# Final Project

Ryan Podzielny, Christopher Hainzl, Christopher Barbieri, Christopher Hakkenberg, Peter Bitanga

2022-12-16

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggridges)
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.2
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
PS4 <- read.csv("PS4_GamesSales.csv")
XboxOne <- read.csv("XboxOne_GameSales.csv")
sales <- read.csv("Video_Games_Sales_as_at_22_Dec_2016.csv")

# Checking for missing values
colSums(is.na(PS4))
```

```
##          Game          Year         Genre     Publisher North.America
##             0             0             0             0             0
##        Europe         Japan Rest.of.World        Global
##             0             0             0             0
```

```
colSums(is.na(XboxOne))
```

```
##           Pos          Game          Year         Genre     Publisher
##             0             0             0             0             0
## North.America        Europe         Japan Rest.of.World        Global
##             0             0             0             0             0
```

```r
colSums(is.na(sales))
```

```
##            Name        Platform Year_of_Release           Genre       Publisher
##               0               0               0               0               0
##        NA_Sales        EU_Sales        JP_Sales     Other_Sales    Global_Sales
##               0               0               0               0               0
##    Critic_Score    Critic_Count      User_Score      User_Count       Developer
##            8582            8582            9129            9129               0
##          Rating
##               0
```

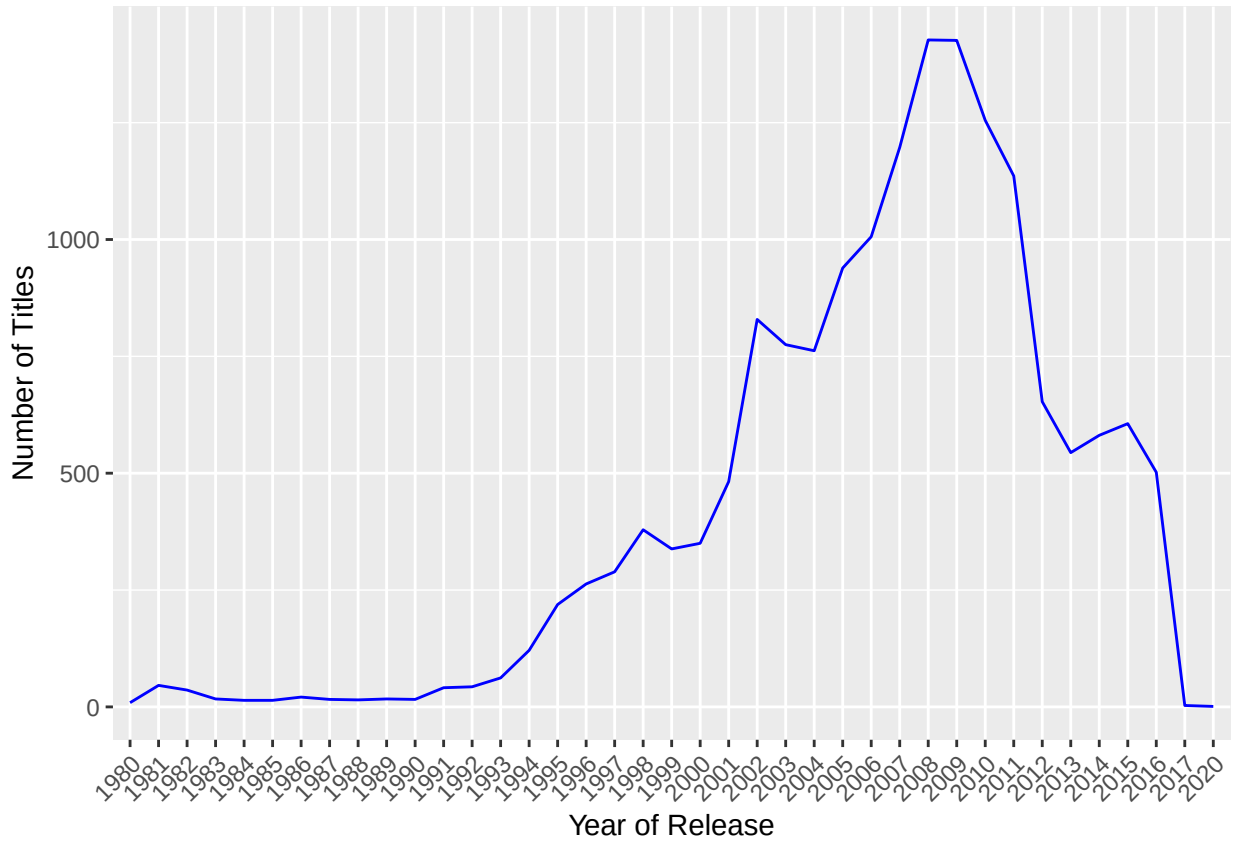
```r
dim(sales)
```

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

```r
# Removing the missing data from years (only 19 cases)
glimpse(sales$Year_of_Release)
```

```
##  chr [1:16719] "2006" "1985" "2008" "2009" "1996" "1989" "2006" "2006" ...
```

```r
counts <- table(sales$Year_of_Release)
counts <- as.data.frame(counts)
counts <- counts %>% filter(Var1 != "N/A")

# Line graph of publications over the years
p <- ggplot(data=counts, aes(x=Var1, y=Freq, group=1)) +
  geom_line(color="blue") +
  xlab("Year of Release") +
  ylab("Number of Titles") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
p
```




```
# Basically get the highest sales
pubs_count <- table(sales$Publisher)
pubs_count <- as.data.frame(pubs_count)

colnames(pubs_count) <- c("pubs", "total_games")

pubs <- c(pubs_count$pubs)

sale_total <- c()

for (x in pubs) {
  currPub <- sales %>% filter(Publisher == x)
  total <- sum(currPub$Global_Sales)
  sale_total <- append(sale_total, total)
}

sale_numbers <- data.frame(pubs, sale_total)
```

```
# Total sales and games published by companies we are looking at
sale_numbers <- sale_numbers %>% arrange(desc(sale_total))
pubs_count <- pubs_count %>% arrange(desc(total_games))

head(sale_numbers)
```

```
##                         pubs sale_total
## 1                   Nintendo    1788.81
## 2             Electronic Arts    1116.96
```

```
## 3                  Activision     731.16
## 4 Sony Computer Entertainment     606.48
## 5                     Ubisoft     471.61
## 6        Take-Two Interactive     403.82
```

```
head(pubs_count)
```

```
##                           pubs total_games
## 1              Electronic Arts        1356
## 2                   Activision         985
## 3           Namco Bandai Games         939
## 4                      Ubisoft         933
## 5 Konami Digital Entertainment         834
## 6                          THQ         715
```

```
pubs_wanted = c("Nintendo", "Electronic Arts", "Microsoft Game Studios", "Sony Computer Entertainment")

looking <- sale_numbers %>% left_join(pubs_count, by="pubs")
looking <- looking %>% filter(pubs %in% pubs_wanted)
looking
```

```
##                          pubs sale_total total_games
## 1                    Nintendo    1788.81         706
## 2             Electronic Arts    1116.96        1356
## 3 Sony Computer Entertainment     606.48         687
## 4      Microsoft Game Studios     248.32         191
```

```
# Get the yearly sales for each game
new_pubs <- sales %>% filter(Publisher %in% pubs_wanted)
# Dropping the N/A (weren't normal NA's but rather strings)
new_pubs <- new_pubs %>% filter(Year_of_Release != "N/A")

# Transform
new_pubs$Global_Sales <- log10(new_pubs$Global_Sales)

new_pubs %>% ggplot(aes(x = Global_Sales, y = Publisher, fill = Publisher)) +
  geom_density_ridges() +
  theme(legend.position = "none") +
  labs(
    x = "Global Sales",
    y = "Publisher",
    title = "Total Global Sales of Top Publishers",
    subtitle = "With a Log 10 Transformation"
  )
```

```
## Picking joint bandwidth of 0.162
```

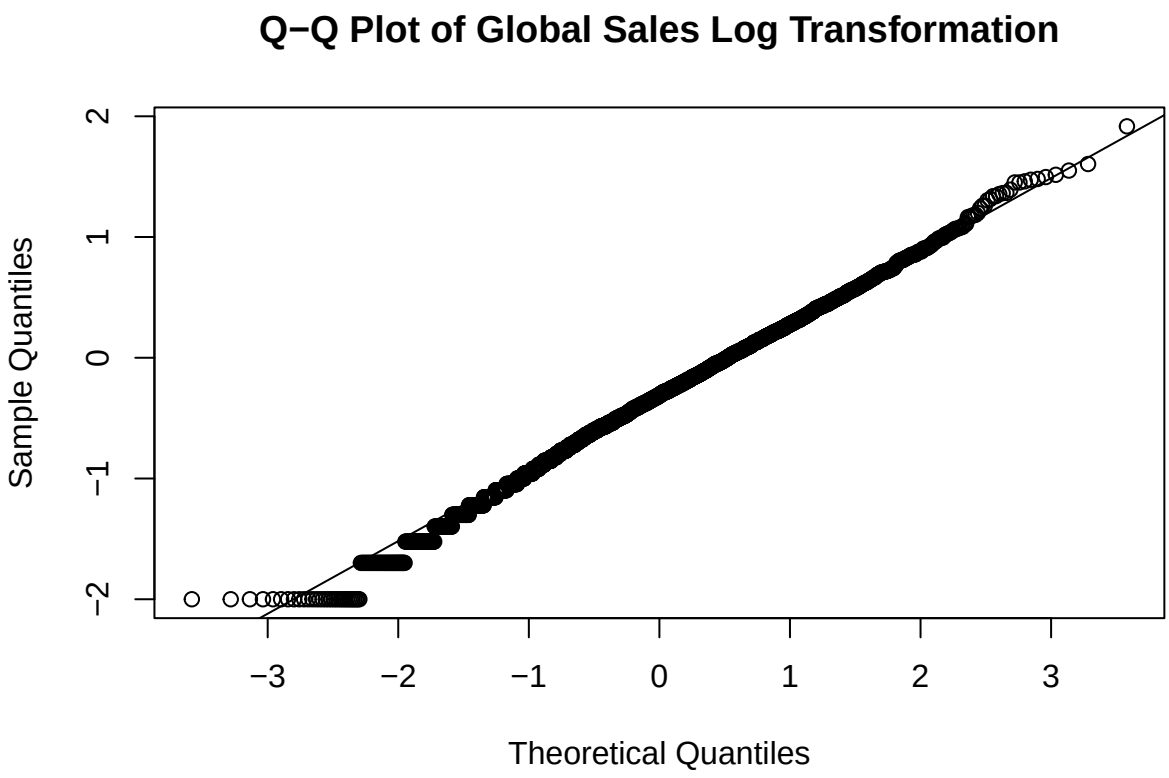

```
ggplot(new_pubs, aes(x=Publisher, y=Global_Sales, color=Publisher)) +
  geom_boxplot() + coord_flip() +
  labs(
    y = "Global Sales",
    x = "Publisher",
    title = "Total Global Sales of Top Publishers",
    subtitle = "With a Log 10 Transformation"
  )
```


```r
years <- unique(new_pubs$Year_of_Release)
years <- sort(years, decreasing = FALSE)

size <- length(years) + 1

sales_per_year <- data.frame(matrix(ncol = size, nrow = 0))
colnames(sales_per_year) <- c("pubs", years)

# Basically for each company we want, we add the total sales for each year
for (x in pubs_wanted) {
  # Get only one publisher
  currPub <- new_pubs %>% filter(Publisher == x)
  # Pub needs to be the in the first column
  year_tots <- c(x)
  for (year in years) {
    # Get the year we are on
    pub_year <- currPub %>% filter(Year_of_Release == year)
    # Add them all up and append to vector
    tot_sales_year <- sum(pub_year$Global_Sales)
    year_tots <- append(year_tots, tot_sales_year)
  }
  # Add as row
  sales_per_year[nrow(sales_per_year) + 1,] = year_tots
}

sales_per_year[,2:ncol(sales_per_year)] <- sapply(sales_per_year[,2:ncol(sales_per_year)], as.numeric)
```

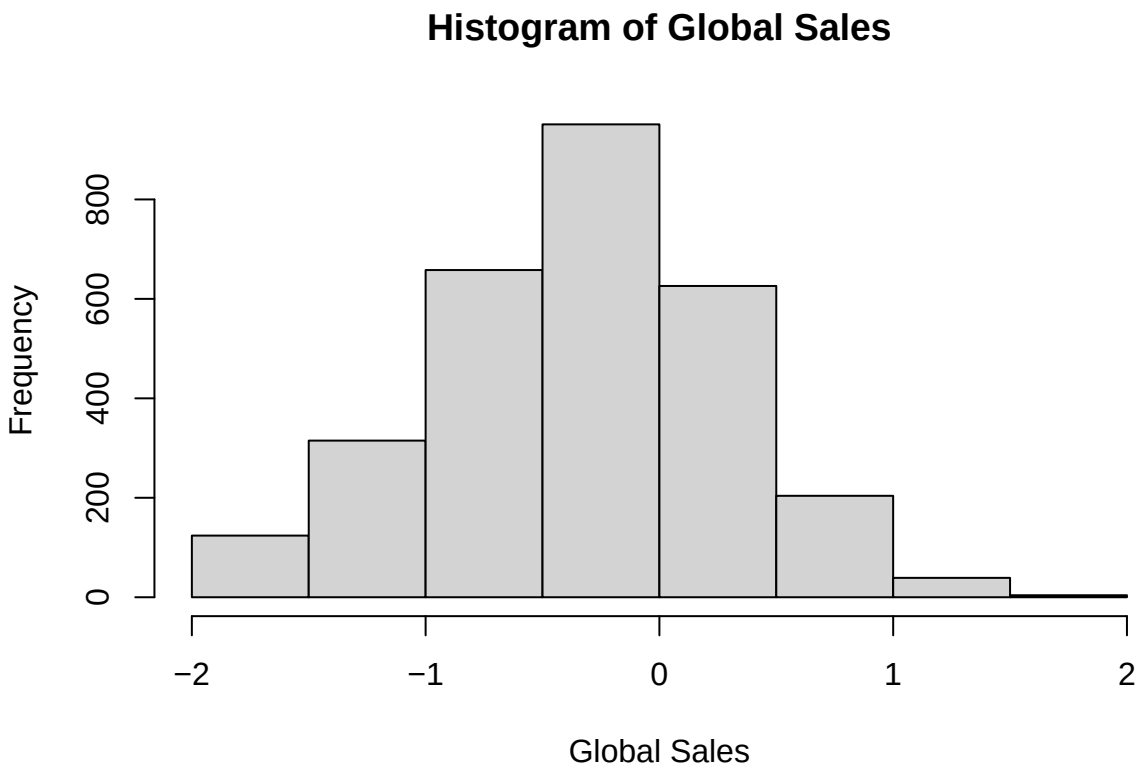
```
sales_per_year
```

```
##                           pubs     1983     1984     1985     1986     1987
## 1                     Nintendo 1.333293 3.638638 2.409005 1.966633 1.834131
## 2             Electronic Arts 0.000000 0.000000 0.000000 0.000000 0.000000
## 3      Microsoft Game Studios 0.000000 0.000000 0.000000 0.000000 0.000000
## 4 Sony Computer Entertainment 0.000000 0.000000 0.000000 0.000000 0.000000
##       1988     1989     1990      1991      1992      1993      1994       1995
## 1 3.764271 4.773255 3.275283 -1.322272  3.793716 0.2855152  2.439689  -1.277453
## 2 0.000000 0.000000 0.000000  0.000000 -1.221849 0.0000000 -1.341035  -9.764336
## 3 0.000000 0.000000 0.000000  0.000000  0.000000 0.0000000  0.000000   0.000000
## 4 0.000000 0.000000 0.000000  0.000000  0.000000 0.0000000 -3.352273 -18.541097
##        1996        1997      1998      1999          2000       2001
## 1  3.690288   0.0833432  4.355879  2.641630  -1.844605016   1.557575
## 2 -4.946057  -3.9894784 -5.785105 -7.008110  -7.977860909  -9.695828
## 3  0.709270   0.0000000  0.000000 -1.045757  -0.004364805  -2.910763
## 4 -5.345205 -14.2934668 -8.411217 -3.535222 -14.858512533 -15.935907
##          2002        2003       2004       2005       2006       2007
## 1  -0.4473017  -0.6959035 -20.604913  -8.353583 -16.336001  -6.036686
## 2 -39.9805914 -34.8171024 -34.962736 -57.215115 -59.409714 -50.266003
## 3 -16.5103728 -20.9171547  -9.038978  -8.931846  -3.901701 -10.690727
## 4 -14.3711991 -11.7868261  -8.803950 -25.982940 -33.981878 -15.529146
##         2008        2009        2010       2011        2012       2013
## 1  -5.231829  -2.8215261  -3.7578157  -3.260969  -5.6650670  2.8033229
## 2 -49.301569 -43.3586079 -23.4894342 -25.382053 -10.1306274 -4.9864997
## 3  -2.299644  -0.8517416   0.6036236  -1.210317  -0.7588444 -0.8585657
## 4 -18.075620 -24.0992224 -26.5353945 -20.397179  -8.5626014 -6.1624772
##         2014        2015       2016
## 1  0.7933938 -14.5207117  -8.623450
## 2 -3.5142192  -9.2602956 -10.901139
## 3 -1.0239545   0.6422707  -2.778424
## 4 -3.3534363  -4.3265533  -7.266854
```

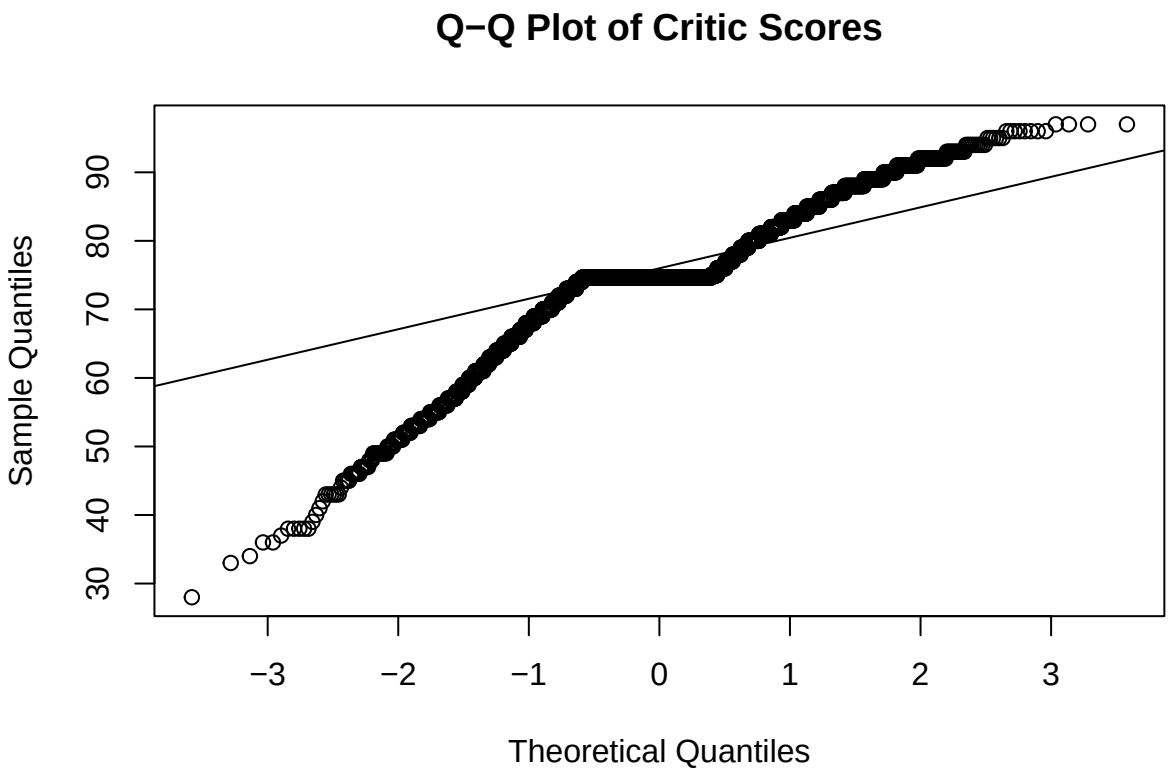
```
# Checking if global sales is right for linear regression after log transform
qqnorm(new_pubs$Global_Sales, main="Q-Q Plot of Global Sales Log Transformation")
qqline(new_pubs$Global_Sales)
```



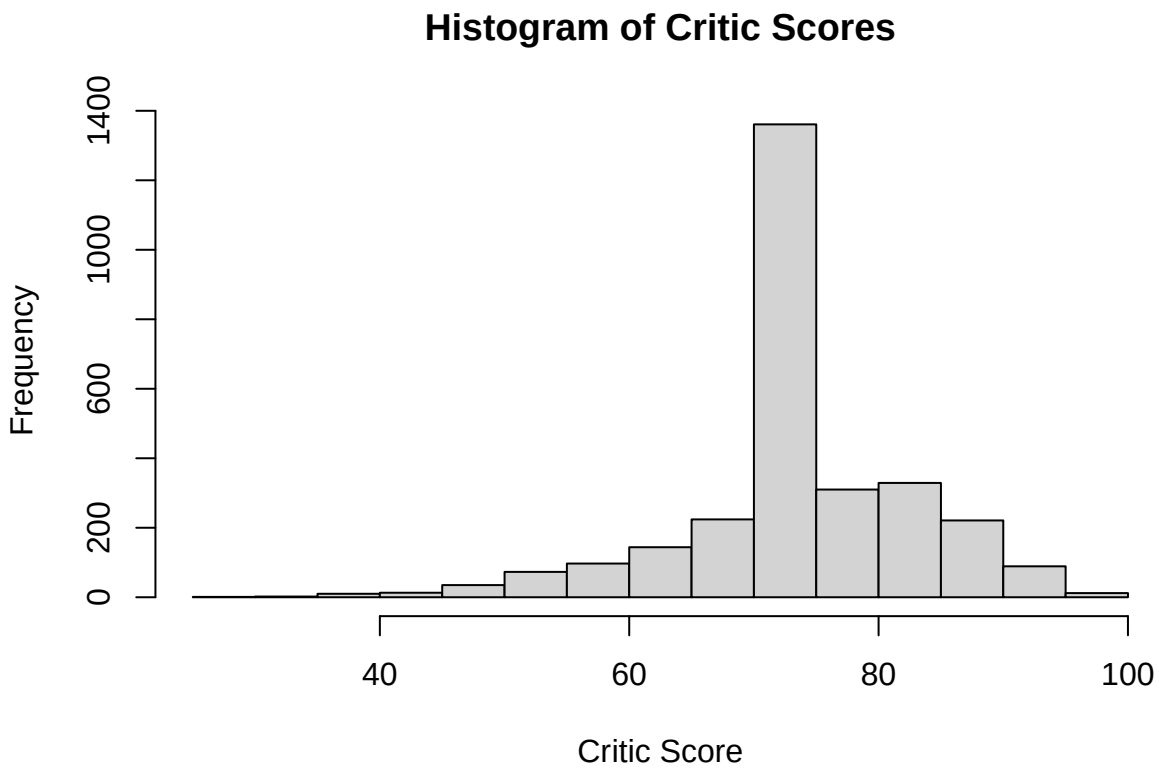

```r
hist(new_pubs$Global_Sales, xlab="Global Sales", main="Histogram of Global Sales")
```



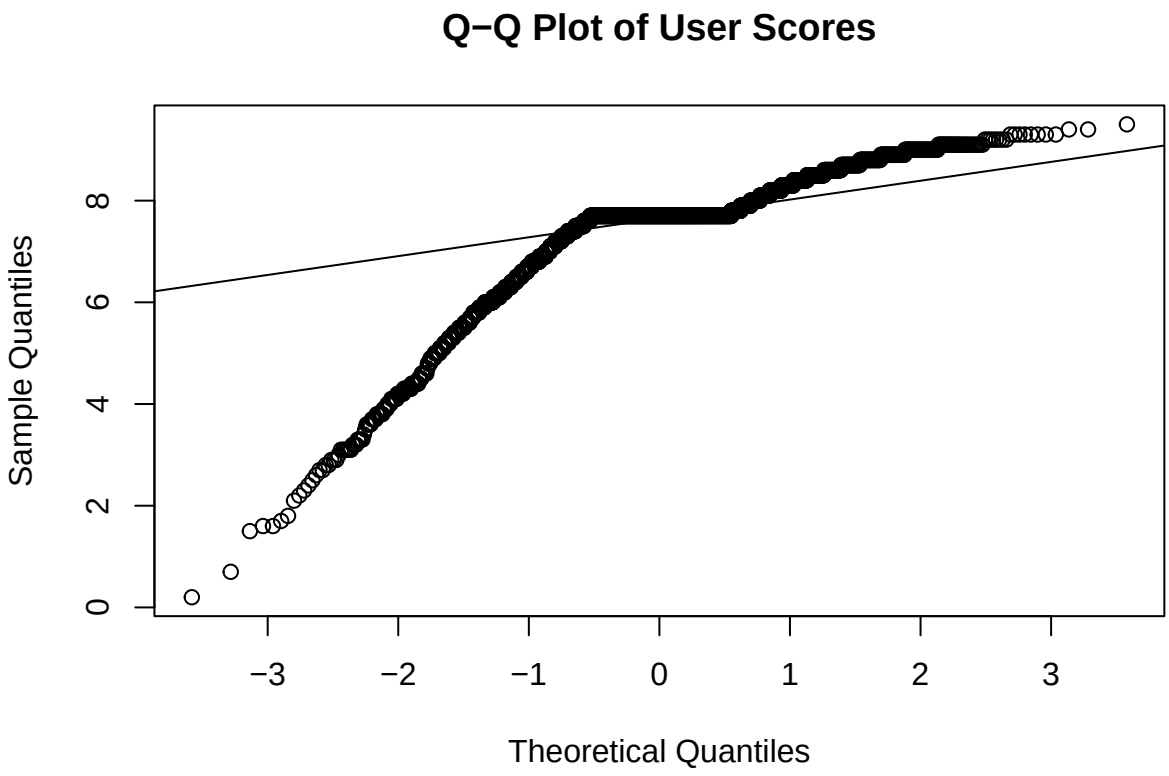

```r
# Some EDA on the critic scores
# Imputing missing values with mean
new_pubs$Critic_Score[is.na(new_pubs$Critic_Score)] <- mean(new_pubs$Critic_Score, na.rm=TRUE)

qqnorm(new_pubs$Critic_Score, main="Q-Q Plot of Critic Scores")
qqline(new_pubs$Critic_Score)
```



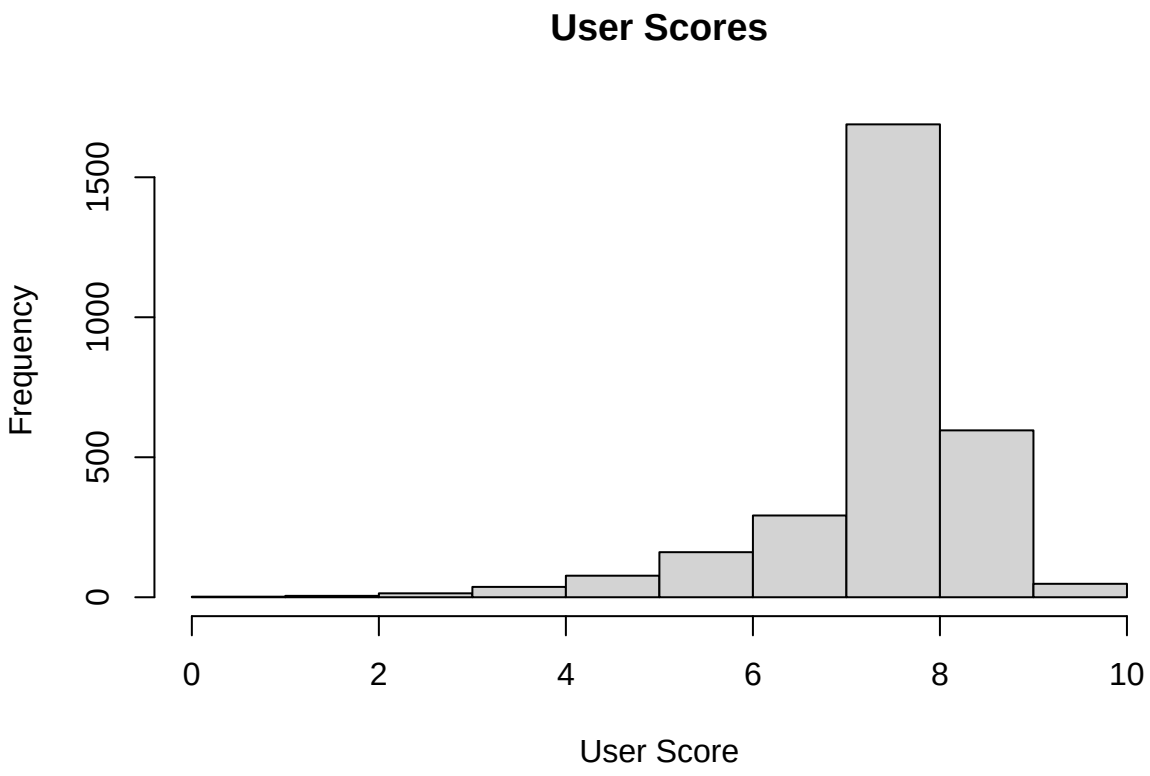

```r
hist(new_pubs$Critic_Score, xlab="Critic Score", main="Histogram of Critic Scores")
```




```r
# EDA on the user cores
# Imputing missing values with median
new_pubs$User_Score[is.na(new_pubs$User_Score)] <- median(new_pubs$User_Score, na.rm=TRUE)

qqnorm(new_pubs$User_Score, main="Q-Q Plot of User Scores")
qqline(new_pubs$User_Score)
```




```
hist(new_pubs$User_Score, xlab="User Score", main="User Scores")
```




```
# Linear regression model for sales and critic/user scores
lin_reg_uc_scores <- lm(Global_Sales ~ User_Score + Critic_Score, data=new_pubs)

summary(lin_reg_uc_scores)
```

```
##
## Call:
## lm(formula = Global_Sales ~ User_Score + Critic_Score, data = new_pubs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.08028 -0.35586  0.03305  0.39597  2.23109
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.644336   0.099152 -16.584   <2e-16 ***
## User_Score   -0.026046   0.011016  -2.364   0.0181 *
## Critic_Score  0.020240   0.001329  15.228   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5972 on 2918 degrees of freedom
## Multiple R-squared:  0.07953,    Adjusted R-squared:  0.0789
## F-statistic: 126.1 on 2 and 2918 DF,  p-value: < 2.2e-16
```

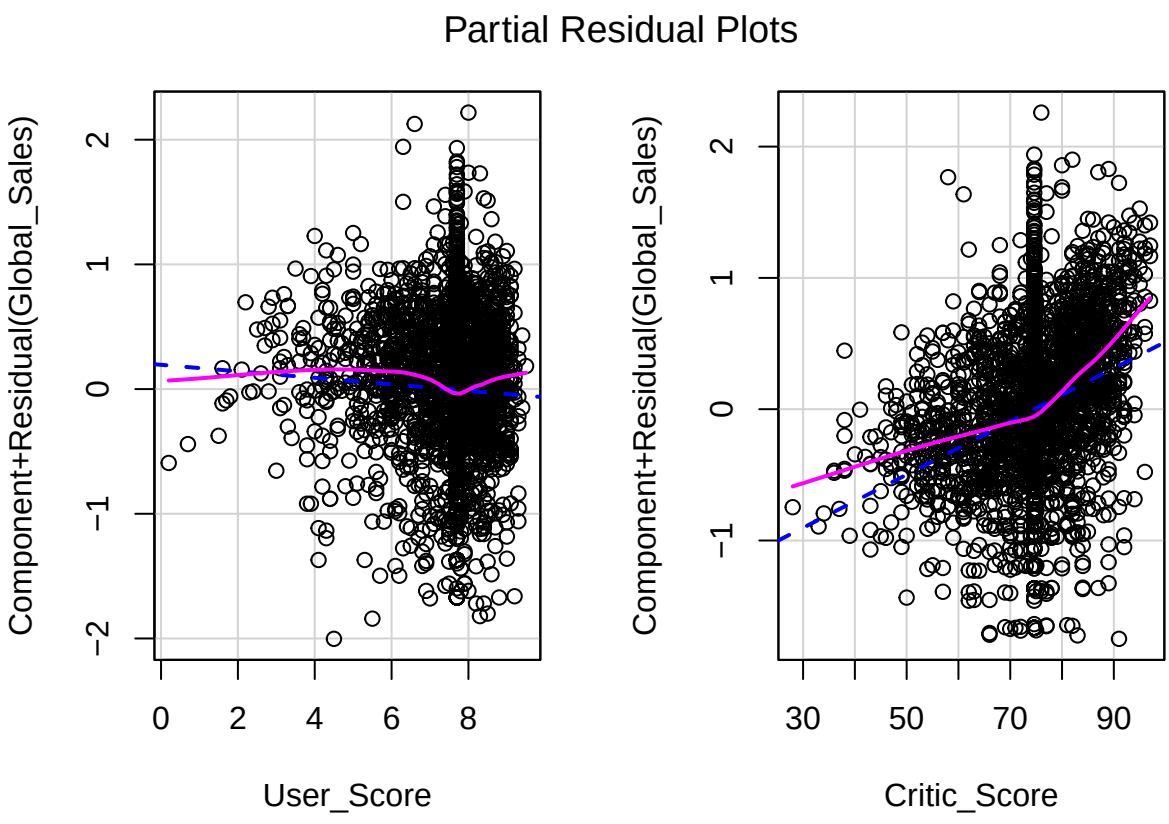
```r
confint(lin_reg_uc_scores) # 95% confidence interval
```

```
##                   2.5 %       97.5 %
## (Intercept)  -1.83875022 -1.449921081
## User_Score   -0.04764510 -0.004446358
## Critic_Score  0.01763377  0.022845837
```

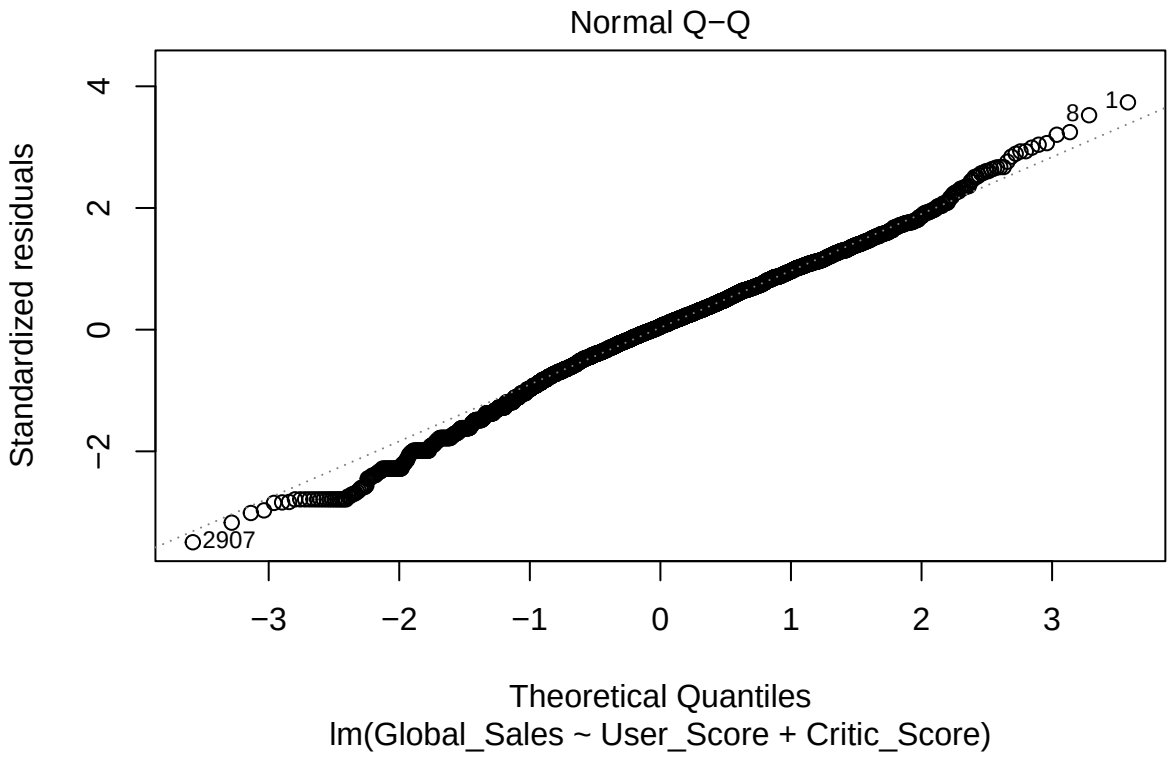
```r
correlation_m <- data.frame(new_pubs$Global_Sales, new_pubs$User_Score, new_pubs$Critic_Score)
cor(correlation_m)
```

```
##                       new_pubs.Global_Sales new_pubs.User_Score
## new_pubs.Global_Sales            1.00000000          0.07984132
## new_pubs.User_Score              0.07984132          1.00000000
## new_pubs.Critic_Score            0.27886275          0.42277990
##                       new_pubs.Critic_Score
## new_pubs.Global_Sales             0.2788627
## new_pubs.User_Score               0.4227799
## new_pubs.Critic_Score             1.0000000
```

```r
# Assumption checking linear regression of sales and critic/user scores
crPlots(lin_reg_uc_scores, main="Partial Residual Plots")
```




```r
plot(lin_reg_uc_scores,2)
```




```
vif(lin_reg_uc_scores)
```

```
##   User_Score Critic_Score
##     1.217645     1.217645
```

```
stocks <- read.csv("Stocks - DATA101 - Sheet1.csv")
stats_combined <- looking %>% left_join(sales, by = c("pubs" = "Publisher"))
```

```
stats_arranged <- stats_combined %>% arrange(Year_of_Release) %>% filter(Year_of_Release >= 2004)
stocks_with_stats <- stocks %>% inner_join(stats_arranged, by = c("company" = "pubs"))
```

```
# Linear regression for stocks
stock_lm <- lm(change_.last. ~ sale_total, data = stocks_with_stats)

summary(stock_lm)
```

```
##
## Call:
## lm(formula = change_.last. ~ sale_total, data = stocks_with_stats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.780  -3.570   0.489   6.640  22.665
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.4479796  0.3023355   11.40  < 2e-16 ***
## sale_total  -0.0018242  0.0002321   -7.86 4.03e-15 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.35 on 18862 degrees of freedom
##   (1572 observations deleted due to missingness)
## Multiple R-squared:  0.003265,   Adjusted R-squared:  0.003212
## F-statistic: 61.79 on 1 and 18862 DF,  p-value: 4.032e-15
```