BangAndOlufsen

Henry Reith, Terence Carey, Bailey Williamson

November 20, 2018

Introduction

The purpose of this project is the utilize a data-set from the Google App Store (hosted on Kaggle) to determine the most important predictors of the success of a mobile application. We define a successful mobile application as having a relatively high number of installs. This data-set contains the following categories: App, Category, Rating, Reviews, Size, Installs, Type, Price, Content Rating, and Updates.

1. Can you even predict if an app will be successful given these categories?

2. If so, what features are most important in predicting the relative success of a mobile application?

3. What genre of mobile application is most likely to be successful?

The results of these analyses could be useful for app developers attempting to develop a successful mobile application. A person investing capital (who wants to maximize their return) in the development of such an application could use these results to determine what type of application will be most successful.

Cleaning Data

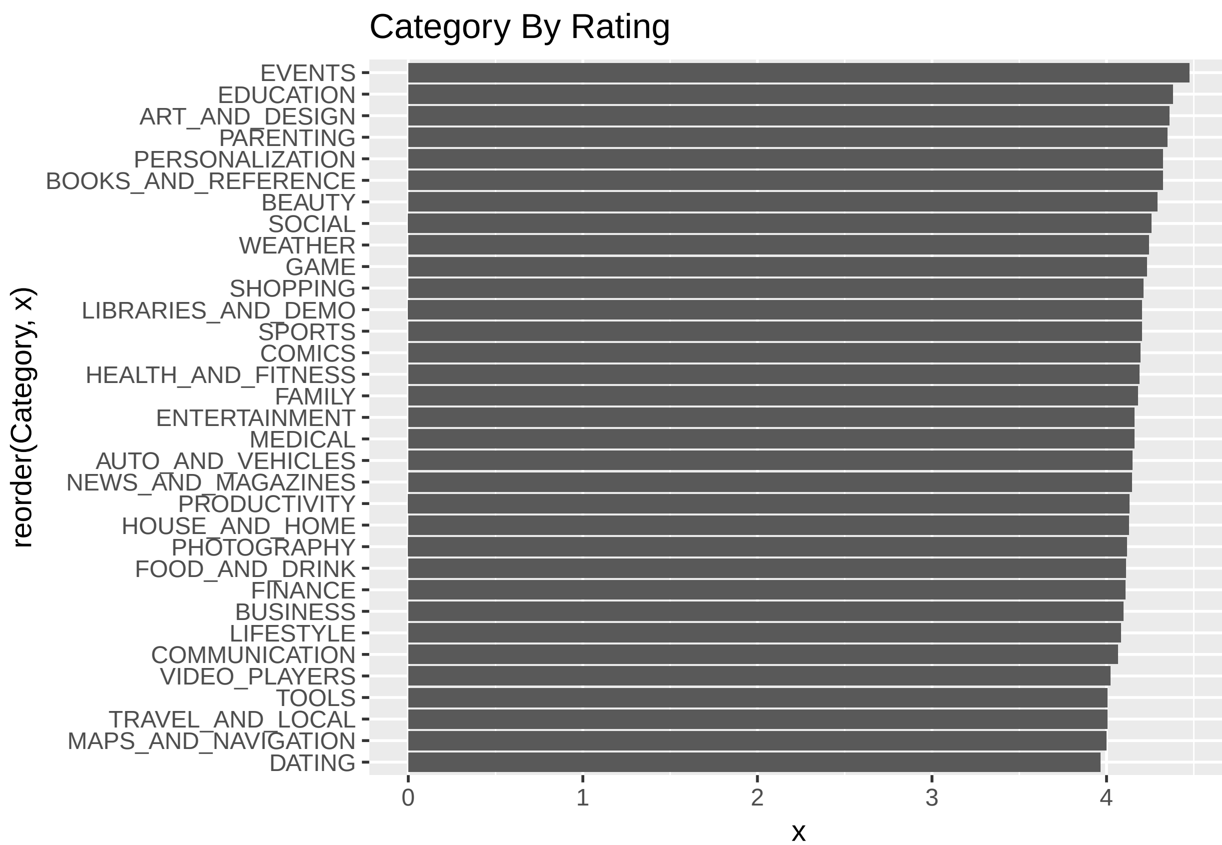
Got rid of unconsistencies, NA's, and items that we did not feel were useful. Turned installs from a numeric to factor. Added new column that put the number of updates into tiers and removed apps that had a much higher number of installations than the others. Also creates a separate data frame that still has the outliers, to show the significance of them later.

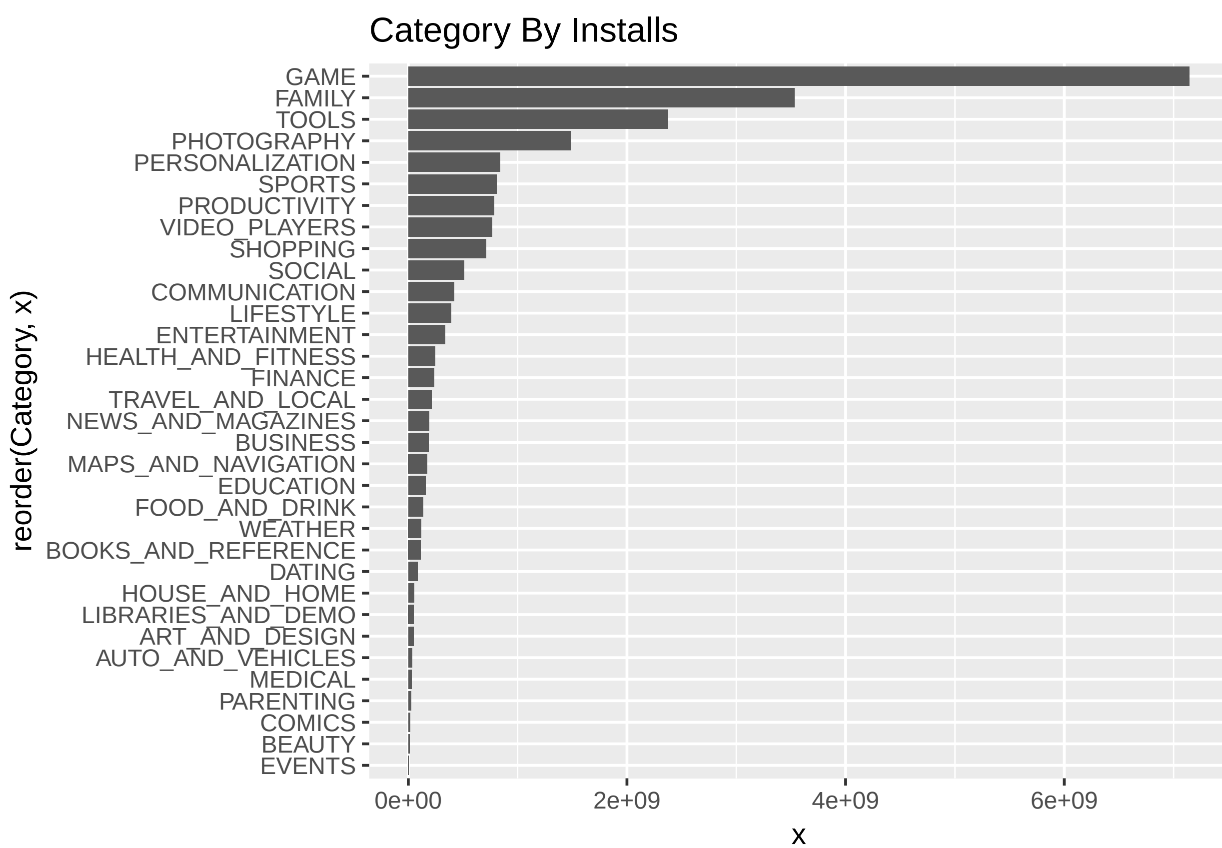
Subsetting our data and aggregating so that we can reorder for the barplots

```r
yz <- subset(d, select =c('Category', 'Installs'))
yz<-aggregate(yz$Installs, by=list(Category=yz$Category), FUN=sum)

xz <- subset(d, select =c('Category', 'Rating'))
xz<-aggregate(xz$Rating, by=list(Category=xz$Category), FUN=mean)
```

Bar Plots


Category By Rating
EVENTS
EDUCATION
ART_AND_DESIGN
PARENTING
PERSONALIZATION
BOOKS_AND_REFERENCE
BEAUTY
SOCIAL
WEATHER
GAME
SHOPPING
LIBRARIES_AND_DEMO
SPORTS
COMICS
HEALTH_AND_FITNESS
FAMILY
ENTERTAINMENT
MEDICAL
AUTO_AND_VEHICLES
NEWS_AND_MAGAZINES
PRODUCTIVITY
HOUSE_AND_HOME
PHOTOGRAPHY
FOOD_AND_DRINK
FINANCE
BUSINESS
LIFESTYLE
COMMUNICATION
VIDEO_PLAYERS
TOOLS
TRAVEL_AND_LOCAL
MAPS_AND_NAVIGATION
DATING
reorder(Category, x)
0
1
2
3
4
x


Category By Installs
GAME
FAMILY
TOOLS
PHOTOGRAPHY
PERSONALIZATION
SPORTS
PRODUCTIVITY
VIDEO_PLAYERS
SHOPPING
SOCIAL
COMMUNICATION
LIFESTYLE
ENTERTAINMENT
HEALTH_AND_FITNESS
FINANCE
TRAVEL_AND_LOCAL
NEWS_AND_MAGAZINES
BUSINESS
MAPS_AND_NAVIGATION
EDUCATION
FOOD_AND_DRINK
WEATHER
BOOKS_AND_REFERENCE
DATING
HOUSE_AND_HOME
LIBRARIES_AND_DEMO
ART_AND_DESIGN
AUTO_AND_VEHICLES
MEDICAL
PARENTING
COMICS
BEAUTY
EVENTS
reorder(Category, x)
0e+00
2e+09
4e+09
6e+09
x

Random Forest Classifier

Created a random forest classifier to predict which features determined Installs. We were succesful in building the model but could not generate a confusion matrix. Attempts for confusion matrix are commented out in code. This was a successful model with a 75% Variables explained. This is thus the best model that we have.

```
##
## Call:
##  randomForest(formula = Installs ~ Category + Type + Rating +      Reviews, data = TrainSet, nTrees =
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 3.385096e+13
##                     % Var explained: 75.55
```

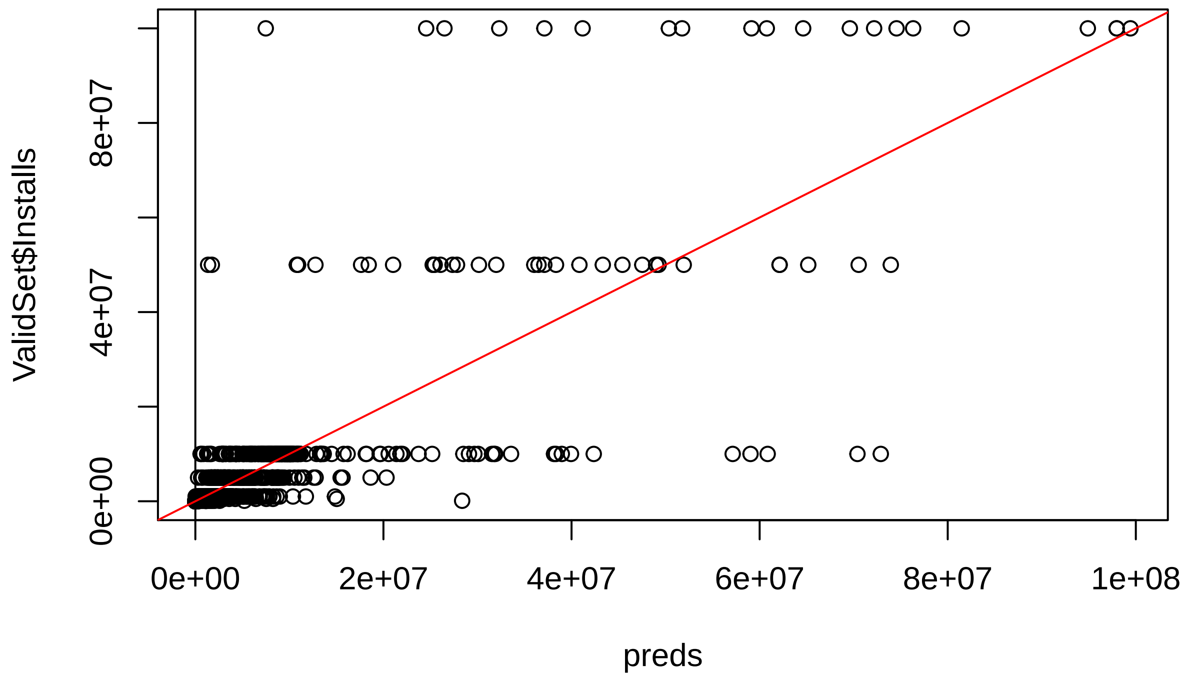
```r
preds <- predict(model1, ValidSet)
head(preds)
```

```
##     1414     1417     1435     1528     1542     1550
## 24544254 97965000 50347817 97965000 72172333 37109489
```

```r
actuals <- predict(model1, TrainSet)
head(actuals)
```

```
##      6534      2883      9266      2145      3684      4522
## 1479948.3  769210.0  862367.0 9764000.0  241748.7  204514.3
```

```r
plot(preds, ValidSet$Installs) + abline(preds,actuals) + abline(0,1, col='red')
```




```
## integer(0)
```

```r
imp <- importance(model1)
imp
```

```
##              %IncMSE IncNodePurity
## Category   14.557996  4.541727e+16
## Type        4.163696  3.495484e+14
## Rating     39.637194  4.310500e+16
## Reviews   124.064375  5.592463e+17
```

```r
model2 <- randomForest(Installs ~ Category +Type + Rating + Reviews + Price + Content.Rating + Updates


model2
```

```
##
## Call:
##  randomForest(formula = Installs ~ Category + Type + Rating +      Reviews + Price + Content.Rating +
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 3.130581e+13
##                     % Var explained: 77.38
```

```r
imp2 <- importance(model2)
imp2
```

```
##                   %IncMSE IncNodePurity
## Category        16.769087  4.824751e+16
## Type             4.015244  7.306791e+14
## Rating          30.520897  3.905009e+16
## Reviews        101.092952  5.190384e+17
## Price            4.803777  7.474450e+14
## Content.Rating  12.304594  1.597718e+16
## Updates          3.745392  1.051222e+16
```

```
FeatureImportance <- as.data.frame(imp2)
names(FeatureImportance) <- c("IncMSE", "IncNodePurity")
FeatureImportance <- FeatureImportance[order(FeatureImportance$IncMSE, decreasing =  TRUE),]
FeatureImportance
```

```
##                    IncMSE IncNodePurity
## Reviews        101.092952  5.190384e+17
## Rating          30.520897  3.905009e+16
## Category        16.769087  4.824751e+16
## Content.Rating  12.304594  1.597718e+16
## Price            4.803777  7.474450e+14
## Type             4.015244  7.306791e+14
## Updates          3.745392  1.051222e+16
```

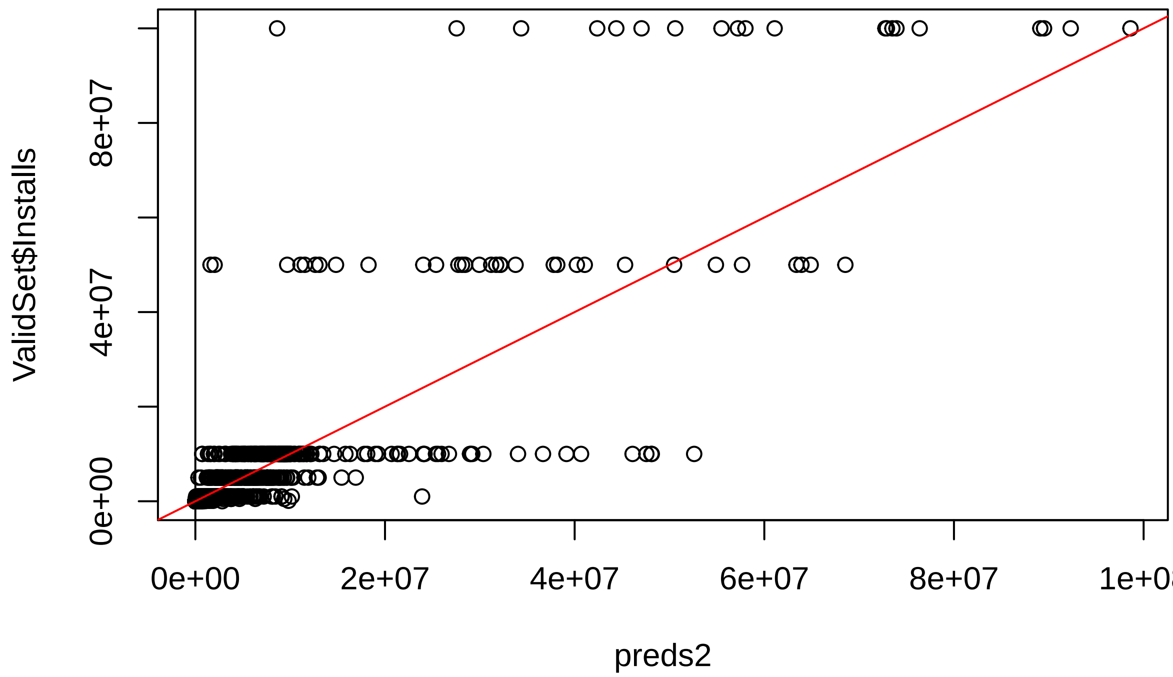
```
preds2 <- predict(model2, ValidSet)
head(preds)
```

```
##     1414     1417     1435     1528     1542     1550
## 24544254 97965000 50347817 97965000 72172333 37109489
```

```
actuals2 <- predict(model2, TrainSet)
head(actuals)
```

```
##      6534      2883      9266      2145      3684      4522
## 1479948.3  769210.0  862367.0 9764000.0  241748.7  204514.3
```

```
plot(preds2, ValidSet$Installs) + abline(preds2,actuals2) + abline(0,1, col='red')
```




```
## integer(0)
```

```r
ValidSet$preds <- preds2

write.table(ValidSet, file = 'validset.csv', sep = ",", row.names = F)
```

RPart Model Before Removal of Outliers

Created RPart model with RSquared value of .4

```
## CART
##
## 6955 samples
## 1918 predictors
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 5217, 5216, 5216, 5216
## Resampling results across tuning parameters:
##
##   cp            RMSE      Rsquared   MAE
##   1.160018e-12  20872311  0.4012833  2926554
##   3.117500e-12  20872311  0.4012833  2926550
##   9.076746e-12  20872311  0.4012833  2926554
##   4.190331e-10  20872311  0.4012833  2926520
##   9.507814e-10  20872311  0.4012833  2926645
```

```
##   8.657806e-09  20872309  0.4012832  2926536
##   1.249219e-08  20872316  0.4012828  2926306
##   2.257311e-08  20872340  0.4012812  2926134
##   1.545559e-06  20872316  0.4012805  2930147
##   4.629350e-04  20849974  0.4018835  3090333
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.000462935.
```

RPart Unsuccessful

Created RPart model with RSquared Value of .07. This model did not include our category of Reviews which shows the importance of this variable.

```
## CART
##
## 6921 samples
##   34 predictor
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 5189, 5191, 5193, 5190
## Resampling results across tuning parameters:
##
##   cp            RMSE      Rsquared    MAE
##   1.712975e-10  11296089  0.07697158  4355915
##   2.541085e-08  11296098  0.07696983  4355951
##   6.663326e-06  11296049  0.07696855  4353826
##   1.376575e-05  11295001  0.07703593  4350385
##   3.395675e-05  11291498  0.07720723  4350156
##   9.808593e-05  11297216  0.07644142  4349236
##   2.581932e-04  11309576  0.07446749  4374175
##   6.228827e-04  11323939  0.07128536  4388583
##   8.965422e-04  11330844  0.06933856  4395351
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 3.395675e-05.
```

After Removal of Outliers

RPart model with RSquared value of .69. This model was created after removing outliers and includes Reviews as a factor.

```
## CART
##
## 6921 samples
##   35 predictor
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 5190, 5191, 5191, 5191
## Resampling results across tuning parameters:
##
##   cp            RMSE     Rsquared   MAE
```

```
##   7.634509e-12  6493429  0.6945728  1705240
##   2.664735e-11  6493429  0.6945728  1705232
##   1.394441e-09  6493428  0.6945732  1705162
##   2.681428e-09  6493428  0.6945733  1705172
##   3.352496e-09  6493427  0.6945733  1705157
##   2.003040e-07  6493394  0.6945769  1703739
##   3.293002e-07  6493338  0.6945825  1703787
##   7.707849e-06  6491577  0.6947318  1705642
##   2.557046e-04  6470648  0.6963449  1737011
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.0002557046.
```

RPart Unsuccessful

Created RPart model with RSquared Value of .07. This model did not include our category of Reviews which shows the importance of this variable.

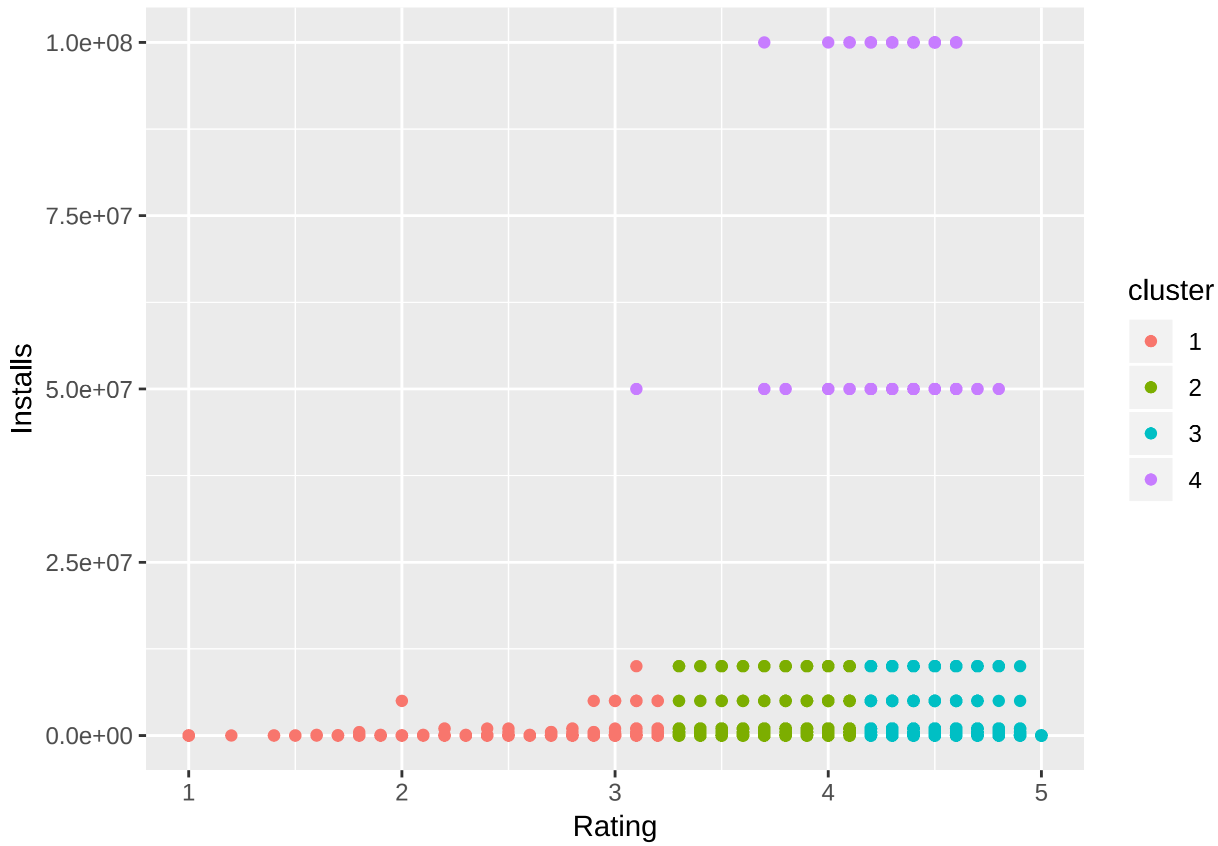
```
## CART
##
## 6921 samples
##   34 predictor
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 5192, 5191, 5191, 5189
## Resampling results across tuning parameters:
##
##   cp            RMSE      Rsquared    MAE
##   1.732176e-07  11293916  0.07318876  4336603
##   2.793866e-07  11293823  0.07320068  4336461
##   8.994665e-07  11293852  0.07319531  4336929
##   1.060652e-06  11294356  0.07313268  4336993
##   9.265291e-06  11294597  0.07304075  4335027
##   1.572876e-05  11294109  0.07306509  4334010
##   2.011649e-05  11294067  0.07303052  4332145
##   3.032684e-04  11311964  0.07012274  4357452
##   5.283633e-04  11324262  0.06720517  4379752
##   7.723539e-04  11313896  0.06802394  4380153
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 2.793866e-07.
```

Clustering

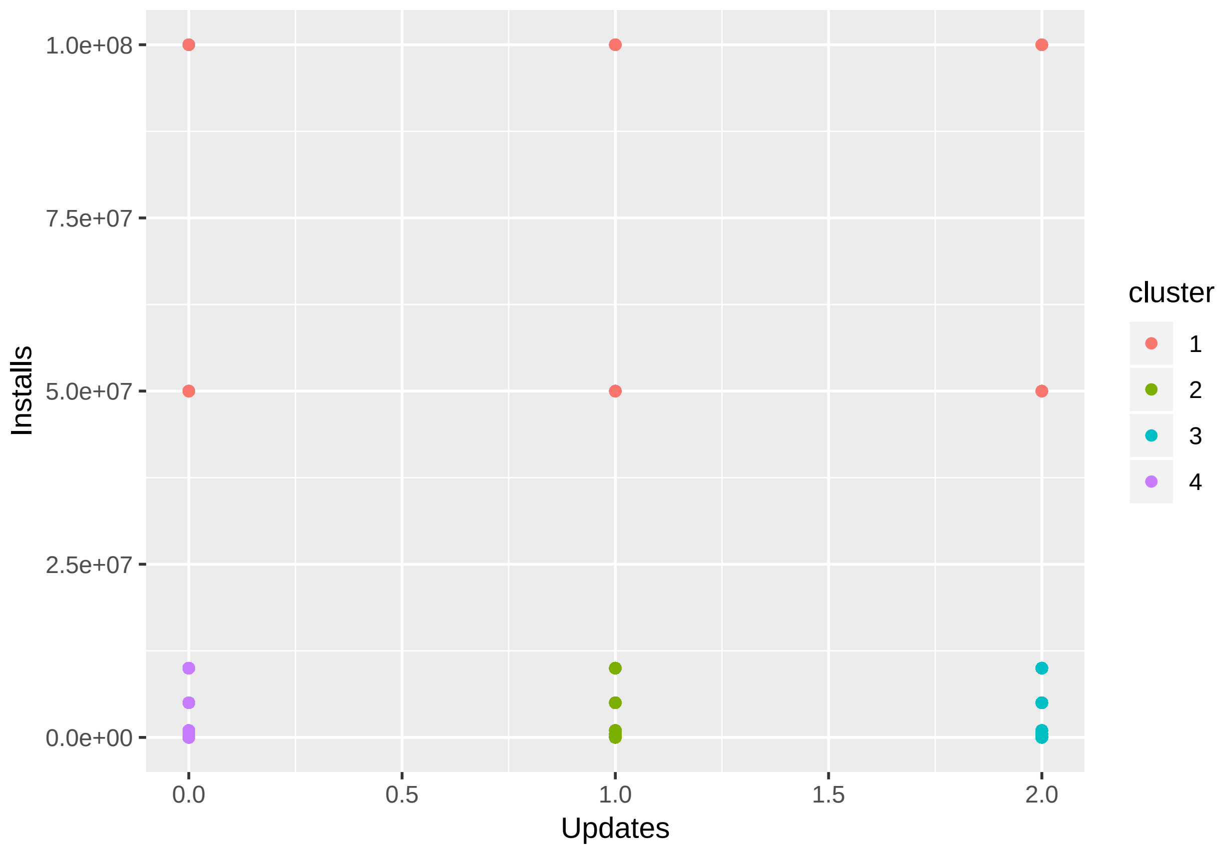
These 3 cluster models attempt to cluster ratings with various other categories in our dataset. We found little predictive capability from these models. The only potentially useful clustered result in the Ratings vs. Installs, where a clear correlation is visible in the cluster plot.

```r
#Initialize random variable
set.seed(30)
d$Reviews <- as.numeric(d$Reviews)
```

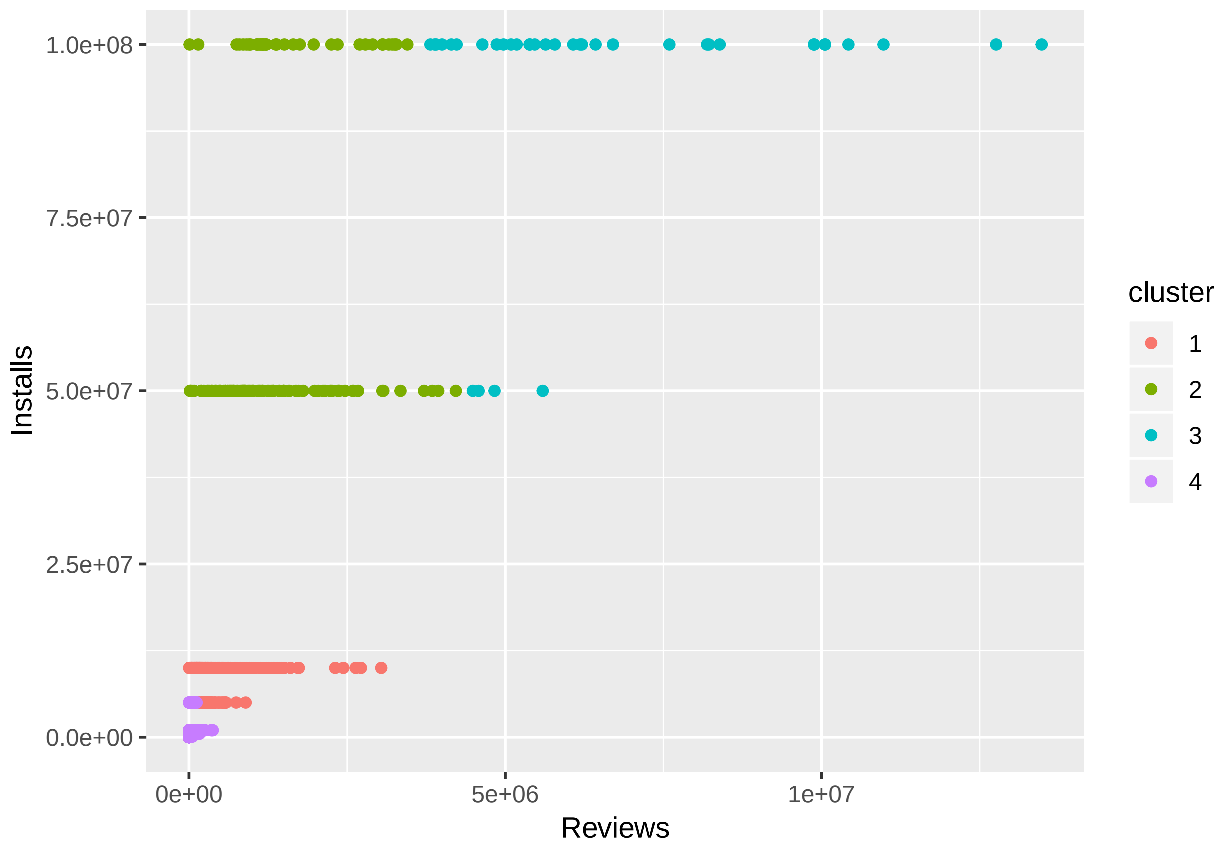
```r
d1<- subset(d, select=c("Installs", "Rating"))
clusters2<- kmeans(scale(d1), 4, nstart=25)
d1$cluster=as.factor(clusters2$cluster)
ggplot(d1, aes(x=Rating, y=Installs, color=cluster)) +geom_point()
```




```r
d2<- subset(d, select=c("Installs", "Updates"))
clusters3<- kmeans(scale(d2), 4, nstart=25)
d2$cluster=as.factor(clusters3$cluster)
ggplot(d2, aes(x=Updates, y=Installs, color=cluster)) +geom_point()
```




```
d3<- subset(d, select=c("Installs", "Reviews"))
clusters4<- kmeans(scale(d3), 4, nstart=25)
d3$cluster=as.factor(clusters4$cluster)
ggplot(d3, aes(x=Reviews, y=Installs, color=cluster)) +geom_point()
```



Executive Summary

The objective of this project was to determine which characteristics contribute to a successful mobile application. Data for this project was sourced from a Google Play Store database hosted on Kaggle. This data was then cleaned by removing unnecessary columns and removing problematic data entries (e.g. NaN values). A Random Forest classifier scheme was utilized to classify which categories in our data-set were most important in predicting a successful mobile application. From this scheme, it was determined that "Reviews", "Rating", and "Category" were the most important categories. An R-part method was utilized in R to try to predict what resulted in a high number of installs and it had a R-squared . K-means clustering was used to determine the following relationships: "Rating" vs. "Installs", "Rating" vs. "Reviews", and "Rating" vs. "Updates". The K-means analyses were ultimately not useful for predicting a successful app, since there were little to no useful clusters produced.. However, a weak positive correlation was noted for the "Rating" vs. "Installs" relationship. Ultimately, the most useful results were produced by the Random Forest analysis of our data, yielding an R-squared value of 0.77.