Roli Srivastava

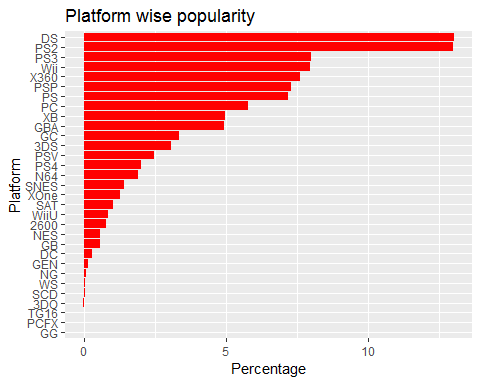
September 29, 2018

## Loading data and library files

Mode <- function(x) {  
 if (is.numeric(x)) {  
 x\_table <- table(x)  
 return(as.numeric(names(x\_table)[which.max(x\_table)]))  
 }  
}

* **Note**
  + The dataset file (*vgsales.csv*)and RMarkdown code/script need to co-exist in same directory for successful execution of this script.
  + DataSet Source: <https://www.kaggle.com/kedokedokedo/vgsales>

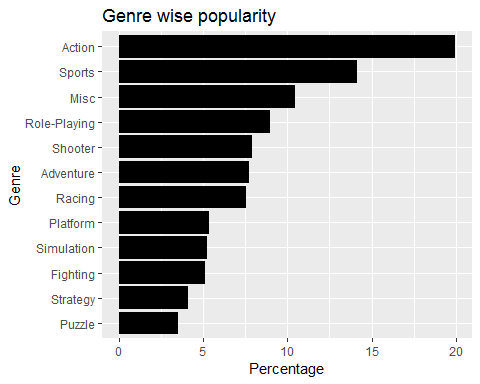
# Histogram to show the most popular platform  
Platform <- data.frame(table(vgsale$Platform))  
Platform$Percentage\_Change <- (Platform$Freq/nrow(vgsale))\*100  
  
  
ggplot(Platform, aes(x = reorder(Platform$Var1, Platform$Percentage\_Change), y = Platform$Percentage\_Change))+  
 labs(title = "Platform wise popularity" , x= "Platform",y= "Percentage")+  
 geom\_bar(stat="identity" , fill = "red")+  
 coord\_flip()



## Genre wise Analysis  
summary(vgsale$Genre)

## Action Adventure Fighting Misc Platform   
## 3316 1286 848 1739 886   
## Puzzle Racing Role-Playing Shooter Simulation   
## 582 1249 1488 1310 867   
## Sports Strategy   
## 2346 681

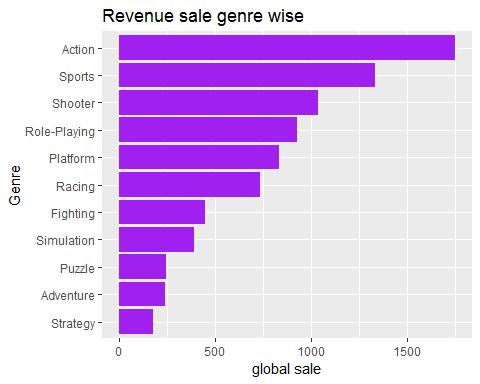
# Histogram to show the most popular genre  
Genre <- data.frame(table(vgsale$Genre))  
Genre$Percentage\_Change <- (Genre$Freq/nrow(vgsale))\*100  
  
ggplot(Genre, aes(x = reorder(Genre$Var1, Genre$Percentage\_Change), y = Genre$Percentage\_Change))+  
 labs(title = "Genre wise popularity" , x= "Genre",y= "Percentage")+  
 geom\_bar(stat="identity" , fill = "black")+  
 coord\_flip()



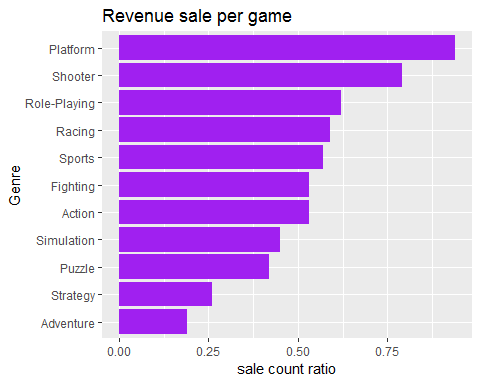
## Sale to count ratio > revenue sale per game  
best\_genre <- vgsale %>% subset(Genre != "Misc") %>% select(Genre ,Name, Global\_Sales) %>% group\_by(Genre) %>%  
summarise(count = n() , sales = sum(Global\_Sales)) %>% arrange(desc(count))  
best\_genre$sales\_count\_Ratio <- round(best\_genre$sales / best\_genre$count , 2)  
kable(best\_genre)

|  |  |  |  |
| --- | --- | --- | --- |
| Genre | count | sales | sales\_count\_Ratio |
| Action | 3316 | 1751.18 | 0.53 |
| Sports | 2346 | 1330.93 | 0.57 |
| Role-Playing | 1488 | 927.37 | 0.62 |
| Shooter | 1310 | 1037.37 | 0.79 |
| Adventure | 1286 | 239.04 | 0.19 |
| Racing | 1249 | 732.04 | 0.59 |
| Platform | 886 | 831.37 | 0.94 |
| Simulation | 867 | 392.20 | 0.45 |
| Fighting | 848 | 448.91 | 0.53 |
| Strategy | 681 | 175.12 | 0.26 |
| Puzzle | 582 | 244.95 | 0.42 |

## revenue sale Genre wise  
ggplot(best\_genre, aes(x = reorder(best\_genre$Genre,best\_genre$sales), y = best\_genre$sales))+  
 labs(title = "Revenue sale genre wise" , x= "Genre",y= "global sale")+  
 geom\_bar(stat="identity" , fill = "purple")+  
 coord\_flip()



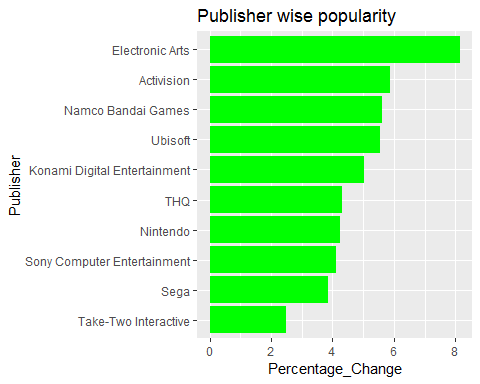
## revenue sale genre wise is Action , sports , shooter , role-playing , platform  
  
## revenue sale per game  
ggplot(best\_genre, aes(x = reorder(best\_genre$Genre,best\_genre$sales\_count\_Ratio), y = best\_genre$sales\_count\_Ratio))+  
 labs(title = "Revenue sale per game" , x= "Genre",y= "sale count ratio")+  
 geom\_bar(stat="identity" , fill = "purple")+  
 coord\_flip()



# revenue sale per game is Platform , shooter , role-playing , racing , sports  
  
# Histogram to show the most popular top 10 publisher of the games  
Publisher <- data.frame(table(vgsale$Publisher))  
Publisher <- subset(Publisher , Publisher$Freq >=400)  
kable(Publisher)

|  |  |  |
| --- | --- | --- |
|  | Var1 | Freq |
| 17 | Activision | 975 |
| 139 | Electronic Arts | 1351 |
| 282 | Konami Digital Entertainment | 832 |
| 352 | Namco Bandai Games | 932 |
| 369 | Nintendo | 703 |
| 453 | Sega | 639 |
| 465 | Sony Computer Entertainment | 683 |
| 499 | Take-Two Interactive | 413 |
| 515 | THQ | 715 |
| 533 | Ubisoft | 921 |

Publisher$Percentage\_Change <- (Publisher$Freq/nrow(vgsale))\*100  
ggplot(Publisher, aes(x = reorder(Publisher$Var1, Publisher$Percentage\_Change), y = Publisher$Percentage\_Change))+  
 labs(title = "Publisher wise popularity" , x= "Publisher",y= "Percentage\_Change")+  
 geom\_bar(stat="identity" , fill = "green")+  
 coord\_flip()



gameshare <- group\_by(vgsale,vgsale$Year)  
gameshare <- summarize(gameshare,World=sum(Global\_Sales),NA.=sum(NA\_Sales),EU=sum(EU\_Sales),JP=sum(JP\_Sales),Other=sum(Other\_Sales))  
  
  
  
cor(gameshare$NA.,gameshare$World)

## [1] 0.992268

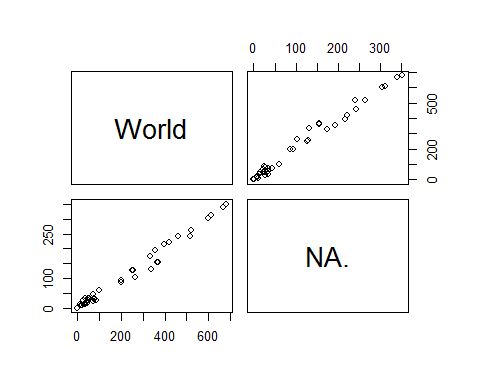
cor(gameshare$EU,gameshare$World)

## [1] 0.9860041

cor(gameshare$JP,gameshare$World)

## [1] 0.8502794

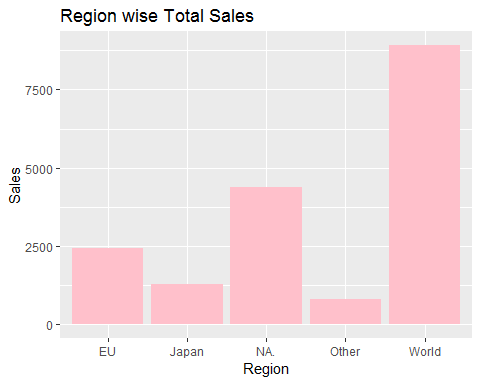
#positive correlation between NA , EU , JP sale of games vs global sales of game , when NA , EU ,JP sale increases so is global sale of games  
pairs(gameshare[,2:3])



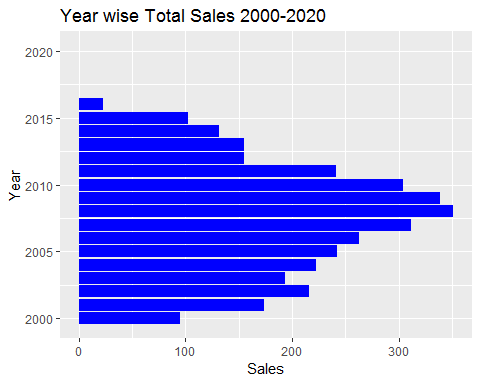
## Region wise Total Sales  
x <- c("World" ,"NA.","EU","Japan","Other")  
y <- c(sum(gameshare$World) , sum(gameshare$NA.), sum(gameshare$EU), sum(gameshare$JP), sum(gameshare$Other))  
gameV <- data.frame(x,y)  
  
kable(gameV)

|  |  |
| --- | --- |
| x | y |
| World | 8920.44 |
| NA. | 4392.95 |
| EU | 2434.13 |
| Japan | 1291.02 |
| Other | 797.75 |

ggplot(gameV , aes(x,y))+  
 labs(title = "Region wise Total Sales" , x= "Region",y= "Sales")+  
 geom\_bar(stat="identity" , fill = "pink")

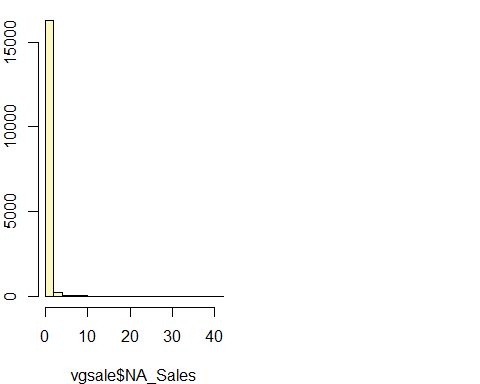


## year-region wise sale from year 2000 - 2020  
yearwise <- vgsale  
yearwise <- within(yearwise, {  
 YearSale <- as.numeric(as.vector.factor(yearwise$Year))})  
 yearwise <- subset(yearwise ,yearwise$YearSale >= 2000)  
ggplot(yearwise , aes(yearwise$YearSale,yearwise$NA\_Sales))+  
 labs(title = "Year wise Total Sales 2000-2020" , x= "Year",y= "Sales")+  
 geom\_bar(stat="identity" , fill = "blue")+  
 coord\_flip()



## 1.Calculate a 95% confidence interval for the average NA Sales and interpret it in context.  
  
inference(y = vgsale$NA\_Sales, est = "mean", type = "ci", null = 0, conflevel = 0.95 ,  
 alternative = "twosided", method = "theoretical")

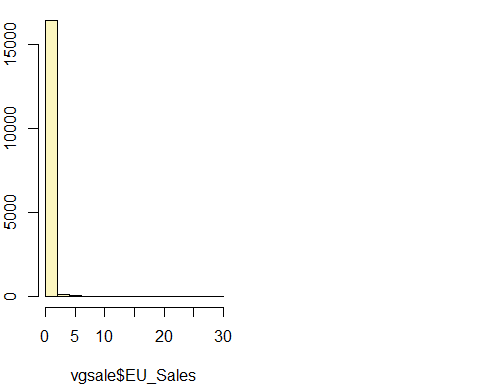
## Single mean   
## Summary statistics:



## mean = 0.2647 ; sd = 0.8167 ; n = 16598   
## Standard error = 0.0063   
## 95 % Confidence interval = ( 0.2522 , 0.2771 )

## 1.Calculate a 95% confidence interval for the average Europe Sales and interpret it in context.  
  
inference(y = vgsale$EU\_Sales, est = "mean", type = "ci", null = 0, conflevel = 0.95 ,  
 alternative = "twosided", method = "theoretical")

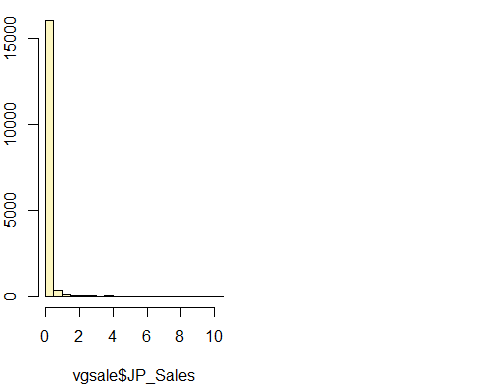
## Single mean   
## Summary statistics:



## mean = 0.1467 ; sd = 0.5054 ; n = 16598   
## Standard error = 0.0039   
## 95 % Confidence interval = ( 0.139 , 0.1543 )

## 1.Calculate a 95% confidence interval for the average Japan Sales and interpret it in context.  
  
inference(y = vgsale$JP\_Sales, est = "mean", type = "ci", null = 0, conflevel = 0.95 ,  
 alternative = "twosided", method = "theoretical")

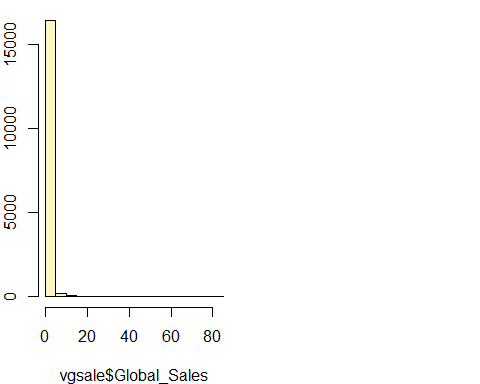
## Single mean   
## Summary statistics:



## mean = 0.0778 ; sd = 0.3093 ; n = 16598   
## Standard error = 0.0024   
## 95 % Confidence interval = ( 0.0731 , 0.0825 )

## 1.Calculate a 95% confidence interval for the average Global Sales and interpret it in context.  
  
inference(y = vgsale$Global\_Sales, est = "mean", type = "ci", null = 0, conflevel = 0.95 ,  
 alternative = "twosided", method = "theoretical")

## Single mean   
## Summary statistics:

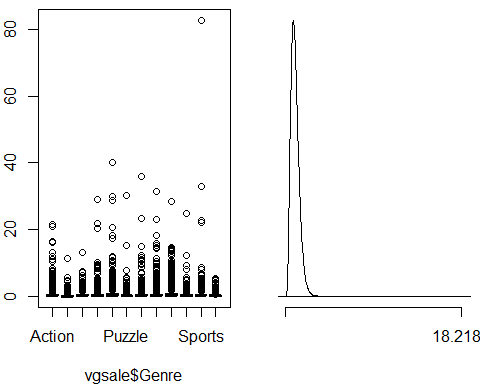


## mean = 0.5374 ; sd = 1.555 ; n = 16598   
## Standard error = 0.0121   
## 95 % Confidence interval = ( 0.5138 , 0.5611 )

inference(y = vgsale$Global\_Sales, x = vgsale$Genre, est = "mean", type = "ht", null = 0, alternative = "greater", method = "theoretical")

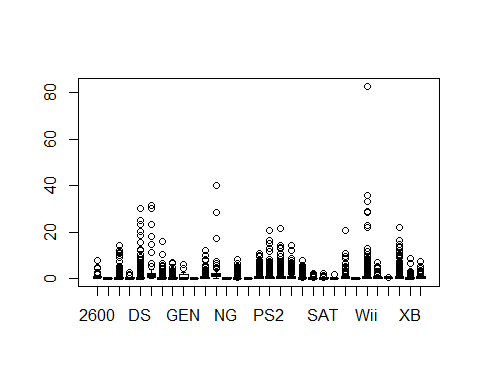
## Response variable: numerical, Explanatory variable: categorical  
## ANOVA  
##   
## Summary statistics:  
## n\_Action = 3316, mean\_Action = 0.5281, sd\_Action = 1.1564  
## n\_Adventure = 1286, mean\_Adventure = 0.1859, sd\_Adventure = 0.5133  
## n\_Fighting = 848, mean\_Fighting = 0.5294, sd\_Fighting = 0.956  
## n\_Misc = 1739, mean\_Misc = 0.4658, sd\_Misc = 1.3149  
## n\_Platform = 886, mean\_Platform = 0.9383, sd\_Platform = 2.5853  
## n\_Puzzle = 582, mean\_Puzzle = 0.4209, sd\_Puzzle = 1.5617  
## n\_Racing = 1249, mean\_Racing = 0.5861, sd\_Racing = 1.6624  
## n\_Role-Playing = 1488, mean\_Role-Playing = 0.6232, sd\_Role-Playing = 1.7079  
## n\_Shooter = 1310, mean\_Shooter = 0.7919, sd\_Shooter = 1.8173  
## n\_Simulation = 867, mean\_Simulation = 0.4524, sd\_Simulation = 1.1953  
## n\_Sports = 2346, mean\_Sports = 0.5673, sd\_Sports = 2.0897  
## n\_Strategy = 681, mean\_Strategy = 0.2572, sd\_Strategy = 0.5209

## H\_0: All means are equal.  
## H\_A: At least one mean is different.  
## Analysis of Variance Table  
##   
## Response: y  
## Df Sum Sq Mean Sq F value Pr(>F)  
## x 11 479 43.557 18.218 < 2.2e-16  
## Residuals 16586 39654 2.391   
##   
## Pairwise tests: t tests with pooled SD   
## Action Adventure Fighting Misc Platform Puzzle Racing  
## Adventure 0.0000 NA NA NA NA NA NA  
## Fighting 0.9829 0.0000 NA NA NA NA NA  
## Misc 0.1733 0.0000 0.3260 NA NA NA NA  
## Platform 0.0000 0.0000 0.0000 0.0000 NA NA NA  
## Puzzle 0.1229 0.0024 0.1924 0.5444 0.0000 NA NA  
## Racing 0.2585 0.0000 0.4097 0.0359 0.0000 0.0333 NA  
## Role-Playing 0.0486 0.0000 0.1583 0.0039 0.0000 0.0074 0.5315  
## Shooter 0.0000 0.0000 0.0001 0.0000 0.0295 0.0000 0.0008  
## Simulation 0.1991 0.0001 0.3025 0.8349 0.0000 0.7039 0.0504  
## Sports 0.3471 0.0000 0.5403 0.0379 0.0000 0.0409 0.7288  
## Strategy 0.0000 0.3308 0.0006 0.0028 0.0000 0.0607 0.0000  
## Role-Playing Shooter Simulation Sports  
## Adventure NA NA NA NA  
## Fighting NA NA NA NA  
## Misc NA NA NA NA  
## Platform NA NA NA NA  
## Puzzle NA NA NA NA  
## Racing NA NA NA NA  
## Role-Playing NA NA NA NA  
## Shooter 0.0040 NA NA NA  
## Simulation 0.0097 0 NA NA  
## Sports 0.2752 0 0.0614 NA  
## Strategy 0.0000 0 0.0137 0



## Descriptive analysis for global sale

|  |  |
| --- | --- |
|  | A |
| mean | 5.374407e-01 |
| mode | 2.000000e-02 |
| median | 1.700000e-01 |
| sd | 1.555028e+00 |
| var | 2.418112e+00 |
| kurtosis | 6.067501e+02 |
| skewness | 1.739907e+01 |
| max | 8.274000e+01 |
| min | 1.000000e-02 |
| range | 8.273000e+01 |
| sum | 8.920440e+03 |
| Count | 1.659800e+04 |



## Linear equation for global sale on Genre wise

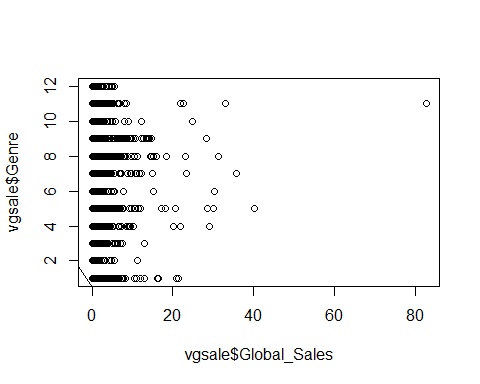
m1 <- lm(Global\_Sales ~ Genre, data = vgsale)  
summary(m1)

##   
## Call:  
## lm(formula = Global\_Sales ~ Genre, data = vgsale)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.928 -0.458 -0.307 -0.037 82.173   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.528100 0.026851 19.668 < 2e-16 \*\*\*  
## GenreAdventure -0.342221 0.050795 -6.737 1.67e-11 \*\*\*  
## GenreFighting 0.001275 0.059501 0.021 0.9829   
## GenreMisc -0.062338 0.045780 -1.362 0.1733   
## GenrePlatform 0.410241 0.058476 7.016 2.38e-12 \*\*\*  
## GenrePuzzle -0.107224 0.069491 -1.543 0.1229   
## GenreRacing 0.058001 0.051334 1.130 0.2585   
## GenreRole-Playing 0.095132 0.048247 1.972 0.0486 \*   
## GenreShooter 0.263785 0.050458 5.228 1.74e-07 \*\*\*  
## GenreSimulation -0.075736 0.058980 -1.284 0.1991   
## GenreSports 0.039219 0.041715 0.940 0.3471   
## GenreStrategy -0.270949 0.065052 -4.165 3.13e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.546 on 16586 degrees of freedom  
## Multiple R-squared: 0.01194, Adjusted R-squared: 0.01128   
## F-statistic: 18.22 on 11 and 16586 DF, p-value: < 2.2e-16

sum(m1$residuals^2)

## [1] 39654.28

plot(vgsale$Global\_Sales ,vgsale$Genre)  
abline(m1)



## R-squared: 0.01194   
   
## Y = Global\_Sales , X = Genre -(X0 = Action = 0 , Xa = Adventure , Xb =Fighting , Xc = Misc , Xd = Platform , xe = Puzzle , Xf = Racing , Xg = Role-Playing, Xh = Shooter , Xi = Simulation , Xj = Sports , Xk = Strategy )  
## Y = 0.528100 - 0.342221 \* Xa + 0.001275 \* Xb - 0.062338 \* Xc + 0.410241 \* Xd - 0.107224 \* Xe + 0.058001 \* Xf + 0.095132 \* Xg + 0.263785 \* Xh - 0.075736 \* Xi+ 0.039219 \* Xj - 0.270949 \* Xk  
## So Breakdown is 0 and running late is 1   
## for Action genre - Y(Global sale) = 0.528100  
## for Adventure genre- Y(Global sale) = 0.528100 - 0.342221 \* 1 = 0.185879  
## so on and so forth  
  
m1 <- lm(Global\_Sales ~ Platform, data = vgsale)  
m1 <- lm(Global\_Sales ~ Publisher, data = vgsale)

## Market-wise Regression, ANNOVA & Prediction Analysis

### For North America (NA) Market

First, we build a multiple linear regression model.

#Subsetting the sales data to create a data frame with just NA sales and relevant independant variables  
naSales <- salesData[, c("Platform", "Genre", "NA\_Sales")]  
#Studying the first few rows of the data frame  
head(naSales, 10)

## Platform Genre NA\_Sales  
## 1 Wii Sports 41.49  
## 2 NES Platform 29.08  
## 3 Wii Racing 15.85  
## 4 Wii Sports 15.75  
## 5 GB Role-Playing 11.27  
## 6 GB Puzzle 23.20  
## 7 DS Platform 11.38  
## 8 Wii Misc 14.03  
## 9 Wii Platform 14.59  
## 10 NES Shooter 26.93

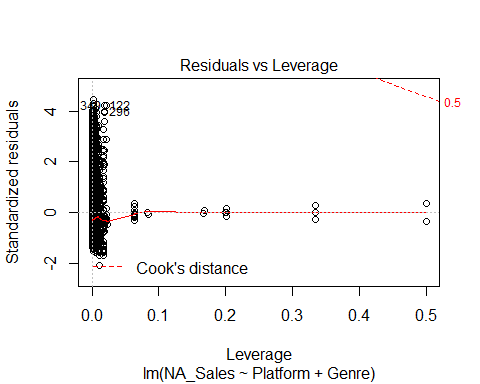
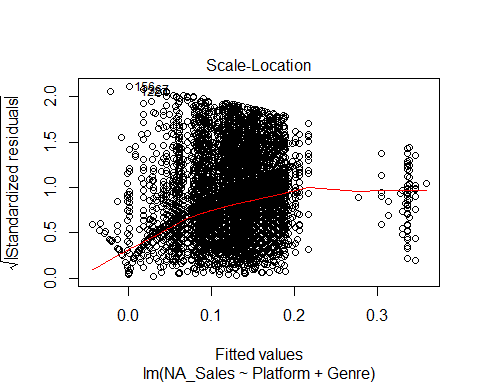
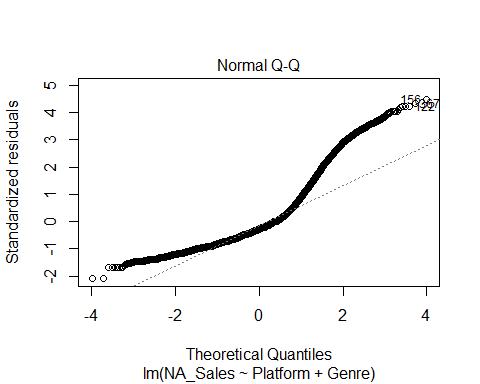
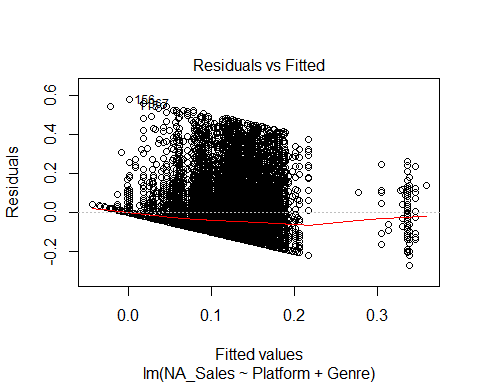
#Removing outliers from the codewith the 1.5 times the Inter Quartile range (IQR) rule  
naSalesNoOutlier <- filter(naSales, NA\_Sales <= (quantile(naSales$NA\_Sales, 0.75) + (1.5 \* IQR(naSales$NA\_Sales))), NA\_Sales >= (quantile(naSales$NA\_Sales, 0.25) - (1.5 \* IQR(naSales$NA\_Sales))))  
#Creating a multiple regression model  
modelNA <- lm(NA\_Sales ~ Platform + Genre, data = naSalesNoOutlier)  
#Creating an ANOVA analysis  
modelNA.aov <- aov(NA\_Sales ~ Platform + Genre, data = naSales)

Next comes displaying the results of the regression analysis.

#Displaying the output of the multiple linear regression  
summary(modelNA)

##   
## Call:  
## lm(formula = NA\_Sales ~ Platform + Genre, data = naSalesNoOutlier)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.26858 -0.08706 -0.03603 0.04182 0.57950   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.336223 0.013767 24.422 < 2e-16 \*\*\*  
## Platform3DO -0.304844 0.076348 -3.993 6.56e-05 \*\*\*  
## Platform3DS -0.251937 0.014954 -16.848 < 2e-16 \*\*\*  
## PlatformDC -0.299547 0.023316 -12.847 < 2e-16 \*\*\*  
## PlatformDS -0.216489 0.014077 -15.379 < 2e-16 \*\*\*  
## PlatformGB -0.259749 0.021863 -11.881 < 2e-16 \*\*\*  
## PlatformGBA -0.199058 0.014553 -13.678 < 2e-16 \*\*\*  
## PlatformGC -0.193729 0.014935 -12.972 < 2e-16 \*\*\*  
## PlatformGEN -0.321410 0.035308 -9.103 < 2e-16 \*\*\*  
## PlatformGG -0.358456 0.130814 -2.740 0.006147 \*\*   
## PlatformN64 -0.152877 0.015943 -9.589 < 2e-16 \*\*\*  
## PlatformNES -0.141451 0.021463 -6.590 4.53e-11 \*\*\*  
## PlatformNG -0.328541 0.040219 -8.169 3.37e-16 \*\*\*  
## PlatformPC -0.290804 0.014445 -20.132 < 2e-16 \*\*\*  
## PlatformPCFX -0.308773 0.130772 -2.361 0.018231 \*   
## PlatformPS -0.206102 0.014339 -14.373 < 2e-16 \*\*\*  
## PlatformPS2 -0.211650 0.014082 -15.030 < 2e-16 \*\*\*  
## PlatformPS3 -0.184531 0.014247 -12.952 < 2e-16 \*\*\*  
## PlatformPS4 -0.222920 0.015724 -14.177 < 2e-16 \*\*\*  
## PlatformPSP -0.249167 0.014281 -17.448 < 2e-16 \*\*\*  
## PlatformPSV -0.278826 0.015184 -18.363 < 2e-16 \*\*\*  
## PlatformSAT -0.312850 0.016991 -18.413 < 2e-16 \*\*\*  
## PlatformSCD -0.311085 0.059777 -5.204 1.97e-07 \*\*\*  
## PlatformSNES -0.300598 0.016429 -18.297 < 2e-16 \*\*\*  
## PlatformTG16 -0.310995 0.092978 -3.345 0.000825 \*\*\*  
## PlatformWii -0.169213 0.014271 -11.857 < 2e-16 \*\*\*  
## PlatformWiiU -0.191357 0.017840 -10.726 < 2e-16 \*\*\*  
## PlatformWS -0.302950 0.054912 -5.517 3.51e-08 \*\*\*  
## PlatformX360 -0.157560 0.014322 -11.001 < 2e-16 \*\*\*  
## PlatformXB -0.189942 0.014546 -13.058 < 2e-16 \*\*\*  
## PlatformXOne -0.184250 0.016931 -10.882 < 2e-16 \*\*\*  
## GenreAdventure -0.059299 0.004443 -13.348 < 2e-16 \*\*\*  
## GenreFighting -0.008595 0.005436 -1.581 0.113895   
## GenreMisc -0.023704 0.004106 -5.773 7.95e-09 \*\*\*  
## GenrePlatform 0.022233 0.005492 4.048 5.18e-05 \*\*\*  
## GenrePuzzle -0.031555 0.006199 -5.090 3.62e-07 \*\*\*  
## GenreRacing -0.005913 0.004648 -1.272 0.203271   
## GenreRole-Playing -0.027450 0.004313 -6.365 2.01e-10 \*\*\*  
## GenreShooter 0.008843 0.004680 1.889 0.058876 .   
## GenreSimulation -0.003285 0.005304 -0.619 0.535693   
## GenreSports 0.002357 0.003782 0.623 0.533226   
## GenreStrategy -0.044919 0.005737 -7.829 5.24e-15 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.13 on 14875 degrees of freedom  
## Multiple R-squared: 0.1307, Adjusted R-squared: 0.1283   
## F-statistic: 54.53 on 41 and 14875 DF, p-value: < 2.2e-16

#Displaying the reidual plots  
plot(modelNA)



#Displaying results of ANOVA analysis  
summary(modelNA.aov)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Platform 30 423 14.113 22.16 <2e-16 \*\*\*  
## Genre 11 104 9.414 14.78 <2e-16 \*\*\*  
## Residuals 16556 10543 0.637   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Prediction a NA sales (in million units) for a particular platform and genre.

predict(modelNA, data.frame(Platform = "PS3", Genre = "Shooter"))

## 1   
## 0.1605351

### For European (EU) Market

First, we build a multiple linear regression model.

#Subsetting the sales data to create a data frame with just EU sales and relevant independant variables  
euSales <- salesData[, c("Platform", "Genre", "EU\_Sales")]  
#Studying the first few rows of the data frame  
head(euSales, 10)

## Platform Genre EU\_Sales  
## 1 Wii Sports 29.02  
## 2 NES Platform 3.58  
## 3 Wii Racing 12.88  
## 4 Wii Sports 11.01  
## 5 GB Role-Playing 8.89  
## 6 GB Puzzle 2.26  
## 7 DS Platform 9.23  
## 8 Wii Misc 9.20  
## 9 Wii Platform 7.06  
## 10 NES Shooter 0.63

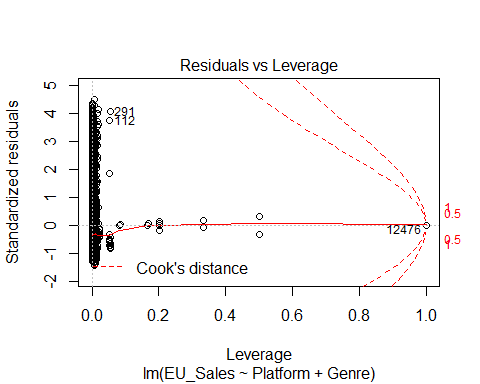
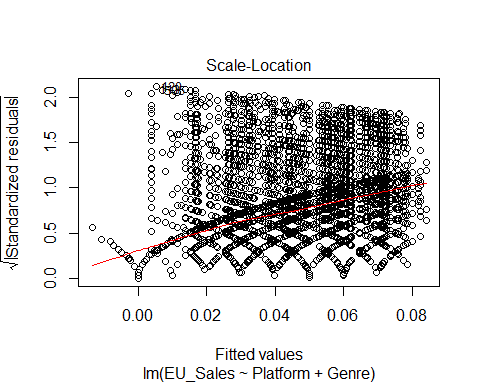
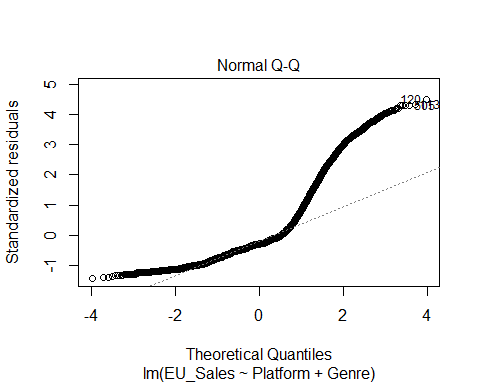
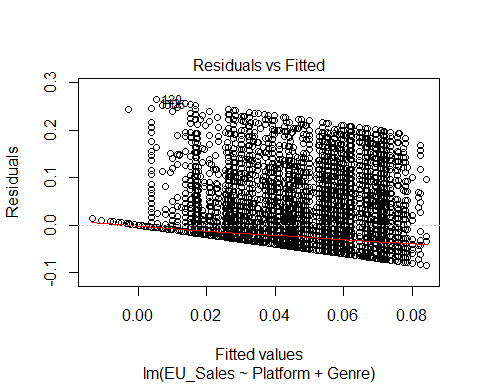
#Removing outliers from the codewith the 1.5 times the Inter Quartile range (IQR) rule  
euSalesNoOutlier <- filter(euSales, EU\_Sales <= (quantile(euSales$EU\_Sales, 0.75) + (1.5 \* IQR(euSales$EU\_Sales))), EU\_Sales >= (quantile(euSales$EU\_Sales, 0.25) - (1.5 \* IQR(euSales$EU\_Sales))))  
#Creating a multiple regression model  
modelEU <- lm(EU\_Sales ~ Platform + Genre, data = euSalesNoOutlier)  
#Creating an ANOVA analysis  
modelEU.aov <- aov(EU\_Sales ~ Platform + Genre, data = euSales)

Next comes displaying the results of the regression analysis.

#Displaying the output of the multiple linear regression  
summary(modelEU)

##   
## Call:  
## lm(formula = EU\_Sales ~ Platform + Genre, data = euSalesNoOutlier)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.08417 -0.03489 -0.01704 0.01078 0.26454   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.0389838 0.0051945 7.505 6.51e-14 \*\*\*  
## Platform3DO -0.0246045 0.0344970 -0.713 0.475711   
## Platform3DS -0.0008953 0.0058471 -0.153 0.878310   
## PlatformDC -0.0193977 0.0098416 -1.971 0.048745 \*   
## PlatformDS -0.0125440 0.0053426 -2.348 0.018893 \*   
## PlatformGB 0.0030542 0.0091302 0.335 0.737997   
## PlatformGBA 0.0167610 0.0055832 3.002 0.002686 \*\*   
## PlatformGC 0.0087954 0.0057653 1.526 0.127140   
## PlatformGEN 0.0023564 0.0145033 0.162 0.870933   
## PlatformGG -0.0450668 0.0593038 -0.760 0.447309   
## PlatformN64 0.0156925 0.0062549 2.509 0.012124 \*   
## PlatformNES 0.0264392 0.0084669 3.123 0.001796 \*\*   
## PlatformNG -0.0342733 0.0179251 -1.912 0.055892 .   
## PlatformPC 0.0148762 0.0055696 2.671 0.007572 \*\*   
## PlatformPCFX -0.0263827 0.0592895 -0.445 0.656340   
## PlatformPS 0.0325602 0.0054943 5.926 3.17e-09 \*\*\*  
## PlatformPS2 0.0274816 0.0053439 5.143 2.74e-07 \*\*\*  
## PlatformPS3 0.0302364 0.0054843 5.513 3.58e-08 \*\*\*  
## PlatformPS4 0.0288744 0.0064000 4.512 6.48e-06 \*\*\*  
## PlatformPSP -0.0110296 0.0054667 -2.018 0.043650 \*   
## PlatformPSV -0.0086958 0.0059685 -1.457 0.145152   
## PlatformSAT -0.0273411 0.0068650 -3.983 6.85e-05 \*\*\*  
## PlatformSCD -0.0296211 0.0269206 -1.100 0.271213   
## PlatformSNES -0.0224297 0.0065404 -3.429 0.000607 \*\*\*  
## PlatformTG16 -0.0300319 0.0420789 -0.714 0.475421   
## PlatformWii 0.0004139 0.0054599 0.076 0.939575   
## PlatformWiiU 0.0367488 0.0074694 4.920 8.75e-07 \*\*\*  
## PlatformWS -0.0250802 0.0246915 -1.016 0.309768   
## PlatformX360 0.0275226 0.0054837 5.019 5.26e-07 \*\*\*  
## PlatformXB 0.0094474 0.0055808 1.693 0.090508 .   
## PlatformXOne 0.0391060 0.0068683 5.694 1.27e-08 \*\*\*  
## GenreAdventure -0.0223704 0.0020458 -10.935 < 2e-16 \*\*\*  
## GenreFighting -0.0045040 0.0024858 -1.812 0.070027 .   
## GenreMisc -0.0091163 0.0018886 -4.827 1.40e-06 \*\*\*  
## GenrePlatform 0.0060829 0.0025217 2.412 0.015866 \*   
## GenrePuzzle -0.0096782 0.0028364 -3.412 0.000646 \*\*\*  
## GenreRacing 0.0005287 0.0021782 0.243 0.808210   
## GenreRole-Playing -0.0126011 0.0019816 -6.359 2.09e-10 \*\*\*  
## GenreShooter 0.0044666 0.0021815 2.047 0.040632 \*   
## GenreSimulation -0.0110893 0.0024440 -4.537 5.74e-06 \*\*\*  
## GenreSports -0.0069818 0.0017405 -4.011 6.07e-05 \*\*\*  
## GenreStrategy -0.0165085 0.0026353 -6.264 3.85e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.05904 on 14475 degrees of freedom  
## Multiple R-squared: 0.1076, Adjusted R-squared: 0.1051   
## F-statistic: 42.59 on 41 and 14475 DF, p-value: < 2.2e-16

#Displaying the reidual plots  
plot(modelEU)



#Displaying results of ANOVA analysis  
summary(modelEU.aov)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Platform 30 99 3.288 13.25 <2e-16 \*\*\*  
## Genre 11 32 2.907 11.71 <2e-16 \*\*\*  
## Residuals 16556 4108 0.248   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Prediction a NA sales (in million units) for a particular platform and genre.

predict(modelNA, data.frame(Platform = "PS4", Genre = "Platform"))

## 1   
## 0.1355363

### For Japanese (JP) Market

First, we build a multiple linear regression model.

#Subsetting the sales data to create a data frame with just JP sales and relevant independant variables  
jpSales <- salesData[, c("Platform", "Genre", "JP\_Sales")]  
#Studying the first few rows of the data frame  
head(jpSales, 10)

## Platform Genre JP\_Sales  
## 1 Wii Sports 3.77  
## 2 NES Platform 6.81  
## 3 Wii Racing 3.79  
## 4 Wii Sports 3.28  
## 5 GB Role-Playing 10.22  
## 6 GB Puzzle 4.22  
## 7 DS Platform 6.50  
## 8 Wii Misc 2.93  
## 9 Wii Platform 4.70  
## 10 NES Shooter 0.28

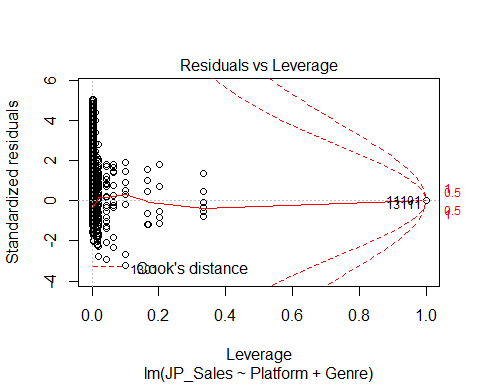
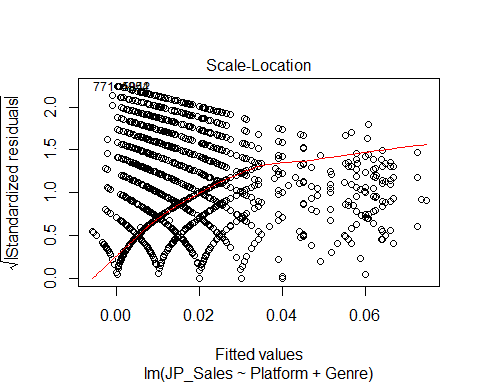
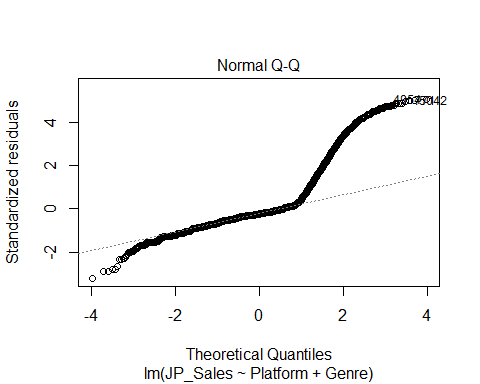
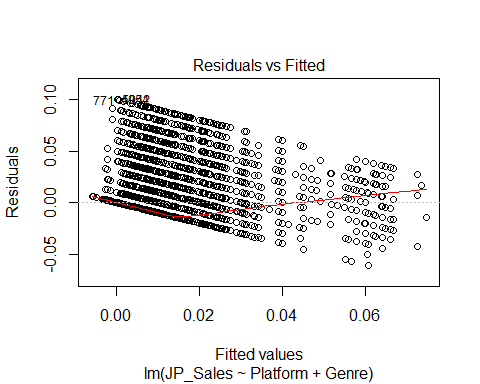
#Removing outliers from the codewith the 1.5 times the Inter Quartile range (IQR) rule  
jpSalesNoOutlier <- filter(jpSales, JP\_Sales <= (quantile(jpSales$JP\_Sales, 0.75) + (1.5 \* IQR(jpSales$JP\_Sales))), JP\_Sales >= (quantile(jpSales$JP\_Sales, 0.25) - (1.5 \* IQR(jpSales$JP\_Sales))))  
#Creating a multiple regression model  
modelJP <- lm(JP\_Sales ~ Platform + Genre, data = jpSalesNoOutlier)  
#Creating an ANOVA analysis  
modelJP.aov <- aov(JP\_Sales ~ Platform + Genre, data = jpSales)

Next comes displaying the results of the regression analysis.

#Displaying the output of the multiple linear regression  
summary(modelJP)

##   
## Call:  
## lm(formula = JP\_Sales ~ Platform + Genre, data = jpSalesNoOutlier)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.060673 -0.009925 -0.004764 0.001604 0.099963   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.0007460 0.0017462 0.427 0.669243   
## Platform3DO 0.0324485 0.0116439 2.787 0.005331 \*\*   
## Platform3DS 0.0247858 0.0020325 12.195 < 2e-16 \*\*\*  
## PlatformDC 0.0591831 0.0044290 13.363 < 2e-16 \*\*\*  
## PlatformDS 0.0099537 0.0018005 5.528 3.29e-08 \*\*\*  
## PlatformGB 0.0630977 0.0065395 9.649 < 2e-16 \*\*\*  
## PlatformGBA 0.0034794 0.0018870 1.844 0.065229 .   
## PlatformGC 0.0040177 0.0019480 2.062 0.039183 \*   
## PlatformGEN 0.0374398 0.0052811 7.089 1.41e-12 \*\*\*  
## PlatformGG 0.0424247 0.0200174 2.119 0.034075 \*   
## PlatformN64 0.0069558 0.0021410 3.249 0.001161 \*\*   
## PlatformNES 0.0050185 0.0116409 0.431 0.666394   
## PlatformNG 0.0440462 0.0083458 5.278 1.33e-07 \*\*\*  
## PlatformPC -0.0025696 0.0018603 -1.381 0.167211   
## PlatformPCFX 0.0157183 0.0200159 0.785 0.432299   
## PlatformPS 0.0062114 0.0018586 3.342 0.000834 \*\*\*  
## PlatformPS2 0.0083178 0.0017984 4.625 3.78e-06 \*\*\*  
## PlatformPS3 0.0165228 0.0018338 9.010 < 2e-16 \*\*\*  
## PlatformPS4 0.0198389 0.0020831 9.524 < 2e-16 \*\*\*  
## PlatformPSP 0.0170249 0.0018499 9.203 < 2e-16 \*\*\*  
## PlatformPSV 0.0257102 0.0020338 12.641 < 2e-16 \*\*\*  
## PlatformSAT 0.0584386 0.0029675 19.693 < 2e-16 \*\*\*  
## PlatformSCD 0.0604644 0.0090850 6.655 2.93e-11 \*\*\*  
## PlatformSNES 0.0476121 0.0032243 14.766 < 2e-16 \*\*\*  
## PlatformTG16 0.0183929 0.0200106 0.919 0.358028   
## PlatformWii 0.0055398 0.0018281 3.030 0.002447 \*\*   
## PlatformWiiU 0.0117914 0.0025457 4.632 3.65e-06 \*\*\*  
## PlatformWS 0.0259464 0.0200244 1.296 0.195088   
## PlatformX360 0.0065323 0.0018230 3.583 0.000340 \*\*\*  
## PlatformXB 0.0008754 0.0018691 0.468 0.639541   
## PlatformXOne 0.0003904 0.0022058 0.177 0.859512   
## GenreAdventure 0.0048678 0.0007009 6.945 3.96e-12 \*\*\*  
## GenreFighting 0.0068944 0.0008762 7.869 3.84e-15 \*\*\*  
## GenreMisc -0.0012374 0.0006366 -1.944 0.051926 .   
## GenrePlatform -0.0031707 0.0008415 -3.768 0.000165 \*\*\*  
## GenrePuzzle -0.0041879 0.0009977 -4.198 2.71e-05 \*\*\*  
## GenreRacing -0.0039419 0.0006986 -5.643 1.70e-08 \*\*\*  
## GenreRole-Playing 0.0135358 0.0007532 17.970 < 2e-16 \*\*\*  
## GenreShooter 0.0008611 0.0006894 1.249 0.211655   
## GenreSimulation -0.0002633 0.0008334 -0.316 0.752060   
## GenreSports -0.0032251 0.0005841 -5.522 3.42e-08 \*\*\*  
## GenreStrategy 0.0033077 0.0009556 3.461 0.000539 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01993 on 14131 degrees of freedom  
## Multiple R-squared: 0.2001, Adjusted R-squared: 0.1978   
## F-statistic: 86.24 on 41 and 14131 DF, p-value: < 2.2e-16

#Displaying the reidual plots  
plot(modelJP)



#Displaying results of ANOVA analysis  
summary(modelJP.aov)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Platform 30 219.6 7.319 90.84 <2e-16 \*\*\*  
## Genre 11 34.2 3.112 38.63 <2e-16 \*\*\*  
## Residuals 16556 1333.9 0.081   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Prediction a NA sales (in million units) for a particular platform and genre.

predict(modelJP, data.frame(Platform = "PS3", Genre = "Shooter"))

## 1   
## 0.01812985

## For Global (GB) Market

First, we build a multiple linear regression model.

#Subsetting the sales data to create a data frame with just GB sales and relevant independant variables  
gbSales <- salesData[, c("Platform", "Genre", "Global\_Sales")]  
#Studying the first few rows of the data frame  
head(gbSales, 10)

## Platform Genre Global\_Sales  
## 1 Wii Sports 82.74  
## 2 NES Platform 40.24  
## 3 Wii Racing 35.82  
## 4 Wii Sports 33.00  
## 5 GB Role-Playing 31.37  
## 6 GB Puzzle 30.26  
## 7 DS Platform 30.01  
## 8 Wii Misc 29.02  
## 9 Wii Platform 28.62  
## 10 NES Shooter 28.31

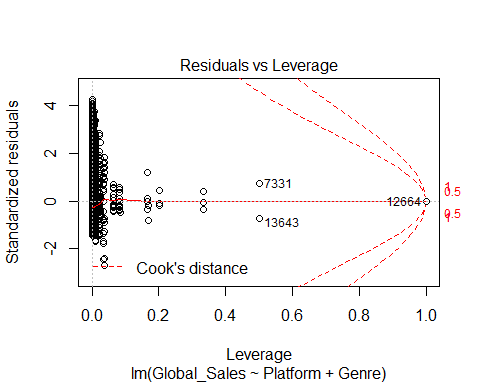
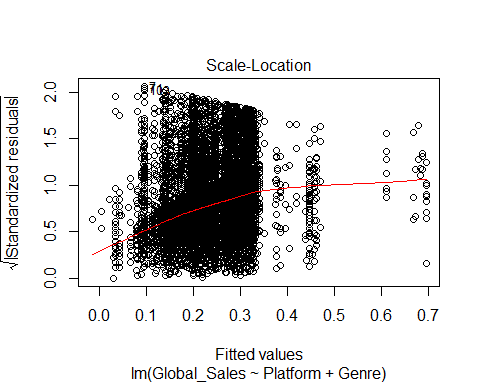
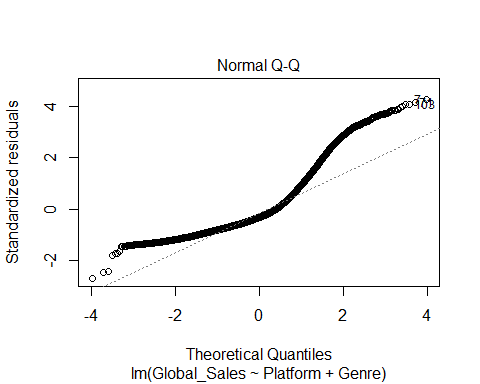
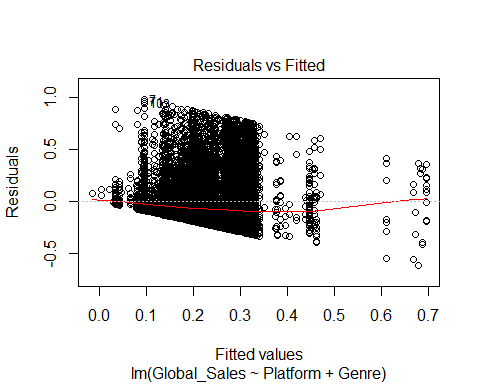
#Removing outliers from the codewith the 1.5 times the Inter Quartile range (IQR) rule  
gbSalesNoOutlier <- filter(gbSales, Global\_Sales <= (quantile(gbSales$Global\_Sales, 0.75) + (1.5 \* IQR(gbSales$Global\_Sales))), Global\_Sales >= (quantile(gbSales$Global\_Sales, 0.25) - (1.5 \* IQR(gbSales$Global\_Sales))))  
#Creating a multiple regression model  
modelGB <- lm(Global\_Sales ~ Platform + Genre, data = gbSalesNoOutlier)  
#Creating an ANOVA analysis  
modelGB.aov <- aov(Global\_Sales ~ Platform + Genre, data = gbSales)

Next comes displaying the results of the regression analysis.

#Displaying the output of the multiple linear regression  
summary(modelGB)

##   
## Call:  
## lm(formula = Global\_Sales ~ Platform + Genre, data = gbSalesNoOutlier)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.60979 -0.15485 -0.07083 0.08405 0.98375   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.446025 0.022207 20.085 < 2e-16 \*\*\*  
## Platform3DO -0.355161 0.135506 -2.621 0.008776 \*\*   
## Platform3DS -0.243866 0.024608 -9.910 < 2e-16 \*\*\*  
## PlatformDC -0.273037 0.040782 -6.695 2.23e-11 \*\*\*  
## PlatformDS -0.243451 0.022792 -10.681 < 2e-16 \*\*\*  
## PlatformGB -0.066051 0.041089 -1.607 0.107967   
## PlatformGBA -0.223253 0.023705 -9.418 < 2e-16 \*\*\*  
## PlatformGC -0.235471 0.024402 -9.649 < 2e-16 \*\*\*  
## PlatformGEN -0.334580 0.061977 -5.398 6.83e-08 \*\*\*  
## PlatformGG -0.430081 0.232616 -1.849 0.064494 .   
## PlatformN64 -0.150616 0.026260 -5.736 9.91e-09 \*\*\*  
## PlatformNES 0.225384 0.048464 4.651 3.34e-06 \*\*\*  
## PlatformNG -0.335011 0.070826 -4.730 2.27e-06 \*\*\*  
## PlatformPC -0.306280 0.023535 -13.014 < 2e-16 \*\*\*  
## PlatformPCFX -0.413293 0.232553 -1.777 0.075556 .   
## PlatformPS -0.148399 0.023339 -6.358 2.10e-10 \*\*\*  
## PlatformPS2 -0.170345 0.022814 -7.467 8.68e-14 \*\*\*  
## PlatformPS3 -0.128678 0.023189 -5.549 2.92e-08 \*\*\*  
## PlatformPS4 -0.200520 0.026251 -7.639 2.33e-14 \*\*\*  
## PlatformPSP -0.259408 0.023213 -11.175 < 2e-16 \*\*\*  
## PlatformPSV -0.296680 0.024988 -11.873 < 2e-16 \*\*\*  
## PlatformSAT -0.246380 0.028450 -8.660 < 2e-16 \*\*\*  
## PlatformSCD -0.349078 0.105894 -3.296 0.000981 \*\*\*  
## PlatformSNES -0.142031 0.027765 -5.116 3.17e-07 \*\*\*  
## PlatformTG16 -0.318622 0.165174 -1.929 0.053748 .   
## PlatformWii -0.173966 0.023162 -7.511 6.21e-14 \*\*\*  
## PlatformWiiU -0.168829 0.030160 -5.598 2.21e-08 \*\*\*  
## PlatformWS -0.190851 0.097196 -1.964 0.049599 \*   
## PlatformX360 -0.132905 0.023240 -5.719 1.09e-08 \*\*\*  
## PlatformXB -0.225683 0.023660 -9.538 < 2e-16 \*\*\*  
## PlatformXOne -0.187696 0.028315 -6.629 3.50e-11 \*\*\*  
## GenreAdventure -0.106326 0.007951 -13.373 < 2e-16 \*\*\*  
## GenreFighting 0.008384 0.009704 0.864 0.387653   
## GenreMisc -0.027719 0.007338 -3.778 0.000159 \*\*\*  
## GenrePlatform 0.024056 0.009865 2.439 0.014758 \*   
## GenrePuzzle -0.061412 0.011170 -5.498 3.91e-08 \*\*\*  
## GenreRacing -0.006505 0.008338 -0.780 0.435265   
## GenreRole-Playing -0.002732 0.007831 -0.349 0.727206   
## GenreShooter 0.011520 0.008373 1.376 0.168861   
## GenreSimulation -0.004853 0.009491 -0.511 0.609087   
## GenreSports 0.015611 0.006761 2.309 0.020965 \*   
## GenreStrategy -0.050059 0.010274 -4.873 1.11e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2314 on 14663 degrees of freedom  
## Multiple R-squared: 0.09178, Adjusted R-squared: 0.08924   
## F-statistic: 36.14 on 41 and 14663 DF, p-value: < 2.2e-16

#Displaying the reidual plots  
plot(modelGB)



#Displaying results of ANOVA analysis  
summary(modelGB.aov)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Platform 30 1450 48.34 20.87 <2e-16 \*\*\*  
## Genre 11 339 30.79 13.29 <2e-16 \*\*\*  
## Residuals 16556 38345 2.32   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Prediction a NA sales (in million units) for a particular platform and genre.

predict(modelGB, data.frame(Platform = "PS3", Genre = "Shooter"))

## 1   
## 0.3288671