ProjectPart3

2023-10-17

Data Cleaning

```r
original_sales<-read.csv("Video_Games.csv")
# make a copy of the dataset
sales<-original_sales

# remove all rows that have N/A or empty information
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",]

#Changing User_Score from a categorical variable into a numerical variable
sales$User_Score<-as.numeric(sales$User_Score)
sales<-sales[,-c(5,6,7,8,9,12,14,15,16)] #Erase the columns we don't need
sales<-sales[sales$Year_of_Release>2000,] #Limit years higher than 2000
sales<-as.data.frame(sales)
#dim(sales)
#names(sales)
#summary(sales)

sales$Genre2 <- sales$Genre # New variable
sales$Genre2[sales$Genre2=="Role-Playing"]<-"Role"
#Remove less popular categories
sales<-sales[sales$Genre2!="Simulation",]
sales<-sales[sales$Genre2!="Puzzle",]
sales<-sales[sales$Genre2!="Adventure",]
sales<-sales[sales$Genre2!="Strategy",]
sales<-sales[sales$Genre2!="Fighting",]
sales<-sales[sales$Genre2!="Misc",]
sales<-sales[sales$Genre2!="Platform",]
table(sales$Genre2)
```

```
##
##  Action  Racing    Role Shooter  Sports
##    1618     564     677     853     936
```

```r
#Changing year from a categorical variable into a numerical variable
sales$Year_of_Release<-as.numeric(sales$Year_of_Release)
#table(sales$Year_of_Release)

sales<-sales[,-c(4)] # Drop the original variable of Genre and leave only Genre2
#summary(sales)
```

```r
table(sales$Platform)
```

```
##
##   3DS    DC    DS   GBA    GC    PC    PS   PS2   PS3   PS4   PSP   PSV   Wii  WiiU  X360    XB
##    92     2   231   135   228   417    26   788   608   198   273    84   280    53   657   440
##  XOne
##   136
```

```r
sort(table(sales$Platform))
```

```
##
##    DC    PS  WiiU   PSV   3DS   GBA  XOne   PS4    GC    DS   PSP   Wii    PC    XB   PS3  X360
##     2    26    53    84    92   135   136   198   228   231   273   280   417   440   608   657
##   PS2
##   788
```

```r
#boxplot(logSales~sales$Platform)
sales$Platform2 <- sales$Platform # New variable
#Remove less popular categories
sales<-sales[sales$Platform2!="DC",]
sales<-sales[sales$Platform2!="PS",]
sales<-sales[sales$Platform2!="WiiU",]
sales<-sales[sales$Platform2!="PSV",]
sales<-sales[sales$Platform2!="3DS",]
sales<-sales[sales$Platform2!="GBA",]
sales<-sales[sales$Platform2!="XOne",]
sales<-sales[sales$Platform2!="PS4",]
sales<-sales[sales$Platform2!="GC",]
sales<-sales[sales$Platform2!="DS",]
sales<-sales[sales$Platform2!="PSP",]
sales<-sales[sales$Platform2!="Wii",]

table(sales$Platform2)
```

```
##
##    PC   PS2   PS3  X360    XB
##   417   788   608   657   440
```

```r
#boxplot(logSales~sales$Platform2)
#summary(sales)

sales<-sales[,-c(2)] #Drop the original variable of Platform and leave only Platform2
summary(sales)
```

```
##       Name           Year_of_Release  Global_Sales      Critic_Score
##   Length:2910        Min.   :2001    Min.   : 0.0100   Min.   :13.00
##   Class :character   1st Qu.:2004    1st Qu.: 0.1300   1st Qu.:63.00
##   Mode  :character   Median :2007    Median : 0.3200   Median :74.00
##                      Mean   :2007    Mean   : 0.8148   Mean   :71.37
##                      3rd Qu.:2010    3rd Qu.: 0.8400   3rd Qu.:82.00
##                      Max.   :2016    Max.   :21.0400   Max.   :98.00
##    User_Score       Genre2            Platform2
##   Min.   :1.000   Length:2910        Length:2910
##   1st Qu.:6.400   Class :character   Class :character
##   Median :7.500   Mode  :character   Mode  :character
##   Mean   :7.171
```

```
##  3rd Qu.:8.200
##  Max.   :9.500
```

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

Fitting the Model

```
sales <- data.frame(read.csv("VideoGamesSales.csv"))
model <- lm (Global_Sales ~ Critic_Score + User_Score + Platform2 + Year_of_Release + Genre2, data = sal
model_1 <- summary(model)
model_1
```

```
##
## Call:
## lm(formula = Global_Sales ~ Critic_Score + User_Score + Platform2 +
##     Year_of_Release + Genre2, data = sales)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0644 -0.6589 -0.2477  0.2824 18.5934
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      9.363365  23.099862   0.405  0.68526
## Critic_Score     0.059556   0.002361  25.226  < 2e-16 ***
## User_Score      -0.205574   0.023938  -8.588  < 2e-16 ***
## Platform2PS2     1.268451   0.111682  11.358  < 2e-16 ***
## Platform2PS3     1.193410   0.089339  13.358  < 2e-16 ***
## Platform2X360    1.261928   0.089053  14.171  < 2e-16 ***
## Platform2XB      0.572129   0.120264   4.757 2.06e-06 ***
## Year_of_Release -0.006061   0.011481  -0.528  0.59757
## Genre2Racing    -0.222026   0.085109  -2.609  0.00913 **
## Genre2Role      -0.198190   0.087863  -2.256  0.02417 *
## Genre2Shooter    0.148227   0.071559   2.071  0.03841 *
## Genre2Sports    -0.497989   0.075181  -6.624 4.16e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.381 on 2898 degrees of freedom
## Multiple R-squared:  0.2315, Adjusted R-squared:  0.2286
## F-statistic: 79.37 on 11 and 2898 DF,  p-value: < 2.2e-16
```

```
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)  9.363364933 23.099861936  0.4053429
## Critic_Score       Critic_Score  0.059555828  0.002360853 25.2264035
## User_Score           User_Score -0.205574224  0.023937995 -8.5877797
## Platform2PS2       Platform2PS2  1.268451124  0.111682384 11.3576652
## Platform2PS3       Platform2PS3  1.193410218  0.089338951 13.3582296
## Platform2X360     Platform2X360  1.261928061  0.089052628 14.1705876
## Platform2XB         Platform2XB  0.572129111  0.120264084  4.7572733
## Year_of_Release Year_of_Release -0.006061278  0.011480772 -0.5279504
## Genre2Racing       Genre2Racing -0.222025641  0.085108872 -2.6087250
## Genre2Role           Genre2Role -0.198189986  0.087863100 -2.2556680
## Genre2Shooter     Genre2Shooter  0.148226671  0.071559277  2.0713830
## Genre2Sports       Genre2Sports -0.497988662  0.075181187 -6.6238467
##                         P.Value
## (Intercept)      6.852554e-01
## Critic_Score    4.155933e-127
## User_Score       1.424172e-17
## Platform2PS2     2.799242e-29
## Platform2PS3     1.529716e-39
## Platform2X360    4.007782e-44
## Platform2XB      2.058292e-06
## Year_of_Release  5.975742e-01
## Genre2Racing     9.134722e-03
## Genre2Role       2.416571e-02
## Genre2Shooter    3.841131e-02
## Genre2Sports     4.155888e-11
```

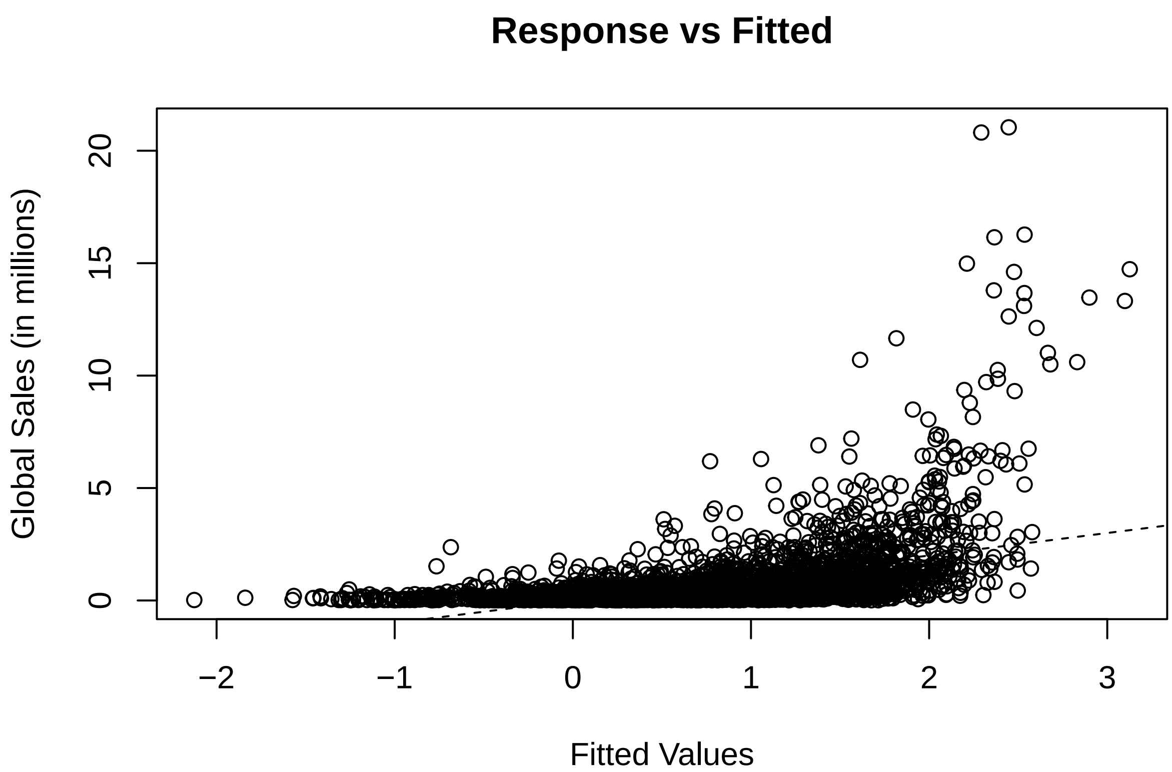
```r
cat (paste("R-squared: ", round (r_squared, 4), "\n"))
```

```
## R-squared:  0.2315
```

Checking MLR Conditions

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.

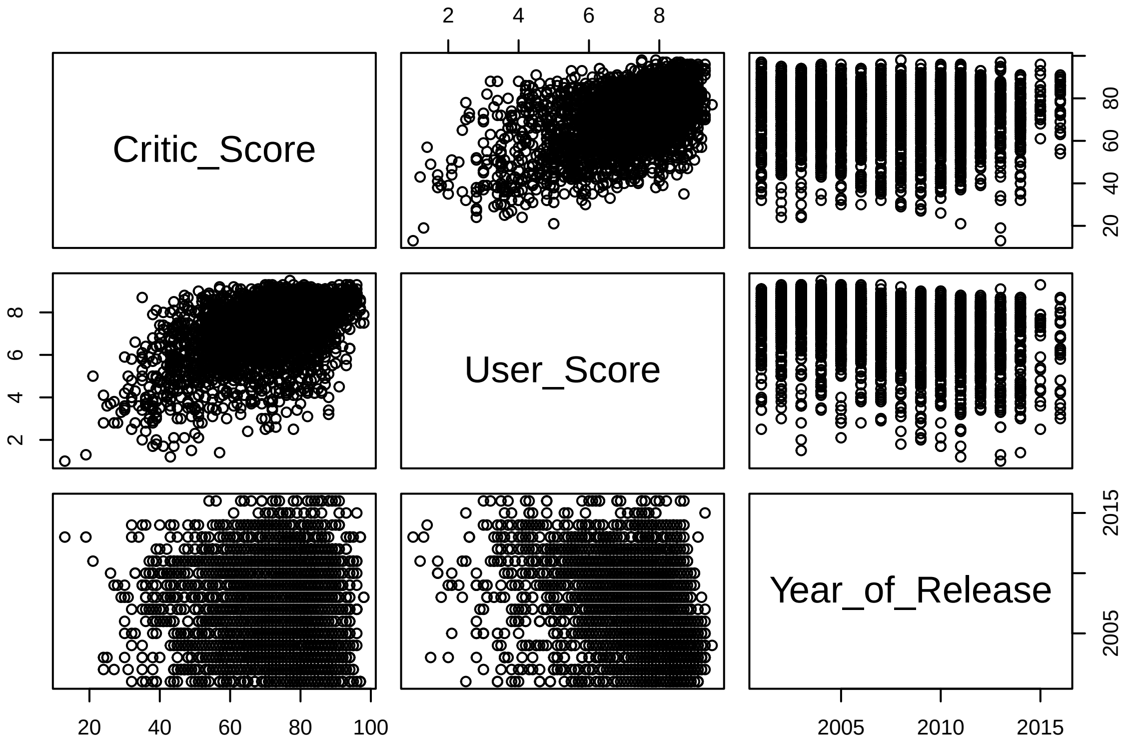
```r
y_hat <- fitted(model)
plot(x = y_hat, y = sales$Global_Sales, main="Response vs Fitted", xlab="Fitted Values", ylab="Global Sa
abline(a = 0, b = 1, lty=2)
```



Based on this plot, we don't observe random diagonal scatter or an easily identifiable non-linear trend so the 1st condition does not seem to hold. As a result, the residual plots will not be reliable.

Next, let's check the 2nd condition.

```r
# a new dataframe with only the numerical predictors
new <- subset(sales, select = c(Critic_Score, User_Score, Year_of_Release))
pairs(new)
```

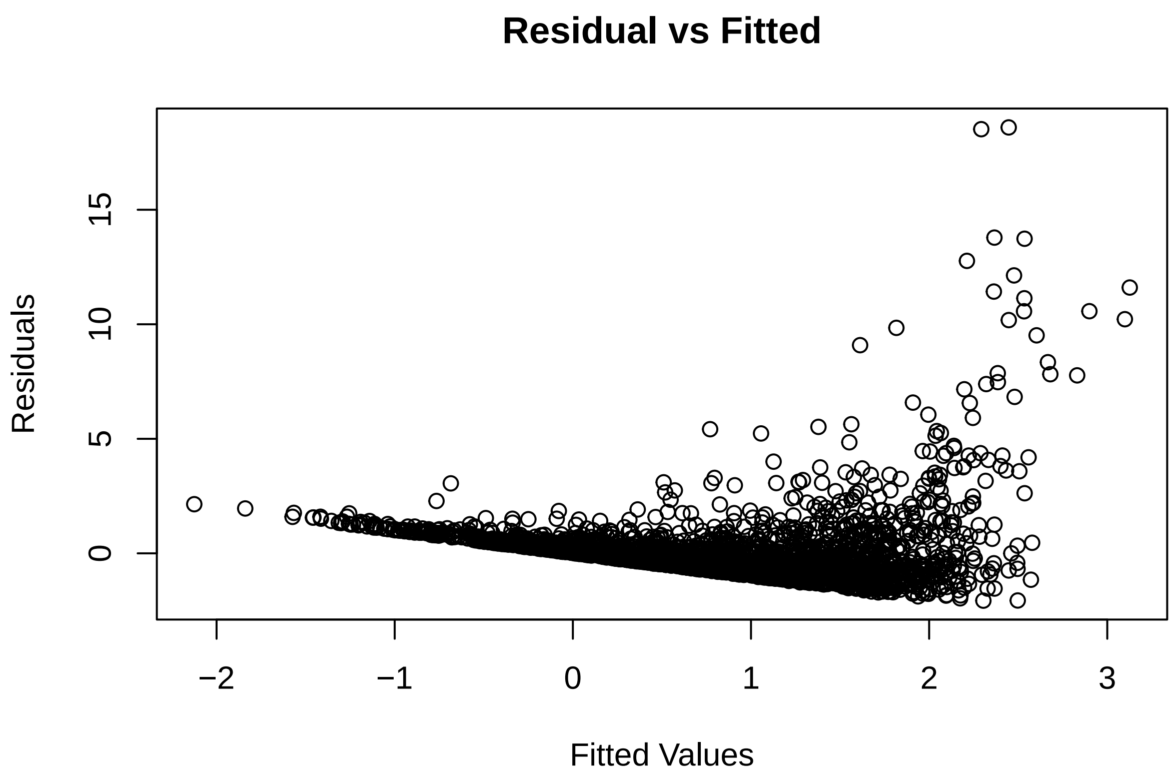


The 2nd condition seems to be satisfied as we observe a lack of curves or other non-linear patterns.

Checking Assumptions

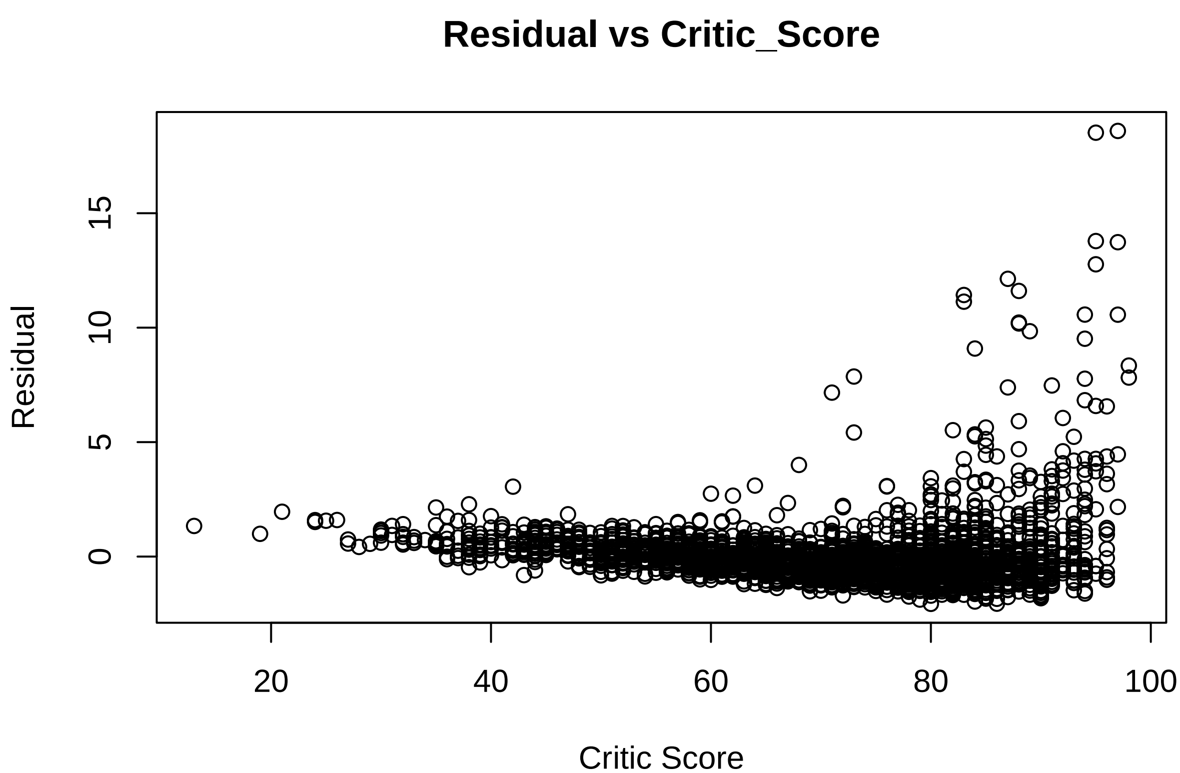
First we make the plot for the residuals versus fitted values.

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

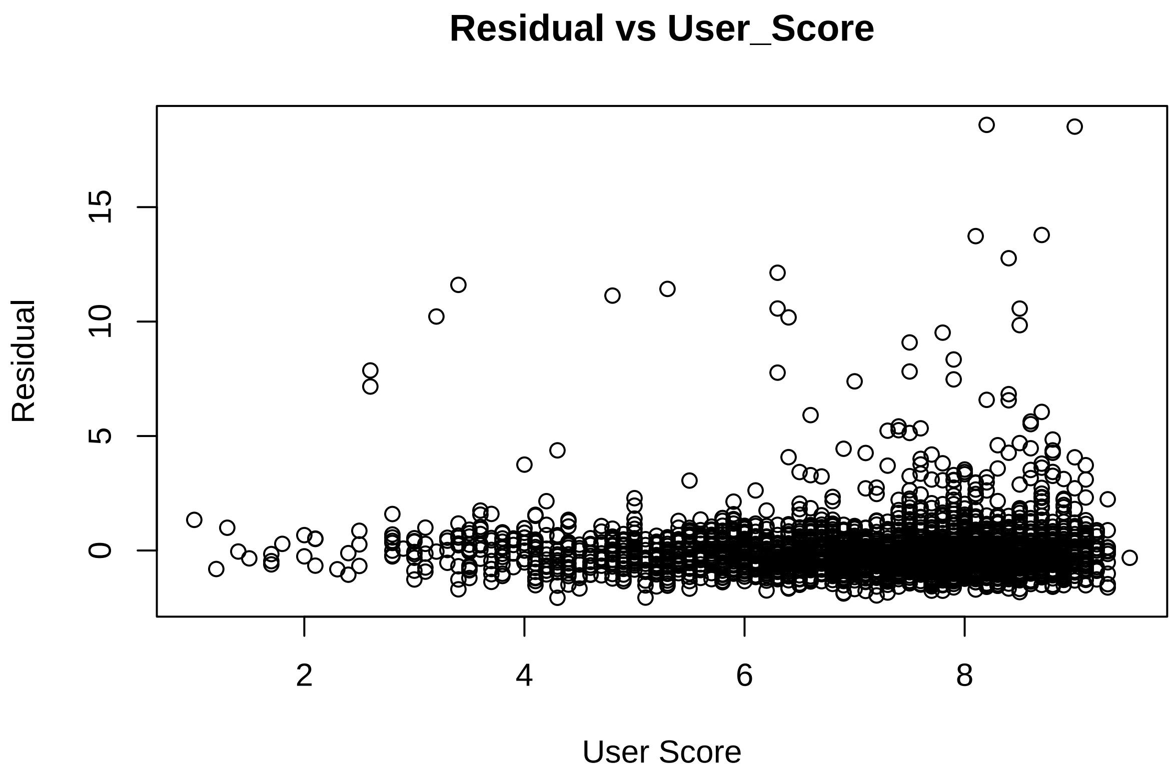


Then we create the residual versus predictor plots for our numerical predictors (Critic_Score, User_Score, Year_of_Release).

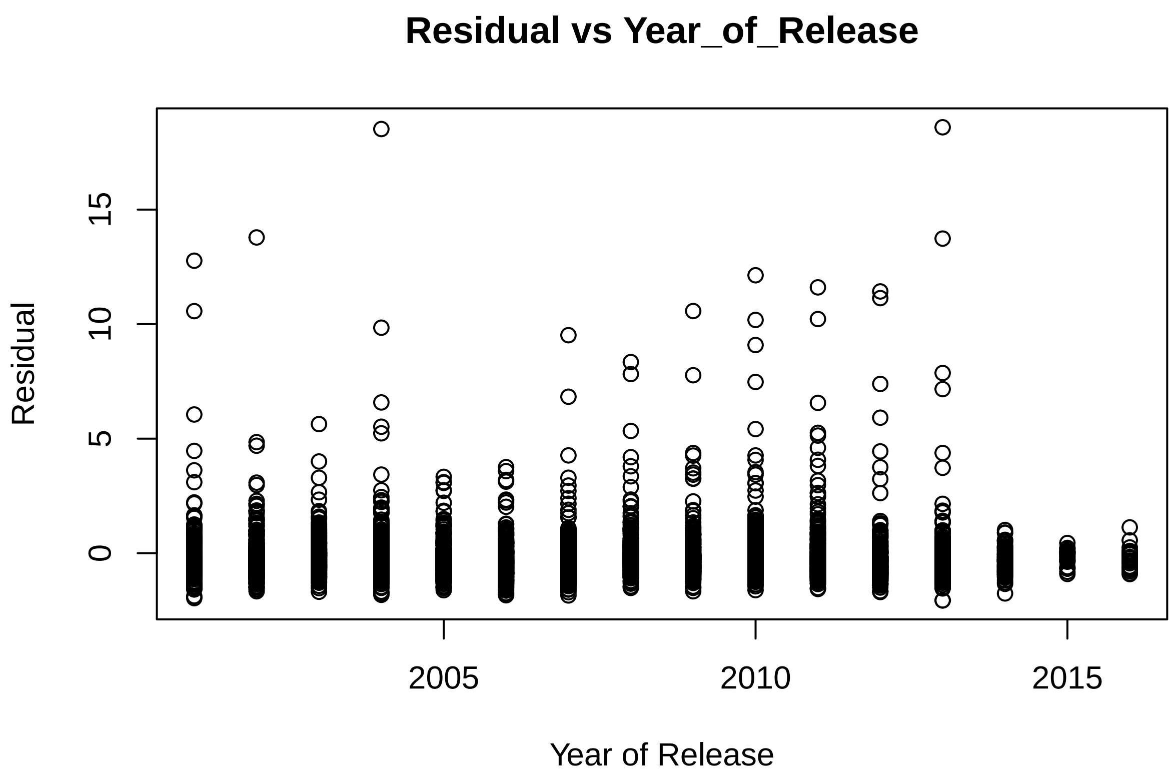
```
plot(x = sales$Critic_Score, y = e_hat, main="Residual vs Critic_Score", xlab="Critic Score", ylab="Res
```



```
plot(x = sales$User_Score, y = e_hat, main="Residual vs User_Score", xlab="User Score", ylab="Residual")
```

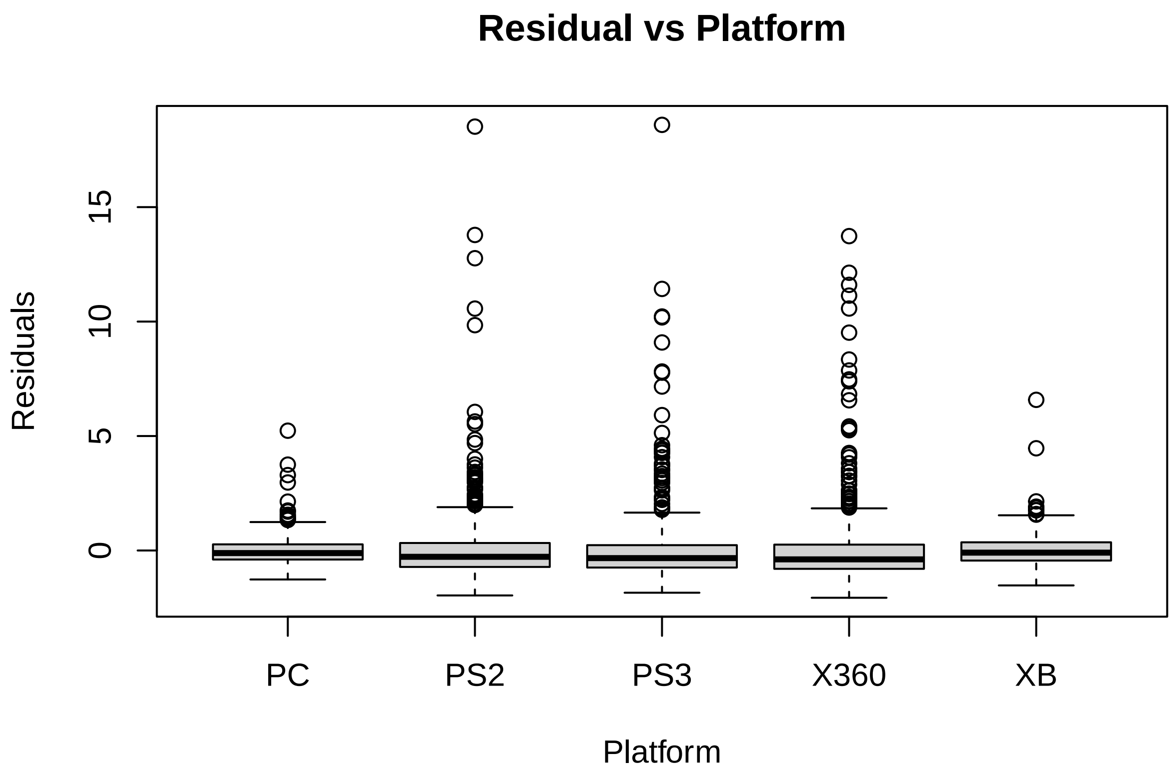



```
plot(x = sales$Year_of_Release, y = e_hat, main="Residual vs Year_of_Release", xlab="Year of Release", y
```

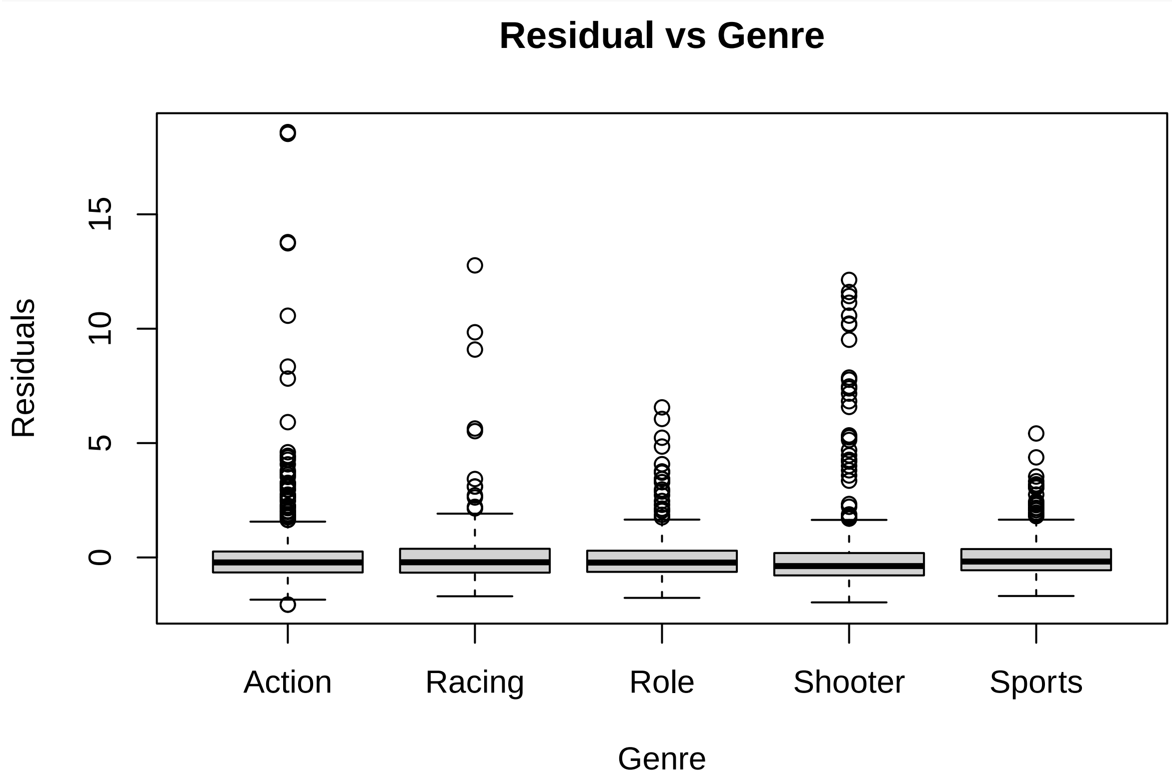


Next we create the residual plots using categorical predictors (Platform, Genre).

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

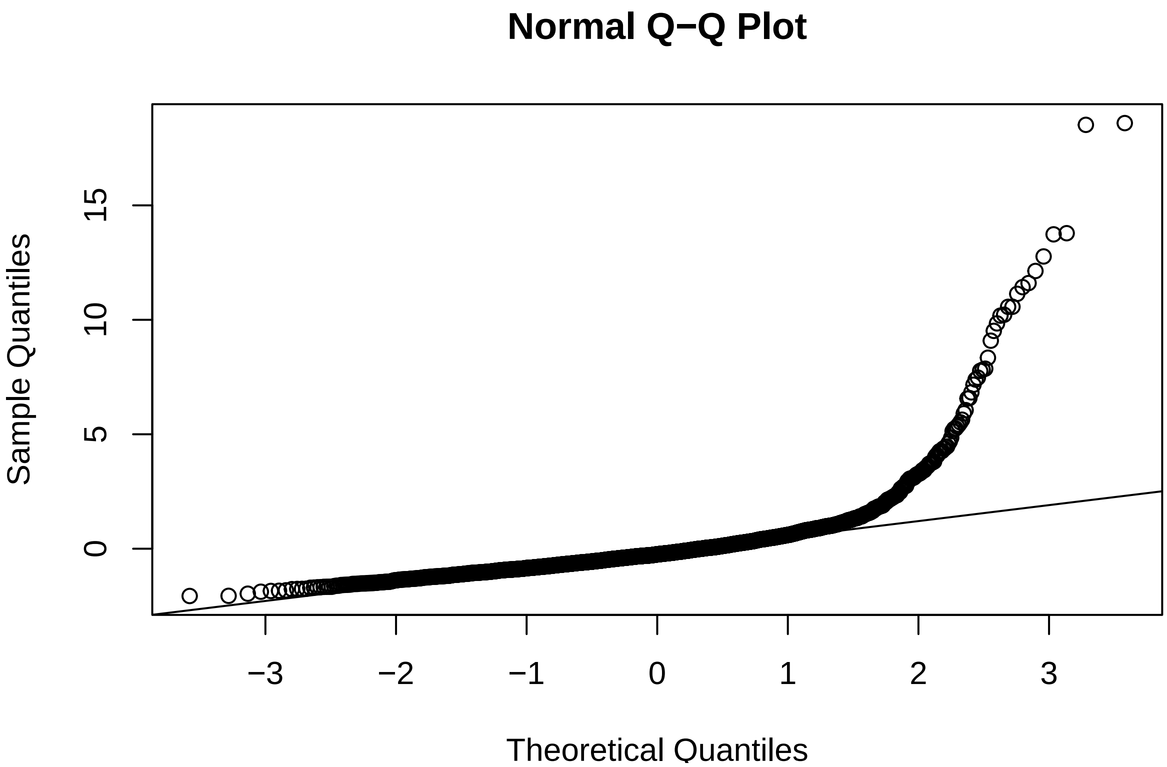



```
boxplot(e_hat ~ sales$Genre , main="Residual vs Genre", xlab="Genre", ylab="Residuals")
```



Lastly, we create the QQ plot.

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



We observe a violation of the normality assumption based on the deviation and curving from the diagonal line that occurs in the QQ plot. We also have some evidence of a violation of the constant variance assumption due to the increase of the spread shown in the residual vs fitted, residual vs user_score, and residual vs critic_score plots. We also have evidence of a violation of the linearity assumption since we observe some systemic patterns in the residual vs fitted, residual vs user_score, and residual vs critic_score plots. As we don't observe any large clusters if points or patterns across time we don't have a violation of the uncorrelated errors assumption.

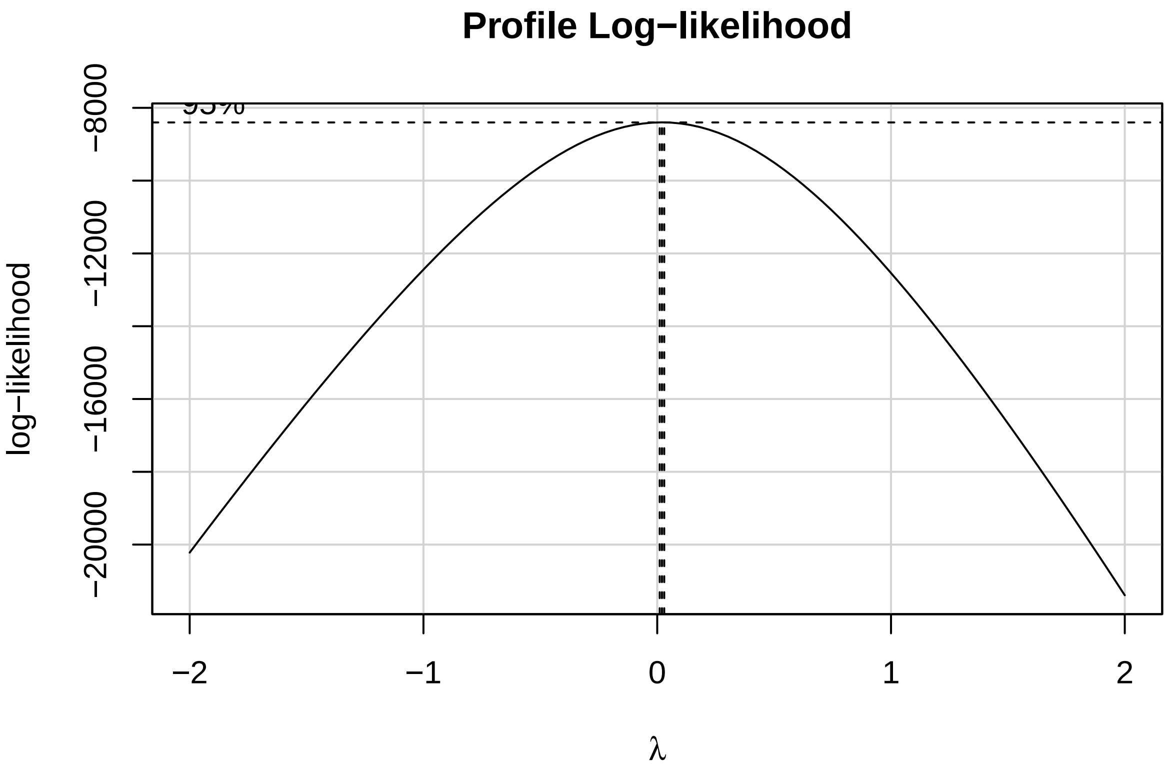
Transformations

We apply Box-Cox transformation to the response to mitigate the observed violation of the linearity assumption.

```r
# Transformation on Y
library(car)
```

```
## Loading required package: carData
```

```r
boxCox(model)
```



The 95% CI on MLE is very close to 0 so ln transformation is reasonable. Based on the transformation on y, we fit a new model:

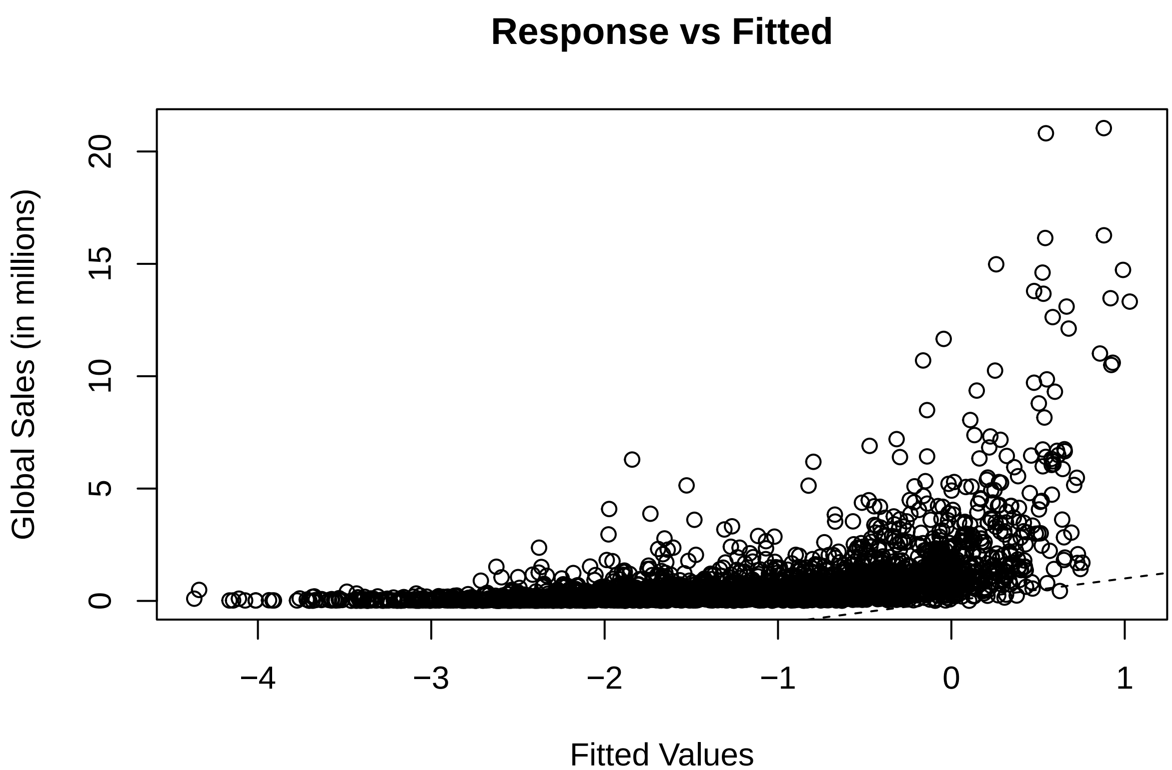
```
model_ln <- lm (log(Global_Sales) ~ Critic_Score + User_Score + Platform2 + Year_of_Release + Genre2, da
summary(model_ln)
```

```
##
## Call:
## lm(formula = log(Global_Sales) ~ Critic_Score + User_Score +
##     Platform2 + Year_of_Release + Genre2, data = sales)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7079 -0.6677  0.0300  0.6913  3.6803
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -49.639739  17.484269  -2.839 0.004555 **
## Critic_Score      0.059145   0.001787  33.099  < 2e-16 ***
## User_Score       -0.132174   0.018119  -7.295 3.84e-13 ***
## Platform2PS2      2.259776   0.084532  26.733  < 2e-16 ***
## Platform2PS3      2.174448   0.067621  32.157  < 2e-16 ***
## Platform2X360     2.161901   0.067404  32.074  < 2e-16 ***
## Platform2XB       1.404005   0.091028  15.424  < 2e-16 ***
## Year_of_Release   0.021705   0.008690   2.498 0.012555 *
## Genre2Racing     -0.300751   0.064419  -4.669 3.17e-06 ***
## Genre2Role       -0.233194   0.066504  -3.506 0.000461 ***
## Genre2Shooter     0.064563   0.054163   1.192 0.233357
## Genre2Sports     -0.283759   0.056905  -4.987 6.51e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.045 on 2898 degrees of freedom
## Multiple R-squared:  0.4511, Adjusted R-squared:  0.449
## F-statistic: 216.5 on 11 and 2898 DF,  p-value: < 2.2e-16
```

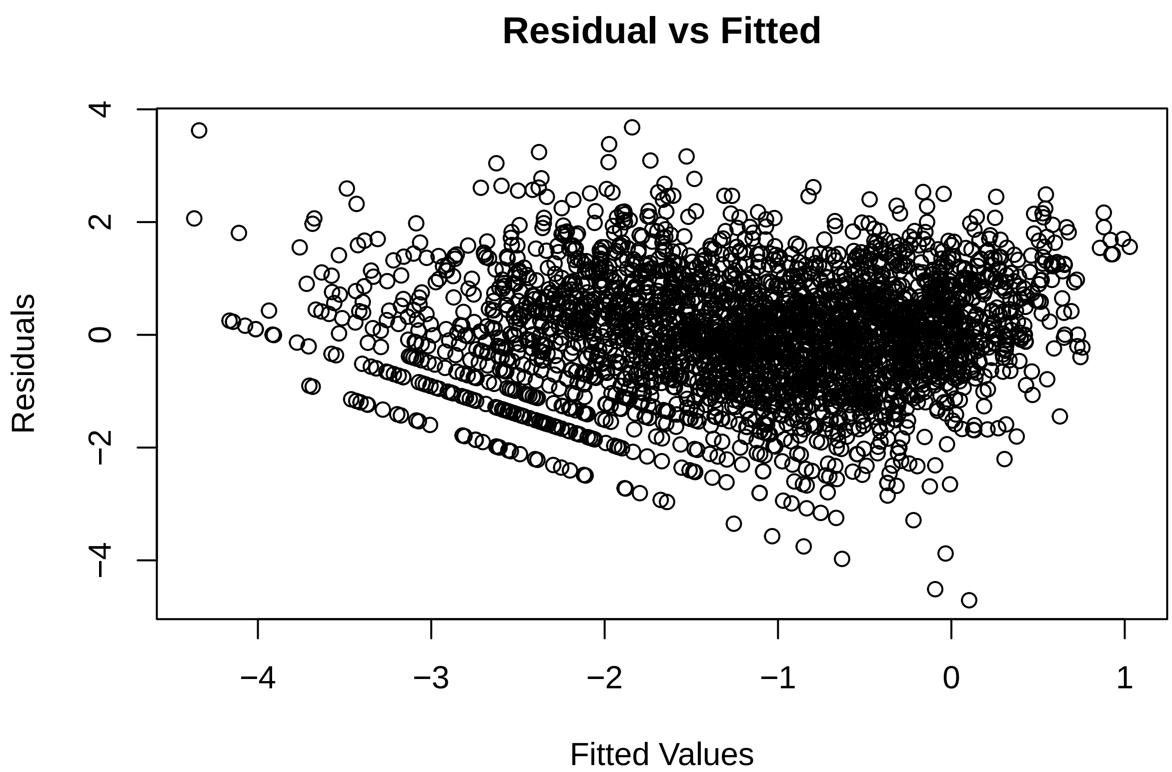
After fitting this new model, we once again check the MLR additional conditions and check for assumption violations.

```r
# condition 1
y_hat <- fitted(model_ln)
plot(x = y_hat, y = sales$Global_Sales, main="Response vs Fitted", xlab="Fitted Values", ylab="Global S
abline(a = 0, b = 1, lty=2)
```

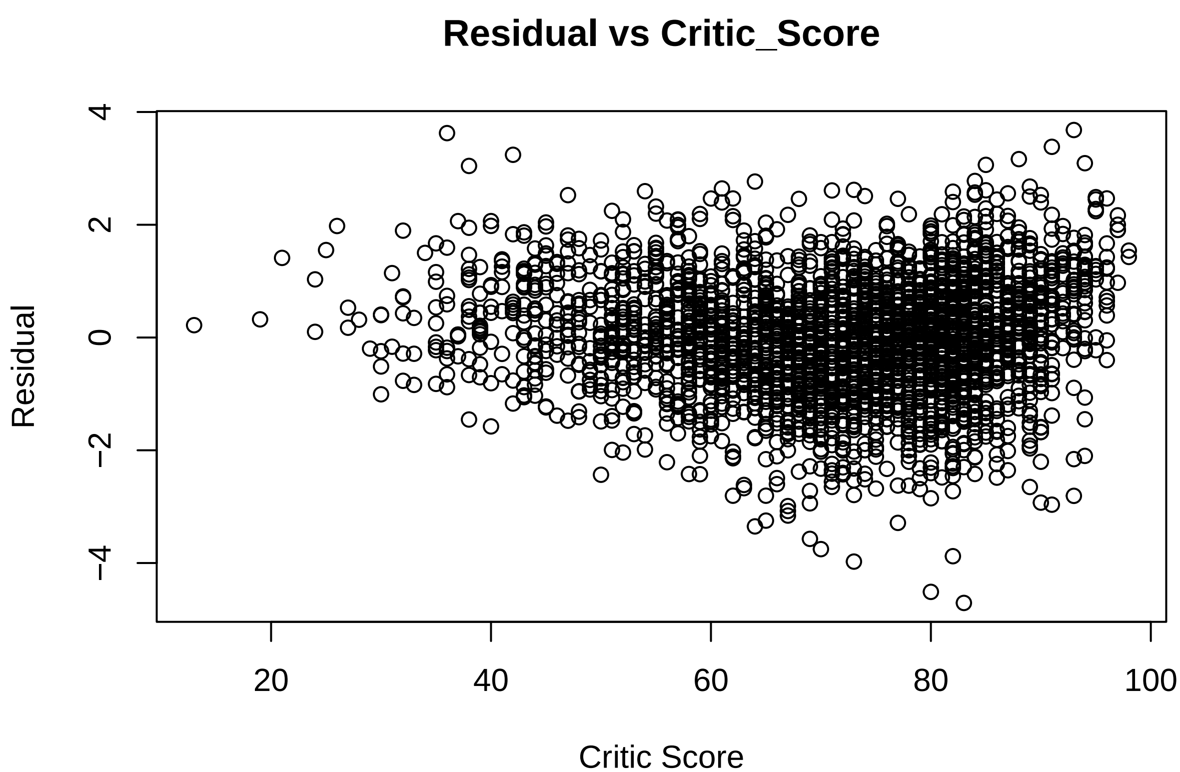


Based on this plot, we don't observe random diagonal scatter or an easily identifiable non-linear trend so the 1st condition does not seem to hold. As a result, the residual plots will not be reliable. Condition 2 still holds as previously shown. Now we check the assumptions one again.

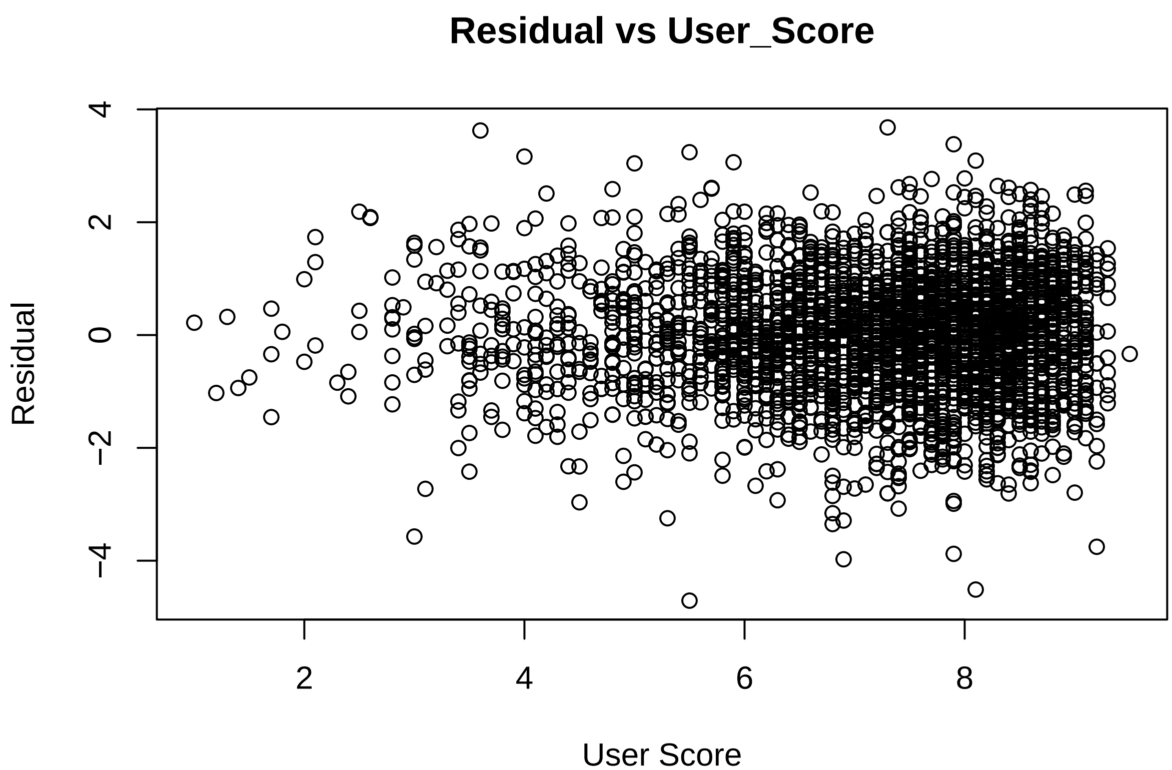
```r
# residuals versus fitted values
e_hat <- resid(model_ln)
plot(x =y_hat, y = e_hat, main="Residual vs Fitted", xlab="Fitted Values", ylab="Residuals")
```



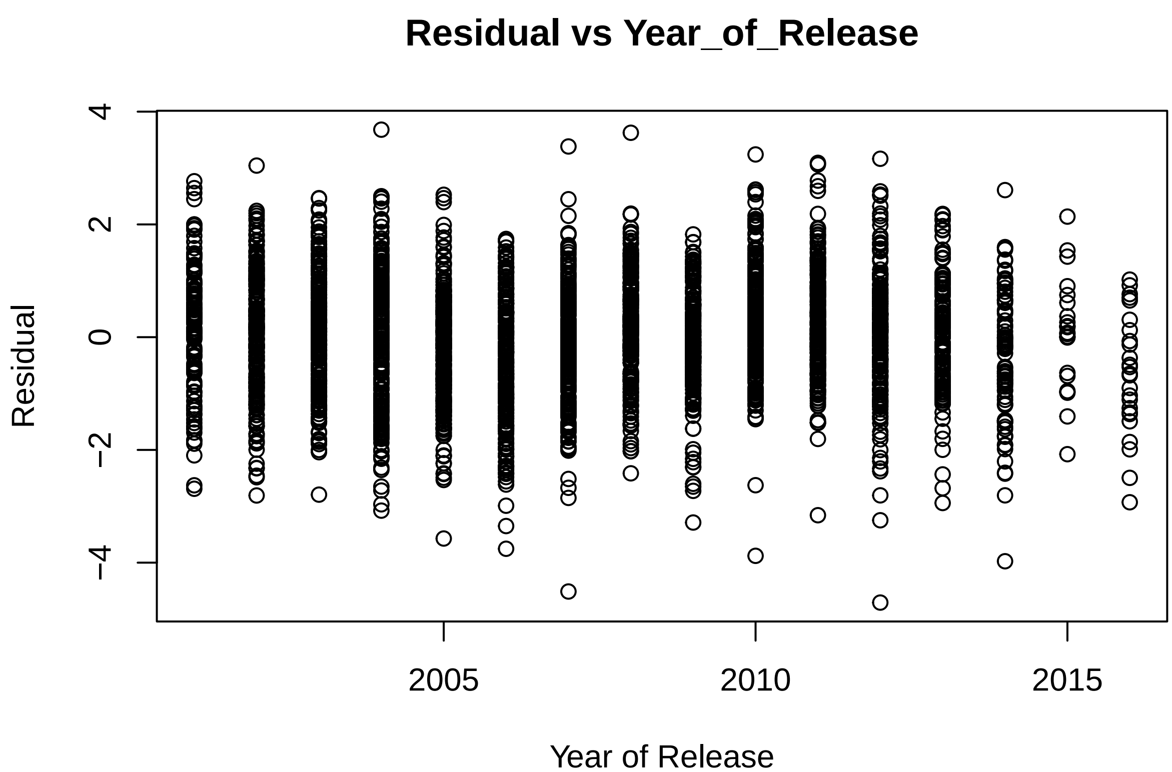

```r
# residual versus predictor plots for numerical variables
plot(x = sales$Critic_Score, y = e_hat, main="Residual vs Critic_Score", xlab="Critic Score", ylab="Resi
```



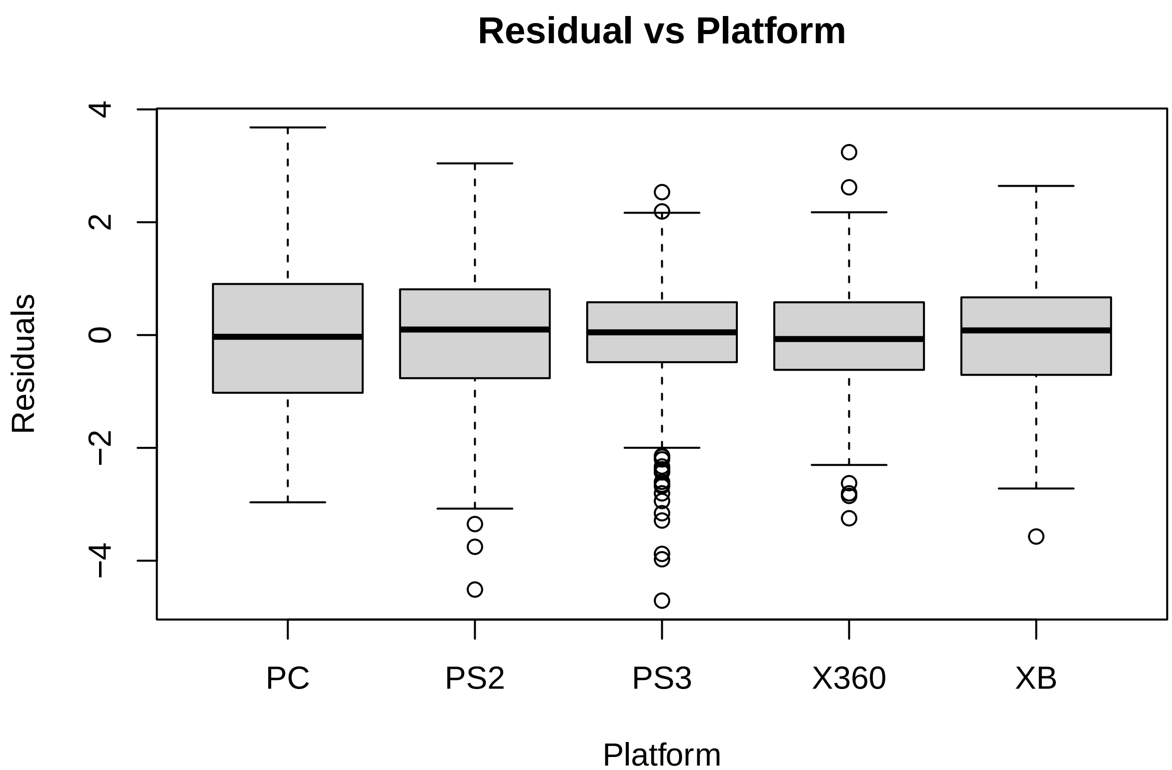

```r
plot(x = sales$User_Score, y = e_hat, main="Residual vs User_Score", xlab="User Score", ylab="Residual")
```



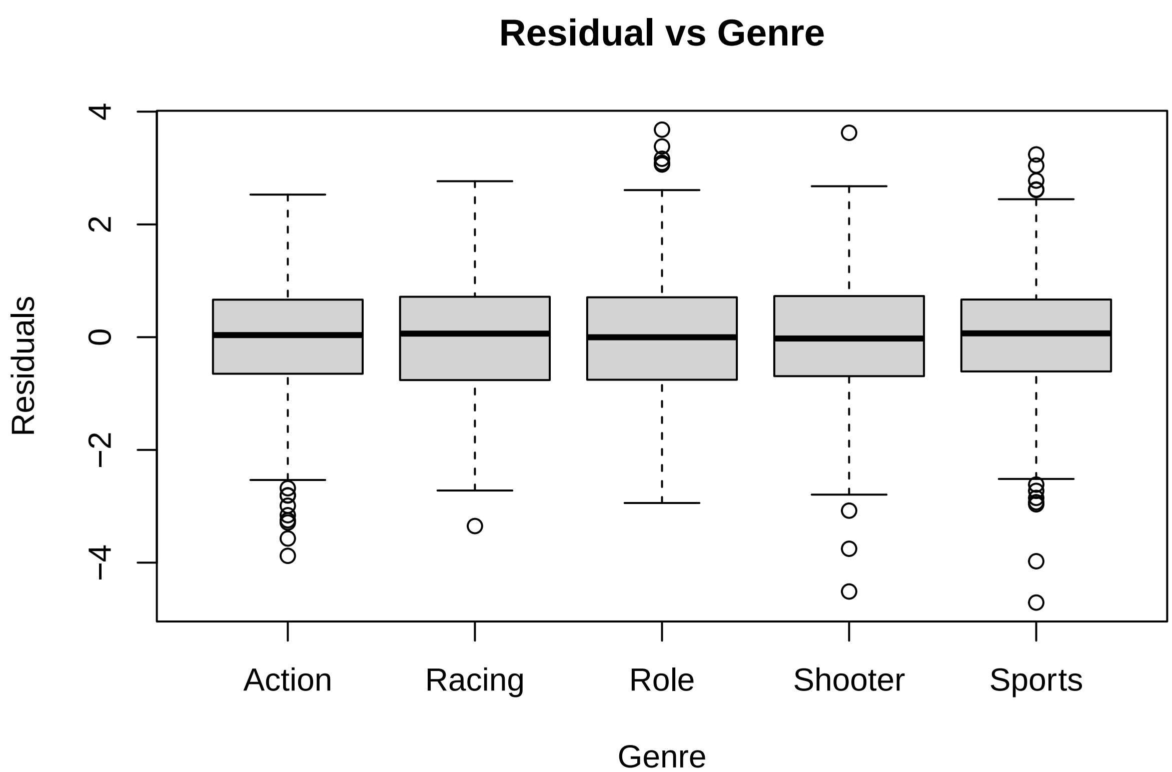

```
plot(x = sales$Year_of_Release, y = e_hat, main="Residual vs Year_of_Release", xlab="Year of Release", y
```



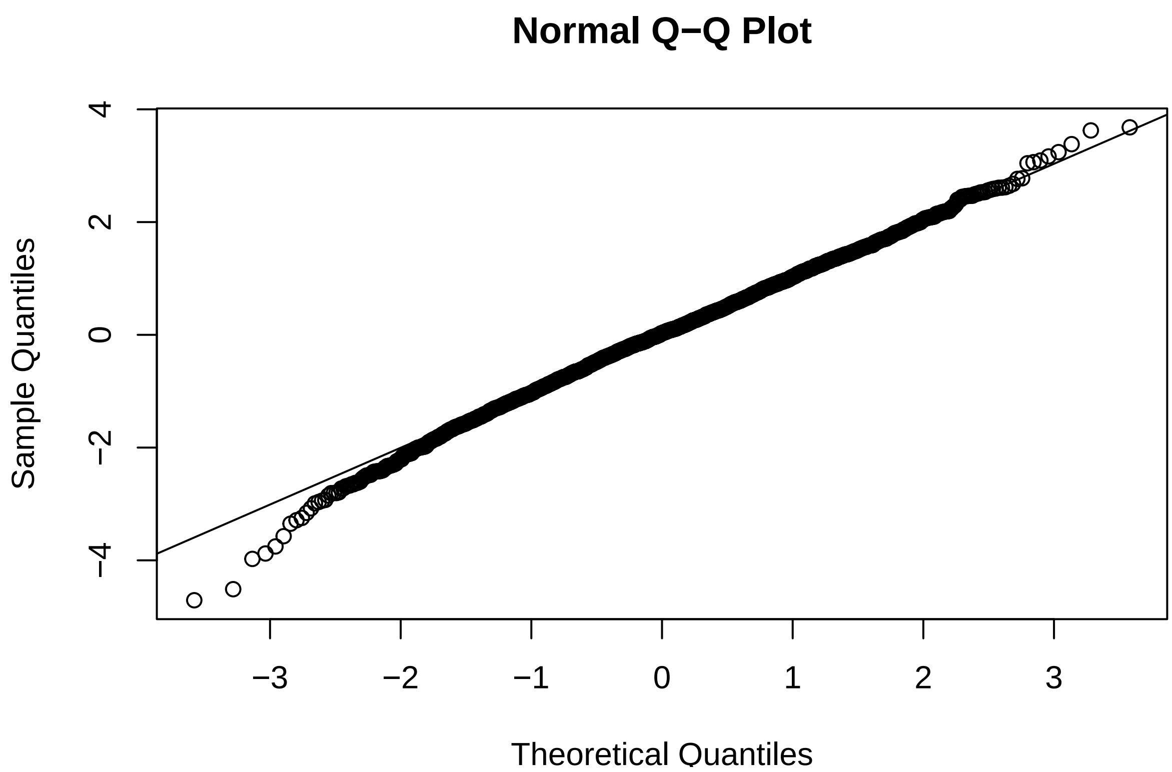

```
# residual plots for categorical predictors
boxplot(e_hat ~ sales$Platform , main="Residual vs Platform", xlab="Platform", ylab="Residuals")
```




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




```
# QQ plot
qqnorm(e_hat)
qqline(e_hat)
```



Based on the new plots we don't observe any assumption violations. Next we perform ANOVA test of overall significance to identify the existence of a linear relationship (null hypothesis: all slopes are zero).

```
summary(model_ln)
```

```
##
## Call:
## lm(formula = log(Global_Sales) ~ Critic_Score + User_Score +
##     Platform2 + Year_of_Release + Genre2, data = sales)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7079 -0.6677  0.0300  0.6913  3.6803
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -49.639739  17.484269  -2.839 0.004555 **
## Critic_Score      0.059145   0.001787  33.099  < 2e-16 ***
## User_Score       -0.132174   0.018119  -7.295 3.84e-13 ***
## Platform2PS2      2.259776   0.084532  26.733  < 2e-16 ***
## Platform2PS3      2.174448   0.067621  32.157  < 2e-16 ***
## Platform2X360     2.161901   0.067404  32.074  < 2e-16 ***
## Platform2XB       1.404005   0.091028  15.424  < 2e-16 ***
## Year_of_Release   0.021705   0.008690   2.498 0.012555 *
## Genre2Racing     -0.300751   0.064419  -4.669 3.17e-06 ***
## Genre2Role       -0.233194   0.066504  -3.506 0.000461 ***
## Genre2Shooter     0.064563   0.054163   1.192 0.233357
## Genre2Sports     -0.283759   0.056905  -4.987 6.51e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.045 on 2898 degrees of freedom
## Multiple R-squared:  0.4511, Adjusted R-squared:  0.449
## F-statistic: 216.5 on 11 and 2898 DF,  p-value: < 2.2e-16
```

From the summary table we can see that the p-value is 2.2e-16 which is less than $\alpha = 0.05$. So we reject the null and conclude a statistically significant linear relationship exists for at least one predictor.

Next, we perform hypothesis tests for individual coefficients in our model (with the null hypothesis being that the coefficient is 0).

```r
summary(model_ln)
```

```
##
## Call:
## lm(formula = log(Global_Sales) ~ Critic_Score + User_Score +
##     Platform2 + Year_of_Release + Genre2, data = sales)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7079 -0.6677  0.0300  0.6913  3.6803
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -49.639739  17.484269  -2.839 0.004555 **
## Critic_Score      0.059145   0.001787  33.099  < 2e-16 ***
## User_Score       -0.132174   0.018119  -7.295 3.84e-13 ***
## Platform2PS2      2.259776   0.084532  26.733  < 2e-16 ***
## Platform2PS3      2.174448   0.067621  32.157  < 2e-16 ***
## Platform2X360     2.161901   0.067404  32.074  < 2e-16 ***
## Platform2XB       1.404005   0.091028  15.424  < 2e-16 ***
## Year_of_Release   0.021705   0.008690   2.498 0.012555 *
## Genre2Racing     -0.300751   0.064419  -4.669 3.17e-06 ***
## Genre2Role       -0.233194   0.066504  -3.506 0.000461 ***
## Genre2Shooter     0.064563   0.054163   1.192 0.233357
## Genre2Sports     -0.283759   0.056905  -4.987 6.51e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.045 on 2898 degrees of freedom
## Multiple R-squared:  0.4511, Adjusted R-squared:  0.449
## F-statistic: 216.5 on 11 and 2898 DF,  p-value: < 2.2e-16
```

Based on the Pr(>|t|) column in the summary table we reject the null and claim a significant linear relationship exists for all the coefficients ($Pr(> |t|)$ is less than $\alpha = 0.05$ for all coefficients). As all coefficients are significant, we can't make a more reduced model to perform a partial F test and we choose the current model as our final one.