Hotels with R

James Newsome

August 2, 2020

Table of Contents

[I. Import Libraries and Data 2](#_Toc47437666)

[A. Libraires 2](#_Toc47437667)

[B. Data 2](#_Toc47437668)

[1. Quick peek 2](#_Toc47437669)

[2. Adjust column names 3](#_Toc47437670)

[3. Read in full data set 3](#_Toc47437671)

[II. Cleaning 3](#_Toc47437672)

[A. Convert column types 3](#_Toc47437673)

[B. Find NAs 4](#_Toc47437674)

[C. Fix Missing Lat and Long 4](#_Toc47437675)

[1. Preparation 4](#_Toc47437676)

[2. Google Geocoding API 5](#_Toc47437677)

[3. Insert missing Lat and Long 5](#_Toc47437678)

[D. Duplicate Hotel Names or Addresses 6](#_Toc47437679)

[1. How many unique hotels and addresses are there? 6](#_Toc47437680)

[2. Fix Duplicate Hotels 7](#_Toc47437681)

[3. Fix Duplicate Addresses 7](#_Toc47437682)

[E. Touch up 11](#_Toc47437683)

[III. Plotting the Data 12](#_Toc47437684)

[A. Nationality that reviews the most (Barplot) 12](#_Toc47437685)

[B. Word count in positive reviews vs negative reviews (Histogram) 13](#_Toc47437686)

[C. Reviewer score base off word count (Boxplot) 15](#_Toc47437687)

[1. Small number of words 15](#_Toc47437688)

[2. Higher word count 16](#_Toc47437689)

[D. Ratings of hotels (Barplot) 18](#_Toc47437690)

[IV. Abandoned Cleaning 18](#_Toc47437691)

[A. Filter by Word Count 18](#_Toc47437692)

[B. Correct Reviews and Word Count 20](#_Toc47437693)

## I. Import Libraries and Data

### A. Libraires

library(readr)  
library(dplyr)  
library(jsonlite)  
library(RCurl)  
library(stringr)  
library(ggplot2)  
library(gridExtra)  
library(DataCombine)

### B. Data

#### 1. Quick peek

This is large dataset with 515k rows at 80MB, so read in small amount to get a peek and see how to preformat it before importing full set

mindata <- read\_csv('Hotel\_Reviews.csv', n\_max=10)  
View(mindata)

glimpse(mindata)

## Observations: 10  
## Variables: 17  
## $ Hotel\_Address <chr> "s Gravesandestraat...  
## $ Additional\_Number\_of\_Scoring <dbl> 194, 194, 194, 194,...  
## $ Review\_Date <chr> "8/3/2017", "8/3/20...  
## $ Average\_Score <dbl> 7.7, 7.7, 7.7, 7.7,...  
## $ Hotel\_Name <chr> "Hotel Arena", "Hot...  
## $ Reviewer\_Nationality <chr> "Russia", "Ireland"...  
## $ Negative\_Review <chr> "I am so angry that...  
## $ Review\_Total\_Negative\_Word\_Counts <dbl> 397, 0, 42, 210, 14...  
## $ Total\_Number\_of\_Reviews <dbl> 1403, 1403, 1403, 1...  
## $ Positive\_Review <chr> "Only the park outs...  
## $ Review\_Total\_Positive\_Word\_Counts <dbl> 11, 105, 21, 26, 8,...  
## $ Total\_Number\_of\_Reviews\_Reviewer\_Has\_Given <dbl> 7, 7, 9, 1, 3, 1, 6...  
## $ Reviewer\_Score <dbl> 2.9, 7.5, 7.1, 3.8,...  
## $ Tags <chr> "[' Leisure trip ',...  
## $ days\_since\_review <chr> "0 days", "0 days",...  
## $ lat <dbl> 52.36058, 52.36058,...  
## $ lng <dbl> 4.915968, 4.915968,...

rm(mindata)

There are 17 columns. ReviewDate needs to be in date format, but will have to be dealt with after import. Some doubles should be integers. Need to rename the columns to make them smaller.

#### 2. Adjust column names

colnames <- c('Address', 'AddNumOfScoring', 'ReviewDate', 'AvgScore', 'Hotel', 'ReviewerNationality', 'NegReview', 'NegReviewWordCount', 'TotalReviews', 'PosReview', 'PosReviewWordCount', 'TotReviewsByReviewer', 'ReviewerScore', 'Tags', 'DaysSinceReview', 'Lat', 'Long')

#### 3. Read in full data set

alldata <- read\_csv('Hotel\_Reviews.csv', col\_types='cicdccciiciidccdd', skip=1, col\_names=colnames)  
glimpse(alldata)

## Observations: 515,738  
## Variables: 17  
## $ Address <chr> "s Gravesandestraat 55 Oost 1092 AA Amste...  
## $ AddNumOfScoring <int> 194, 194, 194, 194, 194, 194, 194, 194, 1...  
## $ ReviewDate <chr> "8/3/2017", "8/3/2017", "7/31/2017", "7/3...  
## $ AvgScore <dbl> 7.7, 7.7, 7.7, 7.7, 7.7, 7.7, 7.7, 7.7, 7...  
## $ Hotel <chr> "Hotel Arena", "Hotel Arena", "Hotel Aren...  
## $ ReviewerNationality <chr> "Russia", "Ireland", "Australia", "United...  
## $ NegReview <chr> "I am so angry that i made this post avai...  
## $ NegReviewWordCount <int> 397, 0, 42, 210, 140, 17, 33, 11, 34, 15,...  
## $ TotalReviews <int> 1403, 1403, 1403, 1403, 1403, 1403, 1403,...  
## $ PosReview <chr> "Only the park outside of the hotel was b...  
## $ PosReviewWordCount <int> 11, 105, 21, 26, 8, 20, 18, 19, 0, 50, 10...  
## $ TotReviewsByReviewer <int> 7, 7, 9, 1, 3, 1, 6, 1, 3, 1, 2, 12, 7, 6...  
## $ ReviewerScore <dbl> 2.9, 7.5, 7.1, 3.8, 6.7, 6.7, 4.6, 10.0, ...  
## $ Tags <chr> "[' Leisure trip ', ' Couple ', ' Duplex ...  
## $ DaysSinceReview <chr> "0 days", "0 days", "3 days", "3 days", "...  
## $ Lat <dbl> 52.36058, 52.36058, 52.36058, 52.36058, 5...  
## $ Long <dbl> 4.915968, 4.915968, 4.915968, 4.915968, 4...

rm(colnames)

## II. Cleaning

### A. Convert column types

Need to convert ‘DaysSinceReview’ to integer format and convert ‘ReviewDate’ to Datetime format

alldata$DaysSinceReview <- as.integer((gsub("[^0-9]", "", alldata$DaysSinceReview)))  
  
alldata$ReviewDate <- as.Date(alldata$ReviewDate, '%m/%d/%Y')  
glimpse(alldata[c(3,15)]) # Verify that the changes were made

## Observations: 515,738  
## Variables: 2  
## $ ReviewDate <date> 2017-08-03, 2017-08-03, 2017-07-31, 2017-07-3...  
## $ DaysSinceReview <int> 0, 0, 3, 3, 10, 10, 17, 17, 25, 26, 27, 28, 28...

### B. Find NAs

dfNAs <- alldata[colSums(is.na(alldata)) > 0]  
writeLines(paste('There are', length(names(dfNAs)), 'columns with NA values:'))

## There are 5 columns with NA values:

writeLines(names(dfNAs))

## ReviewerNationality  
## NegReview  
## PosReview  
## Lat  
## Long

rm(dfNAs)

Of these 5 columns, I am only going to handle ‘Lat’ and ‘Long’.

### C. Fix Missing Lat and Long

#### 1. Preparation

Collect the rows with NA.

nolatlong <- filter(alldata,is.na((alldata$Lat) | (alldata$Long)))  
writeLines(paste('Number of rows missing a Lat or Long:', nrow(nolatlong)))

## Number of rows missing a Lat or Long: 3268

I assume that if a hotel is missing the Lat, that it will also be missing the Long. So that there aren’t any hotels that have one, but not the other, of the Lat/Long pair.

writeLines(paste('Number of rows missing a Lat and Long:', nrow(filter(alldata,is.na((alldata$Lat) & (alldata$Long))))))

## Number of rows missing a Lat and Long: 3268

Since the number missing a Lat OR Long is the same as the number missing a Lat AND Long, then each row that is missing one is also missing the other.

uniqueaddresses <- distinct(nolatlong, Address)  
writeLines(paste('There are', nrow(uniqueaddresses), 'distinct addresses with no Lat or Long'))

## There are 17 distinct addresses with no Lat or Long

Need to add empty columns to ‘uniqueaddresses’ for the missing ‘Lat’ and ‘Long’

uniqueaddresses <- mutate(uniqueaddresses, Lat='', Long='')

#### 2. Google Geocoding API

Need to use Google’s Geocoding API to find and fill in the missing Lat/Long based off the hotels address. To do this, I will need a Google APIKey. This key is not provided in this document, so you will need to provide your own to run the code. I concatenate the address with the API key and base url to access Google’s Geocoding API to grab the Lat/Long and add it to the ‘uniqueaddresses’ dataframe.

googleaddress <- 'https://maps.googleapis.com/maps/api/geocode/json?address='  
APIKey <- 'Need to provide your own'  
for (i in 1:nrow(uniqueaddresses)){  
 address <- uniqueaddresses[i,1]  
 address <- gsub(' ', '+', address)  
 address <- paste(address, '&key=', APIKey, sep='')  
 address <- paste(googleaddress, address, sep='')  
 jdata <- getURL(address) # Fetch JSON data.  
 rdata <- fromJSON(jdata, flatten=TRUE) # Convert a JSON object into an R object.  
 results <- rdata[[1]]  
 latlong <- select(results,Lat=geometry.location.lat, Long=geometry.location.lng)  
 uniqueaddresses[i,2:3] <- latlong  
}  
rm(list=setdiff(ls(), c('alldata', 'uniqueaddresses', 'nolatlong')))

#### 3. Insert missing Lat and Long

The ‘uniqueaddresses’ dataframe now has the correct Lat/Long with each address and now needs to be added back to the ‘alldata’ dataframe. I start by splitting ‘alldata’ into those that have Lat/Long and those without. I already have those without in the ‘nolatlong’ dataframe. Then sort both dataframes by address and find on which row each new address begins.

havelatlong <- filter(alldata,!is.na((alldata$Lat)))  
nolatlong <- arrange(nolatlong, Address) # Sort by address  
uniqueaddresses <- arrange(uniqueaddresses, Address) # Sort by address  
# Find where each new address begins in nolatlong.  
index <- match(uniqueaddresses$Address, nolatlong$Address)  
index[18] <- nrow(nolatlong)+1 # Create an ending point for the loop

Now I loop through the indexes of where each new row starts, to put correct lat/long in nolatlong based off the start and end points of each address in nolatlong.

for (i in 1:(length(index)-1)){  
 start <- index[i]  
 end <- index[i+1]-1  
 for (j in start:end){  
 nolatlong[j,16:17] <- uniqueaddresses[i,2:3]  
 }  
}

Finally I combine nolatlong with havelatlong to get back the original dataframe with the missing Lat/Long values now filled in. Then verify that there are now only three columns with NA values.

newdata <- rbind(havelatlong, nolatlong)  
# Verify no NAs in Lat or Long  
writeLines(paste('There are now', length(newdata[colSums(is.na(newdata)) > 0]), 'columns with NA values'))

## There are now 3 columns with NA values

rm(list=setdiff(ls(), 'newdata'))

### D. Duplicate Hotel Names or Addresses

#### 1. How many unique hotels and addresses are there?

disthotadd <- distinct(newdata, Hotel, Address)  
writeLines(paste('There are', nrow(disthotadd), 'rows with a distinct pair of hotel names and addresses.'))

## There are 1494 rows with a distinct pair of hotel names and addresses.

writeLines(paste(nrow(distinct(newdata, Hotel)), 'distinct hotels'))

## 1492 distinct hotels

writeLines(paste(nrow(distinct(newdata, Address)), 'distinct addresses'))

## 1493 distinct addresses

So there are hotels with duplicate addresses, and addresses with duplicate hotel names. I need to find hotels with extra addresses, and addresses used by multiple hotels.

# Hotels with more than one address  
extraaddresses <- disthotadd %>% group\_by(Hotel) %>% filter(n()>1)  
glimpse(extraaddresses)

## Observations: 3  
## Variables: 2  
## Groups: Hotel [1]  
## $ Hotel <chr> "Hotel Regina", "Hotel Regina", "Hotel Regina"  
## $ Address <chr> "Bergara 2 4 Eixample 08002 Barcelona Spain", "Rooseve...

Here we see that Hotel Regina has three different locations.

# Address with more than one hotel name  
extrahotels <- disthotadd %>% group\_by(Address) %>% filter(n()>1)  
glimpse(extrahotels)

## Observations: 2  
## Variables: 2  
## Groups: Address [1]  
## $ Hotel <chr> "The Grand at Trafalgar Square", "Club Quarters Hotel ...  
## $ Address <chr> "8 Northumberland Avenue Westminster Borough London WC...

And there are two hotels at ‘8 Northumberland Avenue…’.

#### 2. Fix Duplicate Hotels

For Hotel Regina, the addresses are:

print(extraaddresses[,2])

## # A tibble: 3 x 1  
## Address   
## <chr>   
## 1 Bergara 2 4 Eixample 08002 Barcelona Spain   
## 2 Rooseveltplatz 15 09 Alsergrund 1090 Vienna Austria   
## 3 Via Cesare Correnti 13 Milan City Center 20123 Milan Italy

Each Hotel Regina is in a different county, so I can simply rename each based off its location. Thus I will have Hotel Regina Spain, etc.

spainaddress <- as.character(extraaddresses[1,2])  
austriaaddress <- as.character(extraaddresses[2,2])  
italyaddress <- as.character(extraaddresses[3,2])  
newdata[(newdata$Address == spainaddress), 5] <- 'Hotel Regina Spain'  
newdata[(newdata$Address == italyaddress), 5] <- 'Hotel Regina Italy'  
newdata[(newdata$Address == austriaaddress), 5] <- 'Hotel Regina Austria'

There were originally 1492 distinct hotel names. Since I added two additional ones, I should be at 1494.

disthotadd <- distinct(newdata, Hotel, Address)  
writeLines(paste(nrow(distinct(newdata, Hotel)), 'distinct hotels'))

## 1494 distinct hotels

rm(extraaddresses, spainaddress, austriaaddress, italyaddress)

#### 3. Fix Duplicate Addresses

print(extrahotels[,1])

## # A tibble: 2 x 1  
## Hotel   
## <chr>   
## 1 The Grand at Trafalgar Square   
## 2 Club Quarters Hotel Trafalgar Square

These are the two hotel names that share the same address. But here I have to determine if the hotels are the same. I start by seeing if they have reviews during the same time to see if they were in operation at the same time. I need to make sure one didn’t buy out or replace the other, which would be evident by the review dates either not overlapping or just barely overlapping.

hotel1 <- filter(newdata, Hotel==as.character(extrahotels[1,1]))  
hotel2 <- filter(newdata, Hotel==as.character(extrahotels[2,1]))  
mindate1 <- min(hotel1$ReviewDate)  
maxdate1 <- max(hotel1$ReviewDate)  
mindate2 <- min(hotel2$ReviewDate)  
maxdate2 <- max(hotel2$ReviewDate)  
writeLines(paste('The Grand has reviews from', mindate1, 'to', maxdate1))

## The Grand has reviews from 2015-08-04 to 2017-08-03

writeLines(paste('Club Quarters has reviews from', mindate2, 'to', maxdate2))

## Club Quarters has reviews from 2015-08-04 to 2017-08-03

rm(mindate1, mindate2, maxdate1, maxdate2)

They have reviews during the exact same time period. Perhaps this is the the time period that all results lie in. Need to make sure, as this is too coincidental otherwise.

writeLines(paste('Dates in the data set run from', min(newdata$ReviewDate), 'to', max(newdata$ReviewDate)))

## Dates in the data set run from 2015-08-04 to 2017-08-03

So this is the extent of the time period the data is from. Since they have reviews throughtout this same time period, they are either the same hotel or share an address. Possibly they operate under different names but share a staff. I can check the reviews for mentions of the one hotel in a review of the other. I start with reviews of Club Quarters that mention The Grand.

reviews <- filter(hotel2, (grepl('rand', hotel2$NegReview)))  
print(reviews$NegReview)

## [1] "The double room was very small The lobby was not very grand nowhere to sit"   
## [2] "A bit pricey eat out if possible loads of places in Whitehall and The Strand"   
## [3] "Bed was very small for a double and very uncomfortable There were no bar facilities after 11 0 clock so coming in after a lovely evening on the river we were unable to warm up with a coffee and brandy The staff were very dismissive about it It didn t have the feel of a hotel just almost an apartment block Very impersonal"

There is no mention of The Grand in the negative comments. Next scan the positive reviews.

reviews <- filter(hotel2, (grepl('rand', hotel2$PosReview)))  
print(reviews$PosReview)

## [1] "Excellent location within steps to Embankment tube station shopping strand and theatres Small exercise room on site Kettle and flat screen telly in room Bottled water on every floor Quiet in room despite being in the middle of all the action Staff were friendly and helpful throughout Good wifi in room and common areas of hotel"   
## [2] "Always stay at the Grand when we are in London Staff are always pleasant and helpful Rooms Clean and well stocked The effort made for my wife s birthday stay a card a bottle of wine and some sweet treats a gesture well appreciated Well Done"   
## [3] "I mentioned when I booked the hotel that it was my Birthday on the date of stay and when I arrived at the hotel we were informed that we had been placed in 2 rooms with great views which was a nice surprise and they also wished me happpy birthday during the check in which was lovely When I got to my room they had left a bottle of wine and some snacks and a little London fridge magnet which I thought was another lovely touch There was also a card wishing me a happy birthday from the hotel management Needless to say this started my visit to London off very well indeed The room was larger than I expected despite only being the Small room as described on the website and the bed was very comfortable I could hear no road noise or noise from other guests at all over night and I was generally very happy overall with the room standard and all it s decor furniture and supplies The shower was hot and very powerful and the complimentary toiletries were of a high standard Check in out was very quick and the staff were cheerful and professional The entrance hall looks great it is shared by the Grand Hotel and the bar and lobby area is beautiful"  
## [4] "I was with my daughter 2 grandchildren we had everything they wanted to see within walking distance fab location"   
## [5] "We booked in an emergency from the hotel lobby Our other hotel reservation was cancelled by the Hotwire Risk Management team without any kind of notice so we were in total surprise We walked back to The Grand this was our other hotel of choice when we were booking the previous day talked with James on the reception booked online and had two room keys in about 6 7 minutes after entering Two 8 floor rooms with actual view over the adjacent rooftops The potential view less room s was the reason we put The Grand on a second place on our initial choice Breakfast was nice and plentiful tendering to the room and our belongings was perfect the personnel was polite helpful and fast Great stay thanks for saving us in the middle of London P S The mattresses and linens were just perfect"   
## [6] "Great central place to stay when sight seeing Lots of places walkable other can be easily reach by tube from Charing Cross tube The grand children loved every minute"   
## [7] "Lucky to get a room with brand new carpet Great bed Overall an OK stay nothing negative nothing impressive except for location Would stay again value for the money very good"   
## [8] "Amazing location Right in the heart of London with Trafalgar Square a 30 second walk away The Strand 1 minute away and Covent Garden a 10 minute stroll Great lobby impressive bar area Good cocktail menu Like the water refill stations on each floor too Very clean"

Reviews 2 and 5 seem to think they are staying at The Grand or that the two hotels are the same. Next I look at the negative reviews of The Grand that mention Club Quarters.

reviews <- filter(hotel1, (grepl('lub', hotel1$NegReview)))  
print(reviews$NegReview)

## [1] "Effectively this is Club Quarters the two are in the same building and share a front desk The rooms and public spaces are definitely not as nice as the pictures And query the cleanliness when I opened the curtains moths flew out"   
## [2] "Traffic noise heard from front facing room but moved to very quiet room in Club Quarters"   
## [3] "Well we booked the Grand and ended up in Club Quarters I use them on business but would not have used them for a weekend away they make me feel as if I am still working so I felt I had been missold why not promote as CQ if thats what you are selling I guess I am not alone and that fills the rooms at the weekend"  
## [4] "Fairly small room if you think you have booked The Grand you have actually booked Club Quarters We booked Club last year and this year booked the Grand we ended up in the same room as last year just 1 door down No drinking water in the room Only two small lifts"

Review 1 claims that the two are effectively the same as they share a building and front desk. Reviews 3 and 4 say that even when booking The Grand, they may put you in Club. I could probably stop here, but might as well look at the positive reviews.

reviews <- filter(hotel1, (grepl('lub', hotel1$PosReview)))  
print(reviews$PosReview)

## [1] "Beautiful building decor really salubrious beautiful entrance lobby good size rooms and comfortable beds too"

rm(reviews)

No mention of Club Quarters here.

Since these appear to be the same, I could name them the same. However, I’d like to first check their reviews to see if there is a substantial difference.

writeLines(paste('The Average Review Score of The Grand is:', max(hotel1$AvgScore)))

## The Average Review Score of The Grand is: 8.3

writeLines(paste('The Average Review Score of Club Quarters is:', max(hotel2$AvgScore)))

## The Average Review Score of Club Quarters is: 8.5

I think it is acceptable to merge these two hotels into one hotel named, ‘Club Quarters / The Grand’, since they appear to be acting as one.

address <- as.character(extrahotels[1,2])  
newdata[(newdata$Address == address), 5] <- 'Club Quarters / The Grand'  
rm(address, extrahotels, hotel1, hotel2, extraaddresses, austriaaddress, italyaddress, spainaddress)

## Warning in rm(address, extrahotels, hotel1, hotel2, extraaddresses,  
## austriaaddress, : object 'extraaddresses' not found

## Warning in rm(address, extrahotels, hotel1, hotel2, extraaddresses,  
## austriaaddress, : object 'austriaaddress' not found

## Warning in rm(address, extrahotels, hotel1, hotel2, extraaddresses,  
## austriaaddress, : object 'italyaddress' not found

## Warning in rm(address, extrahotels, hotel1, hotel2, extraaddresses,  
## austriaaddress, : object 'spainaddress' not found

I started with 1492 distinct hotels. This increased to 1494 after separating out the Hotel Reginas. After combining these two, I now verify there are only 1493 unique hotels.

disthotadd <- distinct(newdata, Hotel, Address)  
writeLines(paste('Distinct hotels:', nrow(distinct(newdata, Hotel))))

## Distinct hotels: 1493

rm(disthotadd)

### E. Touch up

Great. Now take another quick look at our data.

glimpse(newdata)

## Observations: 515,738  
## Variables: 17  
## $ Address <chr> "s Gravesandestraat 55 Oost 1092 AA Amste...  
## $ AddNumOfScoring <int> 194, 194, 194, 194, 194, 194, 194, 194, 1...  
## $ ReviewDate <date> 2017-08-03, 2017-08-03, 2017-07-31, 2017...  
## $ AvgScore <dbl> 7.7, 7.7, 7.7, 7.7, 7.7, 7.7, 7.7, 7.7, 7...  
## $ Hotel <chr> "Hotel Arena", "Hotel Arena", "Hotel Aren...  
## $ ReviewerNationality <chr> "Russia", "Ireland", "Australia", "United...  
## $ NegReview <chr> "I am so angry that i made this post avai...  
## $ NegReviewWordCount <int> 397, 0, 42, 210, 140, 17, 33, 11, 34, 15,...  
## $ TotalReviews <int> 1403, 1403, 1403, 1403, 1403, 1403, 1403,...  
## $ PosReview <chr> "Only the park outside of the hotel was b...  
## $ PosReviewWordCount <int> 11, 105, 21, 26, 8, 20, 18, 19, 0, 50, 10...  
## $ TotReviewsByReviewer <int> 7, 7, 9, 1, 3, 1, 6, 1, 3, 1, 2, 12, 7, 6...  
## $ ReviewerScore <dbl> 2.9, 7.5, 7.1, 3.8, 6.7, 6.7, 4.6, 10.0, ...  
## $ Tags <chr> "[' Leisure trip ', ' Couple ', ' Duplex ...  
## $ DaysSinceReview <int> 0, 0, 3, 3, 10, 10, 17, 17, 25, 26, 27, 2...  
## $ Lat <chr> "52.3605759", "52.3605759", "52.3605759",...  
## $ Long <chr> "4.9159683", "4.9159683", "4.9159683", "4...

The Lat and Long were converted to characters when I added the missing ones. Let’s convert them to doubles and verify.

newdata$Lat <- as.numeric(newdata$Lat)  
newdata$Long <- as.numeric(newdata$Long)  
glimpse(newdata[16:17])

## Observations: 515,738  
## Variables: 2  
## $ Lat <dbl> 52.36058, 52.36058, 52.36058, 52.36058, 52.36058, 52.3605...  
## $ Long <dbl> 4.915968, 4.915968, 4.915968, 4.915968, 4.915968, 4.91596...

At this point, I am going to save this cleaned data into a csv file so that I can import it into Tableau or if I want to further analyze this data I can speed up the process by starting with this revised version.

write\_csv(newdata, 'CleanedHotelReviews.csv')

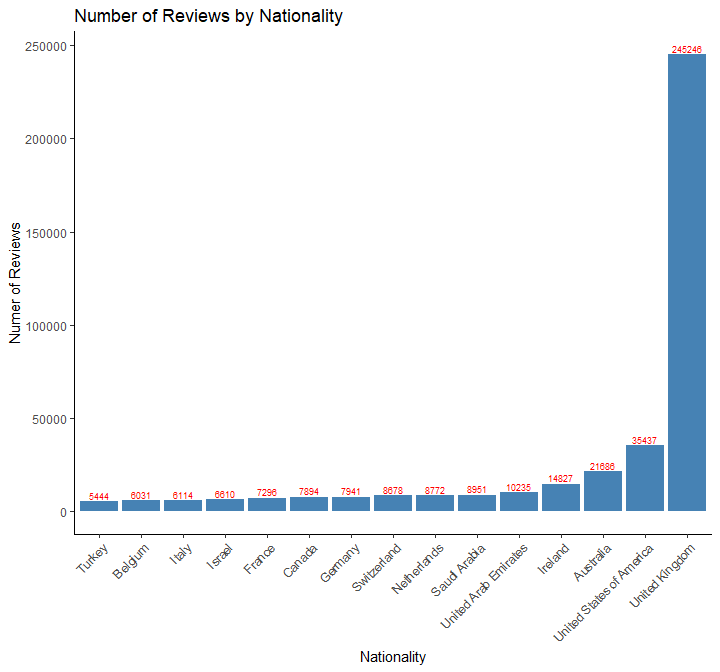
There is a lot more cleaning that can be done, but this will suffice for the purpose of this report. I did look at cleaning the Reviews. The word count is off and much can be done by looking at misspellings and trying to condense this some. I started down this track, but time is limited so I decided not to go further with this. What I did do can be seen in Section IV.

## III. Plotting the Data

### A. Nationality that reviews the most (Barplot)

There are too many nationalities to plot them all. After a quick sort and peek at the data there are only 15 nations with more than 5000 reviews. Let’s just look at those.

nationality <- arrange(filter(count(newdata, ReviewerNationality), n>5000), n)  
ggplot(nationality, aes(x=reorder(ReviewerNationality,n), y=n)) +  
 geom\_bar(stat='identity', fill='steelblue') +   
 ggtitle('Number of Reviews by Nationality') +   
 xlab('Nationality') + ylab('Numer of Reviews') +  
 theme\_classic() + theme(axis.text.x = element\_text(angle = 45, vjust=1, hjust=1)) +  
 geom\_text(aes(label=n), stat='identity', vjust=-.3, color='red', size=2.5)



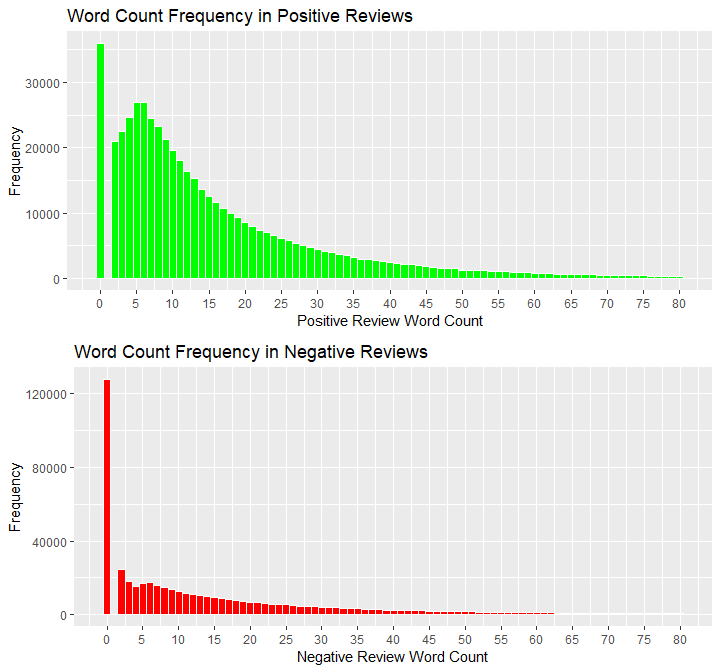
rm(nationality)

The reviewers from the United Kingdom have reviewed about 7 times as many hotels as the next country.

### B. Word count in positive reviews vs negative reviews (Histogram)

After peeking at the data, the reviews contained anywhere from 0 words to over 400. To make a more legible plot, I eliminated any reviews with more than 80 words. As you can see, the number of reviews continues to decrease as the word count increases. Plotting more than 80 words was just not meaningful.

poswords <- filter(newdata, PosReviewWordCount < 81)  
negwords <- filter(newdata, NegReviewWordCount < 81)  
p <- ggplot(poswords, aes(x=PosReviewWordCount)) +  
 geom\_histogram(fill='green', color='white', binwidth=1) +   
 ggtitle('Word Count Frequency in Positive Reviews') +   
 xlab('Positive Review Word Count') + ylab('Frequency') +  
 scale\_x\_continuous(breaks = seq(0,80,5))  
n <- ggplot(negwords, aes(x=NegReviewWordCount)) +  
 geom\_histogram(fill='red', color='white', binwidth=1) +   
 ggtitle('Word Count Frequency in Negative Reviews') +   
 xlab('Negative Review Word Count') + ylab('Frequency') +   
 scale\_x\_continuous(breaks = seq(0,80,5)) +  
 scale\_y\_continuous(breaks = seq(0,120000, 40000))  
grid.arrange(p, n, nrow = 2)



rm(n, p, poswords, negwords)

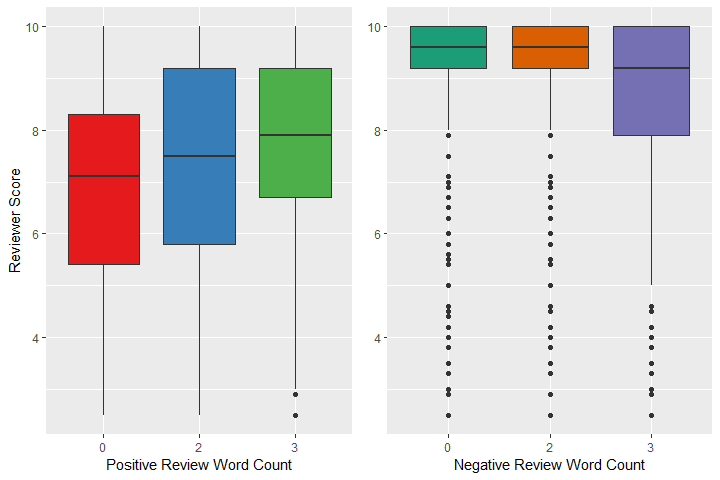
The number of zero word negative reviews is about 4 times that of 0 word positive reviews. As is shown in Section IV, 0 word reviews means ‘nothing negative’ or ‘nothing positive’. Thus people are much more likely to find nothing wrong than to find nothing right.

### C. Reviewer score base off word count (Boxplot)

#### 1. Small number of words

Since the number of words in a review is continuous, I needed to select just a few words to compare so that it was a discrete variable. To start with, I did reviews with 0, 1, 2, or 3 words. There weren’t any reviews with just 1 word. The word count wasn’t right either. It was usually off by 1 or two words. More about this can be found in Section IV.

# Separate out negative reviews with 0, 1, 2, or 3 words.  
bad0 <- filter(newdata, NegReviewWordCount == 0) # 127890 rows  
bad1 <- filter(newdata, NegReviewWordCount == 1) # 0  
bad2 <- filter(newdata, NegReviewWordCount == 2) # 24647  
bad3 <- filter(newdata, NegReviewWordCount == 3) # 18144  
# Concatenate these dataframes into one and convert WordCount to factor  
lownegwords <- rbind(bad0, bad2, bad3)  
lownegwords$NegReviewWordCount <- as.factor(lownegwords$NegReviewWordCount)  
# Do the same thing for positive reviews  
good0 <- filter(newdata, PosReviewWordCount == 0) # 35946 rows  
good1 <- filter(newdata, PosReviewWordCount == 1) # 0  
good2 <- filter(newdata, PosReviewWordCount == 2) # 20934  
good3 <- filter(newdata, PosReviewWordCount == 3) # 22533  
lowposwords <- rbind(good0, good2, good3)  
lowposwords$PosReviewWordCount <- as.factor(lowposwords$PosReviewWordCount)  
  
p <- ggplot(lowposwords,   
 aes(x=PosReviewWordCount, y=ReviewerScore, fill=PosReviewWordCount)) +  
 geom\_boxplot() +   
 scale\_fill\_brewer(palette='Set1') +  
 xlab('Positive Review Word Count') + ylab('Reviewer Score') +   
 theme(legend.position='none')  
n <- ggplot(lownegwords,   
 aes(x=NegReviewWordCount, y=ReviewerScore, fill=NegReviewWordCount)) +  
 geom\_boxplot() +   
 scale\_fill\_brewer(palette='Dark2') +  
 xlab('Negative Review Word Count') + ylab('Reviewer Score') +   
 theme(axis.title.y=element\_blank(), legend.position='none')  
grid.arrange(p, n, nrow = 1)



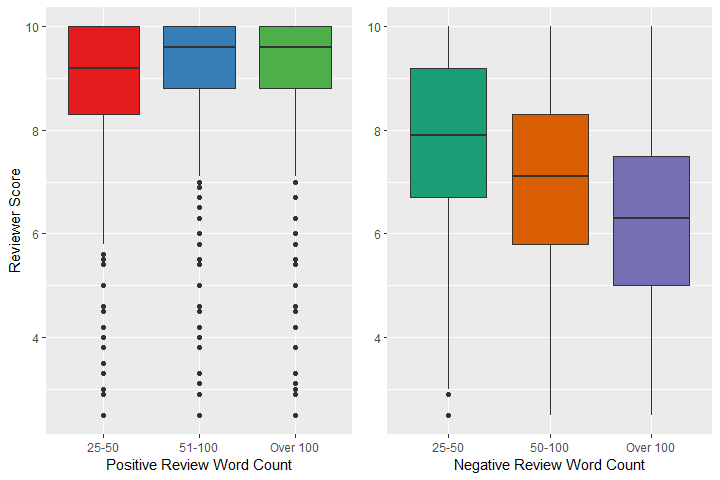
rm(list=setdiff(ls(), 'newdata'))

From this we can conclude that people use less words when there is less to say. So if they gave that hotel high rating, then they had less negative things to say. And if they said little under positive, they likely gave it a lower score. That’s pretty intuitive.

#### 2. Higher word count

How about if they used a lot of words? To test this, I’ll group them by the following: 25-50 words, 51-100 words, 101+ words.

# Separate positive and negative reviews into bins based off word count  
goodlow <- filter(newdata, PosReviewWordCount > 24 & PosReviewWordCount < 51)  
goodlow$PosReviewWordCount <- '25-50'  
goodmed <- filter(newdata, PosReviewWordCount > 50 & PosReviewWordCount < 101)  
goodmed$PosReviewWordCount <- '51-100'  
goodhigh <- filter(newdata, PosReviewWordCount > 100)  
goodhigh$PosReviewWordCount <- 'Over 100'  
badlow <- filter(newdata, NegReviewWordCount > 24 & NegReviewWordCount < 51)  
badlow$NegReviewWordCount <- '25-50'  
badmed <- filter(newdata, NegReviewWordCount > 50 & NegReviewWordCount < 101)  
badmed$NegReviewWordCount <- '50-100'  
badhigh <- filter(newdata, NegReviewWordCount > 100)  
badhigh$NegReviewWordCount <- 'Over 100'  
# Concatenate those dataframes  
poswords <- rbind(goodlow, goodmed, goodhigh)  
poswords$PosReviewWordCount <- as.factor(poswords$PosReviewWordCount)  
negwords <- rbind(badlow, badmed, badhigh)  
negwords$NegReviewWordCount <- as.factor(negwords$NegReviewWordCount)  
# Plot it  
p <- ggplot(poswords,   
 aes(x=PosReviewWordCount, y=ReviewerScore, fill=PosReviewWordCount)) +  
 geom\_boxplot() +   
 scale\_fill\_brewer(palette='Set1') +  
 xlab('Positive Review Word Count') + ylab('Reviewer Score') +   
 theme(legend.position='none')  
n <- ggplot(negwords,   
 aes(x=NegReviewWordCount, y=ReviewerScore, fill=NegReviewWordCount)) +  
 geom\_boxplot() +   
 scale\_fill\_brewer(palette='Dark2') +  
 xlab('Negative Review Word Count') + ylab('Reviewer Score') +   
 theme(axis.title.y=element\_blank(), legend.position='none')  
grid.arrange(p, n, nrow = 1)



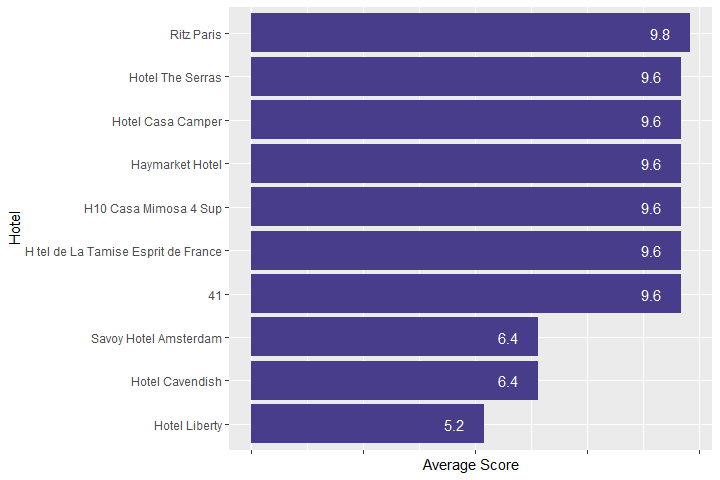
rm(list=setdiff(ls(), 'newdata'))

This reflects what we would expect. People tend to say more in whichever review reflected more of how they felt. For example, the more words a person used under negative reviews, the lower score they gave the hotel.

### D. Ratings of hotels (Barplot)

As we saw earlier, there are 1493 hotels. This is too many to view on a plot. So I took the 3 lowest and 7 highest and plotted their average scores. Their average score rating is listed on each row and is part of the dataset. I did not verify that this average is correct.

highlow <- arrange(newdata[!duplicated(newdata$Hotel), ], AvgScore) # Get one row from each hotel  
highlow <- rbind(head(highlow, 3), tail(highlow, 7)) # Grab the lowest 3 and highest 7  
ggplot(highlow, aes(x=reorder(Hotel, AvgScore), y=AvgScore)) +   
 geom\_bar(stat='identity', fill='darkslateblue') +  
 xlab('Hotel') + ylab('Average Score') +  
 theme(axis.text.x=element\_blank()) +  
 geom\_text(aes(label=AvgScore), stat='identity', hjust=2, color='white', size=4) +   
 coord\_flip()



rm(highlow)

## IV. Abandoned Cleaning

### A. Filter by Word Count

This is being included because it is the direction I would have taken this if more time were available. I would like to clean this data up much more. Time is limited though. Here is what I had started though. I did not change the full data set with this. I made a copy to play with.

As we saw in section III.C.1, there are no 1 word reviews. Actually there are plenty of them, but the ReviewWordCount is not quite accurate. I need to get a group of negative reviews that contain 0, 2, or 3 words.

neg0 <- filter(newdata, NegReviewWordCount == 0)   
neg2 <- filter(newdata, NegReviewWordCount == 2)   
neg3 <- filter(newdata, NegReviewWordCount == 3)  
writeLines(paste('Number of negative reviews that contain 0 words:', nrow(neg0), '\nNumber of negative reviews that contain 2 words:', nrow(neg2), '\nNumber of negative reviews that contain 3 words:', nrow(neg3)))

## Number of negative reviews that contain 0 words: 127890   
## Number of negative reviews that contain 2 words: 24647   
## Number of negative reviews that contain 3 words: 18144

Let’s look at how many distinct reviews I have for each word count.

writeLines(paste('Unique entries for reviews with 0 words: ', nrow(distinct(neg0, NegReview)), '\nUnique entries for reviews with 2 words: ', nrow(distinct(neg2, NegReview)), '\nUnique entries for reviews with 3 words:', nrow(distinct(neg3, NegReview))))

## Unique entries for reviews with 0 words: 1   
## Unique entries for reviews with 2 words: 919   
## Unique entries for reviews with 3 words: 3992

But the word count isn’t accurate:

cat('First 3 reviews with 0 words:', neg0$NegReview[1], ',', neg0$NegReview[2], ',', neg0$NegReview[3],'\n')

## First 3 reviews with 0 words: No Negative , No Negative , No Negative

cat('First 3 reviews with 2 words:', neg2$NegReview[1], ',', neg2$NegReview[2], ',', neg2$NegReview[3],'\n')

## First 3 reviews with 2 words: Nothing , Construction , Nothing

cat('First 3 reviews with 3 words:', neg3$NegReview[1], ',', neg3$NegReview[2], ',', neg3$NegReview[3],'\n')

## First 3 reviews with 3 words: Nothing , n a , Loud aircondition

What is interesting is that the 0 word reviews actually contain the words ‘No Negative’. The 2 word reviews may only contain one word and the 3 word reviews sometimes contain 1 or 2 words. So the word count is off to begin with. Additionally, there are plenty of reviews that only have one word that is “nothing”, “na” or the equivalent. These could all be manipulated so that they read “nothing”. I suspect that in the review form, there was a box that could be checked that indicated that there was “No Negative”. These got marked by most people and was assigned a 0 word review. This was done over 120,000 times, as seen in the plot from section III.B. Others, though, missed that box and decided to put the word “nothing” in the review section. These have been treated differently, but the intent is the same.

writeLines(paste('No Negative was indicated', nrow(neg0), 'times out of', nrow(newdata), 'total reviews.'))

## No Negative was indicated 127890 times out of 515738 total reviews.

cat('That equates to', round(100\*nrow(neg0)/nrow(newdata), 2), '% of the reviews.')

## That equates to 24.8 % of the reviews.

### B. Correct Reviews and Word Count

How many more actually meant that there was nothing negative, but it was scored otherwise? I will check the 2 and 3 word reviews to see what I can find that is equivalent to No Negative. I need to remove the words ‘No Negative’ from the 0 word reviews so that there truly are no words in the review.

neg0$NegReview <- ''

Next I combine all the 0, 2, and 3 word reviews together and do an actual word count to replace what was provided.

negs <- rbind(neg0, neg2, neg3)  
negs$NegReviewWordCount <- sapply(strsplit(negs$NegReview, " "), length)  
rm(neg2, neg3)

After glancing throught the data, I see many misspellings and different use of capital letters. I need to convert all to lower case and arrange in alpha order. I wrote this list as a text file.

negs$NegReview <- str\_to\_lower(negs$NegReview)  
negwords <- negs %>% arrange(NegReview) %>% distinct(NegReview)  
write(negwords$NegReview, 'negreviews.txt')  
rm(negwords)

Opening the text file shows 3848 unique reviews. Many are non-sensical and consist of a single digit or letter. In order to save time, I quickly went down the file looking for things that indicated that nothing was wrong. This included “all positive”, “amazing location”, “cannot complain”, “excellent everything”, etc, as well as assumed misspellings of “nothing”, such as “bothing”, jothing“, etc. I very quickly located 338 words that should most likely be converted into No Negative. These were saved into a file named “negreviews-to-nonegative.txt”. This file will be used to convert all of these to 0 word reviews and see how the total No Negative amount changes.

changenegs <- read\_delim('negreviews-to-nonegative.txt', delim='\n', col\_names='ToChange')  
changenegs <- mutate(changenegs, Replacement='') # Add replacement values  
negs <- as.data.frame(negs)  
negs <- FindReplace(negs, 'NegReview', changenegs, from='ToChange', to='Replacement', exact=TRUE, vector=FALSE)  
rm(changenegs)

Now to change the NegReviewWordCount to accurately reflect the 0 word word count

negs$NegReview <- trimws(negs$NegReview, which='both') # Remove leading/trailing whitespace  
negs <- mutate(negs, NegReviewWordCount=str\_count(negs$NegReview, "\\w+"))

There were originally 127,890 reviews with ‘No Negative’. After this adjustment, how many are there now?

newnonnegs <- nrow(filter(negs, NegReviewWordCount==0))  
cat("There are now", newnonnegs, "reviews with 'No Negative'.\n")

## There are now 155772 reviews with 'No Negative'.

cat('This is an increase of', round(100\*((newnonnegs-nrow(neg0))/nrow(neg0)), 2), '%')

## This is an increase of 21.8 %

rm(list=ls())

I wish I had more time to spend on this as there are many words that can be searched and used to find out the major complaints with the hotel, like”pricey“,”expensive“, etc. Also, looking for qualifiers like”very" or “too” would give more information. I could also do a word cloud that would reflect which words or tags are most associated with a rating less than 7. Those would be interesting and I may look into them in the future.