GPS Analysis

Najja Osiomwan

10/27/2020

Google Play Store Analysis

This is an exploratory data analysis project to examine how different variables such as the size or the price of an app affect the ratings and amount of reviews or installs an app recieves. After exploring the relationship between variables I will attempt to answer some questions about the data by testing for significance and create a linear regression model that can predict the amount of install an app will recieve based on the variables available in the dataset.

Data Cleaning

```
library(psych)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
library(gmodels)
library(MASS)
library(car)
```

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

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

```
library(cluster)
library(fpc)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(FactoMineR)
library(corrplot)
library(scatterplot3d)
library(readr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(tidyr)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.2
```

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

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

```
## The following object is masked from 'package:car':
##
##     recode
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

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

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

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following object is masked from 'package:psych':
##
##     describe
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(gplots)
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(mosaicCore)
```

```
##
## Attaching package: 'mosaicCore'
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, tally
```

```
## The following object is masked from 'package:car':
##
##     logit
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────────────────────
──────────────────── tidyverse 1.2.1 ──
```

```
## ✔ tibble  2.1.3     ✔ stringr 1.4.0
## ✔ purrr   0.3.2     ✔ forcats 0.4.0
```

```
## ── Conflicts ────────────────────────────────────────────────────────────────
──────────────── tidyverse_conflicts() ──
## ✖ ggplot2::%+%()           masks psych::%+%()
## ✖ ggplot2::alpha()         masks psych::alpha()
## ✖ lubridate::as.difftime() masks base::as.difftime()
## ✖ mosaicCore::count()      masks dplyr::count()
## ✖ lubridate::date()        masks base::date()
## ✖ dplyr::filter()          masks stats::filter()
## ✖ lubridate::intersect()   masks base::intersect()
## ✖ dplyr::lag()             masks stats::lag()
## ✖ dplyr::recode()          masks car::recode()
## ✖ dplyr::select()          masks MASS::select()
## ✖ lubridate::setdiff()     masks base::setdiff()
## ✖ purrr::some()            masks car::some()
## ✖ Hmisc::src()             masks dplyr::src()
## ✖ Hmisc::summarize()       masks dplyr::summarize()
## ✖ mosaicCore::tally()      masks dplyr::tally()
## ✖ lubridate::union()       masks base::union()
```

```
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

```
library(nnet)
library(plotrix)
```

```
##
## Attaching package: 'plotrix'
```

```
## The following object is masked from 'package:gplots':
##
##     plotCI
```

```
## The following object is masked from 'package:psych':
##
##     rescale
```

```
g_apps <- read.csv("Google-Playstore-Full.csv", header = T)
head(g_apps)
```

```
##                                          App.Name         Category
## 1                            DoorDash - Food Delivery   FOOD_AND_DRINK
## 2 TripAdvisor Hotels Flights Restaurants Attractions TRAVEL_AND_LOCAL
## 3                                            Peapod         SHOPPING
## 4                     foodpanda - Local Food Delivery   FOOD_AND_DRINK
## 5                           My CookBook Pro (Ad Free)   FOOD_AND_DRINK
## 6                             Safeway Online Shopping   FOOD_AND_DRINK
##        Rating Reviews     Installs               Size Price Content.Rating
## 1 4.548561573  305034   5,000,000+ Varies with device     0       Everyone
## 2 4.400671482 1207922 100,000,000+ Varies with device     0       Everyone
## 3 3.656329393    1967     100,000+               1.4M     0       Everyone
## 4 4.107232571  389154  10,000,000+                16M     0       Everyone
## 5 4.647752285    2291      10,000+ Varies with device $5.99       Everyone
## 6  3.82532239    2559     100,000+                23M     0       Everyone
##         Last.Updated    Minimum.Version     Latest.Version X X.1 X.2 X.3
## 1     March 29, 2019 Varies with device Varies with device            NA
## 2     March 29, 2019 Varies with device Varies with device            NA
## 3 September 20, 2018         5.0 and up              2.2.0            NA
## 4     March 22, 2019         4.2 and up             4.18.2            NA
## 5      April 1, 2019 Varies with device Varies with device            NA
## 6     March 29, 2019         5.0 and up              7.6.0            NA
```

```
str(g_apps)
```

```
## 'data.frame':    267052 obs. of  15 variables:
##  $ App.Name       : Factor w/ 244406 levels "_PRISM","--SB Kiosk App--",..: 74410 223
974 168267 91440 153012 191111 241995 213156 86853 29047 ...
##  $ Category       : Factor w/ 68 levels "","," Accounting",..: 30 66 61 30 30 30 66 30
66 30 ...
##  $ Rating         : Factor w/ 99856 levels " Economics"," Lessons",..: 75011 58414 11
903 30767 85185 16562 44402 69013 16127 74642 ...
##  $ Reviews        : Factor w/ 24545 levels "","1","10","100",..: 11780 2011 7233 1435
2 8799 9912 15192 3790 22757 20222 ...
##  $ Installs       : Factor w/ 38 levels " Xmax X","0+",..: 23 12 13 9 10 13 9 23 10 2
3 ...
##  $ Size           : Factor w/ 1248 levels "1,000,000+","1,000+",..: 1248 1248 31 153
1248 248 1248 1248 454 1248 ...
##  $ Price          : Factor w/ 504 levels "$0.56","$0.67",..: 488 488 488 488 398 488
488 488 488 488 ...
##  $ Content.Rating : Factor w/ 12 levels "$0.99","$2.49",..: 8 8 8 8 8 8 11 8 8 8 ...
##  $ Last.Updated   : Factor w/ 2751 levels "0","500,000+",..: 1761 1761 2621 1706 12 1
761 1753 1786 698 1820 ...
##  $ Minimum.Version: Factor w/ 100 levels "","0","1.0 - 6.0",..: 100 100 80 70 100 80
100 100 72 100 ...
##  $ Latest.Version : Factor w/ 22994 levels "","  1.0.1.6",..: 22890 22890 11185 16974
22890 20210 22890 22890 11936 22890 ...
##  $ X              : Factor w/ 16 levels "","1.0.0","1.0.1",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ X.1            : Factor w/ 4 levels "","1","4.4 and up",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ X.2            : Factor w/ 3 levels "","4.4 and up",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ X.3            : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
summary(g_apps)
```

```
##      App.Name                     Category              Rating
##  ????    :   766   EDUCATION          : 33394   5           : 23804
##  ?????   :   635   TOOLS              : 21592   4           :  5469
##  ??????  :   608   BOOKS_AND_REFERENCE: 21377   4.5         :  3519
##  ??????? :   415   ENTERTAINMENT      : 20604   3           :  2581
##  ????????:   334   MUSIC_AND_AUDIO    : 17876   4.333333492:  2204
##  (Other) :264293   LIFESTYLE          : 15034   4.666666508:  2167
##  NA's    :     1   (Other)            :137175   (Other)    :227308
##     Reviews           Installs                     Size
##  1      :  9203   10,000+ :60531   Varies with device: 11726
##  2      :  7581   1,000+  :48880   11M               :  7312
##  3      :  6445   100,000+:37498   12M               :  6362
##  4      :  5624   5,000+  :26360   13M               :  5569
##  5      :  4962   50,000+ :22795   14M               :  5266
##  6      :  4479   100+    :18502   15M               :  5157
##  (Other):228758   (Other) :52486   (Other)           :225660
##      Price             Content.Rating           Last.Updated
##  0      :255428   Everyone    :241578   April 2, 2019 :  4000
##  $0.99  :  2317   Teen        : 17261   April 1, 2019 :  3331
##  $1.99  :  1552   Everyone 10+:  4661   March 28, 2019:  2736
##  $2.99  :  1351   Mature 17+  :  3489   March 25, 2019:  2681
##  $4.99  :   883   Unrated     :    33   March 29, 2019:  2670
##  $3.99  :   767   0           :    12   March 26, 2019:  2650
##  (Other):  4754   (Other)     :    18   (Other)       :248984
##      Minimum.Version             Latest.Version
##  4.1 and up  :70848   1                 : 33002
##  4.0.3 and up:49324   1.1               : 11714
##  4.0 and up  :37837   Varies with device:  8555
##  4.4 and up  :28250   1.2               :  8205
##  5.0 and up  :17413   2                 :  7126
##  4.2 and up  :13629   1.3               :  5922
##  (Other)     :49751   (Other)           :192528
##                   X                         X.1
##                    :267034                   :267049
##  1.0.0            :     2   1                :     1
##  1.0.1            :     2   4.4 and up       :     1
##  Varies with device:     2   February 14, 2019:     1
##  1.1              :     1
##  1.15.1           :     1
##  (Other)          :    10
##          X.2              X.3
##            :267050   Min.   :9.1
##  4.4 and up:     1   1st Qu.:9.1
##  9.0.3     :     1   Median :9.1
##                      Mean   :9.1
##                      3rd Qu.:9.1
##                      Max.   :9.1
##                      NA's   :267051
```

Looks like there are some rows with misaligned data and columns 12, 13, 14 and 15 are all NA (completely blank) so I'll remove those columns.

```
# Removing misaligned rows
m1 <- which(as.character(g_apps$X) == "1.0.0")
m2 <- which(as.character(g_apps$X) == "1.0.1")
m3 <- which(as.character(g_apps$X) == "Varies with device")
m4 <- which(as.character(g_apps$X) == "1.1")
m5 <- which(as.character(g_apps$X) == "1.15.1")
m6 <- which(as.character(g_apps$X) == "1.2")
m7 <- which(as.character(g_apps$X) == "1.54")
m8 <- which(as.character(g_apps$X) == "1.6")
m9 <- which(as.character(g_apps$X) == "2.2.0")
m10 <- which(as.character(g_apps$X) == "2.3")
m11 <- which(as.character(g_apps$X) == "4.0.0.0")
m12 <- which(as.character(g_apps$X) == "4.0.1")
m13 <- which(as.character(g_apps$X) == "4.0.3 and up")
m14 <- which(as.character(g_apps$X) == "April 2, 2019")
m15 <- which(as.character(g_apps$X) == "Everyone")
m16 <- which(as.character(g_apps$X) == "Varies with device")

misaligned <- c(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13, m14, m15, m16)
g_apps1 <- g_apps[-misaligned, ]
head(g_apps1) # Columns X - X.3 are now empty, let's drop those columns from the dataset
```

```
##                                         App.Name         Category
## 1                          DoorDash - Food Delivery   FOOD_AND_DRINK
## 2 TripAdvisor Hotels Flights Restaurants Attractions TRAVEL_AND_LOCAL
## 3                                         Peapod         SHOPPING
## 4                   foodpanda - Local Food Delivery   FOOD_AND_DRINK
## 5                        My CookBook Pro (Ad Free)   FOOD_AND_DRINK
## 6                          Safeway Online Shopping   FOOD_AND_DRINK
##        Rating Reviews     Installs               Size Price Content.Rating
## 1 4.548561573  305034   5,000,000+ Varies with device     0       Everyone
## 2 4.400671482 1207922 100,000,000+ Varies with device     0       Everyone
## 3 3.656329393    1967     100,000+               1.4M     0       Everyone
## 4 4.107232571  389154  10,000,000+                16M     0       Everyone
## 5 4.647752285    2291      10,000+ Varies with device $5.99       Everyone
## 6  3.82532239    2559     100,000+                23M     0       Everyone
##         Last.Updated    Minimum.Version     Latest.Version X X.1 X.2 X.3
## 1     March 29, 2019 Varies with device Varies with device            NA
## 2     March 29, 2019 Varies with device Varies with device            NA
## 3 September 20, 2018         5.0 and up              2.2.0            NA
## 4     March 22, 2019         4.2 and up             4.18.2            NA
## 5      April 1, 2019 Varies with device Varies with device            NA
## 6     March 29, 2019         5.0 and up              7.6.0            NA
```

```
g_apps1 <- g_apps1[,-c(12, 13, 14,15)]
summary(g_apps1)
```

```
##      App.Name                     Category              Rating
##  ????    :   766   EDUCATION          : 33394   5           : 23804
##  ?????   :   635   TOOLS              : 21592   4           :  5469
##  ??????  :   608   BOOKS_AND_REFERENCE: 21377   4.5         :  3519
##  ??????? :   415   ENTERTAINMENT      : 20604   3           :  2581
##  ????????:   334   MUSIC_AND_AUDIO    : 17876   4.333333492:  2204
##  (Other) :264275   LIFESTYLE          : 15034   4.666666508:  2167
##  NA's    :     1   (Other)            :137157   (Other)    :227290
##     Reviews           Installs                     Size
##  1      :  9203   10,000+ :60531   Varies with device: 11726
##  2      :  7581   1,000+  :48880   11M               :  7312
##  3      :  6445   100,000+:37498   12M               :  6362
##  4      :  5622   5,000+  :26360   13M               :  5569
##  5      :  4960   50,000+ :22795   14M               :  5266
##  6      :  4479   100+    :18502   15M               :  5157
##  (Other):228744   (Other) :52468   (Other)           :225642
##      Price                Content.Rating           Last.Updated
##  0      :255428   Everyone       :241578   April 2, 2019 :  4000
##  $0.99  :  2317   Teen           : 17261   April 1, 2019 :  3331
##  $1.99  :  1552   Everyone 10+   :  4661   March 28, 2019:  2736
##  $2.99  :  1351   Mature 17+     :  3489   March 25, 2019:  2681
##  $4.99  :   883   Unrated        :    33   March 29, 2019:  2670
##  $3.99  :   767   Adults only 18+:    12   March 26, 2019:  2650
##  (Other):  4736   (Other)        :     0   (Other)       :248966
##      Minimum.Version             Latest.Version
##  4.1 and up  :70848   1                 : 33002
##  4.0.3 and up:49324   1.1               : 11714
##  4.0 and up  :37837   Varies with device:  8553
##  4.4 and up  :28250   1.2               :  8205
##  5.0 and up  :17413   2                 :  7126
##  4.2 and up  :13629   1.3               :  5922
##  (Other)     :49733   (Other)           :192512
```

Formating the data for analysis.

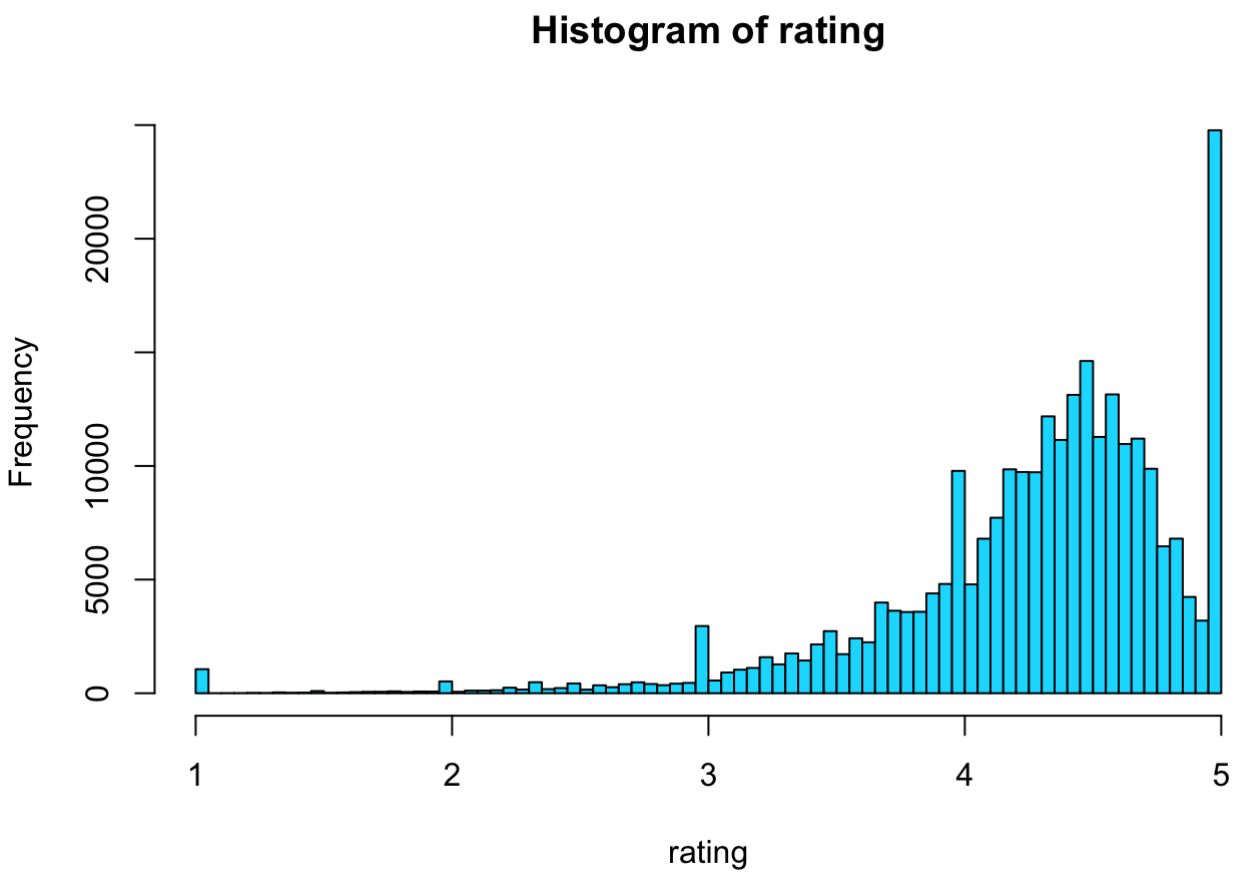
```
g_apps2 <- g_apps1
# Removing k and M from Size variable
g_apps2$Size <- as.character(g_apps2$Size)
g_apps2$Size <- gsub("\\.5k", "500", g_apps2$Size)
g_apps2$Size <- gsub("\\.1M", "100000", g_apps2$Size)
g_apps2$Size <- gsub("\\.2M", "200000", g_apps2$Size)
g_apps2$Size <- gsub("\\.3M", "300000", g_apps2$Size)
g_apps2$Size <- gsub("\\.4M", "400000", g_apps2$Size)
g_apps2$Size <- gsub("\\.5M", "500000", g_apps2$Size)
g_apps2$Size <- gsub("\\.6M", "600000", g_apps2$Size)
g_apps2$Size <- gsub("\\.7M", "700000", g_apps2$Size)
g_apps2$Size <- gsub("\\.8M", "800000", g_apps2$Size)
g_apps2$Size <- gsub("\\.9M", "900000", g_apps2$Size)
g_apps2$Size <- gsub("\\.0M","000000",g_apps2$Size)
g_apps2$Size <- gsub("\\M", "000000", g_apps2$Size)
g_apps2$Size <- gsub("\\k", "000", g_apps2$Size)
g_apps2$Size <- gsub("\\,", "", g_apps2$Size)
g_apps2$Size <- gsub("\\+", "", g_apps2$Size)
g_apps2$Size <- as.factor(g_apps2$Size)

# Removing + and , from Installs
g_apps2$Installs <- gsub("\\+", "", g_apps2$Installs)
g_apps2$Installs <- gsub(",", "", g_apps2$Installs)
g_apps2$Installs <- as.factor(g_apps2$Installs)

# Remove the $ from Price
g_apps2$Price <- gsub("\\$", "", g_apps2$Price)
g_apps2$Price <- as.factor(g_apps2$Price)

# Change last updated to date format
g_apps2$Last.Updated <- gsub("January", "1-", g_apps2$Last.Updated)
g_apps2$Last.Updated <- gsub("February", "2-", g_apps2$Last.Updated)
g_apps2$Last.Updated <- gsub("March", "3-", g_apps2$Last.Updated)
g_apps2$Last.Updated <- gsub("April", "4-", g_apps2$Last.Updated)
g_apps2$Last.Updated <- gsub("May", "5-", g_apps2$Last.Updated)
g_apps2$Last.Updated <- gsub("June", "6-", g_apps2$Last.Updated)
g_apps2$Last.Updated <- gsub("July", "7-", g_apps2$Last.Updated)
g_apps2$Last.Updated <- gsub("August", "8-", g_apps2$Last.Updated)
g_apps2$Last.Updated <- gsub("September", "9-", g_apps2$Last.Updated)
g_apps2$Last.Updated <- gsub("October", "10-", g_apps2$Last.Updated)
g_apps2$Last.Updated <- gsub("November", "11-", g_apps2$Last.Updated)
g_apps2$Last.Updated <- gsub("December", "12-", g_apps2$Last.Updated)
g_apps2$Last.Updated <- gsub(",", "-", g_apps2$Last.Updated)
g_apps2$Last.Updated <- gsub(" ", "", g_apps2$Last.Updated)

g_apps2$Last.Updated <- strptime(g_apps2$Last.Updated,format="%m-%d-%Y")
g_apps2$Last.Updated <- as.Date(g_apps2$Last.Updated, format = "%d-%m-%y")

# Remove "and up" from Minimum Version
g_apps2$Minimum.Version <- gsub(" and up", "+", g_apps2$Minimum.Version)
g_apps2$Minimum.Version <- as.factor(g_apps2$Minimum.Version)

summary(g_apps2)
```

```
##      App.Name                     Category              Rating
##  ????    :   766   EDUCATION          : 33394   5           : 23804
##  ?????   :   635   TOOLS              : 21592   4           :  5469
##  ??????  :   608   BOOKS_AND_REFERENCE: 21377   4.5         :  3519
##  ??????? :   415   ENTERTAINMENT      : 20604   3           :  2581
##  ????????:   334   MUSIC_AND_AUDIO    : 17876   4.333333492:  2204
##  (Other) :264275   LIFESTYLE          : 15034   4.666666508:  2167
##  NA's    :     1   (Other)            :137157   (Other)    :227290
##     Reviews          Installs                      Size
##  1       :  9203   10000  :60531   Varies with device: 11726
##  2       :  7581   1000   :48880   11000000          :  7312
##  3       :  6445   100000 :37498   12000000          :  6362
##  4       :  5622   5000   :26360   13000000          :  5569
##  5       :  4960   50000  :22795   14000000          :  5266
##  6       :  4479   100    :18502   15000000          :  5157
##  (Other):228744   (Other):52468   (Other)           :225642
##      Price                Content.Rating    Last.Updated
##  0       :255428   Everyone       :241578   Min.   :2009-02-11
##  0.99    :  2317   Teen           : 17261   1st Qu.:2018-04-30
##  1.99    :  1552   Everyone 10+   :  4661   Median :2018-11-20
##  2.99    :  1351   Mature 17+     :  3489   Mean   :2018-06-22
##  4.99    :   883   Unrated        :    33   3rd Qu.:2019-02-22
##  3.99    :   767   Adults only 18+:    12   Max.   :2019-04-04
##  (Other):  4736   (Other)        :     0
##  Minimum.Version           Latest.Version
##  4.1+   :70848   1                 : 33002
##  4.0.3+ :49324   1.1               : 11714
##  4.0+   :37837   Varies with device:  8553
##  4.4+   :28250   1.2               :  8205
##  5.0+   :17413   2                 :  7126
##  4.2+   :13629   1.3               :  5922
##  (Other):49733   (Other)           :192512
```

Univariate Analysis

Now I'll take a look at each variable and check for a normal distribution since the tests I'll be performing assume normal distribution.

```
g_apps2$Rating <- as.numeric(as.character(g_apps2$Rating))
g_apps2$Reviews <- as.numeric(as.character(g_apps2$Reviews))
g_apps3 <- g_apps2

# -------- Rating ----------#
rating <- g_apps3$Rating
hist(rating, breaks = 100, col="#00DCFF")
```



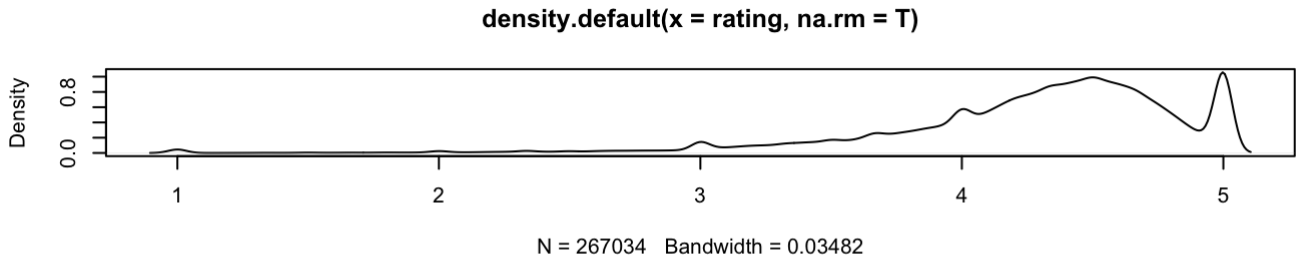

```
summary(rating)
```

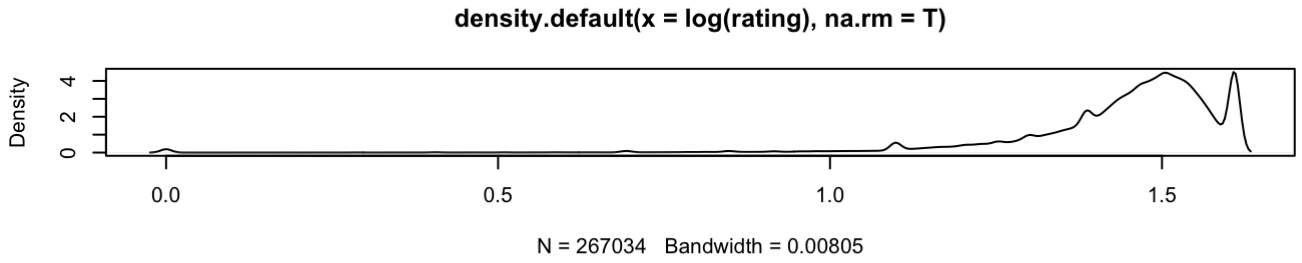
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.018   4.382   4.269   4.649   5.000
```

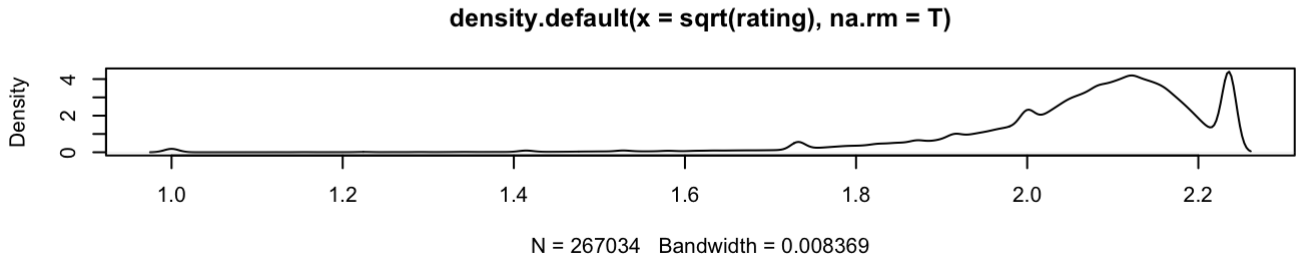
```
len_rate <- length(which(rating>=3))
len_rateP <- len_rate/(length(rating))
len_rateP # 96.9% of apps are rating 3 or higher
```

```
## [1] 0.9695095
```

```
# Look for normal distribution
par(mfrow=c(3,1))
plot(density(rating, na.rm = T))
plot(density(log(rating), na.rm = T))
plot(density(sqrt(rating), na.rm = T))
```








```
low_outliers <- fivenum(rating, na.rm = T)[2]-IQR(rating, na.rm = T)*1.5
low_outliers # 3.07
```

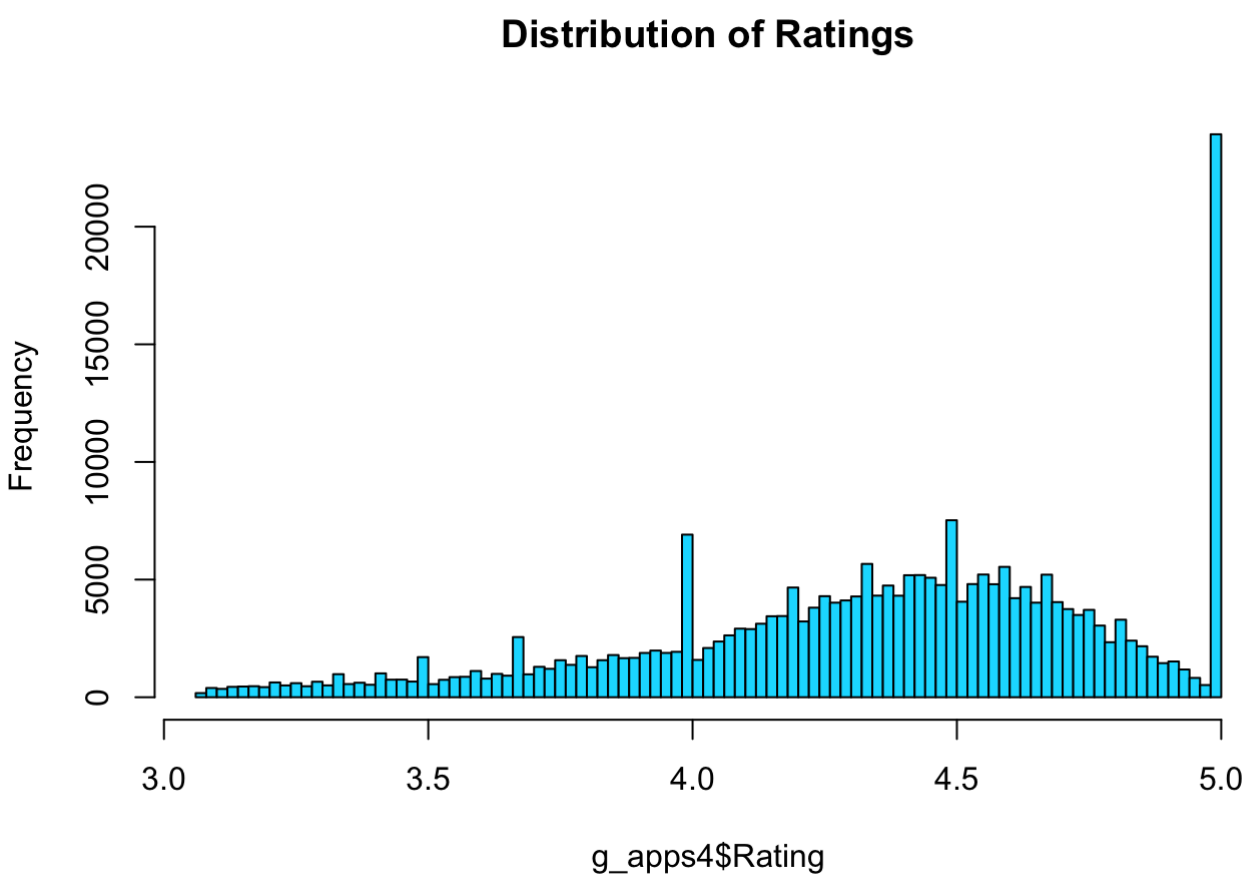
```
## [1] 3.071228
```

```
low <- which(rating < low_outliers)
length(low) # 11627 lower outliers because the data is left skewed, removing for normal
 distribution
```

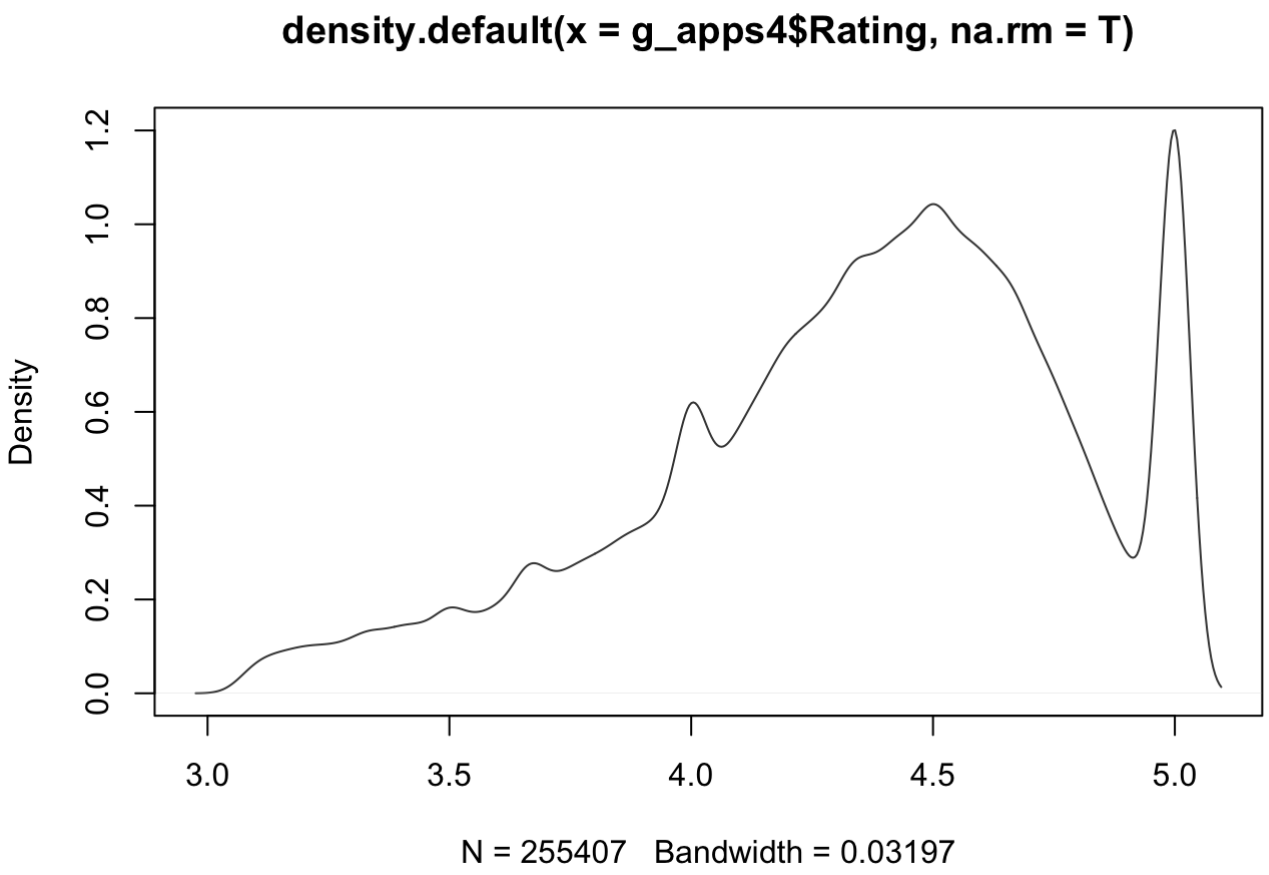
```
## [1] 11627
```

We can see that the all transformations of the data are skewed, so I'll try removing low outliers

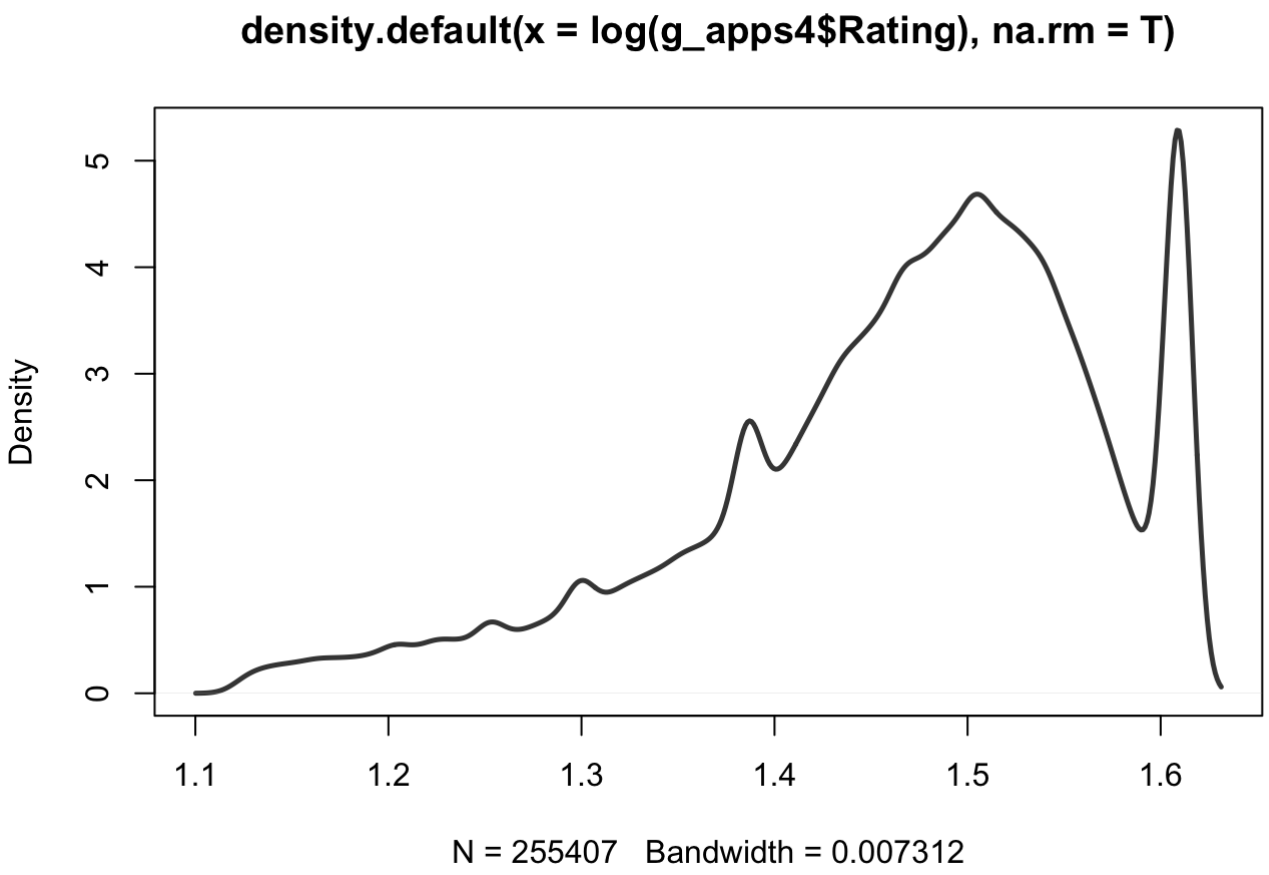
```
g_apps4 <- g_apps3[-low,]
par(mfrow=c(1,1))
hist(g_apps4$Rating, col = c("#00DCFF"), main = "Distribution of Ratings", breaks = 100)
```



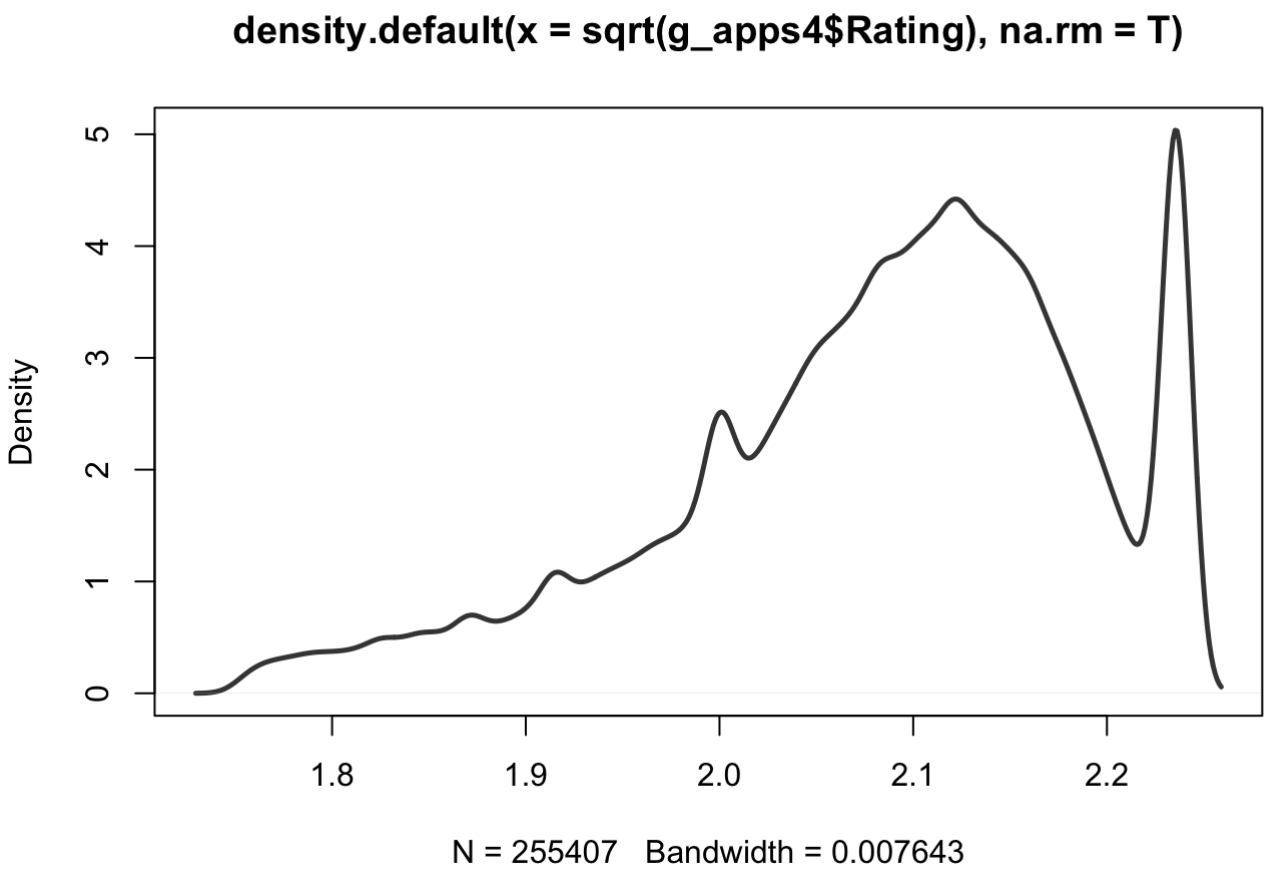

```
plot(density(g_apps4$Rating, na.rm = T), col = "#444444")
```



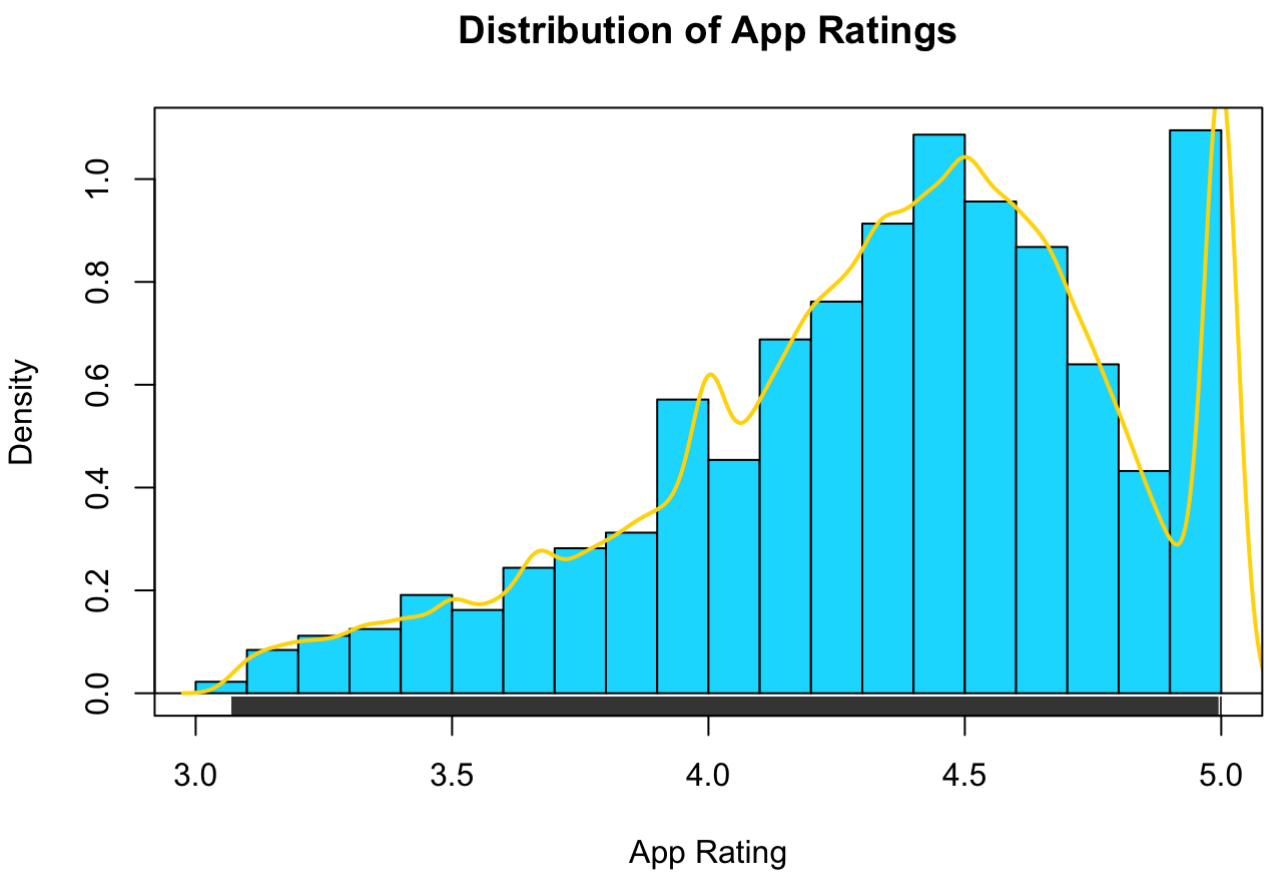

```
plot(density(log(g_apps4$Rating), na.rm = T), col="#444444", lwd=2.5)
```



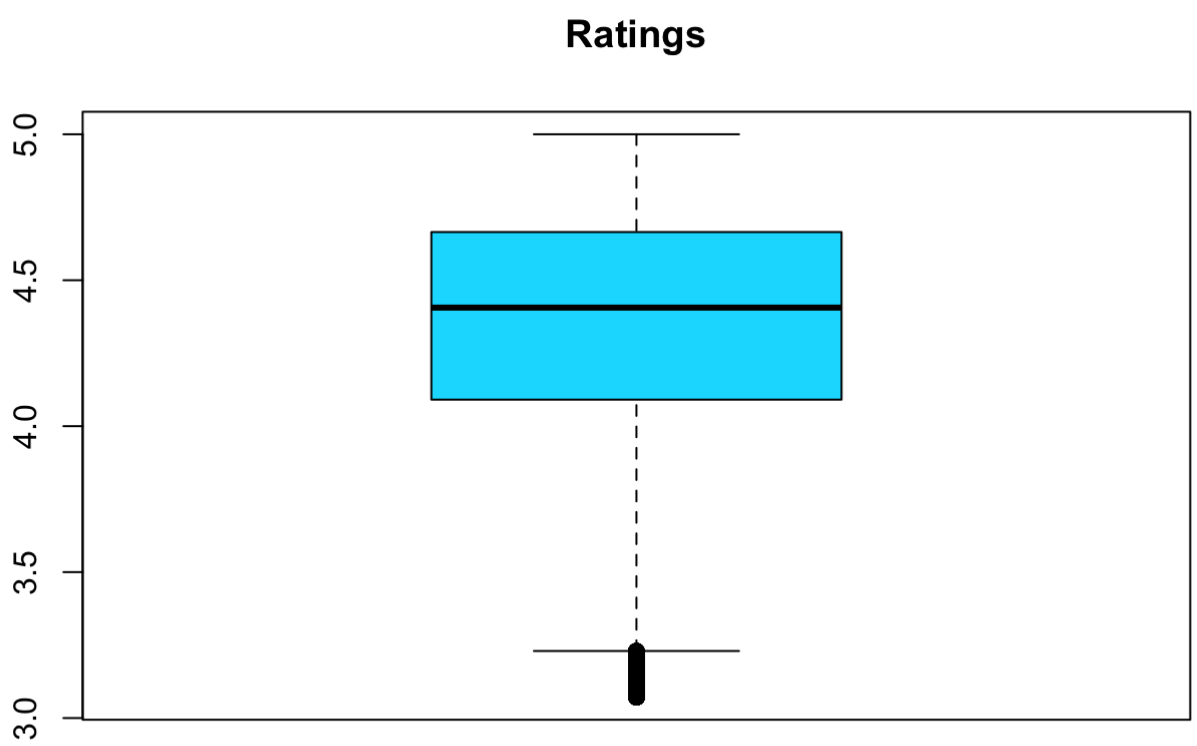

```
plot(density(sqrt(g_apps4$Rating), na.rm = T), col="#444444", lwd=2.5)
```




```
hist(g_apps4$Rating, col=c("#00DCFF"), xlim=c(3,5), freq = F, breaks=seq(.5, 20, 0.1),
     xlab = "App Rating", main = "Distribution of App Ratings")
rug(jitter(g_apps4$Rating), col="#444444")
lines(density(g_apps4$Rating), col="#FFD800", lwd=2)
box()
```




```
boxplot(g_apps4$Rating, main="Ratings", col="#00DCFF")
```



The ratings are still left skewed, but I will leave it as is.

```
# --------- Category --------#
cat <- table(g_apps4$Category)
sortcat <- sort(cat, decreasing = TRUE)
pcat <- round(prop.table(cat), 2)
pcat <- sort(pcat, decreasing = TRUE)
pcat
```

```
##
##                     EDUCATION             BOOKS_AND_REFERENCE
##                          0.13                            0.08
##                 ENTERTAINMENT                           TOOLS
##                          0.08                            0.08
##               MUSIC_AND_AUDIO                       LIFESTYLE
##                          0.07                            0.06
##                      BUSINESS                         FINANCE
##                          0.04                            0.04
##               PERSONALIZATION             HEALTH_AND_FITNESS
##                          0.04                            0.03
##             NEWS_AND_MAGAZINES                    PHOTOGRAPHY
##                          0.03                            0.03
##                  PRODUCTIVITY                  COMMUNICATION
##                          0.03                            0.02
##                      SHOPPING                        SOCIAL
##                          0.02                            0.02
##                        SPORTS               TRAVEL_AND_LOCAL
##                          0.02                            0.02
##                ART_AND_DESIGN              AUTO_AND_VEHICLES
##                          0.01                            0.01
##                FOOD_AND_DRINK                   GAME_ACTION
##                          0.01                            0.01
##                   GAME_ARCADE                   GAME_CASUAL
##                          0.01                            0.01
##              GAME_EDUCATIONAL                   GAME_PUZZLE
##                          0.01                            0.01
##               GAME_SIMULATION            MAPS_AND_NAVIGATION
##                          0.01                            0.01
##                       MEDICAL                 VIDEO_PLAYERS
##                          0.01                            0.01
##                       WEATHER
##                          0.01                            0.00
##                    Accounting                    Alfabe ?ren
##                          0.00                            0.00
##                  Breaking News                  Channel 2 News
##                          0.00                            0.00
##                   ETEA & MDCAT  Islamic Name Boy & Girl+Meaning
##                          0.00                            0.00
##                      Mexpost)        not notified you follow -
##                          0.00                            0.00
##                      Podcasts   Romantic Song Music Love Songs
##                          0.00                            0.00
##                Speaker Pro 2019       super loud speaker booster
##                          0.00                            0.00
##                     Tour Guide                    T rk Alfabesi
##                          0.00                            0.00
##                             )                               6
##                          0.00                            0.00
##                        BEAUTY                          COMICS
##                          0.00                            0.00
##                        DATING                          EVENTS
##                          0.00                            0.00
```

```
##                GAME_ADVENTURE                    GAME_BOARD
##                          0.00                          0.00
##                     GAME_CARD                   GAME_CASINO
##                          0.00                          0.00
##                    GAME_MUSIC                   GAME_RACING
##                          0.00                          0.00
##             GAME_ROLE_PLAYING                   GAME_SPORTS
##                          0.00                          0.00
##                GAME_STRATEGY                   GAME_TRIVIA
##                          0.00                          0.00
##                     GAME_WORD                    Gate ALARM
##                          0.00                          0.00
##               HOUSE_AND_HOME              LIBRARIES_AND_DEMO
##                          0.00                          0.00
##                    PARENTING                        TRAVEL
##                          0.00                          0.00
```

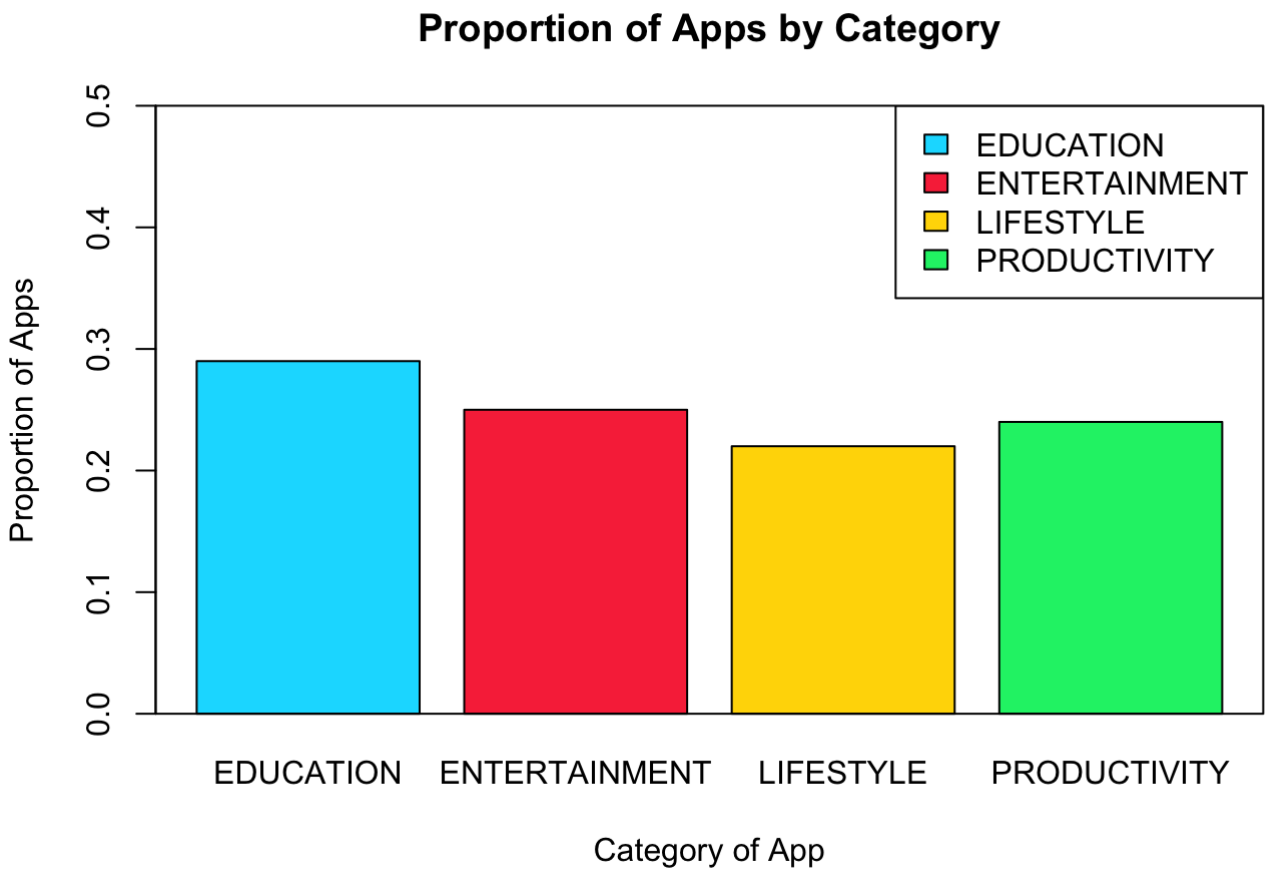
Too many catergories; grouping them together for a more even spread.

```
g_apps5 <- g_apps4
g_apps5$Category <- as.character(g_apps5$Category)

g_apps5$Category[g_apps5$Category %in% "BUSINESS"] <- "EDUCATION"
g_apps5$Category[g_apps5$Category %in% "FINANCE"] <- "EDUCATION"
g_apps5$Category[g_apps5$Category %in% "ETEA & MDCAT"] <- "EDUCATION"
g_apps5$Category[g_apps5$Category %in% "PARENTING"] <- "EDUCATION"
g_apps5$Category[g_apps5$Category %in% "BOOKS_AND_REFERENCE"] <- "EDUCATION"
g_apps5$Category[g_apps5$Category %in% "LIBRARIES_AND_DEMO"] <- "EDUCATION"

g_apps5$Category[g_apps5$Category %in% "HEALTH_AND_FITNESS"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "FOOD_AND_DRINK"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "MEDICAL"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "BEAUTY"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "HEALTH"] <- "LIFESTYLE"


g_apps5$Category[g_apps5$Category %in% "NEWS_AND_MAGAZINES"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "SHOPPING"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "TRAVEL_AND_LOCAL"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "ART_AND_DESIGN"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "AUTO_AND_VEHICLES"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "Mexpost)"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "not notified you follow -"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "TÔøΩrk Alfabesi"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "Alfabe ÔøΩ?ren"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% ")"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "6"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "Tour Guide"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "TRAVEL"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "BEAUTY"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "Mexpost)"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "Breaking News"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "Channel 2 News"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "DATING"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "EVENTS"] <- "LIFESTYLE"
g_apps5$Category[g_apps5$Category %in% "SOCIAL"] <- "LIFESTYLE"


g_apps5$Category[g_apps5$Category %in% "PHOTOGRAPHY"] <- "PRODUCTIVITY"
g_apps5$Category[g_apps5$Category %in% "COMMUNICATION"] <- "PRODUCTIVITY"
g_apps5$Category[g_apps5$Category %in% "MAPS_AND_NAVIGATION"] <- "PRODUCTIVITY"
g_apps5$Category[g_apps5$Category %in% "VIDEO_PLAYERS"] <- "PRODUCTIVITY"
g_apps5$Category[g_apps5$Category %in% "WEATHER"] <- "PRODUCTIVITY"
g_apps5$Category[g_apps5$Category %in% "HOUSE_AND_HOME"] <- "PRODUCTIVITY"
g_apps5$Category[g_apps5$Category %in% "VIDEO_PLAYERS"] <- "PRODUCTIVITY"
g_apps5$Category[g_apps5$Category %in% "Gate ALARM"] <- "PRODUCTIVITY"
g_apps5$Category[g_apps5$Category %in% "Islamic Name Boy & Girl+Meaning"] <- "PRODUCTIVI
TY"
g_apps5$Category[g_apps5$Category %in% "PERSONALIZATION"] <- "PRODUCTIVITY"
g_apps5$Category[g_apps5$Category %in% "TOOLS"] <- "PRODUCTIVITY"
```

```
g_apps5$Category[g_apps5$Category %in% "MUSIC_AND_AUDIO"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "Romantic Song Music Love Songs"] <- "ENTERTAINME
NT"
g_apps5$Category[g_apps5$Category %in% "Speaker Pro 2019"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "super loud speaker booster"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "MUSIC_AND_AUDIO"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "SPORTS"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_ACTION"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_ARCADE"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_CASUAL"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_EDUCATIONAL"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_PUZZLE"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_SIMULATION"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "COMICS"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_ADVENTURE"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_BOARD"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_CARD"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_CASINO"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_MUSIC"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_RACING"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_ROLE_PLAYING"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_SPORTS"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_STRATEGY"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_TRIVIA"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "GAME_WORD"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "Podcasts"] <- "ENTERTAINMENT"
g_apps5$Category[g_apps5$Category %in% "MUSIC"] <- "ENTERTAINMENT"

g_apps5$Category <- as.factor(g_apps5$Category)

ct <- table(g_apps5$Category)
pct <- prop.table(ct)
pct <- round(pct, 2)
pct
```

```
##
##     EDUCATION ENTERTAINMENT     LIFESTYLE  PRODUCTIVITY
##          0.29          0.25          0.22          0.24
```

```
par(mfrow=c(1,1))
barplot(pct, main = "Proportion of Apps by Category", col = c("#00DCFF", "#F83648", "#FF
D800", "#04F075"), ylim = c(0,0.5), xlab = "Category of App", ylab = "Proportion of App
s")
legend("topright", fill = c("#00DCFF", "#F83648", "#FFD800", "#04F075"), legend = levels
(g_apps5$Category))
box()
```



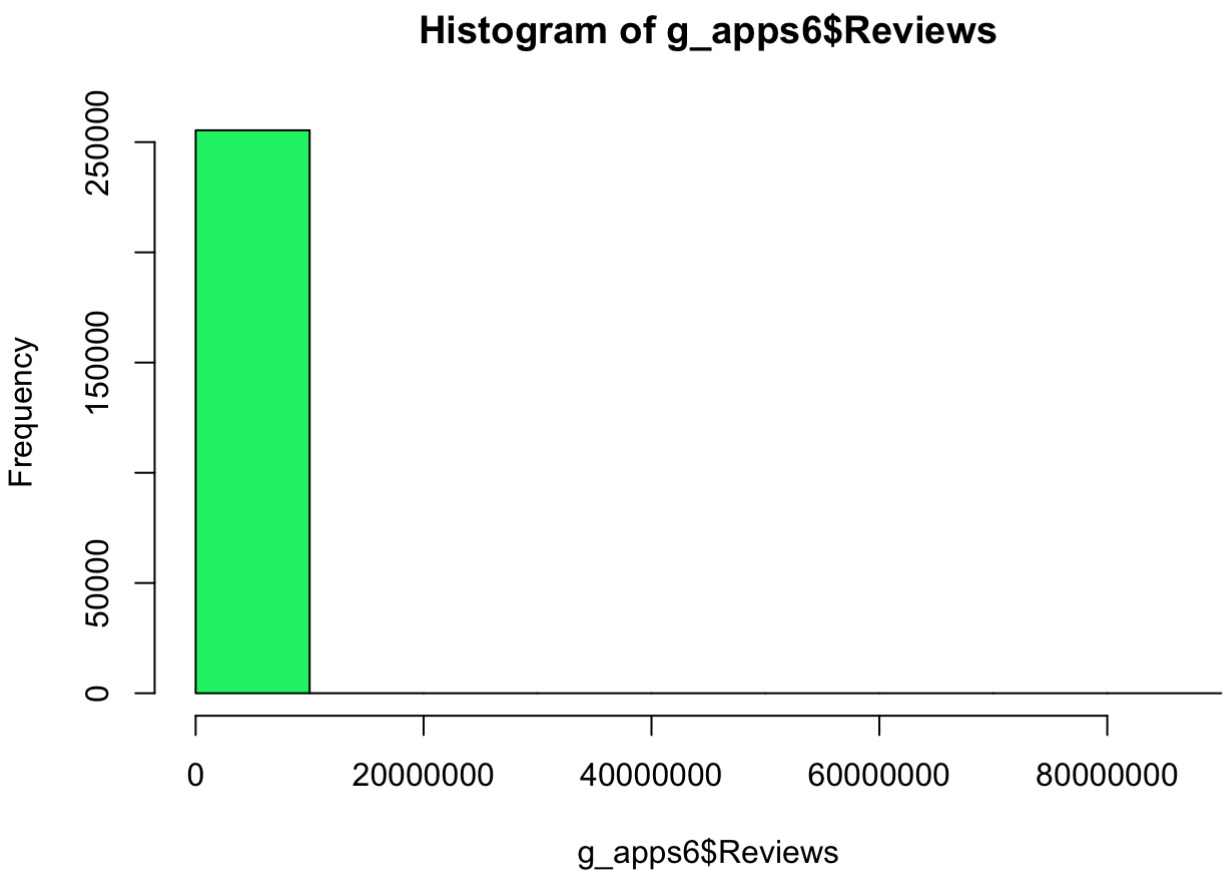

```
g_apps6 <- g_apps5

str(g_apps6$Reviews)
```

```
##  num [1:255407] 305034 1207922 1967 389154 2291 ...
```

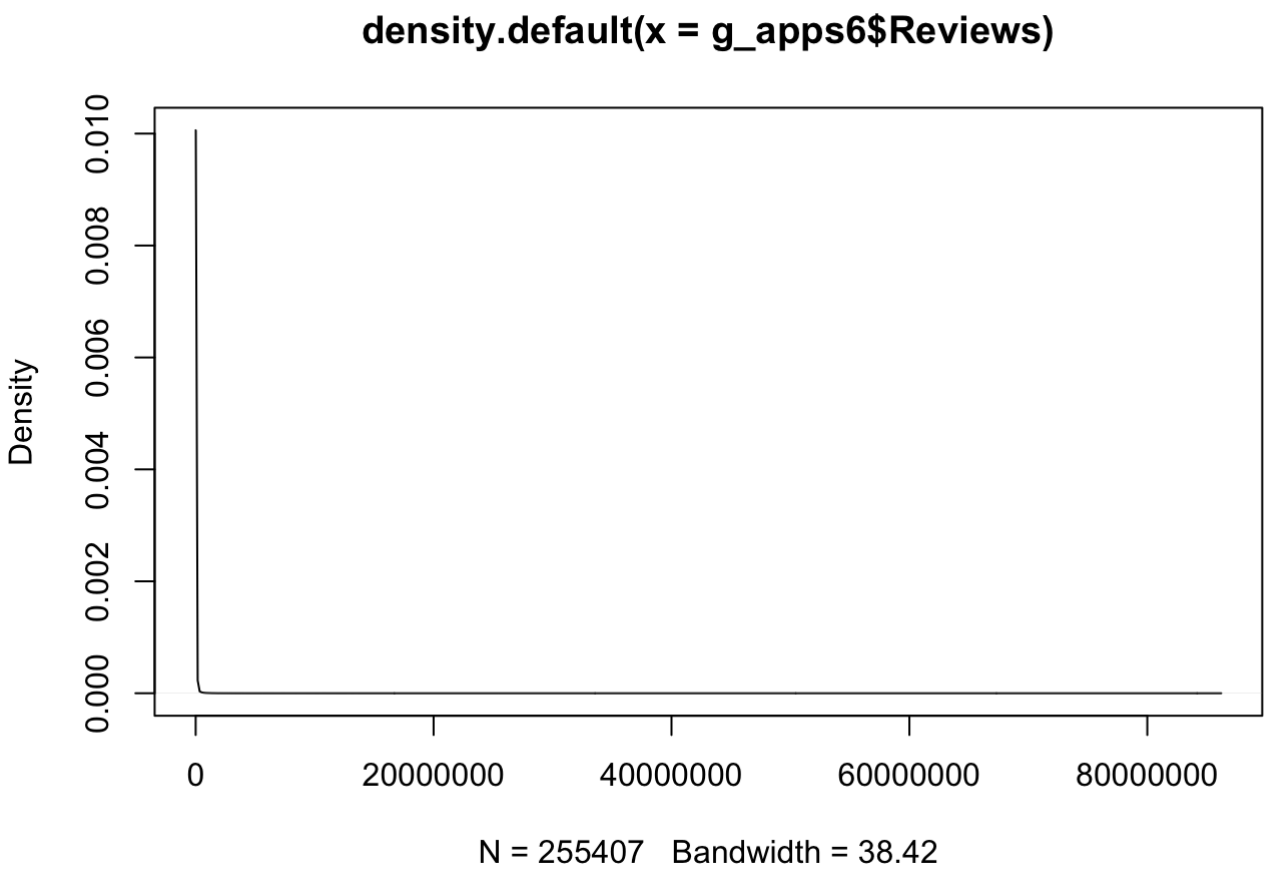
```
summary(g_apps6$Reviews)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##        1       18      101    15246      708 86214292
```

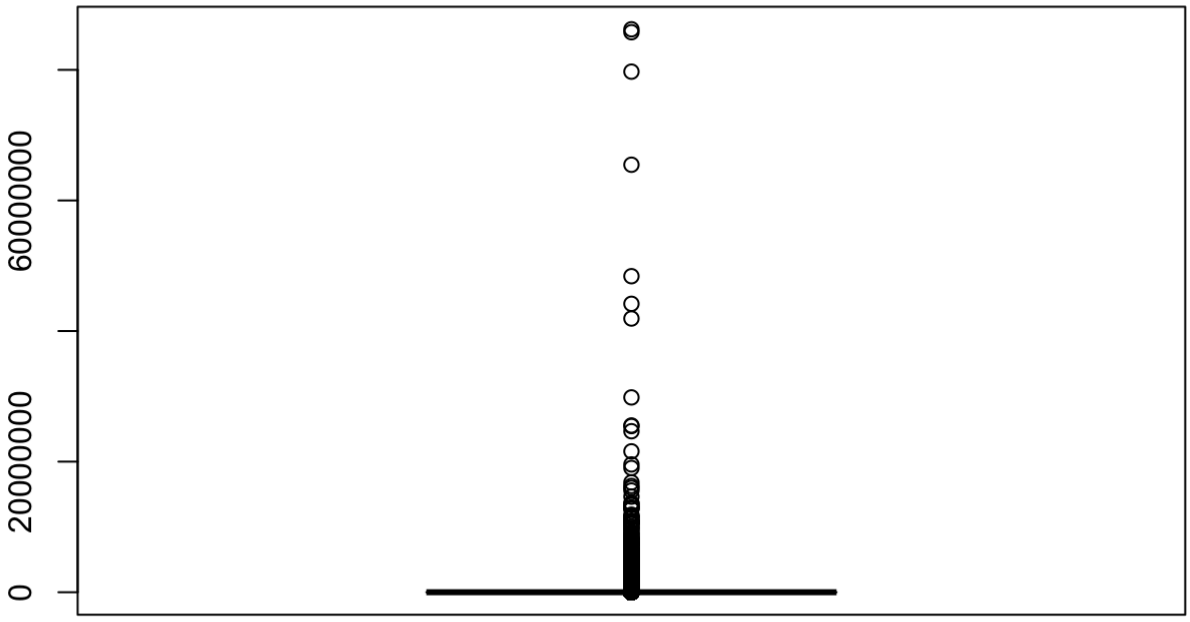
```
hist(g_apps6$Reviews, breaks = 10, col = "#04F075")
```



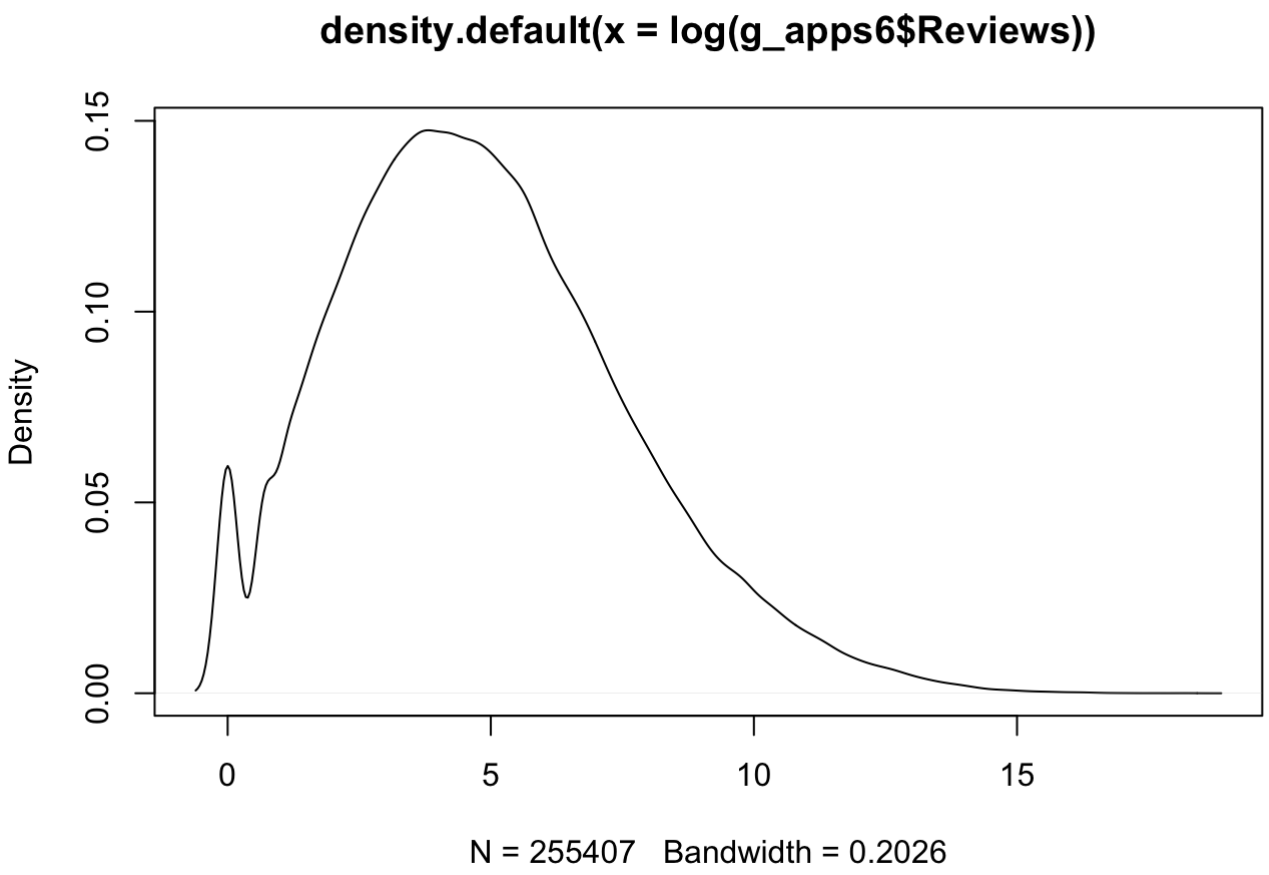

```
plot(density(g_apps6$Reviews)) # Nowhere near normal distribution
```



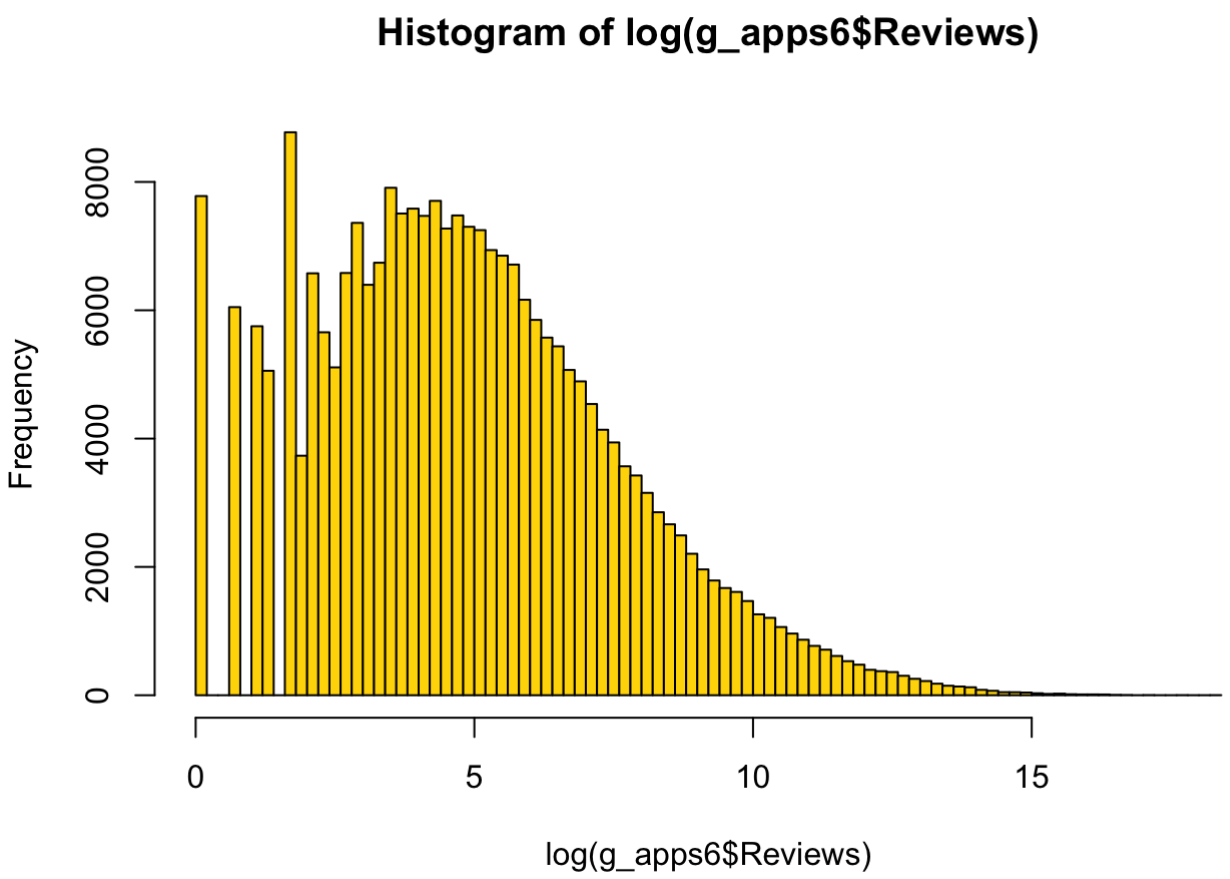

```
boxplot(g_apps6$Reviews, col = "#00DCFF")
```




```
options(scipen = 99)

par(mfrow=c(1,1))
plot(density(log(g_apps6$Reviews)))
```




```
hist(log(g_apps6$Reviews), col = "#FFD800", breaks = 100) # Way better, but right skewed
a bit
```



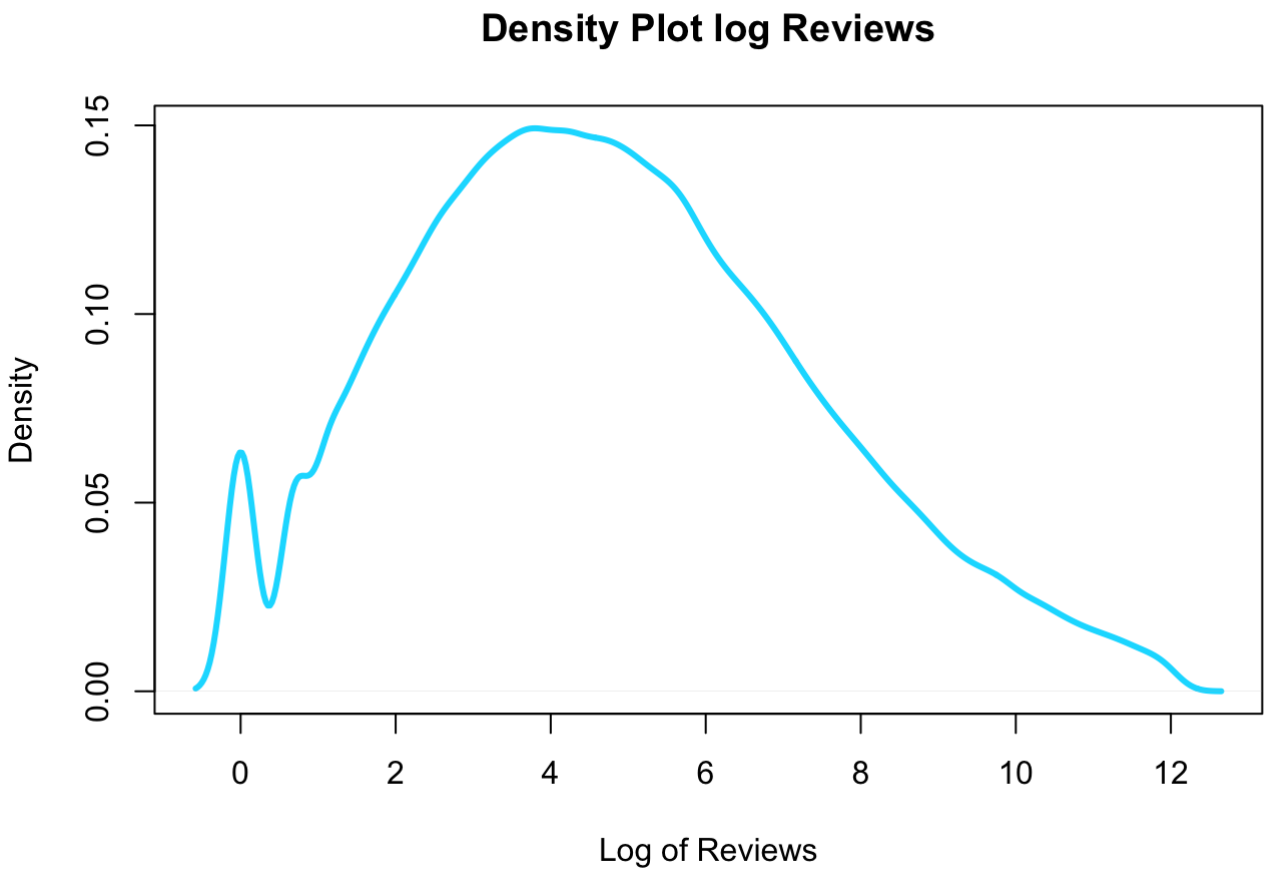

```
# Removing outliers
log_reviews <- log(g_apps6$Reviews)
revup_limit <- fivenum(log_reviews, na.rm=T)[4]+IQR(log_reviews)*1.5
revup_outliers <- which(log_reviews > revup_limit)
length(revup_outliers) # 2796 outliers
```

```
## [1] 2796
```

```
revlow_limit <- fivenum(log_reviews)[2]-IQR(log_reviews)*1.5
revlow_outliers <- which(log_reviews < revlow_limit)
length(revlow_outliers) # 0 outliers
```

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

```
g_apps6$Reviews <- log(g_apps6$Reviews) # Using the log transformation, because although
it is right skewed, it's more normally distruted
g_apps7 <- g_apps6[-revup_outliers,]
plot(density(g_apps7$Reviews), main = "Density Plot log Reviews", lwd = 3, xlab = "Log o
f Reviews", col="#00DCFF")
```



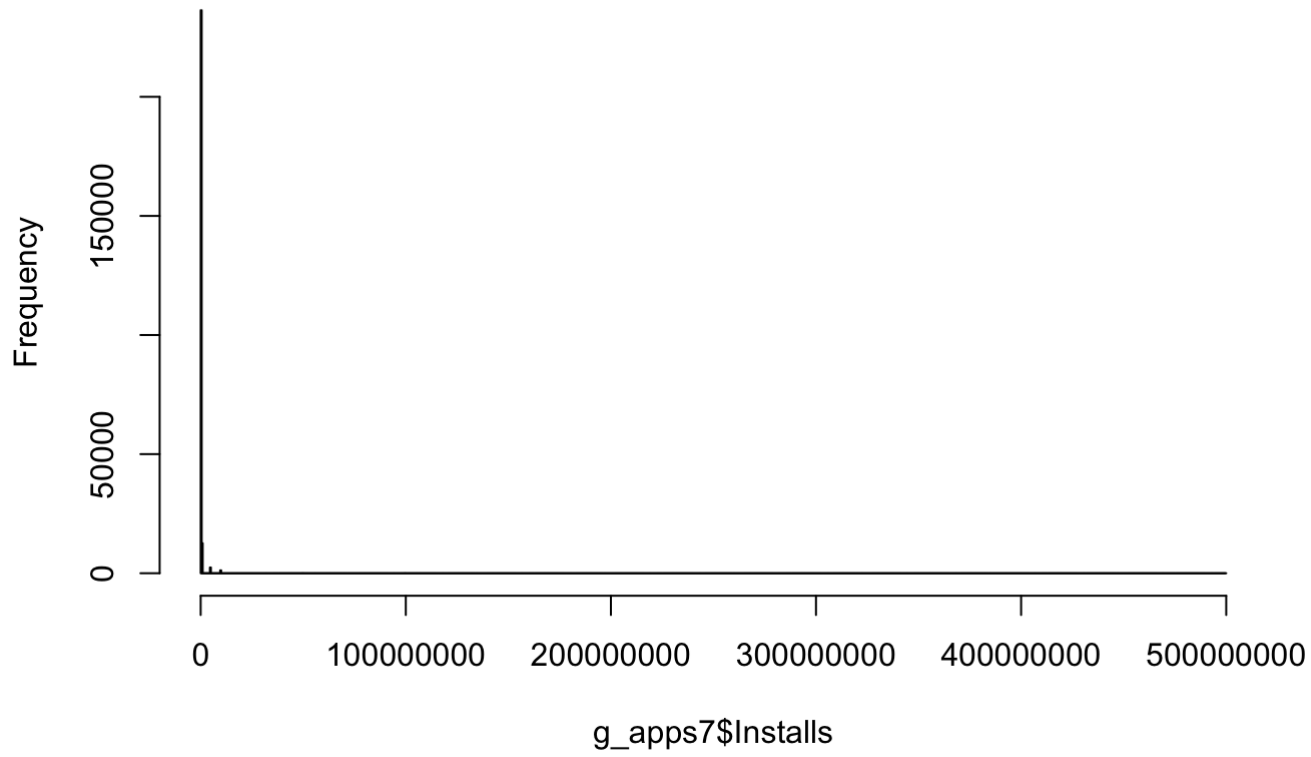

```
#-------------- Installs ---------------#
summary(g_apps7$Installs)
```

```
##          0          1         10        100       1000      10000
##         45        426       4069      16533      45961      58598
##     100000    1000000   10000000  100000000 1000000000          5
##      36736      12599       1221         22          0        640
##         50        500       5000      50000     500000    5000000
##       3486      12693      25268      22184       9685       2414
##   50000000  500000000 5000000000
##         26          5          0
```

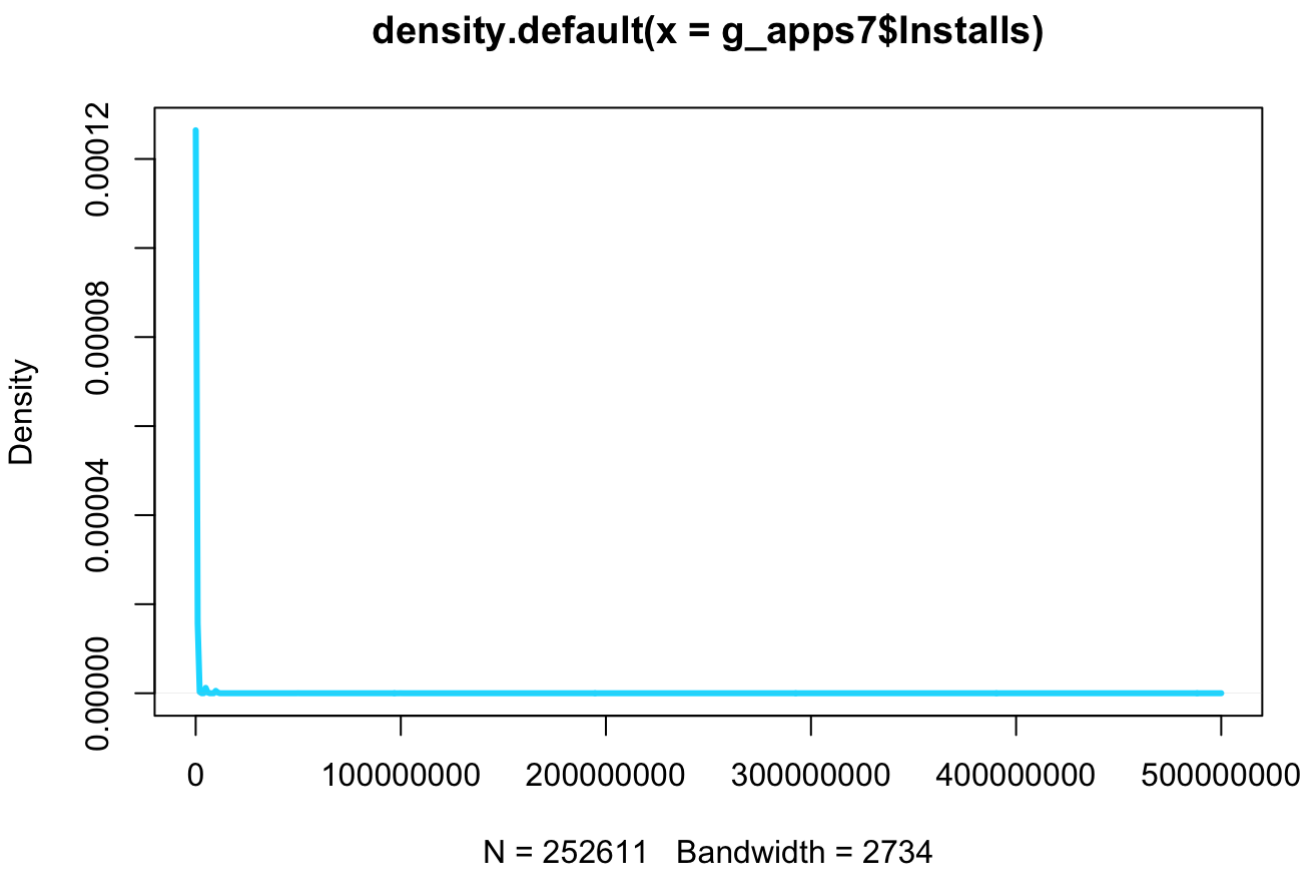
```
summary(as.numeric(as.character(g_apps7$Installs)))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##         0      1000     10000    210881     50000 500000000
```

```
g_apps7$Installs <- as.numeric(as.character(g_apps7$Installs))
hist(g_apps7$Installs, col = "#00DCFF", breaks = 1000)
```



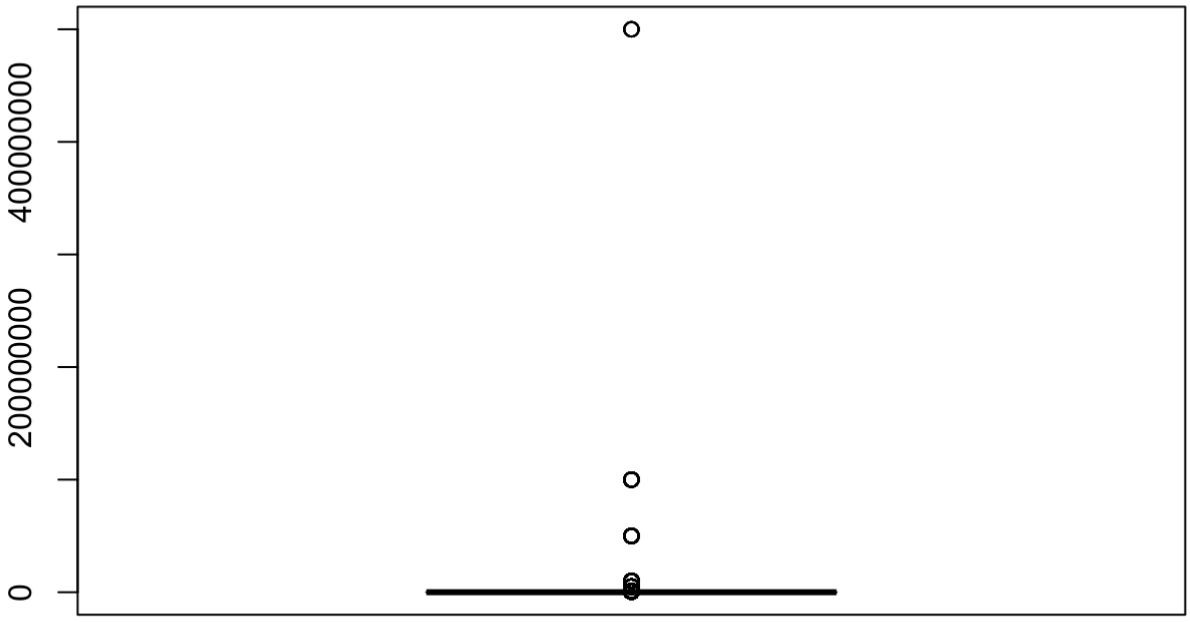

```
plot(density(g_apps7$Installs), col = "#00DCFF", lwd=3)
```



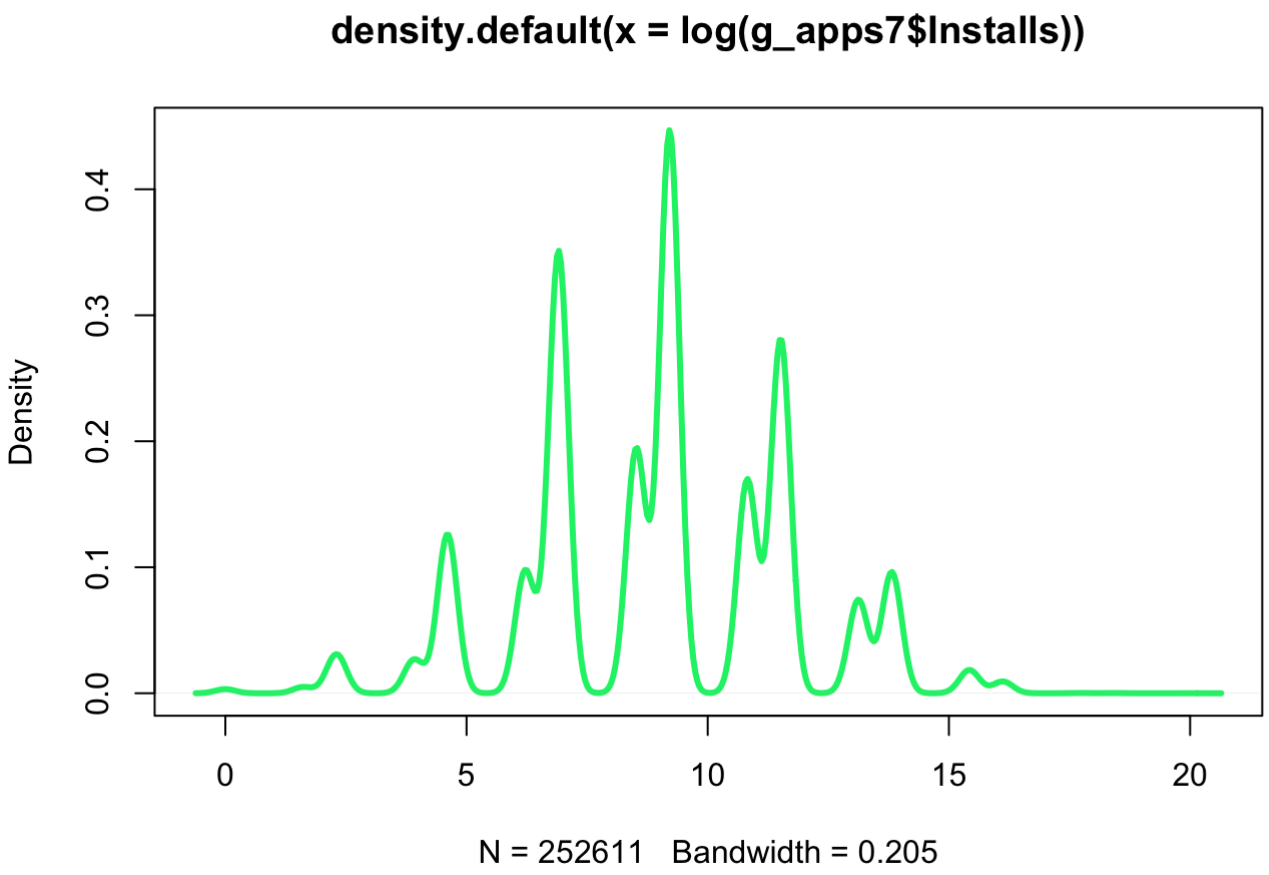

```
# Why is the data so skewed?
toohigh <- which(g_apps7$Installs %in% 500000000)
length(toohigh) # Only 5 apps have up to 5 million installs
```

```
## [1] 5
```

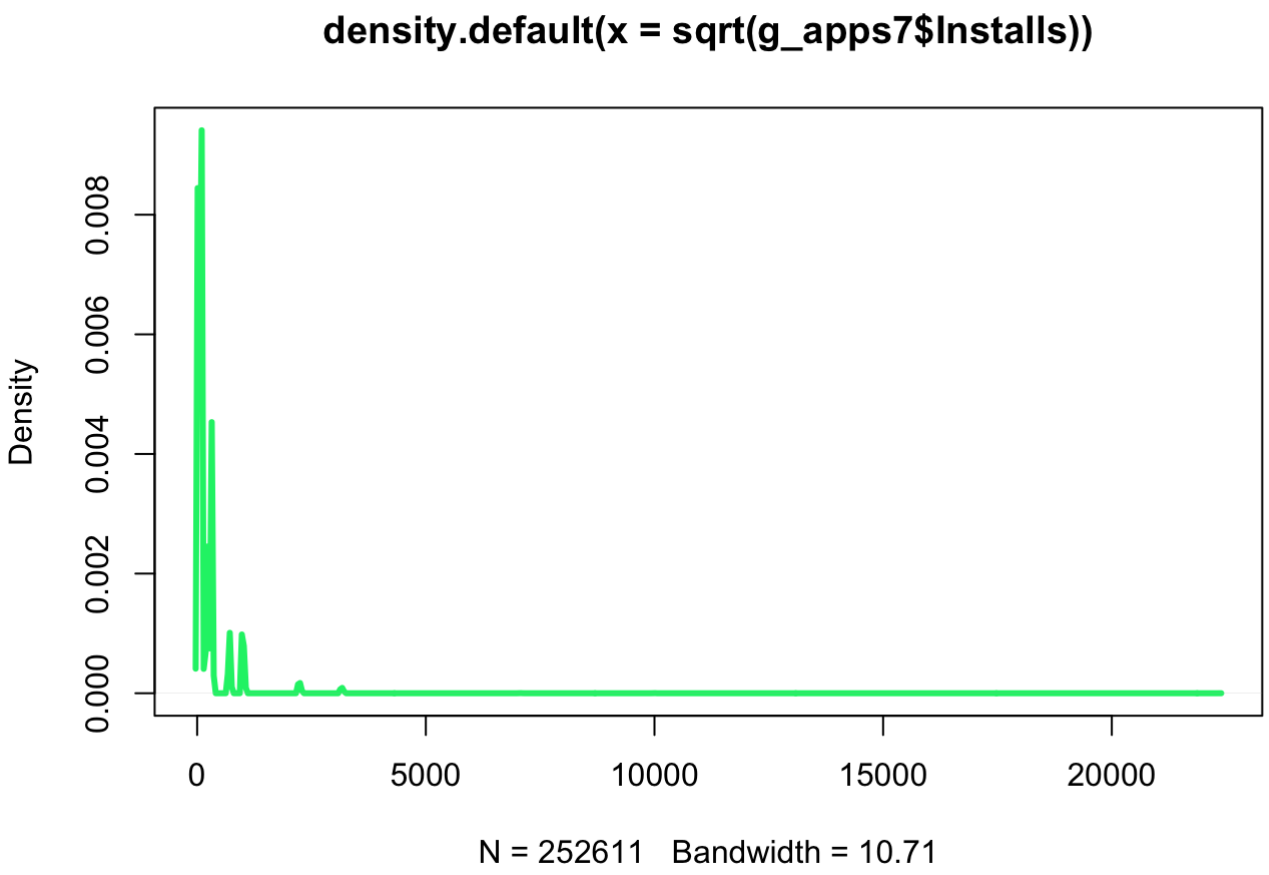
```
boxplot(g_apps7$Installs)
```



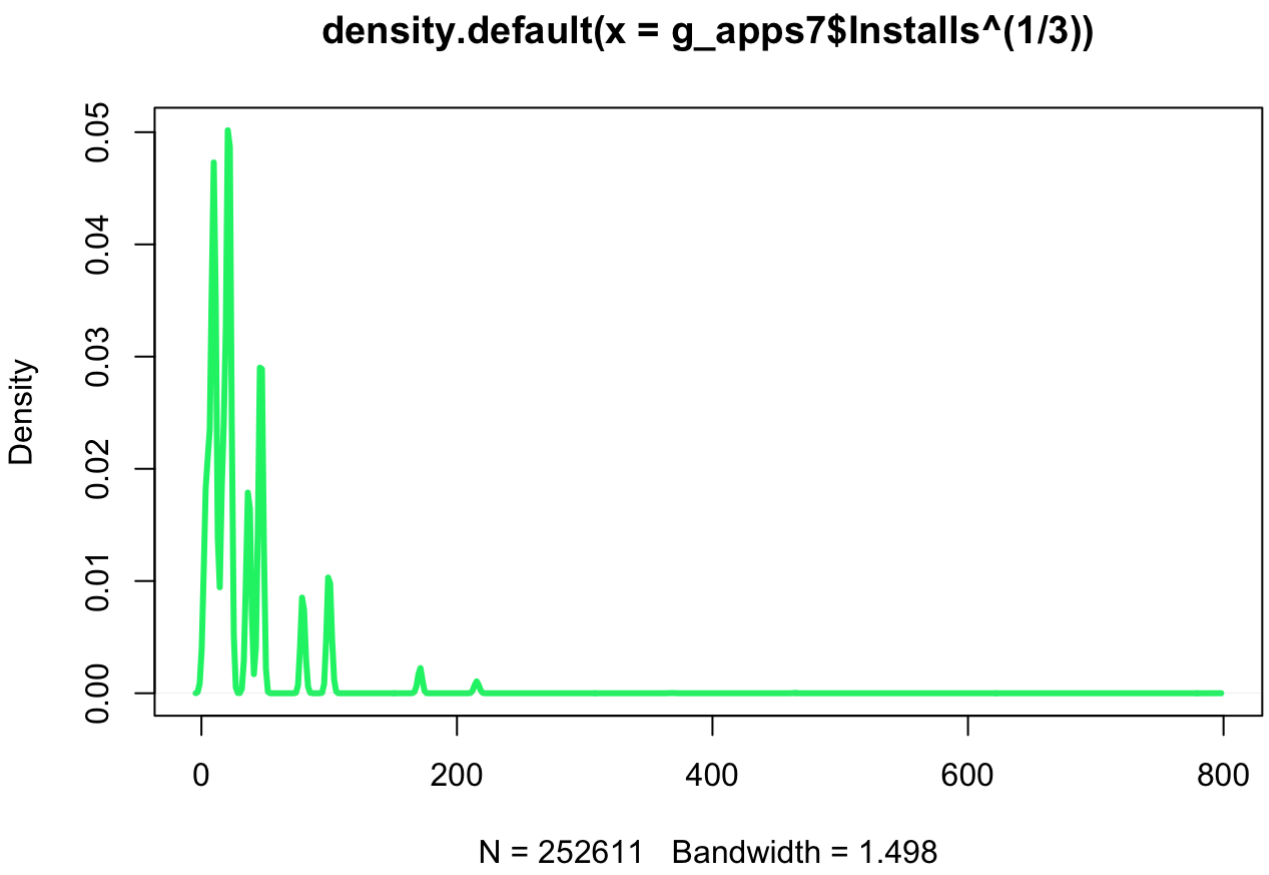

```
# Let me try data transformation to see if I can find a more normal distribution.
plot(density(log(g_apps7$Installs)), col = "#04F075", lwd = 3)
```




```
plot(density(sqrt(g_apps7$Installs)), col = "#04F075", lwd = 3)
```




```
plot(density(g_apps7$Installs^(1/3)), col = "#04F075", lwd = 3) # None of these data tra
nsformations achieve normal distribution.
```



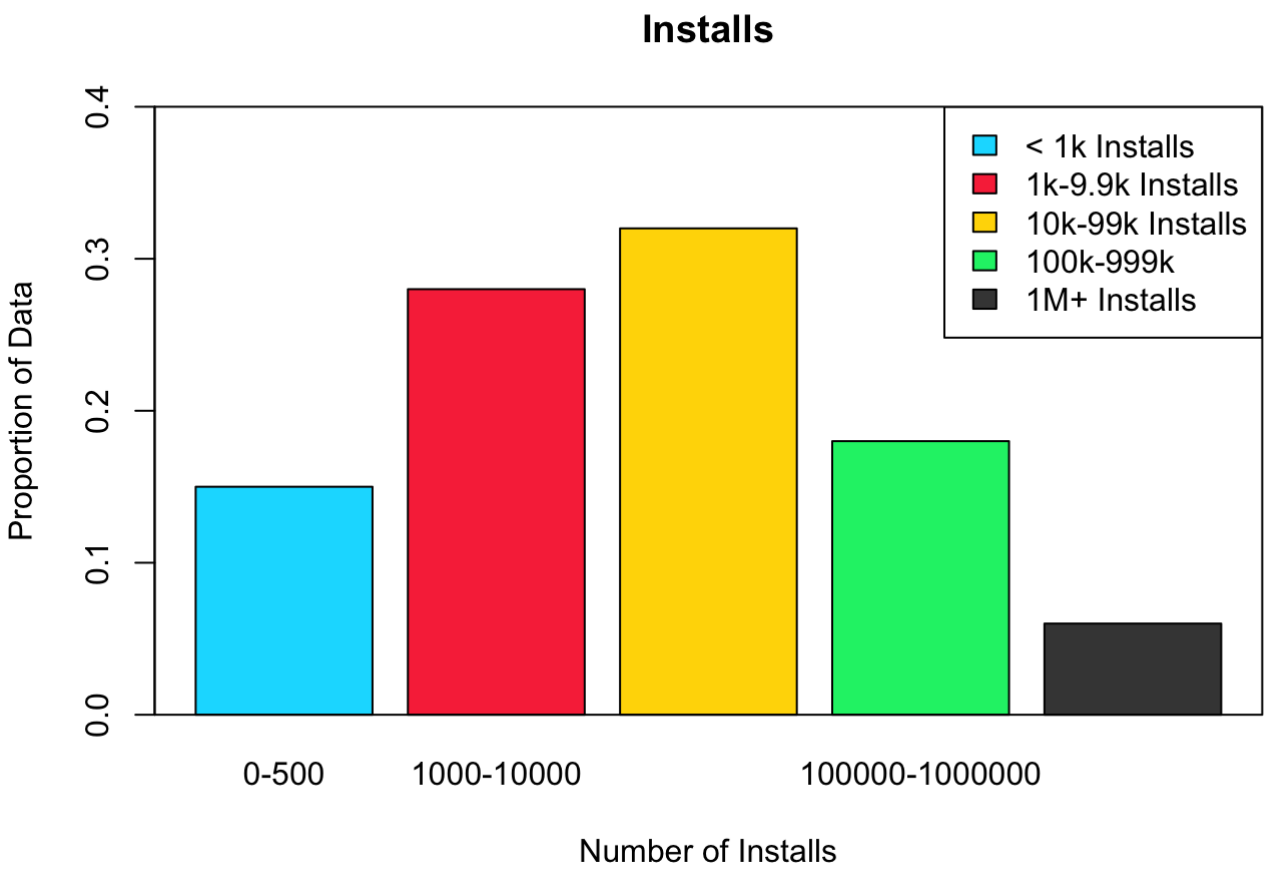

```
# As a numeric variable, Installs is extremely skewed so I will convert it to a factor w
ith 5 groups.
g_apps7$Installs <- as.character(g_apps7$Installs)
g_apps7$Installs[g_apps7$Installs %in% c(0,1,5,10,50,100,500)] <- "0-500"
g_apps7$Installs[g_apps7$Installs %in% c(1000,5000)] <- "1000-10000"
g_apps7$Installs[g_apps7$Installs %in% c(10000,50000)] <- "10000-100000"
g_apps7$Installs[g_apps7$Installs %in% c(100000,500000)] <- "100000-1000000"
g_apps7$Installs[g_apps7$Installs %in% c(1000000,5000000,10000000,50000000, 100000000,50
0000000)] <- "1000000+"

g_apps7$Installs <- as.factor(g_apps7$Installs)
summary(g_apps7$Installs)
```

```
##          0-500     1000-10000   10000-100000 100000-1000000       1000000+
##          37892          71229          80782          46421          16287
```

```
install_tab <- table(g_apps7$Installs)
install_ptab <- round(prop.table(install_tab), 2)
install_ptab
```

```
##
##          0-500     1000-10000   10000-100000 100000-1000000       1000000+
##           0.15           0.28           0.32           0.18           0.06
```

```
barplot(install_ptab, main = "Installs", xlab = "Number of Installs", ylab = "Proportion
of Data", ylim = c(0,0.4), col = c("#00DCFF", "#F83648", "#FFD800", "#04F075", "#444444"
))
legend("topright", fill = c("#00DCFF", "#F83648", "#FFD800", "#04F075", "#444444"), lege
nd = c("< 1k Installs", "1k-9.9k Installs", "10k-99k Installs", "100k-999k", "1M+ Instal
ls"))
box()
```




```
# -------- Size ----------#
g_apps8 <- g_apps7
summary(g_apps8$Size) # This variable cannot be numeric since "Varies with device" is a
 value
```

```
## Varies with device          11000000          12000000
##               10671              6927              6061
##            13000000          14000000          15000000
##                5307              4966              4908
##            16000000          10000000          17000000
##                4315              4236              3567
##            18000000          19000000          20000000
##                3444              3364              3074
##            21000000          23000000          22000000
##                2788              2606              2572
##            24000000           3800000           3400000
##                2556              2517              2501
##             3300000          25000000           3600000
##                2493              2492              2473
##             3200000           3700000           3900000
##                2456              2427              2368
##             3500000           2900000           4000000
##                2363              2305              2279
##             3100000           2800000          26000000
##                2274              2240              2229
##             3000000           4100000           2600000
##                2212              2107              2100
##             4500000           4300000           2700000
##                2077              2067              2014
##             4900000          27000000           4200000
##                2013              1968              1963
##             2300000           2500000           2400000
##                1925              1916              1912
##             4400000           4600000           4700000
##                1874              1801              1797
##             4800000           5000000           5100000
##                1794              1778              1778
##            28000000           2200000           5200000
##                1776              1726              1701
##            30000000           5800000          29000000
##                1686              1670              1612
##             5300000           2000000           6000000
##                1606              1605              1598
##            31000000           5500000           5600000
##                1594              1536              1519
##             5400000           5900000           5700000
##                1511              1490              1480
##             2100000          32000000           6100000
##                1464              1433              1324
##             6400000           6200000           6500000
##                1314              1306              1285
##             1900000           6300000          33000000
##                1284              1280              1277
##            36000000          34000000           1800000
##                1269              1256              1247
##             6600000          35000000           7400000
##                1214              1213              1202
##             1700000           6900000           7200000
```

```
##             1184             1168             1156
##         37000000          7000000          6700000
##             1141             1136             1124
##          7300000          7100000          6800000
##             1116             1111             1074
##          7500000          7600000          1500000
##             1070             1067             1052
##         38000000          7900000          7700000
##             1018             1015             1014
##          8300000          1400000          8200000
##             1011             1005             1003
##          1600000          7800000         39000000
##              997              974              964
##          (Other)
##            46858
```

```
# I'll add a new variable for size groupings
vwd <- which(g_apps8$Size == "Varies with device")
g_apps8.1 <- g_apps8[-vwd,]
summary(g_apps8.1$Size)
```

```
## 11000000 12000000 13000000 14000000 15000000 16000000 10000000 17000000
##     6927     6061     5307     4966     4908     4315     4236     3567
## 18000000 19000000 20000000 21000000 23000000 22000000 24000000  3800000
##     3444     3364     3074     2788     2606     2572     2556     2517
##  3400000  3300000 25000000  3600000  3200000  3700000  3900000  3500000
##     2501     2493     2492     2473     2456     2427     2368     2363
##  2900000  4000000  3100000  2800000 26000000  3000000  4100000  2600000
##     2305     2279     2274     2240     2229     2212     2107     2100
##  4500000  4300000  2700000  4900000 27000000  4200000  2300000  2500000
##     2077     2067     2014     2013     1968     1963     1925     1916
##  2400000  4400000  4600000  4700000  4800000  5000000  5100000 28000000
##     1912     1874     1801     1797     1794     1778     1778     1776
##  2200000  5200000 30000000  5800000 29000000  5300000  2000000  6000000
##     1726     1701     1686     1670     1612     1606     1605     1598
## 31000000  5500000  5600000  5400000  5900000  5700000  2100000 32000000
##     1594     1536     1519     1511     1490     1480     1464     1433
##  6100000  6400000  6200000  6500000  1900000  6300000 33000000 36000000
##     1324     1314     1306     1285     1284     1280     1277     1269
## 34000000  1800000  6600000 35000000  7400000  1700000  6900000  7200000
##     1256     1247     1214     1213     1202     1184     1168     1156
## 37000000  7000000  6700000  7300000  7100000  6800000  7500000  7600000
##     1141     1136     1124     1116     1111     1074     1070     1067
##  1500000 38000000  7900000  7700000  8300000  1400000  8200000  1600000
##     1052     1018     1015     1014     1011     1005     1003      997
##  7800000 39000000  8000000  (Other)
##      974      964      948    45910
```

```
g_apps8.1$Size <- as.numeric(as.character(g_apps8.1$Size))
str(g_apps8.1$Size)
```

```
##  num [1:241940] 1400000 23000000 4100000 39000000 8100000 19000000 30000000 11000000
11000000 23000000 ...
```

```
fivenum(g_apps8.1$Size) # minimum value is 3.1 so I'll set "Varies with device to 1
```

```
## [1]         3.1   4000000.0   8300000.0  19000000.0 334000000.0
```

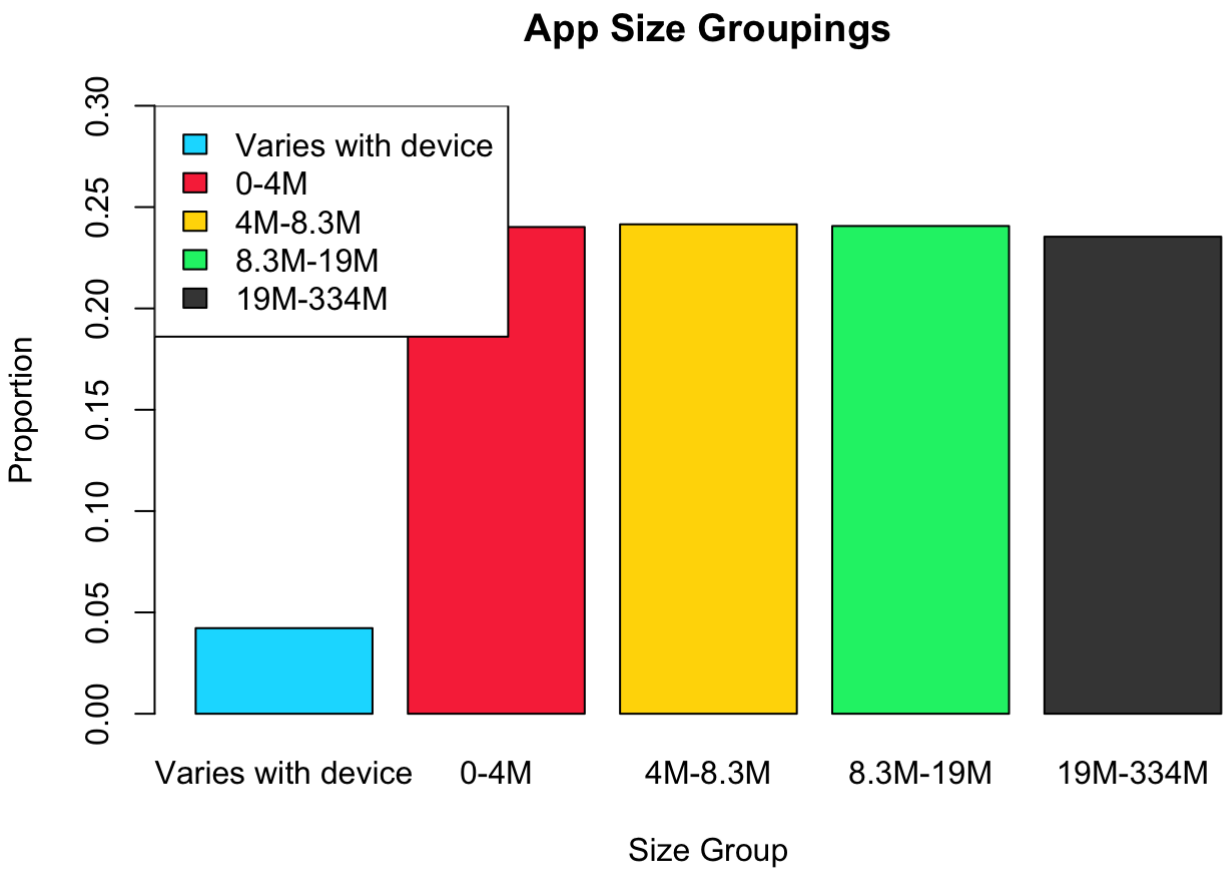
```
g_apps8$Size <- as.character(g_apps8$Size)
g_apps8$Size[g_apps8$Size == "Varies with device"] <- 1
g_apps8$Size <- as.numeric(as.character(g_apps8$Size))
g_apps8$Size.Groups <- cut(g_apps8$Size, breaks = c(0,3,4000000,8300000,19000000,3340000
00), labels = c("Varies with device", "0-4M",  "4M-8.3M", "8.3M-19M", "19M-334M"))
summary(g_apps8$Size.Groups)
```

```
## Varies with device               0-4M            4M-8.3M
##              10671              60671              61008
##           8.3M-19M           19M-334M
##              60798              59463
```

```
sgtab <- table(g_apps8$Size.Groups)
psgtab <- prop.table(sgtab)
psgtab # groupings look proportionate aside from group 5 which is "Varies by device"
```

```
##
## Varies with device               0-4M            4M-8.3M
##         0.04224282         0.24017561         0.24150967
##           8.3M-19M           19M-334M
##         0.24067836         0.23539355
```

```
barplot(psgtab, main = "App Size Groupings", col = c("#00DCFF", "#F83648", "#FFD800", "#
04F075", "#444444"), xlab = "Size Group", ylab = "Proportion", ylim = c(0,0.3))
legend("topleft", fill = c("#00DCFF", "#F83648", "#FFD800", "#04F075", "#444444"), legen
d = levels(g_apps8$Size.Groups))
```




```
g_apps9 <- g_apps8
g_apps9$Price <- as.numeric(as.character(g_apps9$Price))
summary(g_apps9$Price)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##   0.0000   0.0000   0.0000   0.2309   0.0000 399.9900
```

```
pt <- table(as.factor(g_apps9$Price))
ppt <- prop.table(pt)
ppt # over 95% of apps in this dataset cost $0 to download (Free), I'll convert to a fac
tor
```

```
##
##               0            0.67            0.99               1            1.01
## 0.955876822466 0.000003958656 0.008685290823 0.000178139511 0.000011875967
##            1.02            1.03            1.04            1.05            1.06
## 0.000003958656 0.000007917312 0.000003958656 0.000011875967 0.000003958656
##            1.07            1.08            1.09             1.1            1.11
## 0.000003958656 0.000011875967 0.000019793279 0.000015834623 0.000015834623
##            1.12            1.13            1.17            1.18            1.19
## 0.000003958656 0.000003958656 0.000007917312 0.000003958656 0.000059379837
##             1.2            1.21            1.22            1.23            1.25
## 0.000035627902 0.000003958656 0.000003958656 0.000003958656 0.000019793279
##            1.26            1.27            1.28            1.29             1.3
## 0.000007917312 0.000003958656 0.000003958656 0.000063338493 0.000015834623
##            1.32            1.33            1.34            1.35            1.36
## 0.000011875967 0.000015834623 0.000015834623 0.000007917312 0.000011875967
##            1.37            1.38            1.39             1.4            1.42
## 0.000003958656 0.000007917312 0.000015834623 0.000007917312 0.000007917312
##            1.43            1.44            1.45            1.48            1.49
## 0.000003958656 0.000003958656 0.000019793279 0.000011875967 0.002858149487
##             1.5            1.51            1.52            1.53            1.55
## 0.000051462525 0.000007917312 0.000003958656 0.000003958656 0.000003958656
##            1.56            1.58            1.59            1.61            1.62
## 0.000003958656 0.000003958656 0.000039586558 0.000007917312 0.000007917312
##            1.66            1.67            1.68            1.69             1.7
## 0.000003958656 0.000003958656 0.000011875967 0.000011875967 0.000023751935
##            1.72            1.74            1.75            1.77            1.78
## 0.000003958656 0.000007917312 0.000031669246 0.000011875967 0.000007917312
##            1.79             1.8            1.81            1.82            1.84
## 0.000019793279 0.000023751935 0.000011875967 0.000007917312 0.000003958656
##            1.85            1.86            1.88            1.89             1.9
## 0.000007917312 0.000007917312 0.000003958656 0.000007917312 0.000011875967
##            1.92            1.93            1.94            1.95            1.96
## 0.000007917312 0.000003958656 0.000003958656 0.000035627902 0.000015834623
##            1.97            1.98            1.99               2            2.02
## 0.000019793279 0.000003958656 0.005937983698 0.000162304888 0.000003958656
##            2.03            2.04            2.09             2.1            2.13
## 0.000011875967 0.000007917312 0.000015834623 0.000007917312 0.000003958656
##            2.15            2.19             2.2            2.25            2.27
## 0.000003958656 0.000003958656 0.000007917312 0.000007917312 0.000003958656
##            2.28            2.29             2.3            2.31            2.32
## 0.000007917312 0.000015834623 0.000003958656 0.000007917312 0.000007917312
##            2.33            2.35            2.36            2.37            2.39
## 0.000003958656 0.000007917312 0.000003958656 0.000003958656 0.000011875967
##             2.4            2.41            2.42            2.43            2.45
## 0.000007917312 0.000003958656 0.000003958656 0.000003958656 0.000003958656
##            2.46            2.48            2.49             2.5            2.51
## 0.000015834623 0.000011875967 0.001983286555 0.000039586558 0.000003958656
##            2.52            2.54            2.55            2.57            2.59
## 0.000003958656 0.000003958656 0.000003958656 0.000003958656 0.000027710591
##             2.6            2.64            2.65            2.66            2.67
## 0.000003958656 0.000003958656 0.000007917312 0.000003958656 0.000015834623
##            2.69            2.71            2.73            2.74            2.77
## 0.000007917312 0.000007917312 0.000003958656 0.000003958656 0.000011875967
```

```
##           2.79            2.8           2.82           2.84           2.85
## 0.000015834623 0.000003958656 0.000003958656 0.000007917312 0.000011875967
##           2.86           2.89            2.9           2.93           2.94
## 0.000003958656 0.000015834623 0.000031669246 0.000011875967 0.000003958656
##           2.95           2.98           2.99              3           3.01
## 0.000003958656 0.000003958656 0.005205632375 0.000083131772 0.000003958656
##           3.03           3.06           3.08           3.09            3.1
## 0.000003958656 0.000003958656 0.000023751935 0.000003958656 0.000003958656
##           3.11           3.14           3.16           3.22           3.24
## 0.000007917312 0.000003958656 0.000003958656 0.000007917312 0.000003958656
##           3.25           3.26           3.27           3.28           3.29
## 0.000003958656 0.000003958656 0.000003958656 0.000003958656 0.000007917312
##           3.32           3.33           3.36           3.38           3.39
## 0.000007917312 0.000003958656 0.000003958656 0.000007917312 0.000007917312
##            3.4           3.41           3.42           3.43           3.49
## 0.000003958656 0.000003958656 0.000007917312 0.000003958656 0.001302397758
##            3.5           3.55           3.58           3.59           3.63
## 0.000023751935 0.000011875967 0.000003958656 0.000007917312 0.000003958656
##           3.65           3.72           3.75           3.77           3.78
## 0.000003958656 0.000003958656 0.000015834623 0.000003958656 0.000003958656
##            3.8           3.81           3.82           3.83           3.84
## 0.000003958656 0.000007917312 0.000007917312 0.000007917312 0.000003958656
##           3.85           3.89            3.9           3.91           3.93
## 0.000003958656 0.000003958656 0.000019793279 0.000003958656 0.000003958656
##           3.95           3.97           3.98           3.99              4
## 0.000027710591 0.000003958656 0.000007917312 0.002945239914 0.000031669246
##           4.03           4.04           4.06           4.14           4.17
## 0.000003958656 0.000011875967 0.000003958656 0.000003958656 0.000003958656
##           4.19            4.2           4.25           4.26           4.29
## 0.000003958656 0.000007917312 0.000011875967 0.000003958656 0.000007917312
##            4.3           4.31           4.34           4.35           4.38
## 0.000003958656 0.000003958656 0.000003958656 0.000003958656 0.000003958656
##           4.39            4.4           4.41           4.42           4.44
## 0.000003958656 0.000003958656 0.000003958656 0.000003958656 0.000003958656
##           4.48           4.49            4.5           4.53           4.54
## 0.000003958656 0.000977787982 0.000007917312 0.000003958656 0.000003958656
##           4.56           4.57           4.62           4.63           4.64
## 0.000003958656 0.000003958656 0.000003958656 0.000007917312 0.000003958656
##           4.69           4.71           4.72           4.73           4.74
## 0.000011875967 0.000003958656 0.000007917312 0.000003958656 0.000003958656
##           4.77            4.8           4.82           4.85           4.88
## 0.000003958656 0.000003958656 0.000003958656 0.000003958656 0.000003958656
##           4.89            4.9           4.95           4.97           4.98
## 0.000011875967 0.000015834623 0.000019793279 0.000003958656 0.000003958656
##           4.99               5           5.01           5.29            5.3
## 0.003388609364 0.000047503870 0.000003958656 0.000007917312 0.000003958656
##           5.33           5.36           5.38            5.4           5.48
## 0.000007917312 0.000003958656 0.000003958656 0.000003958656 0.000015834623
##           5.49            5.5           5.55           5.57           5.69
## 0.000577963747 0.000003958656 0.000007917312 0.000003958656 0.000003958656
##           5.72           5.74           5.76           5.78           5.79
## 0.000003958656 0.000007917312 0.000003958656 0.000003958656 0.000003958656
##            5.9           5.95           5.96           5.98           5.99
## 0.000003958656 0.000003958656 0.000003958656 0.000003958656 0.000866945620
```

```
##               6            6.04            6.14            6.15            6.16
## 0.000023751935 0.000003958656 0.000007917312 0.000003958656 0.000003958656
##            6.21            6.27            6.28            6.29             6.3
## 0.000003958656 0.000003958656 0.000003958656 0.000003958656 0.000003958656
##            6.49            6.52            6.54            6.57            6.58
## 0.000249395315 0.000003958656 0.000015834623 0.000011875967 0.000003958656
##            6.59            6.64            6.69             6.7            6.71
## 0.000003958656 0.000003958656 0.000003958656 0.000003958656 0.000003958656
##            6.73            6.84            6.85            6.87             6.9
## 0.000003958656 0.000003958656 0.000007917312 0.000003958656 0.000003958656
##            6.98            6.99               7            7.03            7.12
## 0.000007917312 0.000653178207 0.000007917312 0.000003958656 0.000003958656
##            7.14            7.23            7.49            7.53            7.55
## 0.000003958656 0.000003958656 0.000253353971 0.000003958656 0.000003958656
##            7.56            7.64            7.68            7.73            7.74
## 0.000003958656 0.000003958656 0.000003958656 0.000003958656 0.000007917312
##            7.85            7.89            7.99               8            8.34
## 0.000003958656 0.000011875967 0.000676930142 0.000007917312 0.000003958656
##            8.35            8.43            8.47            8.49            8.66
## 0.000003958656 0.000003958656 0.000003958656 0.000134594297 0.000003958656
##            8.69            8.75             8.8             8.9            8.92
## 0.000003958656 0.000003958656 0.000007917312 0.000011875967 0.000003958656
##            8.99               9            9.13            9.17            9.49
## 0.000312733808 0.000003958656 0.000003958656 0.000003958656 0.000186056823
##            9.71            9.76            9.79             9.8             9.9
## 0.000003958656 0.000003958656 0.000003958656 0.000007917312 0.000011875967
##            9.95            9.99              10           10.14           10.15
## 0.000015834623 0.001053002442 0.000015834623 0.000003958656 0.000003958656
##           10.25           10.49            10.7           10.75           10.99
## 0.000003958656 0.000003958656 0.000003958656 0.000003958656 0.000178139511
##              11           11.07            11.1           11.41           11.99
## 0.000003958656 0.000003958656 0.000003958656 0.000003958656 0.000174180855
##              12           12.49           12.62           12.93           12.99
## 0.000011875967 0.000011875967 0.000003958656 0.000003958656 0.000193974134
##           13.37           13.46           13.48           13.52           13.61
## 0.000003958656 0.000003958656 0.000003958656 0.000003958656 0.000003958656
##           13.99           14.01            14.5           14.73           14.93
## 0.000106883707 0.000003958656 0.000003958656 0.000003958656 0.000003958656
##           14.99              15           15.01           15.68           15.99
## 0.000261271283 0.000007917312 0.000003958656 0.000003958656 0.000114801018
##           16.17           16.99           17.92           17.98           17.99
## 0.000003958656 0.000067297149 0.000003958656 0.000003958656 0.000071255804
##              18            18.6           18.99           19.01           19.49
## 0.000003958656 0.000003958656 0.000059379837 0.000003958656 0.000003958656
##             19.5            19.7            19.9           19.95           19.98
## 0.000003958656 0.000003958656 0.000007917312 0.000003958656 0.000003958656
##           19.99           20.99              21           21.99           22.22
## 0.000197932790 0.000039586558 0.000003958656 0.000055421181 0.000003958656
##           22.99           23.32           23.45           23.92           23.99
## 0.000055421181 0.000003958656 0.000003958656 0.000003958656 0.000023751935
##           24.46           24.64           24.95           24.99              25
## 0.000003958656 0.000003958656 0.000003958656 0.000174180855 0.000003958656
##           25.99            26.9           26.99           27.01           27.47
## 0.000003958656 0.000003958656 0.000031669246 0.000003958656 0.000003958656
```

```
##           27.5          27.99          28.99          29.95          29.99
## 0.000007917312 0.000039586558 0.000027710591 0.000007917312 0.000150428920
##          30.99          31.99          32.99          33.99             34
## 0.000003958656 0.000007917312 0.000039586558 0.000011875967 0.000003958656
##          34.99             35          35.99          37.84          37.99
## 0.000023751935 0.000003958656 0.000023751935 0.000003958656 0.000015834623
##          38.99           39.8          39.99          40.99          42.99
## 0.000031669246 0.000003958656 0.000023751935 0.000007917312 0.000003958656
##          43.99          44.99          45.99          46.99             49
## 0.000003958656 0.000003958656 0.000003958656 0.000007917312 0.000003958656
##          49.95          49.99             50          52.25          53.47
## 0.000003958656 0.000031669246 0.000003958656 0.000003958656 0.000003958656
##          54.99          59.99          64.99             69          69.99
## 0.000047503870 0.000007917312 0.000015834623 0.000003958656 0.000027710591
##          74.99          79.99          81.18          84.99          89.99
## 0.000011875967 0.000039586558 0.000003958656 0.000007917312 0.000003958656
##          94.99             99           99.9          99.95          99.99
## 0.000007917312 0.000003958656 0.000007917312 0.000003958656 0.000003958656
##           100         104.99         109.99         114.99         119.99
## 0.000003958656 0.000003958656 0.000003958656 0.000003958656 0.000003958656
##         124.99         129.99         134.99            140         184.99
## 0.000007917312 0.000007917312 0.000003958656 0.000003958656 0.000007917312
##         199.99         234.99         244.99         289.99          299.9
## 0.000011875967 0.000003958656 0.000003958656 0.000003958656 0.000003958656
##         299.99         309.99         369.99         374.99         379.99
## 0.000007917312 0.000003958656 0.000007917312 0.000003958656 0.000007917312
##         389.99         399.99
## 0.000003958656 0.000019793279
```

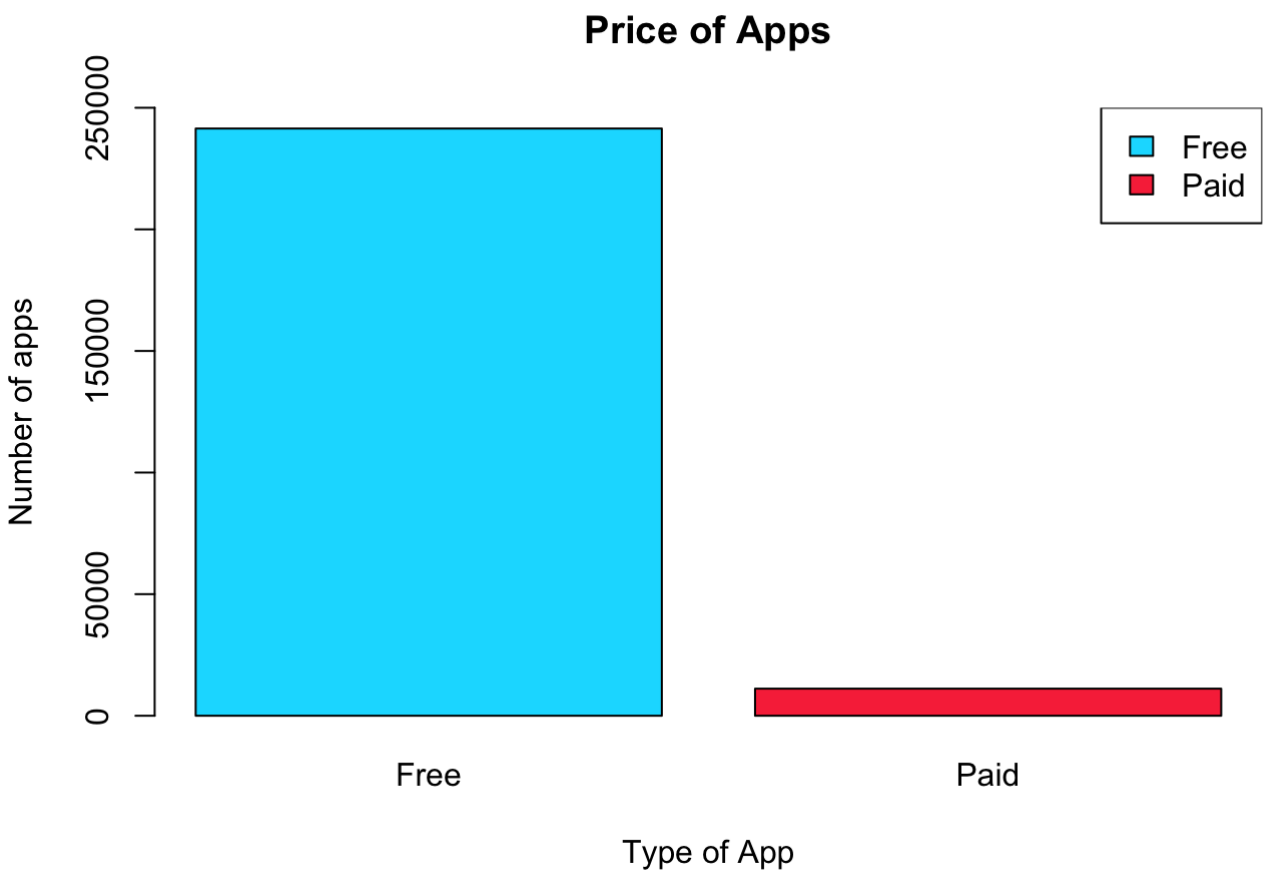
```
g_apps9$Price <- as.character(g_apps9$Price)
g_apps9$Price[g_apps9$Price %in% "0"] <- "Free"
g_apps9$Price[g_apps9$Price != "Free"] <- "Paid"
ptab <- table(g_apps9$Price)
pptab <- prop.table(ptab)
pptab
```

```
##
##       Free       Paid
## 0.95587682 0.04412318
```

```
g_apps9$Price <- as.factor(g_apps9$Price)
summary(g_apps9$Price) # 4.4% of apps cost money
```

```
##   Free   Paid
## 241465  11146
```

```
barplot(ptab, main = "Price of Apps", col = c("#00DCFF", "#F83648"), ylab = "Number of a
pps", xlab = "Type of App", ylim = c(0,250000))
legend("topright", fill = c("#00DCFF", "#F83648"), legend = levels(g_apps9$Price))
```



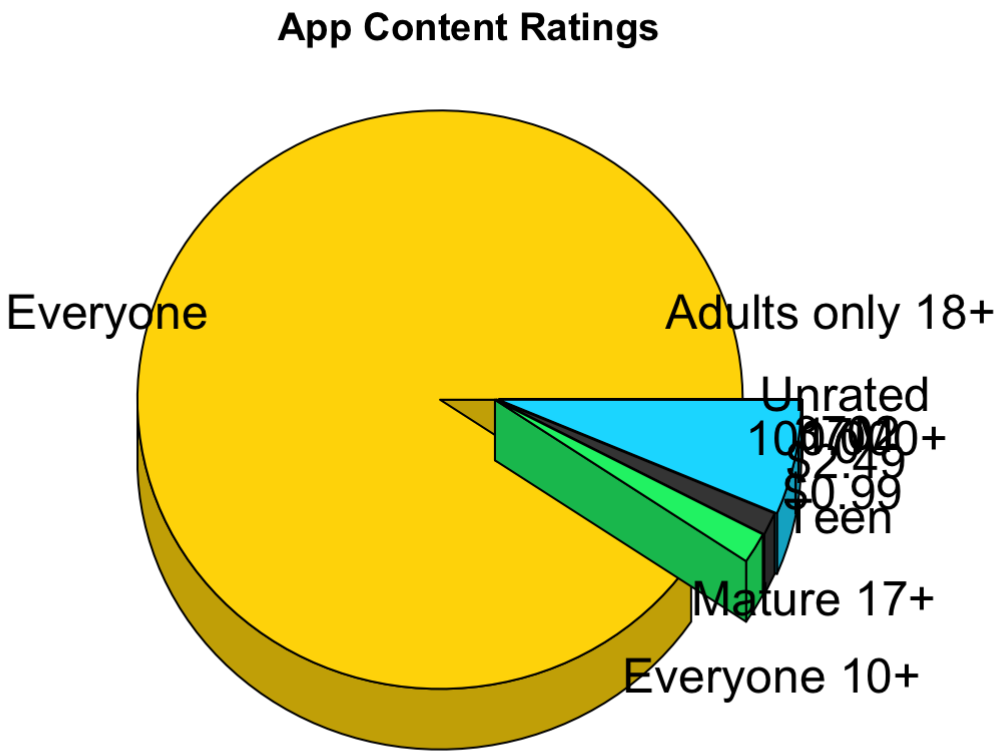

```
g_apps10 <- g_apps9
summary(g_apps10$Content.Rating)
```

```
##           $0.99           $2.49               0         100,000+
##               0               0               0               0
##             17M            3702 Adults only 18+        Everyone
##               0               0              11          228790
##    Everyone 10+      Mature 17+            Teen         Unrated
##            4337            3234           16206              33
```

```
g_apps10$Content.Rating <- as.factor(g_apps10$Content.Rating)
crtab <- table(g_apps10$Content.Rating)
pcrtab <- prop.table(crtab)
pcrtab
```

```
##
##           $0.99           $2.49               0         100,000+
##   0.00000000000   0.00000000000   0.00000000000   0.00000000000
##             17M            3702 Adults only 18+        Everyone
##   0.00000000000   0.00000000000   0.00004354521   0.90570086022
##    Everyone 10+      Mature 17+            Teen         Unrated
##   0.01716869020   0.01280229285   0.06415397588   0.00013063564
```

```
pie3D(crtab, explode = 0.1, main = "App Content Ratings", theta = 1.5,  labels = levels
(g_apps10$Content.Rating), col = c("#00DCFF", "#F83648", "#FFD800", "#04F075", "#444444"
))
```



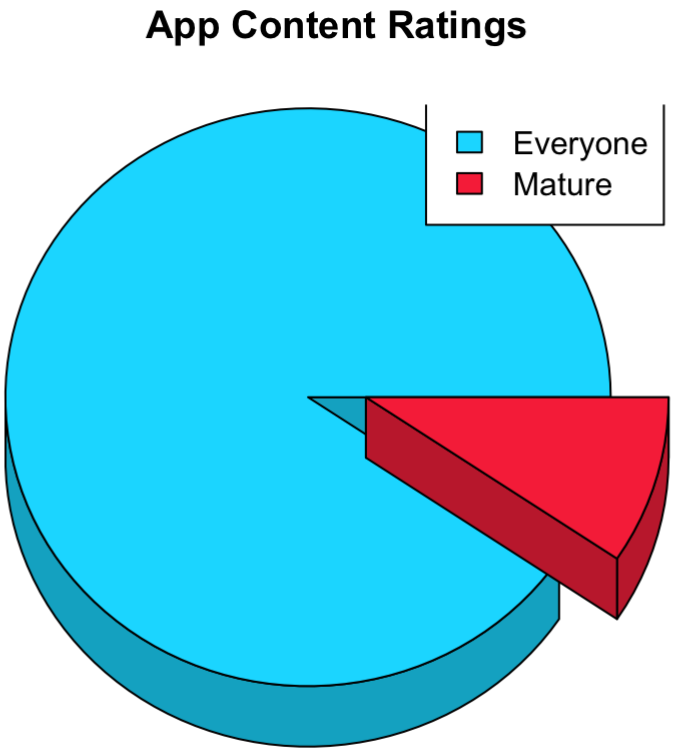

```
# The groups are very disproportionate, lets group some together
g_apps10$Content.Rating <- as.character(g_apps10$Content.Rating)
g_apps10$Content.Rating[g_apps10$Content.Rating == "Everyone 10+"] <- "Mature"
g_apps10$Content.Rating[g_apps10$Content.Rating == "Adults only 18+"] <- "Mature"
g_apps10$Content.Rating[g_apps10$Content.Rating == "Teen"] <- "Mature"
g_apps10$Content.Rating[g_apps10$Content.Rating == "Unrated"] <- "Mature"
g_apps10$Content.Rating[g_apps10$Content.Rating == "Mature 17+"] <- "Mature"
g_apps10$Content.Rating <- as.factor(g_apps10$Content.Rating)
summary(g_apps10$Content.Rating)
```

```
## Everyone   Mature
##   228790    23821
```

```
new_crtab <- table(g_apps10$Content.Rating)
new_pcrtab <- prop.table(new_crtab)
new_pcrtab
```

```
##
##   Everyone     Mature
## 0.90570086 0.09429914
```

```
pie3D(new_pcrtab, explode = 0.1, main = "App Content Ratings", theta = 1.5, col = c("#00
DCFF", "#F83648"))
legend("topright", legend = levels(g_apps10$Content.Rating), fill = c("#00DCFF", "#F8364
8"))
```



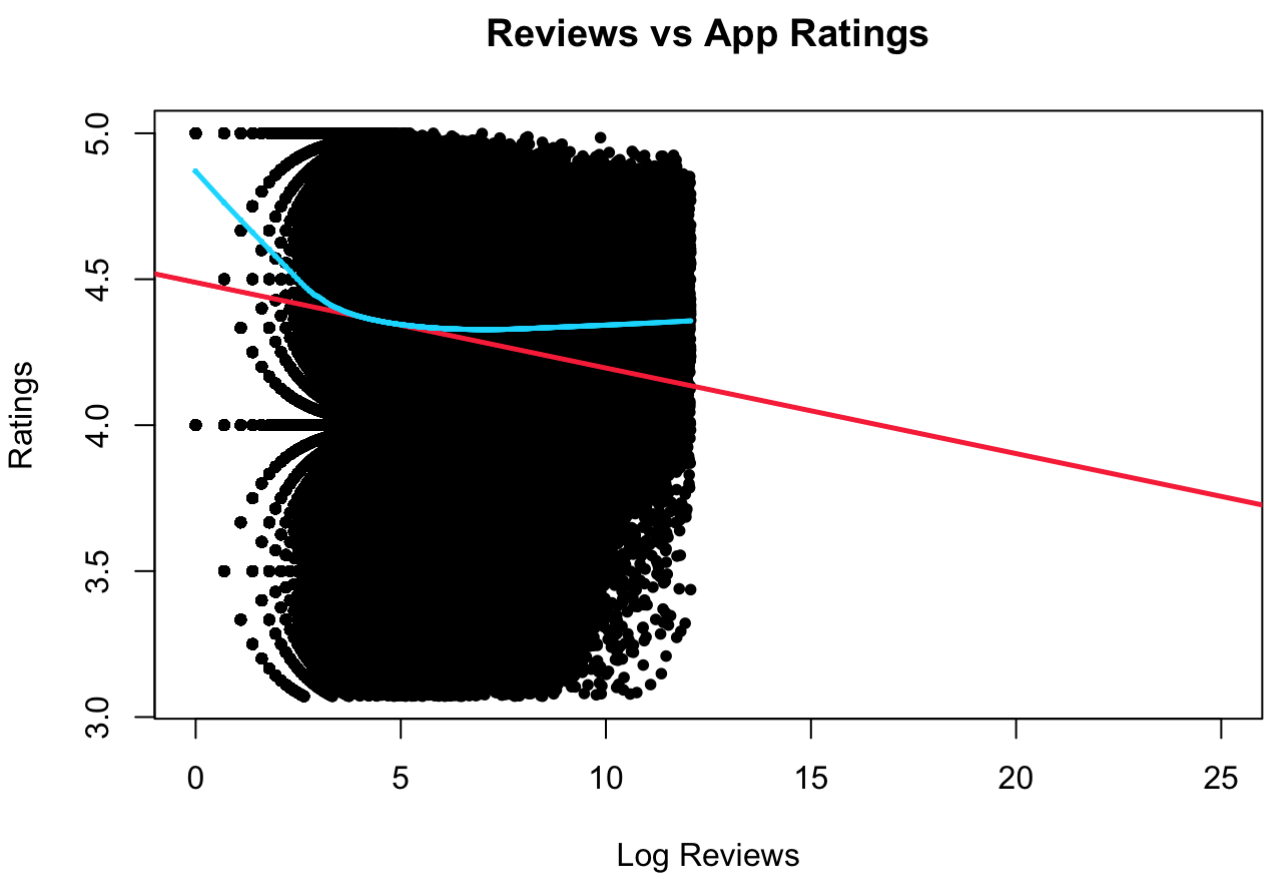

```
g_apps11 <- g_apps10[,-c(10,11)] # Removed last and minimum versions because I won't be
 using those variables
```

Bivariate Analysis

```
table(is.na(g_apps11$Rating)) # No missing values
```

```
##
##  FALSE
## 252611
```

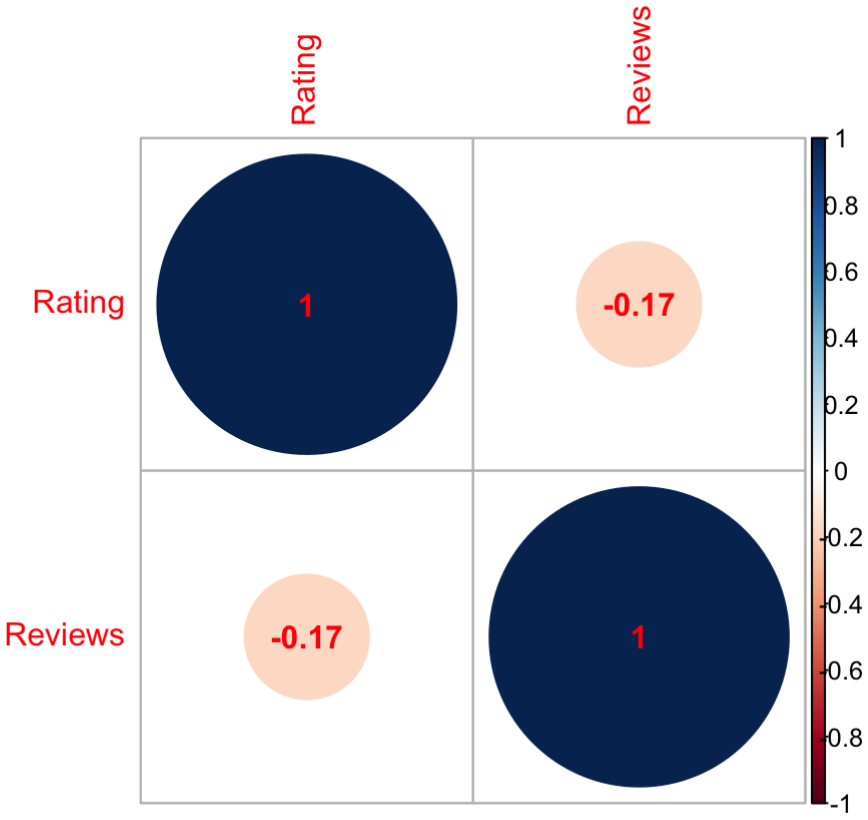
```
plot(g_apps11$Rating~g_apps11$Reviews, col = "black", main="Reviews vs App Ratings",
     xlab = "Log Reviews",
     ylab = "Ratings", pch=20,
     xlim = c(0, 25))
abline(lm(g_apps11$Rating~g_apps11$Reviews), col="#F83648", lwd=2.5)
lines(lowess(g_apps11$Rating~g_apps11$Reviews), col="#00DCFF", lwd=2.5)
```




```
library(corrplot)
cor(g_apps11$Rating, g_apps11$Reviews) # Very weak negative coorelation
```

```
## [1] -0.1726915
```

```
cormat <- cor(g_apps11[,c(3,4)])
corrplot(cormat, method = "circle", addCoef.col = "red")
```




```
cor.test(g_apps11$Rating, g_apps11$Reviews)
```

```
##
##  Pearson's product-moment correlation
##
## data:  g_apps11$Rating and g_apps11$Reviews
## t = -88.119, df = 252609, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1764723 -0.1689057
## sample estimates:
##       cor
## -0.1726915
```

The results from the correlation test show that we can reject the Null hypothesis. There is a correlation between Ratings and Reviews, however the correlation is a very weak, negative one at -0.17.

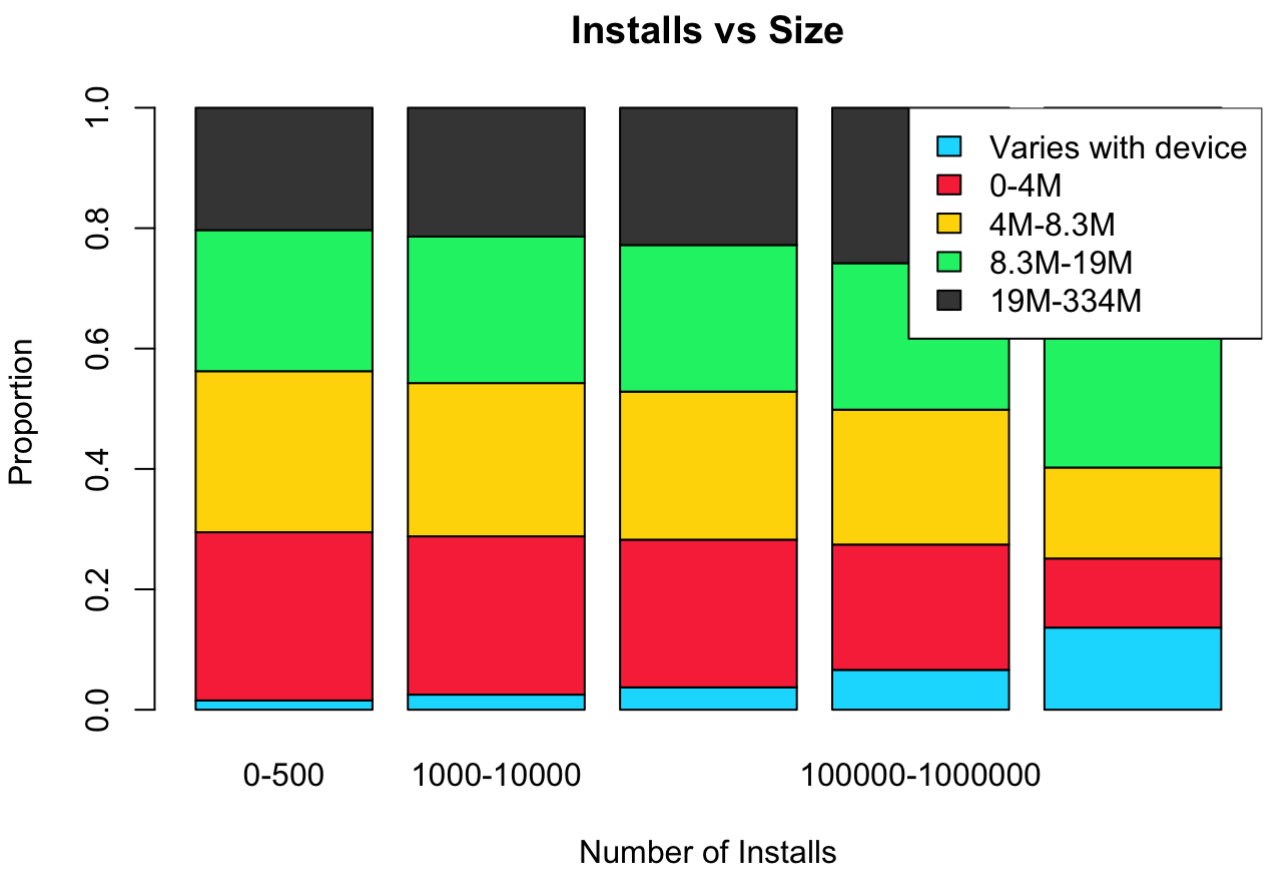
This also tells me that I can include both ratings and reviews in my regression model since multicollinearity is not an issue here.

Now I'll look at how each variable affects the target variable, Installs.

```
install_size <- table(g_apps11$Size.Groups, g_apps11$Installs)
pinstall_size <- prop.table(install_size)
pinstall_size <- round(pinstall_size,2)
addmargins(pinstall_size,c(1,2))
```

```
##
##                      0-500 1000-10000 10000-100000 100000-1000000 1000000+
##   Varies with device  0.00       0.01         0.01            0.01     0.01
##   0-4M                0.04       0.07         0.08            0.04     0.01
##   4M-8.3M             0.04       0.07         0.08            0.04     0.01
##   8.3M-19M            0.04       0.07         0.08            0.04     0.01
##   19M-334M            0.03       0.06         0.07            0.05     0.02
##   Sum                 0.15       0.28         0.32            0.18     0.06
##
##                       Sum
##   Varies with device 0.04
##   0-4M               0.24
##   4M-8.3M            0.24
##   8.3M-19M           0.24
##   19M-334M           0.23
##   Sum                0.99
```

```
ptab_size <- prop.table(install_size, margin = 2)
barplot(ptab_size, col = c("#00DCFF", "#F83648", "#FFD800", "#04F075", "#444444"), main
 = "Installs vs Size", xlab = "Number of Installs", ylab = "Proportion")
legend("topright", fill = c("#00DCFF", "#F83648", "#FFD800", "#04F075", "#444444"), lege
nd = levels(g_apps11$Size.Groups))
```




```
chisq.test(pinstall_size)
```

```
## Warning in chisq.test(pinstall_size): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  pinstall_size
## X-squared = 0.043817, df = 16, p-value = 1
```

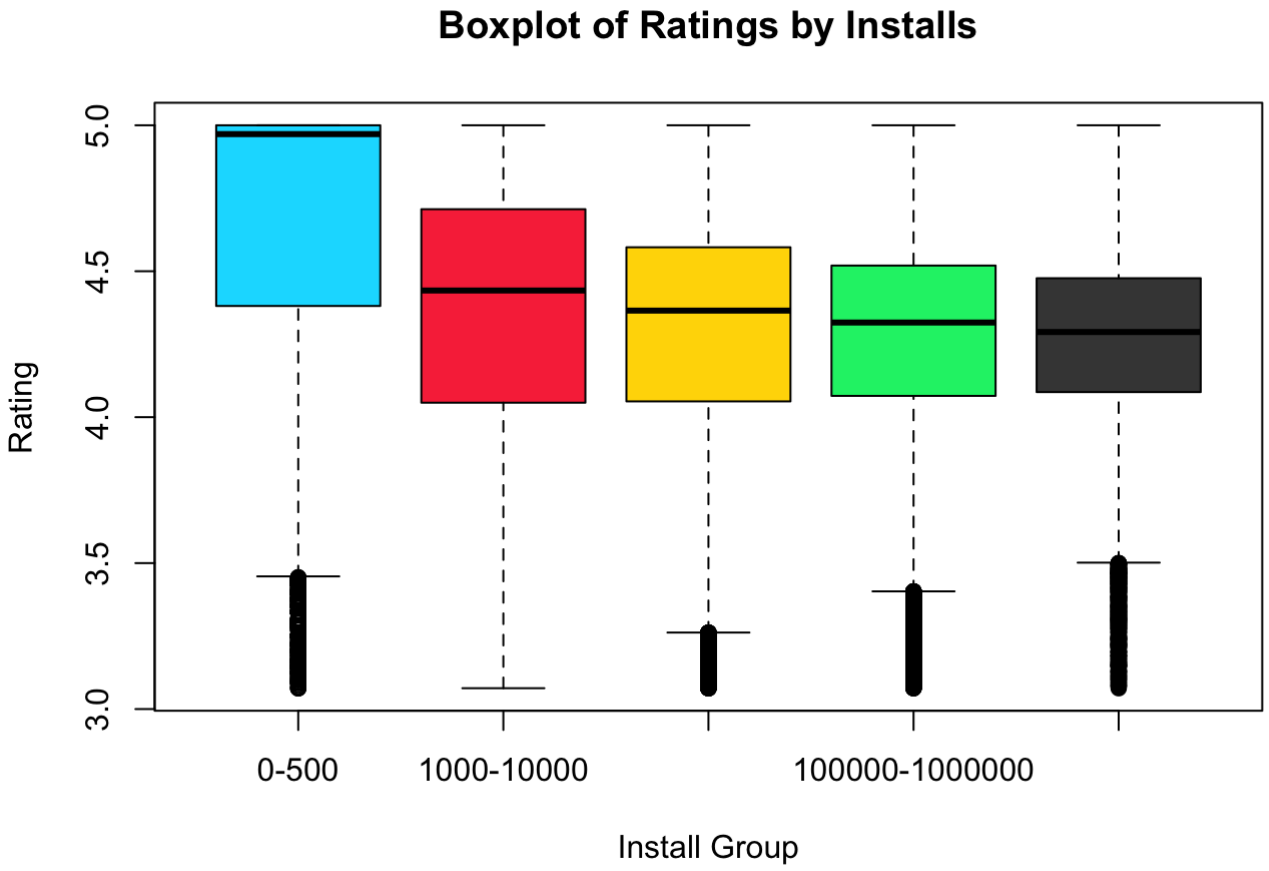
The results of the chi-squared test are higher that 0.05 so we will accept the null hypothesis. It seems that size of app has no effect on the number of installs an app receives. This may actually be true if we assume that many consumers do not look at app size before downloading an app. However, the small amount of data we have on “varies with device” could be the reason the p-value is a 1. More data would be needed to conduct a more succesful test.

```
g_apps11 %>% group_by(Installs) %>% summarise(avg = mean(Rating), median = median(Ratin
g), std = sd(Rating))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 5 x 4
##   Installs         avg median   std
##   <fct>          <dbl>  <dbl> <dbl>
## 1 0-500           4.65   4.97 0.473
## 2 1000-10000      4.35   4.43 0.464
## 3 10000-100000    4.28   4.37 0.403
## 4 100000-1000000  4.26   4.32 0.356
## 5 1000000+        4.25   4.29 0.312
```

```
boxplot(Rating~Installs, data = g_apps11, main="Boxplot of Ratings by Installs", xlab =
"Install Group",
        col = c("#00DCFF", "#F83648", "#FFD800", "#04F075", "#444444")) # Looks like app
s with fewer intalls have a higher rating on average
```



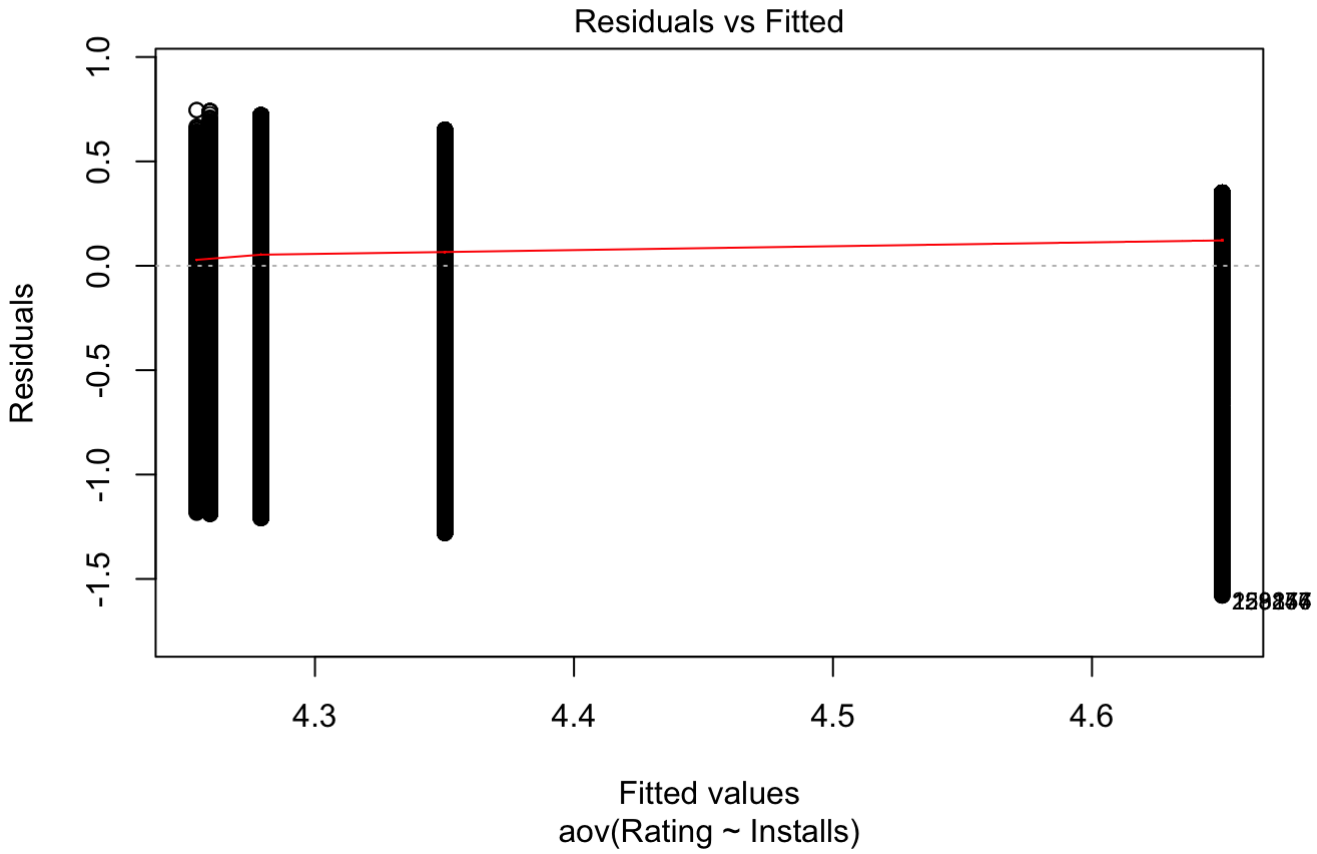

```
rat_install.aov <- aov(Rating~Installs, data = g_apps11)
summary(rat_install.aov)
```

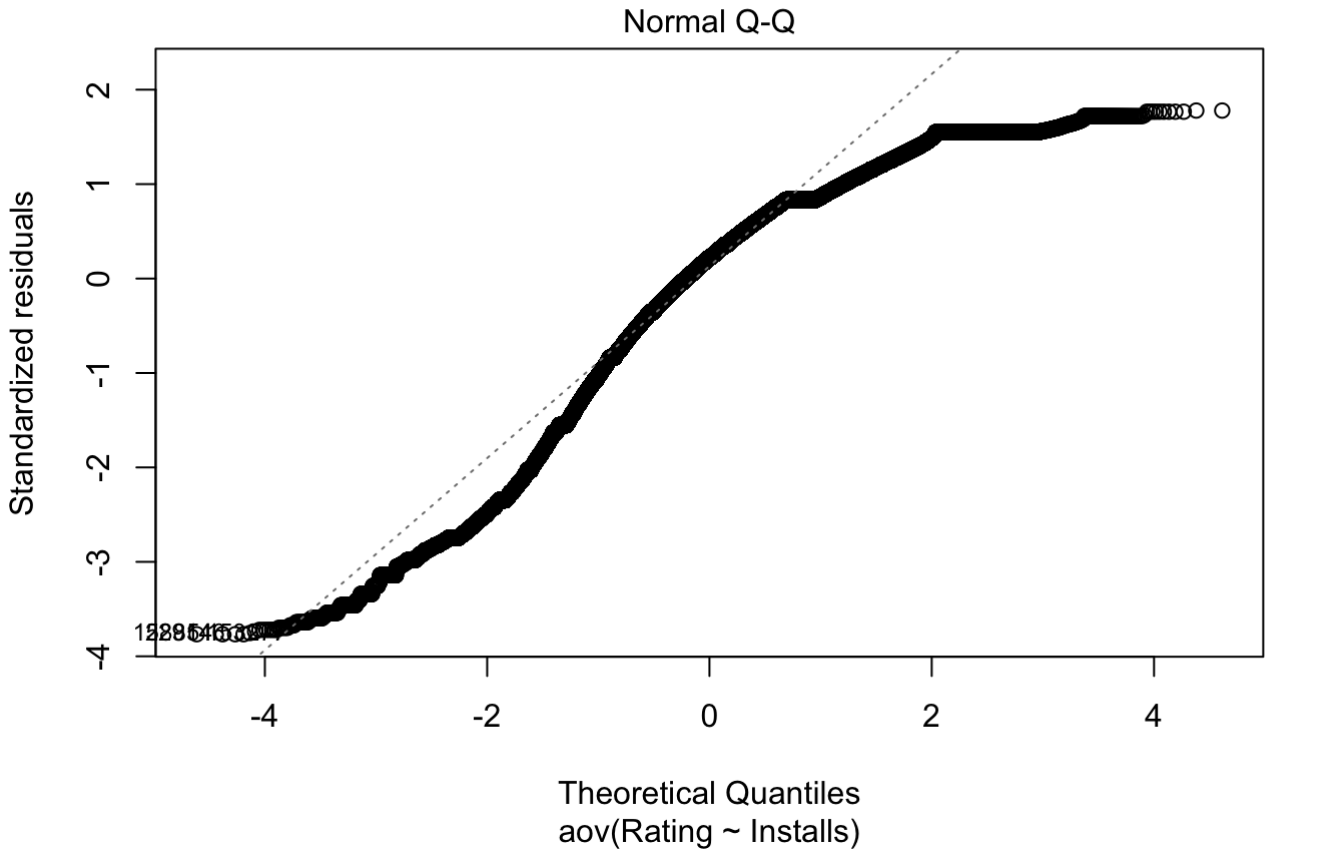
```
##                 Df Sum Sq Mean Sq F value              Pr(>F)
## Installs         4   4352  1087.9    6191 <0.0000000000000002 ***
## Residuals   252606  44389     0.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

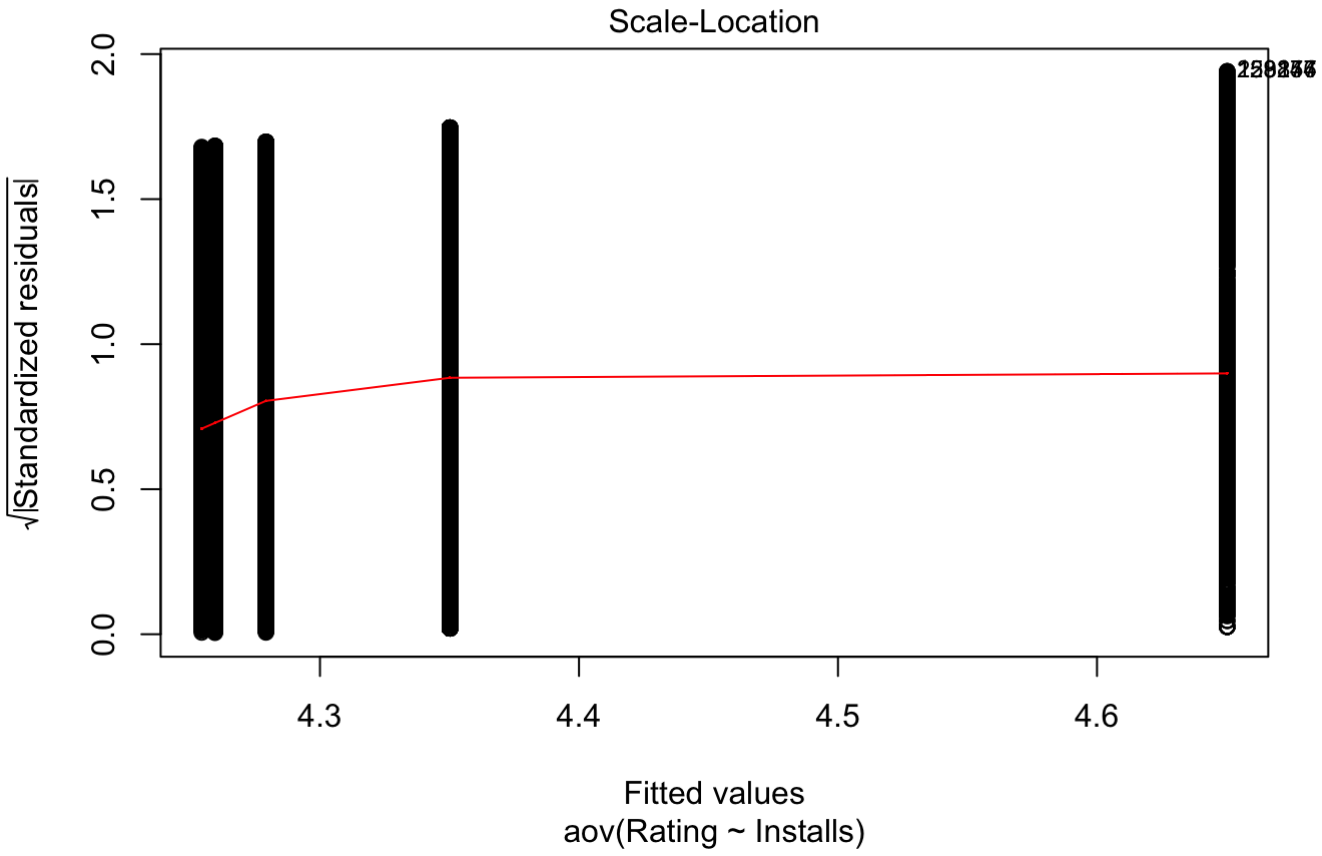
```
TukeyHSD(rat_install.aov) # For coomparison of levels
```

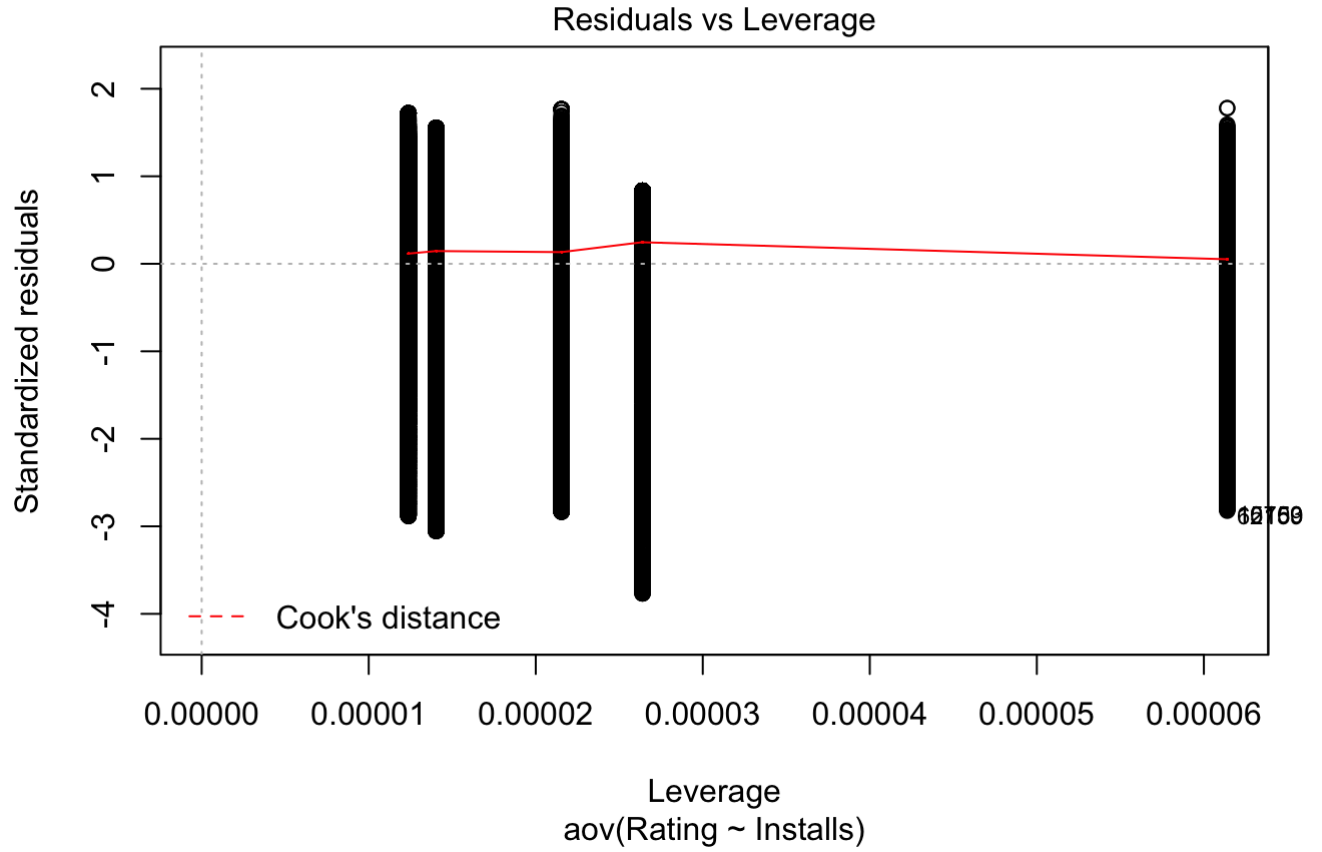
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Rating ~ Installs, data = g_apps11)
##
## $Installs
##                                     diff         lwr          upr
## 1000-10000-0-500            -0.300102885 -0.30737361 -0.292832160
## 10000-100000-0-500          -0.371134212 -0.37825409 -0.364014336
## 100000-1000000-0-500        -0.390823127 -0.39873979 -0.382906467
## 1000000+-0-500              -0.395944779 -0.40665866 -0.385230895
## 10000-100000-1000-10000     -0.071031327 -0.07690862 -0.065154031
## 100000-1000000-1000-10000   -0.090720242 -0.09754105 -0.083899431
## 1000000+-1000-10000         -0.095841894 -0.10577352 -0.085910264
## 100000-1000000-10000-100000 -0.019688915 -0.02634869 -0.013029135
## 1000000+-10000-100000       -0.024810567 -0.03463230 -0.014988833
## 1000000+-100000-1000000     -0.005121652 -0.01553546  0.005292153
##                                  p adj
## 1000-10000-0-500            0.0000000
## 10000-100000-0-500          0.0000000
## 100000-1000000-0-500        0.0000000
## 1000000+-0-500              0.0000000
## 10000-100000-1000-10000     0.0000000
## 100000-1000000-1000-10000   0.0000000
## 1000000+-1000-10000         0.0000000
## 100000-1000000-10000-100000 0.0000000
## 1000000+-10000-100000       0.0000000
## 1000000+-100000-1000000     0.6651233
```

```
#verify ANOVA assumptions with diagnostic plots
plot(rat_install.aov)
```


Residuals vs Fitted
Residuals
1.0
0.5
0.0
-0.5
-1.0
-1.5
4.3
4.4
4.5
4.6
Fitted values
aov(Rating ~ Installs)


Normal Q-Q
Standardized residuals
2
1
0
-1
-2
-3
-4
-4
-2
0
2
4
Theoretical Quantiles
aov(Rating ~ Installs)





Here we can see that apps with fewer installs have higher ratings on average. This makes sense because apps with a

high number of installs would have a large number of ratings. Each rating would carry less weight and at the same time a large number of people rating an app would likely introduce a wider range of ratings, driving the average rating down over time.

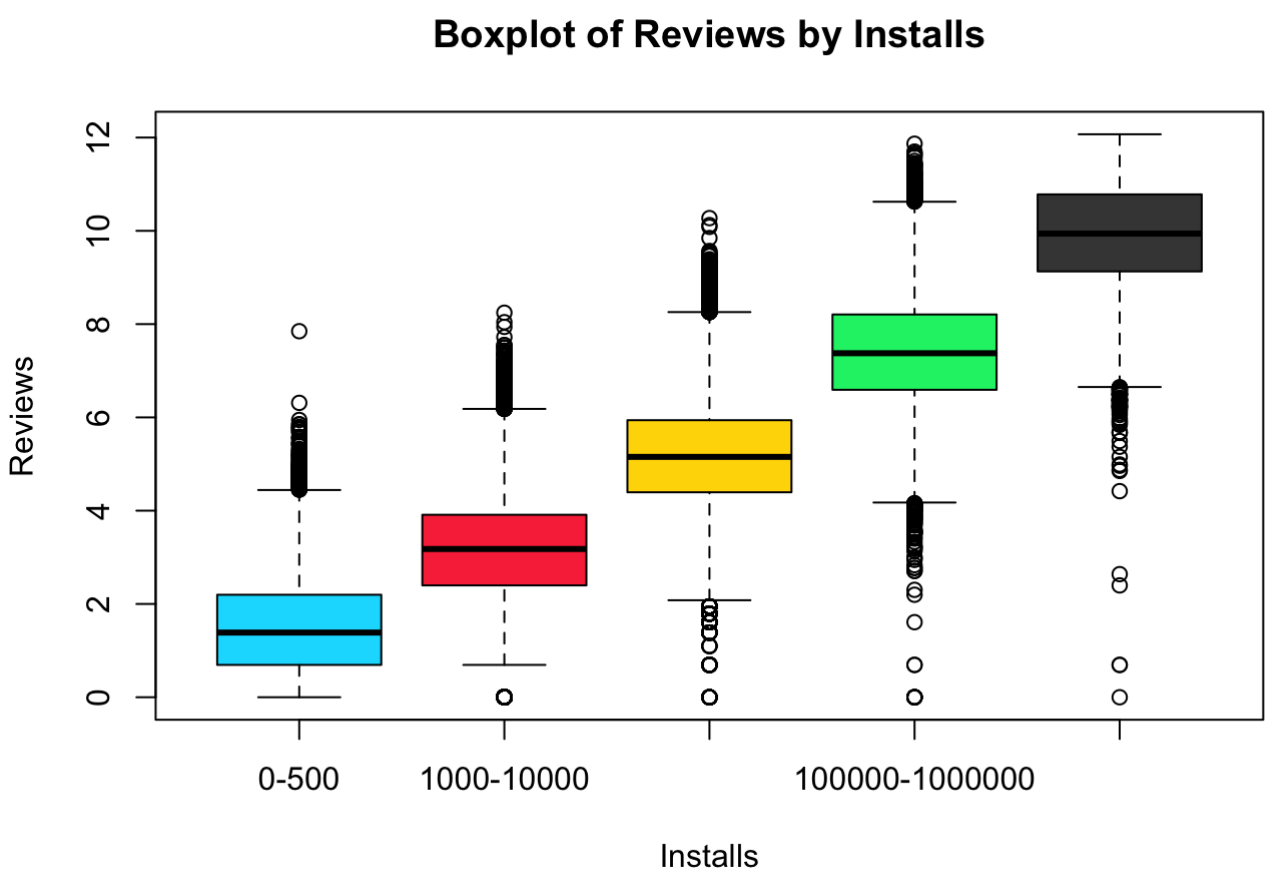
The p-value for the ANOVA test was low enough to reject the null hypothesis. I also used diagnostics plots to verify that the ANOVA assumptions were correct. The data does have some outliers but in general the plots look good. More specifically, the Residuals vs. Fitted plot shows that there is homogeniety among variances as there is no relationship between the residuals of each group. The Normal Q-Q plot tells us that residuals are normally distributed.

```
g_apps11 %>% group_by(Installs) %>% summarise(avg = mean(Reviews), median = median(Revie
ws), std = sd(Reviews))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 5 x 4
##   Installs         avg median   std
##   <fct>          <dbl>  <dbl> <dbl>
## 1 0-500           1.49   1.39  1.08
## 2 1000-10000      3.17   3.18  1.14
## 3 10000-100000    5.19   5.15  1.15
## 4 100000-1000000  7.41   7.38  1.18
## 5 1000000+        9.91   9.94  1.16
```

```
boxplot(Reviews~Installs, data = g_apps11, main="Boxplot of Reviews by Installs",
        col = c("#00DCFF", "#F83648", "#FFD800", "#04F075", "#444444")) # Looks fairly e
qualy, productivity has a slightly less rating.
```



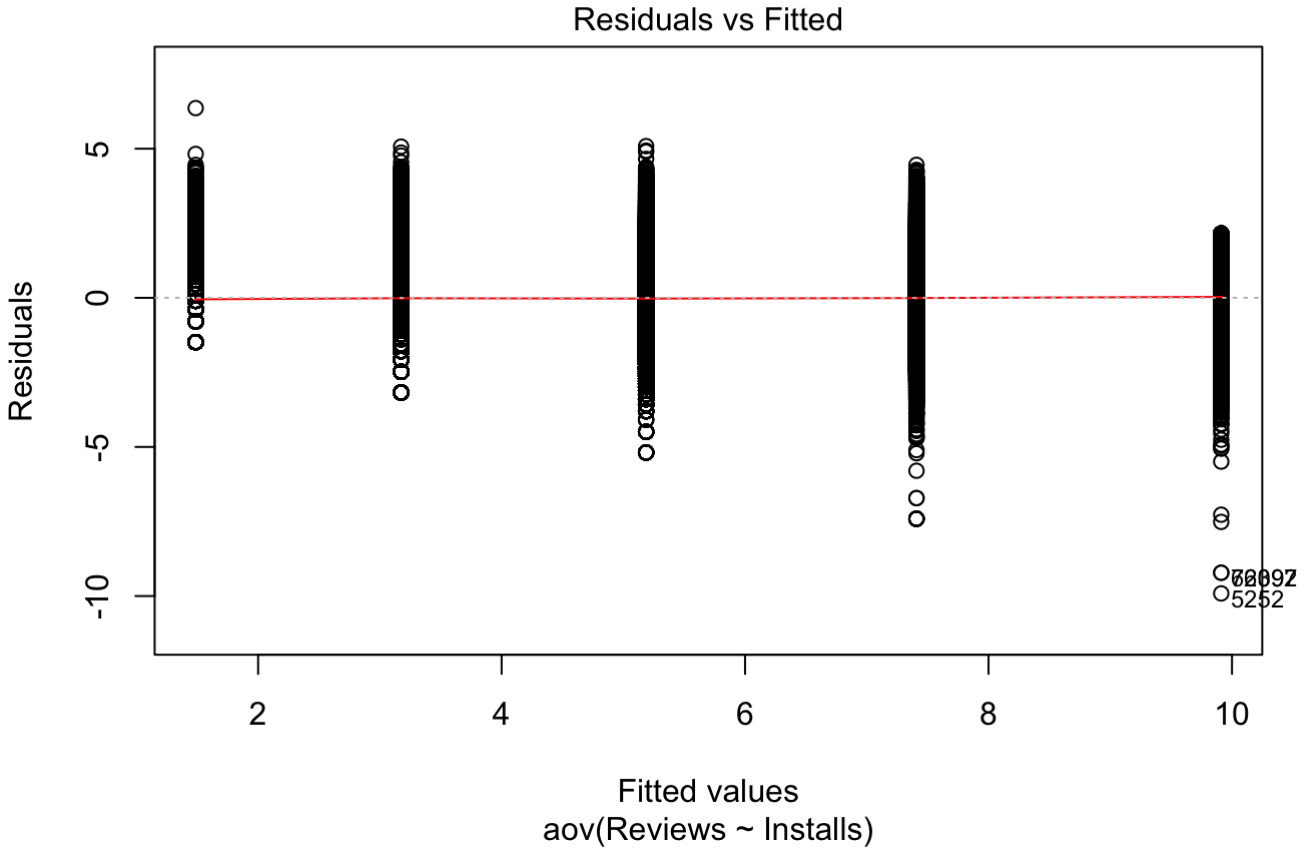

```
rev_installs.aov <- aov(Reviews~Installs, data = g_apps11)
summary(rev_installs.aov)
```

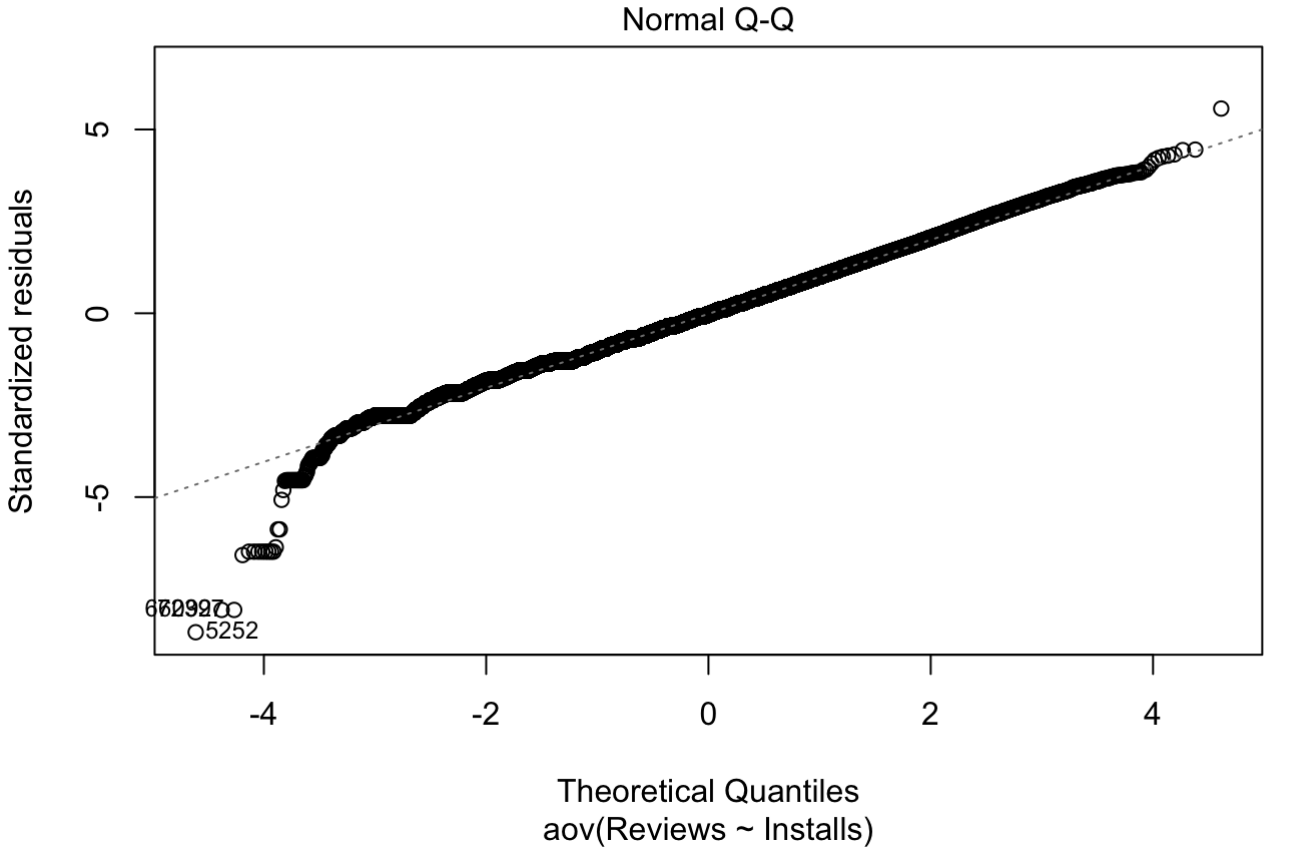
```
##                 Df  Sum Sq Mean Sq F value              Pr(>F)
## Installs         4 1357431  339358  260055 <0.0000000000000002 ***
## Residuals   252606  329638       1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

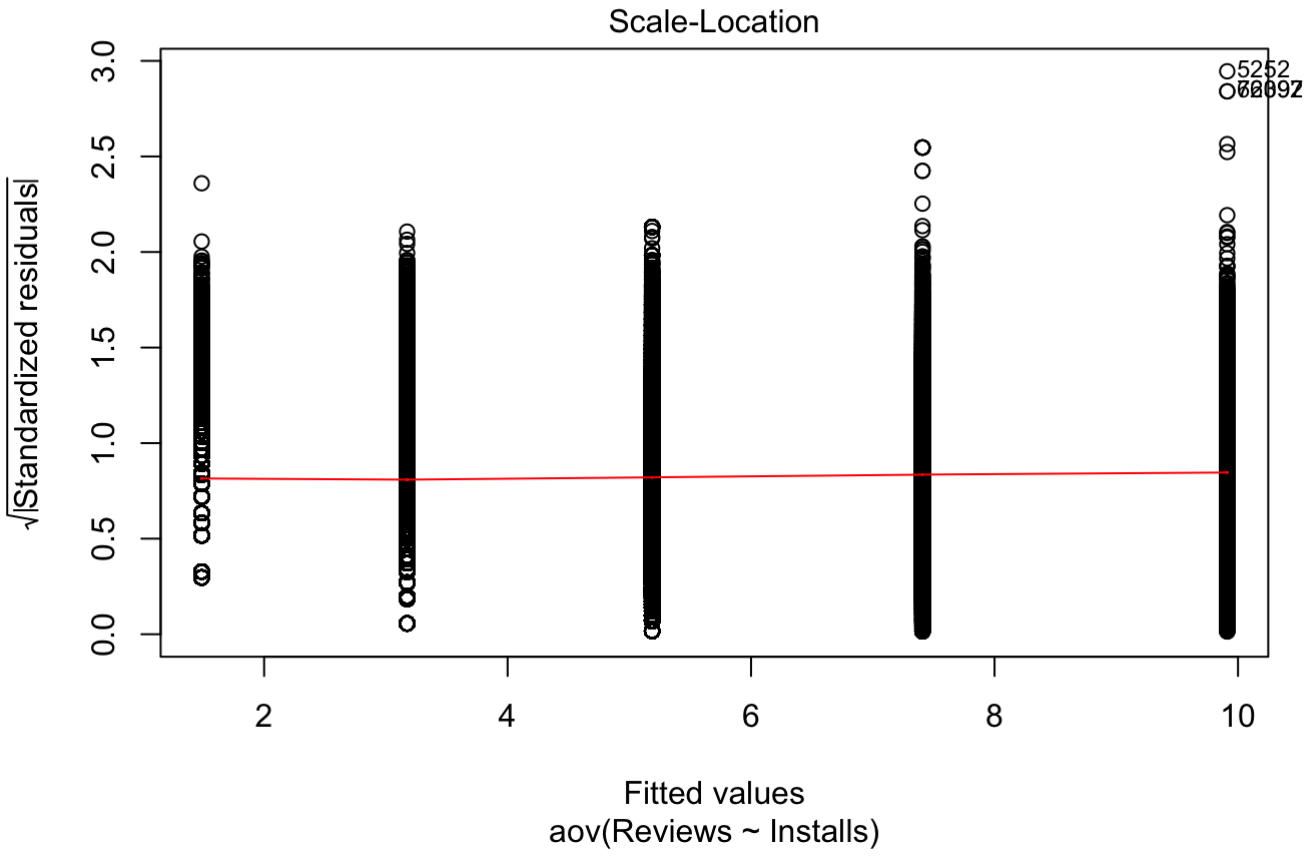
```
TukeyHSD(rev_installs.aov) # For coomparison of levels
```

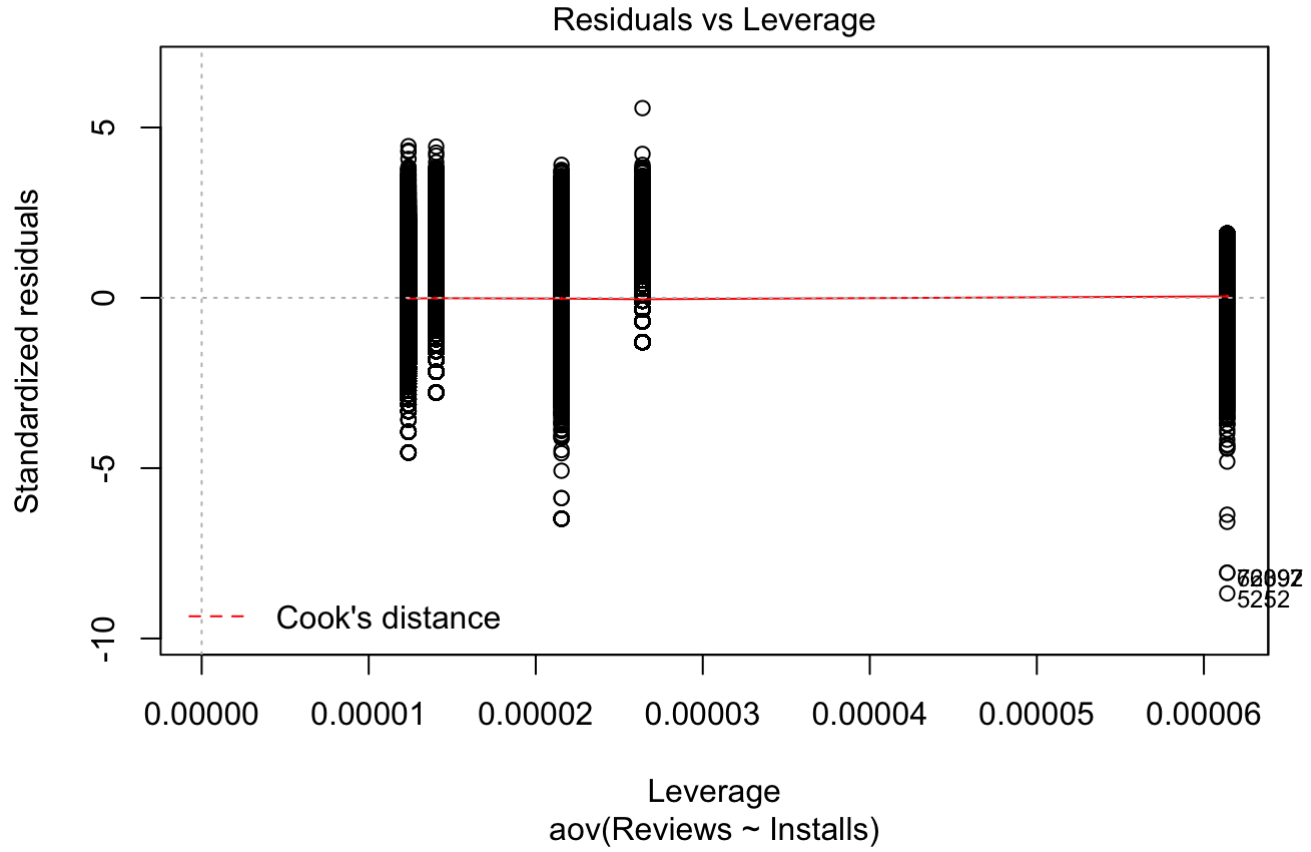
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Reviews ~ Installs, data = g_apps11)
##
## $Installs
##                                   diff      lwr      upr p adj
## 1000-10000-0-500             1.687458 1.667645 1.707272     0
## 10000-100000-0-500           3.700072 3.680670 3.719474     0
## 100000-1000000-0-500         5.921287 5.899714 5.942861     0
## 1000000+-0-500               8.424466 8.395270 8.453662     0
## 10000-100000-1000-10000      2.012613 1.996597 2.028629     0
## 100000-1000000-1000-10000    4.233829 4.215242 4.252416     0
## 1000000+-1000-10000          6.737008 6.709943 6.764072     0
## 100000-1000000-10000-100000  2.221216 2.203067 2.239364     0
## 1000000+-10000-100000        4.724394 4.697629 4.751159     0
## 1000000+-100000-1000000      2.503179 2.474800 2.531557     0
```

```
#verify ANOVA assumptions with diagnostic plots
plot(rev_installs.aov) # we have outliers
```


Residuals vs Fitted
Residuals
5
0
-5
-10
2
4
6
8
10
5252
Fitted values
aov(Reviews ~ Installs)


Normal Q-Q
Standardized residuals
5
0
-5
5252
-4
-2
0
2
4
Theoretical Quantiles
aov(Reviews ~ Installs)





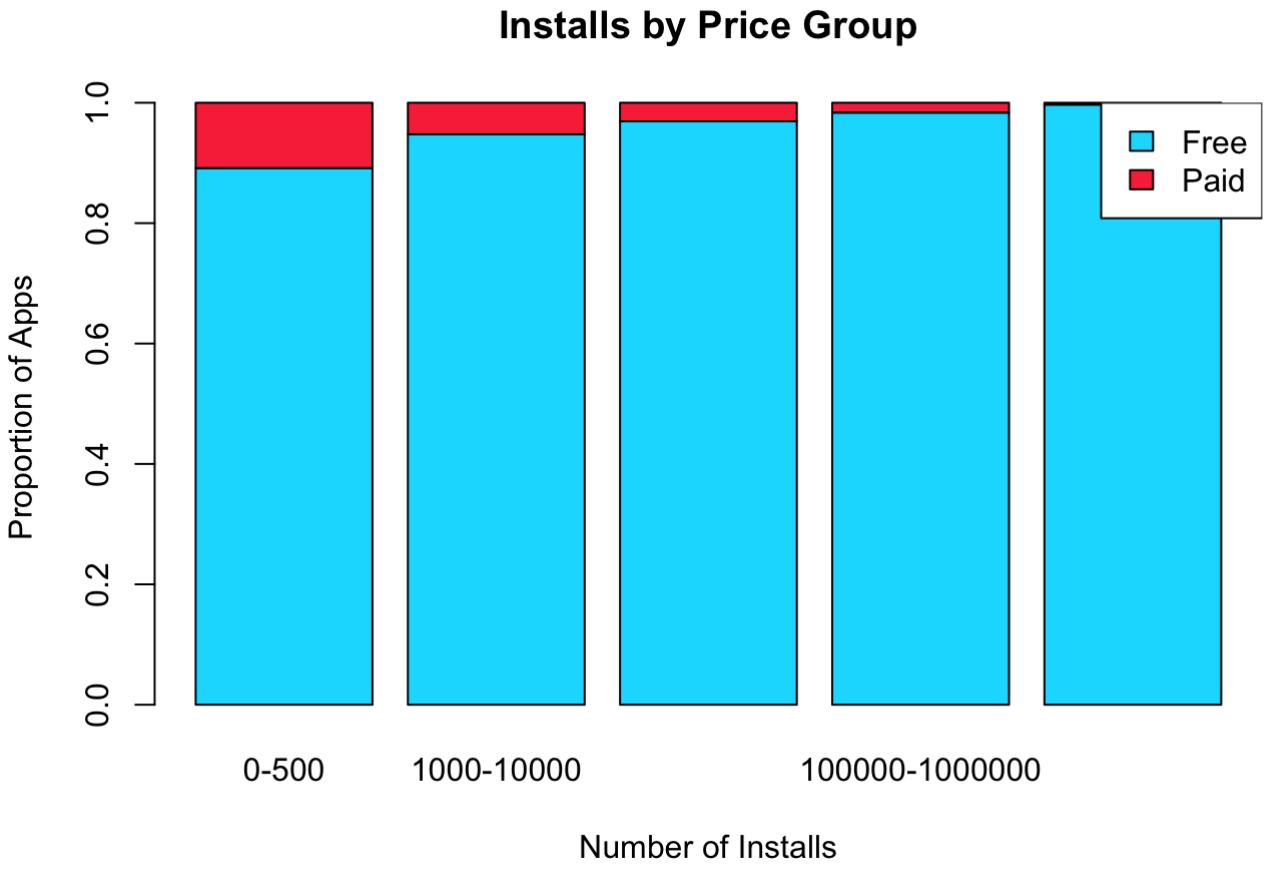
As expected, the apps with over 1 million installs have the highest number of reviews on average and apps with

fewer than 1,000 installs have the lowest number of reviews. The p-value for the ANOVA test is low enough to make the assumption that these results are correct. Installs and Reviews have a positive relationship with each other. Diagnostic plots look good as well.

```
ptab <- table(g_apps11$Price, g_apps11$Installs)
pptab <- prop.table(ptab)
pptab <- round(pptab, 2)
addmargins(pptab, c(1,2))
```

```
##
##         0-500 1000-10000 10000-100000 100000-1000000 1000000+  Sum
##    Free  0.13       0.27          0.31            0.18     0.06 0.95
##    Paid  0.02       0.01          0.01            0.00     0.00 0.04
##    Sum   0.15       0.28          0.32            0.18     0.06 0.99
```

```
ptab_price <- prop.table(ptab, margin = 2)
barplot(ptab_price, col = c("#00DCFF", "#F83648"), main = "Installs by Price Group", xla
b = "Number of Installs", ylab = "Proportion of Apps")
legend("topright", legend = levels(g_apps11$Price), fill = c("#00DCFF", "#F83648")) # di
sproportionate so of course we expect this
```




```
chisq.test(pptab)
```

```
## Warning in chisq.test(pptab): Chi-squared approximation may be incorrect
```

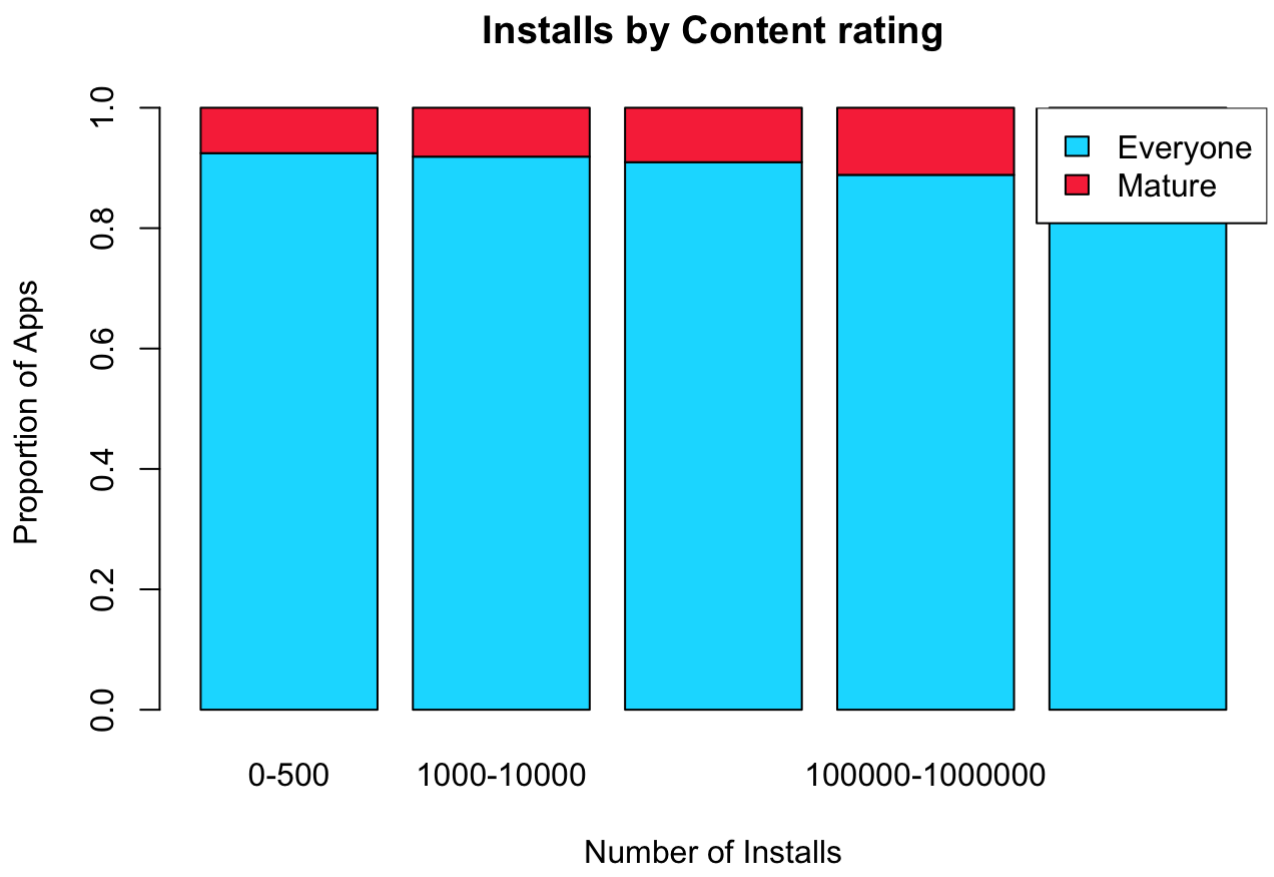
```
##
##  Pearson's Chi-squared test
##
## data:  pptab
## X-squared = 0.044366, df = 4, p-value = 0.9998
```

The p-value for the chi-squared test is 0.99 which is high enough to reject the null. This may be due to the disproportionate amount of data available on the price of apps in the dataset. I will not make any conclusions about Installs compared to the Price of an app.

```
crtab <- table(g_apps11$Content.Rating, g_apps11$Installs)
crtab <- prop.table(crtab)
pcrtab <- round(pptab, 2)
addmargins(pcrtab, c(1,2))
```

```
##
##        0-500 1000-10000 10000-100000 100000-1000000 1000000+  Sum
##   Free  0.13       0.27         0.31           0.18     0.06 0.95
##   Paid  0.02       0.01         0.01           0.00     0.00 0.04
##   Sum   0.15       0.28         0.32           0.18     0.06 0.99
```

```
ptab_cr <- prop.table(crtab, margin = 2)
barplot(ptab_cr, col = c("#00DCFF", "#F83648"), main = "Installs by Content rating", xla
b = "Number of Installs", ylab = "Proportion of Apps")
legend("topright", legend = levels(g_apps11$Content.Rating), fill = c("#00DCFF", "#F8364
8")) # disproportionate so of course we expect this
```




```
chisq.test(pcrtab)
```

```
## Warning in chisq.test(pcrtab): Chi-squared approximation may be incorrect
```

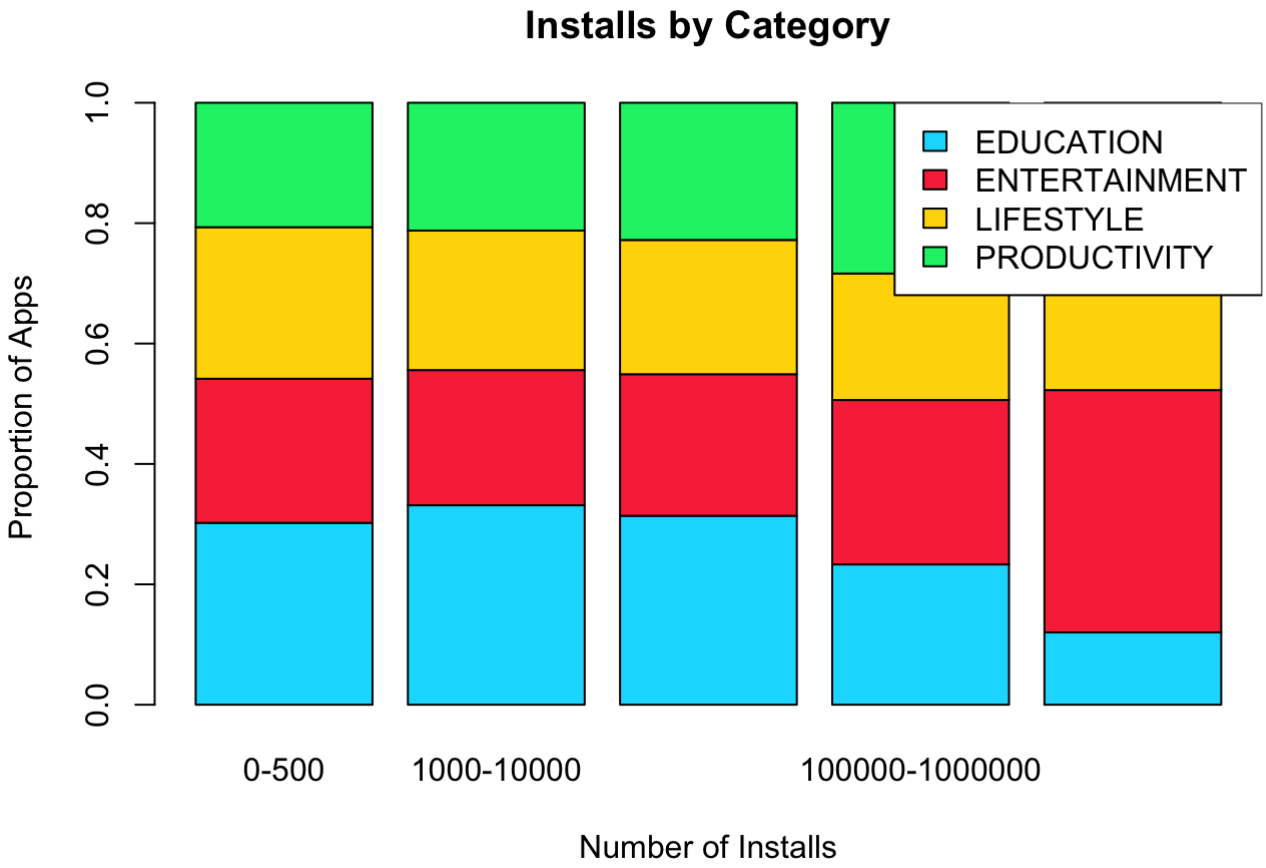
```
##
##  Pearson's Chi-squared test
##
## data:  pcrtab
## X-squared = 0.044366, df = 4, p-value = 0.9998
```

The p-value for the chi-squared test is 0.99 which is high enough to reject the null. This may be due to the disproportionate amount of data available on content rating in the dataset. I will not make any conclusions about Installs compared to the Content Rating of an app.

```
cattab <- table(g_apps11$Category, g_apps11$Installs)
pcattab <- prop.table(cattab)
pcattab <- round(pcattab, 2)
addmargins(pcattab, c(1,2))
```

```
##
##                0-500 1000-10000 10000-100000 100000-1000000 1000000+  Sum
##   EDUCATION     0.05       0.09         0.10           0.04     0.01 0.29
##   ENTERTAINMENT 0.04       0.06         0.08           0.05     0.03 0.26
##   LIFESTYLE     0.04       0.07         0.07           0.04     0.01 0.23
##   PRODUCTIVITY  0.03       0.06         0.07           0.05     0.02 0.23
##   Sum           0.16       0.28         0.32           0.18     0.07 1.01
```

```
ptab_cat <- prop.table(cattab, margin = 2)
barplot(ptab_cat, col = c("#00DCFF", "#F83648", "#FFD800", "#04F075"), main = "Installs
 by Category", xlab = "Number of Installs", ylab = "Proportion of Apps")
legend("topright", legend = levels(g_apps11$Category), fill = c("#00DCFF", "#F83648", "#
FFD800", "#04F075")) # disproportionate so of course we expect this
```




```
chisq.test(pcattab)
```

```
## Warning in chisq.test(pcattab): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  pcattab
## X-squared = 0.028116, df = 12, p-value = 1
```

The p-value for the chi-squared test is 1. It seems that Category has no effect on the number of Installs an app receives.

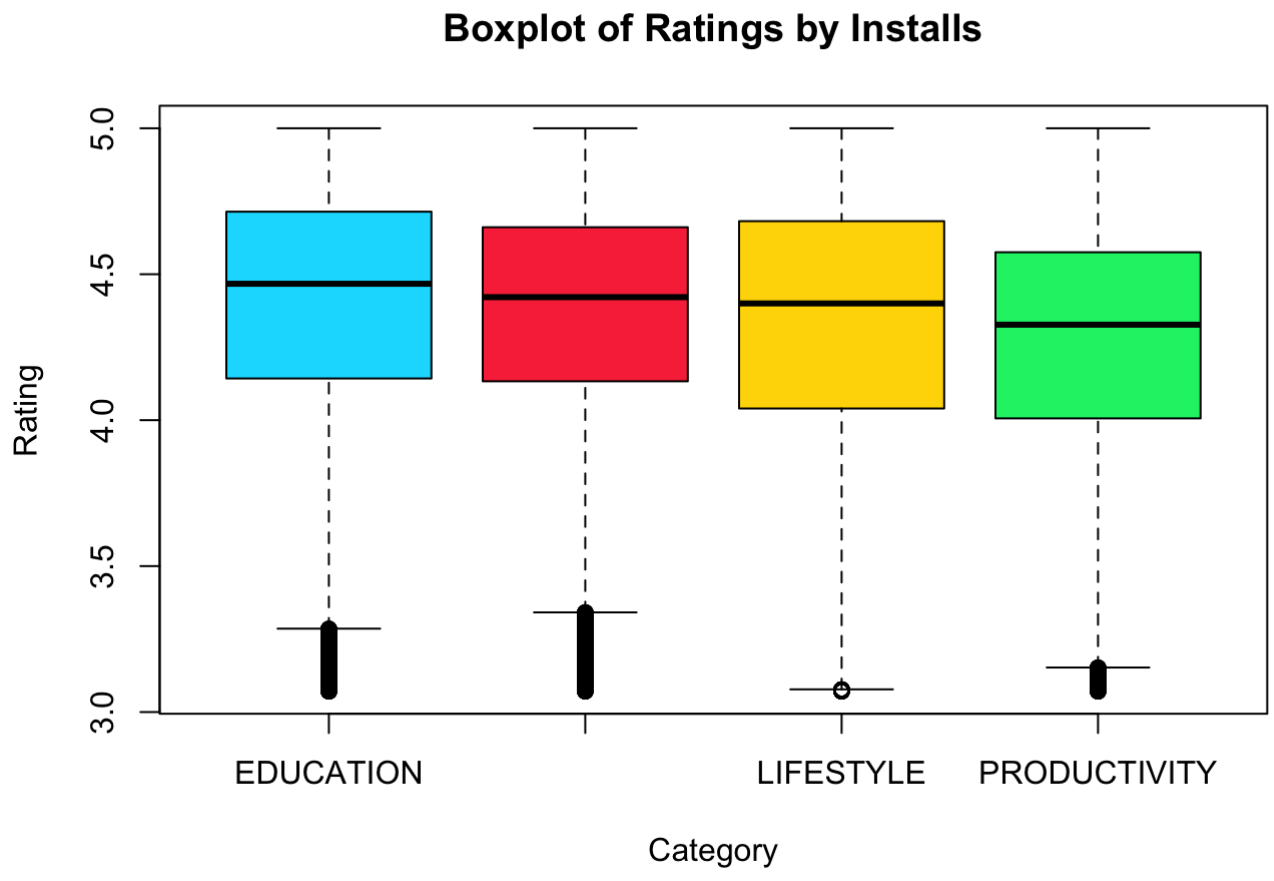
Hypothesis Testing

```
# Hypothesis 1: Education apps will have higher ratings that all other apps on average.

g_apps11 %>% group_by(Category) %>% summarise(avg = mean(Rating), median = median(Ratin
g), std = sd(Rating))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 4 x 4
##   Category        avg median   std
##   <fct>         <dbl>  <dbl> <dbl>
## 1 EDUCATION      4.39   4.47 0.440
## 2 ENTERTAINMENT  4.37   4.42 0.412
## 3 LIFESTYLE      4.34   4.40 0.467
## 4 PRODUCTIVITY   4.28   4.33 0.431
```

```
boxplot(Rating~Category, data = g_apps11, main="Boxplot of Ratings by Installs",
        col = c("#00DCFF", "#F83648", "#FFD800", "#04F075")) # Looks fairly equaly, prod
uctivity has a slightly less rating.
```



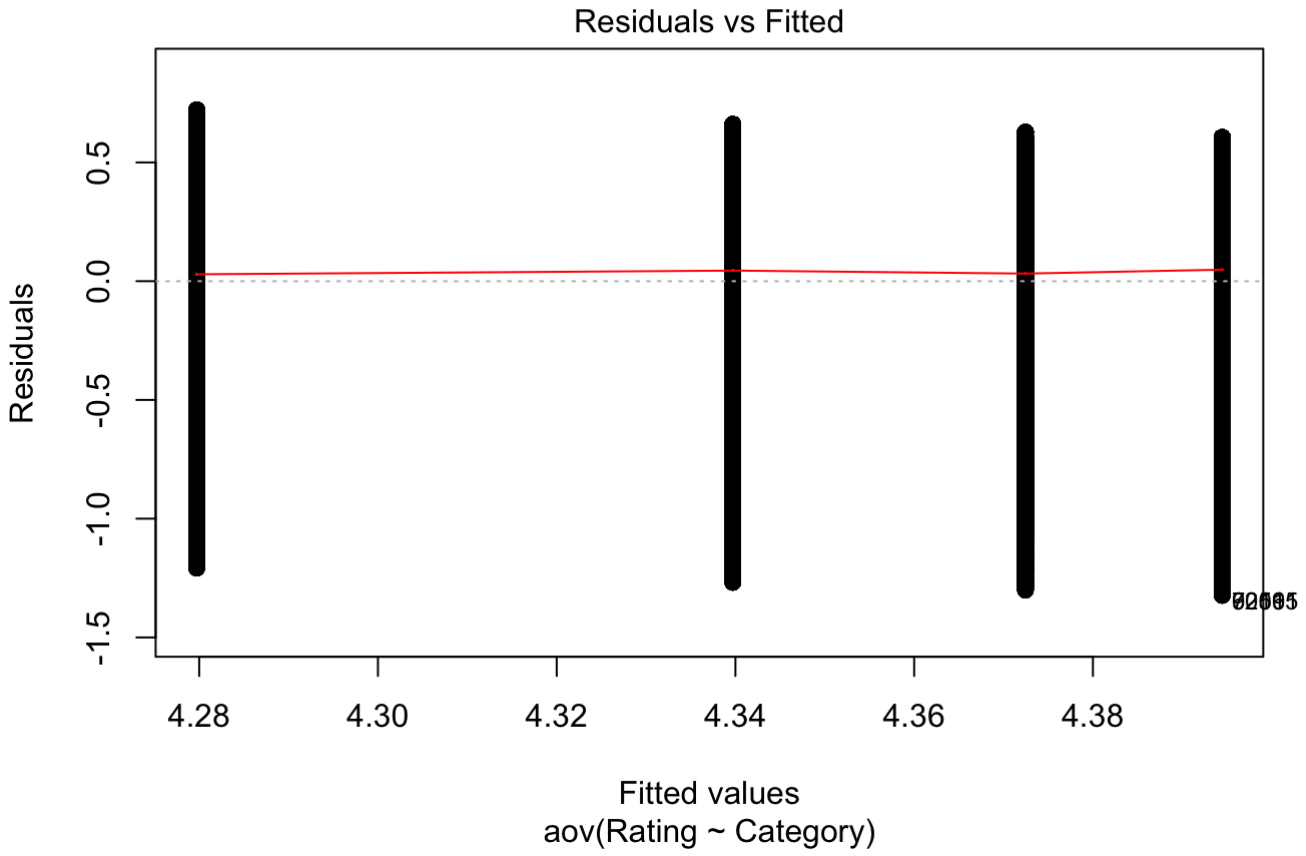
```
rat_category.aov <- aov(Rating~Category, data = g_apps11)
summary(rat_category.aov)
```

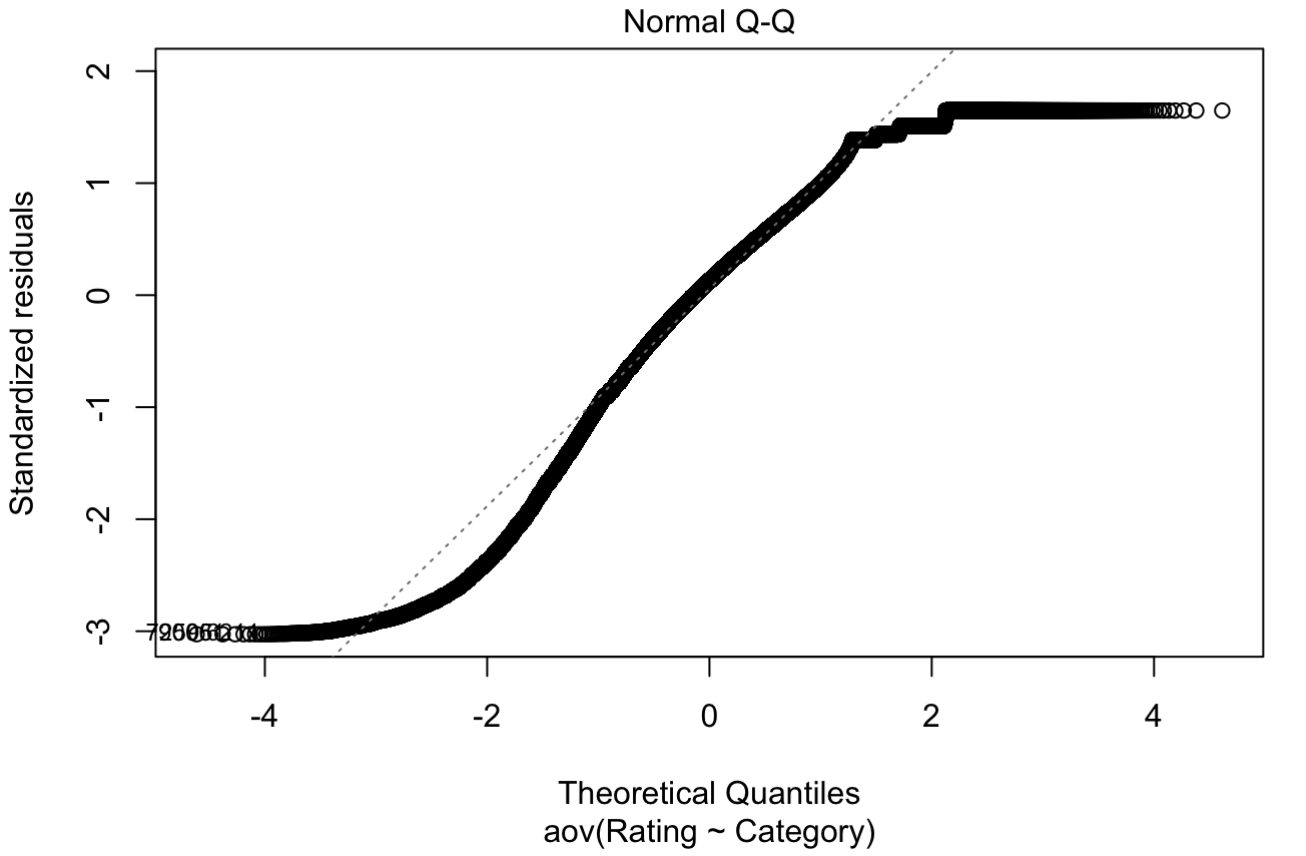
```
##                 Df Sum Sq Mean Sq F value              Pr(>F)
## Category         3    477  159.01   832.3 <0.0000000000000002 ***
## Residuals   252607  48264    0.19
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

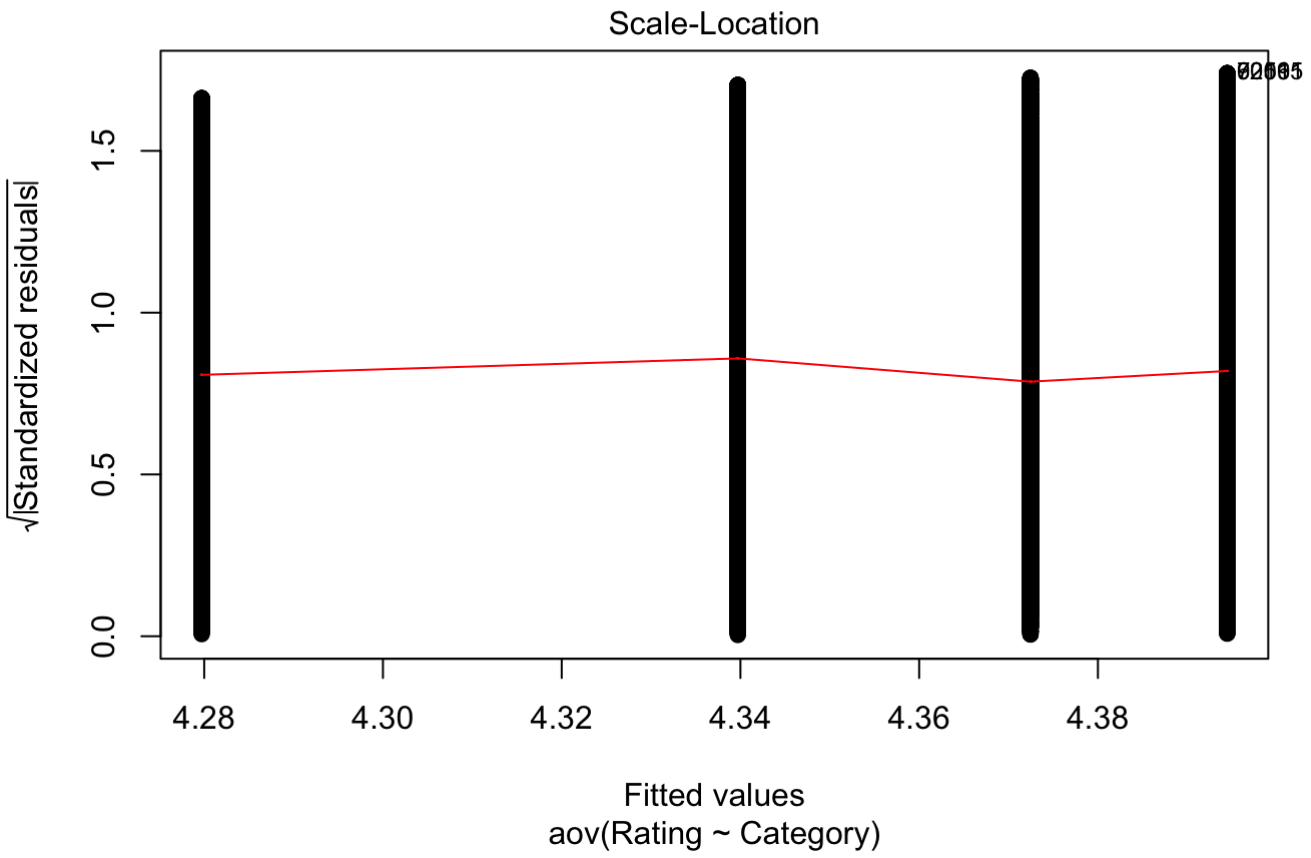
```
TukeyHSD(rat_category.aov) # For coomparison of levels
```

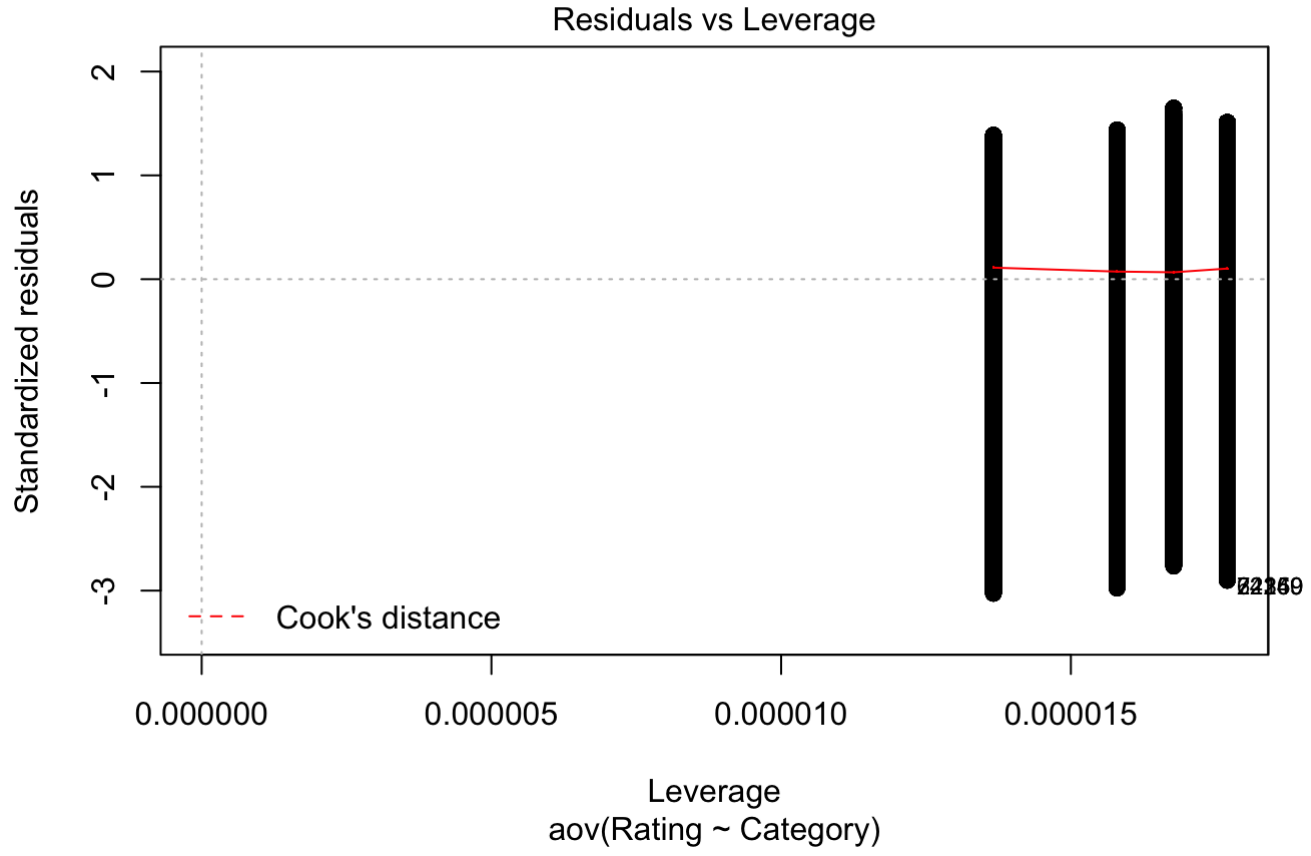
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Rating ~ Category, data = g_apps11)
##
## $Category
##                                   diff         lwr         upr p adj
## ENTERTAINMENT-EDUCATION    -0.02201544 -0.02811043 -0.01592046     0
## LIFESTYLE-EDUCATION        -0.05476119 -0.06104986 -0.04847251     0
## PRODUCTIVITY-EDUCATION     -0.11474899 -0.12094437 -0.10855361     0
## LIFESTYLE-ENTERTAINMENT    -0.03274574 -0.03924470 -0.02624679     0
## PRODUCTIVITY-ENTERTAINMENT -0.09273355 -0.09914227 -0.08632482     0
## PRODUCTIVITY-LIFESTYLE     -0.05998780 -0.06658101 -0.05339460     0
```

```
#verify ANOVA assumptions with diagnostic plots
plot(rat_category.aov) # we have outliers
```


Residuals vs Fitted
Residuals
0.5
0.0
-0.5
-1.0
-1.5
4.28
4.30
4.32
4.34
4.36
4.38
Fitted values
aov(Rating ~ Category)


Normal Q-Q
Standardized residuals
2
1
0
-1
-2
-3
-4
-2
0
2
4
Theoretical Quantiles
aov(Rating ~ Category)


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
4.28
4.30
4.32
4.34
4.36
4.38
Fitted values
aov(Rating ~ Category)


Residuals vs Leverage
Standardized residuals
2
1
0
-1
-2
-3
Cook's distance
0.000000
0.000005
0.000010
0.000015
Leverage
aov(Rating ~ Category)

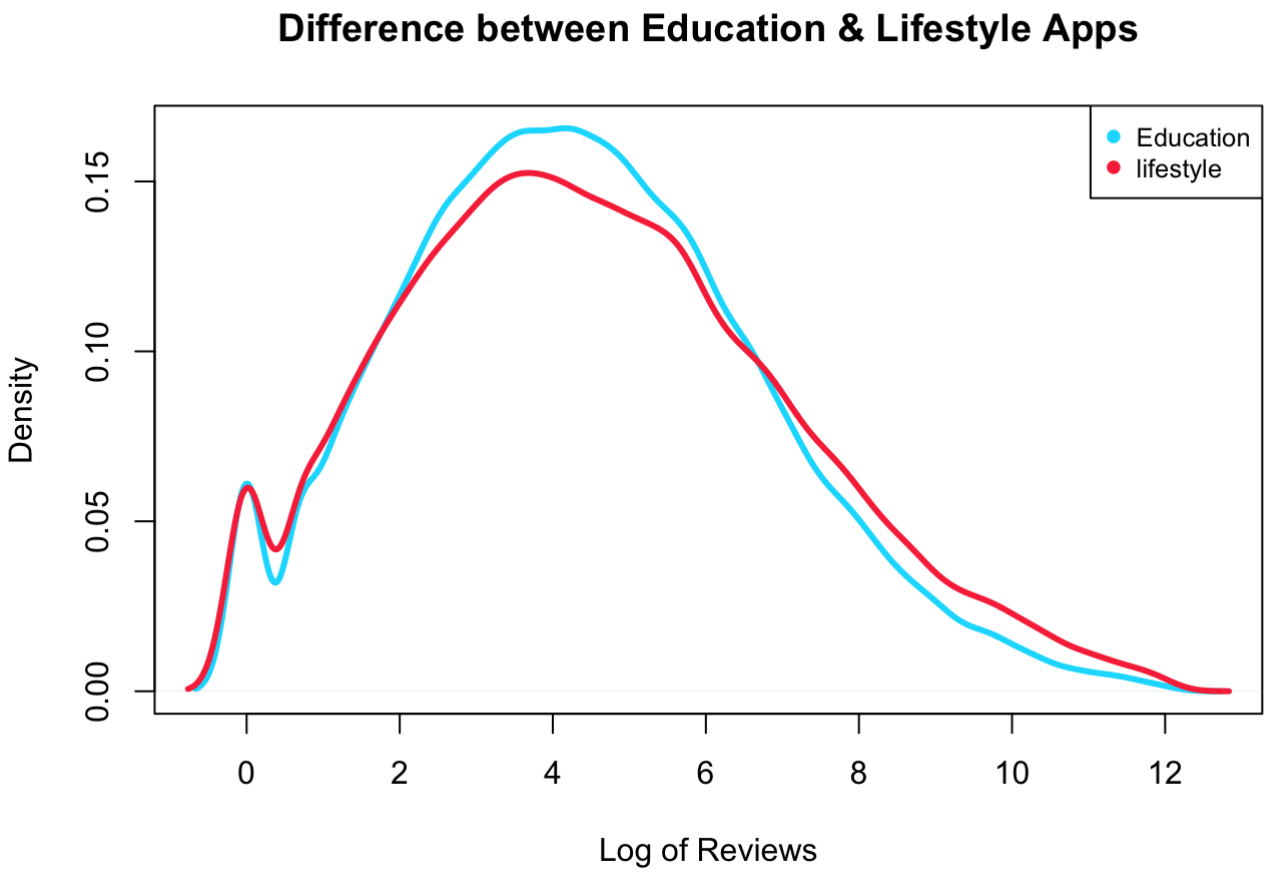
The results show that we can reject the null hypothesis. The Tukey pairwise test shows that there is a significant difference between the means of each category and Education apps actually do have higher ratings on average. Productivity apps have the lowest average rating of the 4 groups. Test results are significant.

I also used diagnostics plots to verify that the ANOVA assumptions were correct. The data does have some outliers but in general the plots look good. More specifically, the Residuals vs. Fitted plot shows that there is homogeniety among variances as there is no relationship between the residuals of each group. The Normal Q-Q plot tells us that residuals are normally distributed.

```
# Hypothesis 2: Education apps will have a higher number of reviews than Lifestyle apps
life_reviews <- mean(g_apps11$Reviews[g_apps11$Category=="LIFESTYLE"])
RC_Hypo <- t.test(g_apps11$Reviews[g_apps11$Category=="EDUCATION"],
                  alternative="greater",
                  mu=life_reviews,
                  conf.level=0.95)
RC_Hypo
```

```
##
##  One Sample t-test
##
## data:  g_apps11$Reviews[g_apps11$Category == "EDUCATION"]
## t = -18.972, df = 73186, p-value = 1
## alternative hypothesis: true mean is greater than 4.535357
## 95 percent confidence interval:
##  4.358875      Inf
## sample estimates:
## mean of x
##  4.372955
```

```
# Let's plot the difference
Edu <- g_apps11$Reviews[g_apps11$Category =="EDUCATION"]
Life <- g_apps11$Reviews[g_apps11$Category == "LIFESTYLE"]
plot(density(Edu), main = "Difference between Education & Lifestyle Apps", lwd= 3, col=
"#00DCFF", xlab = "Log of Reviews")
lines(density(Life), col="#F83648", lwd=3)
legend("topright", c("Education", "lifestyle"), col = c("#00DCFF", "#F83648"), pch = c(1
9,19), cex = 0.8)
```



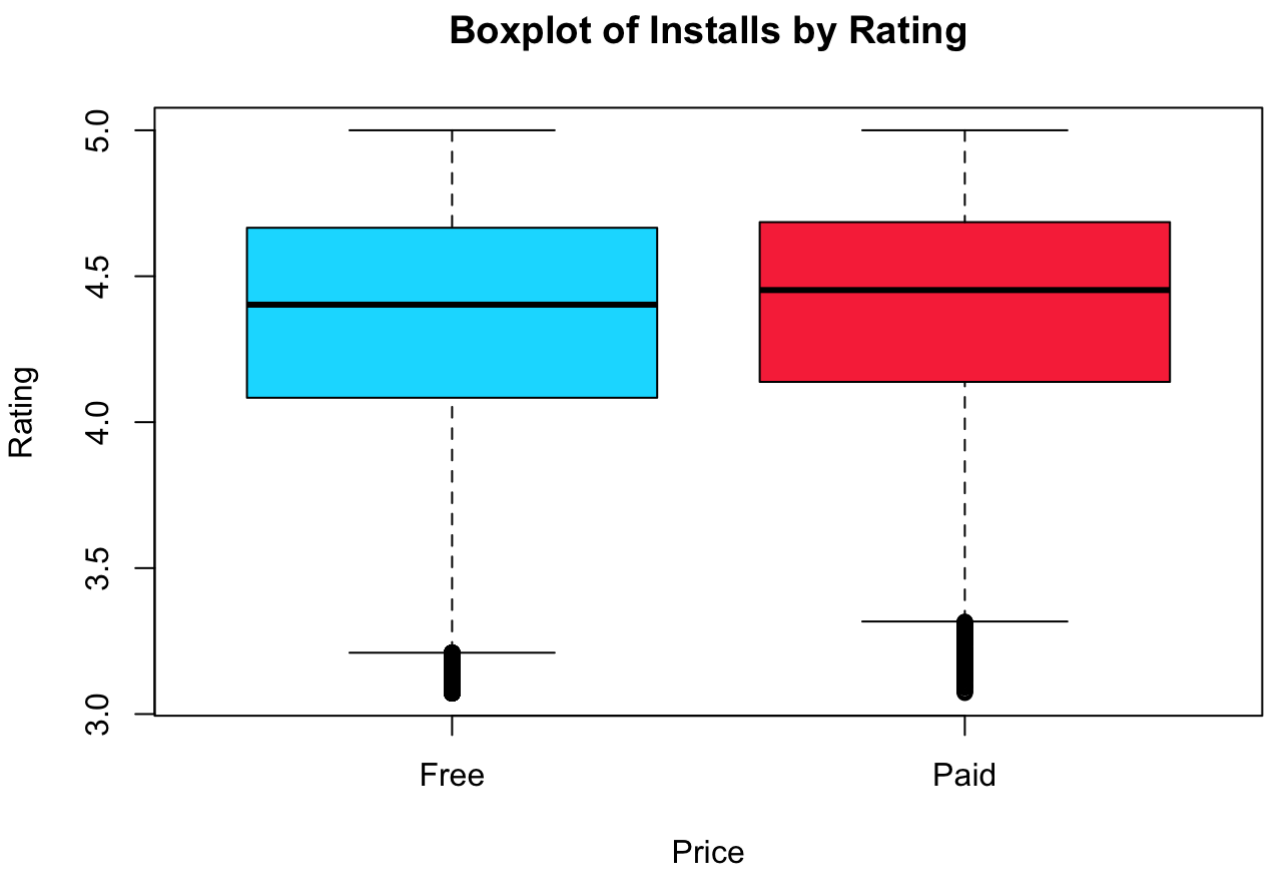
I will accept the null hypothesis for this test as the p-value is equal to 1.

```
# Hypothesis 3: On average, Paid apps will be rated higher than free apps.

g_apps12 <- g_apps11
g_apps12 %>% group_by(Price) %>% summarise(avg = mean(Rating), median = median(Rating),
 std = sd(Rating))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 4
##   Price   avg median   std
##   <fct> <dbl>  <dbl> <dbl>
## 1 Free   4.35   4.40 0.440
## 2 Paid   4.39   4.45 0.420
```

```
boxplot(Rating~Price, data = g_apps12, main="Boxplot of Installs by Rating",
        col = c("#00DCFF", "#F83648")) #Very similar results
```




```
rat_price.aov <- aov(Rating~Price, data=g_apps11)
rat_price.aov
```

```
## Call:
##    aov(formula = Rating ~ Price, data = g_apps11)
##
## Terms:
##                     Price Residuals
## Sum of Squares      16.65  48724.43
## Deg. of Freedom         1    252609
##
## Residual standard error: 0.4391865
## Estimated effects may be unbalanced
```

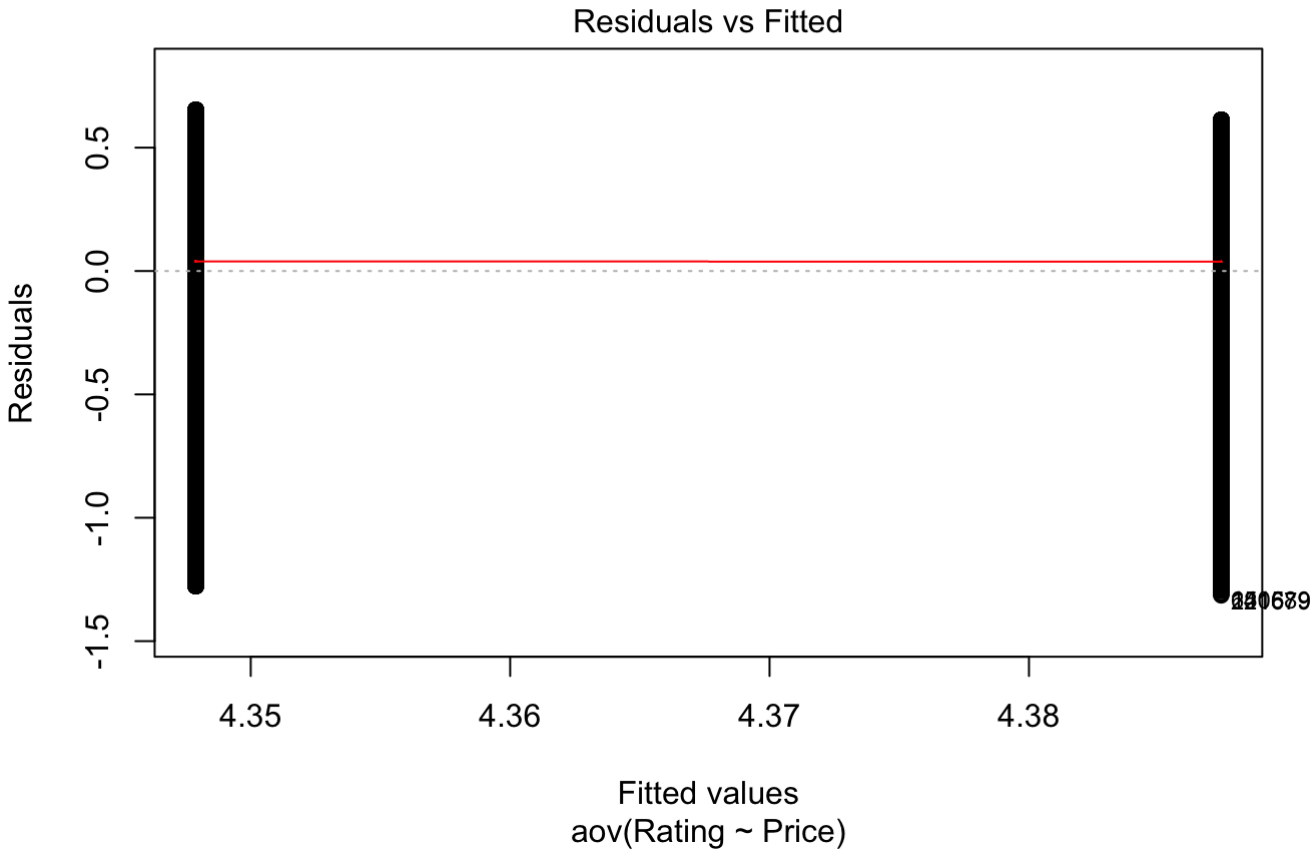
```
summary(rat_price.aov)
```

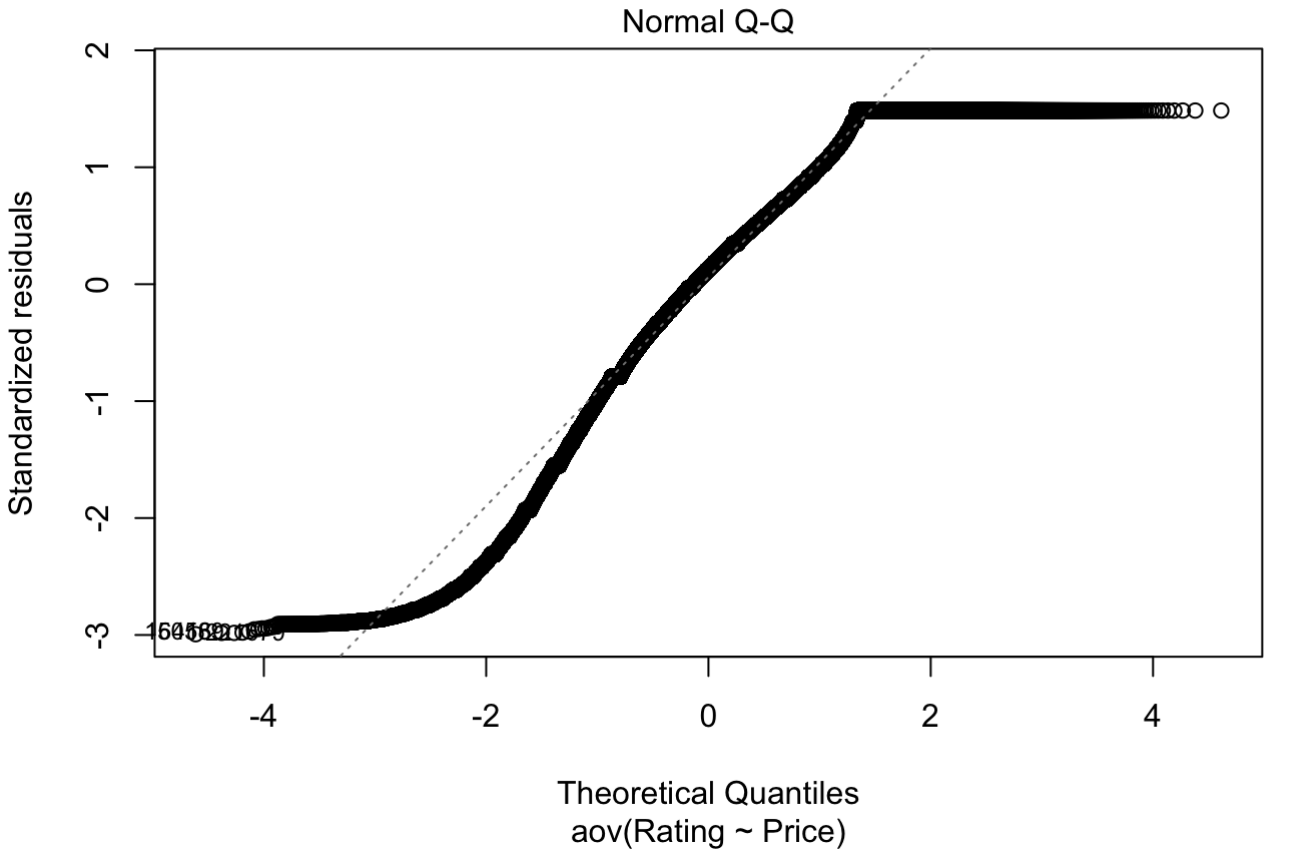
```
##                 Df Sum Sq Mean Sq F value              Pr(>F)
## Price            1     17  16.651   86.33 <0.0000000000000002 ***
## Residuals   252609  48724   0.193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

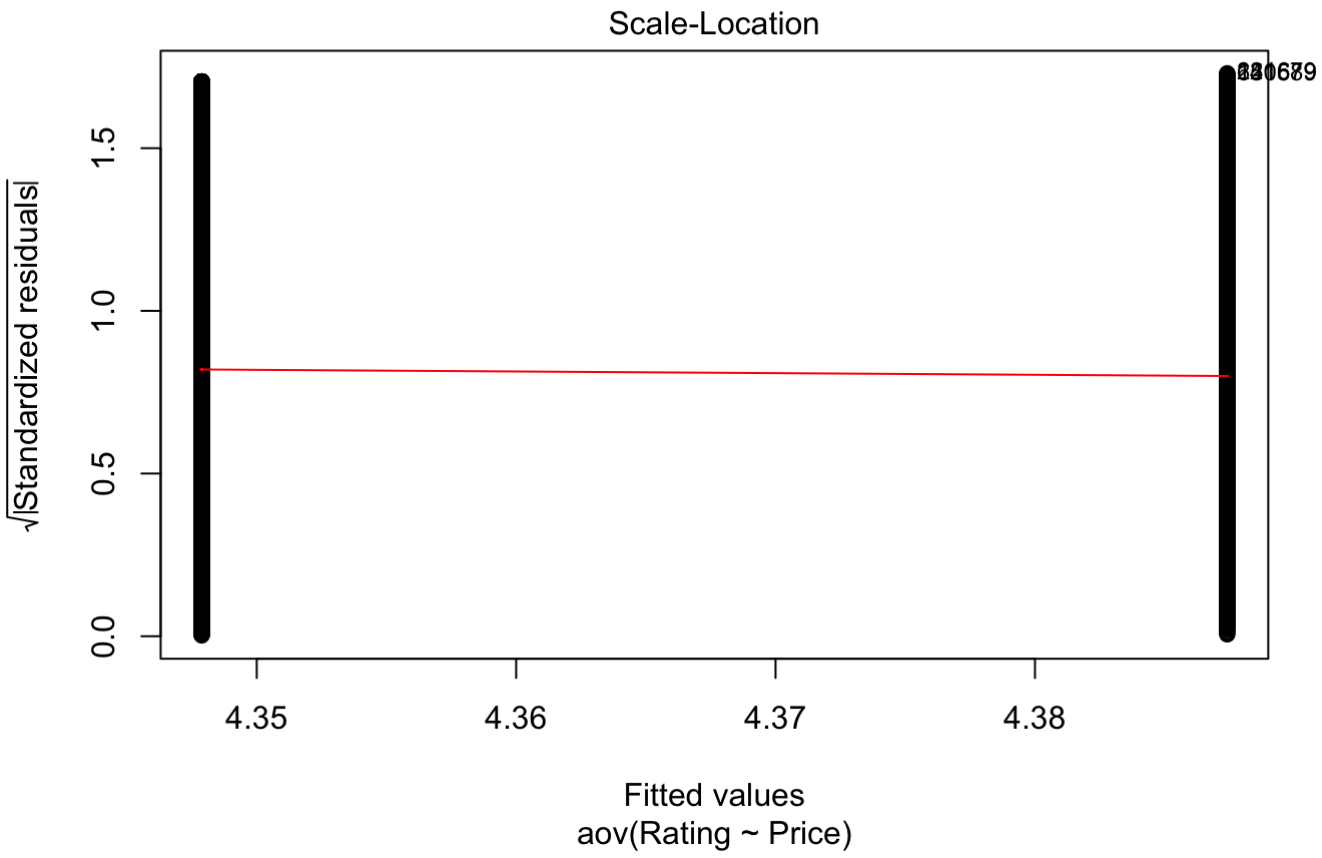
```
TukeyHSD(rat_price.aov)
```

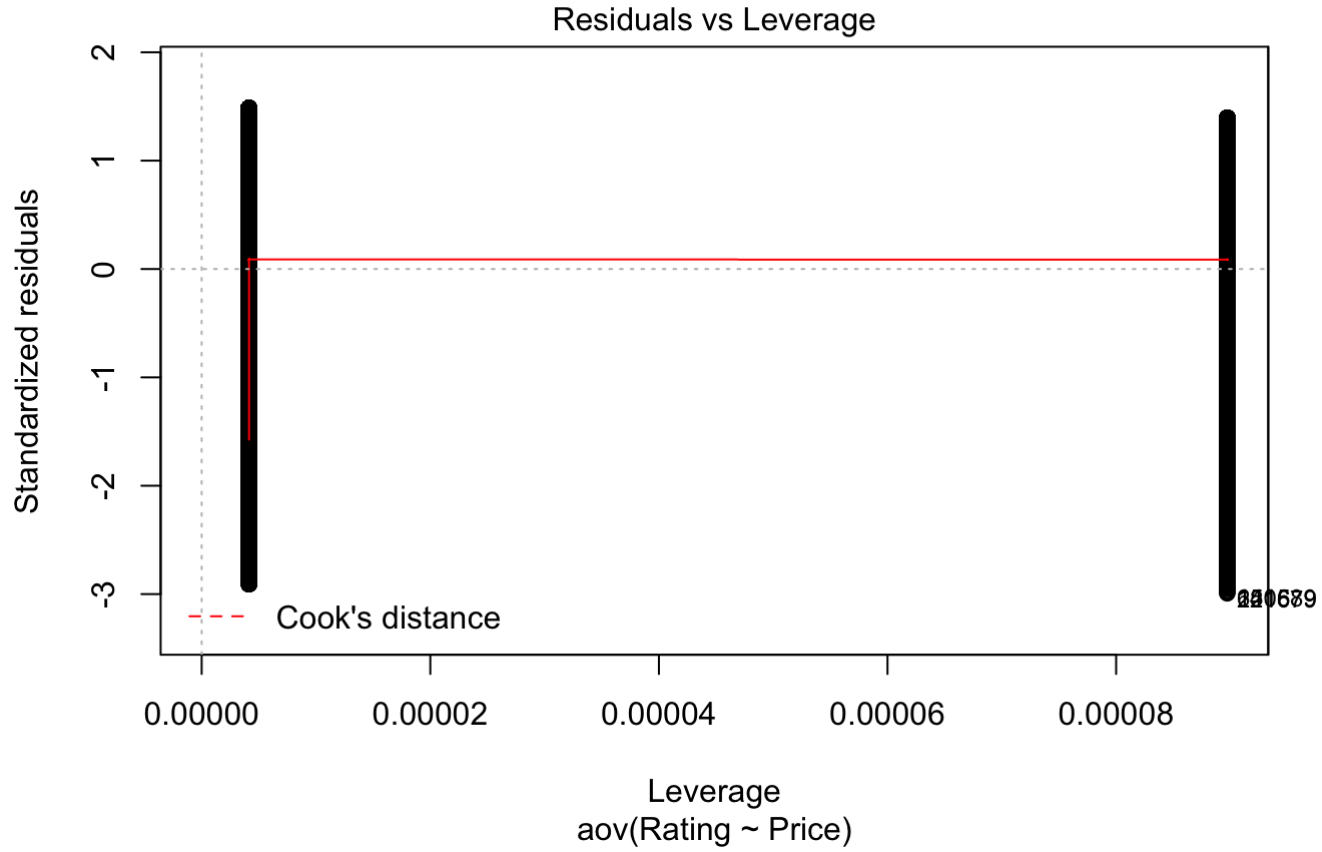
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Rating ~ Price, data = g_apps11)
##
## $Price
##                 diff        lwr        upr p adj
## Paid-Free 0.03953342 0.03119399 0.04787285     0
```

```
plot(rat_price.aov)
```


Residuals vs Fitted
Residuals
0.5
0.0
-0.5
-1.0
-1.5
4.35
4.36
4.37
4.38
Fitted values
aov(Rating ~ Price)


Normal Q-Q
Standardized residuals
2
1
0
-1
-2
-3
-4
-2
0
2
4
Theoretical Quantiles
aov(Rating ~ Price)


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
4.35
4.36
4.37
4.38
Fitted values
aov(Rating ~ Price)


Residuals vs Leverage
Standardized residuals
2
1
0
-1
-2
-3
Cook's distance
0.00000
0.00002
0.00004
0.00006
0.00008
Leverage
aov(Rating ~ Price)

I will

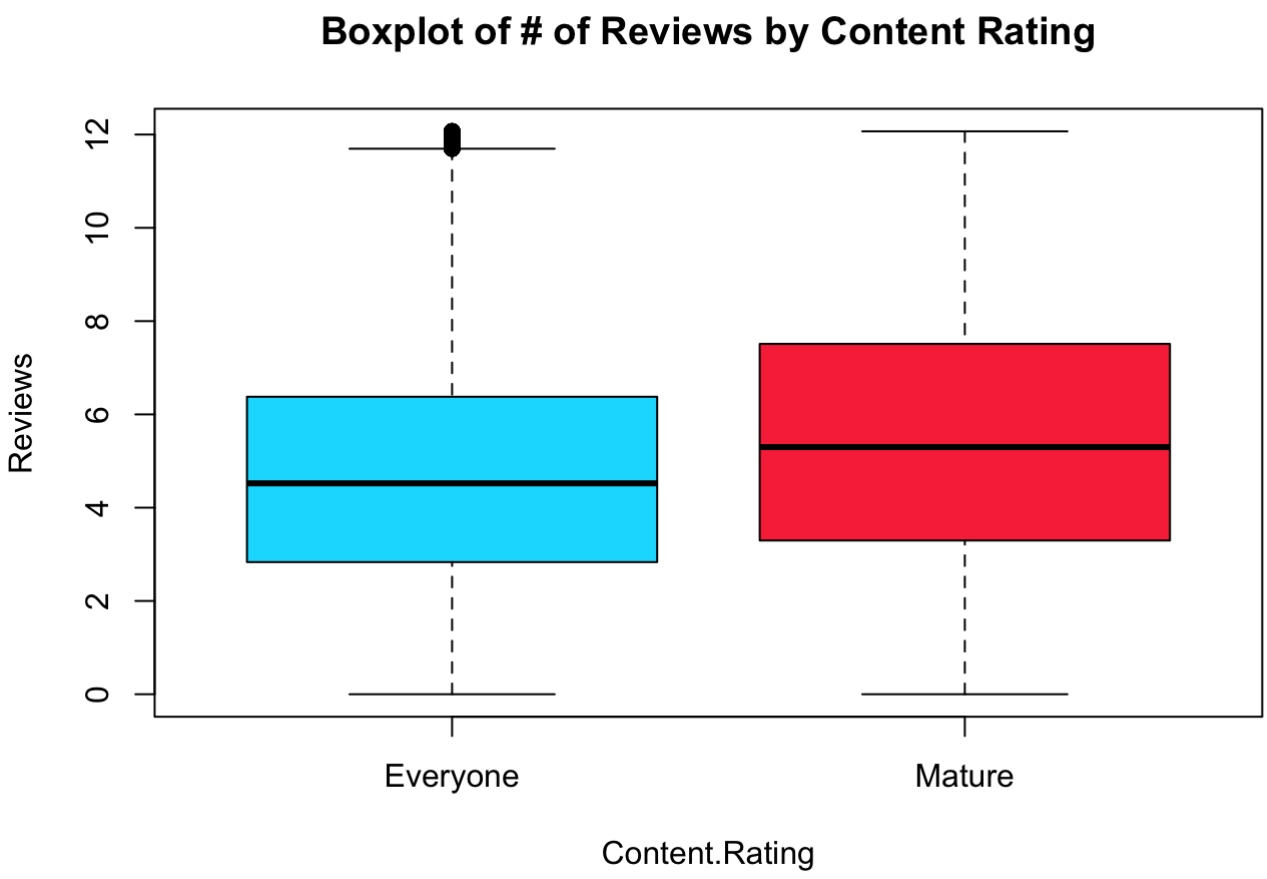
reject the null hypothesis for this test. There is a statistically significant differences between the ratings of free and paid apps. Paid apps actually have a higher rating on average than free apps. Diagnostics plots suggest that the ANOVA test results are valid.

```
# Hypothesis 4: Apps Rated Mature will have more reviews than apps for Everyone
g_apps12 %>% group_by(Content.Rating) %>% summarise(avg = mean(Reviews), median = median
(Reviews), std = sd(Reviews))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 4
##   Content.Rating   avg median   std
##   <fct>          <dbl>  <dbl> <dbl>
## 1 Everyone        4.70   4.52  2.54
## 2 Mature          5.48   5.30  2.87
```

```
boxplot(Reviews~Content.Rating, data = g_apps12, main = "Boxplot of # of Reviews by Cont
ent Rating",
        col=c("#00DCFF", "#F83648")) #Looks like mature apps get more reviews on average
```



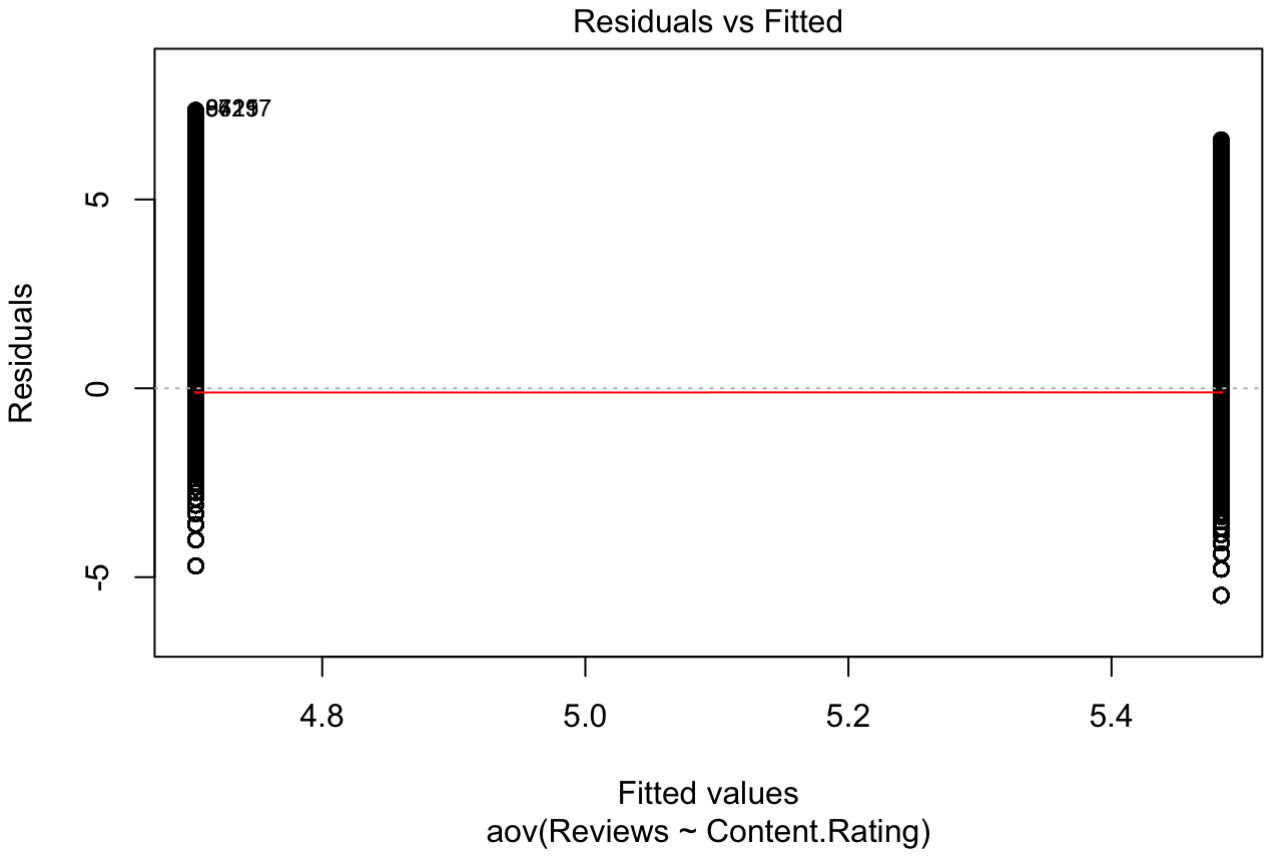

```
rev_content.aov <- aov(Reviews~Content.Rating, data=g_apps11)
summary(rev_content.aov)
```

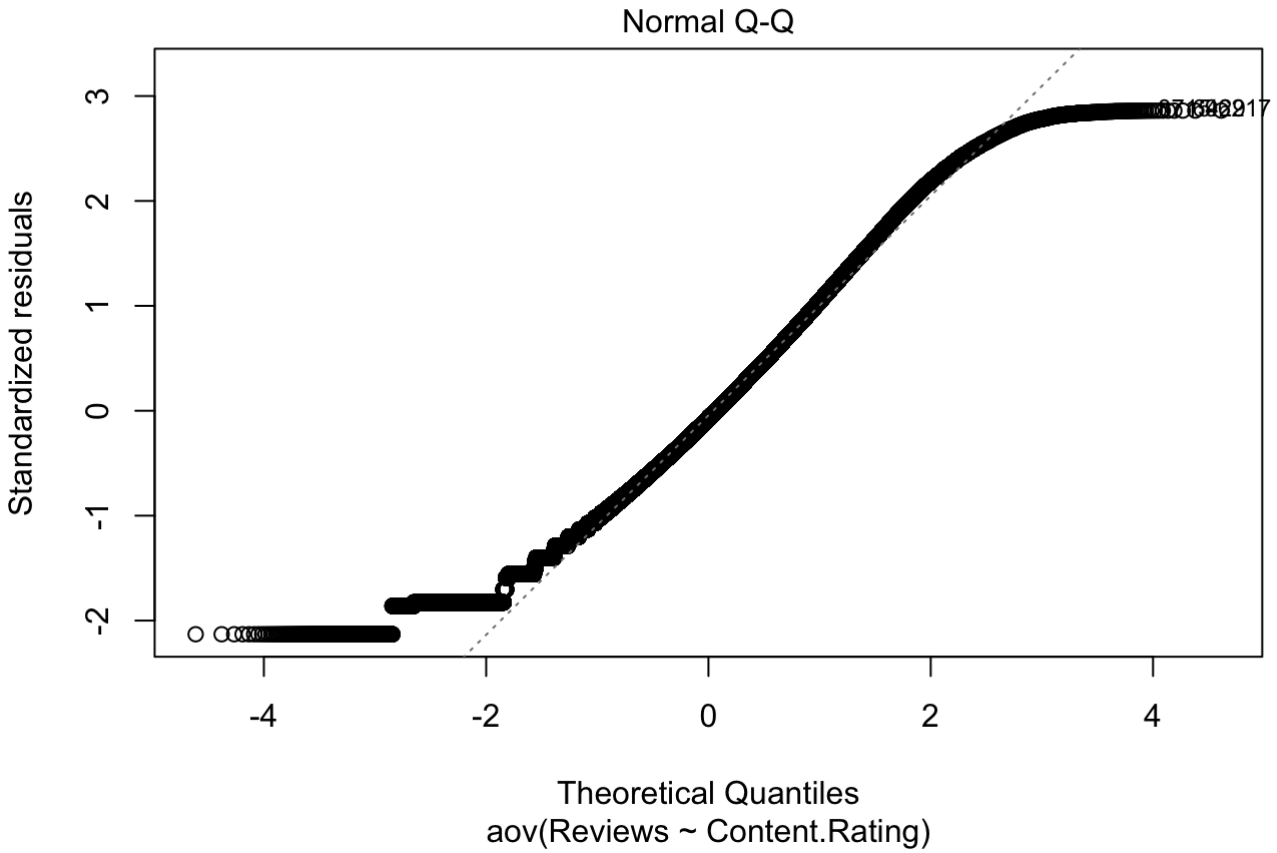
```
##                    Df  Sum Sq Mean Sq F value              Pr(>F)
## Content.Rating      1   13112   13112    1979 <0.0000000000000002 ***
## Residuals      252609 1673956       7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

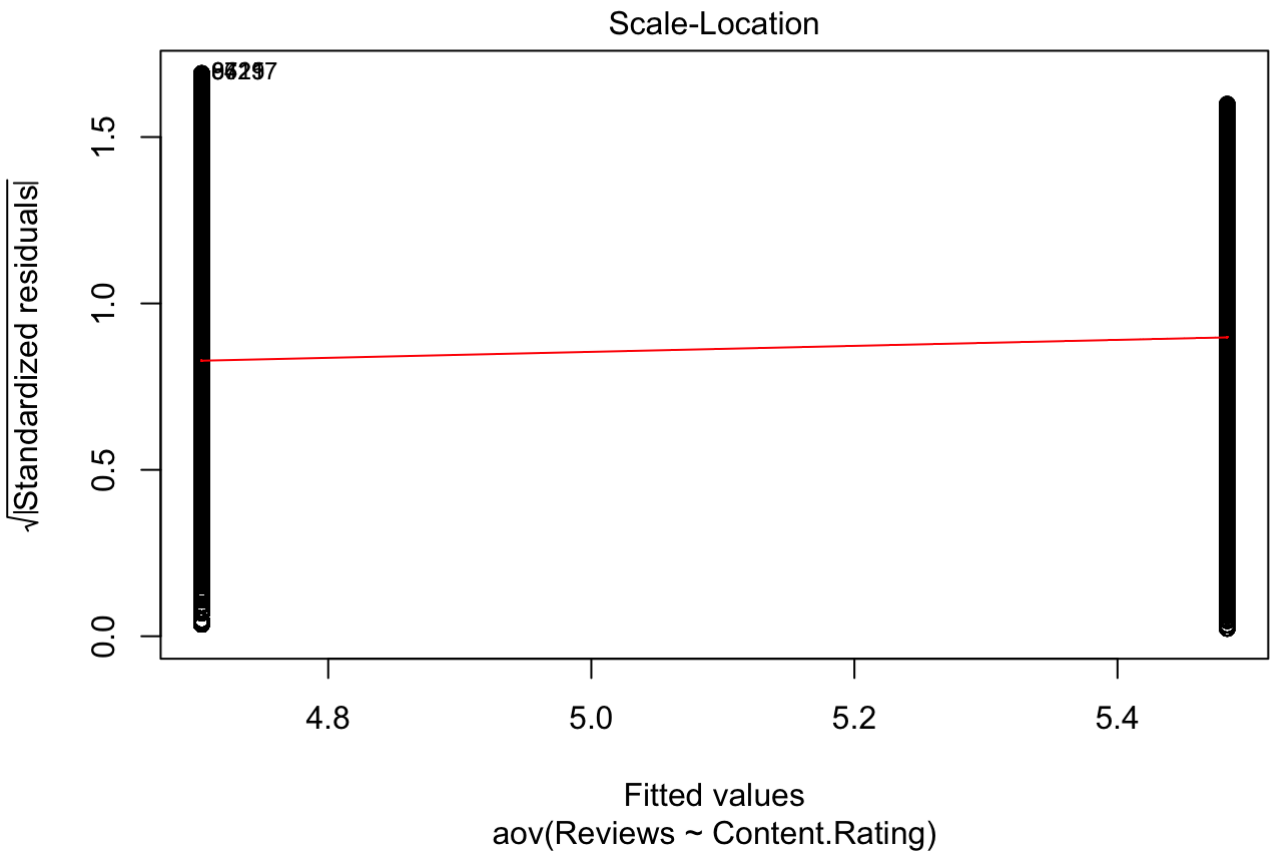
```
TukeyHSD(rev_content.aov)
```

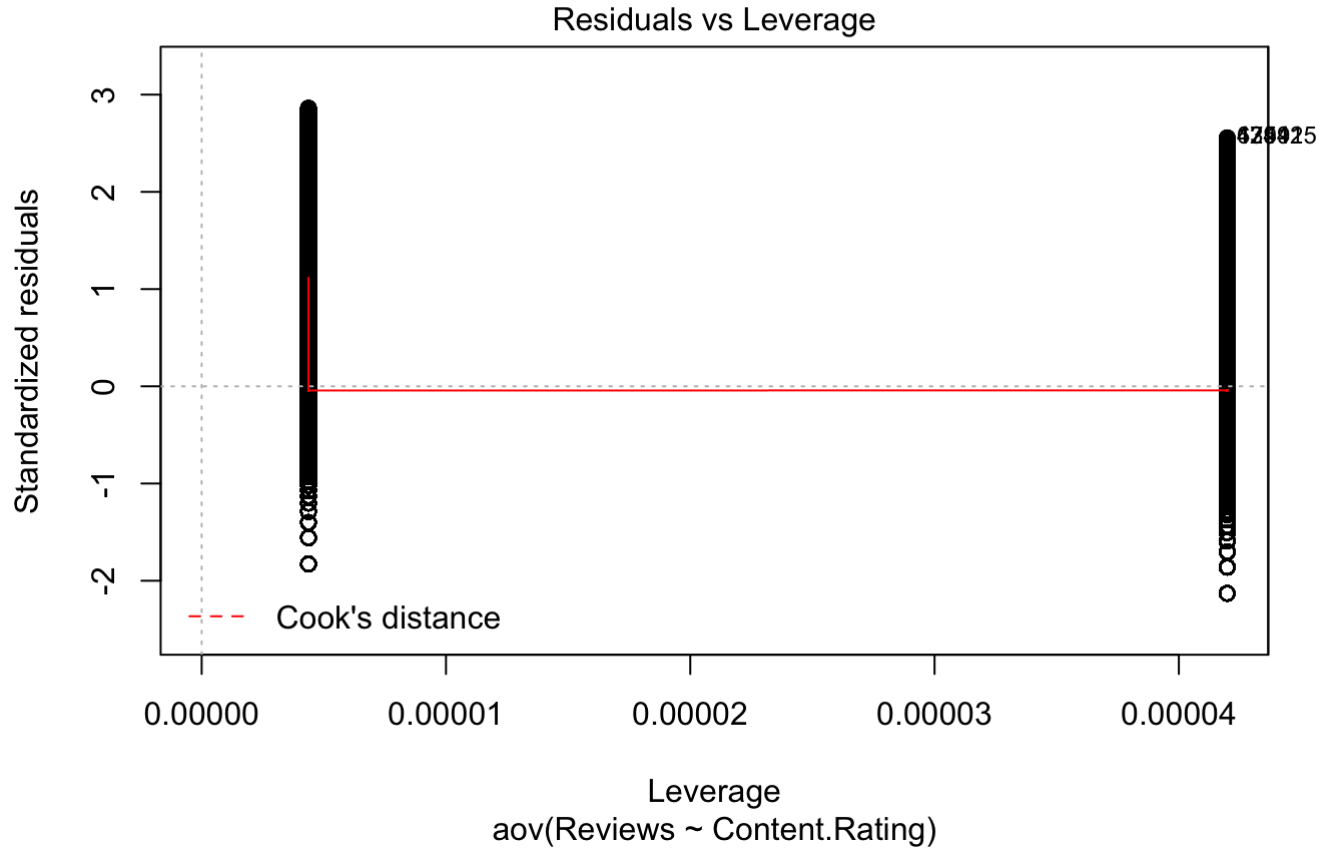
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Reviews ~ Content.Rating, data = g_apps11)
##
## $Content.Rating
##                      diff       lwr       upr p adj
## Mature-Everyone 0.779582 0.7452323 0.8139318     0
```

```
plot(rev_content.aov)
```


Residuals vs Fitted
Residuals
5
0
-5
4.8
5.0
5.2
5.4
Fitted values
aov(Reviews ~ Content.Rating)


Normal Q-Q
Standardized residuals
3
2
1
0
-1
-2
-4
-2
0
2
4
Theoretical Quantiles
aov(Reviews ~ Content.Rating)





I reject the null hypothesis for this test. Mature apps have higher reviews than apps for Everyone.

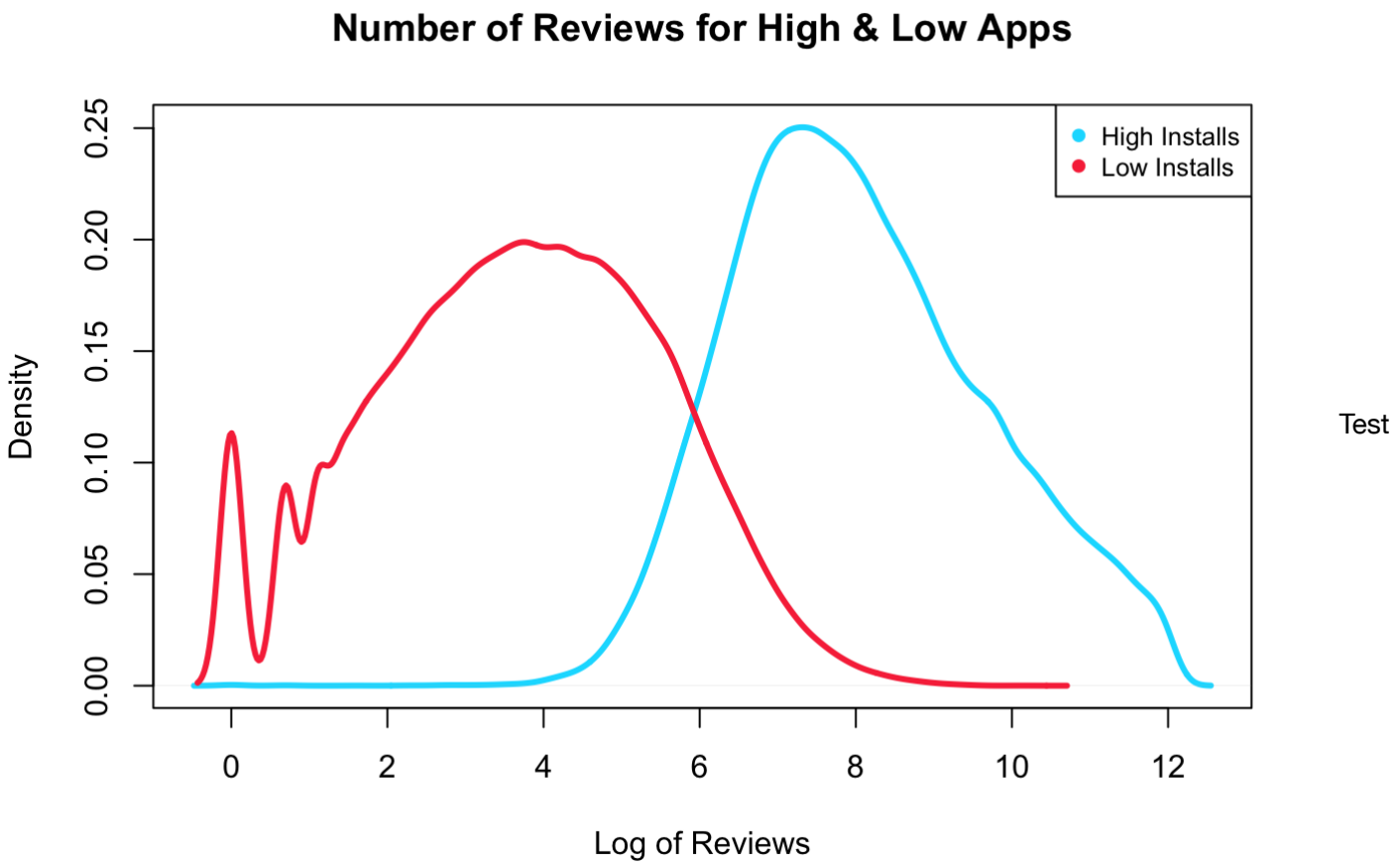
```
# Hypothesis 5: Apps with higher number of Installs will have a higher number of reviews

g_apps12$Installs <- as.character(g_apps12$Installs)
g_apps12$InstallCat <- rep(NA, nrow(g_apps12))
g_apps12$InstallCat[g_apps12$Installs %in% c("100000-1000000","1000000+")] <- "High"
g_apps12$InstallCat[g_apps12$Installs %in% c("10000-100000","1000-10000","0-500")] <- "L
ow"
g_apps12$InstallCat <- as.factor(g_apps12$InstallCat)
g_apps12$Installs <- as.factor(g_apps12$Installs)
summary(g_apps12$InstallCat)
```

```
##   High    Low
##  62708 189903
```

```
low_mean <- mean(g_apps12$Reviews[g_apps12$InstallCat == "Low"])

RI_Hypo <- t.test(g_apps12$Reviews[g_apps12$InstallCat=="High"],
                  alternative="greater",
                  mu=low_mean,
                  conf.level=0.95)
RI_Hypo
```

```
##
##  One Sample t-test
##
## data:  g_apps12$Reviews[g_apps12$InstallCat == "High"]
## t = 678.53, df = 62707, p-value < 0.00000000000000022
## alternative hypothesis: true mean is greater than 3.693875
## 95 percent confidence interval:
##  8.047836      Inf
## sample estimates:
## mean of x
##  8.058416
```

```
# Let's plot the difference
High_Installs <- g_apps12$Reviews[g_apps12$InstallCat =="High"]
Low_Installs <- g_apps12$Reviews[g_apps12$InstallCat == "Low"]
plot(density(High_Installs), main = "Number of Reviews for High & Low Apps", lwd= 3, col
= "#00DCFF", xlab = "Log of Reviews")
lines(density(Low_Installs), col="#F83648", lwd=3)
legend("topright", c("High Installs", "Low Installs"), col = c("#00DCFF", "#F83648"), pc
h = c(19,19), cex = 0.8)
```



Test

results show that we can reject the null hypothesis. Apps with higher installs recieve more reviews on average.

I will also use this created category for logistic regression to determine what variables are best for predicting whether or not an app will recieve a high or low amount of downloads.

Regression

```
options(scipen = 99)
g_apps13 <- g_apps12

g_mod0 <- glm(InstallCat~1, data = g_apps13, family = "binomial")
g_mod <- glm(InstallCat ~ Rating, data=g_apps13, family = "binomial")
summary(g_mod)$coef
```

```
##               Estimate Std. Error   z value      Pr(>|z|)
## (Intercept) -1.5498513 0.04449110 -34.83509 7.161995e-266
## Rating       0.6151776 0.01030832  59.67777  0.000000e+00
```

```
1-logLik(g_mod)/logLik(g_mod0)# Using a calculation for McFaddens R Squared to figure ou
t how much of the variance can be explained by this model.
```

```
## 'log Lik.' 0.01253825 (df=2)
```

I start with simple logistic regression to determine if ratings alone would be a good predictor for installs. The coefficient for rating shows a positive relationship between these two variables. As the rating of an app increases, the log likelihood that the number of Install will increase also increases. These results are statistically significant witha p-value less than 0.05.

By using McFadsdens R squared calculation for logistic regression, I see that the simple logistic regression model only explains about 1% of the variance in Installs. I will need a multiple logistic regression model to improve accuracy.

```
null <- glm(Installs ~ 1, data = g_apps13, family = "binomial") # first model
full <- glm(Installs ~ Category + Reviews + Rating + Size.Groups + Price + Content.Ratin
g, data= g_apps13, family = "binomial") # second model
step(null, scope=list(lower=null, upper=full), direction="forward") # forward selection
```

```
## Start:  AIC=213565.1
## Installs ~ 1
##
##                   Df Deviance    AIC
## + Reviews          1   100229 100233
## + Rating           1   188315 188319
## + Price            1   210157 210161
## + Size.Groups      4   211995 212005
## + Category         3   213197 213205
## + Content.Rating   1   213366 213370
## <none>                 213563 213565
##
## Step:  AIC=100233.4
## Installs ~ Reviews
##
##                   Df Deviance    AIC
## + Price            1    93444  93450
## + Rating           1    96644  96650
## + Category         3   100083 100093
## + Size.Groups      4   100208 100220
## <none>                 100229 100233
## + Content.Rating   1   100229 100235
##
## Step:  AIC=93450.38
## Installs ~ Reviews + Price
##
##                   Df Deviance   AIC
## + Rating           1    89612 89620
## + Category         3    93357 93369
## + Size.Groups      4    93392 93406
## <none>                  93444 93450
## + Content.Rating   1    93444 93452
##
## Step:  AIC=89619.95
## Installs ~ Reviews + Price + Rating
##
##                   Df Deviance   AIC
## + Category         3    89480 89494
## + Size.Groups      4    89576 89592
## <none>                  89612 89620
## + Content.Rating   1    89611 89621
##
## Step:  AIC=89493.79
## Installs ~ Reviews + Price + Rating + Category
##
##                   Df Deviance   AIC
## + Size.Groups      4    89444 89466
## + Content.Rating   1    89476 89492
## <none>                  89480 89494
##
## Step:  AIC=89465.86
## Installs ~ Reviews + Price + Rating + Category + Size.Groups
##
```

```
##                  Df Deviance   AIC
## + Content.Rating  1    89439 89463
## <none>                 89444 89466
##
## Step:  AIC=89463.23
## Installs ~ Reviews + Price + Rating + Category + Size.Groups +
##     Content.Rating
```

```
##
## Call:  glm(formula = Installs ~ Reviews + Price + Rating + Category +
##     Size.Groups + Content.Rating, family = "binomial", data = g_apps13)
##
## Coefficients:
##           (Intercept)                Reviews              PricePaid
##               2.57764                1.59061               -2.98005
##                Rating  CategoryENTERTAINMENT      CategoryLIFESTYLE
##              -1.09773               -0.21786               -0.17365
##  CategoryPRODUCTIVITY        Size.Groups0-4M     Size.Groups4M-8.3M
##              -0.24313               -0.25197               -0.34222
##   Size.Groups8.3M-19M    Size.Groups19M-334M   Content.RatingMature
##              -0.30781               -0.29209                0.06937
##
## Degrees of Freedom: 252610 Total (i.e. Null);  252599 Residual
## Null Deviance:       213600
## Residual Deviance: 89440     AIC: 89460
```

```
step(full, data=g_apps13, direction="backward") # backward selection
```

```
## Start:  AIC=89463.23
## Installs ~ Category + Reviews + Rating + Size.Groups + Price +
##     Content.Rating
##
##                  Df Deviance    AIC
## <none>                 89439  89463
## - Content.Rating  1    89444  89466
## - Size.Groups     4    89476  89492
## - Category        3    89574  89592
## - Rating          1    93305  93327
## - Price           1    96382  96404
## - Reviews         1   182831 182853
```

```
##
## Call:  glm(formula = Installs ~ Category + Reviews + Rating + Size.Groups +
##     Price + Content.Rating, family = "binomial", data = g_apps13)
##
## Coefficients:
##           (Intercept)  CategoryENTERTAINMENT      CategoryLIFESTYLE
##               2.57764               -0.21786               -0.17365
##  CategoryPRODUCTIVITY                Reviews                 Rating
##              -0.24313                1.59061               -1.09773
##       Size.Groups0-4M     Size.Groups4M-8.3M    Size.Groups8.3M-19M
##              -0.25197               -0.34222               -0.30781
##   Size.Groups19M-334M              PricePaid   Content.RatingMature
##              -0.29209               -2.98005                0.06937
##
## Degrees of Freedom: 252610 Total (i.e. Null);  252599 Residual
## Null Deviance:       213600
## Residual Deviance: 89440     AIC: 89460
```

```
step(null, scope = list(upper=full), data=g_apps13, direction="both") # Stepwise selecti
on
```

```
## Start:  AIC=213565.1
## Installs ~ 1
##
##                   Df Deviance    AIC
## + Reviews          1   100229 100233
## + Rating           1   188315 188319
## + Price            1   210157 210161
## + Size.Groups      4   211995 212005
## + Category         3   213197 213205
## + Content.Rating   1   213366 213370
## <none>                 213563 213565
##
## Step:  AIC=100233.4
## Installs ~ Reviews
##
##                   Df Deviance    AIC
## + Price            1    93444  93450
## + Rating           1    96644  96650
## + Category         3   100083 100093
## + Size.Groups      4   100208 100220
## <none>                 100229 100233
## + Content.Rating   1   100229 100235
## - Reviews          1   213563 213565
##
## Step:  AIC=93450.38
## Installs ~ Reviews + Price
##
##                   Df Deviance    AIC
## + Rating           1    89612  89620
## + Category         3    93357  93369
## + Size.Groups      4    93392  93406
## <none>                 93444  93450
## + Content.Rating   1    93444  93452
## - Price            1   100229 100233
## - Reviews          1   210157 210161
##
## Step:  AIC=89619.95
## Installs ~ Reviews + Price + Rating
##
##                   Df Deviance    AIC
## + Category         3    89480  89494
## + Size.Groups      4    89576  89592
## <none>                 89612  89620
## + Content.Rating   1    89611  89621
## - Rating           1    93444  93450
## - Price            1    96644  96650
## - Reviews          1   184785 184791
##
## Step:  AIC=89493.79
## Installs ~ Reviews + Price + Rating + Category
##
##                   Df Deviance    AIC
## + Size.Groups      4    89444  89466
```

```
## + Content.Rating  1    89476  89492
## <none>                 89480  89494
## - Category        3    89612  89620
## - Rating          1    93357  93369
## - Price           1    96414  96426
## - Reviews         1   184493 184505
##
## Step:  AIC=89465.86
## Installs ~ Reviews + Price + Rating + Category + Size.Groups
##
##                  Df Deviance    AIC
## + Content.Rating  1    89439  89463
## <none>                 89444  89466
## - Size.Groups     4    89480  89494
## - Category        3    89576  89592
## - Rating          1    93307  93327
## - Price           1    96387  96407
## - Reviews         1   183080 183100
##
## Step:  AIC=89463.23
## Installs ~ Reviews + Price + Rating + Category + Size.Groups +
##     Content.Rating
##
##                  Df Deviance    AIC
## <none>                 89439  89463
## - Content.Rating  1    89444  89466
## - Size.Groups     4    89476  89492
## - Category        3    89574  89592
## - Rating          1    93305  93327
## - Price           1    96382  96404
## - Reviews         1   182831 182853
```

```
##
## Call:  glm(formula = Installs ~ Reviews + Price + Rating + Category +
##     Size.Groups + Content.Rating, family = "binomial", data = g_apps13)
##
## Coefficients:
##           (Intercept)                Reviews              PricePaid
##               2.57764                1.59061               -2.98005
##                Rating  CategoryENTERTAINMENT      CategoryLIFESTYLE
##              -1.09773               -0.21786               -0.17365
##  CategoryPRODUCTIVITY        Size.Groups0-4M     Size.Groups4M-8.3M
##              -0.24313               -0.25197               -0.34222
##   Size.Groups8.3M-19M    Size.Groups19M-334M   Content.RatingMature
##              -0.30781               -0.29209                0.06937
##
## Degrees of Freedom: 252610 Total (i.e. Null);  252599 Residual
## Null Deviance:       213600
## Residual Deviance: 89440     AIC: 89460
```

```
1-logLik(full)/logLik(null) # Using a calculation for McFaddens R Squared to figure out
 how much of the variance can be explained by this model.
```

```
## 'log Lik.' 0.5812048 (df=12)
```

I used forward selection, backwards elimination and stepwise regression to find the best model for Installs prediction. Each method returned the same model with the same AIC value of 89460. The final model below uses all relevant variables in the dataset however, McFaddens R squared calculation shows that it only explains about 58% of the variance in the Install variable.

```
final <- glm(formula = Installs ~ Reviews + Price + Rating + Category +
    Size.Groups + Content.Rating, family = "binomial", data = g_apps13)
summary(final)
```

```
##
## Call:
## glm(formula = Installs ~ Reviews + Price + Rating + Category +
##     Size.Groups + Content.Rating, family = "binomial", data = g_apps13)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.4283   0.0069   0.0597   0.2439   3.5785
##
## Coefficients:
##                          Estimate Std. Error z value             Pr(>|z|)
## (Intercept)              2.577643   0.107844  23.902 < 0.0000000000000002
## Reviews                  1.590614   0.008831 180.120 < 0.0000000000000002
## PricePaid               -2.980053   0.036327 -82.034 < 0.0000000000000002
## Rating                  -1.097728   0.018374 -59.745 < 0.0000000000000002
## CategoryENTERTAINMENT -0.217857   0.023541  -9.254 < 0.0000000000000002
## CategoryLIFESTYLE     -0.173653   0.023423  -7.414    0.000000000000123
## CategoryPRODUCTIVITY  -0.243129   0.024475  -9.934 < 0.0000000000000002
## Size.Groups0-4M       -0.251971   0.066287  -3.801             0.000144
## Size.Groups4M-8.3M    -0.342221   0.066499  -5.146    0.000000265756169
## Size.Groups8.3M-19M   -0.307805   0.066705  -4.614    0.000003941567053
## Size.Groups19M-334M   -0.292086   0.066933  -4.364    0.000012779868712
## Content.RatingMature   0.069374   0.032315   2.147             0.031811
##
## (Intercept)           ***
## Reviews               ***
## PricePaid             ***
## Rating                ***
## CategoryENTERTAINMENT ***
## CategoryLIFESTYLE     ***
## CategoryPRODUCTIVITY  ***
## Size.Groups0-4M       ***
## Size.Groups4M-8.3M    ***
## Size.Groups8.3M-19M   ***
## Size.Groups19M-334M   ***
## Content.RatingMature  *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 213563  on 252610  degrees of freedom
## Residual deviance:  89439  on 252599  degrees of freedom
## AIC: 89463
##
## Number of Fisher Scoring iterations: 8
```

As we can see in the final model, all variables included in the model are statistically significant. Test data can be introduced to make predictions and test the accuracy of the model but that is beyond the scope of this project. As of right now the McFaddens R squared calcuation shows that the model explains 58% of the variance in Installs. I could also change the probability threshold to be greater than 0.5 and see if that improves the model accuracy.

In sum, we see that all models used in this dataset were important in predicting the amount of install an app would receive. More data on these apps could be included in the data to balance out some of the disproportionate variable and make a more accurate prediction. App developers could also benefit from more data on metrics that explain consumers are using these apps once they have been installed. Further analysis is needed but the analysis in this report is a great start.