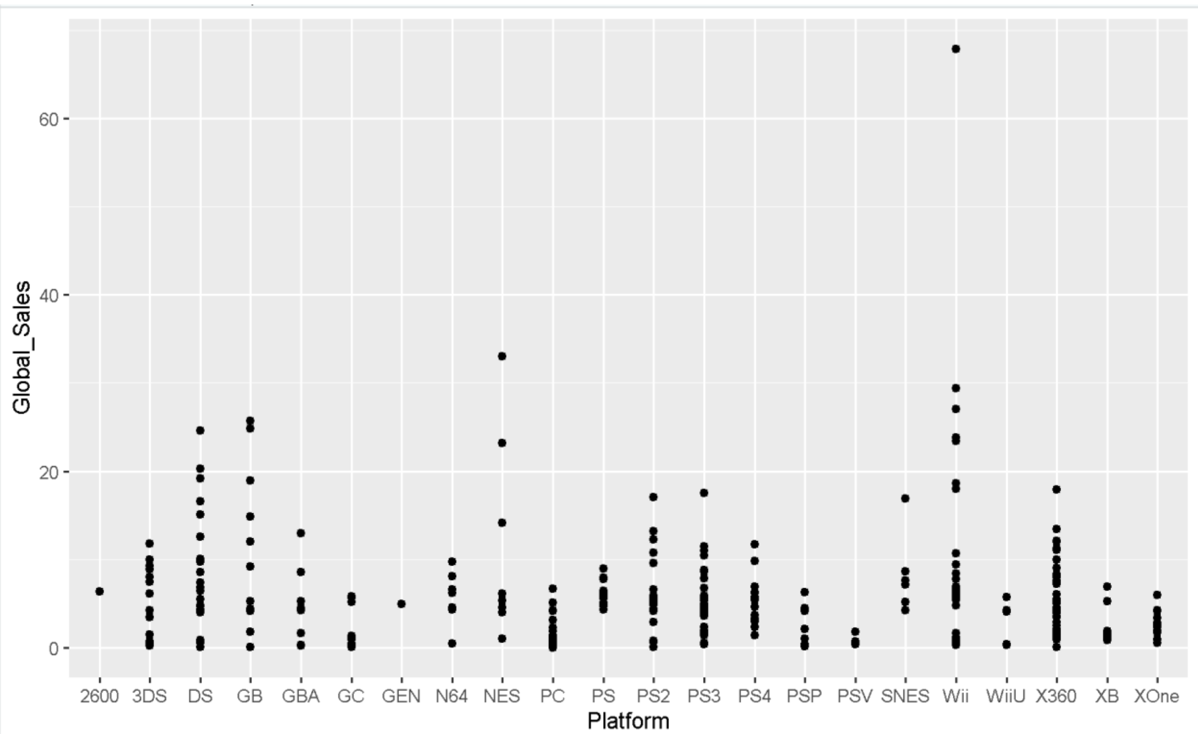
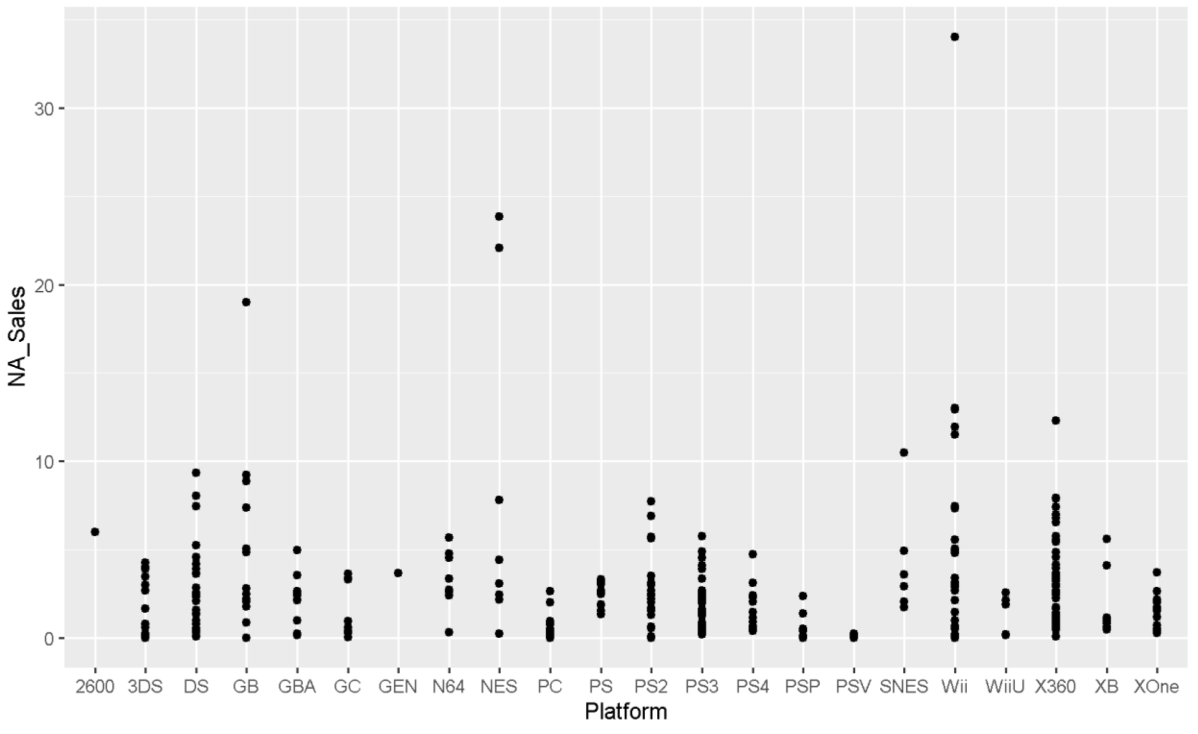
The suggested structure for the report is:

1. **Background/context of the business (100 words):** Briefly describe the business scenario (context) and the business problem you aim to solve for Turtle Games.
2. **Analytical approach (350 words):** Describe the approach taken to import, clean, and analyse data in Python and R. Include a detailed and insightful description of the processes you used and the decisions you made during analysis, such as the choice of libraries, functions, and variables. Ensure that the description of the steps taken to prepare data for analysis is clear, well-organised, and relevant to Turtle Games's objectives.

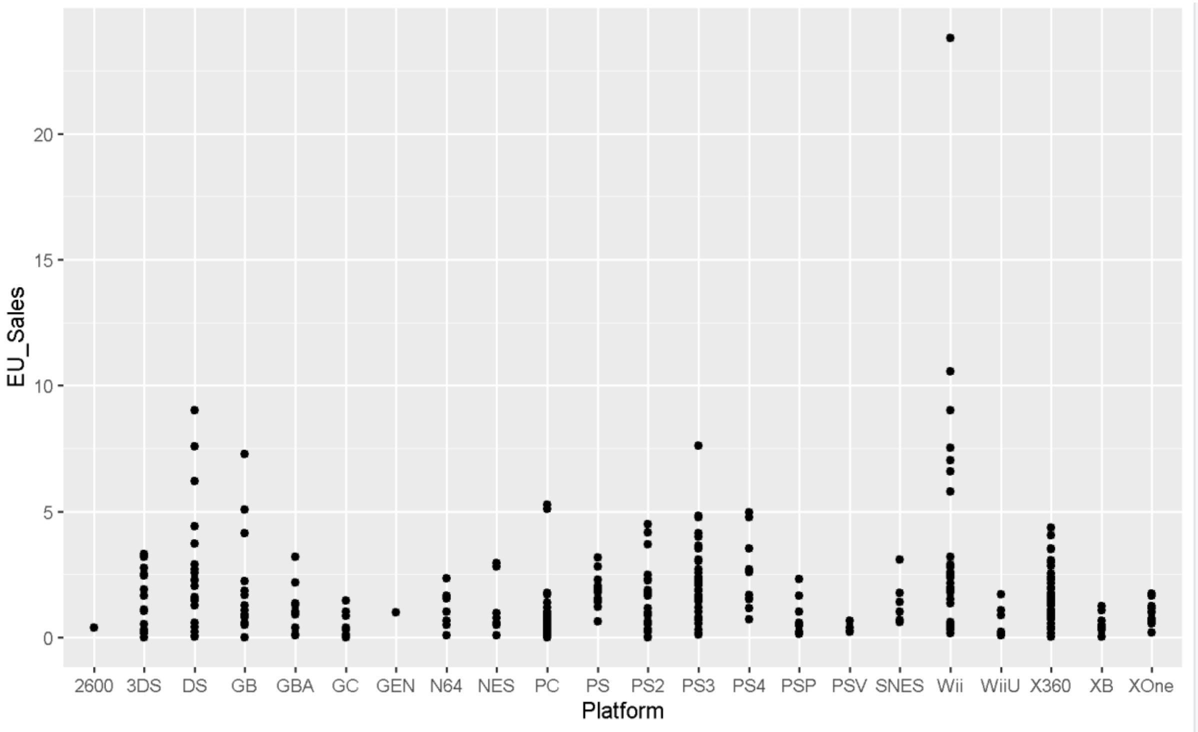
The marketing department also wants to better understand the usefulness of remuneration and spending scores in providing data for analysis but does not know where to begin. You are tasked to identify groups within the customer base that can be used to target specific market segments.

3. **Visualisation and insights (350 words):** Describe the rationale for the selected visualisations, and ensure that the interpretations of the visualisation outputs are detailed, insightful, and relevant to the business objectives.
4. **Patterns and predictions (200 words):** Clearly articulate the patterns you've discovered and the predictions you've made. Describe how they relate to the business scenario, and address Turtle Games's objectives.
5. **Ensure** that you use the correct naming convention before you submit. Save and submit your files as:
    - LastName_FirstName_DA301_Assignment_Report.pdf

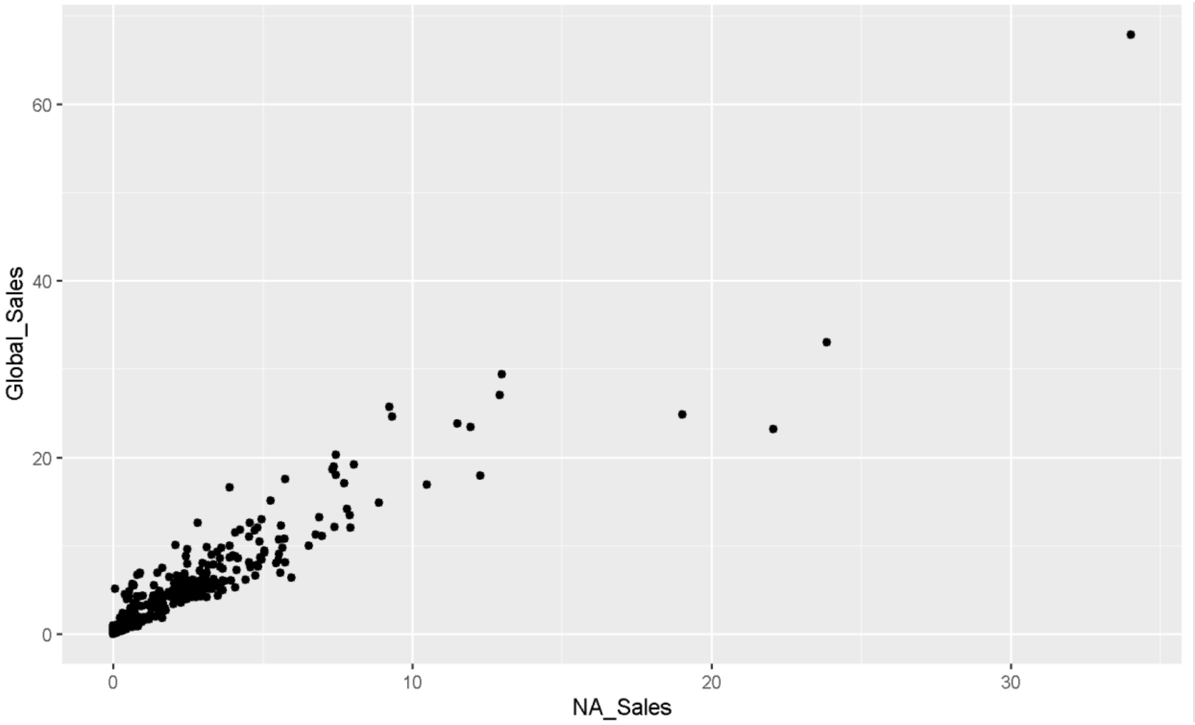




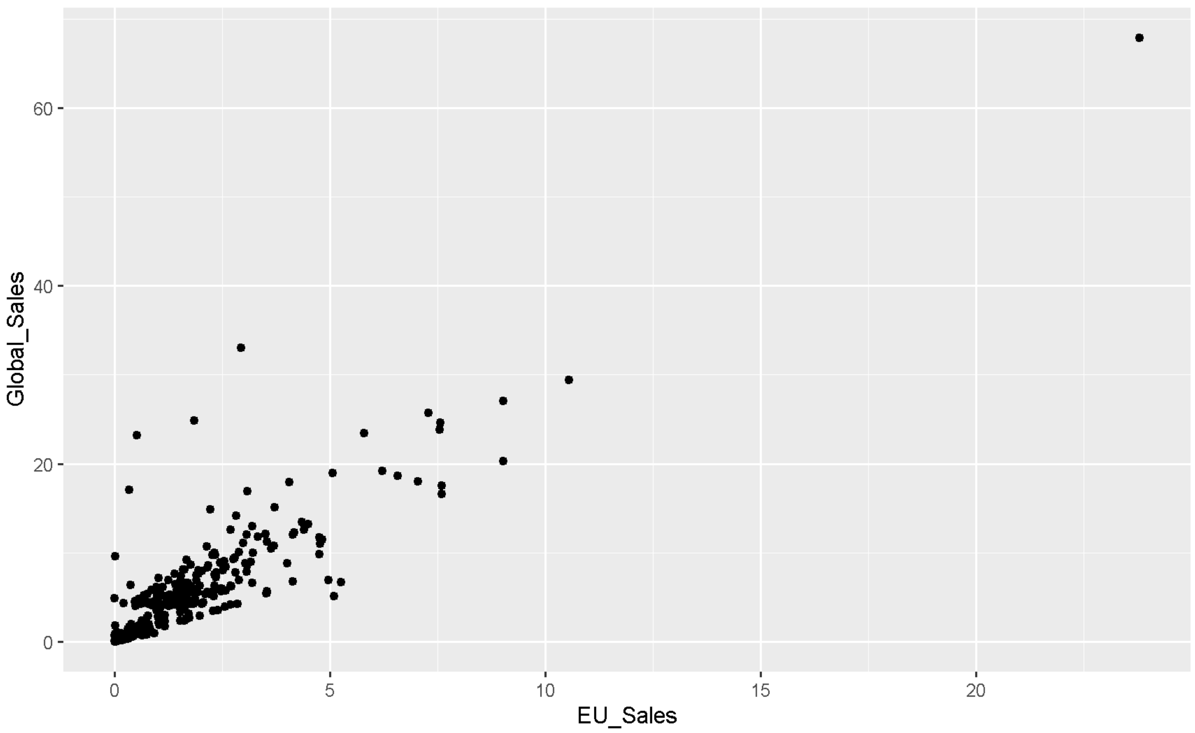
NA_Sales
0
10
20
30
2600 3DS DS GB GBA GC GEN N64 NES PC PS PS2 PS3 PS4 PSP PSV SNES Wii WiiU X360 XB XOne
Platform



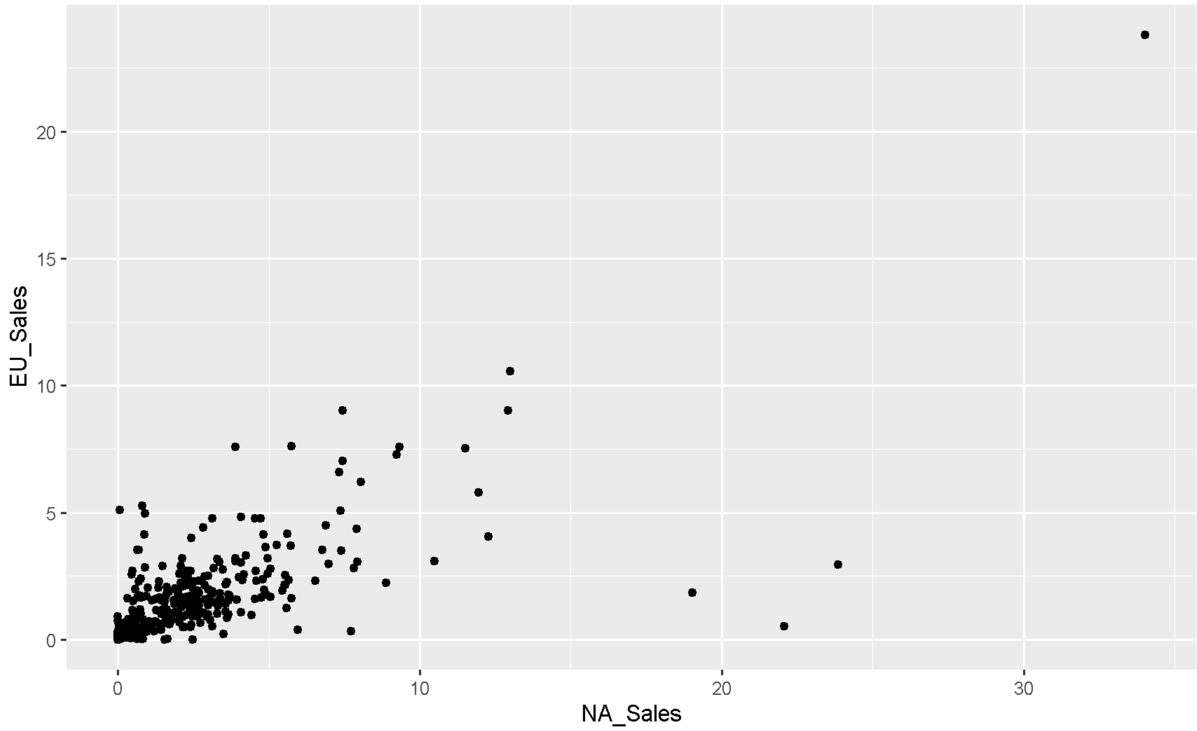
EU_Sales
0
5
10
15
20
2600 3DS DS GB GBA GC GEN N64 NES PC PS PS2 PS3 PS4 PSP PSV SNES Wii WiiU X360 XB XOne
Platform

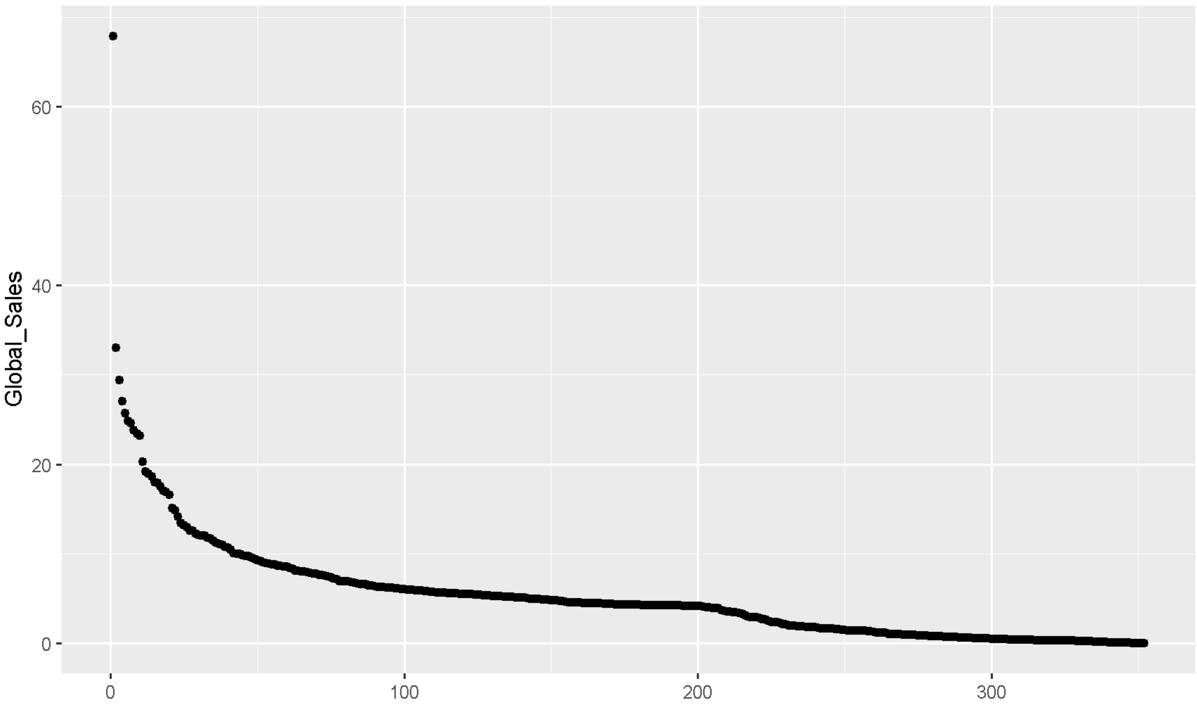


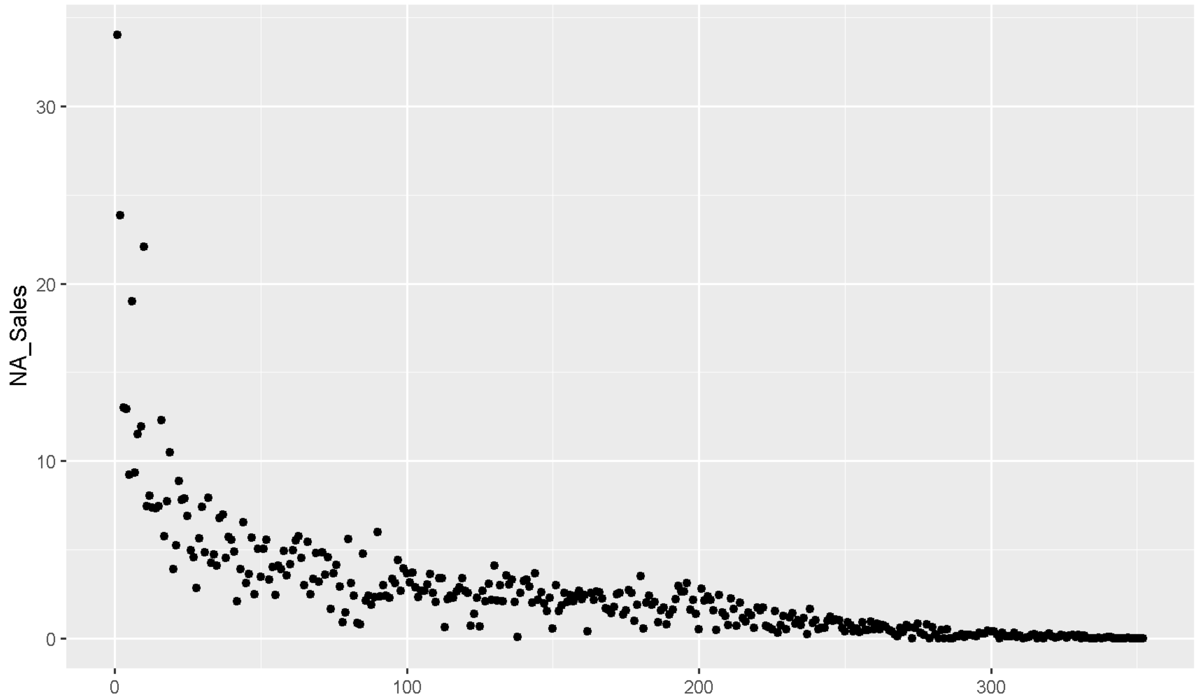
Global_Sales
0
20
40
60
0
10
20
30
NA_Sales

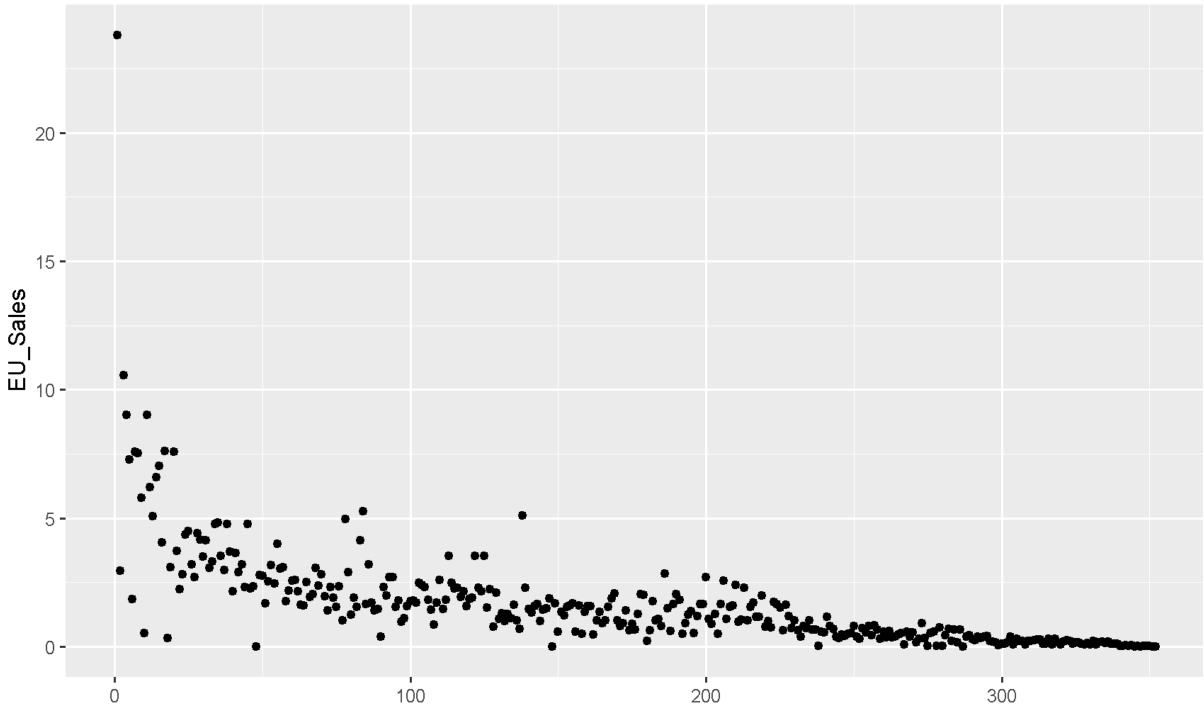


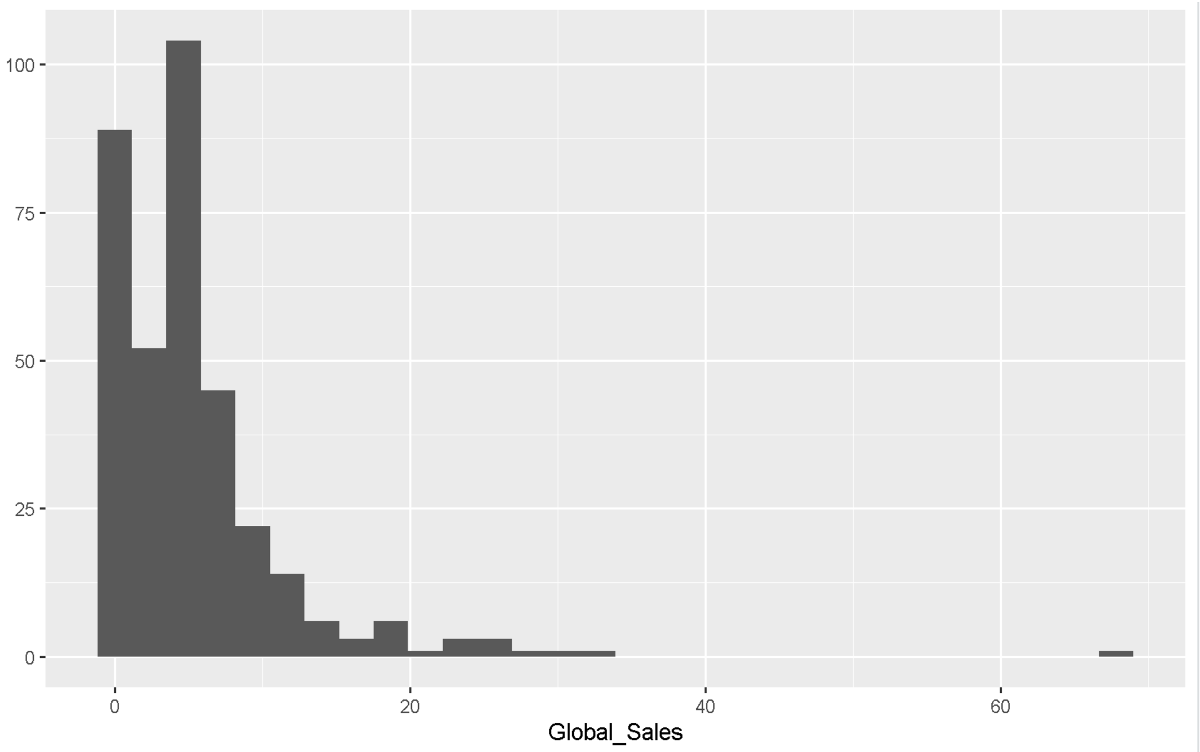
Global_Sales
0
20
40
60
0
5
10
15
20
EU_Sales

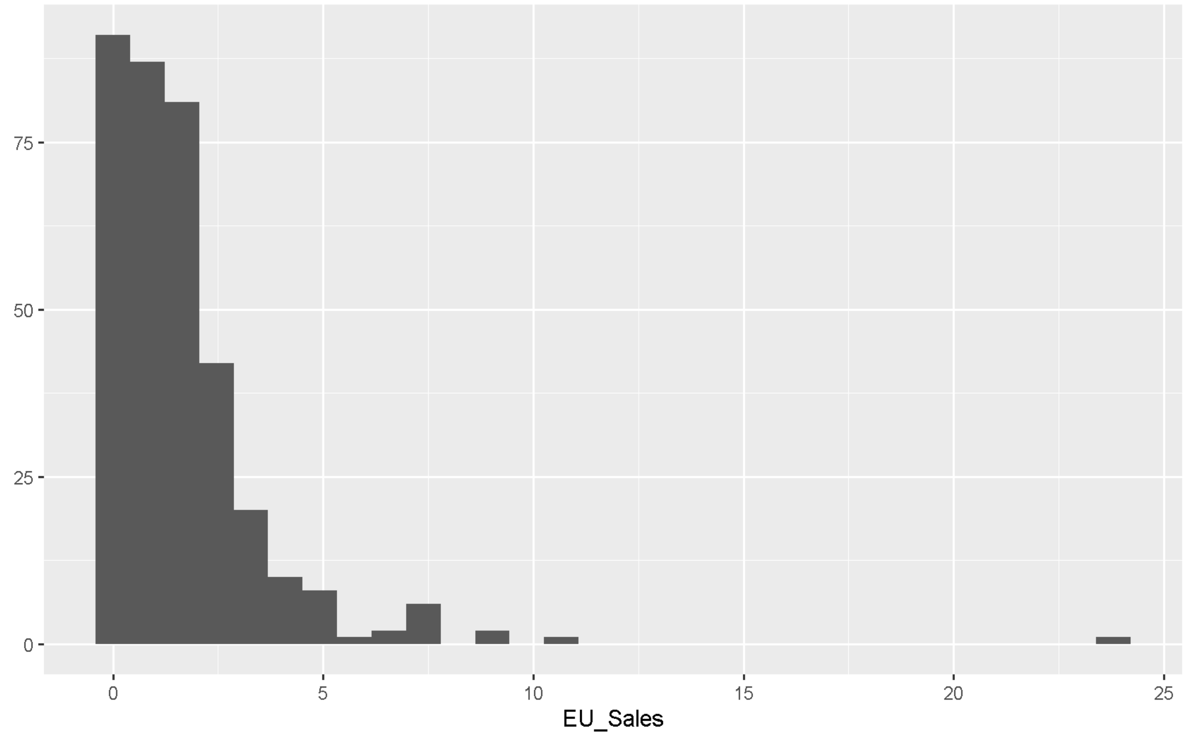

EU_Sales
NA_Sales
0
5
10
15
20
0
10
20
30

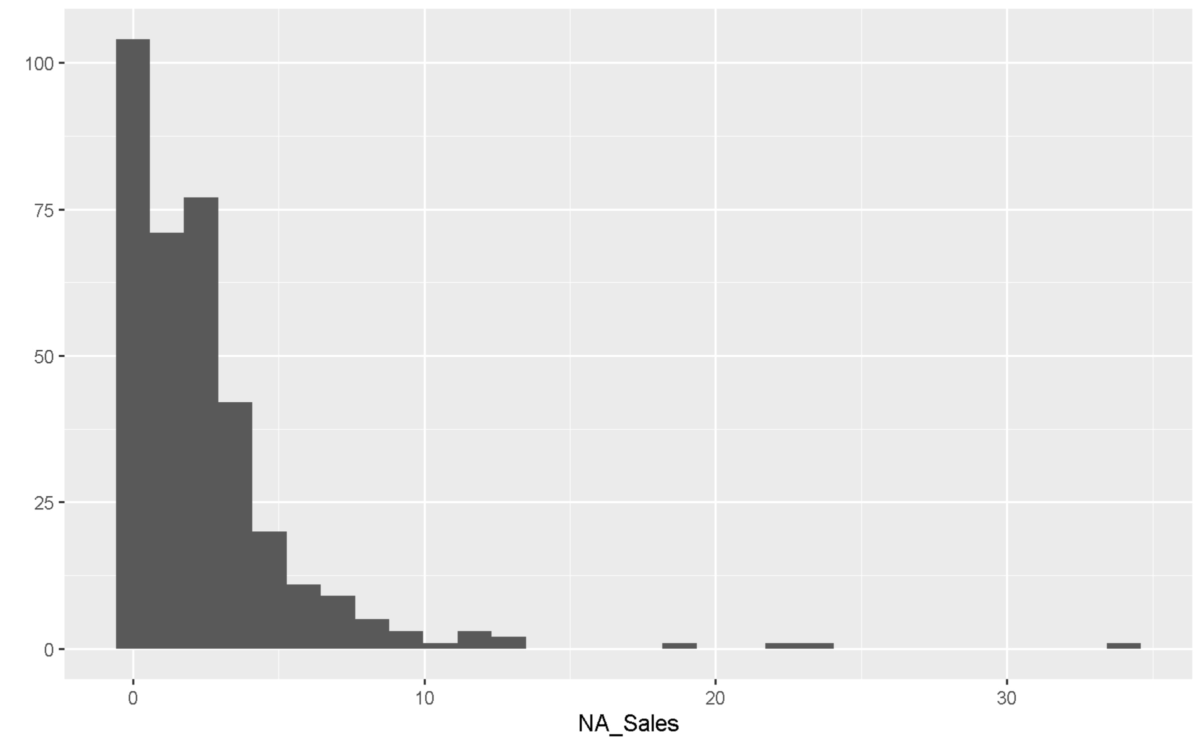

Global_Sales
0
20
40
60
0
100
200
300

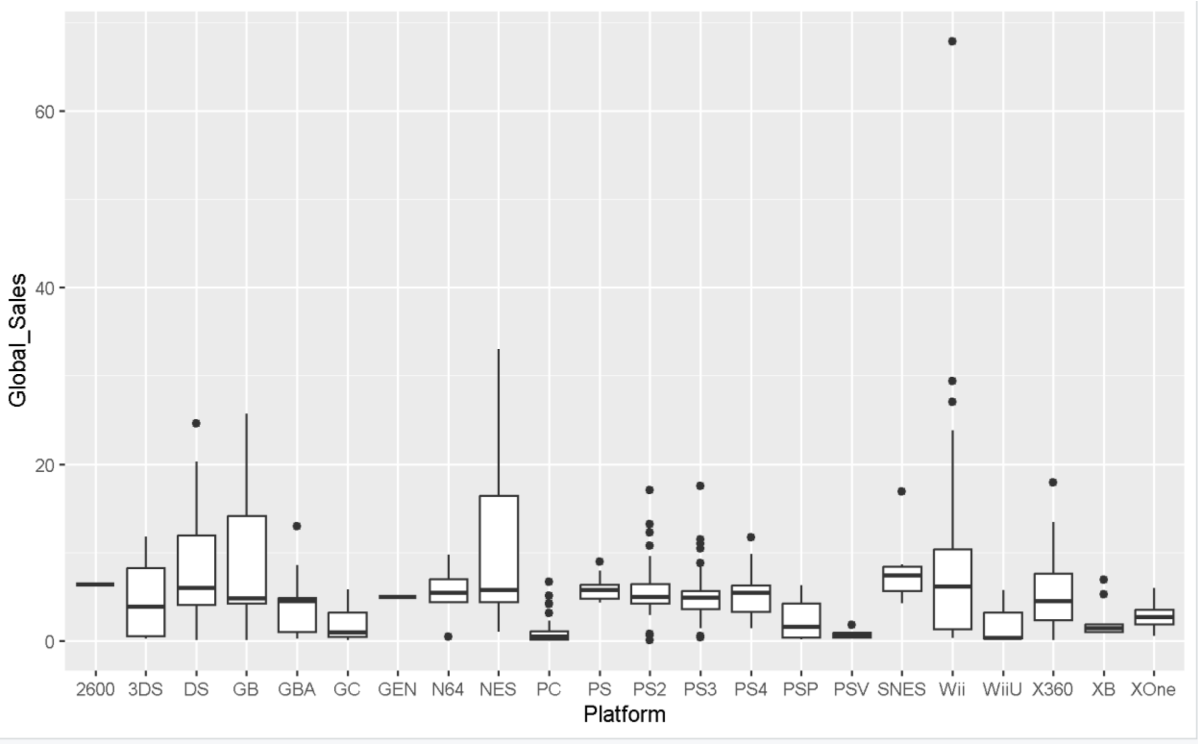

NA_Sales
30
20
10
0
0
100
200
300

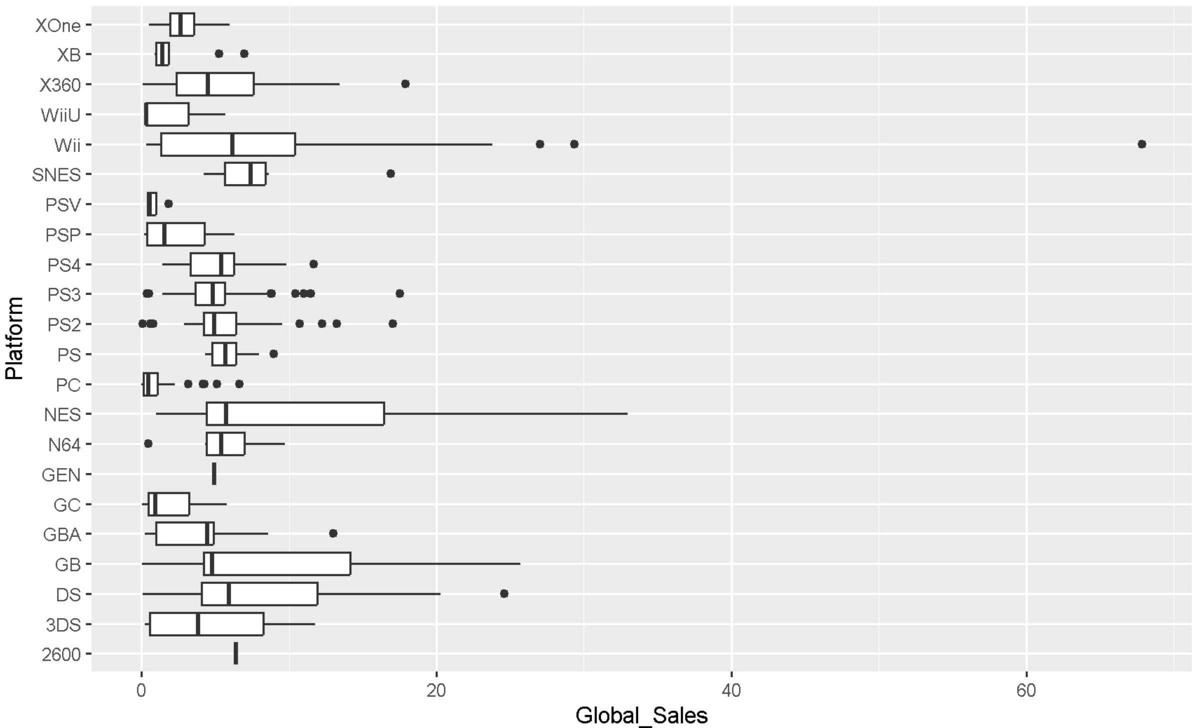

EU_Sales
20
15
10
5
0
0
100
200
300

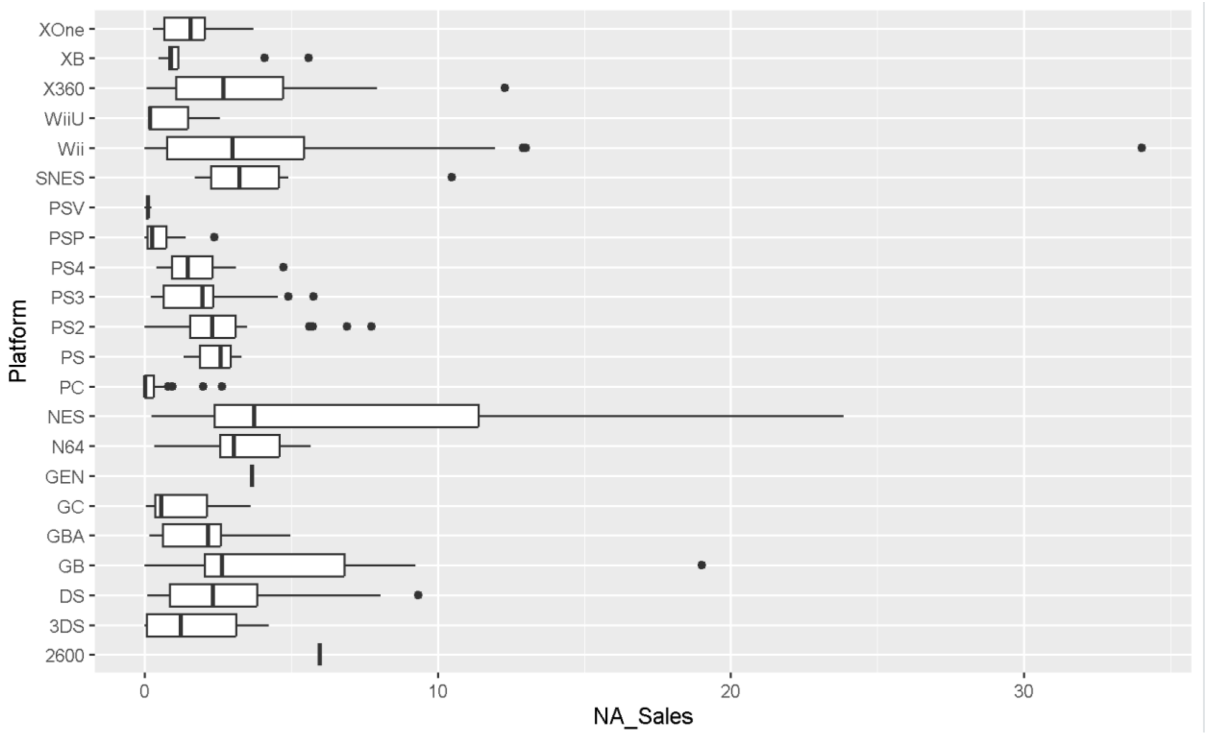

100
75
50
25
0
0
20
40
60
Global_Sales

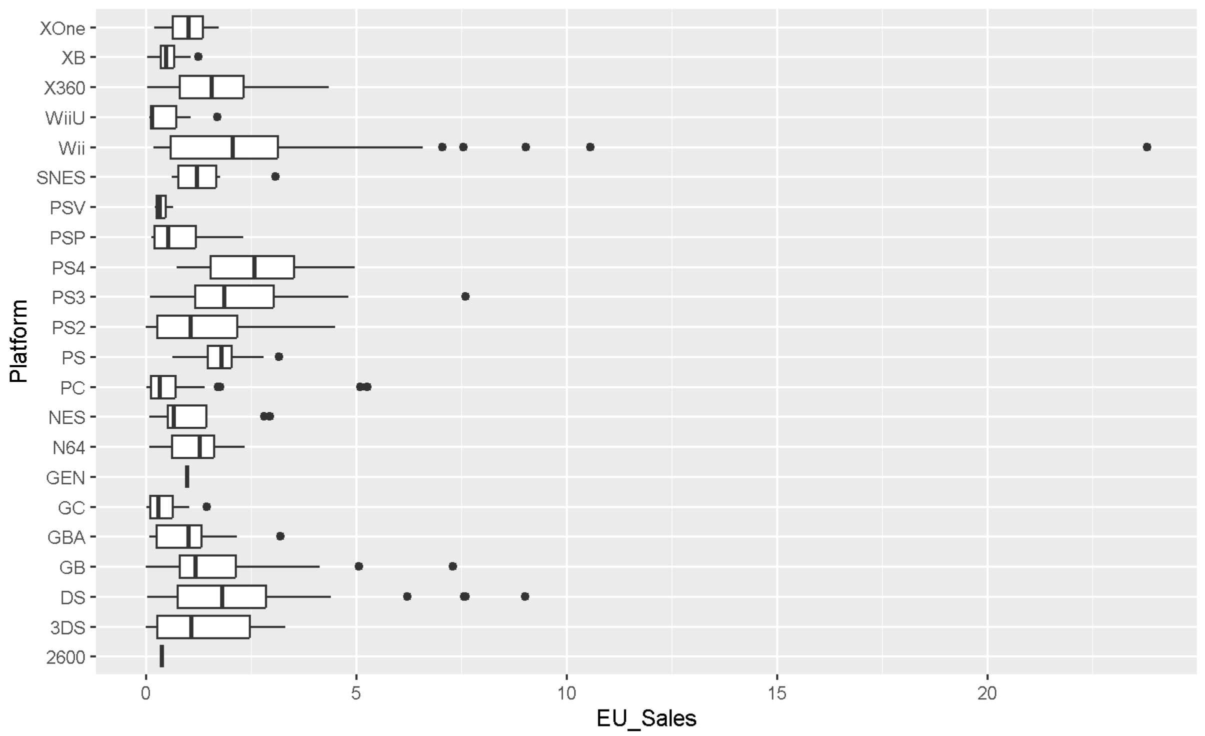

75
50
25
0
0
5
10
15
20
25
EU_Sales


100
75
50
25
0
0
10
20
30
NA_Sales


Global_Sales
0
20
40
60
2600
3DS
DS
GB
GBA
GC
GEN
N64
NES
PC
PS
PS2
PS3
PS4
PSP
PSV
SNES
Wii
WiiU
X360
XB
XOne
Platform


Platform
XOne
XB
X360
WiiU
Wii
SNES
PSV
PSP
PS4
PS3
PS2
PS
PC
NES
N64
GEN
GC
GBA
GB
DS
3DS
2600
0
20
40
60
Global_Sales





2b) Determine which plot is the best to compare game sales.

Create scatterplots.

qplot(Platform,sum_EU, data=df_s_p)

qplot(Platform,sum_NA, data=df_s)

qplot(Platform,sum_global, data=df_s)

qplot(y=sum_global, data=df_s_p)

qplot(y=sum_NA, data=df_s_p)

qplot(y=sum_EU, data=df_s_p)

Create histograms.

qplot(sum_global, bins=5, data=df_s_p)

qplot(sum_global, data=df_s_p)

qplot(sum_EU, data=df_s_p)

qplot(sum_NA, data=df_s_p)

Create boxplots.

qplot(sum_global, Platform, data=df_s_p, geom='boxplot')

qplot(sum_NA, Platform, data=df_s_p, geom='boxplot')

qplot(sum_EU, Platform, data=df_s_p, geom='boxplot')

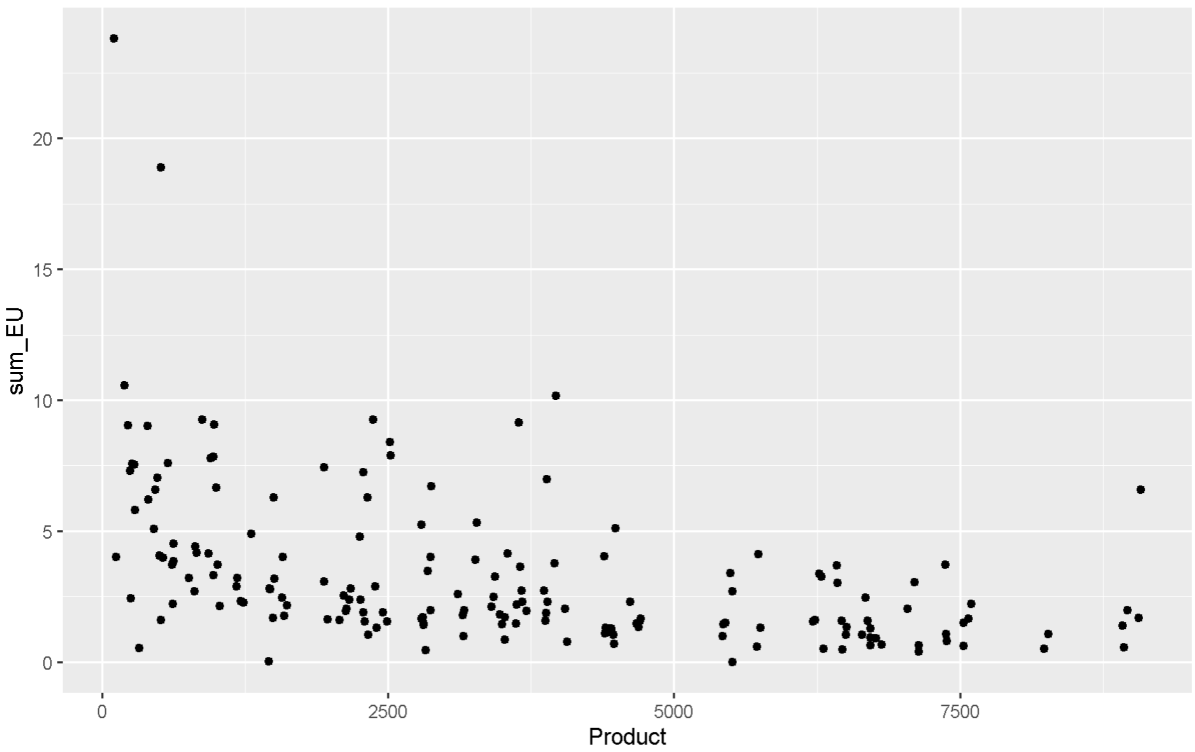
```
# Specify the qqnorm function.
# Draw a qqplot using the total_seconds data.
qqnorm(drive3$total_seconds,
       col='blue',
       xlab="z Value",
       ylab='Time')

# Specify the qqline function.
# Add a reference line to the qqplot.
qqline(drive3$total_seconds,
       col='red',
       lwd=2)

# Run a Shapiro-Wilk test.
shapiro.test(drive3$total_seconds)

# Specify the skewness and kurtosis functions.
```

```r
skewness(drive3$total_seconds)
kurtosis(drive3$total_seconds)
```

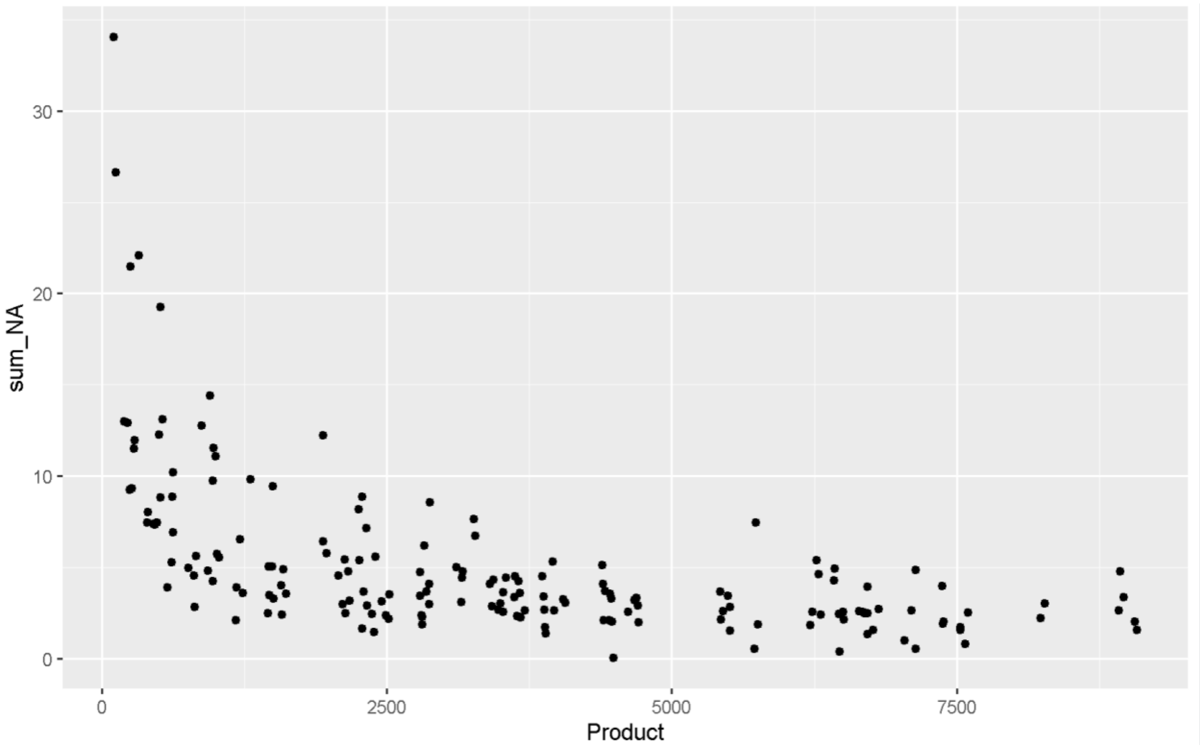
```r
# Specify the t.test function.
# Set the data source, the confidence interval (95%),
# and the theoretical mean.

t.test(drive3$total_seconds,
       conf.level=0.95,
       mu=120)
```

```r
# Run a Shapiro-Wilk test.
shapiro.test(drive3$car_stop)
shapiro.test(drive3$car_go)
shapiro.test(drive3$take_order)
shapiro.test(drive3$hand_over_order)
```

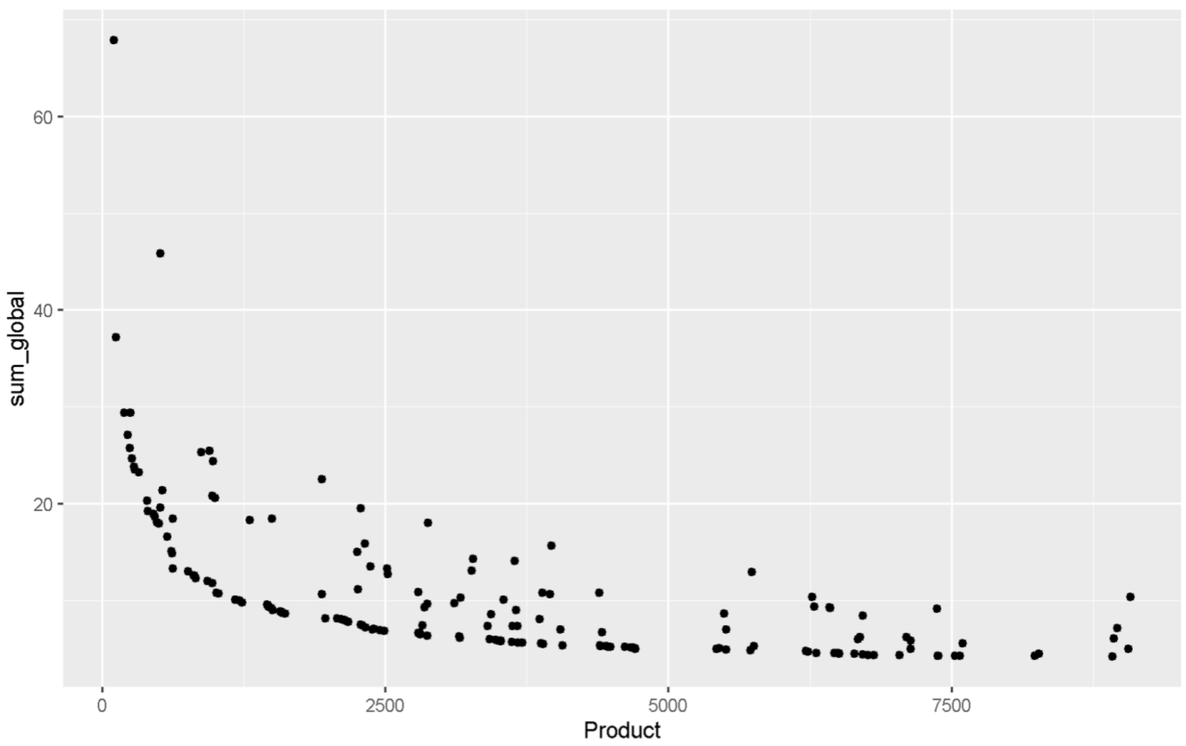
```r
# Specify the cor function.
# Set the first and second variables.

cor(drive3$car_stop, drive3$car_go)
cor(drive3$take_order, drive3$hand_over_order)

# Determine the correlation for the whole data frame.
round (cor(drive3),
       digits=2)
```

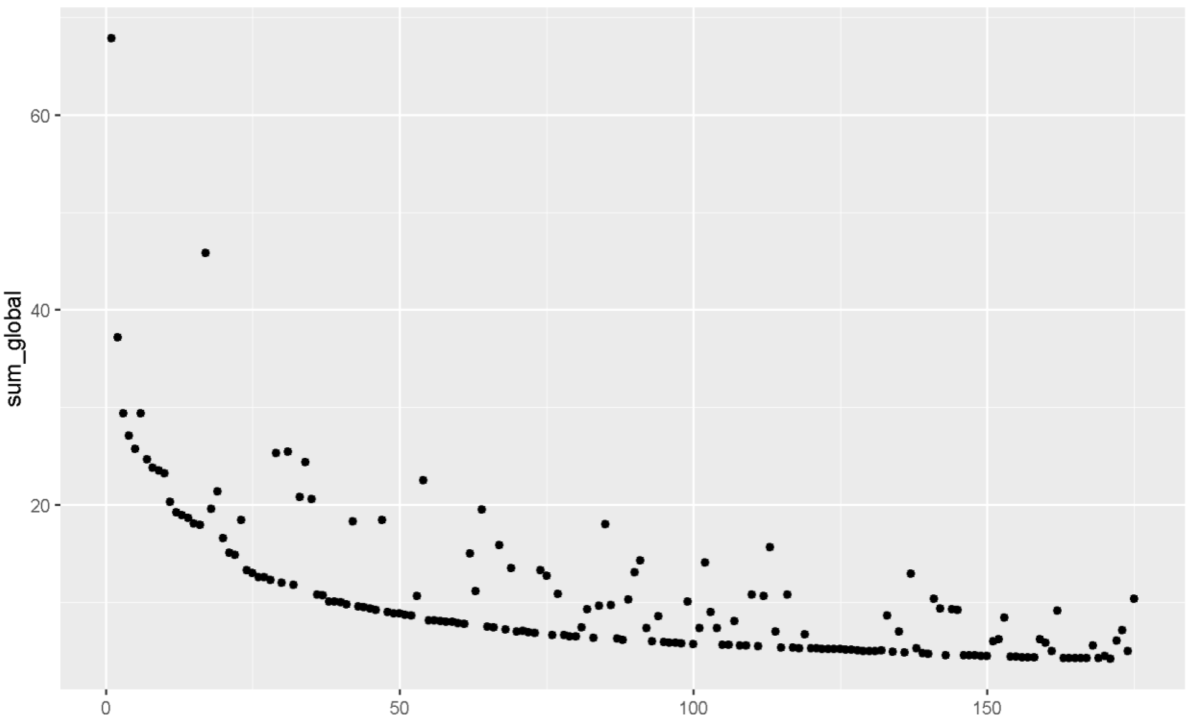


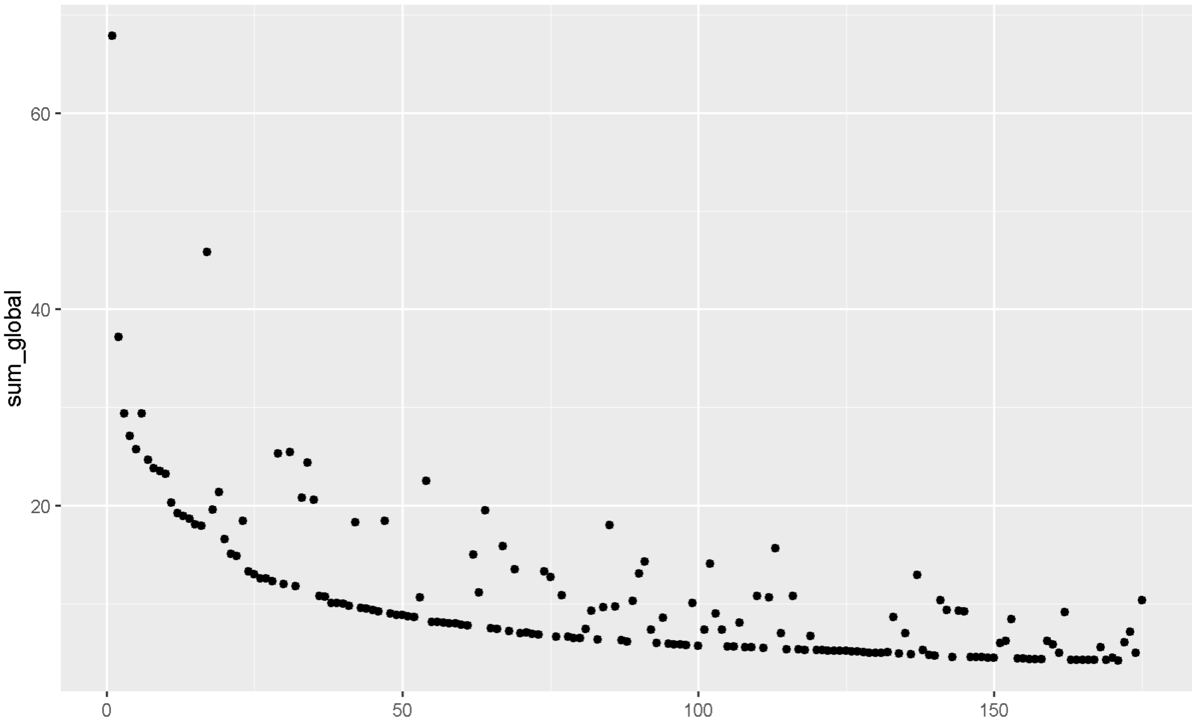
sum_EU
20
15
10
5
0
0
2500
5000
7500
Product

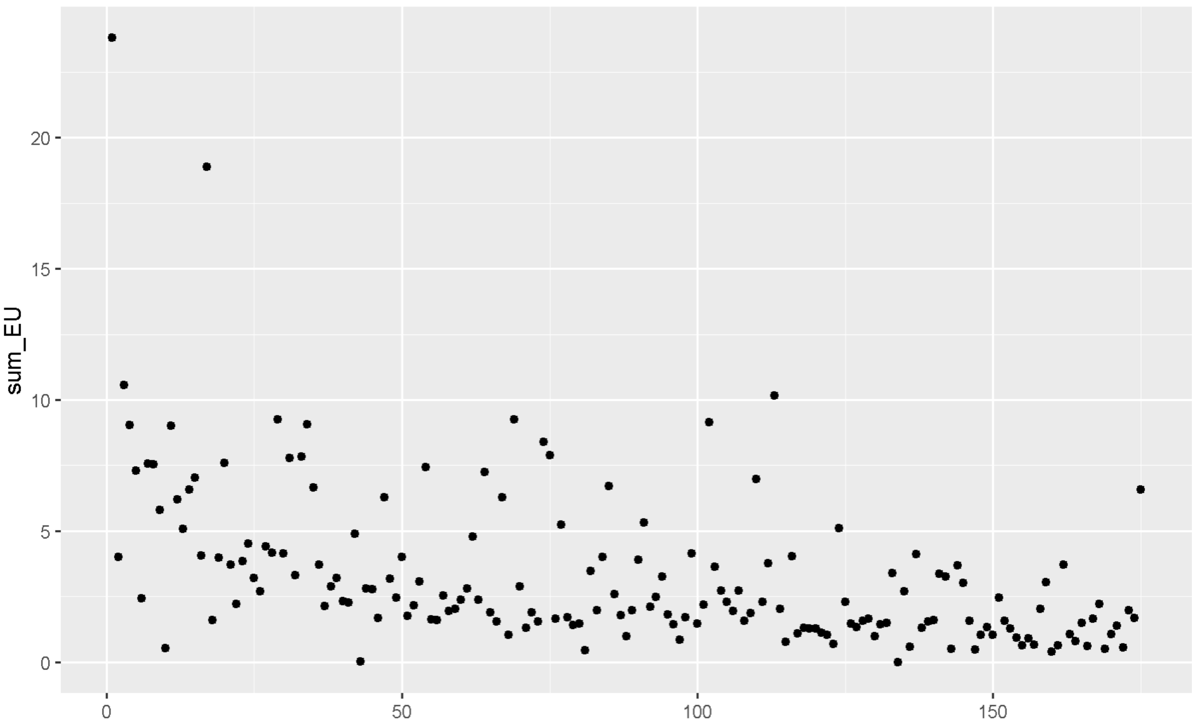


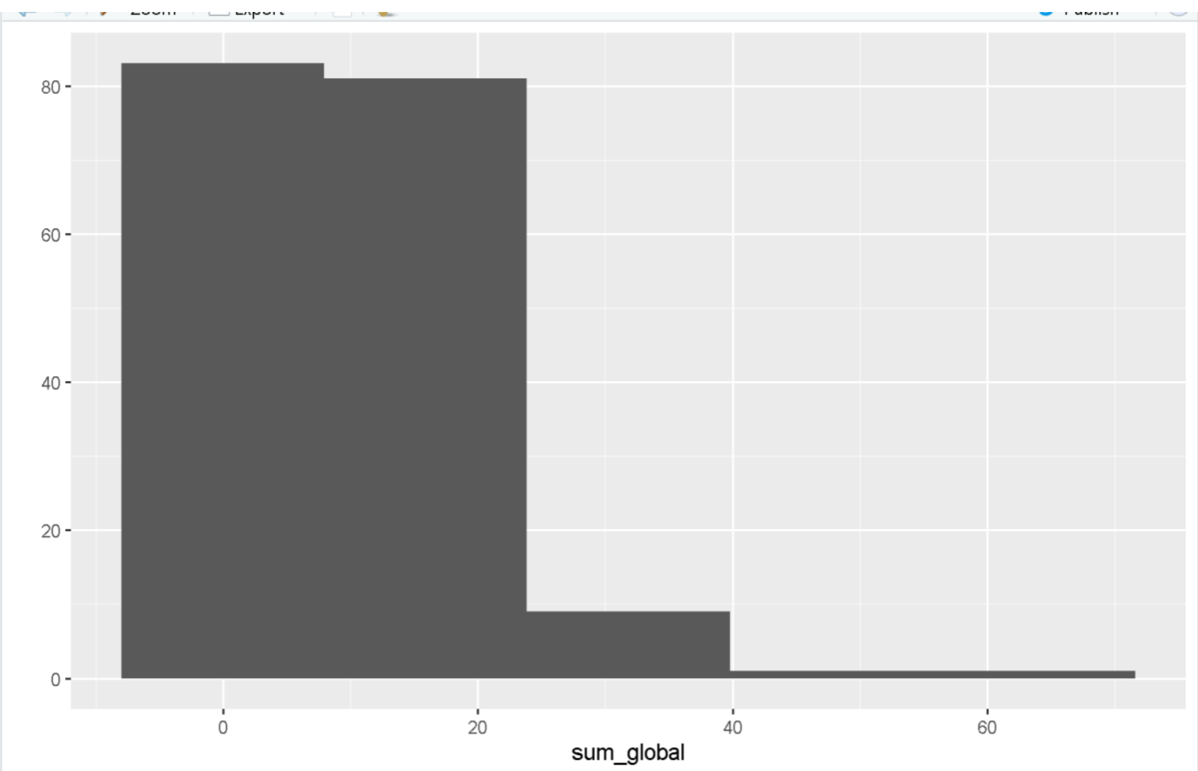
sum_NA
30
20
10
0
0
2500
5000
7500
Product

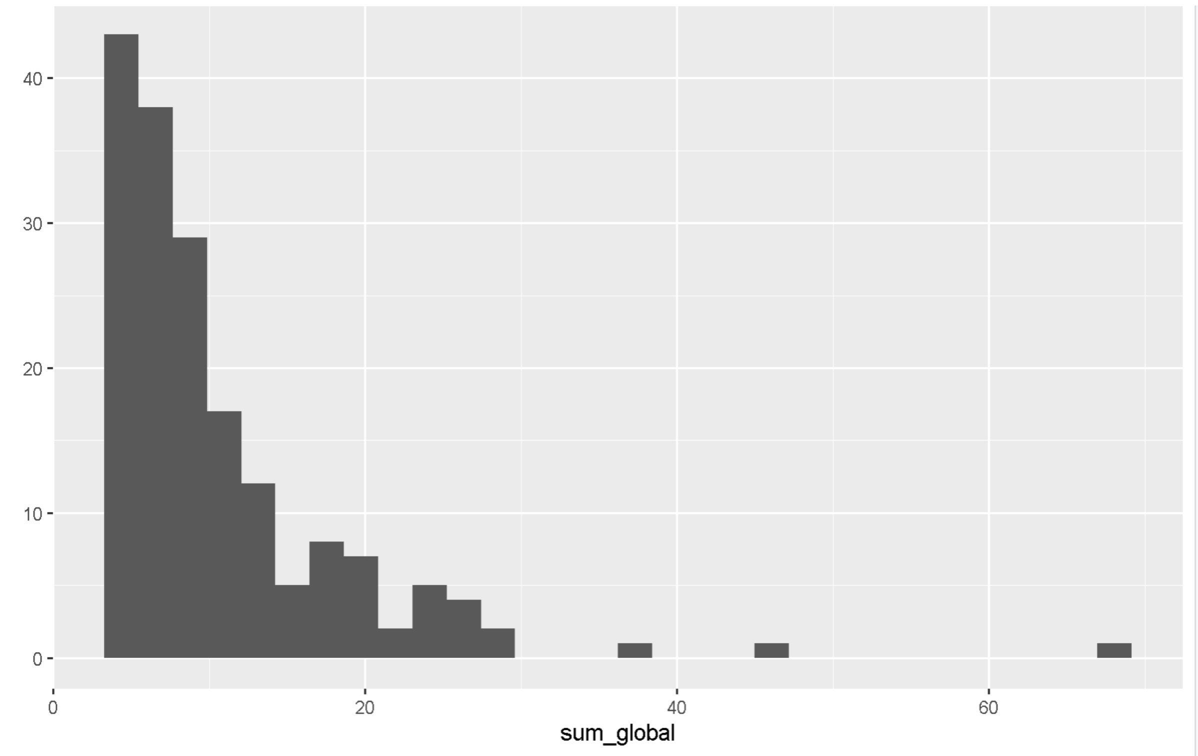

sum_global
60
40
20
0
2500
5000
7500
Product

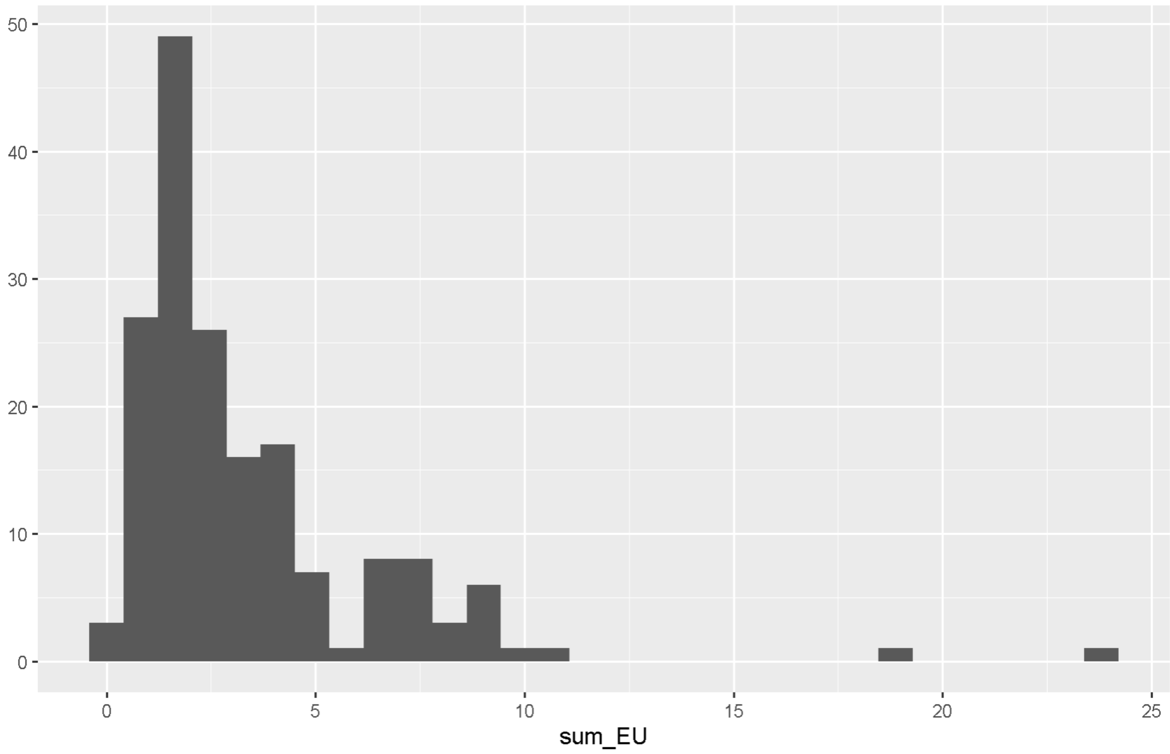

sum_global
60
40
20
0
50
100
150

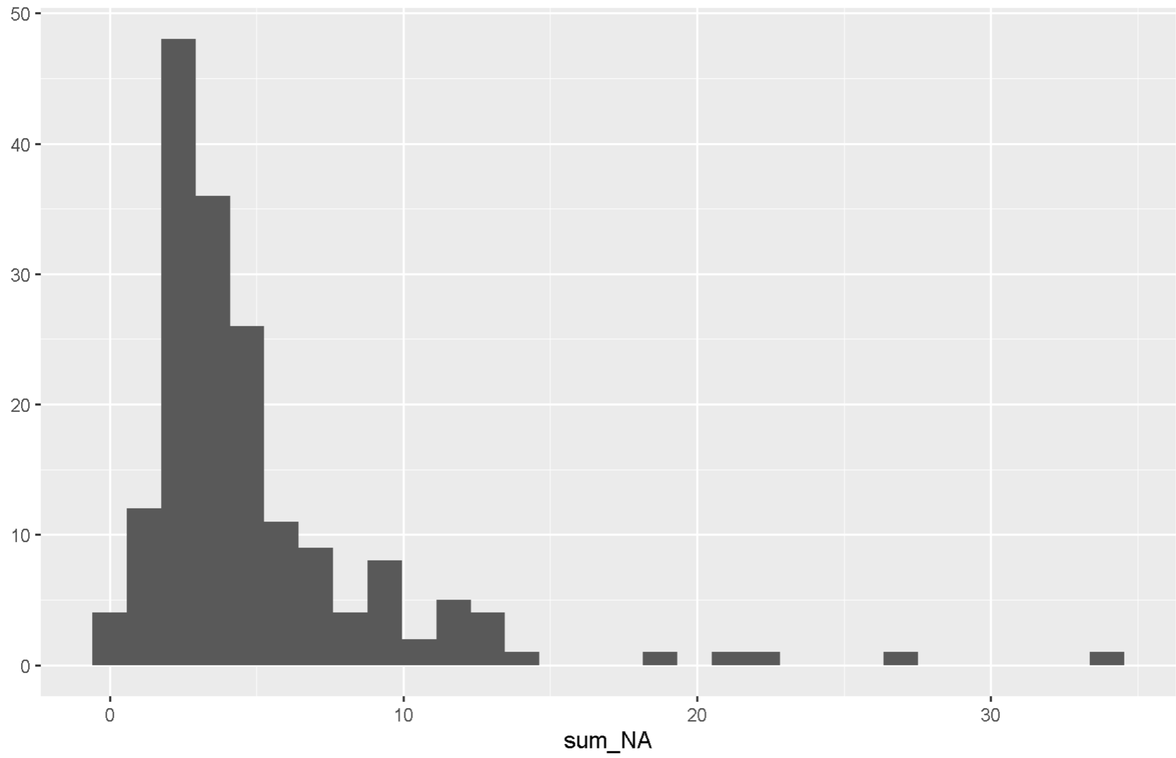

sum_global
60
40
20
0
50
100
150

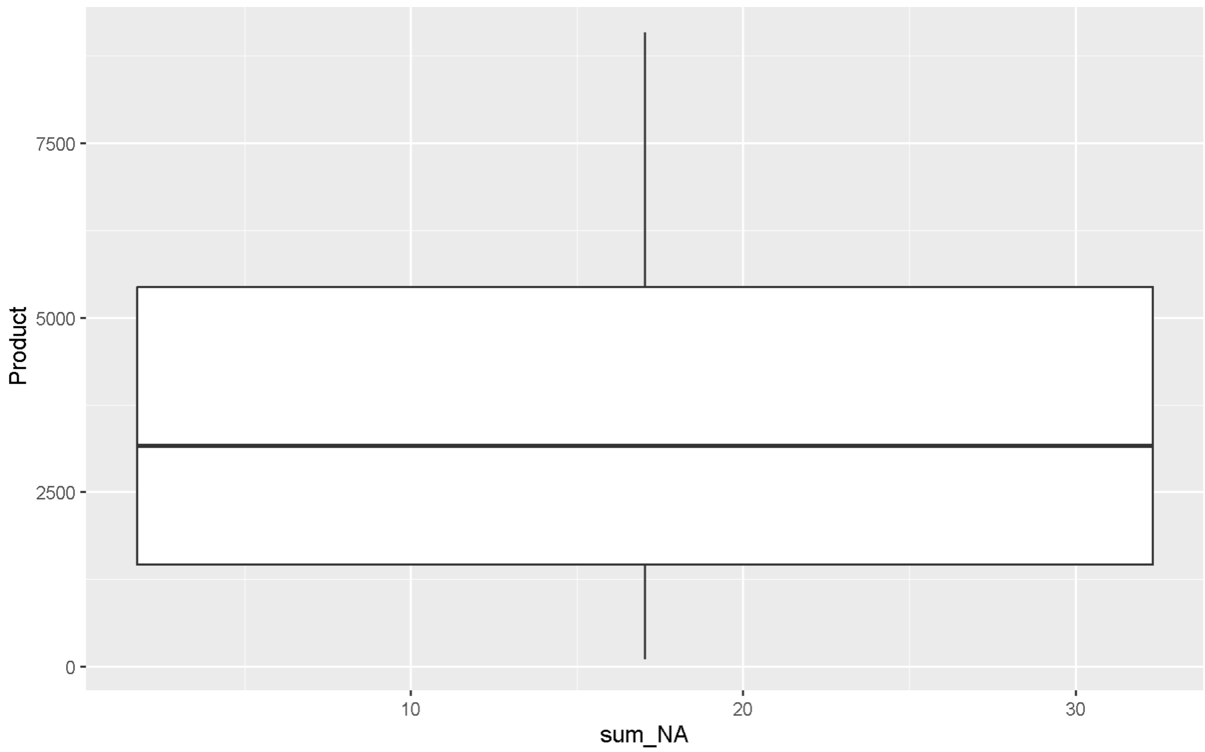

sum_EU
20
15
10
5
0
0
50
100
150

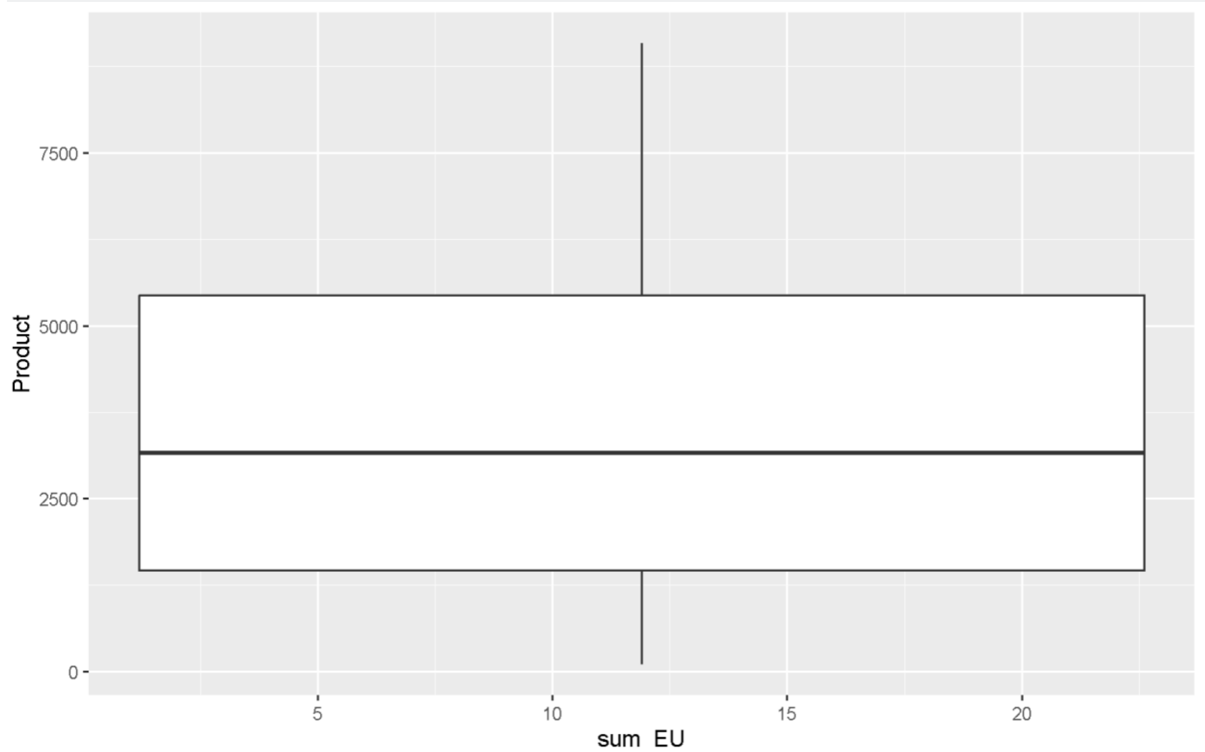

80
60
40
20
0
0
20
40
60
sum_global

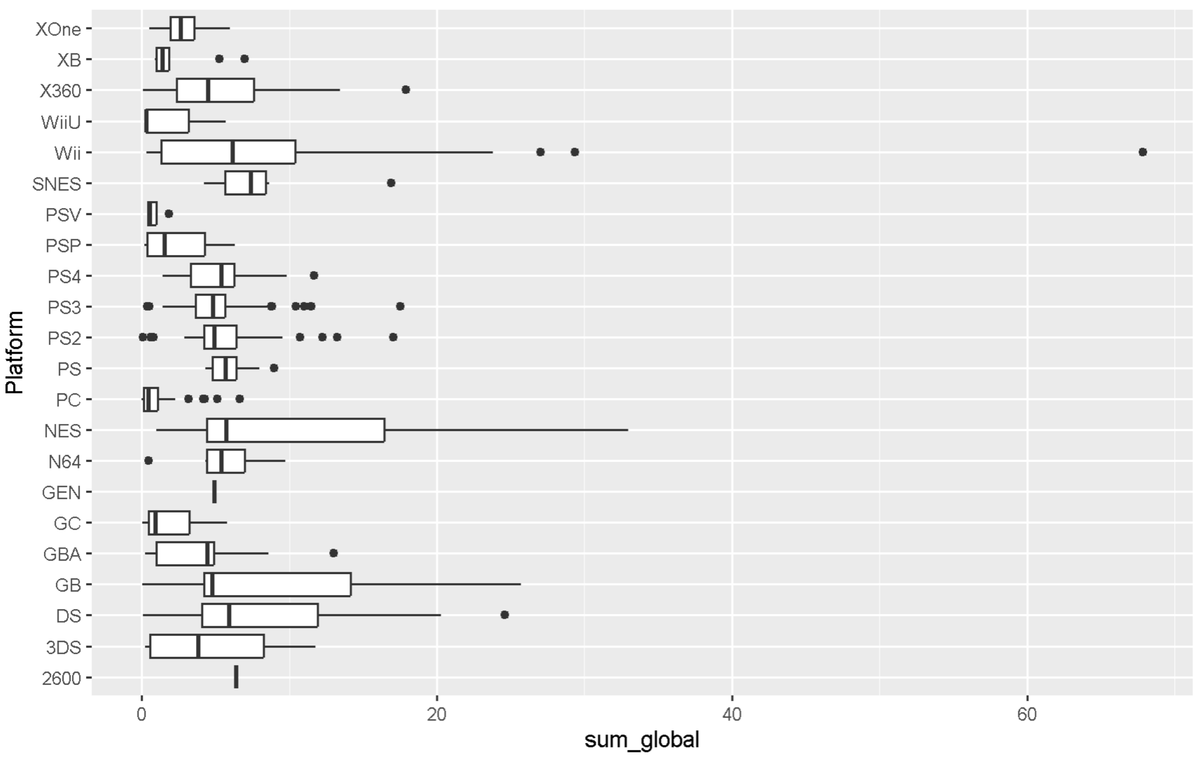

40
30
20
10
0
0
20
40
60
sum_global

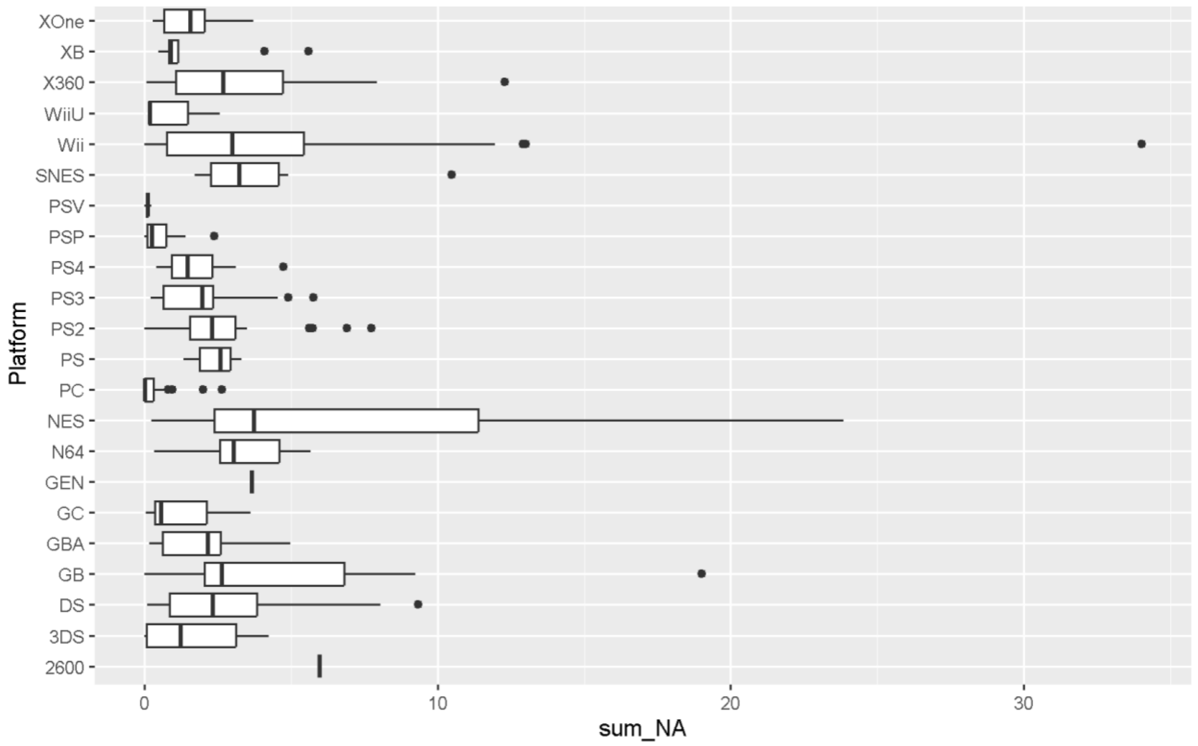

50
40
30
20
10
0
0
5
10
15
20
25
sum_EU

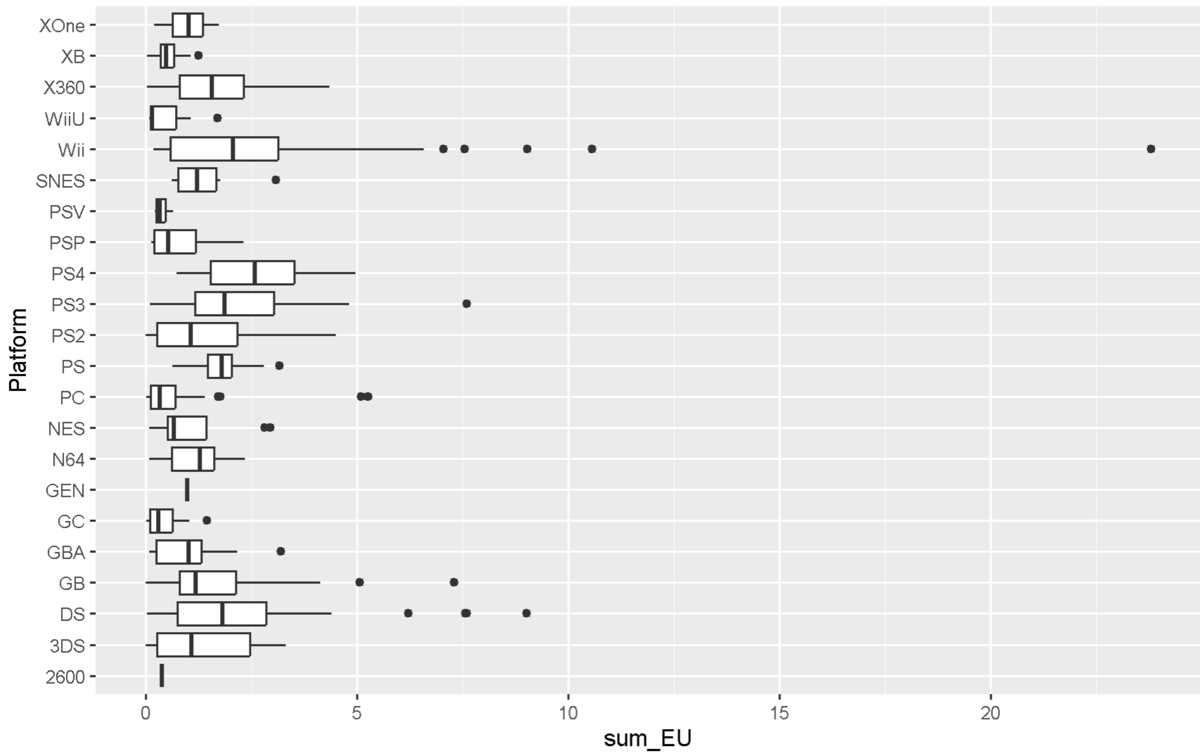

50
40
30
20
10
0
0
10
20
30
sum_NA


Product
7500
5000
2500
0
10
20
30
sum_NA

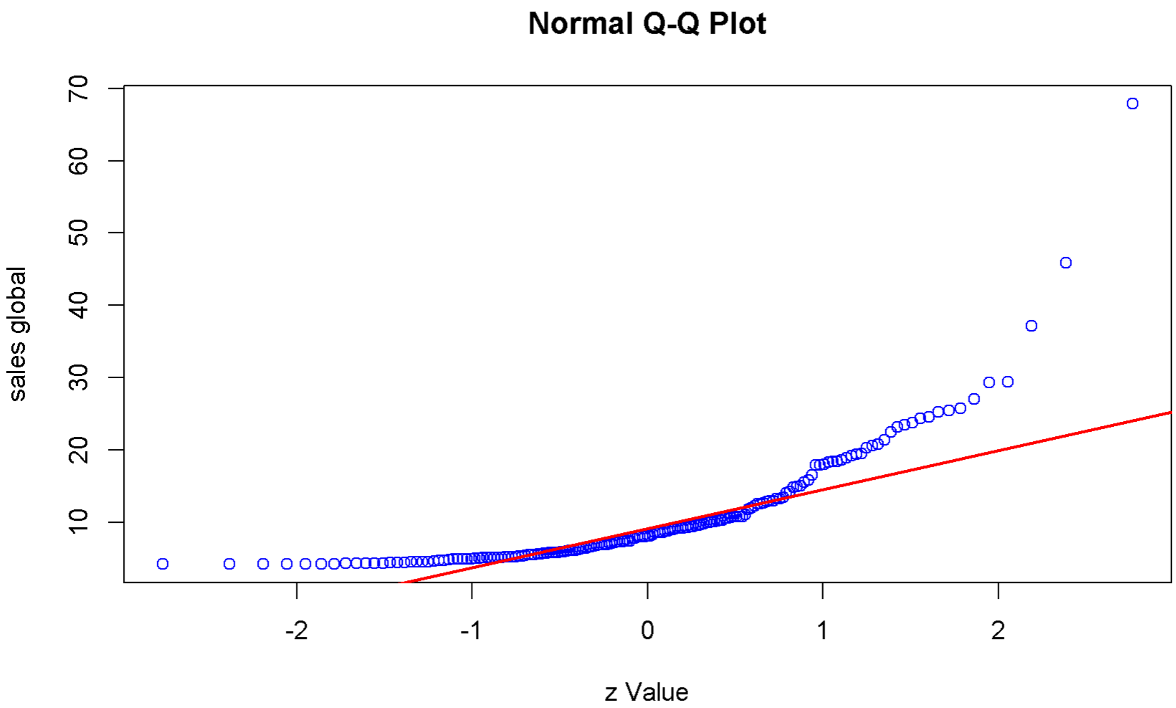

Product
7500
5000
2500
0
5
10
15
20
sum_EU

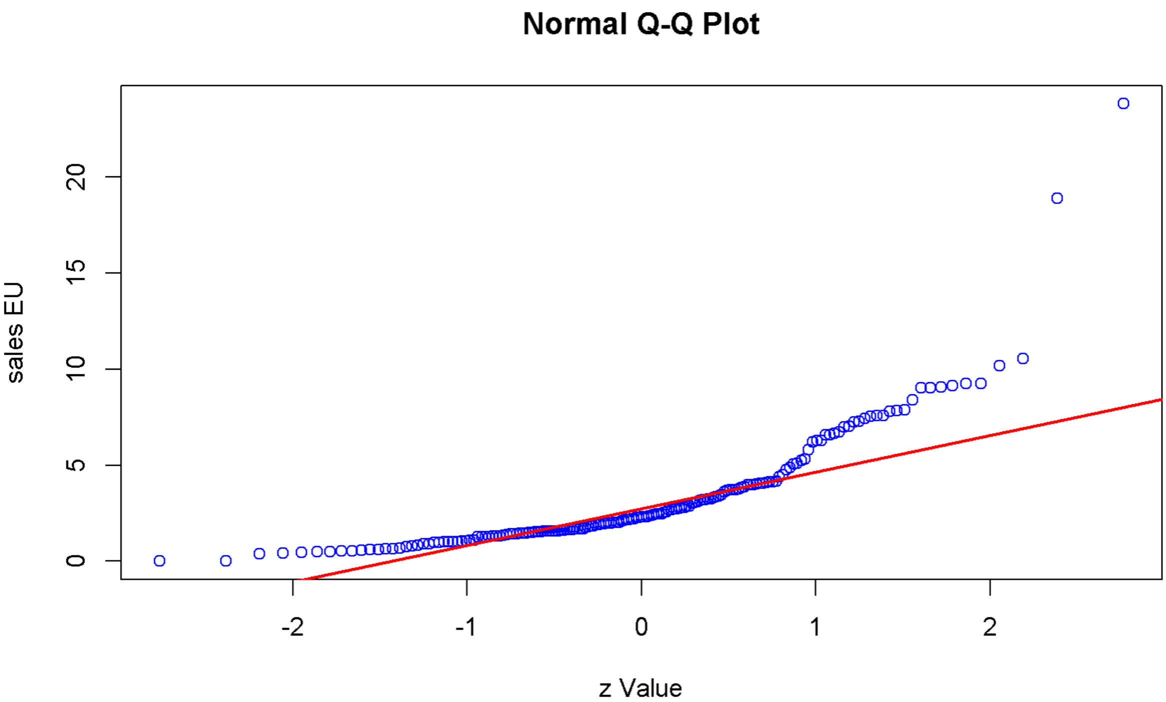

Platform
XOne
XB
X360
WiiU
Wii
SNES
PSV
PSP
PS4
PS3
PS2
PS
PC
NES
N64
GEN
GC
GBA
GB
DS
3DS
2600
0
20
40
60
sum_global


Platform
XOne
XB
X360
WiiU
Wii
SNES
PSV
PSP
PS4
PS3
PS2
PS
PC
NES
N64
GEN
GC
GBA
GB
DS
3DS
2600
0
10
20
30
sum_NA

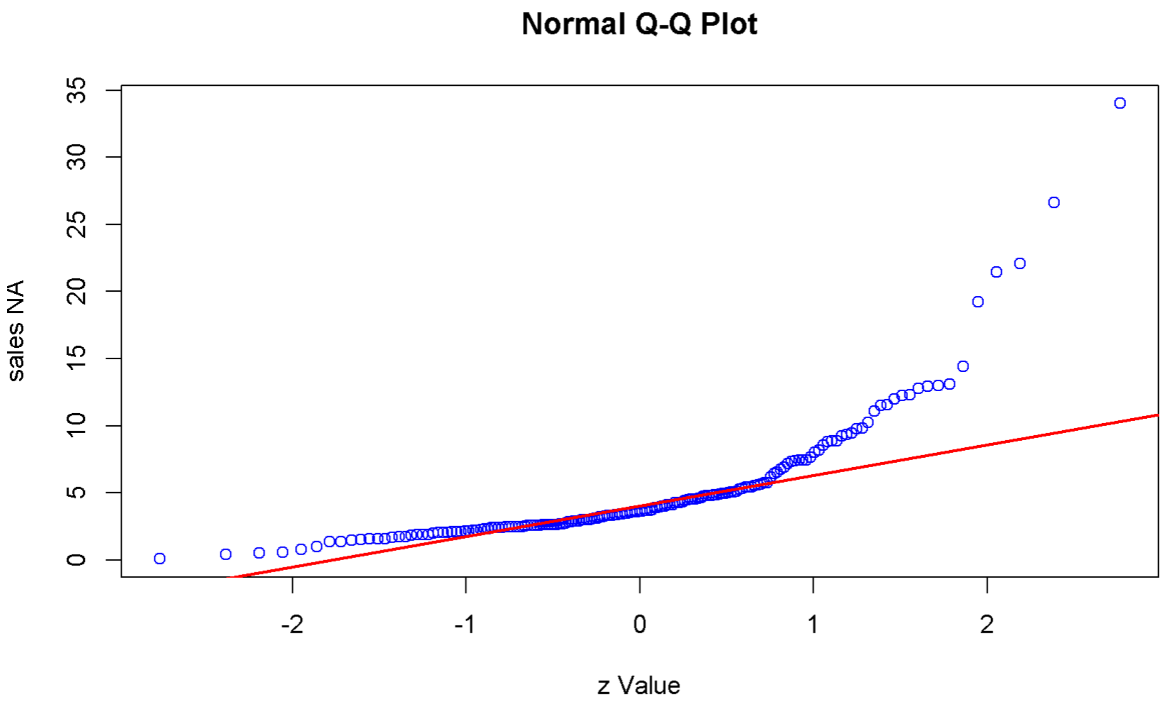



```
> library (moments)
> skewness(df_s_p$sum_global)
[1] 3.066769
> kurtosis(df_s_p$sum_global)
[1] 17.79072
> shapiro.test(df_s_p$sum_global)

        Shapiro-Wilk normality test

data:  df_s_p$sum_global
W = 0.70955, p-value < 2.2e-16
```


Normal Q-Q Plot
sales global
70
60
50
40
30
20
10
-2
-1
0
1
2
z Value




```
> shapiro.test(df_s_p$sum_EU)

        Shapiro-Wilk normality test

data:  df_s_p$sum_EU
W = 0.74058, p-value = 2.987e-16

> skewness(df_s_p$sum_EU)
[1] 2.886029
> kurtosis(df_s_p$sum_EU)
[1] 16.22554
```




```
> shapiro.test(df_s_p$sum_NA)

        Shapiro-Wilk normality test

data:  df_s_p$sum_NA
W = 0.69813, p-value < 2.2e-16

> skewness(df_s_p$sum_NA)
[1] 3.048198
> kurtosis(df_s_p$sum_NA)
[1] 15.6026
> 
```

```
> shapiro.test(df_s_p$sum_NA)

        Shapiro-Wilk normality test

data:  df_s_p$sum_NA
W = 0.69813, p-value < 2.2e-16

> shapiro.test(df_s_p$sum_EU)

        Shapiro-Wilk normality test

data:  df_s_p$sum_EU
W = 0.74058, p-value = 2.987e-16

> shapiro.test(df_s_p$sum_global)

        Shapiro-Wilk normality test

data:  df_s_p$sum_global
W = 0.70955, p-value < 2.2e-16
```