Regression

Jonathan Ho

September 23, 2022

Linear Regression

Linear regression is the concept of comparing two variables to observe their relationship. More specifically, the predictors x will be used to try and predict the target y. In linear regression, the target would be a quantitative value. Through the use of residuals along side statistical values, an analysis of the linear regression can be carried out to check if it is a good fit given the predictors and target values. Linear regression is relatively simple to understand, is powerful when the data supports a linear relationship, and it has low variance. However, a drawback would be the high bias it has due to the linear shape being assumed with the data.

Loading in the data

Loading in the csv file.

```
df <- read.csv("vgsales.csv")
str(df)
```

```
## 'data.frame':    16598 obs. of  11 variables:
##  $ Rank        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Name        : chr  "Wii Sports" "Super Mario Bros." "Mario Kart Wii" "Wii Sports Resort"
...
##  $ Platform    : chr  "Wii" "NES" "Wii" "Wii" ...
##  $ Year        : chr  "2006" "1985" "2008" "2009" ...
##  $ Genre       : chr  "Sports" "Platform" "Racing" "Sports" ...
##  $ Publisher   : chr  "Nintendo" "Nintendo" "Nintendo" "Nintendo" ...
##  $ NA_Sales    : num  41.5 29.1 15.8 15.8 11.3 ...
##  $ EU_Sales    : num  29.02 3.58 12.88 11.01 8.89 ...
##  $ JP_Sales    : num  3.77 6.81 3.79 3.28 10.22 ...
##  $ Other_Sales : num  8.46 0.77 3.31 2.96 1 0.58 2.9 2.85 2.26 0.47 ...
##  $ Global_Sales: num  82.7 40.2 35.8 33 31.4 ...
```

Divding up the data

Splitting the data into 80/20 train/test. There were two rows within the dataset that were corrupted or wrong, so the removal of them was necessary.

```
# Remove two genres that are outliers
new_df <- df[!grepl("Sony Computer Entertainment", df$Genre),]
new_df <- new_df[!grepl("Idea Factory", new_df$Genre),]

set.seed(1234)
x <- sample(1:nrow(new_df), nrow(new_df)*0.8, replace=FALSE)
train <- new_df[x,]
test <- new_df[-x,]
```

Data exploration

1. See all the columns and what data type they are.

```
str(train)
```

```
## 'data.frame':	13276 obs. of  11 variables:
##  $ Rank        : int  7453 8017 7163 8087 9197 623 15245 10886 935 12690 ...
##  $ Name        : chr  "Mortal Kombat: Special Forces" "Reel Fishing II" "Dark Souls II" "Batt
le Commander: Hachibushu Shura no Heihou" ...
##  $ Platform    : chr  "PS" "PS" "XOne" "SNES" ...
##  $ Year        : chr  "2000" "2000" "2015" "1991" ...
##  $ Genre       : chr  "Fighting" "Sports" "Role-Playing" "Strategy" ...
##  $ Publisher   : chr  "Midway Games" "Victor Interactive" "Namco Bandai Games" "Banpresto"
...
##  $ NA_Sales    : num  0.12 0.1 0.13 0 0.11 1.15 0.02 0.09 0.85 0 ...
##  $ EU_Sales    : num  0.08 0.07 0.07 0 0.03 1.14 0 0 0.71 0.05 ...
##  $ JP_Sales    : num  0 0 0 0.18 0 0.06 0 0 0.13 0 ...
##  $ Other_Sales : num  0.01 0.01 0.02 0 0 0.13 0 0.01 0.16 0.01 ...
##  $ Global_Sales: num  0.21 0.18 0.22 0.18 0.14 2.48 0.02 0.09 1.86 0.06 ...
```

2. First and last six rows of each column.

```
head(train)
```

	R… <int>	Name <chr>	Platform <chr>	Y… <chr>	Genre <chr>
7452	7453	Mortal Kombat: Special Forces	PS	2000	Fighting
8016	8017	Reel Fishing II	PS	2000	Sports
7162	7163	Dark Souls II	XOne	2015	Role-Playing
8086	8087	Battle Commander: Hachibushu Shura no Heihou	SNES	1991	Strategy
9196	9197	NFL Blitz 20-03	GC	2002	Sports
623	623	Tomb Raider: The Last Revelation	PS	1998	Action

6 rows | 1-6 of 12 columns

```
tail(train)
```

	Rank <int>	Name <chr>	Platform <chr>	Y... <chr>	Genre <chr>	Publisher <chr>
16418	16420	th!nk Logic Trainer	Wii	2009	Puzzle	Conspiracy Entertainment
12963	12964	Cities in Motion	PC	2011	Simulation	Paradox Interactive
13301	13302	Gummy Bears Mini Golf	DS	2010	Sports	Storm City Games
11025	11026	Hello Kitty's Cube Frenzy	PS	1998	Puzzle	Culture Publishers
14726	14728	Auto Modellista	GC	2003	Racing	Capcom
11750	11751	Super Baseball	2600	1987	Sports	Atari

6 rows | 1-8 of 12 columns

3. Checking how many NAs there are.

```
sapply(train, function(y) sum(is.na(y)))
```

```
##         Rank         Name     Platform         Year        Genre    Publisher 
##            0            0            0            0            0            0 
##     NA_Sales     EU_Sales     JP_Sales  Other_Sales Global_Sales 
##            0            0            0            0            0
```

4. Display how many rows and columns there are in the data.

```
dim(train)
```

```
## [1] 13276    11
```

5. Summary of the statistics of each column.

```
summary(train)
```

```
##       Rank           Name             Platform            Year          
##  Min.   :    1   Length:13276       Length:13276       Length:13276      
##  1st Qu.: 4164   Class :character   Class :character   Class :character  
##  Median : 8352   Mode  :character   Mode  :character   Mode  :character  
##  Mean   : 8326                                                           
##  3rd Qu.:12475                                                           
##  Max.   :16600                                                           
##     Genre            Publisher            NA_Sales         EU_Sales      
##  Length:13276       Length:13276       Min.   : 0.000   Min.   : 0.0000  
##  Class :character   Class :character   1st Qu.: 0.000   1st Qu.: 0.0000  
##  Mode  :character   Mode  :character   Median : 0.080   Median : 0.0200  
##                                        Mean   : 0.262   Mean   : 0.1448  
##                                        3rd Qu.: 0.240   3rd Qu.: 0.1100  
##                                        Max.   :41.490   Max.   :29.0200  
##     JP_Sales         Other_Sales        Global_Sales    
##  Min.   : 0.00000   Min.   : 0.00000   Min.   : 0.0100  
##  1st Qu.: 0.00000   1st Qu.: 0.00000   1st Qu.: 0.0600  
##  Median : 0.00000   Median : 0.01000   Median : 0.1700  
##  Mean   : 0.07772   Mean   : 0.04825   Mean   : 0.5346  
##  3rd Qu.: 0.04000   3rd Qu.: 0.03000   3rd Qu.: 0.4700  
##  Max.   :10.22000   Max.   :10.57000   Max.   :82.7400  
```

6. See all the qualitative descriptions in the Genre column for the train and test.

```
table(train$Genre)
```

```
## 
##       Action    Adventure     Fighting         Misc     Platform       Puzzle 
##         2637         1039          675         1397          705          465 
##       Racing Role-Playing      Shooter   Simulation       Sports     Strategy 
##          998         1191         1056          708         1858          547 
```

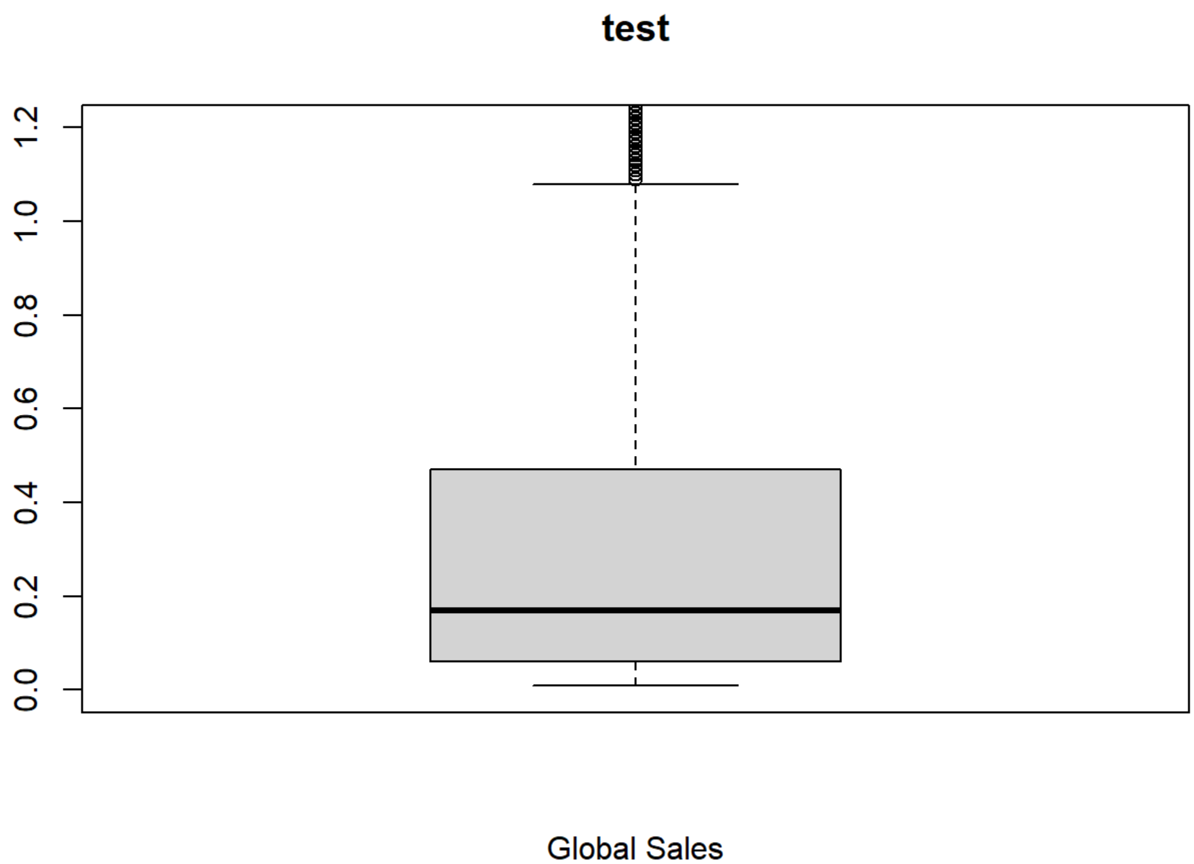
```
table(test$Genre)
```

```
## 
##       Action    Adventure     Fighting         Misc     Platform       Puzzle 
##          679          245          173          342          181          117 
##       Racing Role-Playing      Shooter   Simulation       Sports     Strategy 
##          251          297          254          159          488          134 
```

Data visualization

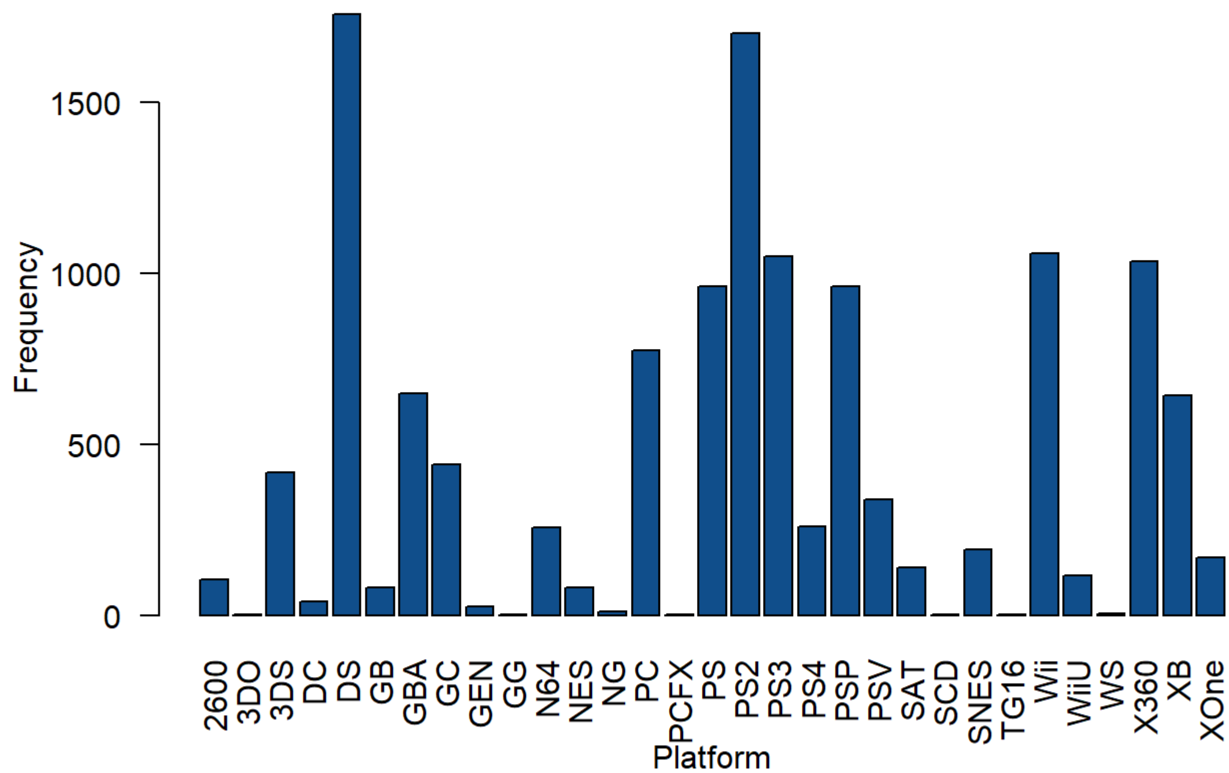
Box plot of the Global_Sales. Within the dataset, there are a good amount of numbers outside the middle half of the sample making it seem like there are a lot of outliers.

```
boxplot(train$Global_Sales, xlab="Global Sales", main="test", ylim=c(0,1.2))
```



A bar plot of Platform to see the frequency of each within the dataset.

```
counts <- table(train$Platform)
barplot(counts, xlab="Platform", ylab="Frequency", col="dodgerblue4", las=2) # las=2 displays al
l the Platforms
```

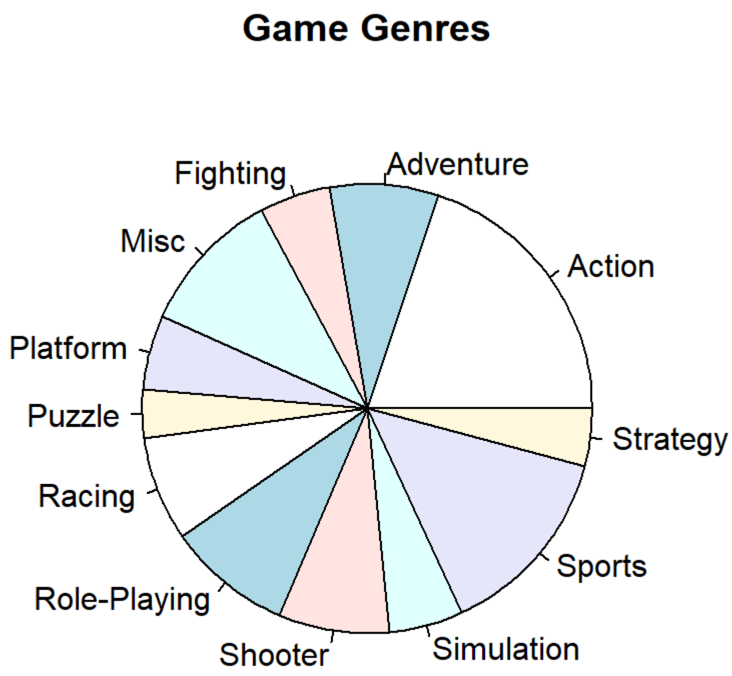


A pie chart of Genre to visually see the distribution of game genres.

```
slices <- c(sum(train$Genre=="Action"), sum(train$Genre=="Adventure"), sum(train$Genre=="Fightin
g"), sum(train$Genre=="Misc"), sum(train$Genre=="Platform"), sum(train$Genre=="Puzzle"), sum(tra
in$Genre=="Racing"), sum(train$Genre=="Role-Playing"), sum(train$Genre=="Shooter"), sum(train$Ge
nre=="Simulation"), sum(train$Genre=="Sports"), sum(train$Genre=="Strategy"))

lbls <- c("Action", "Adventure", "Fighting", "Misc", "Platform", "Puzzle", "Racing", "Role-Playi
ng", "Shooter", "Simulation", "Sports", "Strategy")

pie(slices, labels=lbls, main="Game Genres")
```



Simple linear regression

A single predictor is used to see the impact of Genre on Global_Sales.

```
lm1 <- lm(Global_Sales~Genre, data=train)
summary(lm1)
```

```
##
## Call:
## lm(formula = Global_Sales ~ Genre, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.868 -0.462 -0.308 -0.042 82.145
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.53198    0.03031  17.549  < 2e-16 ***
## GenreAdventure    -0.33847    0.05702  -5.936 2.99e-09 ***
## GenreFighting     -0.02819    0.06715  -0.420 0.674652
## GenreMisc         -0.07257    0.05151  -1.409 0.158891
## GenrePlatform      0.34567    0.06600   5.237 1.65e-07 ***
## GenrePuzzle       -0.08888    0.07829  -1.135 0.256292
## GenreRacing        0.01634    0.05785   0.282 0.777645
## GenreRole-Playing  0.10233    0.05435   1.883 0.059733 .
## GenreShooter       0.25353    0.05669   4.472 7.80e-06 ***
## GenreSimulation   -0.09978    0.06589  -1.514 0.129972
## GenreSports        0.06256    0.04715   1.327 0.184597
## GenreStrategy     -0.26765    0.07313  -3.660 0.000254 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.557 on 13264 degrees of freedom
## Multiple R-squared:  0.01074,    Adjusted R-squared:  0.009917
## F-statistic: 13.09 on 11 and 13264 DF,  p-value: < 2.2e-16
```

When the summary() function was called, it displayed what the call was for the linear model, residuals, coefficients, as well as a few other value pertaining to the target and predictor. Information within the residuals display a bit of data exploration statistics including the min, max, median, and the first and third quarters.

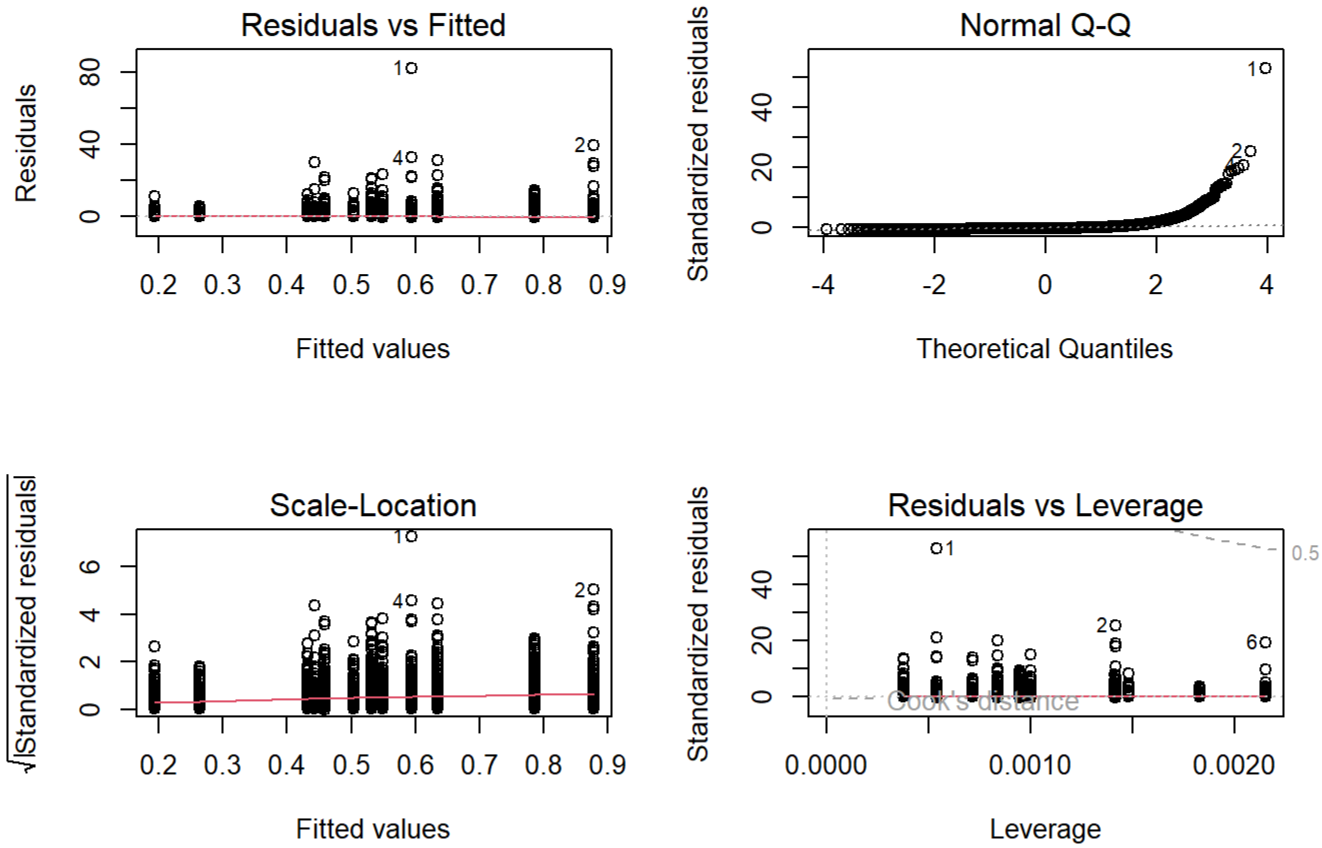
In coefficients are the intercept as well as predictors from the Genre column. An estimated coefficient, standard error, t-value, and p-value are also shown. There are asterisks next to Adventure, Platform, Shooter, and Strategy which suggests that those four genres may be good predictors of Global_Sales. Standard error is a variation of the estimate given for each predictor and the intercept. The t-value signifies how little of a relationship two variables would have, or how true the null hypothesis is. However, the p-value tells the opposite. With a low p-value, the null hypothesis can be rejected meaning that there is a potential relationship between the two variables.

The last part of summary are values that portray how well each coefficient modeled the true data. It seems that the model is off by about 1.6 million copies. An R squared value is also given, which is 0.009917. This implies that Genre most likely does not have a relationship with Global_Sales. However, the F-statistic (accounts for all predictors) is higher than 11 and the p-value is low, so this model may display some kind of significance.

Plotting the residuals

The following are plotting the residuals generated from lm().

```
par(mfrow=c(2,2))
plot(lm1)
```



Starting with the Residuals vs Fitted plot, there technically is a linear pattern. However, the residuals are not spread out evenly and well, but rather clump up in columns at particular values. This may mean that a non-linear pattern can be present. The next plot is the Normal Q-Q which signifies if the residuals follow a straight line well. It seems that way at first, until around the second quantile. The residuals start to significantly deviate from the line which is not ideal. So, the residuals may not be normally distributed. The Scale-Location plot is used to check if residuals are spread equally within the range of predictors. However, like the first plot, the problem lies with how the residuals stack in columns. This could indicate that the predictor of genre may not predict this well. Lastly is the residuals vs leverage plot which indicates if any extremities could influence the regression line. Oddly enough, in this case, none of the residuals appears outside Cook's distance lines. This implies that none of the residuals would affect how the linear regression would be formed.

Multiple linear regression

Multiple linear regression is carried out in an attempt to make the data better. Two predictors are used instead by including Platform into the model. In doing so, warnings are generated about leverages being one. However, these warnings are ignored.

```
lm2 <- lm(Global_Sales~Genre+Platform, data=train)
summary(lm2)
```

```
##
## Call:
## lm(formula = Global_Sales ~ Genre + Platform, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.818 -0.453 -0.240  0.014 81.988
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.69666    0.15252   4.568 4.97e-06 ***
## GenreAdventure    -0.26986    0.05704  -4.731 2.26e-06 ***
## GenreFighting     -0.02836    0.06735  -0.421  0.67369
## GenreMisc         -0.07564    0.05158  -1.467  0.14250
## GenrePlatform      0.29143    0.06641   4.389 1.15e-05 ***
## GenrePuzzle       -0.11820    0.07890  -1.498  0.13412
## GenreRacing        0.02011    0.05780   0.348  0.72797
## GenreRole-Playing  0.11139    0.05421   2.055  0.03990 *
## GenreShooter       0.24288    0.05659   4.292 1.78e-05 ***
## GenreSimulation   -0.06234    0.06615  -0.942  0.34598
## GenreSports        0.02670    0.04720   0.566  0.57163
## GenreStrategy     -0.19144    0.07379  -2.594  0.00949 **
## Platform3DO       -0.46263    1.09500  -0.422  0.67267
## Platform3DS       -0.19659    0.16914  -1.162  0.24514
## PlatformDC        -0.36316    0.28398  -1.279  0.20098
## PlatformDS        -0.29076    0.15630  -1.860  0.06288 .
## PlatformGB         1.90021    0.22935   8.285  < 2e-16 ***
## PlatformGBA       -0.32820    0.16317  -2.011  0.04430 *
## PlatformGC        -0.39778    0.16826  -2.364  0.01809 *
## PlatformGEN        0.28180    0.34814   0.809  0.41827
## PlatformGG        -0.94809    1.54114  -0.615  0.53844
## PlatformN64       -0.02068    0.17948  -0.115  0.90828
## PlatformNES        1.66099    0.22907   7.251 4.37e-13 ***
## PlatformNG        -0.56380    0.51070  -1.104  0.26962
## PlatformPC        -0.41029    0.16183  -2.535  0.01125 *
## PlatformPCFX      -0.77805    1.54080  -0.505  0.61359
## PlatformPS        -0.09624    0.15960  -0.603  0.54651
## PlatformPS2       -0.11348    0.15621  -0.726  0.46755
## PlatformPS3        0.00384    0.15863   0.024  0.98069
## PlatformPS4        0.19940    0.17875   1.116  0.26464
## PlatformPSP       -0.43679    0.15978  -2.734  0.00627 **
## PlatformPSV       -0.51230    0.17331  -2.956  0.00312 **
## PlatformSAT       -0.49615    0.19964  -2.485  0.01296 *
## PlatformSCD       -0.57102    1.54066  -0.371  0.71092
## PlatformSNES      -0.02773    0.18837  -0.147  0.88294
## PlatformTG16      -0.28680    1.54090  -0.186  0.85235
## PlatformWii        0.02881    0.15884   0.181  0.85606
## PlatformWiiU      -0.12558    0.20817  -0.603  0.54633
## PlatformWS        -0.47414    0.78232  -0.606  0.54448
## PlatformX360       0.00931    0.15871   0.059  0.95322
## PlatformXB        -0.44202    0.16314  -2.709  0.00675 **
## PlatformXOne      -0.05645    0.19162  -0.295  0.76831
```

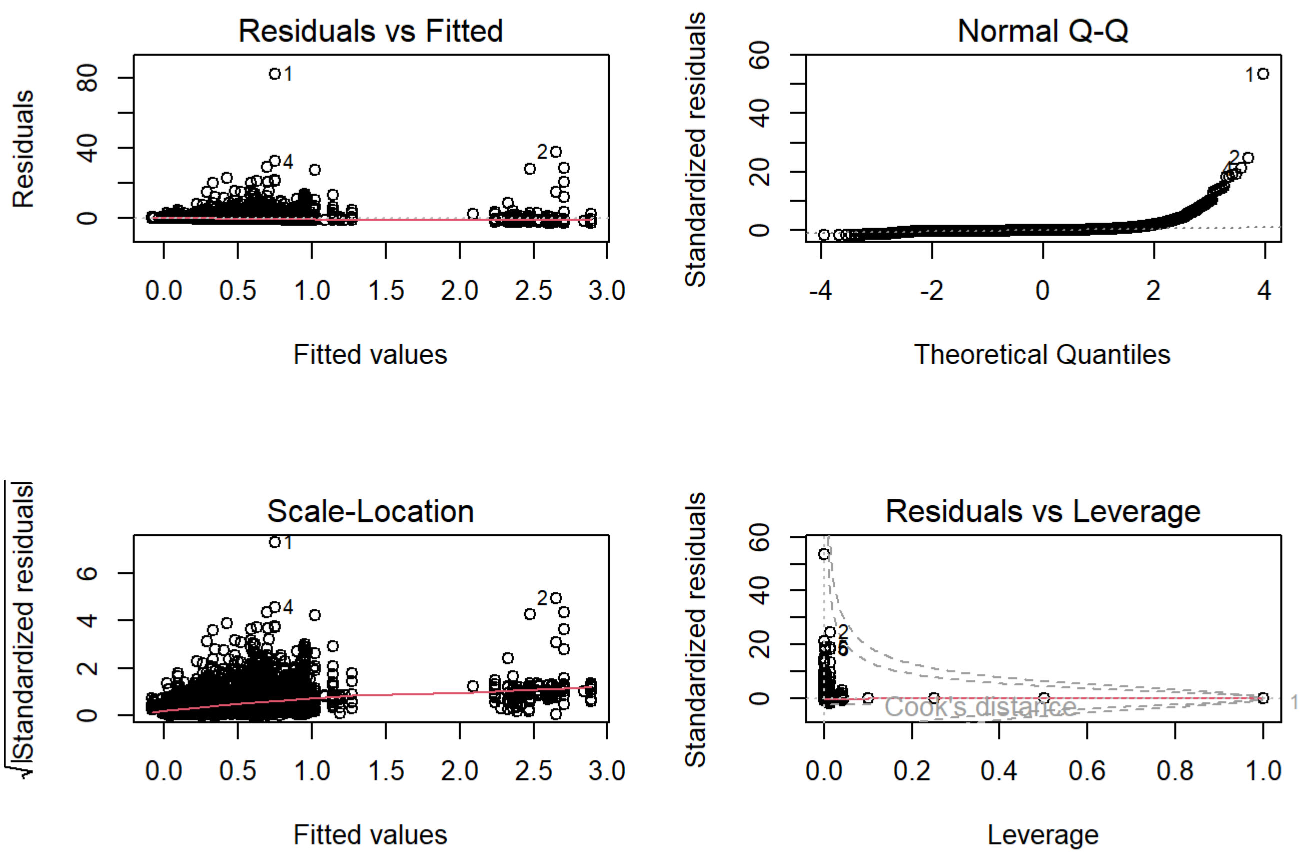
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.533 on 13234 degrees of freedom
## Multiple R-squared:  0.04321,    Adjusted R-squared:  0.04025
## F-statistic: 14.58 on 41 and 13234 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm2)
```

```
## Warning: not plotting observations with leverage one:
##   2214, 2767
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



Linear regression improvement attempt

In a third attempt to better the data, JP_Sales was used instead as there seems to be less outliers associated with it. This was in an attempt to check if the Global_Sales was a problem. Warnings were also issued and ignored.

```
lm3 <- lm(JP_Sales~Platform+Genre, data=train) #Tried an interaction between platform and Genre
summary(lm3)
```

```
##
## Call:
## lm(formula = JP_Sales ~ Platform + Genre, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0969 -0.0631 -0.0376  0.0078  9.2587
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.0010110  0.0282302   0.036  0.97143
## Platform3DO        0.0491214  0.2026798   0.242  0.80851
## Platform3DS        0.1707898  0.0313064   5.455 4.97e-08 ***
## PlatformDC         0.1484749  0.0525630   2.825  0.00474 **
## PlatformDS         0.0623856  0.0289311   2.156  0.03107 *
## PlatformGB         0.8085782  0.0424515  19.047  < 2e-16 ***
## PlatformGBA        0.0439586  0.0302015   1.456  0.14555
## PlatformGC         0.0240793  0.0311437   0.773  0.43944
## PlatformGEN        0.0740447  0.0644388   1.149  0.25055
## PlatformGG         0.0120412  0.2852583   0.042  0.96633
## PlatformN64        0.1036536  0.0332204   3.120  0.00181 **
## PlatformNES        1.0542246  0.0424005  24.864  < 2e-16 ***
## PlatformNG         0.0849948  0.0945279   0.899  0.36859
## PlatformPC        -0.0196797  0.0299540  -0.657  0.51119
## PlatformPCFX      -0.1226870  0.2851960  -0.430  0.66707
## PlatformPS         0.0965209  0.0295421   3.267  0.00109 **
## PlatformPS2        0.0498715  0.0289133   1.725  0.08458 .
## PlatformPS3        0.0469945  0.0293626   1.600  0.10951
## PlatformPS4        0.0249116  0.0330851   0.753  0.45149
## PlatformPSP        0.0313156  0.0295744   1.059  0.28968
## PlatformPSV        0.0211725  0.0320784   0.660  0.50925
## PlatformSAT        0.1594190  0.0369532   4.314 1.61e-05 ***
## PlatformSCD        0.0392584  0.2851701   0.138  0.89051
## PlatformSNES       0.4130571  0.0348658  11.847  < 2e-16 ***
## PlatformTG16       0.1503946  0.2852149   0.527  0.59799
## PlatformWii        0.0479449  0.0294004   1.631  0.10296
## PlatformWiiU       0.0899884  0.0385308   2.335  0.01953 *
## PlatformWS         0.0976776  0.1448036   0.675  0.49997
## PlatformX360      -0.0006301  0.0293757  -0.021  0.98289
## PlatformXB        -0.0036254  0.0301966  -0.120  0.90444
## PlatformXOne      -0.0075311  0.0354688  -0.212  0.83185
## GenreAdventure    -0.0114056  0.0105582  -1.080  0.28005
## GenreFighting      0.0270406  0.0124670   2.169  0.03010 *
## GenreMisc         -0.0002694  0.0095467  -0.028  0.97749
## GenrePlatform      0.0269478  0.0122912   2.192  0.02837 *
## GenrePuzzle       -0.0088593  0.0146031  -0.607  0.54408
## GenreRacing       -0.0082239  0.0106987  -0.769  0.44210
## GenreRole-Playing  0.1516760  0.0100334  15.117  < 2e-16 ***
## GenreShooter      -0.0089410  0.0104745  -0.854  0.39334
## GenreSimulation    0.0184207  0.0122434   1.505  0.13247
## GenreSports       -0.0034230  0.0087371  -0.392  0.69523
## GenreStrategy      0.0159468  0.0136583   1.168  0.24301
```

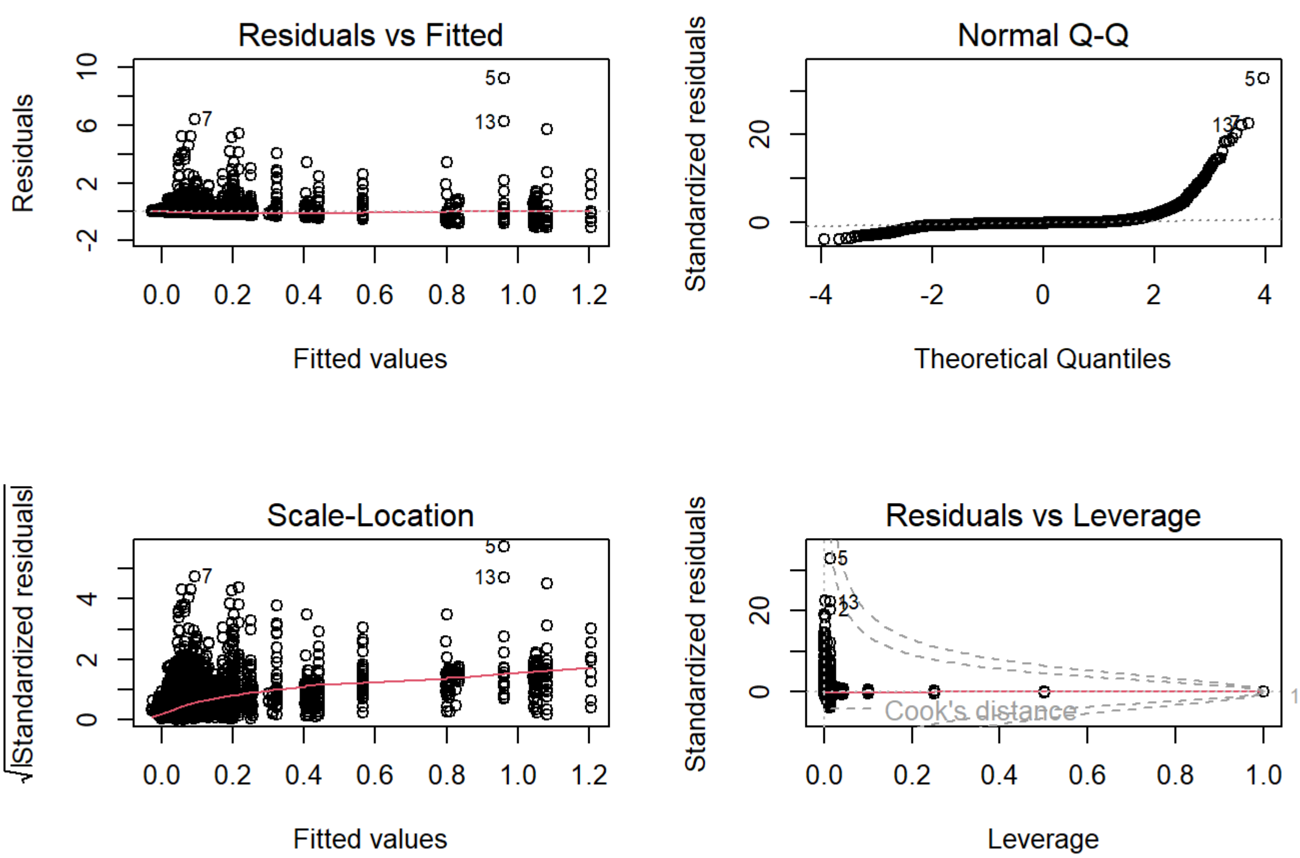
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2837 on 13234 degrees of freedom
## Multiple R-squared:  0.163,  Adjusted R-squared:  0.1604
## F-statistic: 62.88 on 41 and 13234 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm3)
```

```
## Warning: not plotting observations with leverage one:
##   2767, 6391
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



Results

Between the three models, models two and three looks better in terms of the first and third plot. Unlike the first plot, the residuals are less consolidated and more spread out. At the same time though, it is not a drastic improvement over plot one as there is still some clumping in the lower and upper end of the fitted values for plot two. In plot three it is just the lower end. I believe before any more analysis would be done, the third model could be eliminated just because the data did not seem to be any better for JP_Sales over Global_Sales in the second model. I believe the second model would be better over the first as there is more data to work with in the first and third plots. However, there is a bit of concern with plot two as the lower end does not fit the linear line. Even more concern is expressed with the fourth plot since the residual 1 is at the tip of the Cook's distance, meaning that there could be something affecting the linear regression.

Correlation and MSE

Model 1 Evaluation

```
pred1 <- predict(lm1, newdata=test)
cor1 <- cor(pred1, test$Global_Sales)
mse1 <- mean((pred1-test$Global_Sales)^2)
rmse1 <- sqrt(mse1)

print(paste('correlation:', cor1))
```

```
## [1] "correlation: 0.127974758171416"
```

```
print(paste('mse:', mse1))
```

```
## [1] "mse: 2.265806009761"
```

```
print(paste('rmse:', rmse1))
```

```
## [1] "rmse: 1.50525944931796"
```

Model 2 Evaluation

```
pred2 <- predict(lm2, newdata=test)
cor2 <- cor(pred2, test$Global_Sales)
mse2 <- mean((pred2-test$Global_Sales)^2)
rmse2 <- sqrt(mse2)

print(paste('correlation:', cor2))
```

```
## [1] "correlation: 0.218379234764752"
```

```
print(paste('mse:', mse2))
```

```
## [1] "mse: 2.19279293135131"
```

```
print(paste('rmse:', rmse2))
```

```
## [1] "rmse: 1.48080820208132"
```

Model 3 Evaluation

```
pred3 <- predict(lm3, newdata=test)
cor3 <- cor(pred3, test$Global_Sales)
mse3 <- mean((pred3-test$Global_Sales)^2)
rmse3 <- sqrt(mse3)

print(paste('correlation:', cor3))
```

```
## [1] "correlation: 0.152110255322097"
```

```
print(paste('mse:', mse3))
```

```
## [1] "mse: 2.48497871004454"
```

```
print(paste('rmse:', rmse3))
```

```
## [1] "rmse: 1.57638152426516"
```

Between all three models, it seems like model two has the highest correlation, but it is still a bad correlation being only approximately 0.22. Despite being a poor model, model two is still the best as comparing the mse and rmse values with model one, they are both slightly lower. This implies that model two is just a slightly better fit of a model for Global_Sales. I think this may have occurred because with including platform as another predictor in model two, it gave a few more predictors that were good for the model thus helping to support correlation to Global_Sales.