Boston airbnb- Sentiment Analysis

Rohan Chouthai

July 5, 2018

SENTIMENT ANALYSIS OF BOSTON AIRBNB

This analysis aims to do a sentiment analysis of the guest reviews to understand which areas in Boston are more liked by the travellers for booking an airbnb.

This Dataset consists of 3 individual datasets: Calendar, Listings and Reviews. I have combined the Listings and Reviews datasets at a later point in my project. I'd mostly be working with the Listings dataset.

a.IMPORTING THE DATA:

First let us load all the three datasets into R.

```
Calendar<- read.csv("C:/Users/rohan/Desktop/DMML/Boston AIr BNB/calendar.csv")

Listings<- read.csv("C:/Users/rohan/Desktop/DMML/Boston AIr BNB/listings.csv")

Reviews<-read.csv("C:/Users/rohan/Desktop/DMML/Boston AIr BNB/reviews.csv")
```

a. EXPLORATORY DATA ANALYSIS:

In a city as expensive as Boston, there is a lot of curiosity around which areas in Boston are the most expensive. The Listings dataset has a lot of information about the neighbourhood and the price of the listings therein. I will now explore which areas are the most expensive in Boston.

First, let us subset the columns we require for visualizing this.

```
suppressWarnings(library(tidyverse))
```

```
## -- Attaching packages ----------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 2.2.1     v purrr   0.2.4
## v tibble  1.4.1     v dplyr   0.7.4
## v tidyr   0.7.2     v stringr 1.2.0
## v readr   1.1.1     v forcats 0.2.0
```

```
## -- Conflicts -------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
Daily_Price<- Listings%>% select(host_since,host_location,host_response_time,host_acceptance_rat
e,host_is_superhost,neighbourhood_cleansed,is_location_exact,property_type,room_type,accommodate
s,bathrooms,bedrooms,beds,bed_type,price,security_deposit,minimum_nights,maximum_nights)
dim(Daily_Price)
```

```
## [1] 3585   18
```

Now, let us explore the most expensive neighbourhoods.

```
suppressWarnings(library(ggplot2))

Daily_Price$price<-as.integer(Daily_Price$price)


Neighbourhoods<-Daily_Price%>% group_by(neighbourhood_cleansed)%>% summarise(Avg_price=mean(pric
e))%>% arrange(desc(Avg_price))
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.3
```

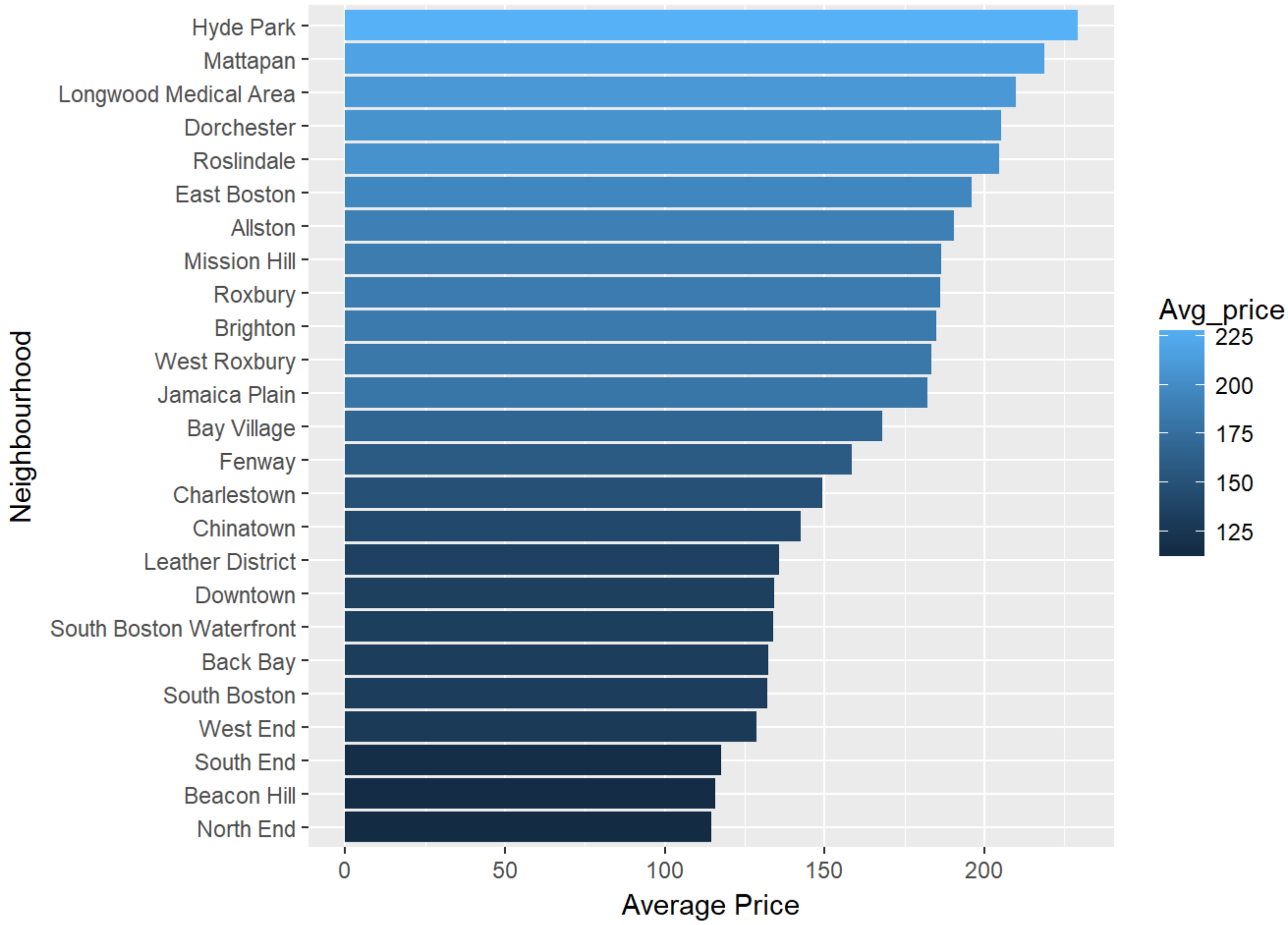
```
head(Neighbourhoods)
```

```
## # A tibble: 6 x 2
##   neighbourhood_cleansed Avg_price
##   <fct>                      <dbl>
## 1 Hyde Park                    229
## 2 Mattapan                     219
## 3 Longwood Medical Area        210
## 4 Dorchester                   205
## 5 Roslindale                   205
## 6 East Boston                  196
```

Looks like Hyde Park is the most expensive neighbourhood in Boston to be renting a bnb in. It costs a whopping $229 per night. Sure we now have the average price per neighbourhood.

Now, let us visualize the most expensive areas.

```
ggplot(Neighbourhoods)+geom_bar(mapping = aes(reorder(neighbourhood_cleansed,Avg_price),Avg_pric
e,fill=Avg_price),stat = "identity")+coord_flip()+xlab("Neighbourhood")+ylab("Average Price")
```



We can easily see that North End seems to be the cheaper place to rent out an bnb in. But what do the people who have stayed here got to say about Northend? (Sentiment analysis to follow in the last part)

b. SENTIMENT ANALYSIS

We saw in the Exploratory Data Analysis the most expensive neighbourhoods in Boston. But what did people who stayed there have to say about the neighbourhood? I want to explore the general sentiment of the neighbourhood and compare it with the average price paid for the listing in that neighbourhood.

I will use the Reviews dataset for this purpose. Then, after tokenizing the reviews per listing, I will join the price, neighbourhood and few other important columns from the Listings dataset. And then, I will proceed to analyze the sentiments and plot them for the neighbourhood.

Let us load the data in the Tidytext format.

```
suppressWarnings(library(tidyr))
suppressWarnings(library(tidytext))
```

We need our comments to be of character type. So I will first convert it into character and then unnest tokens by words.

```
Reviews$comments<-as.character(Reviews$comments)

Reviews_words<-Reviews%>% select(listing_id,comments)%>%unnest_tokens(word,comments)
```

Now, let us remove all the stop words from our dataframe.

```
Reviews_words<-Reviews_words%>% anti_join(stop_words,by="word")
```

Positive and Negative sentiment per review:

I now wish to see the overall sentiment for each neighbourhood. I will use the “Bing” sentiments for assigning a total positive and negative score to each listing, Basically, each word is matched with the Bing sentiments as falling in either positive or negative sentiment and then the number of positive and negative sentiments are counted. Ultimately, my mutating a new column called Sentiment which is the difference between the positive and negative word score for each listing, we get the overall sentiment. Lastly, I have grouped the listings by area.

```
Sentiment_reviews<-Reviews_words%>% inner_join(get_sentiments("bing"),by="word")%>% count(listin
g_id,sentiment)%>% spread(sentiment,n)%>% mutate(sentiment=positive-negative)

Sentiment_reviews<-as.tibble(Sentiment_reviews)

# Making sure I remove the NAs.
Sentiment_reviews$negative<-ifelse(Sentiment_reviews$negative %in% NA,0,Sentiment_reviews$negati
ve)
Sentiment_reviews$positive<-ifelse(Sentiment_reviews$positive %in% NA,0,Sentiment_reviews$positi
ve)

Sentiment_reviews$sentiment<-Sentiment_reviews$positive-Sentiment_reviews$negative

Sentiment_reviews_top10<-Sentiment_reviews%>% arrange(desc(Sentiment_reviews$sentiment))%>% top_
n(10)
```

```
## Selecting by sentiment
```

```
str(Sentiment_reviews)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    2749 obs. of  4 variables:
##  $ listing_id: int  3353 5506 6695 6976 8792 9273 9765 9824 9855 9857 ...
##  $ negative  : num  32 7 34 10 4 10 7 25 0 11 ...
##  $ positive  : num  150 167 233 232 106 67 33 81 15 69 ...
##  $ sentiment : num  118 160 199 222 102 57 26 56 15 58 ...
```

```
colnames(Sentiment_reviews)<-c("id","negative","positive","sentiment") # amking sure the primary
 key of the dataframe matches with the primary key of the Listings dataframe.
```

Let us determine how the sentiment is related to the average rating of the listing. I will join the Listings dataframe to the Sentiment_reviews dataframe so that we get information about all the listings we are analyzing sentiments for.

```
Sentiment_analysis<-Sentiment_reviews%>% left_join(Listings,by="id")

head(Sentiment_analysis)
```

```
## # A tibble: 6 x 98
##      id negative positive sentiment listing_url     scrape_id last_scraped
##   <int>    <dbl>    <dbl>     <dbl> <fct>               <dbl> <fct>
## 1  3353    32.0     150       118   https://www.ai~   2.02e13 2016-09-07
## 2  5506     7.00    167       160   https://www.ai~   2.02e13 2016-09-07
## 3  6695    34.0     233       199   https://www.ai~   2.02e13 2016-09-07
## 4  6976    10.0     232       222   https://www.ai~   2.02e13 2016-09-07
## 5  8792     4.00    106       102   https://www.ai~   2.02e13 2016-09-07
## 6  9273    10.0      67.0      57.0 https://www.ai~   2.02e13 2016-09-07
## # ... with 91 more variables: name <fct>, summary <fct>, space <fct>,
## #   description <fct>, experiences_offered <fct>, neighborhood_overview
## #   <fct>, notes <fct>, transit <fct>, access <fct>, interaction <fct>,
## #   house_rules <fct>, thumbnail_url <fct>, medium_url <fct>, picture_url
## #   <fct>, xl_picture_url <fct>, host_id <int>, host_url <fct>, host_name
## #   <fct>, host_since <fct>, host_location <fct>, host_about <fct>,
## #   host_response_time <fct>, host_response_rate <fct>,
## #   host_acceptance_rate <fct>, host_is_superhost <fct>,
## #   host_thumbnail_url <fct>, host_picture_url <fct>, host_neighbourhood
## #   <fct>, host_listings_count <int>, host_total_listings_count <int>,
## #   host_verifications <fct>, host_has_profile_pic <fct>,
## #   host_identity_verified <fct>, street <fct>, neighbourhood <fct>,
## #   neighbourhood_cleansed <fct>, neighbourhood_group_cleansed <lgl>, city
## #   <fct>, state <fct>, zipcode <fct>, market <fct>, smart_location <fct>,
## #   country_code <fct>, country <fct>, latitude <dbl>, longitude <dbl>,
## #   is_location_exact <fct>, property_type <fct>, room_type <fct>,
## #   accommodates <int>, bathrooms <dbl>, bedrooms <int>, beds <int>,
## #   bed_type <fct>, amenities <fct>, square_feet <int>, price <fct>,
## #   weekly_price <fct>, monthly_price <fct>, security_deposit <fct>,
## #   cleaning_fee <fct>, guests_included <int>, extra_people <fct>,
## #   minimum_nights <int>, maximum_nights <int>, calendar_updated <fct>,
## #   has_availability <lgl>, availability_30 <int>, availability_60 <int>,
## #   availability_90 <int>, availability_365 <int>, calendar_last_scraped
## #   <fct>, number_of_reviews <int>, first_review <fct>, last_review <fct>,
## #   review_scores_rating <int>, review_scores_accuracy <int>,
## #   review_scores_cleanliness <int>, review_scores_checkin <int>,
## #   review_scores_communication <int>, review_scores_location <int>,
## #   review_scores_value <int>, requires_license <fct>, license <lgl>,
## #   jurisdiction_names <lgl>, instant_bookable <fct>, cancellation_policy
## #   <fct>, require_guest_profile_picture <fct>,
## #   require_guest_phone_verification <fct>, calculated_host_listings_count
## #   <int>, reviews_per_month <dbl>
```

Now, let us select the neighbourhood and the price along with the sentiments.

```
Sa<-Sentiment_analysis%>% group_by(neighbourhood_cleansed)%>% summarise(Sentiment=mean(sentimen
t),Price=mean(as.numeric(price)))

head(Sa)
```

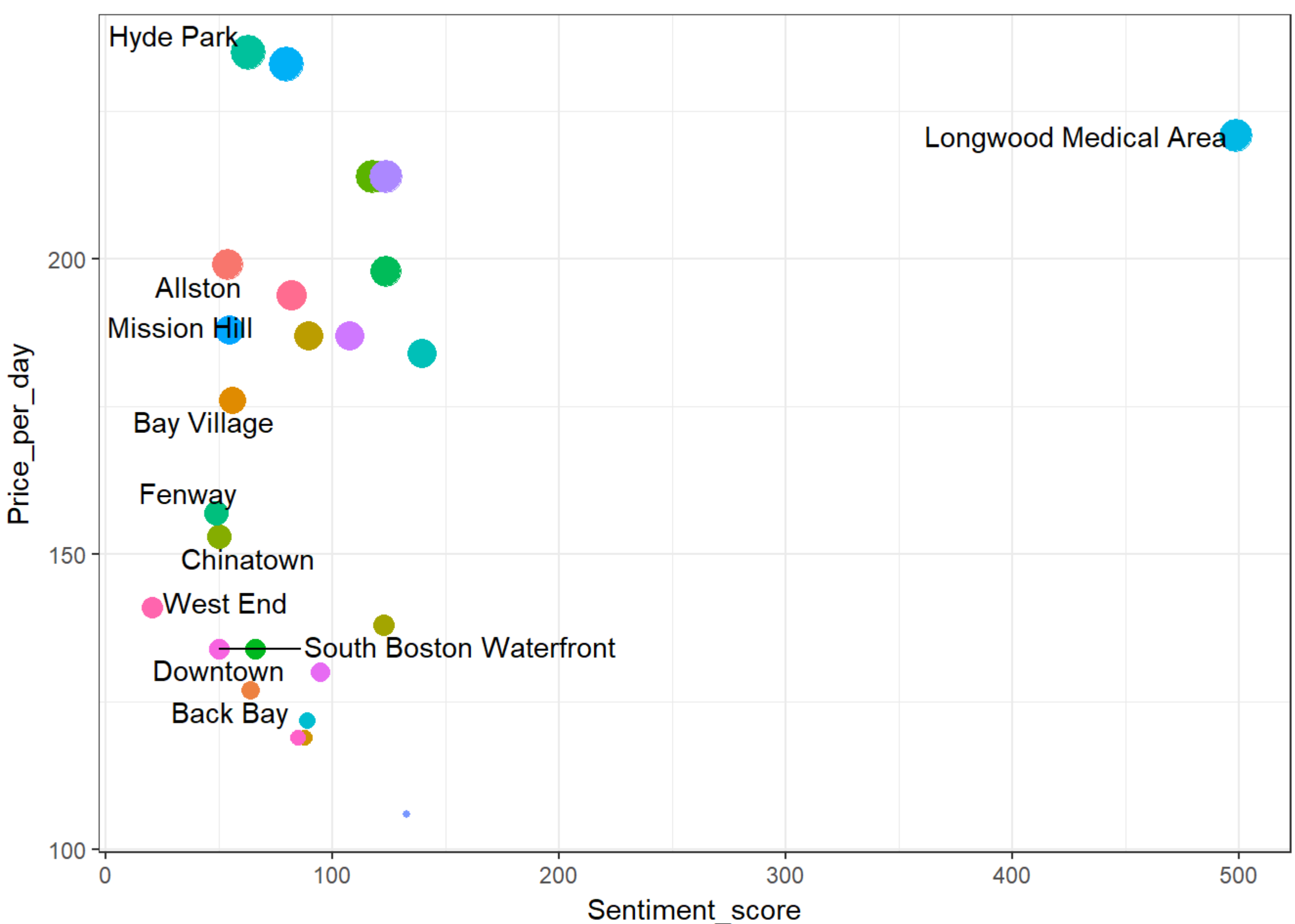
```
## # A tibble: 6 x 3
##   neighbourhood_cleansed Sentiment Price
##   <fct>                      <dbl> <dbl>
## 1 Allston                     54.7   200
## 2 Back Bay                    64.9   128
## 3 Bay Village                 56.5   176
## 4 Beacon Hill                 88.9   119
## 5 Brighton                    91.0   187
## 6 Charlestown                123     139
```

Now, let us plot the sentiment vs price.

```
suppressWarnings(library(ggrepel))

ggplot(data = Sa,mapping = aes(x=as.integer(Sa$Sentiment),y=as.integer(Sa$Price)))+geom_point(ae
s(color=neighbourhood_cleansed,size=(Price)))+xlab("Sentiment_score")+ylab("Price_per_day")+geom
_text_repel(aes(x=as.integer(Sa$Sentiment),y=as.integer(Sa$Price), hjust = 1 ,label=ifelse(as.in
teger(Sa$Sentiment)>400,as.character(neighbourhood_cleansed),''))) + theme_bw() + theme(legend.p
osition="none") + geom_text_repel(aes(as.integer(Sa$Sentiment),as.integer(Sa$Price) ,label=ifels
e((as.integer(Sa$Sentiment) < 70),as.character(neighbourhood_cleansed),''))) + theme_bw() + them
e(legend.position="none")
```

```
## Warning: Ignoring unknown aesthetics: hjust
```



Thus, we can see that even if Hyde park neighbourhood is one of the most expensive, it’s sentiment score isnt all that high. Longwood medical area, on the other hand has a high sentiment score as well as a high daily price.