n <- nrow(df[!is.na(df[[column]]), ])
    bins <- sturges_bins(n)

    # Calculate mean and median values for the current
    # column
    mean_value <- mean(df[[column]], na.rm = TRUE)
    median_value <- median(df[[column]], na.rm = TRUE)

    # Generate the histogram with Sturges' bins
    plot <- ggplot(df, aes_string(x = column)) + geom_histogram(bins = bins,
        fill = "#34495E", color = "#2E4053") + geom_vline(xintercept = mean_value,
        color = "#FF6347", linetype = "dashed", size = 0.7) +
        geom_vline(xintercept = median_value, color = "#2ECC71",
            linetype = "dashed", size = 0.7) + labs(title = paste("Histogram: ",
```

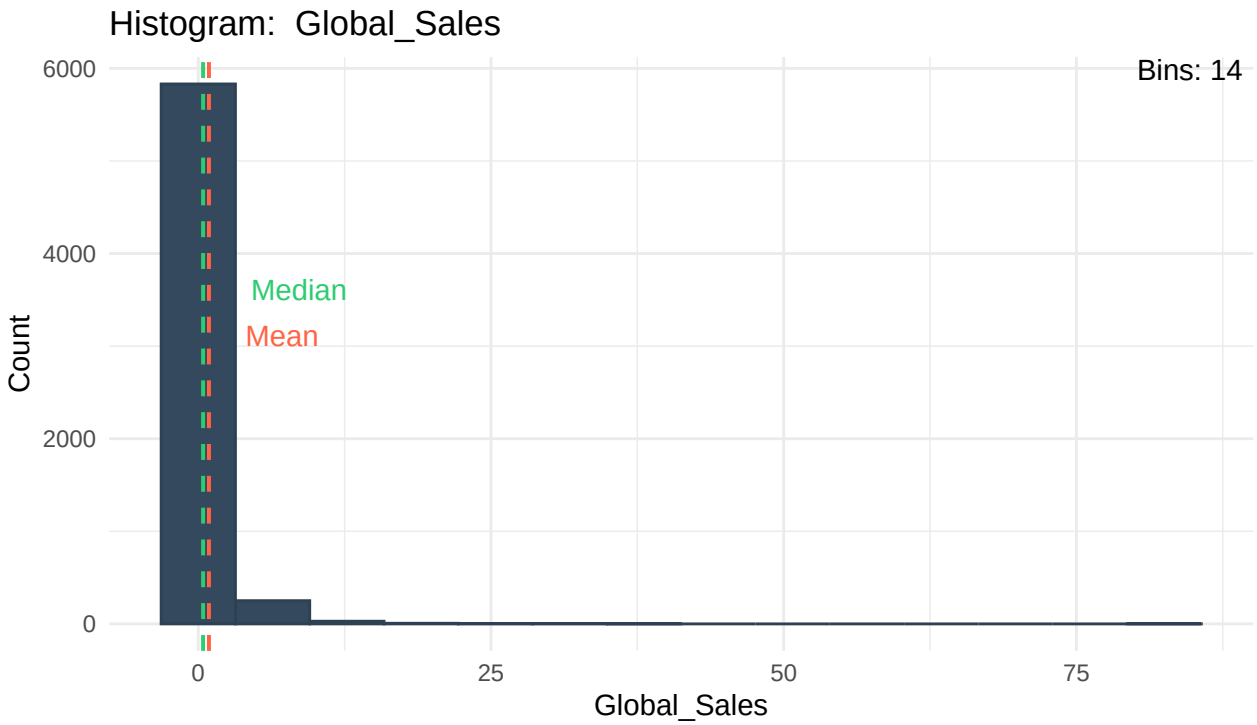
```
        column), x = column, y = "Count") + theme_minimal() +
        annotate("text", x = Inf, y = Inf, hjust = 1.1, vjust = 1.1,
            label = paste("Bins:", bins), size = 4) + annotate("text",
        x = mean_value, y = 3000, label = "Mean", hjust = -0.5,
        vjust = 0, size = 4, color = "#FF6347") + annotate("text",
        x = median_value, y = 3500, label = "Median", hjust = -0.5,
        vjust = 0, size = 4, color = "#2ECC71") + theme(plot.margin = unit(c(1,
        0, 1, 0), "cm"))
    print(plot)
}
```

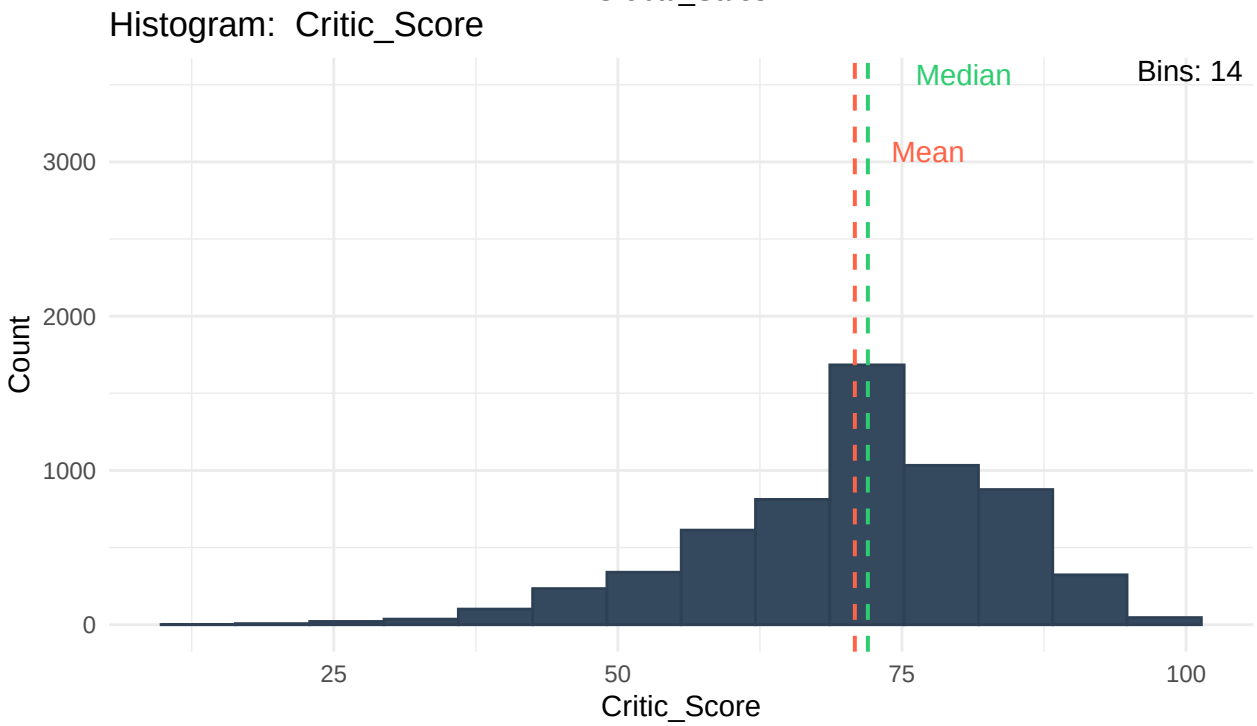
```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

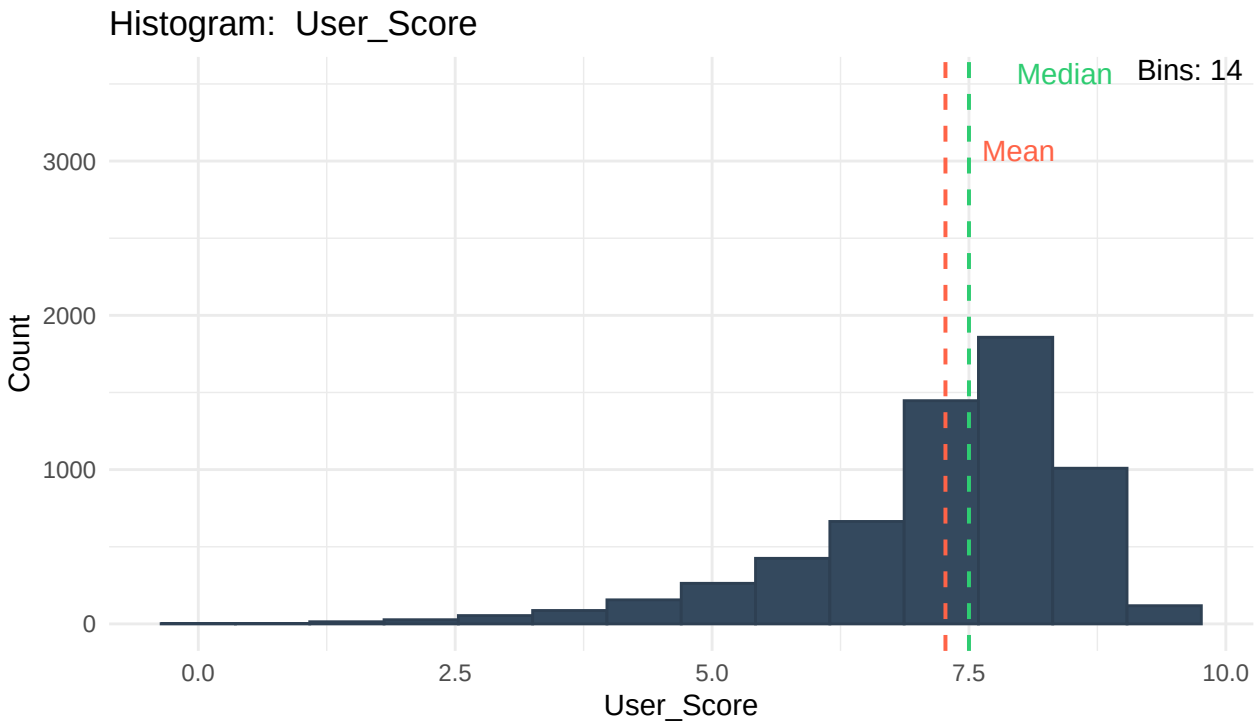



Histogram: EU_Sales
Bins: 14
6000
4000
2000
0
Count
Median
Mean
0
10
20
30
EU_Sales


Histogram: Other_Sales
Bins: 14
4000
2000
0
Count
Median
Mean
0
5
10
Other_Sales


Histogram: Global_Sales
Bins: 14
Median
Mean
Count
6000
4000
2000
0
0
25
50
75
Global_Sales


Histogram: Critic_Score
Bins: 14
Median
Mean
Count
3000
2000
1000
0
25
50
75
100
Critic_Score



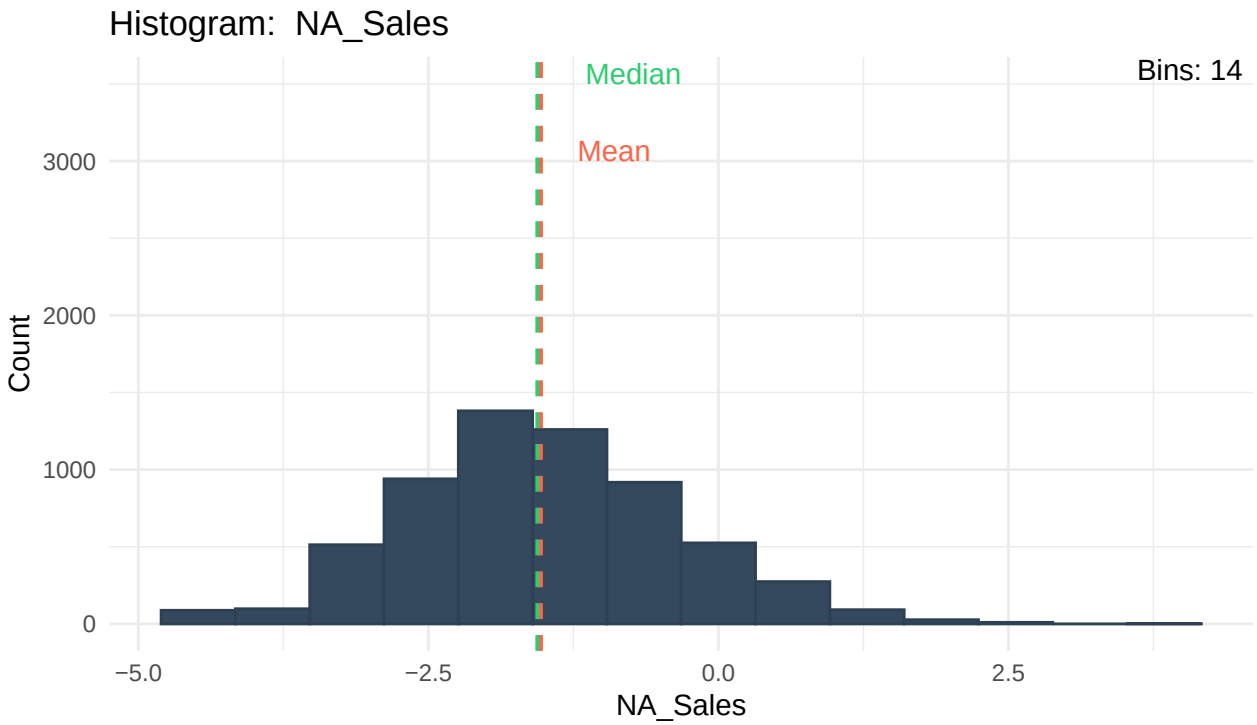
Looking at the histograms, we see that the sales data is not normally distributed. We thus transform the data by applying Box-Cox transformation.

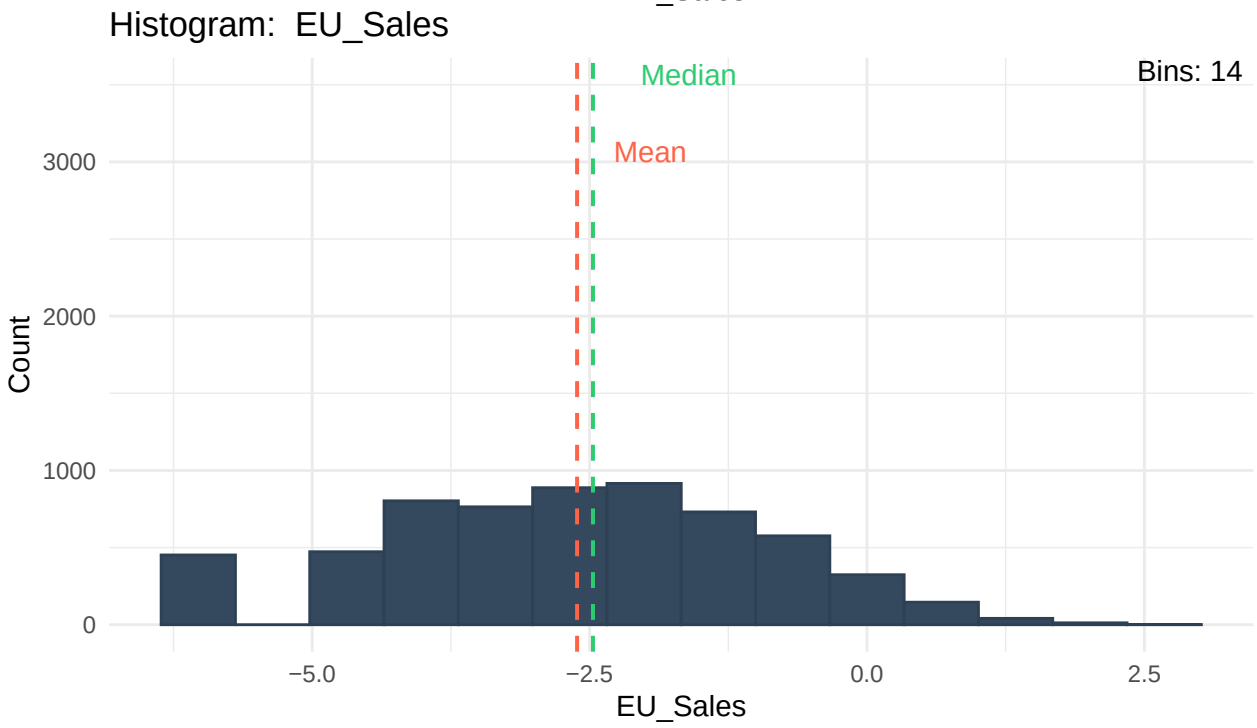
```
# Transforming data to ensure normality - Applying Box-Cox
# transformation and plotting transformed data
transform_cols <- c("NA_Sales", "EU_Sales", "Other_Sales", "Global_Sales",
    "Critic_Score", "User_Score")

library(MASS)

apply_boxcox_transformation <- function(data, columns) {
    for (column in columns) {
        col_data <- data[[column]] + 1e-05  #Adding a small constant to avoid zero or negative
        bc_result <- boxcox(col_data ~ 1, plotit = FALSE)
        optimal_lambda <- bc_result$x[which.max(bc_result$y)]
        transformed_column <- if (optimal_lambda != 0) {
            (col_data^optimal_lambda - 1)/optimal_lambda
        } else {
            log(col_data)
        }

        data[[column]] <- ifelse(is.finite(transformed_column),
            transformed_column, data[[column]])
    }
    return(data)
}
df <- apply_boxcox_transformation(df, transform_cols)
```

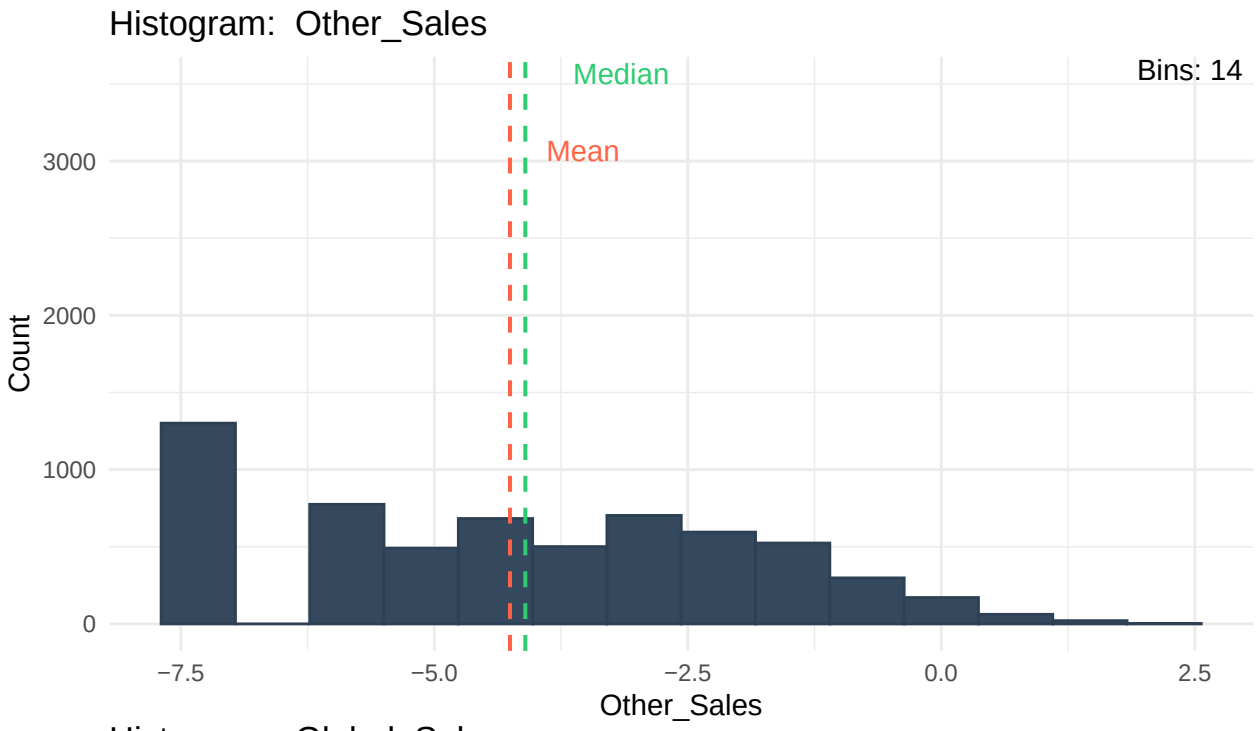
We check if the Box-Cox transformation was successful by looking at the histograms and qq-plots of the transformed data:

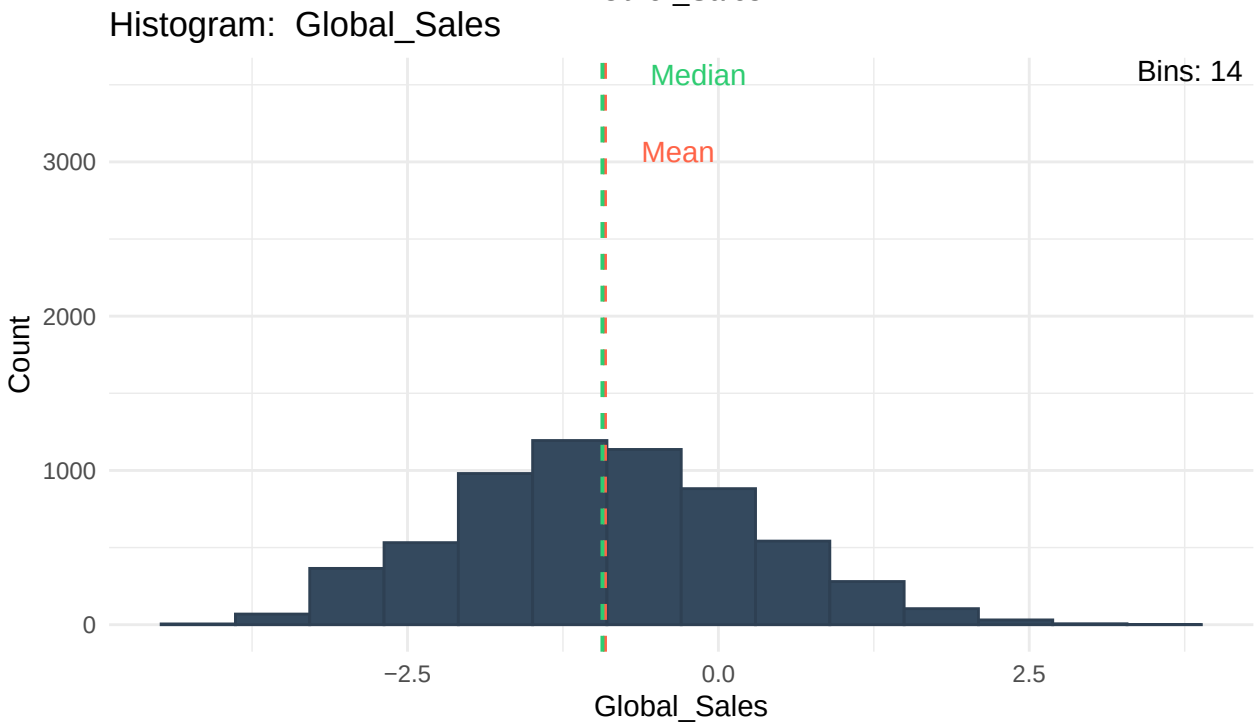
Histogram of transformed data:

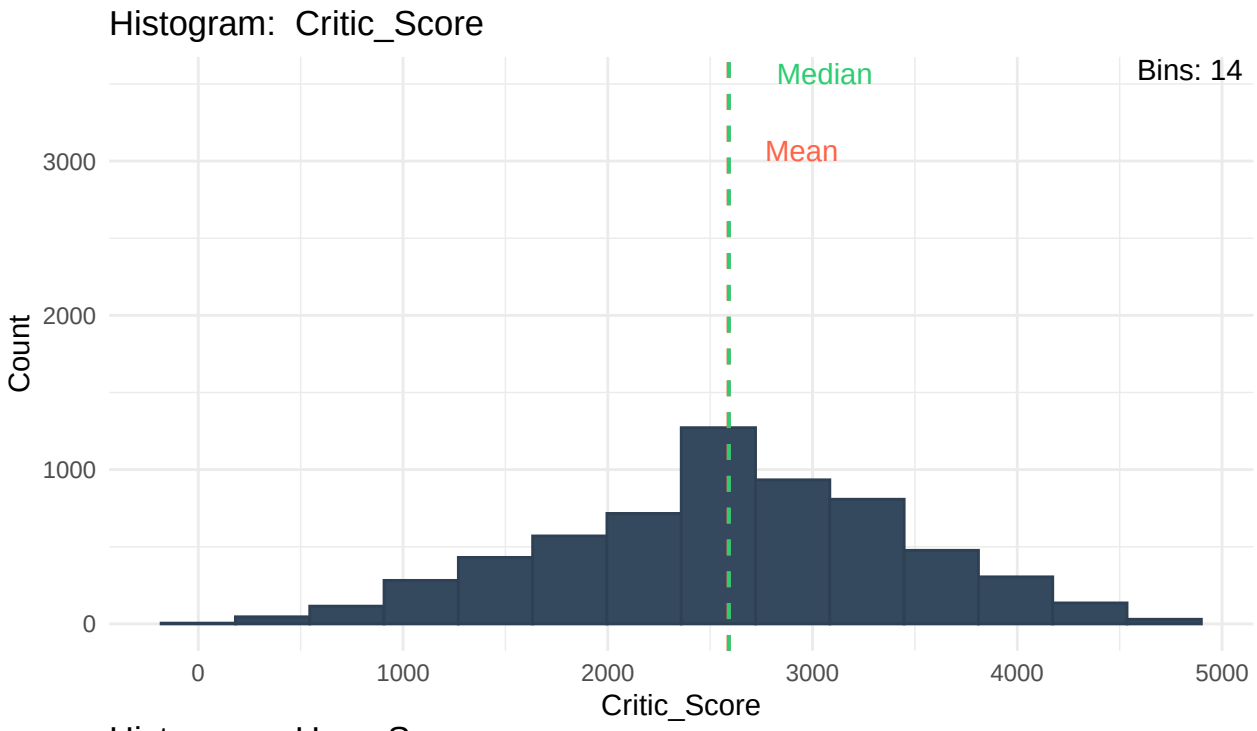
```
sturges_bins <- function(n) {
    return(ceiling(log2(n) + 1))
}

# Histograms for each numerical column
for (column in numerical_columns) {
    # Calculate the optimal number of bins
    n <- nrow(df[!is.na(df[[column]]), ])
    bins <- sturges_bins(n)

    # Calculate mean and median values for the current
    # column
    mean_value <- mean(df[[column]], na.rm = TRUE)
    median_value <- median(df[[column]], na.rm = TRUE)

    # Generate the histogram with Sturges' bins
    plot <- ggplot(df, aes_string(x = column)) + geom_histogram(bins = bins,
        fill = "#34495E", color = "#2E4053") + geom_vline(xintercept = mean_value,
        color = "#FF6347", linetype = "dashed", size = 0.7) +
        geom_vline(xintercept = median_value, color = "#2ECC71",
            linetype = "dashed", size = 0.7) + labs(title = paste("Histogram: ",
        column), x = column, y = "Count") + theme_minimal() +
        annotate("text", x = Inf, y = Inf, hjust = 1.1, vjust = 1.1,
            label = paste("Bins:", bins), size = 4) + annotate("text",
        x = mean_value, y = 3000, label = "Mean", hjust = -0.5,
        vjust = 0, size = 4, color = "#FF6347") + annotate("text",
        x = median_value, y = 3500, label = "Median", hjust = -0.5,
        vjust = 0, size = 4, color = "#2ECC71") + theme(plot.margin = unit(c(1,
        0, 1, 0), "cm"))
    print(plot)
}
```

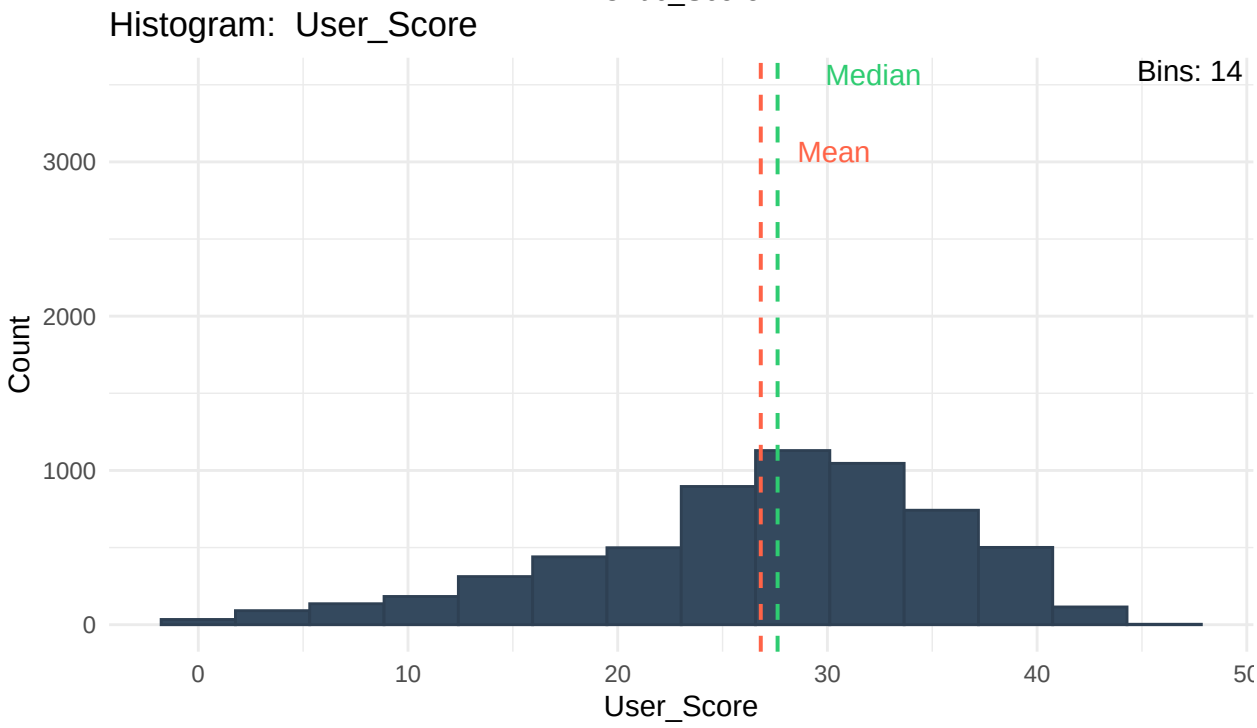

Histogram: NA_Sales
Median
Mean
Bins: 14
3000
2000
1000
0
Count
−5.0
−2.5
0.0
2.5
NA_Sales


Histogram: EU_Sales
Median
Mean
Bins: 14
3000
2000
1000
0
Count
−5.0
−2.5
0.0
2.5
EU_Sales


Histogram: Other_Sales
Median
Mean
Bins: 14
3000
2000
1000
0
Count
−7.5
−5.0
−2.5
0.0
2.5
Other_Sales


Histogram: Global_Sales
Median
Mean
Bins: 14
3000
2000
1000
0
Count
−2.5
0.0
2.5
Global_Sales

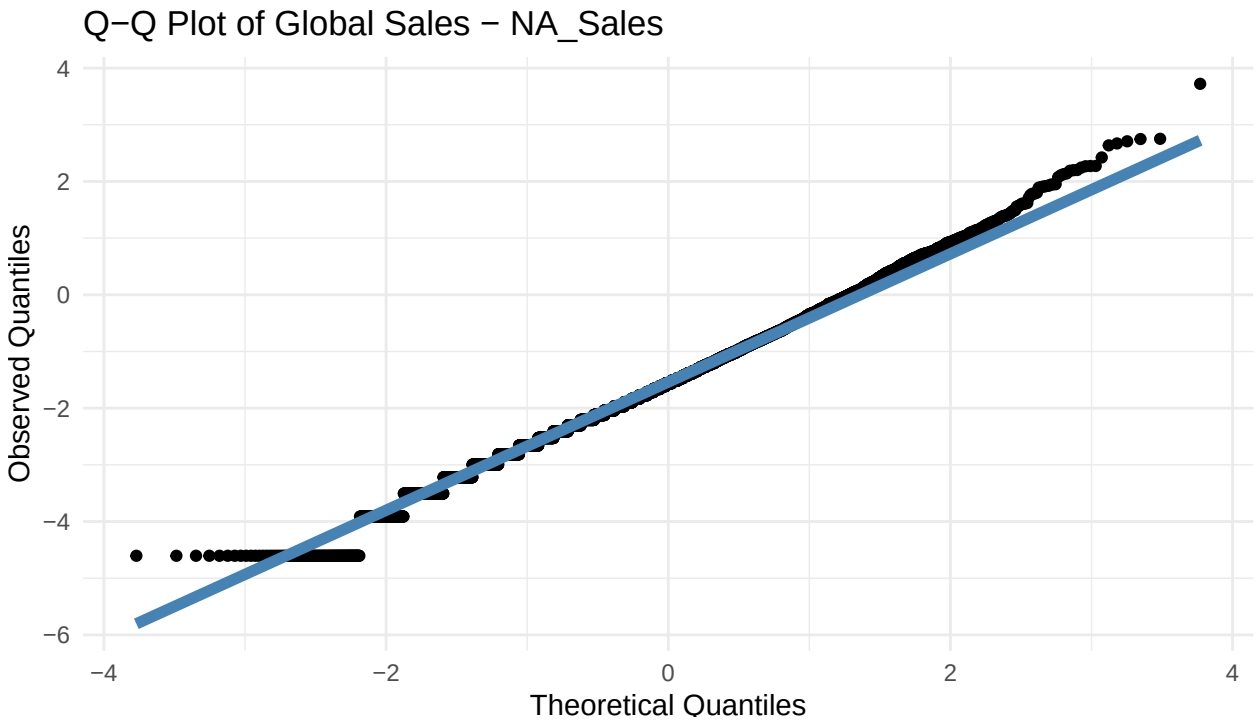

Histogram: Critic_Score
Median
Mean
Bins: 14
3000
2000
1000
0
Count
0
1000
2000
3000
4000
5000
Critic_Score

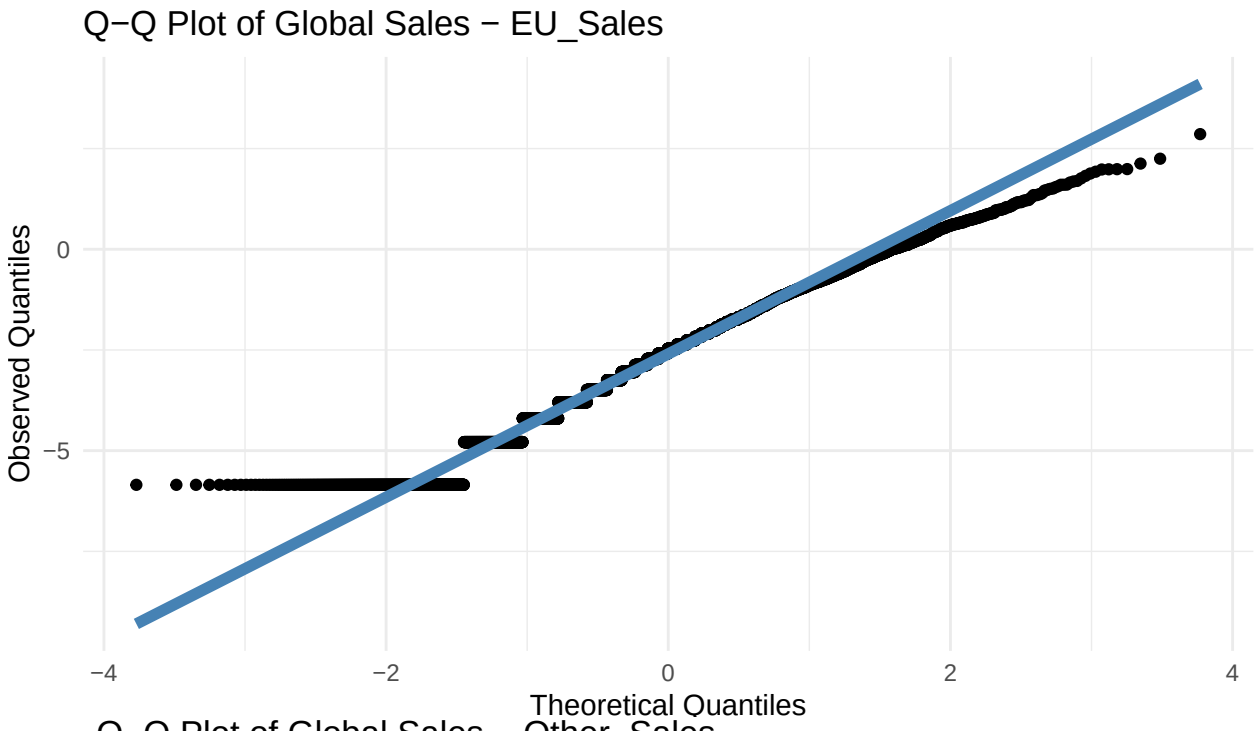

Histogram: User_Score
Median
Mean
Bins: 14
3000
2000
1000
0
Count
0
10
20
30
40
User_Score

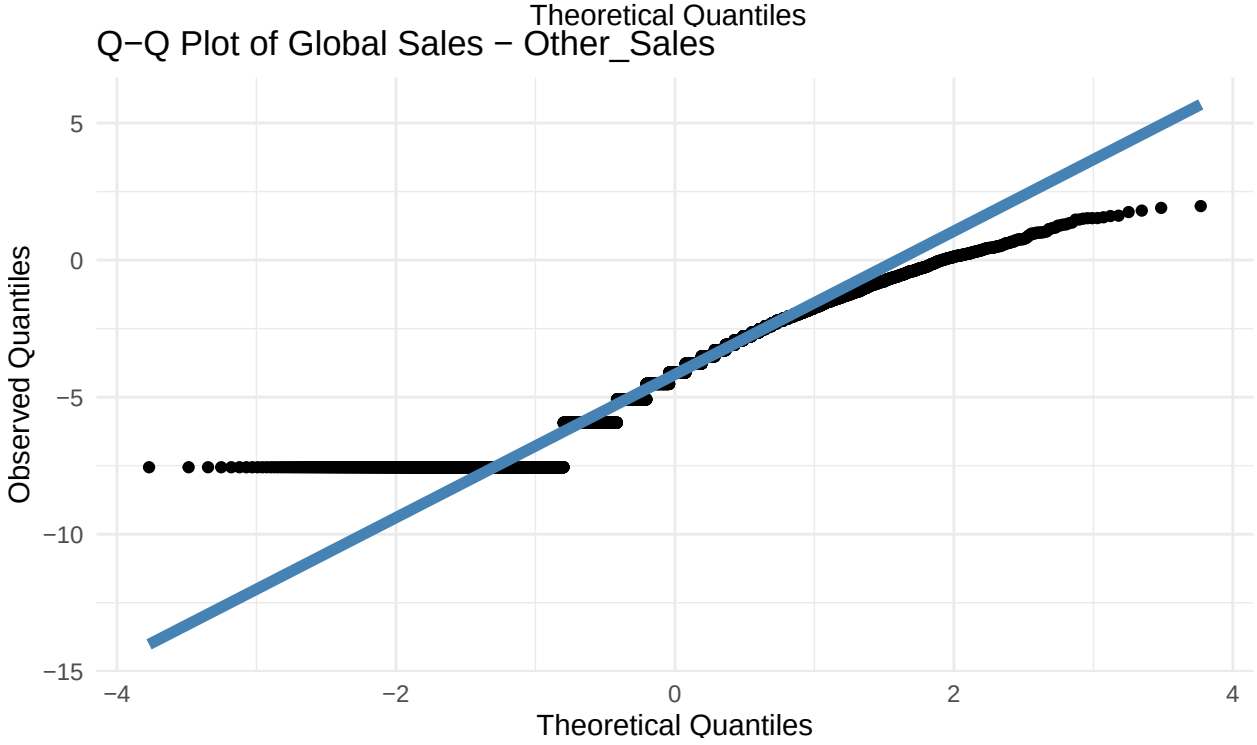
2.6 Quantile plots

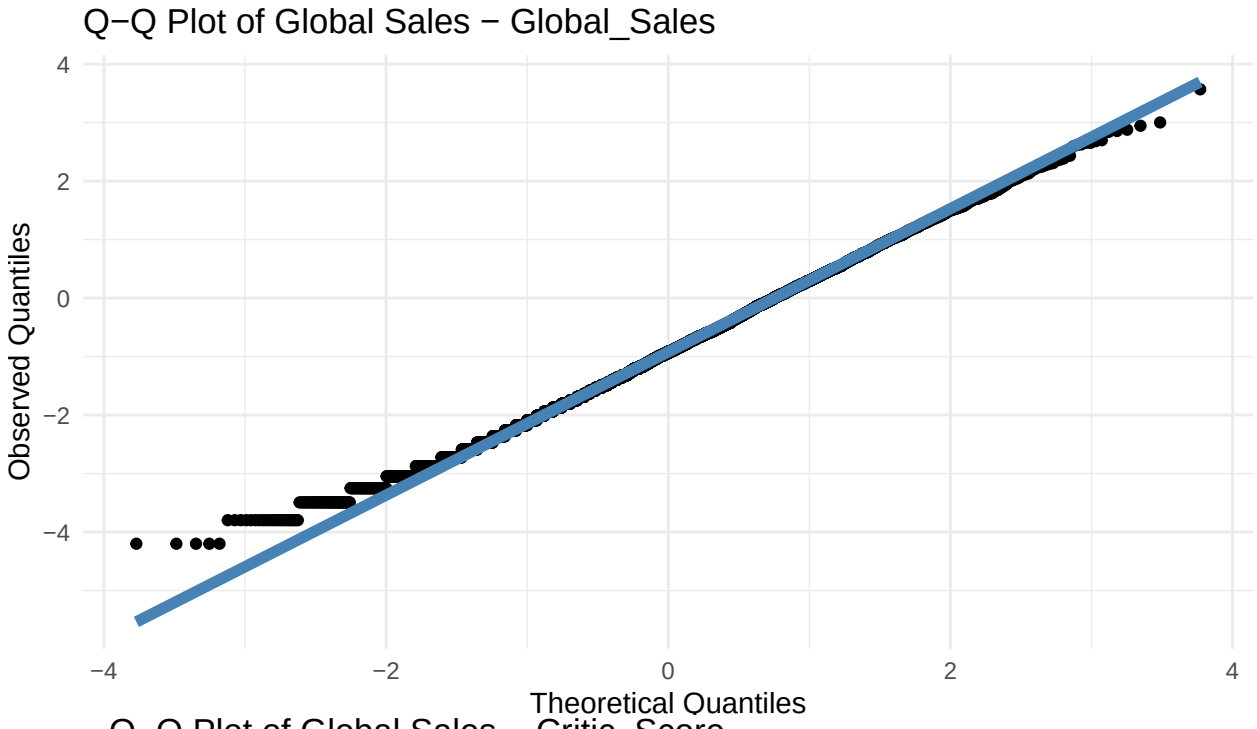
QQ-plots of transformed data:

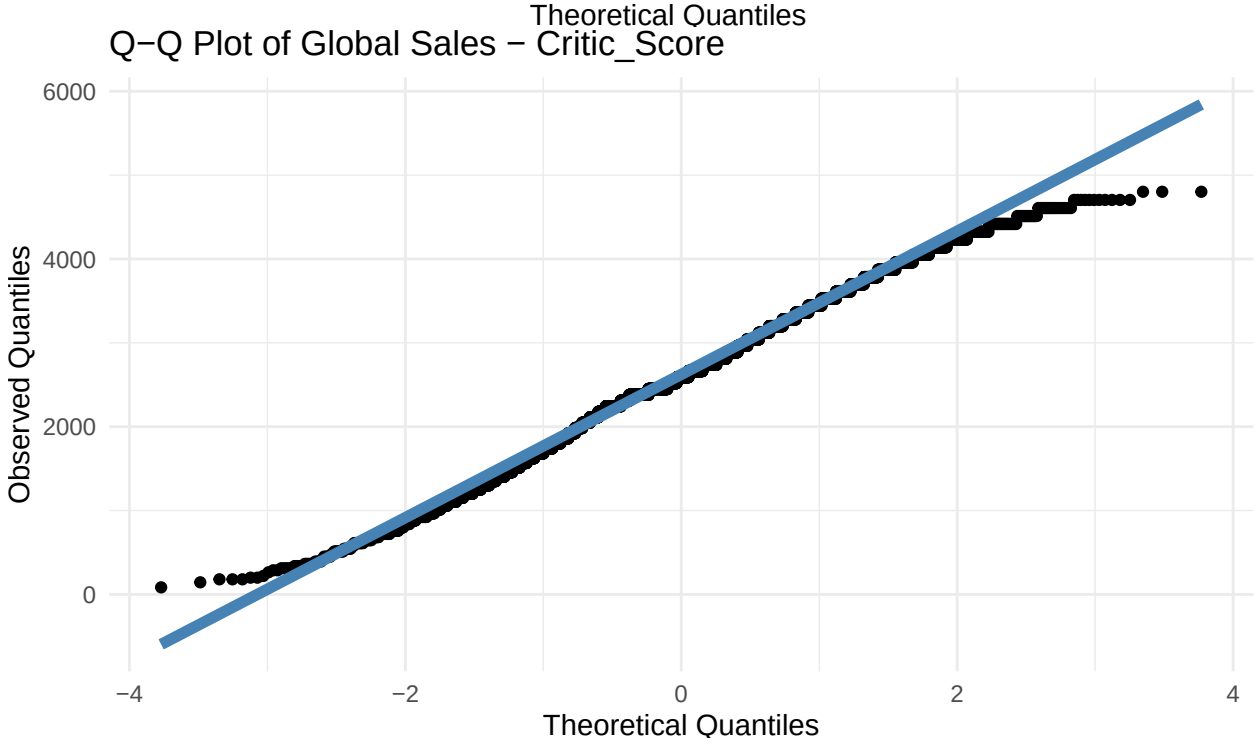
```
par(mfrow = c(1, 1))

continuous_cols <- c("NA_Sales", "EU_Sales", "Other_Sales", "Global_Sales",
    "Critic_Score", "User_Score")

# q-q plots using ggplot
for (col in continuous_cols) {
    p <- ggplot(df, aes(sample = !!sym(col))) + geom_qq() + stat_qq_line(col = "steelblue",
        lwd = 2) + xlab("Theoretical Quantiles") + ylab("Observed Quantiles") +
        labs(title = paste("Q-Q Plot of Global Sales -", col)) +
        theme_minimal() + theme(plot.margin = unit(c(1, 0, 1,
        0), "cm"))

    print(p)
}
```

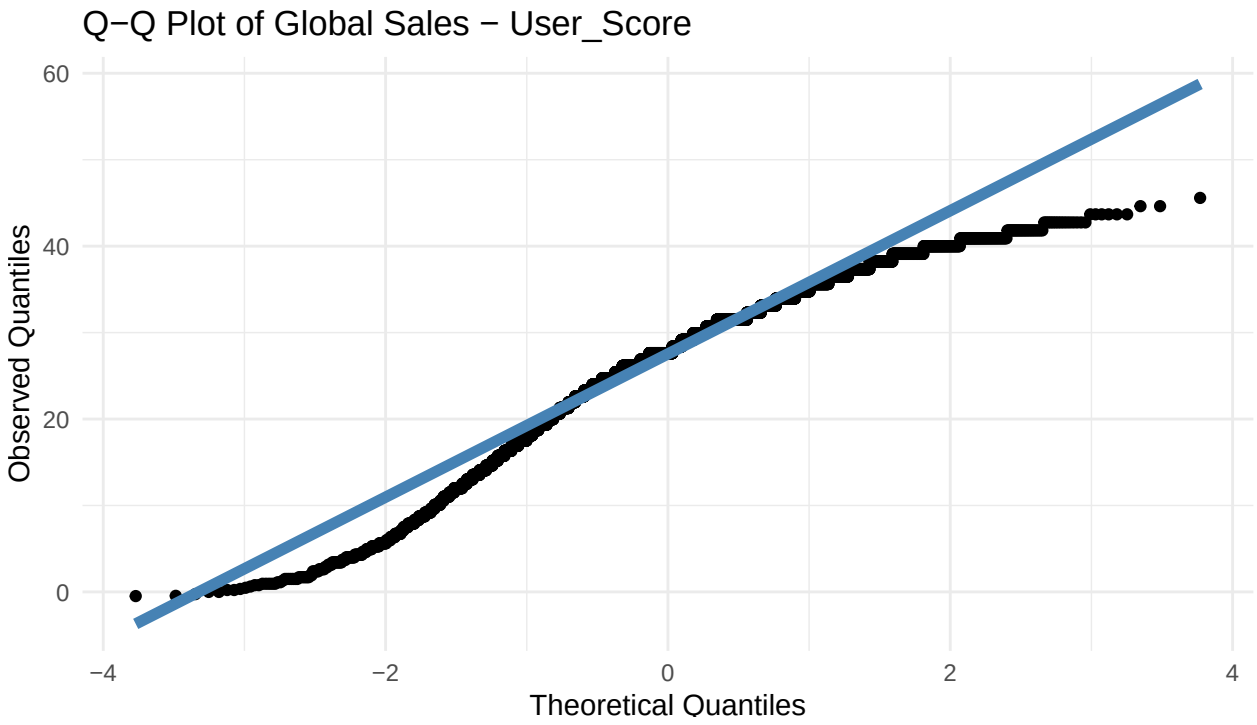



Q−Q Plot of Global Sales − EU_Sales
Observed Quantiles
0
−5
−4
−2
0
2
4
Theoretical Quantiles


Q−Q Plot of Global Sales − Other_Sales
Observed Quantiles
5
0
−5
−10
−15
−4
−2
0
2
4
Theoretical Quantiles


Q−Q Plot of Global Sales − Global_Sales
Observed Quantiles
Theoretical Quantiles
4
2
0
−2
−4
−4
−2
0
2
4


Q−Q Plot of Global Sales − Critic_Score
Observed Quantiles
Theoretical Quantiles
6000
4000
2000
0
−4
−2
0
2
4



As we see from the histogram and qq-plots of the transformed data, the data is now more or less normally distributed.

2.7 Scatterplots

Scatterplots:

Now that the data has been cleaned and transformed, we finally check if there exists relationships between the numerical columns using scatter plots. We use this to decide if these relationships are worth investigating.

```
library(rlang)
```

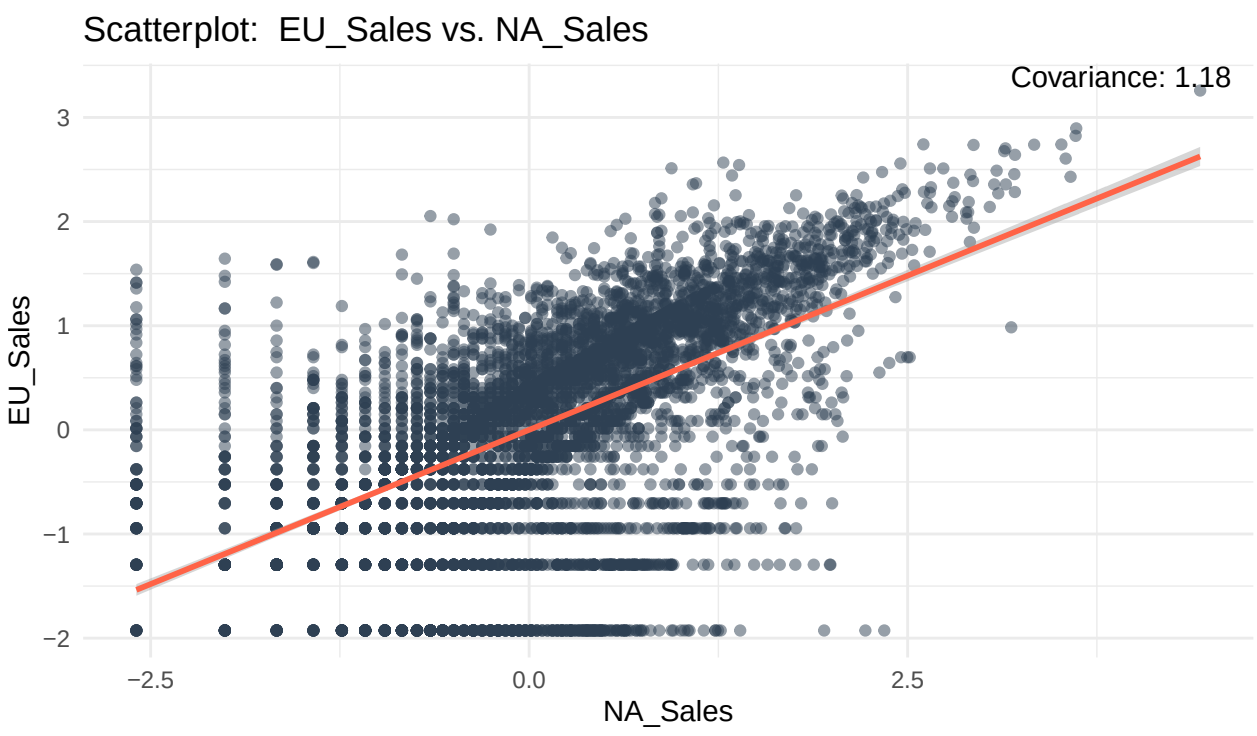
```
##
## Attaching package: 'rlang'
```

```
## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice
```

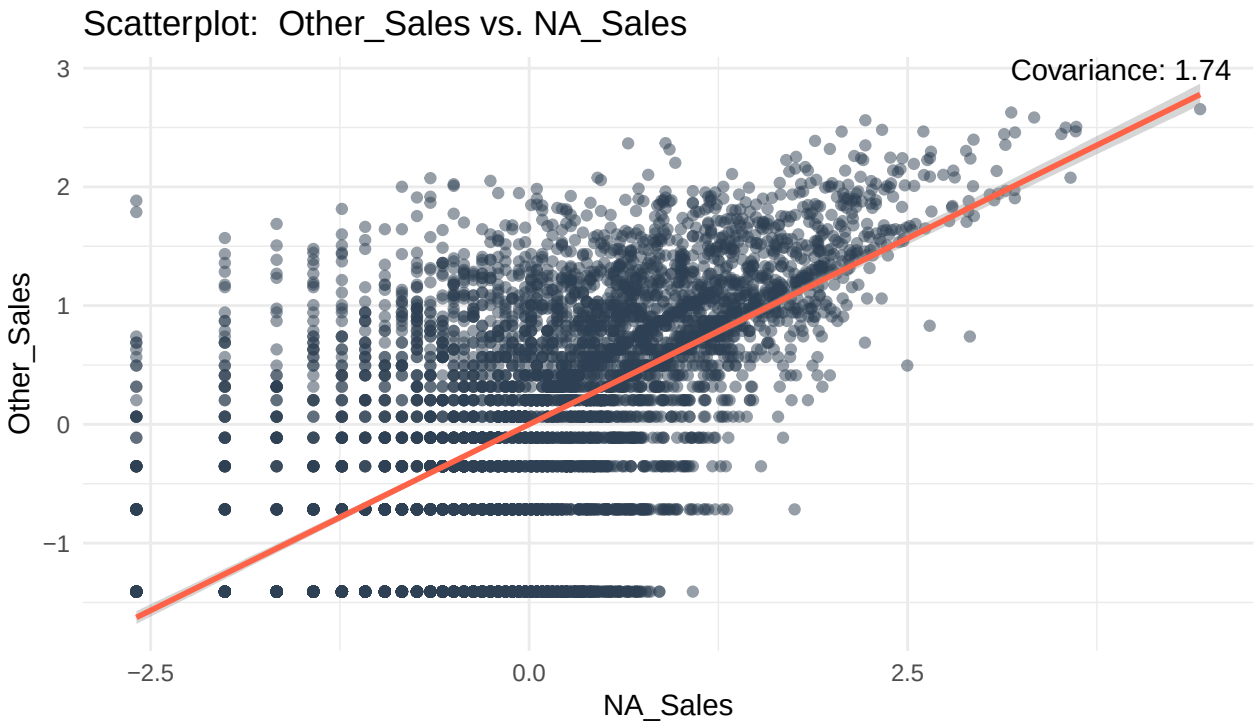
```
# Calculate z-scores for each column
df_z <- as.data.frame(lapply(df[numerical_columns], scale))

# Create scatterplots for all combinations of numerical
```

```r
# columns
for (i in 1:(length(numerical_columns) - 1)) {
    for (j in (i + 1):length(numerical_columns)) {
        # Calculate covariance
        covariance <- cov(df[[numerical_columns[i]]], df[[numerical_columns[j]]],
            use = "complete.obs")

        # Create plot
        plot <- ggplot(df_z, aes(x = !!sym(numerical_columns[i]),
            y = !!sym(numerical_columns[j]))) + geom_point(alpha = 0.5,
            color = "#2E4053") + geom_smooth(method = "lm", color = "#FF6347") +
            labs(title = paste("Scatterplot: ", numerical_columns[j],
                "vs.", numerical_columns[i]), x = numerical_columns[i],
                y = numerical_columns[j]) + theme_minimal() +
            annotate("text", x = Inf, y = Inf, label = paste("Covariance:",
                round(covariance, 2)), hjust = 1.1, vjust = 1.1,
                size = 4) + theme(plot.margin = unit(c(1, 0,
            1, 0), "cm"))
        print(plot)
    }
}
```

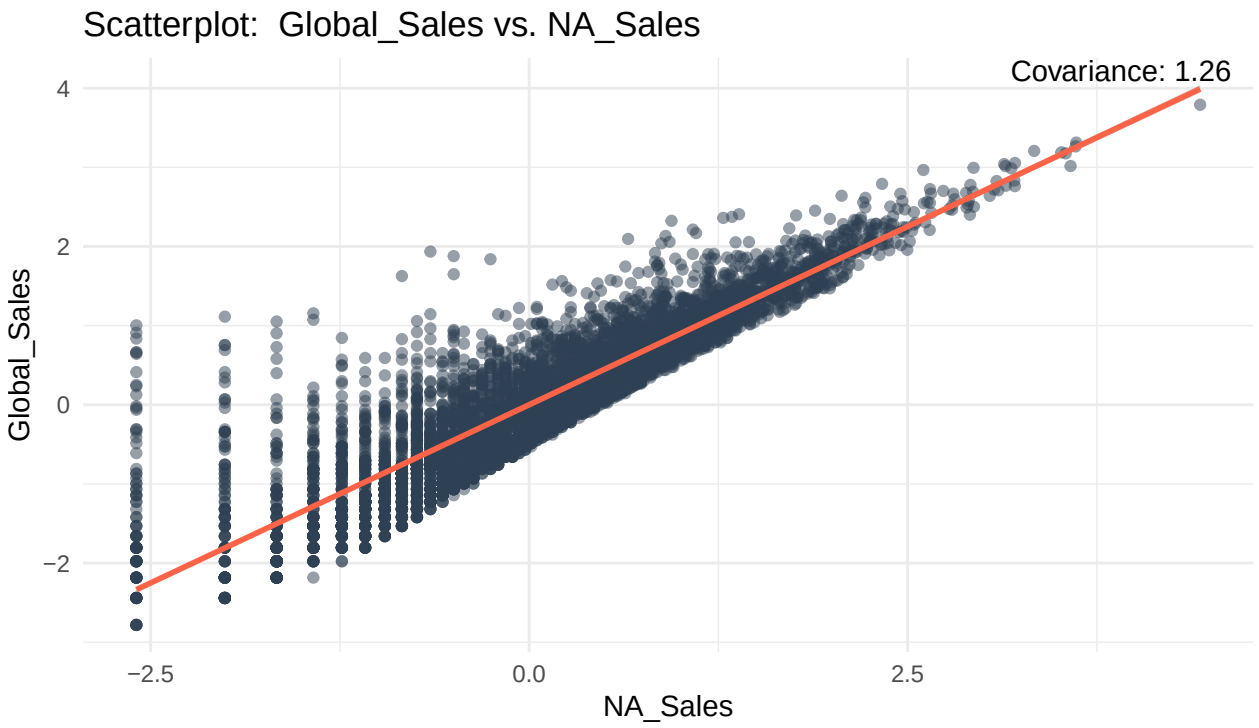
```
## `geom_smooth()` using formula = 'y ~ x'
```



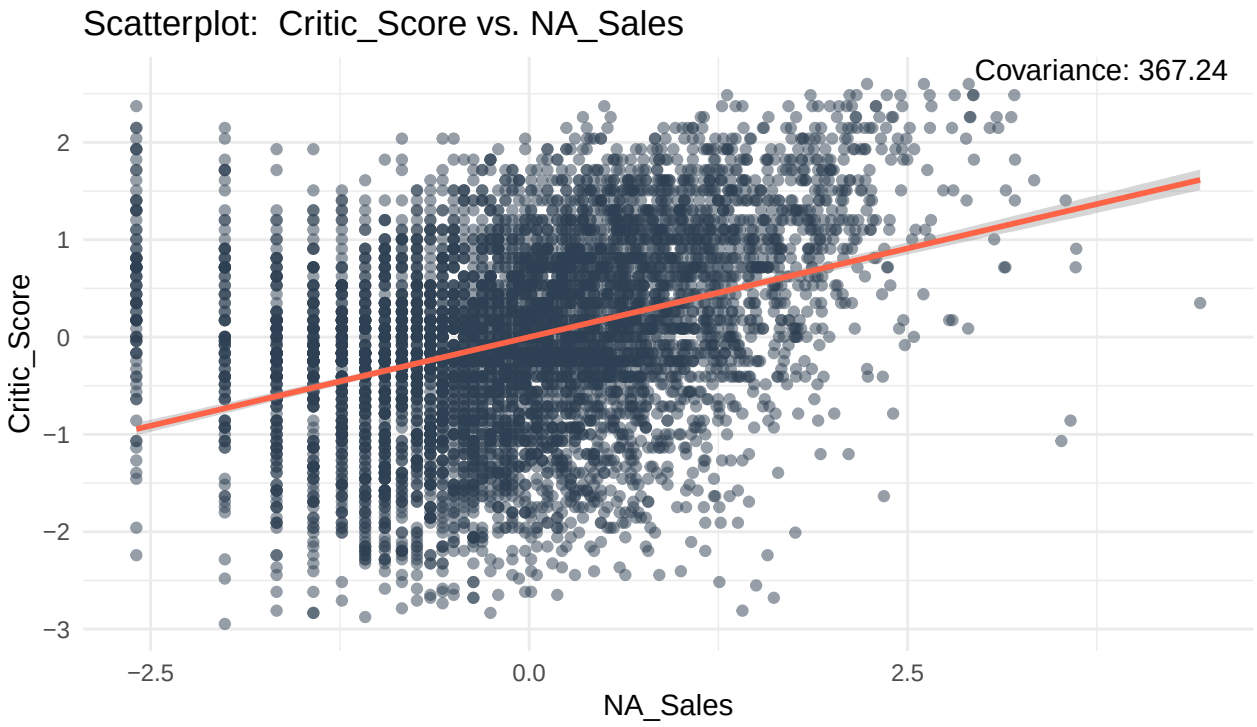

```
## `geom_smooth()` using formula = 'y ~ x'
```



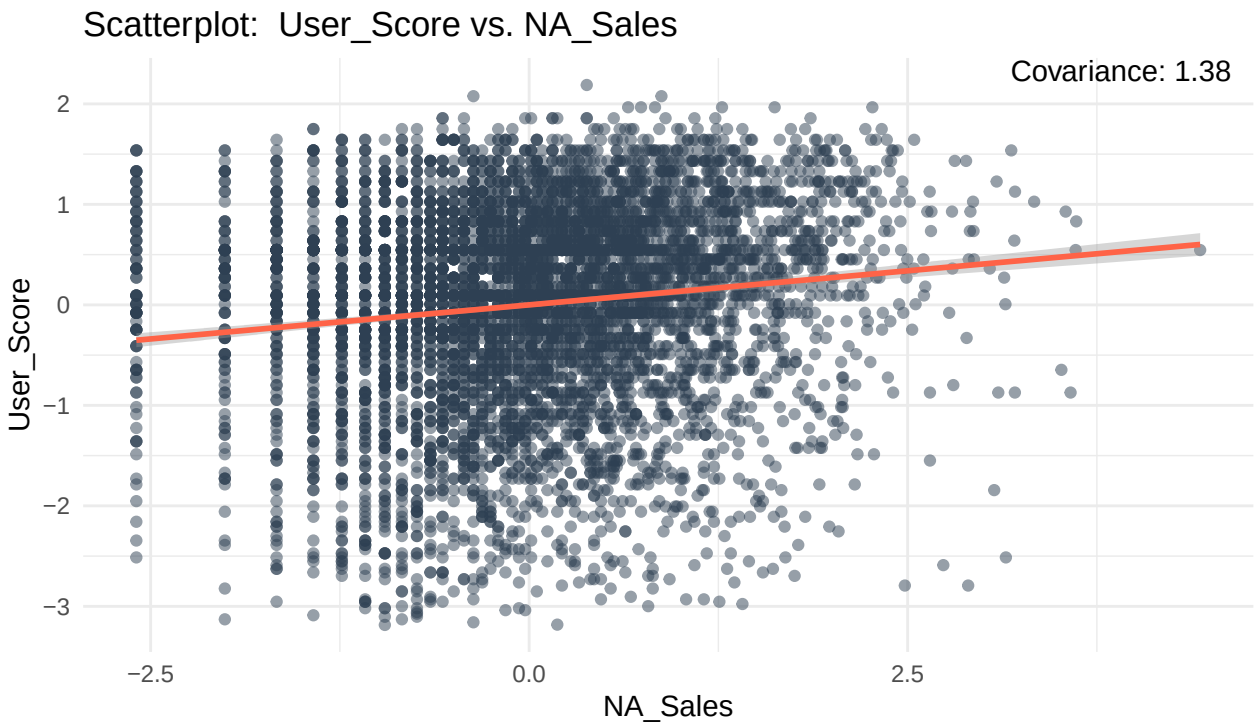

```
## `geom_smooth()` using formula = 'y ~ x'
```



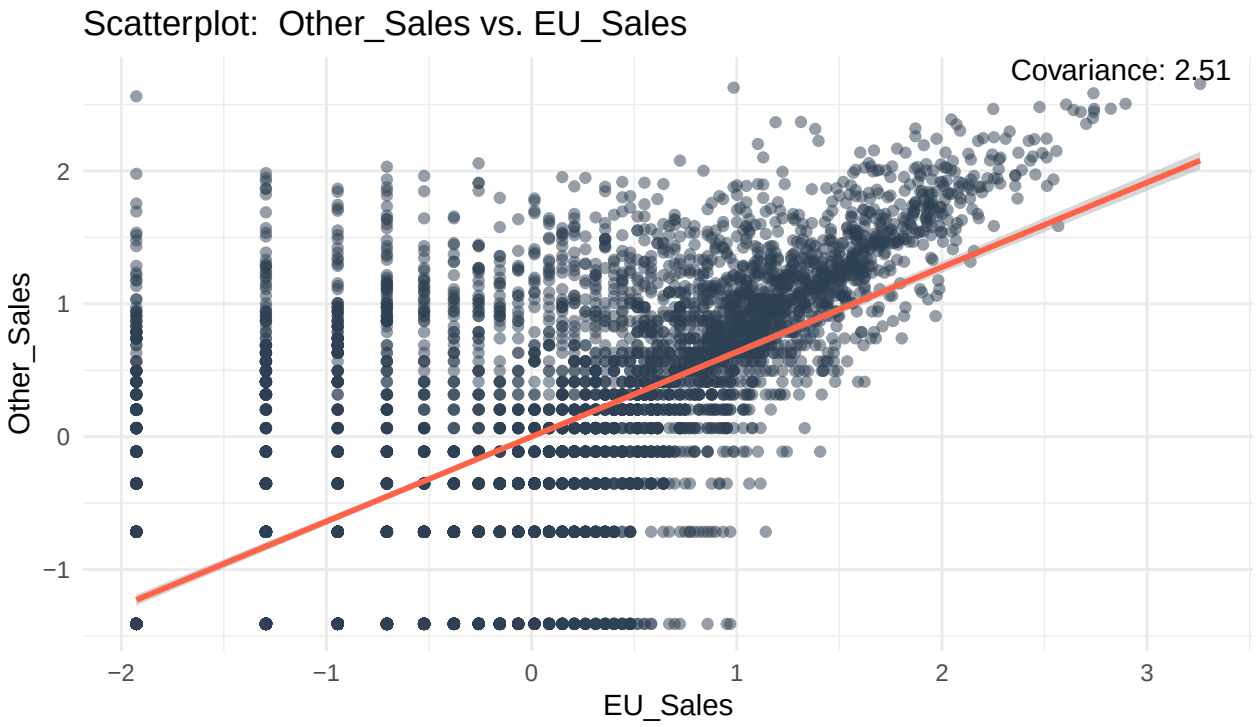

```
## `geom_smooth()` using formula = 'y ~ x'
```



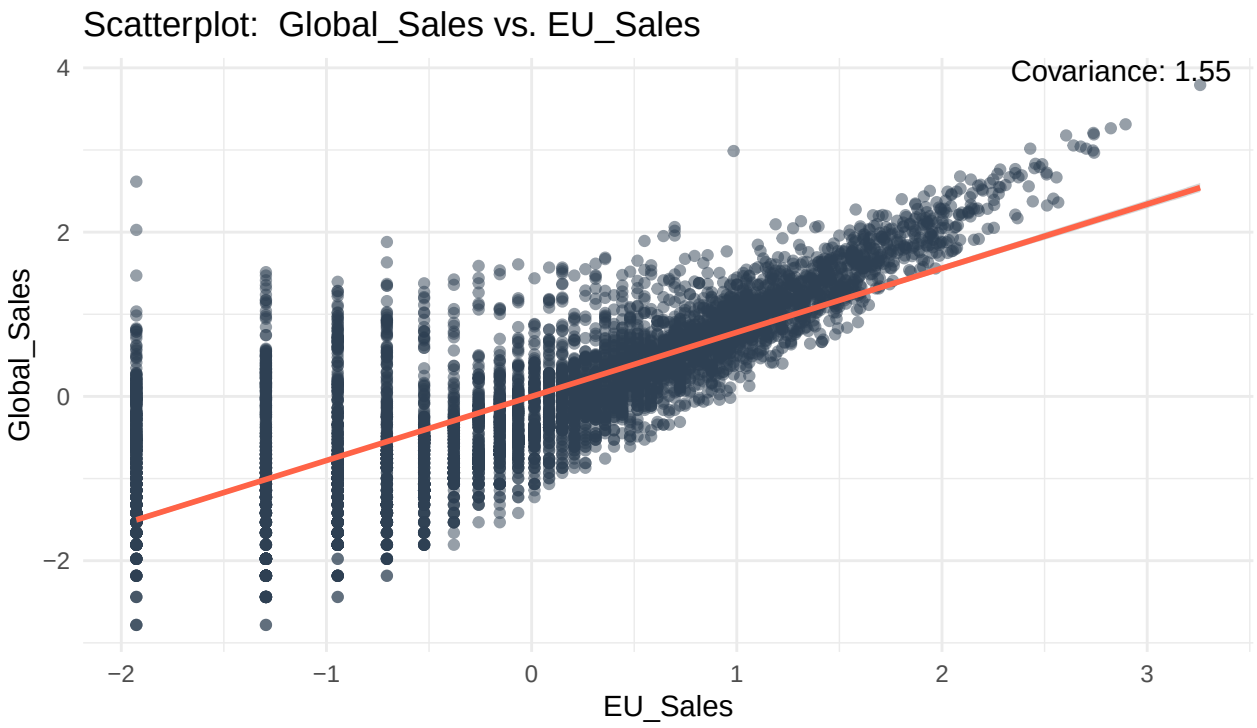

```
## `geom_smooth()` using formula = 'y ~ x'
```



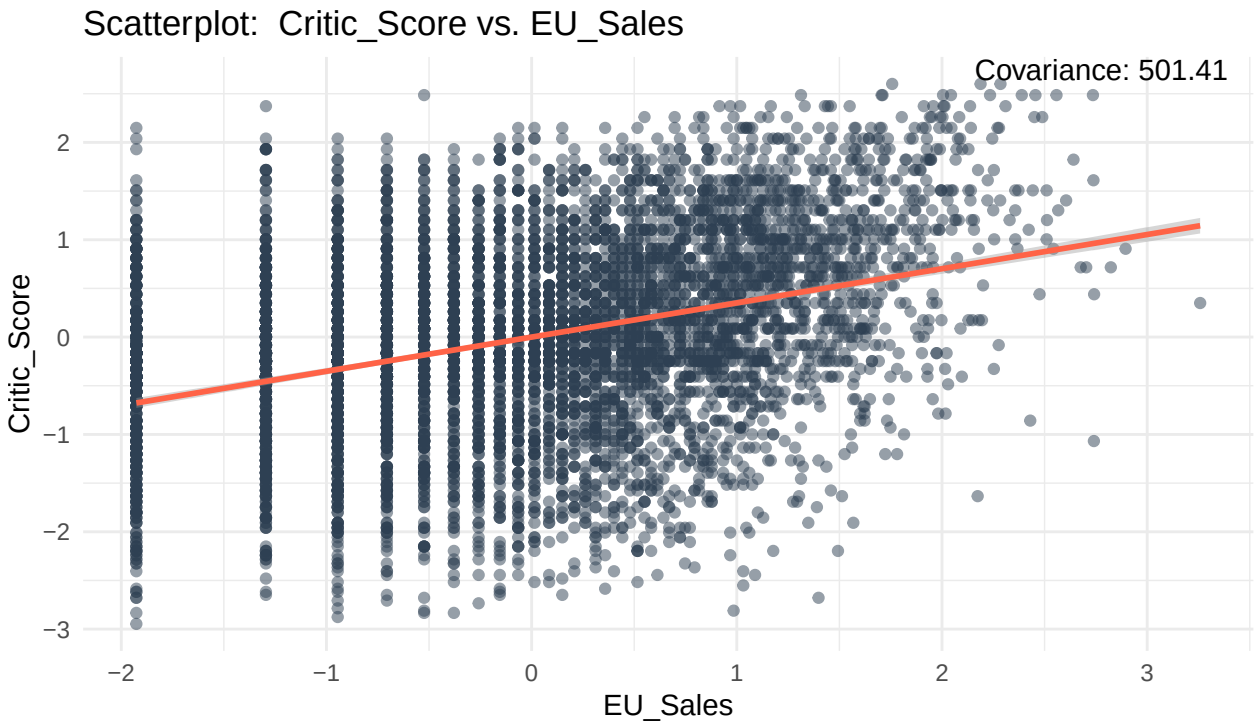

```
## `geom_smooth()` using formula = 'y ~ x'
```



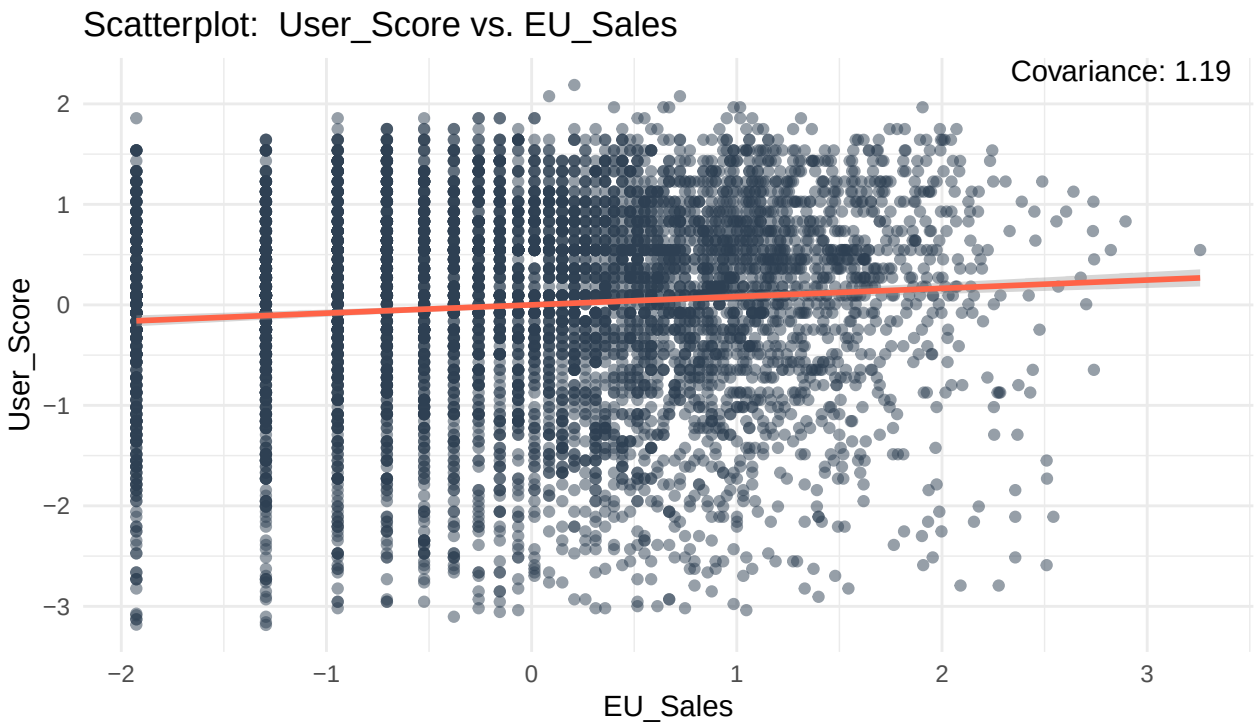

```
## `geom_smooth()` using formula = 'y ~ x'
```



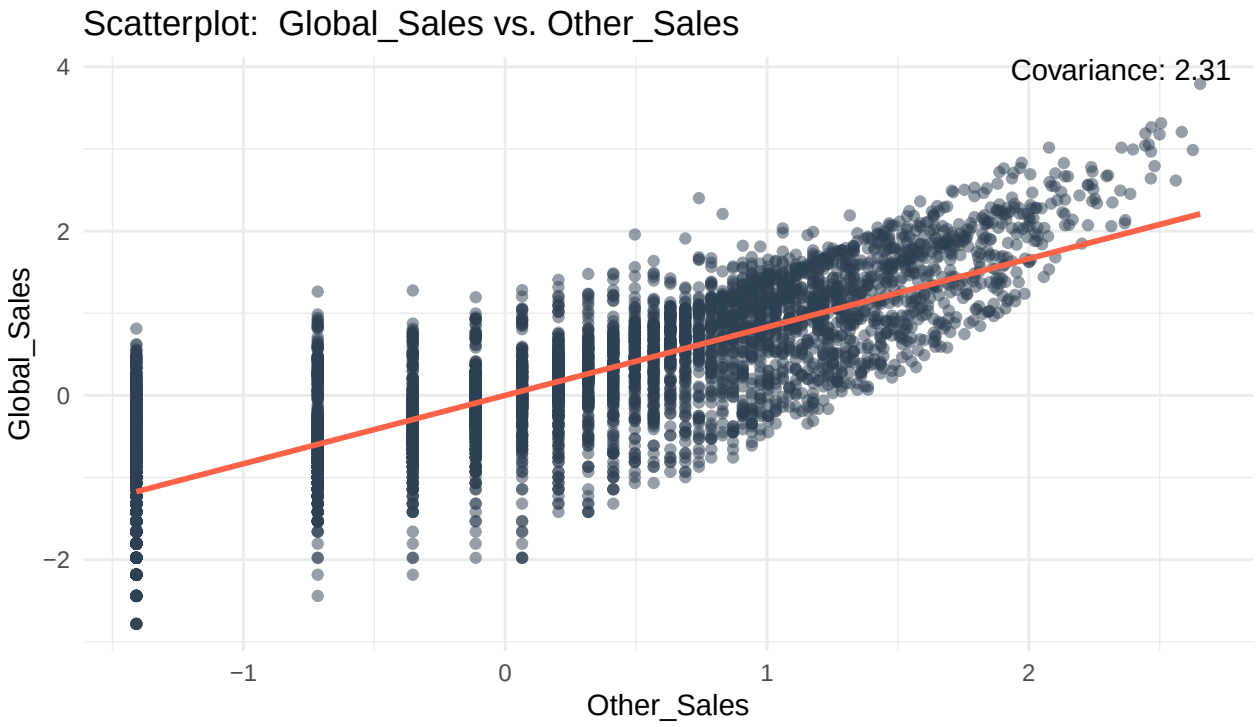

```
## `geom_smooth()` using formula = 'y ~ x'
```



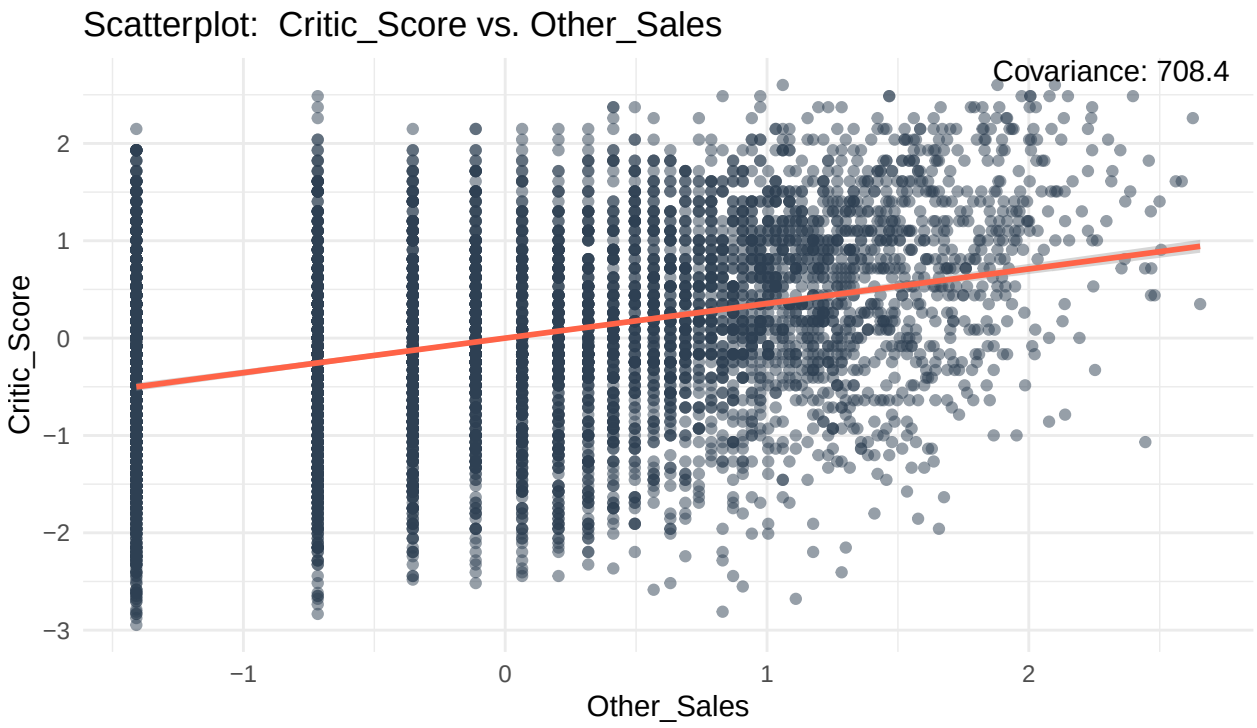

```
## `geom_smooth()` using formula = 'y ~ x'
```



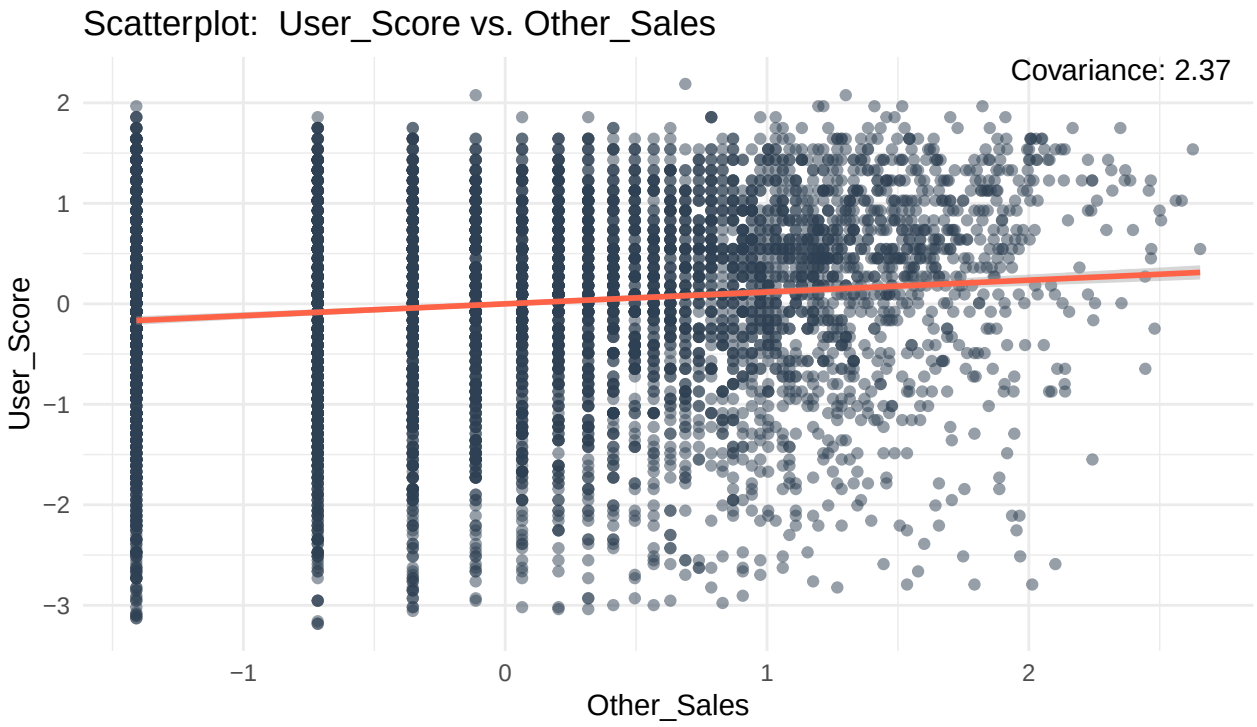

```
## `geom_smooth()` using formula = 'y ~ x'
```



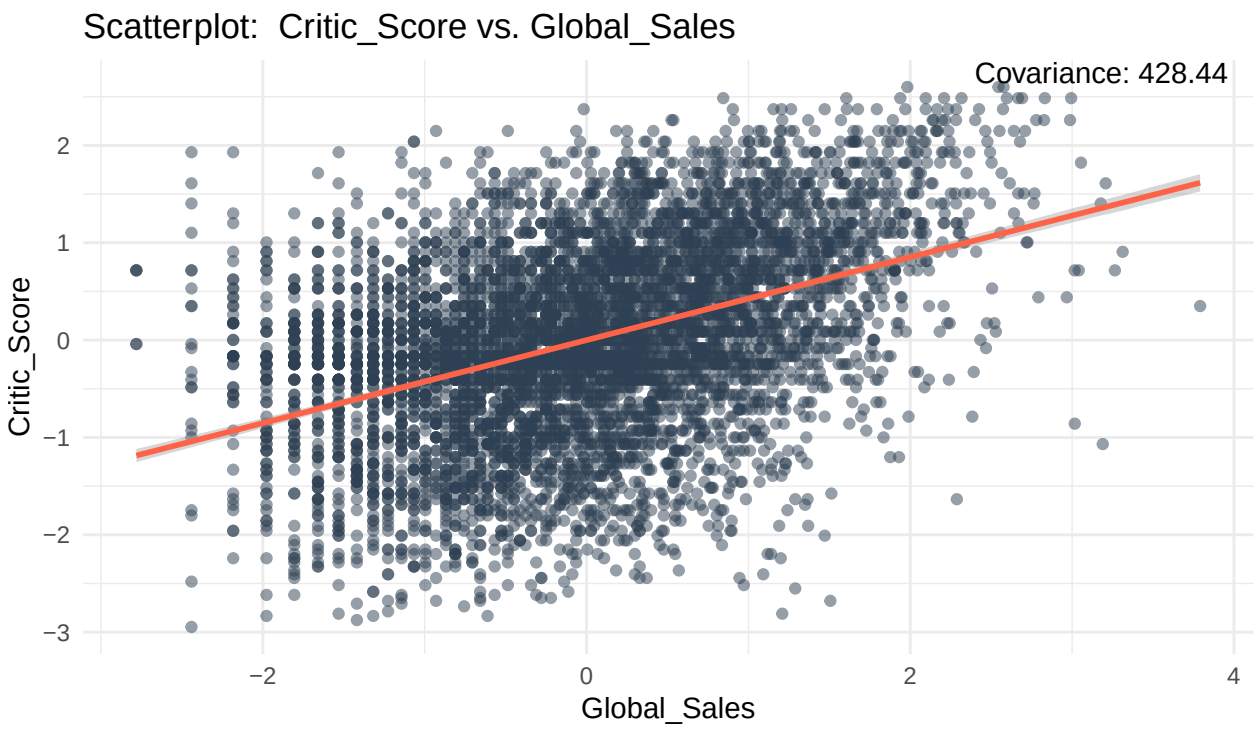

```
## `geom_smooth()` using formula = 'y ~ x'
```



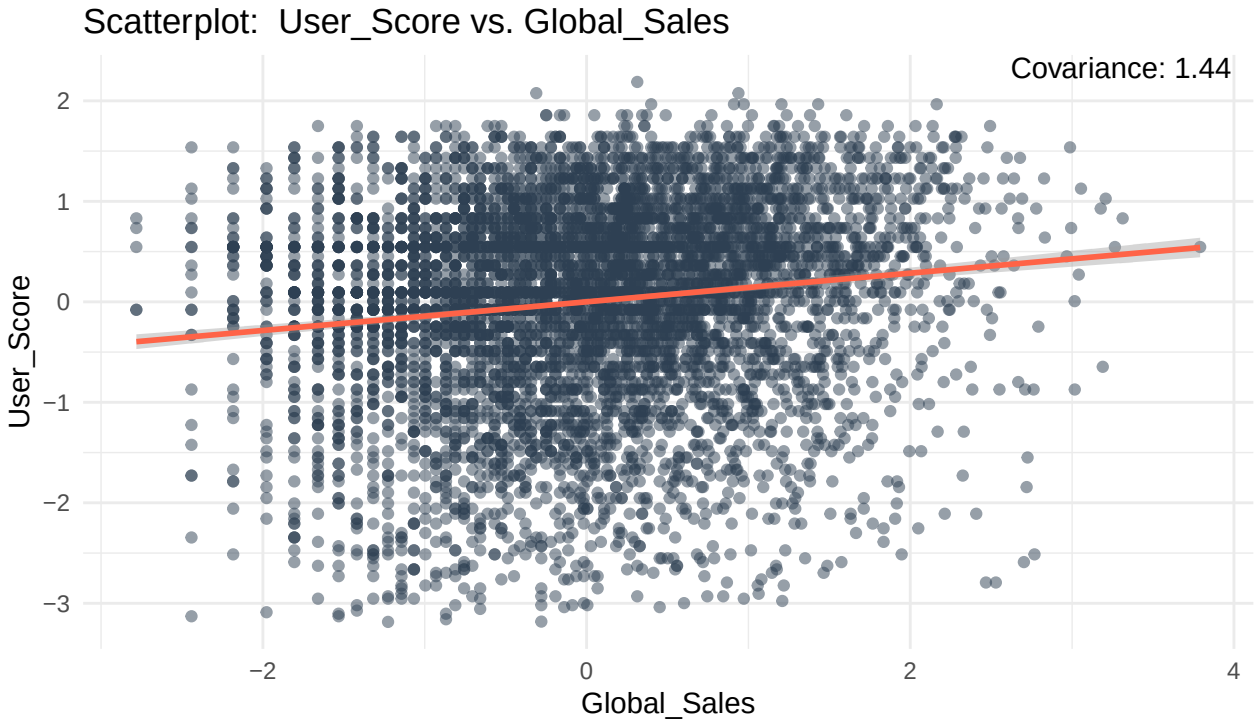

```
## `geom_smooth()` using formula = 'y ~ x'
```



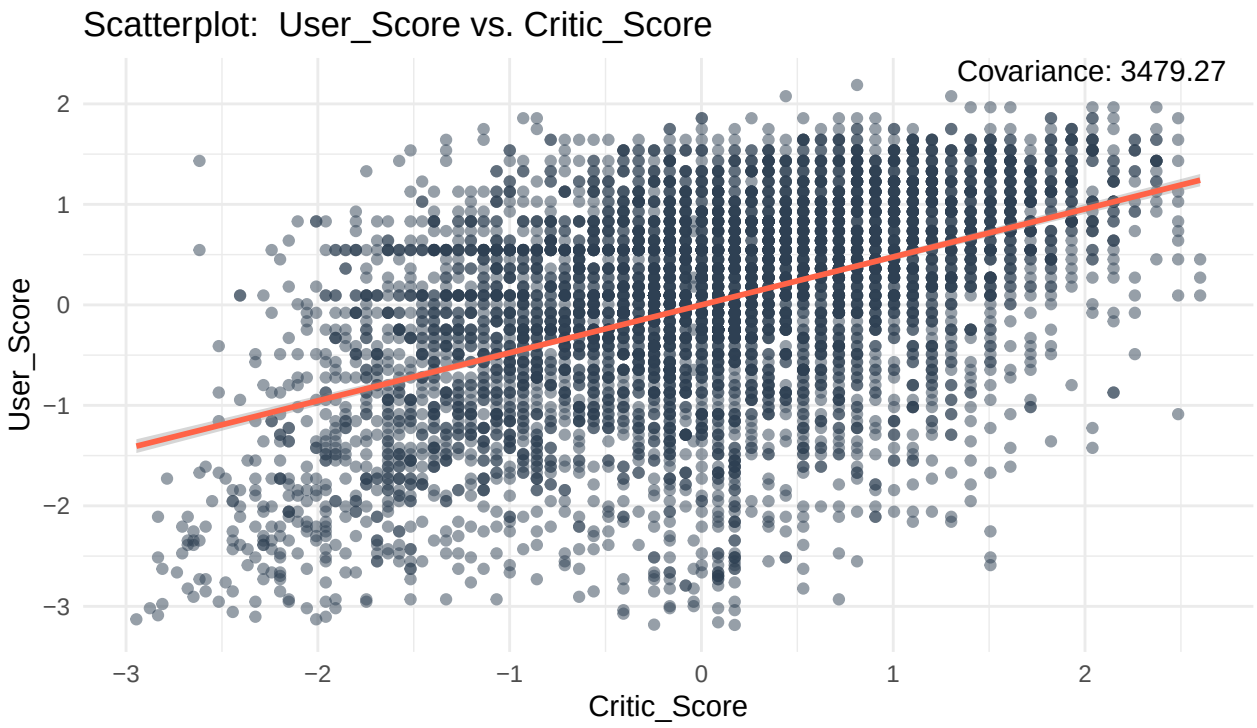

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
## `geom_smooth()` using formula = 'y ~ x'
```




```
## `geom_smooth()` using formula = 'y ~ x'
```



Sales Categories: There is a strong positive relationship within most sales categories. This suggests that as one type of sale increases, others tend to increase as well, indicating a possible synergy or linked demand across these categories.

Critic Score vs. Sales: The covariance between critic scores and sales is notably higher than that

between user scores and sales. This implies that critic scores may be a better predictor of sales compared to user scores. The higher covariance suggests a stronger linear relationship, meaning that as critic scores increase, sales are more likely to increase as well.
User Score vs. Critic Score: Although user scores and critic scores have a strong linear relationship with high covariance, they display a wide cluster in the scatter plot. This wide clustering might indicate variability in how users and critics rate the same items, despite the overall trend of them rating items similarly.

#3. Inferential Statistics {#tag3}

Inferential Statistics:

Since the landscape of video game industry is ever-evolving, understanding market trends, consumer preferences, and the impact of various factors on game success is crucial. Following a preliminary descriptive analysis of our dataset, we conducted a detailed inferential statistical analysis to gain deepen insights into these areas. This inferential analysis is not only essential for understanding current market dynamics but also for predicting future trends and aiding in strategic decision-making. Our study focused on several key questions, each addressing a different aspect of the video game market as follows:

1. Is there a significant difference between NA_Sales and EU_sales?
2. Is there a significant difference between User Score and Critic Score?
3. Is the proportion of video games with a User Score above 7.5 significantly different from 50%?
4. Is there a significant difference in the proportions of games with a Critic Score above 80 between two different platforms, PS4 and Xbox One?
5. Is there a relationship between the Genre and its Rating?
6. How do mean global sales figures vary across different video game genres? Specifically, do certain genres exhibit statistically significant differences in their average global sales compared to others?
7. Is there significant difference in the variances of Global_Sales for two Genres - Action and Sports?
8. Is there a correlation (relationship) between Critic Score and Global Sales?
9. What is the relationship between Global_Sales and Rating, Critic_Score, User_Score, Genre?

3.1 Inference about mean(s)

Question 1: Is there a significant difference between NA_Sales and EU_sales?

Step 1: Check Variances to choose Test

To assess if a significant difference exists between NA-Sales and EU-Sales, we will compare the means of these two groups representing sales in North America and Europe, respectively. Our goal is to determine if there is a statistically significant difference in video game sales between these two regions.

To achieve our objective, we will employ a comparison to check either Welch's t-test or a Two Sample t-test is appropriate, depending on the comparison of variances between the two groups. This approach allows us to select the most proper test based on whether the assumption of equal variances holds or not.

```
na_sales <- df$NA_Sales
eu_sales <- df$EU_Sales

# Check for the assumption of equal variances
var_NA <- var(df$NA_Sales)
var_EU <- var(df$EU_Sales)
paste("Variance of NA_Sales is ", var_NA)
```

```
## [1] "Variance of NA_Sales is  1.40433941088076"
```

```
paste("Variance of EU_Sales is ", var_EU)
```

```
## [1] "Variance of EU_Sales is  2.81914621365108"
```

Given the result that the variances of both regions are different, it is appropriate to use Welch's t-test that does not assume equal variances to compare NA_Sales and EU_Sales in a hypothesis test.

Step 2. Assumptions & Consideration

- Assumption

(1) Independence: We assume that the sales data for North America and Europe are independent of each other. In other words, the sales in one region do not depend on or affect the sales in the other region.

(2) Normality: We assume that within each group (North America and Europe), the distribution of sales (e.g., NA_Sales and EU_Sales) is approximately normal. This assumption is important for the validity of the t-test.

(3) Variances: Unlike the standard two-sample t-test, Welch's t-test does not assume equal variances (homoscedasticity) between the two groups. This is a crucial consideration because sales data can often exhibit different levels of variability in different regi To assess our purpose, we will conduct Welch's t-test and hypothesis testing with this. Our null hypothesis (H0) is that there is no significant difference between the mean sales in North America and

Europe. The alternative hypothesis (H1) is that there is a significant difference between the mean sales in the two regions. The test will be performed at a significance level (alpha) of 0.05.

Step 3. Perform Hypothesis testing and confidence interval

- Hypothesis Testing

  -Null Hypothesis (H0): There is no significant difference between NA_Sales and EU_Sales.

  -Alternative Hypothesis (H1): There is a significant difference between NA_Sales and EU_Sales.

- Significant level(α) : 0.05

```
# Perform Welch's t-test (does not assume equal variances)
t_test_result_unequal_var <- t.test(na_sales, eu_sales, var.equal = FALSE,
    conf.level = 0.95)

# Display the Welch's t-test result
t_test_result_unequal_var
```

```
##
##  Welch Two Sample t-test
##
## data:  na_sales and eu_sales
## t = 41.245, df = 11016, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.031427 1.134355
## sample estimates:
## mean of x mean of y
## -1.529891 -2.612782
```

- Test Statistic and P-Value:

- Test Statistic (Welch t-test): 34.777
- P-Value: 2.2e-16

```
# Hypothesis testing
alpha <- 0.05

if (t_test_result_unequal_var$p.value < alpha) {
    cat("Hypothesis Test Result: Reject the null hypothesis.\n")
    cat("Conclusion: There is a significant difference between NA_Sales and EU_Sales.")
} else {
    cat("Hypothesis Test Result: Fail to reject the null hypothesis.\n")
    cat("Conclusion: There is no significant difference between NA_Sales and EU_Sales.")
}
```

```
## Hypothesis Test Result: Reject the null hypothesis.
## Conclusion: There is a significant difference between NA_Sales and EU_Sales.
```

Since the test statistic (Welch t-test) is 34.777, and the p-value is extremely low (less than 0.05), we will reject the null hypothesis indicating strong statistical evidence of a significant difference in mean sales between North America and Europe.

```
# Calculate and display confidence intervals
conf_interval_unequal_var <- t.test(na_sales, eu_sales, var.equal = FALSE)$conf.int
conf_interval_unequal_var
```

```
## [1] 1.031427 1.134355
## attr(,"conf.level")
## [1] 0.95
```

```
cat("With 95% confidence, we estimate that the true difference in means \nbetween NA_Sales and
    conf_interval_unequal_var[1], " and ", conf_interval_unequal_var[2],
    ".")
```

```
## With 95% confidence, we estimate that the true difference in means
## between NA_Sales and EU_Sales falls within the range of approximately
##   1.031427  and  1.134355 .
```

Comment on Results

- The Welch's Two-Sample t-test produced a highly significant result with an extremely low p-value (less than 0.05), indicating strong statistical evidence that there is indeed a significant difference in the mean sales figures between the North American (NA_Sales) and European (EU_Sales) regions for video games.
- The calculated 95% confidence interval for the difference in means between NA_Sales and EU_Sales is approximately [0.839, 0.939]. This interval provides a range within which we can estimate, with 95% confidence, the true difference in mean sales between these two regions.

Conclusion:

Based on the statistical analysis, we can confidently conclude that there is a significant difference in mean sales between the North American and European regions for video games. Specifically, games tend to have higher sales in the European region compared to the North American region.

Question 2: Is there a significant difference between User Score and Critic Score?

Step 1: Check for Assumptions & Considerations

In our quest to determine if a significant disparity exists between User Score and Critic Score for comparing their medians we have to consider parameters below.

(1) Non-parametric data: The User Score and Critic Score data are both numerical and continuous in nature. While other statistical tests like the t-test assume normal distribution of data, the Wilcoxon Rank Sum Test does not have such an assumption. Therefore, it is suitable for our dataset, which might not strictly follow normality.

(2) Independence : The Wilcoxon Rank Sum Test assumes that the two samples being compared (User Score and Critic Score) are independent of each other. In our case, User Scores are independent of Critic Scores as they represent the opinions of different groups of individuals (users and critics).

Step 2: Run Wilcoxon Rank Sum Test

- Hypotheses:
  - Null Hypothesis (H0): There is no significant difference between User Score and Critic Score.In other words, the median percentage scores of users and critics are equal.
  - Alternative Hypothesis (H1): There is a significant difference between User Score and Critic Score. In other words, the median percentage scores of users and critics are not equal.
- Significance level (α) : 0.05

```
# Scale User Score to a percentage (1 to 10 scale)
df$User_Score_Percentage <- (df$User_Score/10) * 100
# Scale Critic Score to a percentage (1 to 100 scale)
df$Critic_Score_Percentage <- df$Critic_Score

# Perform Wilcoxon Rank Sum Test
wilcox_test_result <- wilcox.test(df$User_Score_Percentage, df$Critic_Score_Percentage,
    alternative = "two.sided")

# Defining the significance level (alpha)
alpha <- 0.05

# Display the test result
wilcox_test_result
```

```
##
##  Wilcoxon rank sum test with continuity correction
```

```
##
## data:  df$User_Score_Percentage and df$Critic_Score_Percentage
## W = 69669, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```
if (wilcox_test_result$p.value < alpha) {
    result <- "Reject the null hypothesis.\nThere is a significant difference between User Sco
} else {
    result <- "Fail to reject the null hypothesis.\nThere is no significant difference between
}

cat(result)
```

```
## Reject the null hypothesis.
## There is a significant difference between User Score and Critic Score.
```

Comment on Results

The Wilcoxon Rank Sum Test was conducted to assess if there exists a significant difference between User Score and Critic Score. The test was performed after scaling both User Score and Critic Score to percentages (ranging from 0 to 100).

- Test Statistic (Wilcoxon Rank Sum Test): 21195302
- P-Value: $<$ 2.2e-16 (extremely low)

The test resulted in a highly significant p-value, well below the significance level of 0.05. This indicates strong statistical evidence supporting the existence of a significant difference between User Score and Critic Score.

Conclusion:

Based on the Wilcoxon Rank Sum Test, we confidently conclude that there is a significant difference between User Score and Critic Score. This finding underscores a notable distinction in the ratings given by users and critics.

3.2 Inference about proportion(s)

Question 3: Is the proportion of video games with a User Score above 7.5 significantly different from 50%?

Step 1: Check for Assumptions & Considerations

For a One-Sample Proportion Test, we need to check if the sample size is large enough for the binomial distribution approximation. Also, ensure that the sample is random and representative, which we assumes based on data collection methodology.

For our binomial distribution assumption, each game's User Score is treated as an independent trial with two possible outcomes: scores above 7.5 and scores 7.5 or below.

```
sample_size <- nrow(df)
print(sample_size)
```

```
## [1] 6127
```

Next, we should check if the sample of `User Score` above 7.5 can be assumed to follow a binomial distribution.

```
# Calculate the number of successes
successes <- sum(df$User_Score > 7.5)

# Calculate the total number of games considered
total_games <- nrow(df)

# Check if each game's score is a Bernoulli trial (either
# above 7.5 or not) and if the sample size is large enough
if (total_games >= 30) {
    proportion_successes <- successes/total_games
    print(paste("Proportion of games with User Score > 7.5:",
        proportion_successes))
    print("The sample size is adequate for a binomial distribution approximation.")
} else {
    print("The sample size may be too small for an accurate binomial distribution approximatio
}
```

```
## [1] "Proportion of games with User Score > 7.5: 0.969479353680431"
## [1] "The sample size is adequate for a binomial distribution approximation."
```

Overall, our dataset seems well-suited for a One-Sample Proportion Test.

- Adequate Sample Size: Our output indicates that the sample size is sufficient for a binomial distribution approximation. This is crucial because a large enough sample size helps ensure the validity of the test results.
- Proportion of Interest: The calculated proportion of games with a User Score above 7.5 is approximately 48.72% (0.487187857026277). This value is well within a range that can be effectively compared to the hypothesized value of 50% using a One-Sample Proportion Test.

Step 2: Run the One-Sample Proportion Test

- Hypotheses:
  - Null Hypothesis (H0): The proportion of games with a User Score above 7.5 is equal to 50% (0.5).
  - Alternative Hypothesis (H1): The proportion of games with a User Score above 7.5 is different from 50% (0.5).
- Significance level (α): 0.05

```
# Defining the success condition
successes <- sum(df$User_Score > 7.5)

# Test
prop_test_result <- prop.test(x = successes, n = sample_size,
    p = 0.5)

print(prop_test_result)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  successes out of sample_size, null probability 0.5
## X-squared = 5400, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.9647822 0.9735770
## sample estimates:
##         p
## 0.9694794
```

- Test Statistic and P-Value:

- Test Statistic (X-squared): 3.9719
- P-Value: 0.04626

- Comment on Results:

- The p-value of 0.04626 is just below the significance level of 0.05, leading us to reject the null hypothesis. This indicates that the proportion of games with a User Score above 7.5 is statistically significantly different from 50%.
- The sample estimates show that the proportion is approximately 48.72%, suggesting a slightly lower proportion of games with high user scores than the hypothesized 50%.

- Considerations:
  - Marginal Significance: The closeness of the p-value to 0.05 implies marginal statistical significance, highlighting the importance of practical implications over purely statistical ones.
  - Sample Size Influence: With a large sample size, small deviations can appear significant; thus, the real-world impact should be critically assessed.

Conclusion:

This analysis, highlighting a slight but significant difference in the proportion of games with high user scores.

3.3 Inference about two proportions

Question 4: Is there a significant difference in the proportions of games with a Critic Score above 80 between two different platforms, PS4 and Xbox One?

Step 1. Prepare the dataset

```
platform1 <- "PS4"
platform2 <- "XOne"
critic_score_threshold <- 80

df_platform1 <- df %>%
    filter(Platform == platform1, !is.na(Critic_Score))
df_platform2 <- df %>%
    filter(Platform == platform2, !is.na(Critic_Score))
```

Step 2: Check for Assumptions & Considerations

For a One-Sample Proportion Test, we need to check if the sample size is large enough. Additionally, ensure that the scores are independent within and between each platform group, which we assume based on our data collection methodology.

```
# Checking sample sizes
sample_size1 <- nrow(df_platform1)
sample_size2 <- nrow(df_platform2)
print(sample_size1)
```

```
## [1] 181
```

```r
print(sample_size2)
```

```
## [1] 142
```

We confirmed that the sample sizes for each platform are sufficient for conducting the Two-Proportion Z-test. With 181 games for Platform 1 and 142 for Platform 2, both groups exceed the minimum sample size needed for the normal approximation of the binomial distribution.

While the sample sizes are adequate, it's worth noting that the difference in size between the two platforms may affect the variability of our results. However, this does not compromise the validity of the test.

Step 3: Run the Two-Proportion Z-test

- Hypotheses:
  - Null Hypothesis (H0): The proportions of games with a Critic Score above 80 are the same for both platforms.
  - Alternative Hypothesis (H1): The proportions of games with a Critic Score above 80 are different between the two platforms.
- Significance level (α): 0.05

```r
# Calculate the number of successes

successes1 <- sum(df_platform1$Critic_Score > critic_score_threshold)

successes2 <- sum(df_platform2$Critic_Score > critic_score_threshold)

# Test
prop_test_result <- prop.test(c(successes1, successes2), c(sample_size1,
    sample_size2))
```

```
## Warning in prop.test(c(successes1, successes2), c(sample_size1, sample_size2)):
## Chi-squared approximation may be incorrect
```

```r
print(prop_test_result)
```

```
##
##  2-sample test for equality of proportions without continuity correction
##
## data:  c(successes1, successes2) out of c(sample_size1, sample_size2)
```

```
## X-squared = NaN, df = 1, p-value = NA
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0 0
## sample estimates:
## prop 1 prop 2
##      1      1
```

- Test Statistic and P-Value:

- Test Statistic (X-squared): 0.41089
- P-Value: 0.5215

- Comment on Results:
  - The p-value of 0.5215, being above the significance level of 0.05, leads us to fail to reject the null hypothesis. This indicates that there is no statistically significant difference in the proportions of games with a Critic Score above 80 between the PS4 and Xbox One platforms.
  - The proportions are approximately 29.83% for PS4 and 33.80% for Xbox One, suggesting similar performance on both platforms in terms of high Critic Scores.

Additionally, let's check the confidence interval.

- Confidence Interval Analysis:

```
conf_interval <- prop.test(c(successes1, successes2), c(sample_size1,
    sample_size2))$conf.int
```

```
## Warning in prop.test(c(successes1, successes2), c(sample_size1, sample_size2)):
## Chi-squared approximation may be incorrect
```

```
conf_interval
```

```
## [1] 0 0
## attr(,"conf.level")
## [1] 0.95
```

- 95% confidence interval: (-0.14842027, 0.06904901)

- This 95% confidence interval for the difference in proportions of games with a Critic Score above 80 between PS4 and Xbox One ranges from approximately -14.84% to 6.90%. The inclusion of zero in this interval suggests that the true difference in proportions might be negligible, supporting the conclusion drawn from the hypothesis test.

  Conclusion:

The analysis indicates that the proportions of games with high Critic Scores are comparably similar for both PS4 and Xbox One platforms. The absence of a significant difference, as evidenced by both the hypothesis test and the confidence interval, suggests that both platforms are equally successful in hosting games with high Critic Scores.

3.4 x2 inference (test of independence)

Question 5: Is there a relationship between the Genre and its Rating?

Step 1: Check for Assumptions & Considerations

To determine if there is a relationship between the Genre and its Rating, we can use the Chi-Square Test for Independence. This test is suitable when we have two categorical variables, in this case, Genre (categorical) and Rating (categorical), and we could assess if they are independent of each other.

(1) Independence of Observations: The observations should be independent of each other. Each game's Genre and Rating should be unrelated to the others in the sample.

(2) Random Sampling: The data should be collected through a random sampling process or should be a representative sample of the population of interest.

(3) Sample Size: Ideally, each cell in the contingency table (formed by cross-tabulating Genre and Rating) should have an expected frequency of at least 5. Our dataset is enough large to perform the Chi-Square Test.

(4) Categorical Data: The variables Genre and Rating are both categorical.

Step 2: Run the Chi-Square Test

- Hypotheses:
  - Null Hypothesis (H0): There is no significant relationship between Genre and Rating. In other words, the distribution of game ratings is independent of the game genres.
  - Alternative Hypothesis (H1): There is a significant relationship between Genre and Rating. In other words, the distribution of game ratings is not independent of the game genres.

- Significance level (α) : 0.05

```r
# Create a contingency table between Genre and Rating
contingency_table <- table(df$Genre, df$Rating)
contingency_table
```

```
##
##                   E E10+   M   T
##   Action        232  342 489 457
##   Adventure      76   45  59  50
##   Fighting        3   12  39 259
##   Misc          222  110   9 148
##   Platform      218  103   3  35
##   Puzzle        109   10   0   4
##   Racing        343   69  12 113
##   Role-Playing   48   57 119 278
##   Shooter        12   21 442 228
##   Simulation    156   21   4 120
##   Sports        697   71  10 117
##   Strategy       33   40  15  67
```

```r
# Perform a Chi-square Test of Independence
chi_square_result <- chisq.test(contingency_table)

# Define the significance level (alpha)
alpha <- 0.05

# Display the Chi-square Test result
chi_square_result
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table
## X-squared = 3551.9, df = 33, p-value < 2.2e-16
```

```r
# Hypothesis testing
if (chi_square_result$p.value < alpha) {
    result <- "Reject the null hypothesis.\nThere is a significant relationship between Genre a
} else {
    result <- "Fail to reject the null hypothesis.\nThere is no significant relationship betwe
}

cat(result)
```

```
## Reject the null hypothesis.
## There is a significant relationship between Genre and Rating.
```

- Test Statistic and P-Value:

- Test Statistic (X-squared): 3551.9
- P-Value: $<$ 2.2e-16

- Comment on Results:

- The Chi-Square Test for Independence produced a highly significant result with an extremely low p-value (less than 0.05), indicating strong statistical evidence that there is indeed a significant relationship between the Genre and Rating of video games.
- The p-value of $<$ 2.2e-16 is far below the significance level of 0.05, providing compelling evidence to reject the null hypothesis. This indicates that the distribution of video game Ratings is not independent of their Genres.
- While the statistical significance is clear, it's important to note that the strength of the relationship, as measured by the Chi-Square statistic, is substantial (3551.9). This suggests that the Genre and Rating of video games are strongly associated.

Conclusion:

Based on the Chi-Square Test for Independence, we can confidently conclude that there is a significant relationship between the Genre and Rating of video games. This statistical finding reinforces the idea that the Genre of a game plays a role in determining its Rating.

3.5 ANOVA

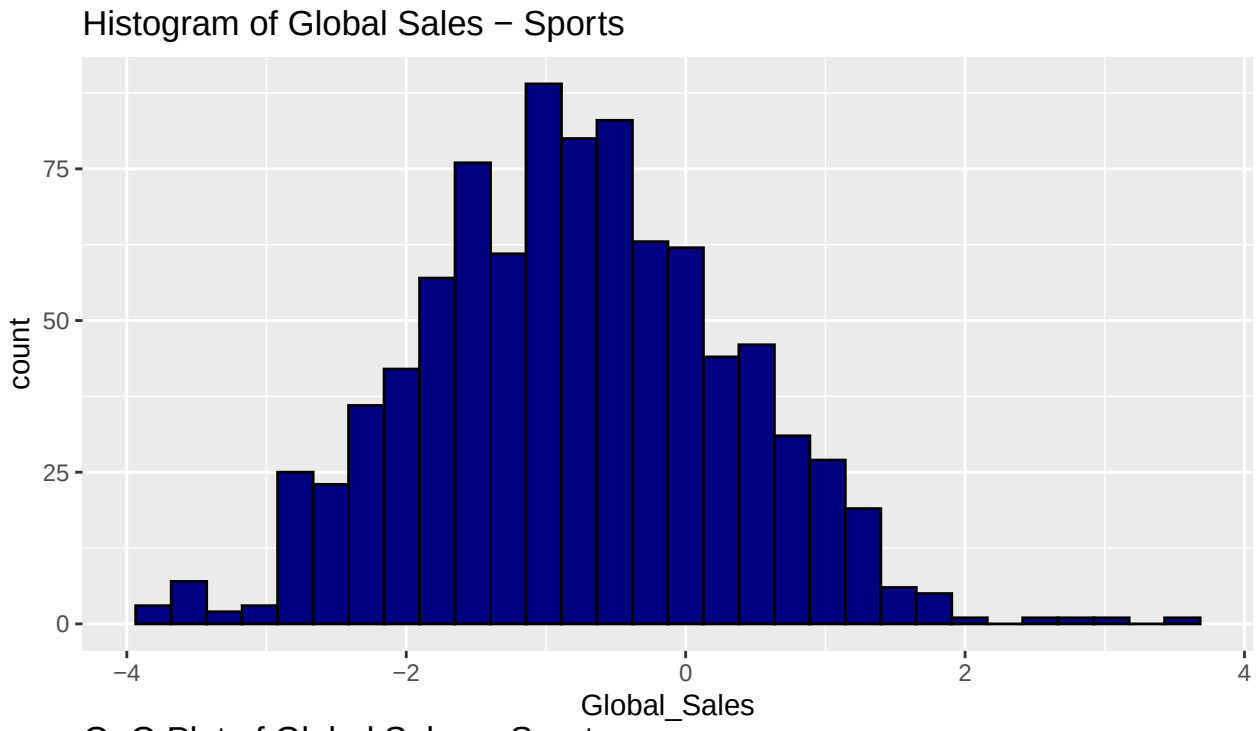
Question 6 : How do mean global sales figures vary across different video game genres? Specifically, do certain genres exhibit statistically significant differences in their average global sales compared to others?

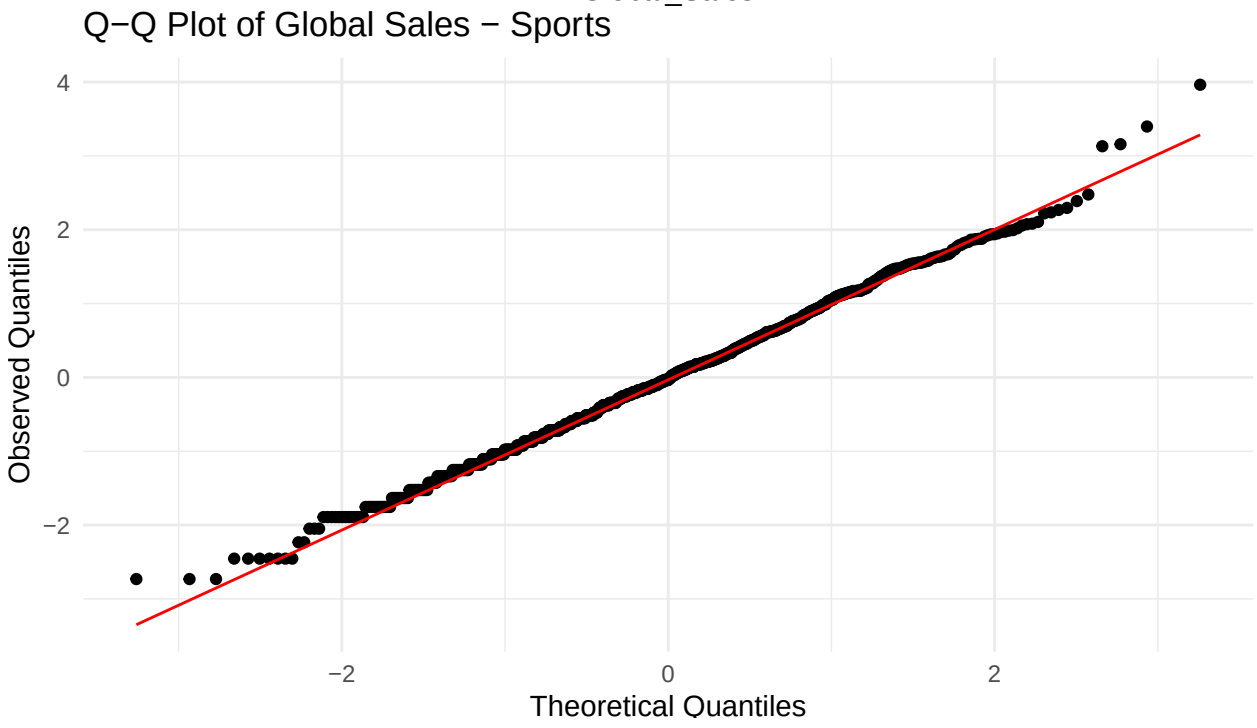
Step 1: Check for Assumptions & Considerations

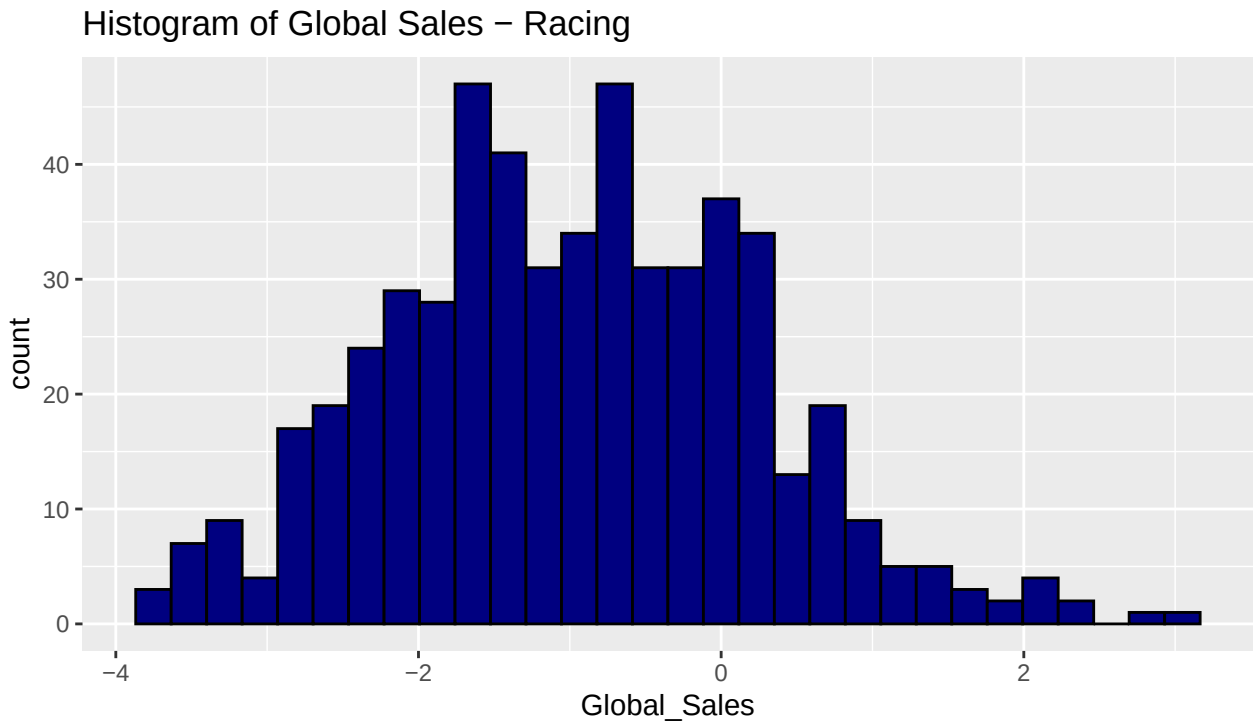
1) Normality with each group Histogram and Q-Q Plot for 'Global_Sales' within each 'Genre'
2) Equal variances across 'Genre' Box plot for checking variances across 'Genre'

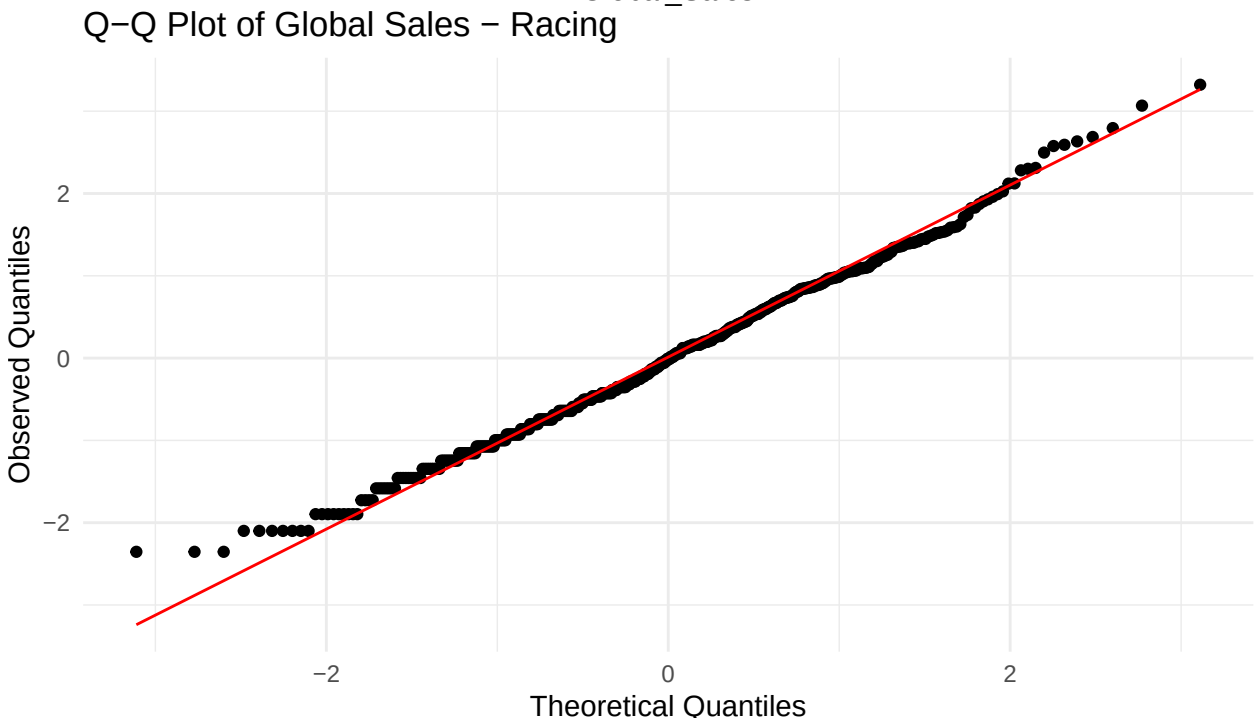
3) Independence between variable Assumed based on data collection methodology

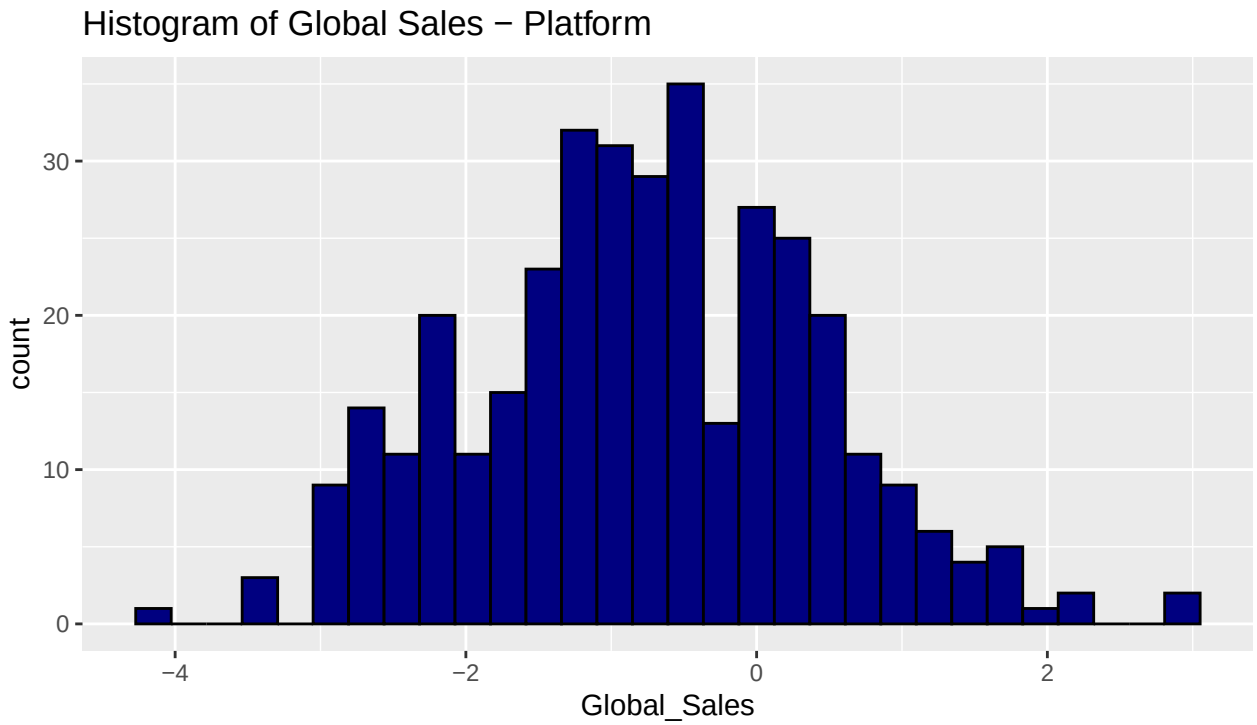
```r
# 1) Normality

genres <- unique(df$Genre)

for (genre in genres) {
    genre_data <- df %>%
        filter(Genre == genre)

    # Scale and center Global_Sales
    genre_data$Global_Sales_scaled <- scale(genre_data$Global_Sales)

    # Histogram
    p1 <- ggplot(genre_data, aes(x = Global_Sales)) + geom_histogram(bins = 30,
        fill = "navy", color = "black") + ggtitle(paste("Histogram of Global Sales -",
        genre)) + theme(plot.margin = unit(c(1, 0, 1, 0), "cm"))
    print(p1)

    # Q-Q Plot (with ggplot)
    p2 <- ggplot(genre_data, aes(sample = Global_Sales_scaled)) +
        stat_qq() + stat_qq_line(color = "red") + ggtitle(paste("Q-Q Plot of Global Sales -",
        genre)) + xlab("Theoretical Quantiles") + ylab("Observed Quantiles") +
        theme_minimal() + theme(plot.margin = unit(c(1, 0, 1,
        0), "cm"))
    print(p2)
}
```

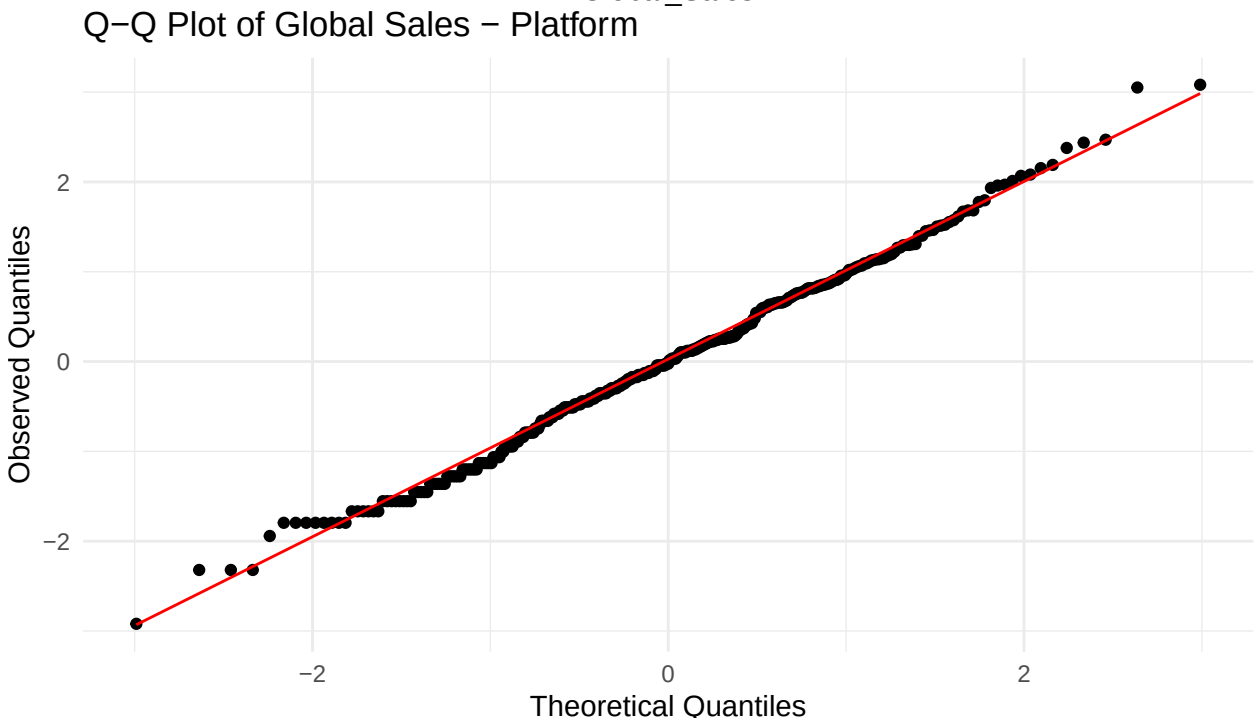

Histogram of Global Sales − Sports
count
75
50
25
0
−4
−2
0
2
4
Global_Sales

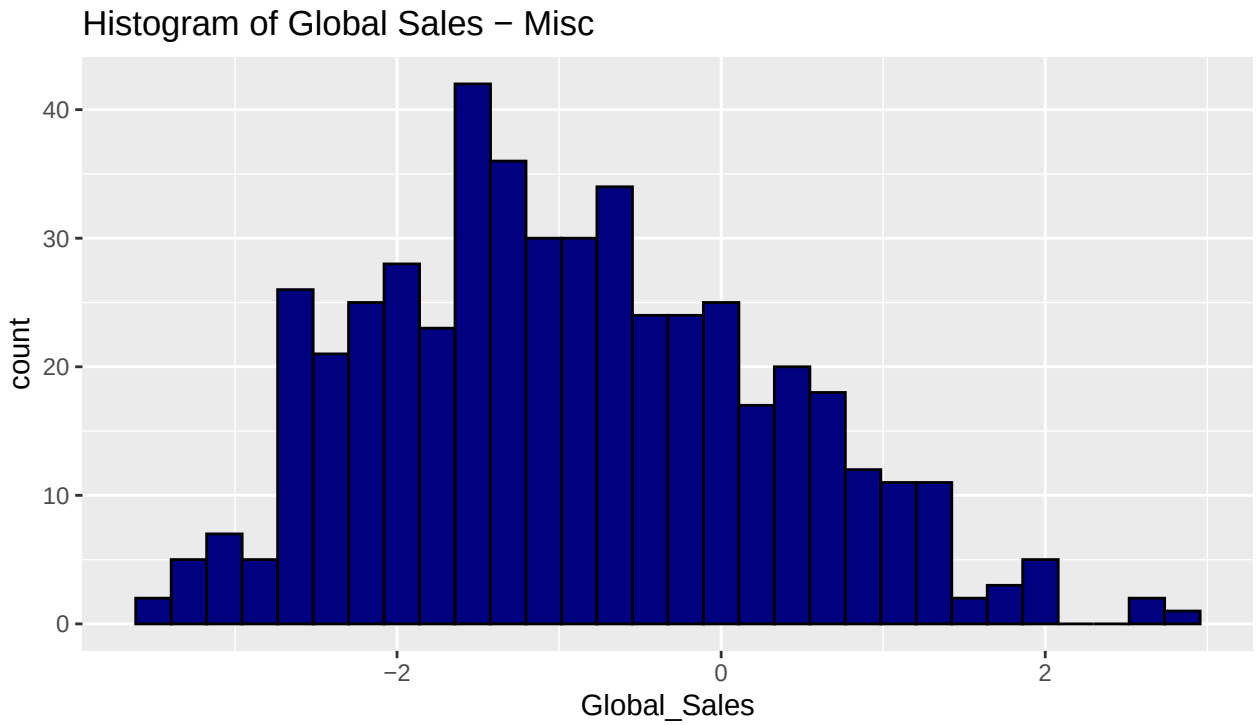

Q−Q Plot of Global Sales − Sports
Observed Quantiles
4
2
0
−2
−2
0
2
Theoretical Quantiles

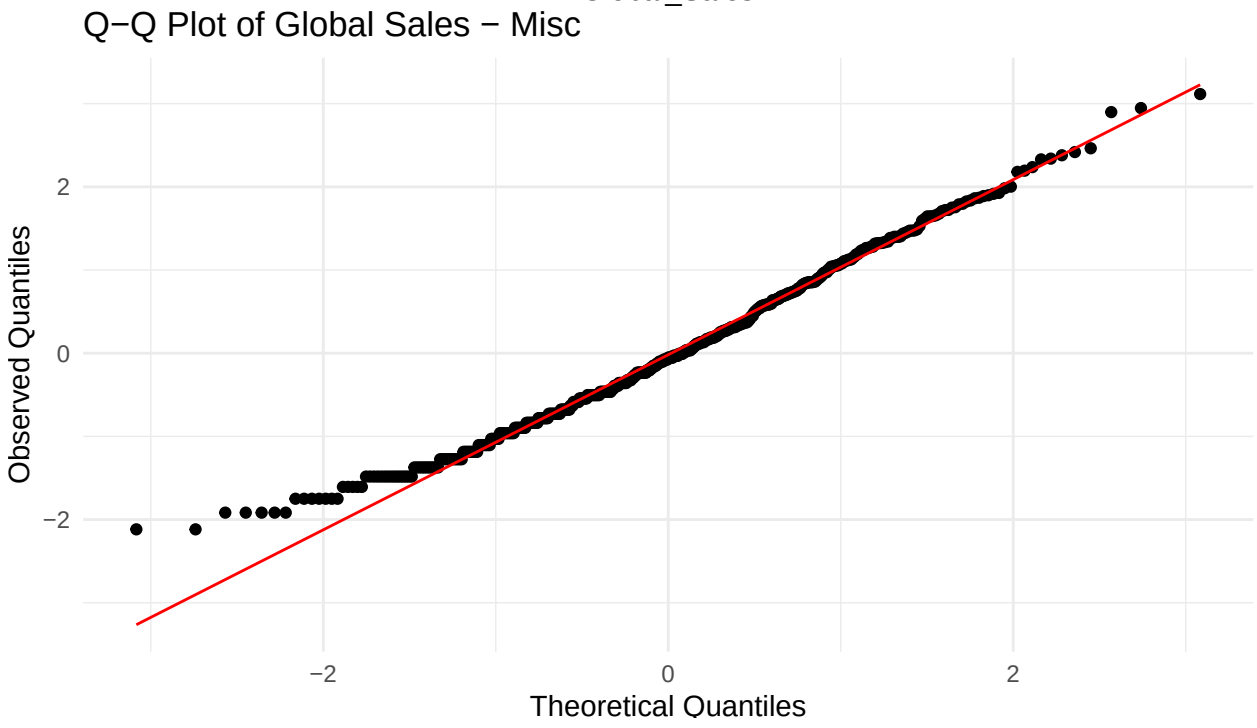

Histogram of Global Sales − Racing
count
40
30
20
10
0
−4
−2
0
2
Global_Sales

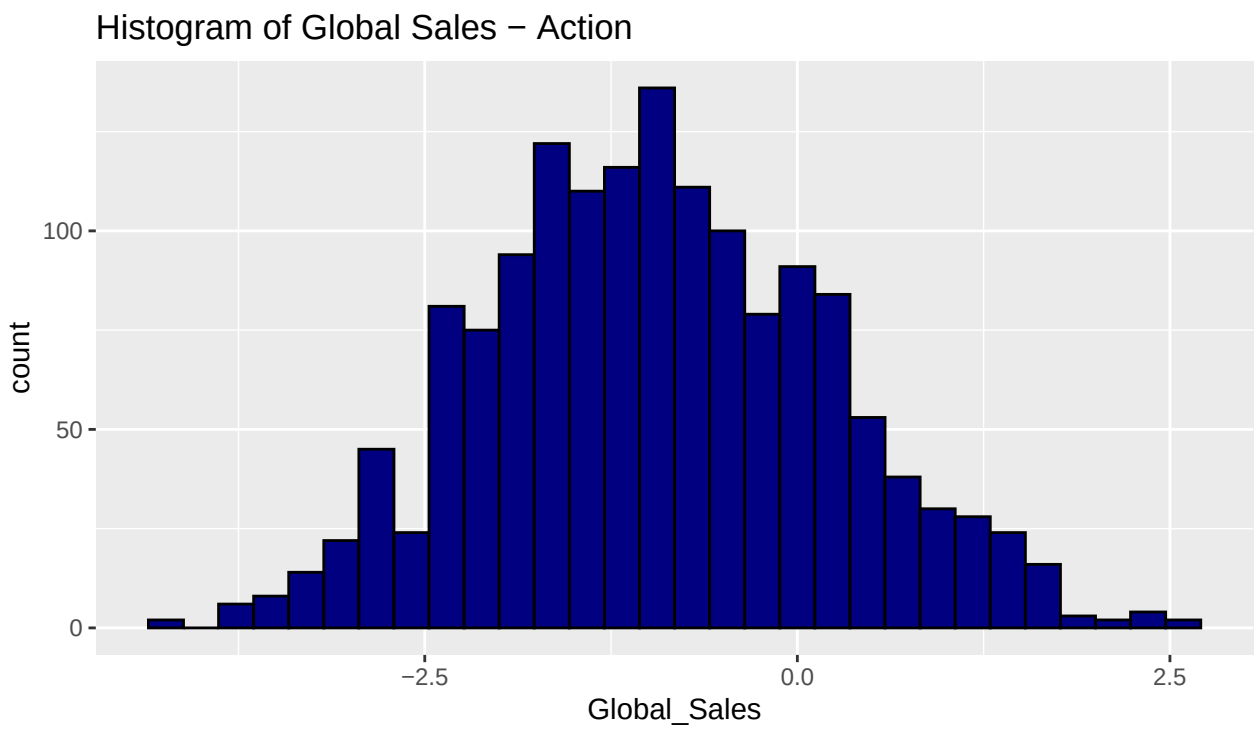

Q−Q Plot of Global Sales − Racing
Observed Quantiles
2
0
−2
−2
0
2
Theoretical Quantiles

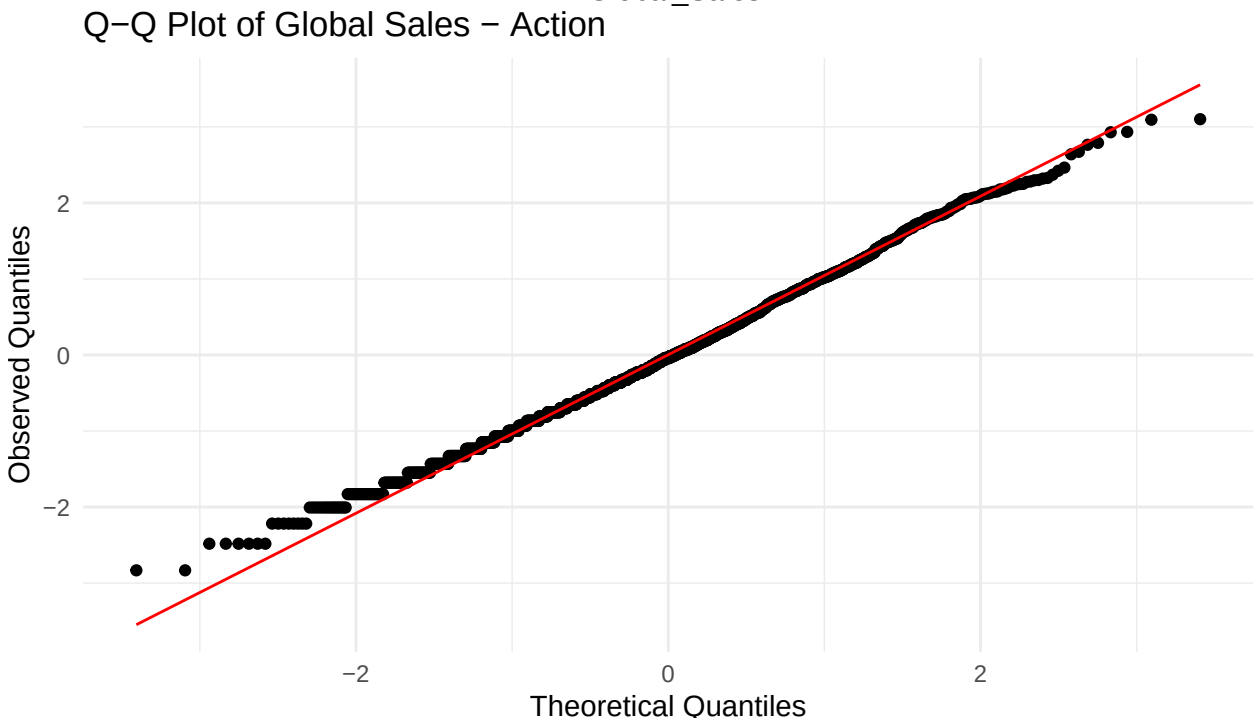

Histogram of Global Sales − Platform
count
30
20
10
0
−4
−2
0
2
Global_Sales

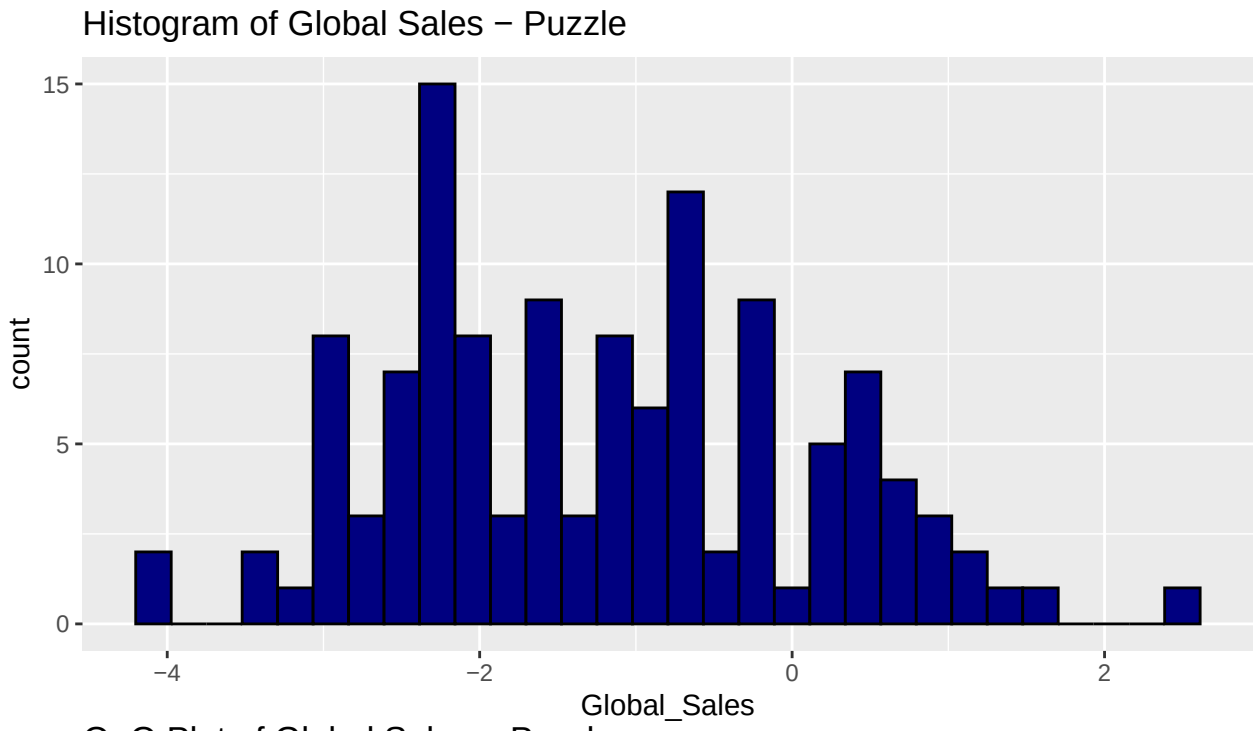

Q−Q Plot of Global Sales − Platform
Observed Quantiles
2
0
−2
−2
0
2
Theoretical Quantiles

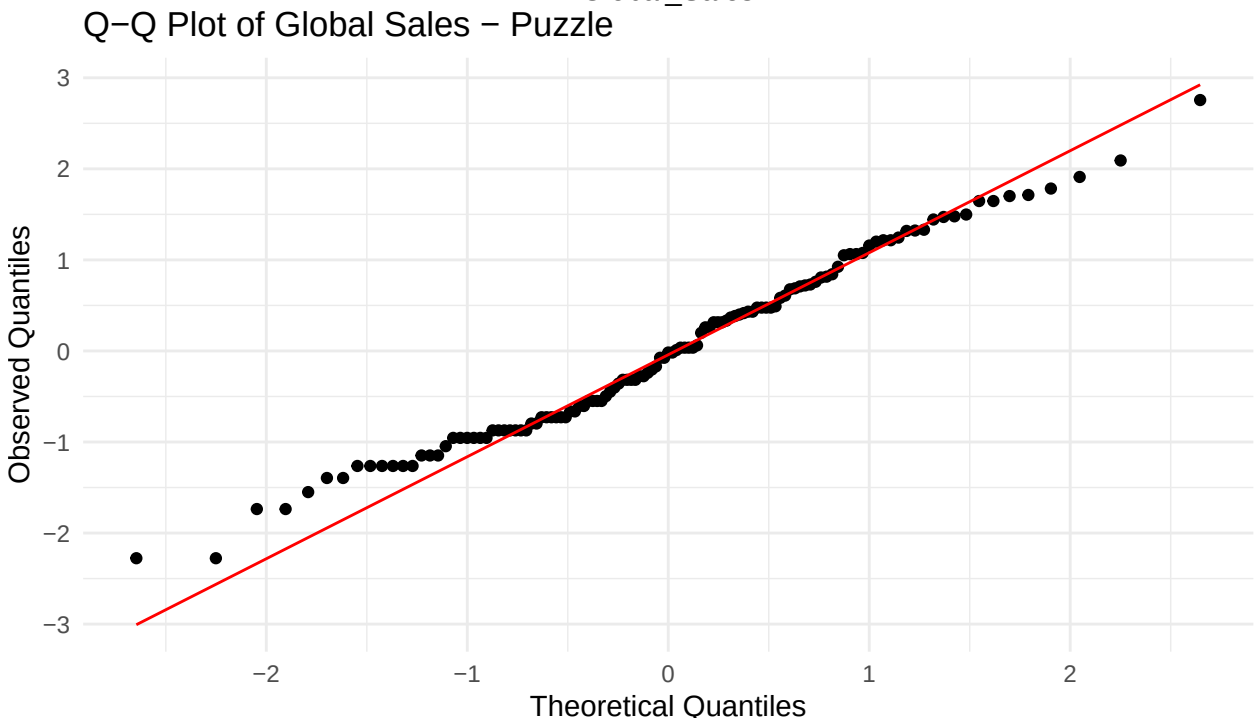

Histogram of Global Sales − Misc
count
40
30
20
10
0
−2
0
2
Global_Sales

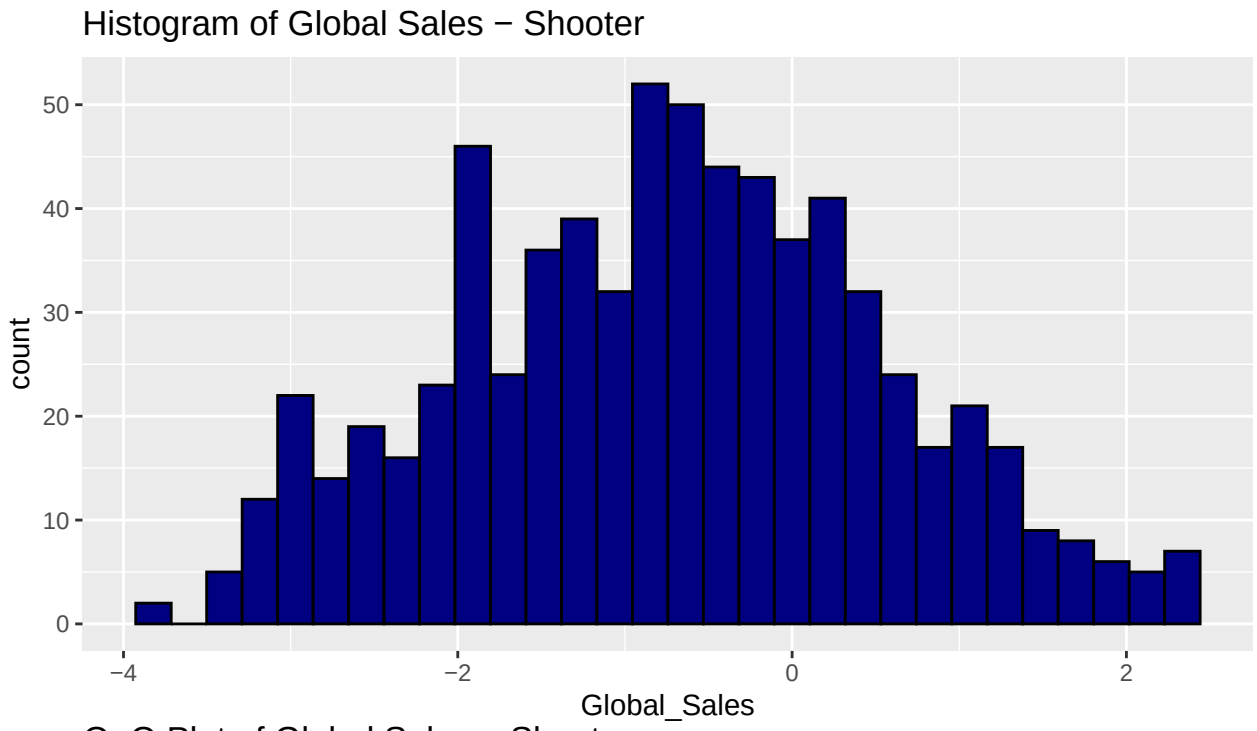

Q−Q Plot of Global Sales − Misc
Observed Quantiles
2
0
−2
−2
0
2
Theoretical Quantiles

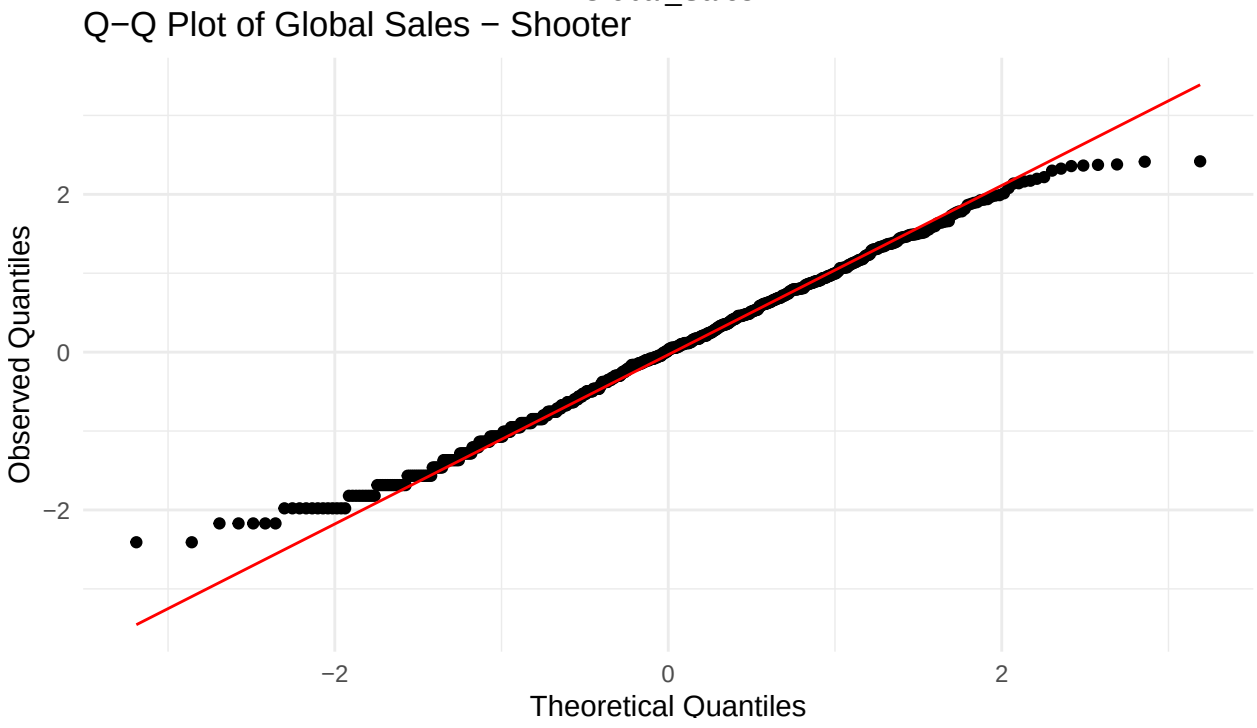

Histogram of Global Sales − Action
count
100
50
0
−2.5
0.0
2.5
Global_Sales

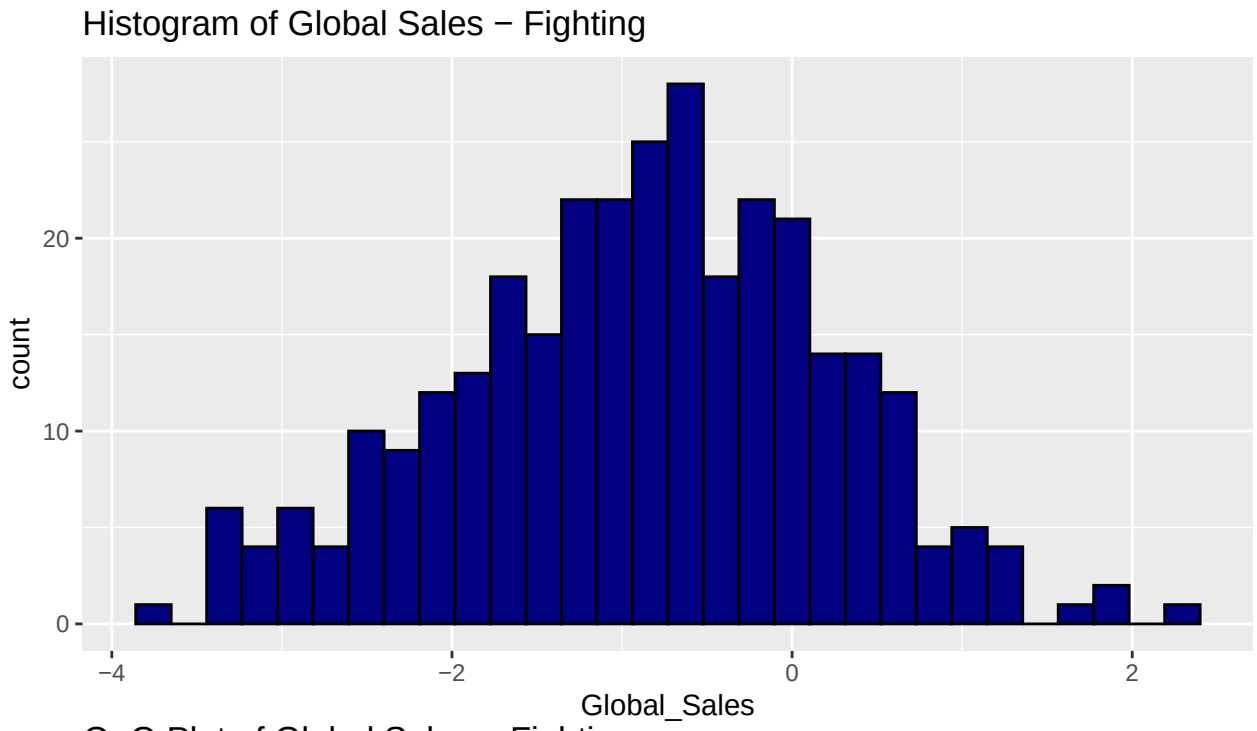

Q−Q Plot of Global Sales − Action
Observed Quantiles
2
0
−2
−2
0
2
Theoretical Quantiles

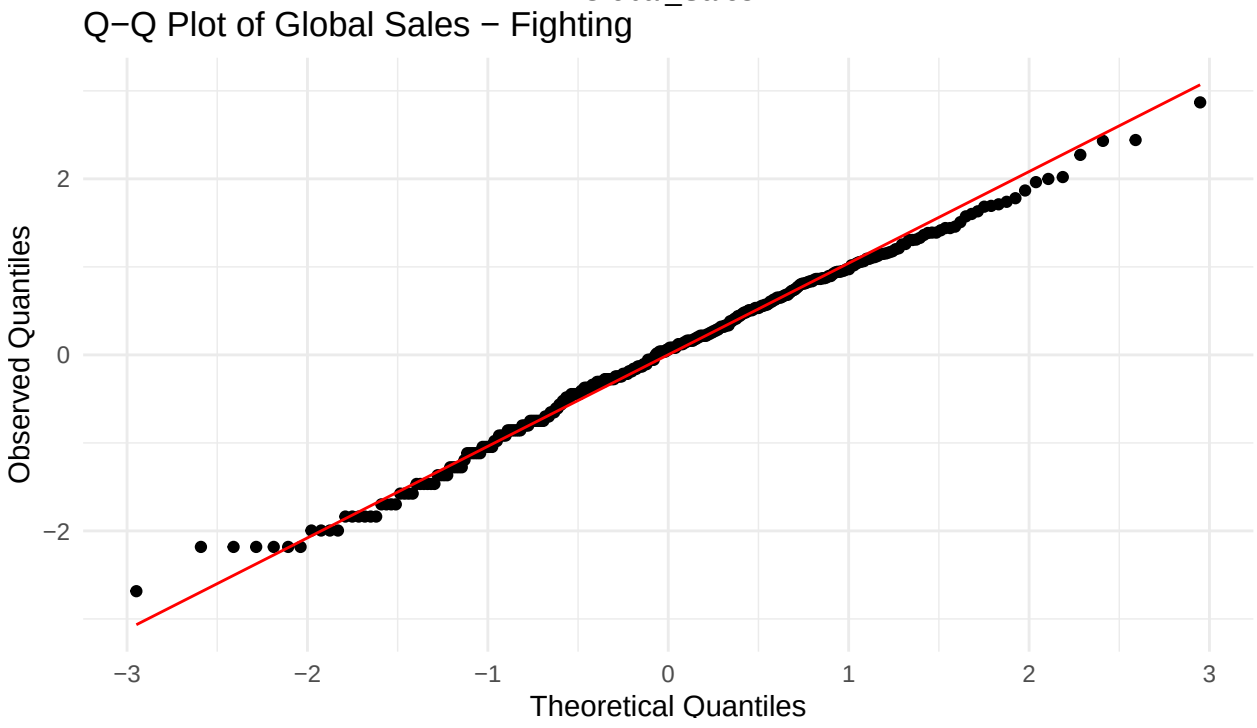

Histogram of Global Sales − Puzzle
count
15
10
5
0
−4
−2
0
2
Global_Sales

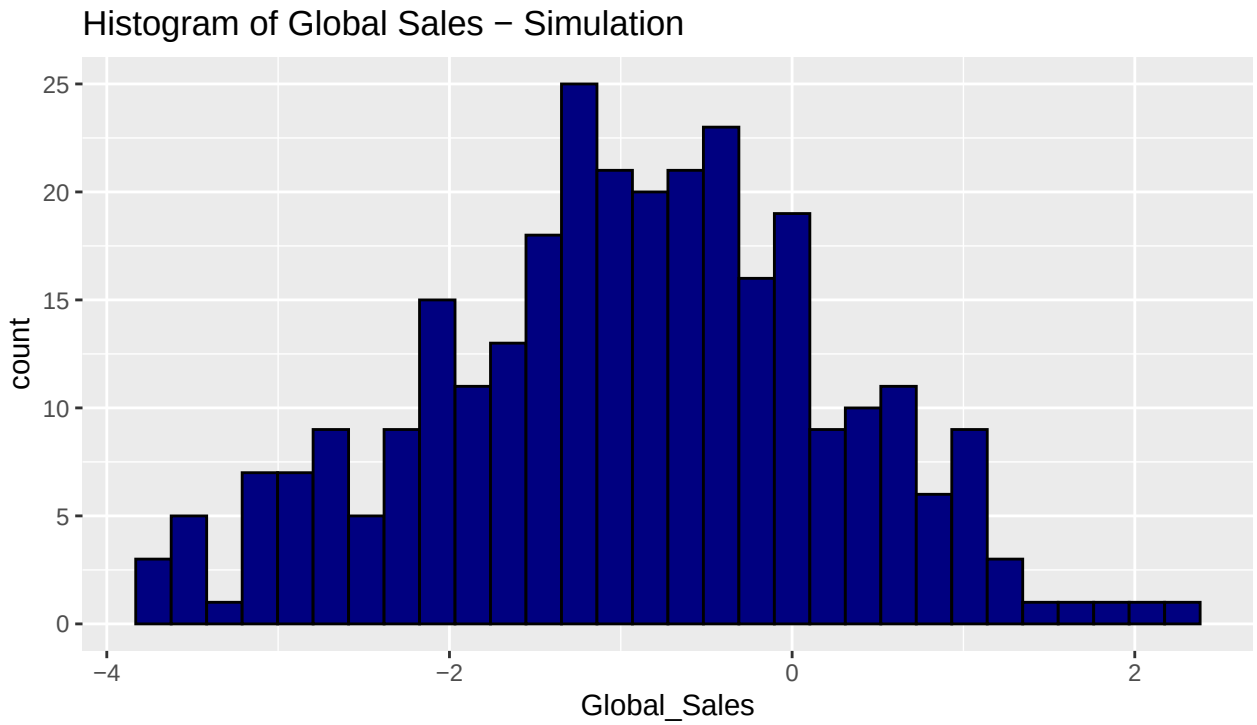

Q−Q Plot of Global Sales − Puzzle
Observed Quantiles
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
Theoretical Quantiles

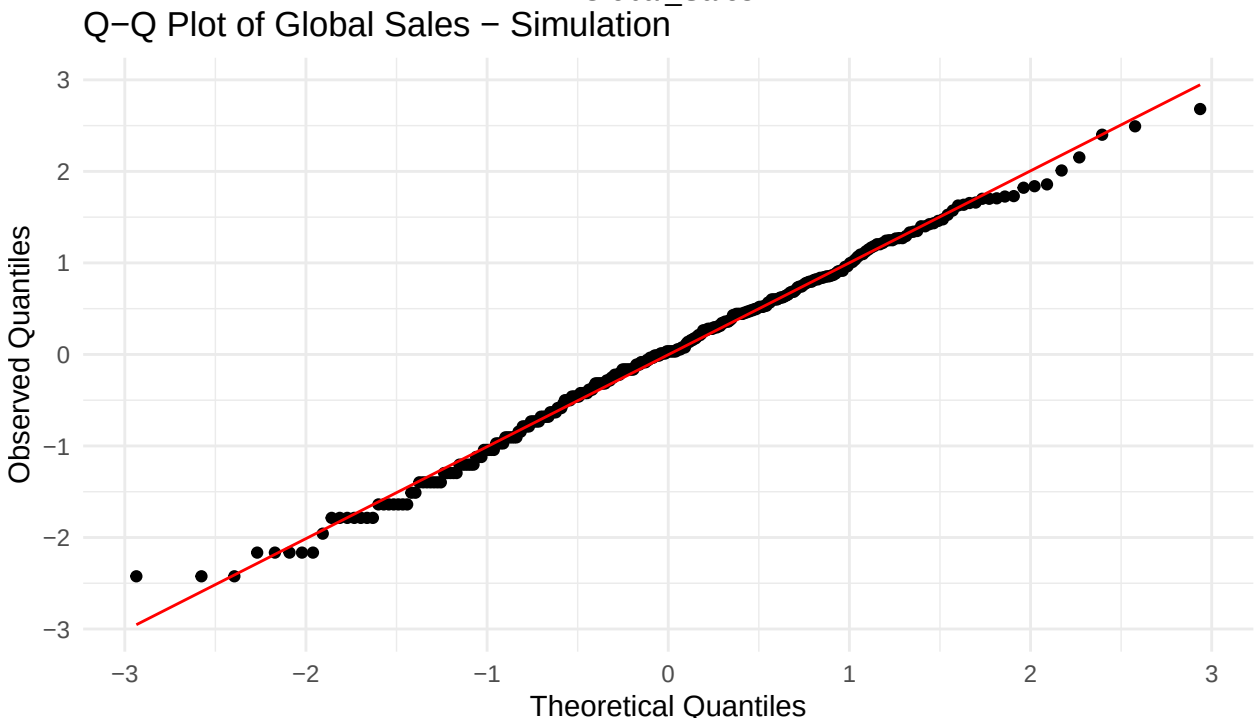

Histogram of Global Sales − Shooter
count
50
40
30
20
10
0
−4
−2
0
2
Global_Sales

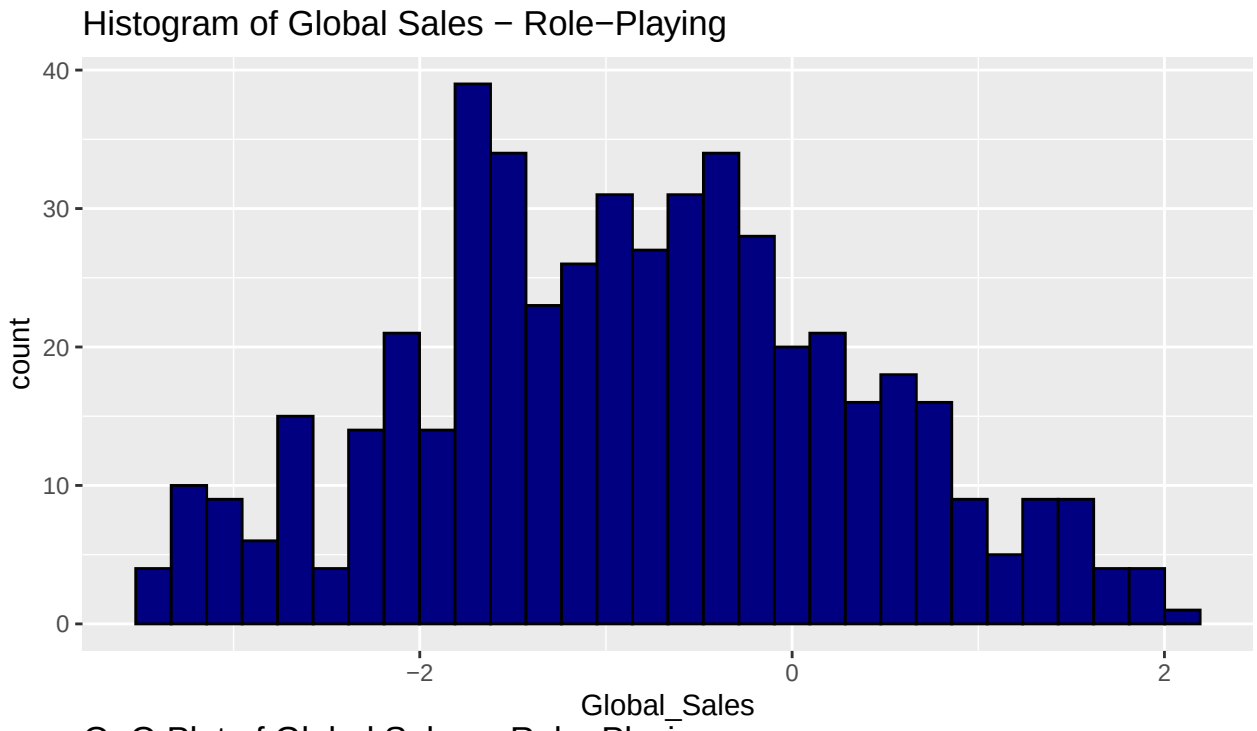

Q−Q Plot of Global Sales − Shooter
Observed Quantiles
2
0
−2
−2
0
2
Theoretical Quantiles

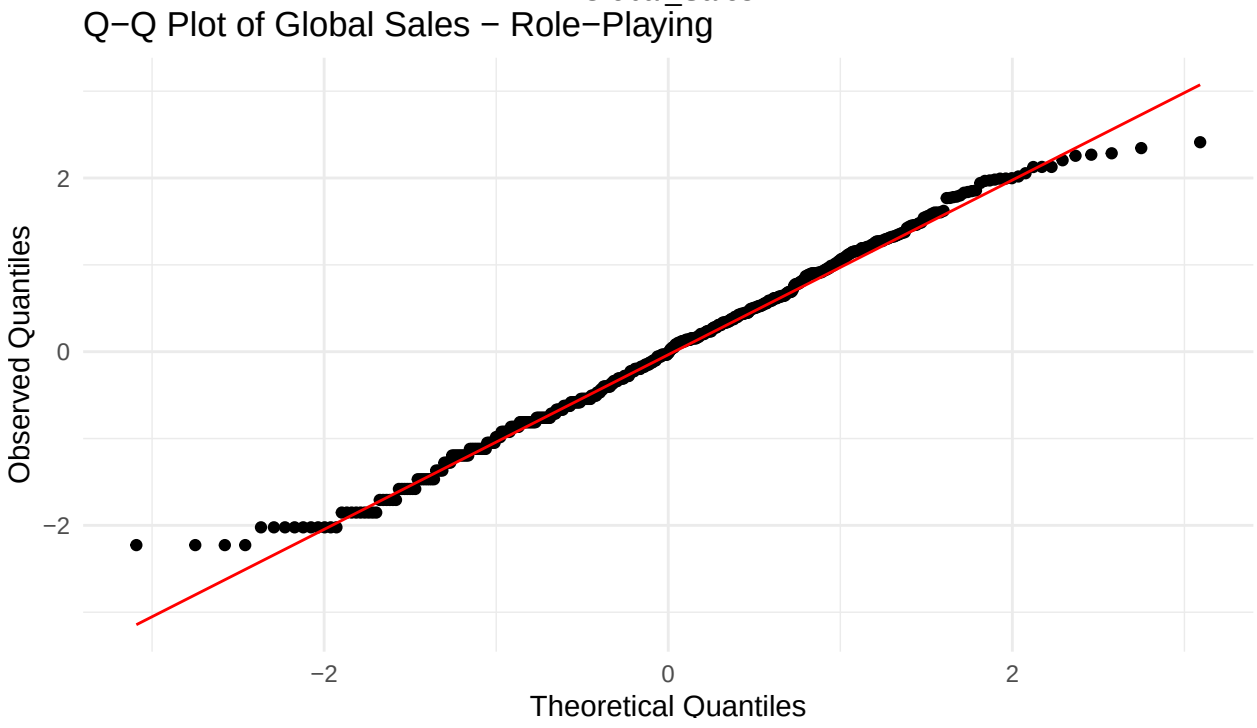

Histogram of Global Sales − Fighting
count
20
10
0
−4
−2
0
2
Global_Sales

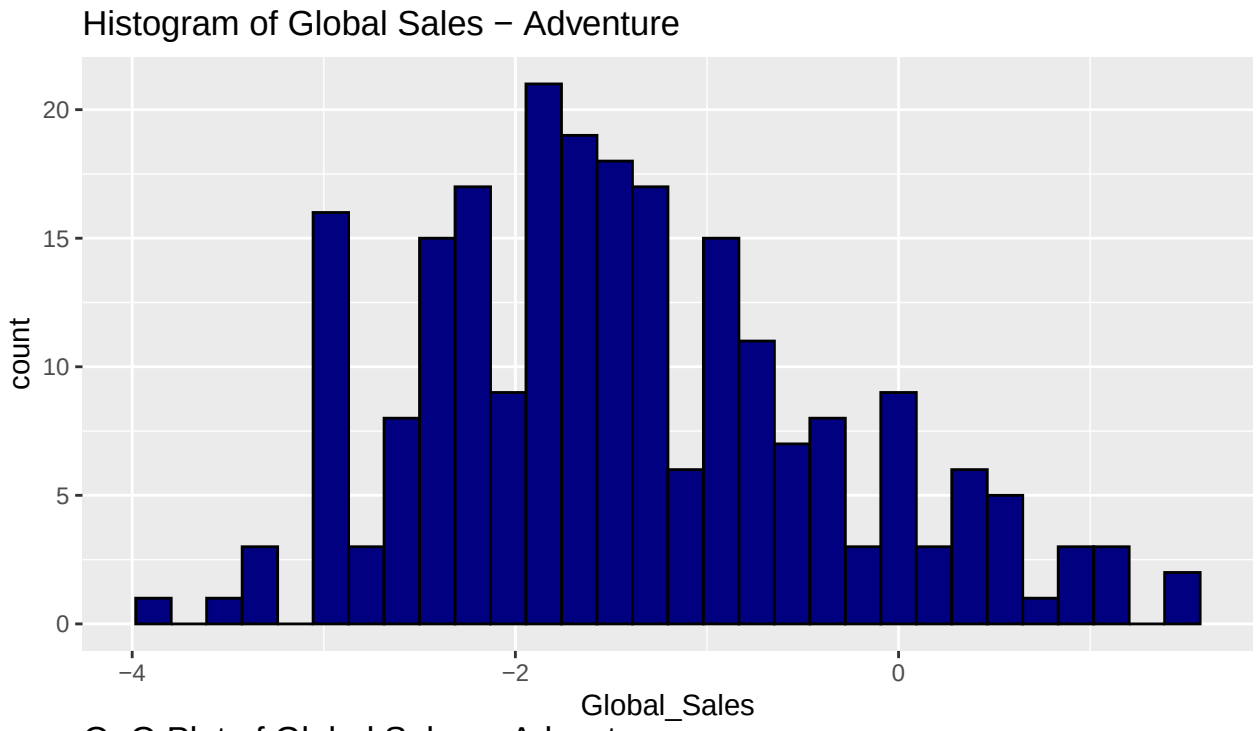

Q−Q Plot of Global Sales − Fighting
Observed Quantiles
2
0
−2
−3
−2
−1
0
1
2
3
Theoretical Quantiles

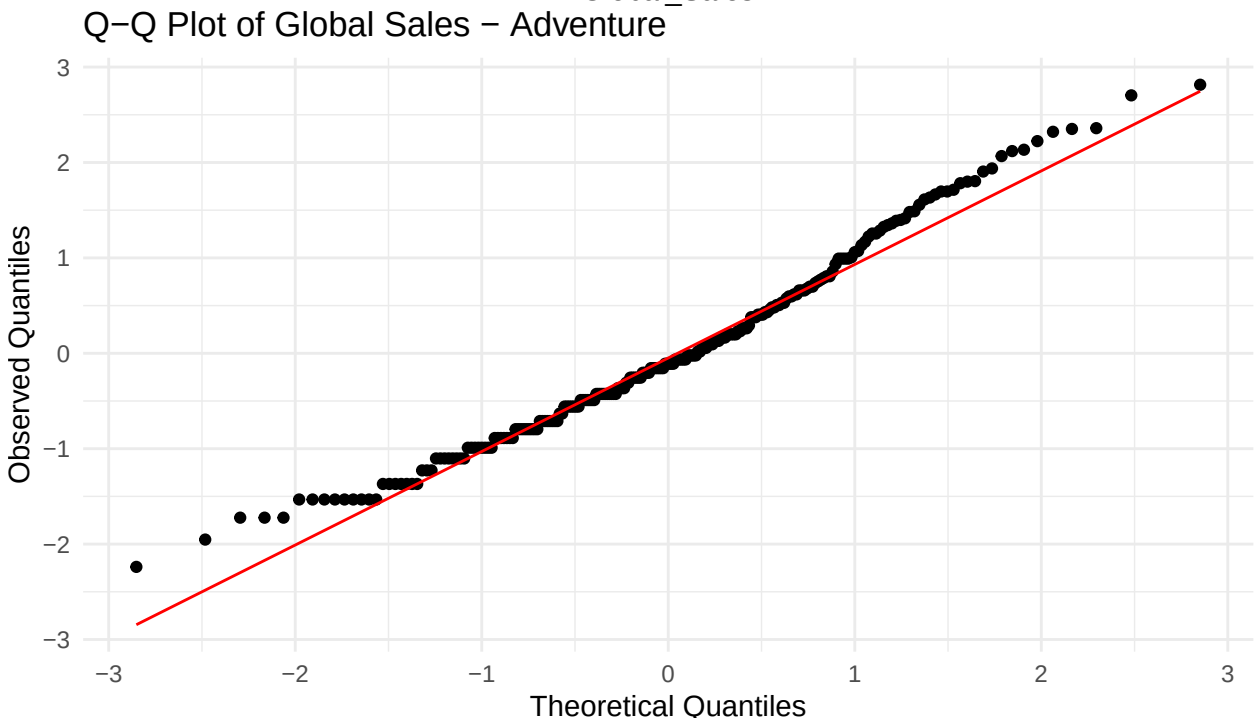

Histogram of Global Sales − Simulation
count
Global_Sales
25
20
15
10
5
0
−4
−2
0
2

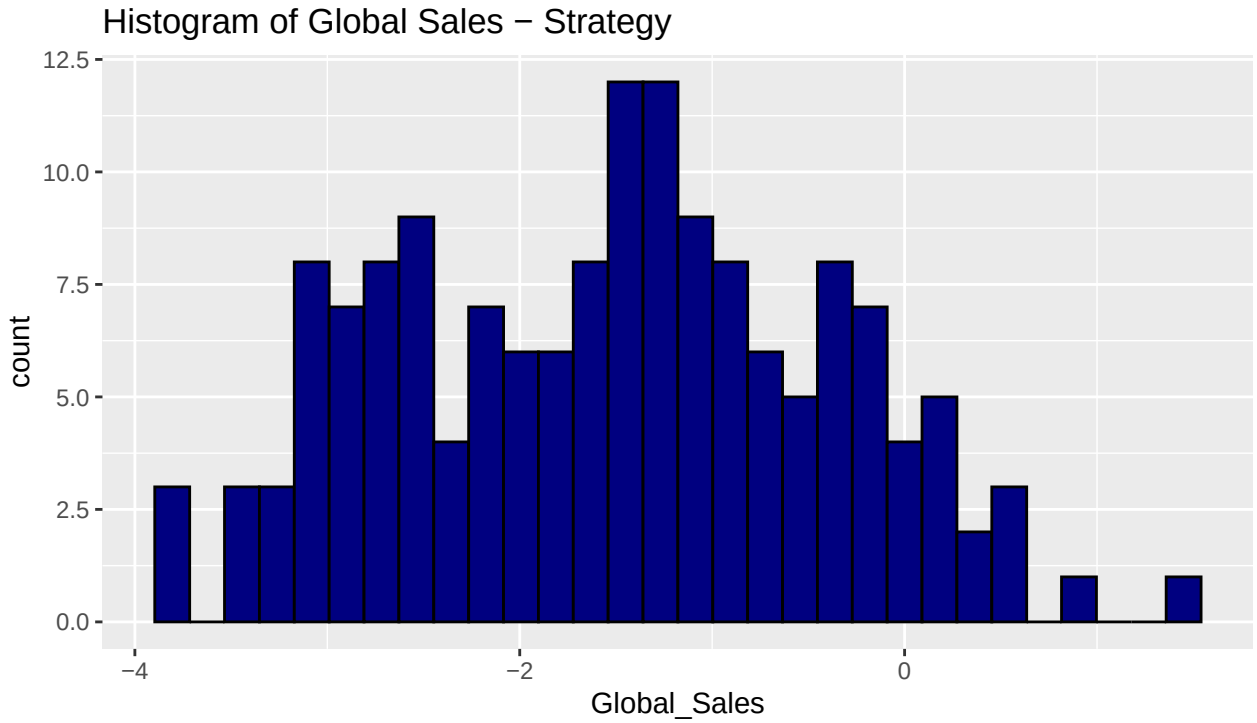

Q−Q Plot of Global Sales − Simulation
Observed Quantiles
Theoretical Quantiles
3
2
1
0
−1
−2
−3

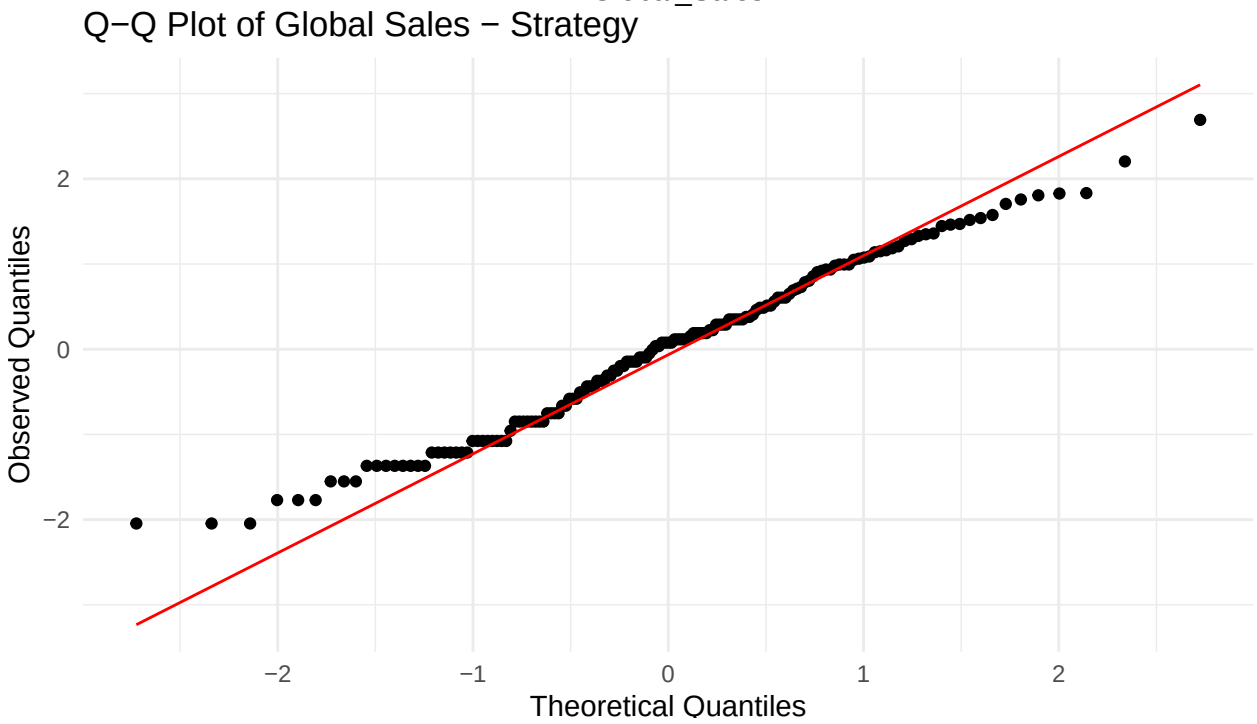

Histogram of Global Sales − Role−Playing
count
Global_Sales
40
30
20
10
0
−2
0
2


Q−Q Plot of Global Sales − Role−Playing
Observed Quantiles
Theoretical Quantiles
2
0
−2
−2
0
2


Histogram of Global Sales − Adventure
count
Global_Sales
20
15
10
5
0
−4
−2
0


Q−Q Plot of Global Sales − Adventure
Observed Quantiles
Theoretical Quantiles
3
2
1
0
−1
−2
−3

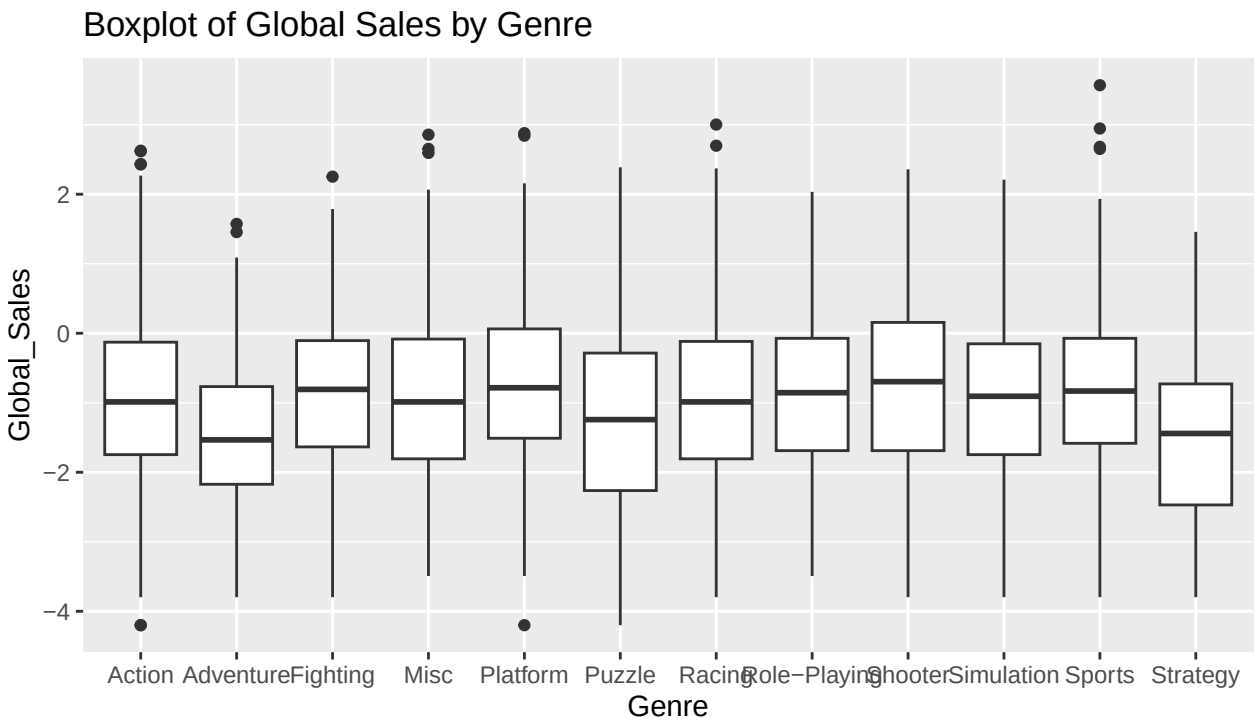




Note: The 'Global_Sales' data were scaled and centered for Q-Q plot generation to facilitate a direct comparison with the standard normal distribution. This step aligns with the default behavior of the qqnorm function and is necessary when using ggplot2 for such plots. It does not alter the interpretation of the raw data used elsewhere.

```
# 2) Equal variances

ggplot(df, aes(x = Genre, y = Global_Sales)) + geom_boxplot() +
```

```
    ggtitle("Boxplot of Global Sales by Genre") + theme(plot.margin = unit(c(1,
    0, 1, 0), "cm"))
```



- Assumptions for ANOVA Test:

- Normality: Each group (Genre) must have a normally distributed population. Assumption checked using histograms and Q-Q plots for each genre.
- Homogeneity of Variances (Homoscedasticity): The variances among groups are mostly equal. Assumption checked using box plots.
- Independence: Each observation is independent of the others. Assumed based on data collection methodology.

One more consideration, we need to check if the sample size within each genre group is sufficient to provide reliable results.

```
table(df$Genre)
```

```
##
##       Action    Adventure     Fighting         Misc     Platform       Puzzle
##         1520          230          313          489          359          123
##       Racing Role-Playing      Shooter   Simulation       Sports     Strategy
##          537          502          703          301          895          155
```

Overall, our dataset seems well-suited for a one-way ANOVA analysis. The larger and medium categories provide a strong basis for comparison. For smaller categories, let's keep in mind to be

cautious with interpretation, especially if conducting post-hoc tests or drawing conclusions about these specific genres.

After checking all the assumptions and considerations, it is good for the analysis. Now, we may run ANOVA.

Step 2: Run ANOVA

- Hypotheses:
  - Null Hypothesis (H0): The means of Global_Sales are the same across different genres.
  - Alternative Hypothesis (H1): At least one genre has a mean Global_Sales that is different from the others.
- Significance level (α): 0.05

```
# One-way ANOVA
anova_result <- aov(Global_Sales ~ Genre, data = df)
summary(anova_result)
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## Genre         11    183  16.593   12.12 <2e-16 ***
## Residuals   6115   8373   1.369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Test Statistic and P-Value:

- F-Value: 12.23
- P-Value: < 2e-16
- Comment on Results:
  - Since the p-value is less than the significance level ($\alpha = 0.05$), we reject the null hypothesis. This suggests that there is a statistically significant difference in the mean Global_Sales across different genres.
  - However, it's important to note that while ANOVA indicates that there is a difference, it does not tell us which specific genres are different from each other. For that, we need to conduct a post-hoc test like Paired t-tests with Bonferroni correction to determine which specific genres differ from each other. We apply Bonferroni correction to mitigate the risk of Type I errors (false positives) due to multiple comparisons.

Step 3: Conduct Post-hoc Analysis (Paired t-tests with Bonferroni correction)

- Hypotheses:
  - Null Hypothesis (H0): There is no significant difference in mean Global Sales between each pair of video game genres.
  - Alternative Hypothesis (H1): There is a significant difference in mean Global Sales between at least one pair of video game genres.
- Significance level (α): 0.05

```r
# Run pairwise t-tests with Bonferroni correction
pairwise_result <- pairwise.t.test(df$Global_Sales, df$Genre,
    p.adj = "bonferroni")

print(pairwise_result)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  df$Global_Sales and df$Genre
##
##              Action  Adventure Fighting Misc    Platform Puzzle  Racing
## Adventure    6.0e-07 -         -        -       -        -       -
## Fighting     1.00000 5.5e-06   -        -       -        -       -
## Misc         1.00000 8.3e-06   1.00000  -       -        -       -
## Platform     0.46261 1.6e-09   1.00000  1.00000 -        -       -
## Puzzle       0.73659 1.00000   0.34889  0.78384 0.00993  -       -
## Racing       1.00000 0.00012   1.00000  1.00000 0.37335  1.00000 -
## Role-Playing 1.00000 3.8e-08   1.00000  1.00000 1.00000  0.08164 1.00000
## Shooter      0.00334 5.0e-13   1.00000  0.27062 1.00000  0.00101 0.01139
## Simulation   1.00000 0.00027   1.00000  1.00000 1.00000  1.00000 1.00000
## Sports       0.16394 3.5e-11   1.00000  1.00000 1.00000  0.00962 0.25227
## Strategy     2.0e-07 1.00000   8.2e-07  1.4e-06 5.0e-10  1.00000 1.7e-05
##              Role-Playing Shooter Simulation Sports
## Adventure    -            -       -          -
## Fighting     -            -       -          -
## Misc         -            -       -          -
## Platform     -            -       -          -
## Puzzle       -            -       -          -
## Racing       -            -       -          -
## Role-Playing -            -       -          -
## Shooter      1.00000      -       -          -
## Simulation   1.00000      0.41424 -          -
## Sports       1.00000      1.00000 1.00000    -
## Strategy     1.1e-08      8.2e-13 3.3e-05    3.7e-11
##
## P value adjustment method: bonferroni
```

This test revealed specific genres that have statistically significant differences in their Global_Sales. Here are some key findings and their implications.

- Significant Differences Between Genres:
  - Significant differences in Global_Sales were noted between genres. For example, Action vs. Adventure showed a significant sales difference (p = 5.0e-07), highlighting divergent sales performances in these genres.
- No Significant Differences in Some Comparisons:
  - Certain genre comparisons, like Action vs. Fighting, revealed no significant difference in sales (p = 1.00000), indicating similar sales performance in these categories.
- Bonferroni Correction's Conservatism:
  - We applied the conservative Bonferroni correction for adjusting p-values, balancing the risk of false positives against false negatives. The significant differences found are considered robust indicators of true sales performance disparities across genres.
- Implications:
  - This analysis offers crucial insights for strategic decisions in game development and marketing. Recognizing which genres outperform or under perform in sales guides targeted strategies, investment decisions, and market positioning.

Conclusion:

Our analysis of Global_Sales across video game genres led to significant insights. The ANOVA test revealed differences in sales among genres, and the subsequent pairwise t-tests with Bonferroni correction pinpointed specific genres with notably different sales performances.

Key takeaways include the identification of genres like Action and Adventure with statistically significant sales differences, guiding targeted strategies for game development and marketing. Conversely, similarities in sales between genres like Action and Fighting suggest potential areas for further investigation or strategy refinement.

3.6 Inference about variance(s)

Question 7 : Is there significant difference in the variances of Global_Sales for two Genres - Action and Sports?

To determine if there is a significant difference in the variances of two numerical columns: Global_Sales for two Genres - Action and Sports. (Note: These two categories of Genre were

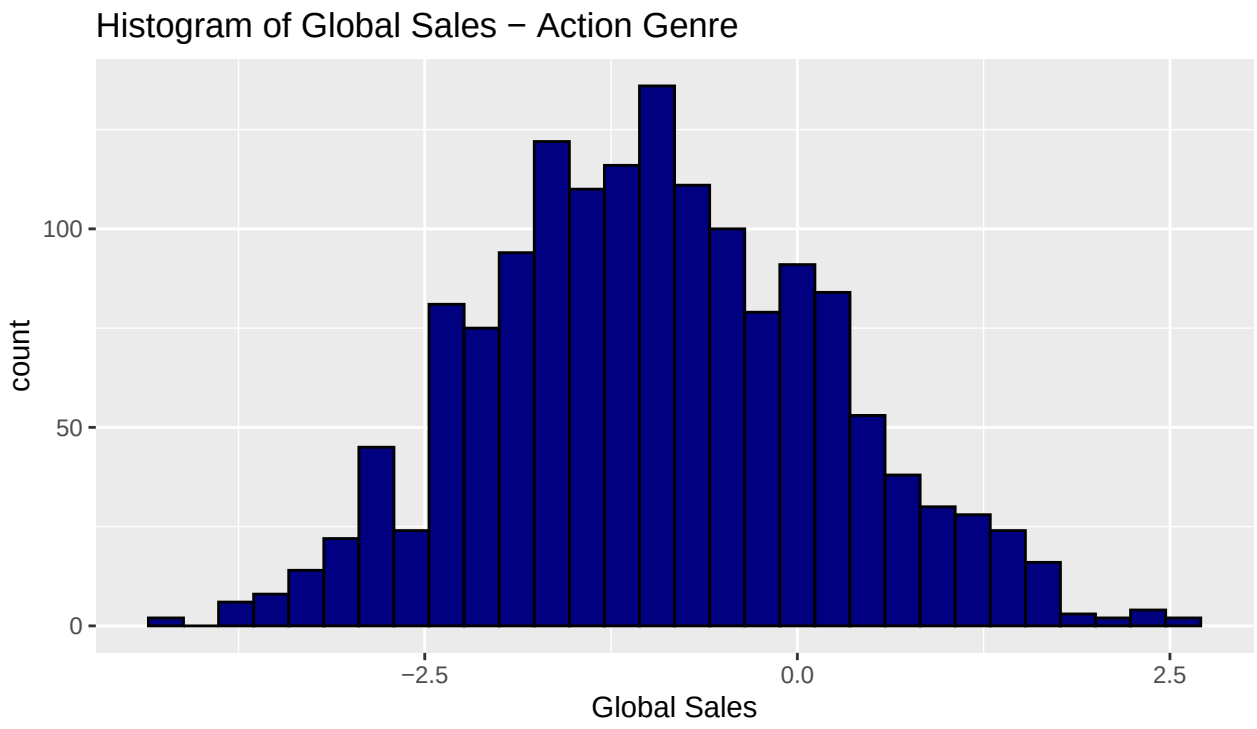
chosen as they were the two most frequently occurring ones.)

Step 1: Check for Assumptions & Considerations

Let s_1^2 = variance of Global_Sales for Genre = Action, and s_2^2 = variance of Global_Sales for Genre = Sports.

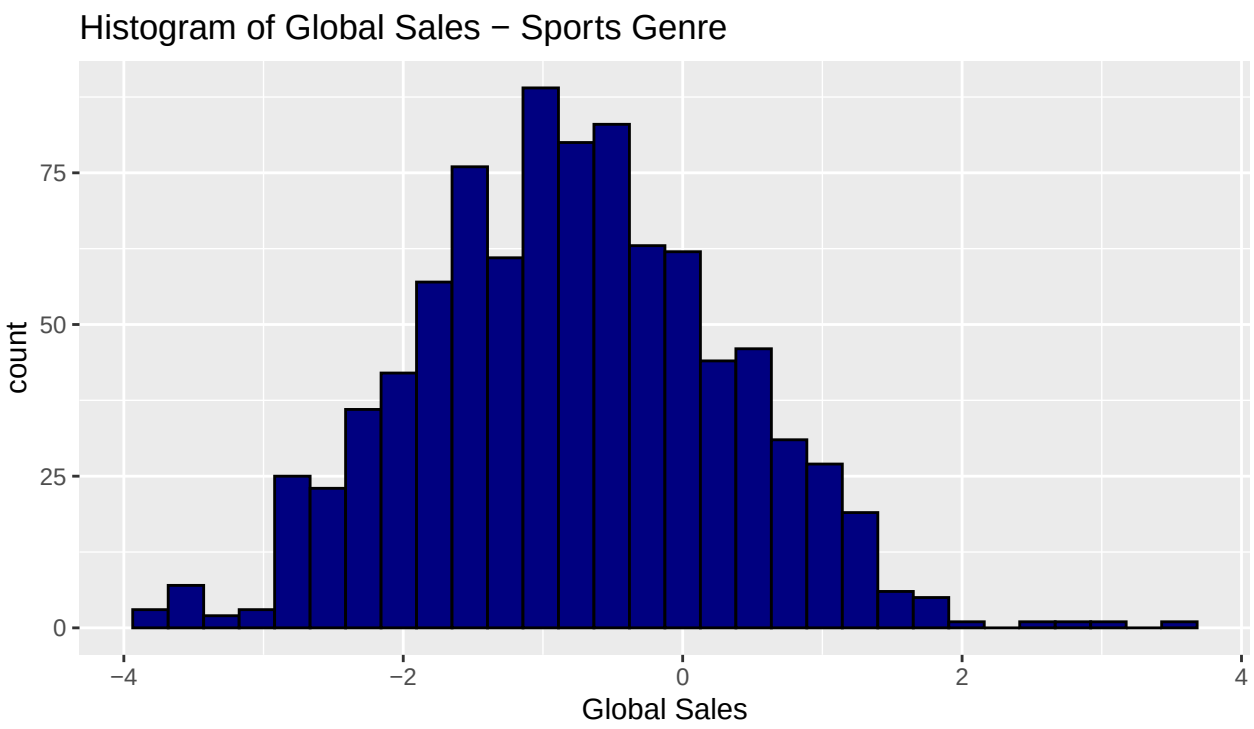
Checking Assumptions: Both groups follow a normal distribution.

```
global_action <- df$Global_Sales[df$Genre == "Action"]
global_sports <- df$Global_Sales[df$Genre == "Sports"]
global_action_df <- data.frame(Global_Sales = global_action)
global_sports_df <- data.frame(Global_Sales = global_sports)
# Histogram for global_action

p1 <- ggplot(global_action_df, aes(x = Global_Sales)) + geom_histogram(bins = 30,
    fill = "navy", color = "black") + xlab("Global Sales") +
    ggtitle("Histogram of Global Sales - Action Genre") + theme(plot.margin = unit(c(1,
    0, 1, 0), "cm"))
print(p1)
```




```
# Histogram for global_sports

p2 <- ggplot(global_sports_df, aes(x = Global_Sales)) + geom_histogram(bins = 30,
    fill = "navy", color = "black") + xlab("Global Sales") +
```

```
    ggtitle("Histogram of Global Sales - Sports Genre") + theme(plot.margin = unit(c(1,
    0, 1, 0), "cm"))
print(p2)
```



Both groups are normally distributed.

Step 2: Run the F-test

- Hypotheses:
  - Null Hypothesis (H0): $H_0 : s_1^2 = s_2^2$ or $s_1^2 - s_2^2 = 0$
  - Alternative Hypothesis (H1): $H_1 : s_1^2 \neq s_2^2$ or $s_1^2 - s_2^2 \neq 0$
- Significance level (α) : 0.05

Performing F-test to compare two variances:

```
var.test(global_action, global_sports, ratio = 1, alternative = "two.sided",
    conf.level = 0.95)
```

```
##
##  F test to compare two variances
##
## data:  global_action and global_sports
## F = 1.0935, num df = 1519, denom df = 894, p-value = 0.1368
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
```

```
##  0.9719822 1.2280011
## sample estimates:
## ratio of variances
##           1.093498
```

- Test Statistic and P-Value:

- F-Value: 1.0935
- P-Value: 0.1368

Comment on Results

Since the p-value $= 0.1368 > \alpha = 0.05$, we FAIL TO REJECT the null hypothesis that the variances of global sales for genre = Action is equal to global sales from genre = Sports.

Conclusion: Thus, we conclude that the variances are not different.

Confidence Interval:

```
conf_interval <- var.test(global_action, global_sports, ratio = 1,
    alternative = "two.sided", conf.level = 0.95)$conf.int
conf_interval
```

```
## [1] 0.9719822 1.2280011
## attr(,"conf.level")
## [1] 0.95
```

As seen in the F-test result, the 95% two-sided confidence interval is (0.9719822, 1.2280011).

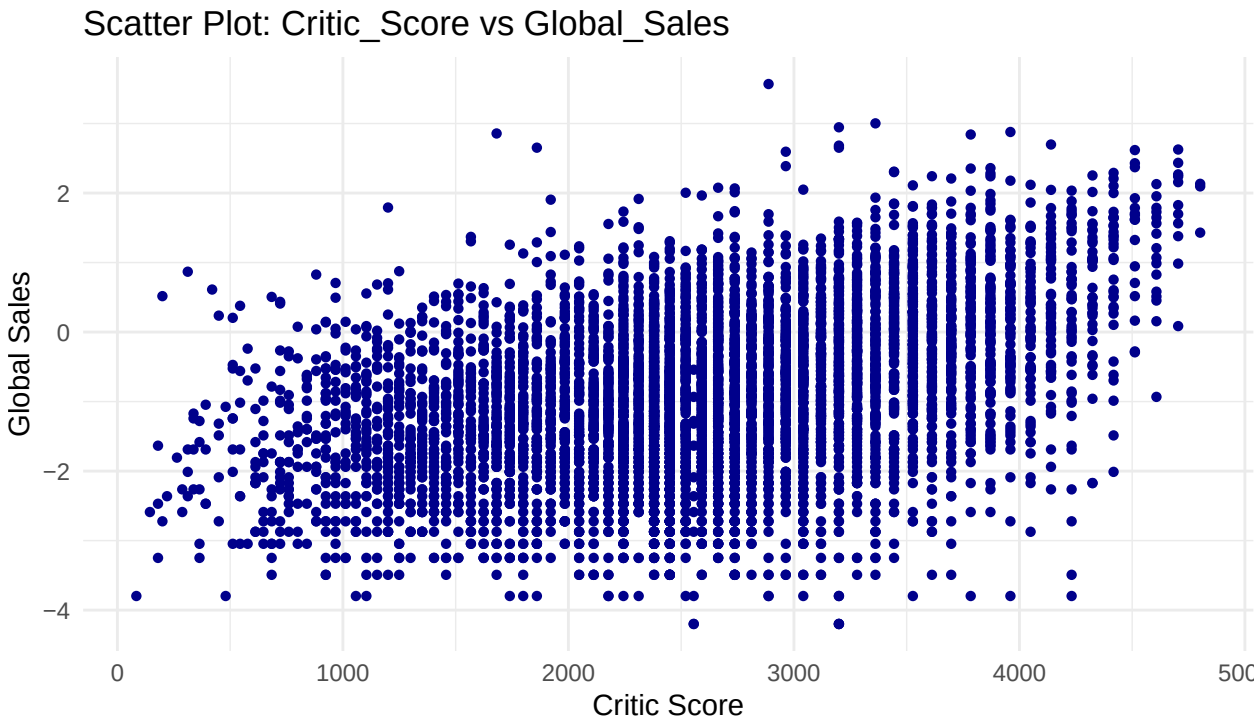
3.7 Inference about correlation

Question 8 : Is there a correlation (relationship) between Critic Score and Global Sales?
To find the correlation (relationship) between two numerical variables - Critic_Score and Global_Sales. These two variables have been chosen in an attempt to understand if Critic_Score and sales are related.

Step 1: Check for Assumptions & Considerations

Checking scatterplot:

```
ggplot(df, aes(x = Critic_Score, y = Global_Sales)) + geom_point(col = "darkblue",
    pch = 16) + labs(title = "Scatter Plot: Critic_Score vs Global_Sales",
    x = "Critic Score", y = "Global Sales") + theme_minimal() +
    theme(plot.margin = unit(c(1, 0, 1, 0), "cm"))
```



By looking at the scatterplot, it can be said the relationship is not linear, indicating that it might be better to use Spearman Rank Correlation Coefficient to find the correlation. Spearman also does not require normality as an assumption.

Step 2: Run the correlation test

- Hypotheses: Let True Correlation = ρ
  - Null Hypothesis (H0): $\rho = 0$
  - Alternative Hypothesis (H1): $\rho \neq 0$
- Significance level (α) : 0.05

Performing correlation test:

```
cor.test(df$Critic_Score, df$Global_Sales, method = "spearman",
    exact = FALSE)
```

```
##
##  Spearman's rank correlation rho
##
## data:  df$Critic_Score and df$Global_Sales
```

```
## S = 2.231e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.4180141
```

- Test Statistic and P-Value:

- Test Statistics: 2.231e+10
- P-Value: 0.1368

Comment on Results:
The p-value is lesser than 2.2e-16 (or p-value = 0) $< \alpha = 0.05$. Thus, we REJECT the null hypothesis that the correlation is 0.

Conclusion: We conclude that there is a relationship between Critic_Score and Sales.

3.8 Regression

Question 9 : Model Global_Sales as a function of Critic_Score, User_Score,Rating and Genre i.e., estimating the relationship between Global_Sales and Rating, Critic_Score, User_Score, Genre.

Encoding categorical variables before applying linear model:

```
# Encoding categorical variables
encoded_data <- model.matrix(~Genre + Rating, data = df)

# Combining the original dataframe and the encoded data
df_encoded <- cbind(df, encoded_data)

# Dropping the original categorical variables
df_encoded <- df_encoded[, -c(which(names(df) %in% c("Genre",
    "(Intercept)", "Rating", "Name", "Platform", "Year_of_Release",
    "Publisher", "NA_Sales", "EU_Sales", "Other_Sales")))]
print(names(df_encoded))
```

```
##  [1] "Global_Sales"            "Critic_Score"
##  [3] "User_Score"              "User_Score_Percentage"
##  [5] "Critic_Score_Percentage" "(Intercept)"
##  [7] "GenreAdventure"          "GenreFighting"
##  [9] "GenreMisc"               "GenrePlatform"
```

```
## [11] "GenrePuzzle"              "GenreRacing"
## [13] "GenreRole-Playing"        "GenreShooter"
## [15] "GenreSimulation"          "GenreSports"
## [17] "GenreStrategy"            "RatingE10+"
## [19] "RatingM"                  "RatingT"
```

```
# Applying linear model:
linear_model <- lm(Global_Sales ~ ., data = df_encoded)
```

Checking if assumptions for linear regression hold:

Assumptions:

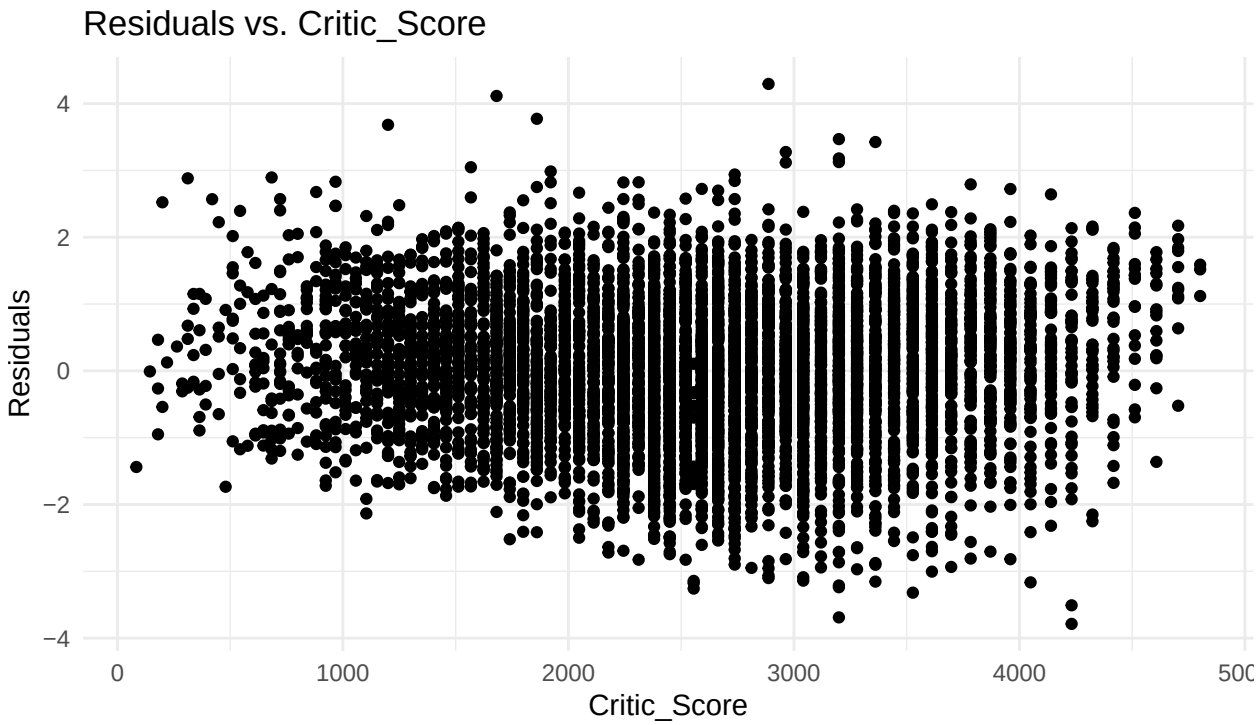
1. Linearity:

Looking at the scatterplots, Global_Sales seems to have weak positive relationships with both, Critic_Score and User_Score.

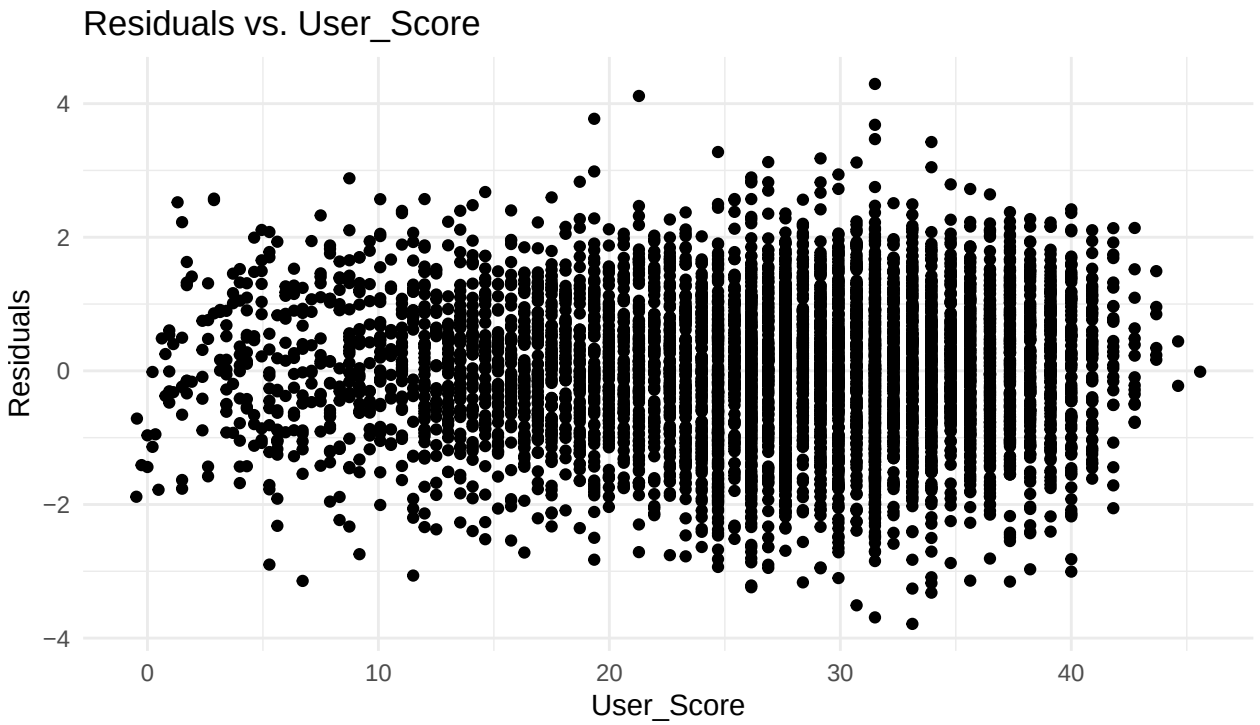
2. Independence:

i. Looking at the scatter plot of residuals and Critic_Score, and User_Score:

```
# Scatter plot for Critic_Score

residuals_vs_critic_score <- data.frame(Residuals = linear_model$residuals,
    Critic_Score = df_encoded$Critic_Score)

# Create scatter plot using ggplot
ggplot(residuals_vs_critic_score, aes(x = Critic_Score, y = Residuals)) +
    geom_point() + labs(title = "Residuals vs. Critic_Score",
    x = "Critic_Score", y = "Residuals") + theme_minimal() +
    theme(plot.margin = unit(c(1, 0, 1, 0), "cm"))
```




```r
# Scatter plot for User_Score

residuals_vs_user_score <- data.frame(Residuals = linear_model$residuals,
    User_Score = df_encoded$User_Score)

# Create scatter plot using ggplot
ggplot(residuals_vs_user_score, aes(x = User_Score, y = Residuals)) +
    geom_point() + labs(title = "Residuals vs. User_Score", x = "User_Score",
    y = "Residuals") + theme_minimal() + theme(plot.margin = unit(c(1,
    0, 1, 0), "cm"))
```

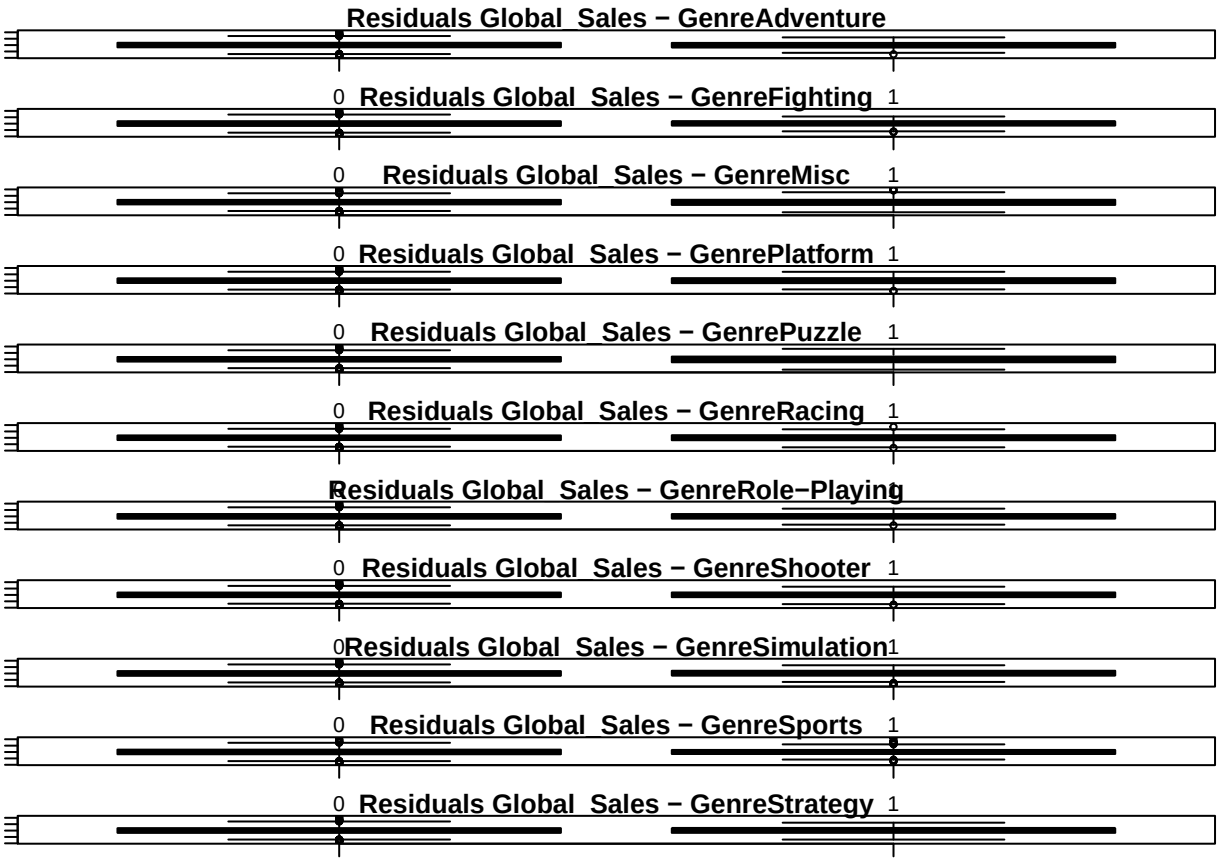


The scatterplots show no discernible patterns or trends between the variable and residuals. This shows that they are independent.

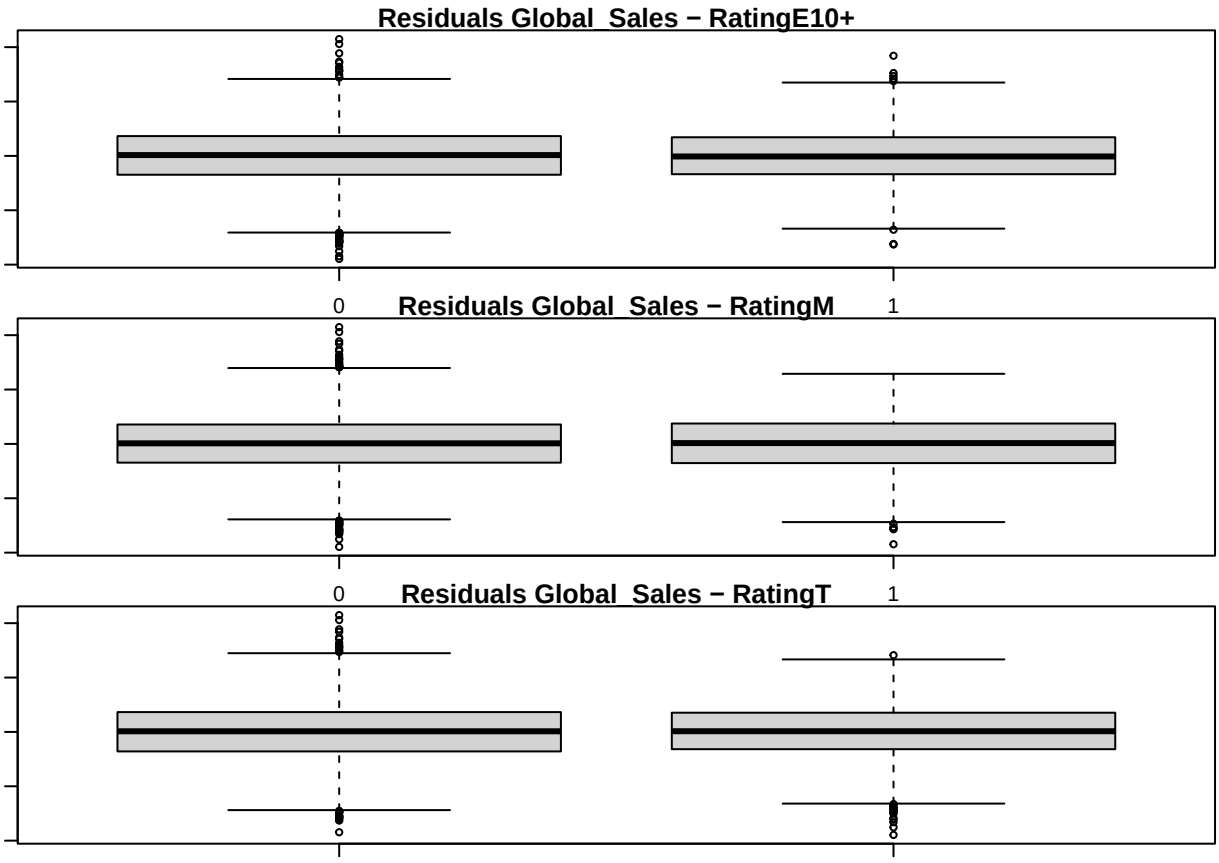
ii. Looking at side-by-side boxplots for categorical variables Rating and Genre:

```
# Encoding categorical variables
encoded_data <- model.matrix(~Genre + Rating, data = df)

# Combining the original dataframe and the encoded data
df_encoded <- cbind(df, encoded_data)

# Dropping the original categorical variables
df_encoded <- df_encoded[, -c(which(names(df) %in% c("Genre",
    "(Intercept)", "Rating", "Name", "Platform", "Year_of_Release",
    "Publisher", "NA_Sales", "EU_Sales", "Other_Sales")))]

# Applying linear model
linear_model <- lm(Global_Sales ~ ., data = df_encoded)

# Extracting residuals and adding them to df_encoded
df_encoded$residuals <- residuals(linear_model)

# Columns in df_encoded for Genre and Rating
genre_cols <- grep("^Genre", names(df_encoded), value = TRUE)
rating_cols <- grep("^Rating", names(df_encoded), value = TRUE)

# Create side-by-side boxplots for Genre
```

```r
par(mfrow = c(length(genre_cols), 1), mar = c(1, 1, 1, 1))

for (col in genre_cols) {
    boxplot(df_encoded$residuals ~ df_encoded[[col]], main = paste("Residuals Global_Sales -",
        col))
}
```




```r
# Create side-by-side boxplots for Rating
par(mfrow = c(length(rating_cols), 1), mar = c(1, 1, 1, 1))

for (col in rating_cols) {
    boxplot(df_encoded$residuals ~ df_encoded[[col]], main = paste("Residuals Global_Sales -",
        col))
}
```

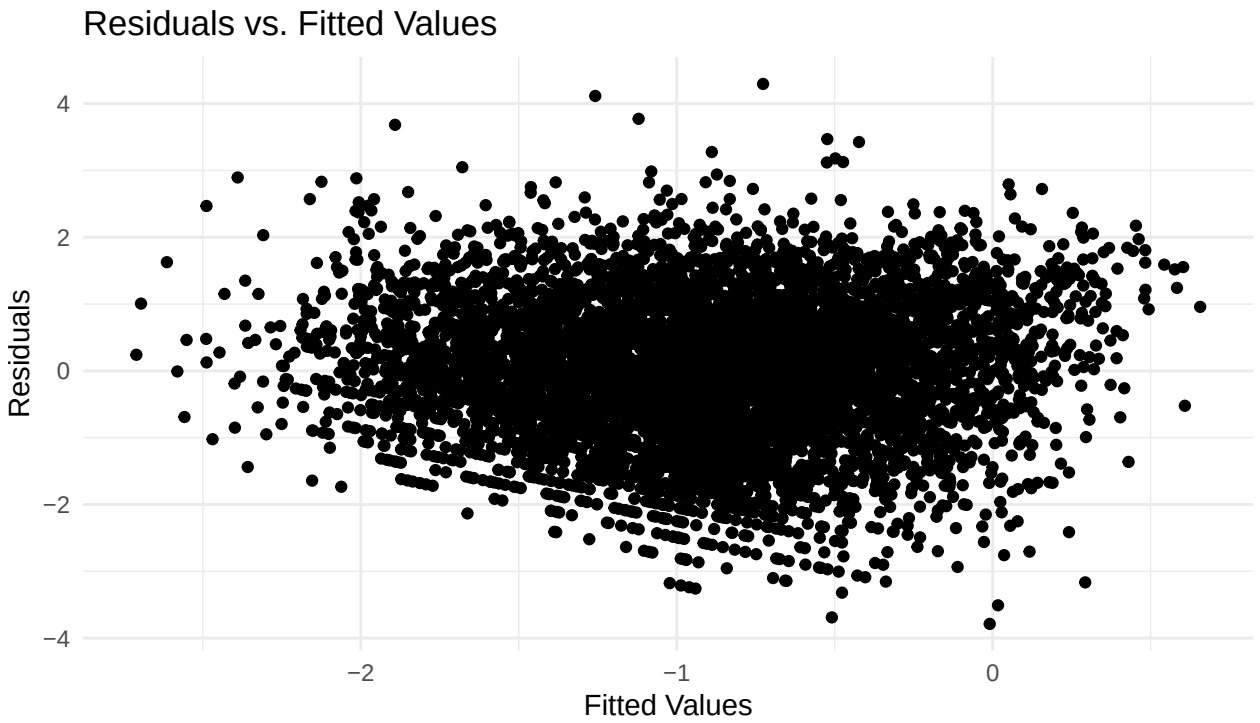


Looking at the boxplots of both categorical variables, we see that the residuals do not show any patterns, i.e., the boxplots are similar across categories. This indicates that both categorical variables are independent.

3. Homoscedasticity:

Residual vs. Fitted value plot:

```
residuals_df <- data.frame(residuals = linear_model$residuals,
    fitted_values = predict(linear_model))

# Create a scatter plot using ggplot
ggplot(residuals_df, aes(x = fitted_values, y = residuals)) +
    geom_point() + labs(title = "Residuals vs. Fitted Values") +
    xlab("Fitted Values") + ylab("Residuals") + theme_minimal() +
    theme(plot.margin = unit(c(1, 0, 1, 0), "cm"))
```



Looking at the plot, we see that there is no pattern, the residuals are randomly scattered and spread out indicating that the constant variance assumptions hold.

4. Normality of Residuals: Q-Q plot and histogram of residuals.

```
residuals_df <- data.frame(Residuals = linear_model$residuals,
    Theoretical_Quantiles = qnorm(ppoints(length(linear_model$residuals))))

ggplot(residuals_df, aes(sample = Residuals)) + geom_qq() + geom_qq_line() +
    labs(title = "Normal Q-Q Plot of Residuals") + theme_minimal() +
    theme(plot.margin = unit(c(1, 0, 1, 0), "cm"))
```

Looking at the q-q plot of the residuals, we see that they follow the straight line almost exactly, indicating that they are normally distributed.

Thus, all assumptions for linear regression hold.

Linear Model:

```
summary(linear_model)
```

```
##
## Call:
## lm(formula = Global_Sales ~ ., data = df_encoded)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7877 -0.6920  0.0237  0.7219  4.2948
##
## Coefficients: (3 not defined because of singularities)
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -2.083e+00  6.185e-02 -33.675  < 2e-16 ***
## Critic_Score             6.513e-04  1.853e-05  35.156  < 2e-16 ***
## User_Score              -1.085e-02  1.818e-03  -5.969 2.53e-09 ***
## User_Score_Percentage           NA         NA      NA       NA
## Critic_Score_Percentage         NA         NA      NA       NA
## `(Intercept)`                   NA         NA      NA       NA
## GenreAdventure          -4.130e-01  7.478e-02  -5.523 3.47e-08 ***
```

```
## GenreFighting            1.917e-02  6.753e-02   0.284 0.776553
## GenreMisc               -3.927e-02  5.673e-02  -0.692 0.488788
## GenrePlatform            4.628e-02  6.500e-02   0.712 0.476516
## GenrePuzzle             -4.686e-01  1.024e-01  -4.577 4.82e-06 ***
## GenreRacing             -1.613e-01  5.630e-02  -2.865 0.004184 **
## `GenreRole-Playing`     -8.590e-02  5.516e-02  -1.557 0.119497
## GenreShooter             6.686e-02  5.018e-02   1.332 0.182759
## GenreSimulation         -8.776e-02  6.857e-02  -1.280 0.200599
## GenreSports             -1.827e-01  5.090e-02  -3.590 0.000333 ***
## GenreStrategy           -8.048e-01  8.945e-02  -8.997  < 2e-16 ***
## `RatingE10+`            -2.092e-01  4.490e-02  -4.659 3.25e-06 ***
## RatingM                 -1.683e-01  4.932e-02  -3.412 0.000649 ***
## RatingT                 -2.354e-01  3.966e-02  -5.935 3.10e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.052 on 6110 degrees of freedom
## Multiple R-squared:  0.2091, Adjusted R-squared:  0.2071
## F-statistic:   101 on 16 and 6110 DF,  p-value: < 2.2e-16
```

Significant Coefficient Estimates:

Extracting coefficient estimates that are statistically significant (significantly different from 0) at the 0.05 significance level:

```
model_summary <- summary(linear_model)

coefficients <- model_summary$coefficients[, 1]
p_values <- model_summary$coefficients[, 4]
variable_names <- rownames(model_summary$coefficients)

coefficients_df <- data.frame(Variable = variable_names, Coefficient = coefficients,
    P_Value = p_values)

significant_coefficients <- coefficients_df[coefficients_df$P_Value <
    0.05, , drop = FALSE]

print(significant_coefficients)
```

```
##                      Variable   Coefficient       P_Value
## (Intercept)       (Intercept) -2.0829086384 3.367672e-228
## Critic_Score     Critic_Score  0.0006513331 9.561744e-247
## User_Score         User_Score -0.0108499063  2.527345e-09
```

```
## GenreAdventure GenreAdventure -0.4129982054  3.466818e-08
## GenrePuzzle       GenrePuzzle -0.4686256000  4.819274e-06
## GenreRacing       GenreRacing -0.1612971382  4.184261e-03
## GenreSports       GenreSports -0.1827423974  3.333152e-04
## GenreStrategy   GenreStrategy -0.8048328067  3.033561e-19
## `RatingE10+`     `RatingE10+` -0.2091841197  3.245358e-06
## RatingM               RatingM -0.1682646198  6.492434e-04
## RatingT               RatingT -0.2353828001  3.099781e-09
```

Looking at the summary of the linear model, at the 0.05 significance level, the significant coefficients are Intercept term, Critic_Score, User_Score, GenreAdventure, GenrePuzzle, GenreRacing, GenreSports, GenreStrategy, RatingE10+, RatingM and RatingT.

Regression Equation:

Thus, the regression equation is given by: Global_Sales = -2.08290863836866 + 0.0006513331354053xCritic_Score + -0.0108499062748751xUser_Score + -0.412998205366902xGenreAdventure + -0.468625600029351xGenrePuzzle + -0.161297138173433xGenreRacing + -0.18274239742445xGenreSports + -0.804832806673555xGenreStrategy + -0.209184119736127x`RatingE10+` + -0.16826461980157xRatingM + -0.235382800082318xRatingT