VGChartz Sales Prediction

Leonard MOK

12/21/2020

Data Science: Capstone IDV Learners

This is an R Markdown document for the edX course Data Science: Capstone IDV Learners project.
The dataset chosen for this machine learning project is "Video Game Sales with Ratings" (reference: https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings).
This dataset contains a list of video games with sales greater than 100,000 copies. It was generated by a scrape of vgchartz.com and another web scrape from Metacritic for ratings.

Dataset preparation

```r
# Video Game Sales with Ratings
# https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings
ratings <- read.csv("https://raw.githubusercontent.com/longyatmok/video_game_sales/main/Video_Games_Sal
summary(ratings)
```

```
     Name             Platform         Year_of_Release       Genre
 Length:16719       Length:16719       Length:16719       Length:16719
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character




  Publisher            NA_Sales          EU_Sales          JP_Sales
 Length:16719       Min.   : 0.0000   Min.   : 0.000   Min.   : 0.0000
 Class :character   1st Qu.: 0.0000   1st Qu.: 0.000   1st Qu.: 0.0000
 Mode  :character   Median : 0.0800   Median : 0.020   Median : 0.0000
                    Mean   : 0.2633   Mean   : 0.145   Mean   : 0.0776
                    3rd Qu.: 0.2400   3rd Qu.: 0.110   3rd Qu.: 0.0400
                    Max.   :41.3600   Max.   :28.960   Max.   :10.2200

  Other_Sales        Global_Sales      Critic_Score     Critic_Count
 Min.   : 0.00000   Min.   : 0.0100   Min.   :13.00   Min.   :  3.00
 1st Qu.: 0.00000   1st Qu.: 0.0600   1st Qu.:60.00   1st Qu.: 12.00
 Median : 0.01000   Median : 0.1700   Median :71.00   Median : 21.00
 Mean   : 0.04733   Mean   : 0.5335   Mean   :68.97   Mean   : 26.36
 3rd Qu.: 0.03000   3rd Qu.: 0.4700   3rd Qu.:79.00   3rd Qu.: 36.00
 Max.   :10.57000   Max.   :82.5300   Max.   :98.00   Max.   :113.00
```

```
                                            NA's   :8582    NA's   :8582
  User_Score          User_Count        Developer           Rating
 Length:16719       Min.   :    4.0   Length:16719       Length:16719
 Class :character   1st Qu.:   10.0   Class :character   Class :character
 Mode  :character   Median :   24.0   Mode  :character   Mode  :character
                    Mean   :  162.2
                    3rd Qu.:   81.0
                    Max.   :10665.0
                    NA's   :9129
```

Here is a description of the columns of the dataset.
Name - The games name
Platform - Platform of the games release (i.e. PC,PS4, etc.)
Year - Year of the game's release
Genre - Genre of the game
Publisher - Publisher of the game
NA_Sales - Sales in North America (in millions)
EU_Sales - Sales in Europe (in millions)
JP_Sales - Sales in Japan (in millions)
Other_Sales - Sales in the rest of the world (in millions)
Global_Sales - Total worldwide sales.
Critic_score - Aggregate score compiled by Metacritic staff
Criticcount - The number of critics used in coming up with the Criticscore
User_score - Score by Metacritic's subscribers
Usercount - Number of users who gave the userscore
Developer - Party responsible for creating the game
Rating - The ESRB ratings

It is observed Year_of_Release and User_score is character, will change to numeric first.

```
ratings <- ratings %>% mutate(Year_of_Release = as.numeric(Year_of_Release),
                              User_Score = as.numeric(User_Score))
head(ratings)
```

```
                      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
```

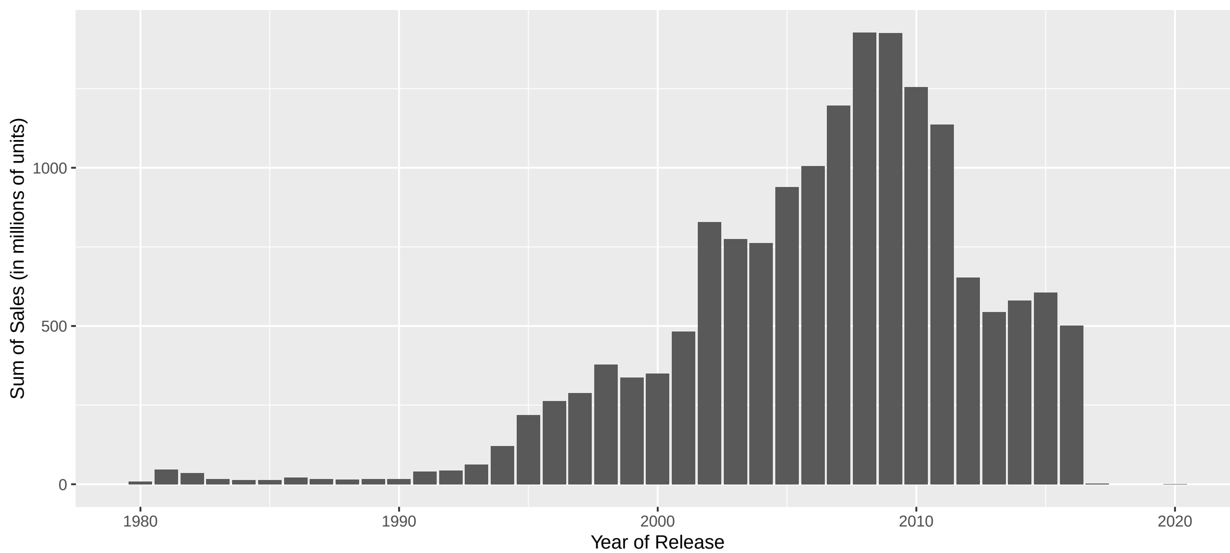
```
6          NA          NA
```

From summary we also observed that there is quite a lot of N/A in the dataset.

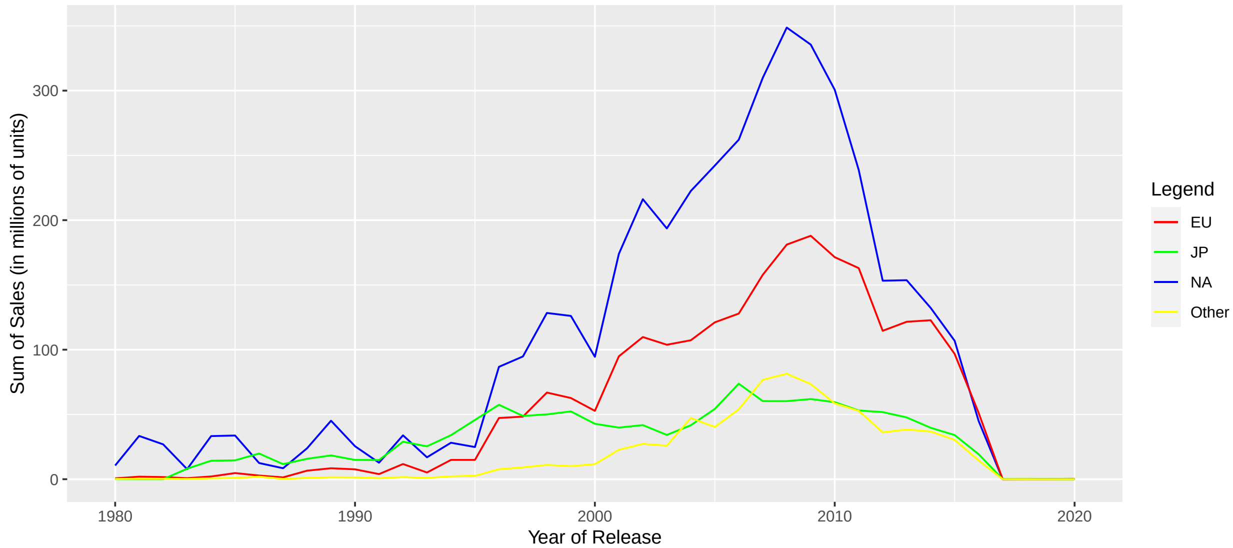
Data exploration and visualization

Game Sales by Year of Release

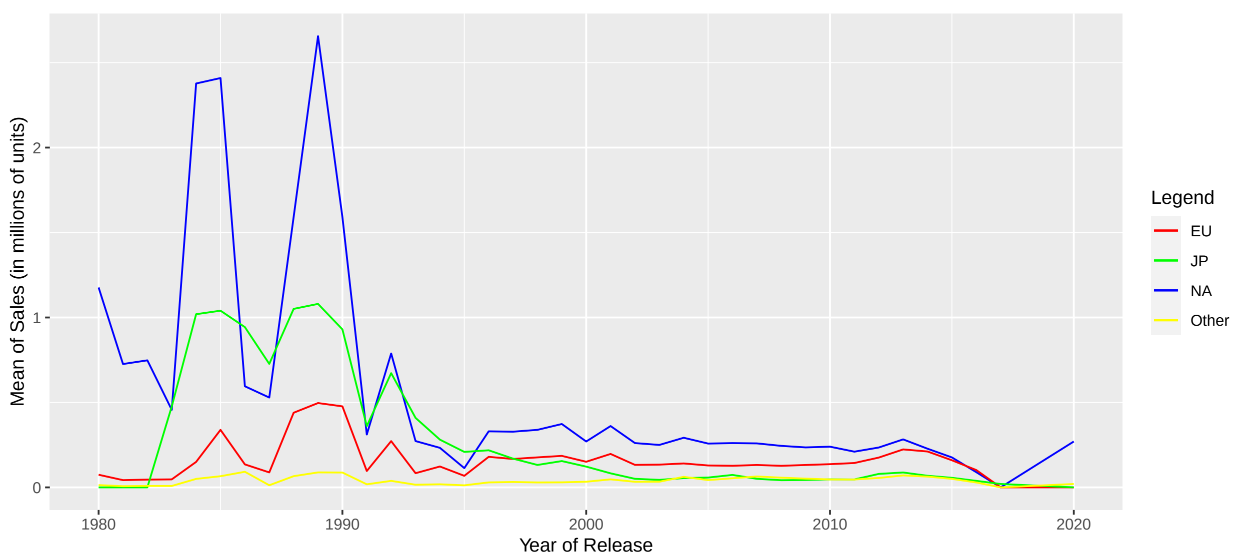
```
# Visualization on data
# Year_of_Release, dataset small before 1995
ggplot(ratings) +
  geom_bar(aes(Year_of_Release))+
  labs(x = "Year of Release",
       y = "Sum of Sales (in millions of units)")
```




```
# Sales by region
ratings %>% group_by(Year_of_Release) %>%
  summarise(Sum_NA_Sales=sum(NA_Sales), Sum_EU_Sales=sum(EU_Sales),
            Sum_JP_Sales=sum(JP_Sales), Sum_Other_Sales=sum(Other_Sales)) %>%
  ggplot() +
  geom_line(aes(x = Year_of_Release, y = Sum_NA_Sales, color = "NA"))+
  geom_line(aes(x = Year_of_Release, y = Sum_EU_Sales, color = "EU")) +
  geom_line(aes(x = Year_of_Release, y = Sum_JP_Sales, color = "JP")) +
  geom_line(aes(x = Year_of_Release, y = Sum_Other_Sales, color = "Other")) +
  labs(x = "Year of Release",
       y = "Sum of Sales (in millions of units)",
       color = "Legend") +
  scale_color_manual(values = c("NA" = "blue", "EU" = "red",
                                "JP" = "green", "Other" = "yellow"))
```




```r
# Average sales per region per year of release
ratings %>% group_by(Year_of_Release) %>%
  summarise(Mean_NA_Sales=mean(NA_Sales), Mean_EU_Sales=mean(EU_Sales),
            Mean_JP_Sales=mean(JP_Sales), Mean_Other_Sales=mean(Other_Sales)) %>%
  ggplot() +
  geom_line(aes(x = Year_of_Release, y = Mean_NA_Sales, color = "NA"))+
  geom_line(aes(x = Year_of_Release, y = Mean_EU_Sales, color = "EU")) +
  geom_line(aes(x = Year_of_Release, y = Mean_JP_Sales, color = "JP")) +
  geom_line(aes(x = Year_of_Release, y = Mean_Other_Sales, color = "Other")) +
  labs(x = "Year of Release",
       y = "Mean of Sales (in millions of units)",
       color = "Legend") +
  scale_color_manual(values = c("NA" = "blue", "EU" = "red",
                                "JP" = "green", "Other" = "yellow"))
```



We can see that the number of games released reached its peak at around 2008-2009. From the second graph, NA region contribute the most sales.

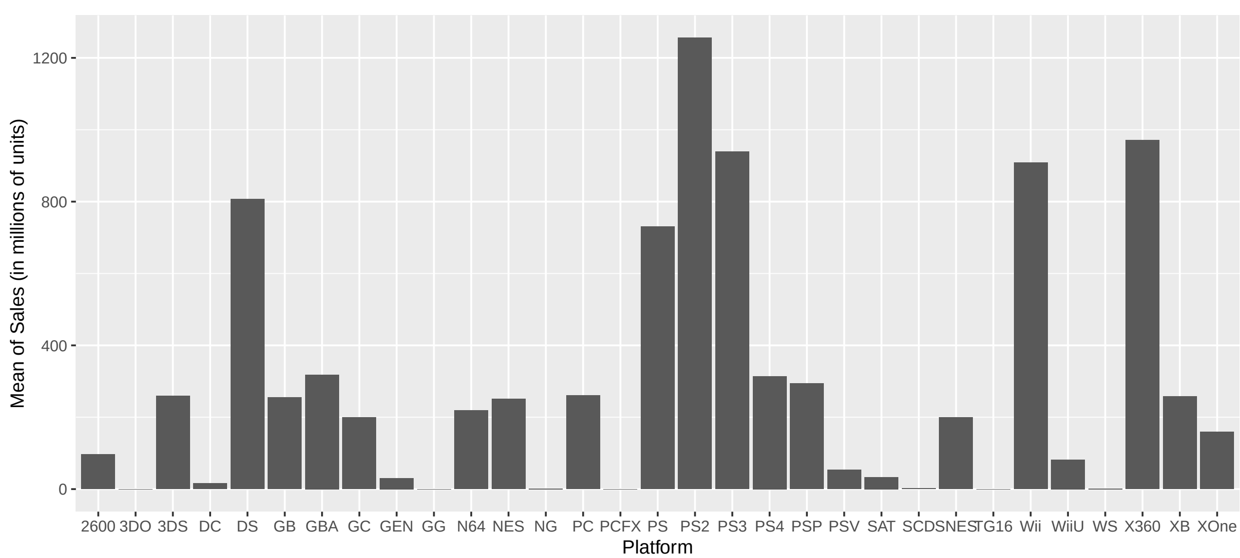
However if we plot the average sales by Year of Release, it is observed that the average sales was greatly declined from 1990 onward. This suggests there are increased competition throughout the market. This finding also means we have to take care on the Year_of_Release when building our model, cause the average sales per year is not stationary throughout the dataset.

Game Sales by Platform

```r
unique(ratings$Platform)
```

```
 [1] "Wii"   "NES"   "GB"    "DS"    "X360" "PS3"  "PS2"   "SNES" "GBA"  "PS4"
[11] "3DS"   "N64"   "PS"    "XB"    "PC"   "2600" "PSP"   "XOne" "WiiU" "GC"
[21] "GEN"   "DC"    "PSV"   "SAT"   "SCD"  "WS"   "NG"    "TG16" "3DO"  "GG"
[31] "PCFX"
```

```r
ratings %>% group_by(Platform) %>%
  summarise(Sales=sum(Global_Sales)) %>%
  ggplot(aes(x=Platform, y=Sales)) +
  geom_bar(stat="identity") +
  labs(x = "Platform",
       y = "Mean of Sales (in millions of units)")
```




```r
ratings %>% group_by(Platform) %>%
  summarise(Sales=sum(Global_Sales)) %>%
  arrange(Sales)
```

```
# A tibble: 31 x 2
   Platform Sales
   <chr>    <dbl>
 1 PCFX      0.03
 2 GG        0.04
 3 3DO       0.1
 4 TG16      0.16
```

```
 5 WS         1.42
 6 NG         1.44
 7 SCD        1.87
 8 DC        16.0
 9 GEN       30.8
10 SAT       33.6
# ... with 21 more rows
```

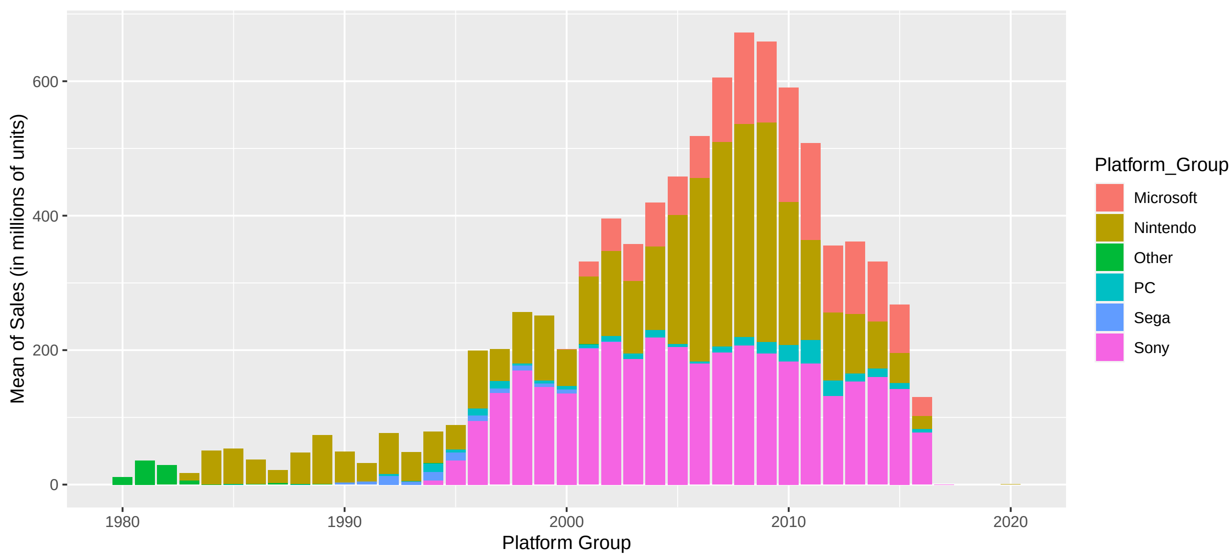
There are total 31 different platforms in the dataset. From the graphs, PS2 has the most sales throughout the dataset, it is also observed some platforms has nearly no sales, such as PCFX, GG, 3DO, TG16.

Next we will try to group the platforms by manufacturer. The grouping of platforms is referencing https://www.kaggle.com/leonardf/releases-and-sales.

```
nintendoplatforms = c("3DS","DS","GB","GBA","N64","GC", "NES","SNES","Wii","WiiU")
sonyplatforms = c("PS","PS2","PSP","PS3","PS4","PSV")
segaplatforms = c("GEN","SCD","DC","GG","SAT")
msplatforms = c("XB","X360", "XOne")
otherplatforms = c("2600","3DO","NG","PCFX","TG16","WS")
pc= c('PC')

ratings$Platform_Group[ratings$Platform %in% nintendoplatforms] <- "Nintendo"
ratings$Platform_Group[ratings$Platform %in% sonyplatforms] <- "Sony"
ratings$Platform_Group[ratings$Platform %in% msplatforms] <- "Microsoft"
ratings$Platform_Group[ratings$Platform %in% segaplatforms] <- "Sega"
ratings$Platform_Group[ratings$Platform %in% pc] <- "PC"
ratings$Platform_Group[ratings$Platform %in% otherplatforms] <- "Other"

ratings %>% group_by(Platform_Group, Year_of_Release) %>%
  summarise(Sales=sum(Global_Sales)) %>%
  ggplot(aes(fill=Platform_Group, y=Sales, x=Year_of_Release)) +
  geom_bar(position="stack", stat="identity") +
  labs(x = "Platform Group",
       y = "Mean of Sales (in millions of units)")
```

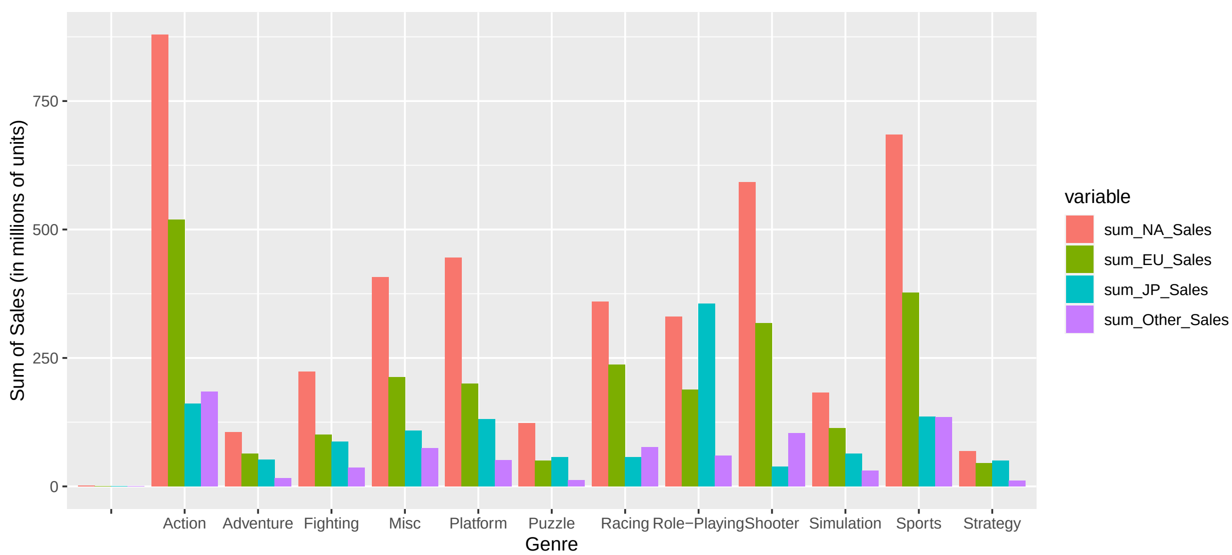


The result is pretty obvious that major platforms are Sony(Playstation), Microsoft(Xbox) and Nintendo. Meanwhile PC is not a mainstream of gaming and Sega products discontinued in mid 90s. The findings suggested that Platform is another factor that affects the sales.

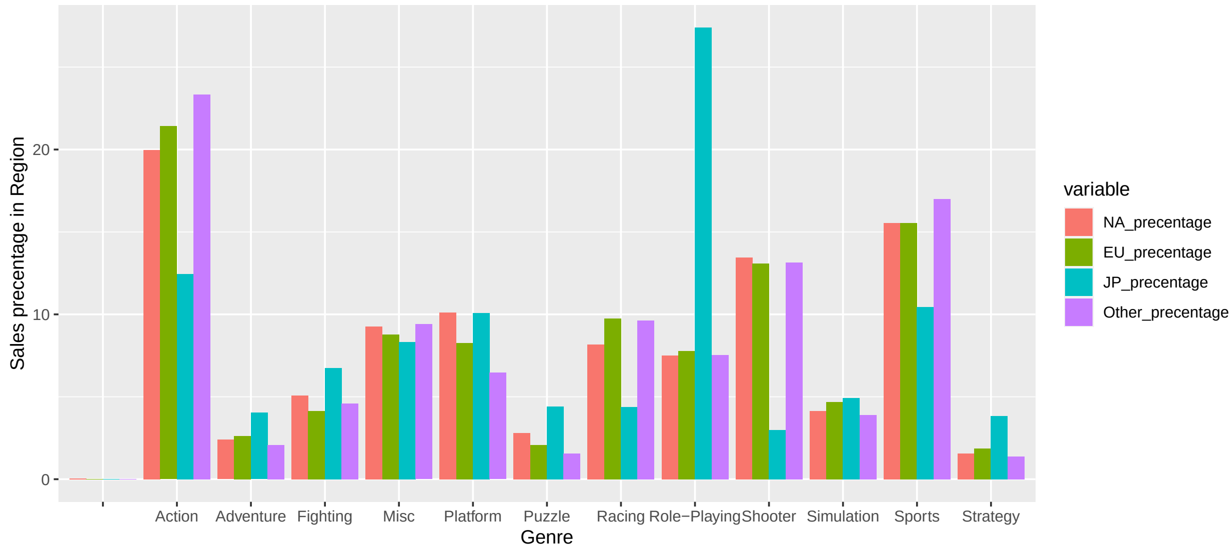
Game Sales by Genre

```
# Check by genre with region preference
sales_by_genre <- ratings %>% group_by(Genre) %>%
  summarise(sum_NA_Sales=sum(NA_Sales), sum_EU_Sales=sum(EU_Sales),
            sum_JP_Sales=sum(JP_Sales), sum_Other_Sales=sum(Other_Sales),
            avg_NA_Sales=mean(NA_Sales), avg_EU_Sales=mean(EU_Sales),
            avg_JP_Sales=mean(JP_Sales), avg_Other_Sales=mean(Other_Sales)) %>%
  mutate(NA_precentage = sum_NA_Sales/sum(ratings$NA_Sales)*100,
         EU_precentage = sum_EU_Sales/sum(ratings$EU_Sales)*100,
         JP_precentage = sum_JP_Sales/sum(ratings$JP_Sales)*100,
         Other_precentage = sum_Other_Sales/sum(ratings$Other_Sales)*100)

melt(select(sales_by_genre,
            c("Genre", "sum_NA_Sales", "sum_EU_Sales", "sum_JP_Sales", "sum_Other_Sales")),
     c("Genre")) %>%
  ggplot(aes(fill=variable, y=value, x=Genre))+
  geom_bar(position="dodge", stat="identity") +
  labs(x = "Genre",
       y = "Sum of Sales (in millions of units)")
```




```
melt(select(sales_by_genre,
            c("Genre", "NA_precentage", "EU_precentage", "JP_precentage", "Other_precentage")),
     c("Genre")) %>%
  ggplot(aes(fill=variable, y=value, x=Genre))+
  geom_bar(position="dodge", stat="identity") +
  labs(x = "Genre",
       y = "Sales precentage in Region")
```



From graphs, Action, Sports, Platform and Shooter are popular genre among 12 game genres.
However if we plot genre sales against sales of the region, we can see different region has different preference on genre. For example, Japan market highly interested in Role Playing and not interested in Shooting. Generally speaking, Japan has a different genre preference with other region.
This finding suggested that Genre is another factor of sales, and region has effect on the genre sales.

Game sales by Publisher

```
length(unique(ratings$Publisher))
```

```
[1] 582
```

```
publisher_sales <- ratings %>% group_by(Publisher) %>%
  summarise(sum_Sales=sum(Global_Sales), avg_Sales=mean(Global_Sales))
arrange(publisher_sales, -sum_Sales)
```

```
# A tibble: 582 x 3
   Publisher                    sum_Sales avg_Sales
   <chr>                            <dbl>     <dbl>
 1 Nintendo                         1789.     2.53
 2 Electronic Arts                  1117.     0.824
 3 Activision                        731.     0.742
 4 Sony Computer Entertainment       606.     0.883
 5 Ubisoft                           472.     0.505
 6 Take-Two Interactive              404.     0.957
 7 THQ                               338.     0.473
 8 Konami Digital Entertainment      282.     0.339
 9 Sega                              270.     0.424
10 Namco Bandai Games                255.     0.271
# ... with 572 more rows
```

There are total 582 publisher in the dataset, if we take a look at the top sales publisher, we can see Nintendo is the top saler, but its average sales is not the highest, may suggest that Nintendo only have a lot of games which the total Sales.

Modeling building

For model building, I only pick some columns, in previous sections, although we know Genre has impact on regional sales, but for simplicity, I will use Global_Sales as the prediction.

```
# For modelling, only take the useful columns
# Global sales will be the prediction, may try other sales later
# Developer is too similar with Publisher, so I dropped it first
# Platform is too wide, will be using Platform_Group
d_model <-  ratings[,c('Global_Sales',
                  'Name',
                  'Year_of_Release',
                  'Publisher',
                  'Platform_Group',
                  'Genre',
                  'Critic_Score',
                  'Critic_Count',
                  'User_Score',
                  'User_Count')]
```

In Dataset Preparation, there are N/As in the dataset, namely Year_of_Release, User_Score, User_Count, Critic_Score and Critic_Count, we will need to deal with them first.

```
# NAs, remove Year Na's as 269/16719 not a great problem
d_model <- d_model %>% filter(!is.na(d_model$Year_of_Release))

# Name empty, 2 record
d_model <- d_model %>% filter(d_model$Name != "")
summary(d_model)
```

```
  Global_Sales         Name           Year_of_Release  Publisher
 Min.   : 0.0100   Length:16448       Min.   :1980    Length:16448
 1st Qu.: 0.0600   Class :character   1st Qu.:2003    Class :character
 Median : 0.1700   Mode  :character   Median :2007    Mode  :character
 Mean   : 0.5362                      Mean   :2006
 3rd Qu.: 0.4700                      3rd Qu.:2010
 Max.   :82.5300                      Max.   :2020

 Platform_Group        Genre            Critic_Score    Critic_Count
 Length:16448       Length:16448       Min.   :13.00   Min.   :  3.00
 Class :character   Class :character   1st Qu.:60.00   1st Qu.: 12.00
 Mode  :character   Mode  :character   Median :71.00   Median : 22.00
                                       Mean   :68.99   Mean   : 26.44
                                       3rd Qu.:79.00   3rd Qu.: 36.00
                                       Max.   :98.00   Max.   :113.00
                                       NA's   :8465    NA's   :8465
   User_Score      User_Count
 Min.   :0.000   Min.   :    4
 1st Qu.:6.400   1st Qu.:   10
 Median :7.500   Median :   24
 Mean   :7.126   Mean   :  163
 3rd Qu.:8.200   3rd Qu.:   81
 Max.   :9.700   Max.   :10665
 NA's   :8985    NA's   :8985
```

Checking on the NAs of User_Score, User_Count, Critic_Score and Critic_Count.

```
colMeans(is.na(d_model))
```

```
   Global_Sales            Name Year_of_Release       Publisher  Platform_Group
      0.0000000       0.0000000       0.0000000       0.0000000       0.0000000
          Genre    Critic_Score    Critic_Count      User_Score      User_Count
      0.0000000       0.5146522       0.5146522       0.5462670       0.5462670
```

Both User_Score and Critic_Score have more than 50% NA. Although I assume the score is useful for predicting the sales. I will drop them first the first model building.

```
# over 50% of critic_score and user_score is missing.
# So continue dropping those NA
d_model <- d_model %>% filter(!is.na(d_model$User_Score))
colMeans(is.na(d_model))
```

```
   Global_Sales            Name Year_of_Release       Publisher  Platform_Group
      0.0000000       0.0000000       0.0000000       0.0000000       0.0000000
          Genre    Critic_Score    Critic_Count      User_Score      User_Count
      0.0000000       0.0762428       0.0762428       0.0000000       0.0000000
```

```
# Fill the 7.6% of Critic_Score to median
d_model[is.na(d_model$Critic_Score), "Critic_Score"] <- mean(d_model$Critic_Score, na.rm=TRUE)
d_model[is.na(d_model$Critic_Count), "Critic_Count"] <- mean(d_model$Critic_Count, na.rm=TRUE)
colMeans(is.na(d_model))
```

```
   Global_Sales            Name Year_of_Release       Publisher  Platform_Group
              0               0               0               0               0
          Genre    Critic_Score    Critic_Count      User_Score      User_Count
              0               0               0               0               0
```

Now all the variables used for prediction is ready. For better prediction, I build dummy variables on the categories columns. The final model looks like this.

```
# Create dummy vars
dummies <- dummyVars(Global_Sales ~ Platform_Group+Genre, data = d_model)
dummies_model <- predict(dummies, newdata = d_model)

# Create final prediction model to be use
p_model = cbind(d_model, dummies_model) %>% select(-c(Platform_Group, Genre, Name))
p_model$Publisher <- as.factor(p_model$Publisher)
head(p_model)
```

```
  Global_Sales Year_of_Release Publisher Critic_Score Critic_Count User_Score
1        82.53            2006  Nintendo           76           51        8.0
2        35.52            2008  Nintendo           82           73        8.3
3        32.77            2009  Nintendo           80           73        8.0
4        29.80            2006  Nintendo           89           65        8.5
5        28.92            2006  Nintendo           58           41        6.6
6        28.32            2009  Nintendo           87           80        8.4
```

```
  User_Count Platform_GroupMicrosoft Platform_GroupNintendo Platform_GroupPC
1        322                       0                      1                0
2        709                       0                      1                0
3        192                       0                      1                0
4        431                       0                      1                0
5        129                       0                      1                0
6        594                       0                      1                0
  Platform_GroupSega Platform_GroupSony GenreAction GenreAdventure
1                  0                  0           0              0
2                  0                  0           0              0
3                  0                  0           0              0
4                  0                  0           0              0
5                  0                  0           0              0
6                  0                  0           0              0
  GenreFighting GenreMisc GenrePlatform GenrePuzzle GenreRacing
1             0         0             0           0           0
2             0         0             0           0           1
3             0         0             0           0           0
4             0         0             1           0           0
5             0         1             0           0           0
6             0         0             1           0           0
  GenreRole-Playing GenreShooter GenreSimulation GenreSports GenreStrategy
1                 0            0               0           1             0
2                 0            0               0           0             0
3                 0            0               0           1             0
4                 0            0               0           0             0
5                 0            0               0           0             0
6                 0            0               0           0             0
```

We can start the model training. I used 4 methods to find the best one, namely normal linear regression, elastic net, support-vector machines and random forest. I separated the dataset into 80% training and 20% validation.

```r
# Start modeling
set.seed(255)

trainRowNumbers <- createDataPartition(p_model$Global_Sales, p=0.8, list=FALSE)
trainData <- p_model[trainRowNumbers,]
testData <- p_model[-trainRowNumbers,]

# Start training
train_ctr <- trainControl(method="LGOCV", number=3)
```

As stated in previous section, I think Year of Release has effect on sales, also older games should have longer sale period hence higher sales, therefore I will add weights to Year_of_Release to cater recently released games.

Linear Regression

```r
# Linear regression
set.seed(255)
M_lm <- train(Global_Sales ~ .,
```

```r
                weights=exp(trainData$Year_of_Release-min(trainData$Year_of_Release)+1),
                data=trainData,
                trControl=train_ctr,
                method="lm")
train_lm <- predict(M_lm,trainData)
RMSE(train_lm, trainData$Global_Sales)
```

```
[1] 1.988054
```

The RMSE for linear regression shown above.

Elastic Net

```r
# Elastic net
set.seed(255)
glmnetgrid <-expand.grid(alpha=c(0.1,0.55,1),lambda=seq(0,0.5,0.1))
M_glmnet<- train(Global_Sales ~ .,
                 weights=exp(trainData$Year_of_Release-min(trainData$Year_of_Release)+1),
                 data=trainData,
                 trControl=train_ctr,
                 method="glmnet",
                 tuneGrid=glmnetgrid)
train_glmnet <- predict(M_glmnet,trainData)
RMSE(train_glmnet, trainData$Global_Sales)
```

```
[1] 1.845451
```

The RMSE for elastic net shown above.

Support-vector machines

```r
# SVM
set.seed(255)
M_svm <- train(Global_Sales ~ .,
               weights=exp(trainData$Year_of_Release-min(trainData$Year_of_Release)+1),
               data=trainData,
               method="svmRadial",
               trControl=train_ctr)
train_svm <- predict(M_svm, trainData)
RMSE(train_svm, trainData$Global_Sales)
```

```
[1] 1.778926
```

The RMSE for svm shown above.

Random Forest

```r
# RF
set.seed(255)
M_rf <- train(Global_Sales~.,
              weights=exp(trainData$Year_of_Release-min(trainData$Year_of_Release)+1),
              data=trainData,
              method="rf",
              tuneLength=2,
              trControl=train_ctr)
train_rf <- predict(M_rf,trainData)
RMSE(train_rf, trainData$Global_Sales)
```

```
[1] 0.7139237
```

The RMSE for random forest shown above.

It seems random forest provide a better model, we will move on to use the validation set.

Validation

```r
# Model testing
test_lm <- predict(M_lm, testData)
RMSE(test_lm, testData$Global_Sales)
```

```
[1] 1.731459
```

```r
test_glmnet <- predict(M_glmnet, testData)
RMSE(test_glmnet, testData$Global_Sales)
```

```
[1] 1.571988
```

```r
test_svm <- predict(M_svm, testData)
RMSE(test_svm, testData$Global_Sales)
```

```
[1] 1.552982
```

```r
test_rf <- predict(M_rf, testData)
RMSE(test_rf, testData$Global_Sales)
```

```
[1] 1.158901
```

Conclusion

Random forest model provided the lowest RMSE among all 4 models. Actually the RMSE result is not that good as expected.
I suspected there is some extreme outliers, such as Nintendo with a lot of games published, and some indie companies which only publish one game with extreme good sales.

The factor of user score and critic score is also not fully used in the model, as we dropped over 50% of the records because of NAs. I think there should be ways to fill up the missing values, such as using knnImpute() in Caret library to preprocess the values. Or maybe using SVD on this sparse dataset to predict the score. Increasing the datapoints may improve the model.

Finally, as stated in the analytic part, some features such as region genre preference is not used in this model. With separate prediction of regional sales may provide a better prediction. Also adding predictors such as sales of Publisher, Genre average sales may increase the accuaracy of the model.

For further implementation of the model to provide prediction on sales, it should cater new games which can be achieved using nearest neighbor to find similar publisher and games with similar genres and try to predict the sales with reference to existing data.