Download Prediction model

Narendranath Edara

5/11/2022

load libraries

```r
# Importing all libraries
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(ggplot2)
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(MLmetrics)
```

```
##
## Attaching package: 'MLmetrics'

## The following objects are masked from 'package:caret':
##
##     MAE, RMSE

## The following object is masked from 'package:base':
##
##     Recall
```

```r
library(tinytex)
```

load data

```r
df <- read.csv("Data set.csv")
str(df)
```

```
## 'data.frame':    80 obs. of  20 variables:
##  $ gname           : chr  "What remains of Edith Finch" "Lego Star Wars: The Skywalker Saga" "Grand T
##  $ gprice          : num  5.99 49.99 29.98 14.99 20.99 ...
##  $ dev             : chr  "Giant Sparrow" "TT Games" "Rockstar North" "Crystal Dynamics" ...
##  $ rel_days        : chr  "4/24/17" "4/5/22" "4/14/15" "3/14/13" ...
##  $ diff_days       : int  1832 25 2573 3334 1340 1499 173 1355 569 233 ...
##  $ all_rev         : int  25372 20888 1334556 211865 70405 33031 71408 13343 56403 21657 ...
##  $ rec_rev         : int  374 1100 15614 884 1150 614 3423 87 163 1393 ...
##  $ pos_rev         : int  24235 19519 1127845 203793 53533 29323 61939 12599 42820 11796 ...
##  $ neg_rev         : chr  "1137" "1369" "206711" "8072" ...
##  $ ytc_name        : chr  "JackSepticEye" "gameranx" "IGN" "IGN" ...
##  $ ytc_subs        : int  2820000 7060000 16600000 16600000 11400 16600000 16600000 16600000 1150000
##  $ ytc_view        : chr  "2598555" "1717937" "3922790" "1382585" ...
##  $ ytc_likes       : int  93000 58000 52000 13000 989 16000 53000 4400 934 5000 ...
##  $ ytc_com         : int  9530 3073 14187 5791 161 1135 3898 1233 432 538 ...
##  $ twt_view        : int  14000 228300 753700 306700 727900 221000 108200 235400 555000 31400 ...
##  $ twt_flw         : int  13300 114000 370400 270200 50000 32900 269900 958300 8900000 3400000 ...
##  $ twi_flw         : int  20600 38500 54100000 334000 614000 344000 485000 99500 3800000 464000 ...
##  $ platform_mac    : int  1 1 1 1 0 0 0 1 0 1 ...
##  $ platform_win    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ platform_win.mac: int  1 1 1 1 0 0 0 1 0 1 ...
```

```r
head(df)
```

```
##                                gname gprice              dev rel_days diff_days
## 1        What remains of Edith Finch   5.99    Giant Sparrow  4/24/17      1832
## 2 Lego Star Wars: The Skywalker Saga  49.99         TT Games   4/5/22        25
## 3                  Grand Theft Auto 5  29.98   Rockstar North  4/14/15      2573
## 4                        Tomb Raider  14.99 Crystal Dynamics  3/14/13      3334
## 5                               Scum  20.99        Gamepires  8/29/18      1340
## 6                          A Way Out  42.73        Hazelight  3/23/18      1499
##   all_rev rec_rev pos_rev neg_rev      ytc_name ytc_subs ytc_view ytc_likes
## 1   25372     374   24235    1137 JackSepticEye  2820000  2598555     93000
## 2   20888    1100   19519    1369      gameranx  7060000  1717937     58000
## 3 1334556   15614 1127845  206711           IGN 16600000  3922790     52000
## 4  211865     884  203793    8072           IGN 16600000  1382585     13000
## 5   70405    1150   53533   17872  game advisor    11400    73859       989
## 6   33031     614   29323    3708           IGN 16600000  1100000     16000
##   ytc_com twt_view twt_flw  twi_flw platform_mac platform_win platform_win.mac
## 1    9530    14000   13300    20600            1            1                1
## 2    3073   228300  114000    38500            1            1                1
## 3   14187   753700  370400 54100000            1            1                1
## 4    5791   306700  270200   334000            1            1                1
```

```
## 5     161    727900    50000    614000              0              1                 0
## 6    1135    221000    32900    344000              0              1                 0
```

Data cleansing and variable selection

```
# variable selection

data <- df %>%
  select(gprice, diff_days, all_rev,    rec_rev,    pos_rev,    neg_rev, ytc_subs, ytc_view, ytc_likes,

data <- data.frame(data)


# converting all character values to numeric

#numeric <- c("gprice", "diff_days", "all_rev", "rec_rev",  "pos_rev",  "neg_rev", "ytc_subs", "ytc_vie


#data[numeric] <- lapply(data[numeric], as.numeric)
```

```
str(data)
```

```
## 'data.frame':    80 obs. of  16 variables:
##  $ gprice          : num  5.99 49.99 29.98 14.99 20.99 ...
##  $ diff_days       : int  1832 25 2573 3334 1340 1499 173 1355 569 233 ...
##  $ all_rev         : int  25372 20888 1334556 211865 70405 33031 71408 13343 56403 21657 ...
##  $ rec_rev         : int  374 1100 15614 884 1150 614 3423 87 163 1393 ...
##  $ pos_rev         : int  24235 19519 1127845 203793 53533 29323 61939 12599 42820 11796 ...
##  $ neg_rev         : chr  "1137" "1369" "206711" "8072" ...
##  $ ytc_subs        : int  2820000 7060000 16600000 16600000 11400 16600000 16600000 16600000 1150000
##  $ ytc_view        : chr  "2598555" "1717937" "3922790" "1382585" ...
##  $ ytc_likes       : int  93000 58000 52000 13000 989 16000 53000 4400 934 5000 ...
##  $ ytc_com         : int  9530 3073 14187 5791 161 1135 3898 1233 432 538 ...
##  $ twt_view        : int  14000 228300 753700 306700 727900 221000 108200 235400 555000 31400 ...
##  $ twt_flw         : int  13300 114000 370400 270200 50000 32900 269900 958300 8900000 3400000 ...
##  $ twi_flw         : int  20600 38500 54100000 334000 614000 344000 485000 99500 3800000 464000 ...
##  $ platform_mac    : int  1 1 1 1 0 0 0 1 0 1 ...
##  $ platform_win    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ platform_win.mac: int  1 1 1 1 0 0 0 1 0 1 ...
```

Data Manipulation

```
data$platform_mac <- as.factor(data$platform_mac)
data$platform_mac <- unclass(data$platform_mac)
data$platform_mac <- as.numeric(as.character(data$platform_mac))

data$platform_win.mac <- as.factor(data$platform_win.mac)
data$platform_win.mac <- unclass(data$platform_win.mac)
```

```r
data$platform_win.mac <- as.numeric(as.character(data$platform_win.mac))

data$platform_win <- as.numeric(as.character(data$platform_win))
data$gprice <- as.numeric(as.character(data$gprice))
data$neg_rev <- as.numeric(as.character(data$neg_rev))
```

```
## Warning: NAs introduced by coercion
```

```r
data$ytc_view <- as.numeric(as.character(data$ytc_view))
```

```
## Warning: NAs introduced by coercion
```

```r
str(data)
```

```
## 'data.frame':    80 obs. of  16 variables:
##  $ gprice          : num  5.99 49.99 29.98 14.99 20.99 ...
##  $ diff_days       : int  1832 25 2573 3334 1340 1499 173 1355 569 233 ...
##  $ all_rev         : int  25372 20888 1334556 211865 70405 33031 71408 13343 56403 21657 ...
##  $ rec_rev         : int  374 1100 15614 884 1150 614 3423 87 163 1393 ...
##  $ pos_rev         : int  24235 19519 1127845 203793 53533 29323 61939 12599 42820 11796 ...
##  $ neg_rev         : num  1137 1369 206711 8072 17872 ...
##  $ ytc_subs        : int  2820000 7060000 16600000 16600000 11400 16600000 16600000 16600000 1150000
##  $ ytc_view        : num  2598555 1717937 3922790 1382585 73859 ...
##  $ ytc_likes       : int  93000 58000 52000 13000 989 16000 53000 4400 934 5000 ...
##  $ ytc_com         : int  9530 3073 14187 5791 161 1135 3898 1233 432 538 ...
##  $ twt_view        : int  14000 228300 753700 306700 727900 221000 108200 235400 555000 31400 ...
##  $ twt_flw         : int  13300 114000 370400 270200 50000 32900 269900 958300 8900000 3400000 ...
##  $ twi_flw         : int  20600 38500 54100000 334000 614000 344000 485000 99500 3800000 464000 ...
##  $ platform_mac    : num  2 2 2 2 1 1 1 2 1 2 ...
##  $ platform_win    : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ platform_win.mac: num  2 2 2 2 1 1 1 2 1 2 ...
```

Data manipulation for categorical variables set with factor levels

```r
data$platform_mac
```

```
##  [1] 2 2 2 2 1 1 1 2 1 2 1 1 1 1 1 1 2 1 1 1 2 1 2 2 1 2 1 2 2 2 2 2 1 2 2 2 2 2
## [39] 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 2 2 2 1 1 1 2 1 1 1 1 1 1 2
## [77] 1 2 2 2
```

```r
data$platform_win.mac
```

```
##  [1] 2 2 2 2 1 1 1 2 1 2 1 1 1 1 1 1 2 1 1 1 2 1 2 2 1 2 1 2 2 2 2 2 1 2 2 2 2 2
## [39] 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 2 2 2 1 1 1 2 1 1 1 1 1 1 2
## [77] 1 2 2 2
```

```r
data$platform_win
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [39] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [77] 1 1 1 1
```

Dealing with missing values "N/A"

```
data[is.na(data)] <- min(data, na.rm = TRUE)
sum(is.na(data))
```

```
## [1] 0
```

```
summary(data)
```

```
##      gprice        diff_days         all_rev            rec_rev
##  Min.   : 0.00   Min.   :   8.0   Min.   :       0   Min.   :       0
##  1st Qu.:14.99   1st Qu.: 225.5   1st Qu.:   16709   1st Qu.:     376
##  Median :29.98   Median :1063.5   Median :   59162   Median :    1018
##  Mean   :32.08   Mean   :1320.3   Mean   :  402654   Mean   :  162722
##  3rd Qu.:49.99   3rd Qu.:1946.0   3rd Qu.:  164981   3rd Qu.:    2914
##  Max.   :94.04   Max.   :5283.0   Max.   :13000000   Max.   :12600000
##     pos_rev            neg_rev            ytc_subs            ytc_view
##  Min.   :      0   Min.   :      0   Min.   :    11400   Min.   :       0
##  1st Qu.:  10798   1st Qu.:   1096   1st Qu.:   767250   1st Qu.:  646753
##  Median :  44314   Median :   3836   Median :  4910000   Median : 2333182
##  Mean   : 238224   Mean   : 113964   Mean   : 11696861   Mean   : 4862955
##  3rd Qu.: 100346   3rd Qu.:  14469   3rd Qu.: 16600000   3rd Qu.: 5300083
##  Max.   :8820000   Max.   :6295746   Max.   :111000000   Max.   :62396088
##    ytc_likes          ytc_com           twt_view            twt_flw
##  Min.   :     0   Min.   :    0.0   Min.   :      76   Min.   :       0
##  1st Qu.:  9175   1st Qu.:  688.8   1st Qu.:   16250   1st Qu.:   15004
##  Median : 31500   Median : 2440.5   Median :  157350   Median :  113250
##  Mean   : 99672   Mean   : 7822.6   Mean   :  995247   Mean   : 1127117
##  3rd Qu.: 98250   3rd Qu.: 8609.0   3rd Qu.:  376575   3rd Qu.:  479675
##  Max.   :637000   Max.   :85140.0   Max.   :38000000   Max.   :24800000
##     twi_flw           platform_mac    platform_win platform_win.mac
##  Min.   :       0   Min.   :1.000   Min.   :1     Min.   :1.000
##  1st Qu.:   40200   1st Qu.:1.000   1st Qu.:1     1st Qu.:1.000
##  Median :  333500   Median :1.000   Median :1     Median :1.000
##  Mean   : 3637546   Mean   :1.438   Mean   :1     Mean   :1.438
##  3rd Qu.:  813000   3rd Qu.:2.000   3rd Qu.:1     3rd Qu.:2.000
##  Max.   :82400000   Max.   :2.000   Max.   :1     Max.   :2.000
```

The summary statistics now show that there are no missing values in the data set

Creating the response variable

The response variable is "downloads" which is a product of diff days (difference between game release date and April 30, 2022) and all_reviews

```
data <- data %>%
  mutate(downloads = diff_days * all_rev)
data$downloads
```

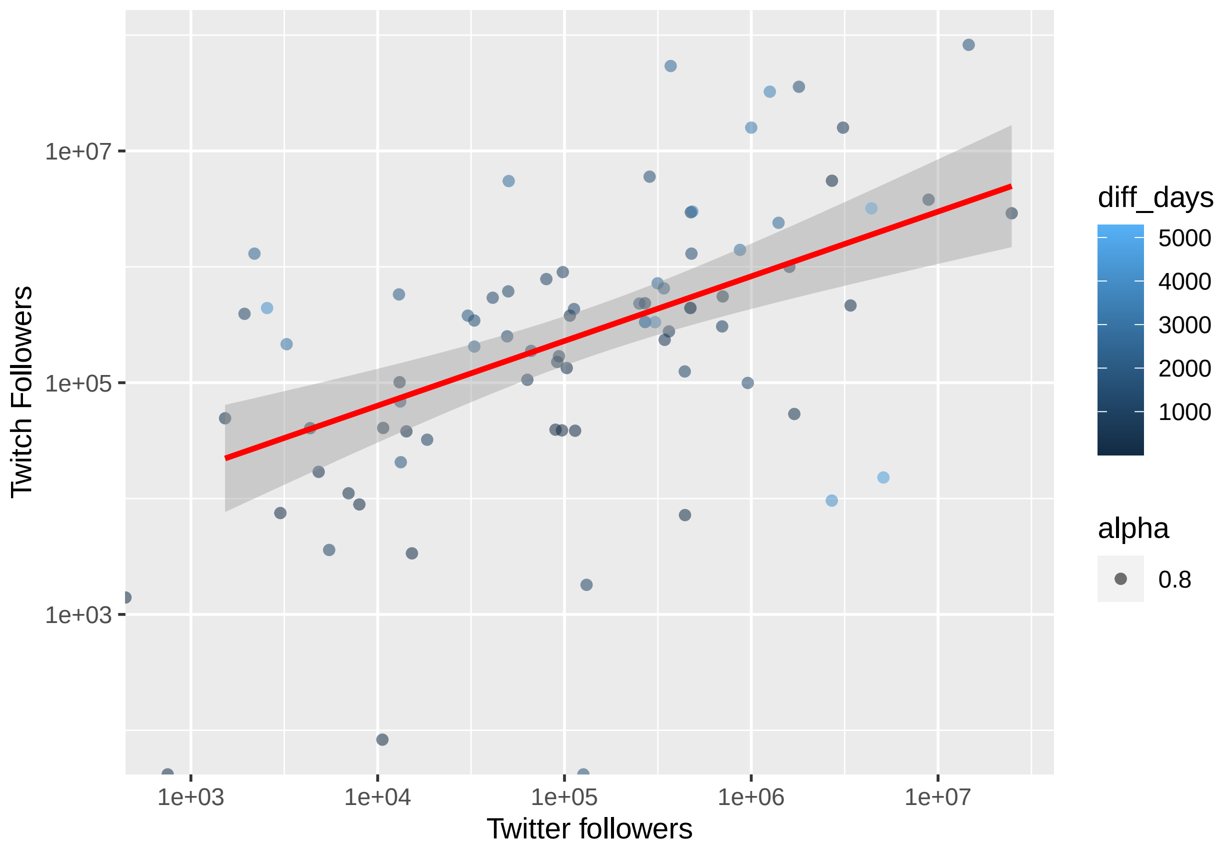
```
##  [1]    46481504      522200  3433812588   706357910    94342700    49513469
##  [7]    12353584    18079765    32093307     5046081   204123510     1430782
## [13]    42125496     2393910   184539080     8404000     5999688    83325408
## [19]  1596020478    80950078    14827728    13535366      550950           0
## [25]     1625122    98676100   205532610    42231970    27364775       26013
## [31]    12844232 22204000000    85728039     5105030   708774176  1181104340
## [37]   211074804   342967384    22354728           0     3147200        1968
## [43] 22523873486   127277520    22499685    23130932       95274      425825
## [49]      214272    82494390   101379110      127359   122370140    10131100
## [55]   238719022  3198501852       70990   951390310   612640798  5547845180
## [61]   259526202    50436456   167587292    32472836     2604825     2521831
## [67]     5544204  1595192858    91802691    26240028   154705056   124577270
## [73]   477645429      553426      259888  1198263528       25878   145569865
## [79]   349762512    44091954
```

Data Visulalization

Scatter plot with regression line

```
ggplot(data, aes(twt_flw, twi_flw)) + geom_point(aes(color = diff_days, alpha = 0.8)) + geom_smooth(met
```

```
## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```



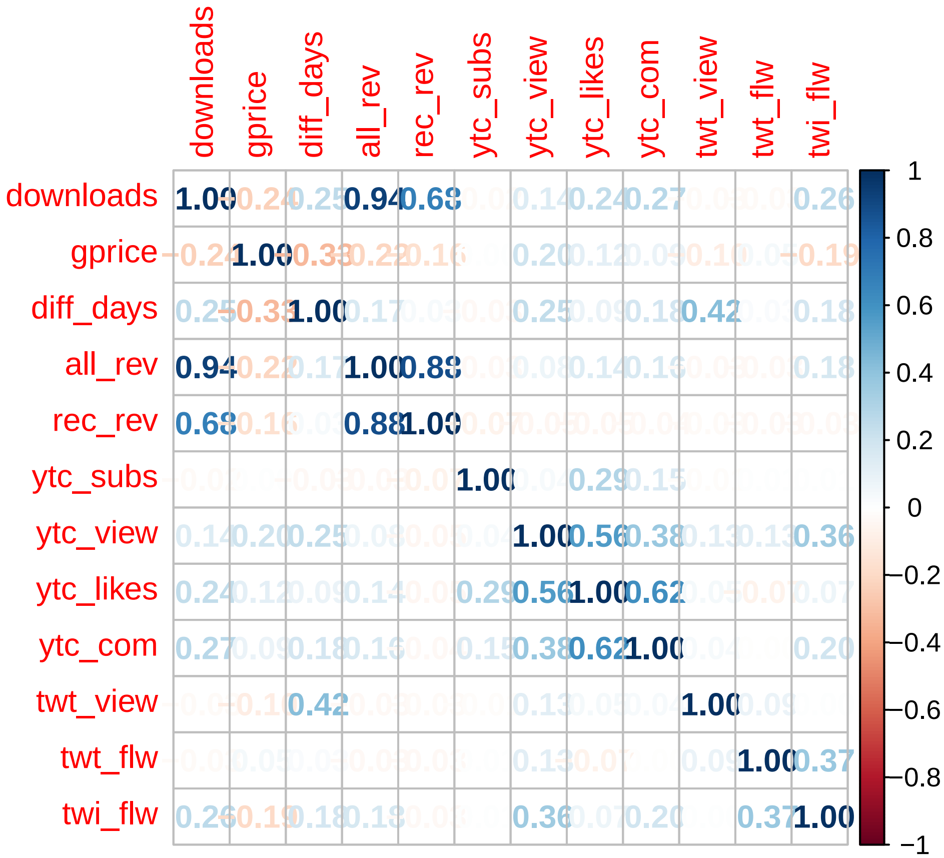
From the above scatter plot of Twitter vs Twitch followers by the number of days since the game released seems to have a positive correlation between their values

Correlation plot

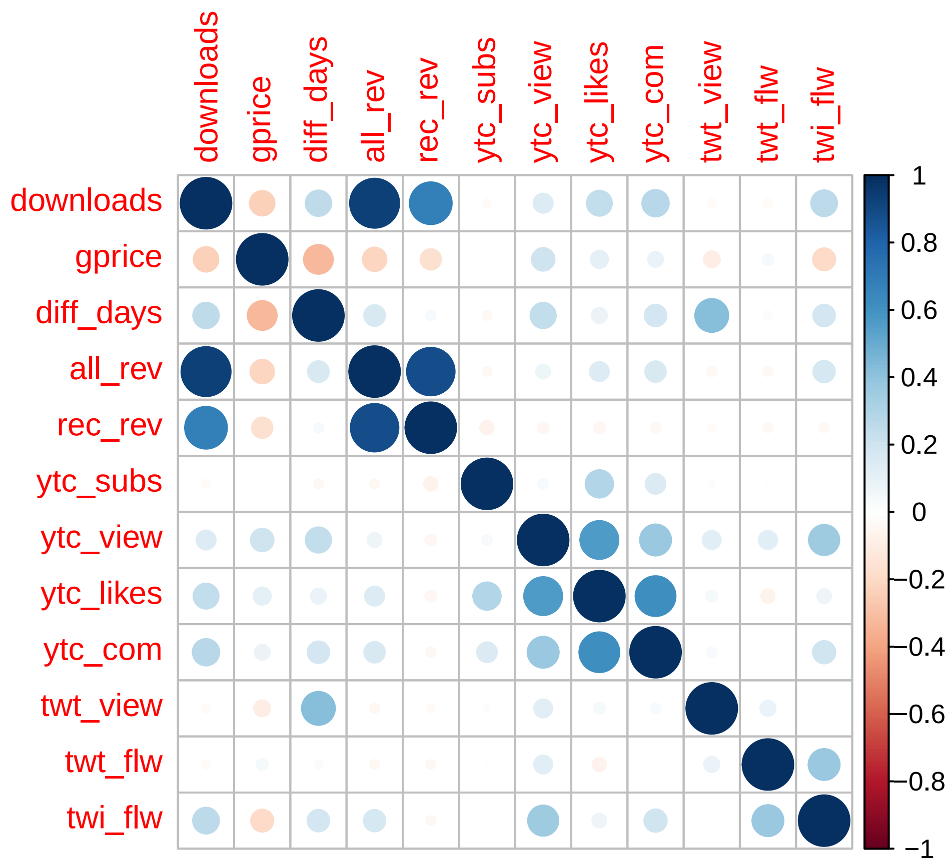
```
# correlation plot
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
data_plot <- data %>%
  select(downloads, gprice, diff_days, all_rev, rec_rev, ytc_subs, ytc_view, ytc_likes, ytc_com,    twt_
data_plot <- data.frame(data_plot)
M <- cor(data_plot)
corrplot(M, method="number")
```




```
corrplot(M, method= "circle")
```



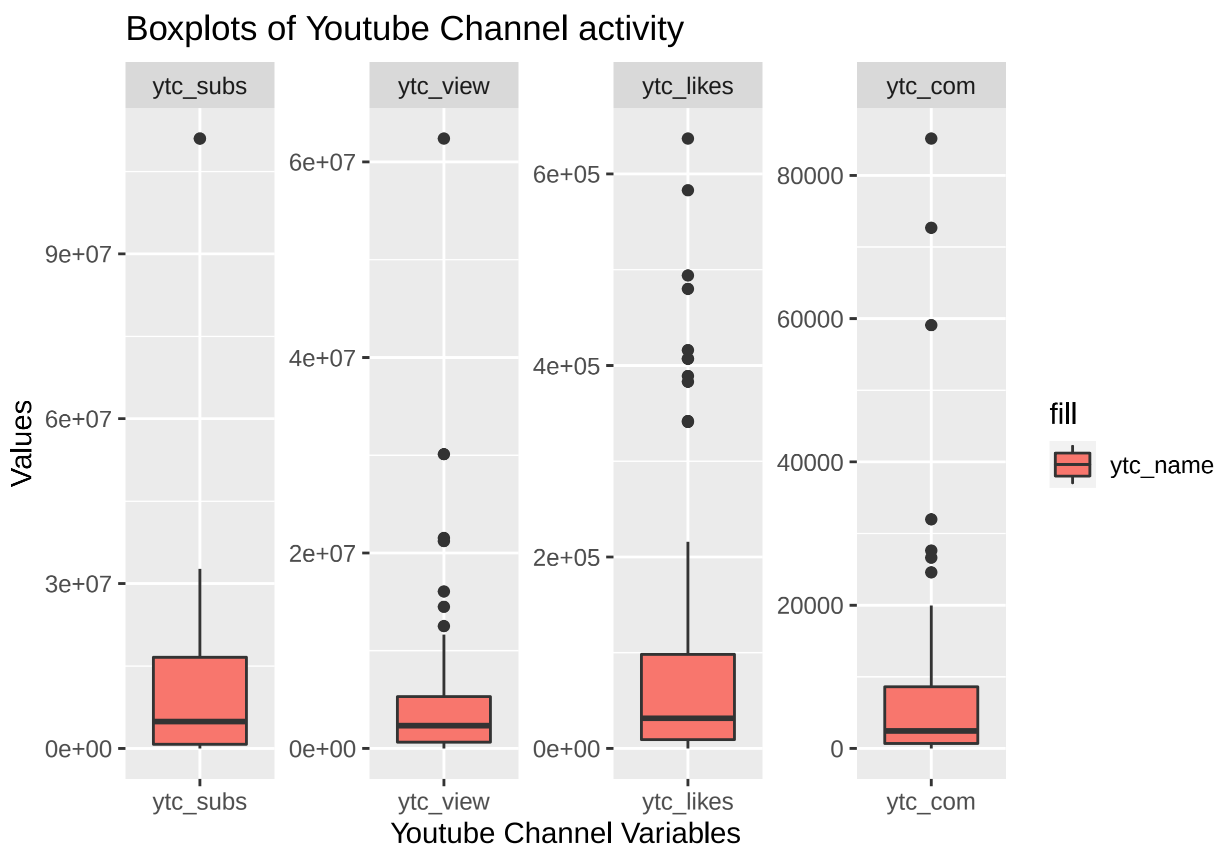
The correlation plot show that variables all_rev, rec_rev to downloads, and twt_view to diff_days are highly correlated. Other variables like ytc_view to ytc_likes, ytc_likes to ytc_com, twt_view to diff_days, and twi_flw to ytc_view have a strong correlation

Grouped Boxplot

```r
# Box plots of Youtube channel, subs, likes, and comments
library(reshape)
```

```
##
## Attaching package: 'reshape'

## The following object is masked from 'package:dplyr':
##
##     rename

## The following objects are masked from 'package:tidyr':
##
##     expand, smiths
```

```r
bplot <- melt(data = data, measure.vars = c(7,8,9,10), variable_name = "variable")

ggplot(bplot, aes(x = variable, y = value, fill = "ytc_name")) +
  geom_boxplot() + facet_wrap(~ variable, scales = "free", ncol = 4) + labs(title = "Boxplots of Youtube
```



The boxplots ytc__view, ytc__likes, ytc__com have more outliers as compared to ytc__subs which has just 1 outlier. Data in ytc__subs and ytc__likes look kind of having a similar range of values and mediean and ytc__view and ytc__com look the same with minor difference in ytc__view

Data Partitioning

```
# Data partition: randomly split the data set into a train (80%) and a test set (20%)
index <- 1:nrow(data)
set.seed(123)
train_index <- sample(index, round(length(index)*0.8))
train_set <- data[train_index,]
test_set <- data[-train_index,]
```

```
nrow(train_set)
```

```
## [1] 64
```

```
nrow(test_set)
```

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

The above table result displays that out of 80 observations the train-test-split of 80-20 assigned 64 observations for training set and 16 observations for test set

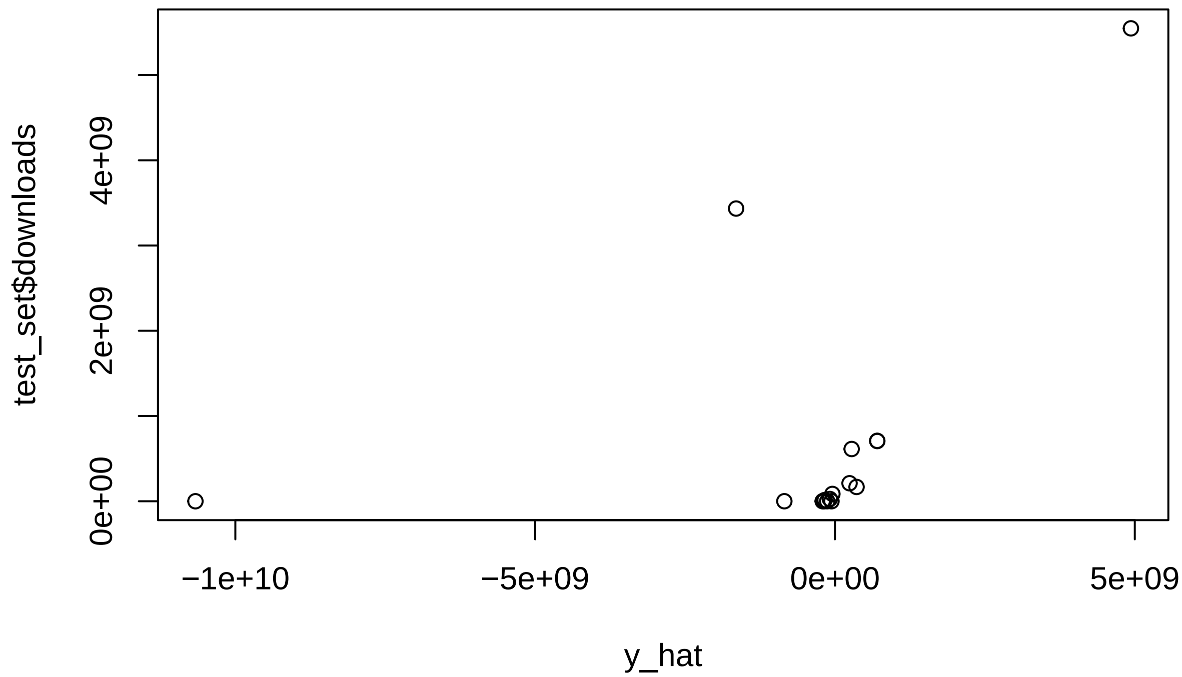
Multiple linear regression Model 1

```r
# Multiple linear regression with selected predictors
model1 <- lm(downloads ~ gprice + diff_days + all_rev + rec_rev + ytc_subs + ytc_view + ytc_likes + ytc_
summary(model1)
```

```
##
## Call:
## lm(formula = downloads ~ gprice + diff_days + all_rev + rec_rev +
##     ytc_subs + ytc_view + ytc_likes + ytc_com + twt_view + twt_flw +
##     twi_flw, data = train_set)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -913328601  -87671620   50723799  125333556  819078009
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.731e+07  1.140e+08  -0.240  0.81158
## gprice      -2.855e+06  2.364e+06  -1.208  0.23265
## diff_days    2.203e+04  4.502e+04   0.489  0.62671
## all_rev      4.186e+03  1.003e+02  41.754  < 2e-16 ***
## rec_rev     -2.557e+03  1.065e+02 -24.010  < 2e-16 ***
## ytc_subs    -6.754e+00  2.182e+00  -3.095  0.00316 **
## ytc_view    -7.809e+00  1.506e+01  -0.519  0.60621
## ytc_likes    5.985e+02  4.835e+02   1.238  0.22128
## ytc_com     -5.953e+03  3.531e+03  -1.686  0.09776 .
## twt_view     1.396e+01  1.058e+01   1.319  0.19289
## twt_flw      2.265e+01  1.316e+01   1.722  0.09108 .
## twi_flw     -1.287e+02  1.239e+01 -10.388 2.74e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.32e+08 on 52 degrees of freedom
## Multiple R-squared:  0.9941, Adjusted R-squared:  0.9928
## F-statistic: 792.7 on 11 and 52 DF,  p-value: < 2.2e-16
```

From the result of multiple linear regression model 1, the predictors all reviews, recent reviews, youtube channel subscribers, and twitch followers are statistically significant

```r
# MLR Model 1 prediction
y_hat <- predict(model1, test_set, type = "response")
mlr1_result <- postResample(y_hat, test_set$downloads)
plot(test_set$downloads ~ y_hat)
```




```
mlr1_result
```

```
##         RMSE     Rsquared          MAE
## 2.968466e+09 1.554486e-01 1.173314e+09
```

Multiple Linear Resgression model 2

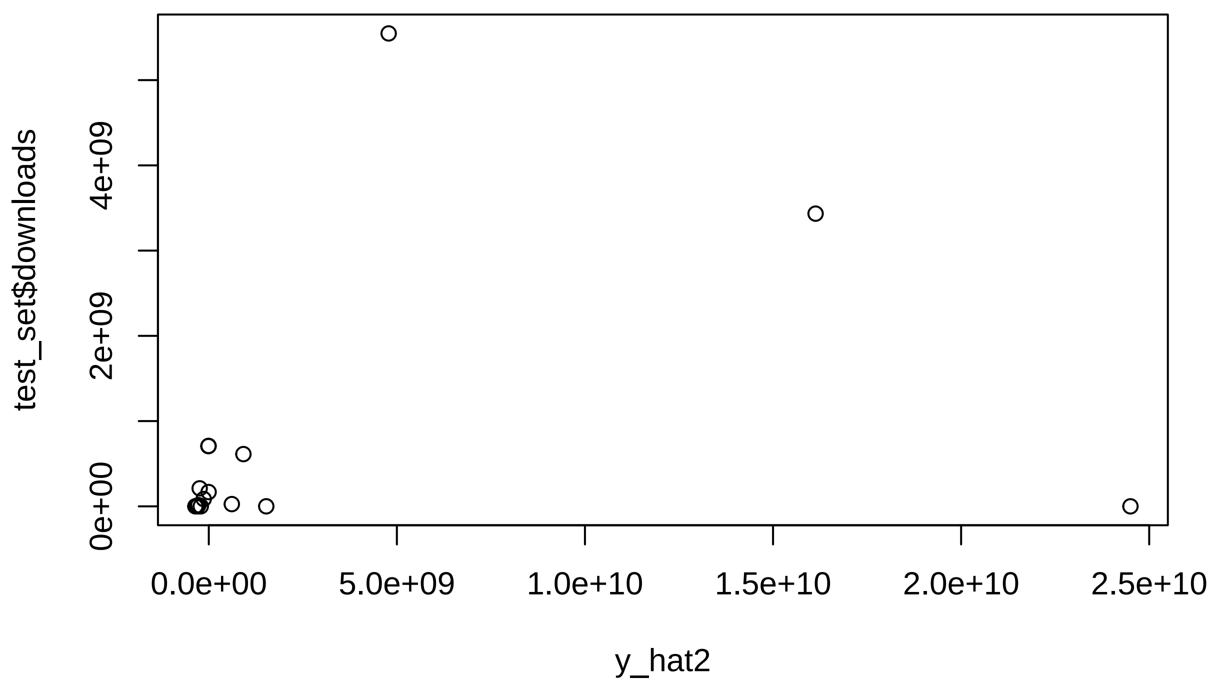
```
# Multiple linear regression with selected predictors
model2 <- lm(downloads ~ ytc_com + twt_view + twi_flw + rec_rev, data = train_set)
summary(model2)
```

```
##
## Call:
## lm(formula = downloads ~ ytc_com + twt_view + twi_flw + rec_rev,
##     data = train_set)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -7.216e+09 -2.216e+07  2.584e+08  3.317e+08  1.068e+10
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -3.531e+08  2.800e+08  -1.261  0.21230
## ytc_com      4.354e+04  1.562e+04   2.787  0.00714 **
## twt_view    -2.953e+01  4.908e+01  -0.602  0.54979
## twi_flw      2.932e+02  4.020e+01   7.292 8.65e-10 ***
## rec_rev      1.784e+03  1.538e+02  11.596  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.919e+09 on 59 degrees of freedom
## Multiple R-squared:  0.7752, Adjusted R-squared:   0.76
## F-statistic: 50.86 on 4 and 59 DF,  p-value: < 2.2e-16
```

From the result of multiple linear regression model 2, the predictors youtube comments, twitch followers, and recent reviews are statistically significant

```
# MLR Model 2 prediction
y_hat2 <- predict(model2, test_set, type = "response")
mlr2_result <- postResample(y_hat2, test_set$downloads)
plot(test_set$downloads ~ y_hat2)
```




```
mlr2_result
```

```
##         RMSE     Rsquared          MAE
## 6.922269e+09 8.890535e-02 2.758602e+09
```

Multiple Linear Resgression model 3

```r
# Multiple linear regression with all predictors
model3 <- lm(downloads ~., data = train_set)
summary(model3)
```

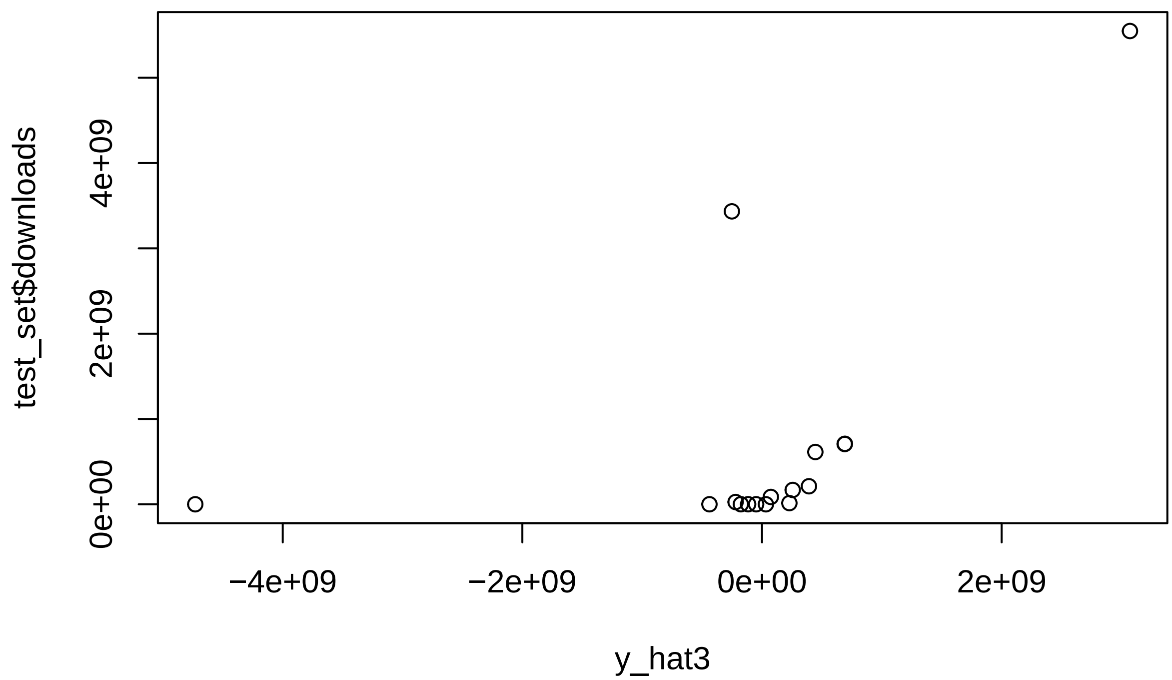
```
##
## Call:
## lm(formula = downloads ~ ., data = train_set)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -569138346  -94739456   14318365   95218221  588297212
##
## Coefficients: (2 not defined because of singularities)
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -4.486e+07  1.158e+08  -0.388  0.70003
## gprice           -2.762e+06  1.649e+06  -1.675  0.10024
## diff_days         1.060e+05  3.384e+04   3.132  0.00293 **
## all_rev           3.394e+03  4.596e+03   0.739  0.46370
## rec_rev          -7.540e+02  1.565e+03  -0.482  0.63204
## pos_rev          -1.429e+03  4.568e+03  -0.313  0.75571
## neg_rev           4.593e+02  4.586e+03   0.100  0.92064
## ytc_subs         -1.973e+00  1.635e+00  -1.207  0.23335
## ytc_view          5.242e+00  1.100e+01   0.477  0.63573
## ytc_likes        -3.245e+02  3.536e+02  -0.918  0.36324
## ytc_com          -2.421e+03  2.600e+03  -0.931  0.35649
## twt_view         -5.577e+00  7.817e+00  -0.714  0.47892
## twt_flw           9.536e+00  1.028e+01   0.928  0.35794
## twi_flw          -6.264e+01  1.204e+01  -5.204 3.83e-06 ***
## platform_mac      2.789e+07  6.607e+07   0.422  0.67483
## platform_win             NA         NA      NA       NA
## platform_win.mac         NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 227200000 on 49 degrees of freedom
## Multiple R-squared:  0.9974, Adjusted R-squared:  0.9966
## F-statistic:  1335 on 14 and 49 DF,  p-value: < 2.2e-16
```

From the result of multiple linear regression 3 with all predictors, the predictors difference date (date between release date and April 30, 2022), and twitch followers are statistically significant

```r
# MLR Model 3 prediction
y_hat3 <- predict(model3, test_set, type = "response")
```

```
## Warning in predict.lm(model3, test_set, type = "response"): prediction from a
## rank-deficient fit may be misleading
```

```
mlr3_result <- postResample(y_hat3, test_set$downloads)
plot(test_set$downloads ~ y_hat3)
```




```
mlr3_result
```

```
##         RMSE     Rsquared          MAE
## 1.629903e+09 2.588370e-01 7.905544e+08
```

Model 2: Logistic Regression

```
# Logistic regression (out of sample prediction)
logit1 <- glm(as.factor(downloads) ~ gprice + diff_days + all_rev + rec_rev + ytc_subs +    ytc_com +
```

```
## Warning: glm.fit: algorithm did not converge
```

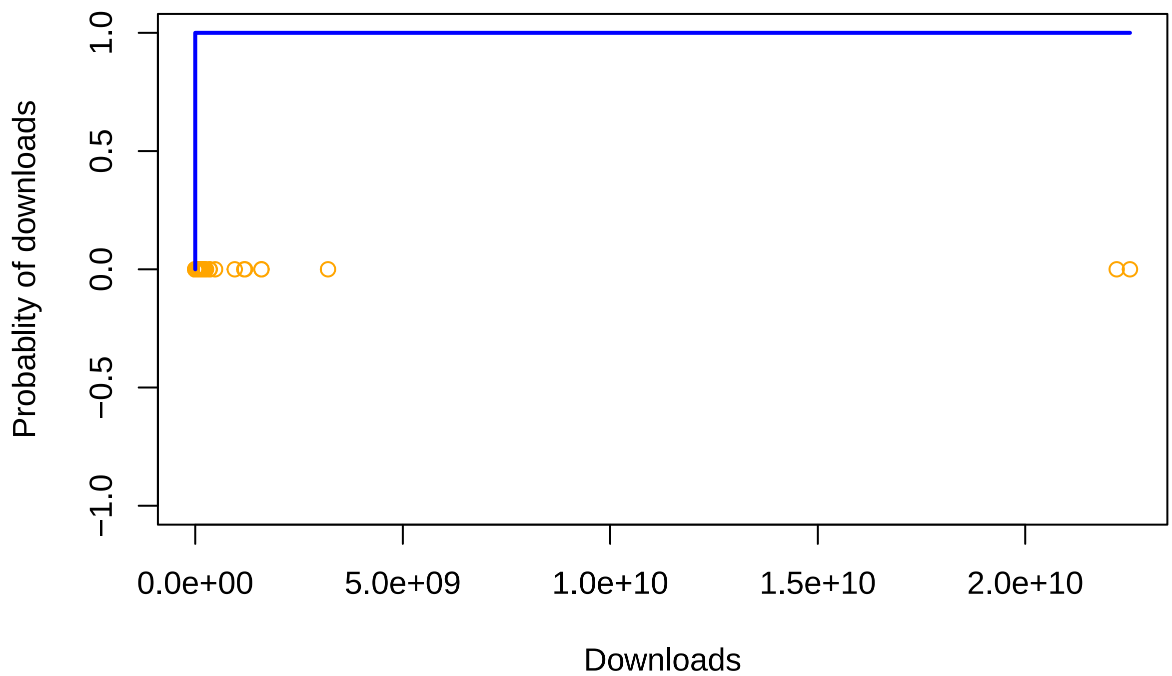
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(logit1)
```

```
##
```

```
## Call:
## glm(formula = as.factor(downloads) ~ gprice + diff_days + all_rev +
##     rec_rev + ytc_subs + ytc_com + twt_view + twi_flw, family = "binomial",
##     data = train_set)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -8.803e-05   2.100e-08   2.100e-08   2.100e-08   8.009e-05
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  9.536e+00  1.340e+04   0.001    0.999
## gprice       1.517e+00  8.053e+02   0.002    0.998
## diff_days   -3.969e-03  5.308e+00  -0.001    0.999
## all_rev      6.364e-04  2.754e-01   0.002    0.998
## rec_rev     -6.560e-04  2.857e-01  -0.002    0.998
## ytc_subs    -6.751e-07  5.878e-04  -0.001    0.999
## ytc_com      7.000e-03  5.800e+00   0.001    0.999
## twt_view     1.385e-06  1.200e-03   0.001    0.999
## twi_flw     -2.500e-05  7.962e-03  -0.003    0.997
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.0302e+01  on 63  degrees of freedom
## Residual deviance: 2.4655e-08  on 55  degrees of freedom
## AIC: 18
##
## Number of Fisher Scoring iterations: 25
```

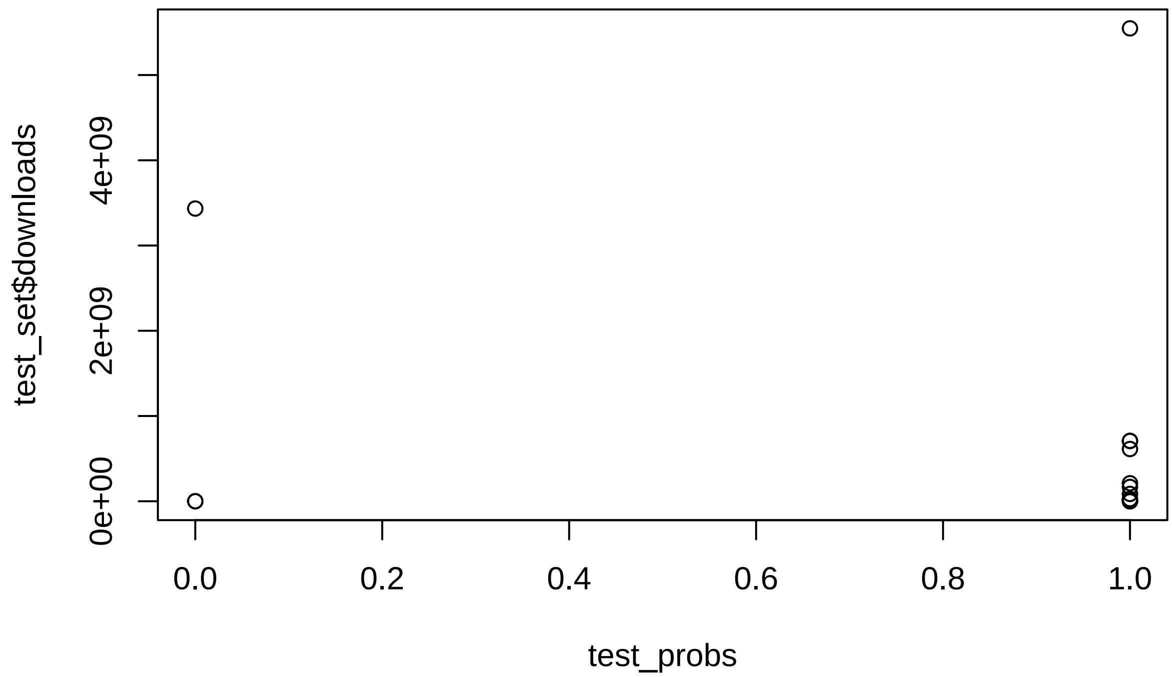
```r
# Calculate predicted probability
logit1.prob <- predict(logit1, type = "response")

plot(x = train_set$downloads, y = ifelse(train_set$downloads == "Yes", 1, 0),
     col = "orange", xlab = "Downloads", ylab = "Probablity of downloads")

points(x = train_set$downloads[order(train_set$downloads)],
       y = logit1.prob[order(train_set$downloads)],
       type = "l", col="blue", lwd = 2)
```




```r
# Calculate probability of downloads
test_probs <- predict(logit1, newdata = test_set, type="response")

# Show the first 10 values
test_probs[1:10]
```

```
##            2            3            4           22           24           30 
## 1.000000e+00 2.220446e-16 1.000000e+00 1.000000e+00 2.220446e-16 1.000000e+00 
##           33           35           37           47 
## 1.000000e+00 1.000000e+00 1.000000e+00 2.045477e-11
```

```r
# Logit model prediction
# Calculate predicted downloads
test_pred <- ifelse(test_probs >.5, "1", "0")

# Show Result
logit_result <- postResample(test_probs, test_set$downloads)
plot(test_set$downloads ~ test_probs)
```




```
logit_result
```

```
##         RMSE     Rsquared          MAE
## 1.658824e+09 1.865689e-02 7.196746e+08
```

Model 3: Gradiant Boosting Machine

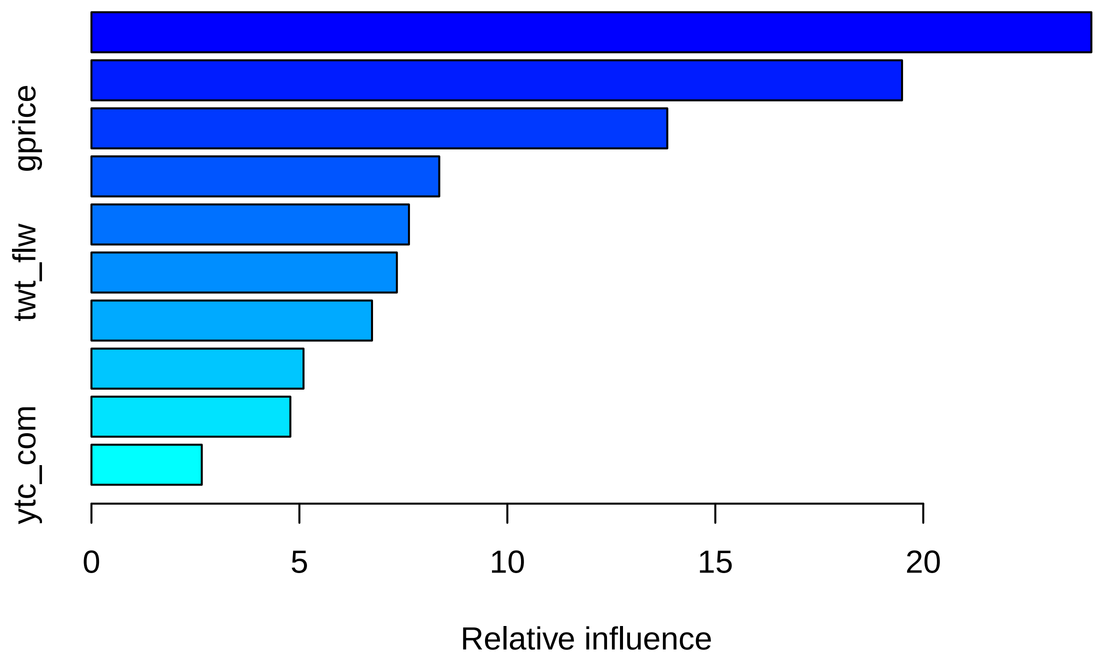
```
set.seed(123)
library(gbm)
```

```
## Loaded gbm 2.1.8.1
```

```
gbm_model <- gbm(train_set$downloads ~ gprice + diff_days + all_rev +  ytc_subs + ytc_view + ytc_likes +
                 distribution = "gaussian",
                 cv.folds = 20,
                 shrinkage =.01,
                 n.minobsinnode = 10,
                 n.trees = 1000)

print(gbm_model)
```

```
## gbm(formula = train_set$downloads ~ gprice + diff_days + all_rev +
##     ytc_subs + ytc_view + ytc_likes + ytc_com + twt_view + twt_flw +
##     twi_flw, distribution = "gaussian", data = train_set, n.trees = 1000,
##     n.minobsinnode = 10, shrinkage = 0.01, cv.folds = 20)
## A gradient boosted model with gaussian loss function.
## 1000 iterations were performed.
## The best cross-validation iteration was 217.
## There were 10 predictors of which 10 had non-zero influence.
```

```
summary(gbm_model)
```

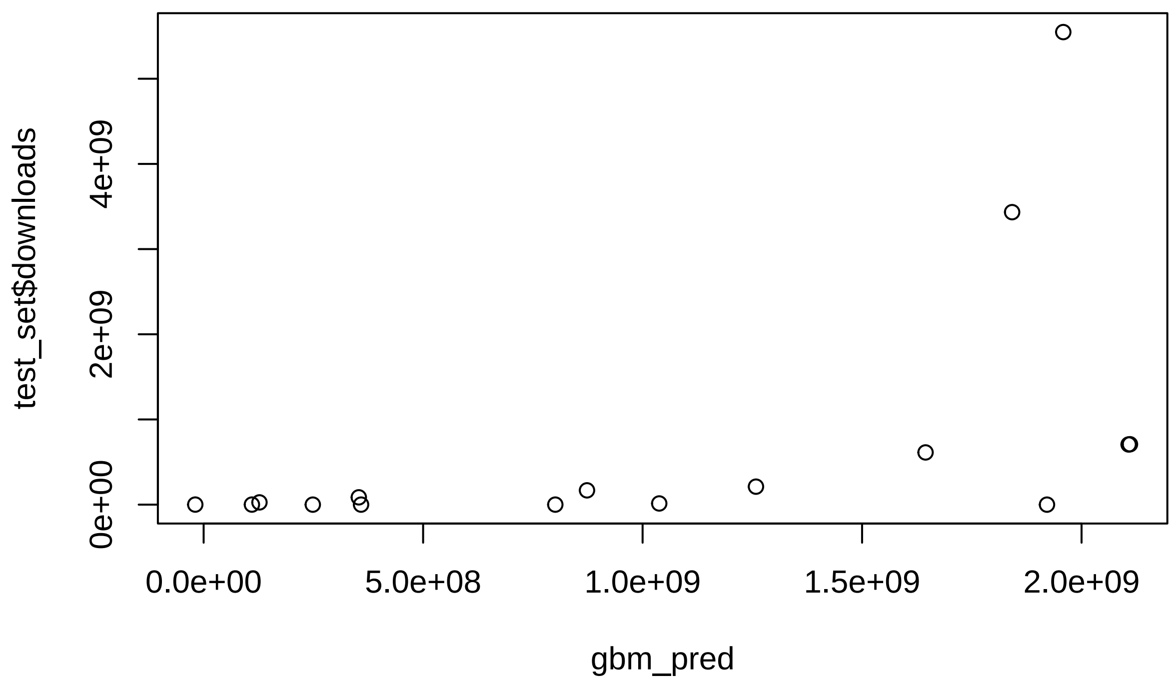



```
##                 var   rel.inf
## all_rev     all_rev 24.042015
## diff_days diff_days 19.490310
## gprice       gprice 13.845321
## twi_flw     twi_flw  8.363042
## ytc_subs   ytc_subs  7.634202
## twt_flw     twt_flw  7.343904
## twt_view   twt_view  6.746496
## ytc_view   ytc_view  5.099514
## ytc_likes ytc_likes  4.782221
## ytc_com     ytc_com  2.652974
```

```r
# GBM model prediction
test1 <- test_set[,-16]
test2 <- test_set[,16]
gbm_pred <- predict.gbm(gbm_model, test_set)
```

```
## Using 217 trees...
```

```r
gbm_result <- postResample(gbm_pred, test_set$downloads)
plot(test_set$downloads ~ gbm_pred)
```




```r
gbm_result
```

```
##         RMSE     Rsquared          MAE
## 1.315105e+09 2.734500e-01 9.762733e+08
```

Model Comparison

```r
modelcomp_df = data.frame(Model =
                        c('Multiple Linear regression1','Multiple Linear regression2',
                          'Multiple Linear regression3',
                          'Logistic regression', 'Gradiant Boosting Machine'),
```

```
                          RMSE = c(mlr1_result[["RMSE"]],
                                   mlr2_result[["RMSE"]],
                                   mlr3_result[["RMSE"]],
                                   logit_result[["RMSE"]],
                                   gbm_result[["RMSE"]]),
                          R2 = c(mlr1_result[["Rsquared"]],
                                   mlr2_result[["Rsquared"]],
                                   mlr3_result[["Rsquared"]],
                                   logit_result[["Rsquared"]],
                                   gbm_result[["Rsquared"]]),
                          MAE = c(mlr1_result[["MAE"]],
                                   mlr2_result[["MAE"]],
                                   mlr3_result[["MAE"]],
                                   logit_result[["MAE"]],
                                   gbm_result[["MAE"]]))
print(modelcomp_df)
```

```
##                          Model       RMSE         R2        MAE
## 1 Multiple Linear regression1 2968465600 0.15544860 1173314439
## 2 Multiple Linear regression2 6922269231 0.08890535 2758601520
## 3 Multiple Linear regression3 1629902533 0.25883702  790554359
## 4         Logistic regression 1658823862 0.01865689  719674552
## 5   Gradiant Boosting Machine 1315104735 0.27344998  976273259
```

Conclusion: After running all models on the data set to predict the number of downloads, the resultant performance figures does not favor the models, as their RMSE and MAE values seems to not convince that their performance was optimal. Although comparatively GBM alone did a good job with its highest R2 value among all the models, but with the results of the model, we will not choose to go with any of the above supervised learning models to predict PC game downloads w.r.t data collected from steam, twitter, youtube, and twitch