ProjectPart1

2023-10-17

Data Cleaning

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
sales<-read.csv("Video_Games.csv")
summary(sales)
```

```
##      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
```

```
dim(sales)
```

```
## [1] 16719    16
```

```
names(sales)
```

```
##  [1] "Name"            "Platform"        "Year_of_Release" "Genre"
##  [5] "Publisher"       "NA_Sales"        "EU_Sales"        "JP_Sales"
##  [9] "Other_Sales"     "Global_Sales"    "Critic_Score"    "Critic_Count"
## [13] "User_Score"      "User_Count"      "Developer"       "Rating"
```

```
sales<-sales[is.na(sales$User_Count)==FALSE,]
sales<-sales[is.na(sales$Critic_Count)==FALSE,]
sales<-sales[is.na(sales$Developer)==FALSE,]
sales<-sales[is.na(sales$Year_of_Release)==FALSE,]
sales<-sales[sales$Year_of_Release!="N/A",]

#sales<-drop_na(sales)

sales$User_Score<-as.numeric(sales$User_Score)
sales<-sales[,-c(5,6,7,8,9,12,14,15,16)]
sales<-sales[sales$Year_of_Release>1995,]
sales<-as.data.frame(sales)
dim(sales)
```

```
## [1] 6890    7
```

```
names(sales)
```

```
## [1] "Name"            "Platform"        "Year_of_Release" "Genre"
## [5] "Global_Sales"    "Critic_Score"    "User_Score"
```

```
summary(sales)
```

```
##      Name             Platform         Year_of_Release       Genre
##  Length:6890        Length:6890        Length:6890        Length:6890
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   Global_Sales      Critic_Score     User_Score
##  Min.   : 0.0100   Min.   :13.00   Min.   :0.500
##  1st Qu.: 0.1100   1st Qu.:62.00   1st Qu.:6.500
##  Median : 0.2900   Median :72.00   Median :7.500
##  Mean   : 0.7717   Mean   :70.26   Mean   :7.185
##  3rd Qu.: 0.7500   3rd Qu.:80.00   3rd Qu.:8.200
##  Max.   :82.5300   Max.   :98.00   Max.   :9.600
```

```
write.csv(sales,file="VideoGamesSales.csv")


prop.table(table(sales$Platform))*100
```

```
##
##       3DS        DC        DS       GBA        GC        PC        PS       PS2
##  2.264151  0.203193  6.748911  3.439768  5.050798  9.941945  2.206096 16.545718
##       PS3       PS4       PSP       PSV       Wii      WiiU      X360        XB
## 11.248186  3.613933  5.660377  1.712627  6.966618  1.291727 12.496372  8.214804
##      XOne
##  2.394775
```

```
prop.table(table(sales$Year_of_Release))*100
```

```
##
##      1996      1997      1998      1999      2000      2001      2002      2003
## 0.1161103 0.2031930 0.3773585 0.4354136 1.4804064 3.7155298 6.6037736 7.2423803
##      2004      2005      2006      2007      2008      2009      2010      2011
## 6.9230769 8.1567489 7.6632801 8.5631350 8.6357039 8.0406386 6.2554427 6.7634253
##      2012      2013      2014      2015      2016
## 4.6589260 3.9477504 3.7155298 3.2075472 3.2946299
```

```
prop.table(table(sales$Genre))*100
```

```
##
##       Action    Adventure     Fighting         Misc     Platform       Puzzle
##    23.860668     3.831640     5.486212     5.602322     5.849057     1.712627
##       Racing Role-Playing      Shooter   Simulation       Sports     Strategy
##     8.505080    10.377358    12.583454     4.354136    13.802612     4.034833
```

```
summary(sales)
```

```
##      Name             Platform         Year_of_Release       Genre
##  Length:6890        Length:6890        Length:6890        Length:6890
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   Global_Sales      Critic_Score     User_Score
##  Min.   : 0.0100   Min.   :13.00   Min.   :0.500
##  1st Qu.: 0.1100   1st Qu.:62.00   1st Qu.:6.500
##  Median : 0.2900   Median :72.00   Median :7.500
##  Mean   : 0.7717   Mean   :70.26   Mean   :7.185
##  3rd Qu.: 0.7500   3rd Qu.:80.00   3rd Qu.:8.200
##  Max.   :82.5300   Max.   :98.00   Max.   :9.600
```

Fitting the Model

```
VideoGamesSales <- data.frame(read.csv("VideoGamesSales.csv"))
model <- lm (Global_Sales ~ Critic_Score + User_Score + Platform + Year_of_Release + Genre, data = Vide
model_1 <- summary(model)

coef_table <- data.frame(
  Coefficient = rownames (model_1$coefficients),
  Estimate = model_1$coefficients [, 1],
  Std.Error = model_1$coefficients [, 2],
  T.Value = model_1$coefficients [, 3],
  P.Value = model_1$coefficients [, 4]
)
```

```
# R-squared value
r_squared <- model_1$r.squared
# Print the coefficient table and R-squared
print(coef_table)
```

```
##                         Coefficient    Estimate    Std.Error    T.Value
## (Intercept)             (Intercept) 69.83636790 21.766530671  3.2084290
## Critic_Score           Critic_Score  0.04626204  0.002125067 21.7696825
## User_Score               User_Score -0.12290939  0.020749513 -5.9234831
## PlatformDC               PlatformDC -1.56441525  0.534917768 -2.9245902
## PlatformDS               PlatformDS -0.03592639  0.180586447 -0.1989429
## PlatformGBA             PlatformGBA -0.61893960  0.216505888 -2.8587657
## PlatformGC               PlatformGC -0.75033619  0.204578194 -3.6677232
## PlatformPC               PlatformPC -0.95257884  0.170772731 -5.5780500
## PlatformPS               PlatformPS -0.06116299  0.254552215 -0.2402768
## PlatformPS2             PlatformPS2 -0.26824209  0.184853977 -1.4511026
## PlatformPS3             PlatformPS3 -0.05354109  0.166021555 -0.3224948
## PlatformPS4             PlatformPS4  0.04147435  0.191850625  0.2161805
## PlatformPSP             PlatformPSP -0.48791204  0.185391850 -2.6317880
## PlatformPSV             PlatformPSV -0.56109391  0.226548645 -2.4767039
## PlatformWii             PlatformWii  0.57411159  0.177252367  3.2389502
## PlatformWiiU           PlatformWiiU -0.19666191  0.246725686 -0.7970873
## PlatformX360           PlatformX360 -0.02194428  0.166023541 -0.1321757
## PlatformXB               PlatformXB -0.85768939  0.193999268 -4.4210960
## PlatformXOne           PlatformXOne -0.27857087  0.209938265 -1.3269181
## Year_of_Release     Year_of_Release -0.03541161  0.010808035 -3.2764150
## GenreAdventure       GenreAdventure -0.30488610  0.123376194 -2.4711907
## GenreFighting         GenreFighting -0.18909481  0.106123915 -1.7818303
## GenreMisc                 GenreMisc  0.16097459  0.105781380  1.5217668
## GenrePlatform         GenrePlatform  0.10214749  0.104597691  0.9765750
## GenrePuzzle             GenrePuzzle -0.34892880  0.181338283 -1.9241872
## GenreRacing             GenreRacing  0.01895743  0.090132309  0.2103289
## GenreRole-Playing GenreRole-Playing -0.16442662  0.084558025 -1.9445418
## GenreShooter           GenreShooter  0.14581029  0.078733649  1.8519437
## GenreSimulation     GenreSimulation -0.06800288  0.118225238 -0.5751976
## GenreSports             GenreSports -0.17216091  0.077835484 -2.2118564
## GenreStrategy         GenreStrategy -0.42174683  0.124745466 -3.3808590
##                         P.Value
## (Intercept)        1.340761e-03
## Critic_Score      1.166783e-101
## User_Score         3.304410e-09
## PlatformDC         3.460396e-03
## PlatformDS         8.423133e-01
## PlatformGBA        4.265750e-03
## PlatformGC         2.465731e-04
## PlatformPC         2.525023e-08
## PlatformPS         8.101229e-01
## PlatformPS2        1.467970e-01
## PlatformPS3        7.470877e-01
## PlatformPS4        8.288535e-01
## PlatformPSP        8.512701e-03
## PlatformPSV        1.328412e-02
## PlatformWii        1.205419e-03
## PlatformWiiU       4.254279e-01
## PlatformX360       8.948492e-01
## PlatformXB         9.971514e-06
## PlatformXOne       1.845800e-01
## Year_of_Release    1.056564e-03
## GenreAdventure     1.349051e-02
## GenreFighting      7.482113e-02
## GenreMisc          1.281136e-01
## GenrePlatform      3.288140e-01
## GenrePuzzle        5.437245e-02
## GenreRacing        8.334173e-01
## GenreRole-Playing  5.187097e-02
## GenreShooter       6.407685e-02
## GenreSimulation    5.651765e-01
## GenreSports        2.700943e-02
## GenreStrategy      7.266311e-04
```

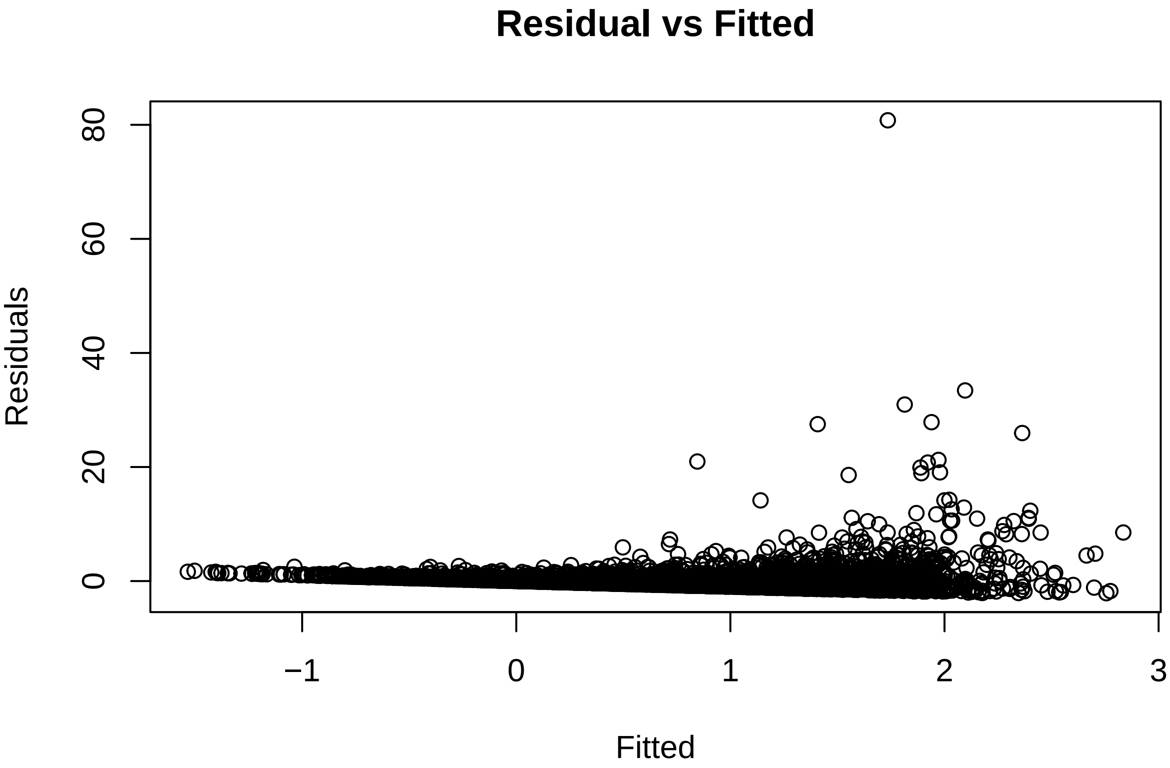
```
cat (paste("R-squared: ", round (r_squared, 4), "\n"))
```

```
## R-squared:  0.1079
```

Checking Assumptions

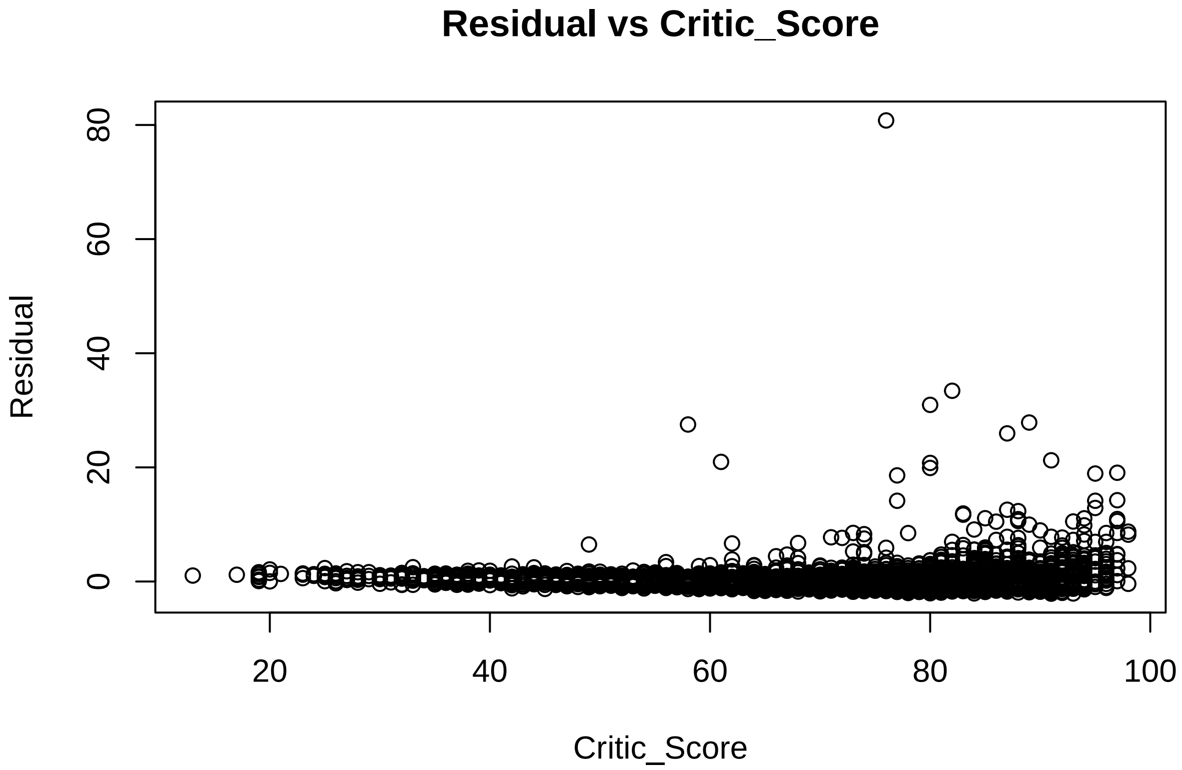
First we make the plot for the residuals versus fitted values.

```
y_hat <- fitted(model)
e_hat <- resid(model)
plot(x =y_hat, y = e_hat, main="Residual vs Fitted", xlab="Fitted", ylab="Residuals")
```

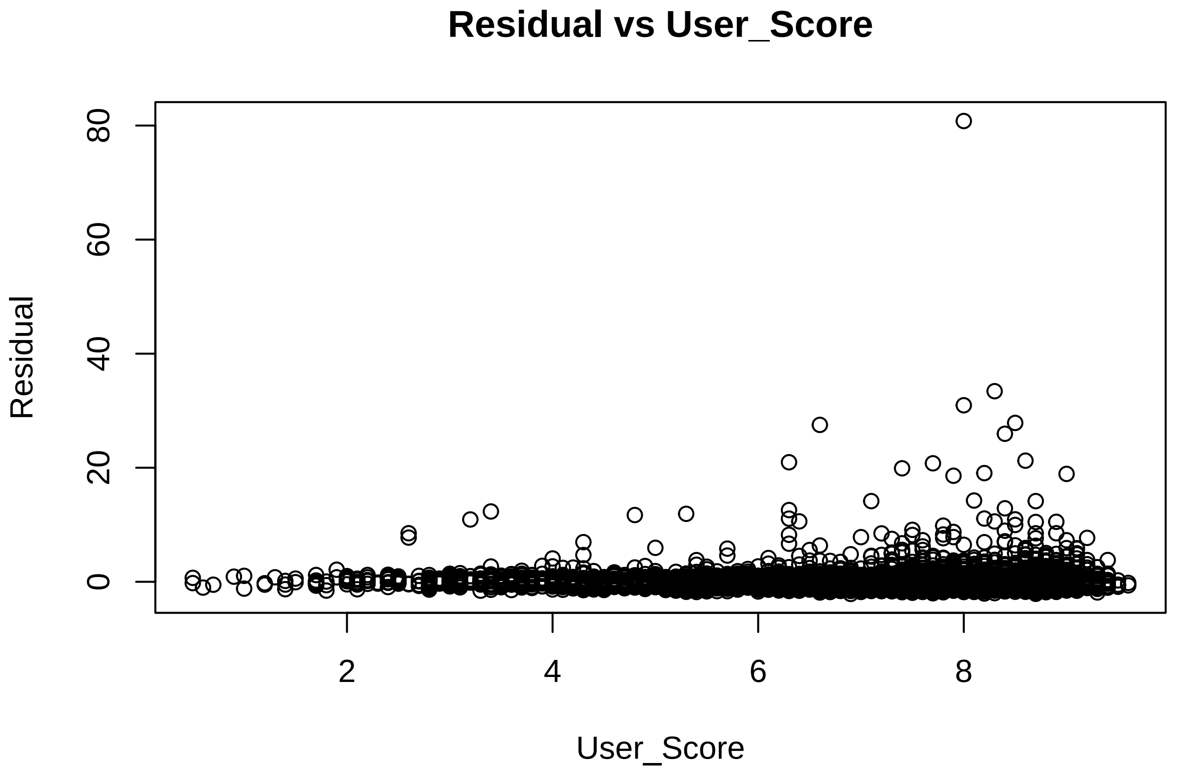


Then we create the residual versus predictor plots for our numerical predictors (Critic_Score, User_Score, Year_of_Release).

```
plot(x = VideoGamesSales$Critic_Score, y = e_hat, main="Residual vs Critic_Score", xlab="Critic_Score",
```

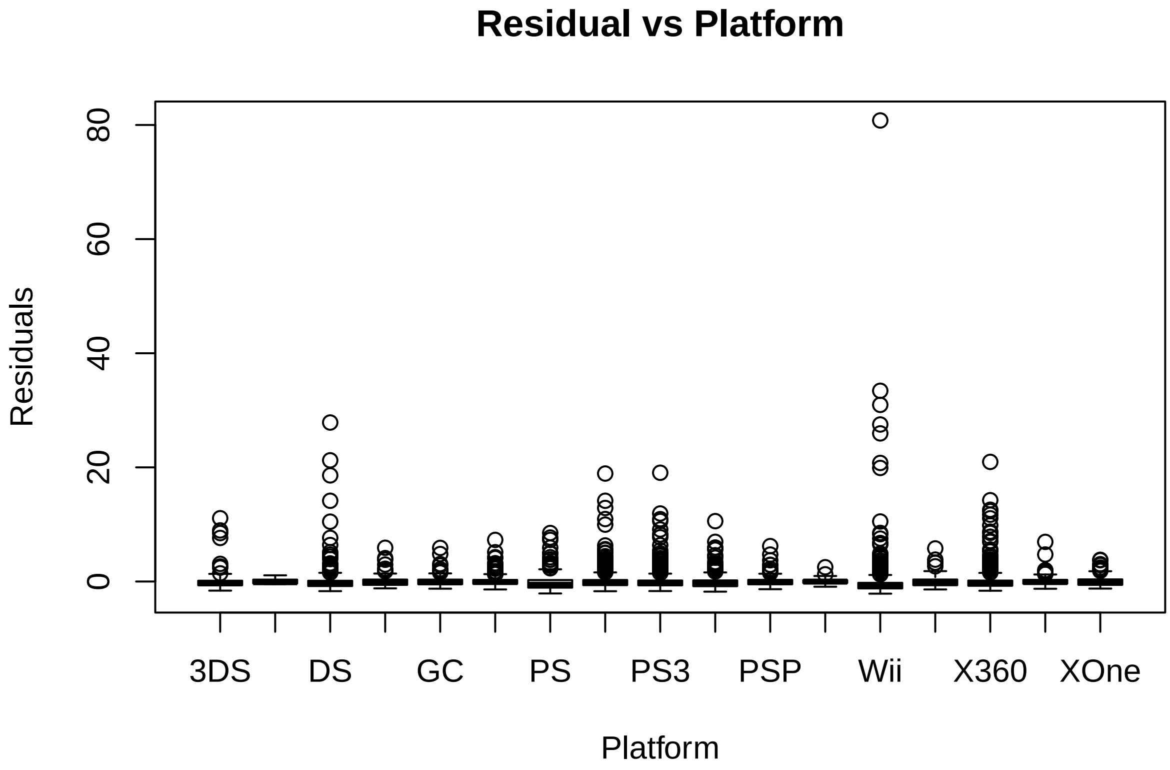


```
plot(x = VideoGamesSales$User_Score, y = e_hat, main="Residual vs User_Score", xlab="User_Score", ylab=
```

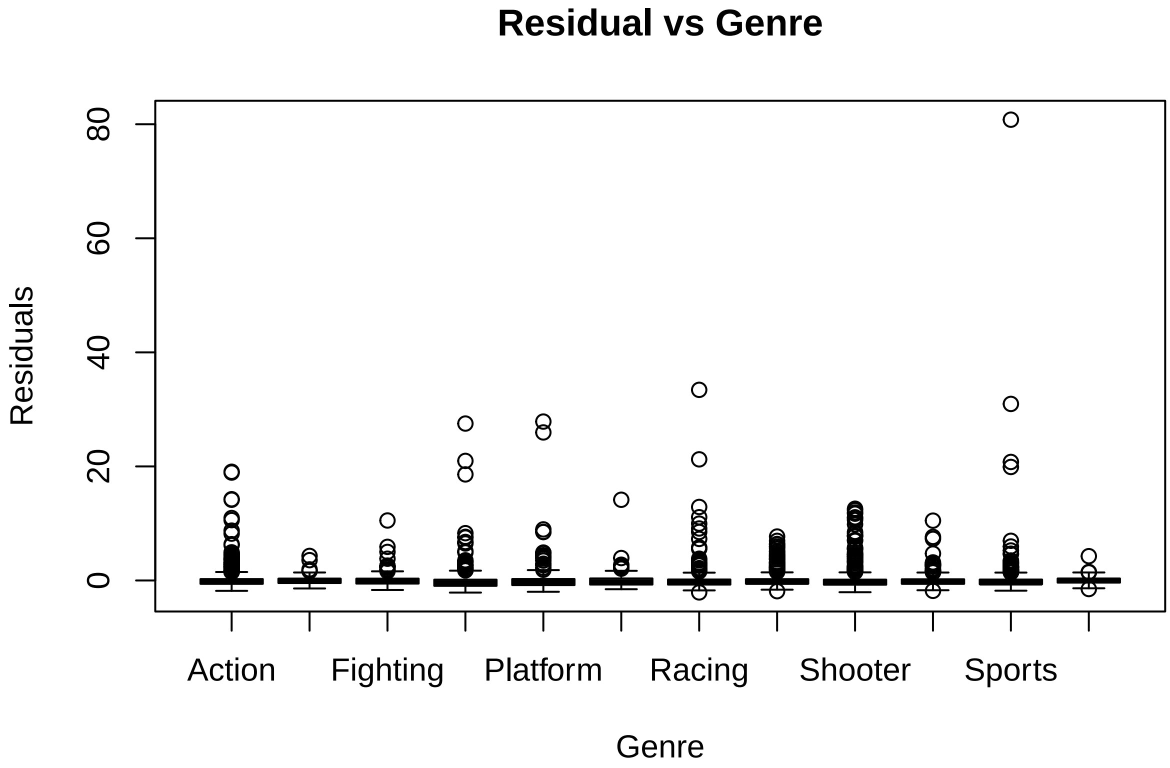


Next we create the residual plots using categorical predictors (Platform, Genre, Year_of_Release).

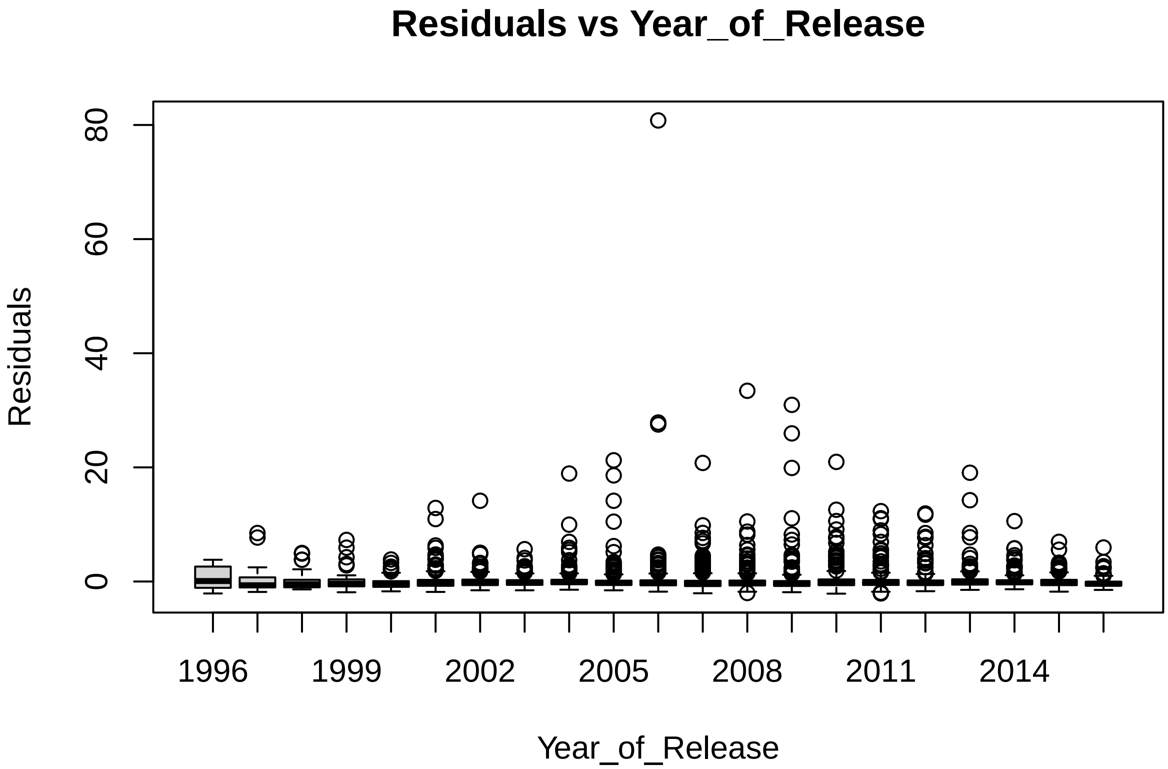
```
boxplot(e_hat ~ VideoGamesSales$Platform , main="Residual vs Platform", xlab="Platform", ylab="Residual
```



```
boxplot(e_hat ~ VideoGamesSales$Genre , main="Residual vs Genre", xlab="Genre", ylab="Residuals")
```



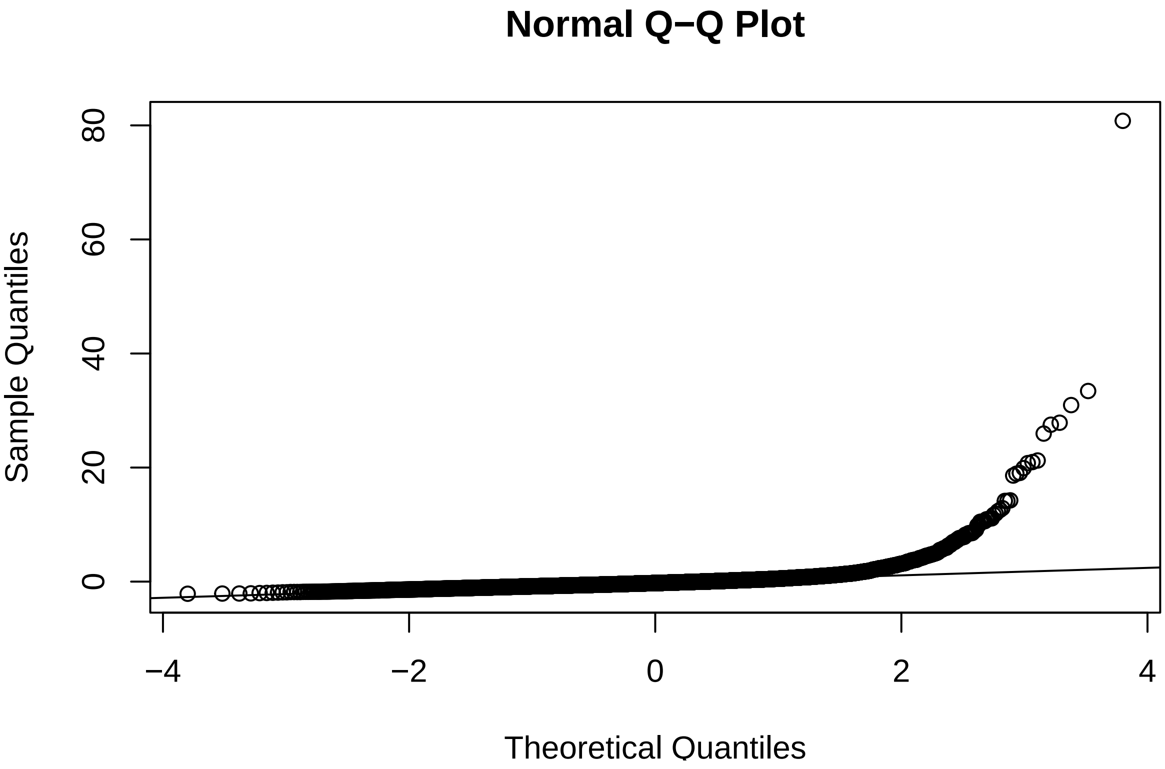
```
boxplot(e_hat ~ VideoGamesSales$Year_of_Release , main="Residuals vs Year_of_Release", xlab="Year_of_Rel
```



Currently, as we have many categories for these variables, these box plots are not very readable. I propose that during the next part of our project, we could limit our dataset to observations that fall into the most popular categories and remove ones whose categories have very few members (for example Action for Genre has a 23.86 percentage so we would keep its observations as a popular category). I also think we could consider year to be a numerical variable in the future.

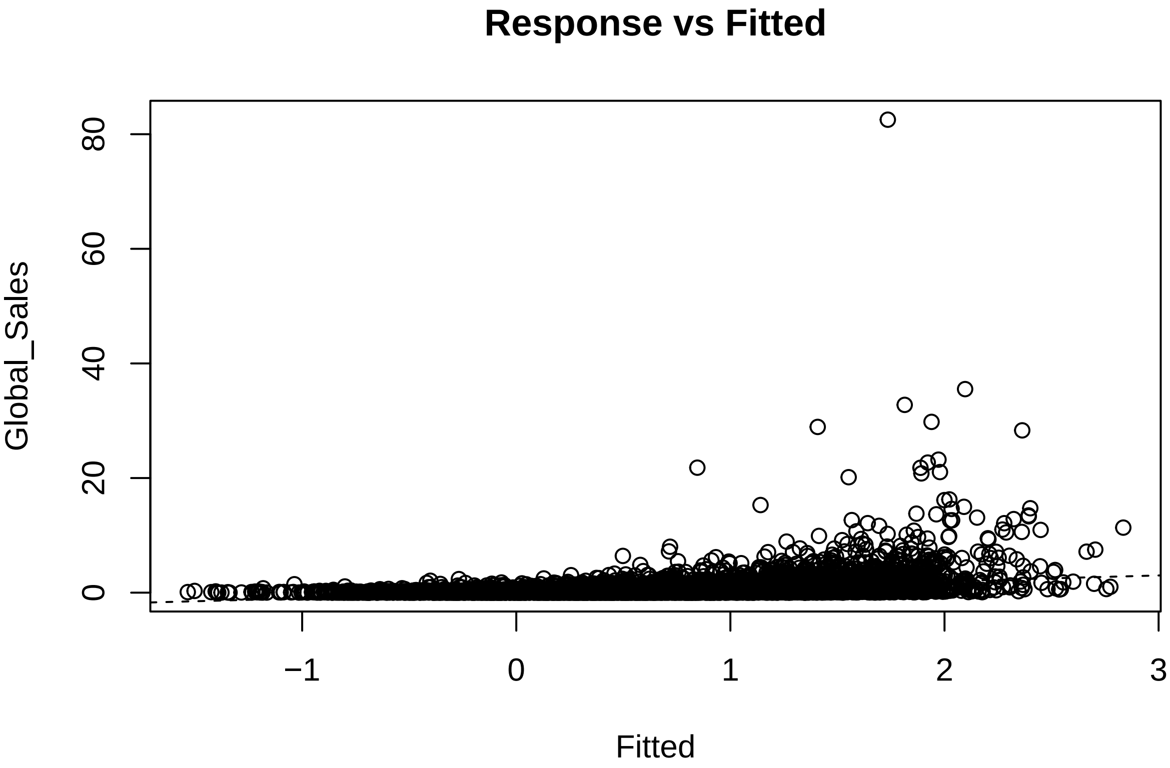
Lastly, we create the QQ plot.

```
qqnorm(e_hat)
qqline(e_hat)
```



Next, let's check the additional conditions for multiple linear models: 1. Conditional mean response condition 2. Conditional mean predictor condition Let's make a scatterplot of our response versus fitted values to check condition 1.

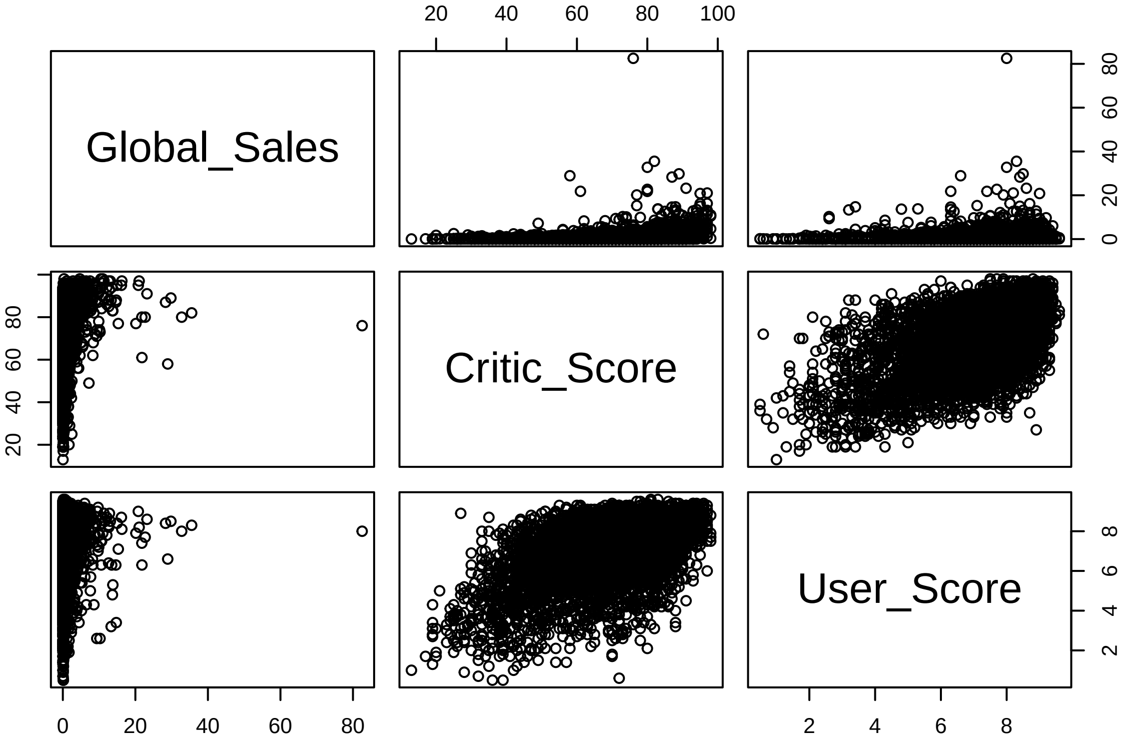
```
plot(x = y_hat, y = VideoGamesSales$Global_Sales, main="Response vs Fitted", xlab="Fitted", ylab="Globa
abline(a = 0, b = 1, lty=2)
```



Based on this plot, we seem to have roughly random scatter around the diagonal line so the 1st condition holds.

Next, let's check the 2nd condition.

```r
# a new dataframe with only the numerical values
new <- subset(VideoGamesSales, select = c(Global_Sales, Critic_Score, User_Score))
pairs(new)
```



I got the following error when trying to include categorical attributes so I only included numerical attributes for the above plot: Error in pairs.default(new) : non-numeric argument to 'pairs'

The 2nd condition seems to be satisfied as well as there are no non-linear patterns present.