Assignment 1 - Group 1

Introduction

This dataset contains a list of video games with sales greater than 100,000 copies from Kaggle. The sales numbers in the dataset are in millions. There are 16,598 records in the dataset and the data was last updated in 2016. This dataset will be used to perform data analysis for the purpose of Assignment 1.

Load the necessary packages

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.2     v purrr   0.3.4
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library (readr)
```

Load dataset

Because the file location will be different for everyone, we load the dataset directly from Github raw file

```
urlfile="https://raw.githubusercontent.com/trngminhtrang/DataAnalysis--Video-Games-Sales--Historical-Da

vgsales <-read_csv(url(urlfile))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Rank = col_double(),
##   Name = col_character(),
##   Platform = col_character(),
##   Year = col_character(),
##   Genre = col_character(),
##   Publisher = col_character(),
##   NA_Sales = col_double(),
##   EU_Sales = col_double(),
##   JP_Sales = col_double(),
##   Other_Sales = col_double(),
##   Global_Sales = col_double()
## )
```

Print the structure of the dataset

```
str(vgsales)
```

```
## spec_tbl_df [16,598 x 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Rank        : num [1:16598] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Name        : chr [1:16598] "Wii Sports" "Super Mario Bros." "Mario Kart Wii" "Wii Sports Resort"
##  $ Platform    : chr [1:16598] "Wii" "NES" "Wii" "Wii" ...
##  $ Year        : chr [1:16598] "2006" "1985" "2008" "2009" ...
##  $ Genre       : chr [1:16598] "Sports" "Platform" "Racing" "Sports" ...
##  $ Publisher   : chr [1:16598] "Nintendo" "Nintendo" "Nintendo" "Nintendo" ...
##  $ NA_Sales    : num [1:16598] 41.5 29.1 15.8 15.8 11.3 ...
##  $ EU_Sales    : num [1:16598] 29.02 3.58 12.88 11.01 8.89 ...
##  $ JP_Sales    : num [1:16598] 3.77 6.81 3.79 3.28 10.22 ...
##  $ Other_Sales : num [1:16598] 8.46 0.77 3.31 2.96 1 0.58 2.9 2.85 2.26 0.47 ...
##  $ Global_Sales: num [1:16598] 82.7 40.2 35.8 33 31.4 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Rank = col_double(),
##   ..   Name = col_character(),
##   ..   Platform = col_character(),
##   ..   Year = col_character(),
##   ..   Genre = col_character(),
##   ..   Publisher = col_character(),
##   ..   NA_Sales = col_double(),
##   ..   EU_Sales = col_double(),
##   ..   JP_Sales = col_double(),
##   ..   Other_Sales = col_double(),
##   ..   Global_Sales = col_double()
##   .. )
```

List the variables in the dataset

```
ls(vgsales)
```

```
##  [1] "EU_Sales"     "Genre"        "Global_Sales" "JP_Sales"     "NA_Sales"
##  [6] "Name"         "Other_Sales"  "Platform"     "Publisher"    "Rank"
## [11] "Year"
```

Print the top 15 rows of the dataset

```
head(vgsales,15)
```

```
## # A tibble: 15 x 11
##     Rank Name         Platform Year  Genre  Publisher NA_Sales EU_Sales JP_Sales
##    <dbl> <chr>        <chr>    <chr> <chr>  <chr>        <dbl>    <dbl>    <dbl>
##  1     1 Wii Sports   Wii      2006  Sports Nintendo     41.5     29.0      3.77
##  2     2 Super Mario~ NES      1985  Platf~ Nintendo     29.1      3.58     6.81
##  3     3 Mario Kart ~ Wii      2008  Racing Nintendo     15.8     12.9      3.79
##  4     4 Wii Sports ~ Wii      2009  Sports Nintendo     15.8     11.0      3.28
##  5     5 Pokemon Red~ GB       1996  Role-~ Nintendo     11.3      8.89    10.2
##  6     6 Tetris       GB       1989  Puzzle Nintendo     23.2      2.26     4.22
##  7     7 New Super M~ DS       2006  Platf~ Nintendo     11.4      9.23     6.5
##  8     8 Wii Play     Wii      2006  Misc   Nintendo     14.0      9.2      2.93
##  9     9 New Super M~ Wii      2009  Platf~ Nintendo     14.6      7.06     4.7
## 10    10 Duck Hunt    NES      1984  Shoot~ Nintendo     26.9      0.63     0.28
## 11    11 Nintendogs   DS       2005  Simul~ Nintendo      9.07    11        1.93
## 12    12 Mario Kart ~ DS       2005  Racing Nintendo      9.81     7.57     4.13
## 13    13 Pokemon Gol~ GB       1999  Role-~ Nintendo      9        6.18     7.2
## 14    14 Wii Fit      Wii      2007  Sports Nintendo      8.94     8.03     3.6
## 15    15 Wii Fit Plus Wii      2009  Sports Nintendo      9.09     8.59     2.53
## # ... with 2 more variables: Other_Sales <dbl>, Global_Sales <dbl>
```

Write a user defined function

User defined function "model" calculates the sum of two variables namely NA_Sales and EU_Sales from the data set and stores it into new variable called "sum" in vgsales dataset

```
model<-function(x,y){x+y}
vgsales$sum = model(vgsales$NA_Sales, vgsales$EU_Sales)
head(vgsales$sum)
```

```
## [1] 70.51 32.66 28.73 26.76 20.16 25.46
```

Using filter command to filter out sales where Global_Sales are > 10

```
vgsalesnew2 = as.data.frame(vgsales %>% filter(Global_Sales > 10))
summary(vgsalesnew2)
```

```
##       Rank           Name             Platform              Year          
##  Min.   : 1.00   Length:62          Length:62          Length:62         
##  1st Qu.:16.25   Class :character   Class :character   Class :character  
##  Median :31.50   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :31.50                                                           
##  3rd Qu.:46.75                                                           
##  Max.   :62.00                                                           
##     Genre            Publisher             NA_Sales         EU_Sales     
##  Length:62          Length:62          Min.   : 2.550   Min.   : 0.010  
##  Class :character   Class :character   1st Qu.: 5.103   1st Qu.: 3.120  
##  Mode  :character   Mode  :character   Median : 6.805   Median : 3.980  
##                                        Mean   : 8.939   Mean   : 5.201  
##                                         3rd Qu.: 9.607   3rd Qu.: 5.817
##                                         Max.   :41.490   Max.   :29.020
##     JP_Sales        Other_Sales      Global_Sales        sum
##  Min.   : 0.0000   Min.   : 0.230   Min.   :10.21   Min.   : 3.020
##  1st Qu.: 0.4175   1st Qu.: 0.770   1st Qu.:11.89   1st Qu.: 8.938
##  Median : 2.3300   Median : 1.170   Median :14.64   Median :11.150
##  Mean   : 2.5198   Mean   : 1.713   Mean   :18.37   Mean   :14.140
##  3rd Qu.: 3.9300   3rd Qu.: 1.995   3rd Qu.:21.71   3rd Qu.:16.468
##  Max.   :10.2200   Max.   :10.570   Max.   :82.74   Max.   :70.510
```

Identify the dependent & independent variables and create a new data frame by joining these variables

As Global_Sales the total sales worldwide, which is the number of all regions sales combined, we can identify it as dependent variable. We select NA_Sales as the independent variable for this task. In this case, we extract the 1st & 6th coloumns and create a new data frame called "vgsalesnew1"

```
vgsalesnew1 = as.data.frame(vgsales %>% select(7,11))
summary(vgsalesnew1)
```

```
##     NA_Sales        Global_Sales
##  Min.   : 0.0000   Min.   : 0.0100
##  1st Qu.: 0.0000   1st Qu.: 0.0600
##  Median : 0.0800   Median : 0.1700
##  Mean   : 0.2647   Mean   : 0.5374
##  3rd Qu.: 0.2400   3rd Qu.: 0.4700
##  Max.   :41.4900   Max.   :82.7400
```

Remove missing values

Replace N/A values in the dataset with NA

We notice that the missing values in the dataset were recorded as N/A, which appears to not be treated as NA. We will replace the N/A values with NA so R can recognize the missing values

```
vgsales[vgsales=="N/A"]=NA
```

Identify the number of missing values

```
sum(is.na(vgsales))
```

```
## [1] 329
```

Remove missing values from the dataset

```
vgsales1 <- vgsales[complete.cases(vgsales), ]
str(vgsales1)
```

```
## tibble [16,291 x 12] (S3: tbl_df/tbl/data.frame)
##  $ Rank        : num [1:16291] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Name        : chr [1:16291] "Wii Sports" "Super Mario Bros." "Mario Kart Wii" "Wii Sports Resort"
##  $ Platform    : chr [1:16291] "Wii" "NES" "Wii" "Wii" ...
##  $ Year        : chr [1:16291] "2006" "1985" "2008" "2009" ...
##  $ Genre       : chr [1:16291] "Sports" "Platform" "Racing" "Sports" ...
##  $ Publisher   : chr [1:16291] "Nintendo" "Nintendo" "Nintendo" "Nintendo" ...
##  $ NA_Sales    : num [1:16291] 41.5 29.1 15.8 15.8 11.3 ...
##  $ EU_Sales    : num [1:16291] 29.02 3.58 12.88 11.01 8.89 ...
##  $ JP_Sales    : num [1:16291] 3.77 6.81 3.79 3.28 10.22 ...
##  $ Other_Sales : num [1:16291] 8.46 0.77 3.31 2.96 1 0.58 2.9 2.85 2.26 0.47 ...
##  $ Global_Sales: num [1:16291] 82.7 40.2 35.8 33 31.4 ...
##  $ sum         : num [1:16291] 70.5 32.7 28.7 26.8 20.2 ...
```

There are 16,291 records in the new data frame while the original dataframe has 16,598 records. This means 329 missing values have been removed from the new dataframe.

Identify and remove duplicated data

Identify duplicated data throughout the dataset

```
duplicate <- duplicated(vgsales)
sum(duplicate)
```

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

There is no duplicated rows in the dataset. We will remove duplicated data in the EU_Sales variable.

Identify and remove duplicated data in EU_Sales variable

Identify the number of duplicated in EU_Sales

```
sum(duplicated(vgsales$EU_Sales))
```

```
## [1] 16293
```

Remove duplicated data

```
noduplicate_EU <- vgsales %>% distinct(Distinct_EU = EU_Sales)
noduplicate_EU
```

```
## # A tibble: 305 x 1
##    Distinct_EU
##          <dbl>
##  1       29.0
##  2        3.58
##  3       12.9
##  4       11.0
##  5        8.89
##  6        2.26
##  7        9.23
##  8        9.2
##  9        7.06
## 10        0.63
## # ... with 295 more rows
```

There are 305 distinct records from the new dataset noduplicate_EU. This means 16,293 duplicated data has been removed from EU_Sales.

Reorder multiple rows in descending order

We reorder the dataset in descending order of the EU_Sales variable

```
vgsales %>% arrange(desc(EU_Sales))
```

```
## # A tibble: 16,598 x 12
##     Rank Name        Platform Year  Genre  Publisher  NA_Sales EU_Sales JP_Sales
##    <dbl> <chr>       <chr>    <chr> <chr>  <chr>         <dbl>    <dbl>    <dbl>
##  1     1 Wii Sports  Wii      2006  Sports Nintendo      41.5     29.0      3.77
##  2     3 Mario Kart~ Wii      2008  Racing Nintendo      15.8     12.9      3.79
##  3     4 Wii Sports~ Wii      2009  Sports Nintendo      15.8     11.0      3.28
##  4    11 Nintendogs  DS       2005  Simul~ Nintendo       9.07    11        1.93
##  5    17 Grand Thef~ PS3      2013  Action Take-Two ~     7.01     9.27     0.97
##  6    20 Brain Age:~ DS       2005  Misc   Nintendo       4.75     9.26     4.16
##  7     7 New Super ~ DS       2006  Platf~ Nintendo      11.4      9.23     6.5
##  8     8 Wii Play    Wii      2006  Misc   Nintendo      14.0      9.2      2.93
##  9     5 Pokemon Re~ GB       1996  Role-~ Nintendo      11.3      8.89    10.2
## 10    15 Wii Fit Pl~ Wii      2009  Sports Nintendo       9.09     8.59     2.53
## # ... with 16,588 more rows, and 3 more variables: Other_Sales <dbl>,
## #   Global_Sales <dbl>, sum <dbl>
```

Rename some column names in the dataset

Rename the "Rank" column with "Ranking" and "Name" column with "Games".

```
names(vgsales)[1] <- 'Ranking'
names(vgsales)[2] <- 'Games'
head(vgsales)
```

```
## # A tibble: 6 x 12
##   Ranking Games       Platform Year  Genre  Publisher NA_Sales EU_Sales JP_Sales
##     <dbl> <chr>       <chr>    <chr> <chr>  <chr>        <dbl>    <dbl>    <dbl>
## 1       1 Wii Sports  Wii      2006  Sports Nintendo      41.5    29.0      3.77
## 2       2 Super Mari~ NES      1985  Platf~ Nintendo      29.1     3.58     6.81
## 3       3 Mario Kart~ Wii      2008  Racing Nintendo      15.8    12.9      3.79
## 4       4 Wii Sports~ Wii      2009  Sports Nintendo      15.8    11.0      3.28
## 5       5 Pokemon Re~ GB       1996  Role-~ Nintendo      11.3     8.89    10.2
## 6       6 Tetris      GB       1989  Puzzle Nintendo      23.2     2.26     4.22
## # ... with 3 more variables: Other_Sales <dbl>, Global_Sales <dbl>, sum <dbl>
```

Add new variables by using a mathematical function

```
vgsales$New_JP_Sales = vgsales$JP_Sales*2
head(vgsales)
```

```
## # A tibble: 6 x 13
##   Ranking Games       Platform Year  Genre  Publisher NA_Sales EU_Sales JP_Sales
##     <dbl> <chr>       <chr>    <chr> <chr>  <chr>        <dbl>    <dbl>    <dbl>
## 1       1 Wii Sports  Wii      2006  Sports Nintendo      41.5    29.0      3.77
## 2       2 Super Mari~ NES      1985  Platf~ Nintendo      29.1     3.58     6.81
## 3       3 Mario Kart~ Wii      2008  Racing Nintendo      15.8    12.9      3.79
## 4       4 Wii Sports~ Wii      2009  Sports Nintendo      15.8    11.0      3.28
## 5       5 Pokemon Re~ GB       1996  Role-~ Nintendo      11.3     8.89    10.2
## 6       6 Tetris      GB       1989  Puzzle Nintendo      23.2     2.26     4.22
## # ... with 4 more variables: Other_Sales <dbl>, Global_Sales <dbl>, sum <dbl>,
## #   New_JP_Sales <dbl>
```

Create a training set using random number generator engine

```r
set.seed(1)
vgsales%>% sample_n (15, replace = FALSE)
```

```
## # A tibble: 15 x 13
##    Ranking Games      Platform Year  Genre Publisher  NA_Sales EU_Sales JP_Sales
##      <dbl> <chr>      <chr>    <chr> <chr> <chr>         <dbl>    <dbl>    <dbl>
##  1    4776 Asura's W~ PS3      2012  Acti~ Capcom         0.18     0.12     0.06
##  2   13219 World Ser~ GC       2005  Misc  Activisio~     0.04     0.01     0
##  3   10540 Blue Stin~ DC       1999  Adve~ Activision     0        0        0.1
##  4    8463 Thrillvil~ X360     2007  Stra~ LucasArts      0.13     0.02     0
##  5    4051 Doom 3 BF~ X360     2012  Shoo~ Bethesda ~     0.28     0.17     0
##  6   13500 Shin Fort~ PS       1996  Misc  Media Wor~     0        0        0.04
##  7   11572 Need For ~ PC       2008  Raci~ Electroni~     0        0.07     0
##  8   12258 Motion Ex~ X360     2011  Misc  505 Games      0.05     0.01     0
##  9   14264 SBK 2011:~ PC       2011  Raci~ Black Bea~     0        0.03     0
## 10   13904 Dramatic ~ PS2      2002  Spor~ Enix Corp~     0        0        0.04
## 11    9942 Ridge Rac~ X360     2012  Raci~ Namco Ban~     0.05     0.05     0
## 12    8230 Rugrats: ~ GC       2002  Plat~ THQ            0.13     0.03     0
## 13     879 FIFA Socc~ X360     2008  Spor~ Electroni~     0.49     1.26     0.01
## 14    6527 Armorines~ N64      1999  Shoo~ Acclaim E~     0.21     0.05     0
## 15   12205 TRON: Evo~ PC       2010  Acti~ Disney In~     0.06     0        0
## # ... with 4 more variables: Other_Sales <dbl>, Global_Sales <dbl>, sum <dbl>,
## #   New_JP_Sales <dbl>
```

Print the summary statistics of the dataset

```r
summary(vgsales)
```

```
##     Ranking         Games              Platform             Year
##  Min.   :    1   Length:16598       Length:16598       Length:16598
##  1st Qu.: 4151   Class :character   Class :character   Class :character
##  Median : 8300   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 8301
##  3rd Qu.:12450
##  Max.   :16600
##     Genre            Publisher            NA_Sales          EU_Sales
##  Length:16598       Length:16598       Min.   : 0.0000   Min.   : 0.0000
##  Class :character   Class :character   1st Qu.: 0.0000   1st Qu.: 0.0000
##  Mode  :character   Mode  :character   Median : 0.0800   Median : 0.0200
##                                        Mean   : 0.2647   Mean   : 0.1467
##                                        3rd Qu.: 0.2400   3rd Qu.: 0.1100
##                                        Max.   :41.4900   Max.   :29.0200
##     JP_Sales         Other_Sales        Global_Sales         sum
##  Min.   : 0.00000   Min.   : 0.00000   Min.   : 0.0100   Min.   : 0.0000
##  1st Qu.: 0.00000   1st Qu.: 0.00000   1st Qu.: 0.0600   1st Qu.: 0.0200
##  Median : 0.00000   Median : 0.01000   Median : 0.1700   Median : 0.1200
##  Mean   : 0.07778   Mean   : 0.04806   Mean   : 0.5374   Mean   : 0.4113
##  3rd Qu.: 0.04000   3rd Qu.: 0.04000   3rd Qu.: 0.4700   3rd Qu.: 0.3700
##  Max.   :10.22000   Max.   :10.57000   Max.   :82.7400   Max.   :70.5100
##   New_JP_Sales
##  Min.   : 0.0000
##  1st Qu.: 0.0000
##  Median : 0.0000
##  Mean   : 0.1556
##  3rd Qu.: 0.0800
##  Max.   :20.4400
```

Perform statistical functions using EU_Sales varable

Calculate Mean

```
mean(vgsales$EU_Sales)
```

```
## [1] 0.146652
```

Calculate Median

```
median(vgsales$EU_Sales)
```

```
## [1] 0.02
```

Calculate Mode

As R does not have a standard built-in function to calculate mode, we create a user function to calculate mode of EU_Sales in the dataset.

Create the function to calculate Mode

```
getmode <- function(v) {
     uniqv <- unique(v)
     uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

Assign v to the EU_Sales variable of the dataset

```
v <- vgsales$EU_Sales
```

Create a variable to store the Mode result

```
mode <- getmode(v)
```

Print the Mode result

```
print(mode)
```

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

The mode value of EU_Sales is 0 which could mean that a majority of games were not available or were not released in EU.

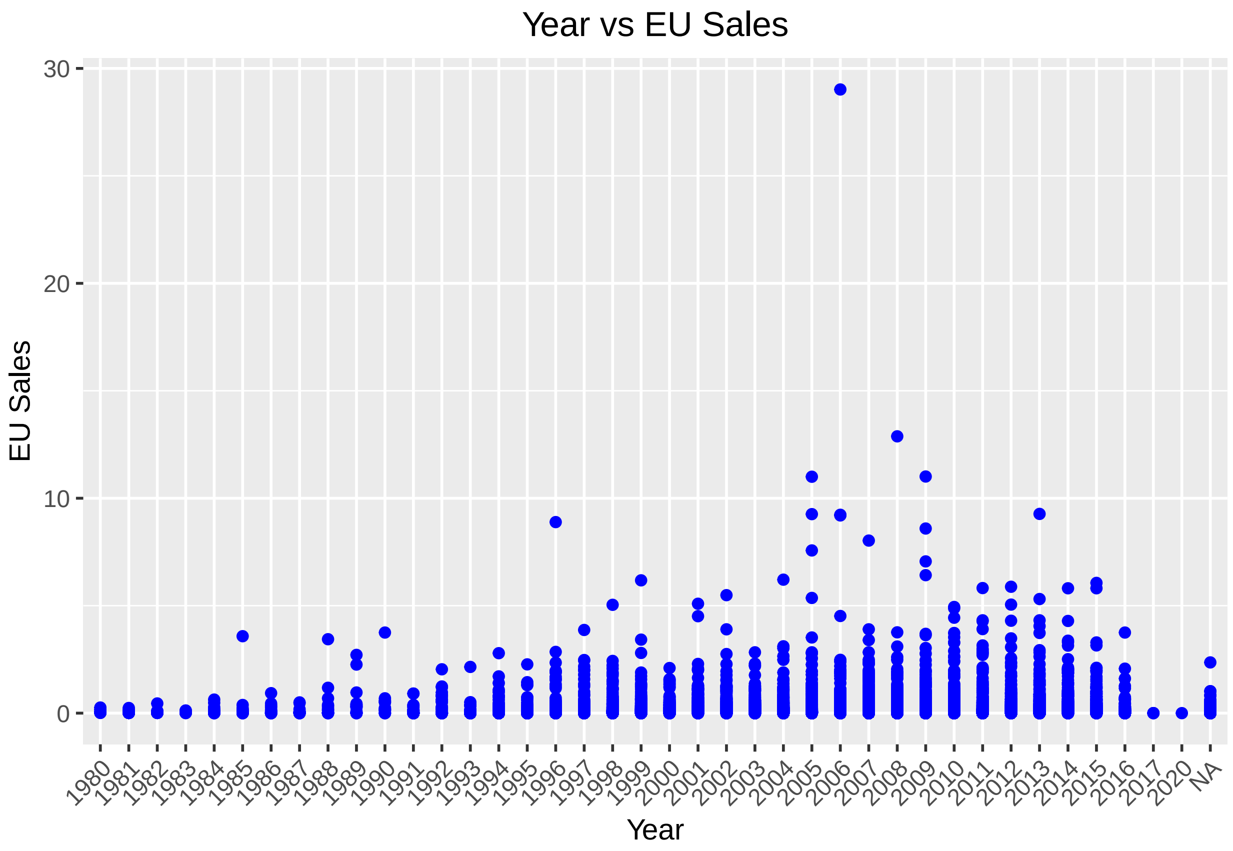
Calculate Range

```
range(vgsales$EU_Sales)
```

```
## [1]  0.00 29.02
```

Plot a scatter plot for Year and EU_Sales variables

```
ggplot(data = vgsales, aes(Year, EU_Sales)) + geom_point(color = "blue") + theme(axis.text.x = element_
```



Plot a bar plot for any two variables in dataset

Filter data for the last 10 years

We filter the dataset by the last 10 years to reduce the data size and make it more relevant for analysis. Since the dataset was last updated in 2016, we filter the time range from 2006 to 2016.

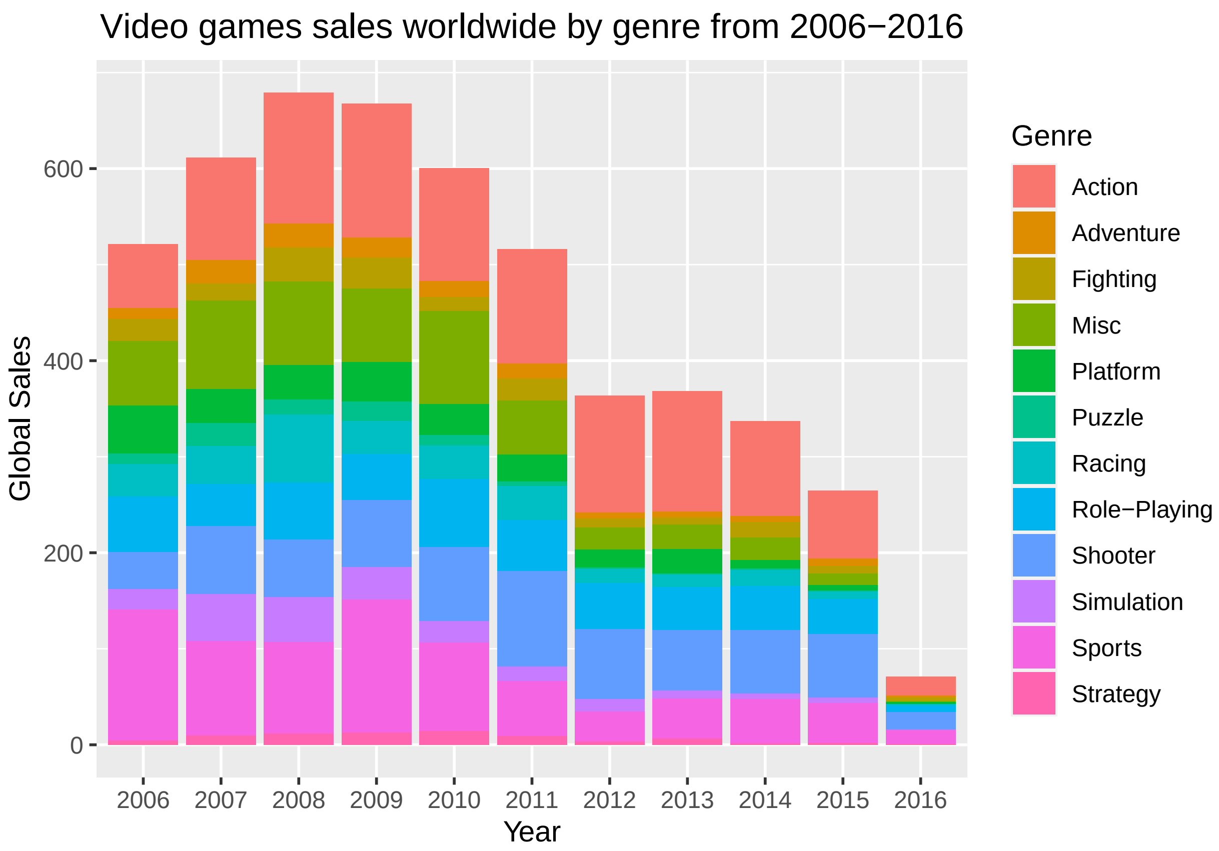
```
filtered_years <- filter(vgsales, Year >= 2006, Year<=2016)
```

The filtered dataset is stored in "filtered_years".

Plot a bar plot for Year and Global Sales variables

This bar plot shows the global sales of videos games by genre over the last 10 years from the last year the dataset was updated (2016).

```
ggplot(data = filtered_years, aes(Year, Global_Sales)) + geom_bar(aes(fill=Genre), stat = "identity") +
```



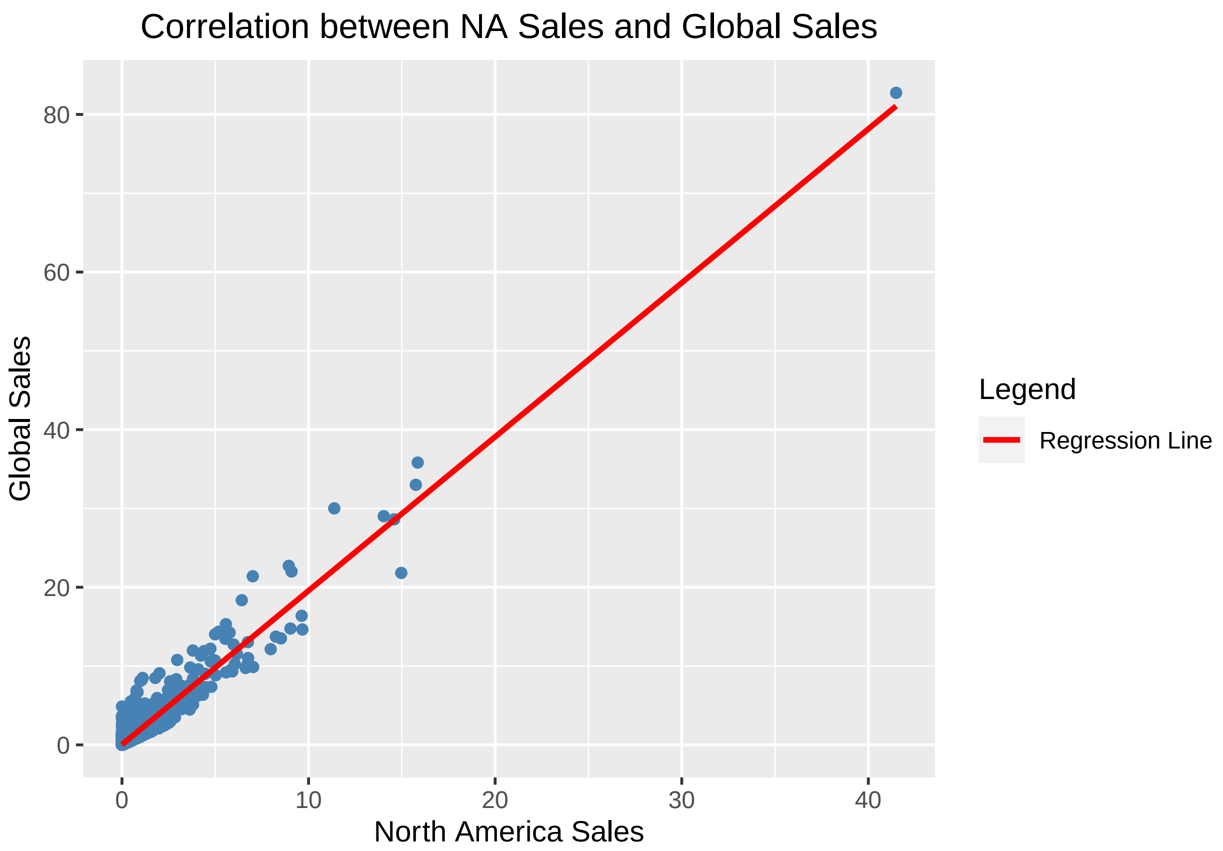
Find the correlation between NA_Sales and Global_Sales

Plot a scatter plot for NA_Sales and Global_Sales with a regression line

This scatter plot aims to find a correlation between the video games sales in the North America and the sales worldwide.

```
ggplot(data = filtered_years, aes(NA_Sales, Global_Sales))+geom_point(color="steelblue")+geom_smooth(aes
```

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



Define X and Y variables for regression model

For the regression model, we are using the full dataset instead of the filtered dataset above to have an overall view

```
Y = vgsales$NA_Sales
X= vgsales$Global_Sales
```

Find the correlation between NA_Sales (Y) and Global_Sales (X)

```
corrl = cor(X, Y,method = "pearson")
corrl
```

```
## [1] 0.9410474
```

Conclusion of Analysis

The global sales of a video game had reached as high as 82.7400 and as low as 0.01. The minimum sales of all regions are 0, and NA has the highest maximum sales of 41.49 while JP has the lowest maximum sales of 10.22.

From the "Video Games Sales Worldwide by Genre from 2006-2016" bar plot, we can see that Sports genre had generated the most sales in 2006 and remained the one of top genres in 2 years before it started to lose traction since 2010. Over the time, Action genre has remained the most popular genre since 2007 while Strategy games made the lowest to zero sales. Further research is needed to determine whether the game publishers dropped Strategy games out of their roadmap or they still rolled out new Strategy games but were unsuccessfully to make revenue from it.

For the “Correlation between NA Sales and Global Sales” scatter plot, it seems to be a correlation between these two variables. Moreover, the correlation coefficient value of 0.9410474 is very close to 1 which means that the NA_Sales and Global_Sales variables have a positive correlation. An increase in NA_Sales will be likely to generate an increase in the Global_Sales to a respective extent, which also means NA_Sales variable is made up a very high percentage of the Global_Sales value.