EDA

Sarah Gaeta

4/25/2021

```
airbnb =  read_csv("clean_airbnb.csv", col_types = cols(last_review =
col_date(format = "%Y-%m-%d")))
```

First, let's clean the data

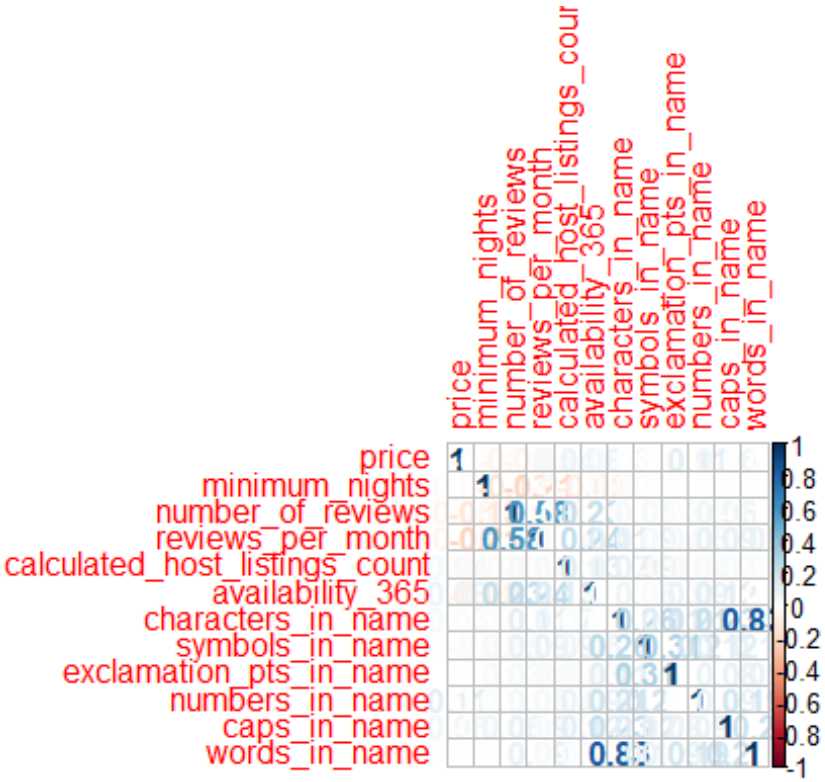
```
#next remove all 0's from price
price_0 = which(airbnb$price >0, )
airbnb = airbnb[price_0, ]
min_nights20 = which(airbnb$minimum_nights<=20)
airbnb = airbnb[min_nights20, ]
```

First, we will plot the correlograms between the columns of numerical type

```
updated_airbnb = airbnb %>% keep(is.numeric)
updated = updated_airbnb[,-c(1,2,3,4)] %>% na.omit()
#pairs(updated)
```

Next, we will plot the respective correlation matrix between the numerical variables:

```
corrplot(cor(updated), method = "number")
```



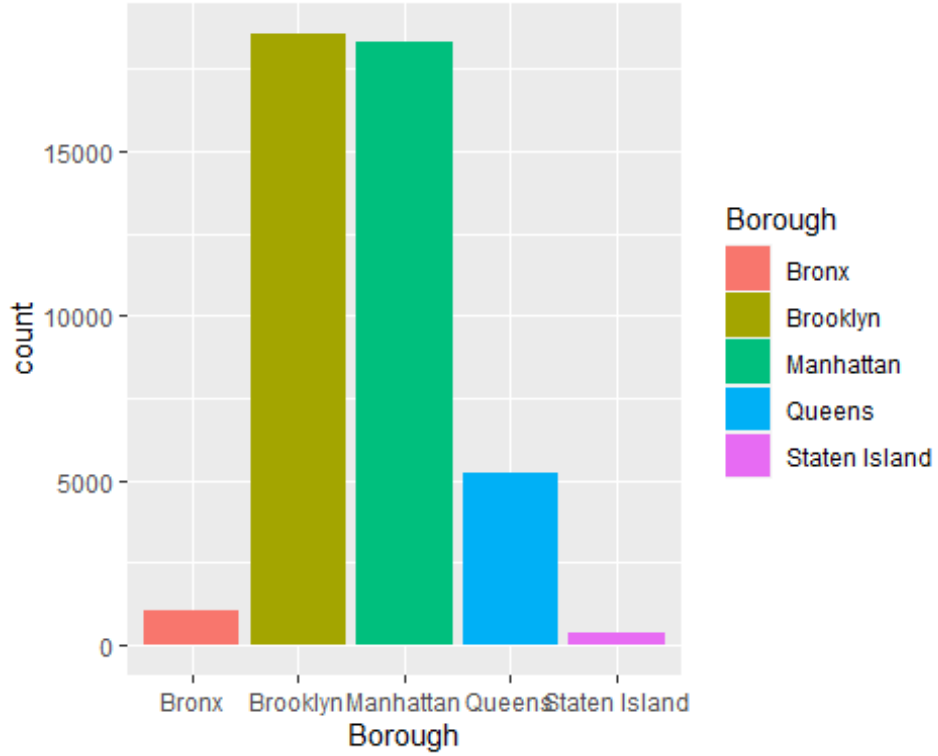
```
summary(airbnb)

##        id                 name               host_id            host_name
##  Min.   :    2539   Length:43521       Min.   :     2571   Length:43521
##  1st Qu.: 9394266   Class :character   1st Qu.:  7775447   Class
:character
##  Median :19554164   Mode  :character   Median : 30848788   Mode
:character
##  Mean   :18904349                      Mean   : 66957564
##  3rd Qu.:28888147                      3rd Qu.:104927746
##  Max.   :36487245                      Max.   :274321313
##
##  neighbourhood_group neighbourhood        latitude       longitude
##  Length:43521        Length:43521       Min.   :40.50   Min.   :-74.24
##  Class :character    Class :character   1st Qu.:40.69   1st Qu.:-73.98
##  Mode  :character    Mode  :character   Median :40.72   Median :-73.95
##                                         Mean   :40.73   Mean   :-73.95
##                                         3rd Qu.:40.76   3rd Qu.:-73.93
##                                         Max.   :40.91   Max.   :-73.71
##
##   room_type             price         minimum_nights   number_of_reviews
##  Length:43521       Min.   :   10.0   Min.   : 1.000   Min.   :  0.0
##  Class :character   1st Qu.:   69.0   1st Qu.: 1.000   1st Qu.:  1.0
##  Mode  :character   Median :  100.0   Median : 2.000   Median :  6.0
##                     Mean   :  148.6   Mean   : 3.073   Mean   : 25.1
##                     3rd Qu.:  175.0   3rd Qu.: 4.000   3rd Qu.: 27.0
##                     Max.   :10000.0   Max.   :20.000   Max.   :629.0
##
##   last_review         reviews_per_month calculated_host_listings_count
##  Min.   :2011-03-28   Min.   : 0.000    Min.   :  1.000
##  1st Qu.:2018-07-14   1st Qu.: 0.050    1st Qu.:  1.000
##  Median :2019-05-26   Median : 0.460    Median :  1.000
##  Mean   :2018-10-07   Mean   : 1.187    Mean   :  3.361
##  3rd Qu.:2019-06-23   3rd Qu.: 1.800    3rd Qu.:  2.000
##  Max.   :2019-07-08   Max.   :58.500    Max.   :327.000
##  NA's   :7913
##  availability_365 characters_in_name symbols_in_name
exclamation_pts_in_name
##  Min.   :  0.0    Min.   :  1.00     Min.   : 0.0000   Min.   :0.0000
##  1st Qu.:  0.0    1st Qu.: 30.00     1st Qu.: 0.0000   1st Qu.:0.0000
##  Median : 29.0    Median : 36.00     Median : 1.0000   Median :0.0000
##  Mean   :100.1    Mean   : 36.59     Mean   : 0.9808   Mean   :0.1582
##  3rd Qu.:184.0    3rd Qu.: 46.00     3rd Qu.: 1.0000   3rd Qu.:0.0000
##  Max.   :365.0    Max.   :179.00     Max.   :43.0000   Max.   :9.0000
##
##  numbers_in_name    caps_in_name    words_in_name
##  Min.   : 0.0000   Min.   : 0.000   Min.   : 1.000
##  1st Qu.: 0.0000   1st Qu.: 3.000   1st Qu.: 5.000
##  Median : 0.0000   Median : 4.000   Median : 6.000
##  Mean   : 0.4658   Mean   : 5.344   Mean   : 5.726
```

```
##  3rd Qu.: 1.0000   3rd Qu.: 6.000   3rd Qu.: 7.000
##  Max.   :11.0000   Max.   :45.000   Max.   :27.000
##
```

We will get rid of zero price, because a price of 0 is senseless:

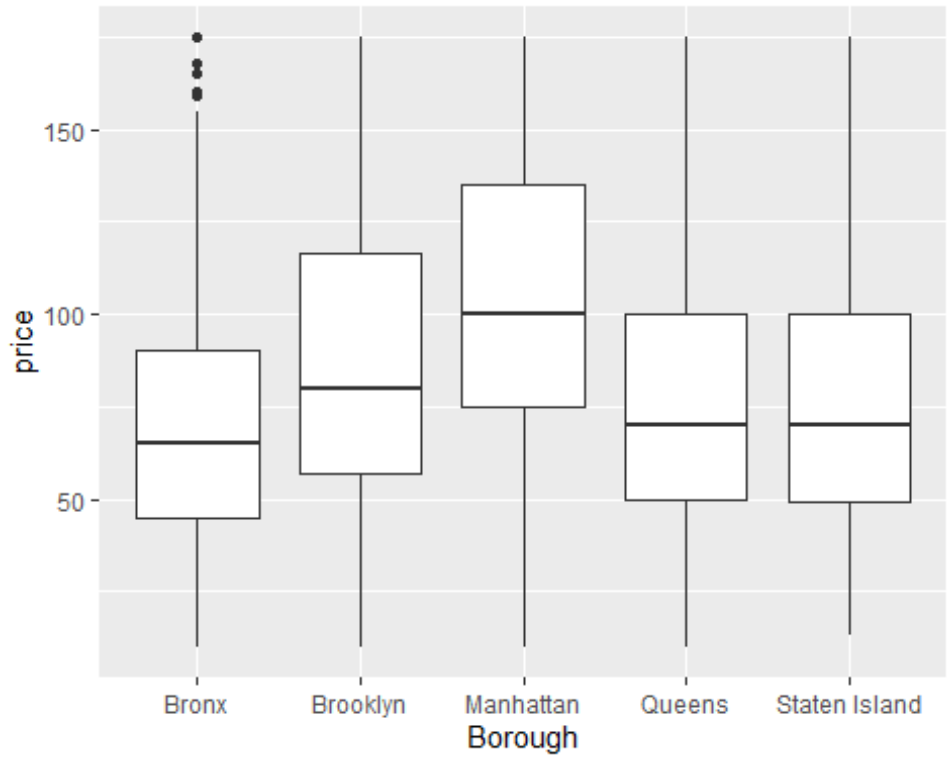
Then, get the count of each borough:

```
ggplot(airbnb, aes(x=neighbourhood_group, fill =
as.factor(neighbourhood_group))) + geom_bar() + scale_fill_discrete(name =
"Borough") + xlab("Borough")
```



First, we want to see if prices are greater for certain neighborhood groups: #cheaper airbnbs

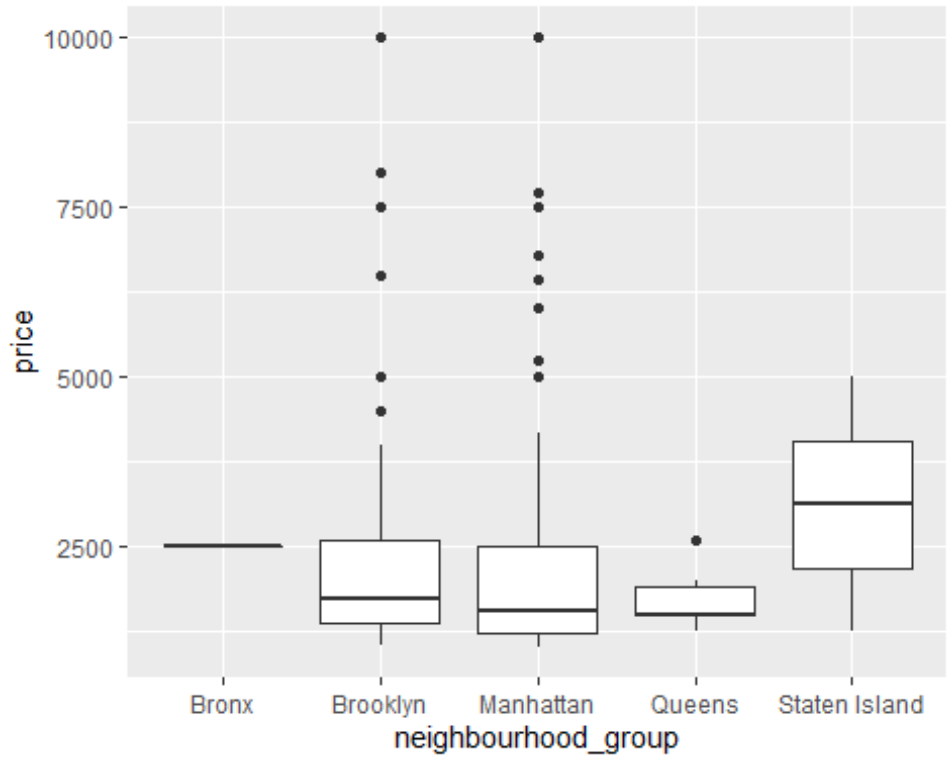
```
ggplot(airbnb[which(airbnb$price <= 175), ], aes(x= neighbourhood_group , y=
price)) + geom_boxplot() + xlab("Borough")
```



average listenings in Manhattan more expensive ish than average of other groups.

#more expensive:

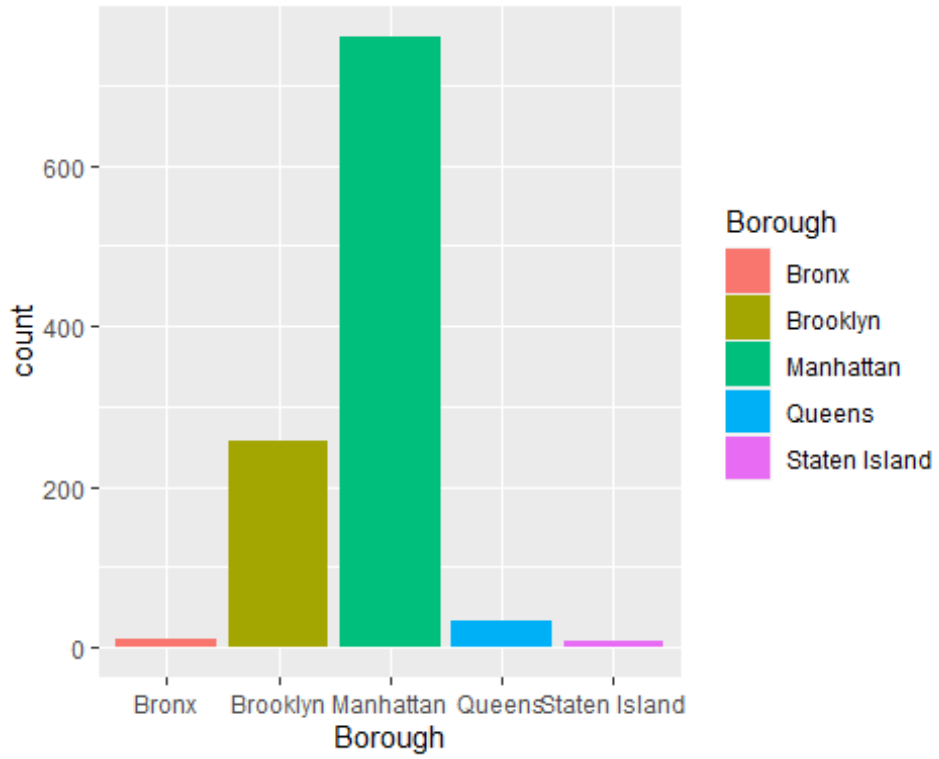
```
ggplot(airbnb[which(airbnb$price > 1000), ], aes(x= neighbourhood_group , y=
price)) + geom_boxplot()
```



more variability around the more expensive groups in Broklyn and Queens

what kind of airbnb has prices over 500?-could be interesting(what is similar-could be specific people)

```
price_500 = which(airbnb$price >= 500)
airbnb_over_500 = airbnb[price_500, ]
ggplot(airbnb_over_500, aes(neighbourhood_group, fill = neighbourhood_group))
+ geom_bar() + scale_fill_discrete(name = "Borough") +xlab("Borough")
```



A lot of the more expensive ones seem to be from Manhattan and Brooklyn.

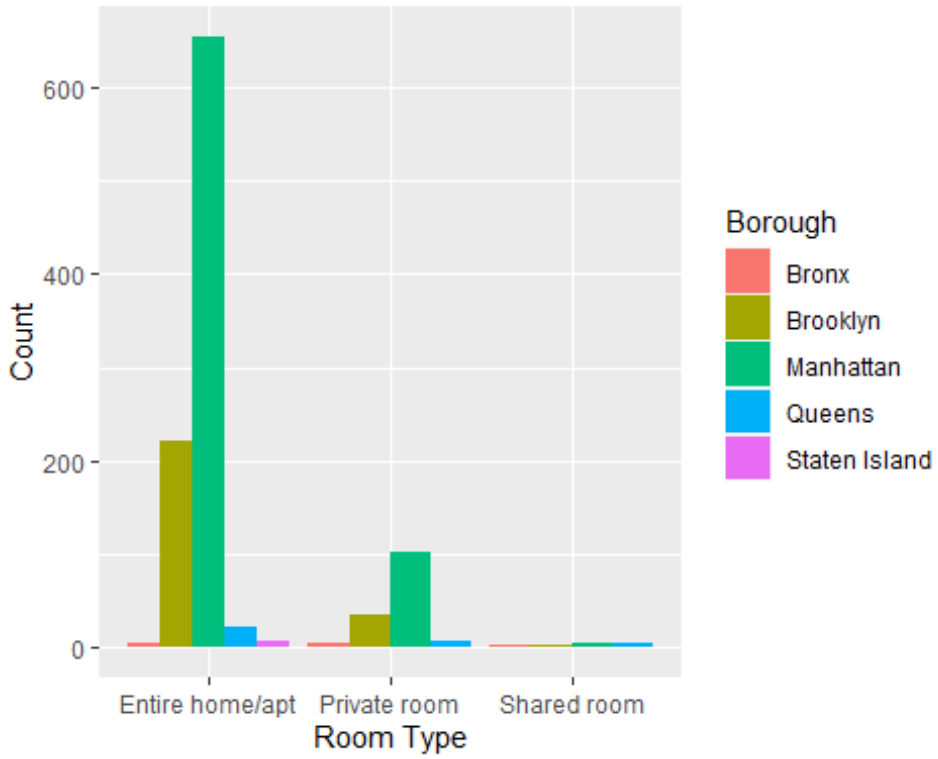
#dont plot below

```
# price_1000 = which(airbnb$price >= 500)
# airbnb_over_1000 = airbnb[price_1000, ]
# manhattan_1000 = which(airbnb_over_1000$neighbourhood_group == "Manhattan")
# airbnb_man_1000 = airbnb_over_1000[manhattan_1000, ]
# ggplot(airbnb_man_1000, aes(price, fill = room_type)) + geom_histogram()
```

From the plot above, we can see that for prices over $1000, we see that across all the prices, the ones that cost the most are entire homes/apts.

Let's find out which borough has more of which room type for the more expensive prices(over 500).

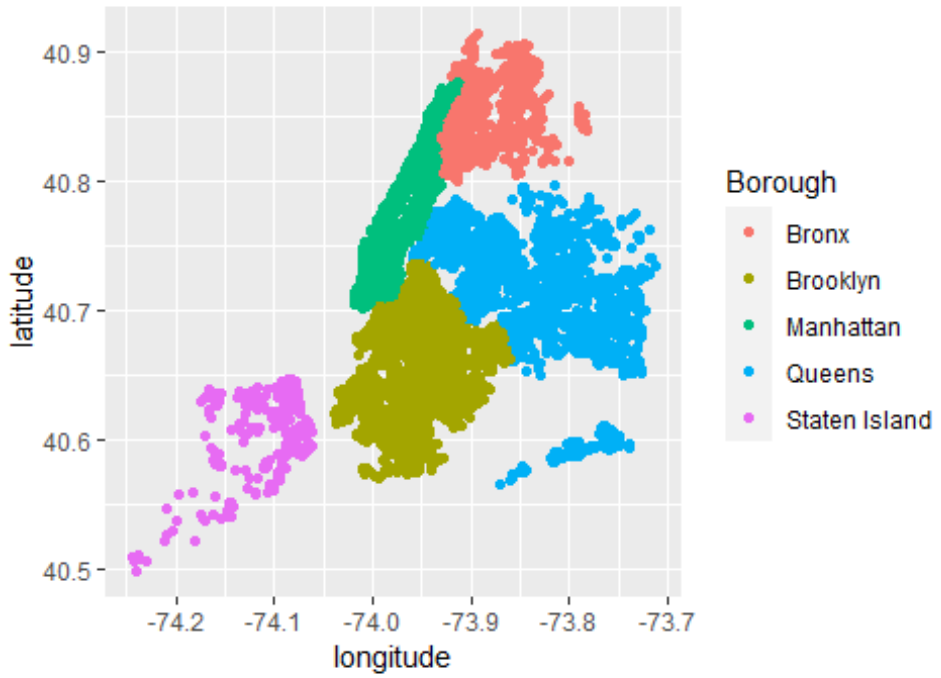
```
price_500 = which(airbnb$price >= 500)
airbnb_over_500 = airbnb[price_500, ]
expensive_air = airbnb_over_500 %>% group_by(neighbourhood_group) %>%
count(room_type)
ggplot(data = expensive_air, aes(x=room_type, y=n, fill =
neighbourhood_group)) + geom_bar(stat = "identity", position
=position_dodge()) + ylab("Count") + scale_fill_discrete(name = "Borough")
+xlab("Room Type")
```



This pattern of more expensive apartments(1000+) being entire homes holds true.

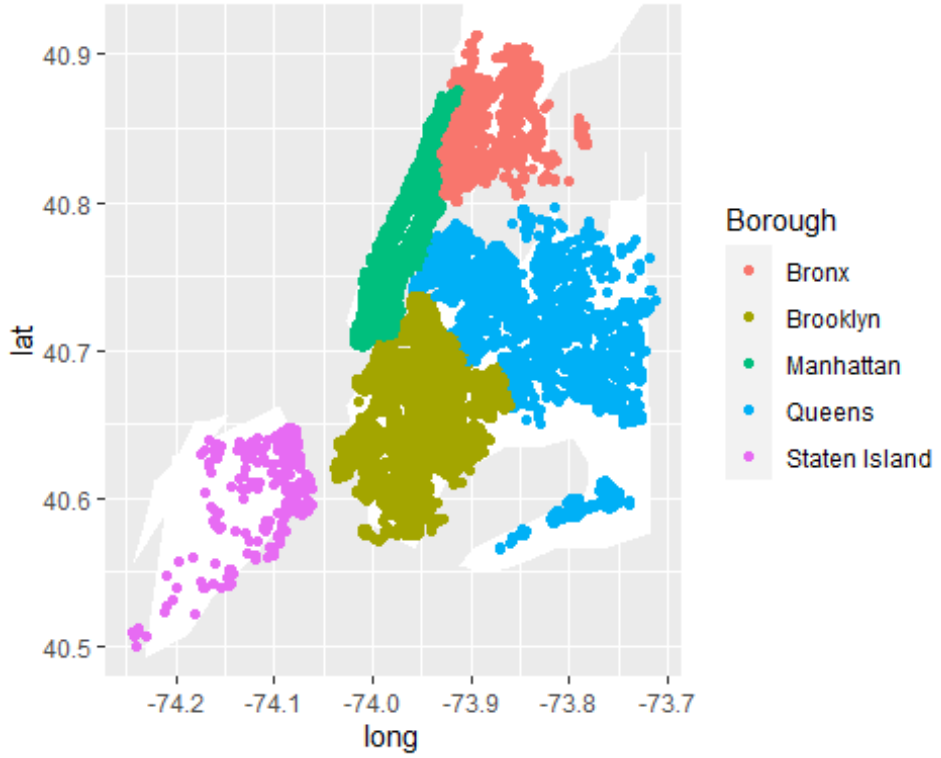
We will remove the name and host_name columns since they are represented by id and host_id:

```
new_air = airbnb[, -c(2, 4)]

ny = ggplot(data = airbnb, aes(longitude, latitude, color
=neighbourhood_group)) + coord_quickmap() + geom_point() +
scale_color_discrete(name = "Borough")
ny
```



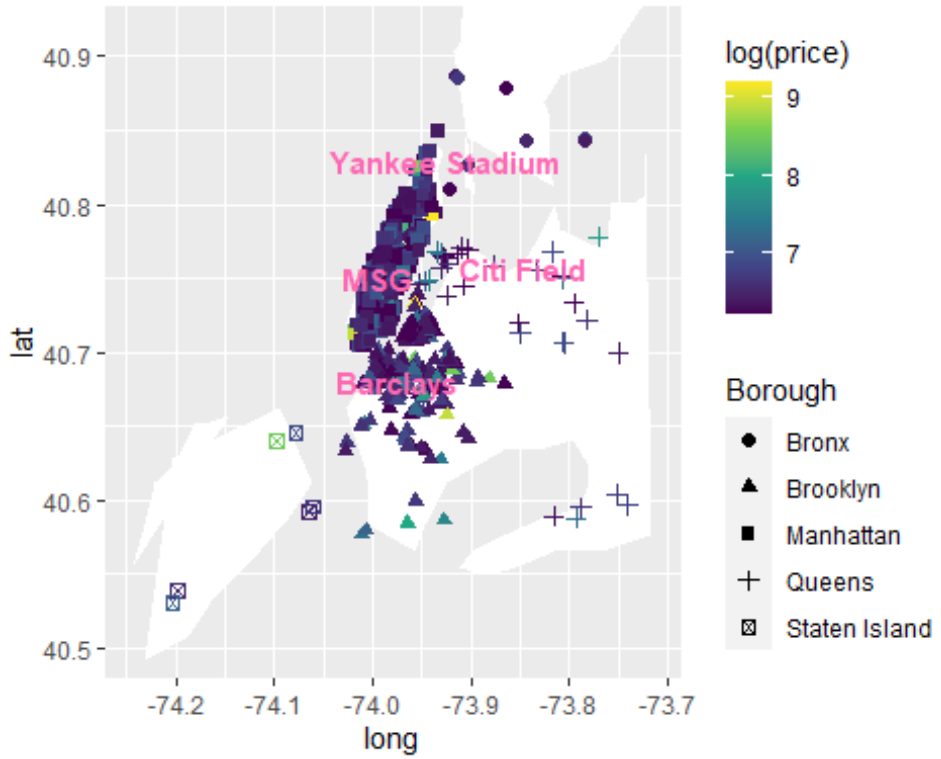

```
usa <- map_data("usa")
states <- map_data("state")

# ggplot() +
#   geom_polygon(data = worldmap,
#                aes(x = long, y = lat, group = group),
#                fill = 'gray90', color = 'black') +
#   coord_fixed(ratio = 1.3, xlim = c(-100,-70), ylim = c(50, 59)) +
#   theme_void() +
#   geom_point(data = lc_with_geo_counts,
#              aes(x = as.numeric(lng),
#                  y = as.numeric(lat), size = n, color = log(n)), alpha =
.7) +
#   scale_size_area(max_size = 8) +
#   scale_color_viridis_c() +
#   theme(legend.position = 'none') +
#   theme(title = element_text(size = 12))
# usa <- map_data("usa")
# states <- map_data("state")
# ny_df <- subset(states, region=="new york")
# ny_base <- ggplot(data=ny_df, mapping=aes(x=long, y=lat, group=group))+
#   coord_fixed(1.3) +
#   geom_polygon(color="black", fill="gray")
# ny_base+theme_nothing() + geom_point(data = airbnb,
aes(x=as.numeric(longitude), y=as.numeric(latitude)))

world <- map_data('world')
```

```
min_lat = min(airbnb$latitude)
max_lat = max(airbnb$latitude)
min_long = min(airbnb$longitude)
max_long = max(airbnb$longitude)
east_coast <- subset(states, region %in% c("new york"))
p = ggplot(data = east_coast) +  geom_polygon(aes(x = long, y = lat, group  =
group), fill = "white") + coord_fixed(1.3)
p + geom_point(data = airbnb, aes(x = longitude, y=latitude, color =
as.factor(neighbourhood_group))) + xlim(c(min_long, max_long)) +
coord_cartesian(ylim=c(min_lat,max_lat)) + scale_color_discrete(name =
"Borough")
```



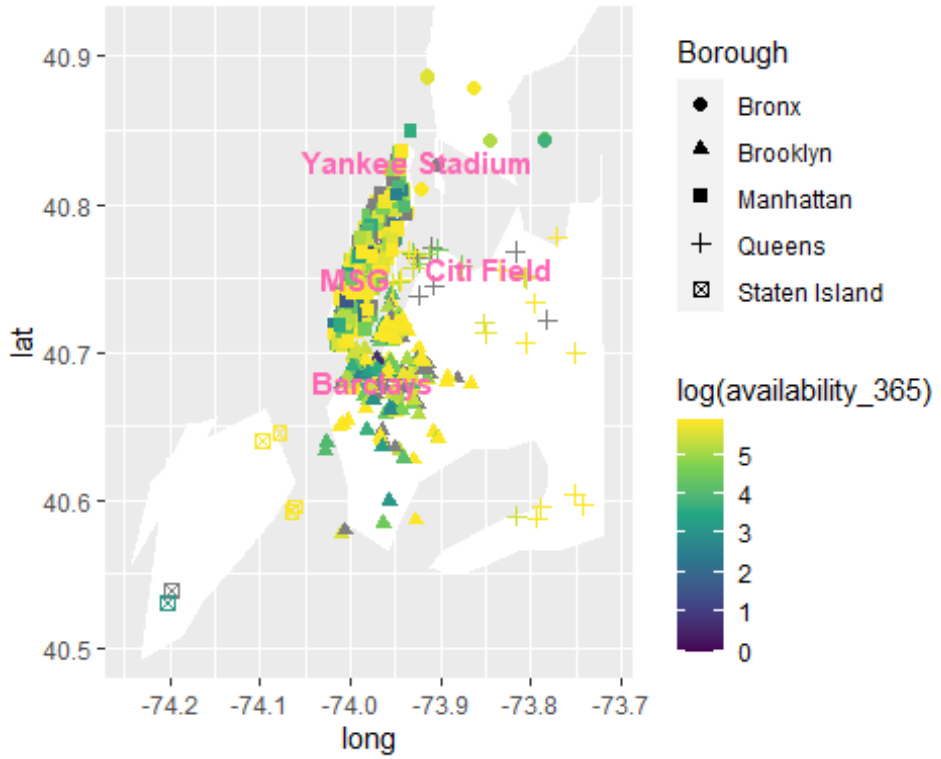

```
min_lat = min(airbnb$latitude)
max_lat = max(airbnb$latitude)
min_long = min(airbnb$longitude)
max_long = max(airbnb$longitude)
east_coast <- subset(states, region %in% c("new york"))
airbnb1 = airbnb[which(airbnb$price>=500), ]
p = ggplot(data = east_coast) +  geom_polygon(aes(x = long, y = lat, group  =
group), fill = "white") + coord_fixed(1.3)
p + geom_point(data = airbnb1, aes(x = longitude, y=latitude, color =
log(price), shape = as.factor(neighbourhood_group)), size = 2) +
xlim(c(min_long, max_long)) + coord_cartesian(ylim=c(min_lat,max_lat)) +
annotate("text", x = c(-73.9934, -73.9754, -73.9262, -73.8458), y = c(40.75,
40.68, 40.8296, 40.7571),
           label = c("MSG", "Barclays", "Yankee Stadium", "Citi Field") ,
color="hot pink",
```

```
          size=4 , angle=360, fontface="bold") +scale_color_viridis_c() +
scale_shape_discrete(name = "Borough")
```

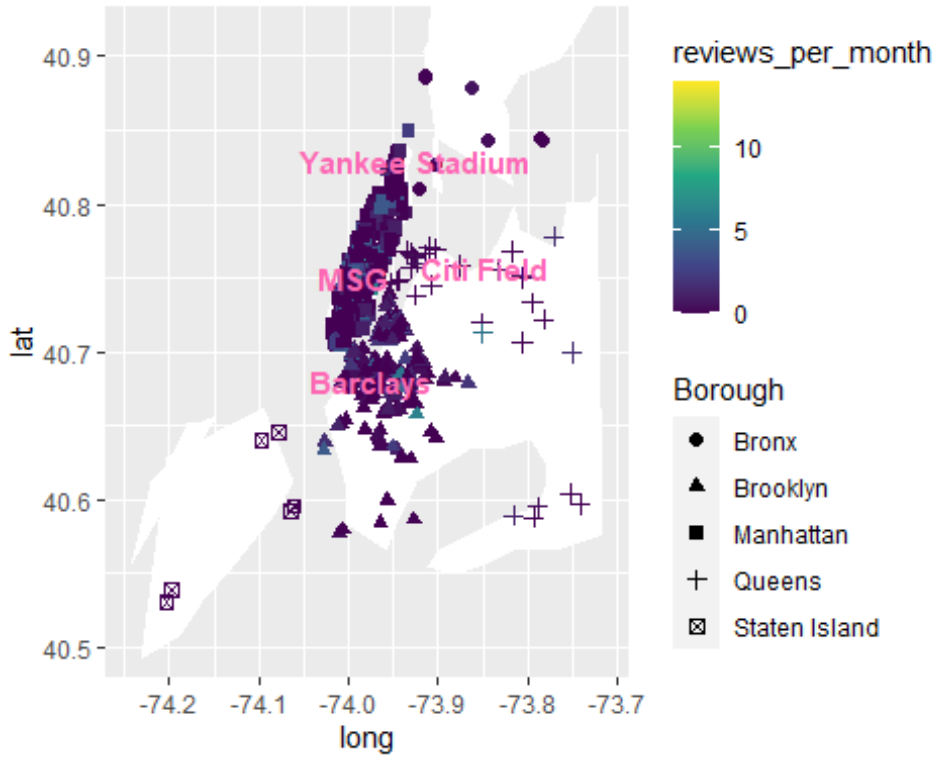



```
#maybe that not many expensive airbnbs around citi field-mets suck compared
to the yankees
#also a lot of ppl don't stay in queens
#compare distribution of price in thosez areas(boxplot)

min_lat = min(airbnb$latitude)
max_lat = max(airbnb$latitude)
min_long = min(airbnb$longitude)
max_long = max(airbnb$longitude)
east_coast <- subset(states, region %in% c("new york"))
airbnb1 = airbnb[which(airbnb$price>=500), ]
p = ggplot(data = east_coast) +  geom_polygon(aes(x = long, y = lat, group  =
group), fill = "white") + coord_fixed(1.3)
p + geom_point(data = airbnb1, aes(x = longitude, y=latitude, color =
log(availability_365), shape = as.factor(neighbourhood_group)), size = 2) +
xlim(c(min_long, max_long)) + coord_cartesian(ylim=c(min_lat,max_lat)) +
annotate("text", x = c(-73.9934, -73.9754, -73.9262, -73.8458), y = c(40.75,
40.68, 40.8296, 40.7571),
          label = c("MSG", "Barclays", "Yankee Stadium", "Citi Field") ,
color="hot pink",
          size=4 , angle=360, fontface="bold") + scale_color_viridis_c() +
scale_shape_discrete(name = "Borough")
```

```
## Coordinate system already present. Adding new coordinate system, which
will replace the existing one.
```




```
## Coordinate system already present. Adding new coordinate system, which
will replace the existing one.
```



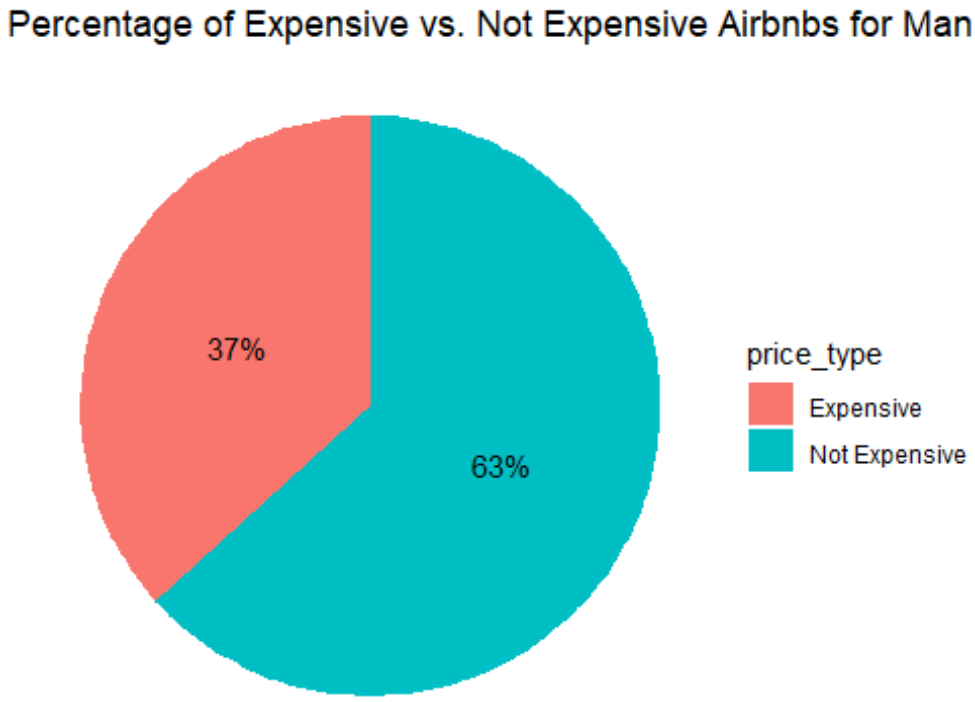

```
# convert price into 2 categorical groups
price_categorical = 1:length(airbnb$price)
for (i in price_categorical) {
  if (airbnb$price[i] > 175){
   price_categorical[i] = "expensive"
  }
  else {
    price_categorical[i] = "not expensive"
  }
}
airbnb_price_categorical = cbind(airbnb, price_categorical)
```

Boxplot of neighbor group vs room types for prices over 1000

```
# subset to only include airbnbs with prices over 1000
# price_1000 = which(airbnb$price >= 1000)
# airbnb_over_1000 = airbnb[price_1000, ]
# # plot room vs price for each neightborhood group
# ggplot(airbnb_over_1000, aes(x = room_type, y=log(price), color =
neighbourhood_group))+ geom_boxplot() + ggtitle("Price Per Room Type for Each
Neighbourhood Group") + scale_color_discrete(name = "Borough")

#subset airbnb for only manhattan data
manhattan = which(airbnb_price_categorical$neighbourhood_group ==
"Manhattan")
airbnb_manhattan = airbnb_price_categorical[manhattan, ]
```

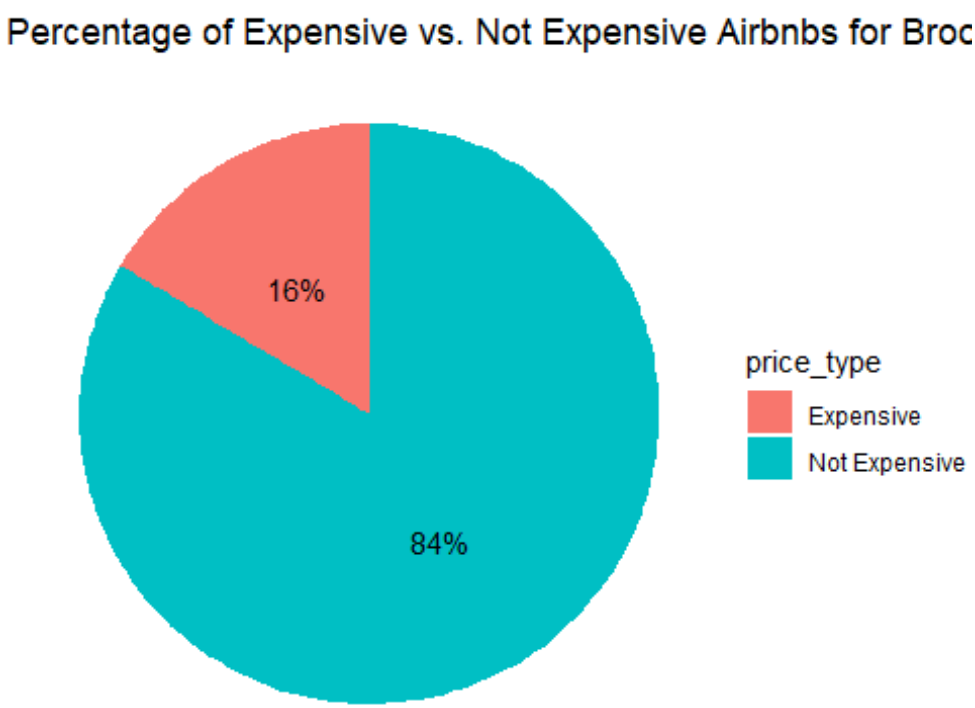
```
#subset data for only expensive airbnbs in manhattan
manhattan_expensive = which(airbnb_manhattan$price_categorical ==
"expensive")
airbnb_manhattan_expensive = airbnb_manhattan[manhattan_expensive, ]
#find number of expensive and inexpensive airbnbs
num_exp_manhattan = length(airbnb_manhattan_expensive$price)
num_inexp_manhattan = length(airbnb_manhattan$price) - num_exp_manhattan
#subset airbnb for only brooklyn data
brooklyn = which(airbnb_price_categorical$neighbourhood_group == "Brooklyn")
airbnb_brooklyn = airbnb_price_categorical[brooklyn, ]
#subset data for only expensive airbnbs in brooklyn
brooklyn_expensive = which(airbnb_brooklyn$price_categorical == "expensive")
airbnb_brooklyn_expensive = airbnb_brooklyn[brooklyn_expensive, ]
#find number of expensive and inexpensive airbnbs
num_exp_brooklyn = length(airbnb_brooklyn_expensive$price)
num_inexp_brooklyn = length(airbnb_brooklyn$price) - num_exp_brooklyn
#subset airbnb for only queens data
queens = which(airbnb_price_categorical$neighbourhood_group == "Queens")
airbnb_queens = airbnb_price_categorical[queens, ]
#subset data for only expensive airbnbs in queens
queens_expensive = which(airbnb_queens$price_categorical == "expensive")
airbnb_queens_expensive = airbnb_queens[queens_expensive, ]
#find number of expensive and inexpensive airbnbs
num_exp_queens = length(airbnb_queens_expensive$price)
num_inexp_queens = length(airbnb_queens$price) - num_exp_queens
#subset airbnb for only bronx data
bronx = which(airbnb_price_categorical$neighbourhood_group == "Bronx")
airbnb_bronx = airbnb_price_categorical[bronx, ]
bronx_expensive = which(airbnb_bronx$price_categorical == "expensive")
#subset data for only expensive airbnbs in bronx
airbnb_bronx_expensive = airbnb_bronx[bronx_expensive, ]
#find number of expensive and inexpensive airbnbs
num_exp_bronx = length(airbnb_bronx_expensive$price)
num_inexp_bronx = length(airbnb_bronx$price) - num_exp_bronx
#subset airbnb for only staten data
staten_island = which(airbnb_price_categorical$neighbourhood_group == "Staten
Island")
airbnb_staten_island = airbnb_price_categorical[staten_island, ]
#subset data for only expensive airbnbs in staten island
staten_island_expensive = which(airbnb_staten_island$price_categorical ==
"expensive")
airbnb_staten_island_expensive =
airbnb_staten_island[staten_island_expensive, ]
#find number of expensive and inexpensive airbnbs
num_exp_staten_island = length(airbnb_staten_island_expensive$price)
num_inexp_staten_island = length(airbnb_staten_island$price) -
num_exp_staten_island

# categories of pie chart
data <- data.frame(slices = c(num_exp_manhattan, num_inexp_manhattan),
```

```
                    price_type = c("Expensive","Not Expensive"),
                    stringsAsFactors = F)
# find percentages of categories
data <- data %>%
  mutate(per=slices/sum(slices)) %>%
  arrange(desc(price_type))
data$label <- scales::percent(data$per)
# Plot pie graph
ggplot(data=data)+
  geom_bar(aes(x="", y=per, fill=price_type), stat="identity", width = 1)+
  coord_polar("y", start=0)+
  theme_void()+
  geom_text(aes(x=1, y = cumsum(per) - per/2, label=label))+
  ggtitle("Percentage of Expensive vs. Not Expensive Airbnbs for Manhattan")
```

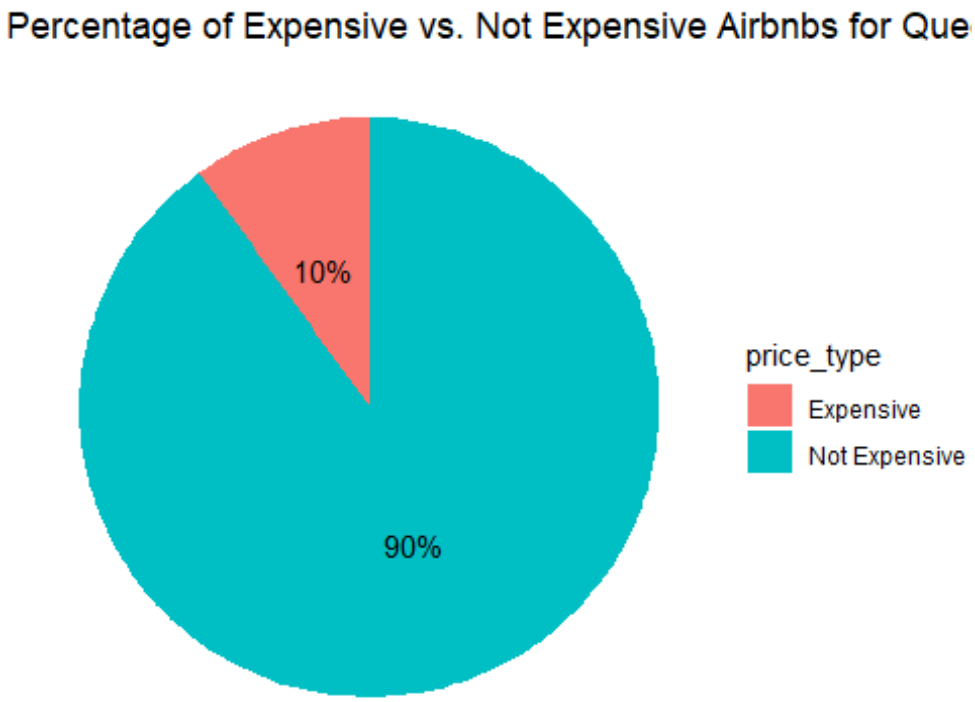



```
# categories of pie chart
data <- data.frame(slices = c(num_exp_brooklyn, num_inexp_brooklyn),
                    price_type = c("Expensive","Not Expensive"),
                    stringsAsFactors = F)
# find percentages of categories
data <- data %>%
  mutate(per=slices/sum(slices)) %>%
  arrange(desc(price_type))
data$label <- scales::percent(data$per)
# Plot pie graph
ggplot(data=data)+
  geom_bar(aes(x="", y=per, fill=price_type), stat="identity", width = 1)+
```

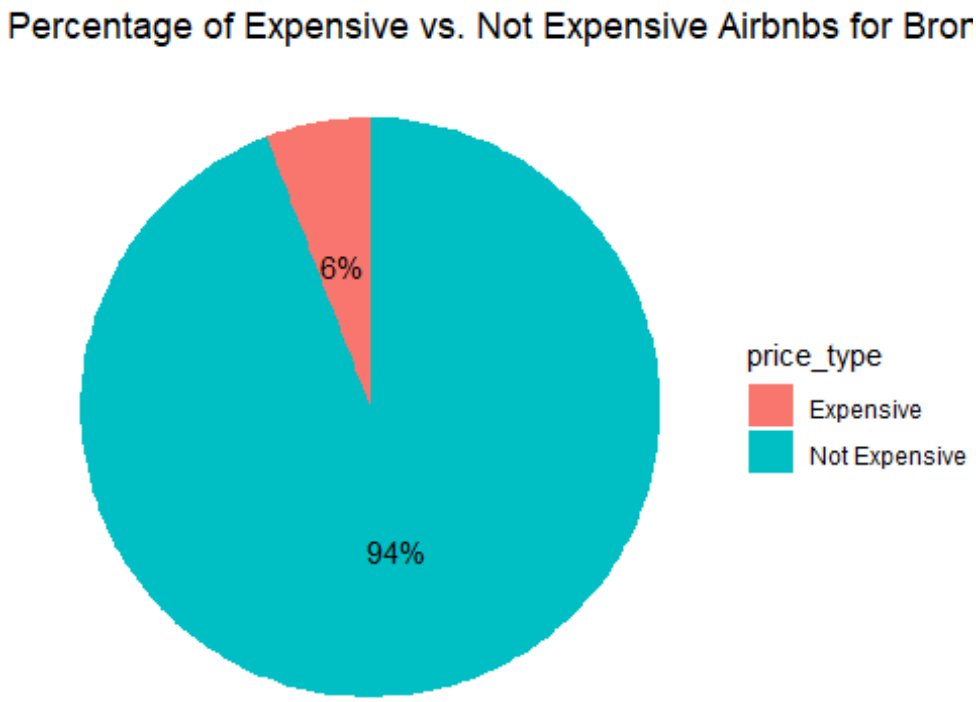
```
  coord_polar("y", start=0)+
  theme_void()+
  geom_text(aes(x=1, y = cumsum(per) - per/2, label=label))+
  ggtitle("Percentage of Expensive vs. Not Expensive Airbnbs for Brooklyn")
```



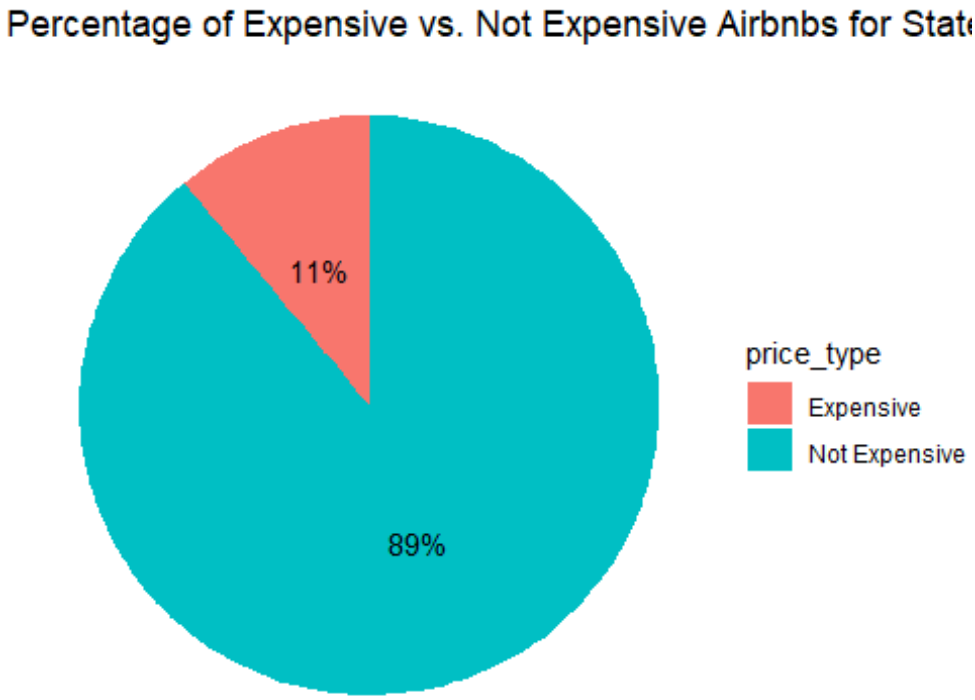

```
# categories of pie chart
data <- data.frame(slices = c(num_exp_queens, num_inexp_queens),
                   price_type = c("Expensive","Not Expensive"),
                   stringsAsFactors = F)
# find percentages of categories
data <- data %>%
  mutate(per=slices/sum(slices)) %>%
  arrange(desc(price_type))
data$label <- scales::percent(data$per)
# Plot pie graph
ggplot(data=data)+
  geom_bar(aes(x="", y=per, fill=price_type), stat="identity", width = 1)+
  coord_polar("y", start=0)+
  theme_void()+
  geom_text(aes(x=1, y = cumsum(per) - per/2, label=label))+
  ggtitle("Percentage of Expensive vs. Not Expensive Airbnbs for Queens")
```




```
# categories of pie chart
data <- data.frame(slices = c(num_exp_bronx, num_inexp_bronx),
                   price_type = c("Expensive","Not Expensive"),
                   stringsAsFactors = F)
# find percentages
data <- data %>%
  mutate(per=slices/sum(slices)) %>%
  arrange(desc(price_type))
data$label <- scales::percent(data$per)
# Plot pie graph
ggplot(data=data)+
  geom_bar(aes(x="", y=per, fill=price_type), stat="identity", width = 1)+
  coord_polar("y", start=0)+
  theme_void()+
  geom_text(aes(x=1, y = cumsum(per) - per/2, label=label))+
  ggtitle("Percentage of Expensive vs. Not Expensive Airbnbs for Bronx")
```

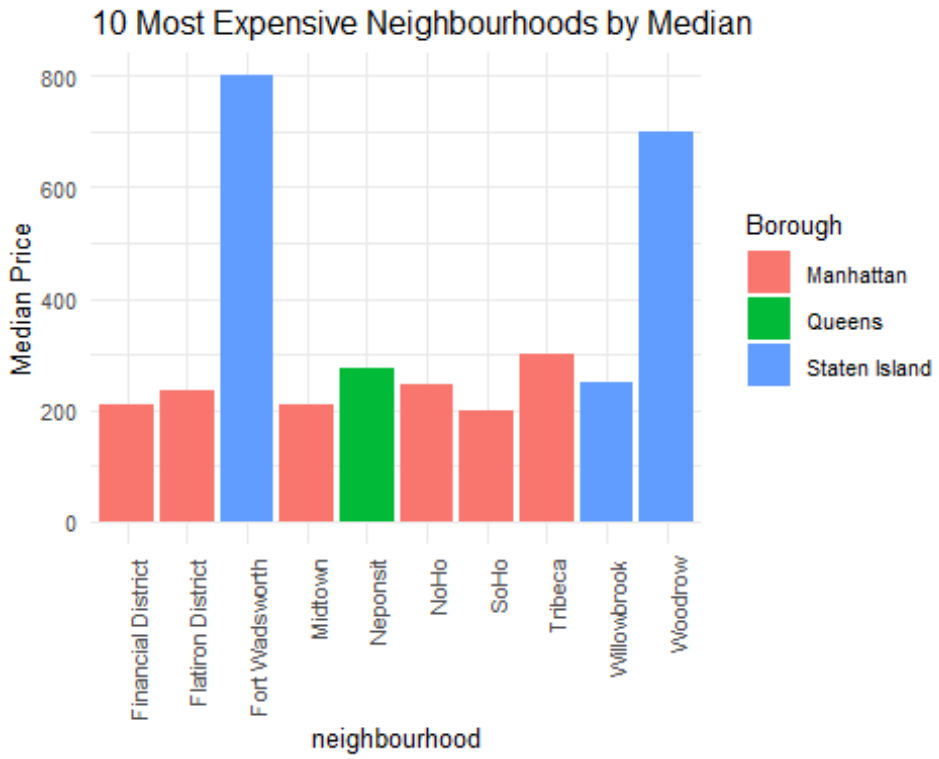



```
# categories of pie chart
data <- data.frame(slices = c(num_exp_staten_island,
num_inexp_staten_island),
                   price_type = c("Expensive","Not Expensive"),
                   stringsAsFactors = F)
# find percentages
data <- data %>%
  mutate(per=slices/sum(slices)) %>%
  arrange(desc(price_type))
data$label <- scales::percent(data$per)
# Plot pie graph
ggplot(data=data)+
  geom_bar(aes(x="", y=per, fill=price_type), stat="identity", width = 1)+
  coord_polar("y", start=0)+
  theme_void()+
  geom_text(aes(x=1, y = cumsum(per) - per/2, label=label))+
  ggtitle("Percentage of Expensive vs. Not Expensive Airbnbs for Staten
Island")
```



Top 10 Most Expensive Airbnbs

```
# arrange neighbourhood prices in decreasing order
med_price_neighborhoods <- airbnb %>%
  group_by(neighbourhood, neighbourhood_group) %>%
  summarize(median_price = median(price, na.rm = TRUE)) %>%
  arrange(desc(median_price))
# 10 most expensive airbnbs
most_exp = med_price_neighborhoods[1:10,]
p<-ggplot(most_exp, aes(x=neighbourhood, y=median_price, fill =
neighbourhood_group)) +
  geom_bar(stat="identity")+theme_minimal() + ggtitle("10 Most Expensive
Neighbourhoods by Median") + theme(axis.ticks.x=element_blank()) + labs(fill
= "Borough") + ylab("Median Price") + theme(text = element_text(size=10),
        axis.text.x = element_text(angle=90, hjust=1))
p
```



#we used median, averages too influenced by outliers/large values.