SemesterArbeit

Sayyed Ahmad Hosseini, Astritt Zyberaj

15/11/2021

Semester Arbeit

Description of dataset

Context

While many public datasets (on Kaggle and the like) provide Apple App Store data, there are not many counterpart datasets available for Google Play Store apps anywhere on the web. It is because, ITunes App Store page deploys a nicely indexed appendix-like structure to allow for simple and easy web scraping. On the other hand, Google Play Store uses sophisticated modern-day techniques (like dynamic page load) using JQuery making scraping more challenging. Content

Each app (row) has values for catergory, rating, size, and more.

Acknowledgements

All information is scraped from the Google Play Store. I downloaded this dataset from Kaggle.com

Inspiration

The Play Store apps data has enormous potential to drive app-making businesses to success. Actionable insights can be drawn for developers to work on and capture the Android market!

---

loading the dataset

Summary of infos for each column:

- Are they numeric?
- If numeric, what is min, max, and quartiles
- Are there any NAs?

```
googleplaystore <- read.csv("~/Studium/Data-Science-General/HS2021/EXPD/Projects/googleplaystore/googlep
playstore<-as.data.frame(googleplaystore)
summary(playstore)
```

```
##      App              Category              Rating         Reviews
##  Length:10841       Length:10841       Min.   : 1.000   Length:10841
##  Class :character   Class :character   1st Qu.: 4.000   Class :character
##  Mode  :character   Mode  :character   Median : 4.300   Mode  :character
##                                        Mean   : 4.193
##                                        3rd Qu.: 4.500
##                                        Max.   :19.000
##                                        NA’s   :1474
##      Size             Installs              Type               Price
##  Length:10841       Length:10841       Length:10841       Length:10841
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Content.Rating        Genres            Last.Updated       Current.Ver
##  Length:10841       Length:10841       Length:10841       Length:10841
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Android.Ver
##  Length:10841
##  Class :character
##  Mode  :character
##
##
##
##
```

The first 10 lines of dataset:

```
head(playstore,10)
```

```
##                                                     App       Category Rating
## 1        Photo Editor & Candy Camera & Grid & ScrapBook ART_AND_DESIGN    4.1
## 2                                   Coloring book moana ART_AND_DESIGN    3.9
## 3  U Launcher Lite â\200" FREE Live Cool Themes, Hide Apps ART_AND_DESIGN    4.7
## 4                                 Sketch - Draw & Paint ART_AND_DESIGN    4.5
## 5                 Pixel Draw - Number Art Coloring Book ART_AND_DESIGN    4.3
## 6                            Paper flowers instructions ART_AND_DESIGN    4.4
## 7               Smoke Effect Photo Maker - Smoke Editor ART_AND_DESIGN    3.8
## 8                                      Infinite Painter ART_AND_DESIGN    4.1
## 9                                  Garden Coloring Book ART_AND_DESIGN    4.4
## 10                        Kids Paint Free - Drawing Fun ART_AND_DESIGN    4.7
##    Reviews Size    Installs Type Price Content.Rating                    Genres
## 1      159  19M     10,000+ Free     0       Everyone              Art & Design
## 2      967  14M    500,000+ Free     0       Everyone Art & Design;Pretend Play
## 3    87510 8.7M  5,000,000+ Free     0       Everyone              Art & Design
## 4   215644  25M 50,000,000+ Free     0           Teen              Art & Design
## 5      967 2.8M    100,000+ Free     0       Everyone   Art & Design;Creativity
## 6      167 5.6M     50,000+ Free     0       Everyone              Art & Design
## 7      178  19M     50,000+ Free     0       Everyone              Art & Design
## 8    36815  29M  1,000,000+ Free     0       Everyone              Art & Design
## 9    13791  33M  1,000,000+ Free     0       Everyone              Art & Design
## 10     121 3.1M     10,000+ Free     0       Everyone   Art & Design;Creativity
##           Last.Updated        Current.Ver  Android.Ver
## 1     January 7, 2018              1.0.0 4.0.3 and up
## 2    January 15, 2018              2.0.0 4.0.3 and up
## 3      August 1, 2018              1.2.4 4.0.3 and up
## 4        June 8, 2018 Varies with device   4.2 and up
## 5       June 20, 2018                1.1   4.4 and up
## 6      March 26, 2017                1.0   2.3 and up
## 7      April 26, 2018                1.1 4.0.3 and up
## 8       June 14, 2018           6.1.61.1   4.2 and up
## 9  September 20, 2017              2.9.2   3.0 and up
## 10       July 3, 2018                2.8 4.0.3 and up
```

clean up of the dataset:

Cleanups:

- removing line 10473 because it has a rating of 19. And it is most likey false data
- making reviews, size and price vectors to numeric vectors

```r
#to find the NAs after getting:"NAs introduced by coercion"
#I used: which(is.na(x)) after applyin function to find
#what are the abnormalities
#the first abnormality is line 10473. I think it is faulty:
#rating is by 19 and price is everyone and size is 1,000+
#so we delete this line:
playstore[10473,]
playstore<- playstore[-10473,]


#cleaning the Reviews:
str(playstore$Reviews)
playstore$Reviews<-as.numeric(playstore$Reviews)


#cleaning the size:
str(playstore$Size)


#To replace 1.5k with 1500
#we need the following library to do all these:
#install.packages("stringr")
library("stringr")
nonDecimalVec<-stringr::str_extract(string = playstore$Size,pattern ="\\.([0-9])*")
#replace NAs with empty string , so we have an easier
#job, when we use paste function later
nonDecimalVec[is.na(nonDecimalVec)]<-""
```

```
playstore$Size<-sub(pattern = "\\.[0-9]*k","000",playstore$Size)
playstore$Size<-sub(pattern = "\\.[0-9]*M","000000",playstore$Size)
#finally: adding the nachkommastellen back to the number
#if they had any example: 1.05k= (1+0.05)*1000:
vsel<-nonDecimalVec!=""
temp_size<-(as.numeric(playstore$Size[vsel]))
temp_size2<-rep(1,sum(vsel))
temp_size3<-as.numeric(nonDecimalVec[vsel])
temp_size4<-temp_size2+temp_size3
temp_size<-temp_size4*temp_size
playstore$Size[vsel]<-temp_size

#to replace the likes of 10k with 1000 or m with 1000000:
playstore$Size<-sub(pattern = "k","000",playstore$Size)
playstore$Size<-sub(pattern = "M","000000",playstore$Size)
#size also contains the string: "Varies with device"
#So we should be carful about that!
playstore$Size[playstore$Size=="Varies with device"]<-NA
playstore$Size<-as.numeric(playstore$Size)
# coercedNas<-which(is.na(playstore$Size))
# coercedNas





#cleaning the price:
str(playstore$Price)
playstore$Price<-sub(pattern = "\\$", replacement = "", x = playstore$Price)
playstore$Price<-as.numeric(playstore$Price)
#The following two lines helped me with debugging and cleaning
# coercedNas<-which(is.na(playstore$Price))
# coercedNas





#I removed the following line because we don't want to
#lose information, forexmpale if an app doesn't have
#ratings, it could be that it was downloaded very
#little.
#playstore<-playstore[complete.cases(playstore), ]
```

Introduction of new Kategorial Variables

```
reviewCut<-cut(playstore$Reviews,breaks = c(0,1000,10000,100000,80000000))
playstore$reviewCut<-factor(reviewCut,levels = levels(reviewCut),labels = c("0+","1000+","10k+","100k+"
playstore$reviewSuperCut<-factor(reviewCut,levels = levels(reviewCut),labels = c("0+","1000+","10k+","1

priceCut<-cut(playstore$Price,breaks = c(0,10,30,500))
playstore$priceCut<-factor(priceCut,levels = levels(priceCut),labels = c("0+","10+","30+"),ordered = T)
playstore$priceSuperCut<-factor(priceCut,levels = levels(priceCut),labels = c("0+","10+","10+"),ordered

installsfac<-factor(playstore$Installs, labels =c("0","0+","1+","5+","10+","50+","100+","500+","1,000+"
playstore$InstallsFac<-installsfac

#labels(installsfac)<-c("0","0+","1+","5+","10+","50+","100+","500+","1,000+","5,000+","10,000+","50,00
#playstore$InstallsFac<-installsfac

#It might make sense to convert this to numeric
#to calculate the mean. This is an ordnial category
#but the difference from categroy to other is not the
#same between each pair of categories:
#(500-100)=400 but (100-50)=50
Installs_<-sub(pattern = "\\+", replacement = "", x = playstore$Installs)
Installs_<-gsub(pattern = ",", replacement = "", x = Installs_)
Installs_<-as.numeric(Installs_)

installsCut<-cut(Installs_,breaks = c(0,1e+5,5e+05,1e+06,5e+06,1e+07,5e+07,1e+08,5e+08,1e+09))
insatllsCut<-factor(installsCut,levels = levels(installsCut),ordered = T)
levels(installsCut)<-c("0+","100k+","500k+","1m+","5m+","10m+","50m+","100m+","500m+")
playstore$installsCut<-insatllsCut
installsCut<-cut(Installs_,breaks = c(0,1e+2,1e+03,1e+04,1e+05,5e+05,5e+07,1e+08,5e+08,1e+09))
playstore$installsSuperCut<-factor(installsCut,levels = levels(installsCut),ordered = T)
levels(installsCut)<-c("0+","100+","1k+","10k+","50k+","50k+","50k+","50k+","50k+")

ratingCut<-cut(playstore$Rating,breaks = c(0.99,1.99,2.99,3.49,3.99,5))
ratingCut<-factor(ratingCut,levels = levels(ratingCut),labels = c("1+","2+","3+","3.5+","4+"),ordered =
playstore$ratingCut<-ratingCut

playstore$categoryCut<-factor(playstore$Category,levels =names(sort(table(playstore$Category),decreasin
tmp<-head(levels(playstore$categoryCut),10)
playstore$categoryCut<-factor(playstore$categoryCut,labels = append(tmp,rep("other",23),after = length(
```

Interesting numbers and tables:

Means of 4 Variables:

```
mean_table<-colMeans(playstore[,c("Rating","Reviews","Size","Price")],na.rm = T)
mean_table
```

```
##       Rating      Reviews         Size        Price
## 4.191757e+00 4.441529e+05 2.211759e+07 1.027368e+00
```

mean of Installs variable:

```
Installs_mean<-mean(Installs_)
Installs_mean
```

```
## [1] 15464339
```

Quntiles of Reviews

```
quantile(playstore$Reviews,probs = c(0.25,0.5,0.75),type=7)
```

```
##     25%     50%     75%
##    38.0  2094.0 54775.5
```

Applications with most number of Reviews:

```
##                                                      App  Reviews
## 4296                                          Facebook 78143257
## 9377                                WhatsApp Messenger 69116101
## 5611                                         Instagram 66560497
## 6394         Messenger â\200" Text and Video Chat for Free 56644091
## 2461                                    Clash of Clans 44889695
## 2471           Clean Master- Space Cleaner & Antivirus 42916526
## 8387                                    Subway Surfers 27721993
## 9582                                           YouTube 25639427
## 7993 Security Master - Antivirus, VPN, AppLock, Booster 24900999
## 2463                                      Clash Royale 23132575
## 2050                                  Candy Crush Saga 22427591
## 9009       UC Browser - Fast Download Private & Secure 17713565
## 8183                                          Snapchat 17011253
## 67      360 Security - Free Antivirus, Booster, Cleaner 16771865
## 6711                                    My Talking Tom 14889643
## 112                                        8 Ball Pool 14198028
## 3528  DU Battery Saver - Battery Charger & Battery Life 13479633
## 1136                       BBM - Free Calls & Messages 12843148
## 1971 Cache Cleaner-DU Speed Booster (booster & cleaner) 12759739
## 8979                                           Twitter 11664259
```

Application with most number of installs:

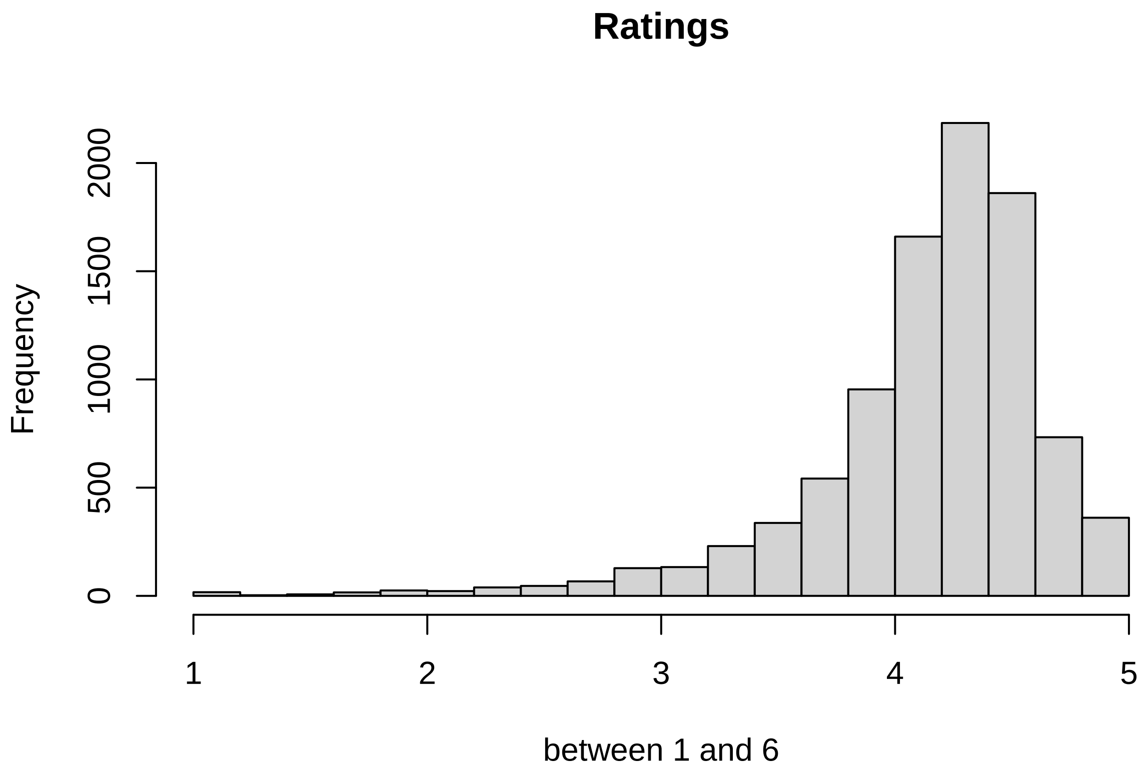
```
grouped_by_app<-aggregate(Installs_~App,data=playstore,FUN = mean)
vsel<-head(order(grouped_by_app$Installs_,decreasing=T),20)
grouped_by_app[vsel,]
```

```
##                                              App Installs_
## 4296                                   Facebook     1e+09
## 5025                                      Gmail     1e+09
## 5083                                     Google     1e+09
## 5093               Google Chrome: Fast & Secure     1e+09
## 5096                               Google Drive     1e+09
## 5105                                Google News     1e+09
## 5109                              Google Photos     1e+09
## 5110                          Google Play Books     1e+09
## 5111                          Google Play Games     1e+09
## 5112                    Google Play Movies & TV     1e+09
## 5116                         Google Street View     1e+09
## 5120                                    Google+     1e+09
## 5238                                   Hangouts     1e+09
## 5611                                  Instagram     1e+09
## 6264                  Maps - Navigate & Explore     1e+09
## 6394 Messenger â\200" Text and Video Chat for Free     1e+09
## 8130              Skype - free IM & video calls     1e+09
## 8387                             Subway Surfers     1e+09
## 9377                         WhatsApp Messenger     1e+09
## 9582                                    YouTube     1e+09
```

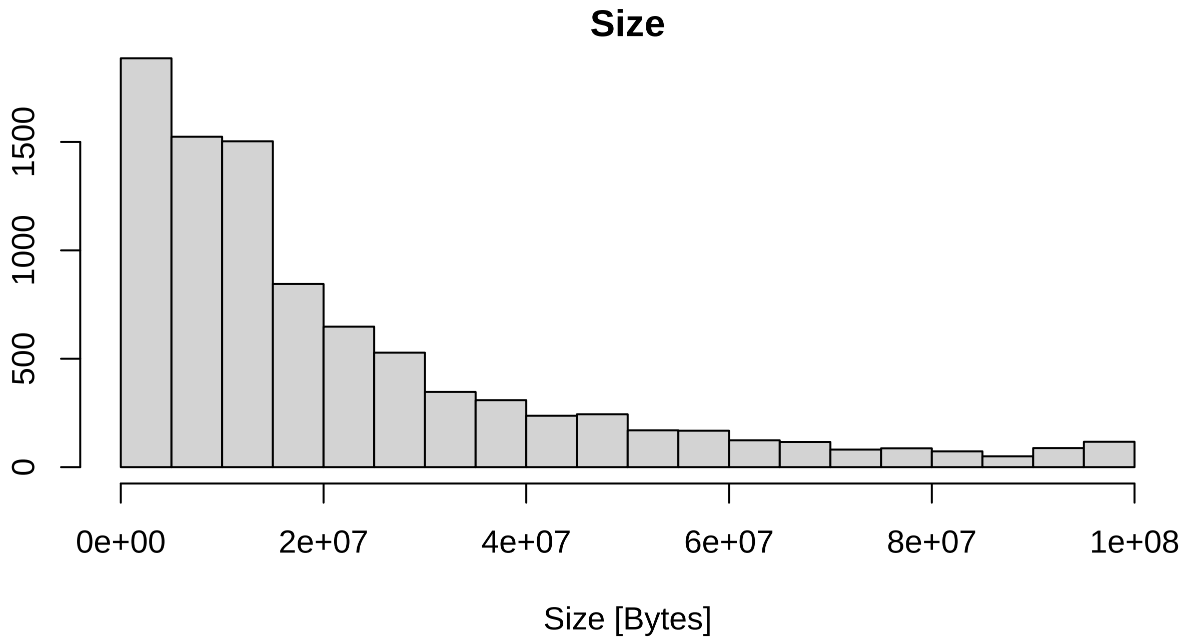
some univariate plots

Below we see a Distribution of Ratings and Size.

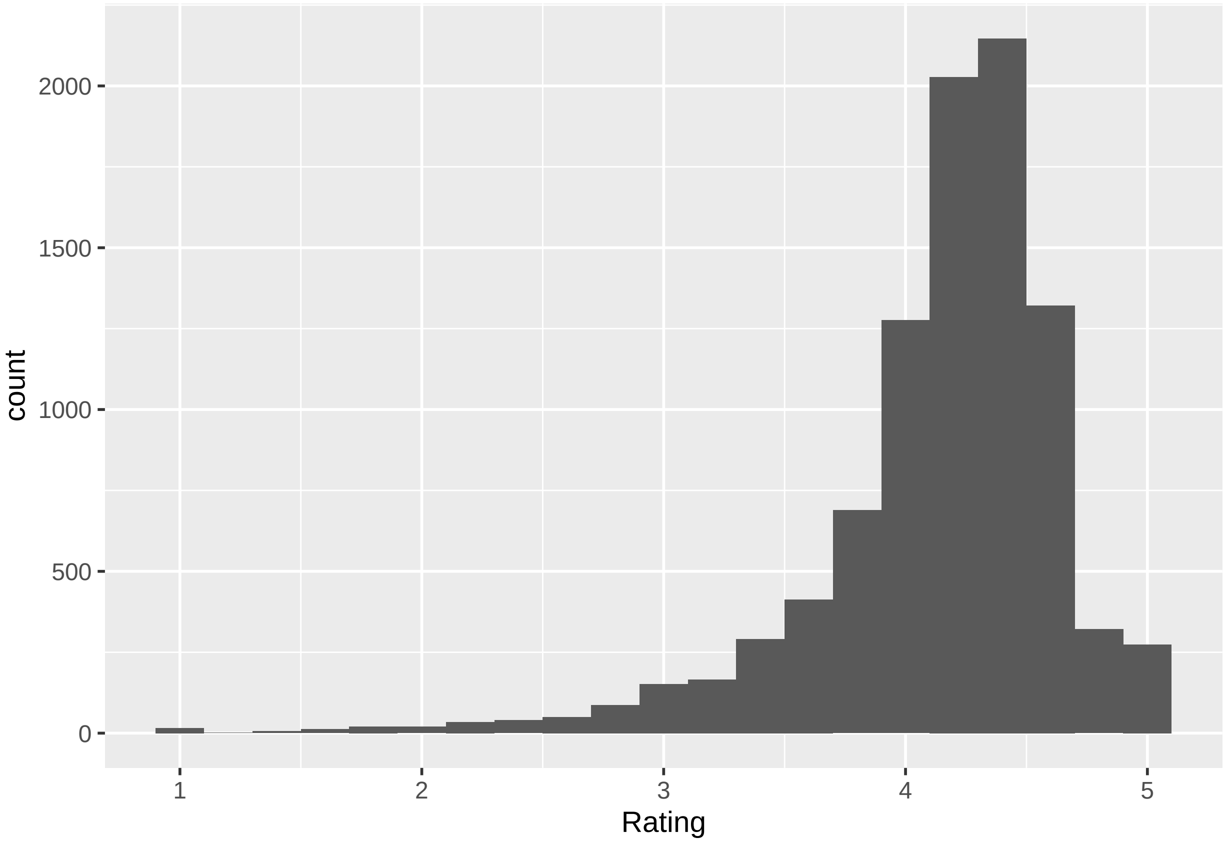
```
library(ggplot2)
hist((playstore$Rating[!is.na(playstore$Rating)]),xlab = "between 1 and 6",main = "Ratings",breaks = 20
```



```
par(mar=c(10,3,1,1))
hist(playstore$Size,xlab = "Size [Bytes]",main = "Size")
```



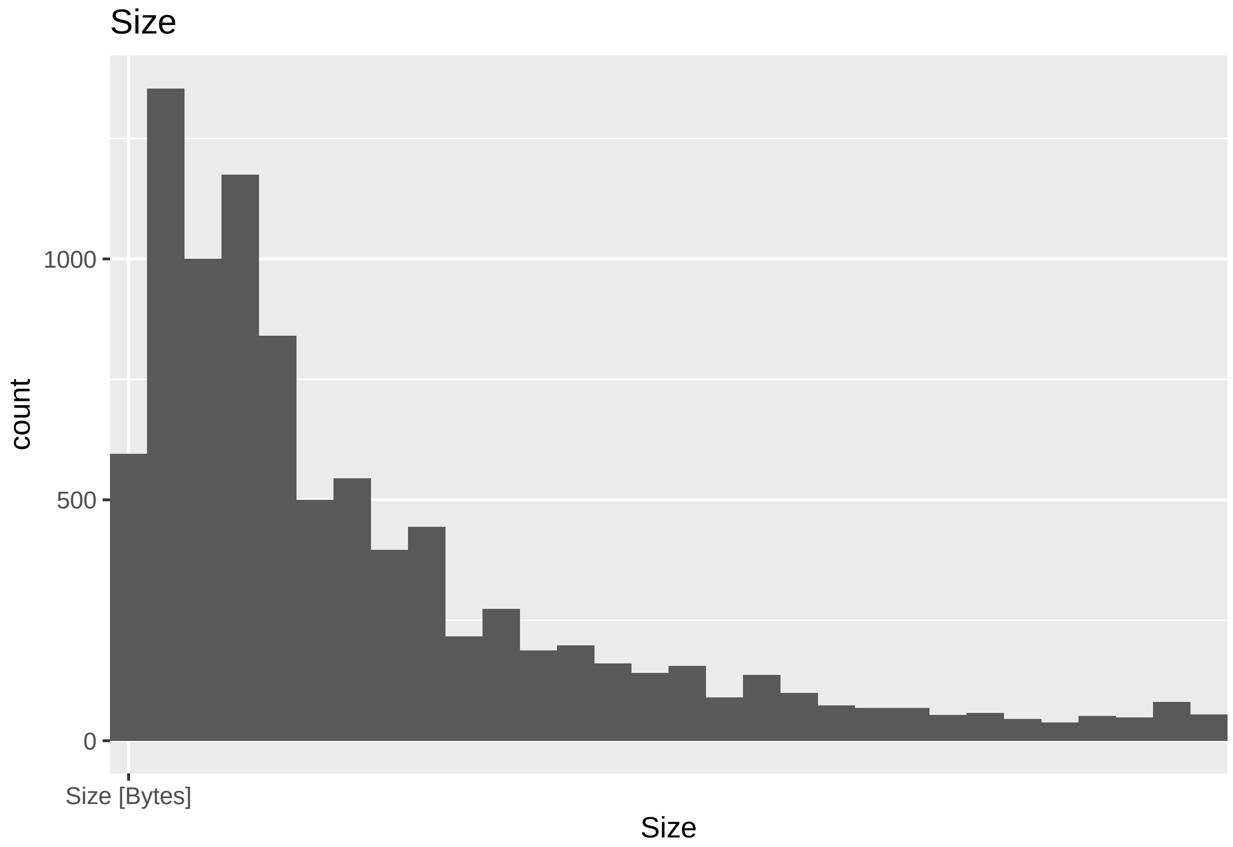
```r
#ggplot versions;
ggplot(data = playstore[!is.na(playstore$Rating),],aes(x=Rating))+
  geom_histogram(binwidth = 0.2)
```



```
ggplot(data = playstore,aes(x=Size))+geom_histogram()+ggtitle("Size")+xlim("Size [Bytes]")
```

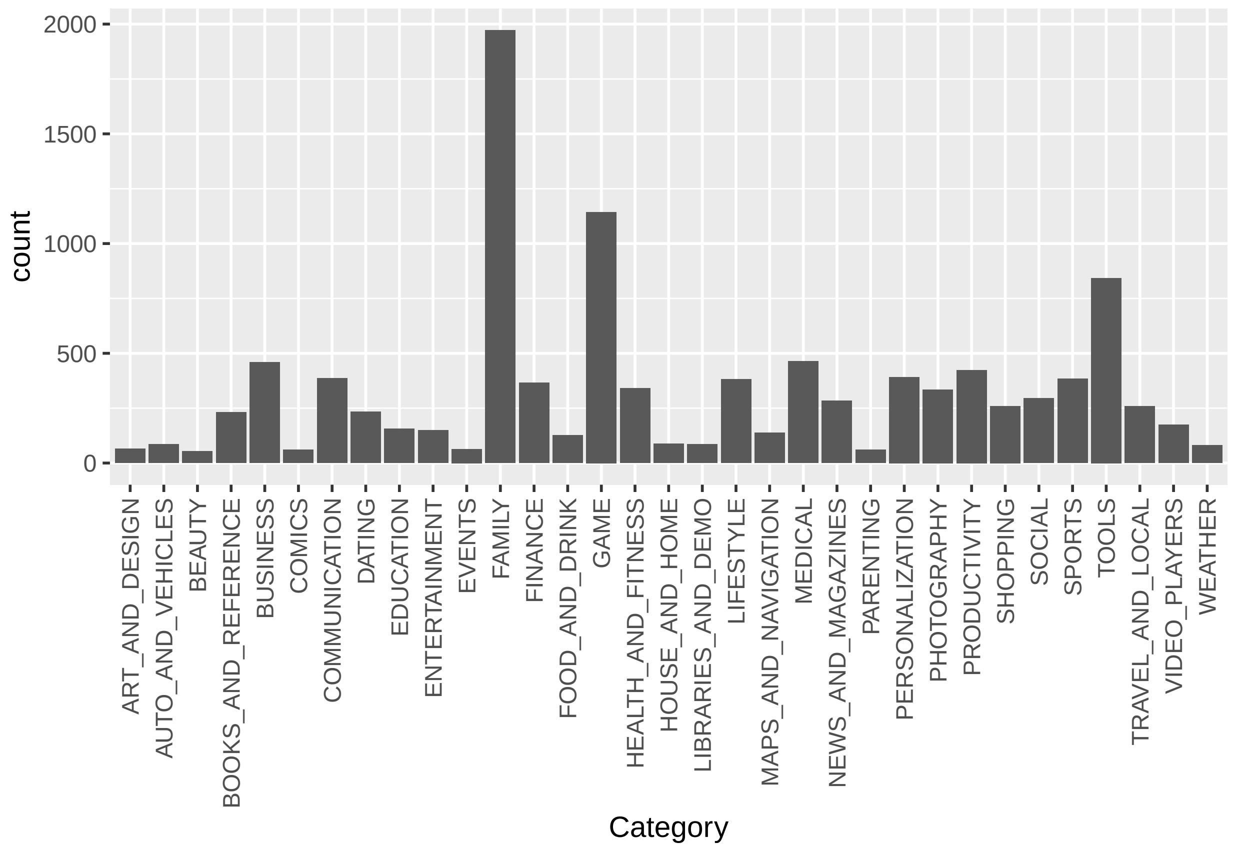
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 1695 rows containing non-finite values (stat_bin).
```



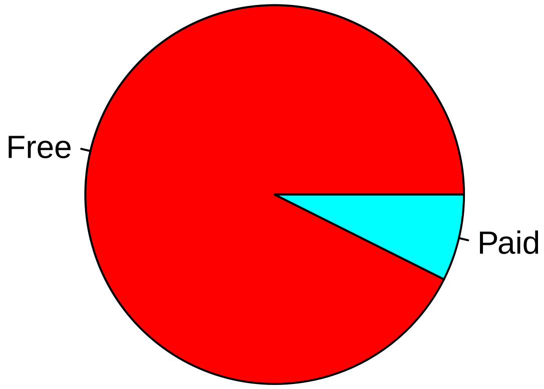
The most frequent category:

```
ggplot(playstore,aes(x=Category))+geom_bar()+
   guides(x = guide_axis(angle = 90))
```



Are paid apps or free apps the most frequent?

```
pie(table(playstore$Type),col = rainbow(2))
```



some bivariate plots

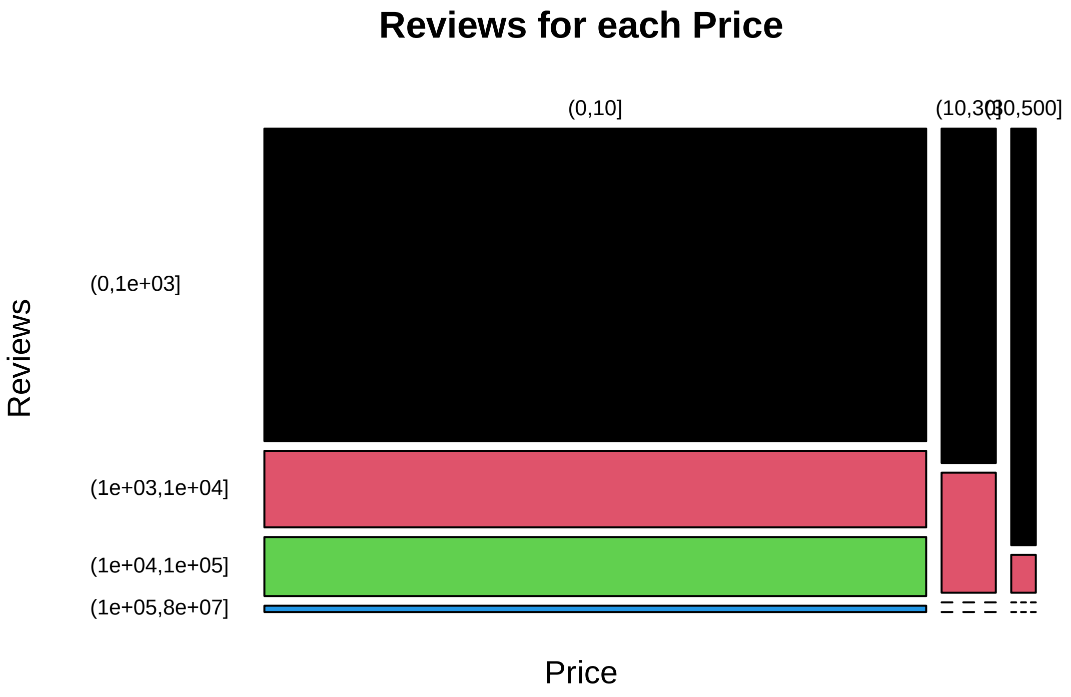
Review Vs Price

Which price and review range contains the most apps?

```
library(vcd)
```

```
## Loading required package: grid
```

```
mosaicplot(table(priceCut,reviewCut),ylab = "Reviews",xlab = "Price",main="Reviews for each Price",col=
```



The exact percentages for the above Mosaic Plot:

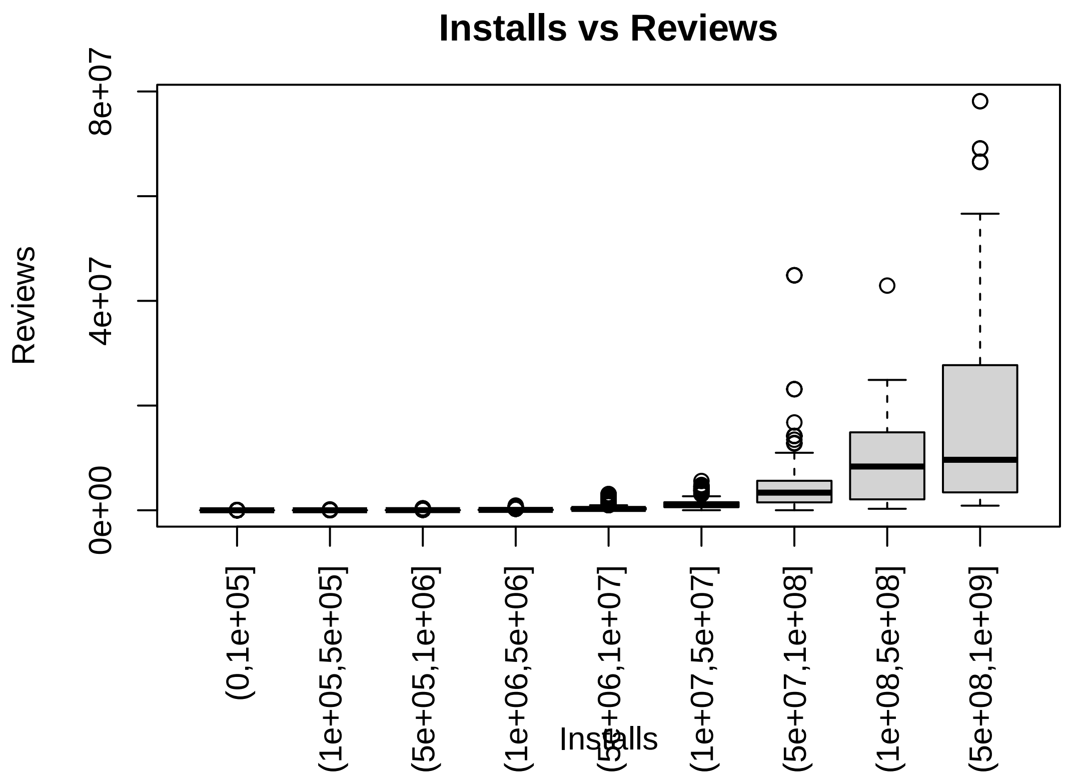
```
prop.table(table(priceCut,reviewCut))
```

```
##            reviewCut
## priceCut      (0,1e+03] (1e+03,1e+04] (1e+04,1e+05] (1e+05,8e+07]
##   (0,10]    0.614640884   0.150552486   0.116022099   0.012430939
##   (10,30]   0.053867403   0.019337017   0.000000000   0.000000000
##   (30,500]  0.030386740   0.002762431   0.000000000   0.000000000
```

Reviews Vs Number of Installs

As number of installs grows, the median of number of reviews increases as well:

```
par(xpd=T,mar=c(8,4,3,5))


boxplot(Reviews~installsCut,data = playstore,las=3,xlab = "",main="Installs vs Reviews")
mtext(text = "Installs",side = 1,line = 5)
```



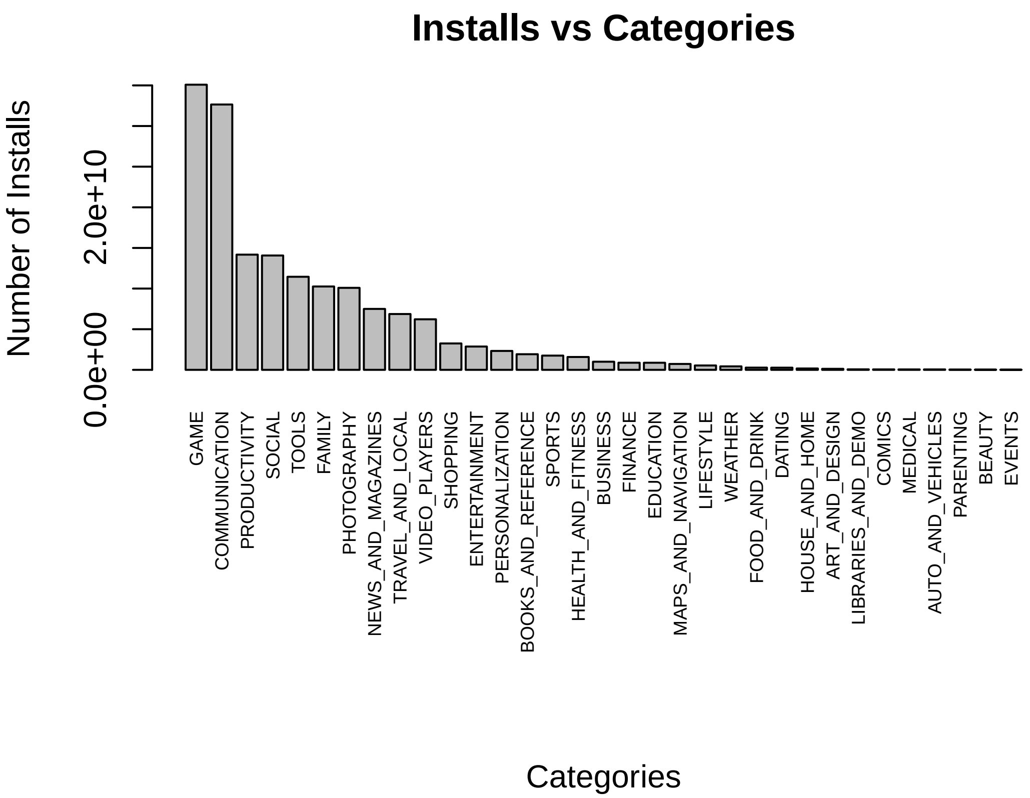
```
par(cex=1)
cor(playstore$Reviews,Installs_,method = "spearman")
```

```
## [1] 0.9712189
```

Installs per Category

Category with most number of installs:

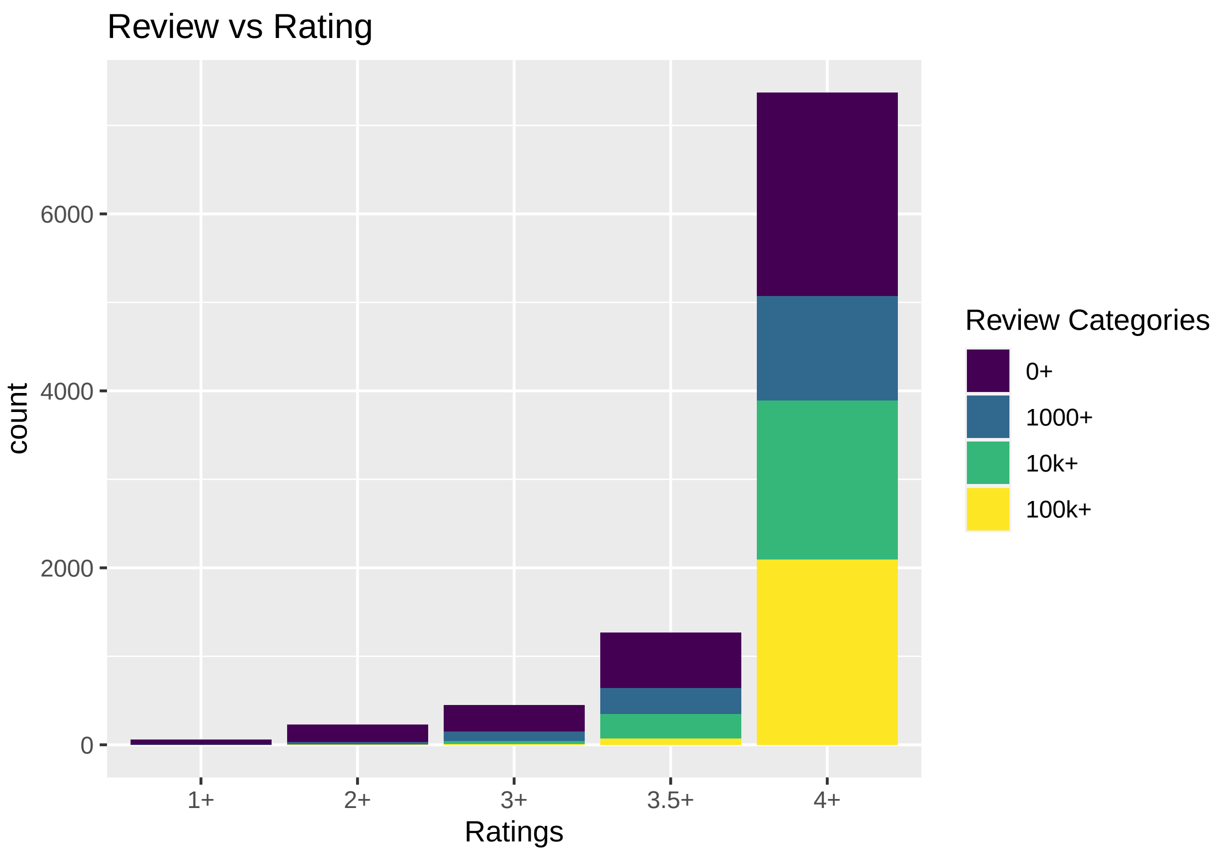
```
par(xpd=T,mar=c(12,4,3,5))
install_cat_table<-aggregate(Installs_~Category,data = playstore,FUN=sum)
vsel<-order(install_cat_table[,2],decreasing=T)
tab.agg<-install_cat_table[,2]
names(tab.agg)<-install_cat_table[,1]

df.bar<-barplot(tab.agg[vsel],las=3,cex.names = 0.6,ylab = "Number of Installs",main="Installs vs Categ
par(cex=1)
mtext(text = "Categories",side = 1,line = 10)
```



Reviews Vs Rating

What is the distribution of reviews for each rating?

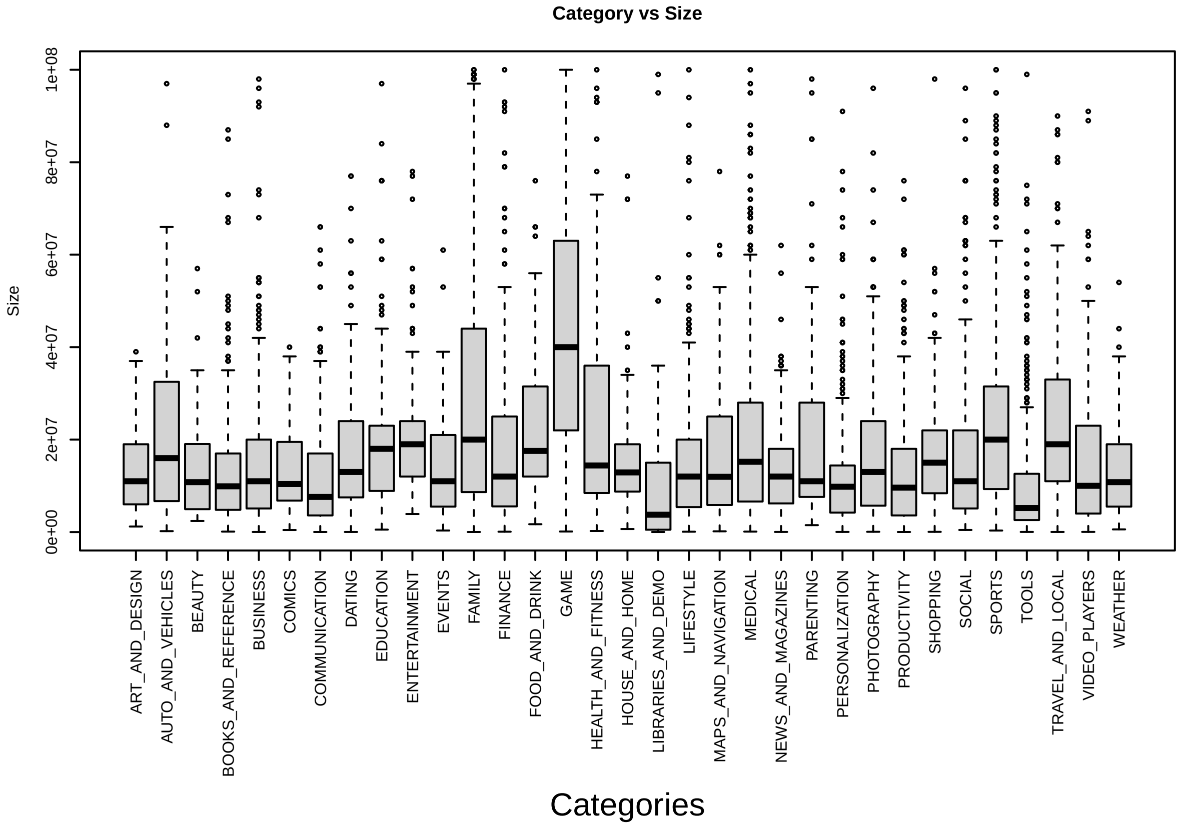
```
ggplot(subset(playstore,!is.na(ratingCut)),aes(x=ratingCut, fill=reviewCut))+
         geom_bar(position = "stack")+ggtitle("Review vs Rating")+
  xlab("Ratings")+guides(fill=guide_legend(title="Review Categories"))
```



Category vs Size

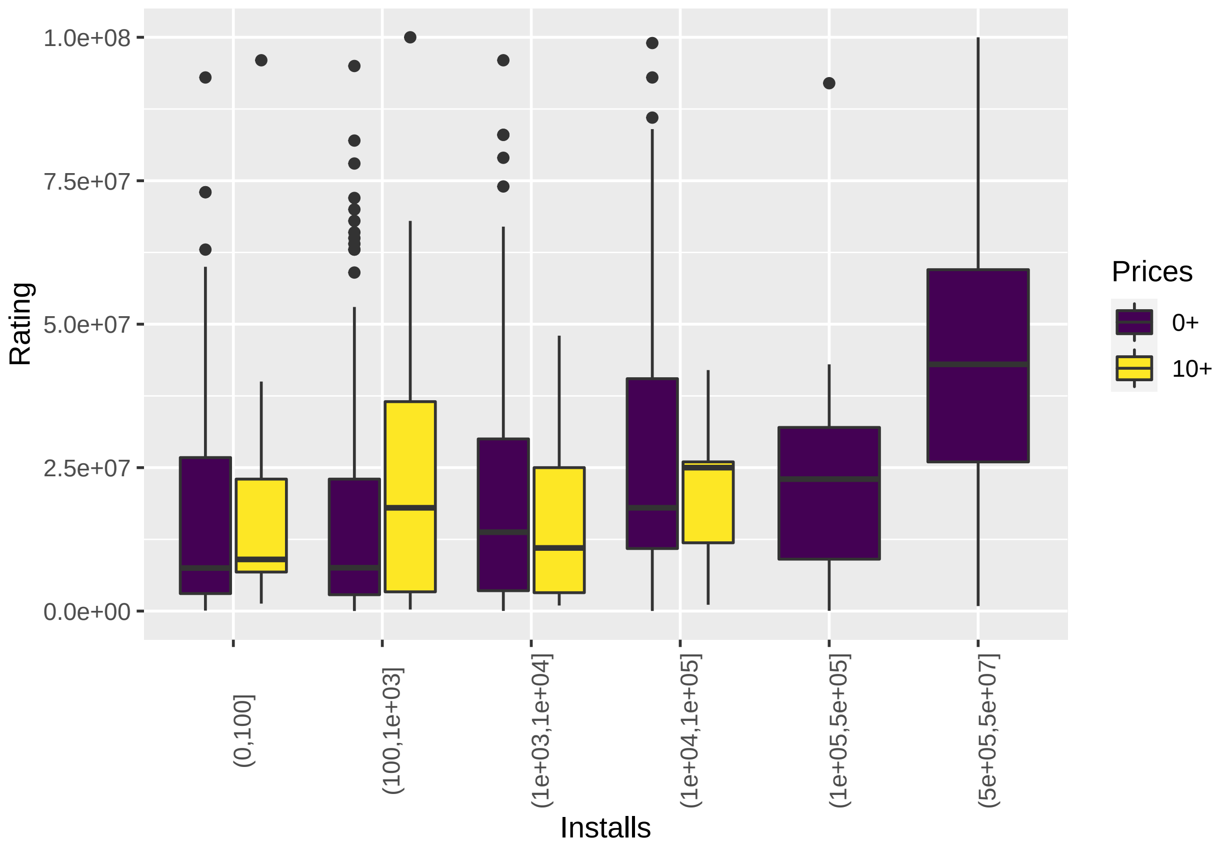
The most size instensive category:

```
par(xpd=NA,mar=c(15,5,4,3),cex=0.5)
boxplot(Size~Category,data=playstore,las=3,ylab = "Size",xlab = "",main="Category vs Size")
par(cex=1)
mtext(text = "Categories",side = 1,line = 13)
```



## some Multivariate plots

```
playstore_complete=playstore[complete.cases(playstore),]
ggplot(playstore_complete,aes(x=installsSuperCut,y=Size,fill=priceSuperCut))+geom_boxplot()+xlab("Insta
```



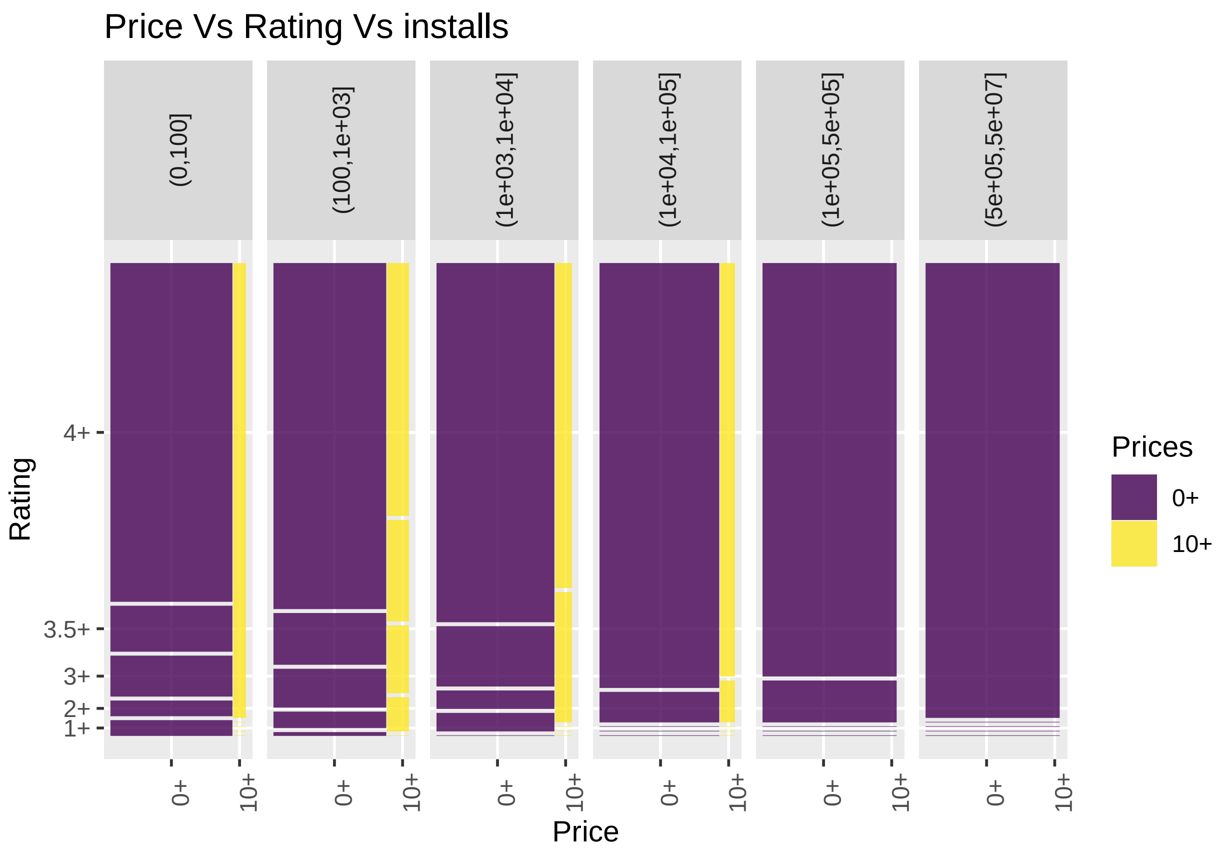
```
library(ggmosaic)
```

```
## Warning: package 'ggmosaic' was built under R version 4.1.2
```

```
##
## Attaching package: 'ggmosaic'
```

```
## The following objects are masked from 'package:vcd':
##
##     mosaic, spine
```

```
# ggplot(playstore_complete)+geom_mosaic(aes(x=product(ratingCut,priceCut,insatllsCut),fill=installsCut

ggplot(playstore_complete)+geom_mosaic(aes(x=product(ratingCut,priceSuperCut),fill=priceSuperCut))+face
```



Inferences and Conclusions

In this notebook we did Exploratory Data Analysis on Google Play Store Apps datset. We drew interesting inferences from this dataset:

- Based on the dataset we can infer that most of the Apps in play store belongs to Family and Gaming categories followed by Tools, Medical and Business.
- Also based on the type metric it seems that only 7% of apps are paid and around 93% of apps are free to install.
- From the ratings, it appears that people are tend to give ratings in the range of 3 to 5. And the more reviews an app has, the better is its rating.
- From the sizes, it appears that most apps tend to be below 20 MB in size. And the most size intensive apps are mostly games.
- Based on the popularity, the apps in Gaming category were installed most number of time followed by Communication, Productivity and Social.
- Based on the number of installs, the facebook app instlled most number of time followed by gmail.
- Based on the reviews, Facebook, WhatsApp and instagram has most no of Reviews in google platy store.

- Most (61%) of the apps are [0,10) dollars and have [0,1000) Review
- There is also a big correlation between number of installs and number of reviews for each app. The more an app is installed, the more reviews it has.