EDA. Case Study 2 - Group 4

Keru Wu

Contents

1. Load Data

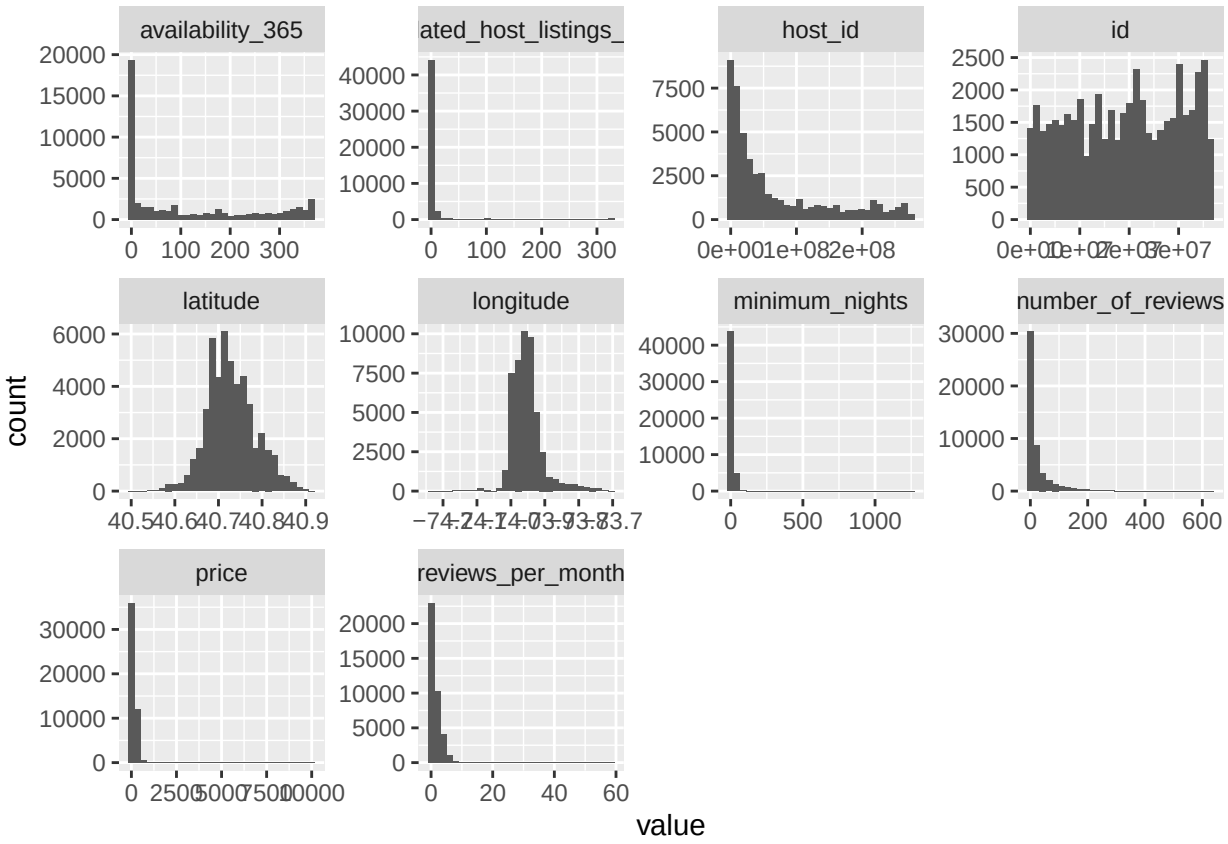
```
dat = read.csv('AB_NYC_2019.csv', na.strings = c("", "NA"))

library(purrr)
library(tidyr)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
dat %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram()
```

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

```
## Warning: Removed 10052 rows containing non-finite values (stat_bin).
```




```
lapply(dat, class)
```

```
## $id
## [1] "integer"
##
## $name
## [1] "factor"
##
## $host_id
## [1] "integer"
##
## $host_name
## [1] "factor"
##
## $neighbourhood_group
## [1] "factor"
##
## $neighbourhood
## [1] "factor"
##
## $latitude
## [1] "numeric"
##
## $longitude
## [1] "numeric"
##
## $room_type
```

```
## [1] "factor"
##
## $price
## [1] "integer"
##
## $minimum_nights
## [1] "integer"
##
## $number_of_reviews
## [1] "integer"
##
## $last_review
## [1] "factor"
##
## $reviews_per_month
## [1] "numeric"
##
## $calculated_host_listings_count
## [1] "integer"
##
## $availability_365
## [1] "integer"
```

2. Missing Data Manipulation

```
apply(dat, 2, function(x)(sum(is.na(x))))
```

```
##                             id                           name
##                              0                             16
##                        host_id                      host_name
##                              0                             21
##            neighbourhood_group                  neighbourhood
##                              0                              0
##                       latitude                      longitude
##                              0                              0
##                      room_type                          price
##                              0                              0
##                minimum_nights              number_of_reviews
##                              0                              0
##                   last_review             reviews_per_month
##                          10052                          10052
## calculated_host_listings_count               availability_365
##                              0                              0
```

```
dat = dat[, !names(dat) %in% c('id', 'host_name','last_review')]
dat$reviews_per_month[is.na(dat$reviews_per_month)] = 0
apply(dat, 2, function(x)(sum(is.na(x))))
```

```
##                             name                          host_id
##                               16                                0
##            neighbourhood_group                  neighbourhood
##                              0                                0
##                       latitude                      longitude
```

```
##                             0                             0
##                     room_type                         price
##                             0                             0
##                minimum_nights             number_of_reviews
##                             0                             0
##             reviews_per_month calculated_host_listings_count
##                             0                             0
##              availability_365
##                             0
```

3. EDA plots

```
library(jpeg)
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 3.5.2
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

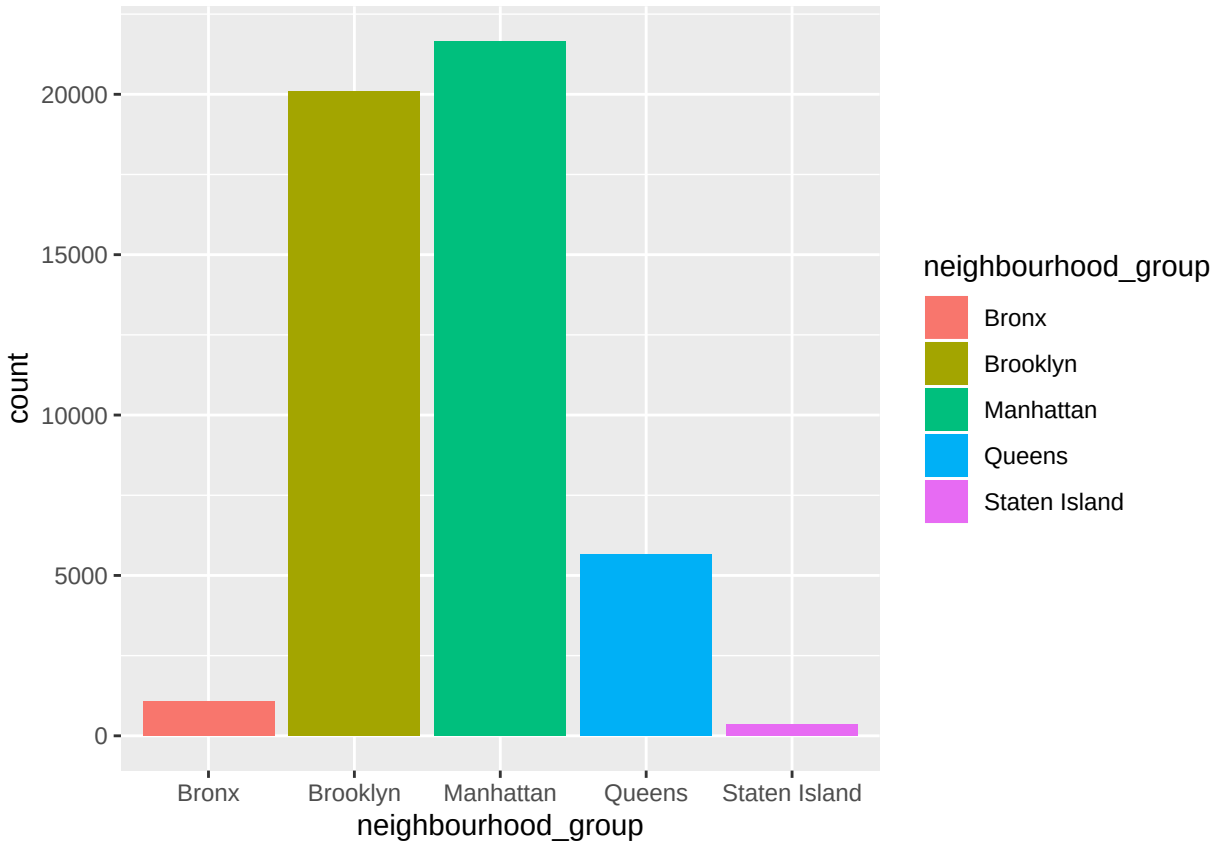
```
## The following object is masked from 'package:tidyr':
##
##     extract
```

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

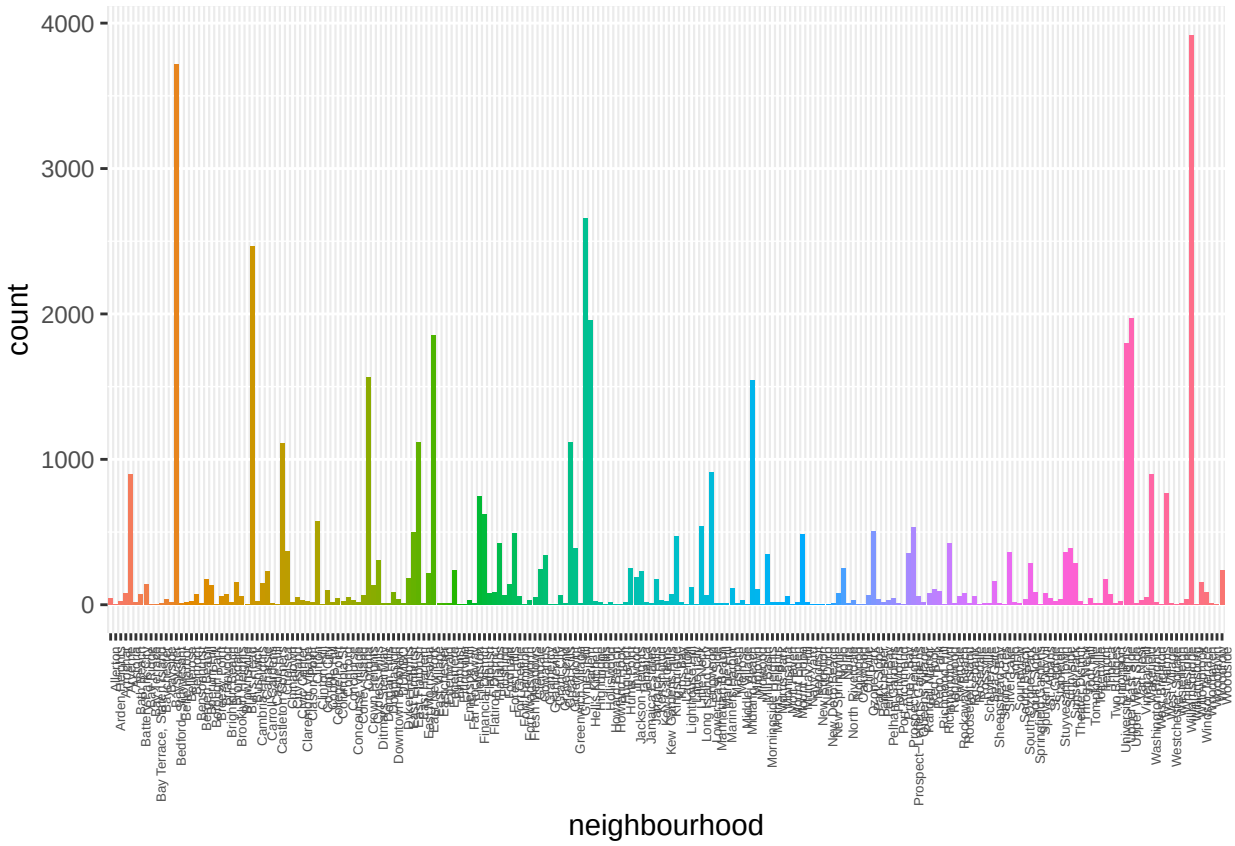
```
library(grid)

img = readJPEG("New_York_City_.jpg")
```

3.1 Neighbourhood Count

```
ggplot(dat, aes(x=neighbourhood_group)) +
  geom_bar(aes(fill=neighbourhood_group))
```

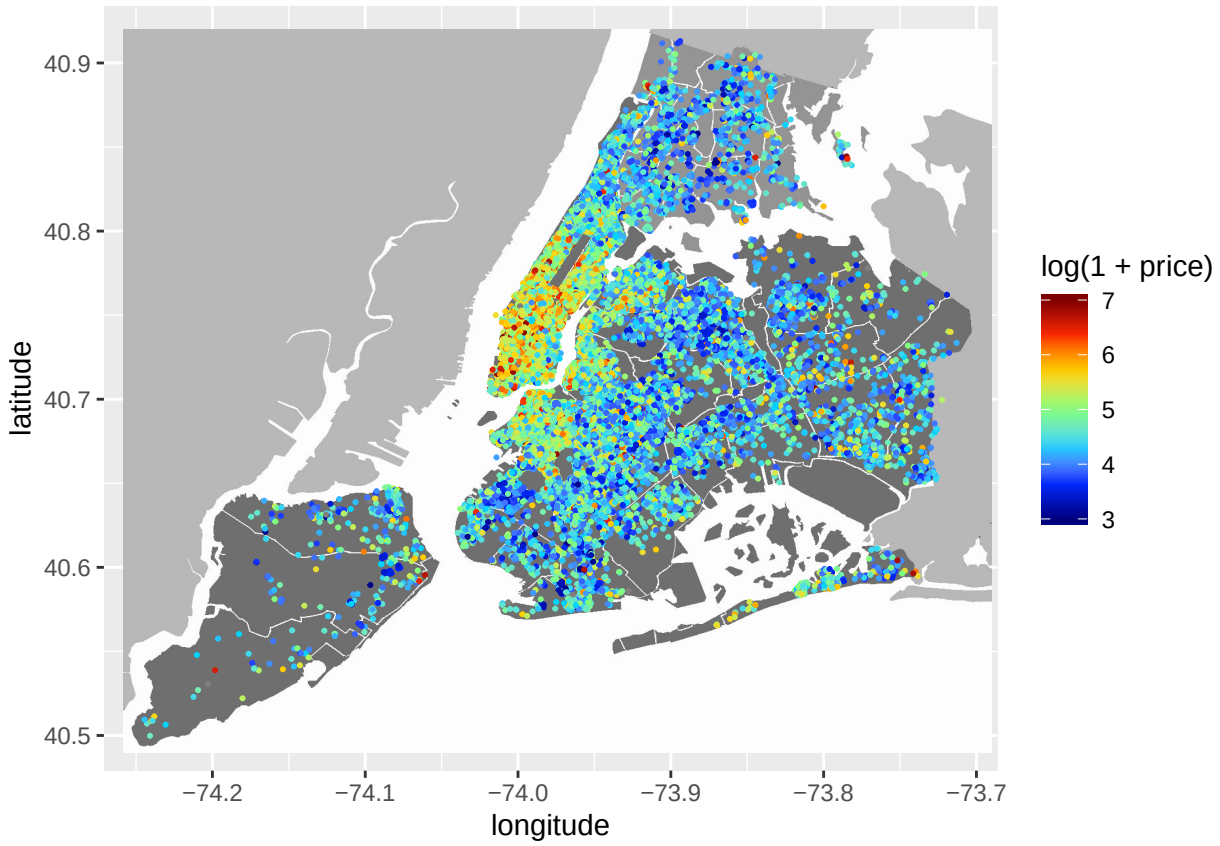



```
ggplot(dat, aes(x=neighbourhood)) +
  geom_bar(aes(fill=neighbourhood), show.legend = FALSE) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size=5))
```

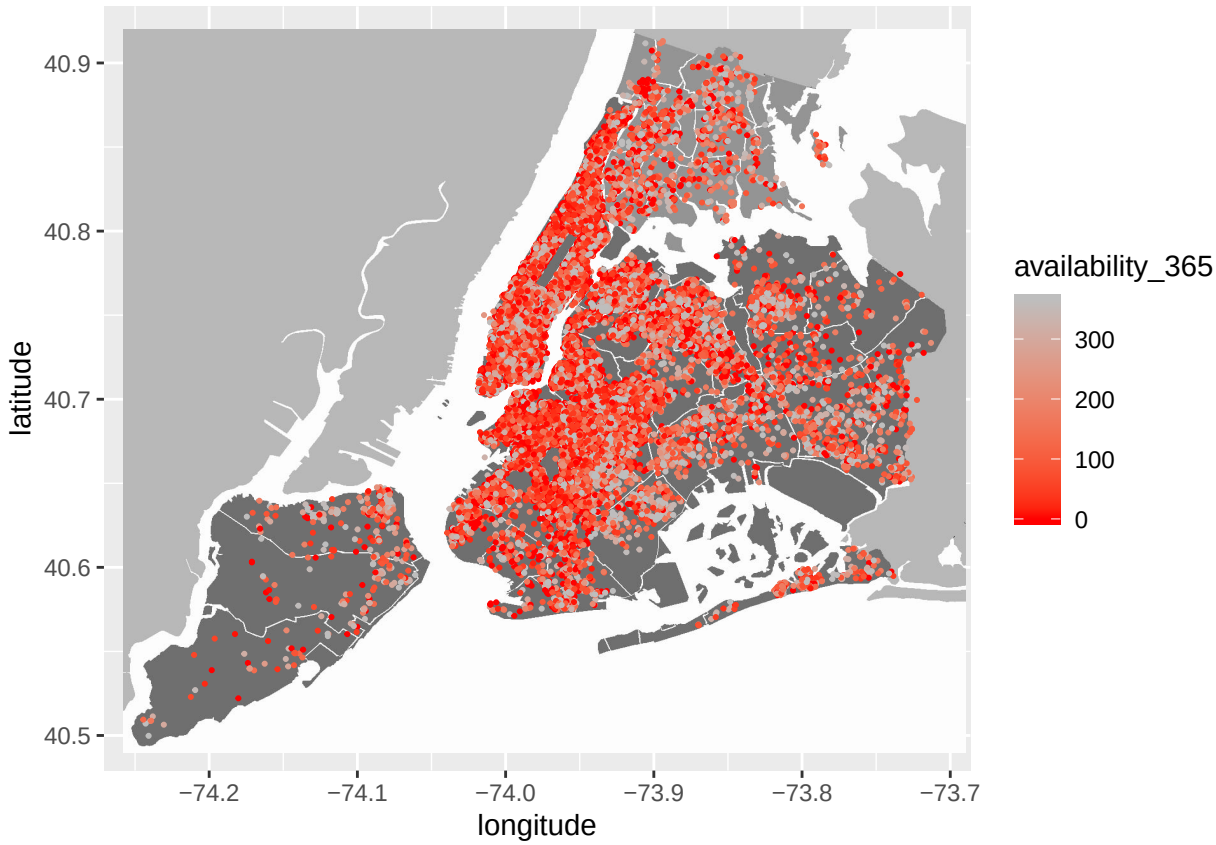


3.2 Maps

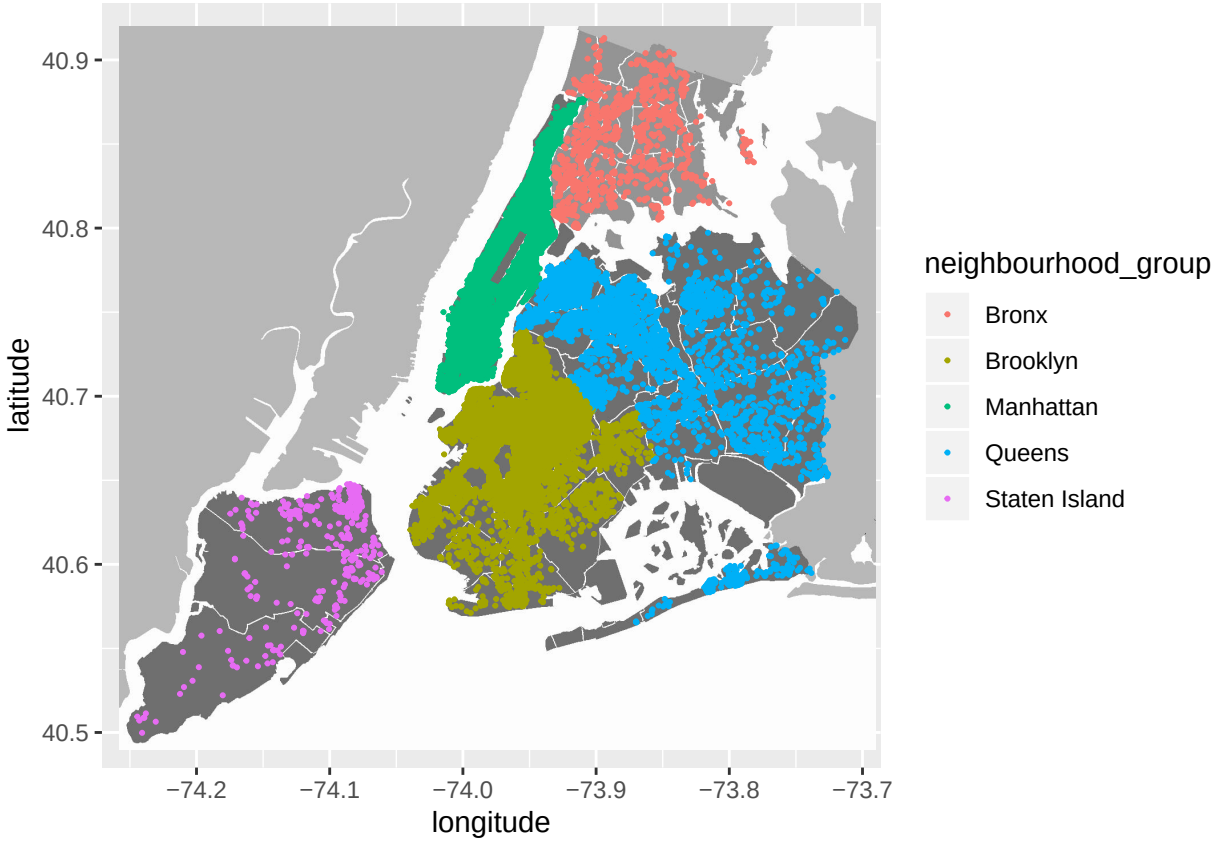
```
jet.colors <- colorRampPalette(c("#00007F", "blue", "#007FFF", "cyan", "#7FFF7F", "yellow", "#FF7F00",

ggplot(dat, aes(x=longitude, y = latitude, color = log(1+price)))+
  annotation_custom(rasterGrob(img,
                               width = unit(1,"npc"),
                               height = unit(1,"npc")),
                    -74.258, -73.69, 40.49,40.92) +
  geom_point(cex = 0.4) +
  scale_colour_gradientn(colors = jet.colors(7), limits = c(3,7))
```




```
ggplot(dat, aes(x=longitude, y = latitude, color = availability_365))+
  annotation_custom(rasterGrob(img,
                               width = unit(1,"npc"),
                               height = unit(1,"npc")),
                    -74.258, -73.69, 40.49,40.92) +
  geom_point(cex = 0.4) +
  scale_colour_gradient(low = 'red', high = 'grey')
```

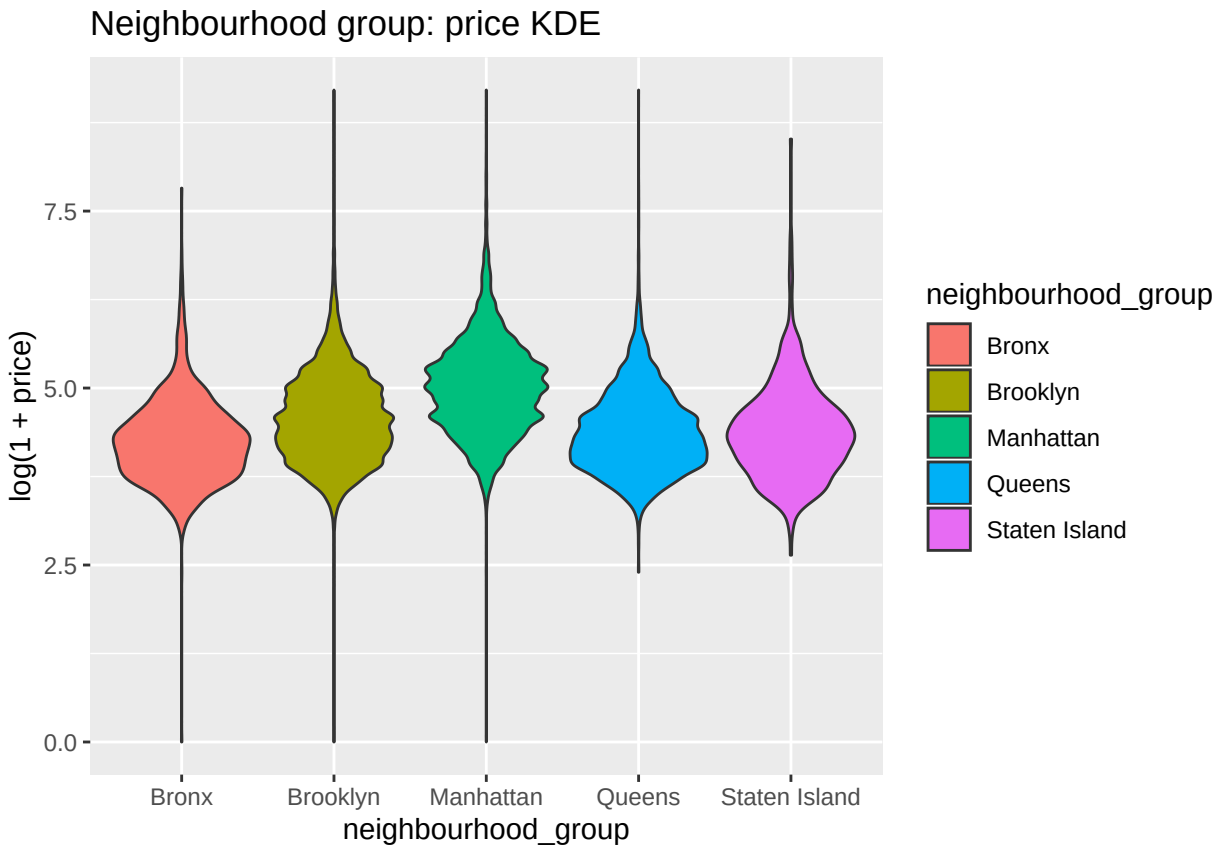



```
ggplot(dat, aes(x=longitude, y = latitude, color = neighbourhood_group))+
  annotation_custom(rasterGrob(img,
                               width = unit(1,"npc"),
                               height = unit(1,"npc")),
                    -74.258, -73.69, 40.49,40.92) +
  geom_point(cex = 0.4)
```

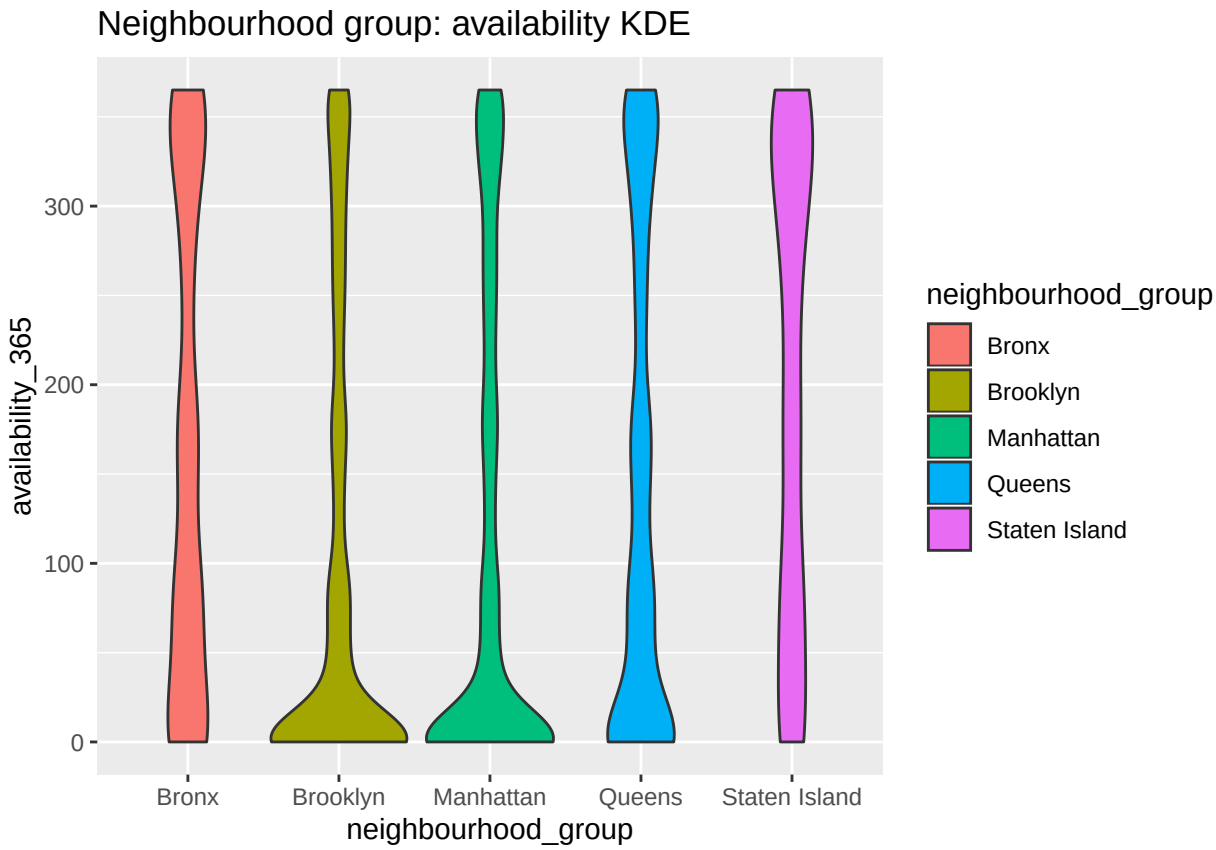


3.3 Neighbourhood Group

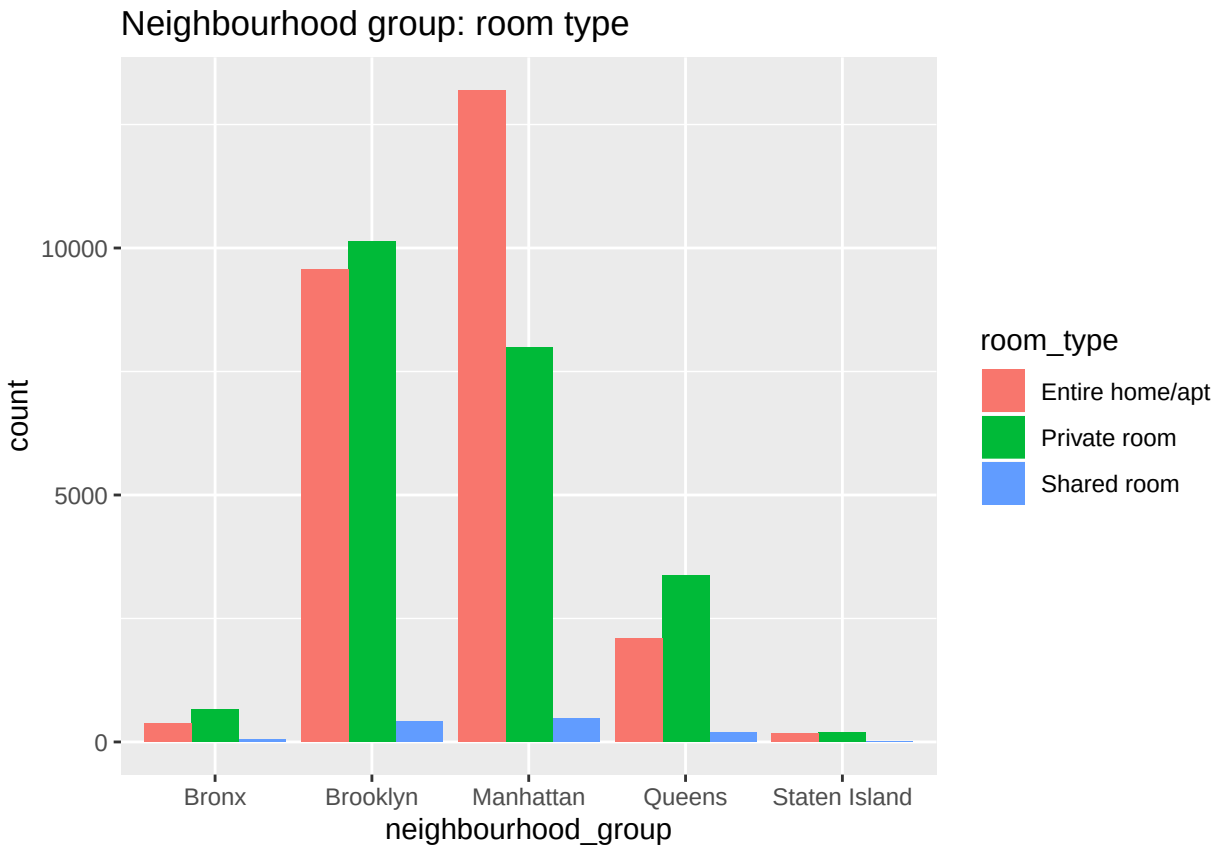
```
ggplot(dat, aes(x=neighbourhood_group, y = log(1+price), fill = neighbourhood_group))+
  geom_violin() +
  ggtitle('Neighbourhood group: price KDE')
```




```
ggplot(dat, aes(x=neighbourhood_group, y = availability_365, fill = neighbourhood_group))+
  geom_violin() +
  ggtitle('Neighbourhood group: availability KDE')
```

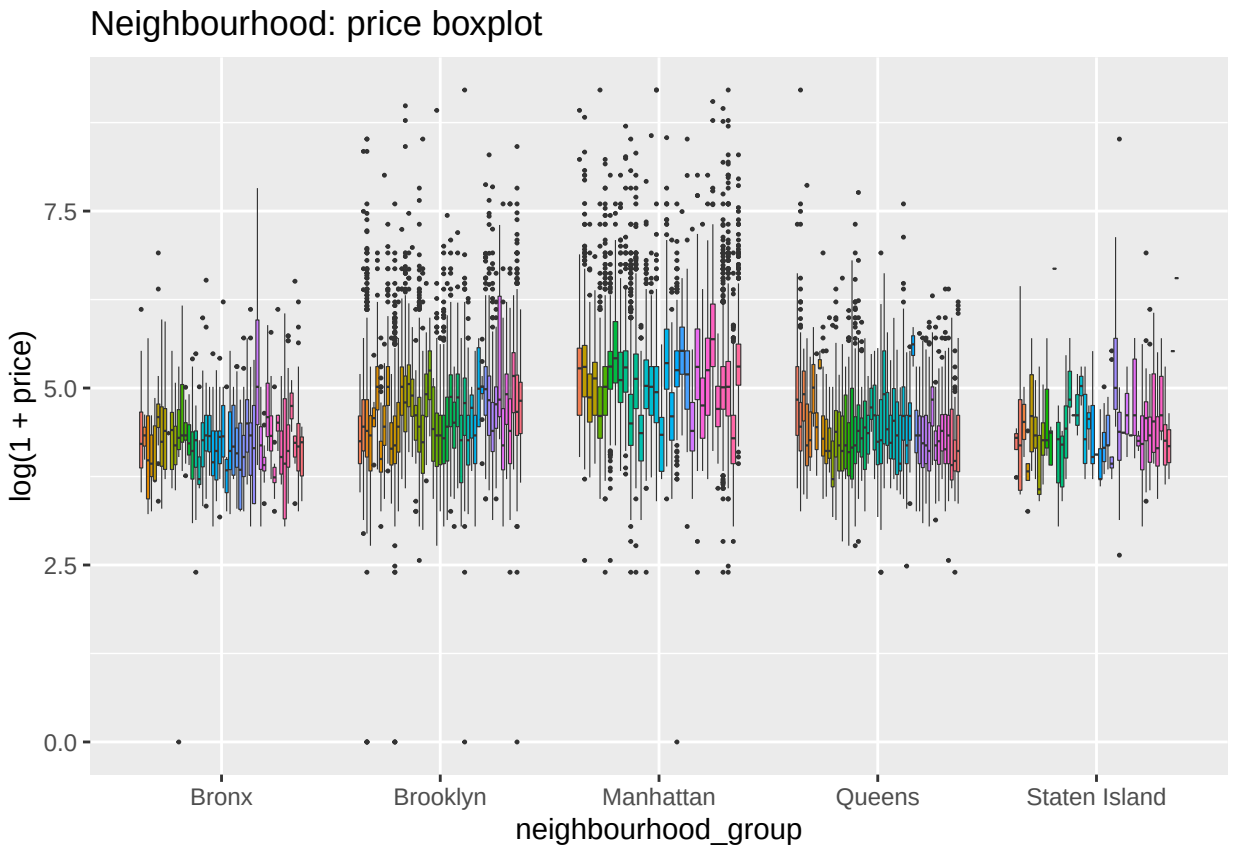



```
ggplot(dat, aes(x = neighbourhood_group))+
  geom_bar(aes(fill = room_type), position='dodge')+
  ggtitle("Neighbourhood group: room type")
```

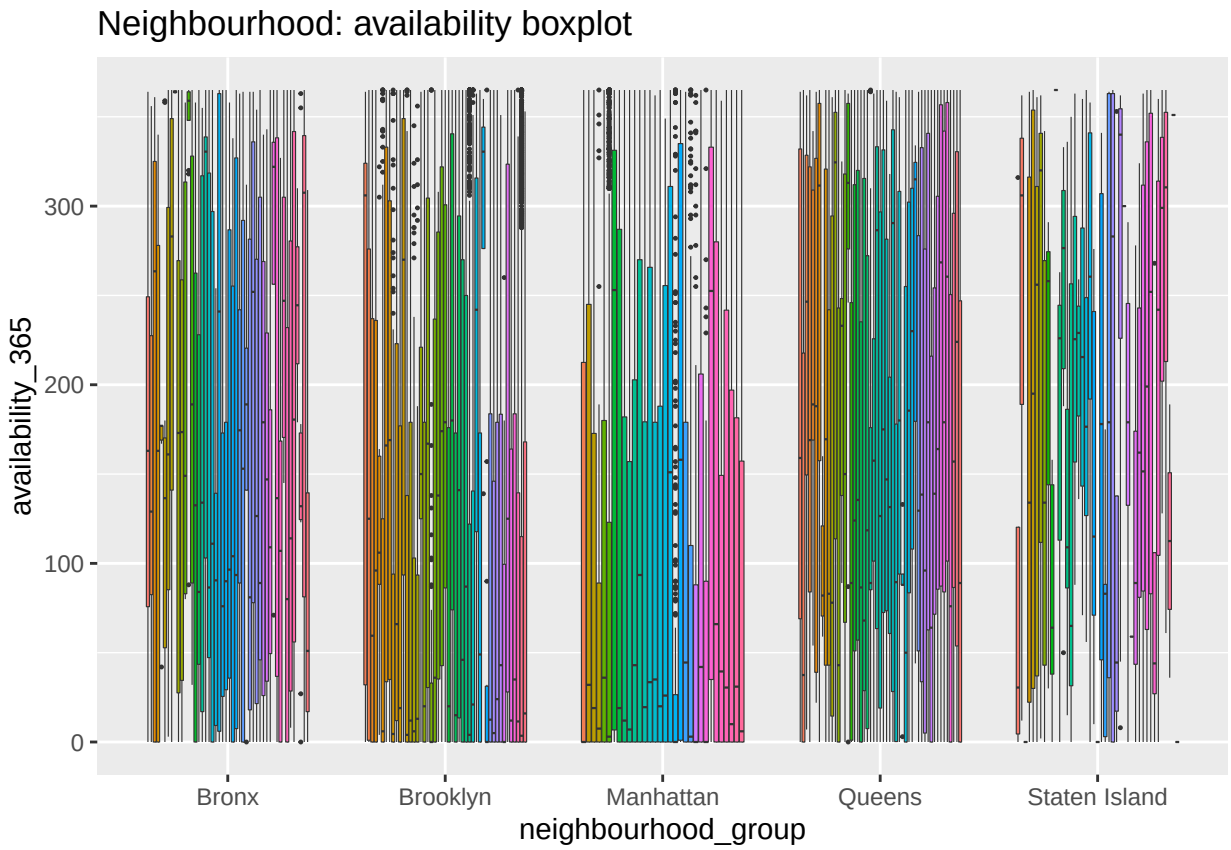


3.4 Neighbourhood

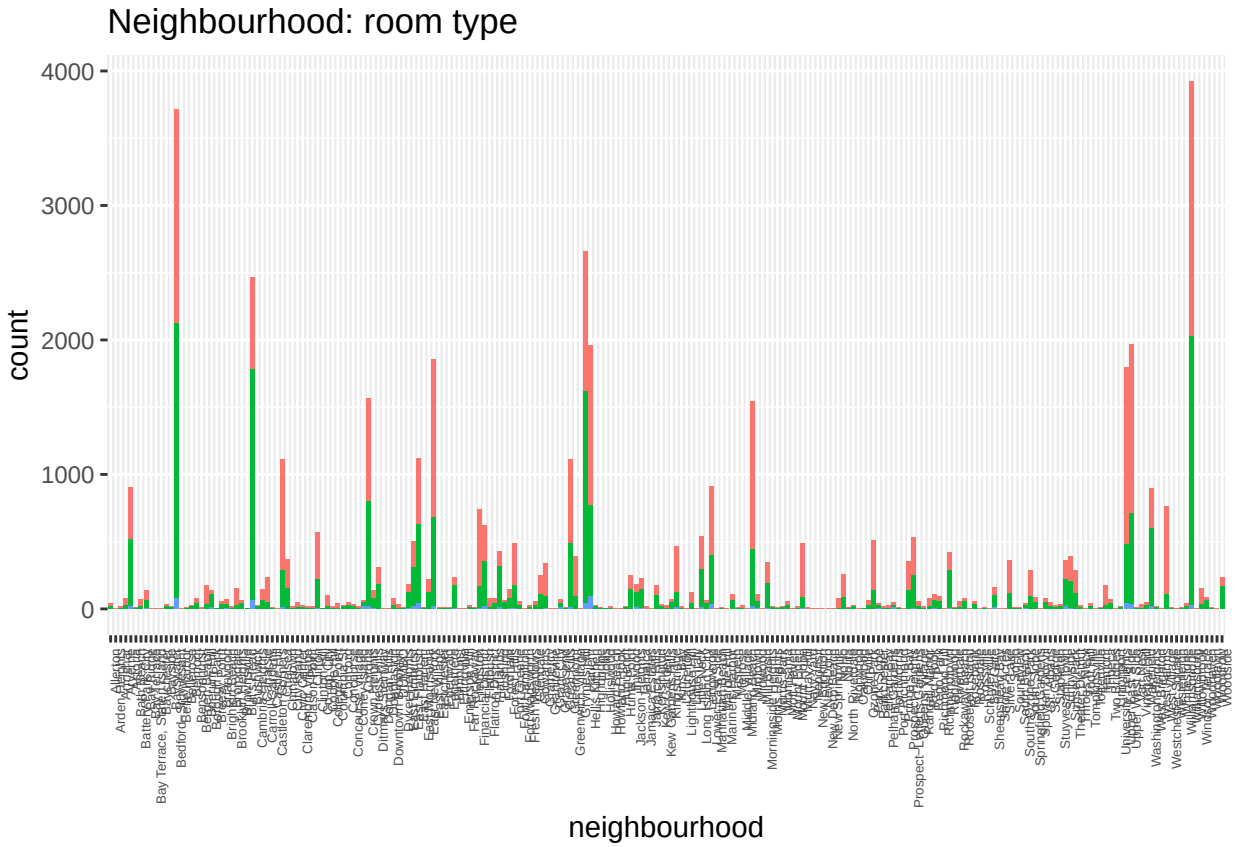
```
ggplot(dat, aes(x=neighbourhood_group, y = log(1+price), fill = neighbourhood))+
  geom_boxplot(show.legend = FALSE, outlier.size=0.2, lwd=0.2) +
  ggtitle('Neighbourhood: price boxplot')
```




```
ggplot(dat, aes(x=neighbourhood_group, y = availability_365, fill = neighbourhood))+
  geom_boxplot(show.legend = FALSE, outlier.size=0.2, lwd=0.2) +
  ggtitle('Neighbourhood: availability boxplot')
```




```
ggplot(dat, aes(x = neighbourhood))+
  geom_bar(aes(fill = room_type), show.legend = FALSE) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size=5))+
  ggtitle("Neighbourhood: room type")
```



3.5 Number of Reviews

```
dat1 <- dat[with(dat,order(-number_of_reviews)),]

dat1[1:10,]
```

```
##                                                 name  host_id
## 11760                        Room near JFK Queen Bed 47621202
## 2032                        Great Bedroom in Manhattan  4734398
## 2031                     Beautiful Bedroom in Manhattan  4734398
## 2016                       Private Bedroom in Manhattan  4734398
## 13496                        Room Near JFK Twin Beds 47621202
## 10624                Steps away from Laguardia airport 37312959
## 1880           Manhattan Lux Loft.Like.Love.Lots.Look !  2369681
## 20404 Cozy Room Family Home LGA Airport NO CLEANING FEE 26432133
## 4871                 Private brownstone studio Brooklyn 12949460
## 472                    LG Private Room/Family Friendly   792159
##       neighbourhood_group   neighbourhood latitude longitude
## 11760              Queens         Jamaica 40.66730 -73.76831
## 2032            Manhattan          Harlem 40.82085 -73.94025
## 2031            Manhattan          Harlem 40.82124 -73.93838
## 2016            Manhattan          Harlem 40.82264 -73.94041
## 13496              Queens         Jamaica 40.66939 -73.76975
## 10624              Queens   East Elmhurst 40.77006 -73.87683
## 1880            Manhattan Lower East Side 40.71921 -73.99116
```

```
## 20404              Queens   East Elmhurst 40.76335 -73.87007
## 4871             Brooklyn      Park Slope 40.67926 -73.97711
## 472              Brooklyn        Bushwick 40.70283 -73.92131
##             room_type price minimum_nights number_of_reviews
## 11760    Private room    47              1               629
## 2032     Private room    49              1               607
## 2031     Private room    49              1               597
## 2016     Private room    49              1               594
## 13496    Private room    47              1               576
## 10624    Private room    46              1               543
## 1880     Private room    99              2               540
## 20404    Private room    48              1               510
## 4871  Entire home/apt   160              1               488
## 472      Private room    60              3               480
##       reviews_per_month calculated_host_listings_count availability_365
## 11760             14.58                               2              333
## 2032               7.75                               3              293
## 2031               7.72                               3              342
## 2016               7.57                               3              339
## 13496             13.40                               2              173
## 10624             11.59                               5              163
## 1880               6.95                               1              179
## 20404             16.22                               5              341
## 4871               8.14                               1              269
## 472                6.70                               1                0
```

4. Word Count

```
Names = paste(dat$name, collapse = " ")
Names = strsplit(Names, " ")[[1]]
Names = table(Names)

library(dplyr)
```

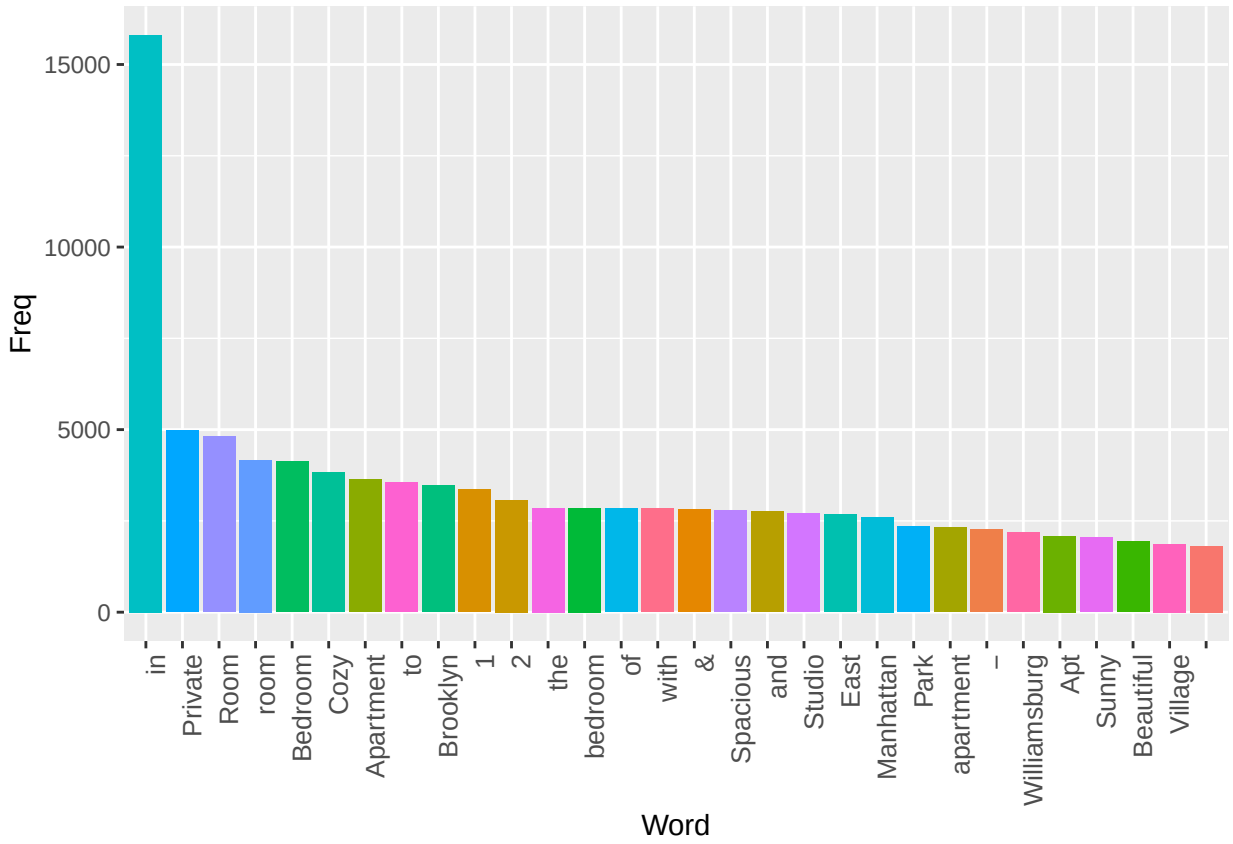
```
## Warning: package 'dplyr' was built under R version 3.5.2
```

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

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

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

```
freq = Names %>% as.data.frame() %>% arrange(desc(Freq))

ggplot(freq[1:30,], aes(x = reorder(Names, -Freq), y = Freq))+
  geom_bar(stat = "identity", aes(fill = Names), show.legend = FALSE)+
  xlab("Word")+
  theme(axis.text.x = element_text(angle = 90, hjust = 1, size=10))
```



5. Regression

5.0 Train Test

```
dat = dat %>% mutate(id = row_number())
train = dat %>% sample_frac(.7)
test  <- anti_join(dat, train, by = 'id')
```

5.1 Linear Regression

5.2 LASSO

5.3 GAM

5.4 Decision Tree & Random Forest