Exploratory Data Analysis

Yu-Chen (Amber) Lu

Problem 1

Explore realdirect.com thinking about how buyers and sellers would navigate, and how the website is organized. Use the datasets provided for Bronx, Brooklyn, Manhattan, Queens, and Staten Island. Do the following:

Load in and clean the data.

Solution:

```
library(plyr)
library(gdata)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
##
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
##
## Attaching package: 'gdata'
## The following object is masked from 'package:stats':
##
##     nobs
## The following object is masked from 'package:utils':
##
##     object.size
## The following object is masked from 'package:base':
##
##     startsWith
```

```
library(ggplot2)
library(lattice)

setwd("/Users/yuchenlu/Github")
getwd()
```

```
## [1] "/Users/yuchenlu/GitHub"
```

```
AllFiles = c("bronx", "brooklyn", "manhattan", "queens", "statenisland")

for ( AllFiles in AllFiles)
{
  df = read.xls(paste("DataForExploratoryDataAnalysis/rollingsales_",AllFiles,".xls",sep=""), pattern="

  names(df) = tolower(names(df))   # lower all the columns' names
  # make sure the dates are formatted and values are numerical
  df$sale.price.n = as.numeric(gsub("[^[:digit:]]","",df$sale.price))
  df$gross.sqft = as.numeric(gsub("[^[:digit:]]","",df$gross.square.feet))
```

```
  df$land.sqft = as.numeric(gsub("[^[:digit:]]","",df$land.square.feet))
  df$borough = as.character(df$borough)
  df$sale.date = as.Date(df$sale.date)
  df$year.built = as.numeric(as.character(df$year.built))

  if (AllFiles=="bronx"){ bx=df }
  else if (AllFiles=="brooklyn"){bk=df}
  else if (AllFiles=="manhattan"){mh=df}
  else if (AllFiles=="queens"){qn=df}
  else {si=df}

}
```

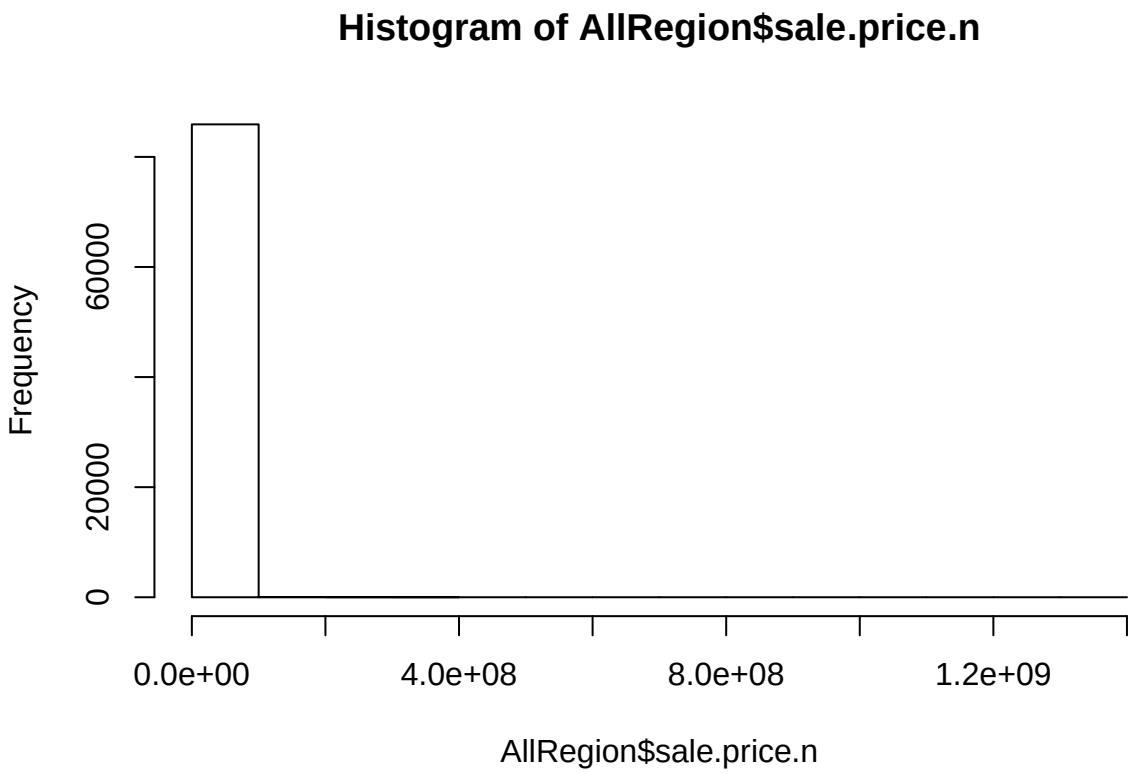
Conduct exploratory data analysis in order to find out where there are outliers or missing values, decide how you will treat them, make sure the dates are formatted correctly, make sure values you think are numerical are being treated as such, etc.

Solution:

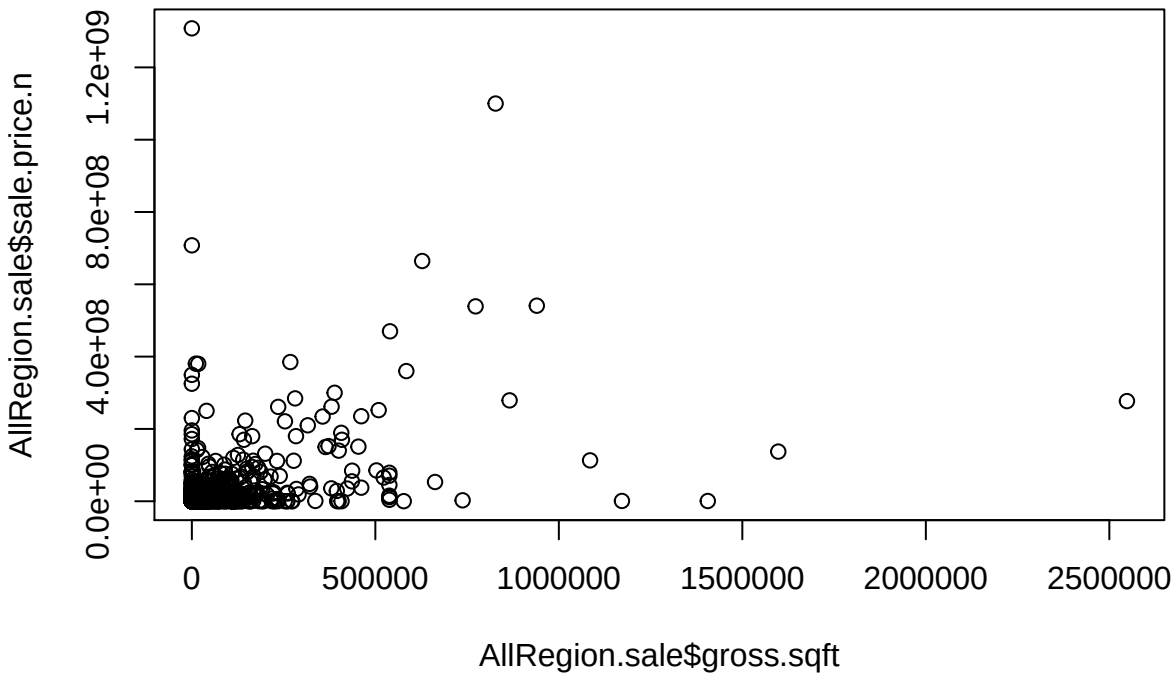
```
# combine all regions into into one dataframe
AllRegion = rbind(bx,qn,mh,si,bk)
```

```
## Warning in `[<-.factor`(`*tmp*`, ri, value = c(0L, 0L, 0L, 0L, 0L, 0L,
## 0L, : invalid factor level, NA generated

## Warning in `[<-.factor`(`*tmp*`, ri, value = c(0L, 0L, 0L, 0L, 0L, 0L,
## 0L, : invalid factor level, NA generated
```

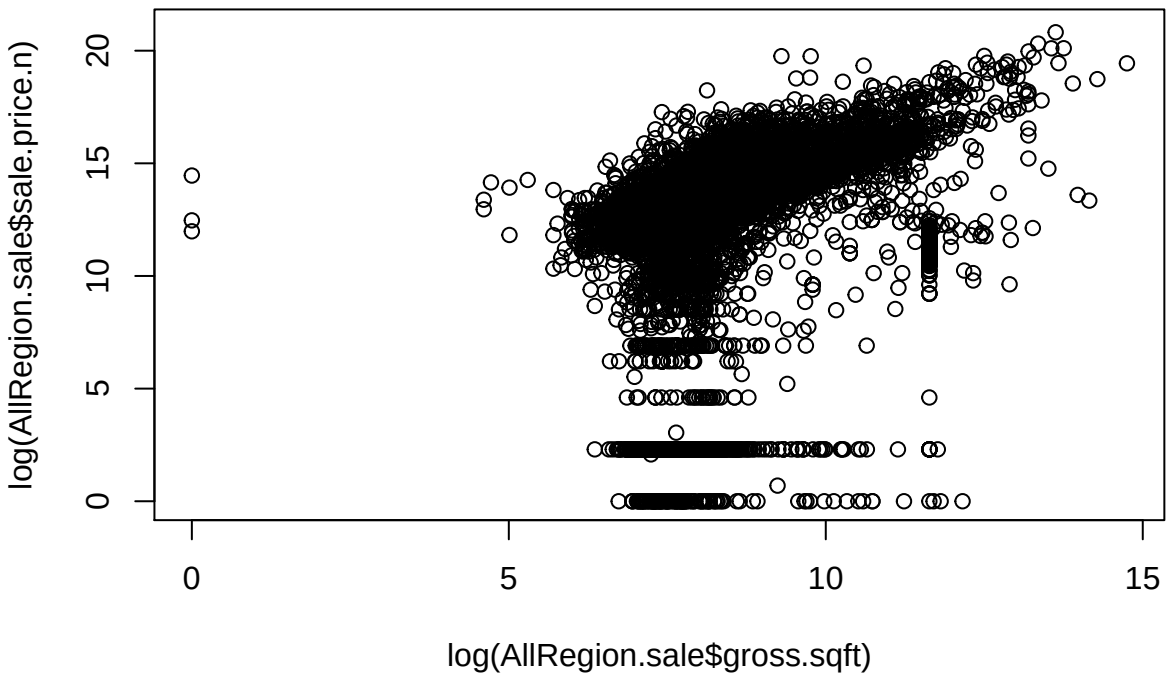
```
# do a bit of exploration on **All Region** to make sure the data is ok
hist(AllRegion$sale.price.n)
```



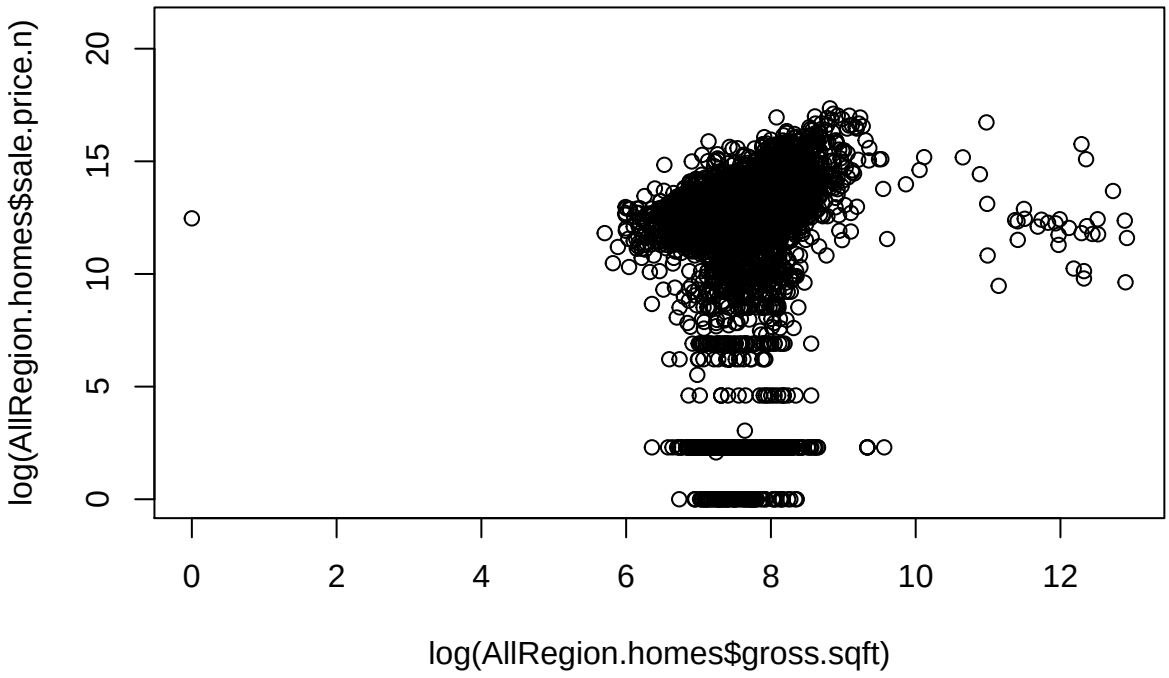

```
## NOTE: The range of sale prices is large and most of them are close to 0.
##       Therefore, I keep only the actual sales in the following code.}

# keep only the actual sales
AllRegion.sale = AllRegion[AllRegion$sale.price.n!=0,]
plot(AllRegion.sale$gross.sqft,AllRegion.sale$sale.price.n)
```



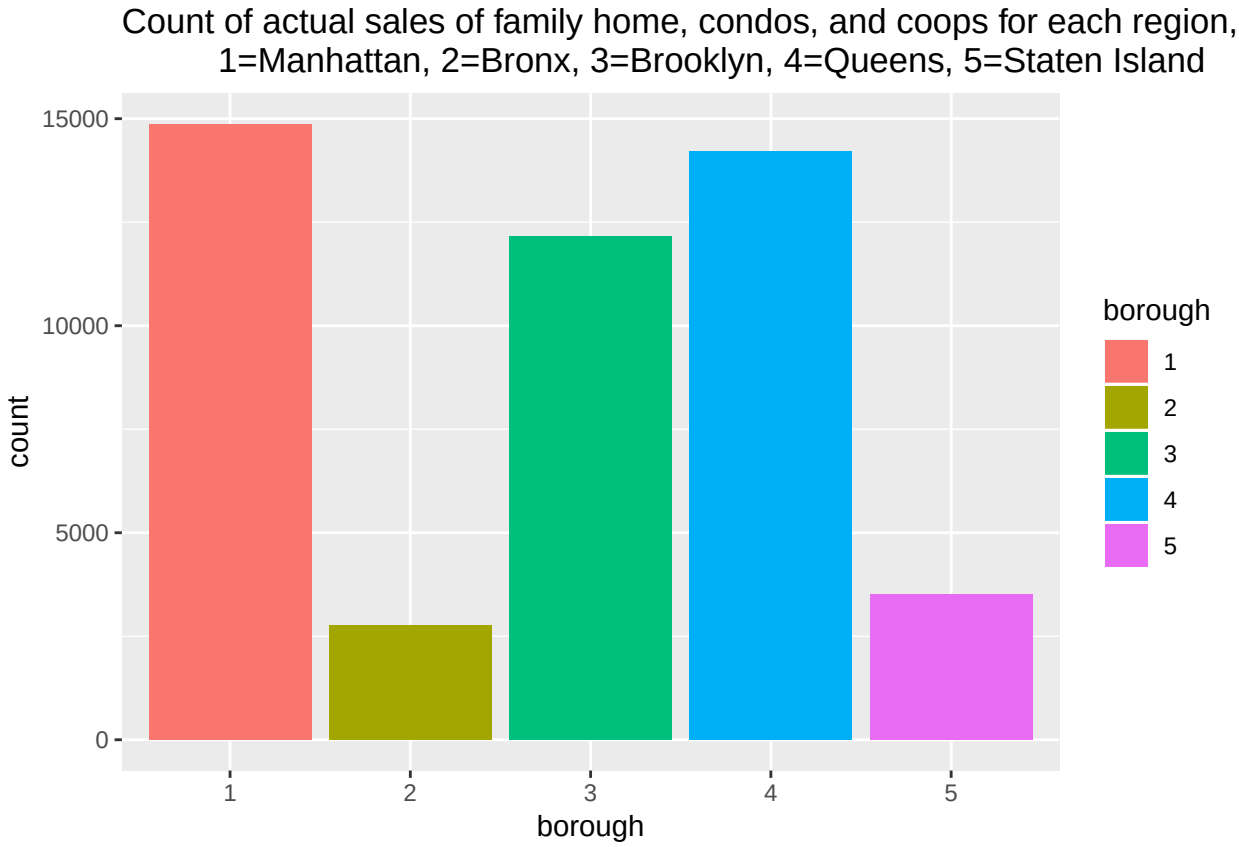

```
## NOTE: Most points locate about the origin in the Quadrant 1,
##       so I take log of both two variables.
plot(log(AllRegion.sale$gross.sqft),log(AllRegion.sale$sale.price.n))
```



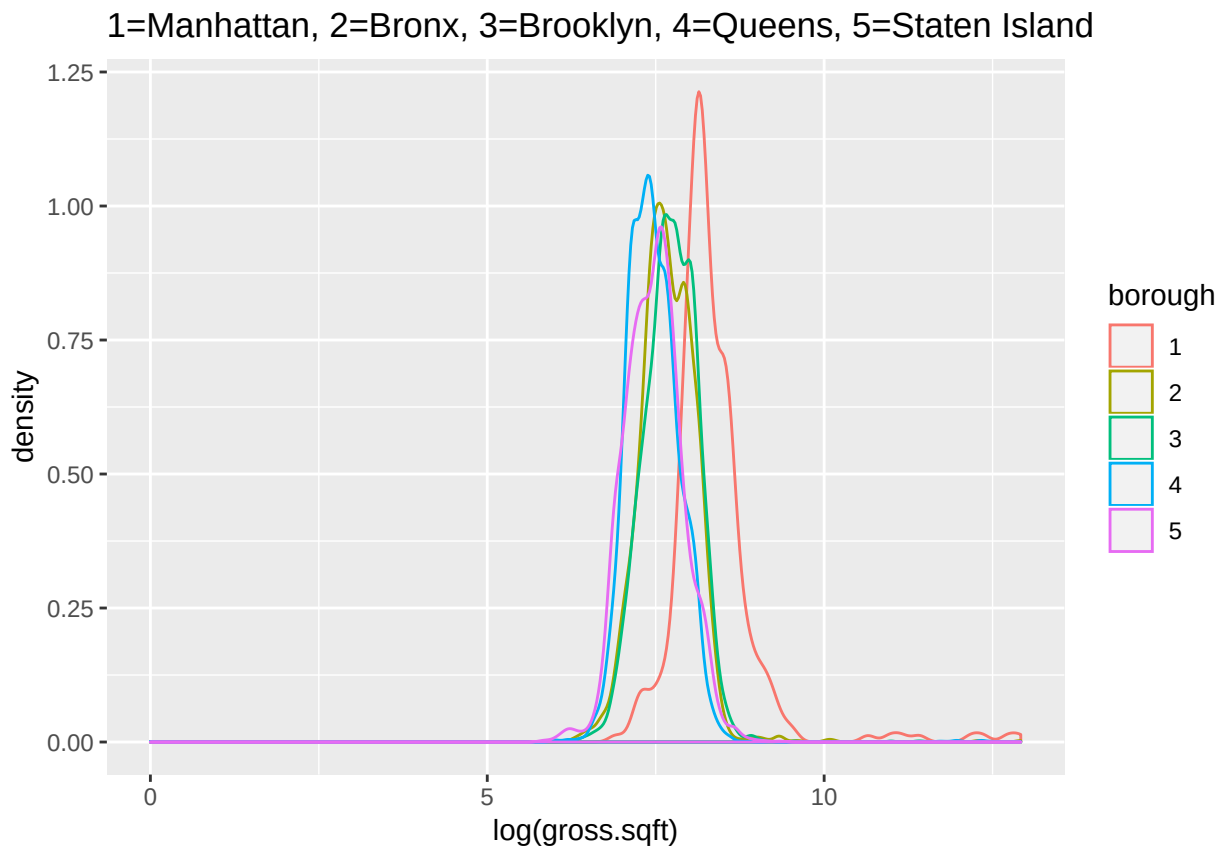

```
## NOTE: This graph is much better, but there are still some noises.

# for now, let's look at 1, 2, 3 family homes, condos, and coops
AllRegion.homes = AllRegion.sale [which(grepl("FAMILY | CONDOS | COOPS",
          AllRegion.sale$building.class.category)),]
plot(log(AllRegion.homes$gross.sqft),log(AllRegion.homes$sale.price.n))
```



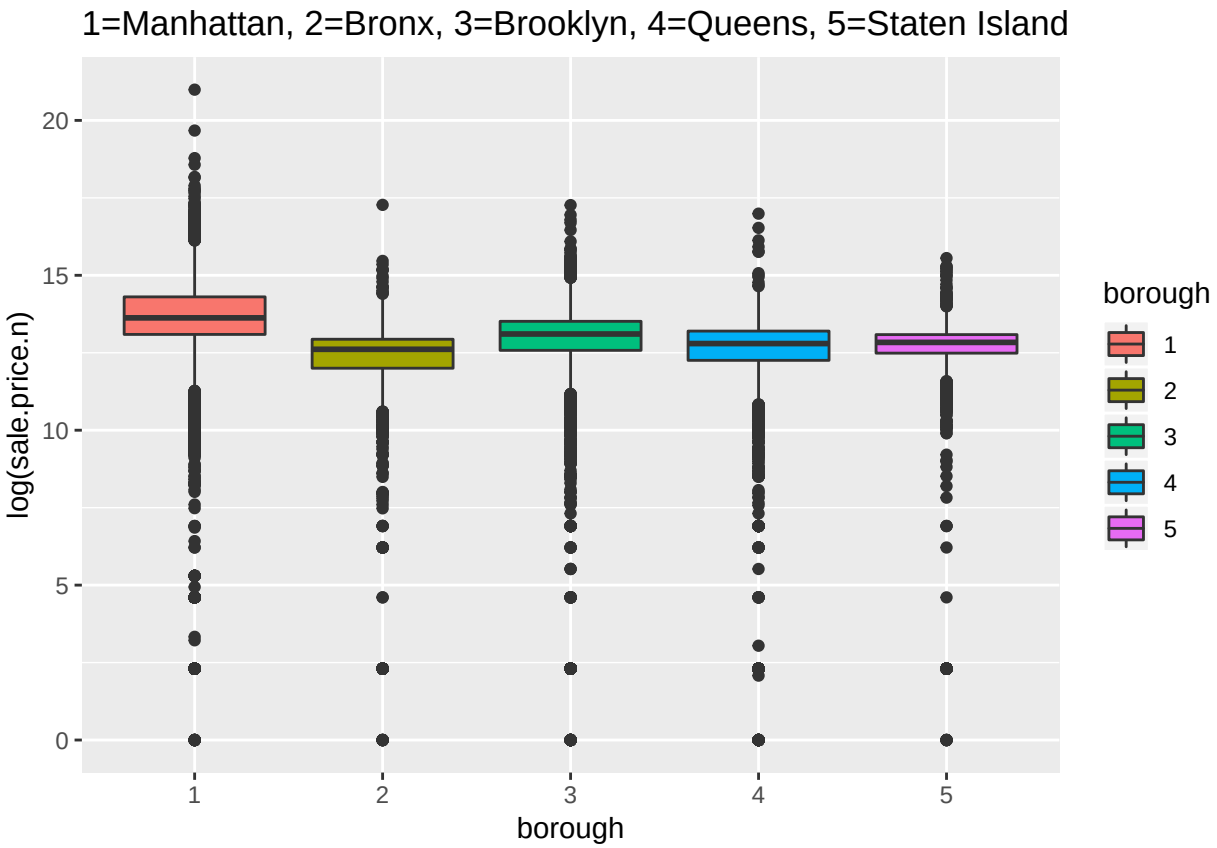

```
## NOTE: Less noises appear in this graph.

library("ggplot2")
ggplot(data= AllRegion.homes, aes(x=borough, fill=borough))+geom_bar()+
  ggtitle("Count of actual sales of family home, condos, and coops for each region,
          1=Manhattan, 2=Bronx, 3=Brooklyn, 4=Queens, 5=Staten Island")
```




```
# sale price
ggplot(subset(AllRegion.homes, gross.sqft>0), aes(x=log(gross.sqft),
                                                  colour=borough))+geom_density()+
  ggtitle("1=Manhattan, 2=Bronx, 3=Brooklyn, 4=Queens, 5=Staten Island")
```




```
ggplot(data= AllRegion.homes, aes(x=borough,
                                  y=log(sale.price.n),fill=borough))+geom_boxplot()+
  ggtitle("1=Manhattan, 2=Bronx, 3=Brooklyn, 4=Queens, 5=Staten Island")
```

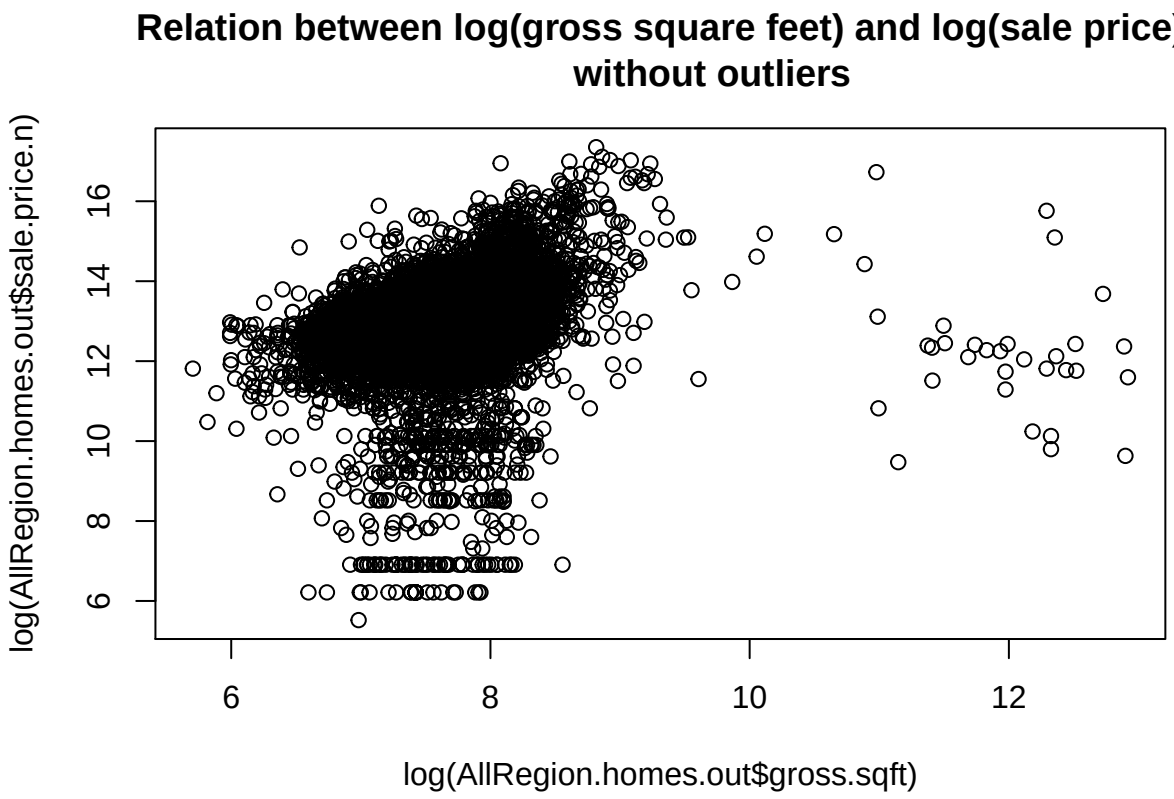



```
tapply(log(AllRegion.homes$sale.price.n), AllRegion.homes$borough, summary)
```

```
## $`1`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   13.09   13.63   13.55   14.30   20.99
##
## $`2`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   12.00   12.61   12.18   12.94   17.27
##
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   12.58   13.11   12.80   13.51   17.27
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   12.25   12.80   12.50   13.20   16.99
##
## $`5`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   12.49   12.83   12.52   13.09   15.56
```

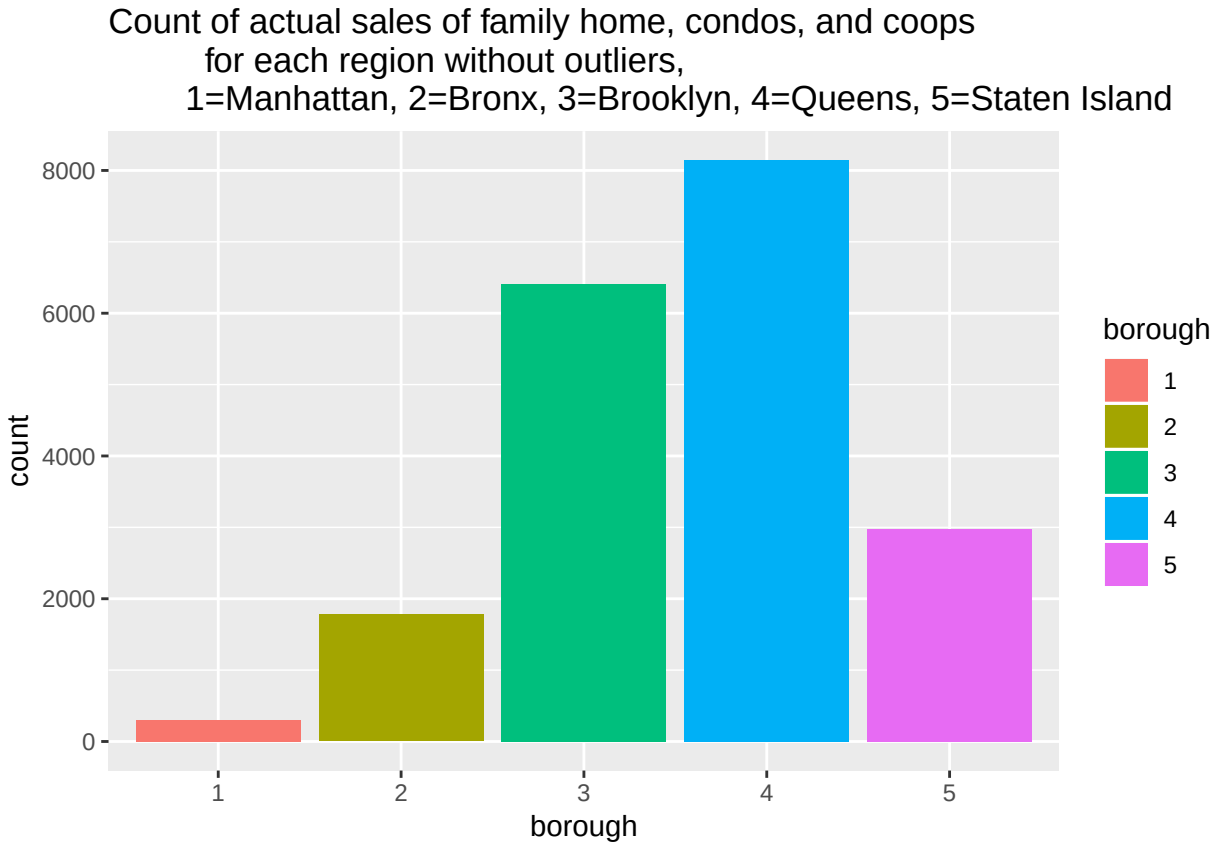
```
## NOTE: Take a closer eye on actual sale prices, the distribution of
##       ones in Manhattan has the hightest mean and higher kurtosis.
##       The prices of other four regions have similar distributions.
```

```r
# remove outliers that seem like they weren't actual sales
# and the extreme cases
AllRegion.homes$outliers = NULL
AllRegion.homes$outliers = (log(AllRegion.homes$sale.price.n) <= 5 |
                            log(AllRegion.homes$gross.sqft) <= 2) + 0
AllRegion.homes.out = AllRegion.homes[which(AllRegion.homes$outlier==0),]

plot(log(AllRegion.homes.out$gross.sqft),log(AllRegion.homes.out$sale.price.n))+ title("Relation betwee
          without outliers")
```

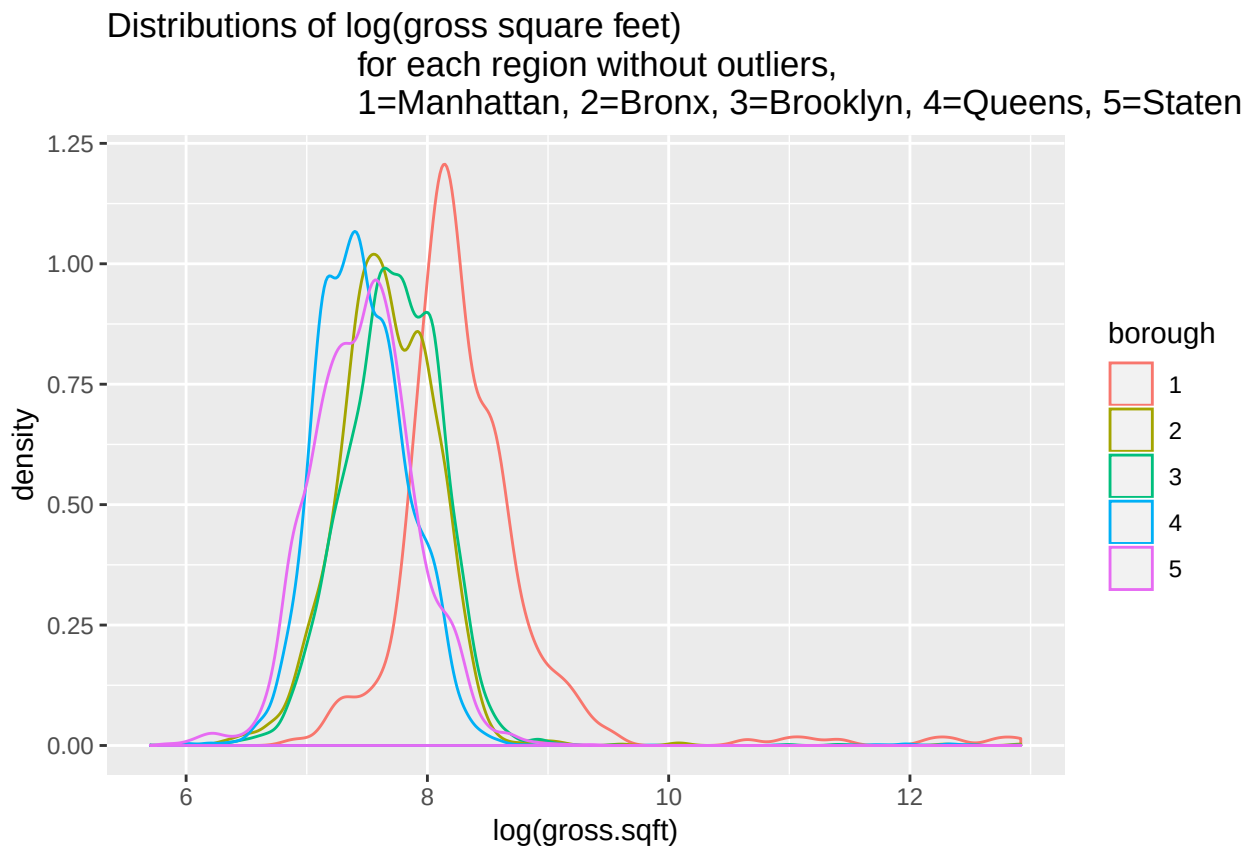



```
## integer(0)
```

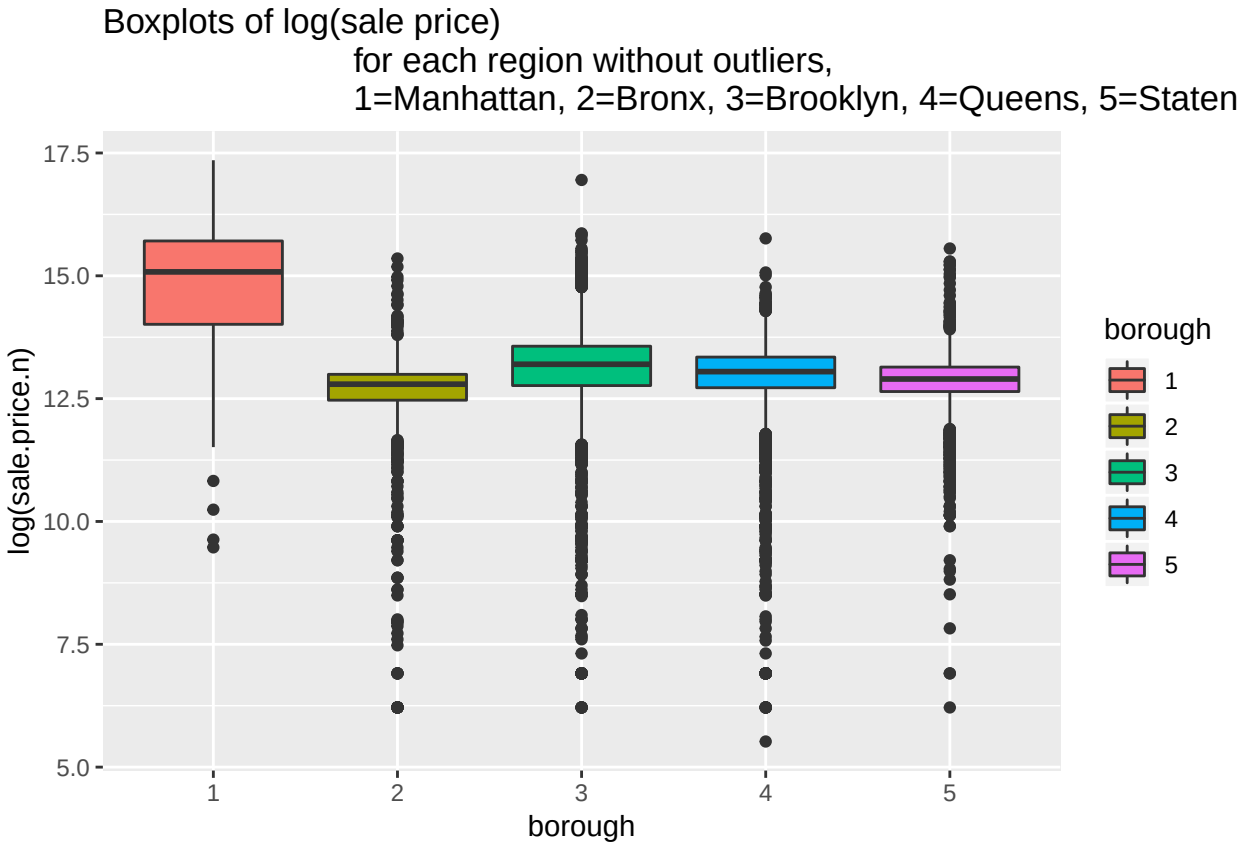
```r
## NOTE: Gross square feet doesn't like relative to the sale prices.

ggplot(AllRegion.homes.out, aes(x=borough, fill=borough))+geom_bar()+
ggtitle("Count of actual sales of family home, condos, and coops
          for each region without outliers,
        1=Manhattan, 2=Bronx, 3=Brooklyn, 4=Queens, 5=Staten Island")
```




```
## NOTE: Without removing the outliers, the number of the actual
##       sales in manhattan is the largest,
##       but now it is the smallest one.

# sale price
ggplot(AllRegion.homes.out, aes(x=log(gross.sqft), colour=borough))+
  geom_density()+ ggtitle("Distributions of log(gross square feet)
                          for each region without outliers,
                          1=Manhattan, 2=Bronx, 3=Brooklyn, 4=Queens, 5=Staten Island")
```




```
ggplot(AllRegion.homes.out, aes(x=borough, y=log(sale.price.n), fill=borough))+
  geom_boxplot()+ ggtitle("Boxplots of log(sale price)
                          for each region without outliers,
                          1=Manhattan, 2=Bronx, 3=Brooklyn, 4=Queens, 5=Staten Island")
```




```
## NOTE: The first quartile, median and third quartile in Manhattan
##       are the highest than other regions.
```

Conduct exploratory data analysis to visualize and make comparisons for residential building category classes across boroughs and across time (select the following: 1-, 2-, and 3-family homes, coops, and condos). Use histograms, boxplots, scatterplots or other visual graphs. Provide summary statistics along with your conclusions.

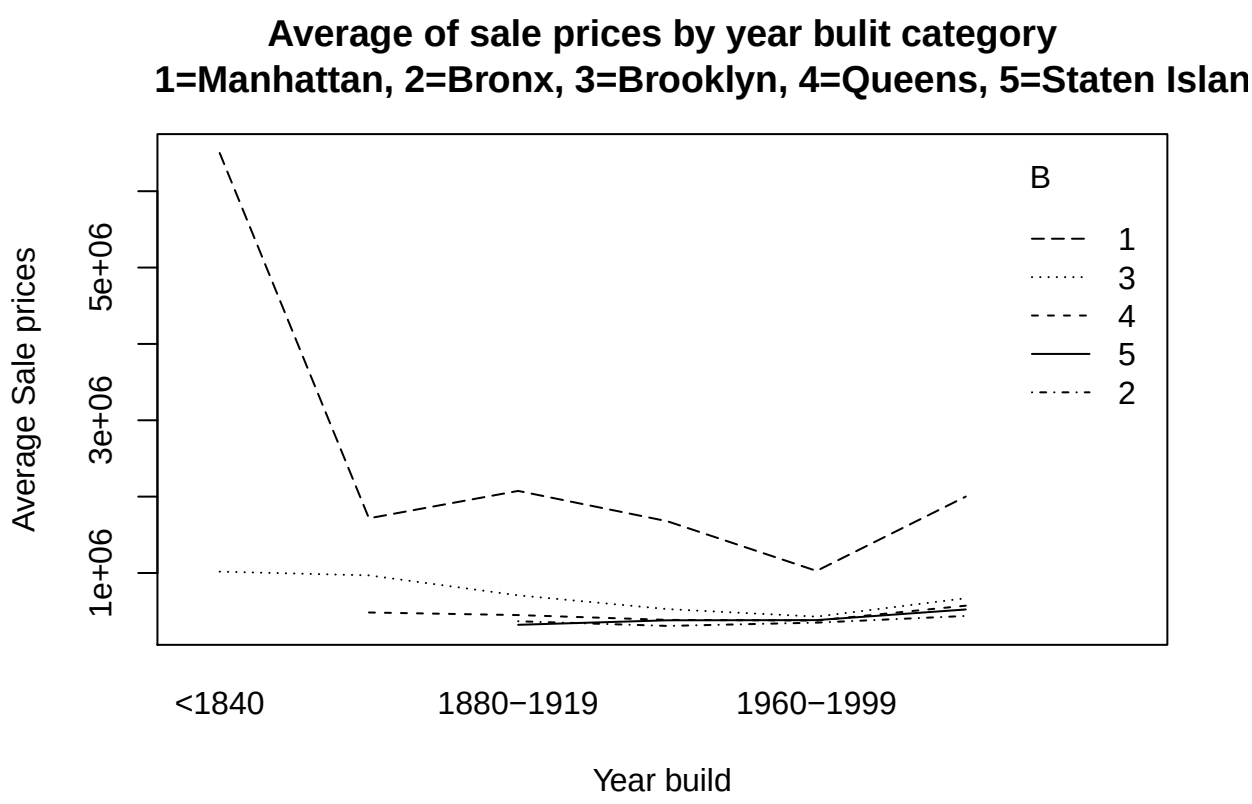
Solution:

```
AllRegion.homes = AllRegion.homes[AllRegion.homes$year.built!=0,]
AllRegion.homes$yearcat =cut(AllRegion.homes$year.built, c(0,1840,1880,1920,1960,2000,Inf),right = FALSE
unique(AllRegion.homes$building.class.category)
```

```
##  [1] 01  ONE FAMILY HOMES
##  [2] 02  TWO FAMILY HOMES
##  [3] 03  THREE FAMILY HOMES
##  [4] 10  COOPS - ELEVATOR APARTMENTS
##  [5] 04  TAX CLASS 1 CONDOS
##  [6] 28  COMMERCIAL CONDOS
##  [7] 09  COOPS - WALKUP APARTMENTS
##  [8] 13  CONDOS - ELEVATOR APARTMENTS
##  [9] 12  CONDOS - WALKUP APARTMENTS
```

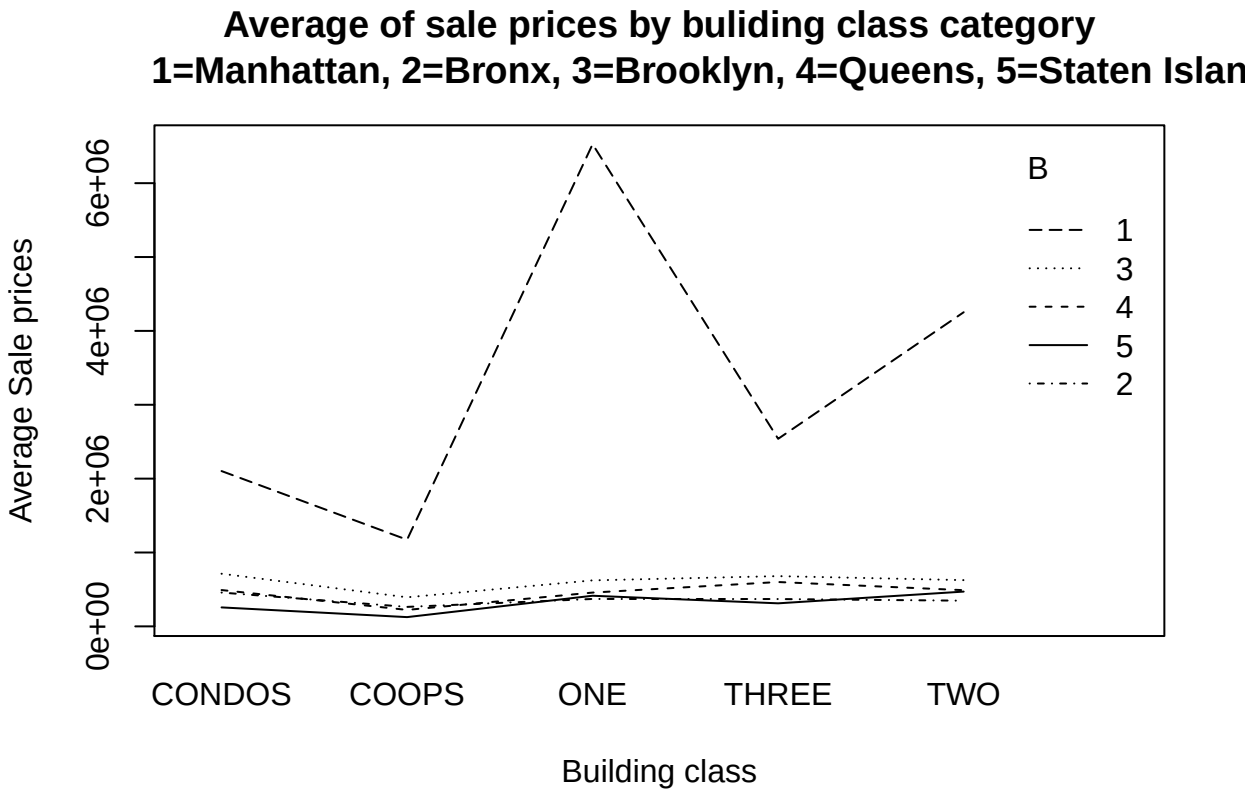
```
## [10] 15  CONDOS - 2-10 UNIT RESIDENTIAL
## [11] 16  CONDOS - 2-10 UNIT WITH COMMERCIAL UNIT
## 40 Levels:                                                ...
```

```
means =with(AllRegion.homes,aggregate(x=list(Y=sale.price.n),by=list(A=yearcat, B=borough),mean))
with(means,interaction.plot(x.factor=A, trace.factor=B, response=Y, type='l',
          main = "Average of sale prices by year bulit category
          1=Manhattan, 2=Bronx, 3=Brooklyn, 4=Queens, 5=Staten Island",
                          xlab = "Year build",ylab= "Average Sale prices"))
```




```
## NOTE: The sales prices are similar no matter which year the houses built,
##       except for ones in Manhattan.

AllRegion.homes$newbuildingcat = AllRegion.homes$building.class.category

tt = as.vector(AllRegion.homes$newbuildingcat)

tt[grep("COOPS", tt)] = "COOPS"
tt[grep("CONDOS",tt)] = "CONDOS"
tt[grep("ONE FAMILY",tt)] = "ONE"
tt[grep("TWO FAMILY",tt)] = "TWO"
tt[grep("THREE FAMILY",tt)] = "THREE"
AllRegion.homes$newbuildingcat = tt

homes =with(AllRegion.homes,aggregate(x=list(Y=sale.price.n),by=list(A=newbuildingcat, B=borough),mean))
with(homes,interaction.plot(x.factor=A, trace.factor=B, response=Y, type='l',
          main = "Average of sale prices by buliding class category
```

```
          1=Manhattan, 2=Bronx, 3=Brooklyn, 4=Queens, 5=Staten Island",
                              xlab = "Building class",ylab= "Average Sale prices"))
```




```
## NOTE: The sale prices in Manhattan are very different by the buliding class.
##       Other regions are not so different.
```

Problem 2

The datasets provided nyt1.csv, nyt2.csv, and nyt3.csv represents three (simulated) days of ads shown and clicks recorded on the New York Times homepage. Each row represents a single user. There are 5 columns: age, gender (0=female, 1=male), number impressions, number clicks, and logged-in. Use R to handle this data. Perform some exploratory data analysis:

Create a new variable, age_group, that categorizes users as "<20", "20-29", "30-39", "40-49", "50-59", "60-69", and "70+".

Solution:

```
setwd("/Users/yuchenlu/Github")
allnyt = c('nyt1', 'nyt2', 'nyt3')
for ( allnyt in allnyt)
{
```

```
  df = read.csv(paste("DataForExploratoryDataAnalysis/",allnyt,".csv",sep=""))
  #categorize

  df$agecat = cut(df$Age, c(0,20,30,40,50,60,70,Inf),
                  right = FALSE,
                  labels = c("<20", "20-29", "30-39",
                        "40-49", "50-59", "60-69", "70+"))

  as.character(df$Gender)

  if (allnyt=="nyt1"){ data1=df }
  else if (allnyt=="nyt2"){data2=df}
  else {data3=df}
}
summary(data3)
```

```
##       Age             Gender        Impressions         Clicks
##  Min.   :  0.00   Min.   :0.0000   Min.   : 0.000   Min.   :0.00000
##  1st Qu.:  0.00   1st Qu.:0.0000   1st Qu.: 3.000   1st Qu.:0.00000
##  Median : 31.00   Median :0.0000   Median : 5.000   Median :0.00000
##  Mean   : 29.47   Mean   :0.3668   Mean   : 4.996   Mean   :0.09226
##  3rd Qu.: 48.00   3rd Qu.:1.0000   3rd Qu.: 6.000   3rd Qu.:0.00000
##  Max.   :109.00   Max.   :1.0000   Max.   :19.000   Max.   :6.00000
##
##    Signed_In        agecat
##  Min.   :0.0000   <20  :156822
##  1st Qu.:0.0000   20-29: 55527
##  Median :1.0000   30-39: 61464
##  Mean   :0.7001   40-49: 64699
##  3rd Qu.:1.0000   50-59: 52668
##  Max.   :1.0000   60-69: 31249
##                   70+  : 17941
```

For each day:

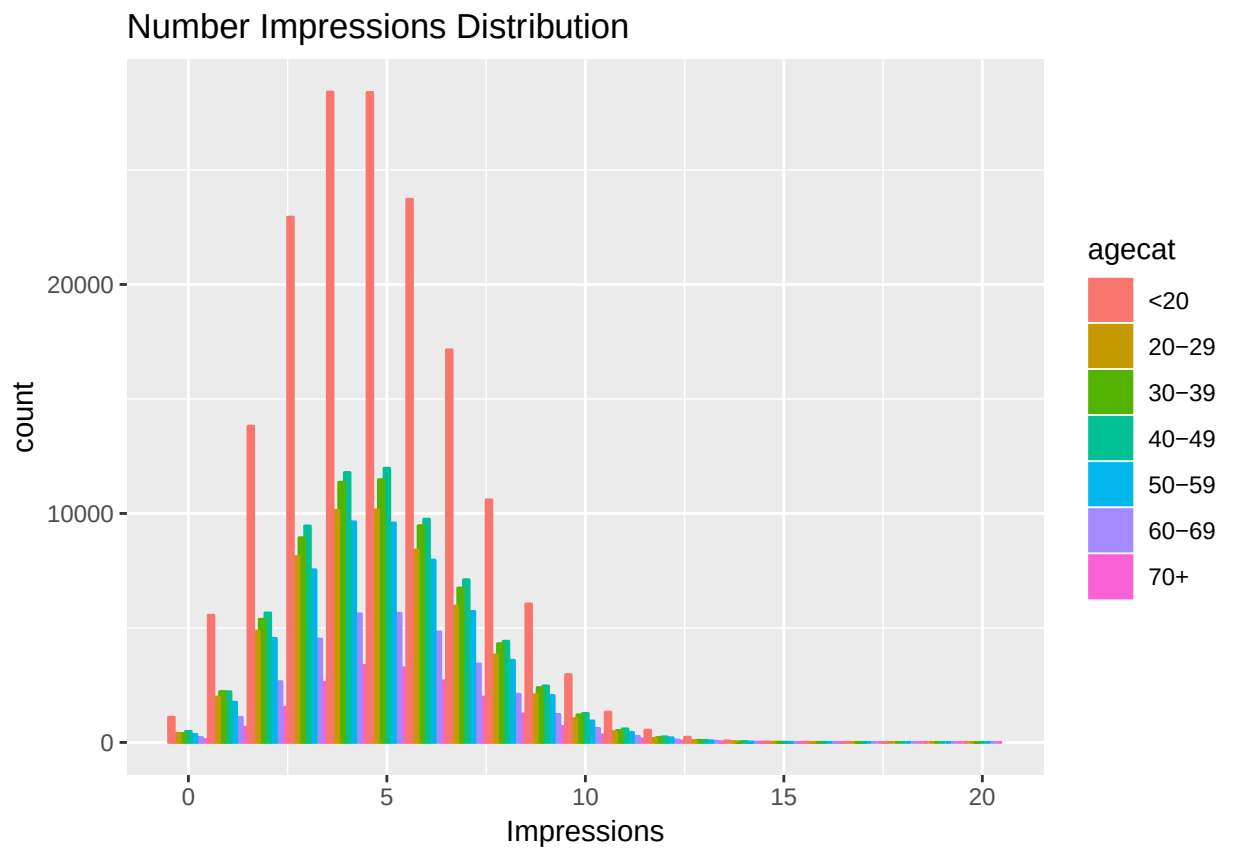
o Plot the distribution of number of impressions and click-through-rate (CTR =#clicks / #impressions) for these age categories

o Define a new variable to segment or categorize users based on their click behavior.

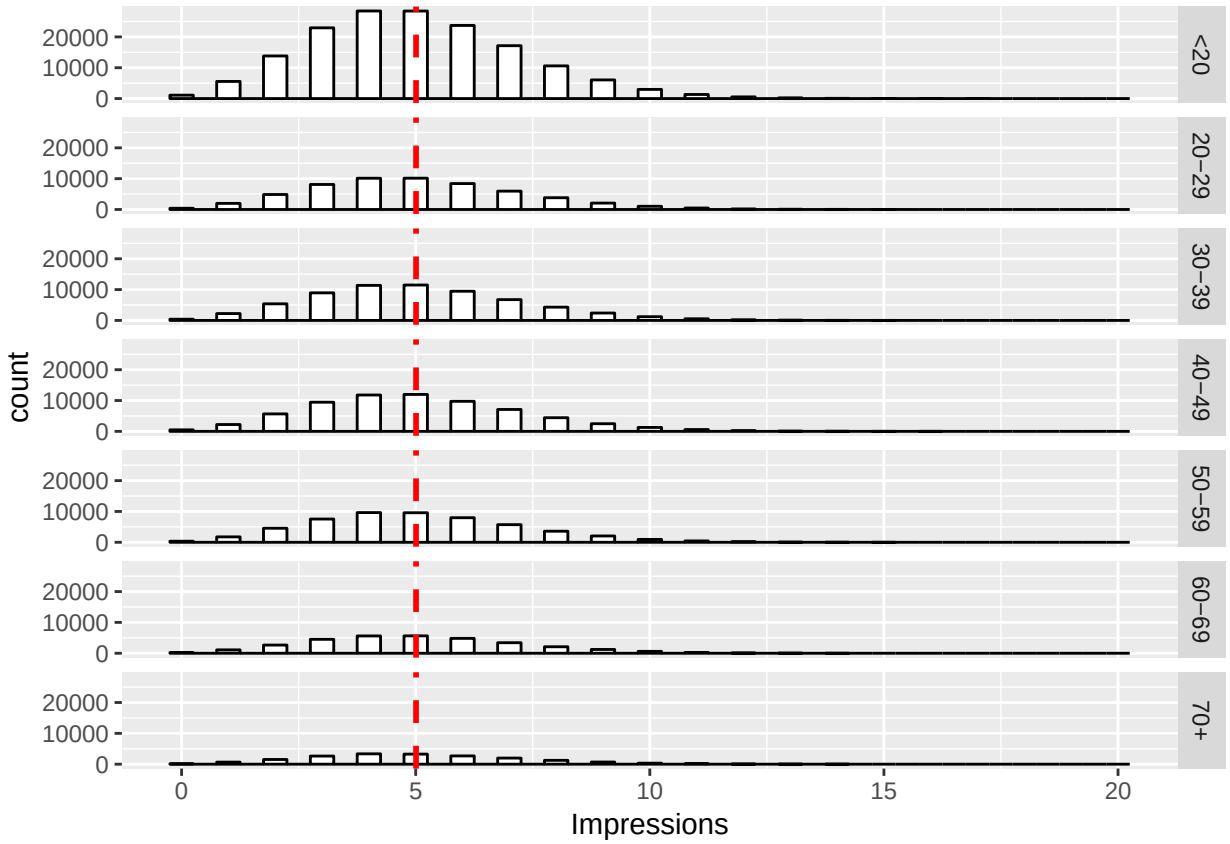
o Explore the data and make visual and quantitative comparisons across user segments/demographics (<20-year-old males versus <20-year-old females or logged-in versus not, for example).

```
# Plot the distribution of number of impressions for these age categories
impressions.plot = ggplot(data = data1,
                          aes(Impressions,
                              fill = agecat,
                              color = agecat))
```

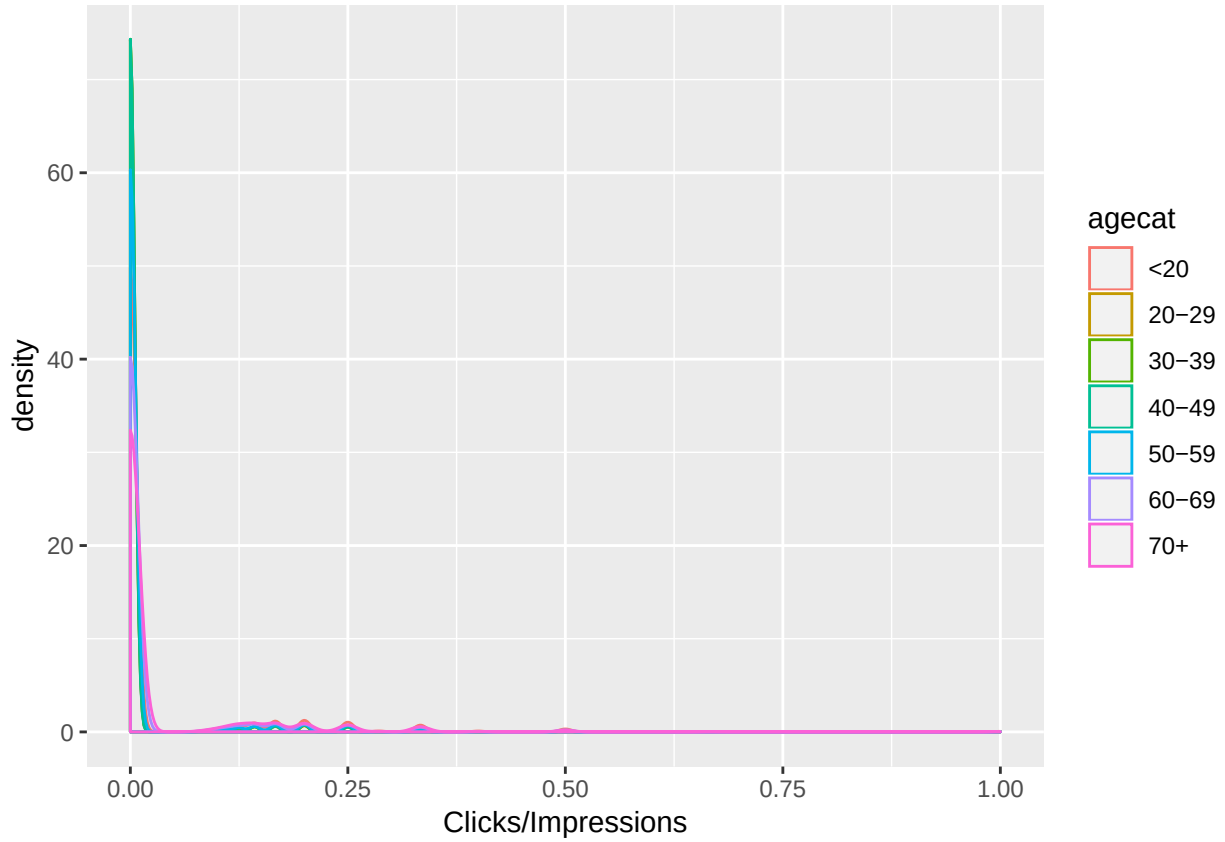
```
impressions.plot +  geom_histogram(position = "dodge", binwidth = 1) +
  labs(title = "Number Impressions Distribution")
```



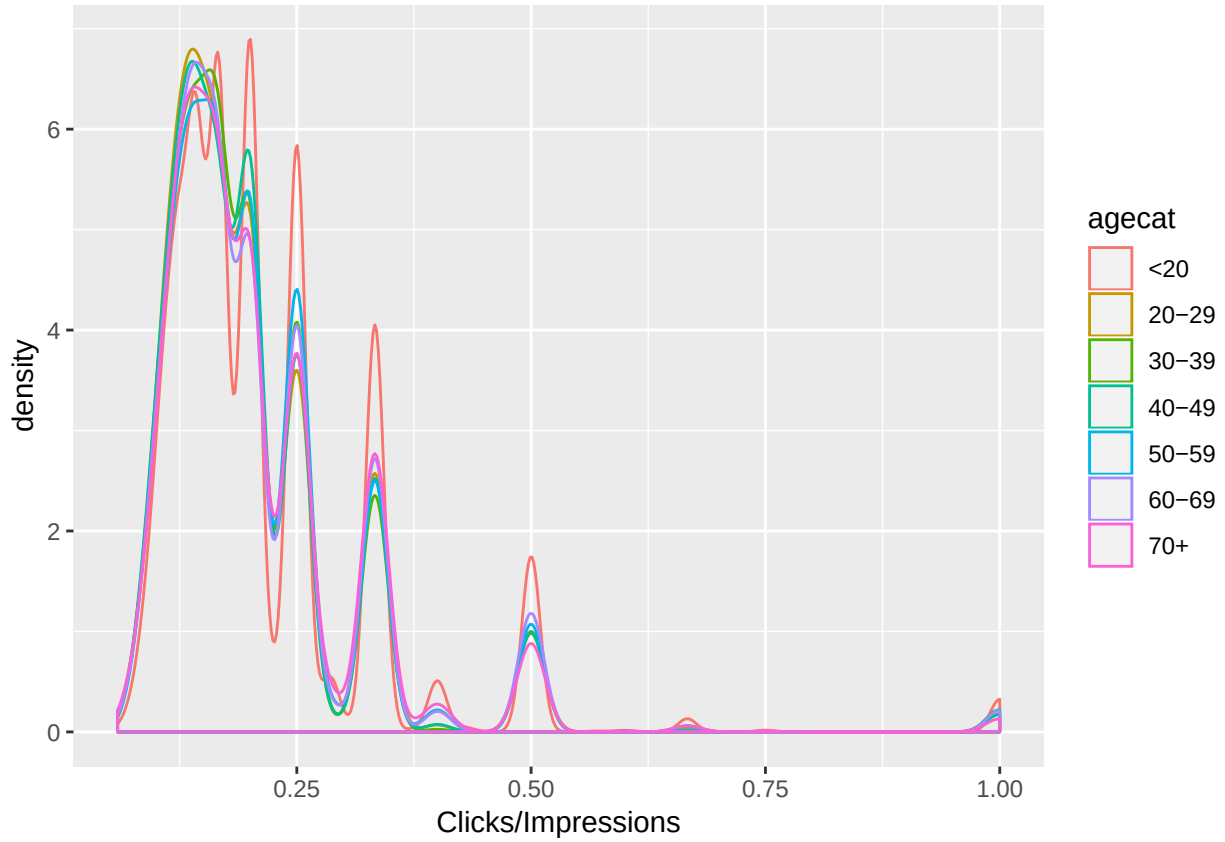

```
## NOTE: We can see that there are more impressions in the group
##       with age below 20. Moreover, the highest density of each
##       age group is about 5 impressions.

ggplot(data1, aes(x=Impressions)) + geom_histogram(binwidth=.5, colour="black", fill="white") +
  facet_grid(agecat ~ .) +
  geom_vline(data=data1, aes(xintercept=mean(Impressions)),
             linetype="dashed", size=1, colour="red")
```




```
## NOTE: We can see the distributions of each group,
##       which are normally distributed, seperately in the graph.

# Plot the distribution of number of click-through-rate
# (CTR = clicks / #impressions) for these age categories

ggplot(subset(data1, Impressions >0),aes(x=Clicks/Impressions,colour=agecat))+geom_density()
```

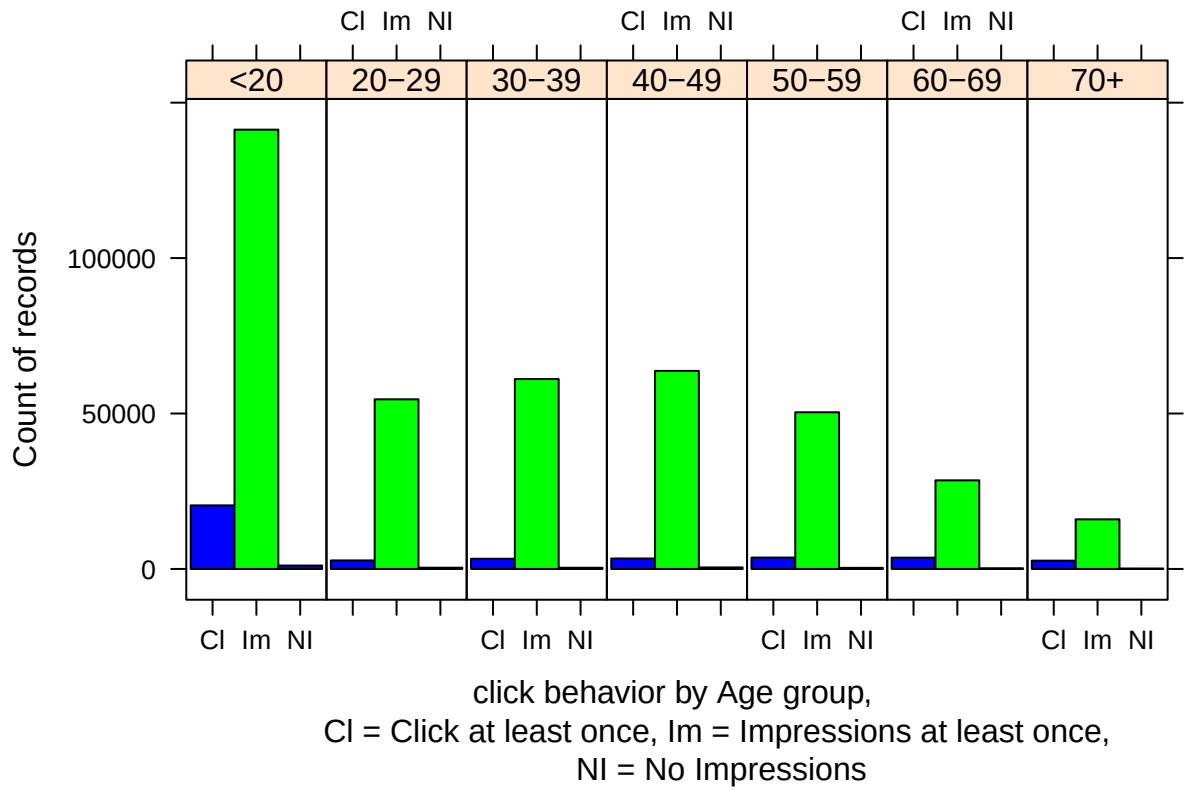



```
## NOTE: It is hard to tell the distributions by this graph.

ggplot(subset(data1, Clicks >0),aes(x=Clicks/Impressions,colour=agecat))+geom_density()
```




```
## NOTE: the distributions seem right-skewed distributions
##       if we set Clicks > 0.
```

```
# Define a new variable to segment or categorize users
# based on their click behavior.
data1$scode[data1$Impressions==0] = "NI"
data1$scode[data1$Impressions >0] = "Im"
data1$scode[data1$Clicks >0] = "Cl"

histogram(~as.factor(data1$scode) | data1$agecat,
          data=data1, type = "count",layout=c(7,1),
          col=c("blue", "green"), xlab= "click behavior by Age group,
          Cl = Click at least once, Im = Impressions at least once,
          NI = No Impressions",
          ylab ="Count of records")
```




```
data1$scode = factor(data1$scode)
head(data1)
```

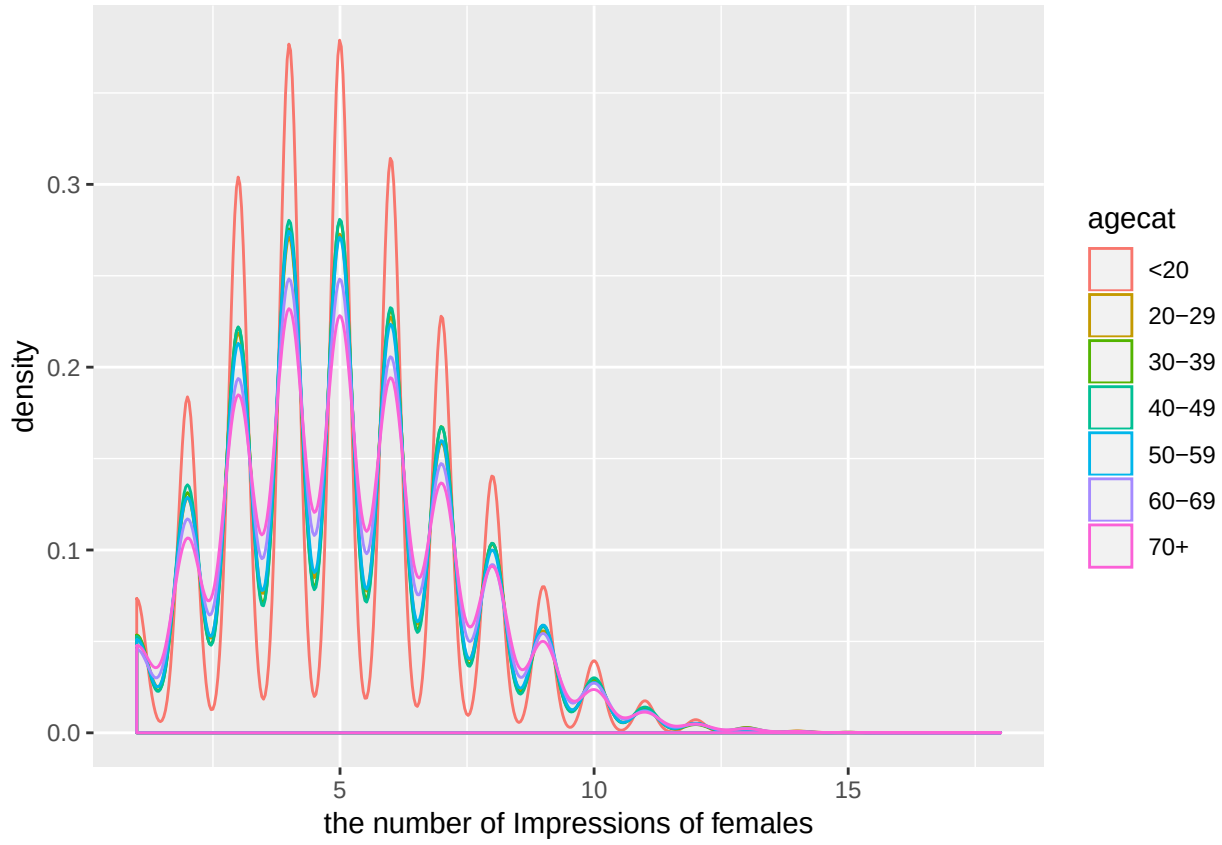
```
##   Age Gender Impressions Clicks Signed_In agecat scode
## 1  36      0           3      0         1  30-39    Im
## 2  73      1           3      0         1    70+    Im
## 3  30      0           3      0         1  30-39    Im
## 4  49      1           3      0         1  40-49    Im
## 5  47      1          11      0         1  40-49    Im
## 6  47      0          11      1         1  40-49    Cl
```

```
summary(data1)
```

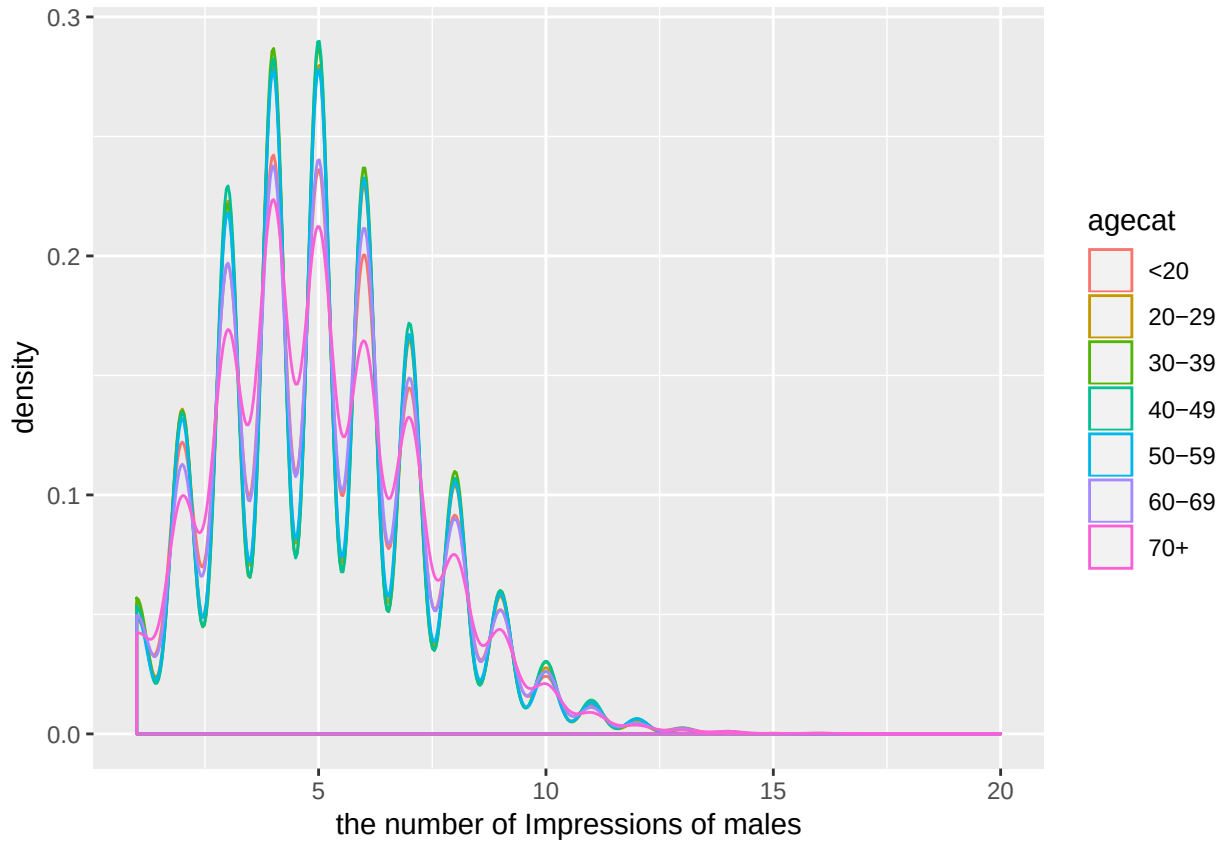
```
##       Age             Gender       Impressions         Clicks
##  Min.   :  0.00   Min.   :0.000   Min.   : 0.000   Min.   :0.00000
##  1st Qu.:  0.00   1st Qu.:0.000   1st Qu.: 3.000   1st Qu.:0.00000
##  Median : 31.00   Median :0.000   Median : 5.000   Median :0.00000
##  Mean   : 29.48   Mean   :0.367   Mean   : 5.007   Mean   :0.09259
##  3rd Qu.: 48.00   3rd Qu.:1.000   3rd Qu.: 6.000   3rd Qu.:0.00000
##  Max.   :108.00   Max.   :1.000   Max.   :20.000   Max.   :4.00000
##
##    Signed_In        agecat       scode
##  Min.   :0.0000   <20  :162867   Cl: 39838
##  1st Qu.:0.0000   20-29: 57715   Im:415537
##  Median :1.0000   30-39: 64763   NI:  3066
##  Mean   :0.7009   40-49: 67565
##  3rd Qu.:1.0000   50-59: 54406
```

```
##  Max.   :1.0000   60-69: 32358
##                   70+  : 18767
```

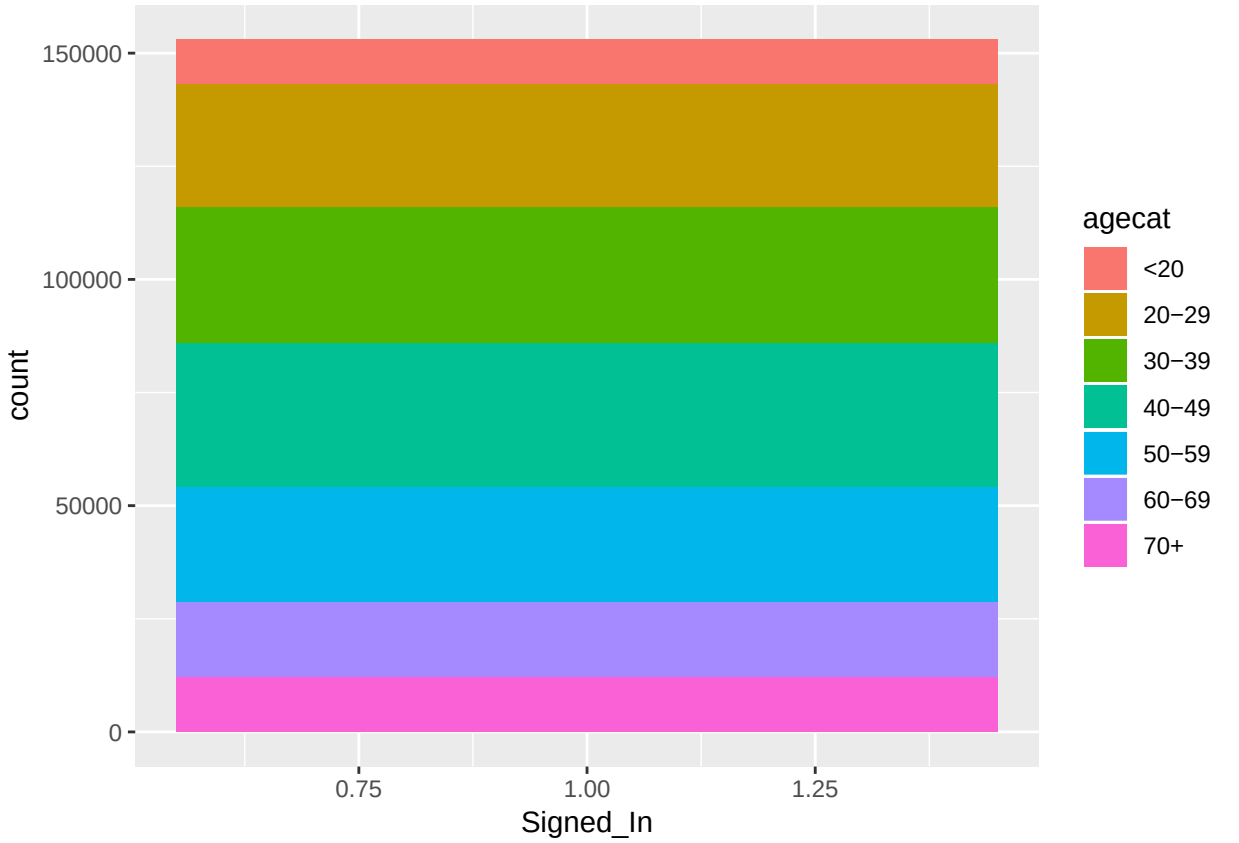
```
# Explore the data and make visual and quantitative comparisons
# across user segments/demographics
# (<20-year-old males versus <20-year-old females
# or logged-in versus not, for example).

ggplot(subset(data1, Impressions >0 & Gender=="0"),
       aes(x=Impressions,colour=agecat))+geom_density()+
  xlab("the number of Impressions of females")
```



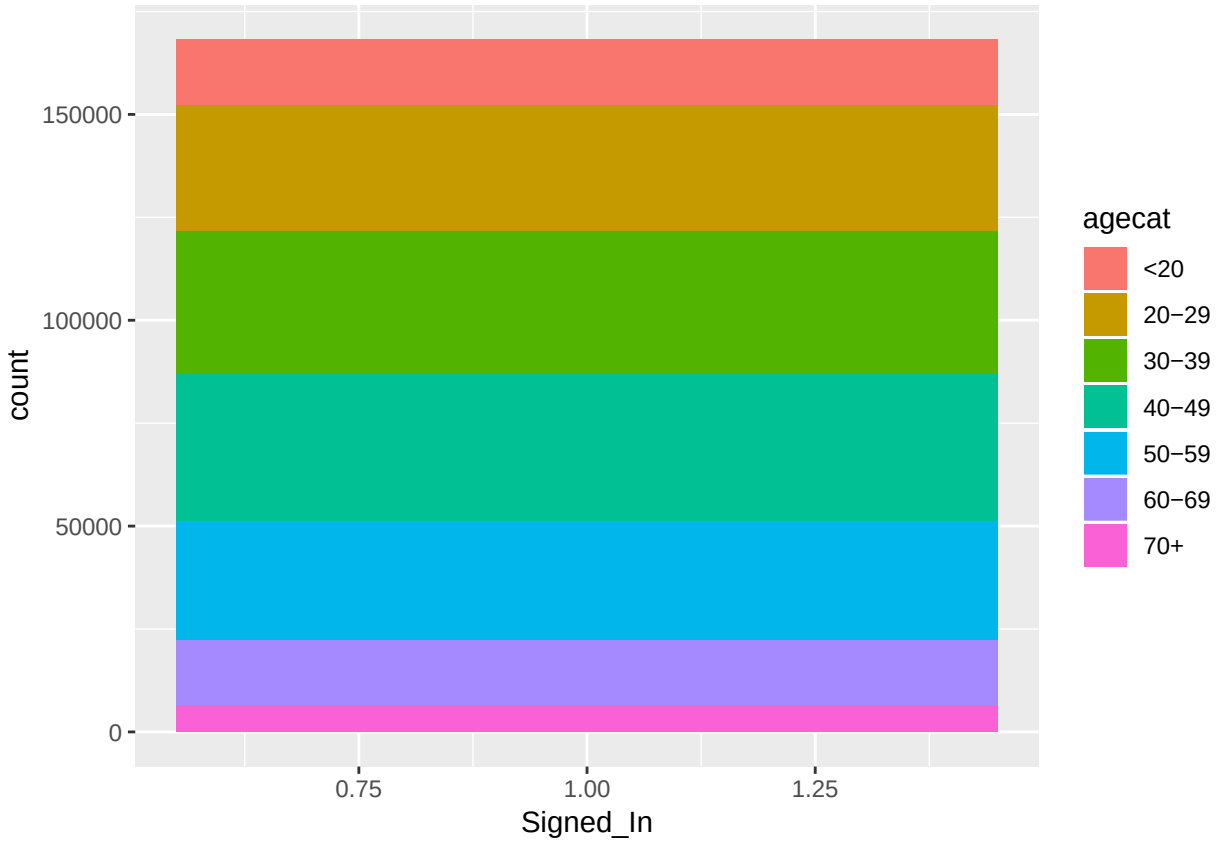

```
ggplot(subset(data1, Impressions >0 & Gender=="1"),
       aes(x=Impressions,colour=agecat))+geom_density()+
  xlab("the number of Impressions of males")
```



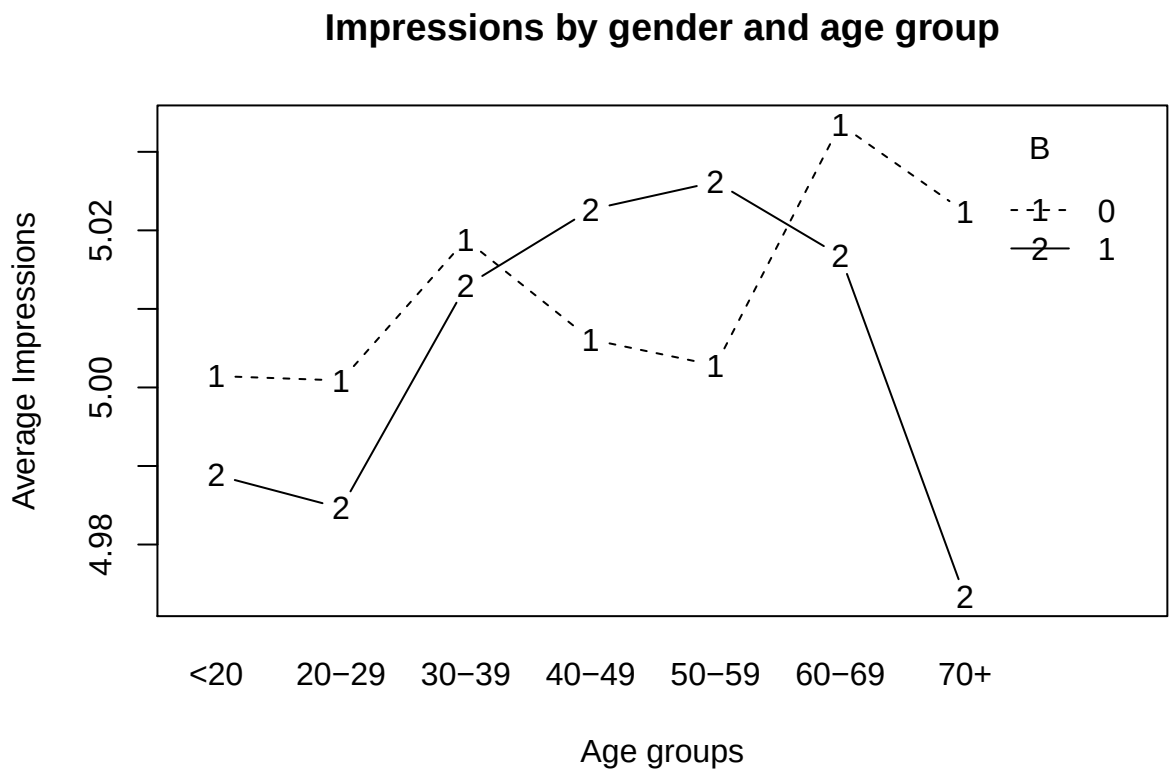

```
## NOTE: Under age 20, the kurtosis of the distribution of females is lagerer than one of males.
##       All distributions by age groups and genders are slightly right-skewed.

ggplot(subset(data1, Age>0 & Gender=="0"),
       aes(x=Signed_In, fill=agecat))+geom_bar()
```




```
ggplot(subset(data1, Age>0 & Gender=="1"),
       aes(x=Signed_In, fill=agecat))+geom_bar()
```




```
## NOTE: Under age 20, there are more males signed in the page.
##       Above age 70, there are more female users.

means = with(data1, aggregate(x=list(Y=Impressions),
                  by=list(A=agecat, B=Gender),mean))
with(means, interaction.plot(x.factor=A, trace.factor=B, response=Y, type='b',
                             main = "Impressions by gender and age group",
                             xlab = "Age groups",
                             ylab= "Average Impressions"))
```



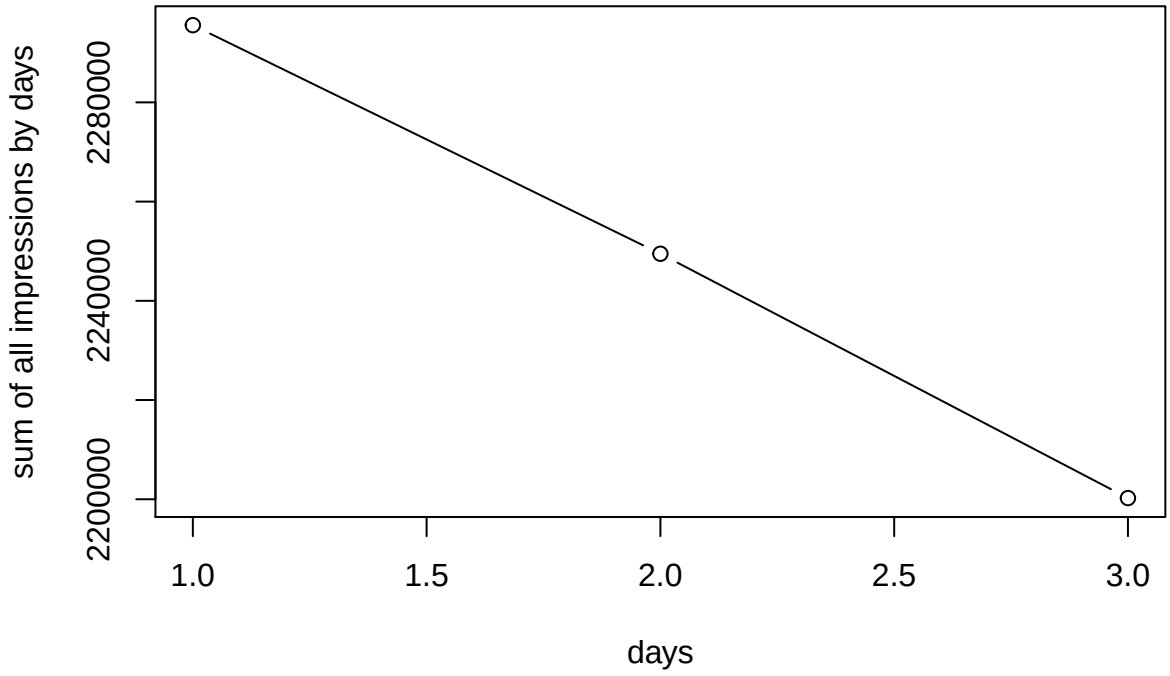

```
## NOTE: From this graph, the means of each age groups
##       between males and females are only little difference.
```

Extend your analysis across days. Visualize some metrics and distributions over time.

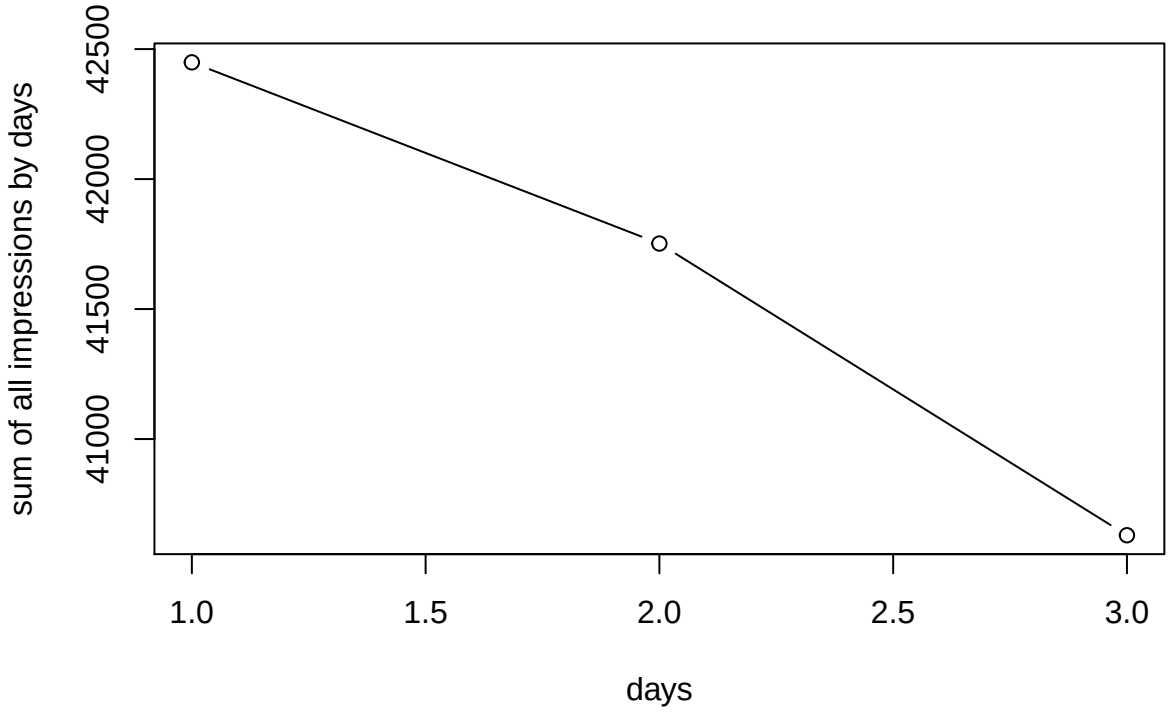
```
# combine three data set into one data frame
data1$Day = 1
data2$Day = 2
data3$Day = 3

data2$scode[data2$Impressions==0] = "NI"
data2$scode[data2$Impressions >0] = "Im"
data2$scode[data2$Clicks >0] = "Cl"

data3$scode[data3$Impressions==0] = "NI"
data3$scode[data3$Impressions >0] = "Im"
data3$scode[data3$Clicks >0] = "Cl"

alldata = rbind(data1,data2,data3)
```

```
## compare the sum of impressions, clicks, and signed in by each day
dayImp = aggregate(alldata$Impressions, by=list(Day=alldata$Day),sum)
dayCli = aggregate(alldata$Clicks, by=list(Day=alldata$Day),sum)
daySig = aggregate(alldata$Signed_In, by=list(Day=alldata$Day),sum)
```

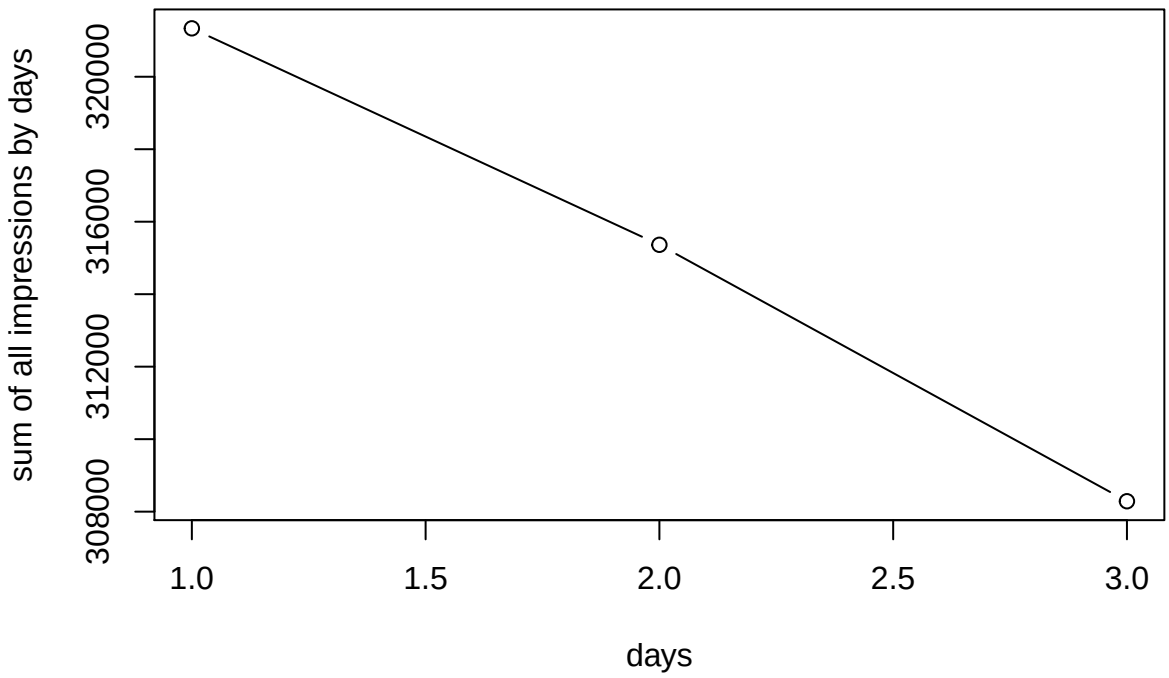
```
plot(dayImp, type="b",xlab="days",
     ylab="sum of all impressions by days")
```




```
## NOTE: Less impressions in Day 3 than other two days.

plot(dayCli, type="b",xlab="days",
     ylab="sum of all impressions by days")
```

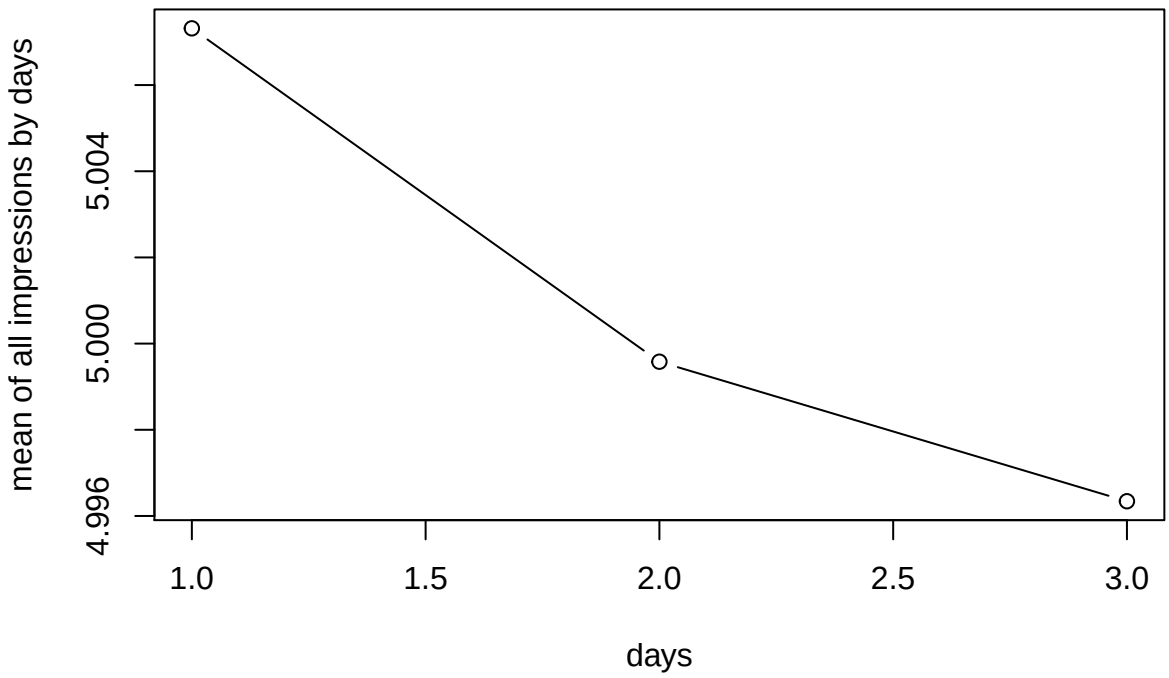



```
## NOTE: Less Clicks in Day 3 than other two days.

plot(daySig, type="b",xlab="days",
     ylab="sum of all impressions by days")
```

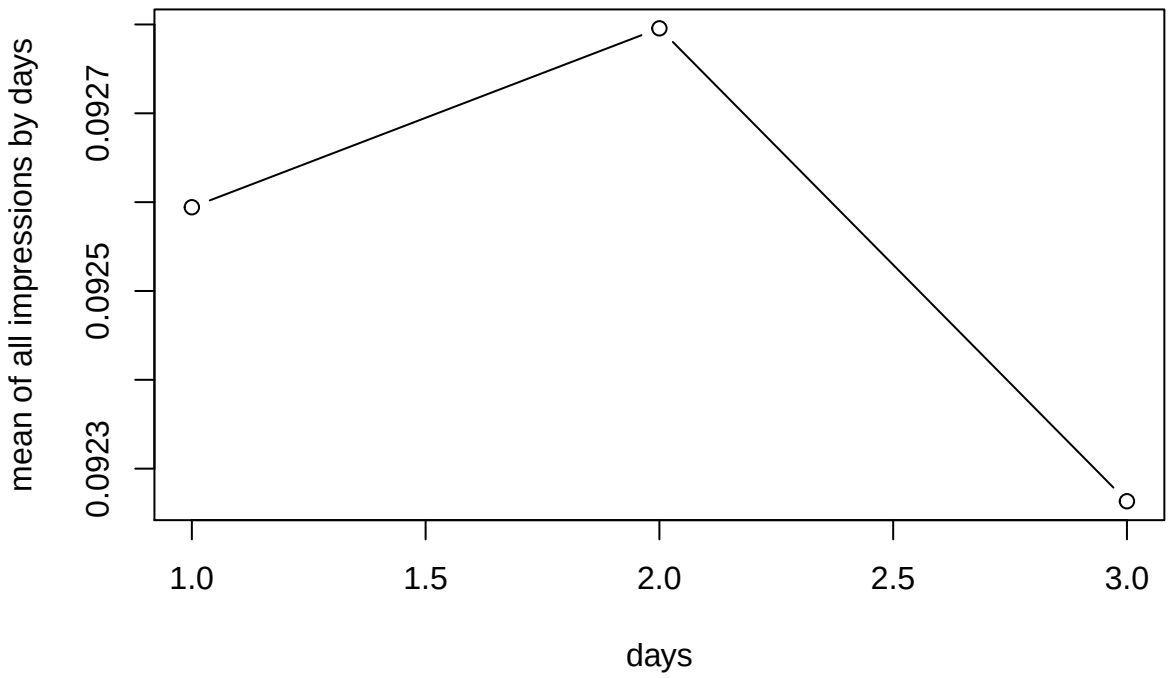



```
## NOTE: Less Clicks in Day 3 than other two days.
```

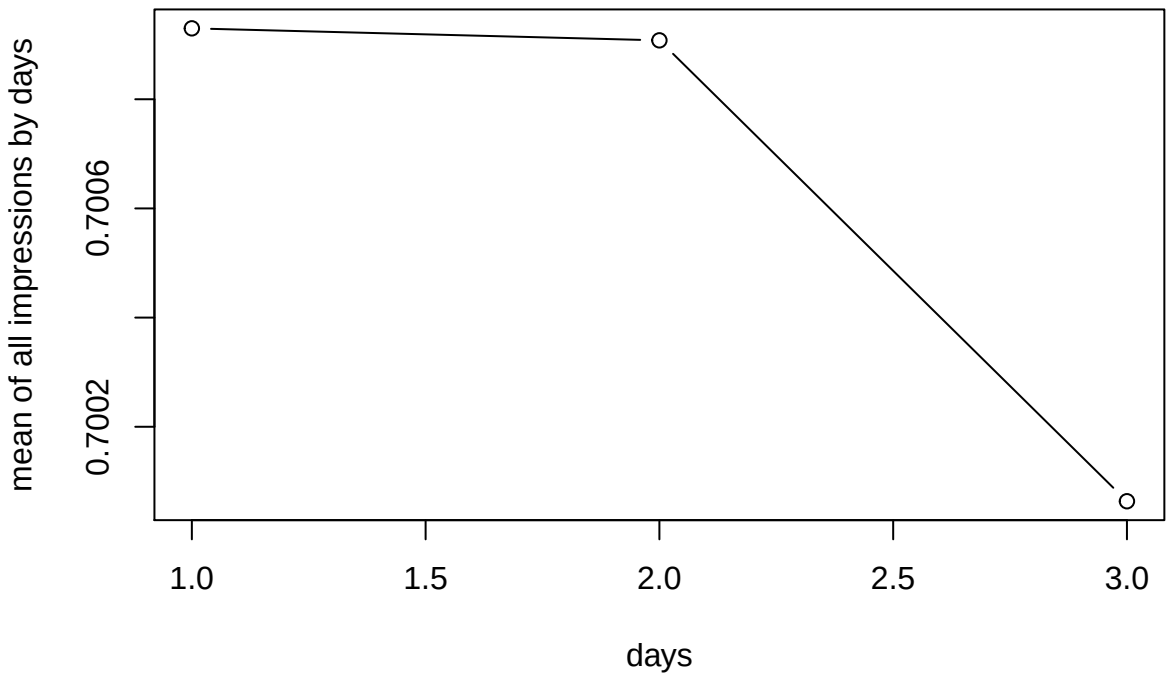
```
## compare the average of impressions, clicks, and signed in by each day
dayImp = aggregate(alldata$Impressions, by=list(Day=alldata$Day),mean)
dayCli = aggregate(alldata$Clicks, by=list(Day=alldata$Day),mean)
daySig = aggregate(alldata$Signed_In, by=list(Day=alldata$Day),mean)

plot(dayImp, type="b",xlab="days",
     ylab="mean of all impressions by days")
```




```
## NOTE: Less impressions in Day 3 than other two days.

plot(dayCli, type="b",xlab="days",
     ylab="mean of all impressions by days")
```




```
## NOTE: Less Clicks in Day 3 than other two days.

plot(daySig, type="b",xlab="days",
     ylab="mean of all impressions by days")
```




```
## NOTE: Less Clicks in Day 3 than other two days.
```