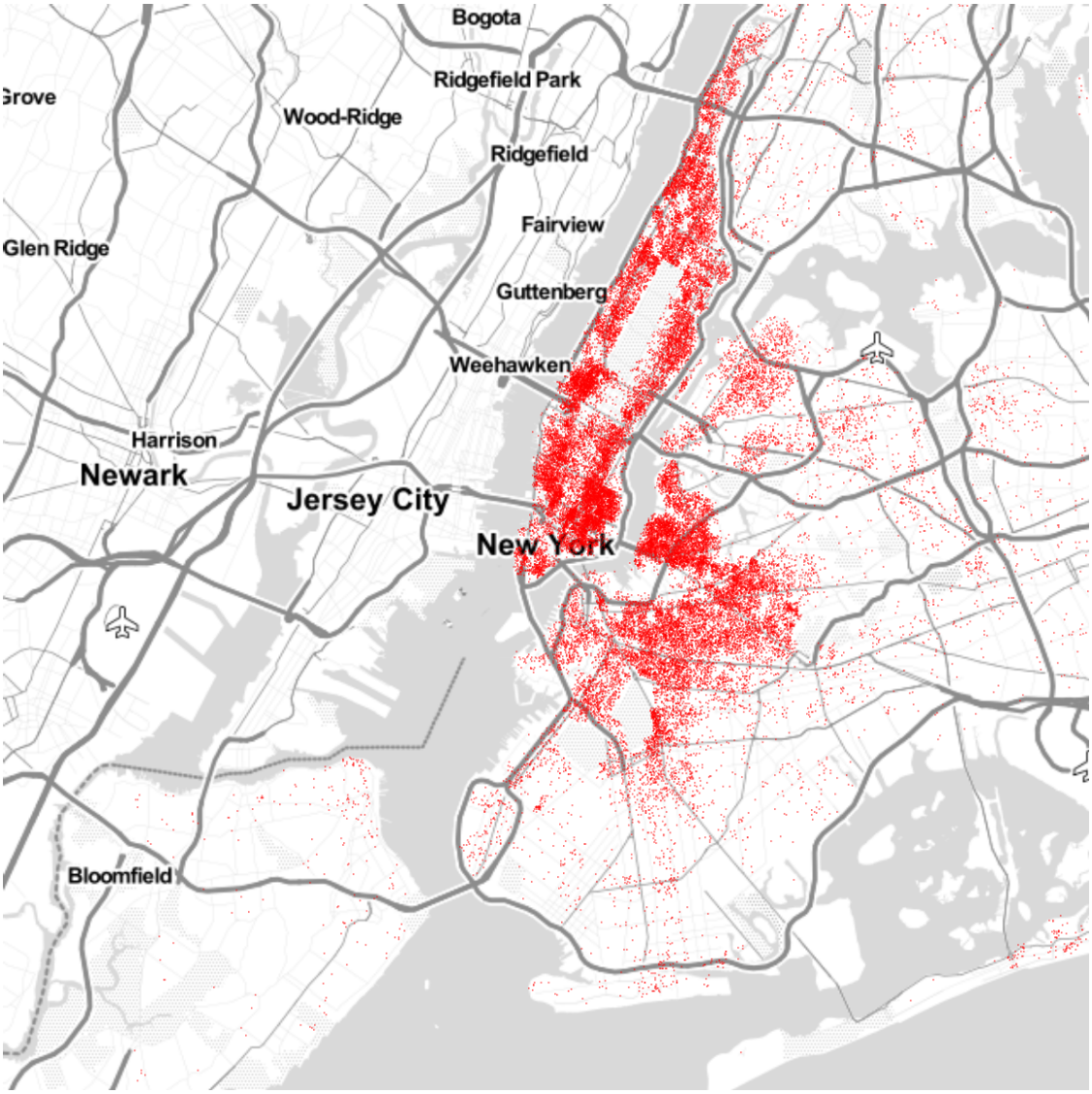
Assignment 2

Brandon Wolff

March 7, 2017

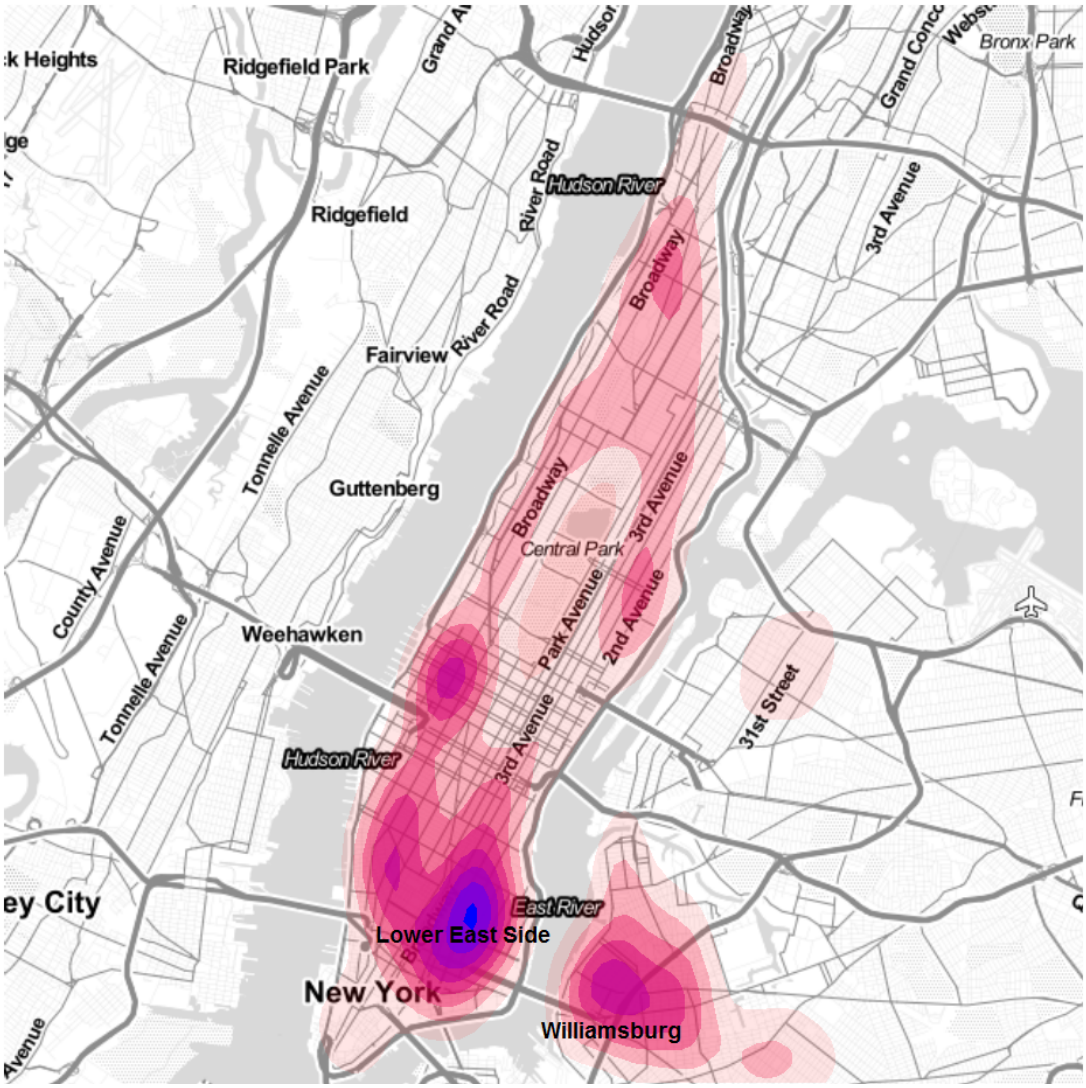
Assignment 2

Airbnb Overall Locations in NYC



This map shows that there are a large sum of Airbnb listing in New York City. When looking more closely at the map it appears that most airbnb listings are in or very close to Manhattan. This could be good news for tourists.

Airbnb Listingins Density Map NYC

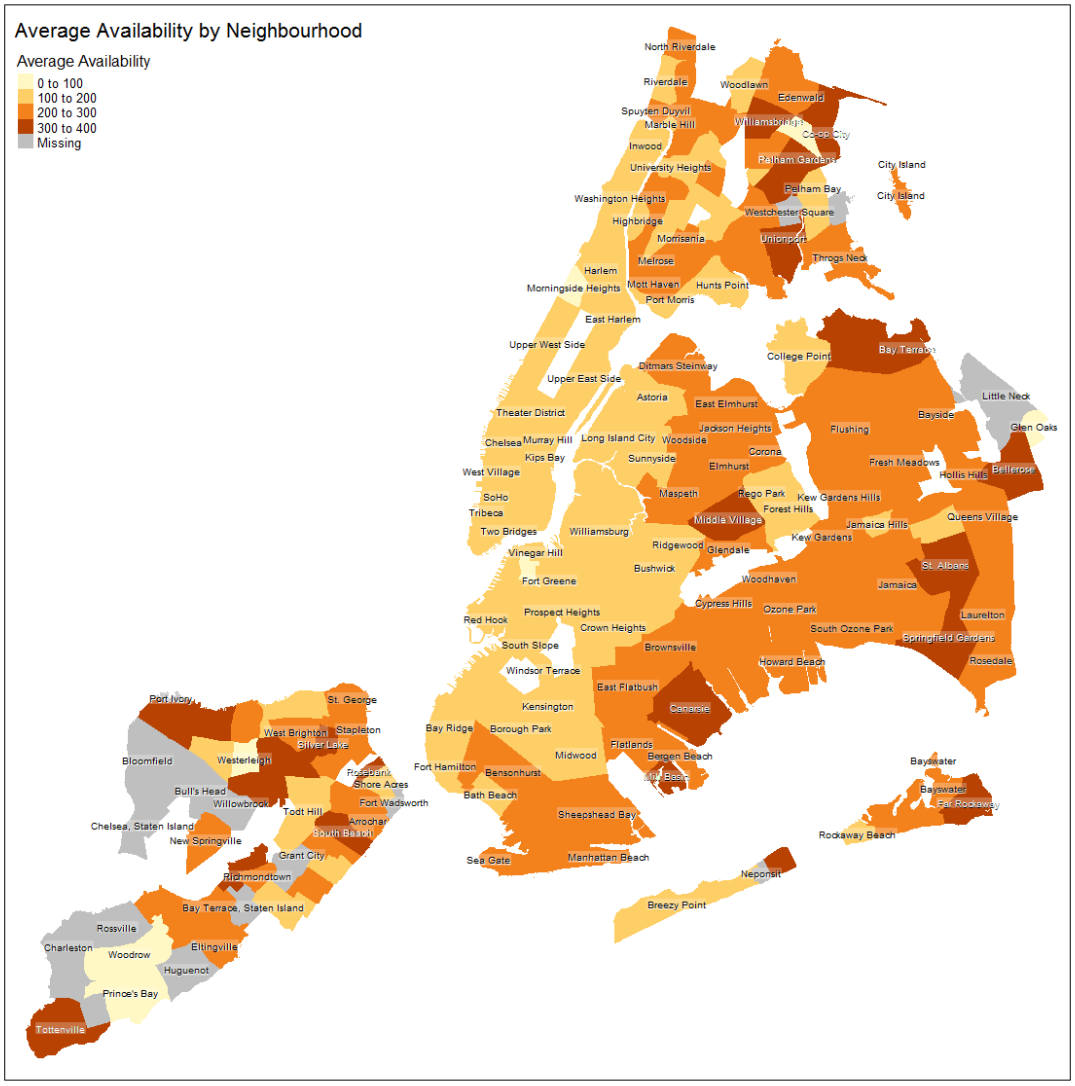


We get a more clear picture of where the most Airbnb listing seem to be located when looking at this Density map. We see that the most populated area seems to be the Lower East Side in Manhattan. The Lower East Side is not to far from tourist attractions like Times Square so it could be a beneficial area to stay for tourists. When looking at the map it can be noted that there are a number of bridges near this area which allows individuals to get from borough to borough more easily. This may be a reason why there are also a large sum of airbnb listings in Williamsburg.

Rental Availability

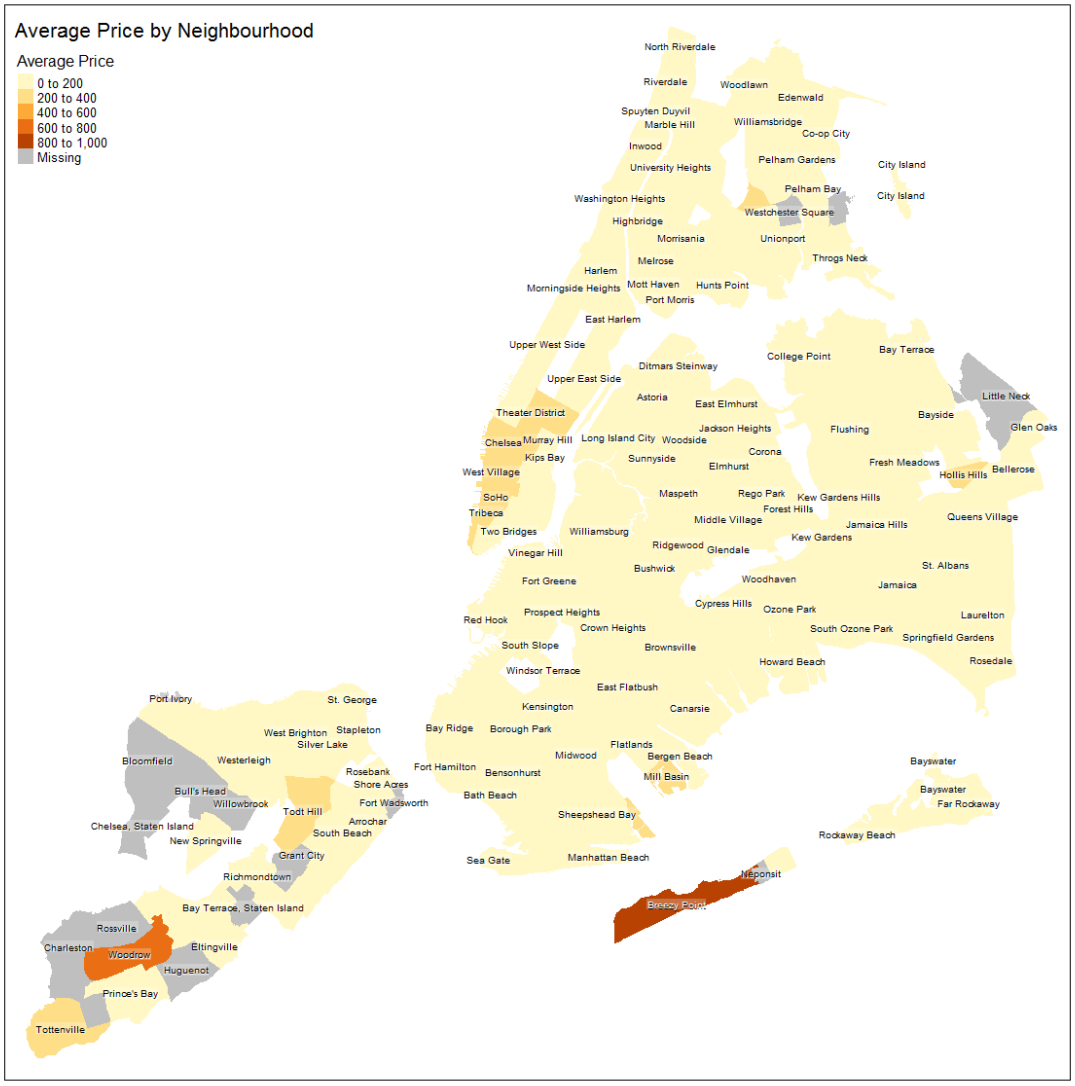
```
## OGR data source with driver: GeoJSON
## Source: "C:\Users\Brandon\Documents\GitHub\QMSS-G4063-Data-Visualiza
```

```
tion\Assignments\Assignment 2\neighbourhoods.geojson", layer: "OGRGeoJS
ON"
## with 233 features
## It has 2 fields

## OGR data source with driver: GeoJSON
## Source: "C:\Users\Brandon\Documents\GitHub\QMSS-G4063-Data-Visualiza
tion\Assignments\Assignment 2\neighbourhoods.geojson", layer: "OGRGeoJS
ON"
## with 233 features
## It has 2 fields
```



This map above displays the average availability by neighborhoods for all of New York City. It appears that the listings in Manhattan seem to be available for between 100 - 200 days of the year. On the other hand the availability in all other boroughs

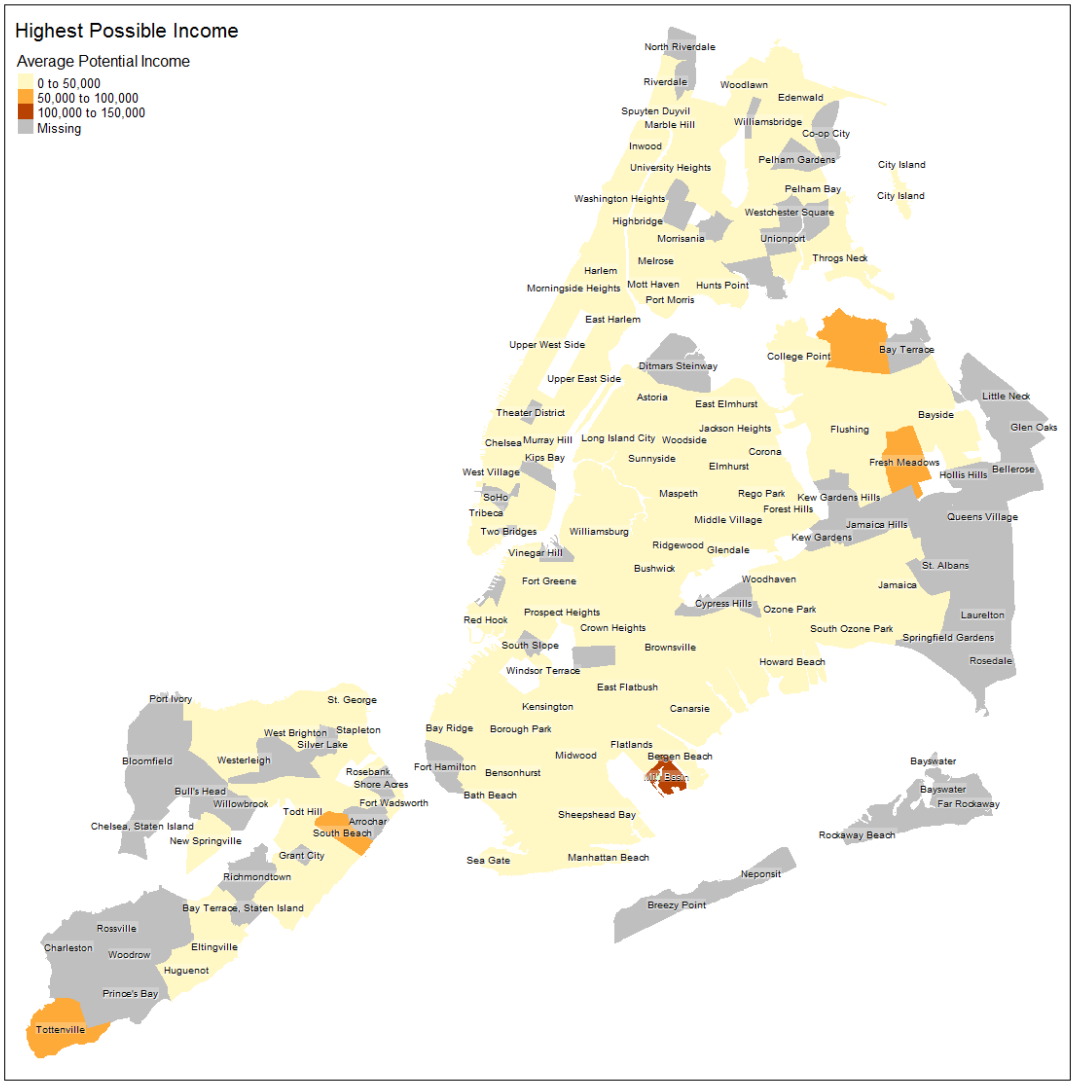
appears to be higher with some neighborhoods being available for more than 300 days of the year.



This map displays the average Price of Airbnb listings by neighborhood. It appears that it is very expensive to stay in Breezy Point, costing between $800 - $1000 a night! The price in this area may be so high because of the size of the homes or potentially the view. We also notice that the Woodrow neighbourhood is also more pricey than most neighborhoods with the nightly rate being between $600 - $800. When looking at Manhattan, the borough most tourists want to stay, there appears to be a group of neighborhoods that cost between $200 - $400 in comparison to the rest of Manhattan which costs only $0 - $200 nightly. The more pricey neighbourhoods of Manhattan are near Times Square and Hell's Kitchen and this could be a potential reason for the higher price.

```
## OGR data source with driver: GeoJSON
## Source: "C:\Users\Brandon\Documents\GitHub\QMSS-G4063-Data-Visualiza
```

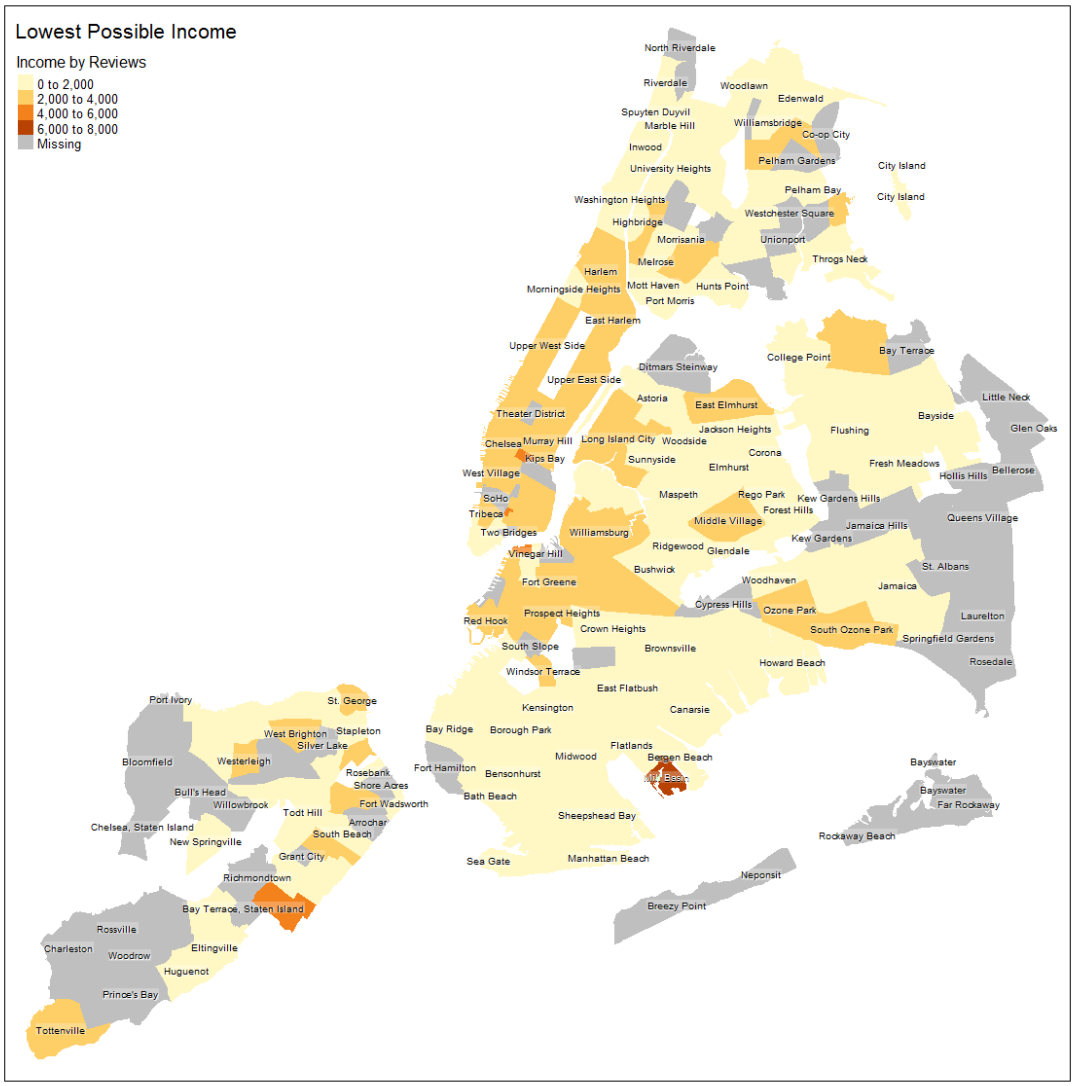
```
tion\Assignments\Assignment 2\neighbourhoods.geojson", layer: "OGRGeoJS
ON"
## with 233 features
## It has 2 fields
```



Now that we have examined price and availability we can go a step further and try to estimate how much income each neighborhood has on average. To do this we must make some assumption due to the lack of information. In this above map the assumption is that the Airbnb listings are booked every single day they are available. So this is actually measuring the Potential Income for each neighborhood. It can be noted that the Mil Basin neighbourhood has by far the most potential Income! There are four other neighborhoods that also have a fairly high potential income.

Potential income is assuming these listings are being booked every single possible day, which is problematic. In order to get a more clear picture of the average income

of Airbnb listings by neighborhood we will not only examine the maximum possible income but also the minimum possible income. In order to do this we will be assuming that the Airbnb listings were only booked if there is an actual review as evidence, to do this we will be using the number of reviews each listing has.



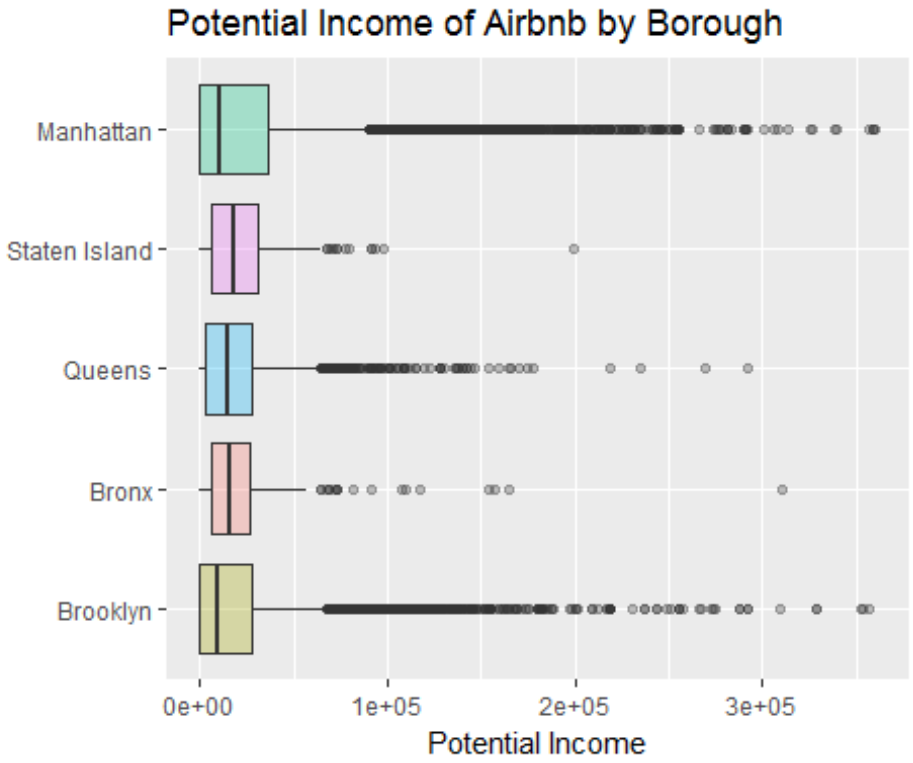
This map is showing us the average income for Airbnb listings by neighborhood based on the number of reviews. We see that Mil Basin still stands out the most with an average income of $6000 - $8000. When looking at the lowest possible income and the highest possible income it appears that Mil Basin stands out. Below is a summary of the highest possible income and the lowest possible income and the Max. in each is representing the income of Mil Basin. Mil Basin makes between $8000 - $149500!

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0   15440   19800   22420   25280  149500      73
```

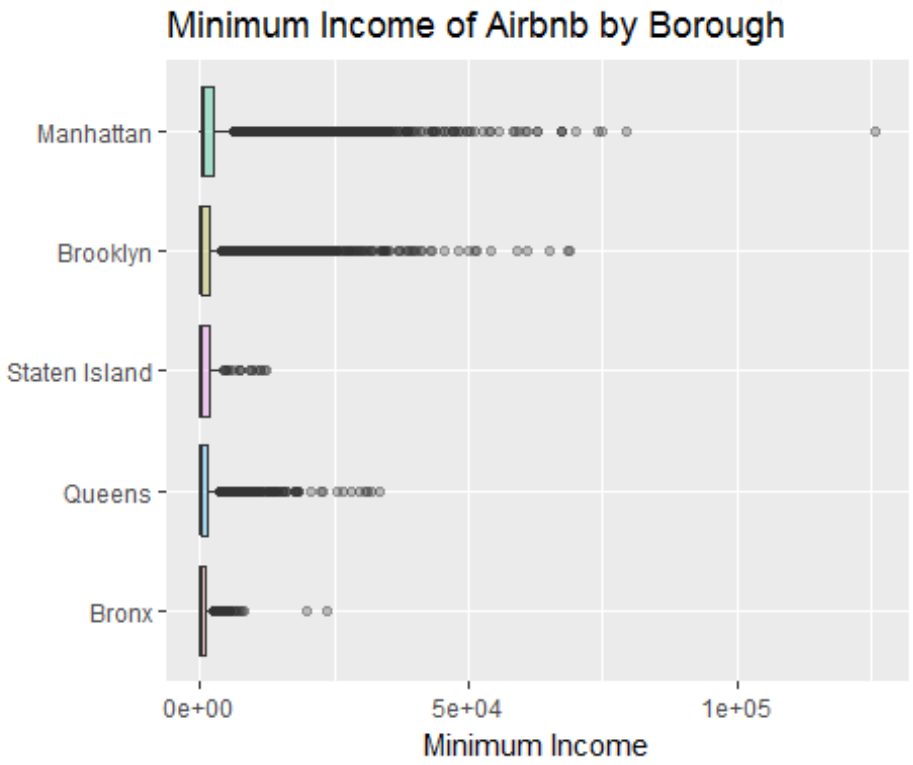
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0   802.2  1247.0  1572.0  2142.0  8000.0      73
```

Income by Borough

It is also informative to examine the potential income of Airbnb listings by New York City Borough.



We see that Manhattan seems to have the highest variability of potential income. Again the income is assuming the listing is booked every available night. Staten Island appears to have the highest potential income. Now let's again compare this information to Income based on number of reviews.



Overall, Airbnb listings do not get a large sum of reviews so there is not a lot of variability here. All boroughs have very similar average incomes based on the number of reviews. Taking this information into account one might assume that Staten Island has the highest potential income.

```
##
## Call:
## lm(formula = potential_income ~ dummy + neighbourhood_group_cleansed
+
##     price + bathrooms + bedrooms, data = airbnb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -175252  -13072   -3388   14529  190568
##
## Coefficients:
##                                          Estimate Std. Error t val
ue
## (Intercept)                             -2619.124   1072.663  -2.4
42
## dummy                                   28736.938    626.267  45.8
86
## neighbourhood_group_cleansedBrooklyn      -4042.679   1017.617  -3.9
73
## neighbourhood_group_cleansedManhattan     -5683.348   1023.053  -5.5
55
```

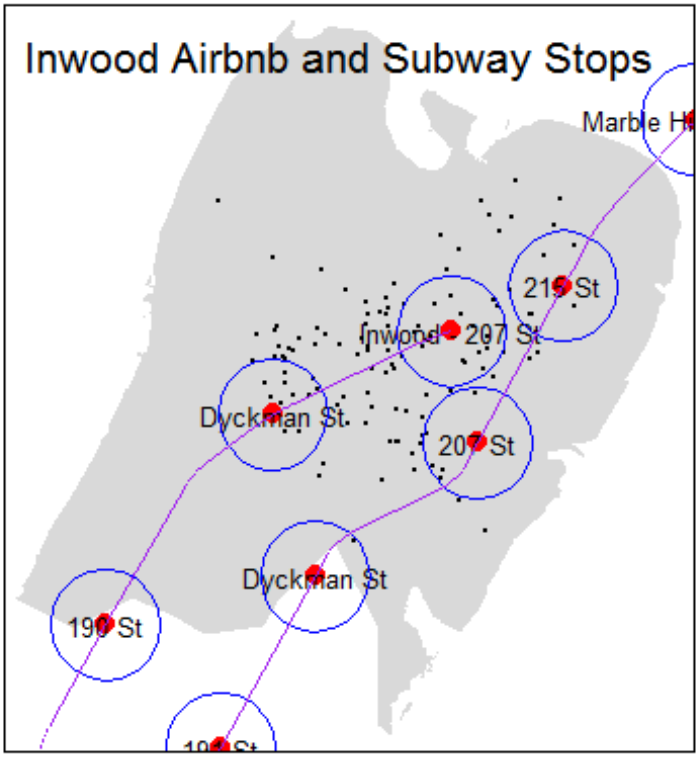
```
## neighbourhood_group_cleansedQueens          -251.174   1081.002  -0.2
32
## neighbourhood_group_cleansedStaten Island  1031.690   1968.829   0.5
24
## price                                       176.076      1.417 124.2
89
## bathrooms                                  -533.172    373.807  -1.4
26
## bedrooms                                   3540.091    225.435  15.7
03
##                                                    Pr(>|t|)
## (Intercept)                                          0.0146 *
## dummy                                              < 2e-16 ***
## neighbourhood_group_cleansedBrooklyn               7.12e-05 ***
## neighbourhood_group_cleansedManhattan              2.79e-08 ***
## neighbourhood_group_cleansedQueens                   0.8163
## neighbourhood_group_cleansedStaten Island            0.6003
## price                                              < 2e-16 ***
## bathrooms                                            0.1538
## bedrooms                                           < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24950 on 39776 degrees of freedom
##   (442 observations deleted due to missingness)
## Multiple R-squared:  0.3961, Adjusted R-squared:  0.396
## F-statistic:  3261 on 8 and 39776 DF,  p-value: < 2.2e-16
```

A linear regression was run on the potential income of airbnb listings. There were a number of control variables added to account for price fluctuations based on bathrooms and bedrooms. We see that Queens and staten Island have no statistical significance in potential income compared to the Bronx. We also see that Brooklyn appears to have a potential income of $4042 less on average than the Bronx and Manhattan appears to have a potential income of $5683 less on average than the Bronx, with a p-value of less than 0.01 these results are statistically significant.

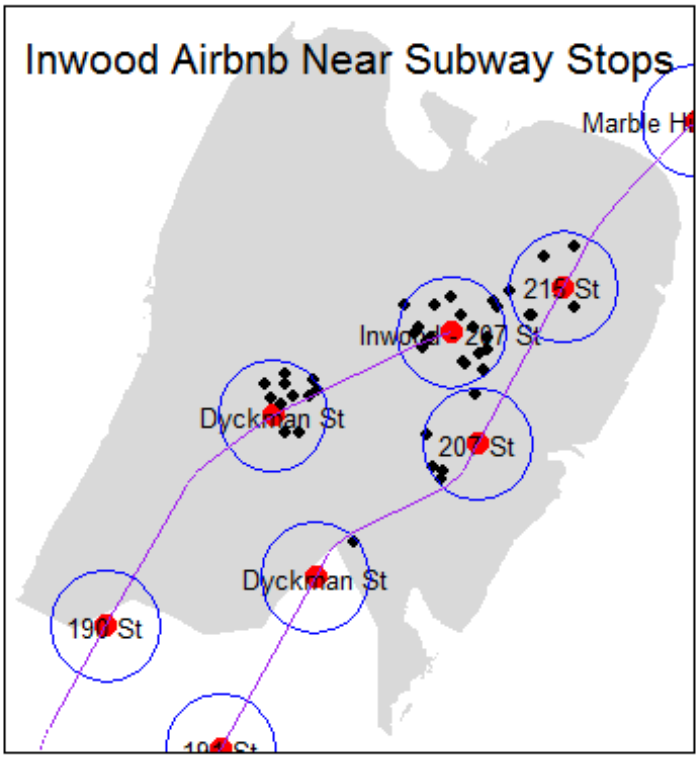
Airbnb and Subway Access

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\Brandon\Documents\GitHub\QMSS-G4063-Data-Visualiza
tion\Assignments\Assignment 2\nyc_subway_map\stops_nyc_subway.", layer:
"stops_nyc_subway_jan2017"
## with 493 features
## It has 8 fields

## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\Brandon\Documents\GitHub\QMSS-G4063-Data-Visualization\Assignments\Assignment 2\nyc_subway_map\routes_nyc_subway.", layer
: "routes_nyc_subway_jan2017"
## with 25 features
```

```
## It has 7 fields
## Integer64 fields read as strings:  OBJECTID

## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\Brandon\Documents\GitHub\QMSS-G4063-Data-Visualiza
tion\Assignments\Assignment 2\nyc_subway_map\entrances_nyc_subway.", la
yer: "subway_entrances_may2016"
## with 1868 features
## It has 32 fields
## Integer64 fields read as strings:  Route_8 Route_9 Route_10 Route_11
```



The map above is of the Manhattan neighborhood Inwood. The purple lines represent the path of the subways in the neighborhood and the red dots represent the actual subway stops. The blue circles around the stops represent a distance of 190 meters from the subway stop. The black dots are all of the actual Airbnb listings. When looking at the map above it appears that most of the Airbnb listings in Inwood are fairly close to the subway stops. There does not appear to be much of any listings that are not somewhat near a subway stop.



The map above is of only the Airbnb listening that are within 190 meters of the subway stops. There are a total of 42 out of 117 listings that are at least 190 meters away from a subway stop. One might assume that being closer to the subway stops might allow the Airbnb to listening to have a higher price.

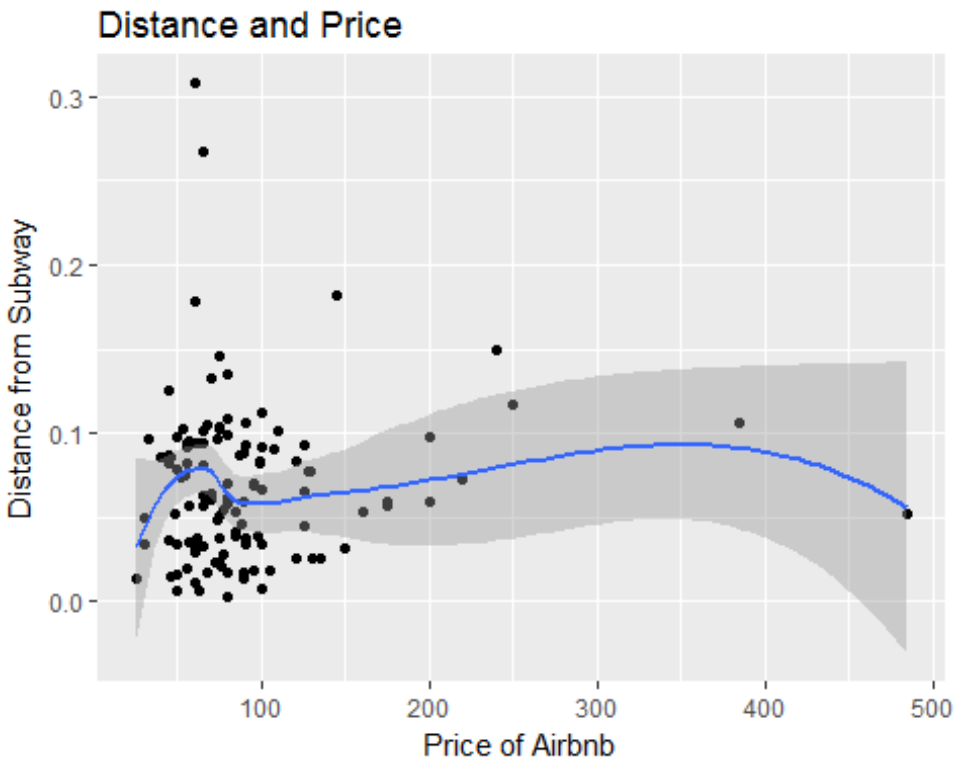
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25.00   60.00   75.00   85.57   99.00  250.00
```

Printed above is the range of price for those Airbnb listings farther than 190 meters from the subway stops. These listenings cost between $25 - $250 per night. Printed below is range of price for those Airbnb listings within 190 meters of the subway stops. They actually have a higher minimum of $30 and a higher maximum of $485. Even the mean price of those within 190 meters are higher at $103.60 in comparison to $85.57. The price of listening seems to be affected by the distance from subway stops but is this statistically significant?

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   30.00   57.75   78.50  103.60  100.00  485.00

##
## Call:
## lm(formula = price ~ distance + bathrooms + bedrooms + number_of_rev
iews +
##     bed_type + beds + accommodates + room_type + property_type,
##     data = inwood)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -103.450  -19.429   -3.346   14.930  130.716
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -75.1570    32.7047  -2.298  0.02358 *
## distance                  60.9649    76.5300   0.797  0.42751
## bathrooms                101.5783    20.8322   4.876 3.95e-06 ***
## bedrooms                  68.6562    10.3029   6.664 1.34e-09 ***
## number_of_reviews         -0.1407     0.1558  -0.903  0.36861
## bed_typeCouch             -7.0549    49.1702  -0.143  0.88619
## bed_typeFuton             22.5119    31.1043   0.724  0.47086
## bed_typePull-out Sofa     -6.9761    48.9171  -0.143  0.88688
## bed_typeReal Bed         -11.5427    26.1035  -0.442  0.65928
## beds                      -9.9151     7.6535  -1.295  0.19805
## accommodates               8.8412     4.6735   1.892  0.06133 .
## room_typePrivate room    -24.6871     7.9702  -3.097  0.00252 **
## room_typeShared room     -59.5501    22.6186  -2.633  0.00977 **
## property_typeDorm        104.7478    35.7379   2.931  0.00416 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.17 on 103 degrees of freedom
## Multiple R-squared:  0.712,  Adjusted R-squared:  0.6757
## F-statistic: 19.59 on 13 and 103 DF,  p-value: < 2.2e-16
```

Above we see a multiple regression on Price of Airbnb listings and Distance from subway stops for Inwood. To control for other possible reasons for differing price we have added the number of bathrooms, bedrooms, accommodates,and reviews, as well as controlling for type of room, type of bed, and type of property. Once we have accounted for all these other factors related to price we see that the Distance from the subway stops does not have a significant effect on the price of listings in Inwood.

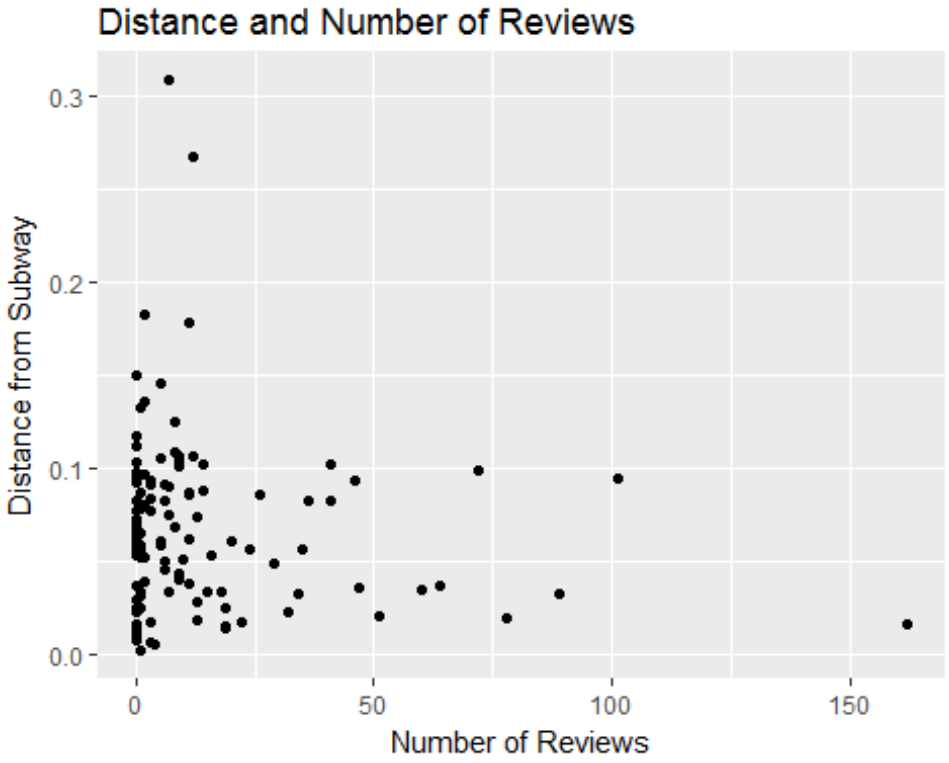


This graph makes it more clear that there does not appear to be a concrete relationship between Airbnb listing price and distance from the subway stops.

```
##
## Call:
## lm(formula = distance ~ price + bathrooms + bedrooms + number_of_rev
iews +
##     bed_type + beds + accommodates + room_type + property_type +
##     potential_income + low_income, data = inwood)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.072838 -0.022770 -0.001572  0.017612  0.210848
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -4.342e-03  4.854e-02  -0.089  0.92889
## price                  1.142e-04  1.776e-04   0.643  0.52154
## bathrooms             -3.433e-02  3.356e-02  -1.023  0.30884
## bedrooms               6.217e-03  1.629e-02   0.382  0.70360
## number_of_reviews     -6.284e-04  8.016e-04  -0.784  0.43496
## bed_typeCouch          1.257e-01  6.325e-02   1.987  0.04959 *
## bed_typeFuton          3.399e-02  4.040e-02   0.841  0.40223
## bed_typePull-out Sofa  1.055e-01  6.267e-02   1.683  0.09539 .
## bed_typeReal Bed       7.221e-02  3.339e-02   2.162  0.03296 *
## beds                  -1.436e-02  1.017e-02  -1.411  0.16117
## accommodates           1.275e-02  6.096e-03   2.091  0.03902 *
```

```
## room_typePrivate room  3.185e-02  1.154e-02   2.760  0.00687 **
## room_typeShared room   2.941e-03  3.181e-02   0.092  0.92652
## property_typeDorm     -3.905e-02  4.838e-02  -0.807  0.42154
## potential_income      -6.310e-08  3.459e-07  -0.182  0.85563
## low_income             2.391e-06  1.177e-05   0.203  0.83943
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04557 on 101 degrees of freedom
## Multiple R-squared:  0.1899, Adjusted R-squared:  0.06964
## F-statistic: 1.579 on 15 and 101 DF,  p-value: 0.09293
```

To examine what things might actually be related to the distance from the subway a multiple regression was run on Distance including the same variables from before. We see that on average the closer an Airbnb listing is to the subway stop the more reviews they received, net of all other factors. The p-value is below 0.05 so this is statistically significant. We see that some types of rooms and types of beds also appear to be significant in comparison to other types.



Here one can visualize the relationship between Distance from subway stops and the number of reviews the airbnb received. This may be because the Airbnb listings closer to subway stops are actually being rented out more often due to convenience.

Project Book

1

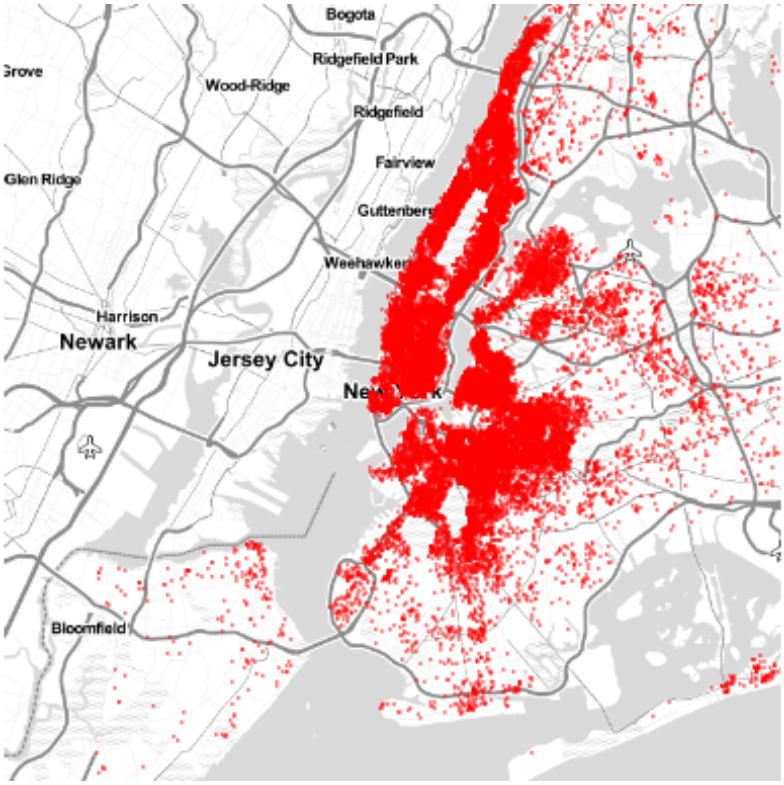
First I loaded the Airbnb dataset and began using get map and ggmap to plot the first section of the assignment. I decided to use get map and ggmap for the first section of the assignment because I wanted to show that I am able and capable to use getmap/ggmap since I will not be using it at all to answer sections 2 and 3. I used stamen and google as a source but decided to only include the stamen/toner-lite maptype in the polished graphs simply because they are less busy/distracting. (Almost an all white background so the dots/heatmaps show up well.) The google hybrid maps look really cool but are too busy when attempting to get a point across. I played around with the colors a lot for the heat maps and for the google maps I really like going from yellow to blue, and for the polished stamen/ toner-lite heat map I used red to blue because I felt it is easiest to see all of the colors even when the color is dim. When using other colors they would be almost invisible in the dimmer (less populated) areas. For the outputs used as polished graphs I choose to enlarge the maps in order to make them more readable.

```
library(readr)
airbnb <- read_csv("C:\\Users\\Brandon\\Documents\\GitHub\\QMSS-G4063-Data-Visualization\\Assignments\\Assignment 2\\airbnb_listings.csv.zip")

## Parsed with column specification:
## cols(
##   .default = col_character(),
##   id = col_integer(),
##   scrape_id = col_double(),
##   last_scraped = col_date(format = ""),
##   host_id = col_integer(),
##   host_since = col_date(format = ""),
##   host_listings_count = col_integer(),
##   host_total_listings_count = col_integer(),
##   zipcode = col_integer(),
##   latitude = col_double(),
##   longitude = col_double(),
##   accommodates = col_integer(),
##   bathrooms = col_double(),
##   bedrooms = col_integer(),
##   beds = col_integer(),
##   square_feet = col_integer(),
##   guests_included = col_integer(),
##   minimum_nights = col_integer(),
##   maximum_nights = col_integer(),
##   availability_30 = col_integer(),
##   availability_60 = col_integer()
```

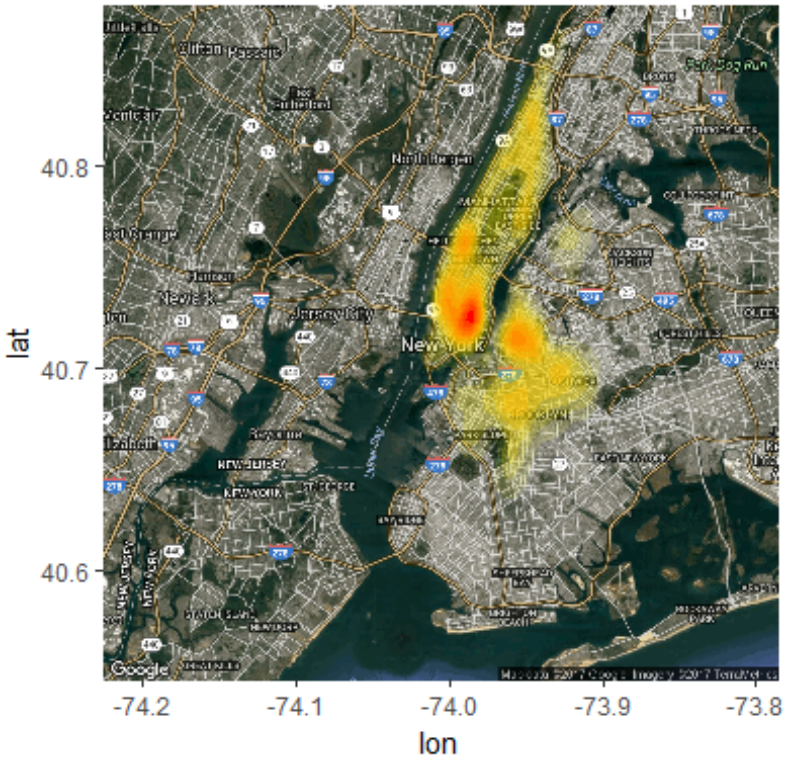
```
##   # ... with 15 more columns
## )

## See spec(...) for full column specifications.

## Warning: 9 parsing failures.
##   row     col               expected  actual
##  1244 zipcode no trailing characters -1175
## 13232 zipcode no trailing characters -2289
## 13668 zipcode no trailing characters
## 11249
## 21790 zipcode no trailing characters  10019
## 23944 zipcode no trailing characters m
## ..... ....... ..................... .......
## See problems(...) for more details.

#library(ggplot2)
#library(ggmap)

map_NYC_st <- get_map("New York", zoom=11,
                  source="stamen",maptype="toner-lite")

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=
New+York&zoom=11&size=640x640&scale=2&maptype=terrain&sensor=false

## Information from URL : http://maps.googleapis.com/maps/api/geocode/j
son?address=New%20York&sensor=false

## Map from URL : http://tile.stamen.com/toner-lite/11/601/768.png

## Warning in file.remove(index[[url]]): cannot remove file
## '13d69437ade7694d3accb298bdb57b63.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/602/768.png

## Warning in file.remove(index[[url]]): cannot remove file
## 'a080478c811d9c99061b3c468e446293.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/603/768.png

## Warning in file.remove(index[[url]]): cannot remove file
## '4af7727f763ca6e868400237a23cf260.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/604/768.png

## Warning in file.remove(index[[url]]): cannot remove file
## 'a625baa8137f526f42da7cd75fcac045.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/601/769.png
```

```
## Warning in file.remove(index[[url]]): cannot remove file
## '46cd0695ad1e1490594618dd58013122.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/602/769.png

## Warning in file.remove(index[[url]]): cannot remove file
## 'b5e1b7dee9389822849326ebc3a15674.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/603/769.png

## Warning in file.remove(index[[url]]): cannot remove file
## '0fdca83ad32b5e7862a80a777ed3f4fc.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/604/769.png

## Warning in file.remove(index[[url]]): cannot remove file
## 'cd4dc96dc14fd520137ab08b978dc4e9.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/601/770.png

## Warning in file.remove(index[[url]]): cannot remove file
## '2a16693cd5de02b00f301576e1b8150f.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/602/770.png

## Warning in file.remove(index[[url]]): cannot remove file
## '6aaa58bea88938ab510bd09a6d879163.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/603/770.png

## Warning in file.remove(index[[url]]): cannot remove file
## '3872ec9326eaa0ef3f8389f37ca08af0.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/604/770.png

## Warning in file.remove(index[[url]]): cannot remove file
## 'f027bc1b5feb562e7756e624752ffbec.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/601/771.png

## Warning in file.remove(index[[url]]): cannot remove file
## 'df509703e5f1908c94c67bfb4d3894e3.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/602/771.png
```

```
## Warning in file.remove(index[[url]]): cannot remove file
## '42bed0068030fe5a2dc14e4d900d4640.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/603/771.png

## Warning in file.remove(index[[url]]): cannot remove file
## '685b353b448a399b6ea38247adb7c529.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/11/604/771.png

## Warning in file.remove(index[[url]]): cannot remove file
## '46751cd5b1bb697422e001cdf0db6487.rds', reason 'No such file or dire
ctory'

a <- ggmap(map_NYC_st, extent = "device")

## Warning: `panel.margin` is deprecated. Please use `panel.spacing` pr
operty
## instead

a <- a + geom_point(aes(x=longitude,y=latitude),data=airbnb, size=0.3,
alpha=0.3, color="red")
a

## Warning: Removed 441 rows containing missing values (geom_point).
```

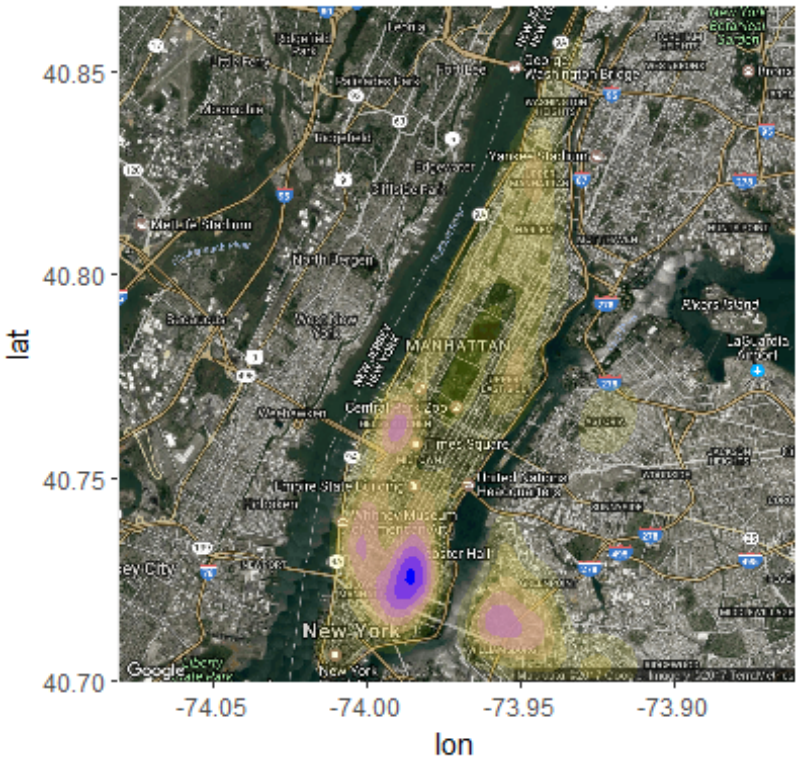


```
map_NYC_gm <- get_map("New York", zoom=11,
                     source="google",maptype="hybrid")

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=
New+York&zoom=11&size=640x640&scale=2&maptype=hybrid&language=en-EN&sen
sor=false

## Information from URL : http://maps.googleapis.com/maps/api/geocode/j
son?address=New%20York&sensor=false

g2 <- ggmap(map_NYC_gm)
g2 <- g2 + stat_density2d(aes(x=longitude,y=latitude,
                              fill=..level..,alpha=..level..),
                          data=airbnb,geom="polygon")
(g2 + scale_fill_gradient(low = "yellow", high = "red") + theme(legend.
position = "none"))

## Warning: Removed 441 rows containing non-finite values (stat_density
2d).
```




```
map_MAN_gm <- get_map("Manhattan", zoom=12,
                     source="google",maptype="hybrid")

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=
Manhattan&zoom=12&size=640x640&scale=2&maptype=hybrid&language=en-EN&se
nsor=false
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/j
son?address=Manhattan&sensor=false

g3 <- ggmap(map_MAN_gm)
g3 <- g3 + stat_density2d(aes(x=longitude,y=latitude,
                              fill=..level..,alpha=..level..),
                          data=airbnb,geom="polygon")
(g3 + scale_fill_gradient(low = "yellow", high = "blue") + theme(legend
.position = "none"))

## Warning: Removed 12784 rows containing non-finite values (stat_densi
ty2d).
```




```
map_MAN_st <- get_map("Manhattan", zoom=12,
                      source="stamen",maptype="toner-lite")

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=
Manhattan&zoom=12&size=640x640&scale=2&maptype=terrain&sensor=false

## Information from URL : http://maps.googleapis.com/maps/api/geocode/j
son?address=Manhattan&sensor=false

## Map from URL : http://tile.stamen.com/toner-lite/12/1205/1537.png

## Warning in file.remove(index[[url]]): cannot remove file
## '9ac0191557edf4f218f75411d1babefd.rds', reason 'No such file or dire
ctory'
```

```
## Map from URL : http://tile.stamen.com/toner-lite/12/1206/1537.png

## Warning in file.remove(index[[url]]): cannot remove file
## '2773f1dc8a0ce9fb552e7feb7cf840cc.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/12/1207/1537.png

## Warning in file.remove(index[[url]]): cannot remove file
## 'fe951c5d8249fae8df8c6221defd01a8.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/12/1205/1538.png

## Warning in file.remove(index[[url]]): cannot remove file
## 'be3eb489a29c8c101dad0fe1d0b0304c.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/12/1206/1538.png

## Warning in file.remove(index[[url]]): cannot remove file
## '43a5d080468b6c80290a7fa46126e912.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/12/1207/1538.png

## Warning in file.remove(index[[url]]): cannot remove file
## '0fbc97b98b41afa39d9b153c8fb944fd.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/12/1205/1539.png

## Warning in file.remove(index[[url]]): cannot remove file
## '67caeae66a62be03e57685cf88077d36.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/12/1206/1539.png

## Warning in file.remove(index[[url]]): cannot remove file
## 'd03761c428186fd94f6a0865bee8e639.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/12/1207/1539.png

## Warning in file.remove(index[[url]]): cannot remove file
## 'a97ff75d693aab0ef6e7af3fbaf76d31.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/12/1205/1540.png

## Warning in file.remove(index[[url]]): cannot remove file
## '871680c4f3554063e05e9c3dddb171d4.rds', reason 'No such file or dire
ctory'
```

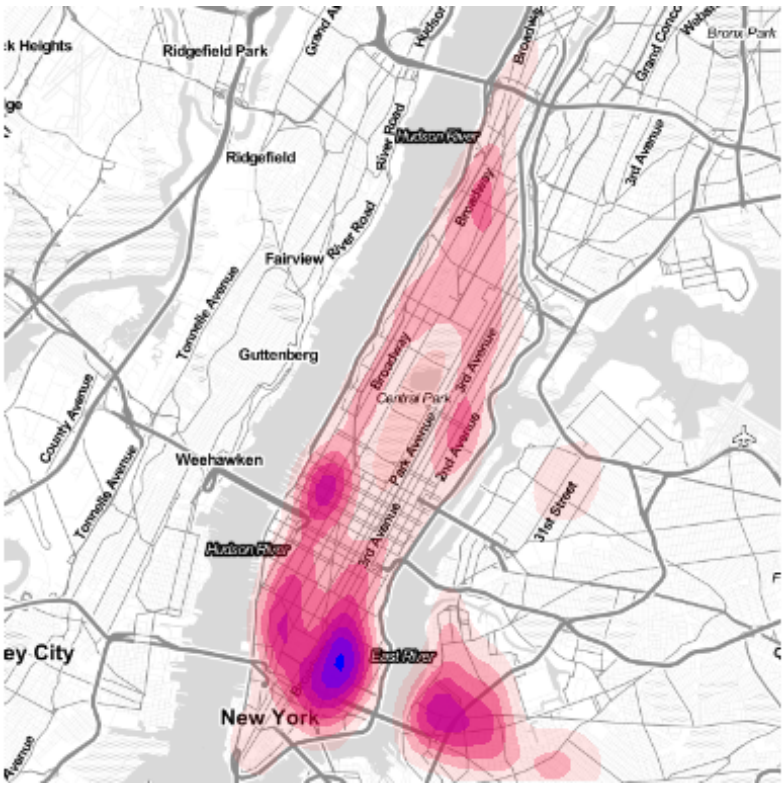
```
## Map from URL : http://tile.stamen.com/toner-lite/12/1206/1540.png

## Warning in file.remove(index[[url]]): cannot remove file
## '985de2c0e2c68fe9a708168b6a53d4df.rds', reason 'No such file or dire
ctory'

## Map from URL : http://tile.stamen.com/toner-lite/12/1207/1540.png

## Warning in file.remove(index[[url]]): cannot remove file
## '4aa4a6edc72db0b5e449e86029470c4e.rds', reason 'No such file or dire
ctory'
```

```
g5 <- ggmap(map_MAN_st, extent = "device")
```

```
## Warning: `panel.margin` is deprecated. Please use `panel.spacing` pr
operty
## instead
```

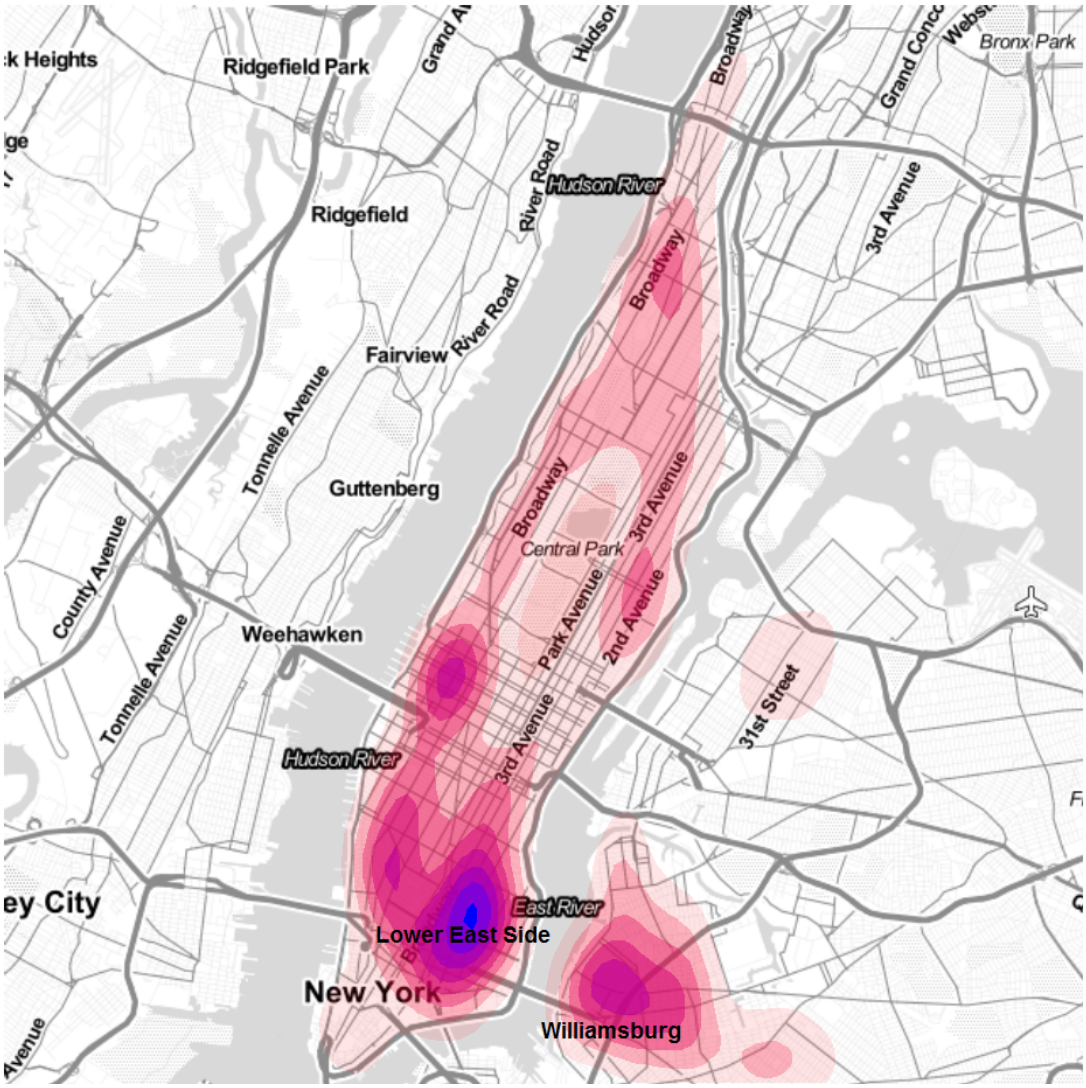
```
g5 <- g5 + stat_density2d(aes(x=longitude,y=latitude,
                              fill=..level..,alpha=..level..),
                          data=airbnb,geom="polygon")
(g5 + scale_fill_gradient(low = "red", high = "blue") +
                          theme(legend.position = "none"))
```

```
## Warning: Removed 12784 rows containing non-finite values (stat_densi
ty2d).
```



```
map_MAN_st <- get_map("Manhattan", zoom=12,
                  source="stamen",maptype="toner-lite")

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=
Manhattan&zoom=12&size=640x640&scale=2&maptype=terrain&sensor=false

## Information from URL : http://maps.googleapis.com/maps/api/geocode/j
son?address=Manhattan&sensor=false

## Map from URL : http://tile.stamen.com/toner-lite/12/1205/1537.png

## Map from URL : http://tile.stamen.com/toner-lite/12/1206/1537.png

## Map from URL : http://tile.stamen.com/toner-lite/12/1207/1537.png

## Map from URL : http://tile.stamen.com/toner-lite/12/1205/1538.png

## Map from URL : http://tile.stamen.com/toner-lite/12/1206/1538.png

## Map from URL : http://tile.stamen.com/toner-lite/12/1207/1538.png

## Map from URL : http://tile.stamen.com/toner-lite/12/1205/1539.png

## Map from URL : http://tile.stamen.com/toner-lite/12/1206/1539.png

## Map from URL : http://tile.stamen.com/toner-lite/12/1207/1539.png

## Map from URL : http://tile.stamen.com/toner-lite/12/1205/1540.png

## Map from URL : http://tile.stamen.com/toner-lite/12/1206/1540.png

## Map from URL : http://tile.stamen.com/toner-lite/12/1207/1540.png

g4 <- ggmap(map_MAN_st, extent = "device")
g4 <- g4 + stat_density2d(aes(x=longitude,y=latitude,
                              fill=..level..,alpha=..level..),
                          data=airbnb,geom="polygon") +
        annotate("text",x=-73.987289, y=40.722933, label="Lower East Si
de",
             fontface=2, size=5) +
        annotate("text",x=-73.95707, y=40.708116, label="Williamsburg",
             fontface=2, size=5) +
        ggtitle("Airbnb Listingins Density Map NYC") +
        theme(plot.title = element_text(hjust = 0.5)) +
        theme(text = element_text(size=20))
(g4 + scale_fill_gradient(low = "red", high = "blue") +
                          theme(legend.position = "none"))
```

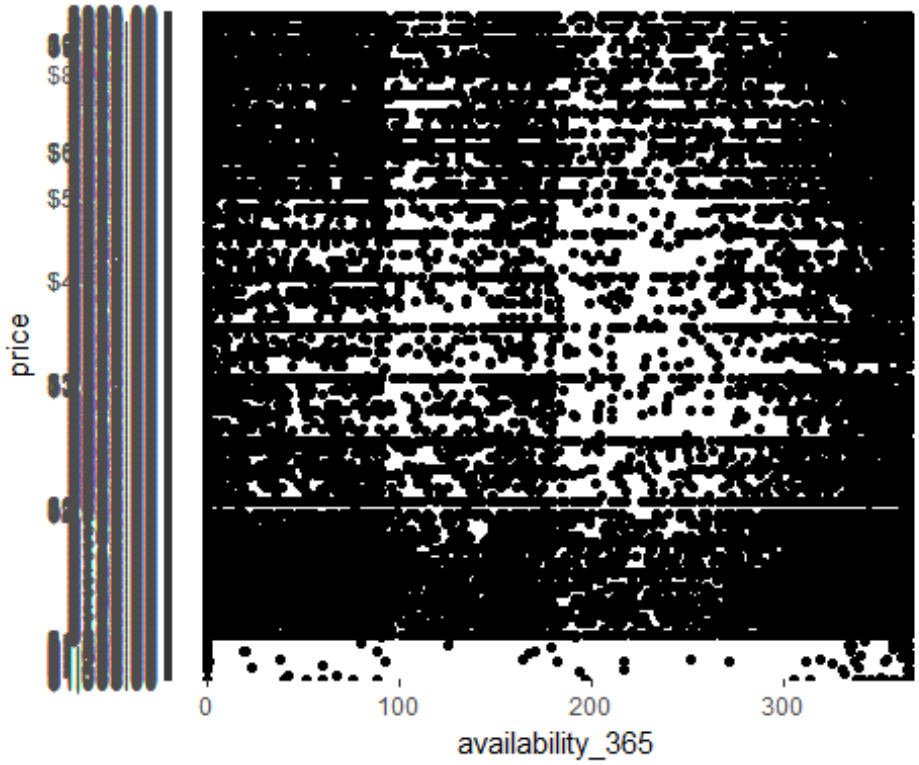
Airbnb Listingins Density Map NYC



2

First I plotted the airbnb data by availability and price just to give myself an idea of how these two variables interact.

```
gg <- ggplot(airbnb)
gg <- gg + geom_point(mapping = aes(x = availability_365, y = price))
gg
```



Next I had to remove the $ symbol from price and made price numeric in order to actually average the price by neighbourhood. I also created a dummy variable for availability so that 1 = always available and 0 = Not always available. Next I created to separate datasets one with the dummy variable and one with the average availability.

```
#rid of $ in price
airbnb$price <- as.numeric(sub("\\$","", airbnb$price))

## Warning: NAs introduced by coercion

airbnb$price <- as.numeric(as.character(airbnb$price))
airbnb$dummy <- as.numeric(airbnb$availability_365 == 365)

# calculating averages
library(dplyr)
neighborhoodbnb <- airbnb %>%
  group_by(neighbourhood_cleansed) %>%
  summarise(avg_availability = mean(availability_365, na.rm = FALSE),
            avg_price = mean(price, na.rm = TRUE),
            total = n())
neighborhoodbnb <- as.data.frame(neighborhoodbnb)

neighborhoodbnb$neighbourhood <- neighborhoodbnb$neighbourhood_cleansed

#calculating averages with binary
neigh_365 <- airbnb %>%
```

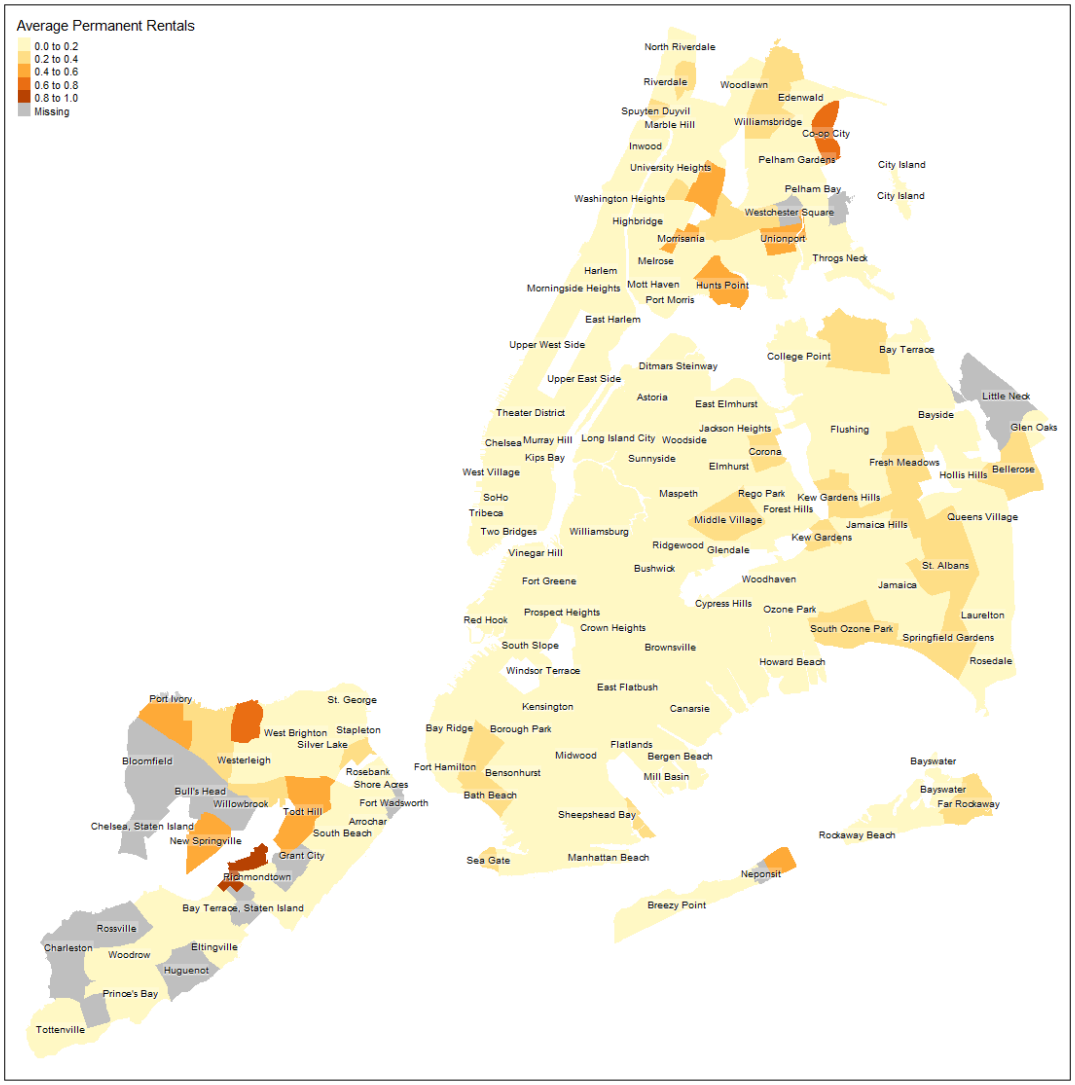
```
  group_by(neighbourhood_cleansed) %>%
  summarise(aval_365 = mean(dummy),
            avg_price = mean(price),
            total = n())
neigh_365 <- as.data.frame(neigh_365)
neigh_365$neighbourhood <- neigh_365$neighbourhood_cleansed
```

Once I made the datasets I then loaded in the geoJSON file twice, once for each dataset and also merged the datasets to the neighborhoods geoJSON file.

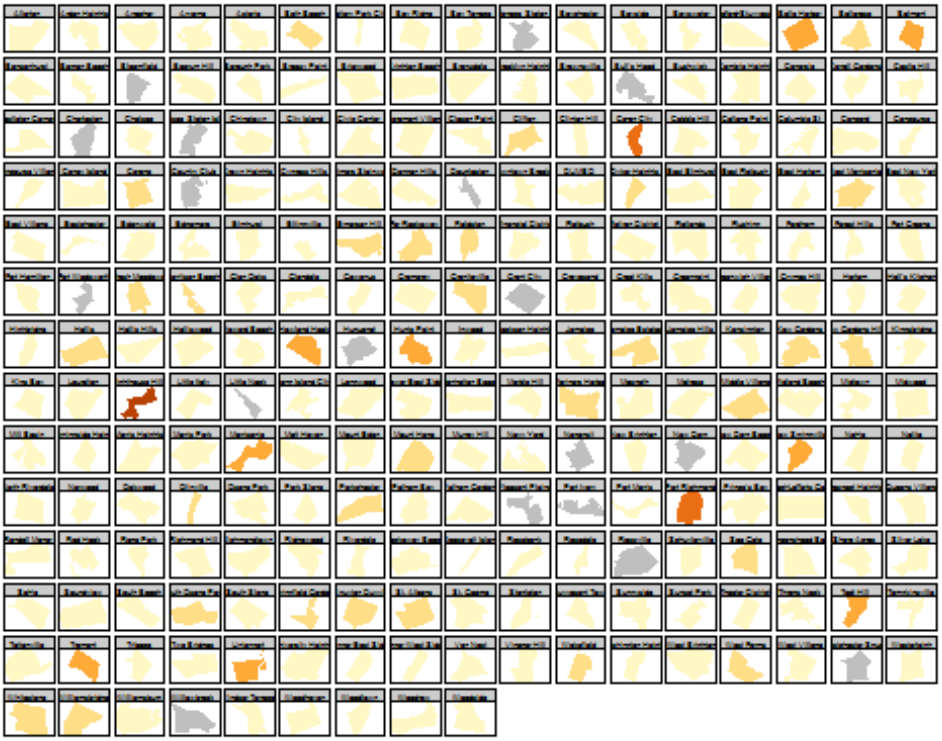
```
library(rgdal)

nyc_neighborhood <- readOGR("C:\\Users\\Brandon\\Documents\\GitHub\\QMSS-G4063-Data-Visualization\\Assignments\\Assignment 2\\neighbourhoods.geojson","OGRGeoJSON")

## OGR data source with driver: GeoJSON
## Source: "C:\Users\Brandon\Documents\GitHub\QMSS-G4063-Data-Visualization\Assignments\Assignment 2\neighbourhoods.geojson", layer: "OGRGeoJSON"
## with 233 features
## It has 2 fields

nyc_neighbor365 <- readOGR("C:\\Users\\Brandon\\Documents\\GitHub\\QMSS-G4063-Data-Visualization\\Assignments\\Assignment 2\\neighbourhoods.geojson","OGRGeoJSON")

## OGR data source with driver: GeoJSON
## Source: "C:\Users\Brandon\Documents\GitHub\QMSS-G4063-Data-Visualization\Assignments\Assignment 2\neighbourhoods.geojson", layer: "OGRGeoJSON"
## with 233 features
## It has 2 fields

# Merge data.frame into the geojson
nyc_neighborhood@data <- data.frame(nyc_neighborhood@data, neighborhoodbnb[match(nyc_neighborhood@data[,"neighbourhood"], neighborhoodbnb[,"neighbourhood"]),])

# Merge data.frame into the geojson
nyc_neighbor365@data <- data.frame(nyc_neighbor365@data, neigh_365[match(nyc_neighbor365@data[,"neighbourhood"], neigh_365[,"neighbourhood"]),])
```

Next I ran a number of graphs using tmap. I did not use the dummy variable in the polished graphs because there was not as much variability and they did not look as visually pleasing. For my own exploration I also used facets by neighborhood to see each neighborhood on their own but could not use it in the polished graphs because they were very difficult to read. I decided to include the plots of average price and average availability to show how each neighborhood differs.

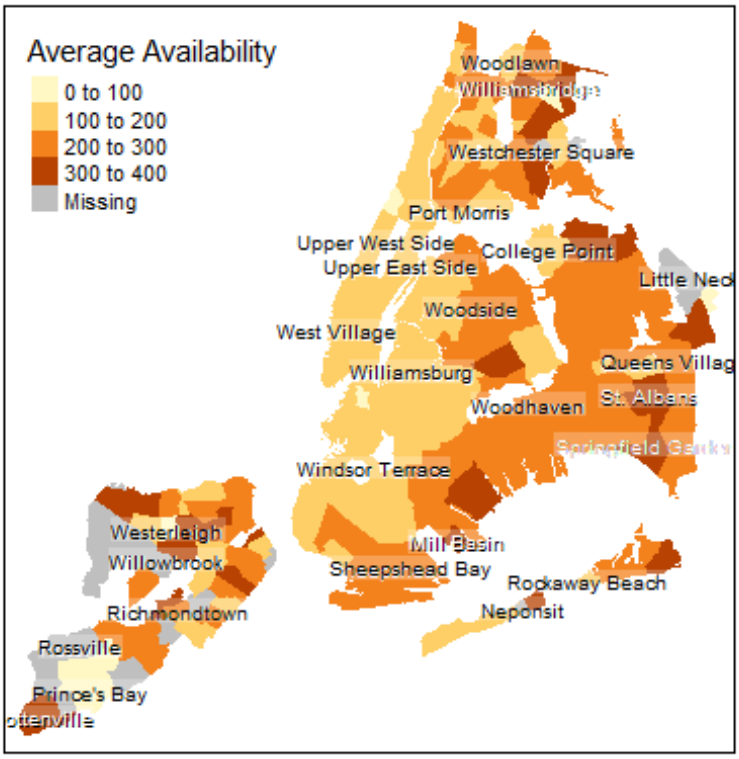
```
library(tmap)

al <- tm_shape(nyc_neighbor365) + layout +
tm_fill("aval_365", title = "Average Permanent Rentals")
al + tm_text("neighbourhood", size=.6, shadow=TRUE,
  bg.color="white", bg.alpha=.25,
  remove.overlap=TRUE)
```



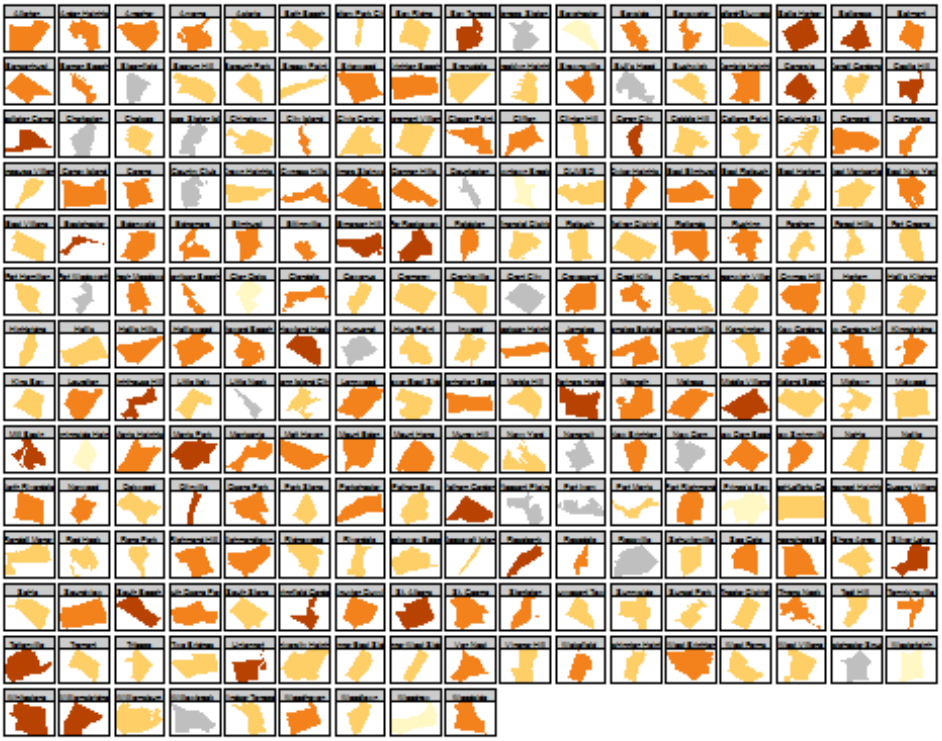

```
options(tmap.limits=c(facets.plot=230, facets.view=4))

tm_shape(nyc_neighbor365) +
    tm_fill("aval_365", legend.show = FALSE) +
    tm_facets("neighbourhood", free.coords=TRUE, drop.units=TRUE, drop.
empty.facets = TRUE)
```



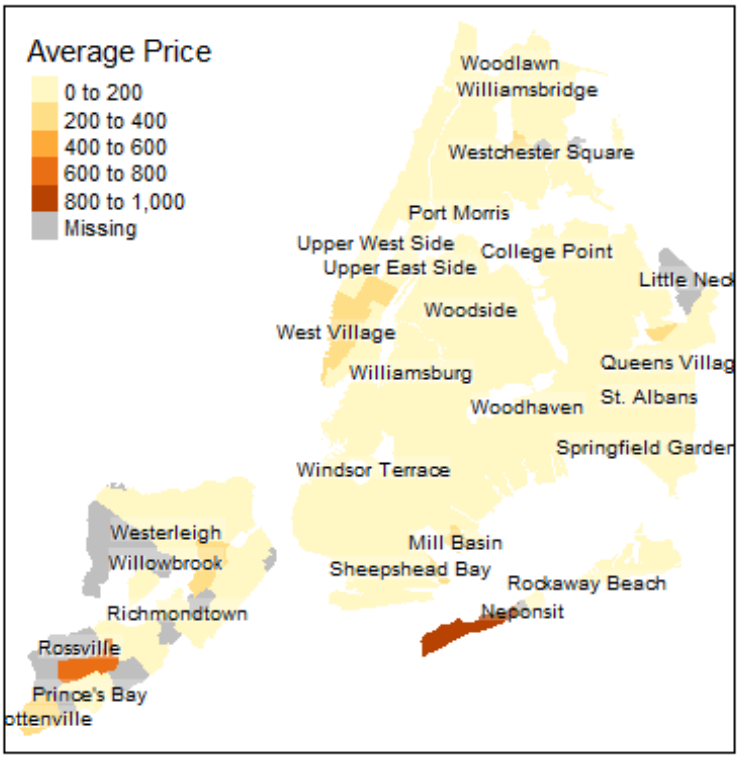
```
av <- tm_shape(nyc_neighborhood) + layout +
tm_fill("avg_availability", title = "Average Availability")
av + tm_text("neighbourhood", size=.6, shadow=TRUE,
  bg.color="white", bg.alpha=.25,
  remove.overlap=TRUE)
```



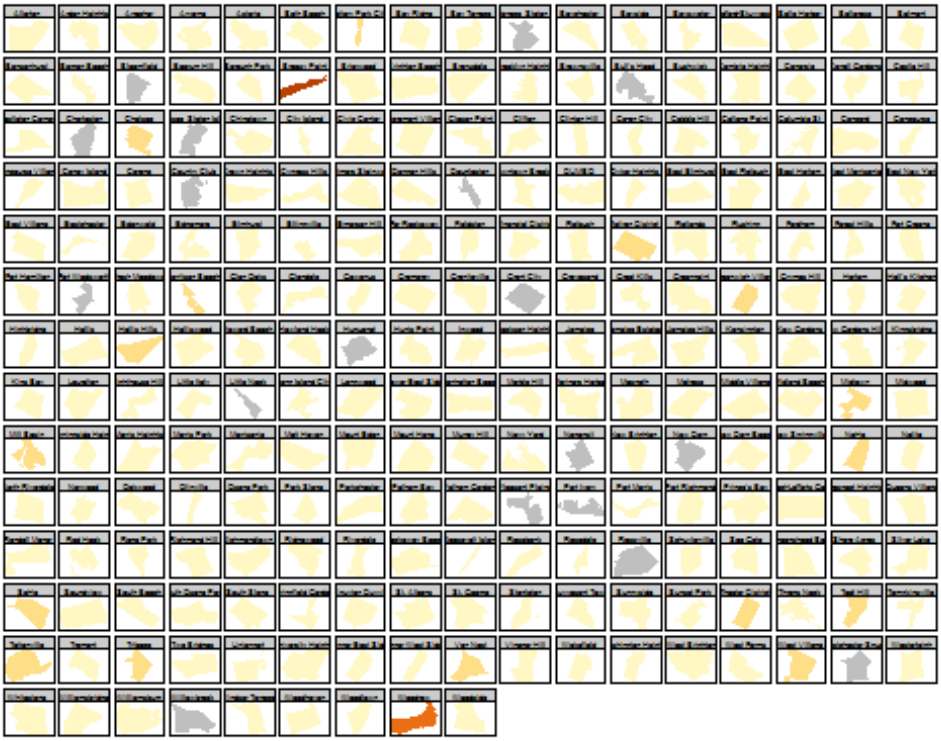

```
tm_shape(nyc_neighborhood) +
    tm_fill("avg_availability", legend.show = FALSE) +
    tm_facets("neighbourhood", free.coords=TRUE, drop.units=TRUE, drop.
empty.facets = TRUE)
```



```
#Average Price
ap <- tm_shape(nyc_neighborhood) + layout +
tm_fill("avg_price", title = "Average Price")
ap + tm_text("neighbourhood", size=.6, shadow=TRUE,
  bg.color="white", bg.alpha=.25,
  remove.overlap=TRUE)
```

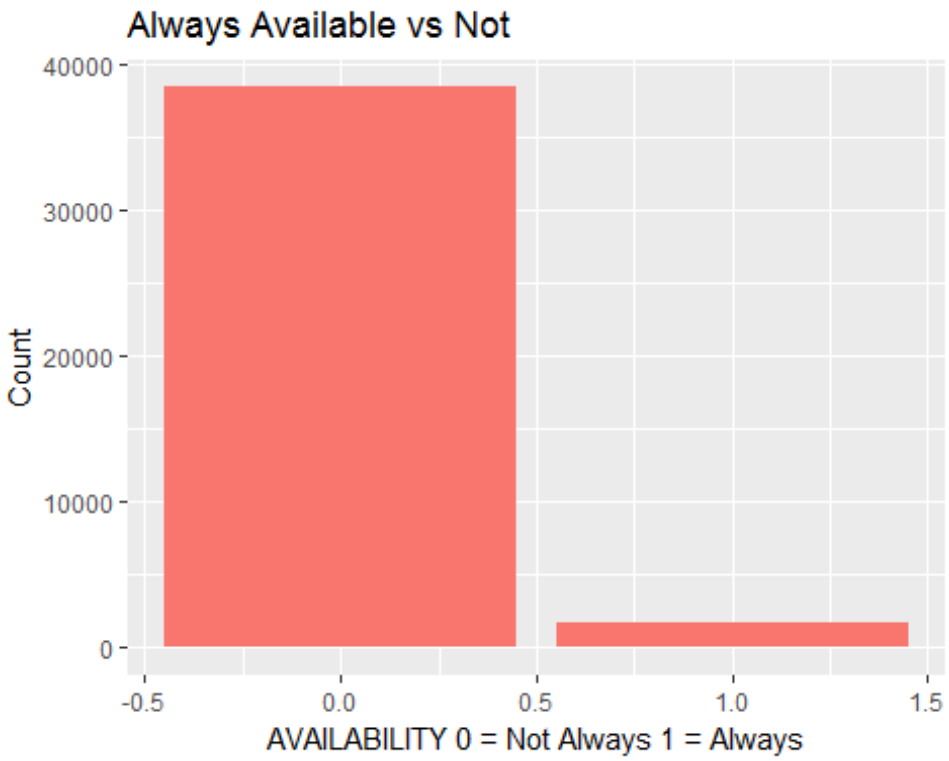



```
tm_shape(nyc_neighborhood) +
    tm_fill("avg_price", legend.show = FALSE) +
    tm_facets("neighbourhood", free.coords=TRUE, drop.units=TRUE, drop.
empty.facets = TRUE)
```

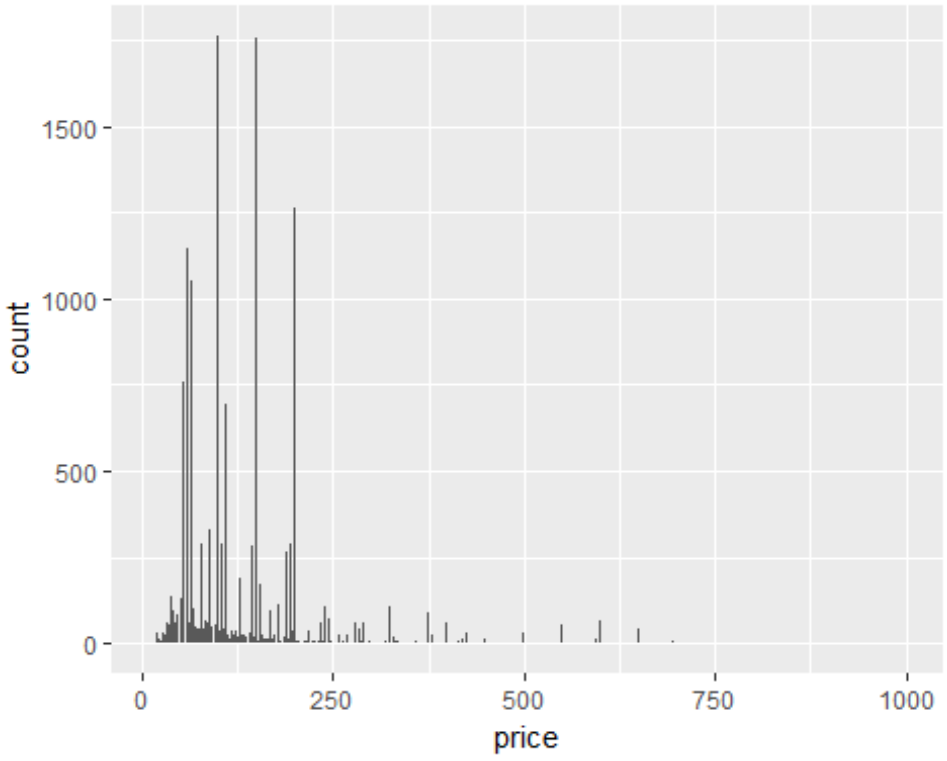


Next I explored the data by using ggplot2 and decided to separate by borough instead of neighborhood for further analysis. I did this because it is easier to compare this smaller number of groups. This was in order to really examine the relationship between availability, price, and number of reviews. I added number of reviews because I feel this is a decent way to see how many nights people actually do stay in the Airbnb listings. None of this was used in the polished graphs because I didn't want to add way to many graphs and overwhelm the reader.

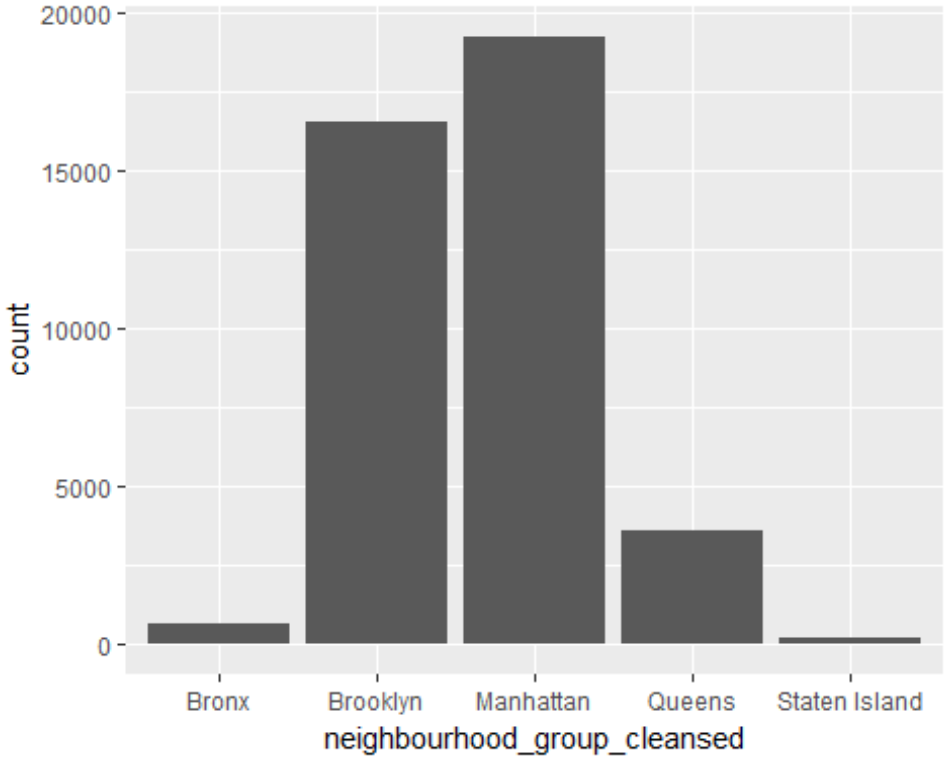
```
##explore price and availability
c <- ggplot(airbnb, aes(dummy))
c + geom_bar(aes(fill = "blue")) +
  ggtitle("Always Available vs Not") +
  xlab("AVAILABILITY 0 = Not Always 1 = Always") +
  ylab("Count") +
  guides(fill=FALSE)
```




```
d <- ggplot(airbnb, aes(price))
d + geom_bar()

## Warning: Removed 205 rows containing non-finite values (stat_count).
```



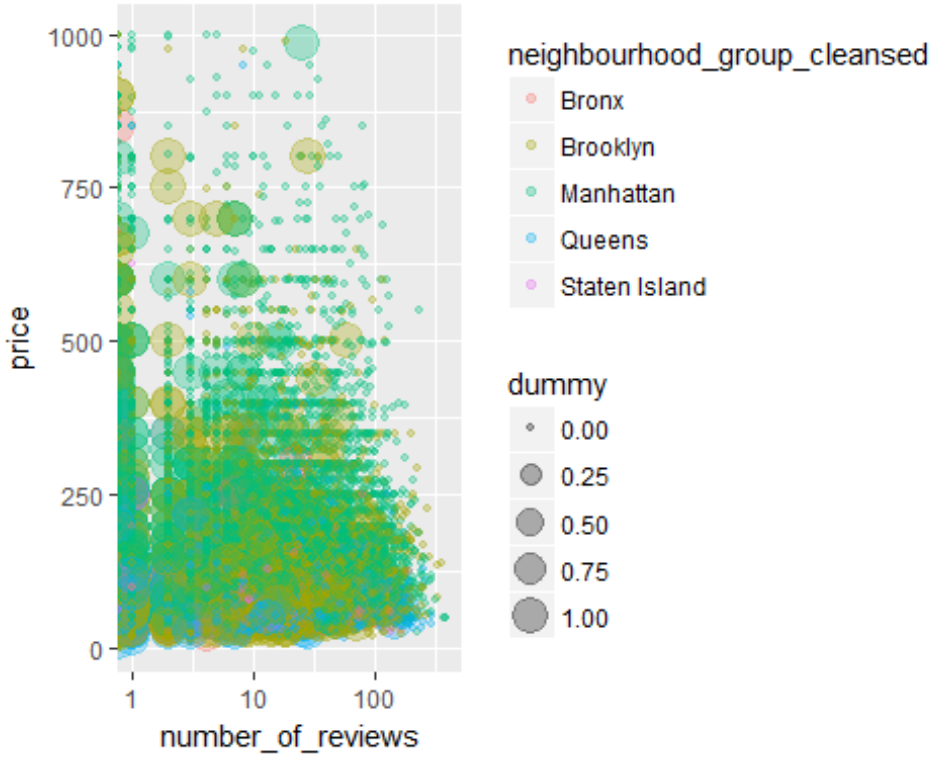

```
k <- ggplot(airbnb, aes(neighbourhood_group_cleansed))
k + geom_bar()
```



```
ggplot(data = airbnb,
       aes(x = number_of_reviews, y = price, size=dummy,
           color=neighbourhood_group_cleansed)) +
       scale_x_log10() + geom_point(alpha=0.3)
```

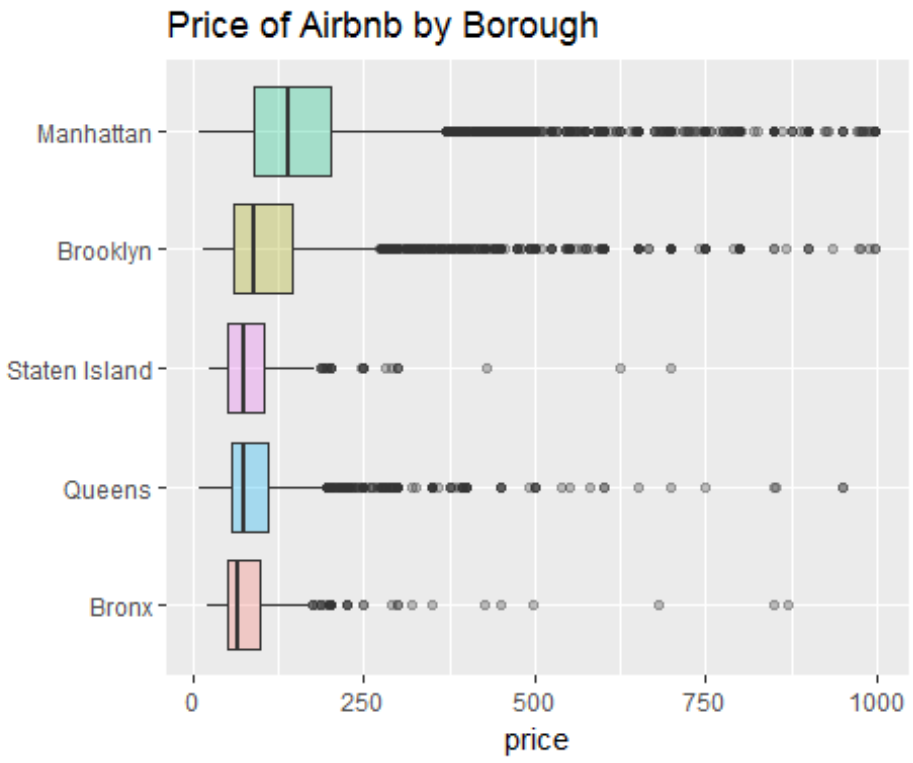
```
## Warning: Transformation introduced infinite values in continuous x-a
xis
```

```
## Warning: Removed 205 rows containing missing values (geom_point).
```

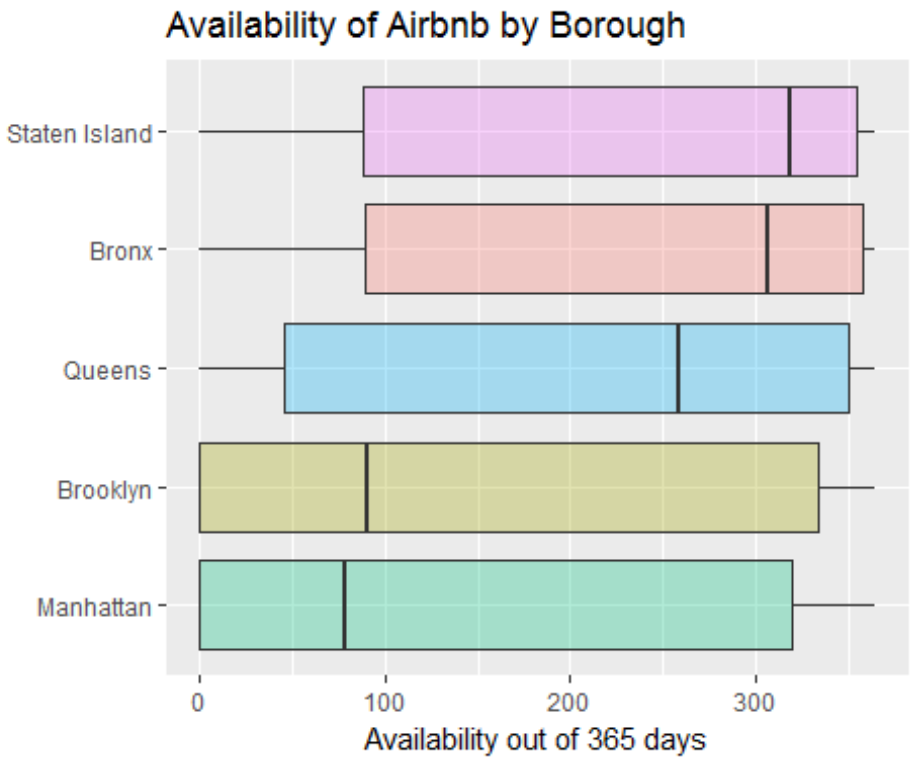



```
f <- ggplot(data=airbnb,aes(x=reorder(neighbourhood_group_cleansed, pri
ce, na.rm=TRUE), y=price))
f + geom_boxplot(aes(fill=neighbourhood_group_cleansed), alpha=0.3) +
    coord_flip() +
    labs(x="", y="price") +
    guides(fill=FALSE) +
    ggtitle("Price of Airbnb by Borough")
```

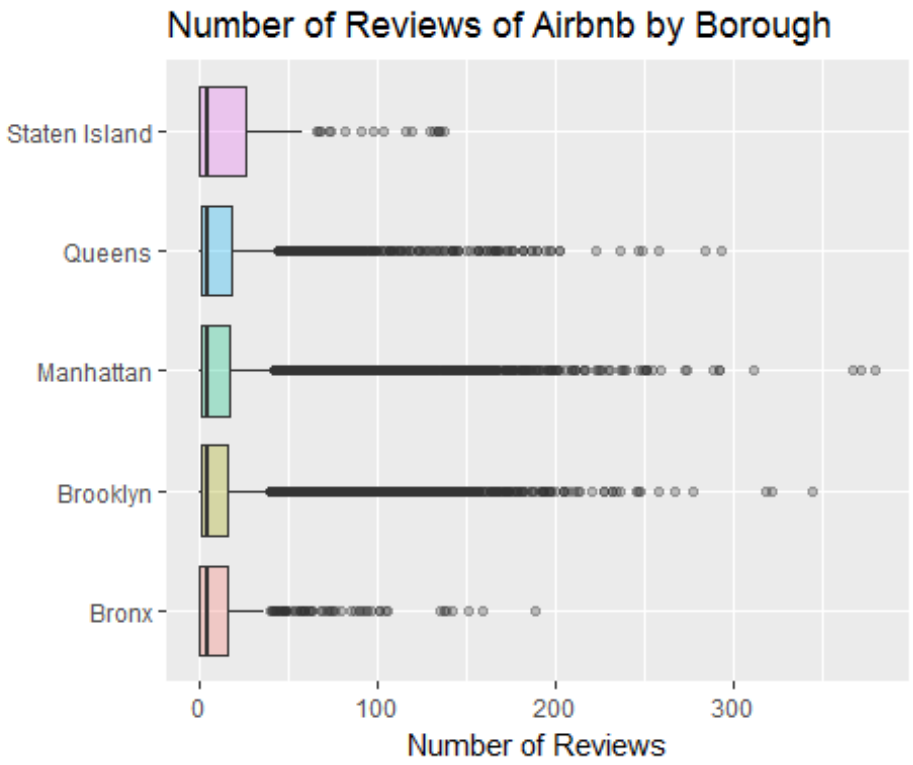
```
## Warning: Removed 205 rows containing non-finite values (stat_boxplot
).
```




```
h <- ggplot(data=airbnb,aes(x=reorder(neighbourhood_group_cleansed, ava
ilability_365, na.rm=TRUE), y=availability_365))
h + geom_boxplot(aes(fill=neighbourhood_group_cleansed), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Availability out of 365 days") +
    guides(fill=FALSE) +
    ggtitle("Availability of Airbnb by Borough")
```




```
j <- ggplot(data=airbnb,aes(x=reorder(neighbourhood_group_cleansed, num
ber_of_reviews, na.rm=TRUE), y=number_of_reviews))
j + geom_boxplot(aes(fill=neighbourhood_group_cleansed), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Number of Reviews") +
    guides(fill=FALSE) +
    ggtitle("Number of Reviews of Airbnb by Borough")
```



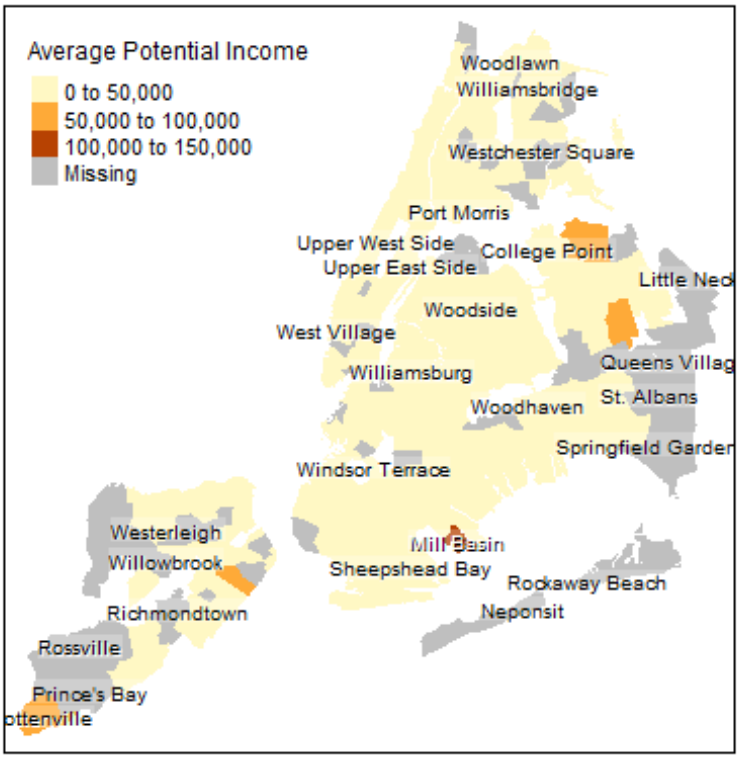
Next I calculated two different measures of Income one based on availability and one based on number of reviews. I did this because either way we are drawing a number of assumptions but if we look at both we have a better idea of where in the middle the real income may lie. Availability is assuming that the Airbnb is booked every possible day, so it is giving use a way higher income estimate then the truth. On the other hand using the number of reviews is giving us the income based on the "known" times the listing was booked, so it is giving us a lower income estimate then the truth. I then plotted both and did use both in the polished graphs because I felt they are very informational and allow the reader to think about where the true income estimate might lie. I also printed the summary statistics of the two measures of income to see the actual range and highlight the max (Mil Basin). Again I choose to enlarge all the maps in the polished section to really allow the reader to see the neighborhoods.

```
#create new variable for income
airbnb$potential_income <- (airbnb$availability_365 * airbnb$price)
airbnb$low_income <- (airbnb$number_of_reviews * airbnb$price)

#predict income
lm = lm(potential_income ~ dummy + neighbourhood_group_cleansed + price
+ bathrooms + bedrooms, airbnb)
summary(lm)

##
## Call:
## lm(formula = potential_income ~ dummy + neighbourhood_group_cleansed
```

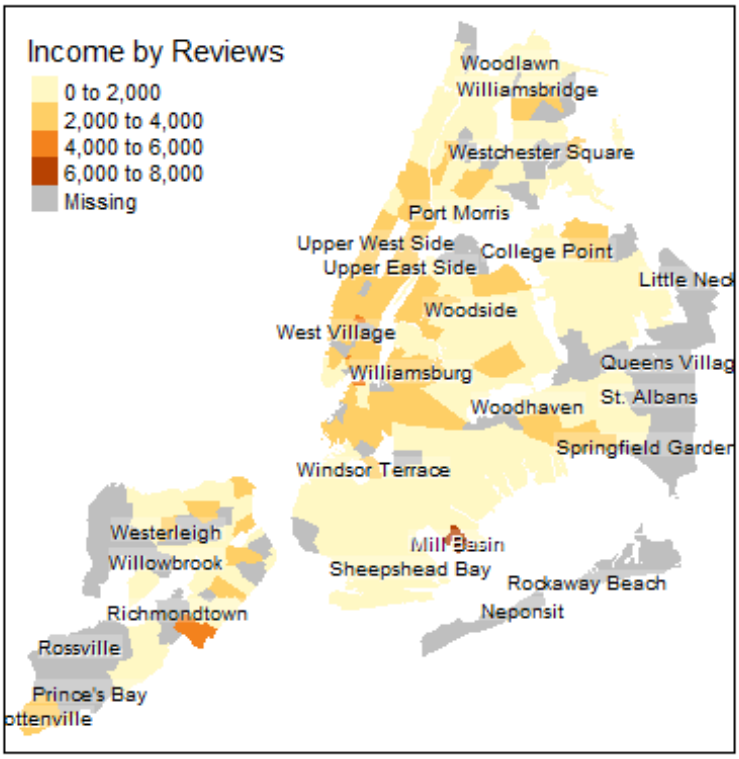
```
+
##     price + bathrooms + bedrooms, data = airbnb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -175252  -13072   -3388   14529  190568
##
## Coefficients:
##                                          Estimate Std. Error t val
ue
## (Intercept)                             -2619.124   1072.663  -2.4
42
## dummy                                   28736.938    626.267  45.8
86
## neighbourhood_group_cleansedBrooklyn      -4042.679   1017.617  -3.9
73
## neighbourhood_group_cleansedManhattan     -5683.348   1023.053  -5.5
55
## neighbourhood_group_cleansedQueens         -251.174   1081.002  -0.2
32
## neighbourhood_group_cleansedStaten Island  1031.690   1968.829   0.5
24
## price                                       176.076      1.417 124.2
89
## bathrooms                                  -533.172    373.807  -1.4
26
## bedrooms                                    3540.091    225.435  15.7
03
##                                          Pr(>|t|)
## (Intercept)                                0.0146 *
## dummy                                     < 2e-16 ***
## neighbourhood_group_cleansedBrooklyn      7.12e-05 ***
## neighbourhood_group_cleansedManhattan     2.79e-08 ***
## neighbourhood_group_cleansedQueens          0.8163
## neighbourhood_group_cleansedStaten Island   0.6003
## price                                     < 2e-16 ***
## bathrooms                                   0.1538
## bedrooms                                  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24950 on 39776 degrees of freedom
##   (442 observations deleted due to missingness)
## Multiple R-squared:  0.3961, Adjusted R-squared:  0.396
## F-statistic:  3261 on 8 and 39776 DF,  p-value: < 2.2e-16

#assuming they have guests every day possible

lm2 = lm(low_income ~ dummy + neighbourhood_group_cleansed + price + ba
```

```
throoms + bedrooms, airbnb)
summary(lm2)

##
## Call:
## lm(formula = low_income ~ dummy + neighbourhood_group_cleansed +
##     price + bathrooms + bedrooms, data = airbnb)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14022  -1671   -970    201 118347
##
## Coefficients:
##                                            Estimate Std. Error t va
lue
## (Intercept)                                -180.7052   186.1490  -0.
971
## dummy                                     -1109.2969   108.6818 -10.
207
## neighbourhood_group_cleansedBrooklyn        368.6826   176.5965   2.
088
## neighbourhood_group_cleansedManhattan       680.4393   177.5397   3.
833
## neighbourhood_group_cleansedQueens          271.4585   187.5961   1.
447
## neighbourhood_group_cleansedStaten Island   173.4697   341.6689   0.
508
## price                                        10.7401     0.2458  43.
686
## bathrooms                                  -312.6534    64.8701  -4.
820
## bedrooms                                    607.3181    39.1218  15.
524
##                                           Pr(>|t|)
## (Intercept)                               0.331676
## dummy                                      < 2e-16 ***
## neighbourhood_group_cleansedBrooklyn      0.036830 *
## neighbourhood_group_cleansedManhattan     0.000127 ***
## neighbourhood_group_cleansedQueens        0.147895
## neighbourhood_group_cleansedStaten Island 0.611658
## price                                      < 2e-16 ***
## bathrooms                                 1.44e-06 ***
## bedrooms                                   < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4329 on 39776 degrees of freedom
##   (442 observations deleted due to missingness)
## Multiple R-squared:  0.0936, Adjusted R-squared:  0.09341
## F-statistic: 513.4 on 8 and 39776 DF,  p-value: < 2.2e-16
```

```r
#assuming only reviews are profitable

#average income
neigh_inc <- airbnb %>%
  group_by(neighbourhood) %>%
  summarise(aval_inc = mean(potential_income, na.rm=TRUE),
            low_inc = mean(low_income, na.rm=TRUE),
            avg_price = mean(price, na.rm=TRUE),
            total = n())
neigh_inc <- as.data.frame(neigh_inc)

nyc_neighborinc <- readOGR("C:\\Users\\Brandon\\Documents\\GitHub\\QMSS-G4063-Data-Visualization\\Assignments\\Assignment 2\\neighbourhoods.geojson","OGRGeoJSON")

## OGR data source with driver: GeoJSON
## Source: "C:\Users\Brandon\Documents\GitHub\QMSS-G4063-Data-Visualization\Assignments\Assignment 2\neighbourhoods.geojson", layer: "OGRGeoJSON"
## with 233 features
## It has 2 fields

# Merge data.frame into the geojson
nyc_neighborinc@data <- data.frame(nyc_neighborinc@data, neigh_inc[match(nyc_neighborinc @data[,"neighbourhood"], neigh_inc[,"neighbourhood"]),])


ap <- tm_shape(nyc_neighborinc) + layout +
tm_fill("aval_inc", title = "Average Potential Income")
ap + tm_text("neighbourhood", size=.6, shadow=TRUE,
  bg.color="white", bg.alpha=.25,
  remove.overlap=TRUE)
```




```
lp <- tm_shape(nyc_neighborinc) + layout +
tm_fill("low_inc", title = "Income by Reviews")
lp + tm_text("neighbourhood", size=.6, shadow=TRUE,
  bg.color="white", bg.alpha=.25,
  remove.overlap=TRUE)
```




```
summary(nyc_neighborinc@data$aval_inc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0   15440   19800   22420   25280  149500      73
```

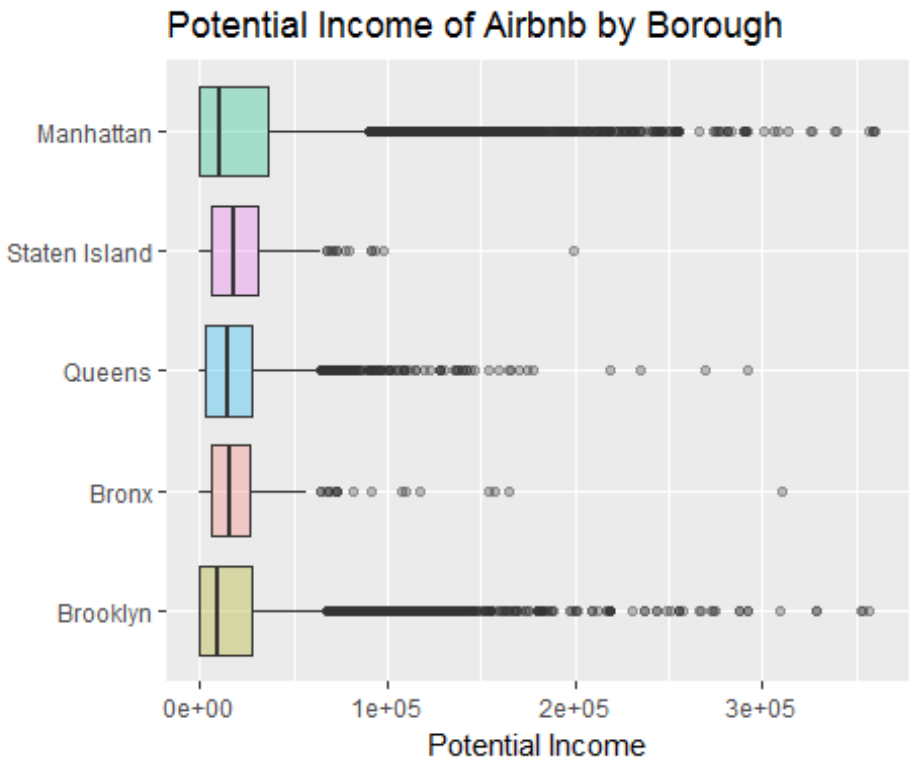
```
summary(nyc_neighborinc@data$low_inc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0   802.2  1247.0  1572.0  2142.0  8000.0      73
```

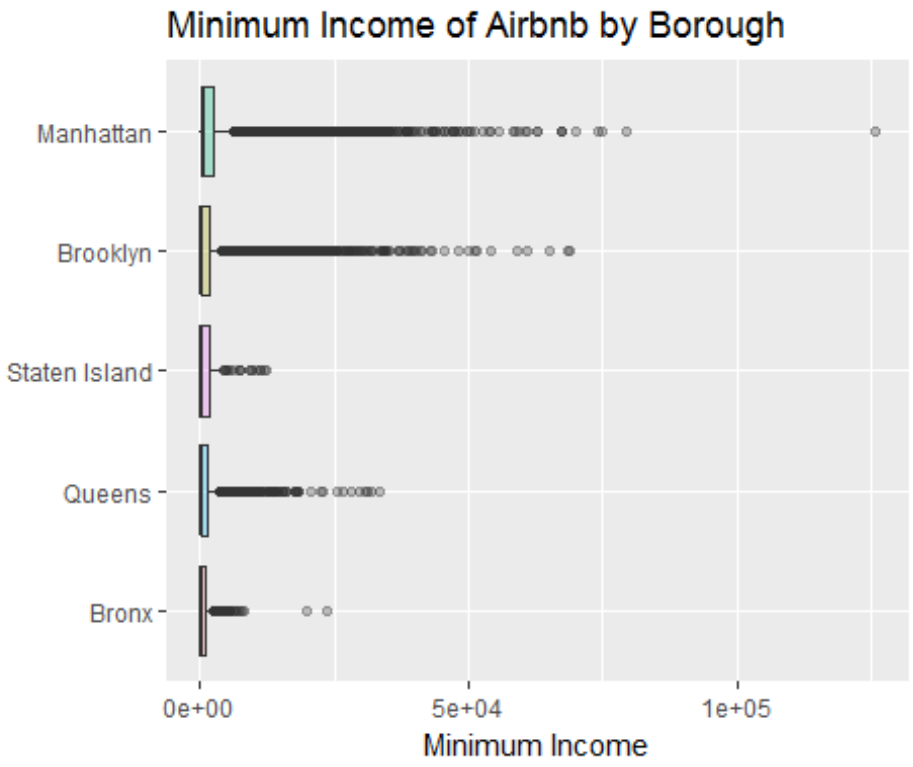
Finally I again wanted to look at the boroughs. I used ggplot2 to visualize the two income measures by borough. I did include this in the polished section because I believe the reader would like to see the results at the borough When looking at the graphs we see that using number of reviews to measure income seems to give us very low variability. I also included a regression of the income based on average availability so the reader can see if the difference are actually significant by borough.

```
j <- ggplot(data=airbnb,aes(x=reorder(neighbourhood_group_cleansed, pot
ential_income, na.rm=TRUE), y=potential_income))
j + geom_boxplot(aes(fill=neighbourhood_group_cleansed), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Potential Income") +
    guides(fill=FALSE) +
    ggtitle("Potential Income of Airbnb by Borough")
```

```
## Warning: Removed 205 rows containing non-finite values (stat_boxplot
).
```




```
j <- ggplot(data=airbnb,aes(x=reorder(neighbourhood_group_cleansed, low
_income, na.rm=TRUE), y=low_income))
j + geom_boxplot(aes(fill=neighbourhood_group_cleansed), alpha=0.3) +
    coord_flip() +
    labs(x="", y="Minimum Income") +
    guides(fill=FALSE) +
    ggtitle("Minimum Income of Airbnb by Borough")

## Warning: Removed 205 rows containing non-finite values (stat_boxplot
).
```




```
#predict income
lm = lm(potential_income ~ dummy + neighbourhood_group_cleansed + price
+ bathrooms + bedrooms, airbnb)
summary(lm)

##
## Call:
## lm(formula = potential_income ~ dummy + neighbourhood_group_cleansed
+
##     price + bathrooms + bedrooms, data = airbnb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -175252  -13072   -3388   14529  190568
##
## Coefficients:
##                                          Estimate Std. Error t val
ue
## (Intercept)                             -2619.124   1072.663  -2.4
42
## dummy                                   28736.938    626.267  45.8
86
## neighbourhood_group_cleansedBrooklyn      -4042.679   1017.617  -3.9
73
## neighbourhood_group_cleansedManhattan     -5683.348   1023.053  -5.5
55
## neighbourhood_group_cleansedQueens         -251.174   1081.002  -0.2
```

```
32
## neighbourhood_group_cleansedStaten Island  1031.690   1968.829   0.5
24
## price                                       176.076      1.417 124.2
89
## bathrooms                                  -533.172    373.807  -1.4
26
## bedrooms                                   3540.091    225.435  15.7
03
##                                                    Pr(>|t|)
## (Intercept)                                          0.0146 *
## dummy                                               < 2e-16 ***
## neighbourhood_group_cleansedBrooklyn               7.12e-05 ***
## neighbourhood_group_cleansedManhattan              2.79e-08 ***
## neighbourhood_group_cleansedQueens                   0.8163
## neighbourhood_group_cleansedStaten Island            0.6003
## price                                               < 2e-16 ***
## bathrooms                                            0.1538
## bedrooms                                            < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24950 on 39776 degrees of freedom
##   (442 observations deleted due to missingness)
## Multiple R-squared:  0.3961, Adjusted R-squared:  0.396
## F-statistic:  3261 on 8 and 39776 DF,  p-value: < 2.2e-16

#assuming they have guests every day possible
```

3

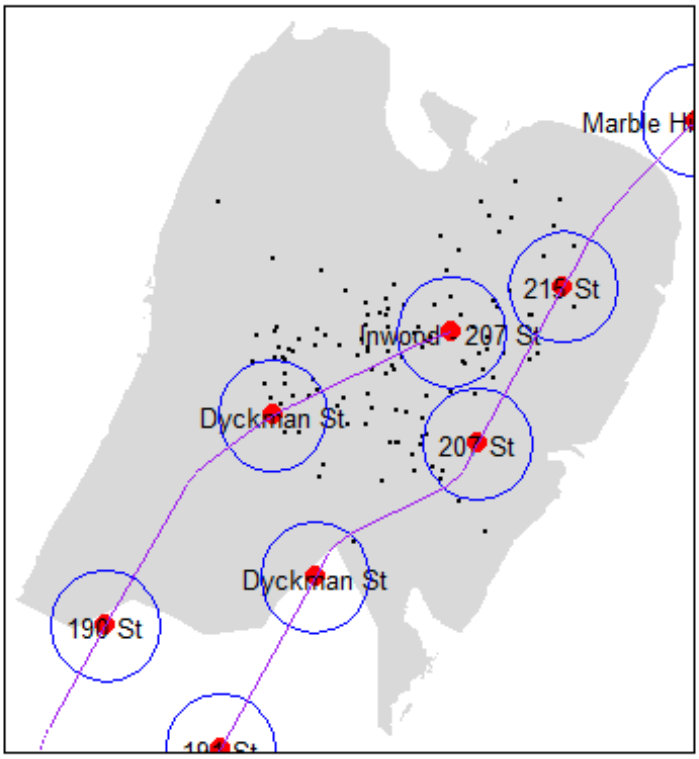
First I load the shapefiles into r. Next I subset the Airbnb dataset to get only the data on the single neighborhood Inwood. I choose this Inwood because it is where I currently live and I felt it would be interesting to see. I then make the inwood dataset a spatial object. I also subset and plot the Inwood neighbourhood from the geojson file.

```
nyc_sub_stops <- readOGR("C:\\Users\\Brandon\\Documents\\GitHub\\QMSS-G
4063-Data-Visualization\\Assignments\\Assignment 2\\nyc_subway_map\\sto
ps_nyc_subway.", "stops_nyc_subway_jan2017")

## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\Brandon\Documents\GitHub\QMSS-G4063-Data-Visualiza
tion\Assignments\Assignment 2\nyc_subway_map\stops_nyc_subway.", layer:
"stops_nyc_subway_jan2017"
## with 493 features
## It has 8 fields
```

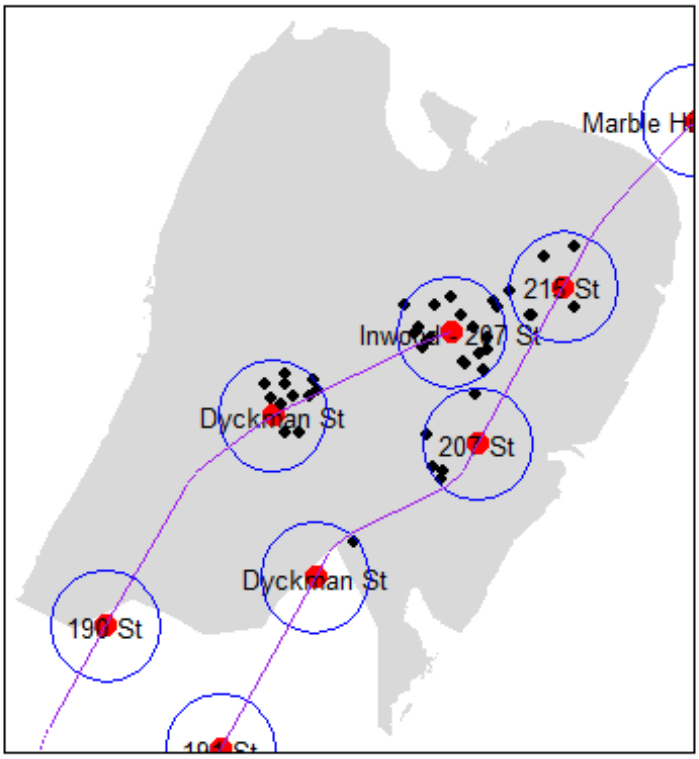
```
nyc_sub_lines <- readOGR("C:\\Users\\Brandon\\Documents\\GitHub\\QMSS-G
4063-Data-Visualization\\Assignments\\Assignment 2\\nyc_subway_map\\rou
tes_nyc_subway.", "routes_nyc_subway_jan2017")

## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\Brandon\Documents\GitHub\QMSS-G4063-Data-Visualiza
tion\Assignments\Assignment 2\nyc_subway_map\routes_nyc_subway.", layer
: "routes_nyc_subway_jan2017"
## with 25 features
## It has 7 fields
## Integer64 fields read as strings:  OBJECTID

nyc_sub_ent <- readOGR("C:\\Users\\Brandon\\Documents\\GitHub\\QMSS-G40
63-Data-Visualization\\Assignments\\Assignment 2\\nyc_subway_map\\entra
nces_nyc_subway.", "subway_entrances_may2016")

## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\Brandon\Documents\GitHub\QMSS-G4063-Data-Visualiza
tion\Assignments\Assignment 2\nyc_subway_map\entrances_nyc_subway.", la
yer: "subway_entrances_may2016"
## with 1868 features
## It has 32 fields
## Integer64 fields read as strings:  Route_8 Route_9 Route_10 Route_11

nyc_sub_stops <- spTransform(nyc_sub_stops, CRS("+proj=utm +zone=18 +da
tum=NAD27"))

## Subset + Converting Airbnb data into spatial file so it can be run i
n tmap
library(dplyr)
# using subset function
inwood <- filter(airbnb, neighbourhood == "Inwood")



bnb_inwood <- as.data.frame(inwood[,c(49:50)])
xy <- bnb_inwood[,c(2,1)]
bnb_inwood <- SpatialPoints(coords = xy, proj4string = CRS("+proj=longl
at +datum=WGS84"))

#subset neighborhood
nyc_inwood <- nyc_neighborhood[nyc_neighborhood$neighbourhood == 'Inwoo
d',]
plot(nyc_inwood)
```

I used tmap to plot the map of inwood including all listings and subway stops and subway lines and a buffer of 190 meters. I choose 190 meters because it was the largest the circles could be without combining into one. I also plotted the same thing but with only the airbnb listings within the circle. I included both as polished graphs because I feel it helps differentiate the two groups.

```
library(rgeos)
a3 <- tm_shape(nyc_inwood) + layout +
        tm_fill()

# set buffer
stop_buffer <- gBuffer(nyc_sub_stops, width = 190)

a3 + tm_shape(nyc_sub_stops) + tm_dots(col="red", size=0.5) +
      tm_text("stop_name", size=0.8) +
      tm_shape(bnb_inwood) + tm_dots(col = "black") +
      tm_shape(stop_buffer) + tm_borders(col="blue") +
      tm_shape(nyc_sub_lines) + tm_lines(col = "purple")
```




```
# print only listings inside buffer
sel_bnb_inw <- spTransform(bnb_inwood, CRS("+proj=utm +zone=18 +datum=N
AD27"))
sel_inwood <- sel_bnb_inw[stop_buffer,]

a3 + tm_shape(nyc_sub_stops) + tm_dots(col="red", size=0.6) +
      tm_text("stop_name", size=0.8) +
      tm_shape(sel_inwood) + tm_dots(col = "black", size = 0.2) +
      tm_shape(stop_buffer) + tm_borders(col="blue") +
      tm_shape(nyc_sub_lines) + tm_lines(col = "purple")
```



Once I had done that I decided to create two new datasets one of the airbnb data for only the inwood listings within the buffer circle and one for the listings from inwood not in the buffer circle. I did this in order to show summary of the price for both groups separately. I was able to see which belong to which group by comparing coordinates from the sel_inwood object and the inwood dataset. Once they see the results it seems that the closer listings cost more but is this true?

```
#subsets for in vs out of buffer
notclose <- inwood[c(3:6, 9, 10, 12:16, 19:20, 22:25, 27, 29:31, 36:38,
40:43, 45, 47:55, 57:58, 62, 64, 67, 69:70, 74:75, 77:85, 88:92, 94, 10
0:103, 105:106, 110:115, 117), ]

close1 <- inwood[c(1:2, 7:8, 11, 17:18, 21, 26, 28, 32:35, 39, 44, 46,
56, 59, 60:61, 63, 65:66, 68, 71:73, 76, 86:87, 93, 95:99, 104, 107:109
, 116), ]

summary(notclose$price)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25.00   60.00   75.00   85.57   99.00  250.00

summary(close1$price)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   30.00   57.75   78.50  103.60  100.00  485.00
```

To analyze the relationship between the distance from subways stations and the price I created a variable using a for loop and added the distance variable to the inwood dataset. I did this in order to run regression using distance from the subway stations. The regression shows there is not a statistically significant difference between price and distance. I used the regression with the control variables in the polished graphs to prove that control variables were used even though the simple regression still does not show significance.

```
## for loop is for calculating distance from nearest stop

subway <- read_csv("C:\\Users\\Brandon\\Documents\\GitHub\\QMSS-G4063-Data-Visualization\\Assignments\\Assignment 2\\NYC_Transit_Subway_Entrance_And_Exit_Data.csv")

## Parsed with column specification:
## cols(
##   .default = col_character(),
##   `Station Latitude` = col_double(),
##   `Station Longitude` = col_double(),
##   Route8 = col_integer(),
##   Route9 = col_integer(),
##   Route10 = col_integer(),
##   Route11 = col_integer(),
##   ADA = col_logical(),
##   `Free Crossover` = col_logical(),
##   `Entrance Latitude` = col_double(),
##   `Entrance Longitude` = col_double()
## )

## See spec(...) for full column specifications.

inwood$distance <- NA
for(i in 1:117){
  for(j in 1:1868){
    lon <- vector()
    lat <- vector()
    lat <- c(lat, (airbnb[[i, 49]] - subway[[j, 4]]))
    lon <- c(lon, (airbnb[[i, 50]] - subway[[j, 5]]))
    min <- min(sqrt((lat^2) + (lon^2)))
    inwood[i, 99] <- min
  }
}

#Price related to distance

lm3 <- lm(price ~ distance, inwood)
summary(lm3)

##
## Call:
```

```
## lm(formula = price ~ distance, data = inwood)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -62.97 -33.37 -14.26   6.71 394.22
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    86.92      10.16   8.555 5.91e-14 ***
## distance       74.18     121.69   0.610    0.543
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61.93 on 115 degrees of freedom
## Multiple R-squared:  0.00322,    Adjusted R-squared:  -0.005447
## F-statistic: 0.3715 on 1 and 115 DF,  p-value: 0.5434
```

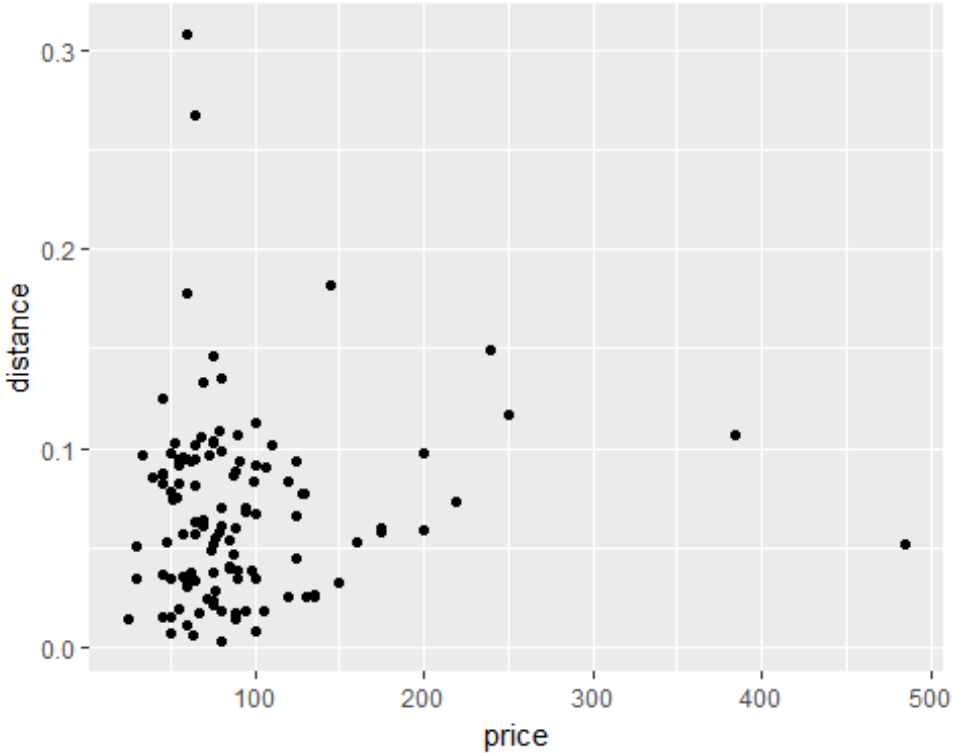
```
lm4 <- lm(price ~ distance + bathrooms + bedrooms + number_of_reviews +
bed_type + beds + accommodates + room_type + property_type, inwood)
summary(lm4)
```

```
##
## Call:
## lm(formula = price ~ distance + bathrooms + bedrooms + number_of_rev
iews +
##     bed_type + beds + accommodates + room_type + property_type,
##     data = inwood)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -103.450  -19.429   -3.346   14.930  130.716
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -75.1570    32.7047  -2.298  0.02358 *
## distance               60.9649    76.5300   0.797  0.42751
## bathrooms             101.5783    20.8322   4.876 3.95e-06 ***
## bedrooms               68.6562    10.3029   6.664 1.34e-09 ***
## number_of_reviews      -0.1407     0.1558  -0.903  0.36861
## bed_typeCouch          -7.0549    49.1702  -0.143  0.88619
## bed_typeFuton          22.5119    31.1043   0.724  0.47086
## bed_typePull-out Sofa  -6.9761    48.9171  -0.143  0.88688
## bed_typeReal Bed      -11.5427    26.1035  -0.442  0.65928
## beds                   -9.9151     7.6535  -1.295  0.19805
## accommodates            8.8412     4.6735   1.892  0.06133 .
## room_typePrivate room -24.6871     7.9702  -3.097  0.00252 **
## room_typeShared room  -59.5501    22.6186  -2.633  0.00977 **
## property_typeDorm     104.7478    35.7379   2.931  0.00416 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 35.17 on 103 degrees of freedom
## Multiple R-squared:  0.712,  Adjusted R-squared:  0.6757
## F-statistic: 19.59 on 13 and 103 DF,  p-value: < 2.2e-16
```

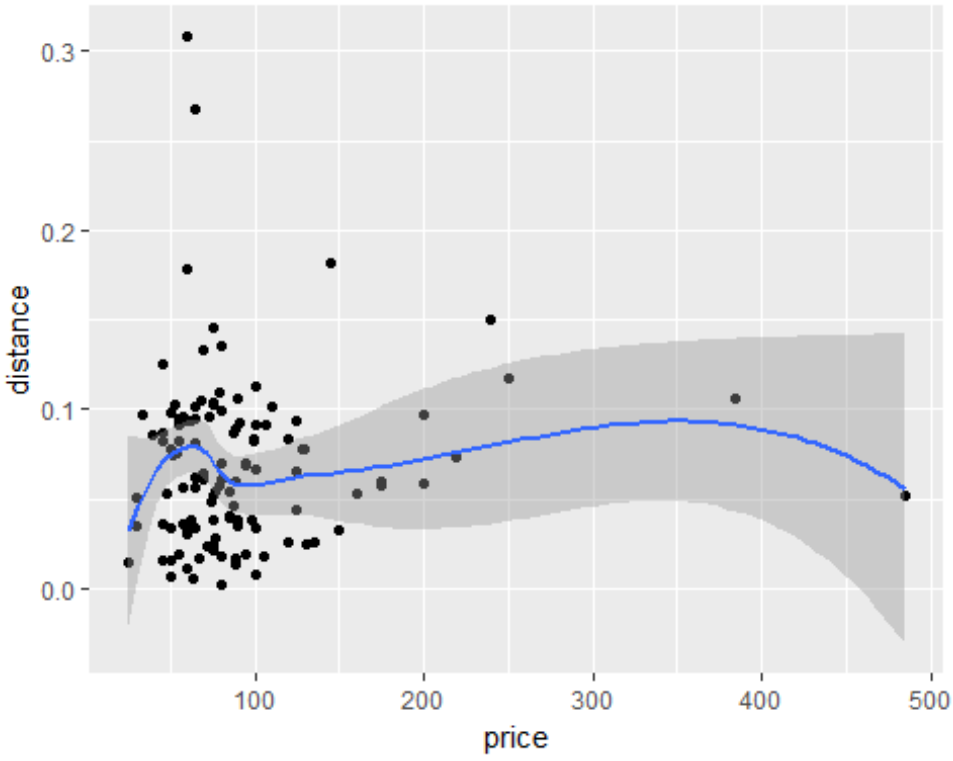
```
#visual of regression

ggplot(inwood, aes(x = price, y = distance)) +
    geom_point() +
    geom_smooth(method=lm)   # Add linear regression line
```

```
## Warning: Computation failed in `stat_smooth()`:
## 'what' must be a function or character string
```




```
ggplot(inwood, aes(x = price, y = distance)) +
    geom_point() +
    geom_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```



Last I decided to run one last regression using distance from the subway stops to see if it is statistically significant with any other variables. We see that distance is related to number of reviews and a number of other things but I concentrate on number of reviews because I feel it was important and shows that these listings "may"" be able to charge more based on distance if they are booked more often (higher demand).

```
lm5 <- lm(distance ~ price + bathrooms + bedrooms + number_of_reviews +
bed_type + beds + accommodates + room_type + property_type, inwood)
summary(lm5)

##
## Call:
## lm(formula = distance ~ price + bathrooms + bedrooms + number_of_rev
iews +
##     bed_type + beds + accommodates + room_type + property_type,
##     data = inwood)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.072200 -0.024415 -0.001375  0.017353  0.210575
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -0.0041883  0.0430392  -0.097  0.92267
## price                  0.0001004  0.0001261   0.797  0.42751
```

```
## bathrooms              -0.0319919  0.0294975  -1.085  0.28065
## bedrooms                0.0054440  0.0158111   0.344  0.73131
## number_of_reviews      -0.0004724  0.0001953  -2.418  0.01735 *
## bed_typeCouch           0.1258466  0.0618891   2.033  0.04458 *
## bed_typeFuton           0.0336654  0.0398879   0.844  0.40063
## bed_typePull-out Sofa   0.1061157  0.0619175   1.714  0.08957 .
## bed_typeReal Bed        0.0713857  0.0327913   2.177  0.03177 *
## beds                   -0.0138409  0.0098091  -1.411  0.16125
## accommodates            0.0125582  0.0059752   2.102  0.03802 *
## room_typePrivate room   0.0307691  0.0102574   3.000  0.00339 **
## room_typeShared room    0.0002973  0.0299933   0.010  0.99211
## property_typeDorm      -0.0406122  0.0475784  -0.854  0.39532
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04515 on 103 degrees of freedom
## Multiple R-squared:  0.1894, Adjusted R-squared:  0.08704
## F-statistic: 1.851 on 13 and 103 DF,  p-value: 0.04486
```

```
#visual of regression

ggplot(inwood, aes(x = number_of_reviews, y = distance)) +
    geom_point() +
    geom_smooth(method=lm)   # Add linear regression line
```

```
## Warning: Computation failed in `stat_smooth()`:
## 'what' must be a function or character string
```