Oberholzer_Amelia_DA301-Assignment_markdown_Rscript.R

2022-12-23

Here I use this markdown document to present the insights I found by analysing the sales data from Turtle Games.

```r
# Import the tidyverse library.
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library('dplyr')
library(tidyr)
require(lattice)
```

```
## Loading required package: lattice
```

```r
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:purrr':
##
##     set_names
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
library(plotly)
```

```
##
## Attaching package: 'plotly'
##
## The following object is masked from 'package:ggplot2':
##
##     last_plot
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following object is masked from 'package:graphics':
```

```
##
##     layout
```

```
library(patchwork)
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library (moments)
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

First I do some data cleaning to prepare the data for analysis.

```
# Set the plots margin size
par(mfrow=c(1,1))

# Import the data set.

data <- read.csv("/Users/ameliaoberholzer/Documents/R Assignment/turtle_sales.csv", header=TRUE, string


# Identifying null values
sum(is.na(data$size))
```

```
## [1] 0
```

```
# Quickly identify max and min of global sales
summary(data)
```

```
##     Ranking            Product       Platform             Year     
##  Min.   :    1.00   Min.   : 107   Length:352         Min.   :1982  
##  1st Qu.:   88.75   1st Qu.:1945   Class :character   1st Qu.:2003  
##  Median :  176.50   Median :3340   Mode  :character   Median :2009  
##  Mean   : 1428.02   Mean   :3607                      Mean   :2007  
##  3rd Qu.: 1439.75   3rd Qu.:5436                      3rd Qu.:2012  
##  Max.   :16096.00   Max.   :9080                      Max.   :2016  
##                                                       NA's   :2     
##     Genre            Publisher            NA_Sales         EU_Sales     
##  Length:352         Length:352         Min.   : 0.0000   Min.   : 0.000  
##  Class :character   Class :character   1st Qu.: 0.4775   1st Qu.: 0.390  
##  Mode  :character   Mode  :character   Median : 1.8200   Median : 1.170  
##                                        Mean   : 2.5160   Mean   : 1.644  
##                                        3rd Qu.: 3.1250   3rd Qu.: 2.160  
##                                        Max.   :34.0200   Max.   :23.800  
```

```
##
##   Global_Sales
##  Min.   : 0.010
##  1st Qu.: 1.115
##  Median : 4.320
##  Mean   : 5.335
##  3rd Qu.: 6.435
##  Max.   :67.850
##
```

```
# Use the glimpse() function.
glimpse(data)
```

```
## Rows: 352
## Columns: 9
## $ Ranking      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17~
## $ Product      <int> 107, 123, 195, 231, 249, 254, 263, 283, 291, 326, 399, 40~
## $ Platform     <chr> "Wii", "NES", "Wii", "Wii", "GB", "GB", "DS", "Wii", "Wii~
## $ Year         <dbl> 2006, 1985, 2008, 2009, 1996, 1989, 2006, 2006, 2009, 198~
## $ Genre        <chr> "Sports", "Platform", "Racing", "Sports", "Role-Playing",~
## $ Publisher    <chr> "Nintendo", "Nintendo", "Nintendo", "Nintendo", "Nintendo~
## $ NA_Sales     <dbl> 34.02, 23.85, 13.00, 12.92, 9.24, 19.02, 9.33, 11.50, 11.~
## $ EU_Sales     <dbl> 23.80, 2.94, 10.56, 9.03, 7.29, 1.85, 7.57, 7.54, 5.79, 0~
## $ Global_Sales <dbl> 67.85, 33.00, 29.37, 27.06, 25.72, 24.81, 24.61, 23.80, 2~
```

```
# Changing the date
data$Year <- lubridate::ymd(data$Year, truncated = 2L)

# Changing product

data$Product <- as.character(data$Product)

# Checking data types
str(data)
```

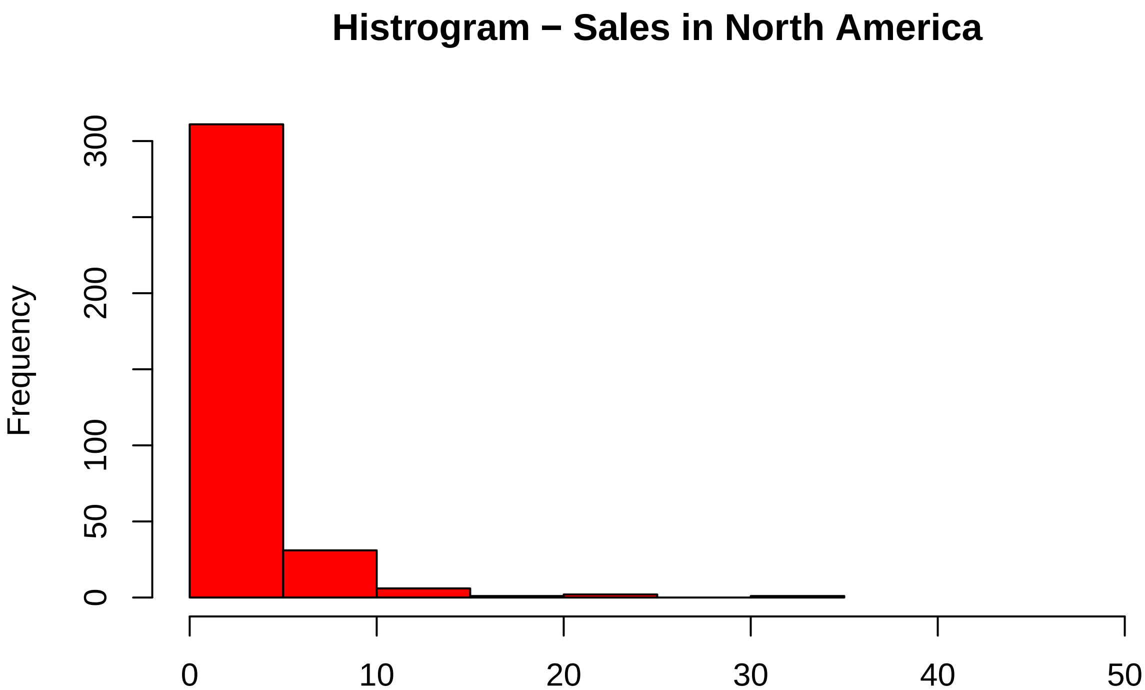
```
## 'data.frame':    352 obs. of  9 variables:
##  $ Ranking     : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Product     : chr  "107" "123" "195" "231" ...
##  $ Platform    : chr  "Wii" "NES" "Wii" "Wii" ...
##  $ Year        : Date, format: "2006-01-01" "1985-01-01" ...
##  $ Genre       : chr  "Sports" "Platform" "Racing" "Sports" ...
##  $ Publisher   : chr  "Nintendo" "Nintendo" "Nintendo" "Nintendo" ...
##  $ NA_Sales    : num  34.02 23.85 13 12.92 9.24 ...
##  $ EU_Sales    : num  23.8 2.94 10.56 9.03 7.29 ...
##  $ Global_Sales: num  67.8 33 29.4 27.1 25.7 ...
```

```
# Now I'm going to get an overview of the sales data
# Drop Ranking, Year, Genre and Publisher columns
data_sales <- select(data, -Ranking, -Year, -Genre, -Publisher)
```

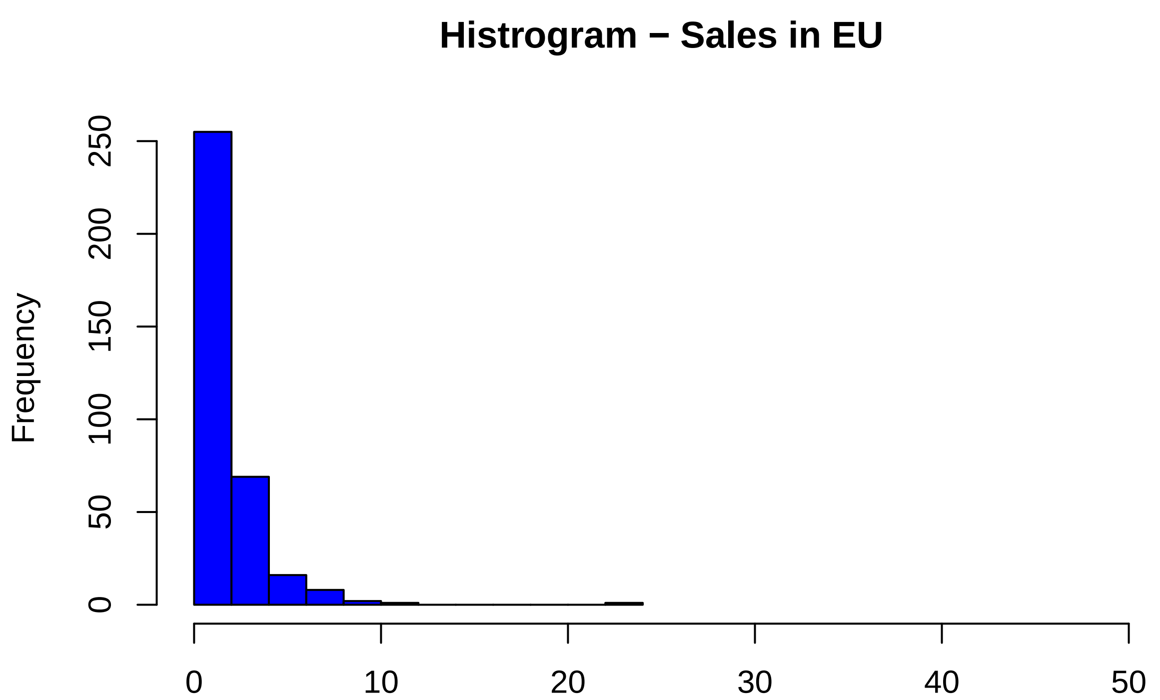
I will now start to create simple visulisations to understand how individual products from Turtle Games affect sales.

```
# See some histograms to show the distribution of sales
# In different regions

hist(data_sales$NA_Sales,
```

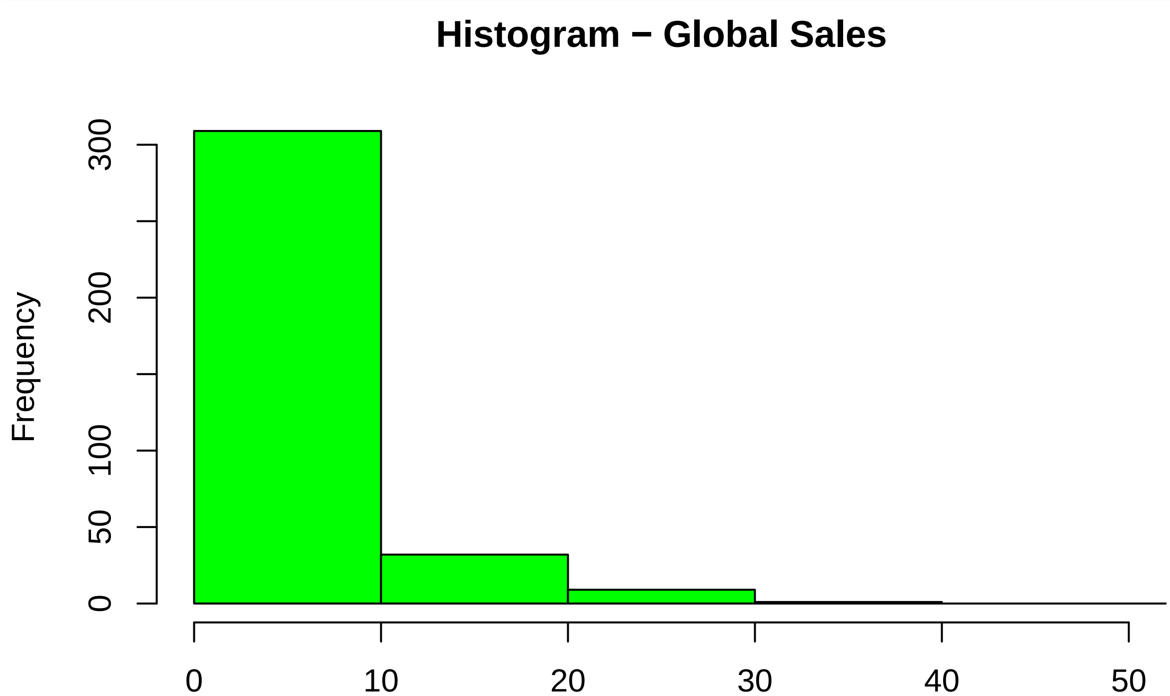
```
     xlim = c(0, 50),
     breaks = 9,
     main = "Histrogram - Sales in North America",
     xlab = "",
     col = "red")
```




```
hist(data_sales$EU_Sales,
     xlim = c(0, 50),
     breaks = 9,
     main = "Histrogram - Sales in EU",
     xlab = "",
     col = "Blue")
```



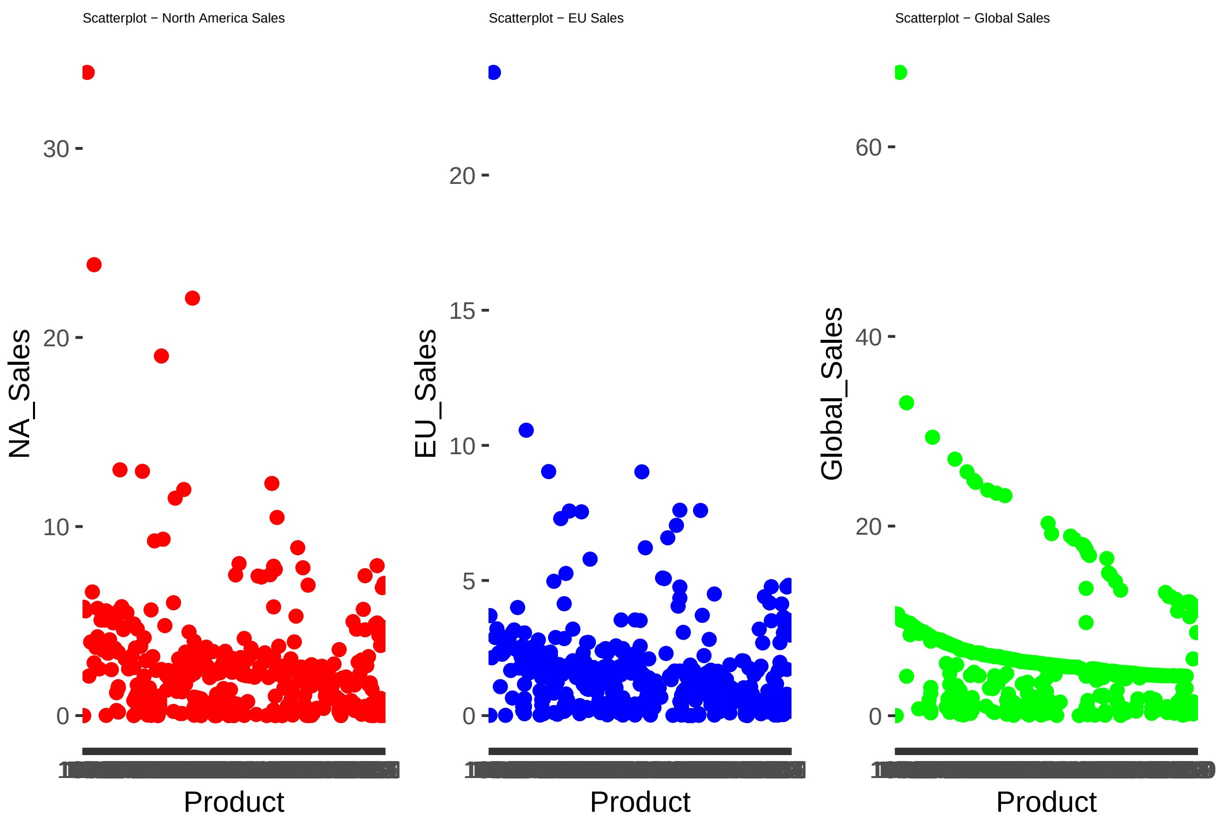

```
hist(data_sales$Global_Sales,
     xlim = c(0, 50),
     breaks = 9,
     main = "Histogram - Global Sales",
     xlab = "",
     col = "Green")
```



These graphs show the distribution of sales across different regions

```
# Distribution of sales in EU, NA and the World
```

```r
# Most basic scatter chart
NA_Sales_distribution <- ggplot(data_sales, aes(x=Product, y=NA_Sales)) +
  geom_point(color="red",size=2) +
  theme(plot.title = element_text(size = 5)) +
  ggtitle("Scatterplot - North America Sales")

EU_Sales_distribution <- ggplot(data_sales, aes(x=Product, y=EU_Sales)) +
  geom_point(color="Blue",size=2) +
  theme(plot.title = element_text(size = 5)) +
  ggtitle("Scatterplot - EU Sales")

Global_Sales_distribution <- ggplot(data_sales, aes(x=Product, y=Global_Sales)) +
  geom_point(color="green",size=2) +
  theme(plot.title = element_text(size = 5)) +
  ggtitle("Scatterplot - Global Sales")

NA_Sales_distribution + EU_Sales_distribution + Global_Sales_distribution
```

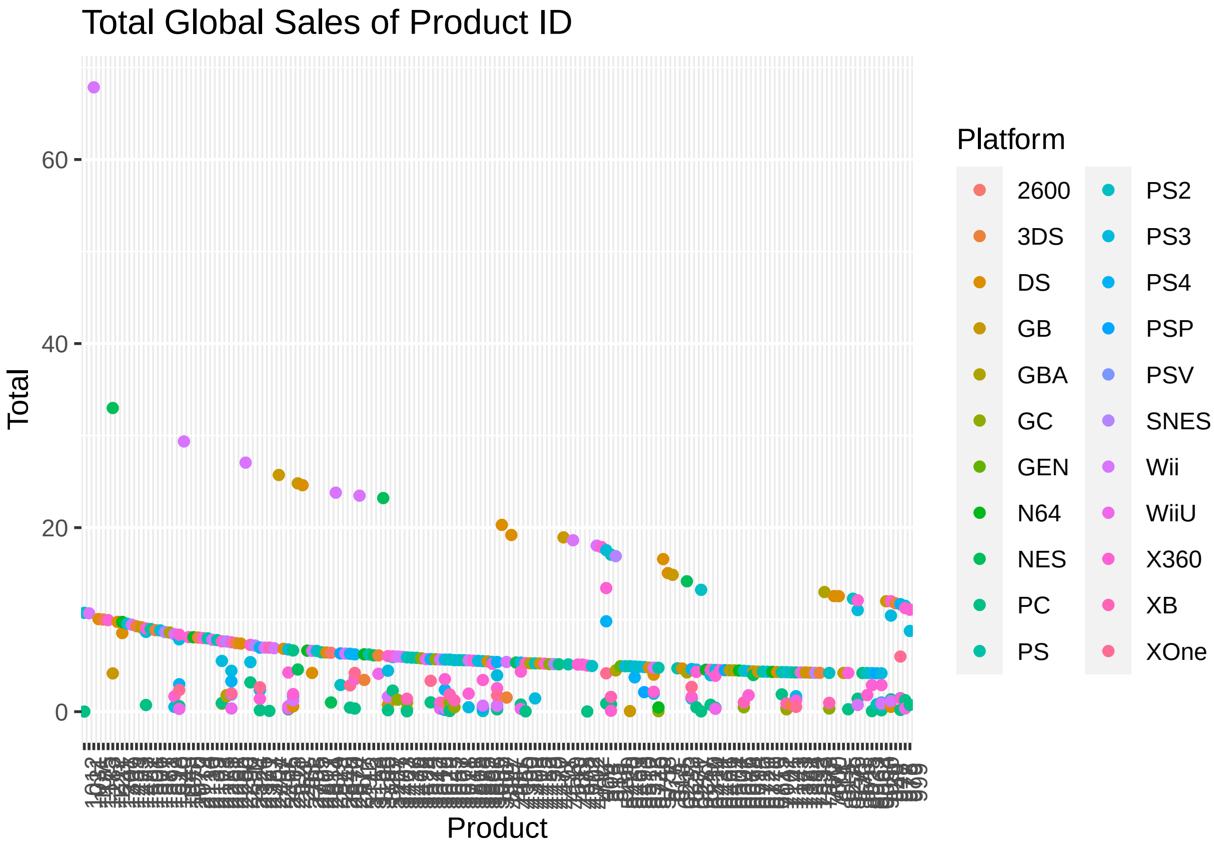


Here most of the products have sales in between 0 and 5 million however some products have sold exceptionally well. I will go ahead to identify top selling products as well as products that haven't sold as well.

```r
# Use group by function() to get the sum
# Of sales for certain platform
sum_sales_product_global <- data %>%
  group_by(Product, Platform, Global_Sales) %>%
  summarise(Total = sum(Global_Sales))
```

```
## `summarise()` has grouped output by 'Product', 'Platform'. You can override
## using the `.groups` argument.
```

```r
# Viewing the count of sales for a certain game
# Creating a scatter plot for this information
global_sales_plot <- ggplot(sum_sales_product_global,
                            aes(x = Product, y = Total, colour = Platform)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 90)) +
  ggtitle("Total Global Sales of Product ID")
global_sales_plot
```



Here we can see which products have higher sales recorded depending on their sales platform. Interestingly Wii has the highest sale of a certain product. Almost by double all other products. In general this product has been a top seller at Turtle Games.

```r
# Top 15 and their platform
ordered_scatter_product = sum_sales_product_global[order(sum_sales_product_global$Total, decreasing = T
ordered_scatter_product
```

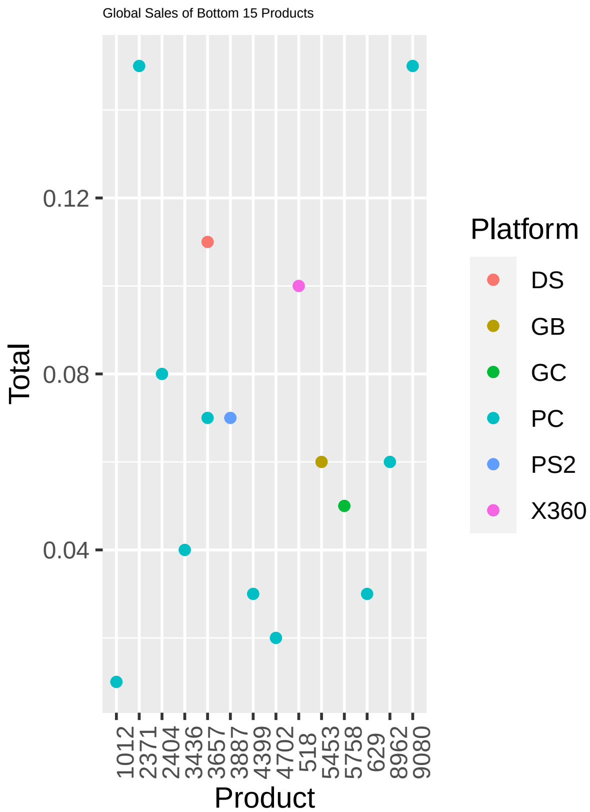
```
## # A tibble: 352 x 4
## # Groups:   Product, Platform [352]
##    Product Platform Global_Sales Total
##    <chr>   <chr>           <dbl> <dbl>
##  1 107     Wii              67.8  67.8
##  2 123     NES              33    33
##  3 195     Wii              29.4  29.4
##  4 231     Wii              27.1  27.1
##  5 249     GB               25.7  25.7
##  6 254     GB               24.8  24.8
##  7 263     DS               24.6  24.6
```

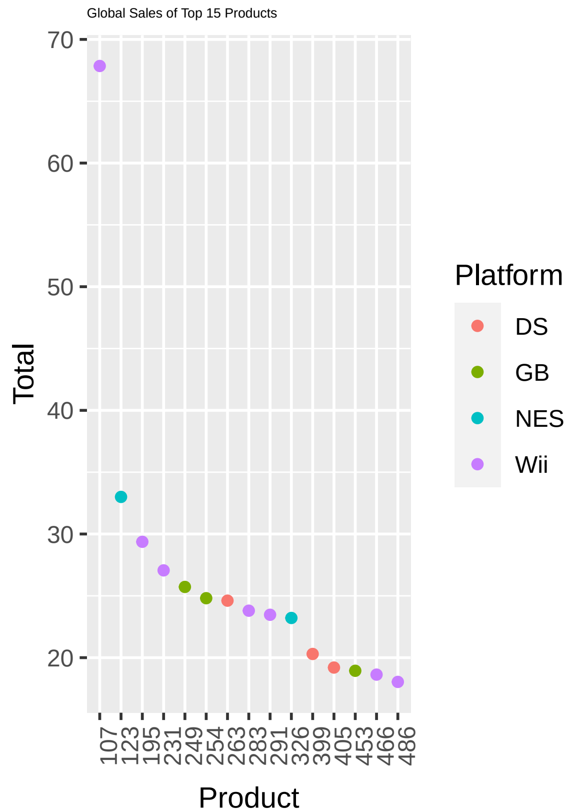
```
##  8 283     Wii               23.8  23.8
##  9 291     Wii               23.5  23.5
## 10 326     NES               23.2  23.2
## # ... with 342 more rows
```

```
ordered_scatter_product_top <- head(ordered_scatter_product, 15)
global_sales_plot_top <- ggplot(ordered_scatter_product_top,
                               mapping = aes(x = Product, y = Total, colour = Platform)) +
  geom_point() +
  theme(plot.title = element_text(size = 5)) +
  theme(axis.text.x = element_text(angle = 90)) +
  ggtitle("Global Sales of Top 15 Products")
# Bottom 15 and their platform
ordered_scatter_product = sum_sales_product_global[order(sum_sales_product_global$Total, decreasing = T
ordered_scatter_product
```

```
## # A tibble: 352 x 4
## # Groups:   Product, Platform [352]
##    Product Platform Global_Sales Total
##    <chr>   <chr>           <dbl> <dbl>
##  1 107     Wii              67.8  67.8
##  2 123     NES              33    33
##  3 195     Wii              29.4  29.4
##  4 231     Wii              27.1  27.1
##  5 249     GB               25.7  25.7
##  6 254     GB               24.8  24.8
##  7 263     DS               24.6  24.6
##  8 283     Wii              23.8  23.8
##  9 291     Wii              23.5  23.5
## 10 326     NES              23.2  23.2
## # ... with 342 more rows
```

```
ordered_scatter_product_bottom <- tail(ordered_scatter_product, 15)
global_sales_plot_bottom <- ggplot(ordered_scatter_product_bottom,
                                  mapping = aes(x = Product, y = Total, colour = Platform)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 90)) +
  theme(plot.title = element_text(size = 5)) +
  ggtitle("Global Sales of Bottom 15 Products")
global_sales_plot_bottom + global_sales_plot_top
```



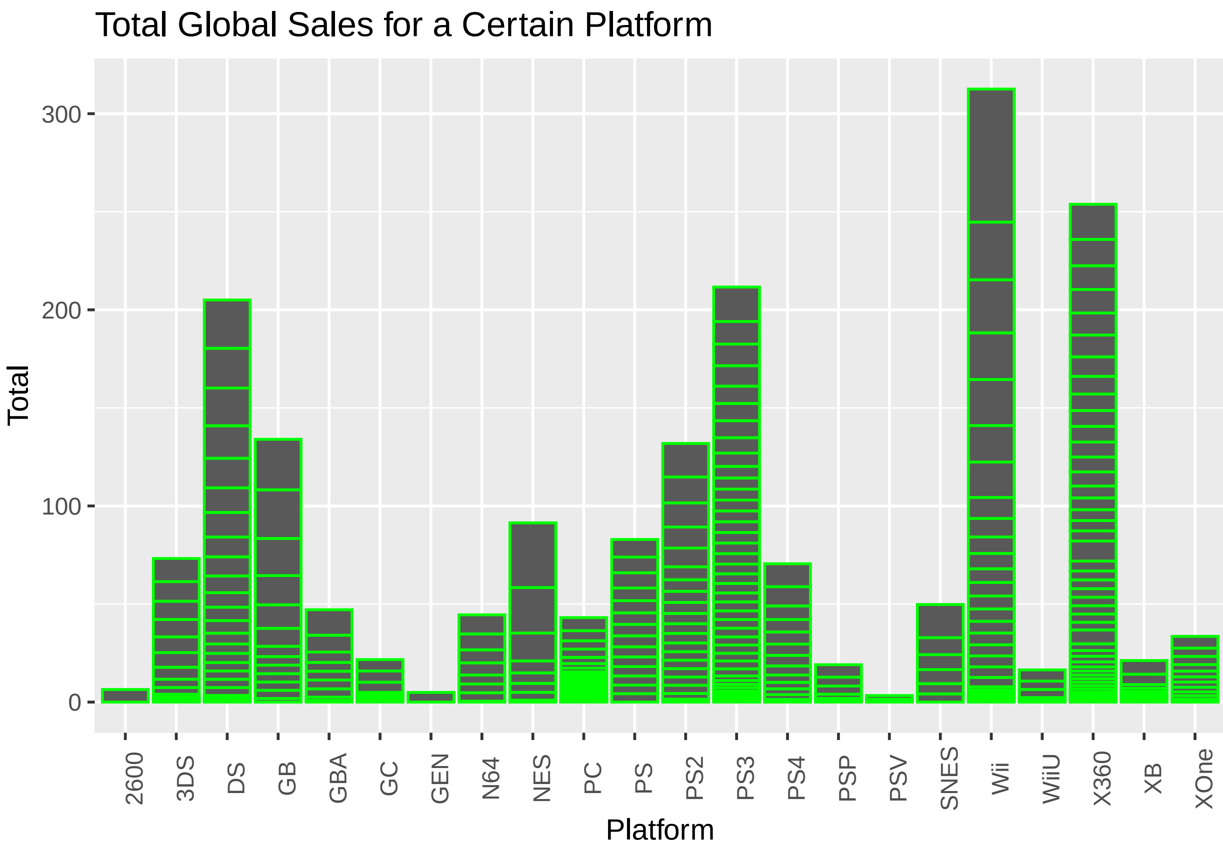


We see here how the bottom and top plots compare. Interestingly PC have the lowest sales performance. Whereas Will, NES and GB have higher sales performance. You can also see the individual product numbers. Turtle Games can use their directory to identify which product corresponds to the number to gain more information. It is interesting to see how some platforms are more popular in different regions. I will now go on to investigate this!

```
# Use group by function() to get the sum
# Of sales for certain platform
sum_sales_platform_global <- data %>%
  group_by(Platform, Global_Sales) %>%
  summarise(Total = sum(Global_Sales))
```

```
## `summarise()` has grouped output by 'Platform'. You can override using the
## `.groups` argument.
```

```
# Barplot showing the total sales for certain platforms
# On some version this bar plot comes out with horizontal lines
# I'm not sure why this is the case
global_sales_platfrom_plot <- ggplot(sum_sales_platform_global,
                                     aes(x = Platform, y = Total)) +
  geom_bar(stat="identity", colour="green") +
  theme(axis.text.x = element_text(angle = 90)) +
  ggtitle("Total Global Sales for a Certain Platform")
global_sales_platfrom_plot
```



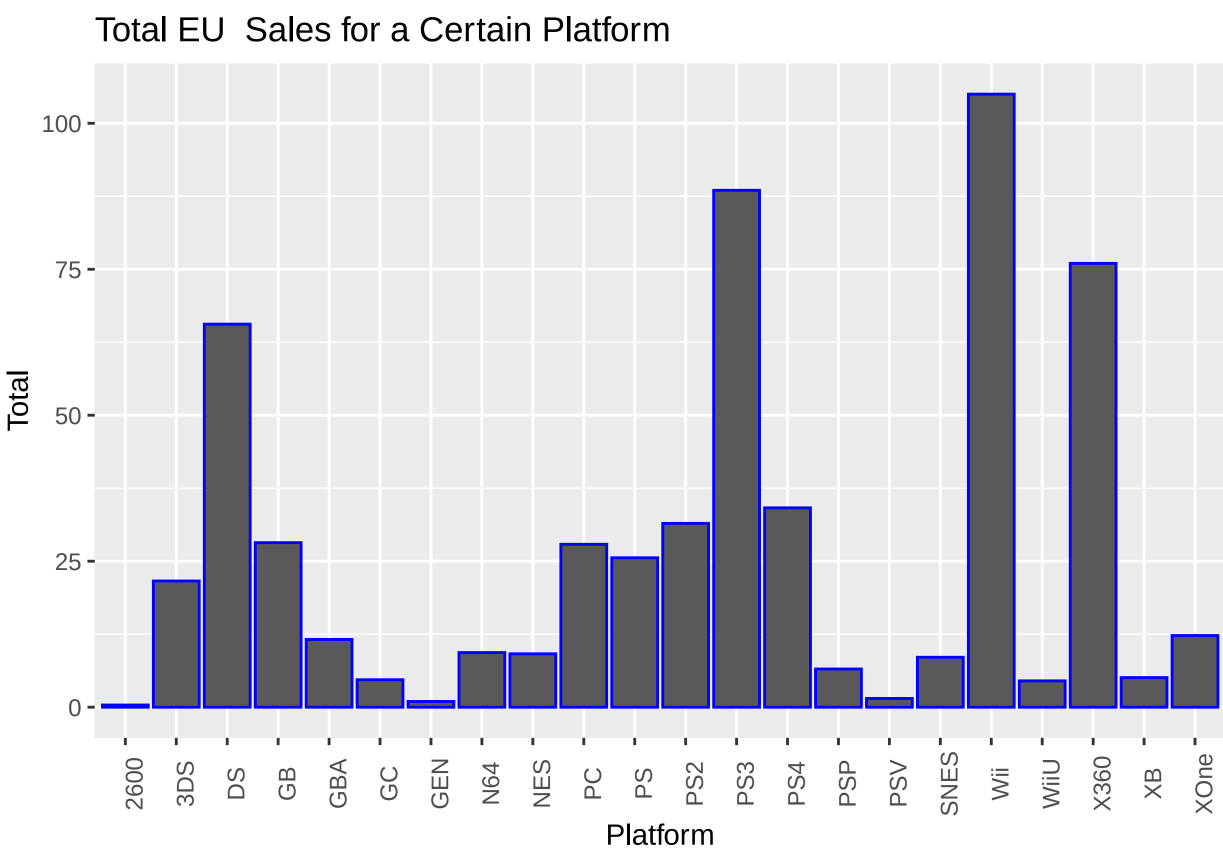

```r
# Comparison of most sales in certain platform for EU versus NA Sales
# Grouping sum of EU sales for different platforms
sum_sales_platform_EU = data %>%
  group_by(Platform) %>%
  summarise(Total = sum(EU_Sales))
sum_sales_platform_EU
```

```
## # A tibble: 22 x 2
##    Platform Total
##    <chr>    <dbl>
##  1 2600      0.37
##  2 3DS      21.6
##  3 DS       65.6
##  4 GB       28.2
##  5 GBA      11.6
##  6 GC        4.7
##  7 GEN       0.98
##  8 N64       9.36
##  9 NES       9.14
## 10 PC       27.9
## # ... with 12 more rows
```

```r
# Grouping sum of NA sales for different platforms
sum_sales_platform_NA = data %>%
  group_by(Platform) %>%
  summarise(Total = sum(NA_Sales))
sum_sales_platform_NA
```

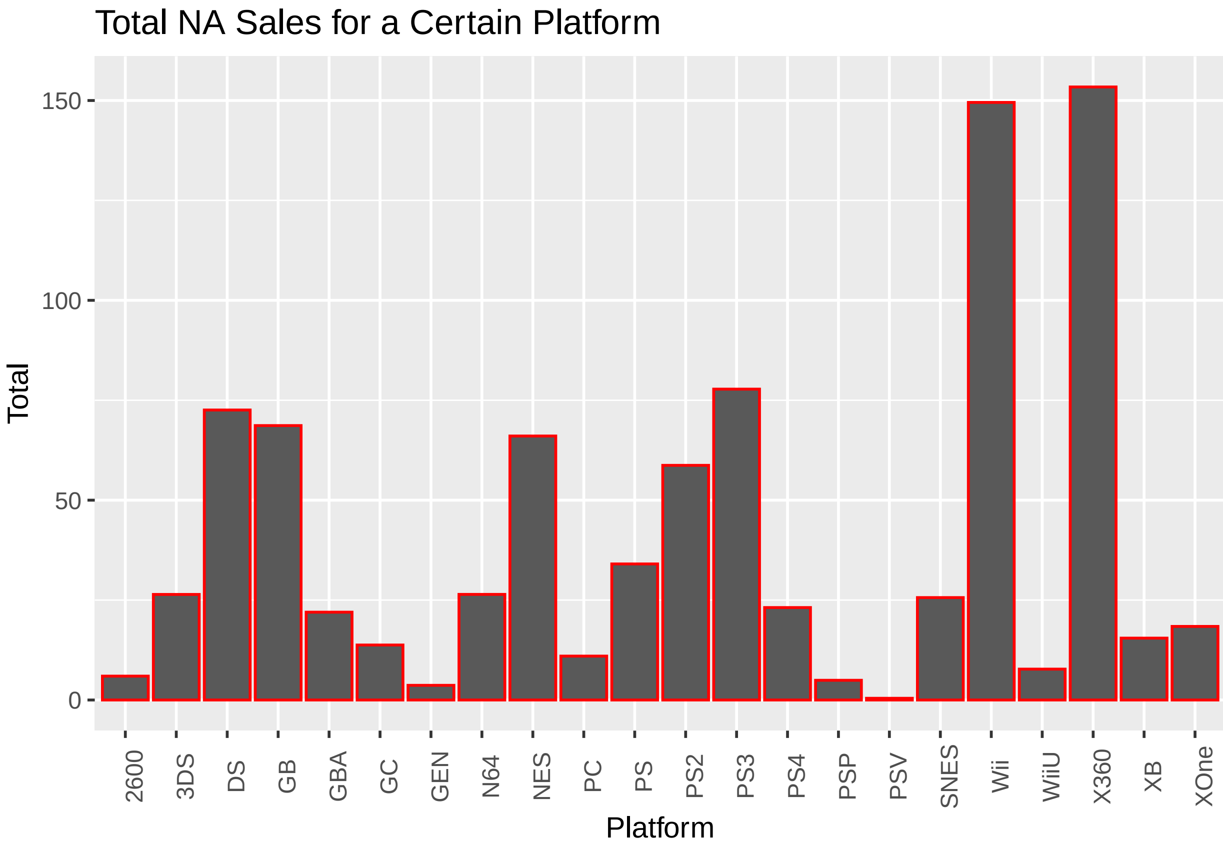
```
## # A tibble: 22 x 2
##    Platform Total
##    <chr>    <dbl>
##  1 2600      5.97
##  2 3DS      26.4
##  3 DS       72.6
##  4 GB       68.7
##  5 GBA      22.0
##  6 GC       13.8
##  7 GEN       3.67
##  8 N64      26.4
##  9 NES      66.0
## 10 PC       11.0
## # ... with 12 more rows
```

```
# Creating Bar Plots
EU_sales_platfrom_plot <- ggplot(sum_sales_platform_EU,
                                 aes(x = Platform, y = Total)) +
  geom_bar(stat="identity", colour="blue") +
  theme(axis.text.x = element_text(angle = 90)) +
  ggtitle("Total EU  Sales for a Certain Platform")
EU_sales_platfrom_plot
```




```
# Bar plot for NA
NA_sales_platfrom_plot <- ggplot(sum_sales_platform_NA,
                                 aes(x = Platform, y = Total)) +
  geom_bar(stat="identity", colour="red") +
  theme(axis.text.x = element_text(angle = 90)) +
```

```
  ggtitle("Total NA Sales for a Certain Platform")
NA_sales_platfrom_plot
```



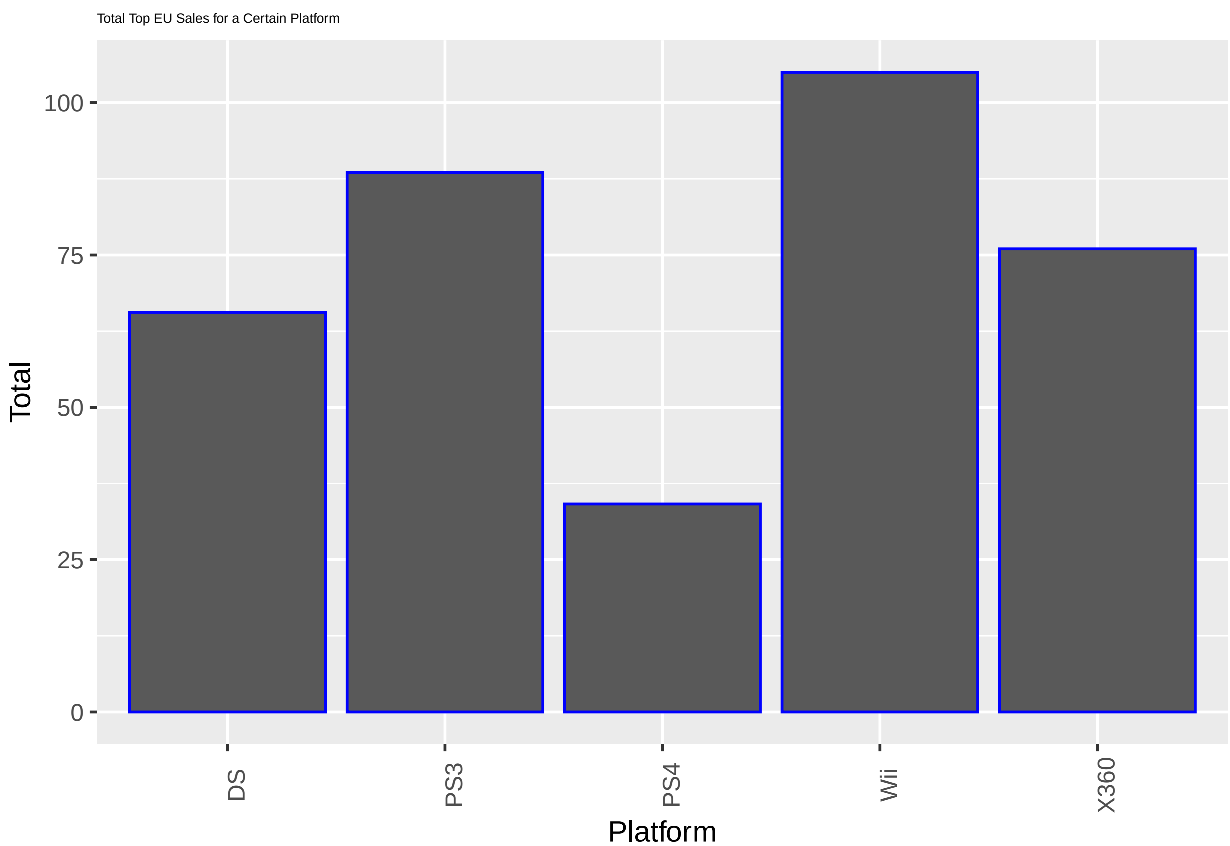
I

will now filter out the highest platform sales in EU and NA for a clear comparison

```
# First EU
EU_highest_platform_sales <- sum_sales_platform_EU %>%
  arrange(desc(Total)) %>%
  group_by(Platform)
EU_highest_platform_sales_top <- head(EU_highest_platform_sales ,5)
EU_highest_platform_sales_top
```

```
## # A tibble: 5 x 2
## # Groups:   Platform [5]
##   Platform Total
##   <chr>    <dbl>
## 1 Wii      105.
## 2 PS3       88.5
## 3 X360      76.0
## 4 DS        65.6
## 5 PS4       34.1
```

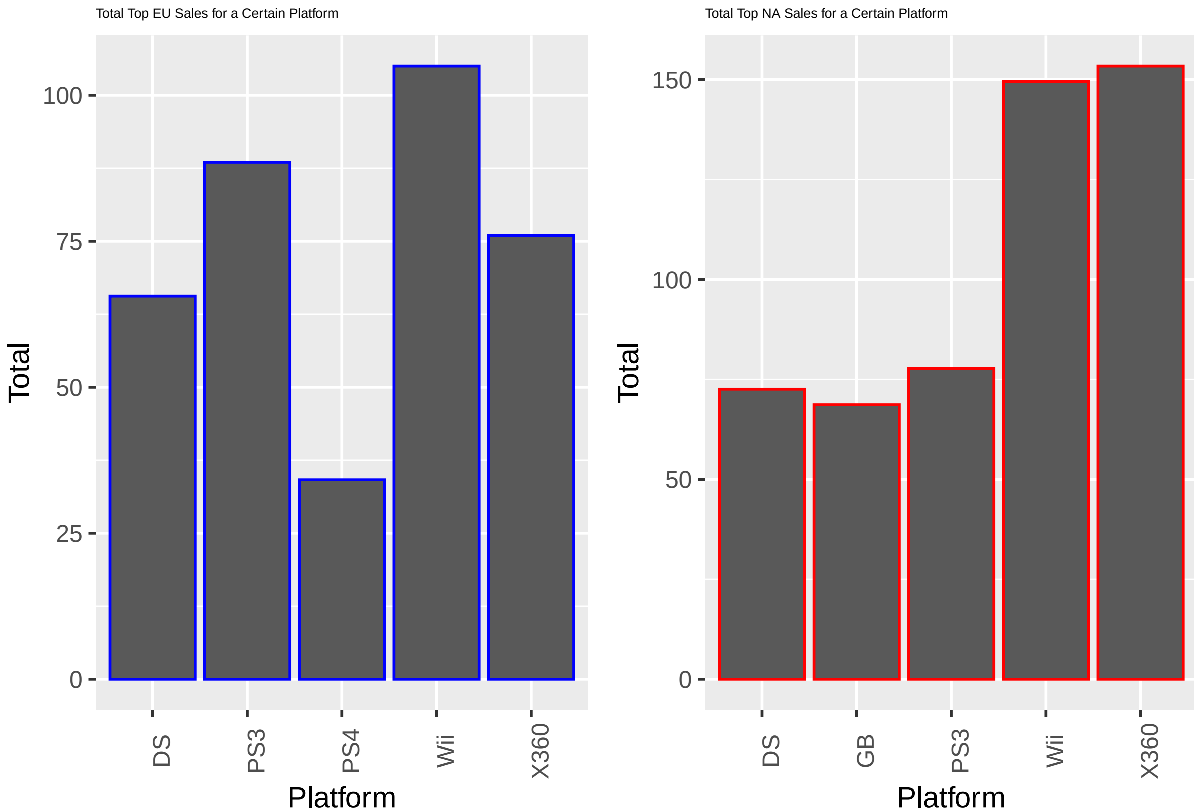
```
EU_highest_platform_sales_top_plot <- ggplot(EU_highest_platform_sales_top,
                                             aes(x = Platform, y = Total)) +
  geom_bar(stat="identity", colour="Blue") +
  theme(plot.title = element_text(size = 5)) +
  theme(axis.text.x = element_text(angle = 90)) +
  ggtitle("Total Top EU Sales for a Certain Platform")
EU_highest_platform_sales_top_plot
```




```
# Now US
NA_highest_platform_sales <- sum_sales_platform_NA %>%
  arrange(desc(Total)) %>%
  group_by(Platform)
NA_highest_platform_sales_top <- head(NA_highest_platform_sales ,5)
NA_highest_platform_sales_top
```

```
## # A tibble: 5 x 2
## # Groups:   Platform [5]
##   Platform Total
##   <chr>    <dbl>
## 1 X360     153.
## 2 Wii      150.
## 3 PS3       77.8
## 4 DS        72.6
## 5 GB        68.7
```

```
NA_highest_platform_sales_top_plot <- ggplot(NA_highest_platform_sales_top,
                                             aes(x = Platform, y = Total)) +
  geom_bar(stat="identity", colour="Red") +
  theme(axis.text.x = element_text(angle = 90)) +
  theme(plot.title = element_text(size = 5)) +
  ggtitle("Total Top NA Sales for a Certain Platform")
EU_highest_platform_sales_top_plot + NA_highest_platform_sales_top_plot
```



Interestingly we see how the Wii is more popular in the EU whereas the Xbox 360 is more popular in the US. This information can information Turtle Games on where to focus marketing when new products have been released from these platforms. Now we're going to observe how sales change over time using group_by to find the total global sales per year.

```r
# using group_by to find the total global sales per year

sum_sales_platform_global_year = data %>%
  group_by(Year, Platform) %>%
  summarise(Total = sum(Global_Sales))
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

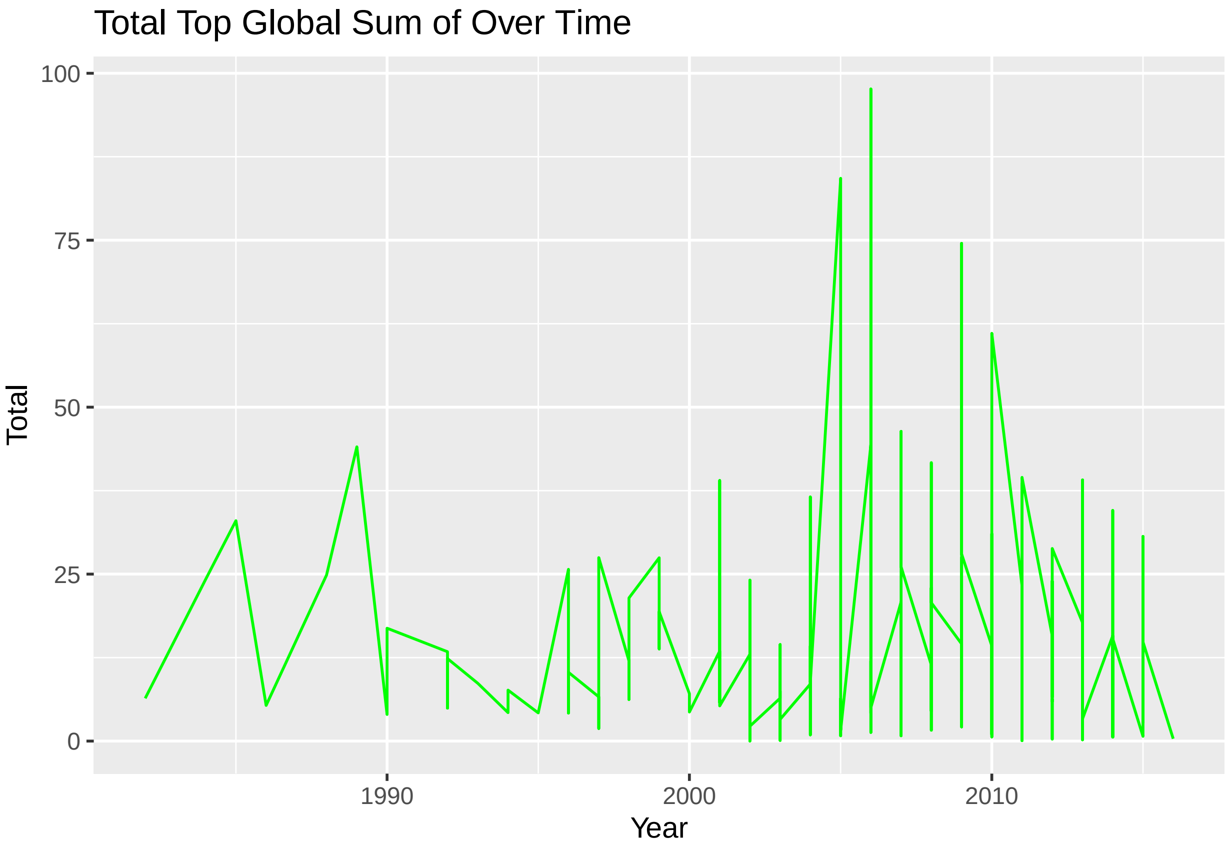
```r
sum_sales_platform_global_year
```

```
## # A tibble: 129 x 3
## # Groups:   Year [33]
##    Year       Platform Total
##    <date>     <chr>    <dbl>
##  1 1982-01-01 2600      6.4
##  2 1984-01-01 NES      24.2
##  3 1985-01-01 NES      33
##  4 1986-01-01 NES       5.34
##  5 1988-01-01 NES      24.9
##  6 1989-01-01 GB       44.1
##  7 1990-01-01 NES       3.98
##  8 1990-01-01 SNES     16.9
##  9 1992-01-01 GB       13.4
## 10 1992-01-01 GEN       4.94
```

```
## # ... with 119 more rows
```

```
sum_sales_platform_global_year_plot <- ggplot(sum_sales_platform_global_year,
                                              aes(x = Year, y = Total)) +
  geom_line(stat="identity", colour="Green") +
  ggtitle("Total Top Global Sum of Over Time")

sum_sales_platform_global_year_plot
```

```
## Warning: Removed 2 rows containing missing values (`geom_line()`).
```



Interesting to see that we see seasonal spikes here. Turtle Games product sales are more successful during certain times of the year.

Creating a time series to see seasonal spikes in more detail…

```
# Convert the data into a time series.
# Create a new data frame and assign time series value,
# and specify the 'ts' function.

sum_sales_platform_global_year
```

```
## # A tibble: 129 x 3
## # Groups:   Year [33]
##    Year       Platform Total
##    <date>     <chr>    <dbl>
##  1 1982-01-01 2600      6.4
##  2 1984-01-01 NES      24.2
##  3 1985-01-01 NES      33
##  4 1986-01-01 NES       5.34
```

```
##  5 1988-01-01 NES      24.9
##  6 1989-01-01 GB       44.1
##  7 1990-01-01 NES       3.98
##  8 1990-01-01 SNES     16.9
##  9 1992-01-01 GB       13.4
## 10 1992-01-01 GEN       4.94
## # ... with 119 more rows
```

```
global_sales_time = sum_sales_platform_global_year[c("Year", "Total")]

global_sales_time
```

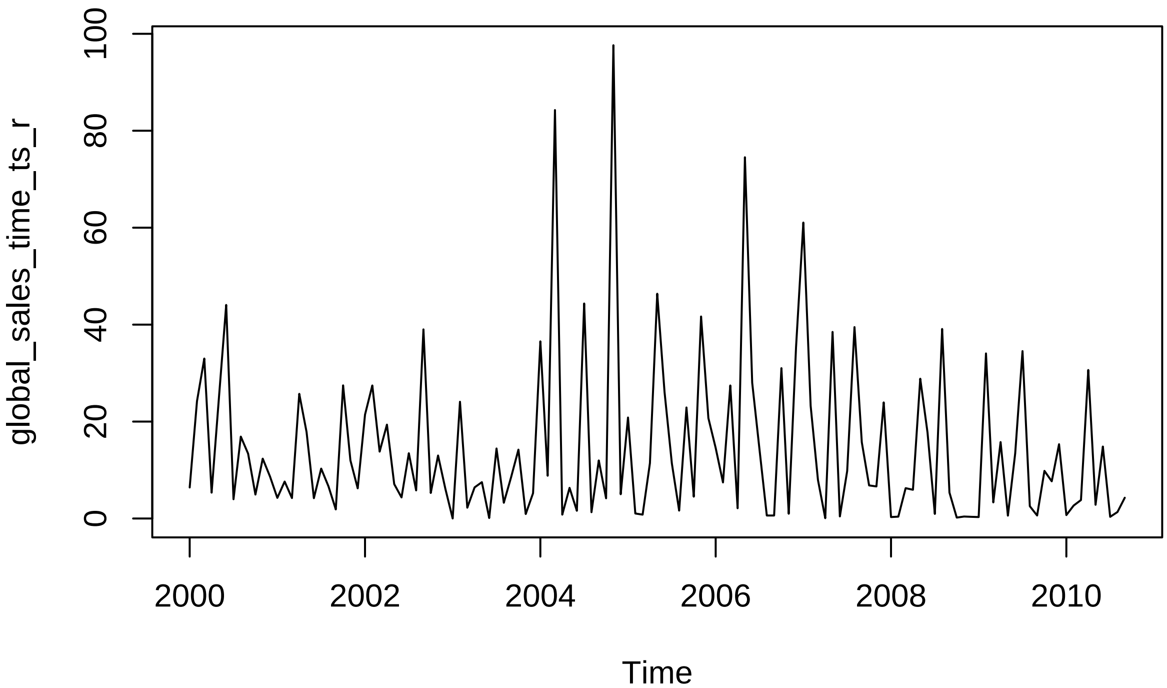
```
## # A tibble: 129 x 2
## # Groups:   Year [33]
##    Year       Total
##    <date>     <dbl>
##  1 1982-01-01  6.4
##  2 1984-01-01 24.2
##  3 1985-01-01 33
##  4 1986-01-01  5.34
##  5 1988-01-01 24.9
##  6 1989-01-01 44.1
##  7 1990-01-01  3.98
##  8 1990-01-01 16.9
##  9 1992-01-01 13.4
## 10 1992-01-01  4.94
## # ... with 119 more rows
```

```
# Change the names of columns by specifying the new column names.
colnames(global_sales_time) <- c('date', 'index')

global_sales_time
```

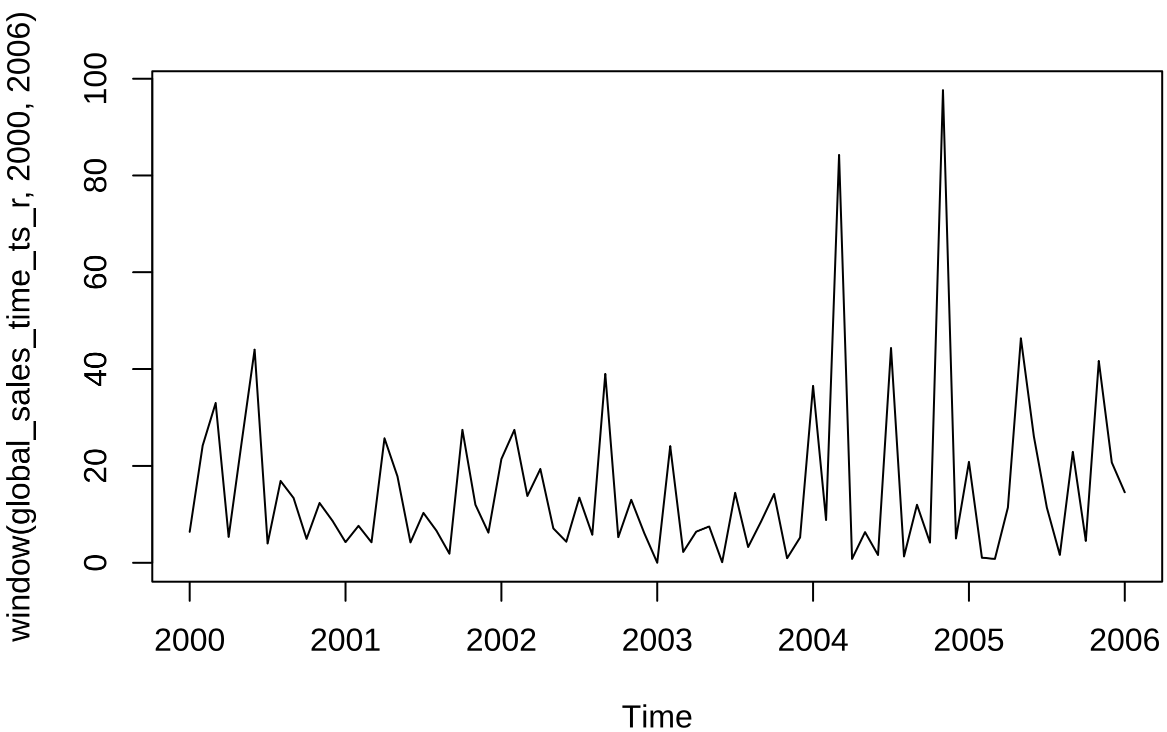
```
## # A tibble: 129 x 2
## # Groups:   date [33]
##    date       index
##    <date>     <dbl>
##  1 1982-01-01  6.4
##  2 1984-01-01 24.2
##  3 1985-01-01 33
##  4 1986-01-01  5.34
##  5 1988-01-01 24.9
##  6 1989-01-01 44.1
##  7 1990-01-01  3.98
##  8 1990-01-01 16.9
##  9 1992-01-01 13.4
## 10 1992-01-01  4.94
## # ... with 119 more rows
```

```
global_sales_time_ts_r <- ts(global_sales_time$index,
                             start = c(2000, 1),
                             # Monthly frequency without missing values in data.
                             frequency = 12)

# Sense-check the new object.
# View the data by creating a smaller sample of the visualisation.
```

```
plot(global_sales_time_ts_r)
```




```
# View the data by creating a smaller sample of the visualisation.
plot(window(global_sales_time_ts_r, 2000, 2006))
```



We see here that sales spike around the end and the beginning of the year. Turtle games should push sales and marketing campaigns during these times. Perhaps it could be the case that new games are released during this time of year?

https://www.ps4playstation4.com/ps4-release-date-countdown-begins

This article contains release dates of PS1, PS2, PS3 and PS4 platforms. Interestingly November is a popular release date. This is in time for Christmas. This indicates that gaming companies like to have big releases before Christmas, the gift giving season, to improve sales. Turtle games should focus on improving sales around this time.

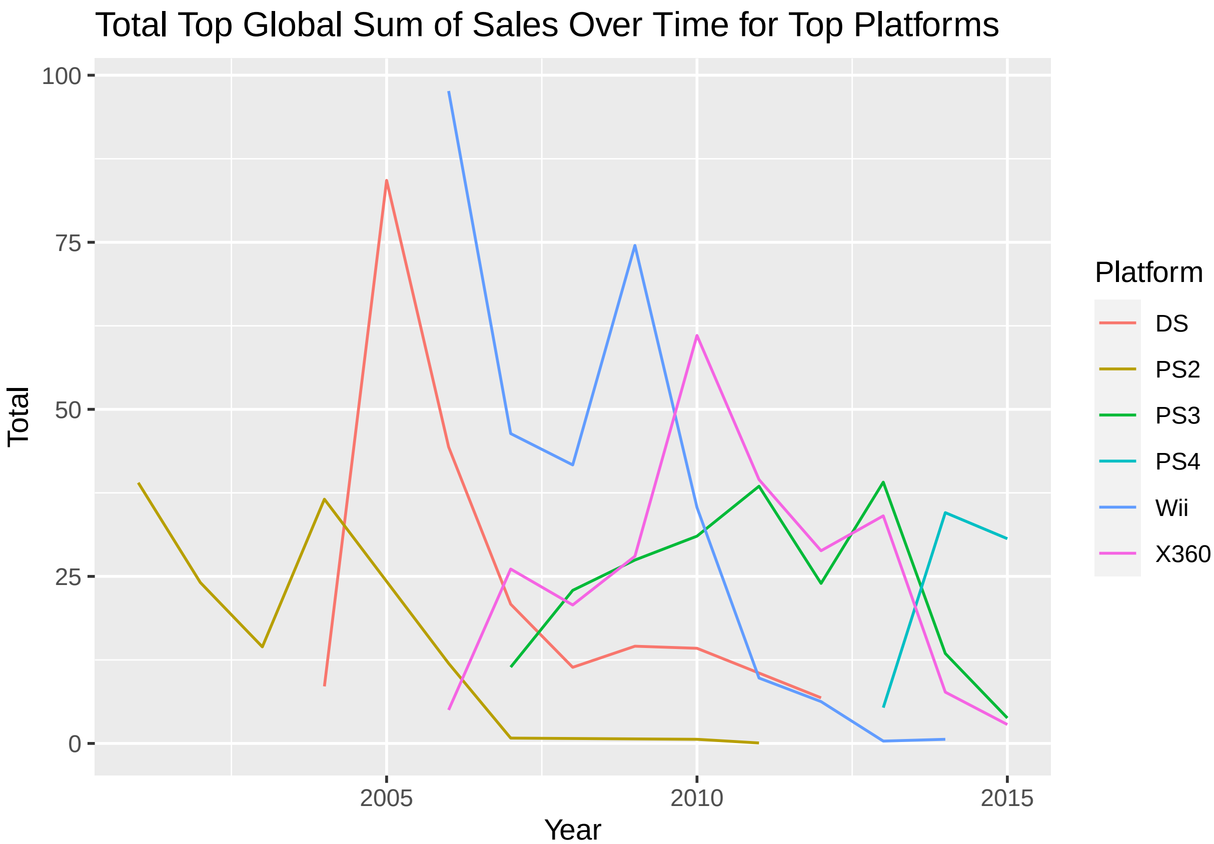
Having a look at seasonal trends in more detail to see how the sales of certain platforms changes over time.

```
# Show top platforms to see sales changing over time

top_sales_over_time <- sum_sales_platform_global_year [sum_sales_platform_global_year$Platform %in% c("

top_sales_over_time
```

```
## # A tibble: 48 x 3
## # Groups:   Year [16]
##    Year       Platform Total
##    <date>     <chr>    <dbl>
##  1 2001-01-01 PS2      39.0
##  2 2002-01-01 PS2      24.1
##  3 2003-01-01 PS2      14.4
##  4 2004-01-01 DS        8.54
##  5 2004-01-01 PS2      36.6
##  6 2005-01-01 DS       84.3
##  7 2006-01-01 DS       44.4
##  8 2006-01-01 PS2      12.0
##  9 2006-01-01 Wii      97.6
## 10 2006-01-01 X360      5.01
## # ... with 38 more rows
```

```
top_sales_over_time_plot <- ggplot(top_sales_over_time,
                                   aes(x = Year, y = Total, colour = Platform)) +
  geom_line() +
  ggtitle("Total Top Global Sum of Sales Over Time for Top Platforms")

top_sales_over_time_plot
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```



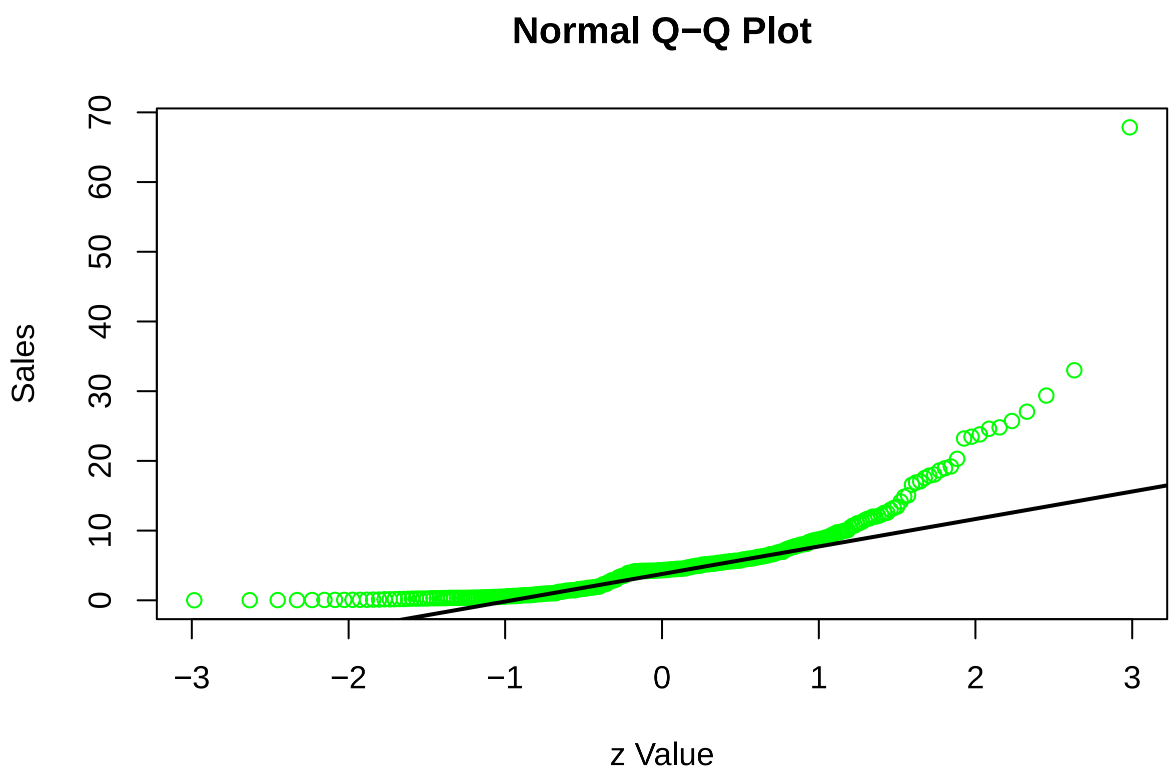
It is interesting to see how the sale of certain platforms over time. We see when games were in higher demand they have the highest sales.

As seen by this article Wii has now been discontinued due to other platforms being released. This shows how Turtle Games needs to understand and create a marketing strategy around new game releases to maximize profit.

https://www.lifewire.com/slow-painful-death-of-the-nintendo-wii-2498653#:~:text=Some%20consoles%2C%20like%20the%20

I am now going to check the reliability of the sales data.

```
# Now I'm going to check the normality of the data
# It is possible to do this using a Q-Q plot

# Specify the qqnorm function.
# Draw a qqplot using the Global_Sales.
qqnorm(data$Global_Sales,
       col='green',
       xlab="z Value",
       ylab='Sales')

# Specify the qqline function.
# Add a reference line to the qqplot.
qqline(data$Global_Sales,
       col='black',
       lwd=2)
```



As we see data points deviate by large amount from the line. Points should be lower according distribution. Values in tails of the distribution are not as extreme as we would expect. Therefore the qqplot shows that the sales data has lighter tales.

```
# Run a Shapiro-Wilk test:
shapiro.test(data$Global_Sales)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$Global_Sales
## W = 0.6818, p-value < 2.2e-16
```

Null hypothesis of the Shapiro-Wilk test is that the data is normally distributed. If there is a small P value we reject the null hypothesis. There is a very small p value here. Here we see the data is not normally distributed

```
# Specify the skewness and kurtosis functions.
skewness(data$Global_Sales)
```

```
## [1] 4.045582
```

Skewness of 4 large amount of positive skewness. If the skewness is between -0.5 and 0.5, the data are fairly symmetrical. If the skewness is between -1 and – 0.5 or between 0.5 and 1, the data are moderately skewed. If the skewness is less than -1 or greater than 1, the data are highly skewed.

Positive Skewness means when the tail on the right side of the distribution is longer or fatter. Positive skewness could be caused by inequality of distribution This means the more sales are distribution towards the lower end of the distribution.

```
kurtosis(data$Global_Sales)
```

```
## [1] 32.63966
```

This measures whether tails are heavy or light tailed. The data has a kurtosis of 32. This means the data has heavier tails than normal. Some values much higher than predicted by the distribution or their are some outliers.

I am now going to see to explore the relationship between the sales data using single and multiple linear regression.

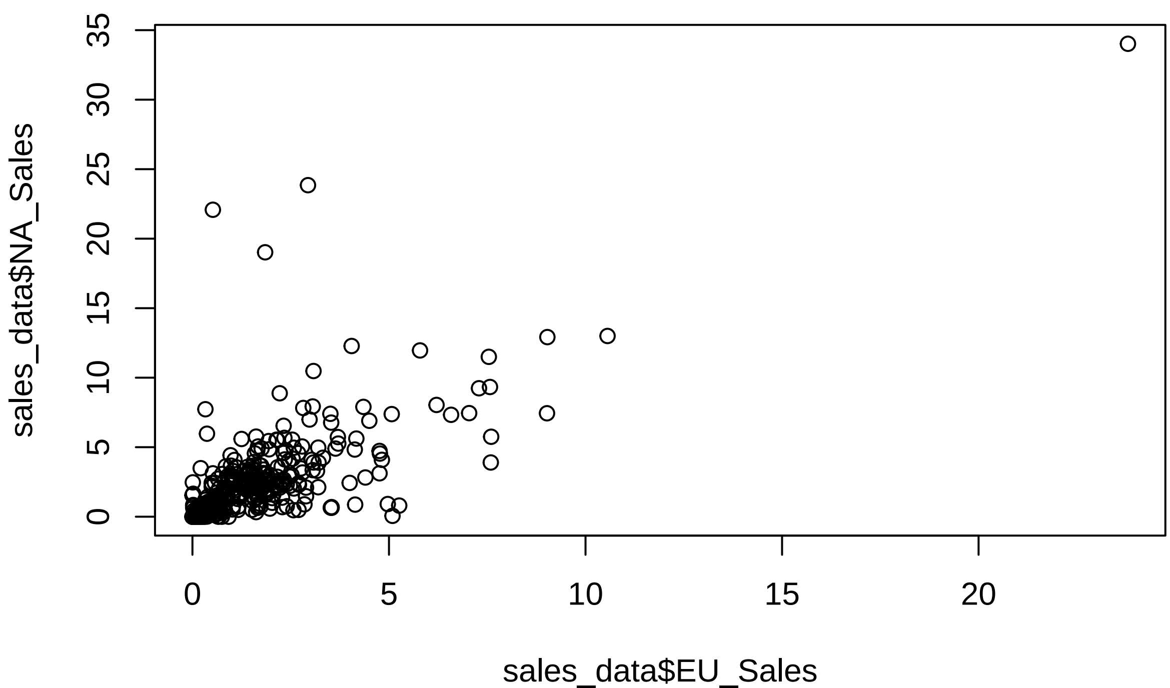
```
# Is there correlation between sales columns
sales_data = data%>%
  select(EU_Sales, NA_Sales, Global_Sales)

cor(sales_data)
```

```
##              EU_Sales  NA_Sales Global_Sales
## EU_Sales     1.0000000 0.7055236    0.8775575
## NA_Sales     0.7055236 1.0000000    0.9349455
## Global_Sales 0.8775575 0.9349455    1.0000000
```

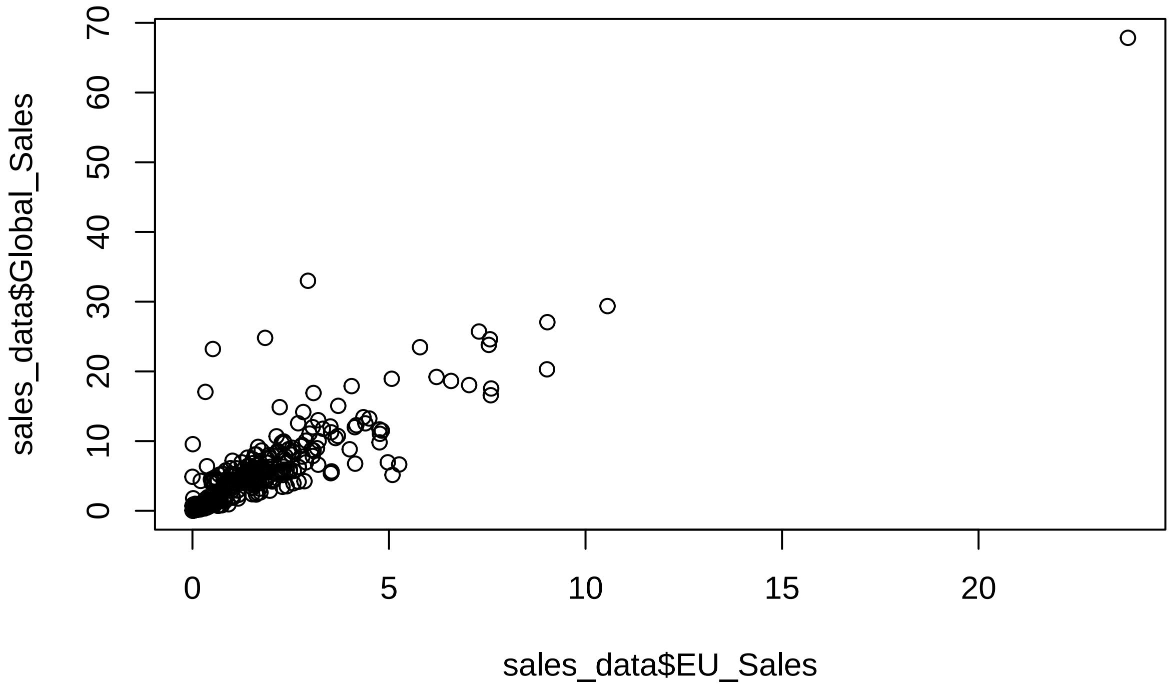
We see here that all the sales data is strongly correlated. The strongest correlation exits between global sales data and EU or North America data.

Here we will try to plot the relationship between the data

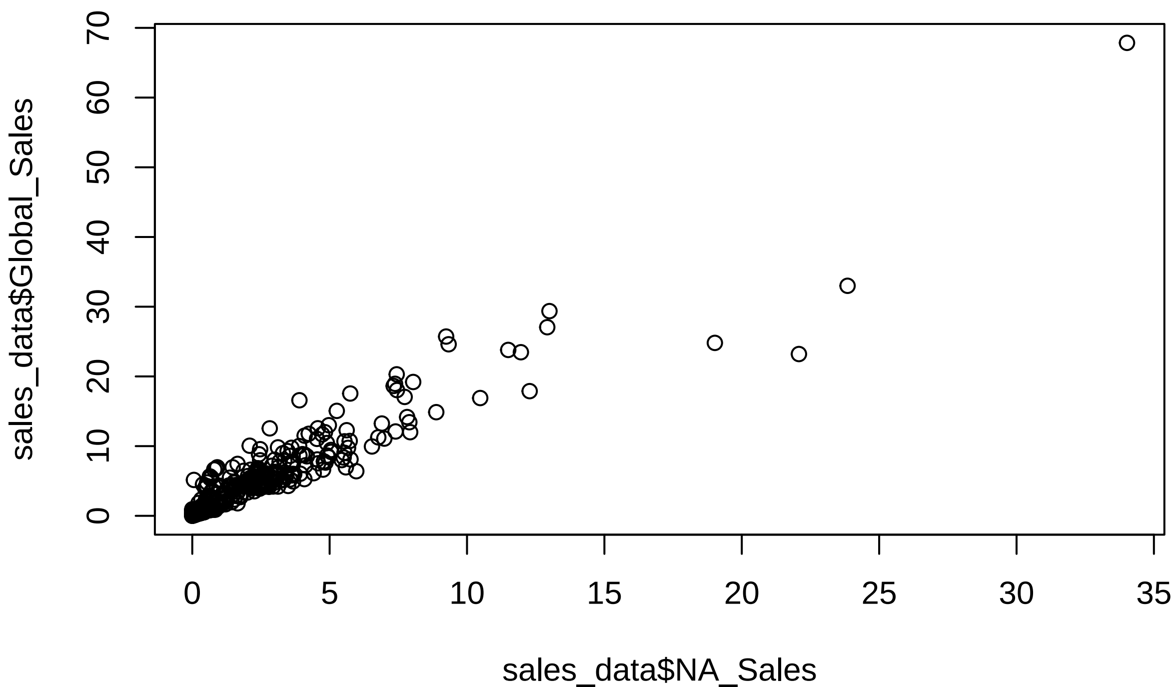
```
# Hard to see any relationship here
plot(sales_data$EU_Sales, sales_data$NA_Sales)
```




```
# A bit more of a relationships here
plot(sales_data$EU_Sales, sales_data$Global_Sales)
```




```r
# More of a relationship here
plot(sales_data$NA_Sales, sales_data$Global_Sales)
```



Makes sense that there is more of a relationship between Global Sales and EU or NA sales. Global Sales is calculated from EU and NA sales

```r
# Creating simple linear regression NA and EU Sales

NA_EU_Sales <- lm(EU_Sales~NA_Sales,
                  data=sales_data)
```

```r
# lm is linear model
# Just one x variable

# View the model.
NA_EU_Sales
```

```
##
## Call:
## lm(formula = EU_Sales ~ NA_Sales, data = sales_data)
##
## Coefficients:
## (Intercept)     NA_Sales
##      0.5891       0.4192
```

```r
summary(NA_EU_Sales)
```

```
##
## Call:
## lm(formula = EU_Sales ~ NA_Sales, data = sales_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3248 -0.5791 -0.2776  0.3439  8.9501
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.58911    0.09528   6.183 1.75e-09 ***
## NA_Sales     0.41919    0.02251  18.625  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.438 on 350 degrees of freedom
## Multiple R-squared:  0.4978, Adjusted R-squared:  0.4963
## F-statistic: 346.9 on 1 and 350 DF,  p-value: < 2.2e-16
```

To some extent these numbers are correlated. NA Sales accounts for 50% of the variance of EU Sales. The p is very small meaning we can reject the null hypothesis that there is no correlation

```r
# Moving on to the relationship between
# Global and NA Sales

EU_Global_Sales <- lm(EU_Sales~Global_Sales,
                      data=sales_data)

# lm is linear model
# Just one x variable

# View the model.
EU_Global_Sales
```

```
##
## Call:
## lm(formula = EU_Sales ~ Global_Sales, data = sales_data)
##
## Coefficients:
##  (Intercept)  Global_Sales
```

```
##       0.1300        0.2838
```

```r
summary(EU_Global_Sales)
```

```
##
## Call:
## lm(formula = EU_Sales ~ Global_Sales, data = sales_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5539 -0.2717 -0.0537  0.2927  4.4172
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.130034   0.068134   1.909   0.0571 .
## Global_Sales 0.283755   0.008287  34.241   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9727 on 350 degrees of freedom
## Multiple R-squared:  0.7701, Adjusted R-squared:  0.7695
## F-statistic:  1172 on 1 and 350 DF,  p-value: < 2.2e-16
```

P value is small rejecting null hypothesis of no correlation. R value is large, this shows that 77% of the variance of global sales is explained by EU Sales.

```r
# Moving on to the relationship between
# Global and EU Sales

NA_Global_Sales <- lm(NA_Sales~Global_Sales,
                      data=sales_data)

# lm is linear model
# Just one x variable

# View the model.
NA_Global_Sales
```

```
##
## Call:
## lm(formula = NA_Sales ~ Global_Sales, data = sales_data)
##
## Coefficients:
##  (Intercept)  Global_Sales
##      -0.1984        0.5088
```

```r
summary(NA_Global_Sales)
```

```
##
## Call:
## lm(formula = NA_Sales ~ Global_Sales, data = sales_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3377 -0.3786  0.0838  0.3743 10.4689
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.19838    0.08485  -2.338     0.02 *
## Global_Sales  0.50881    0.01032  49.300   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.211 on 350 degrees of freedom
## Multiple R-squared:  0.8741, Adjusted R-squared:  0.8738
## F-statistic:  2430 on 1 and 350 DF,  p-value: < 2.2e-16
```

Again we see here that the P value is very small. We can reject null hypothesis of no correlation. 87% of the variance of global sales can be explained by NA sales

```r
# Multiple linear regression

# Create a new object and
# specify the lm function and the variables.
multi_regression = lm(Global_Sales~EU_Sales+NA_Sales, data=sales_data)

# Print the summary statistics.
summary(multi_regression)
```

```
##
## Call:
## lm(formula = Global_Sales ~ EU_Sales + NA_Sales, data = sales_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6186 -0.4234 -0.2692  0.0796  7.4639
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.22175    0.07760   2.858  0.00453 **
## EU_Sales     1.34197    0.04134  32.466  < 2e-16 ***
## NA_Sales     1.15543    0.02456  47.047  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.112 on 349 degrees of freedom
## Multiple R-squared:  0.9687, Adjusted R-squared:  0.9685
## F-statistic:  5398 on 2 and 349 DF,  p-value: < 2.2e-16
```

Combined NA and EU Sales account for 97% of the variation in global sales. Confirmed by the low result of the P value we can reject null hypothesis of no correlation.

Turtle Games should focus on increasing and sustaining sales in the EU and NA as these regions account for a large proportion of sales. Turtle Games could also investigate and find what causes the remaining 3% of variance. This 3% could account for new emerging markets.

Now I'm going to test the accuracy of the model.

```r
# Predicting future global sales
# Testing the model
head(sales_data)
```

```
##   EU_Sales NA_Sales Global_Sales
## 1    23.80    34.02        67.85
## 2     2.94    23.85        33.00
```

```
## 3    10.56    13.00        29.37
## 4     9.03    12.92        27.06
## 5     7.29     9.24        25.72
## 6     1.85    19.02        24.81
```

```r
# Create a new object and specify the predict function.
predictTest = predict(multi_regression, newdata=sales_data,
                      interval='confidence')
# Print the object.
head(predictTest)
```

```
##        fit      lwr      upr
## 1 71.46857 70.16242 72.77472
## 2 31.72426 30.75814 32.69038
## 3 29.41363 28.88685 29.94040
## 4 27.26797 26.81975 27.71619
## 5 20.68094 20.33539 21.02649
## 6 24.68076 23.88671 25.47482
```

Here we can make comparison of the accuracy of the model. If NA_Sales_sum is 34.02 and EU_Sales_sum is 23.80 the model predicts global sales will be 72.77. This is not too far off the value of 67.84 the actual value You can also see the lower and upper values defining the confidence interval in our predicted values. The confidence interval in the predict function will help us to gauge the uncertainty in the predictions.