VideoGamesSalesPredictor_Report

Ragini

June 15, 2019

Preface

This is my submission for the do your own project of the Data Science Capstone course provided by HarvardX in association with edX.org. The objective of this document is to explain the procedure and present results from an approach to the Project unedrtaken. I have selected Video Games Sales dataset fom Kaggle to build a sales prediction model.

1. Introduction

Video games is one of the most popular entertainment among kids, even adults also get fascinated to it. It gets more popularity because user can directly interact and get the feedback/outcome from the device, through which they earn points, which gives motivation, confidence while playing. The dataset consists of following variables Name, Platform, Year_of_Release, Genre, Publisher,Developer, NA_Sales, EU_Sales, JP_Sales, Other_Sales, Global_Sales along with this it has user score, count, critic score, count and ESRB ratings.

```
So in this report let us try to find answers for the questions like,

  1. Most Popular Games

  2. Which Genre is popular

  3. Which year sales was at its peak

  4. Region wise sales

  5. Yearwise game releases and so on....
```

Then I tried to build a prediction model by using linear regression. However, I found that the RMSE is too high for linear regression on this datset. Hence, referring to the correlation between the data fields I applied polynomial regression of degree 3 and result in a better prediction model.

2. Analysis

In order to start our analysis, we first need to get the dataset and required libraries, then we’ll be able to get some insights on the data.

2.1 Data Preparation

Loading the required libraries

Sourcing Data

```
#Data Sourcing
#https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings/downloads/video-game-sales-with
-ratings.zip/2
temp <- tempfile()
wd <- tempdir()
download.file("https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings/downloads/video-g
ame-sales-with-ratings.zip/2",temp, mode="wb")
wd <- getwd()
unzip(temp, wd)
games <- read.csv(paste(wd,"Video_Games_Sales_as_at_22_Dec_2016.csv", sep="/"),,stringsAsFactors
= FALSE)
unlink(c(temp, wd))
str(games)
```

```
## 'data.frame':    16719 obs. of  16 variables:
##  $ Name           : chr  "Wii Sports" "Super Mario Bros." "Mario Kart Wii" "Wii Sports Resor
t" ...
##  $ Platform       : chr  "Wii" "NES" "Wii" "Wii" ...
##  $ Year_of_Release: chr  "2006" "1985" "2008" "2009" ...
##  $ Genre          : chr  "Sports" "Platform" "Racing" "Sports" ...
##  $ Publisher      : chr  "Nintendo" "Nintendo" "Nintendo" "Nintendo" ...
##  $ NA_Sales       : num  41.4 29.1 15.7 15.6 11.3 ...
##  $ EU_Sales       : num  28.96 3.58 12.76 10.93 8.89 ...
##  $ JP_Sales       : num  3.77 6.81 3.79 3.28 10.22 ...
##  $ Other_Sales    : num  8.45 0.77 3.29 2.95 1 0.58 2.88 2.84 2.24 0.47 ...
##  $ Global_Sales   : num  82.5 40.2 35.5 32.8 31.4 ...
##  $ Critic_Score   : int  76 NA 82 80 NA NA 89 58 87 NA ...
##  $ Critic_Count   : int  51 NA 73 73 NA NA 65 41 80 NA ...
##  $ User_Score     : chr  "8" "" "8.3" "8" ...
##  $ User_Count     : int  322 NA 709 192 NA NA 431 129 594 NA ...
##  $ Developer      : chr  "Nintendo" "" "Nintendo" "Nintendo" ...
##  $ Rating         : chr  "E" "" "E" "E" ...
```

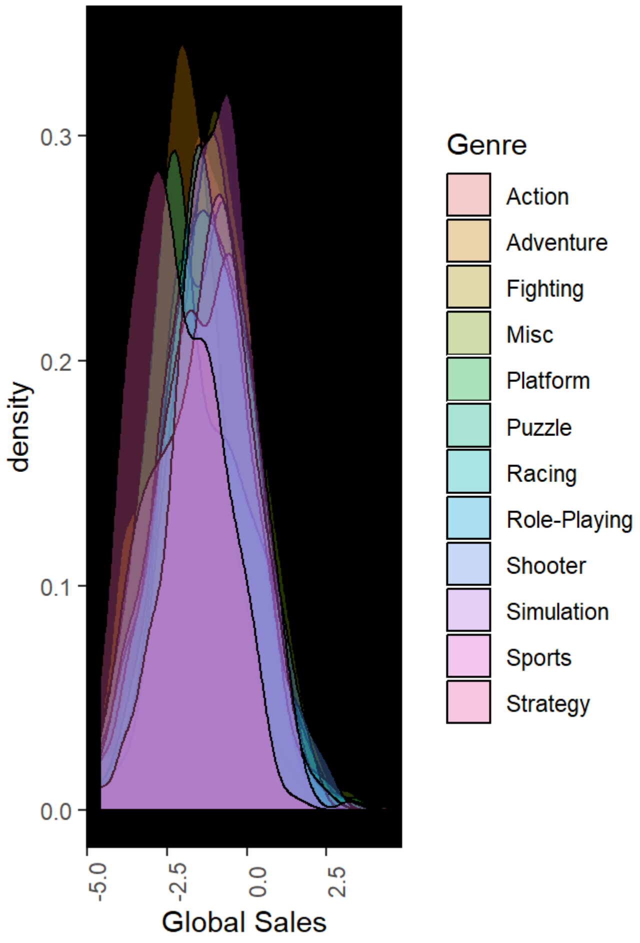
Dataset has details about the video game Name, publishers, year it got released and also has sales and rating details. Video games has been classified in to 12 genres. Sales of the games are given as Global sales and region wise, which has 4 groups(NA,EU, JP,other).

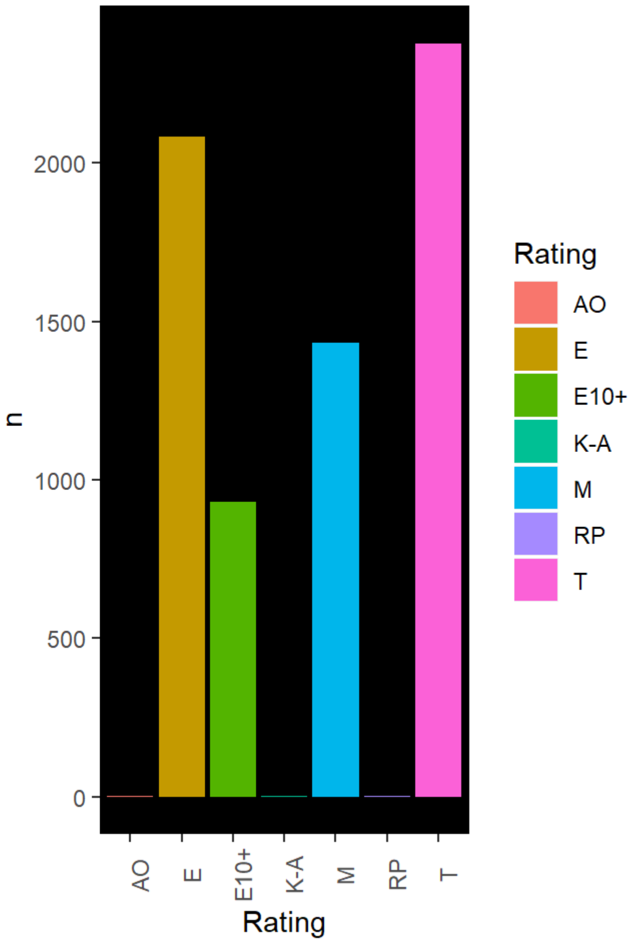
Converting the variables User_Count, User_Score, Critic_Count, Critic_Score to Numeric since it was in character. Also filter the dataset to get data only till year 2017.

```
#change factor into numeric
games$User_Count<-as.numeric(as.character(games$User_Count))
games$User_Score<-as.numeric(as.character(games$User_Score))
games$Critic_Count<-as.numeric(as.character(games$Critic_Count))
games$Critic_Score<-as.numeric(as.character(games$Critic_Score))
games<-games%>%filter(Year_of_Release<=2017)
#create a new dataframe cleanup of all NA's ...
games2 <- na.omit(games)
#there are still few rows for which the Rating is an empty string
games2<-filter(games2,Rating!='')

#create new columns to regroup the Platform by manufacturers
sony<-c('PS','PS2','PS3','PS4' ,'PSP','PSV')
microsoft<-c('PC','X360','XB','XOne')
nintendo<-c('3DS','DS','GBA','GC','N64','Wii','WiiU')
sega<-c('DC')
newPlatform<-function(x){
  if (x %in% sony == TRUE) {return('SONY')}
  else if(x %in% microsoft == TRUE) {return('MICROSOFT')}
  else if(x %in% nintendo == TRUE) {return('NINTENDO')}
  else if(x %in% sega == TRUE) {return('SEGA')}
  else{return('OTHER')}
}
games2$newPlatform<-sapply(games2$Platform, newPlatform)
```

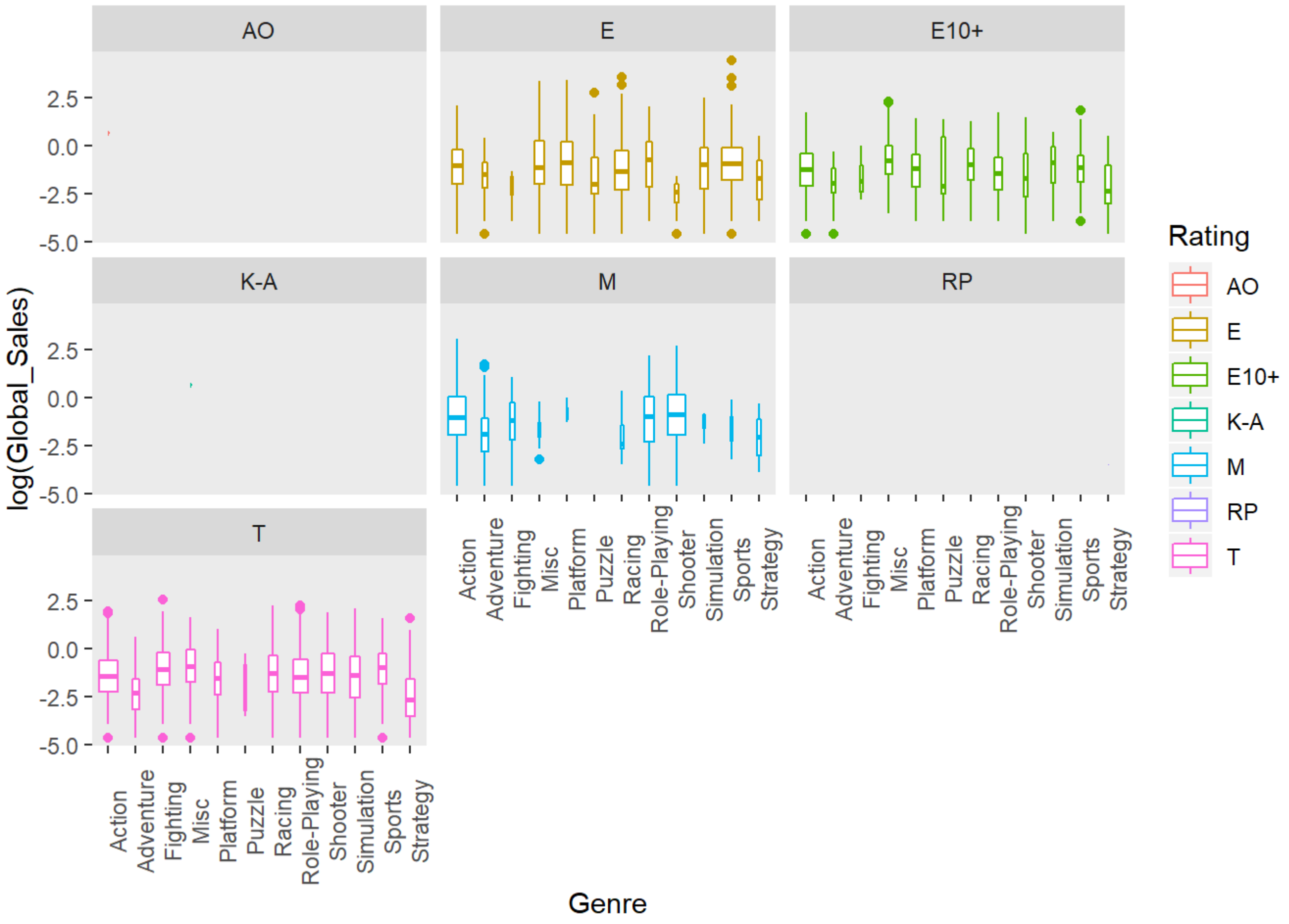
2.2 Data Visualization

Distribution of Global Sales across the genres

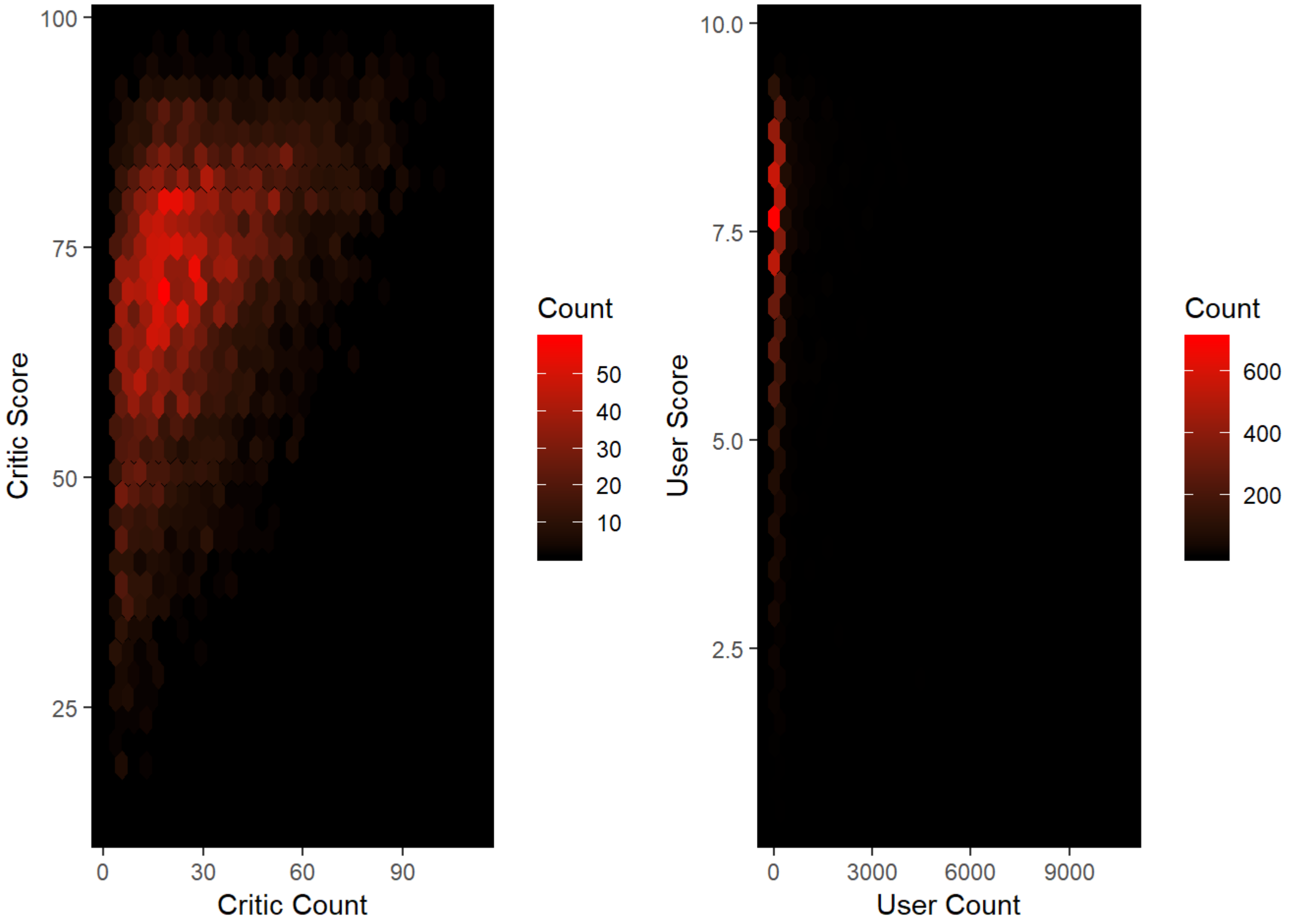




Distribution of Global Sales across Genres and Rating

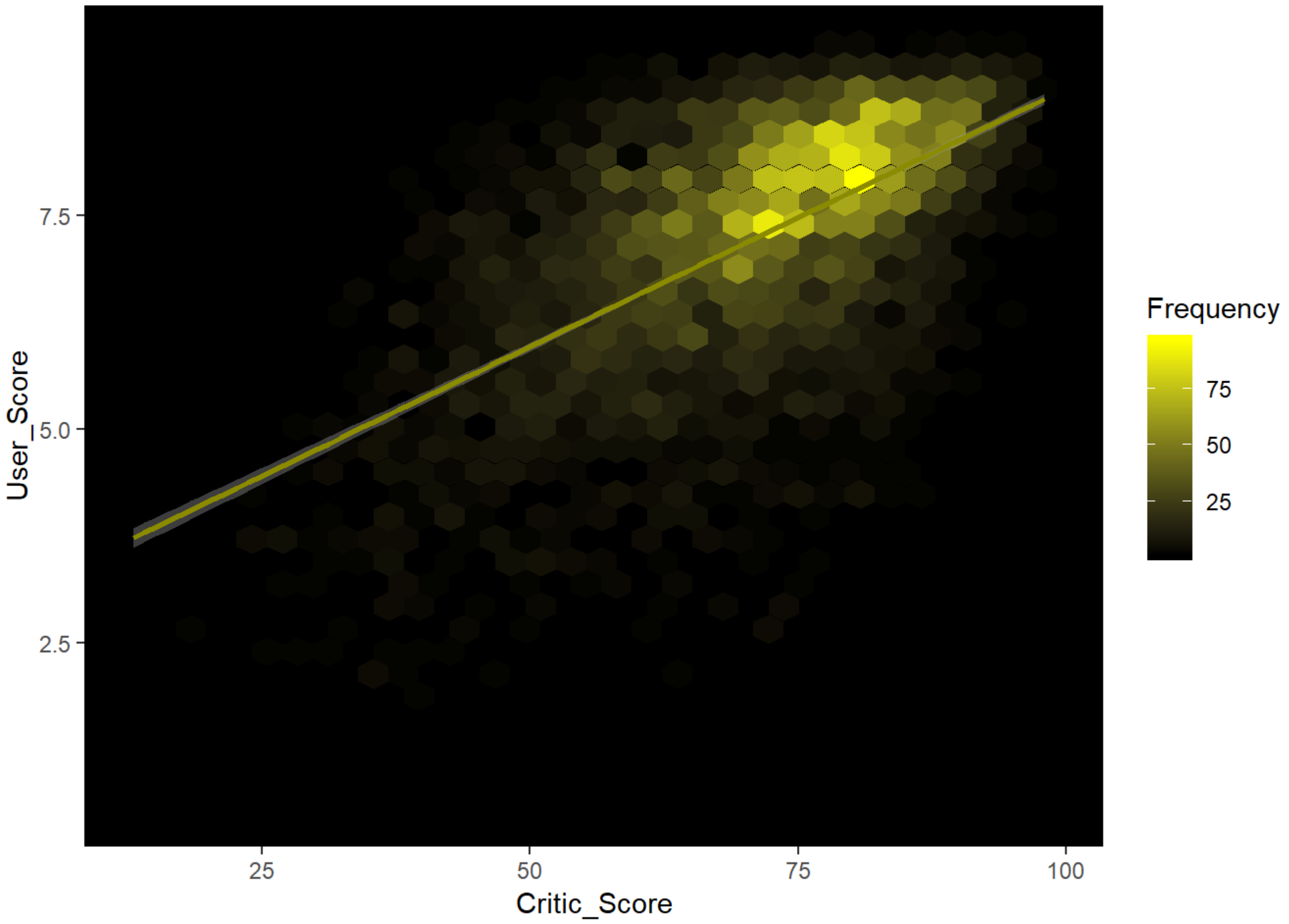


Relationship between Critic Score, Critic Count and User Score, User Count, since there exists blank space in User count and user score, normalizing it for plotting purposes.



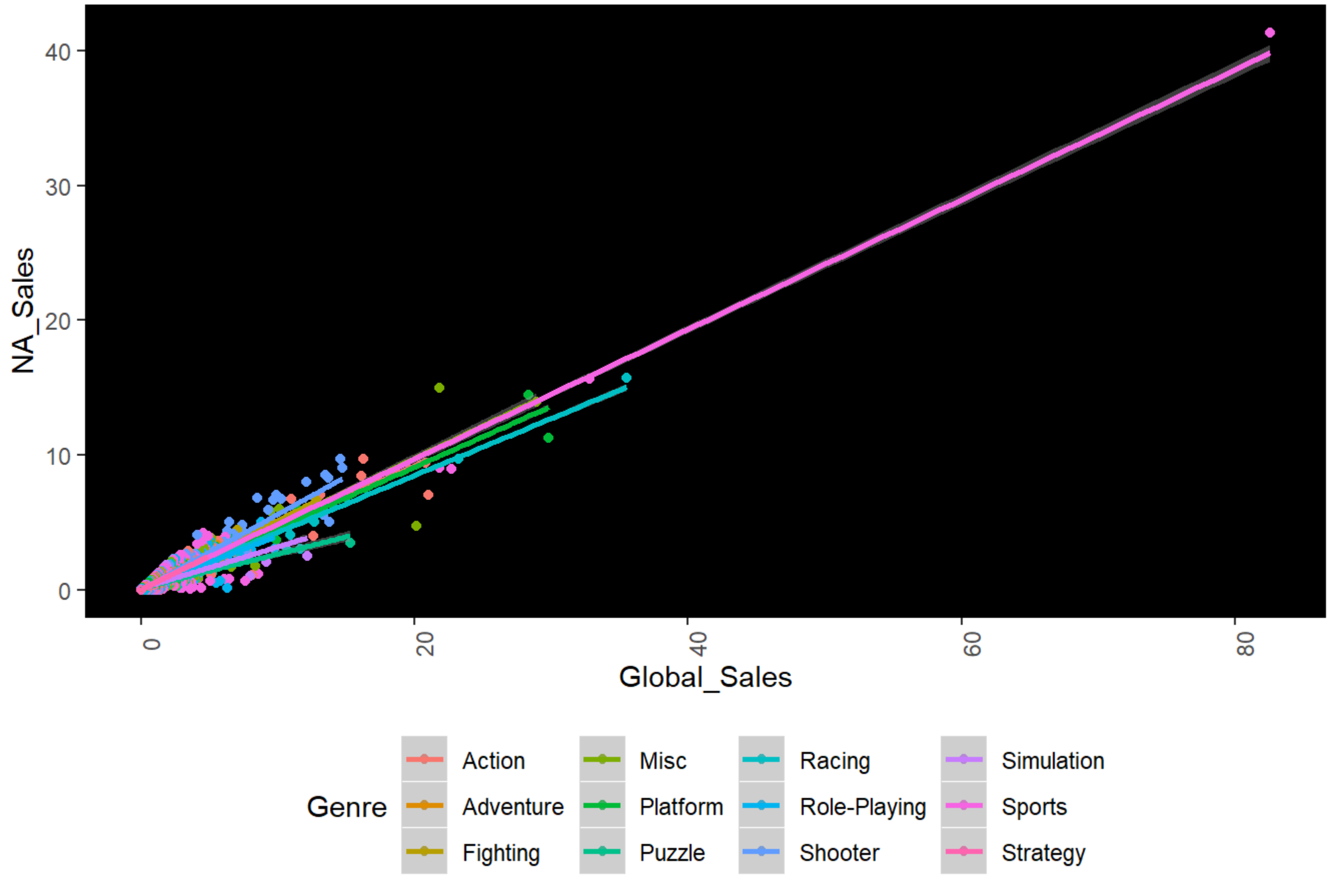
Critic Count and Critic Score are somehow correlated . Critic Count doesn’t have impact over critic Score, whereas User Count and User score doesn’t seem to have any relation. Reason may be due to null values existence in both the columns. Next find the correlation between User Score and critic Score

Correlation between User Score and Critic Score

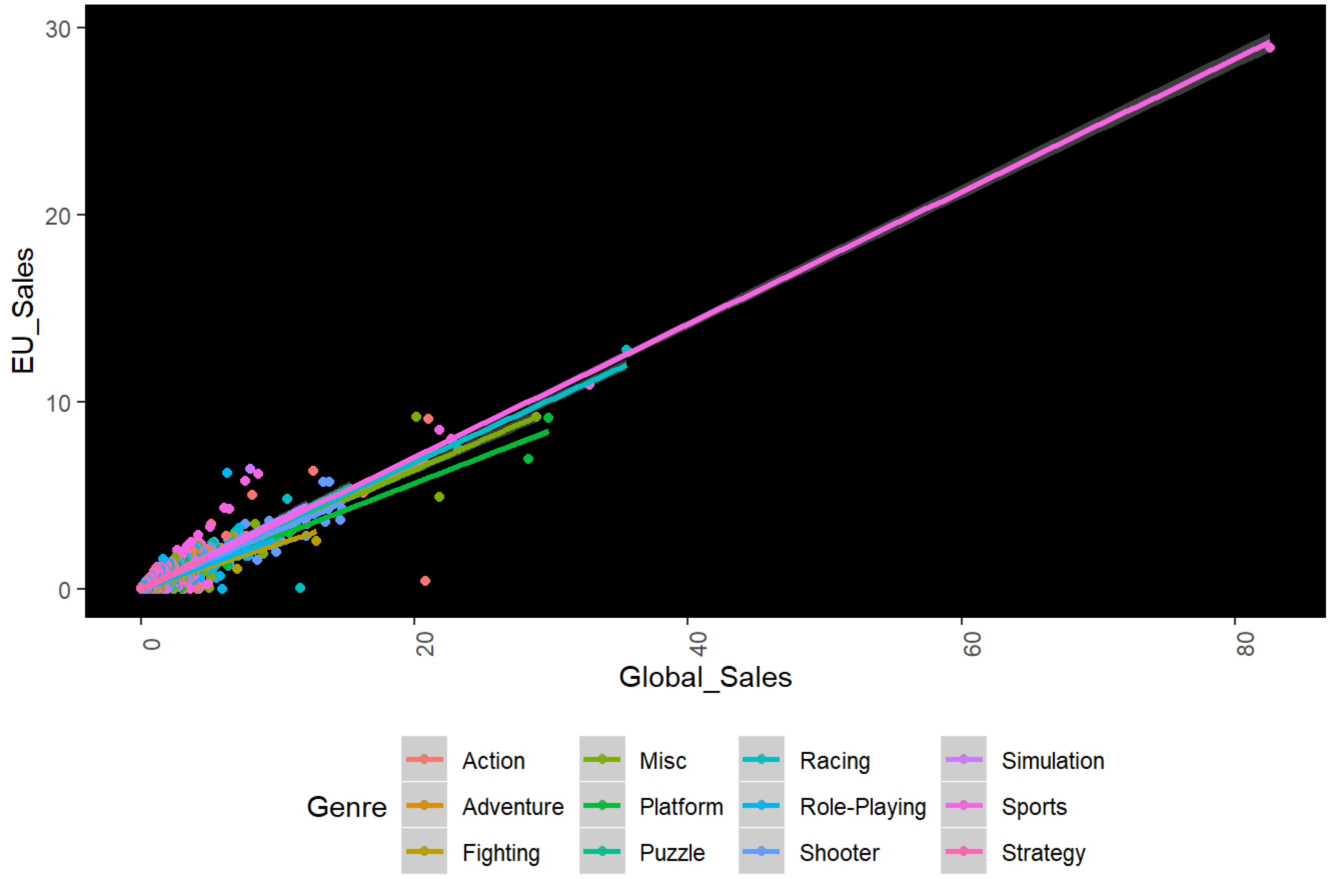


Critic Score and user Score has got positive correlation. As critic score increases, user score also got increases. Next will find out is there any correlaion exists between scores and the sales.

Global Sales Vs NA Sales across Genres

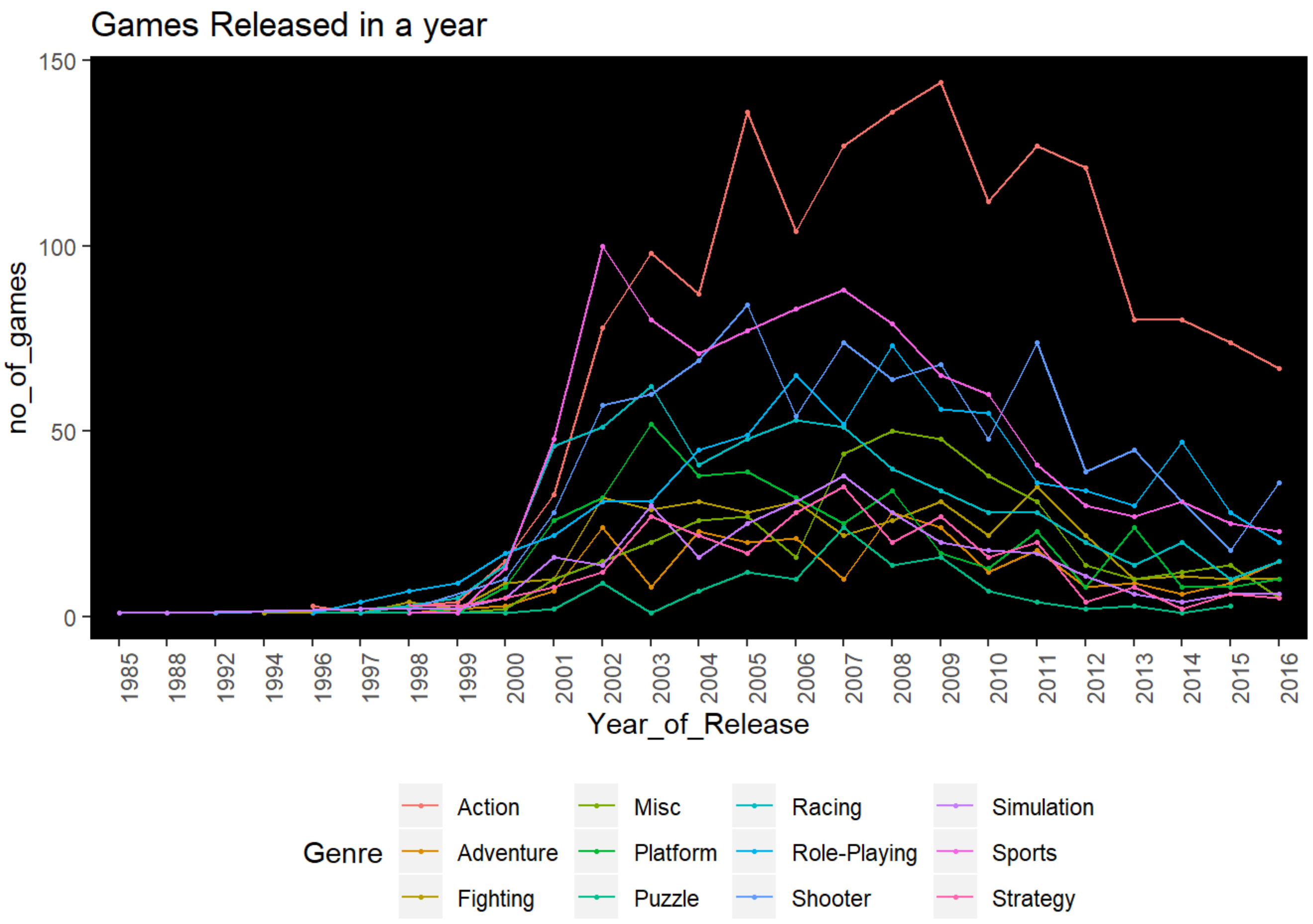


Global Sales Vs EU Sales across Genres



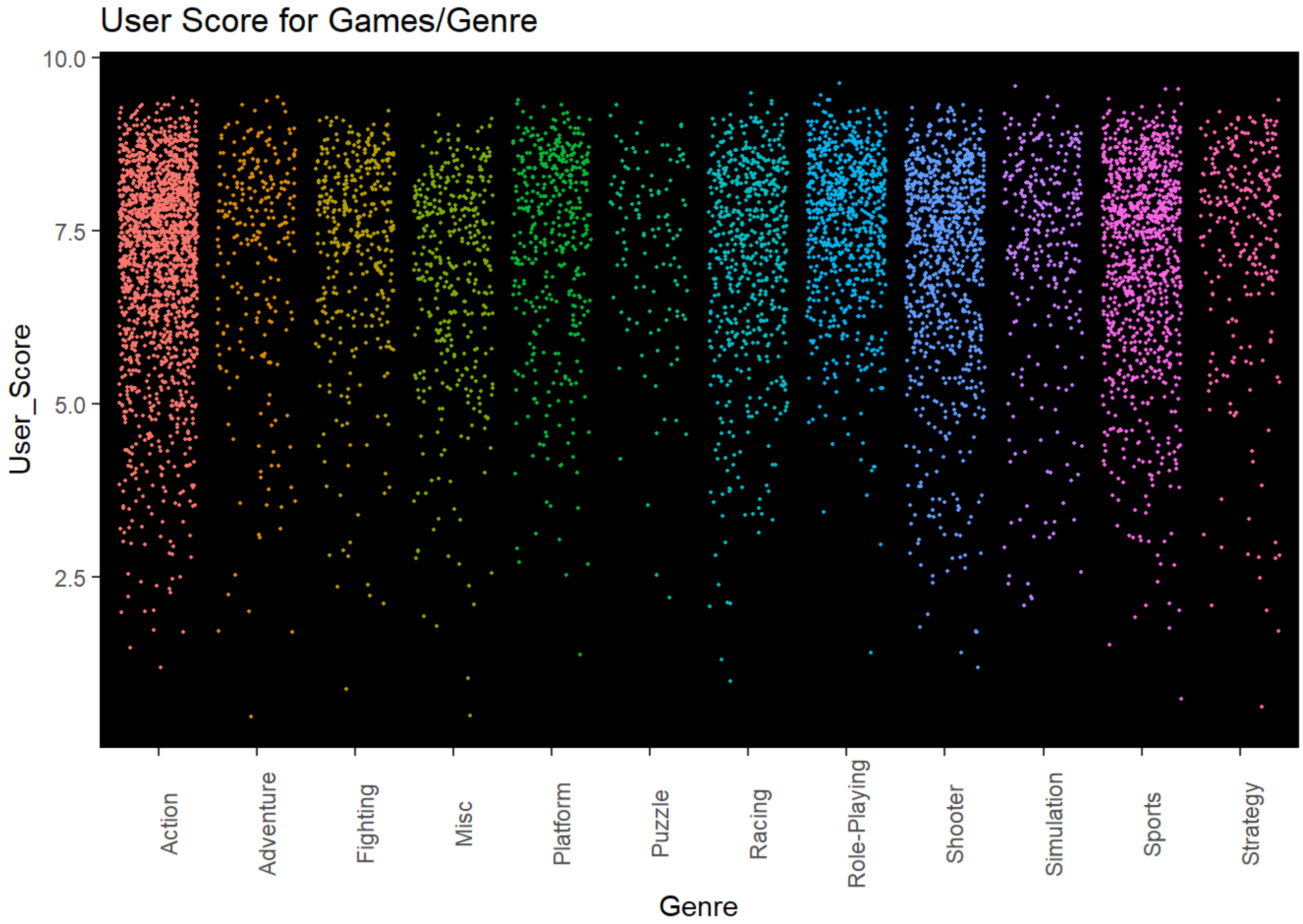
2.3 Data Analysis

1. Number of Video games Released in each year

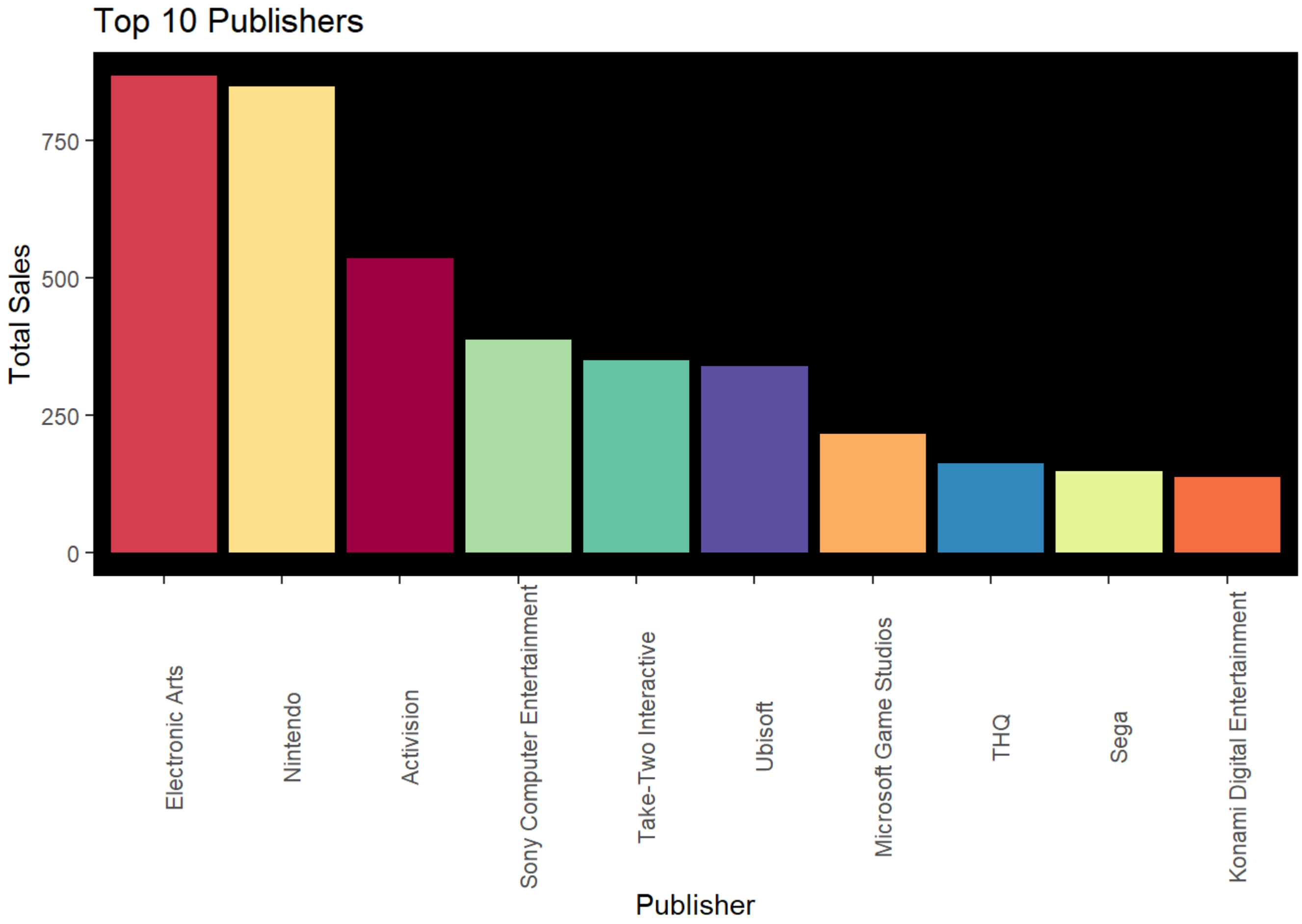


Number of games released was very less during the initial period(1980 to 1995), take off started after that and attain its peak during 2005-2012.Growth was clear for Action,Sports genre games, maximum number of games released per year was around 280 in action genre.

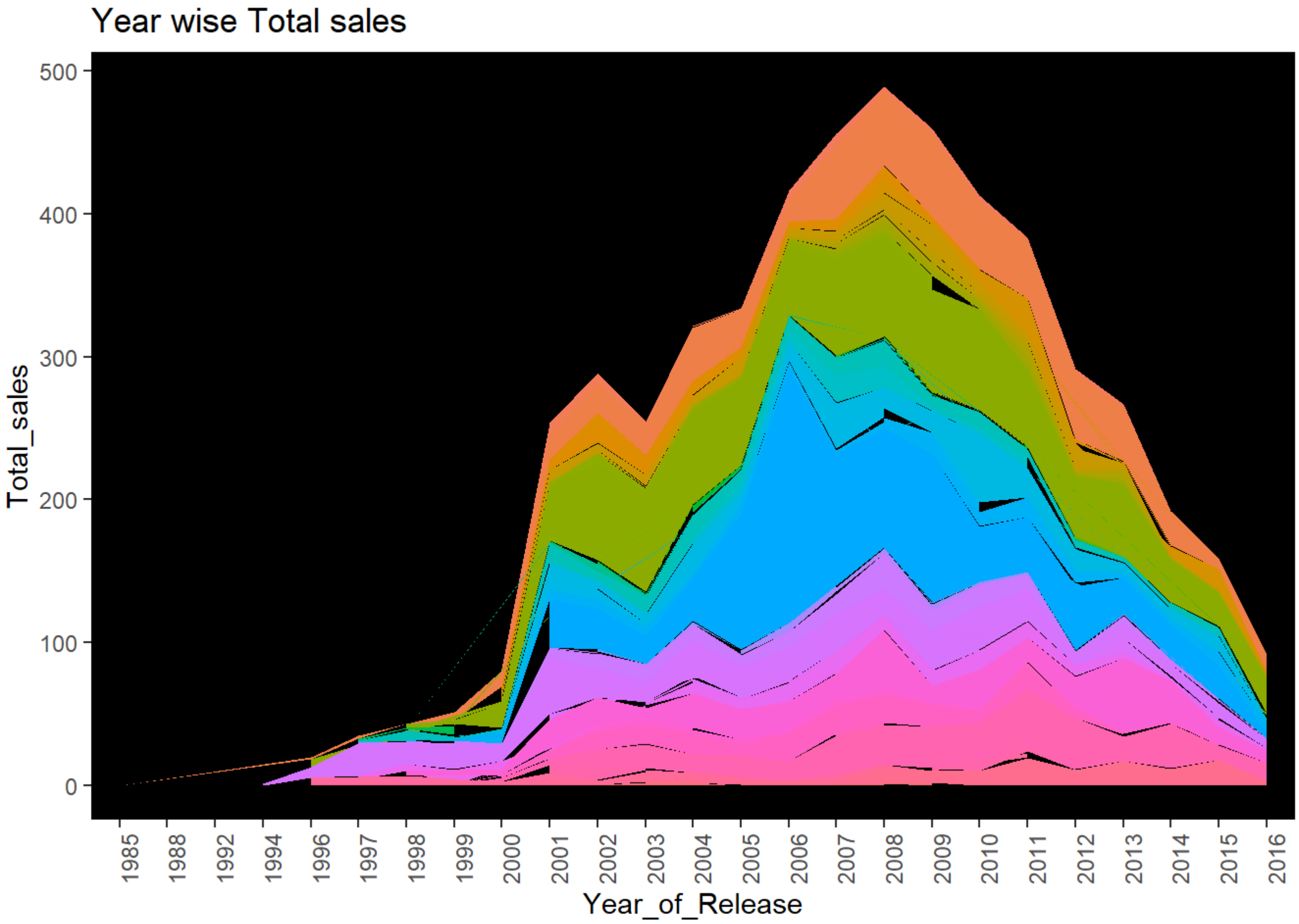
2. User Score for each genre games



3. Top Video Game Publisher

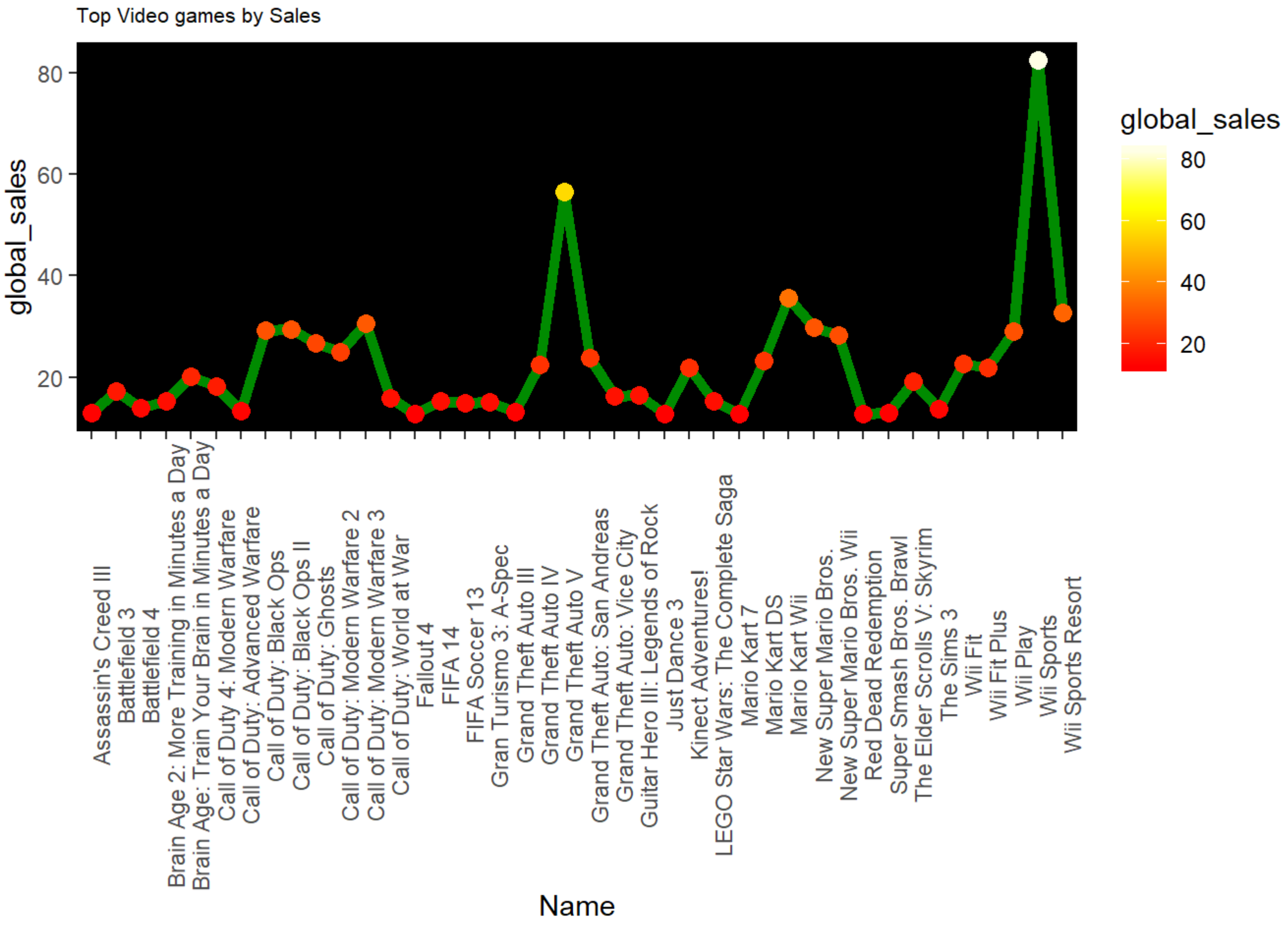


4. Yearly Sales

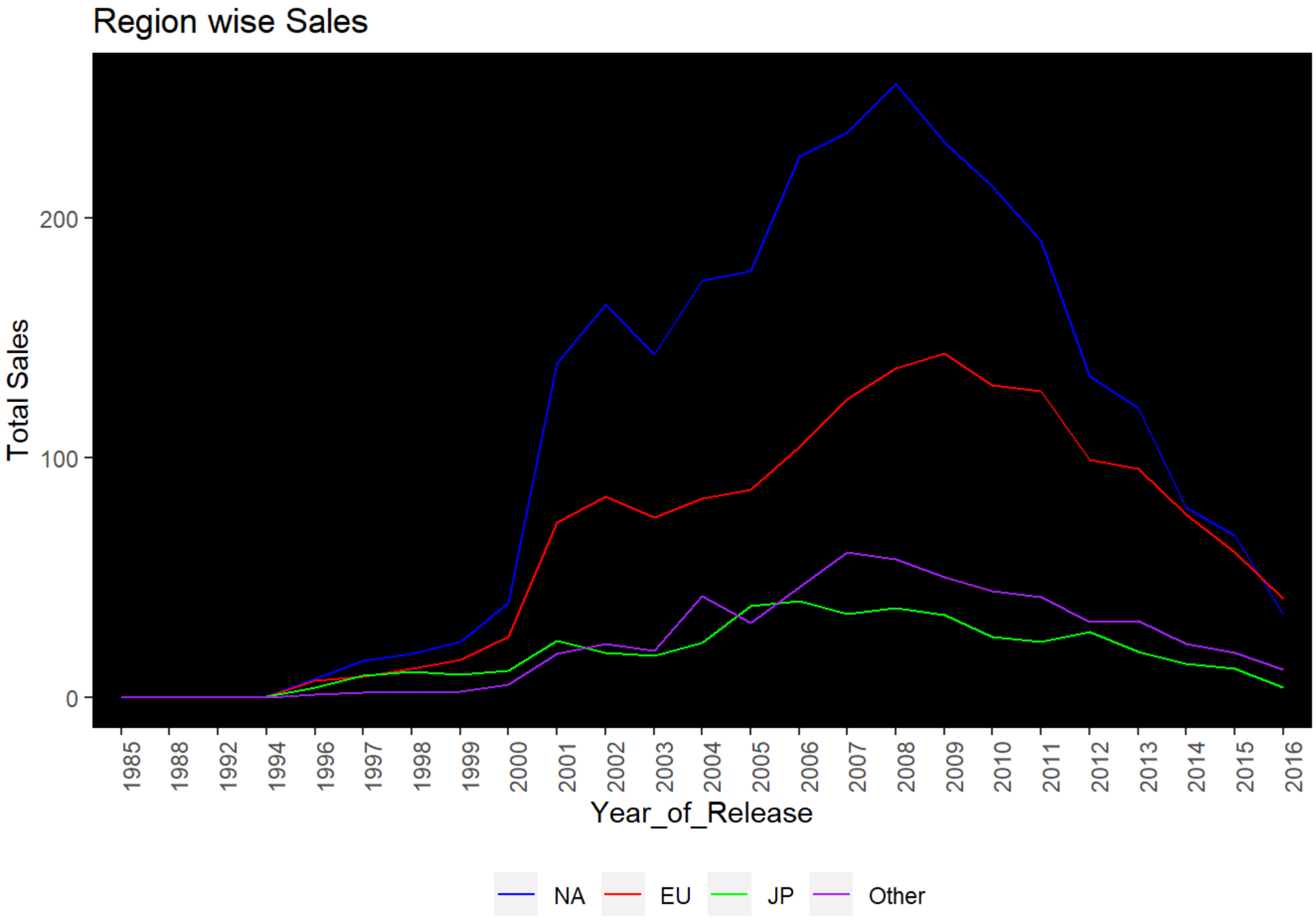


Video game Sales raises from 1980 to 2000 and it attains its peak at 2008, after wards it gradually decreases. The reason might be due to mobile phones , where all games available in mobile as Apps.

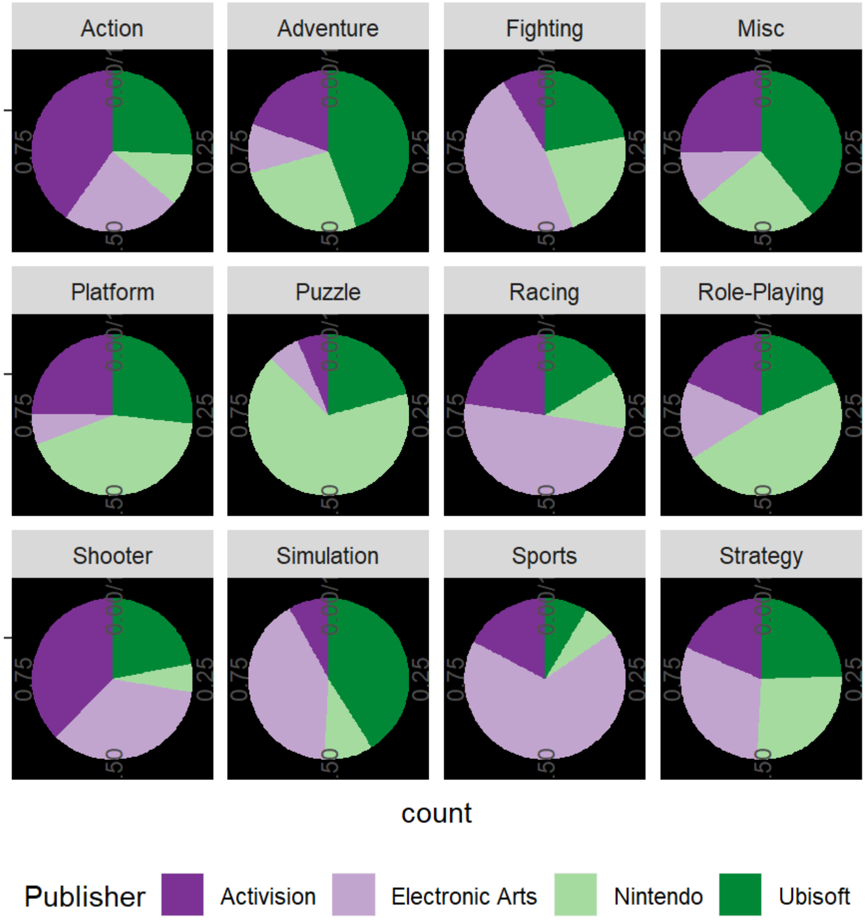
5. Top Video Games and Sales



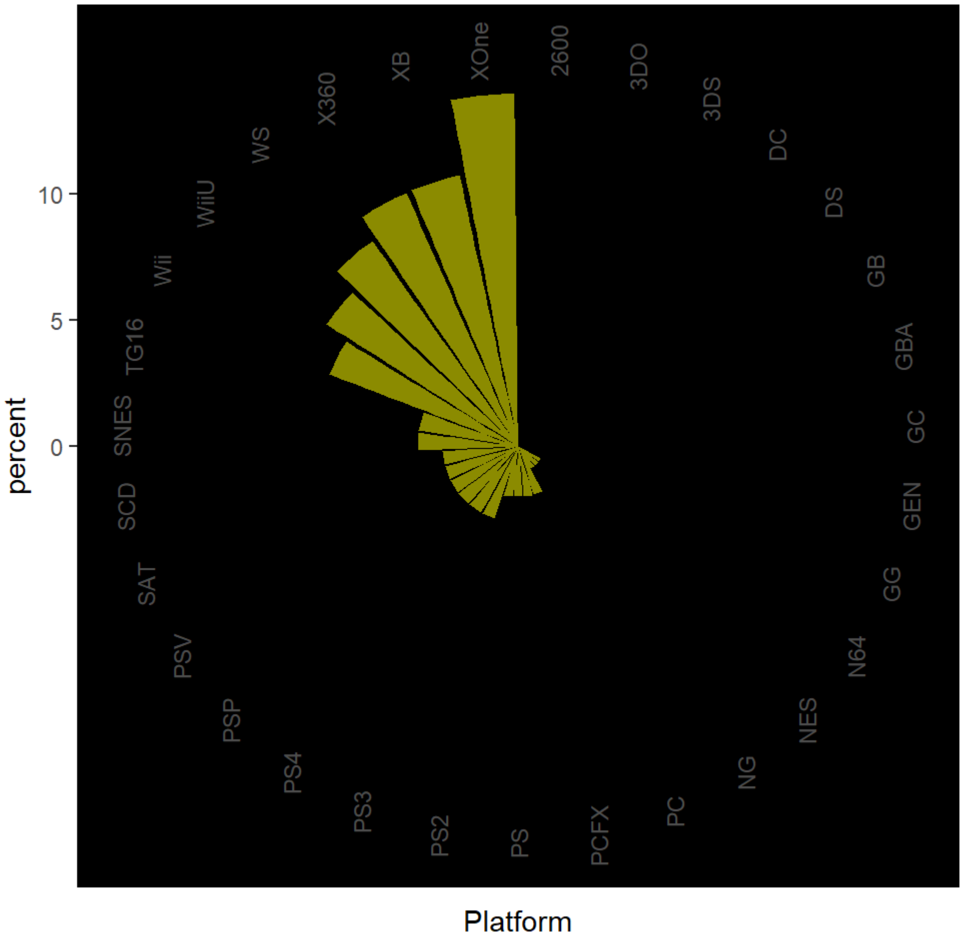
6. Regionwise sales


Region wise Sales
Total Sales
200
100
0
1985
1988
1992
1994
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
Year_of_Release
NA
EU
JP
Other

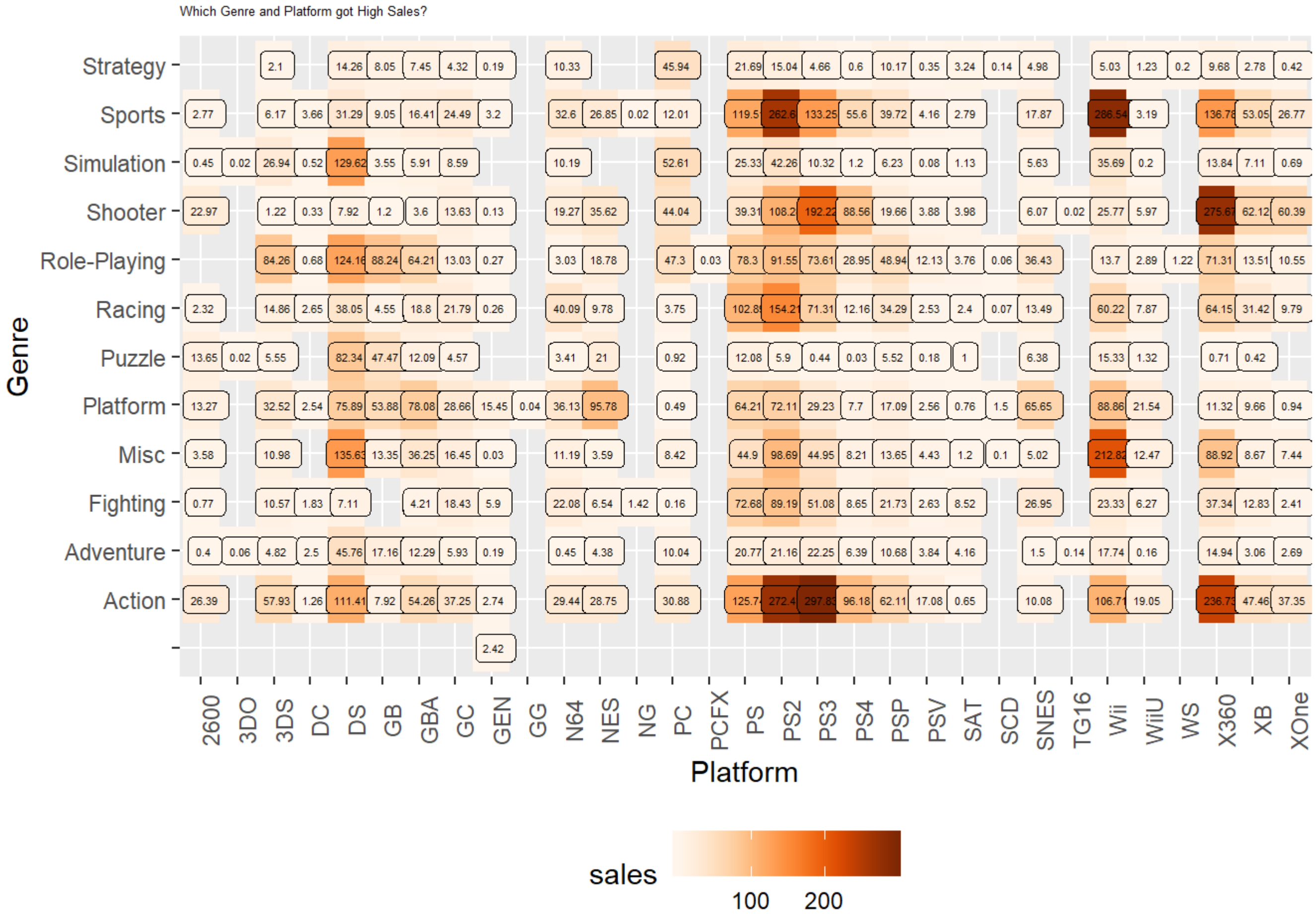
7. Video games Genres and its top publishers



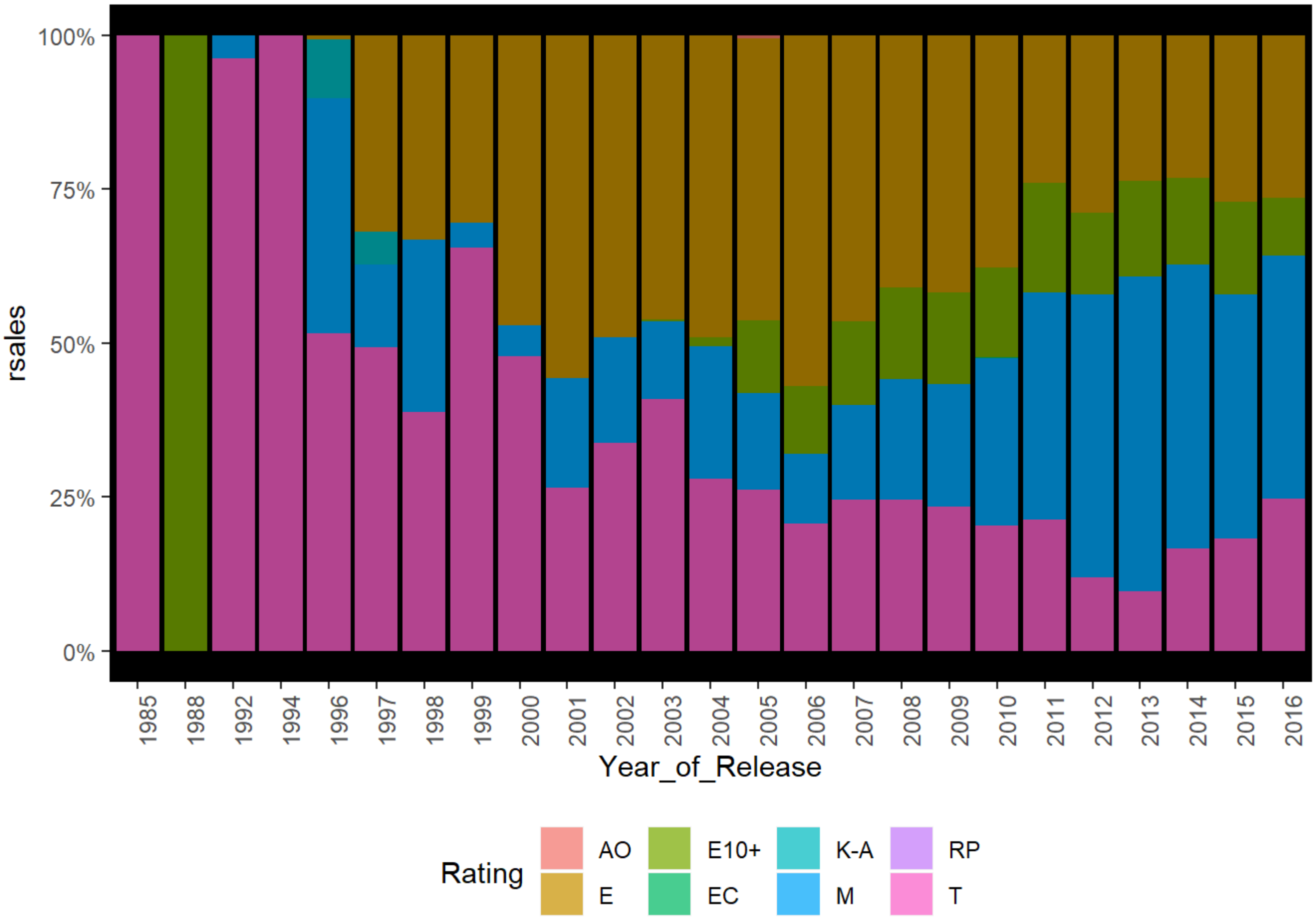
8. Popular Game Platform



9. Game genre and its Platform



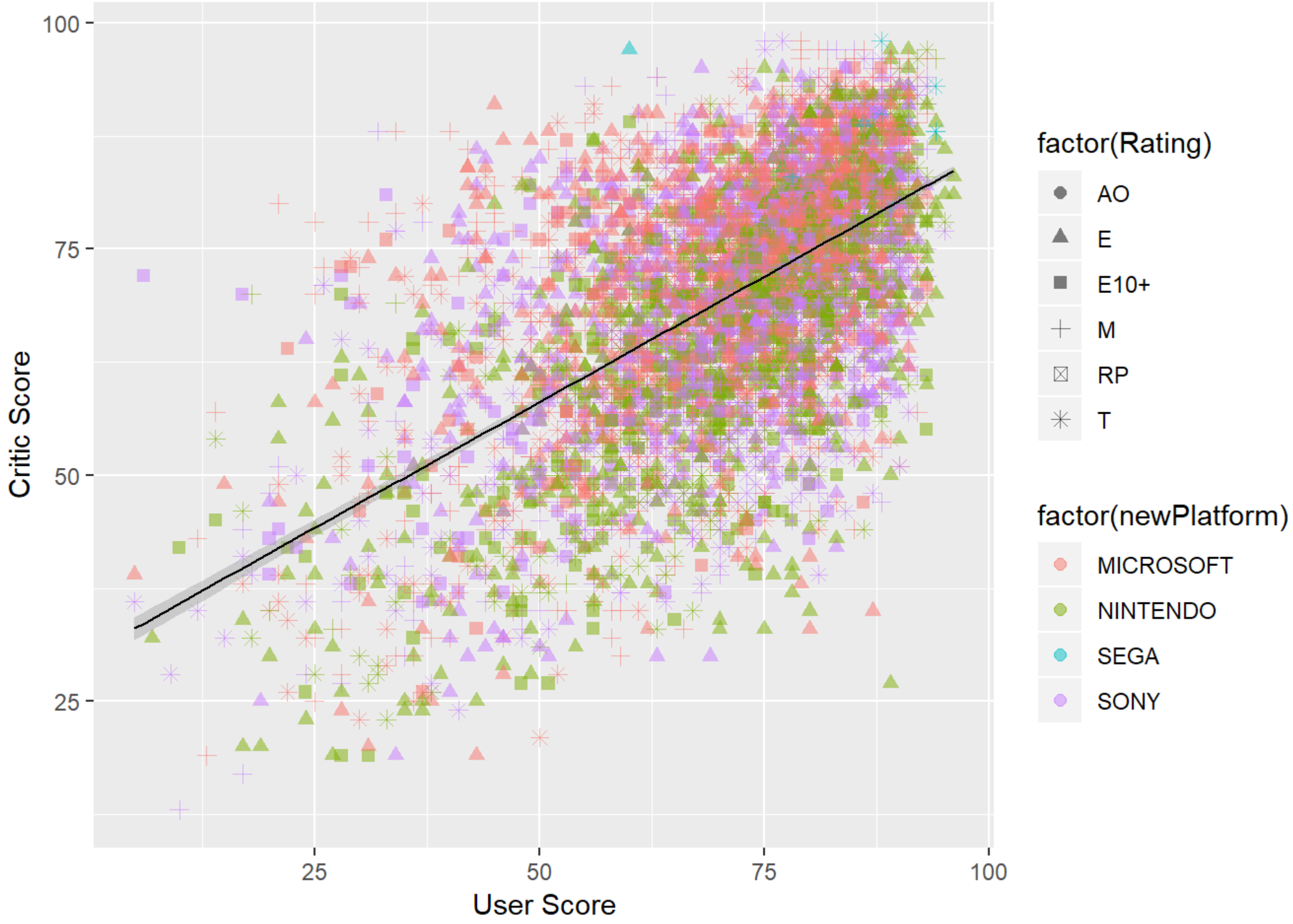
10. Rating and Sales



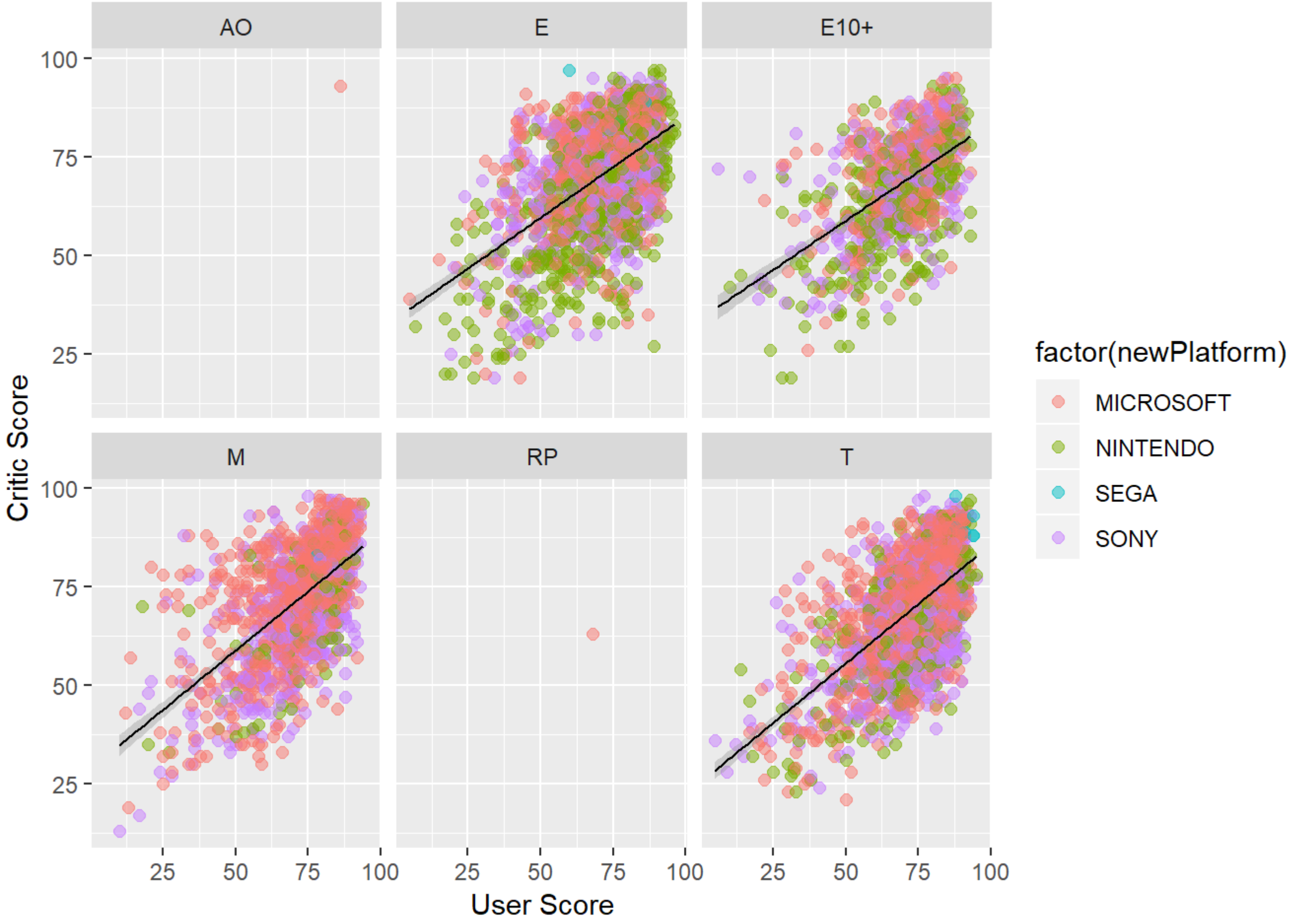
2.4 Modeling and Prediction

Modeling Scores Rescaling User Score and using data cleaned of NA’s.

```
#Modelling Scores / sales

#select years for which Critic was on
games3<-filter(games2,Year_of_Release>=1999)
#rescale the User_Score
games3$User_Score_num = as.numeric(as.character(games3$User_Score)) *10
#quick plot
ggplot(data=games3,aes(x=User_Score_num,y=Critic_Score)) + geom_point(aes(color=factor(newPlatfo
rm), shape=factor(Rating)),size=2,alpha=.5) + geom_smooth(method = "lm", size=.5,color="black",
 formula = y ~ x)+labs(x="User Score",y="Critic Score")
```

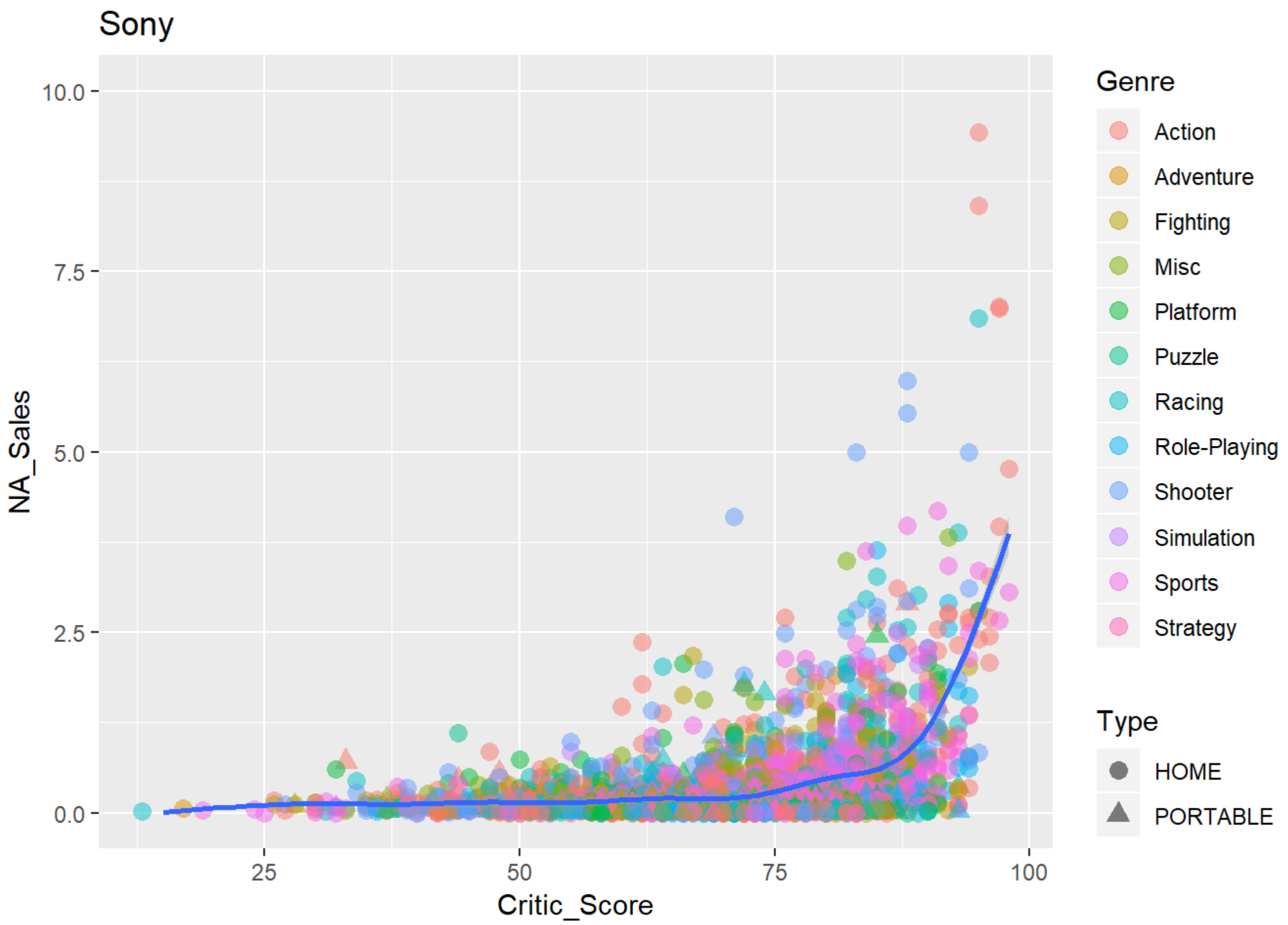


```
#breakdown per Rating
ggplot(data=games3,aes(x=User_Score_num,y=Critic_Score)) + geom_point(aes(color=factor(newPlatfo
rm)),size=2,alpha=.5) + geom_smooth(method = "lm", size=.5,color="black", formula = y ~ x) + fac
et_wrap(~Rating)+labs(x="User Score",y="Critic Score")
```

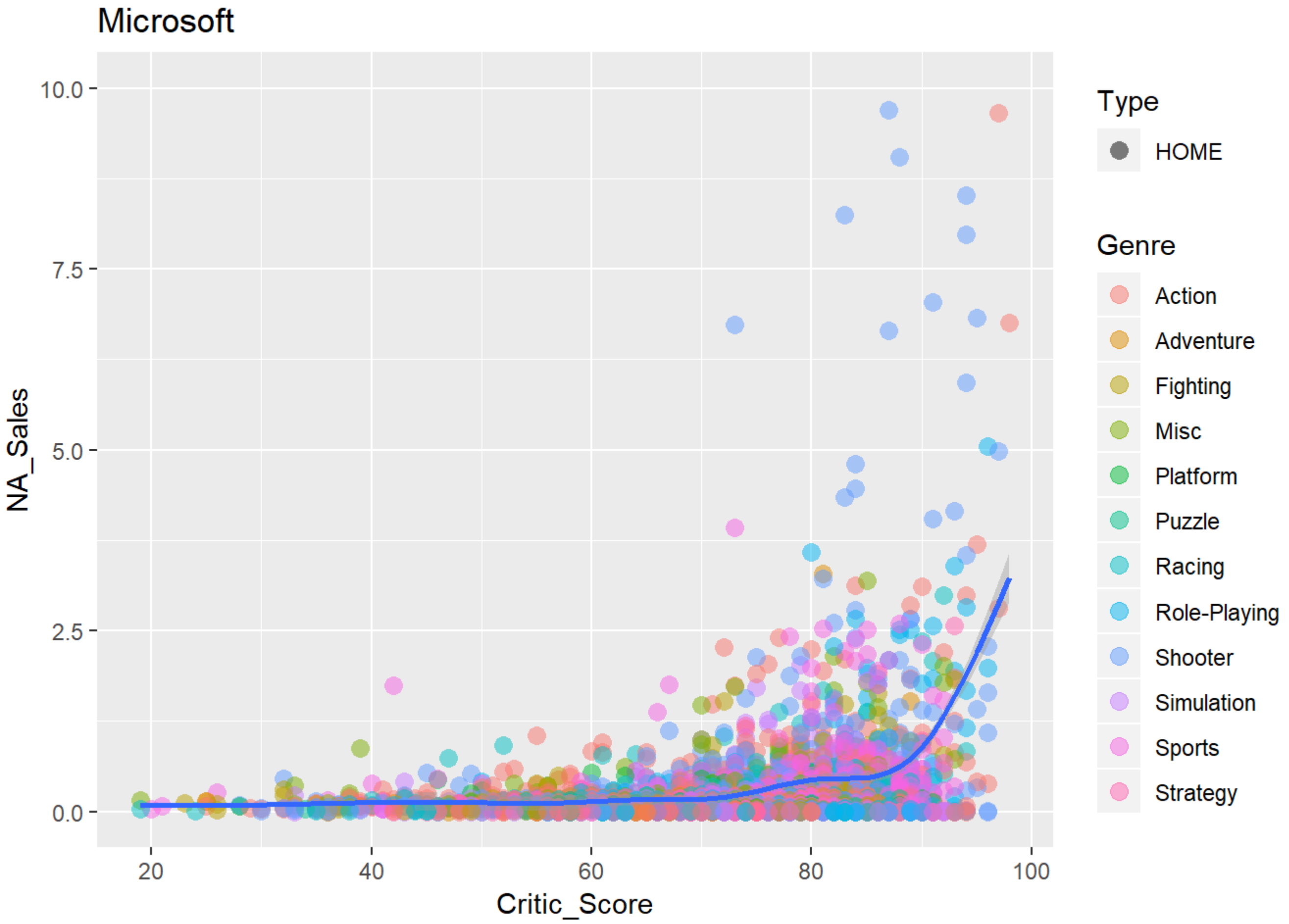


Comments : We have already seen during data analysis that there is correlation between User score and critic score. This section confirms it. Let us see the correlation between score and sales for top platforms.

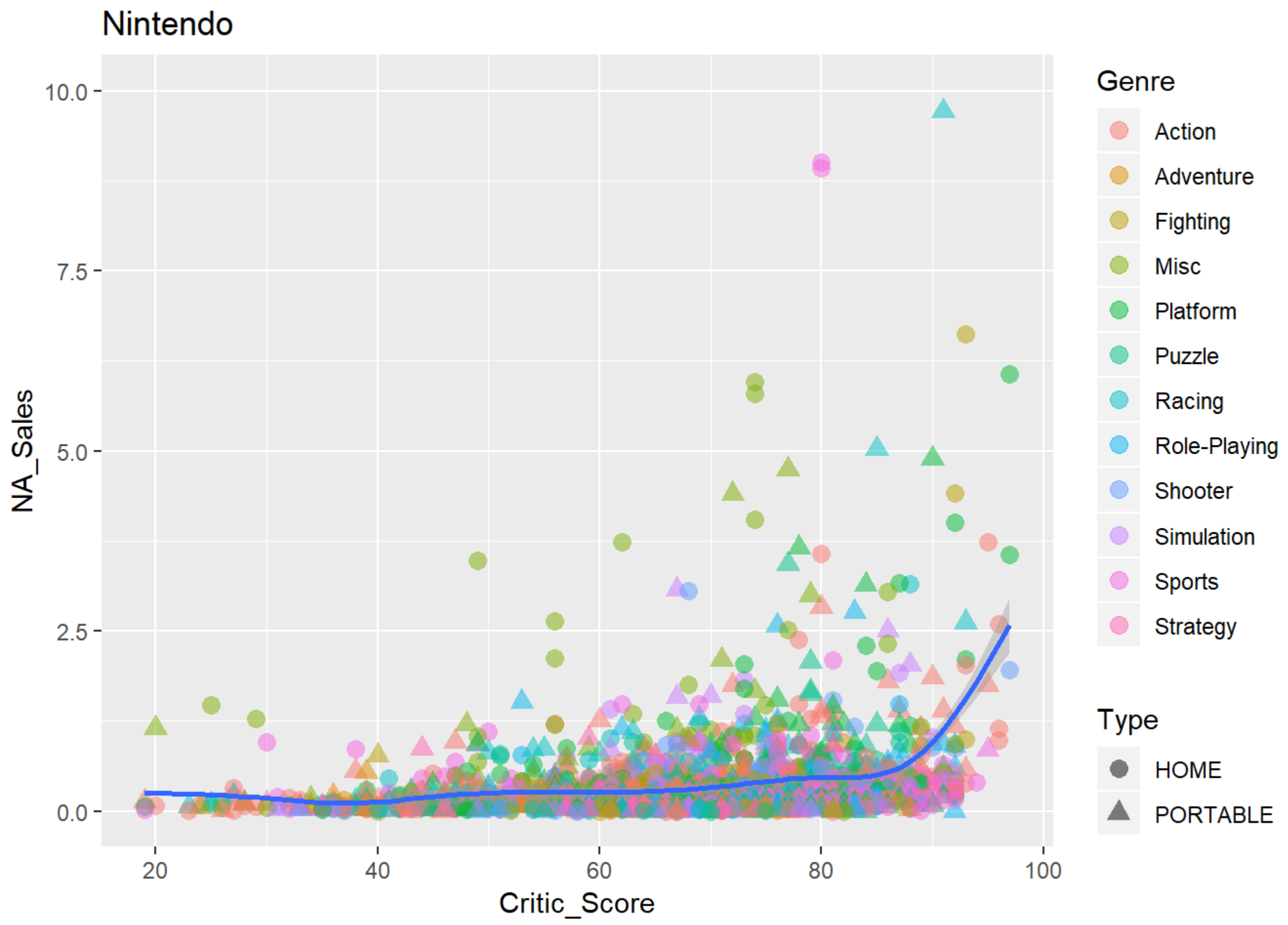
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



We can see that a higher critic score, apparently independently of the genre and type of console, leads to higher sales. It’s somehow expected since players look at reviews before buying a new game.

3. Results

Let us now try to predict the sales on the basis of scores using a linear regression model.

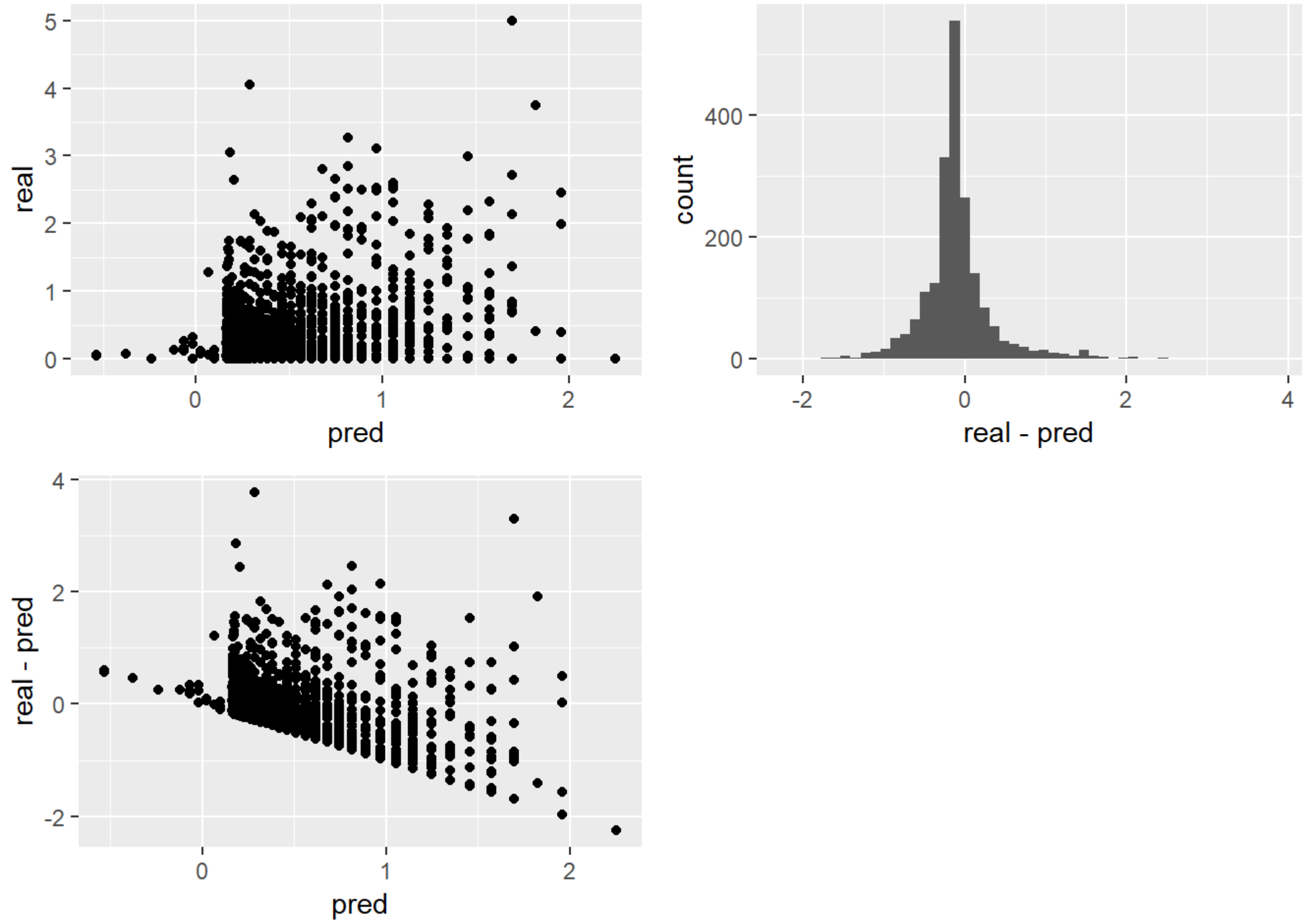
```
# Machine Learning Linear Regression
test_index <- createDataPartition(y = games3$NA_Sales, times = 1,
                                  p = 0.7, list = FALSE)
train_set <- games3[test_index,]
test_set <- games3[-test_index,]

RMSE <- function(true_sales, predicted_sales){

  sqrt(mean((true_sales - predicted_sales)^2, na.rm=T))
}
mu <- mean(train_set$NA_Sales)

#Calculating critic averages
Critic_avgs <- train_set %>%
  group_by(Critic_Score) %>%
  summarize(b_i = mean(NA_Sales - mu))

#Calculating user averages
user_avgs <- train_set %>%
  left_join(Critic_avgs, by='Critic_Score') %>%
  group_by(User_Score) %>%
  summarize(b_u = mean(NA_Sales - mu - b_i))

#Predicting ratings on test set
predicted_sales <- test_set %>%
  left_join(Critic_avgs, by='Critic_Score') %>%
  left_join(user_avgs, by='User_Score') %>%
  mutate(pred = mu + b_i + b_u) %>%
  .$pred

model_rmse <- RMSE(test_set$NA_Sales, predicted_sales)
rmse_results <- data_frame(method="Critic + User Effects Model on test set",
                           RMSE = model_rmse )
rmse_results %>% knitr::kable()
```

method	RMSE
Critic + User Effects Model on test set	0.6953414

The linear regression model results in higher RMSE so let us move to polynomial regression considering the shape of sales vs score correlation.

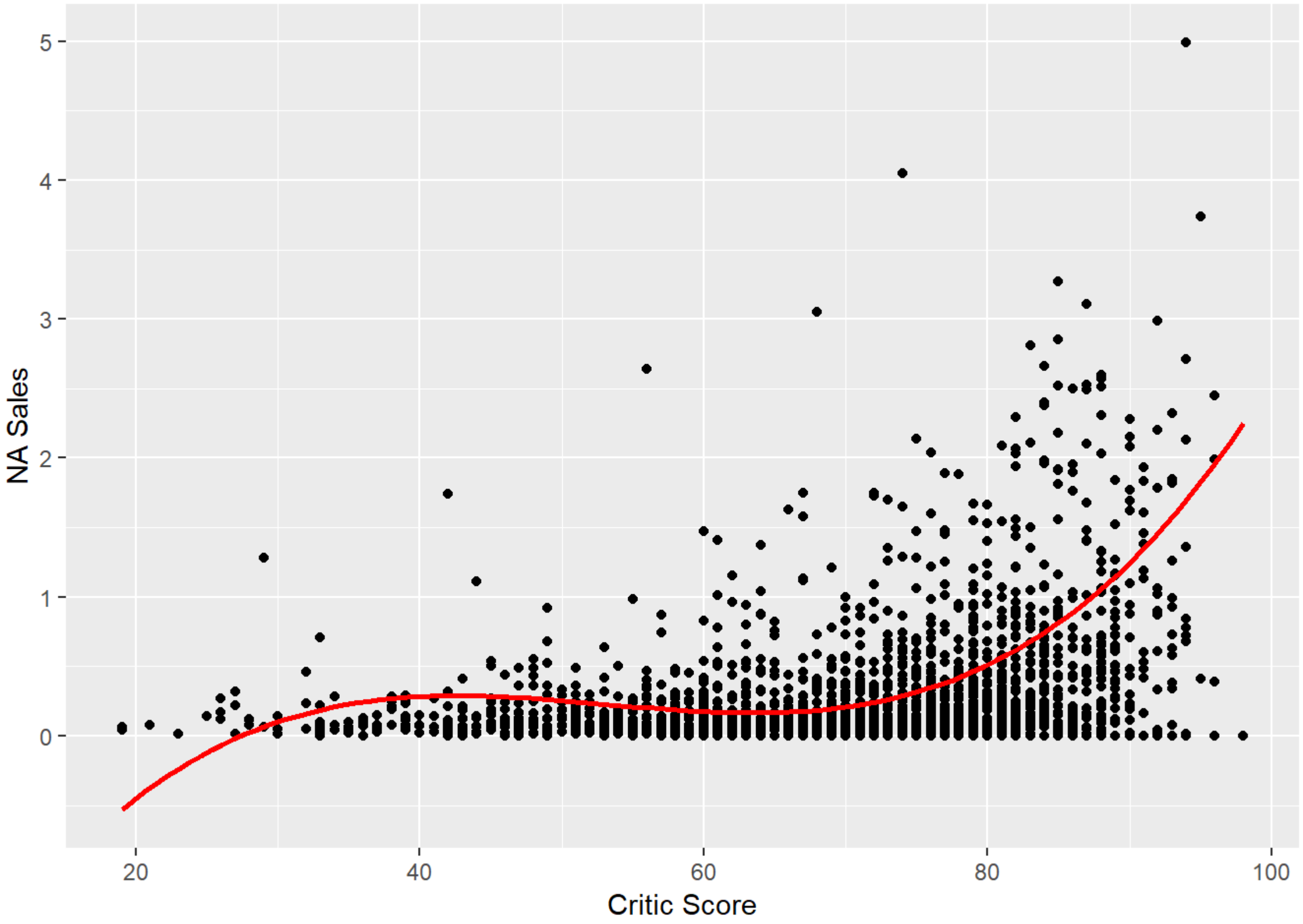
```
#Polynomial regression (1st test)
#define a function to plot the summary of a model(MSE, RMSE, R^2, plots)
plotRes<-function(mod){
  print(mod)
  summary(mod)
  #create DF with prediction and real values
  mod.predictions <- predict(mod,test)
  mod.res<- cbind(mod.predictions,test$NA_Sales)
  colnames(mod.res) <- c('pred','real')
  mod.res <- as.data.frame(mod.res)
  #make plots of residuals,etc...
  g1<-ggplot(data=mod.res,aes(x=pred,y=real)) + geom_point()
  g2<-ggplot(data=mod.res,aes(x=real-pred)) + geom_histogram(bins=50)
  g3<-ggplot(data=mod.res,aes(x=pred,y=real-pred)) + geom_point()
  grid.arrange(g1,g2,g3,nrow=2, ncol=2)
  #calculate metrics
  mse <- mean((mod.res$real-mod.res$pred)^2)
  rmse<-mse^0.5
  SSE = sum((mod.res$pred - mod.res$real)^2)
  SST = sum( (mean(test$NA_Sales) - mod.res$real)^2)
  R2 = 1 - SSE/SST
  sprintf("MSE: %f RMSE : %f R2 :%f", mse,rmse,R2)
}
#Data Partition
set.seed(101)
split<-sample.split(games3$NA_Sales,SplitRatio=.7)
train<-subset(games3,split==T)
test<-subset(games3,split==F)
#Polynomial degree 3 model
model<-lm(NA_Sales ~ Critic_Score + I(Critic_Score^2) + I(Critic_Score^3),train)
plotRes(model)
```

```
##
## Call:
## lm(formula = NA_Sales ~ Critic_Score + I(Critic_Score^2) + I(Critic_Score^3),
##     data = train)
##
## Coefficients:
##       (Intercept)       Critic_Score  I(Critic_Score^2)
##        -3.313e+00          2.175e-01         -4.261e-03
## I(Critic_Score^3)
##         2.674e-05
```



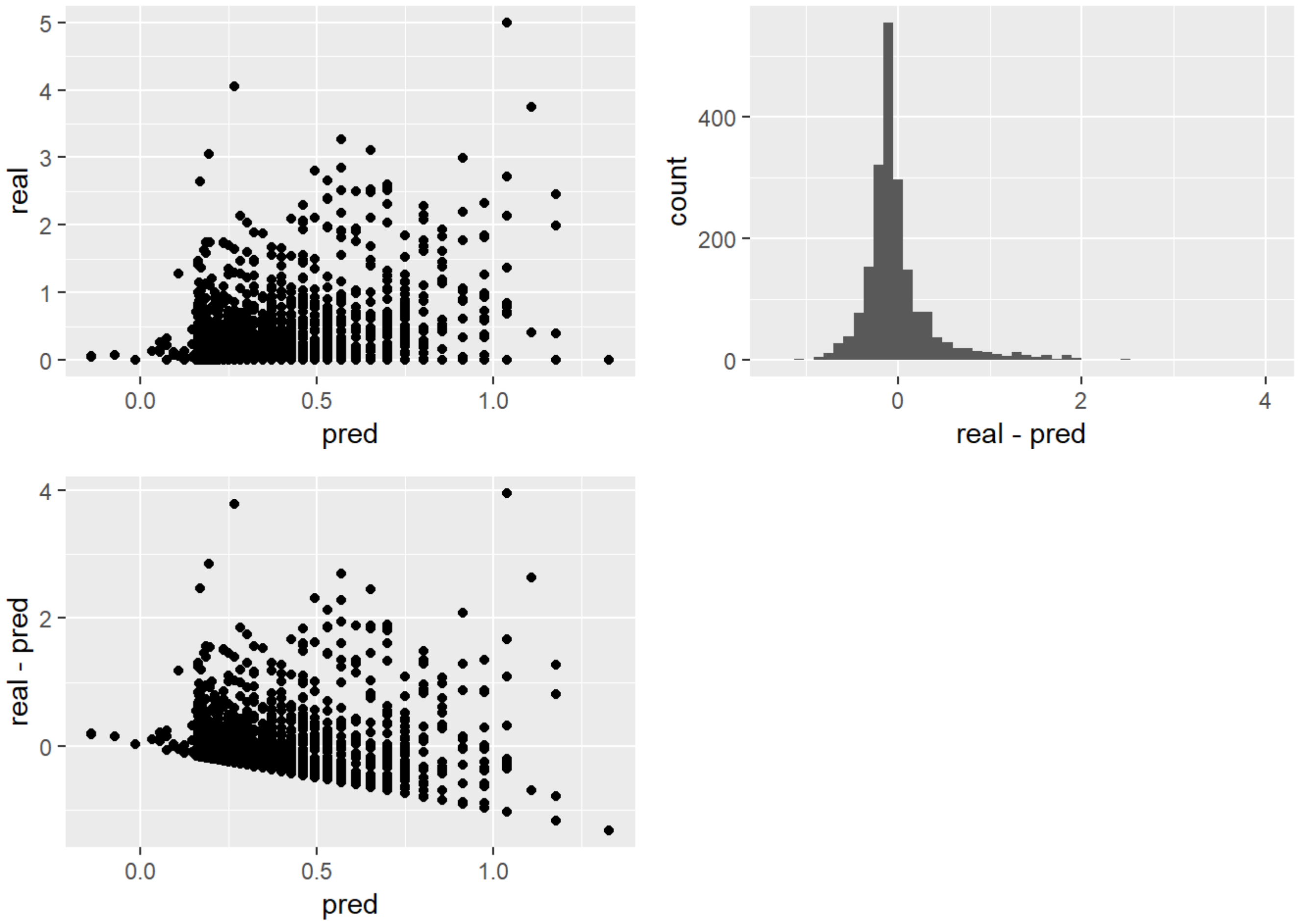
```
## [1] "MSE: 0.221176 RMSE : 0.470294 R2 :0.048036"
```

```
#Test sample
testFunc <- function(x) {model$coefficients[1] + x*model$coefficients[2] + x*x*model$coefficient
s[3] + x*x*x*model$coefficients[4]}
ggplot(data=test,aes(x=Critic_Score,y=NA_Sales)) + geom_point() + stat_function(fun=testFunc,col
or='red',size=1) + xlab("Critic Score") + ylab('NA Sales')
```



```
model_test<-lm(NA_Sales ~ Critic_Score + I(Critic_Score^2) + I(Critic_Score^3),test)
plotRes(model_test)
```

```
##
## Call:
## lm(formula = NA_Sales ~ Critic_Score + I(Critic_Score^2) + I(Critic_Score^3),
##     data = test)
##
## Coefficients:
##       (Intercept)       Critic_Score  I(Critic_Score^2)
##        -1.332e+00          9.441e-02         -1.903e-03
## I(Critic_Score^3)
##         1.241e-05
```



```
## [1] "MSE: 0.193233 RMSE : 0.439583 R2 :0.168307"
```

4. Conclusion