HomeWork1

Viveksinh

2/18/2020

```
install.packages("doBy")
```

Problem 1

```
rm(list = ls())

setwd('E:/STEVENS/study/FE-582/assignments/asst1/HW1_S20')
getwd()
```

```
## [1] "E:/STEVENS/study/FE-582/assignments/asst1/HW1_S20"
```

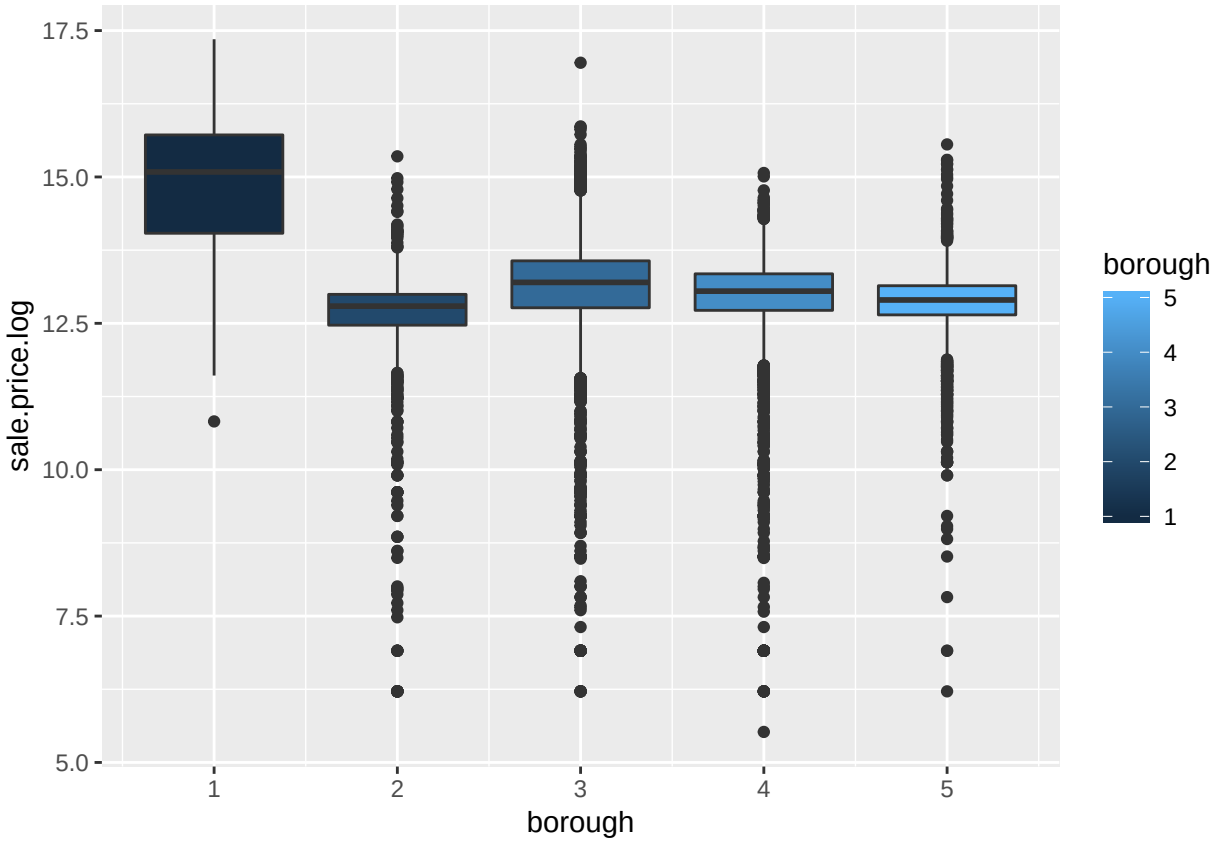
```
library('xlsx')

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
```

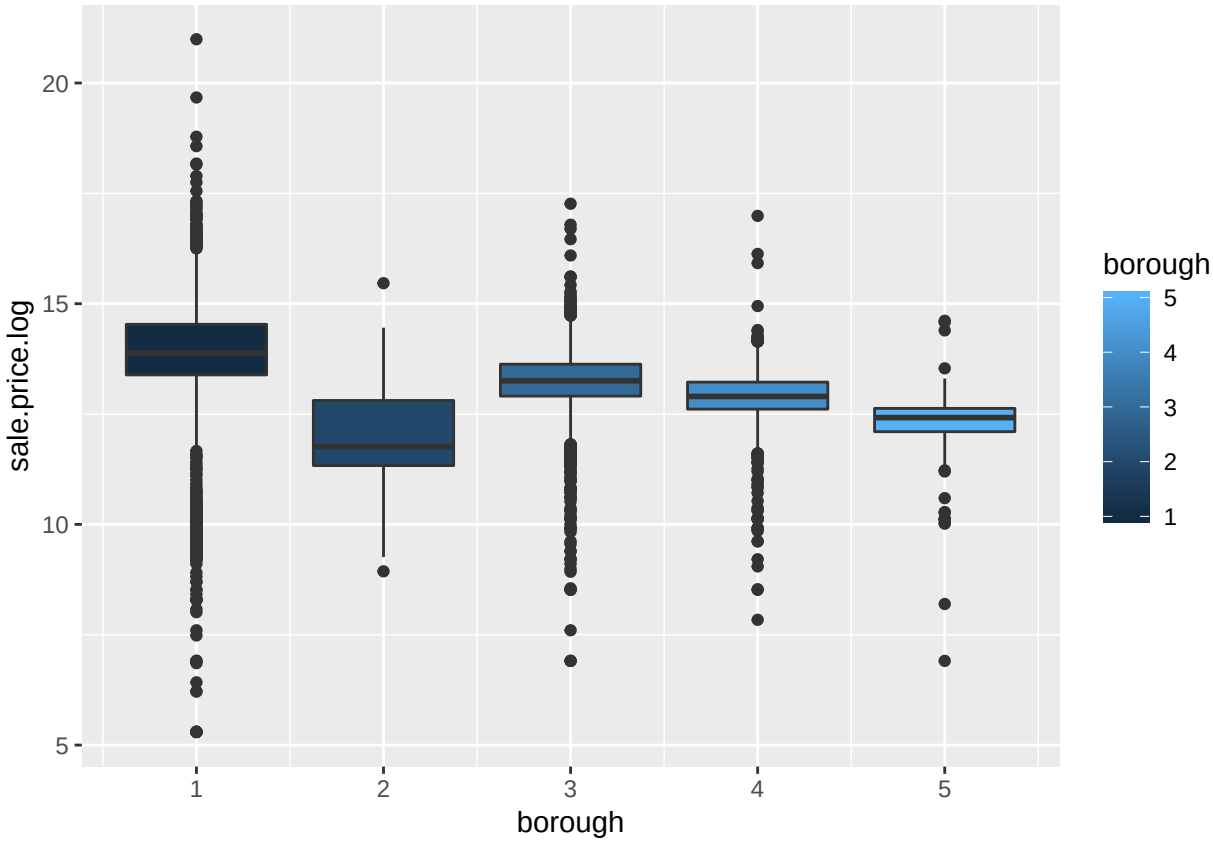
```
# Load datasets
bk <- read.xls("rollingsales_brooklyn.xls",perl = "C:\\Perl64\\bin\\perl.exe", pattern="BOROUGH")
bx <- read.xls("rollingsales_bronx.xls",perl = "C:\\Perl64\\bin\\perl.exe", pattern="BOROUGH")
manh <- read.xls("rollingsales_manhattan.xls",perl = "C:\\Perl64\\bin\\perl.exe", pattern="BOROUGH")
qn <- read.xls("rollingsales_queens.xls",perl = "C:\\Perl64\\bin\\perl.exe", pattern="BOROUGH")
si <- read.xls("rollingsales_statenisland.xls",perl = "C:\\Perl64\\bin\\perl.exe", pattern="BOROUGH")

# convert column names to lowercase
names(bk) <- tolower(names(bk))
names(bx) <- tolower(names(bx))
```

```r
names(manh) <- tolower(names(manh))
names(qn) <- tolower(names(qn))
names(si) <- tolower(names(si))

# Format

# Brooklyn
bk$sale.price.n <- as.numeric(gsub("[^[:digit:]]","",bk$sale.price))

bk$gross.sqft <- as.numeric(gsub("[^[:digit:]]","",bk$gross.square.feet))

bk$land.sqft <- as.numeric(gsub("[^[:digit:]]","",bk$land.square.feet))

bk$sale.date <- as.Date(bk$sale.date)

bk$year.built <- as.numeric(as.character(bk$year.built))

# Bronx
bx$sale.price.n <- as.numeric(gsub("[^[:digit:]]","",bx$sale.price))

bx$gross.sqft <- as.numeric(gsub("[^[:digit:]]","",bx$gross.square.feet))

bx$land.sqft <- as.numeric(gsub("[^[:digit:]]","",bx$land.square.feet))

bx$sale.date <- as.Date(bx$sale.date)

bx$year.built <- as.numeric(as.character(bx$year.built))

# Manhattan
manh$sale.price.n <- as.numeric(gsub("[^[:digit:]]","",manh$sale.price))

manh$gross.sqft <- as.numeric(gsub("[^[:digit:]]","",manh$gross.square.feet))

manh$land.sqft <- as.numeric(gsub("[^[:digit:]]","",manh$land.square.feet))

manh$sale.date <- as.Date(manh$sale.date)

manh$year.built <- as.numeric(as.character(manh$year.built))

# Queens
qn$sale.price.n <- as.numeric(gsub("[^[:digit:]]","",qn$sale.price))

qn$gross.sqft <- as.numeric(gsub("[^[:digit:]]","",qn$gross.square.feet))

qn$land.sqft <- as.numeric(gsub("[^[:digit:]]","",qn$land.square.feet))

qn$sale.date <- as.Date(qn$sale.date)

qn$year.built <- as.numeric(as.character(qn$year.built))

# StatenIsland
si$sale.price.n <- as.numeric(gsub("[^[:digit:]]","",si$sale.price))
```

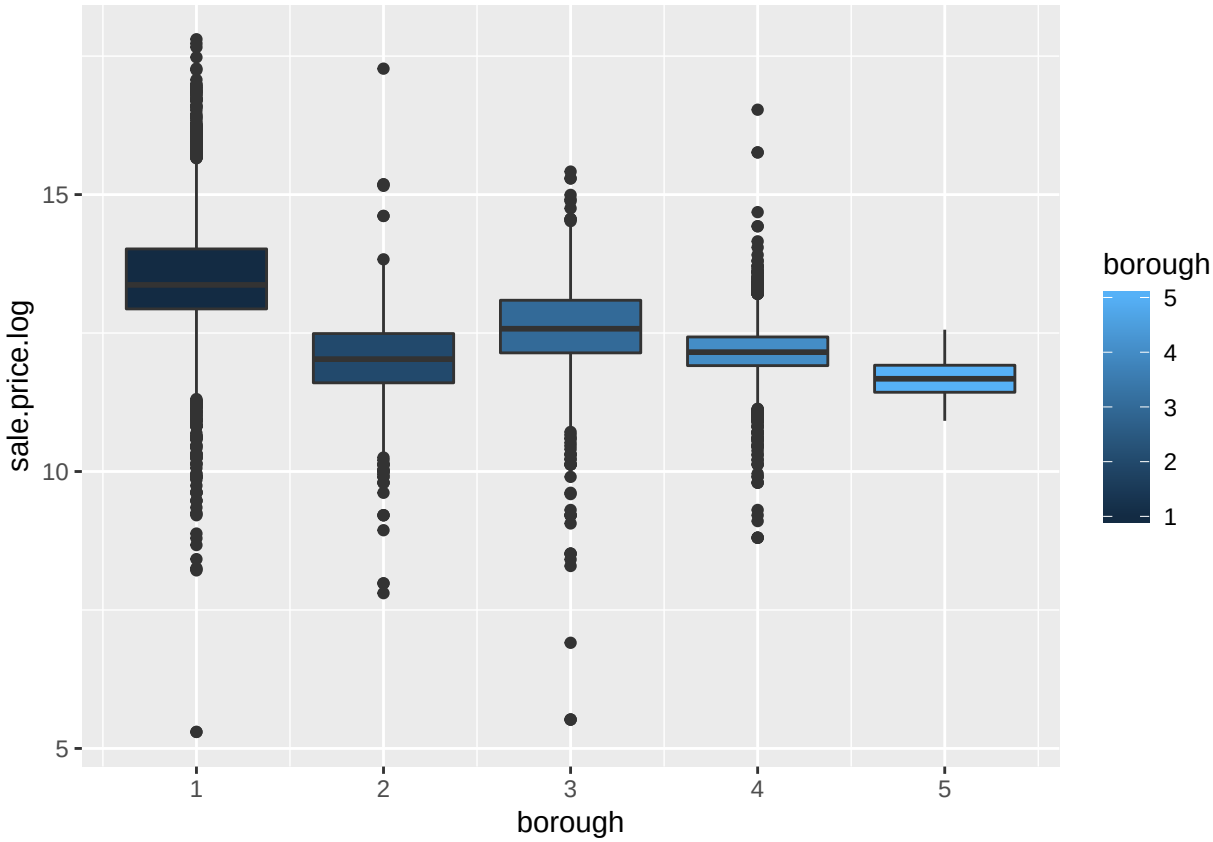
```r
si$gross.sqft <- as.numeric(gsub("[^[:digit:]]","",si$gross.square.feet))

si$land.sqft <- as.numeric(gsub("[^[:digit:]]","",si$land.square.feet))

si$sale.date <- as.Date(si$sale.date)

si$year.built <- as.numeric(as.character(si$year.built))

# Clean: remove records which don't have sale price(=0$)
bk <- bk[bk$sale.price.n!=0,]
bx <- bx[bx$sale.price.n!=0,]
manh <- manh[manh$sale.price.n!=0,]
qn <- qn[qn$sale.price.n!=0,]
si <- si[si$sale.price.n!=0,]

# remove outliers
bk$sale.price.log <- log(bk$sale.price.n)
bk <- bk[bk$sale.price.log > 5, ]

bx$sale.price.log <- log(bx$sale.price.n)
bx <- bx[bx$sale.price.log > 5, ]

manh$sale.price.log <- log(manh$sale.price.n)
manh <- manh[manh$sale.price.log > 5, ]

qn$sale.price.log <- log(qn$sale.price.n)
qn <- qn[qn$sale.price.log > 5, ]

si$sale.price.log <- log(si$sale.price.n)
si <- si[si$sale.price.log > 5, ]

## Comparison and analysis across boroughs


## Family
homes_borough_1 <- bk[which(grepl("FAMILY",bk$building.class.category)),]
homes_borough_2 <- bx[which(grepl("FAMILY",bx$building.class.category)),]
homes_borough_3 <- manh[which(grepl("FAMILY",manh$building.class.category)),]
homes_borough_4 <- qn[which(grepl("FAMILY",qn$building.class.category)),]
homes_borough_5 <- si[which(grepl("FAMILY",si$building.class.category)),]

final_homes_table <- rbind(homes_borough_1, homes_borough_2, homes_borough_3, homes_borough_4, homes_bo

library("ggplot2")
ggplot(final_homes_table,aes(x=borough, y=sale.price.log, fill=borough, group=borough))+geom_boxplot()
```



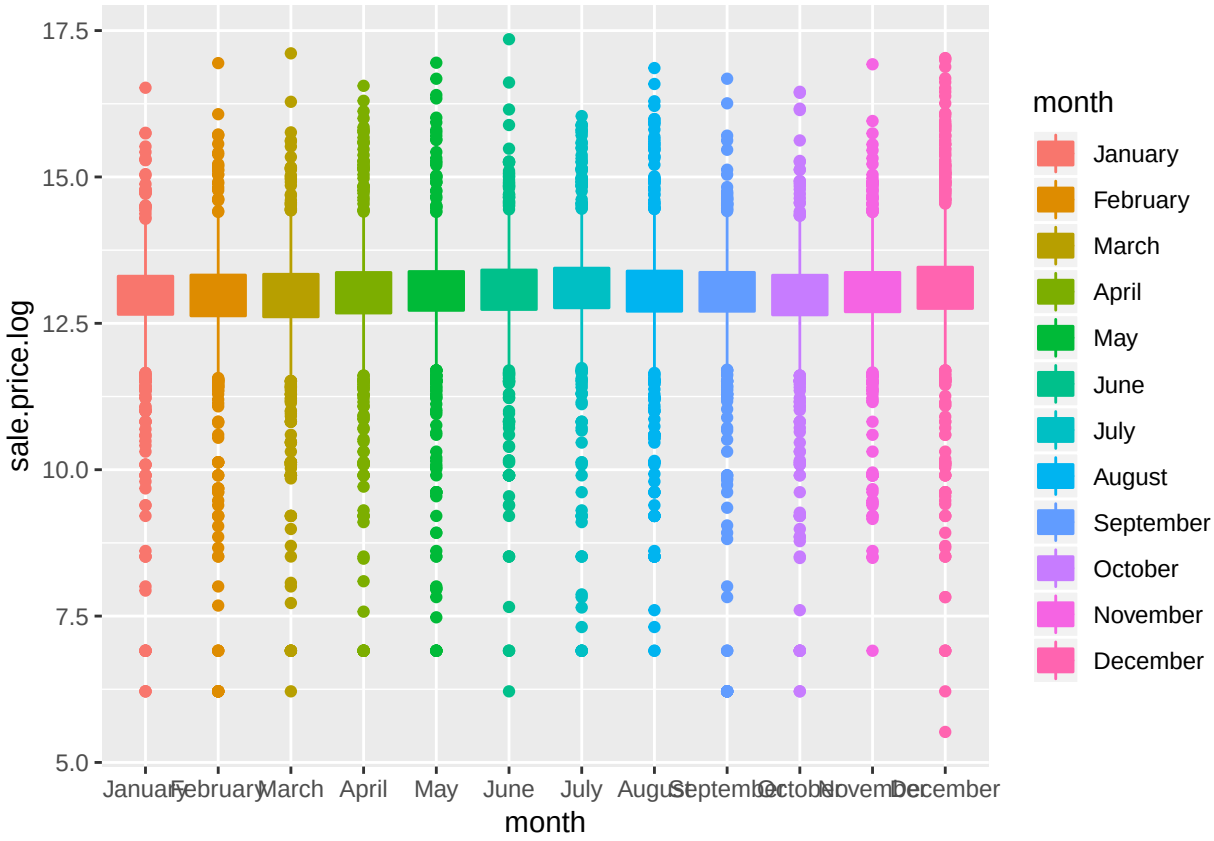

```
## Condos
condos_borough_1 <- bk[which(grepl("CONDOS",bk$building.class.category)),]
condos_borough_2 <- bx[which(grepl("CONDOS",bx$building.class.category)),]
condos_borough_3 <- manh[which(grepl("CONDOS",manh$building.class.category)),]
condos_borough_4 <- qn[which(grepl("CONDOS",qn$building.class.category)),]
condos_borough_5 <- si[which(grepl("CONDOS",si$building.class.category)),]

final_condos_table <- rbind(condos_borough_1, condos_borough_2, condos_borough_3, condos_borough_4, cond


ggplot(final_condos_table,aes(x=borough, y=sale.price.log, fill=borough, group=borough))+geom_boxplot()
```



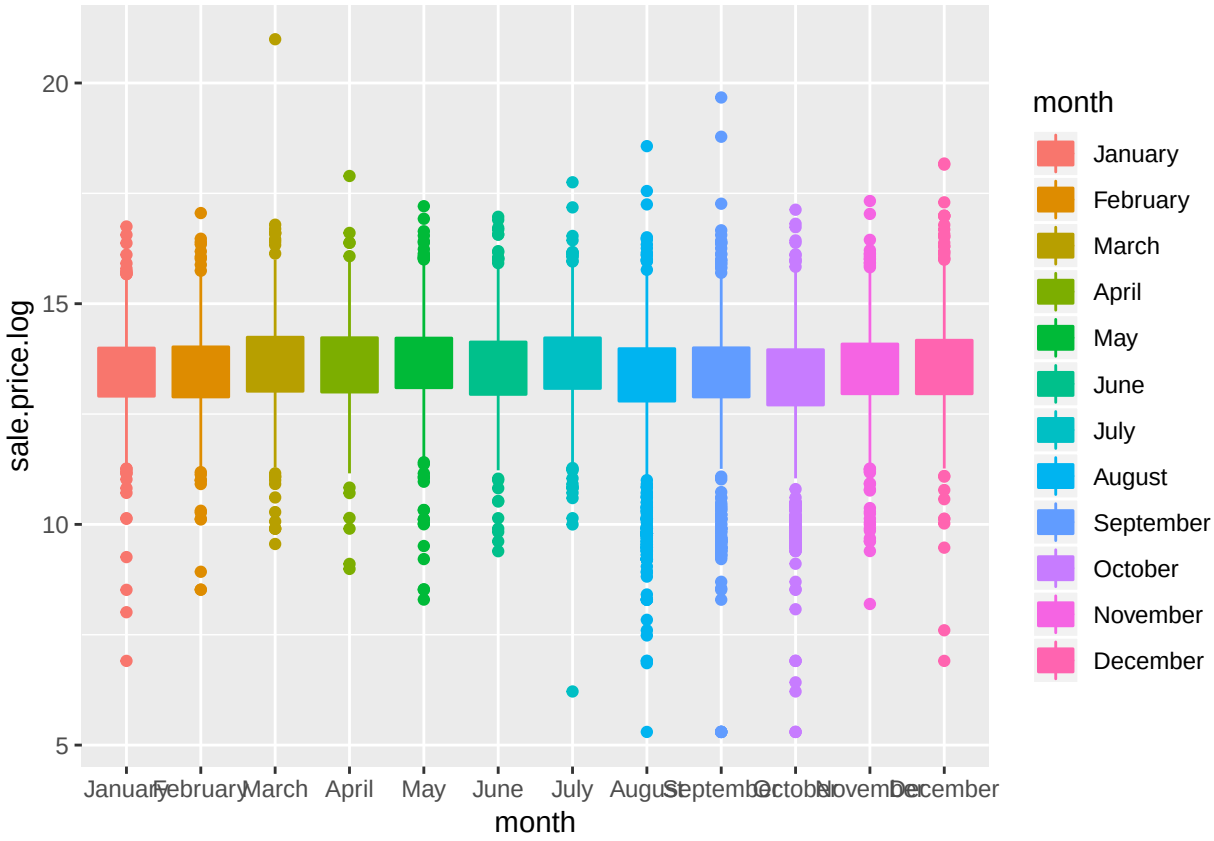

```
## Coops
coops_borough_1 <- bk[which(grepl("COOPS",bk$building.class.category)),]
coops_borough_2 <- bx[which(grepl("COOPS",bx$building.class.category)),]
coops_borough_3 <- manh[which(grepl("COOPS",manh$building.class.category)),]
coops_borough_4 <- qn[which(grepl("COOPS",qn$building.class.category)),]
coops_borough_5 <- si[which(grepl("COOPS",si$building.class.category)),]

final_coops_table <- rbind(coops_borough_1, coops_borough_2, coops_borough_3, coops_borough_4, coops_bo

ggplot(final_coops_table,aes(x=borough, y=sale.price.log, fill=borough, group=borough))+geom_boxplot()
```



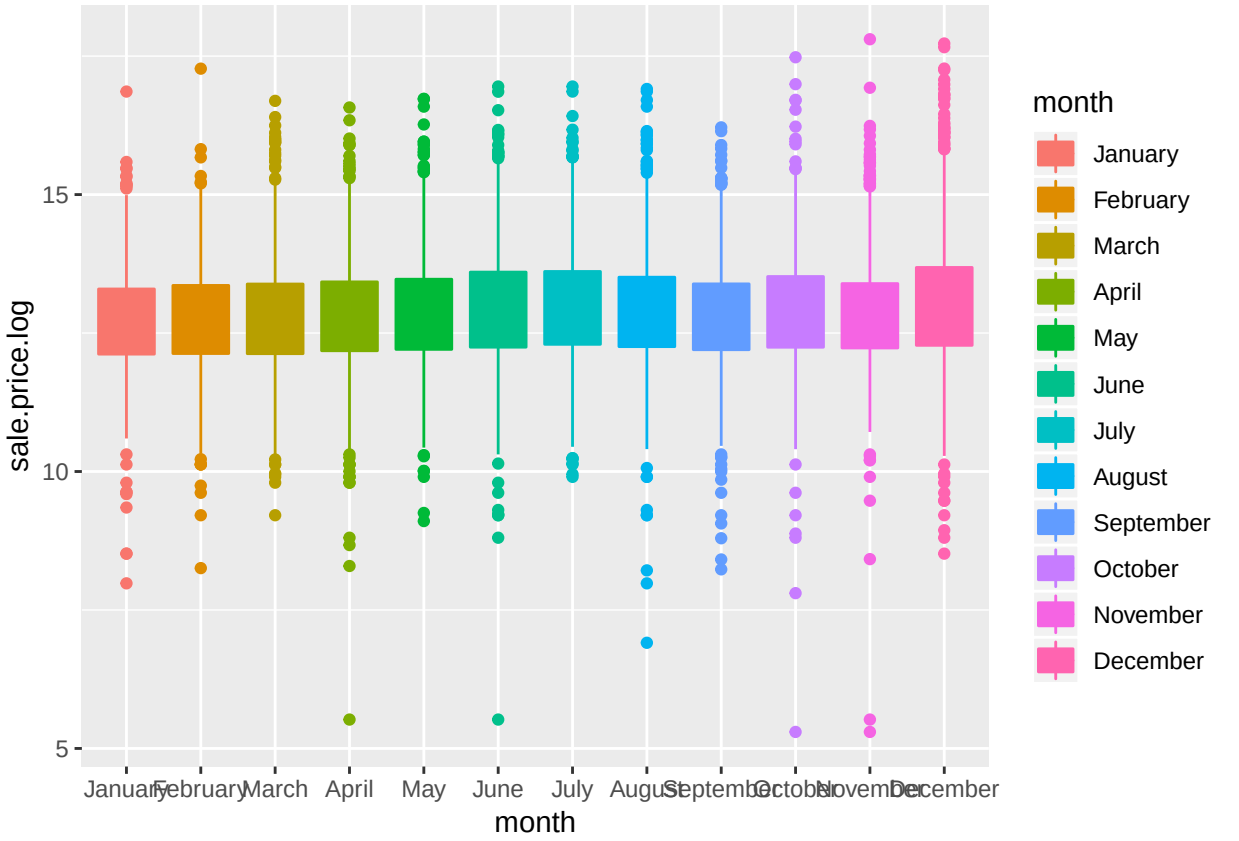

```
## Comparison and analysis across time

# convert date format to days and months
days_month <- function(df){
  df$day <- format(df$sale.date, "%A")
  df$day <- factor(df$day, levels = c("Monday", 'Tuesday',
                                      "Wednesday", "Thursday", "Friday"))

  df$month <- format(df$sale.date, "%B")
  df$month <- factor(df$month, levels = c("January", "February", "March",
                                          "April", "May", "June", "July", "August",
                                          "September", "October", "November",
                                          "December"))

  return(df)
}

final_homes_table_time_transformation <- days_month(final_homes_table)
final_condos_table_time_transformation <- days_month(final_condos_table)
final_coops_table_time_transformation <- days_month(final_coops_table)

## Family homes
ggplot(final_homes_table_time_transformation,aes(x=month, y=sale.price.log,
                                                 fill=month, group=month, colour=month)) + geom_boxplo
```




```
## Condos
ggplot(final_condos_table_time_transformation,aes(x=month, y=sale.price.log,
                                                  fill=month, group=month, colour=month)) + geom_boxplo
```




```
## Coops
ggplot(final_coops_table_time_transformation,aes(x=month, y=sale.price.log,
                                                 fill=month, group=month, colour=month)) + geom_boxplot
```




```
library("doBy")

siterange <- function(x){c(length(x),mean(x),median(x))}


## Summary statistics across boroughs


# Family homes
summaryBy(borough+sale.price+gross.sqft~building.class.category, data=final_homes_table, FUN=siterange)
```

```
##                       building.class.category borough.FUN1 borough.FUN2
## 1 01  ONE FAMILY HOMES                                 9433     3.861126
## 2 02  TWO FAMILY HOMES                                 7995     3.426517
## 3 03  THREE FAMILY HOMES                               2166     3.091874
##   borough.FUN3 sale.price.FUN1 sale.price.FUN2 sale.price.FUN3
## 1            4            9433        2302.324            1875
## 2            3            7995        2330.967            2082
## 3            3            2166        2185.910            2082
##   gross.sqft.FUN1 gross.sqft.FUN2 gross.sqft.FUN3
## 1            9433        1710.353            1520
## 2            7995        2327.915            2200
## 3            2166        3124.282            3060
```

```
# Condos
summaryBy(borough+sale.price+gross.sqft~building.class.category, data=final_condos_table, FUN=siterange)
```

```
##                       building.class.category borough.FUN1 borough.FUN2
## 1 04  TAX CLASS 1 CONDOS                               853     3.808910
## 2 12  CONDOS - WALKUP APARTMENTS                       719     3.061196
## 3 13  CONDOS - ELEVATOR APARTMENTS                    9138     1.850843
## 4 15  CONDOS - 2-10 UNIT RESIDENTIAL                   883     2.408834
## 5 16  CONDOS - 2-10 UNIT WITH COMMERCIAL UNIT           48     1.645833
## 6 28  COMMERCIAL CONDOS                                590     1.427119
##   borough.FUN3 sale.price.FUN1 sale.price.FUN2 sale.price.FUN3
## 1            4             853        2360.098            1681
## 2            3             719        2034.790            1725
## 3            1            9138        2582.424            2324
## 4            3             883        2440.909            2223
## 5            1              48        2307.167            1425
## 6            1             590        3856.912            4759
##   gross.sqft.FUN1 gross.sqft.FUN2 gross.sqft.FUN3
## 1             853               0               0
## 2             719               0               0
## 3            9138               0               0
## 4             883               0               0
## 5              48               0               0
## 6             590               0               0
```

```
# Coops
summaryBy(borough+sale.price+gross.sqft~building.class.category, data=final_coops_table, FUN=siterange)
```

```
##                       building.class.category borough.FUN1 borough.FUN2
## 1 09  COOPS - WALKUP APARTMENTS                       2367     2.800591
## 2 10  COOPS - ELEVATOR APARTMENTS                    12514     2.095493
##   borough.FUN3 sale.price.FUN1 sale.price.FUN2 sale.price.FUN3
## 1            3            2367        1956.845            1313
## 2            1           12514        2075.166            1550
##   gross.sqft.FUN1 gross.sqft.FUN2 gross.sqft.FUN3
## 1            2367        303.2467               0
## 2           12514        424.0642               0
```

```
## Summary statistics across time
# Family homes
summaryBy(sale.price+gross.sqft~month, data=final_homes_table_time_transformation, FUN=siterange)
```

```
##        month sale.price.FUN1 sale.price.FUN2 sale.price.FUN3
## 1    January            1498        2330.566            1990
## 2   February            1452        2314.731            1926
## 3      March            1583        2291.379            1949
## 4      April            1632        2272.080            1949
## 5        May            1861        2337.026            2057
## 6       June            1747        2291.684            2005
## 7       July            1475        2268.840            2055
## 8     August            1965        2279.734            2005
## 9  September            1508        2373.910            2025
## 10   October            1444        2335.124            1963
## 11  November            1410        2273.032            1990
## 12  December            2019        2261.208            1967
##    gross.sqft.FUN1 gross.sqft.FUN2 gross.sqft.FUN3
## 1             1498        2065.942          1880.5
## 2             1452        2108.914          1944.0
## 3             1583        2110.350          1936.0
```

```
## 4             1632        2127.472          1960.0
## 5             1861        2139.499          1960.0
## 6             1747        2075.653          1920.0
## 7             1475        2103.929          1923.0
## 8             1965        2151.002          1944.0
## 9             1508        2089.117          1920.0
## 10            1444        2054.288          1890.0
## 11            1410        2119.741          1941.0
## 12            2019        2228.621          2000.0
```

```
# Condos
summaryBy(sale.price+gross.sqft~month, data=final_condos_table_time_transformation, FUN=siterange)
```

```
##        month sale.price.FUN1 sale.price.FUN2 sale.price.FUN3
## 1    January             847        2626.290          2210.0
## 2   February             750        2509.296          2103.0
## 3      March             841        2688.014          2323.0
## 4      April            1022        2545.070          2242.5
## 5        May            1093        2451.008          2252.0
## 6       June            1220        2366.601          2082.0
## 7       July            1061        2438.437          2223.0
## 8     August            1433        2745.590          2324.0
## 9  September             912        2674.525          2270.0
## 10   October             966        2668.768          2247.0
## 11  November             866        2661.416          2290.0
## 12  December            1220        2654.170          2253.0
##    gross.sqft.FUN1 gross.sqft.FUN2 gross.sqft.FUN3
## 1              847               0               0
## 2              750               0               0
## 3              841               0               0
## 4             1022               0               0
## 5             1093               0               0
## 6             1220               0               0
## 7             1061               0               0
## 8             1433               0               0
## 9              912               0               0
## 10             966               0               0
## 11             866               0               0
## 12            1220               0               0
```

```
# Coops
summaryBy(sale.price+gross.sqft~month, data=final_coops_table_time_transformation, FUN=siterange)
```

```
##        month sale.price.FUN1 sale.price.FUN2 sale.price.FUN3
## 1    January            1044        2044.966          1500.5
## 2   February             954        2014.828          1429.5
## 3      March            1020        2025.698          1444.5
## 4      April            1104        2056.538          1481.0
## 5        May            1395        2040.356          1495.0
## 6       June            1414        2047.463          1592.0
## 7       July            1539        2081.932          1592.0
## 8     August            1915        2006.051          1514.0
## 9  September            1007        1998.734          1481.0
## 10   October            1076        2089.680          1550.0
## 11  November            1013        2125.953          1559.0
## 12  December            1400        2146.330          1587.5
```

```
##    gross.sqft.FUN1 gross.sqft.FUN2 gross.sqft.FUN3
## 1             1044       793.74138               0
## 2              954         0.00000               0
## 3             1020      1573.27647               0
## 4             1104       434.80525               0
## 5             1395        41.97849               0
## 6             1414       341.94484               0
## 7             1539       348.84990               0
## 8             1915       407.56606               0
## 9             1007       484.77756               0
## 10            1076       480.73699               0
## 11            1013         0.00000               0
## 12            1400       175.86357               0
```

Conclusion

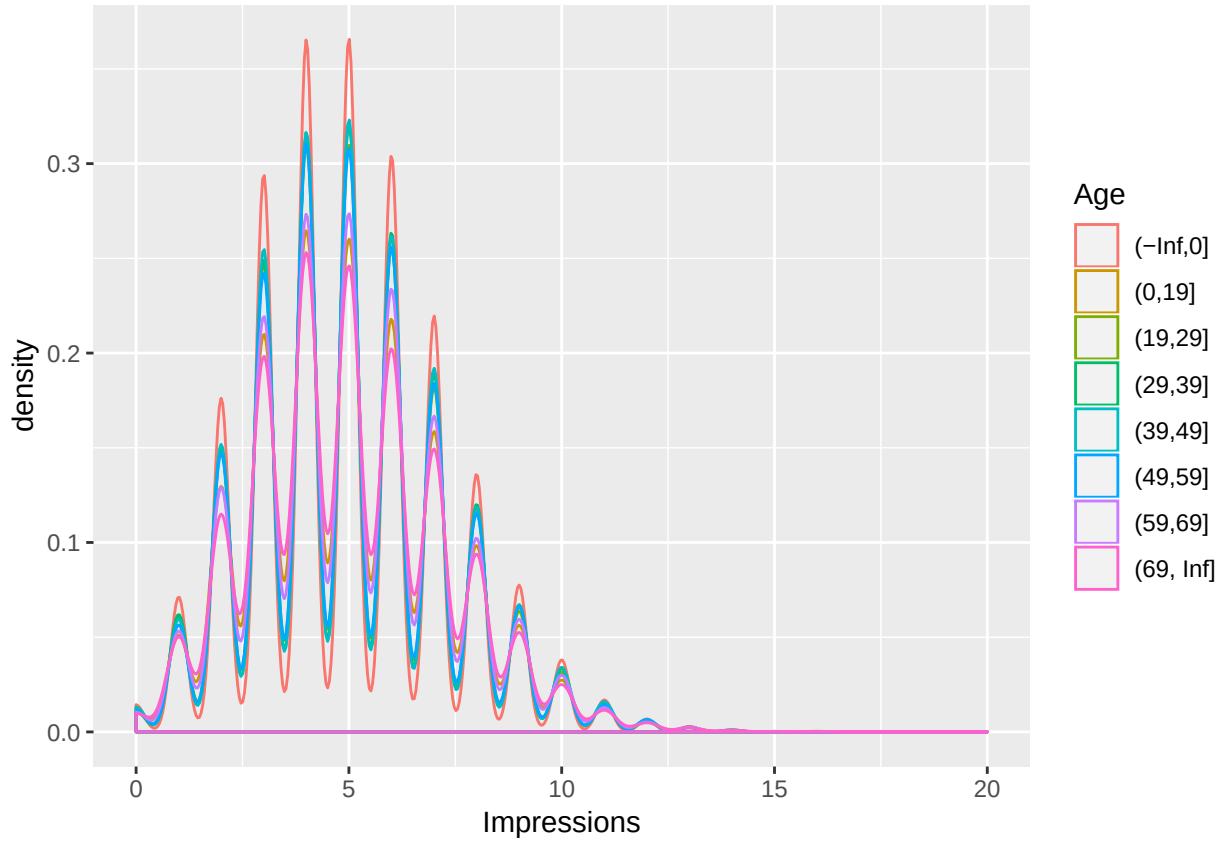
- As compared to other boroughs, sale prices are the highest in brooklyn
- By analysing boroughs over time, we can see that sale prices don't change much throught a year
- From summary stats, we can notice that for family homes, smaller the home, higher the price.

Problem 2

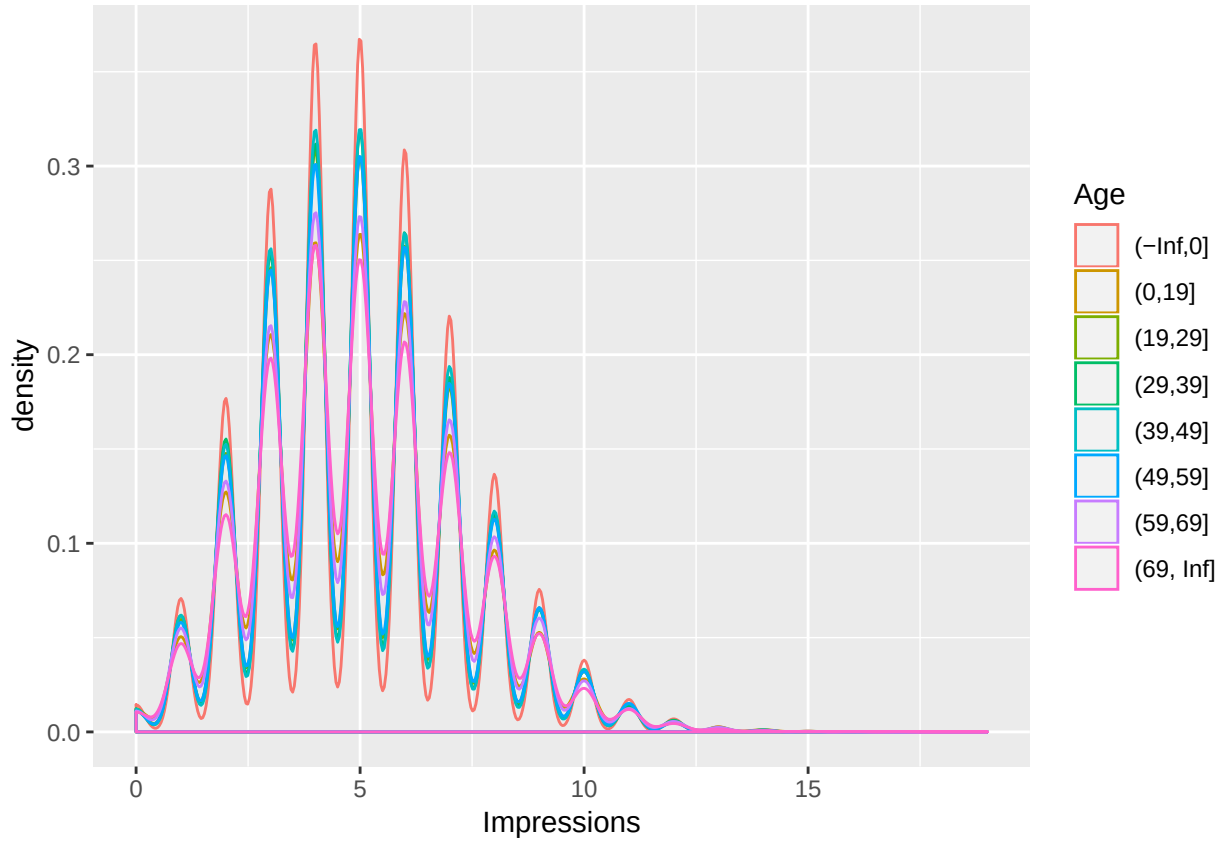
```
setwd('E:/STEVENS/study/FE-582/assignments/asst1/HW1_S20')
getwd()
```

```
## [1] "E:/STEVENS/study/FE-582/assignments/asst1/HW1_S20"
```

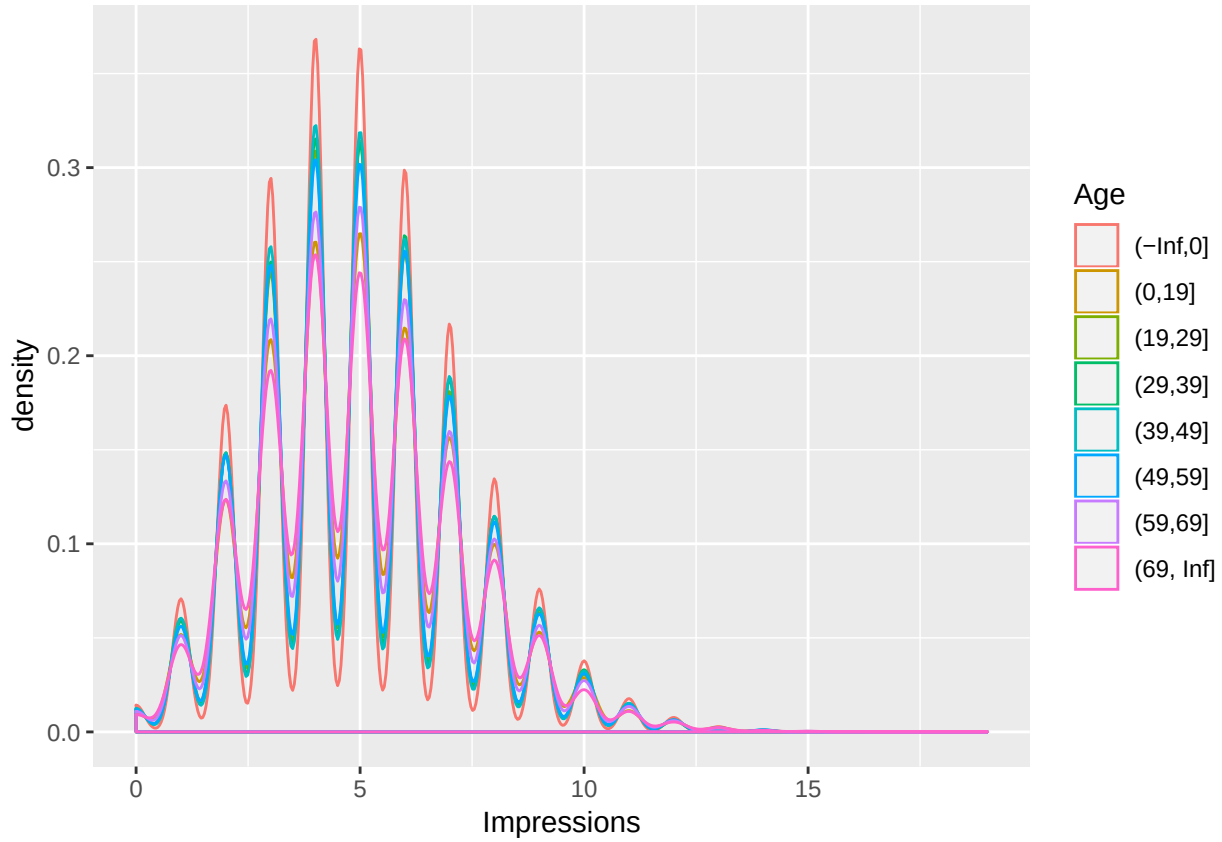
```
day1 <- data.frame(read.csv("nyt1.csv"))
day2 <- data.frame(read.csv("nyt2.csv"))
day3 <- data.frame(read.csv("nyt3.csv"))

day1$age_group <- cut(day1$Age, c(-Inf,0,19,29,39,49,59,69,Inf))
day2$age_group <- cut(day2$Age, c(-Inf,0,19,29,39,49,59,69,Inf))
day3$age_group <- cut(day3$Age, c(-Inf,0,19,29,39,49,59,69,Inf))

day1$ctr <- day1$Clicks / day1$Impressions
day2$ctr <- day2$Clicks / day2$Impressions
day3$ctr <- day3$Clicks / day3$Impressions

library(ggplot2)

##day1: Impressions across age groups
ggplot(day1, aes(x=Impressions, color=age_group)) + geom_density() + xlab("Impressions") + labs(color='
```



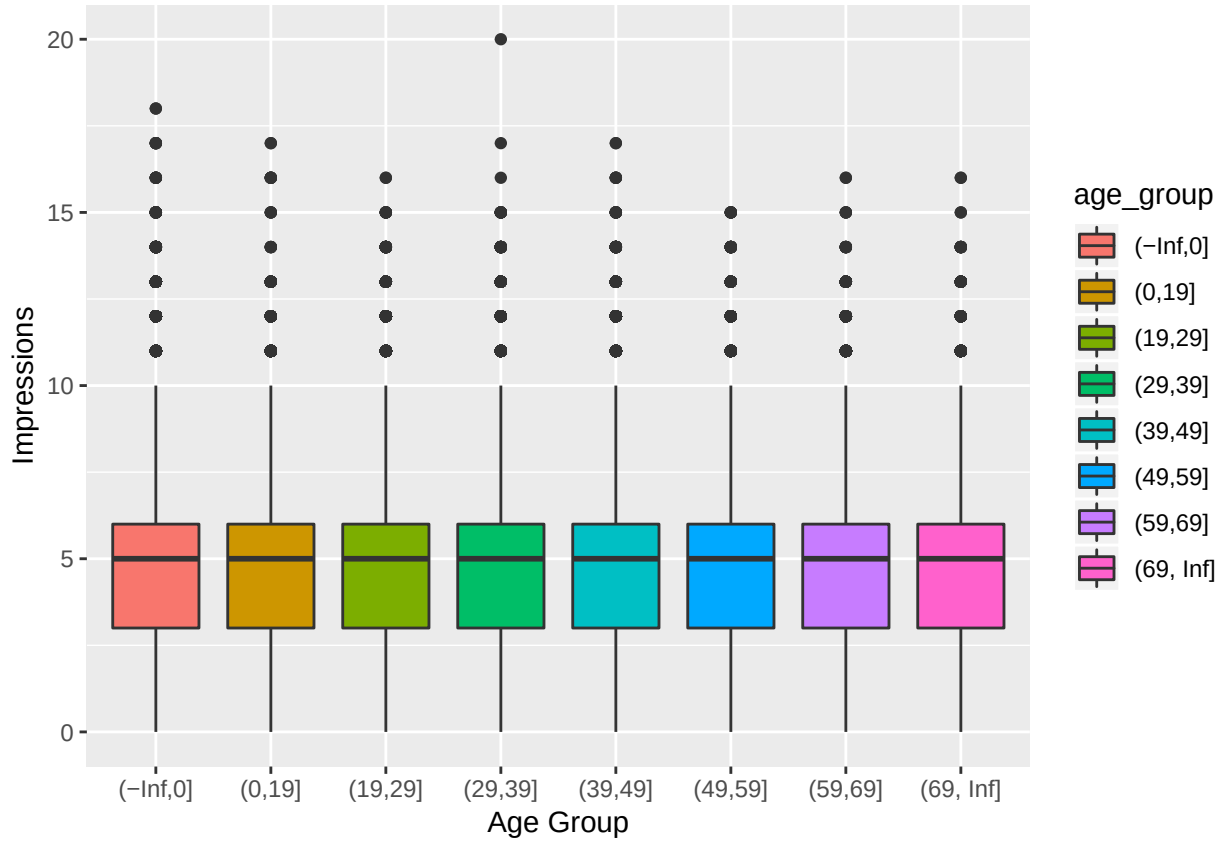

```
##day2: Impressions across age groups
ggplot(day2, aes(x=Impressions, color=age_group)) + geom_density() + xlab("Impressions") + labs(color='
```



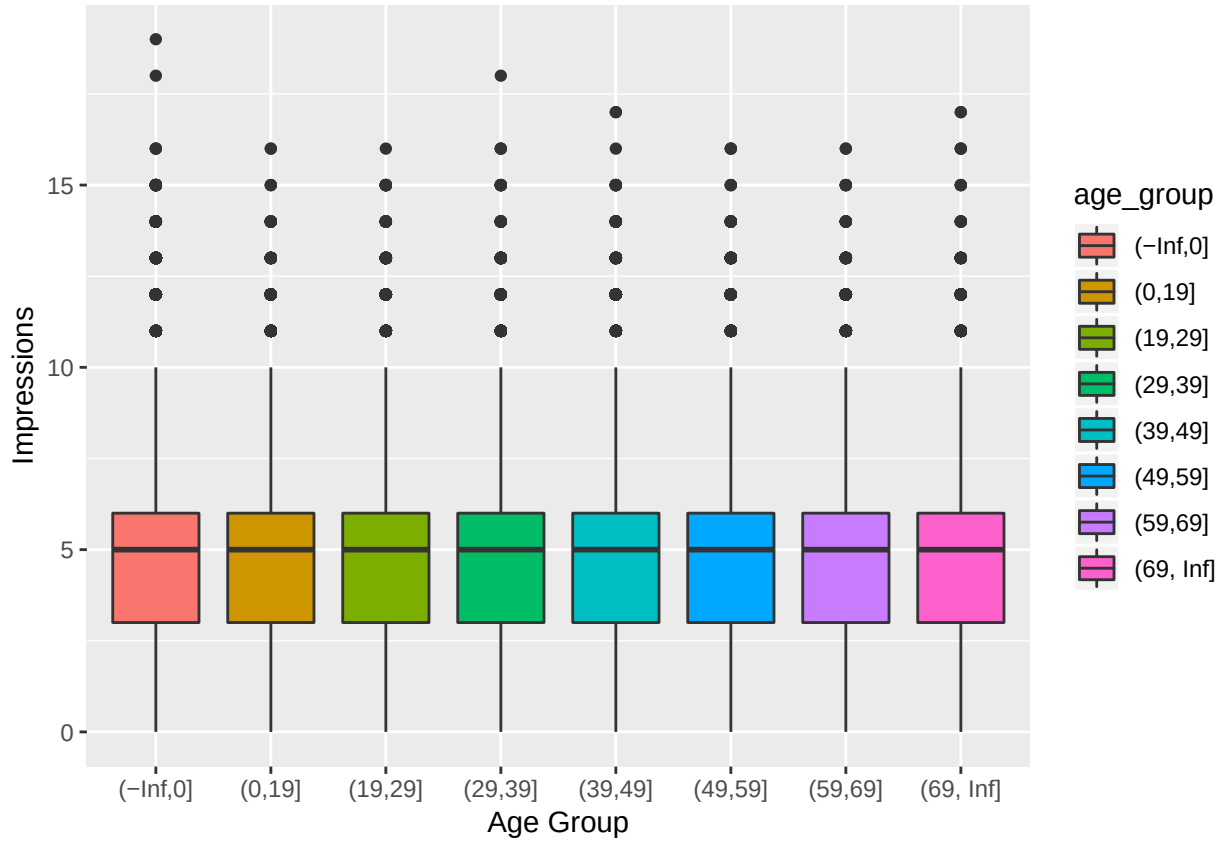

```
##day3: Impressions across age groups
ggplot(day3, aes(x=Impressions, color=age_group)) + geom_density() + xlab("Impressions") + labs(color='
```



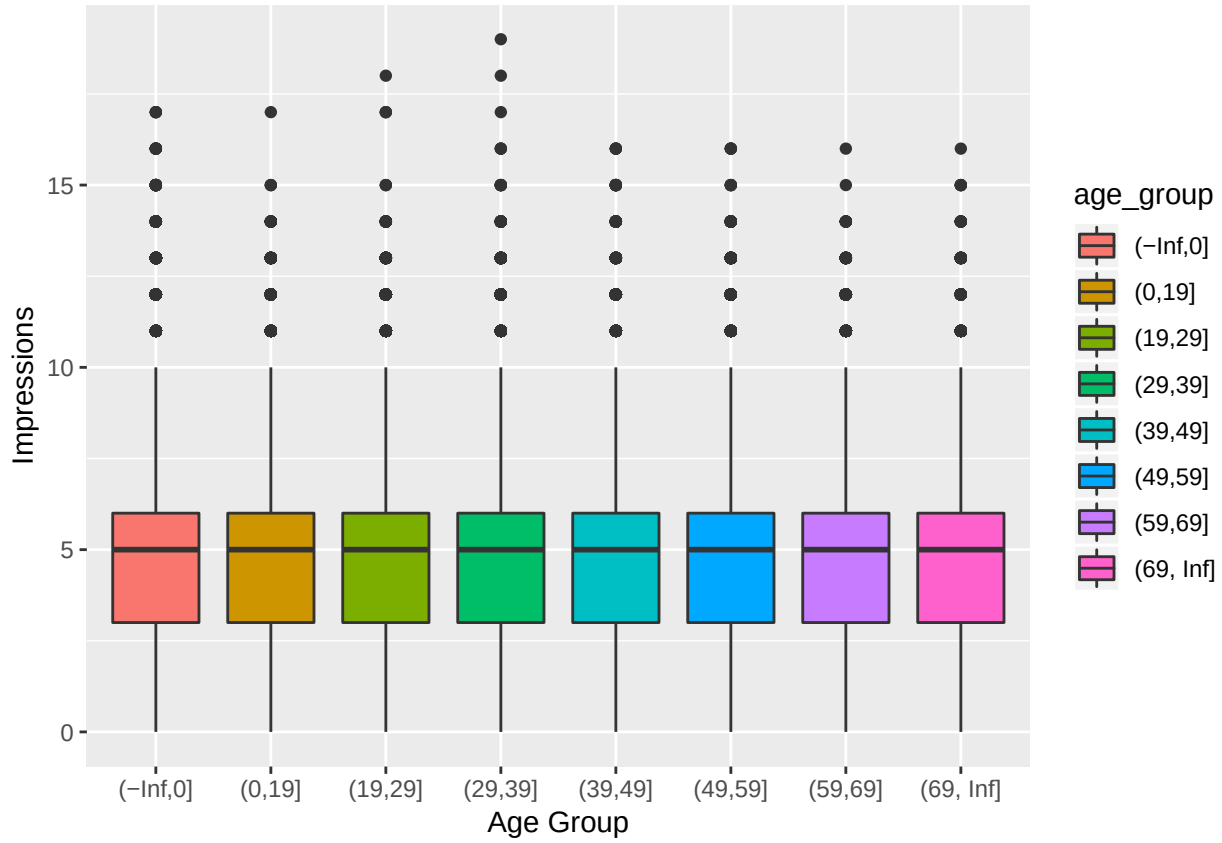

```
day1$click_group[day1$Clicks == 0] <- '0 clicks'
day1$click_group[day1$Clicks > 0] <- '> 0 clicks'

day2$click_group[day2$Clicks == 0] <- '0 clicks'
day2$click_group[day2$Clicks > 0] <- '> 0 clicks'

day3$click_group[day3$Clicks == 0] <- '0 clicks'
day3$click_group[day3$Clicks > 0] <- '> 0 clicks'

# comparison across age groups

##day1
ggplot(day1, aes(x=age_group, y=Impressions, fill=age_group)) +
  geom_boxplot() + xlab("Age Group")
```



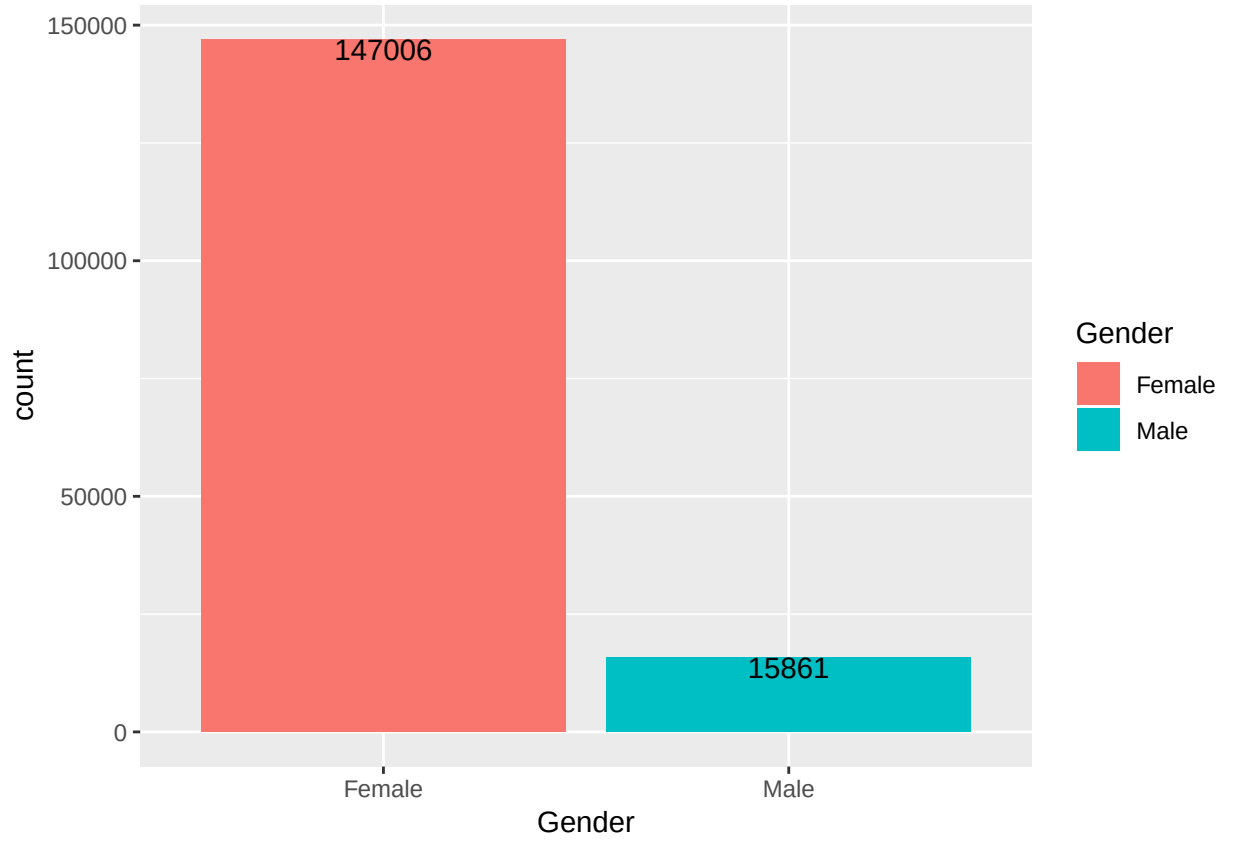

```
##day2
ggplot(day2, aes(x=age_group, y=Impressions, fill=age_group)) +
  geom_boxplot() + xlab("Age Group")
```



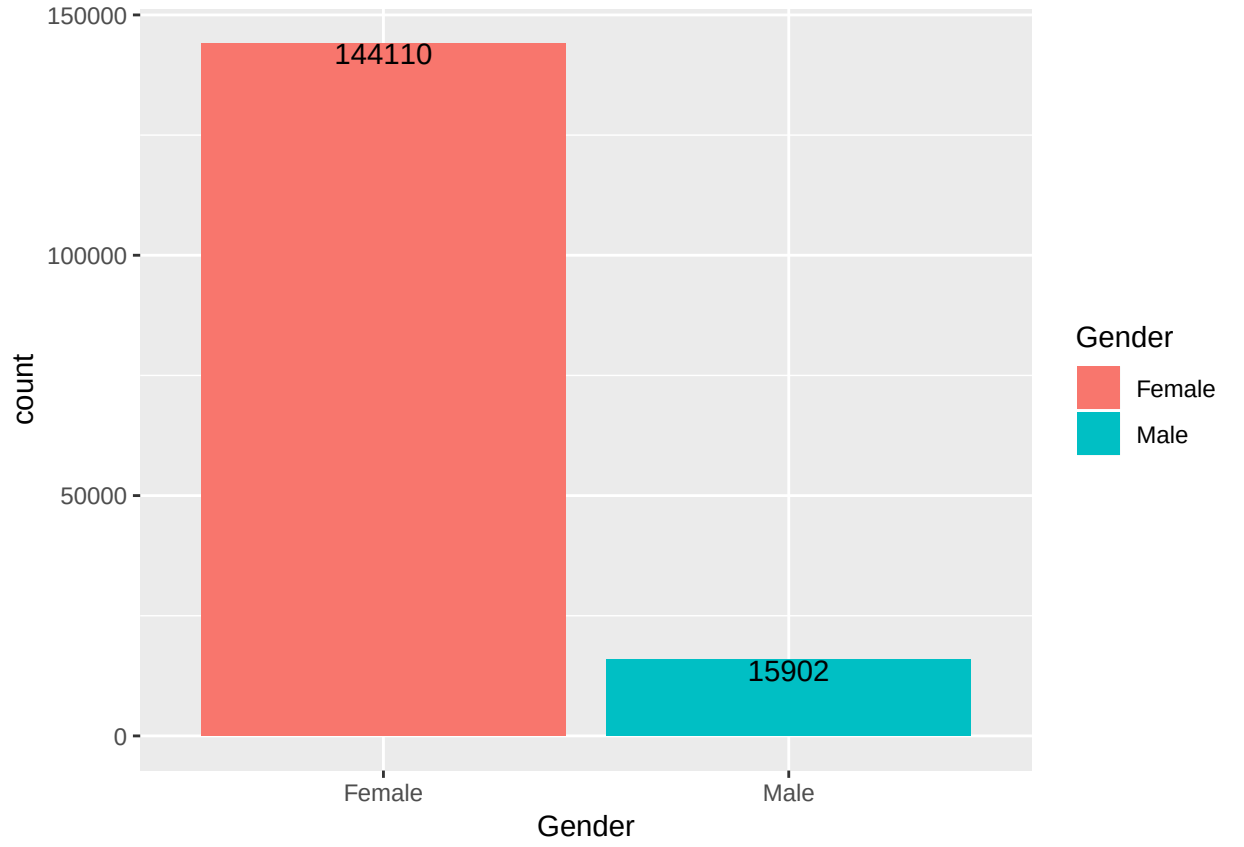

```
##day3
ggplot(day3, aes(x=age_group, y=Impressions, fill=age_group)) +
  geom_boxplot() + xlab("Age Group")
```



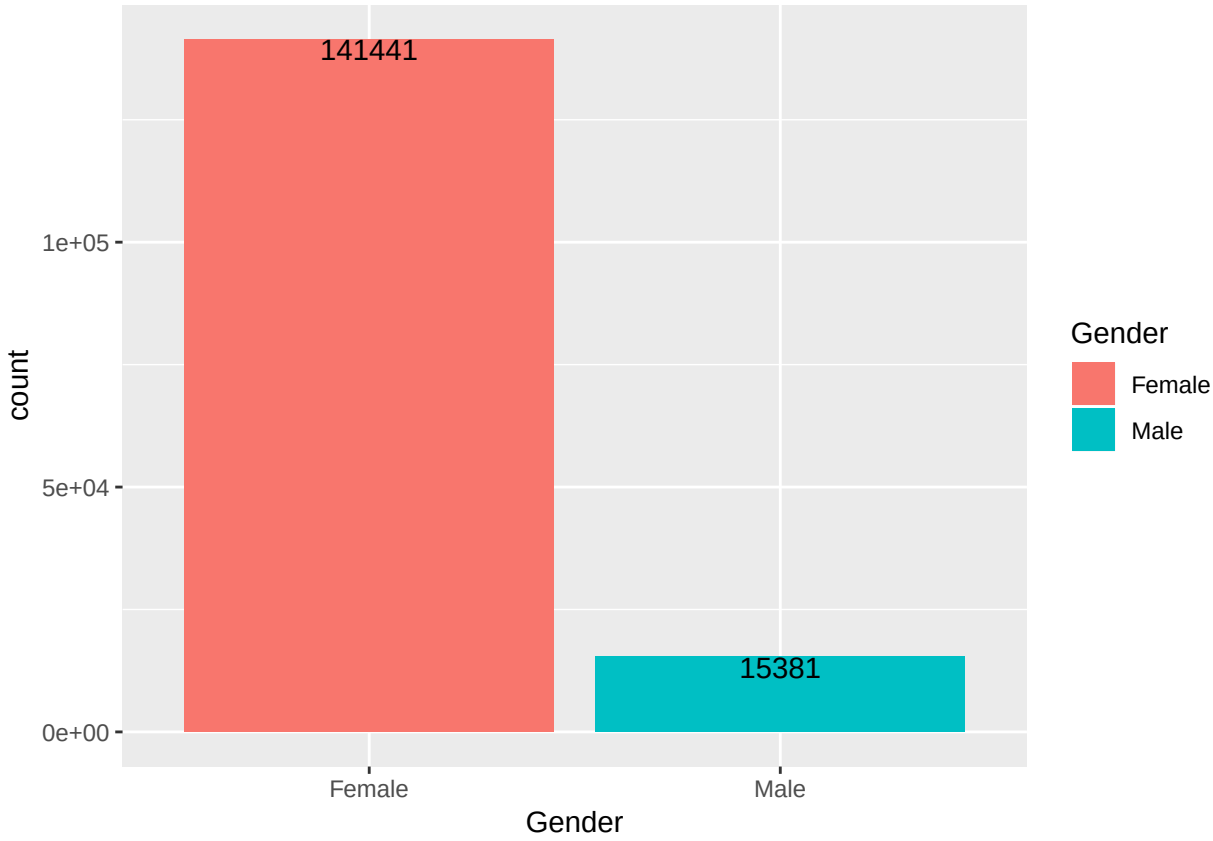

```r
# data transformation

to_category <- function(df){
  df$Gender[df$Gender == 1] <- 'Male'
  df$Gender[df$Gender == 0] <- 'Female'
  df$Signed_In[df$Signed_In == 1] <- 'logged_in'
  df$Signed_In[df$Signed_In == 0] <- 'not logged_in'
  return(df)
}

day1 <- to_category(day1)
day2 <- to_category(day2)
day3 <- to_category(day3)

## comparison across user segments:

# i) <20 years old males vs <20 years old females :
day1_subset1 <- subset(day1, Age<20)
day2_subset1 <- subset(day2, Age<20)
day3_subset1 <- subset(day3, Age<20)

##day1
ggplot(day1_subset1, aes(x=Gender, y=..count..)) +
  geom_bar(aes(fill=Gender)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Gender')
```



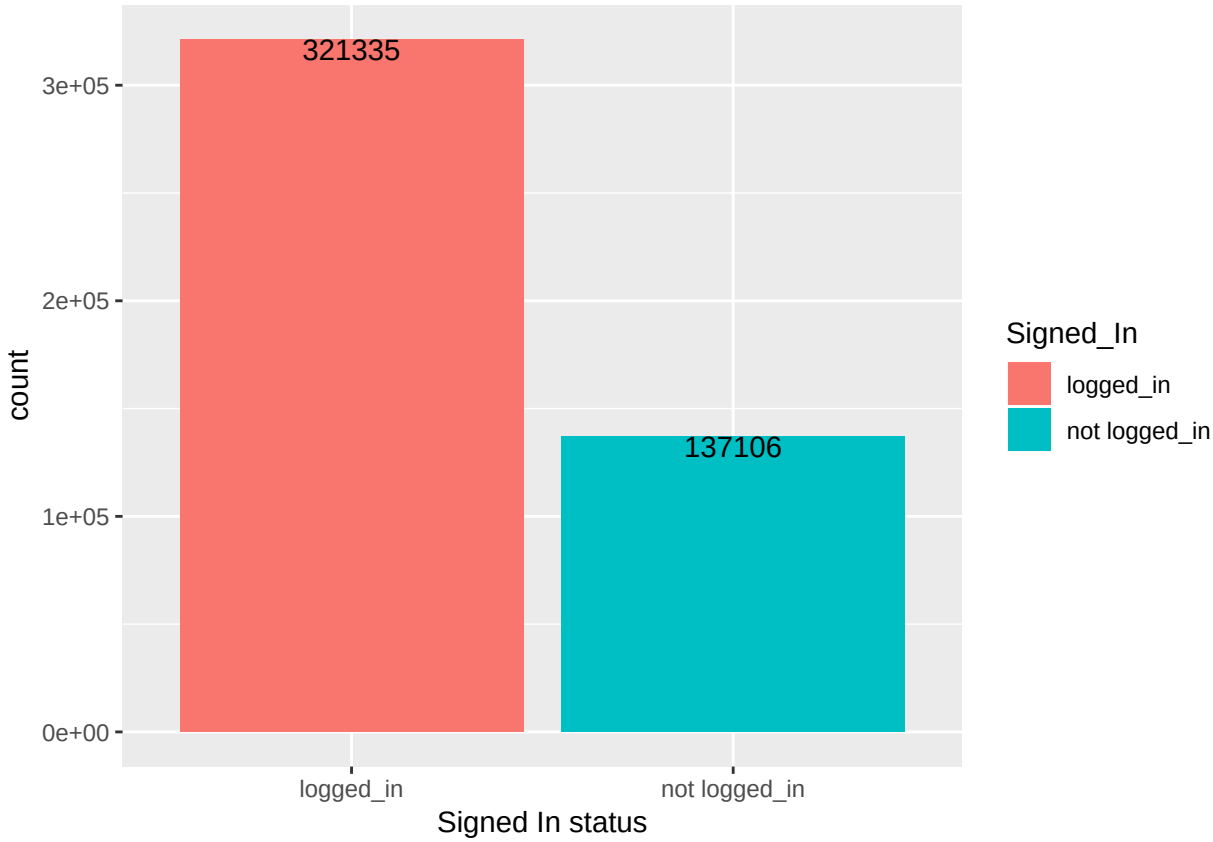

```
##day2
ggplot(day2_subset1, aes(x=Gender, y=..count..)) +
  geom_bar(aes(fill=Gender)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Gender')
```



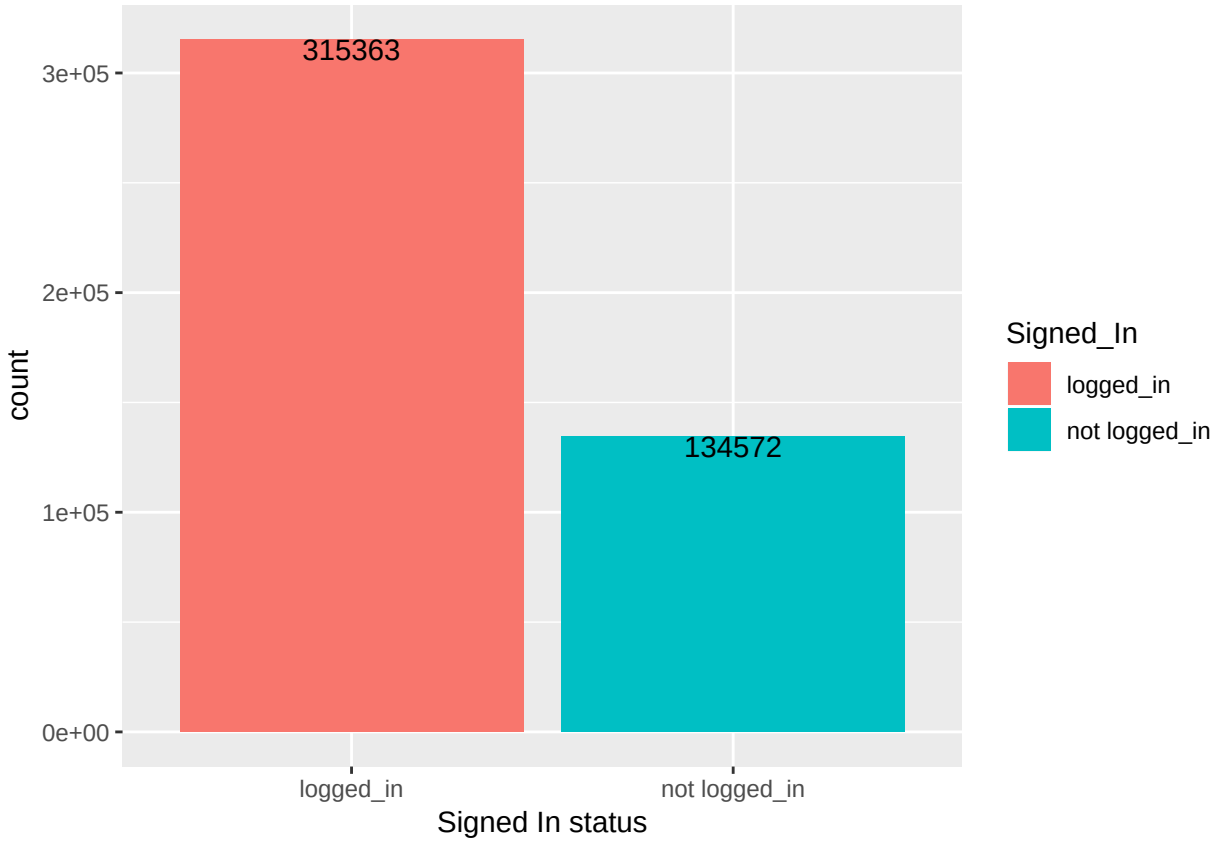

```
##day3
ggplot(day3_subset1, aes(x=Gender, y=..count..)) +
  geom_bar(aes(fill=Gender)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Gender')
```



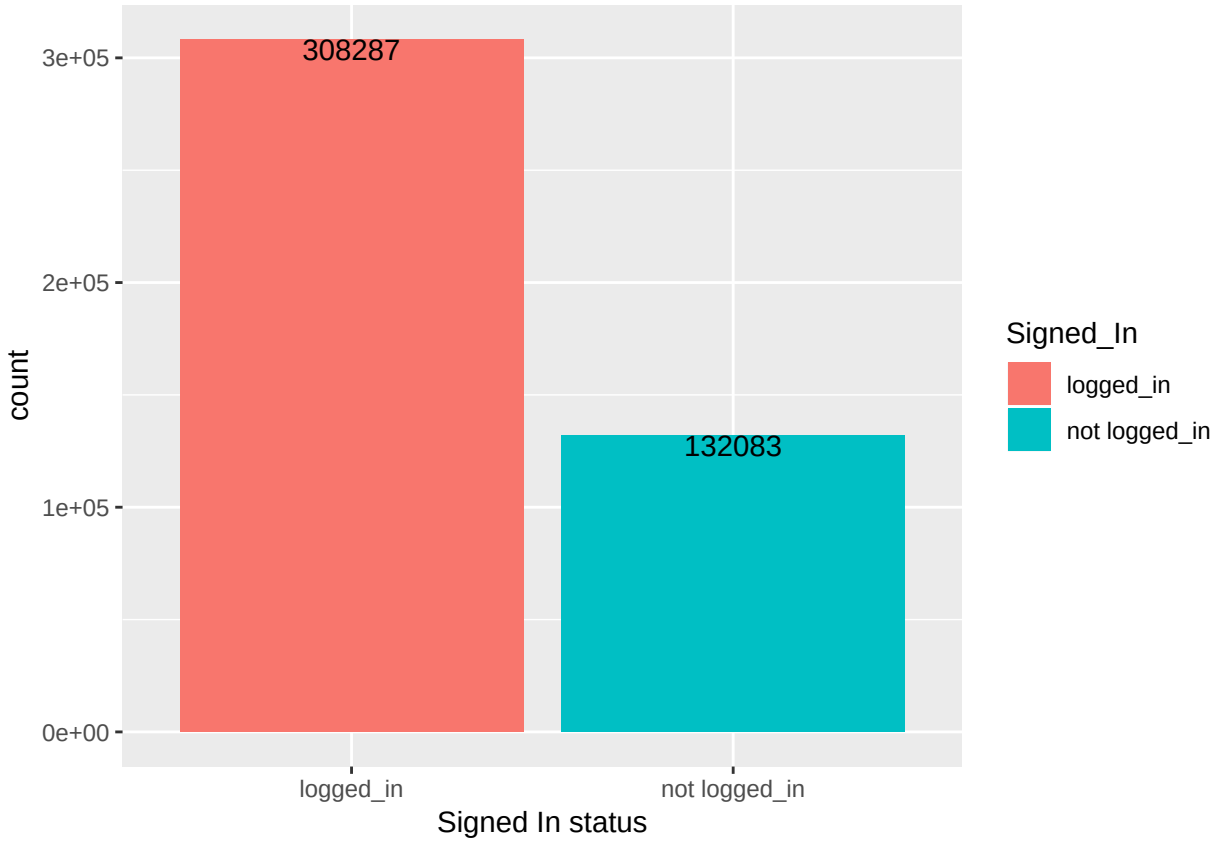

```
# ii) logged in users vs not logged in users:
##day1
ggplot(day1, aes(x=Signed_In, y=..count..)) +
  geom_bar(aes(fill=Signed_In)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Signed In status')
```



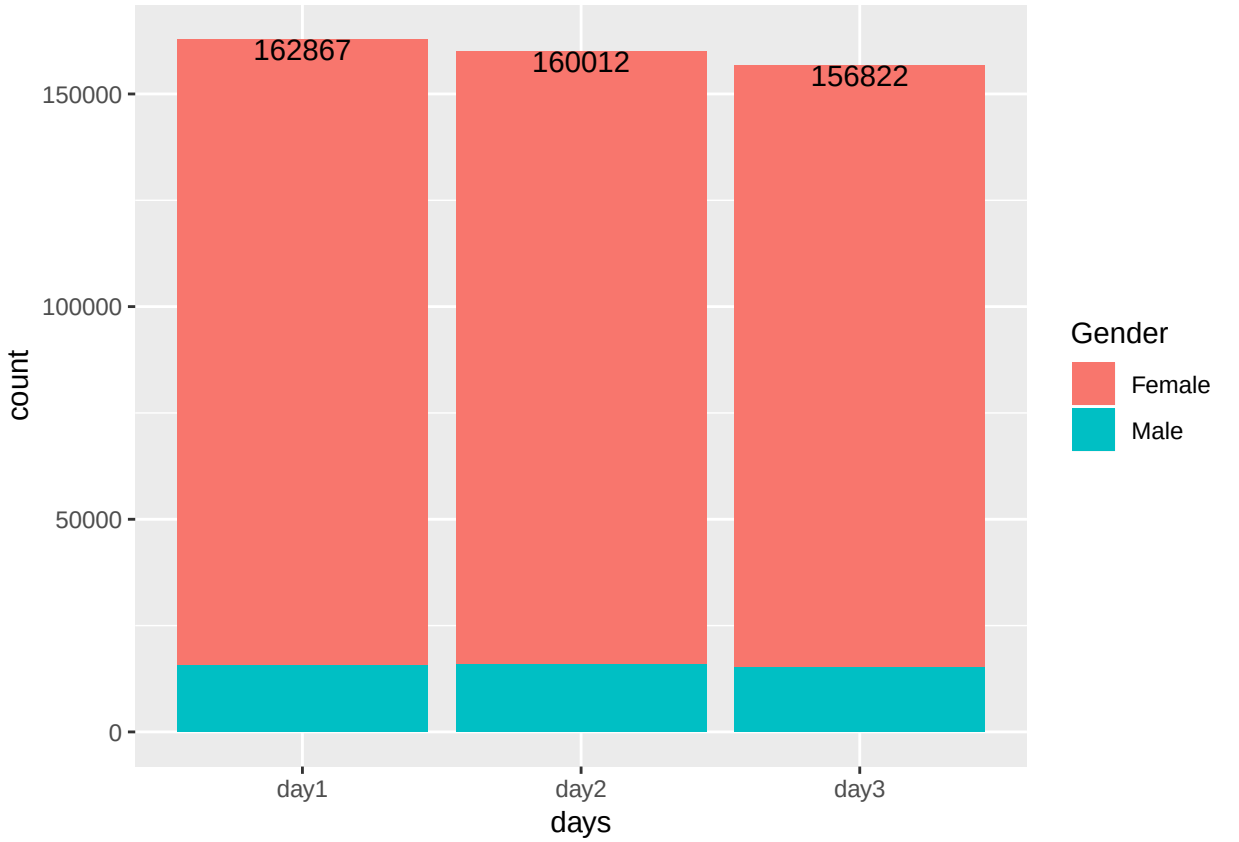

```
##day2
ggplot(day2, aes(x=Signed_In, y=..count..)) +
  geom_bar(aes(fill=Signed_In)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Signed In status')
```



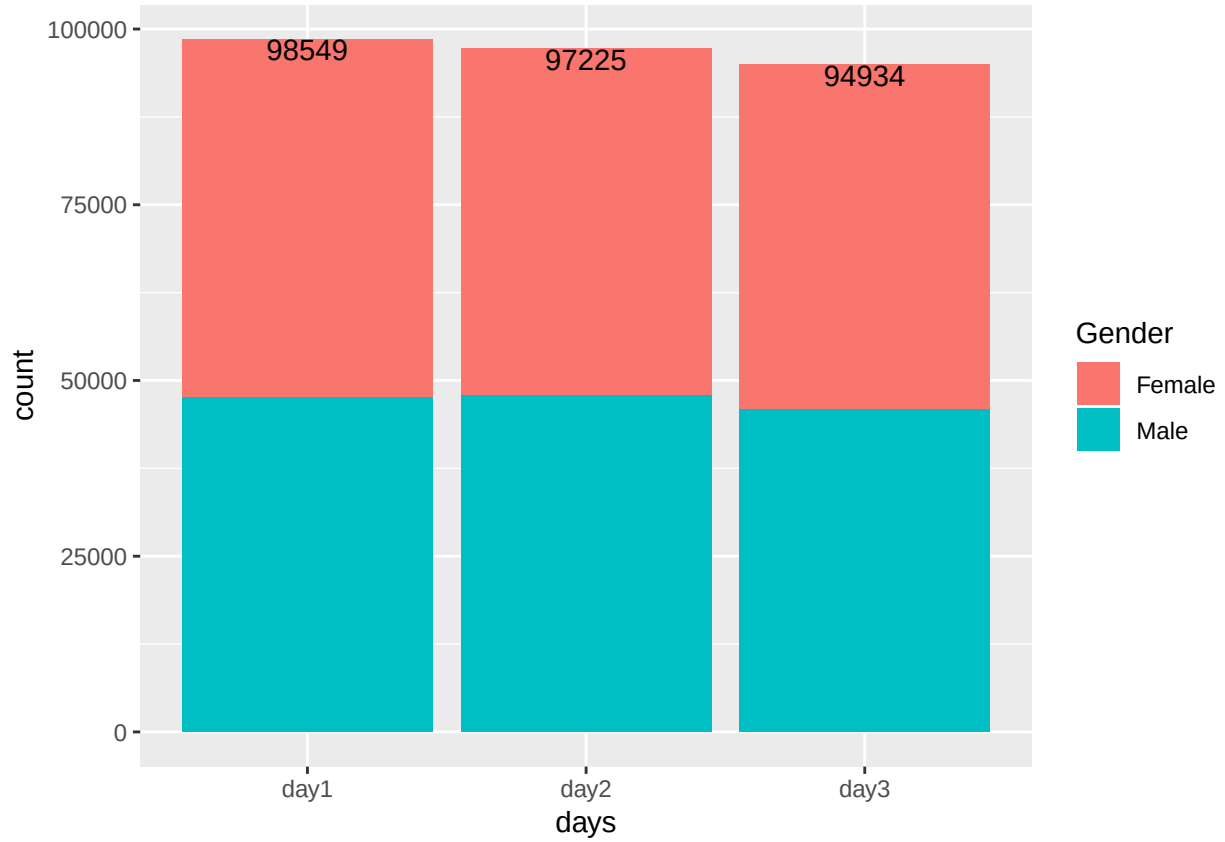

```
##day3
ggplot(day3, aes(x=Signed_In, y=..count..)) +
  geom_bar(aes(fill=Signed_In)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Signed In status')
```



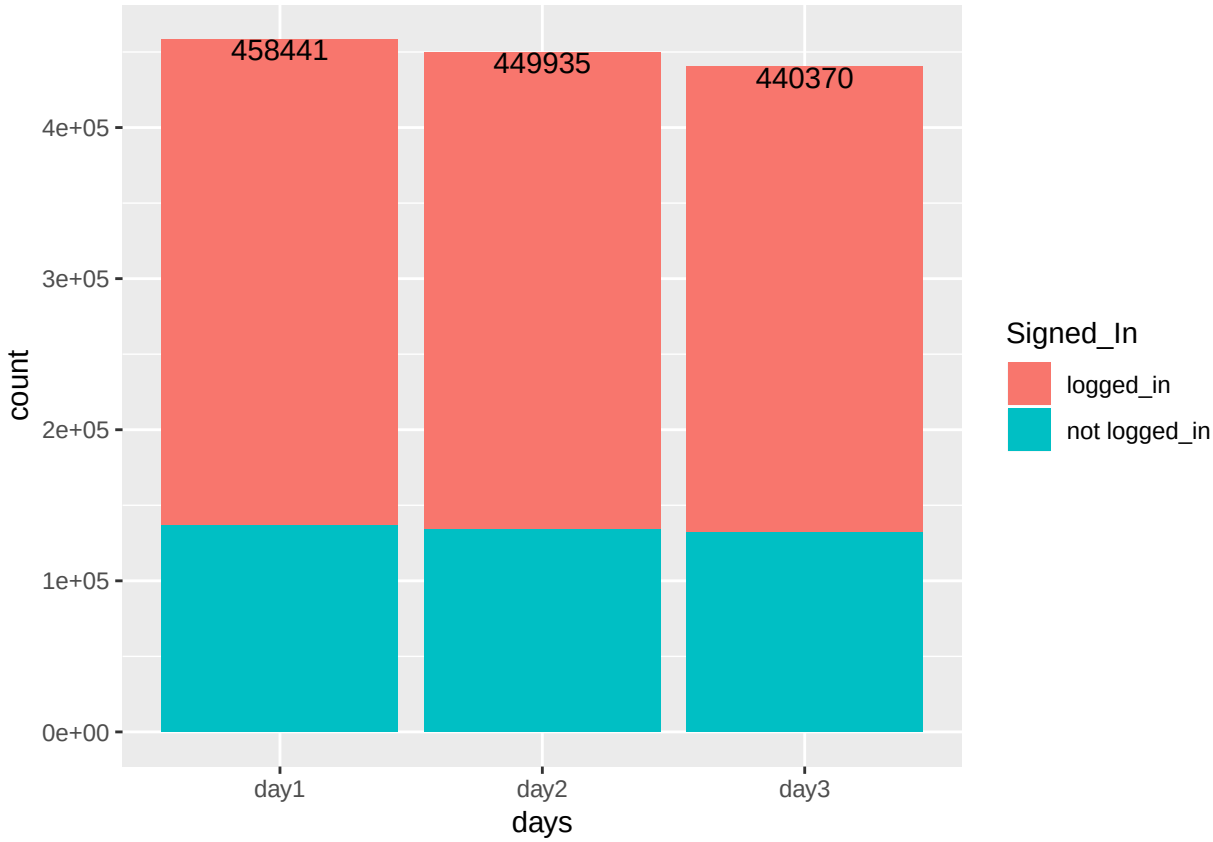

```
## Analysis across days

day1$day_id <- 'day1'
day2$day_id <- 'day2'
day3$day_id <- 'day3'
all_data <- rbind(day1, day2, day3)

# i) <20 years old males vs <20 years old females across days :

all_data_subset1 <- subset(all_data, Age<20)

ggplot(all_data_subset1, aes(x=day_id, y=..count..)) +
  geom_bar(aes(fill=Gender)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('days')
```



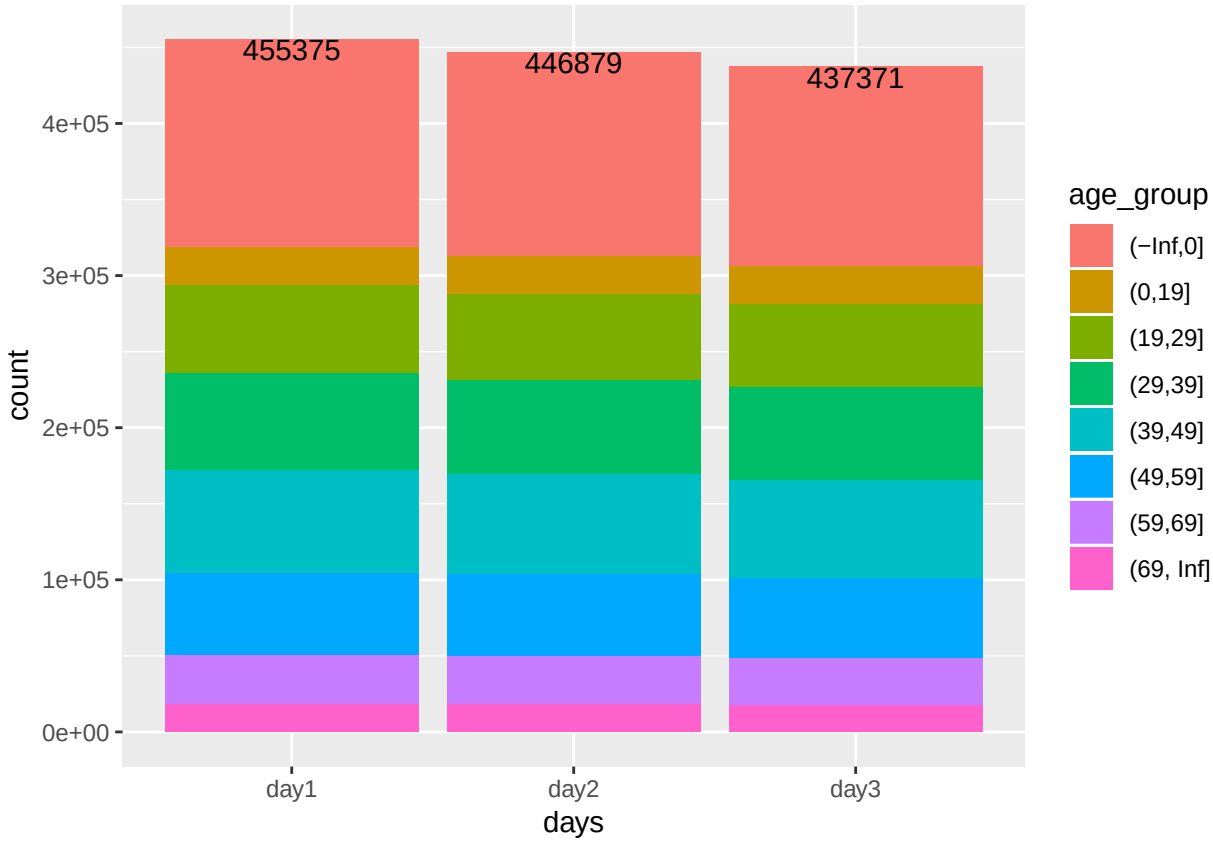

```
# ii) >50 years old males vs >50 years old females across days :

all_data_subset2 <- subset(all_data, Age>50)

ggplot(all_data_subset2, aes(x=day_id, y=..count..)) +
  geom_bar(aes(fill=Gender)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('days')
```



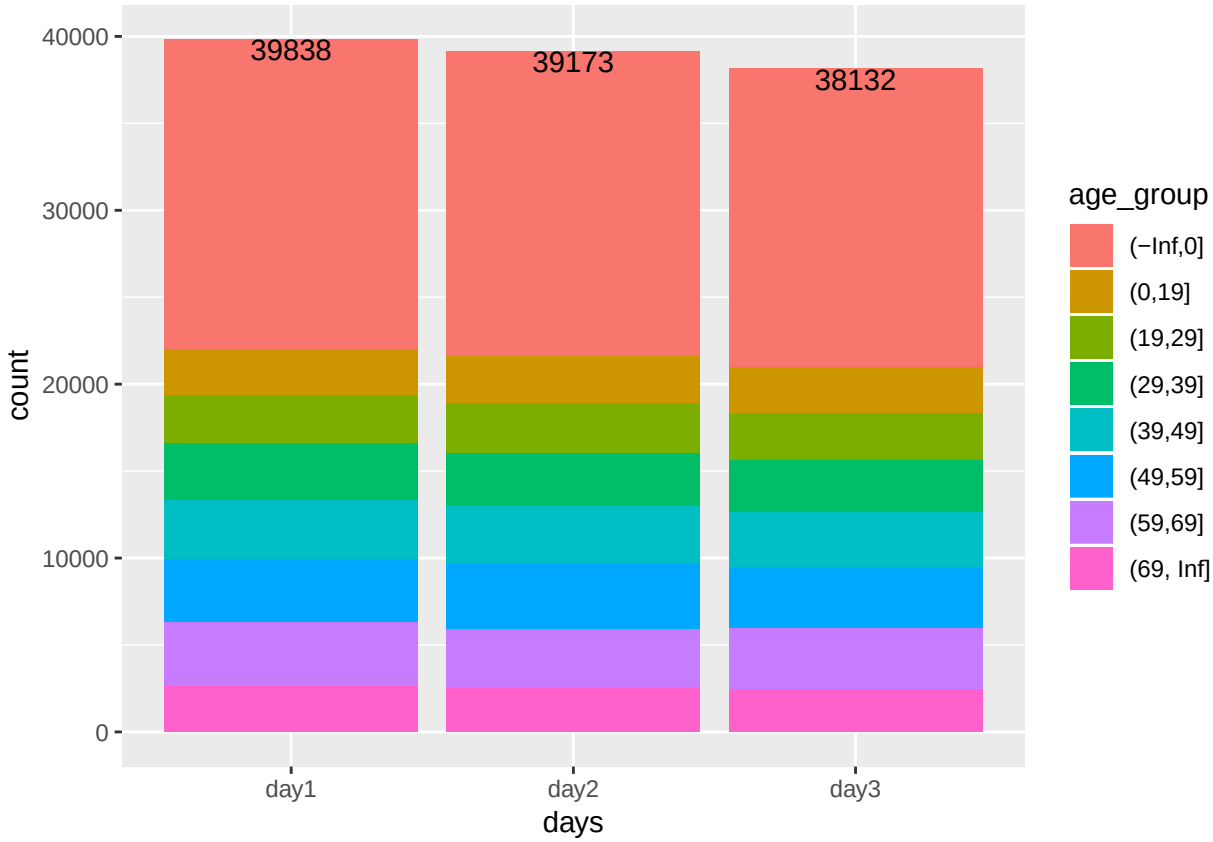

```
# iii) logged in users vs not logged in users across days:

ggplot(all_data, aes(x=day_id, y=..count..)) +
  geom_bar(aes(fill=Signed_In)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('days')
```



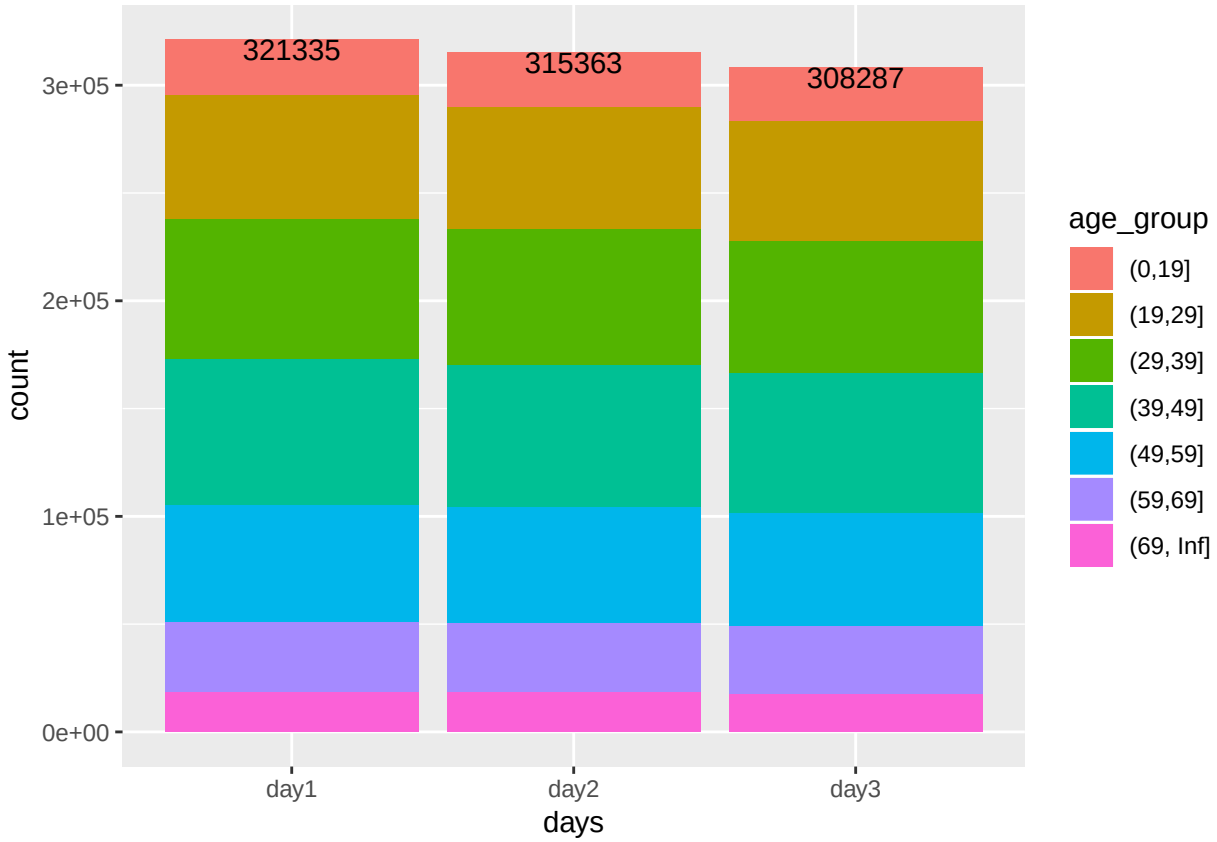

```
# iv) distribution of impressions across days and age groups

ggplot(subset(all_data, Impressions>0), aes(x=day_id, y=..count..)) +
  geom_bar(aes(fill=age_group)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('days')
```




```
# v) distribution of '> clicks' across days and age groups

ggplot(subset(all_data, Clicks>0), aes(x=day_id, y=..count..)) +
  geom_bar(aes(fill=age_group)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('days')
```




```
# vi) distribution of logged in users across days and across age groups

ggplot(subset(all_data, Signed_In=='logged_in'), aes(x=day_id, y=..count..)) +
  geom_bar(aes(fill=age_group)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('days')
```


321335
315363
308287
3e+05
2e+05
1e+05
0e+00
count
day1
day2
day3
days
age_group
(0,19]
(19,29]
(29,39]
(39,49]
(49,59]
(59,69]
(69, Inf]