REDS Midtown Tavern Online Customer Review Analytics

Foundations of Data Science: Capstone Project Report

Antoine Beauchamp
April 6th, 2017

Contents

1 Introduction

The past decade has been met with an explosion in the volume of data that we humans produce every day by way of our technological activity. An important consequence of this ever-growing data output is that we are increasingly becoming saturated with information regarding any number of topics in a wide range of areas of human activity, from personalized healthcare to federal policy to corporate finance. The sheer volume of data available makes the process of deriving insight seem difficult and inaccessible. However with the right tools and the right approach, it is possible to uncover a wealth of important information about various aspects of human and business activity. By manipulating and analyzing the available data, we can uncover and solidify insights that previously may only have been hinted at on an intuitive level. The data, when analyzed properly, provides a solid foundation for making decisions and recommendations based on concrete evidence. In the context of business, this is an extremely useful advantage to leverage in order to promote growth and to allocate resources properly. One especially important source of data is the World Wide Web. The Internet provides a massive agglomeration of human opinion and behaviour from all areas of life. From a business perspective, the clientele's online behaviour often provides abundant information regarding patterns that may be influential to business activity. For businesses that operate within the service and hospitality industries, important data pertaining to customer sentiment is abundantly available on popular review and travel websites.

In this report, I will demonstrate how we can use a variety of tools available in R to acquire and mine the data from such websites in order to uncover insights and make informed business recommendations. In particular, I will focus on gathering and analyzing the online customer review data for REDS Midtown Tavern (RMT). RMT is an upscale restaurant located in Toronto, Ontario, that offers a large selection of beverages with a focus on an extensive gin bar and a gastropub-style menu. RMT is a subsidiary of SIR Corp and has been in business since 2013. Using the online customer review data for RMT, I endeavour to build a data-founded understanding of the restaurant's performance over time, as well as the types of sentiments that are commonly associated with various aspects of the customer experience. Specifically:

- How has RMT performed over the years? What periods of time have been associated with positive, negative reviews? How has the restaurant's performance changed over time? How is it performing now?
- Based on the customer feedback, what are the restaurant's strengths? What are its weaknesses?
- What are the most important aspects of a customer's experience? How can we improve upon these?

In addition to providing data-driven answers to these questions, I will make recommendations for potential improvements to the restaurant.

This report is divided into five primary sections. In Section 2, I will elaborate on the methods and tools I used to acquire the customer review data for RMT using popular review and travel websites. Once the data is collected and stored locally, I will discuss the process of transforming this raw data into a form suitable for analysis. This is done in Section 3. In Section 4, I will dive into the analysis of this customer review data in order to

mine for actionable insights. Finally, Section 5 will be dedicated to summarising the main points of the report and analysis, providing data-driven recommendations, and outlining potential areas for future work. Sections 2 and 3 will be more technical in nature, while Sections 4 and Section 5 will be focused primarily on analysis. For complete access to the programming scripts used to complete this project, as well as additional content, please refer to my GitHub repository.

2 Data Acquisition

As mentioned in the Introduction, the aim of this project is to perform an analysis of online customer review data for RMT. For the purpose of this investigation, I have focussed my efforts on the following review and travel websites: Yelp, OpenTable, TripAdvisor, and Zomato. These review websites are similar in that they provide a 5-star rating scale. The first step in being able to analyze the data from these websites is to gather the data and store it locally. Though this data is readily available to see and read online, we would like to acquire it in a more structured way. This can be done relatively simply using a variety of tools available online. More specifically, I gathered the review data from the aforementioned websites by using the `rvest` package in R and the web scraping tool, SelectorGadget. I will elaborate on the use of these tools below, beginning with the latter.

2.1 SelectorGadget and CSS Selectors

2.1.1 Identifying Content with SelectorGadget

Acquiring data from the web means being able to identify exactly what content needs to be extracted. In practice, this is accomplished with the use of **CSS selectors**. CSS selectors are what allows us to gather data from the web by using packages such as `rvest`. A more detailed discussion of CSS selectors is given in Section 2.1.2. The difficulty in using CSS selectors arises when identifying which selectors are associated with what content on a webpage. Traditionally, one might attempt to identify a selector for specific content by scouring a website's HTML source code. Though it is possible, this isn't exactly effortless. Fortunately, an alternative way of identifying the appropriate CSS selector is to use a tool, such as SelectorGadget. SelectorGadget is an open source tool, developed by Andrew Cantino and Kyle Maxwell, that allows us to identify CSS selectors from a website by **clicking** on select items of interest. SelectorGadget greatly facilitates the process of identifying CSS selectors by providing a point-and-click interface between the webpage content and the underlying selector. In order to better understand how to use SelectorGadget to obtain CSS selectors from a website, I will provide a relevant example.

The way in which we will gather the customer review data to be used in our analysis is to **scrape** the websites for the customer reviews. In this example, I will work with the first page of reviews for RMT on Yelp. Here is the relevant URL: https://www.yelp.ca/biz/reds-midtown-tavern-toronto-2. The reviews are located not too far down the page, as displayed in Figure 1.

The SelectorGadget interface manifests itself on the webpage as a grey search bar at the bottom of the page. To begin identifying the CSS selector that corresponds to the customer review content, we can select the first customer review by clicking on it. The

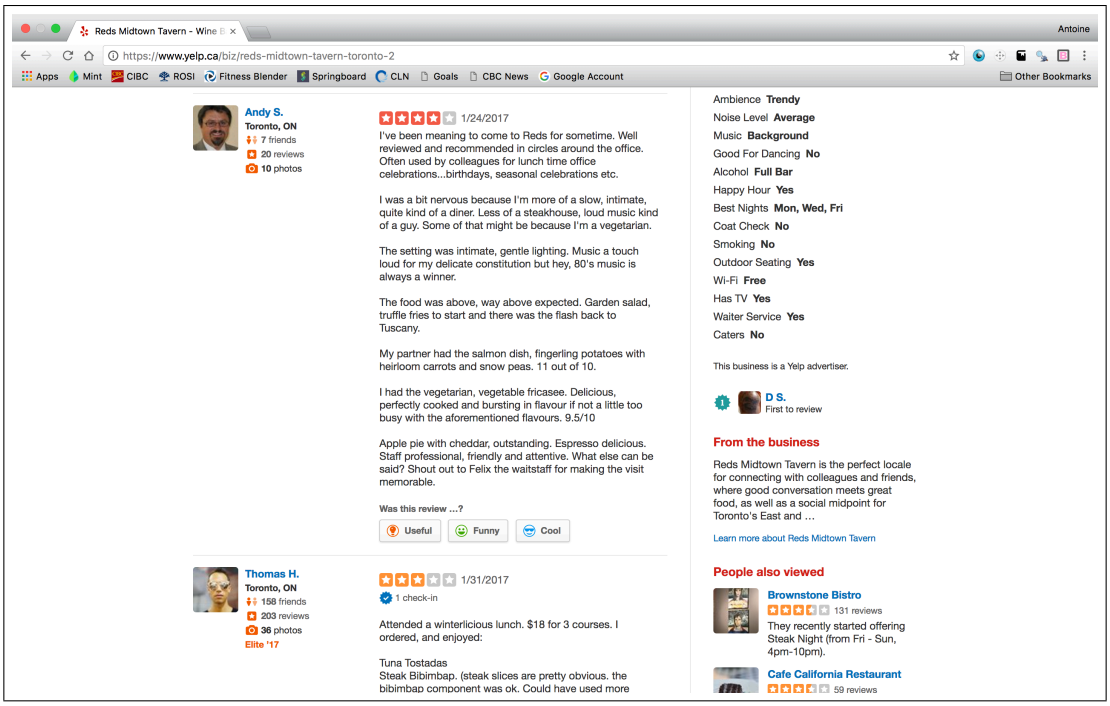

Figure 1: Customer reviews for RMT on Yelp

result of this action is shown in Figure 2. We see that the SelectorGadget has returned the CSS selector "`p`" in the search bar. The content that we have selected via point-and-click has been highlighted in green, while the content that matches that selector has been highlighted in yellow. This selector includes the customer reviews, which is desired, but it also appears to include additional content, such as the text "Was this review ...?" and additional text under the heading "From the business". We don't need these elements of the webpage. The selector "`p`" is returning too broad a selection. We can now choose to add additional content to our filter by clicking on content that is not yet highlighted, or we can remove content from our filter by clicking on content that has been highlighted. In this case, we want to click on the text below "From the business" to remove it from our selection. In doing so, we observe that content that had previously been highlighted but is now de-selected becomes red. This is shown in Figure 3. We also notice that the text "Was this review...?" is no longer included in our selection, so that only the customer reviews remain. It appears that the CSS selector "`.review-content p`" is the correct selector associated with the review content on this webpage. This can verified by noting that SelectorGadget has found 20 items matching this selector (identified in parentheses on the SelectorGadget tool), which corresponds to the number of reviews on this particular page. Having identified the correct CSS selector for the customer review content, we can import the data into R. This process will be described in Section 2.2.

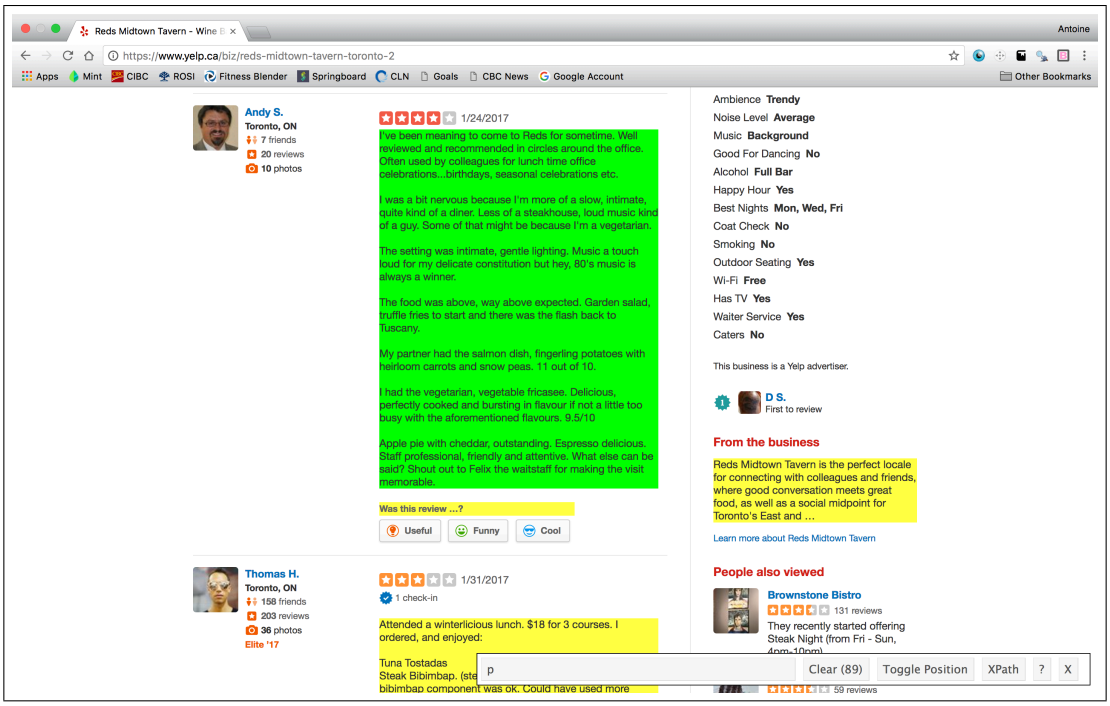

Figure 2: First attempt at selecting Yelp customer reviews with SelectorGadget

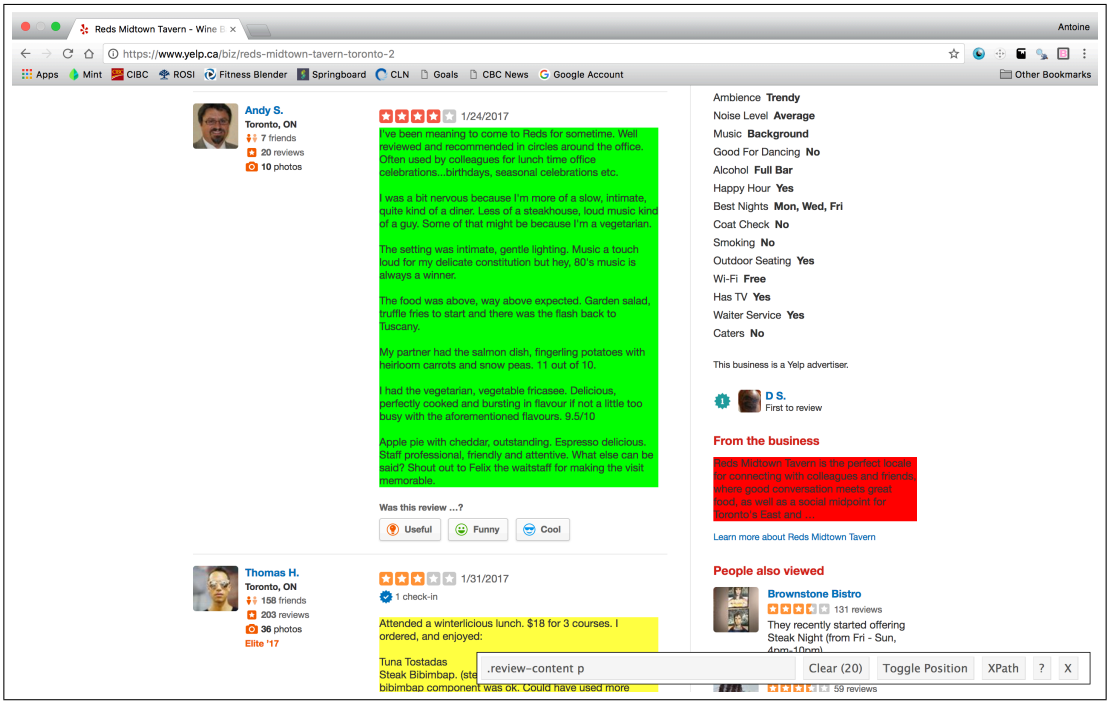

Figure 3: Correct use of SelectorGadget on Yelp

2.1.2 CSS Selectors in Detail

When working with SelectorGadget, it is helpful to have some knowledge of CSS selectors. In essence, CSS selectors allow us to identify given "blocks" of content within an HTML source file. An excellent hands-on tutorial regarding CSS selectors is found at the following website: https://flukeout.github.io/. This section will loosely follow the layout of this tutorial.

An HTML file is comprised of different blocks of content, which are identified by their **type**. E.g., in HTML, `<p>` represents a paragraph block, `<a>` a hyperlink, `<h1>` a heading, and so on. A screenshot of the HTML source file for https://www.yelp.ca/biz/reds-midtown-tavern-toronto-2 is included in Figure 4. To learn more about HTML formatting, visit https://www.w3schools.com/html/default.asp.

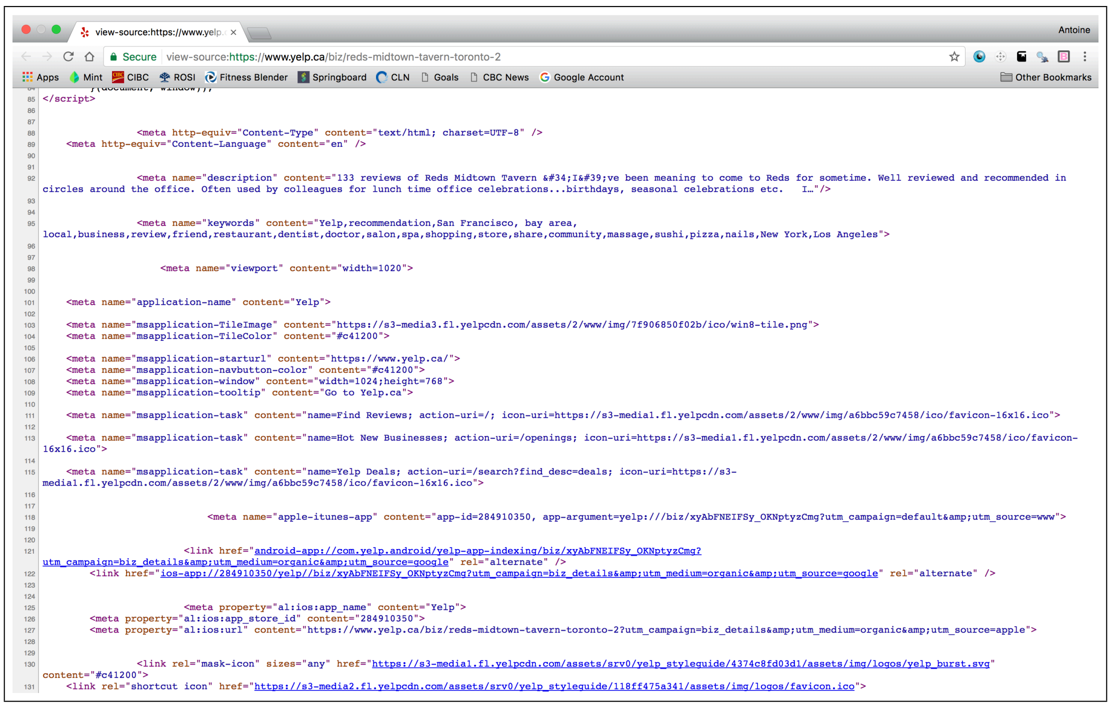

Figure 4: HTML source for REDS Midtown Tavern on Yelp

The most basic CSS selector is simply a selector of type. To obtain all content that is contained within blocks of a certain type, simply use the indicator for that type. E.g., to obtain content that is contained within a paragraph block, use the selector `p`. This is what happened when we used SelectorGadget in Figure 2: the customer review content was contained within a parapraph block, but so was other content that we didn't need. Using the selector `p` identified all of this content.

In order to diversify content, HTML types can be associated with an **ID** or with a **class**. E.g., `<div id="title">` or `<div class="section">`. These are both `<div>` blocks, but one has an ID called "title" and the other has a class called "section". In addition to basic type selection, content can be extracted from a webpage by selecting it based on ID or class. The CSS selector for an ID is a hashtag, `#`. E.g. if there exists a block defined by `<div id="title">` in the HTML file, we can select that element by using the selector

`#title`. This will select all content with `id="title"`, including but not limited to the content that we want. Similarly, the selector for a class is a period, `.`. E.g. we can select a block such as `<div class="content">` using `.content`.

Now that we have these basic selectors, we can combine them to tailor our selection process and access more specific content. One way of doing this is to use the **descendant selector**. This allows us to select content that is nested within another content block. The descendant selector is expressed by writing two selectors separated by a space. E.g. we could select something like `<div id="title"><p> Text </p></div>` with the selector `div p`. This selects all paragraph content within a `<div>` block. We can also use the descendant selector in conjunction with the ID or class selectors. E.g. `#title p` will select all paragraph content contained within a block of any type with `id="title"`. In Section 2.1.1, we used the selector `.review-content p` to obtain the customer review data from Yelp. With our new understanding of CSS selectors, we see that this is comprised of a descendant selector with the class and type selectors. We are selecting the paragraph blocks, `<p>`, within blocks of content with `class="review-content"`.

Finally, specific class content can be selected by combining the class selector with the type selector, in the following way: `type.class`. E.g. if there exists both `<div class="section">` and `<span class="section">` within the HTML file, we can select the `<div>` block specifically using `div.section`, rather than just `.section`, which would additionally select the `<span>` block.

This covers the basics of CSS selectors and should provide some context for what SelectorGadget is returning when clicking on webpage content. For a more detailed look at CSS selectors, follow the complete tutorial at https://flukeout.github.io/.

2.2 Importing Web Data with `rvest`

In the previous Section, I discussed how to work with SelectorGadget to identify content that we want to acquire from webpages. Once the correct CSS selectors are obtained for the content that we want to extract, the next step is to load this data into R for cleaning and processing. I performed this step of the data gathering process using Hadley Wickham's web data R package, `rvest`. `rvest` allows us to gather data from the web using a few basic functions in R: `read_html()`, `html_nodes()`, `html_text()`, and `html_attrs()`.

Consider once again the problem of scraping reviews from Yelp. In Section 2.1.1, we identified that the correct CSS selector for this content was `.review-content p`. This is a good start, but what do we do with this? This is where `rvest` comes in. Let's start by loading the package. If it isn't already installed, install it using `install.packages()`.

```
library(rvest)
```

The first step in working with `rvest` is to provide R with the URL for the website that we want to scrape. This is done using the `read_html()` function.

```
YelpURL <- "https://www.yelp.ca/biz/reds-midtown-tavern-toronto-2"
YelpURL_data <-read_html(YelpURL)
```

This saves the information about the URL to our console. Next, we extract the content that we want using the CSS selector that we identified with SelectorGadget. Here we use `html_nodes()` with the arguments being the HTML documents and the CSS selector.

```
YelpReviews <- html_nodes(YelpURL_data, ".review-content p")
head(YelpReviews)
```

```
## {xml_nodeset (6)}
## [1] <p lang="en">The first time I started dining here was due to summerl ...
## [2] <p lang="en">Attended a winterlicious lunch. $18 for 3 courses. I or ...
## [3] <p lang="en">I've been meaning to come to Reds for sometime. Well re ...
## [4] <p lang="en">Unbelievably bad service. I think everyone course had a ...
## [5] <p lang="en">As a young barrister  I often visited Red's older, fanc ...
## [6] <p lang="en">I came here for dinner and had wine and entrees. I wish ...
```

We now have the customer reviews from Yelp stored in an object of class `xml_nodeset`. This isn't the easiest way to work with the data so the next step is to convert this to `character` class. The first way to do this is simply to use the `as.character()` function.

```
YelpReviews_char1 <- as.character(YelpReviews)
strtrim(head(YelpReviews_char1, n=2),65)
```

```
## [1] "<p lang=\"en\">The first time I started dining here was due to summ"
## [2] "<p lang=\"en\">Attended a winterlicious lunch. $18 for 3 courses. I"
```

```
class(YelpReviews_char1)
```

```
## [1] "character"
```

This converts all of the data to `character` class, including the HTML formatting tags. This can be useful if the formatting is needed. If we are solely interested in the text stored within the paragraph block, however, we can use `html_text()` to extract this data directly.

```
YelpReviews_char2 <- html_text(YelpReviews)
strtrim(head(YelpReviews_char2, n=2),65)
```

```
## [1] "The first time I started dining here was due to summerlicious. My"
## [2] "Attended a winterlicious lunch. $18 for 3 courses. I ordered, and"
```

In this case we have the text in character format without the HTML tags associated with the content. Note that `html_text()` only works if there is text to extract, evidently.

Other useful functions to extract data from these HTML nodes are the `html_attrs()` and `html_attr()` functions. Looking at the output of `YelpReviews` above, we see that the HTML tag is `<p lang="en">`. Thus the paragraph blocks are associated with an **attribute** called `lang`, with a value of "en". We can pull all the attributes from our HTML data using `html_attrs()`.

```r
head(html_attrs(YelpReviews), n=2)
```

```
## [[1]]
## lang
## "en"
##
## [[2]]
## lang
## "en"
```

```r
class(html_attrs(YelpReviews))
```

```
## [1] "list"
```

The output is a list containing the attribute (`lang`) and its value (`"en"`). We can further extract the value of a specific attribute using `html_attr()`.

```r
head(html_attr(YelpReviews, "lang"))
```

```
## [1] "en" "en" "en" "en" "en" "en"
```

```r
class(html_attr(YelpReviews, "lang"))
```

```
## [1] "character"
```

This is particularly useful when information that we want is stored within such attributes, rather than within the main content block. One such example occurs when extracting the numerical ratings that the customers have given to the restaurant. On Yelp this is presented as a number of stars filled in with the colour orange. Let's extract this data and see what it looks like. The correct CSS selector, obtained via SelectorGadget, is `.rating-large`.

```r
YelpRatings <- html_nodes(YelpURL_data, ".rating-large")
as.character(YelpRatings)[1] %>% str_break()
```

```
## [1] "<div class=\"i-stars i-stars--regular-4 rating-large\" title=\""
## [2] "4.0 star rating\">\n        <img class=\"offscreen\" height=\"303"
## [3] "\" src=\"https://s3-media1.fl.yelpcdn.com/assets/srv0/yelp_des"
## [4] "ign_web/41341496d9db/assets/img/stars/stars.png\" width=\"84\" "
## [5] "alt=\"4.0 star rating\">\n</div>"
```

This output has been broken into multiple lines to fit the page. We see that the value that we want, namely "4.0 star rating", is actually contained within the `"title"` attribute of the `<div>` block. It can also be obtained from the `"alt"` attribute of the `<img>` block. Let's see what `html_attrs()` gives us.

```
head(html_attrs(YelpRatings), n=2)
```

```
## [[1]]
##                                       class
## "i-stars i-stars--regular-4 rating-large"
##                                       title
##                           "4.0 star rating"
##
## [[2]]
##                                       class
## "i-stars i-stars--regular-3 rating-large"
##                                       title
##                           "3.0 star rating"
```

These are only the attributes of the `<div>` block. The reason for this is that our selector, `.rating-large`, only selects this block. To select the nested `<img>` block specifically, we would use the descendant selector `.rating-large img`. From the `div` block we can extract the numeric rating by using the `title` attribute.

```
YelpRatings_clean <- html_attr(YelpRatings, "title")
head(YelpRatings_clean)
```

```
## [1] "4.0 star rating" "3.0 star rating" "4.0 star rating" "2.0 star rating"
## [5] "4.0 star rating" "3.0 star rating"
```

The next step here would be to use regular expressions to isolate the numerical value and then convert the data to `numeric` class for quantitative analysis. Now that we have a foundation in how to gather web data using SelectorGadget and `rvest`, let's acquire the full data set of online customer reviews for RMT.

2.3 Gathering Online Customer Review Data

With a basic understanding of CSS selectors, SelectorGadget, and `rvest`, we are equipped to gather our customer review data for RMT. As mentioned previously, the following websites will be used as data sources: Yelp, OpenTable, TripAdvisor, and Zomato. Specifically, the data set will be comprised of the written customer reviews, the numerical customer ratings, and the review dates. Note that the code presented in this section will not be executable due to the longer computation times. Let's start by loading the necessary R packages.

```
#Load required libraries
library(rvest)
library(dplyr)
library(tidyr)
library(readr)
```

I have coded the data gathering process into a number of functions, one for each website, named `YelpScrape()`, `OpenTableScrape()`, `TripAdScrape()`, and `ZomatoScrape()`. The algorithm for each function is straightforward:

1. Read the HTML document using the correct URL
2. Gather review, rating and date data from the webpage using CSS Selectors. Additional data may be gathered as needed.
3. Append new data to vectors for each of the variables
4. Check to see if we have reached the end of the reviews
5. Increment counter and identify URL for next page of reviews

An important part of this process was identifying the structure of the URLs of the various review pages in order to go through them automatically. It was also important to see what data was available from each of the webpages and how to access that data. In the following subsection I will discuss some of the sticking points that arose when scraping the various websites for data. Only the code for the `YelpScrape()` function will be provided in full, since the scraping functions are similar in structure. The full code can found in the .Rmd file associated with this document, or in the file DataGathering.R.

2.3.1 Web Scraping Concerns

The code for the `YelpScrape()` function is as follows.

```
## Function: YelpScrape ##

# This function is used to scrape review data from Yelp.com
# Arguments:
# BaseURL: URL to the first page of reviews
YelpScrape <- function(BaseURL) {

  #Review counter. Yelp = 20 per page
  ReviewCount <- 0
  #Vector intialization
  Reviews <- character(0)
  Ratings <- character(0)
  Dates <- character(0)
  PrevRev <- character(0)
  flag <- 1

  #Iterate over review pages and scrape data
  while(flag==1){

    #URL for the given review page
    page_url <- paste(BaseURL,"?start=",as.character(ReviewCount),sep="")

    #Scrape reviews, ratings, dates from current URL
    ReviewsNew <- read_html(page_url) %>%
```

```
      html_nodes(".review-content p") %>%
      html_text
    RatingsNew <- read_html(page_url) %>%
      html_nodes(".rating-large") %>%
      html_attr("title")
    DatesNew <- read_html(page_url) %>%
      html_nodes(".biz-rating-large .rating-qualifier") %>%
      html_text()
    #Additional data identifying previous/updated reviews
    PrevRevNew <- read_html(page_url) %>%
      html_nodes(".biz-rating-large .rating-qualifier") %>%
      as.character()

    #Print iteration count identifier to std.out
    print(paste("Scraping Yelp page",ceiling(ReviewCount/20)))

    #Append new data to existing vectors
    Reviews <- c(Reviews,ReviewsNew)
    Ratings <- c(Ratings,RatingsNew)
    Dates <- c(Dates, DatesNew)
    PrevRev <- c(PrevRev,PrevRevNew)

    #Increment counter
    ReviewCount=ReviewCount +length(ReviewsNew)

    #Loop ending condition
    flag <- if(length(ReviewsNew)==0){0} else {1}
  }
 return(list("Reviews"=Reviews,
             "Ratings"=Ratings,
             "Dates"=Dates,
             "PrevRev"=PrevRev))
}
```

The primary issue that arose when scraping Yelp was that there was a discrepancy in the number of reviews and the number of ratings. This occurs because the ratings data includes data from **previous reviews** that have since been updated, as well as the corresponding updated reviews. The text review data only picks up the new reviews. The variable `PrevRev` contains information about whether a review is considered a "previous review" or not. This will be used later on to identify which reviews are previous reviews.

The web scraping process for OpenTable is more straightforward than Yelp, though there are some hiccoughs in the data that we will clean in Section 2.3.2.

When scraping reviews from TripAdvisor, a difficulty arises in that the URL for each of the review pages does not appear to have an obvious pattern. Fortunately, the URL for the subsequent review page is contained within the HTML source of the current review page, as part of the link to the next page. Therefore to get the URL for the next page, I

had to identify the selector for the link to the following page, and extract the data.

Another feature that I've implemented in `TripAdScrape()` is that the web scraping process for TripAdvisor actually begins at the landing page for the restaurant, rather than at the first full review page. The scraping function then jumps from this landing page to the first full review page by "clicking" on the title of the first available review. The reason for this is that, when attempting to go straight to the first full review page, I recognized that new reviews were not being included in this "first" page. The review page was effectively dated to when I had first obtained the URL. Jumping from the landing page for to the first full review page bypasses this problem, since new reviews are included on the landing page.

In the `TripAdScrape()` function I have extracted two sets of data for the dates, stored in `Dates1` and `Dates2`. The reason for this is that, for recent reviews, TripAdvisor presents its date information in the following form: "Dined ## days ago" or "Dined yesterday". This is not useful for analysis. However, the **actual** dates for these recent reviews can be obtained using the selector `.relativeDate`. The catch is that this selector does not select the dates for those older reviews that are not expressed in the forms "Dined ## days ago" and so on. This format is only used to express the most recent dates. Older reviews are associated with a proper date format. Consequently, we need a combination of the information gathered by both the selectors `.relativeDate` and `.ratingDate`.

The final website used to extract data is Zomato. The primary problem I ran into with this website was that the reviews are not written onto different URL pages. Rather, the reviews are accessed via a sort of drop down menu on the landing page. Due to this, and in the interest of time, I wasn't able to extract the full set of reviews available on Zomato. One thing to note is that longer reviews on Zomato will be truncated and have an associated "read more" option to show the full review. Reviews of this type are counted twice by SelectorGadget: once for the truncated version, and once for the full expanded version. Additionally, the format in which the ratings have been extracted is such that there will be double the amount of data, half of which consists of `NA` values. We will address these issues in the data cleaning stage.

2.3.2 Web Scraping Executed

With the web scraping functions defined and the idiosyncrasies of the websites accounted for, we can execute the data acquisition process. The primary thing to do is to identify the URLs for the different review pages and pass them to the scraping functions.

```
#Yelp main review page URL
BaseURL_Yelp <-
  "https://www.yelp.ca/biz/reds-midtown-tavern-toronto-2"
#OpenTable main review page URL
BaseURL_OpenTable <-
  paste("https://www.opentable.com/reds-midtown-tavern?",
        "covers=2&dateTime=2017-02-22+19%3A00%23reviews&page=")
#TripAdvisor landing page
LandingURL_TripAd <-
  paste("https://www.tripadvisor.ca/Restaurant_Review-g155019-d5058760",
```

```
        "-Reviews-Reds_Midtown_Tavern-Toronto_Ontario.html")
#Zomato main review page URL
BaseURL_Zomato <-
  paste("https://www.zomato.com/toronto/",
        "reds-midtown-tavern-church-and-wellesley/reviews")

#Scrape data from websites
YelpData <- YelpScrape(BaseURL_Yelp)
OpenTableData <- OpenTableScrape(BaseURL_OpenTable)
TripAdData <- TripAdScrape(LandingURL_TripAd)
ZomatoData <- ZomatoScrape(BaseURL_Zomato)
```

Before saving the raw data to file, we need to do some basic cleaning of the OpenTable date data. The reason for this is that some of the dates are expressed in "Dined ## days ago" format. The proper date data is not available, so we have to create it by subtracting the number of days from the current date. This only works if it is done on the same day that the web data was acquired.

```
#Find all instances of dates in the form "Dined ## days ago"
DatesLogic <- grepl("[0-9]+.*ago",OpenTableData$Dates)

#Subset date info to get the instances matching the above format
DatesTemp <- OpenTableData$Dates[DatesLogic]

#Create empty character vector of proper length
DineDate <- character(length(DatesTemp))

#Extract number of days ago that reviews were posted.
dineDay <-
  regmatches(DatesTemp,regexpr("[0-9]+",DatesTemp)) %>%
  as.numeric()

#Today's date
todayDate <- Sys.Date()

#Subtract number of days from today's date
DineDate <- todayDate - dineDay

#Replace date entries with the proper dates
OpenTableData$Dates[DatesLogic] <- DineDate
```

Finally, having completed the data gathering stage, we can save our raw data to file.

```
#Save raw data to file
save(YelpData,OpenTableData,TripAdData,ZomatoData,
     file="./Data/RMTRawData.RData")
```

3 Data Cleaning

In the previous section, we developed a number of functions to scrape the customer review data for RMT from a number of popular review and travel websites. With the raw data in hand, the next step of the process is to clean the data. There are a number of things that need to be done:

- Remove unnecessary "previous review" data from Yelp data set
- Remove the unnecessary `NA` values from Zomato ratings data
- Remove the duplicate truncated reviews from Zomato
- Consolidate `Dates1` and `Dates2` variables for TripAdvisor
- Create vectors that describe which websites the data belongs to
- Merge all online review data into a data frame
- Clean up all dates and convert to `date` class
- Clean up all ratings and convert to `numeric` class
- Clean up reviews as needed

We will begin by addressing the first six bullet points in a preliminary cleaning stage.

3.1 Preliminary Cleaning

Since the code written in Section 2.3 is not executable, we will load the raw data from the "CapstoneRawData_032017.RData" file.

```
load("./Data/CapstoneRawData_032017.RData")
# Examine Yelp data
str(YelpData, width = 70, strict.width = "cut")
```

```
## List of 4
##  $ Reviews: chr [1:136] "The first time I started dining here was d"..
##  $ Ratings: chr [1:138] "4.0 star rating" "3.0 star rating" "4.0 st"..
##  $ Dates  : chr [1:138] "\n        3/19/2017\n    " "\n        1/31"..
##  $ PrevRev: chr [1:138] "<span class=\"rating-qualifier\">\n       "..
```

```
# Examine OpenTable data
str(OpenTableData, width = 70, strict.width = "cut")
```

```
## List of 3
##  $ Reviews: chr [1:309] "Bibimbap needs more veg and more sauce. Wo"..
##  $ Ratings: chr [1:309] "3" "5" "5" "3" ...
##  $ Dates  : chr [1:309] "17242" "Dined on March 9, 2017" "Dined on "..
```

```
# Examine TripAdvisor data
str(TripAdData, width = 70, strict.width = "cut")
```

```
## List of 4
##  $ Reviews: chr [1:222] "\nGreat place to meet after work for drink"..
##  $ Ratings: chr [1:222] "4 of 5 bubbles" "2 of 5 bubbles" "4 of 5 b"..
##  $ Dates1 : chr [1:6] "22 March 2017" "21 March 2017" "5 March 2017"..
##  $ Dates2 : chr [1:222] "Reviewed yesterday\nNEW " "Reviewed 2 days"..
```

```
# Examine Zomato data
str(ZomatoData, width = 70, strict.width = "cut")
```

```
## List of 3
##  $ Reviews: chr [1:11] "\n                Rated \n                 "..
##  $ Ratings: chr [1:22] "Rated 3.0" NA "Rated 5.0" NA ...
##  $ Dates  : chr [1:10] "2016-08-01 16:22:56" "2016-04-20 05:25:03" "..
```

Beginning with the Yelp data, we will remove "previous reviews", as mentioned in Section 2.3.1. We will also generate a vector that indicates that this is data from Yelp. Finally, we will merge these variables into a data frame specific to Yelp.

```
#Which reviews are not previous reviews?
NoPrevRev <- !grepl("has-previous-review",YelpData$PrevRev)
YelpData$Ratings <- YelpData$Ratings[NoPrevRev]
YelpData$Dates <- YelpData$Dates[NoPrevRev]

#Vector, Yelp identifier
YelpVec <- rep("Yelp",length(YelpData$Reviews))

#Combine to data frame
YelpDF <- data_frame(Reviews=YelpData$Reviews,
                     Ratings=YelpData$Ratings,
                     Dates=YelpData$Dates,
                     Website=YelpVec)
```

Moving on to OpenTable, we only need to create the categorical vector and merge the variables to a data frame, since the dates were cleaned in Section 2.3.2

```
#Vector, OpenTable identifier
OpenTableVec <- rep("OpenTable", length(OpenTableData$Reviews))

#Merge to data frame
OpenTableDF <- data_frame(Reviews=OpenTableData$Reviews,
                          Ratings=OpenTableData$Ratings,
                          Dates=OpenTableData$Dates,
                          Website=OpenTableVec)
```

To clean the Zomato data, we will remove the unnecessary `NA` values in the ratings data, as well as the duplicate truncated reviews.

```
#Remove NAs from ratings
ZomatoData$Ratings <- ZomatoData$Ratings[!is.na(ZomatoData$Ratings)]

#Remove duplicate reviews. These truncated duplicates can be
# identified with the regex "read more"
FullRev <- !grepl("read more",ZomatoData$Reviews)
ZomatoData$Ratings <- ZomatoData$Ratings[FullRev]
ZomatoData$Reviews <- ZomatoData$Reviews[FullRev]

#Vector, Zomato identifier
ZomatoVec <- rep("Zomato", length(ZomatoData$Reviews))

#Merge to data frame
ZomatoDF <- data_frame(Reviews=ZomatoData$Reviews,
                       Ratings=ZomatoData$Ratings,
                       Dates=ZomatoData$Dates,
                       Website=ZomatoVec)
```

Finally, we will clean the TripAdvisor data by consolidating the different date variables.

```
#Replace dates of the form "Reviewed ## days ago" with the proper dates
TripAdData$Dates2[grepl("ago|yesterday|today",TripAdData$Dates2)] <-
  TripAdData$Dates1

##Vector, TripAdvisor identifier
TripAdVec <- rep("TripAdvisor",length(TripAdData$Reviews))

#Merge to data frame
TripAdDF <- data_frame(Reviews=TripAdData$Reviews,
                       Ratings=TripAdData$Ratings,
                       Dates=TripAdData$Dates2,
                       Website=TripAdVec)
```

Now that the data from the individual websites is stored in a data frame, we can join these data frames together to have all of the data in one place.

```
#Merge all data frames
d1 <- full_join(YelpDF,OpenTableDF)
d2 <- full_join(d1,ZomatoDF)
CapstoneDF <- full_join(d2,TripAdDF) %>% group_by(Website)
str(CapstoneDF, width=70, strict.width = "cut", give.attr = FALSE)
```

```
## Classes 'grouped_df', 'tbl_df', 'tbl' and 'data.frame':  677 obs. of  4 variables:
##  $ Reviews: chr  "The first time I started dining here was due to s"..
##  $ Ratings: chr  "4.0 star rating" "3.0 star rating" "4.0 star rati"..
##  $ Dates  : chr  "\n        3/19/2017\n    " "\n        1/31/2017\n"..
##  $ Website: chr  "Yelp" "Yelp" "Yelp" "Yelp" ...
```

```
summary(CapstoneDF)
```

```
##    Reviews             Ratings             Dates
##  Length:677         Length:677         Length:677
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##    Website
##  Length:677
##  Class :character
##  Mode  :character
```

This is a great start, but so far all of the data is of `character` class. In the following sections, we will clean up the date and ratings data in order to express them in more quantitative terms.

3.2 Cleaning Up the Dates

We will begin the date cleaning stage by taking a look at the date data from Yelp.

```
head(subset(CapstoneDF$Dates, CapstoneDF$Website=="Yelp"),n=4)
```

```
## [1] "\n        3/19/2017\n    " "\n        1/31/2017\n    "
## [3] "\n        1/24/2017\n    " "\n        1/10/2017\n    "
```

The first thing that needs to be done is to get rid of the newline and space characters.

```
#Remove newline characters and spaces
CapstoneDF$Dates <- gsub("\n *","",CapstoneDF$Dates)
head(subset(CapstoneDF$Dates, CapstoneDF$Website=="Yelp"),n=10)
```

```
##  [1] "3/19/2017"  "1/31/2017"  "1/24/2017"  "1/10/2017"  "12/27/2016"
##  [6] "7/12/2016"  "7/29/2016"  "3/10/2017"  "11/10/2016" "6/17/2016"
```

Next we look for data that doesn't fit this pattern.

```
#Find data that doesn't fit Yelp pattern
UncleanDates <- CapstoneDF$Dates[!grepl("^[0-9].*[0-9]$",CapstoneDF$Dates)]
head(UncleanDates, n=6)
```

```
## [1] "3/5/2016Updated review"     "2/14/2014Updated review"
## [3] "Dined on March 9, 2017"     "Dined on March 9, 2017"
## [5] "Dined on February 27, 2017" "Dined on February 23, 2017"
```

Given this output, we see that we need to remove the "Updated review" phrase, as well as "Dined on".

```
#Remove "Updated review"
CapstoneDF$Dates <- gsub("Updated review.*$","", CapstoneDF$Dates)

#Remove "Dined on "
CapstoneDF$Dates <- gsub("Dined on ","",CapstoneDF$Dates)
```

With that done, let's take a look at the date data from OpenTable, TripAdvisor, and Zomato.

```
#OpenTable
head(subset(CapstoneDF$Dates, CapstoneDF$Website=="OpenTable"),n=9)
```

```
## [1] "17242"             "March 9, 2017"     "March 9, 2017"
## [4] "February 27, 2017" "February 23, 2017" "February 19, 2017"
## [7] "February 19, 2017" "February 19, 2017" "February 15, 2017"
```

```
#TripAdvisor
head(subset(CapstoneDF$Dates, CapstoneDF$Website=="TripAdvisor"),n=10)
```

```
##  [1] "22 March 2017"            "21 March 2017"
##  [3] "5 March 2017"             "1 March 2017"
##  [5] "20 February 2017"         "17 February 2017"
##  [7] "Reviewed 9 February 2017" "Reviewed 6 February 2017"
##  [9] "Reviewed 2 February 2017" "Reviewed 2 February 2017"
```

```
#Zomato
head(subset(CapstoneDF$Dates, CapstoneDF$Website=="Zomato"),n=6)
```

```
## [1] "2016-08-01 16:22:56" "2016-04-20 05:25:03" "2016-01-15 23:20:46"
## [4] "2015-12-23 00:15:16" "2015-11-19 01:28:48" "2015-10-12 04:06:25"
```

The data from OpenTable and Zomato is already expressed in formats that we can manipulate. All we have to do is remove "Reviewed" from the TripAdvisor data.

```
#Remove "Reviewed " from TripAdvisor data
CapstoneDF$Dates <- gsub("Reviewed ","",CapstoneDF$Dates)
head(subset(CapstoneDF$Dates, CapstoneDF$Website=="TripAdvisor"))
```

```
## [1] "22 March 2017"    "21 March 2017"    "5 March 2017"
## [4] "1 March 2017"     "20 February 2017" "17 February 2017"
```

The dates data should now be stripped of unnecessary characters. The next step in our cleaning process is to express all of this data in terms of a `date` class. This will be done in parts since the date is formatted differently for the different websites.

Yelp:

```
#Yelp date format
head(subset(CapstoneDF$Dates, CapstoneDF$Website=="Yelp"))
```

```
## [1] "3/19/2017"  "1/31/2017"  "1/24/2017"  "1/10/2017"  "12/27/2016"
## [6] "7/12/2016"
```

```
#grep for the Yelp dates
YelpDateRegex <- grep("^[0-9]+/.*[0-9]$",CapstoneDF$Dates)
#Express dates as date variables
CapstoneDF$Dates[YelpDateRegex] <-
  CapstoneDF$Dates[YelpDateRegex] %>%
  as.Date(format="%m/%d/%Y")
```

OpenTable:

```
#Open Table date format:
head(subset(CapstoneDF$Dates, CapstoneDF$Website=="OpenTable"))
```

```
## [1] "17242"             "March 9, 2017"     "March 9, 2017"
## [4] "February 27, 2017" "February 23, 2017" "February 19, 2017"
```

```
#grep for OpenTable dates and express as date
OpenTableDateRegex <-
  grep("^([Jj]|[Ff]|[Mm]|[Aa]|[Jj]|[Ss]|[Oo]|[Nn]|[Dd]).+[0-9]+$",
                           CapstoneDF$Dates)

CapstoneDF$Dates[OpenTableDateRegex] <-
  CapstoneDF$Dates[OpenTableDateRegex] %>%
  as.Date(format="%B %d, %Y")
```

TripAdvisor:

```
#TripAdvisor date format:
head(subset(CapstoneDF$Dates, CapstoneDF$Website=="TripAdvisor"))
```

```
## [1] "22 March 2017"    "21 March 2017"    "5 March 2017"
## [4] "1 March 2017"     "20 February 2017" "17 February 2017"
```

```
#grep for TripAdvisor dates and express as date variable
TripAdRegex <-
  grep("^[0-9]+ ([Jj]|[Ff]|[Mm]|[Aa]|[Jj]|[Ss]|[Oo]|[Nn]|[Dd]).+[0-9]+$",
       CapstoneDF$Dates)

CapstoneDF$Dates[TripAdRegex] <-
  CapstoneDF$Dates[TripAdRegex] %>%
  as.Date(format="%d %B %Y")
```

Finally, since Zomato dates are already in `POSIXct` format, we can convert them trivially to dates.

```
CapstoneDF$Dates[which(CapstoneDF$Website == "Zomato")] <-
  CapstoneDF$Dates[which(CapstoneDF$Website == "Zomato")] %>%
  as.Date()
```

Let's finish by imposing the `date` class on the `Dates` variable of our data frame, just to be sure.

```
class(CapstoneDF$Dates) <- "Date"
str(CapstoneDF$Dates, width=70, strict.width="cut")
```

```
## Date[1:677], format: "2017-03-19" "2017-01-31" "2017-01-24" "2017-01-10" ...
```

The dates data is now properly formatted and ready for analysis.

3.3 Numeric Ratings

In the previous section, we removed unnecessary characters from the date data and converted the data to a common `date` class. Here we will do the same with the ratings data.

Yelp:

```
#Yelp ratings
head(subset(CapstoneDF$Ratings, CapstoneDF$Website=="Yelp"))
```

```
## [1] "4.0 star rating" "3.0 star rating" "4.0 star rating" "4.0 star rating"
## [5] "3.0 star rating" "3.0 star rating"
```

```
#Get rid of "star rating"
CapstoneDF$Ratings <- gsub("star rating","",CapstoneDF$Ratings)
```

OpenTable:

```
#Open Table ratings
head(subset(CapstoneDF$Ratings, CapstoneDF$Website=="OpenTable"))
```

```
## [1] "3" "5" "5" "3" "5" "3"
```

This data is already in a workable format.

TripAdvisor:

```
#TripAdvisor ratings
head(subset(CapstoneDF$Ratings, CapstoneDF$Website=="TripAdvisor"), n=4)
```

```
## [1] "4 of 5 bubbles" "2 of 5 bubbles" "4 of 5 bubbles" "4 of 5 bubbles"
```

```
#Get rid of "of 5 bubbles"
CapstoneDF$Ratings <- gsub("of [0-9] bubbles","",CapstoneDF$Ratings)
```

Zomato:

```
#Zomato ratings
head(subset(CapstoneDF$Ratings, CapstoneDF$Website=="Zomato"))
```

```
## [1] "Rated 3.0" "Rated 5.0" "Rated 4.5" "Rated 1.5" "Rated 4.0" "Rated 1.0"
```

```
#Get rid of "Rated "
CapstoneDF$Ratings <- gsub("Rated ","",CapstoneDF$Ratings)
```

Having removed unnecessary characters, we will now impose the `numeric` class on the data.

```
#Impose numeric class
class(CapstoneDF$Ratings) <- "numeric"
str(CapstoneDF$Ratings)
```

```
##  num [1:677] 4 3 4 4 3 3 4 1 1 3 ...
```

This completes the cleaning process for the numerical ratings.

3.4 Final Touches

Before moving on to data analysis, we will finalize the data cleaning process by making some small adjustments to the data. The first adjustment is to convert the `Website` variable of the data frame to `factor` class, rather than a simple `character` class.

```
#Convert Websites to factor
CapstoneDF$Website <-
  factor(CapstoneDF$Website,order=FALSE,
         levels=c("Yelp","OpenTable","Zomato","TripAdvisor"))

str(CapstoneDF$Website)
```

```
##  Factor w/ 4 levels "Yelp","OpenTable",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
levels(CapstoneDF$Website)
```

```
## [1] "Yelp"        "OpenTable"   "Zomato"      "TripAdvisor"
```

Additionally, we will remove any newline characters from the review data, and remove some unnecessary text in the Zomato review data.

```
# Remove newline characters from reviews
CapstoneDF$Reviews <- gsub("\n", "", CapstoneDF$Reviews)

# Clean up Zomato reviews by removing the 'Rated' beginning
head(subset(CapstoneDF$Reviews, CapstoneDF$Website == "Zomato"), n = 2) %>%
    strtrim(65)
```

```
## [1] "                Rated                                    Had lunch a"
## [2] "                Rated                                    My wife & I"
```

```
CapstoneDF$Reviews <- gsub(" +Rated *", "", CapstoneDF$Reviews)
```

We will finish off the section by writing our newly-cleaned data to file.

```
write_csv(CapstoneDF, "./Data/CapstoneCleanData.csv")
```

This completes the cleaning stage. We are now ready to analyze the data and derive some insights.

4 Data Analysis

Having completed the gritty process of first acquiring the customer review data from the web and subsequently cleaning it, we can now dive into the analysis stage. As mentioned in the Introduction, the analysis will be focussed on two main aspects of the customer reviews: The performance of RMT over time and common sentiments and feedback associated with the restaurant. The first of these points of focus will be explored in Section 4.1 by means of a time-series analysis of the numerical customer ratings. Following this, in Section 4.2, I will perform a text analysis of the customer review data in order to examine the type of feedback that is provided by the restaurant's clientele. In the previous sections, we built a data frame with four variables: `Reviews`, `Ratings`, `Dates`, and `Website`. The variables that we will use primarily to extract insights are the `Reviews` and `Ratings` variables, containing respectively the written customer reviews and the associated numerical ratings. Let's take a moment to load some relevant libraries and read in the clean data.

```
#Load libraries
library(readr)
library(dplyr)
library(tidyr)
```

```
#Read in clean data
CapstoneDF <- read_csv("./Data/CapstoneCleanData.csv")
#Summary of data frame
glimpse(CapstoneDF)
```

```
## Observations: 677
## Variables: 4
## $ Reviews <chr> "The first time I started dining here was due to summe...
## $ Ratings <dbl> 4, 3, 4, 4, 3, 3, 4, 1, 1, 3, 3, 4, 5, 4, 3, 4, 5, 5, ...
## $ Dates   <date> 2017-03-19, 2017-01-31, 2017-01-24, 2017-01-10, 2016-...
## $ Website <chr> "Yelp", "Yelp", "Yelp", "Yelp", "Yelp", "Yelp", "Yelp"...
```

We have data for 677 customer reviews from the four aformentioned websites: Yelp, OpenTable, TripAdvisor, and Zomato. Before diving into the analysis, we will do some basic data wrangling to reformat the structure of the data frame and facilitate analysis.

```
#Create variable describing annual quarters: Quarters
CapstoneDF <- CapstoneDF  %>% mutate(Quarters = quarters.Date(Dates))

#Separate Dates variable into Year, Month, Day variables
CapstoneDF <- CapstoneDF %>% separate(Dates, c("Year","Month","Day"))

#Create variable that describes Month and Year together: YearMonth
tempdf <- CapstoneDF %>% unite("YearMonth", Year, Month, sep="-")
CapstoneDF$YearMonth <- tempdf$YearMonth

#Create variable that describes Quarters and Year together: YearQuarters
tempdf <- CapstoneDF %>% unite("YearQuarters", Year, Quarters, sep="-")
CapstoneDF$YearQuarters <- tempdf$YearQuarters

#Convert Website and Quarters to factor class
CapstoneDF$Website <- factor(CapstoneDF$Website)
CapstoneDF$Quarters <- factor(CapstoneDF$Quarters)

#Create new variables

#Review count by year
t1 <- CapstoneDF %>% group_by(Year) %>% summarise(countYear=n())
CapstoneDF <- left_join(CapstoneDF,t1, by="Year")

#Review count by quarters
t2 <- CapstoneDF %>% group_by(YearQuarters) %>% summarise(countQuarters=n())
CapstoneDF <- left_join(CapstoneDF,t2, by="YearQuarters")

#Review count by month
t3 <- CapstoneDF %>% group_by(YearMonth) %>% summarise(countMonth=n())
```

```
CapstoneDF <- left_join(CapstoneDF,t3, by="YearMonth")

head(CapstoneDF[-1])
```

```
## # A tibble: 6 × 11
##   Ratings  Year Month   Day Website Quarters YearMonth YearQuarters
##     <dbl> <chr> <chr> <chr>  <fctr>   <fctr>     <chr>        <chr>
## 1       4  2017    03    19    Yelp       Q1   2017-03      2017-Q1
## 2       3  2017    01    31    Yelp       Q1   2017-01      2017-Q1
## 3       4  2017    01    24    Yelp       Q1   2017-01      2017-Q1
## 4       4  2017    01    10    Yelp       Q1   2017-01      2017-Q1
## 5       3  2016    12    27    Yelp       Q4   2016-12      2016-Q4
## 6       3  2016    07    12    Yelp       Q3   2016-07      2016-Q3
## # ... with 3 more variables: countYear <int>, countQuarters <int>,
## #   countMonth <int>
```

For the purpose of clarity, the code used to generate the plots in this section has been ommitted. Once again, the full details of the code can be found in the .Rmd file associated with this document, or in the RMTAnalysis.R file. We are ready to begin our analysis.

4.1 Time Series Analysis of Ratings

The first part of our data analysis process will be focussed on examining the "dynamics" of the `Ratings` variable over time in order to find answers to the following questions:

- How has RMT performed over the years?
- What periods of time are associated with positive and negative reviews?
- How has the restaurant's performance changed over time?
- How is the restaurant performing currently?

The primary package used to perform this analysis will be `ggplot2`.

```
library(ggplot2)
library(ggthemes)
```

As a first step, let's compute the cumulative mean value of the ratings for RMT:

```
mean(CapstoneDF$Ratings)
```

```
## [1] 3.79099
```

The mean value of the numerical customer ratings is 3.79. We can think of this as the ratings aggregated at the lowest possible level of granularity, i.e. all of the data is used to compute the average. This is just a single number so it isn't too informative, but it does provides some insight into the overall performance of the restaurant across its entire lifespan. We can deduce that RMT has done relatively well over its years of operation, with

a cumulative rating that is above the basic average value of 3 out of 5. However this leads to more questions, namingly why this particular value? Why is the average not higher? Why not lower? We can mine for further insight into the history of the restaurant's quality by increasing the granularity of the data. Let's begin by aggregating the average value of the ratings by year. This data is presented in Figure 5. The dashed blacked line is plotted to represent the cumulative average value of 3.79.

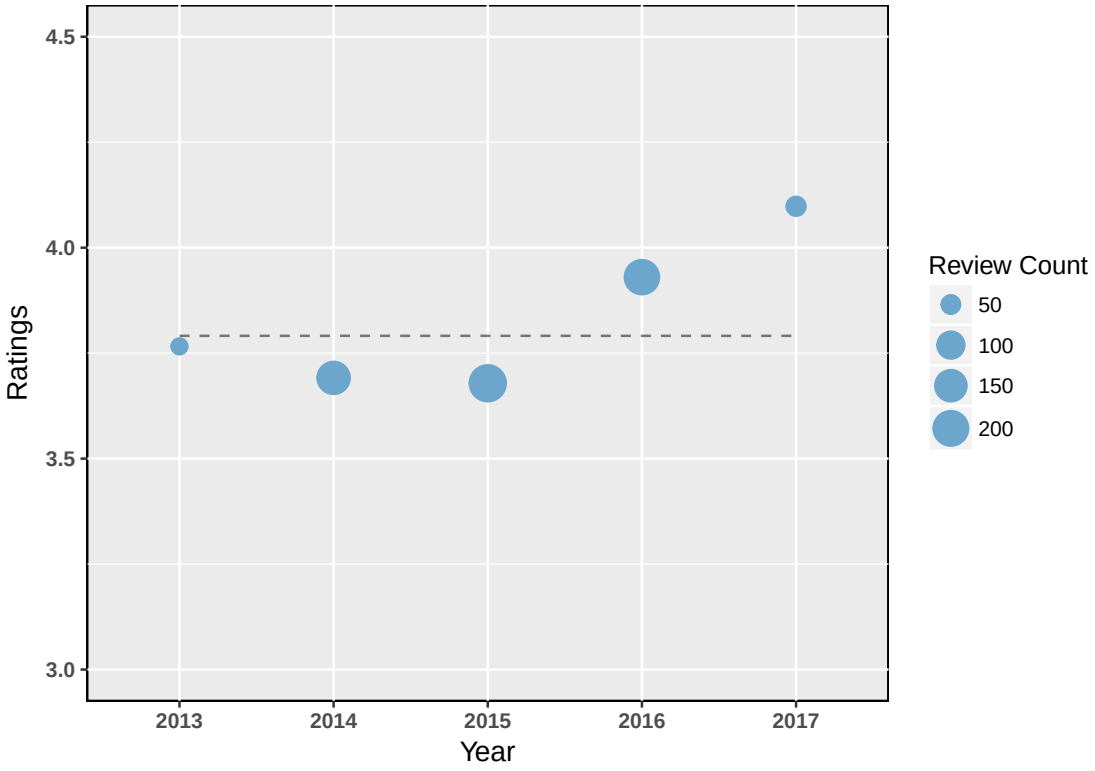

Figure 5: Customer ratings averaged by year

By computing the mean value of the customer ratings for each year, we begin to see some of the dynamics of the ratings. In particular, we can observe that from 2013 to 2015, the restaurant experienced a decreasing trend in its ratings. 2015 was the worst year in terms of quality for the restaurant, as reported by the clientele. Since then the ratings have been rising steadily. Now, one quarter into 2017, the restaurant's annual ratings are at an all time high, with a mean value of 4.10. The interplay between the initial negative trend and the more recent positive growth in the ratings is what causes the global average to be 3.79. The recent improvement in ratings is a promising sign for the near future. However it should be met with some skepticism since things can change rapidly. Another thing to note is that I have encoded the information about the number of reviews for each year in the size of data points. We can see that the years 2014 through 2016 have approximately the same number of reviews, whereas 2013 and 2017 have fewer reviews. This is to be expected since RMT opened in November 2013, and 2017 has only just begun. The average value for 2017 will certainly be subject to change as the year progresses. In order to better understand why the ratings were lowest in 2015 and what we might expect from the present growth

trend, we will mine deeper into the data by computing an average for each yearly quarter. This is shown in Figure 6.

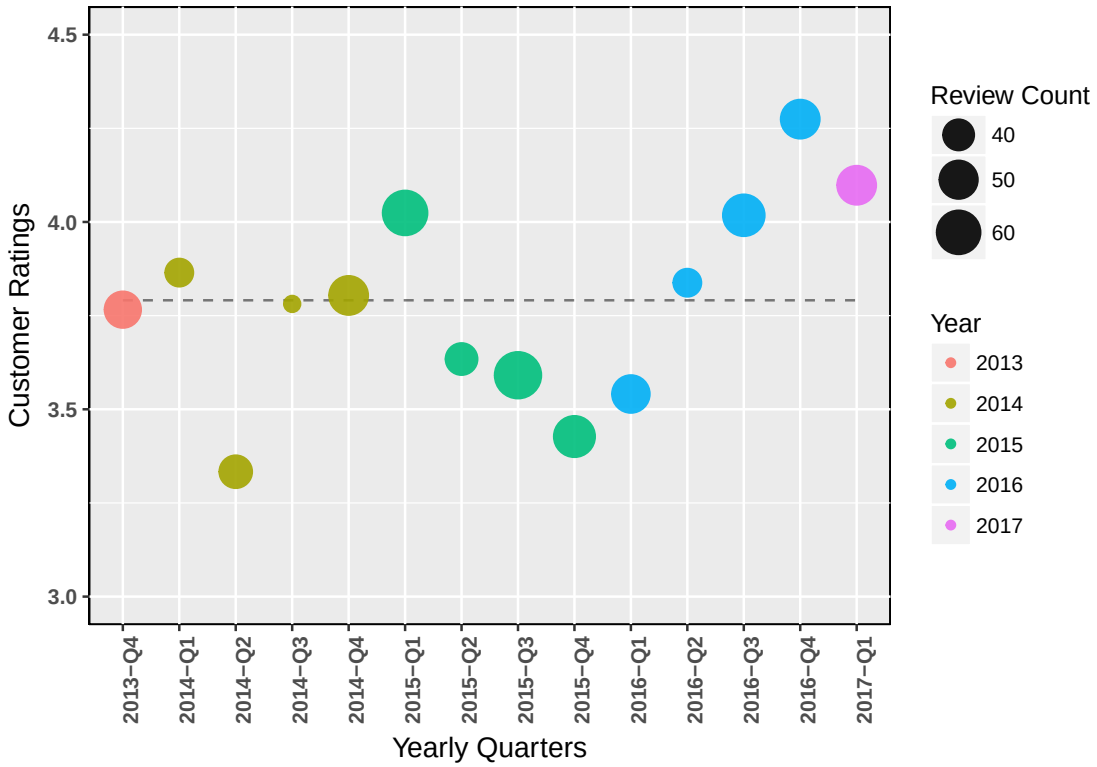

Figure 6: Customer ratings averaged by yearly quarters

Figure 6 presents some interesting details in the data. During RMT's first year in business, from the end of 2013 to the end of 2014, the quarterly ratings are mainly scattered around the global mean of 3.79, presented as the dashed black line. The exception is an unexpectedly low value for the ratings in 2014-Q2, which can be interpreted as an outlier during this period, since the ratings quickly recovered their previous values in the following quarters. In this first year, the restaurant experienced no real growth in terms of quality, though it appears to have avoided the serious decrease in quality associated with 2014-Q2. Moving into the first quarter of 2015, the ratings appeared to promise that things would be different in the upcoming year, with an initial increase in quality. However the data from the rest of the year shows that this wasn't the case. Though the ratings for started out high in 2015, they exhibited a relatively steady decline for the remainder of the year, reaching bottom in the fourth quarter of 2015 with an average value of 3.43. Since the beginning of 2016 however, the restaurant has experienced a tremendous improvement in the quality of the ratings it has received. Following the low in 2015-Q4, the ratings quickly reached their earlier 2014 values in the second quarter of 2016, and continued on to an all-time high in 2016-Q4. This period of growth implies that the restaurant underwent some significant changes in 2016. The final piece to examine is that, since the beginning of 2017, the ratings have dipped slightly, from 4.27 in 2016-Q4 to 4.10 in 2017-Q1. It remains to be seen whether this is simply random scatter around a new high-valued average, or if

this is the beginning of a new downward trend. Ultimately, the outcome will depend on the decisions that are made at RMT over the course of the following months. To get a better sense of the underlying trends in the data, let's take a look at the ratings at a monthly level, described in Figure 7.

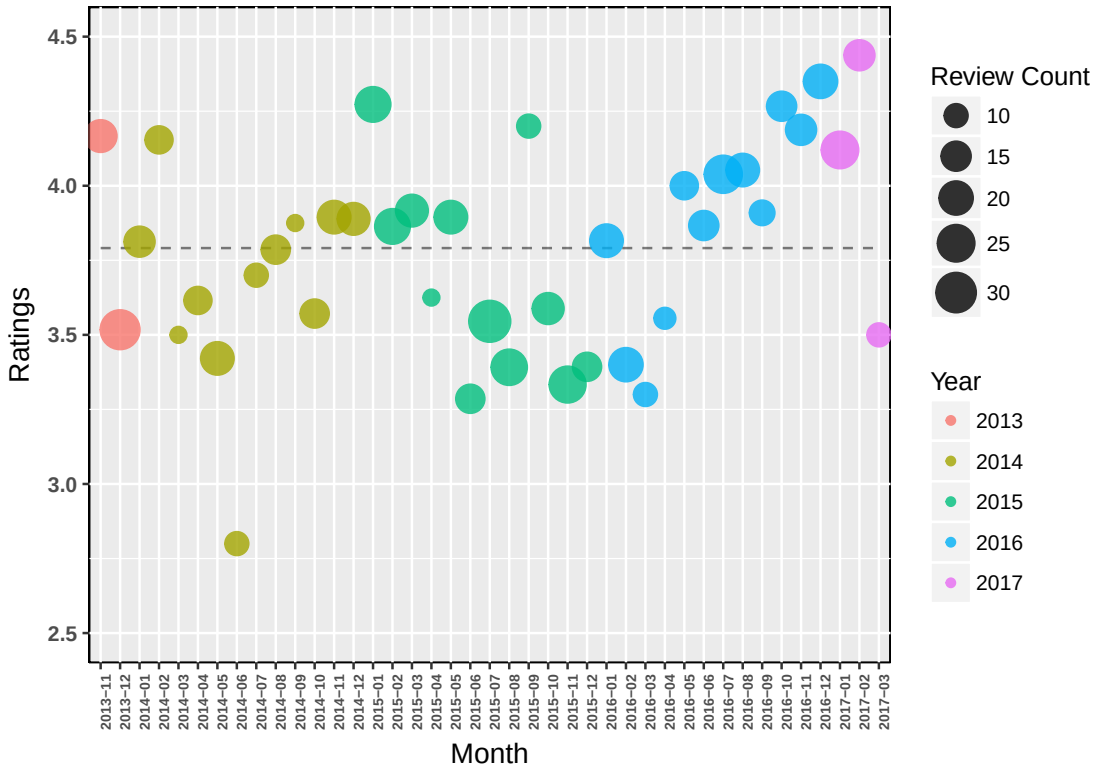

Figure 7: Customer ratings averaged by month

Looking into the monthly ratings, we can observe that the general trends that were present in Figure 6 are still present here, though the data exhibits a larger degree of scatter. We can see quite clearly that the low ratings in 2014-Q2 are due to a particularly poor performance in June 2014. Following the dip in ratings at this time, the restaurant quickly recovered its previous ratings. The strong first quarter of 2015 is due in large part to a strong month of January, which could potentially be interpreted as an outlier in the data. The ratings for the remainder of 2015-Q1 sit around the cumulative mean, indicated by the dashed line. As in the previous discussion of the quarterly ratings, the rest of 2015 is met with a decrease in the ratings for the restaurant. Interestingly, there appears to be a high-valued outlier for the month of September 2015, though this spike in quality does not last. The ratings appear to pick up in January 2016, though in reality the true growth trend of 2016 doesn't begin until the month of April. There is a clear improvement in the ratings throughout 2016 despite the random scatter of the data, though it is unclear whether this trend has been continuing in recent months. It is possible to interpret the data from December 2016, January 2017 and February 2017 as being scattered around a new normal average of approximately 4.25. The month of March 2016 however suggests that this new average might be short-lived, since the ratings have dropped back down to

around 3.5. This explains the dip in the quarterly average for 2017-Q1. Whether this is simply a low-valued outlier, like that of June 2014, or is instead indicative of a forthcoming downward trend will depend on the performance of RMT in the immediate future.

In the next set of plots we will examine how the distribution of ratings has changed over time. Figure 8 displays the overall dispersion of the customer ratings, in which we can see that the majority of the ratings have had the values of 4 and 5, with respective percentages of 38% and 29%. This comprises 67% of the data. The remainder of the ratings have been 3 or less. The fact that the overall distribution in ratings is skewed towards higher values is what causes the cumulative average for RMT to sit at 3.79, rather than at 3, which would be the expected value for a symmetrical distribution.

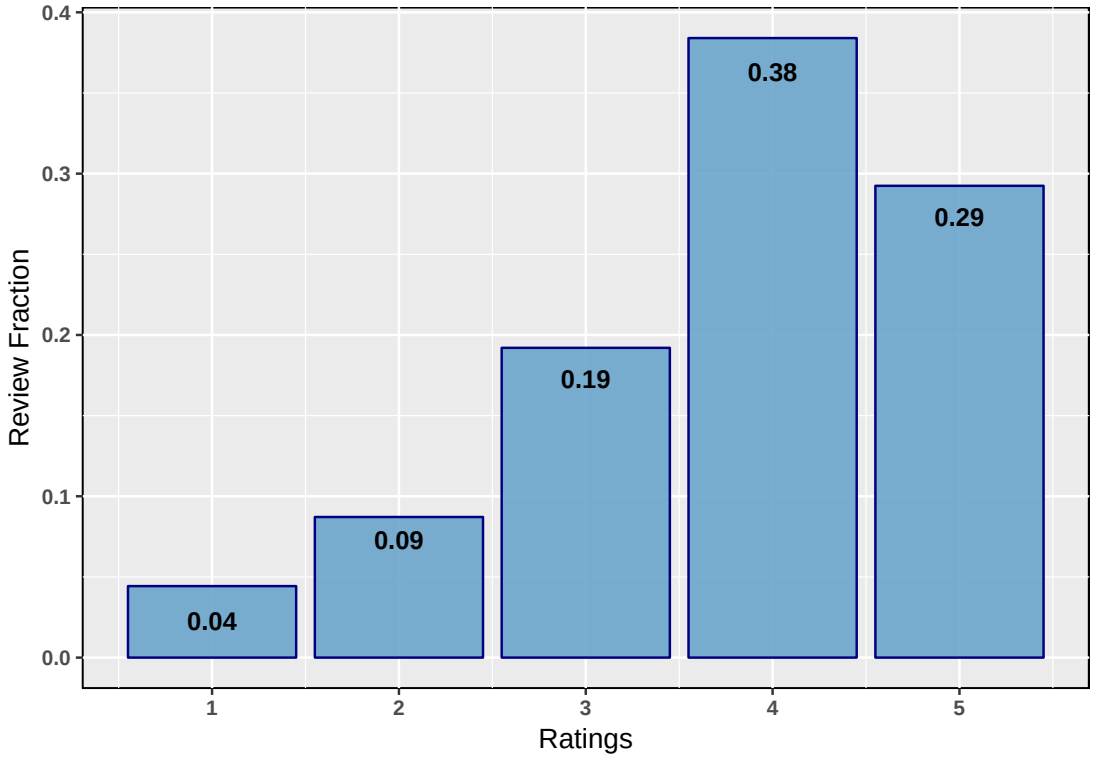

Figure 8: Cumulative dispersion of numerical ratings

To better understand how these ratings are dispersed over time, we will break the data down by years. A histogram of the rating fractions is shown in Figure 9, while a scatter plot of the rating fractions over time is presented in Figure 10. The latter also includes a number of linear regression models that describe the trends of the ratings categories. These models are presented in the form of dashed black lines and shaded error intervals. These plots allow us to observe the distribution of the ratings for each of the years and understand how the ratings have changed over time.

We will go through each rating category separately in order to minimize confusion. In Figure 8, we can observe that 1-star ratings comprise 4% of the data overall. The majority of the 1-star ratings occured in 2014 and 2015, as shown in Figures 9 and 10. The yearly trend of the 1-star ratings has been parabolic, as can be seen in Figure 10. The peak occurred in 2015 and the number of 1-star ratings have been decreasing since then. For 2-

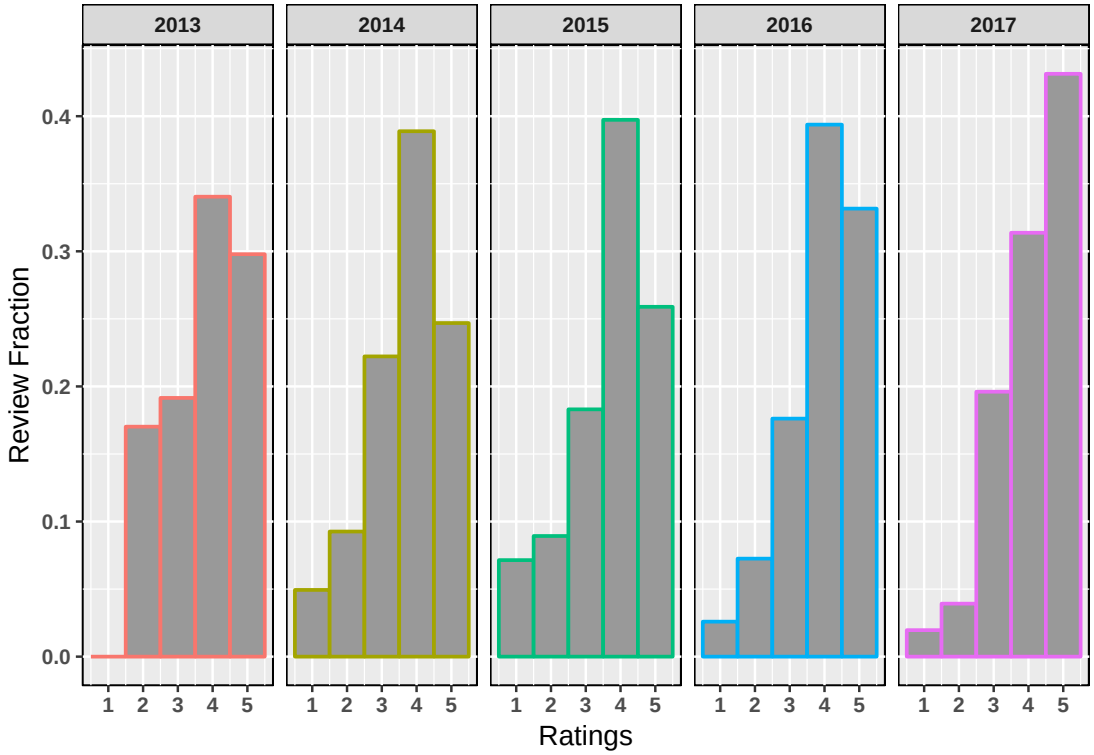

Figure 9: Density histogram of yearly customer ratings

star ratings, the story is slightly different. Though there are more 2-star ratings than 1-star ratings overall, they don't comprise a significant fraction of the data at 9%. Incidentally, the most 2-star ratings were given in 2013. They have since been decreasing and are now at their lowest point in 2017. This trend can be well approximated using a linear model with negative slope, as shown in Figure 10. Moving on to the 3-star ratings, we can see that, though the number of ratings varies slightly from year to year, the overall trend is fairly constant. The percentage of 3-star ratings has not exhibited any real growth or decline over this period. For most of the restaurant's lifespan, 4 stars has been the restaurant's most common rating. This was especially the case from 2014 through to 2016, where 4-star ratings comprised around 40% of the ratings during those years. The percentage of 4-star ratings was slightly lower in 2013, but remained steady from 2014 to 2016. It is only in 2017 that the fraction of 4-star ratings has dipped below its 2013 value, resulting in an overall trend that can be approximated parabolically. This recent dip in the fraction of 4-star ratings might at first seem like a negative occurrence, but taking a look at the fraction of 5-star ratings tells a different story. In 2013, the number of 5-star ratings comprised around 30% of the ratings, but this fraction decreased in 2014 and 2015. This decrease in 5-star ratings and simulatenous increase in 1-star ratings over this period resulted in the low average value for 2015 that we observed in Figures 5 and 6. From those Figures, we also noted that there was a strong growth trend in 2016. This can be explained by the fact that the percentage of 5-star ratings began to increase in 2016, while the fraction of 1- and 2-star ratings began to decrease. This trend continued into 2017 even as the fraction of 4-star ratings decreased. The surge in the percentage of 5-star ratings, coupled with

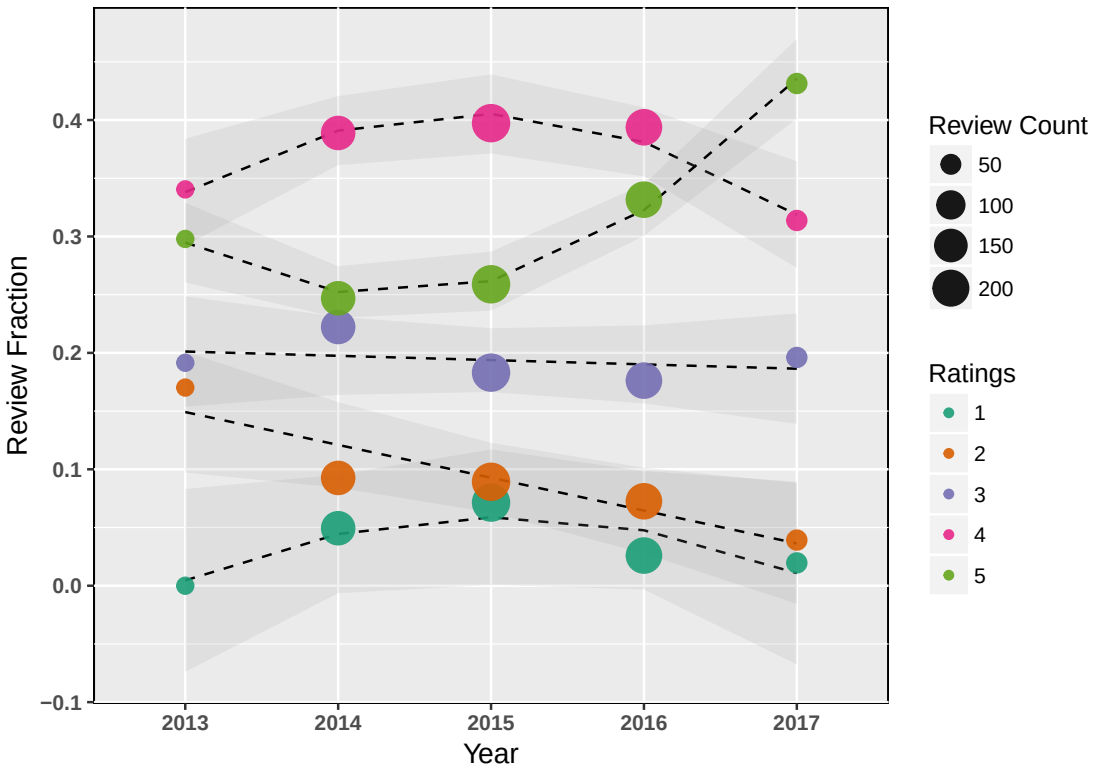

Figure 10: Time series data for ratings Fractions

the stability of 3-star ratings and the decrease in lower ratings has offset the effect of a decreasing fraction of 4-star ratings. This suggests that more customer have been rating RMT as 5 stars in the last few months, rather than 1, 2 or 4 stars.

This concludes the time-series analysis of the ratings. In Section 4.2 we will attempt to understand some of the trends presented here by mining the text data from the online reviews.

–>

4.2 Customer Reviews Text and Sentiment Analysis

In the previous Section, we performed an analysis of the numerical ratings and observed some interesting trends. We were able to derive some data-driven insight into how RMT has performed since its inception, observing periods of high- and low-quality ratings, as well as periods of large improvement. Though the numerical ratings data provides information into how the restaurant has performed in the past, it does not elaborate on why the ratings have been the way they were. Insights into the motivations behind the ratings can be obtained by mining the customer review data. Particularly, we can analyze the text data to obtain answers to the following questions:

- Why has the restaurant been given the ratings that it has?
- What are RMT's strengths? What are its weaknesses?
- What is the most common feedback that is returned by the restaurant's customers?

- What are the most important aspects of a customer's experience at RMT?

We will dive into the customer review data using a number of methods in text analysis. The analysis will be broken down into two Sections. In Section 4.2.1, we will perform a global analysis of the full set of customer review data in order to get an idea of the general sentiment towards RMT. Following this, in Section 4.2.2, we will break the data down into a subset of positive and negative reviews to examine the different themes and sentiments associated with the different review subsets. The analysis will be performed using the following R packages: `tm`, `wordcloud`, and `syuzhet`. Additionally, I have written a function `TextAnalysis()` that performs the main data wrangling steps of the text analysis. For details on this function, refer to the RMTAnalysis.R) file.

```
library(tm)
library(wordcloud)
library(syuzhet)
```

4.2.1 Global Analysis

We will begin the overall text analysis of RMT reviews by calling the `TextAnalysis()` function on the full data set.

```
Capstone_wA <- TextAnalysis(CapstoneDF)
```

The first thing that we can examine is the set of words that occur most frequently within the customer reviews. This data is depicted visually as a word cloud in Figure 11, where the larger words are those with the higher number of occurrences. Another way to examine this data is to take a look at what fraction of reviews contain these common words, as shown in Figure 12.

From this combined output, we can see that the words that show up most often in all of the customer reviews are "food", "good", "great", and "service". We can immediately derive a number of insights from these results. The first is that food is the single most important aspect of the restaurant, the word "food" occurring 517 times in 61% of the online reviews. This is the one word that is mentioned more than any other and so the aspects of a customer's experience related to food will be the most influential in their perception of the experience. Since the data has not yet been parsed according to the quality of ratings, we cannot tell whether the mention of food is placed within a positive or negative context. This will be explored in detail below. However we do have some potential insight into this, since the next most common words are "good" and "great". These are both terms indicating positive sentiment. The fact that these two words are the second and third most common words in the customer reviews indicate that the overall sentiment towards the customers' experience is positive. This is in line with the ratings analysis from Section 4.1, in which we discovered that overall, the restaurant was rated favourably. In Figure 11, we can also see that words such as "delicious" and "excellent" are fairly common, which also suggests an overall positive sentiment towards the food. At this stage, it is important to maintain a degree of criticality towards the occurrence of words such as "good" and "great", since their frequency of occurrence does not take into

Figure 11: Word cloud for RMT customer review data

account instances where the words are preceded by qualifiers or words indicating negation, e.g. "pretty good", "not good", etc. This will be addressed in Section 4.2.2. Moving on, the second most important aspect of the customers' experience is the restaurant's service. The word "service" is the fourth most common word in the analysis. However, as seen in Figure 12, it actually occurs in 44% of reviews, second only to the word "food". Furthermore, from the word cloud in Figure 11 we can also see that the words "staff" and "server" occur with a large frequency as well. This suggests that the customer's experience related to RMT's service and staff will have a large impact, either negative or positive, on their perception. Words such as "nice", "friendly" and "attentive", which show up in the word cloud analysis, suggest that, overall, the customers' experience of the restaurant's service is positive.

This analysis allows us to gain some insight into what customer are talking about in their reviews. In order to dive deeper into the data, we examine what kinds of words are usually associated with these common words within the reviews. This is done using the `tm::findAssocs()` function. Let's begin with the word "food".

```
findAssocs(Capstone_wA$TDM, "food", .18)
```

```
## $food
## service    good   great quality
##    0.23    0.22    0.18    0.18
```

The strongest association for "food" is with the word "service", which reinforces the idea that RMT's food and service are its two most important characteristics. The next strongest

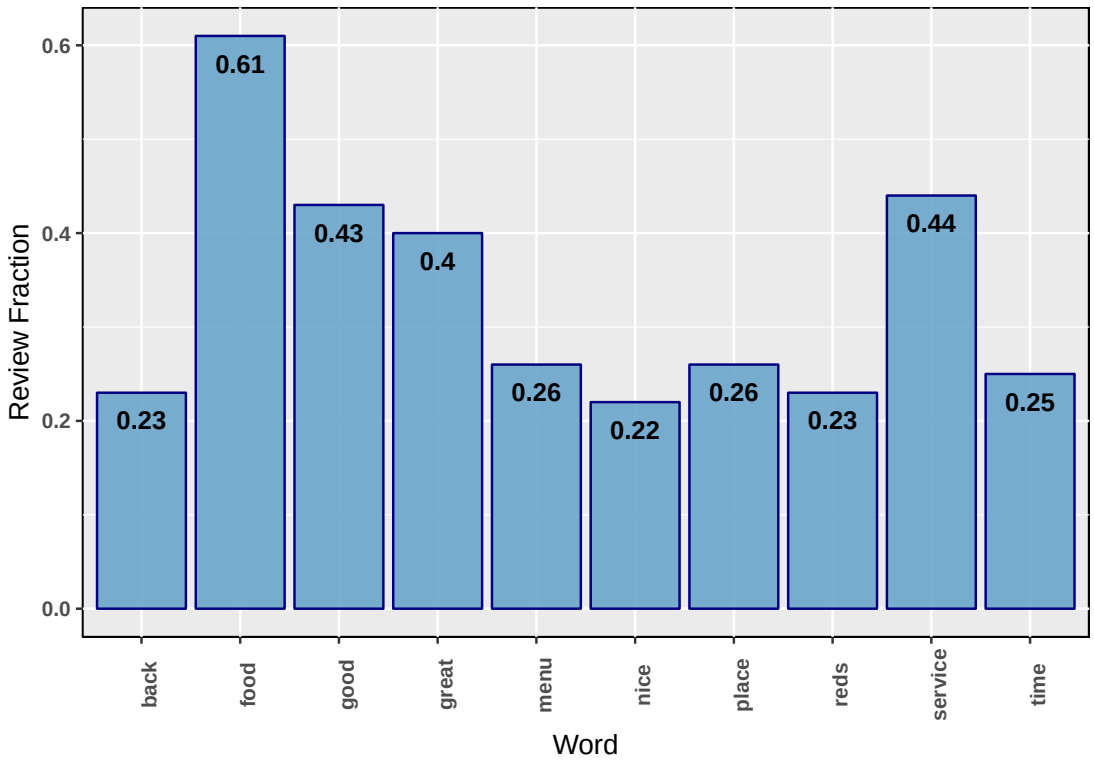

Figure 12: Fraction of reviews containing common words

associations are "good", "great" and "quality". The word "quality" suggests that reviewers often comment on the quality of the food they buy, while the words "good" and "great" speak to the characterization of this quality. The fact that "good" and "great" are strongly associated with "food" are a positive sign since, as mentioned previously, these are generally positive terms. However we notice that "good" has a slightly stronger association with "food" than "great", which means that overall the quality could be improved, since "great" is understood to express strong positive sentiment than "good". As mentioned above, we must maintain some skepticism since the data related to "good" and "great" can possibly include terms indicating negation and qualification.

Next we will look into the associations with the word "service".

```
findAssocs(Capstone_wA$TDM, "service", 0.15)
```

```
## $service
##      slow      food     great  friendly excellent  customer goodgreat
##      0.24      0.23      0.17      0.17      0.16      0.16      0.16
```

Here we find that the strongest association is "slow". It is clear that this suggests that the customers often experience slow service. This is a clear point of improvement for RMT. The next word is "food", which is to be expected from the results above. Following this we find the words "great", "friendly", and "excellent". This is encouraging as it suggests that despite being slow, the service at RMT is of high quality and the waitstaff are friendly.

Finally, looking into the associations for words "good" and "great", we find a number of results.

```
findAssocs(Capstone_wA$TDM, c("good","great"), c(0.15, 0.10))
```

```
## $good
##       food decoration       1995  batmobile      bonus  excluding
##       0.22       0.18       0.18       0.18       0.18       0.18
##     jacket  succulent        tax
##       0.18       0.18       0.18
##
## $great
##  goodgreat       food    service       bold  sharables      lousy
##       0.19       0.18       0.17       0.14       0.14       0.14
##    wrongit atmosphere      staff
##       0.14       0.13       0.11
```

Both of these words are strongly associated with food, as is expected from above. The word "great" is also associated with the words "service", "staff" and "atmosphere", while the word "good" is associated with "decoration". The associations with "atmosphere" and "decoration" suggest that consumers enjoy the decor and ambiance of the restaurant. Note that these results also return some artefacts of the algorithm that have no real generalizable meaning.

Another word that is mentioned frequently in the customer reviews is the word "reds". Though this might simply be a result of reviewers mentioning the restaurant's name in their comments, we can get a better idea of how this word is contextualized by examining some of its associations.

```
#Associations with "reds"
findAssocs(Capstone_wA$TDM, "reds", 0.20)
```

```
## $reds
##  midtown   tavern adelaide
##     0.49     0.36     0.21
```

The words most commonly associated with "reds" are "midtown", "tavern", and "adelaide". The first two suggest that "reds" is used mainly as a statement of the restaurant's name within the review comments. However, the third word, "adelaide", is a reference to RMT's other location, REDS Wine Tavern, on Adelaide Street West in Toronto, indicating that there is a degree of brand recognition among RMT's clientele. References to REDS Wine Tavern imply that customers have had previous experiences at this restaurant before trying the newer location of RMT.

The final way in which we will examine the review data is by performing a sentiment analysis on it. The purpose of this analysis is to identify key words within the data that are associated with basic emotions and sentiments. These word-sentiment and word-emotion associations are based on the NRC Word-Emotion Association Lexicon. R provides functions for performing this type of analysis in the `syuzhet` package. The general idea is that

each review will contain a number of words expressing different emotions and sentiments. These words can be tallied and plotted to show the distribution of emotions and sentiments within the data. This is shown for the review data in Figure 13.

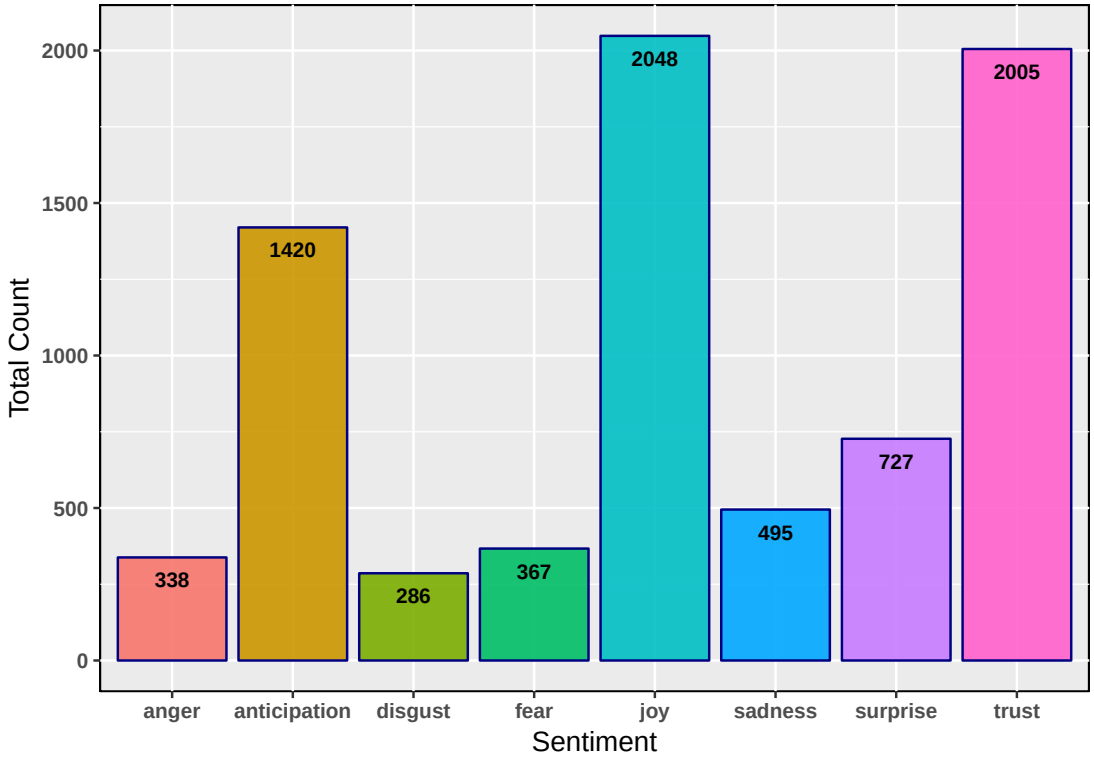

Figure 13: Sentiment Analysis for RMT Customer Reviews

Based on this analysis, the strongest emotions associated with customers' experience of RMT are joy, trust, and anticipation. That joy is the strongest sentiment expressed in the reviews reinforces the analysis so far that RMT provides an experience of high quality for most customers. The sentiment of trust indicates that REDS has a strong brand value and credibility. The clientele trusts the restaurant and expect to be satisfied with great food and service when they visit. This is also outlined in that the next most common sentiment is that of anticipation. Anticipation speaks to how the guests feel before going to RMT, but also how they feel afterwards. Customers may be excited to dine out at RMT based on the restaurant's strong brand and previous experiences at REDS Wine Tavern. The sentiment of anticipation can also indicate that guests expect more value for their money when visiting RMT.

This concludes our analysis of the full set of customer review data. We will now mine more deeply into the data by examining how the data varies for positive and negative reviews.

4.2.2 Text Analysis of Positive and Negative Reviews

In this Section we will dive deeper into the customer review data by parsing the overall data set into positive and negative reviews. For the purpose of this investigation, positive

reviews will be taken to be those reviews that are associated with a 4- or 5-star rating, while negative reviews will be those with numerical ratings of 3 stars or less. The format of the analysis will be similar to that performed in the previous Section. Let's begin by parsing the data into the appropriate subsets and running the `TextAnalysis()` function.

```
#Break data into positive and negative reviews
CapstoneDF_pos <- CapstoneDF %>% subset(Ratings > 3)
CapstoneDF_neg <- CapstoneDF %>% subset(Ratings <= 3)

#Perform text analytics on pos and neg reviews
CapstonePos_wA <- TextAnalysis(CapstoneDF_pos)
CapstoneNeg_wA <- TextAnalysis(CapstoneDF_neg)
```

Of the 677 online reviews in the data set, the positive data subset contains 458 reviews, i.e. 68% of the data, while the negative subset contains 32% of the data at 219 reviews. This information can be examined visually in Figure 8 of Section 4.1. Now, as we did in Section 4.2.1, we will take a look at the words that show up most frequently in both data subsets. The word cloud plots are presented Figures 14 and 15 for the positive and negative reviews, respectively.

Figure 14: Word cloud for positive review data

Figure 15: Word cloud for negative review data

Immediately we can observe that the frequently occurring words are similar for both the negative and positive reviews. As expected, we find a lot of words that came up during the global analysis in the previous Section. Food and service are still the primary aspects of the restaurant related to the customers' experience. However we now see that these words occur in both positive and negative reviews. We can infer that these aspects of the customers' experience will be pivotal to their perception of the restaurant. If they have a poor experience related to food or service, they will give a poor rating, and vice versa. One interesting thing to note is that the words "good" and "great" are still occurring in both the positive and negative reviews. In particular, the word "good" is the second most frequent word within negative reviews. This might seem a little unintuitive at first. In order to get a better idea as to how this word is contextualized within the negative reviews, we can take a look at a random sample of the negative reviews containing the word "good":

```
set.seed(1)
tvec <-
  CapstoneDF_neg$Reviews[grepl("[Gg]ood", CapstoneDF_neg$Reviews)] %>%
  sample(5)
str_break(tvec[1])
```

```
## [1] "Horrible service!!!!! Food is over priced and wasn't even go"
## [2] "od small portions and was served cold !   I had the server b"
## [3] "ring all of our food at once after we told her we wanted the"
## [4] " salads to start. She also forgot about our drinks for about"
```

```
## [5] " half an hour ! Overall terrible experience and I probably w"
## [6] "on't go back !"
```

```
str_break(tvec[2])
```

```
##  [1] "We went because of the winterlicious, a friend and I.  For o"
##  [2] "ur starters ...she had Winter Kale Salad and I had Tuna Tost"
##  [3] "ada .Both were very tasty.For our mains, we had the Madras C"
##  [4] "hicken Sandwich and the  Shrimp Ravioli. The sauce they used"
##  [5] " on the sandwich was good, not as spicy as the waiter stress"
##  [6] "ed it would be.  And for the ravioli..not very impressed. It"
##  [7] " seemed pretty bland. I actually asked for salt which I neve"
##  [8] "r have never done before.Loved the desserts, toffee cake and"
##  [9] " salted caramel dark chocolate mousse. Too full to eat the c"
## [10] "ake so gave it away to my boss when I got back to the office"
## [11] ". The mousse was amazing.Would definitely come back again."
```

```
str_break(tvec[3])
```

```
## [1] "This is a very good looking place, with nice music played li"
## [2] "ve by a DJ adding atmosphere to the context, but service was"
## [3] " really dull (the restaurant was semi-desert and it took 'ag"
## [4] "es' to eat) and the food was not more than fair. A delusion,"
## [5] " considering expectations it creates. And clearly priced to "
## [6] "the look... and not to the food."
```

```
str_break(tvec[4])
```

```
##  [1] "On first impressions, the place is nice. Decor is awesome an"
##  [2] "d staff are well dressed and professional looking. This ends"
##  [3] " as soon as you are seated. I've been in the industry for 20"
##  [4] " years so I notice things and I know how things work. There "
##  [5] "is no excuse for poor service. We were staying in the area a"
##  [6] "nd decided to stop in here for a quick drink and a snack aft"
##  [7] "er a day of shopping. It was late afternoon and in between l"
##  [8] "unch and dinner rushes. We sat in the bar area and from what"
##  [9] " I could see there appeared to be more staff that patrons. W"
## [10] "e ordered drinks and an appetizer to share. Beers are 14oz, "
## [11] "not a standard 20oz pint but you pay 20oz prices. Beer temp "
## [12] "was off and beer was warm while my Caesar was weak and taste"
## [13] "less. There was a situation in the kitchen which slowed our "
## [14] "food down, the manager came over and explained and apologize"
## [15] "d which was perfectly acceptable, these things happen, I get"
## [16] " it. However, while we waited 20+ minutes for a simple appet"
## [17] "izer, we were never offered another round of drinks and sat "
```

```
## [18] "with empty glasses until we were able to flag someone down t"
## [19] "o request another drink! I'd like to add that there were sev"
## [20] "eral staff milling around the bar area in plain view chattin"
## [21] "g with what I can only assume were off duty staff sitting at"
## [22] " the bar. All while we sat with no food and no drinks!!!! Ou"
## [23] "r food was good, fresh and hot but tiny portion given the pr"
## [24] "ice. Undersized and overpriced was my general impression. Th"
## [25] "e service might have made up for it if anyone had bothered p"
## [26] "aying attention to us. Brutal. Don't waste your money here. "
## [27] "There are lots of other great places to eat around here, no "
## [28] "wonder it was empty on a Saturday afternoon."
```

Assuming that this sample is representative of the negative data subset, we can infer that the word "good" is used within the negative reviews primarily in two ways. The first is that the word good is negated so as to express negative sentiment, e.g. "wasn't good", "not good". The second way in which the word is used is one in which positive comments are made about certain aspects of the restaurant but other aspects are described negatively, leading to an overall negative review. E.g. "food was good but service was slow". This provides an explanation as to how the word "good" occurs so frequently in the negative reviews. It also tells us that all important aspects of a customer's experience at the restaurant, i.e. aspects related to food and service, must run smoothly in order for the guest to bestow a positive review. Examining the word clouds in Figures 14 and 15, we can see that, though "good" occurs frequently in both positive and negative reviews, the positive reviews often carry descriptors that express stronger positive sentiment, such as "great", "excellent", "amazing" and "delicious". Words such as these are not as present within the negative reviews. This comparative analysis is depicted graphically in Figures 16 and 17, which present the fraction of positive and negative reviews that contain the various frequent words. Figure 16 depicts the words that occur in both positive and negative data subsets, while Figure 17 depicts the words that are exclusive to either subset.

We can see from these figures that the word "great" occurs in 47% of the positive reviews and only in 26% of the negative reviews, while the word "good" occurs in approximately the same fraction of positive and negative reviews. A slightly larger portion of the negative reviews mention the word "food" compared to the positive reviews, while a larger portion of the positive reviews mention the word "service" compared to the negative reviews, implying that the restaurant's food might be a larger point of concern than its service. Additionally, in Figure 17, we see that the positive reviews frequently mention words like "staff" and "friendly" while the negative reviews do not, indicating that the quality of service has a large impact on the guests having a positive experience.

Now, as we did in the previous Section, we will take a look at some of the associations for the most frequent words.

```
#Positive reviews: "food"
findAssocs(CapstonePos_wA$TDM, "food", 0.15)
```

```
## $food
##   service     great      good   quality goodgreat     tasty
##      0.24      0.24      0.23      0.21      0.17      0.15
```

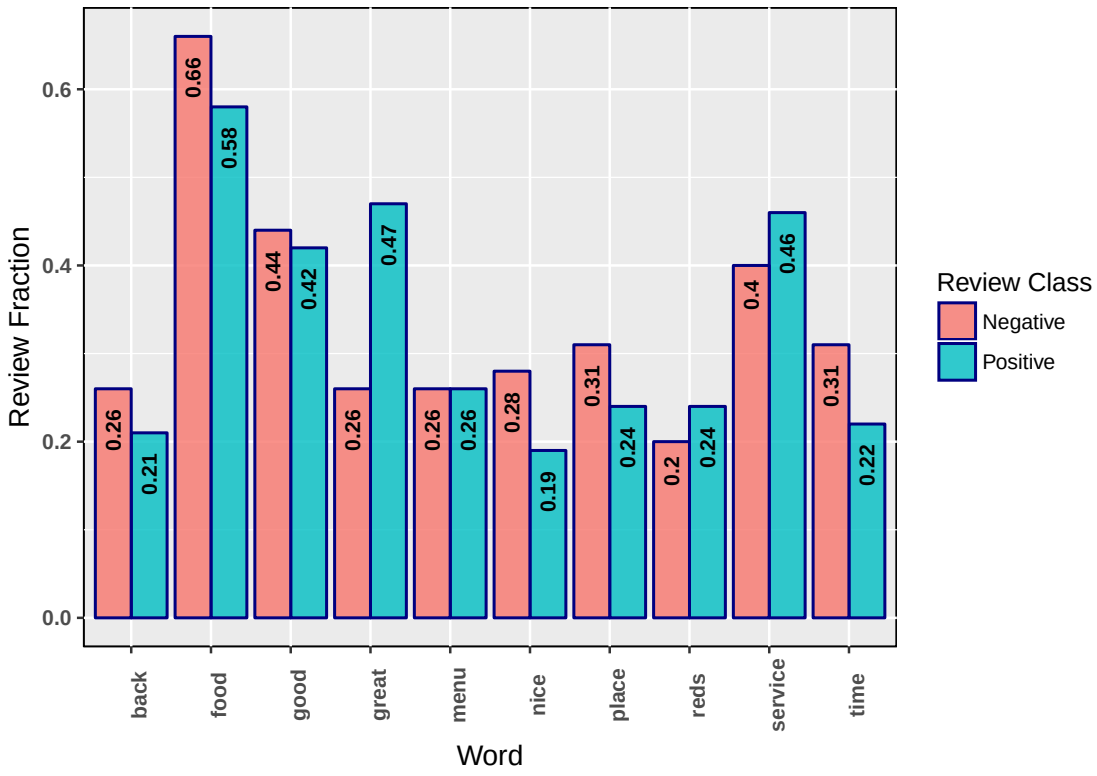

Figure 16: Fraction of frequent words included in positive and negative reviews

```
#Negative reviews: "food"
findAssocs(CapstoneNeg_wA$TDM, "food", 0.16)
```

```
## $food
##   service      good      fast orderwhen starvingi   payment     lousy
##      0.22      0.21      0.21      0.19      0.19      0.19      0.19
##   wrongit     basic      blah  beverage
##      0.19      0.17      0.17      0.17
```

We find that the word "food" is commonly associated with positive descriptors like "great", "quality", and "tasty" within the positive reviews, whereas within the negative reviews it is associated to words like "lousy", "basic", and "blah". Given the importance of food to a customer's experience, it should be expected that the quality of the food served will have a tremendous impact. Another food-related word that we can examine is the word "menu", which shows up in 26% of both the positive and negative reviews.

```
#Positive reviews: "menu"
findAssocs(CapstonePos_wA$TDM, "menu", 0.15)
```

```
## $menu
## winterlicious         items          fixe          prix      selected
```

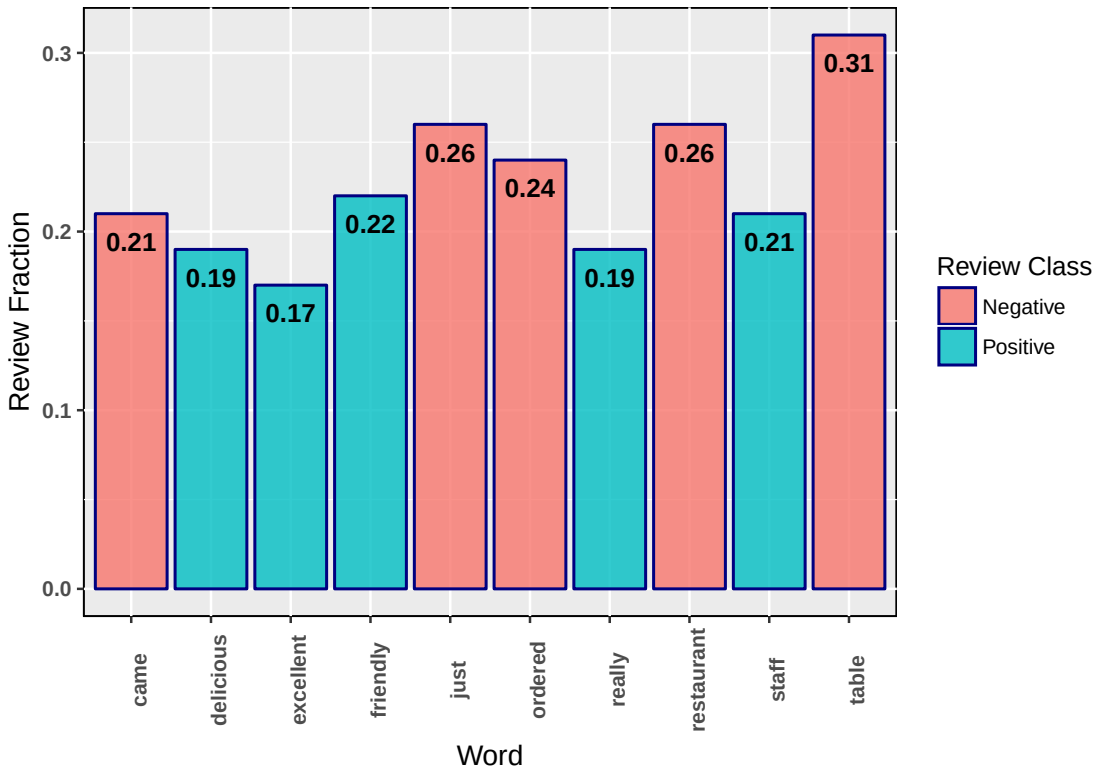

Figure 17: Fraction of frequent words exclusive to positive or negative reviews

```
##          0.23          0.22          0.18          0.18          0.17
##      activity     featuring   promotional      yearmost
##          0.17          0.17          0.17          0.17
```

```
#Negative reviews: "menu"
findAssocs(CapstoneNeg_wA$TDM, "menu", 0.15)
```

```
## $menu
##          quit knowledgeable    reasonably        varied     initially
##          0.33          0.25          0.23          0.23          0.22
##        opted  descriptions     specifics         terms      training
##          0.22          0.22          0.22          0.22          0.22
## summerlicious         items      personal          seem
##          0.19          0.17          0.16          0.15
```

The strongest association to the word "menu" within the positive reviews is the word "winterlicious", which refers to a food festival that occurs during the winter in Toronto. During this festival, a large number of restaurants create prix-fixe menus that showcase some of the restaurant's food. Additional associations to the word "menu" within the positive reviews include the word "fixe" and "prix", which may refer to the Winterlicious menu, but might also suggest a reference to the relatively new prix-fixe menu at RMT. These associations imply that customers have had a positive experiences with these different

non-standard menus at RMT. The associations to "menu" in the negative reviews are not as immediately descriptive, though interestingly, the word "summerlicious" makes an appearance, suggesting that the Summerlicious prix-fixe menus have not been received as favourably as their winter counterpart.

Next we will take a look at the associations for words related to the restaurant's service, such as "service", "staff" and "server", beginning with the positive reviews.

```
findAssocs(CapstonePos_wA$TDM,
           c("service", "staff", "server"),
           c(0.15,0.19,0.16))
```

```
## $service
##      food     great      slow  friendly goodgreat excellent  customer
##      0.24      0.21      0.20      0.20      0.20      0.19      0.19
## attentive
##      0.16
##
## $staff
##          friendly              wait        management            engage
##              0.38              0.29              0.21              0.20
##        interacted        2defiantly          constant           greater
##              0.20              0.20              0.20              0.20
##        companions countdownfriendly               nye          pleasure
##              0.20              0.20              0.20              0.20
##          shouting
##              0.20
##
## $server
##        our  attentive     helped    michael   requests delightful
##       0.39       0.23       0.20       0.19       0.16       0.16
```

From this output, we see that the word "slow" is one of the words that is most associated with the word "service". This indicates that the speed of the service at RMT is an important point of concern. Beyond this however, "service" is associated with words indicating strong positive sentiments such as "great" and "excellent", as well as words that speak to the quality of the service, such as "friendly" and "attentive". These words are also associated with the words "staff" and "server". Other words that show up are "engage", "interacted", and "helped". These associations suggest that customers have a positive experience related to service when the waitstaff is present in their interactions with the customer, is friendly towards them, and is willing to go above and beyond for them. Next we will look into the negative reviews.

```
findAssocs(CapstoneNeg_wA$TDM,
           c("service", "staff", "server"),
           c(0.19, 0.2, 0.2))
```

```
## $service
##        slow        food hospitality personality    great2nd      okayre
##        0.34        0.22        0.20        0.20        0.19        0.19
##    outgoing
##        0.19
##
## $staff
## chatting     duty  milling     view    plain      add friendly
##     0.33     0.33     0.33     0.33     0.24     0.23     0.21
##
## $server
##       biggy differently    partners    switched       versa        vice
##        0.26        0.26        0.26        0.26        0.26        0.26
##        bill
##        0.21
```

We find once again that the word "slow" is strongly associated with "service". Most of the words that show up in this case are not as informative as the ones for the positive reviews, suggesting that the customers' experience related to service wasn't extremely terrible, but wasn't anything special either. Associated with the word "staff" however are words such as "chatting" and "milling", indicating experiences in which the customers observed the waitstaff socializing and being inattentive rather than tending to their guests.

In this Section we extracted the main insights from the online customer reviews, both globally as well as in negative and positive reviews. In the following Section, I will summarise the work that I have done and provide some recommendations and potential directions for future work.

5 Summary, Recommendations and Future Directions

Over the course of this report I have elaborated on some of the tools and methods that can be used in R to gather, process and analyze data from the Internet. Specifically, I applied these techniques with the aim of performing an analysis of the customer review data for REDS Midtown Tavern in Toronto, Ontario. This analysis was performed using data from the travel and review websites Yelp, OpenTable, TripAdvisor and Zomato, resulting in a data set of 677 online reviews, numerical ratings, and review dates. In Section 2, we began by exploring the uses of the SelectorGadget tool and CSS Selectors to identify content on the web for data acquisition. We then discussed how to load this data into R using the `rvest` package. Following this, we spent time in Section 3 elaborating on the process of cleaning the raw data from the Internet in order to have it in a format suitable to analysis. Finally, the clean data was analyzed in Section 4, where we performed first a time series analysis of the numerical ratings, and subsequently a text and sentiment analysis of the written customer reviews. In analyzing the online review data for RMT, we were able to derive a number of insights. In particular, our exploration of the data was framed by the following business questions:

- How has RMT performed over the years? What periods of time have been associated

with positive, negative reviews? How has the restaurant's performance changed over time? How is it performing now?
- What are the restaurant's overall strengths? What are its weaknesses?
- What are the most important aspects of the customer experience? How can we improve upon these?

We were able to find some answers to these questions by analyzing the web data. In Section 4.1, we examined in detail how the restaurant's ratings changed over time. The overall average rating for RMT was 3.79. Looking at the yearly data, we found that the ratings began around this average value, hit their lowest value in 2015 and are now at all-time high in 2017, with a mean rating of 4.1. The quarterly ratings data gave us more insight into this, where we observed no real movement in 2014, an initial surge followed by a steady decline in 2015, reaching a low in 2015-Q4, and a subsequent period of tremendous growth in 2016. The restaurant's highest quarterly ratings were in 2016-Q4. The first quarter of 2017 has been met with a slight decline in ratings. The monthly ratings data allowed us to observe that this decline has been due to poor ratings in March 2017. In the remainder of Section 4.1 we examined the different distributions of ratings over the years. In particular, we found that since 2016, the ratings have been characterized by an increasing fraction of 5-star ratings and a decreasing fraction of 1-, 2-, and 4-star ratings. This is positive as it indicates that more customer are rating the restaurant favourably. The significant improvements in ratings made in 2016 are a positive sign for the restaurant's future. Ideally we would like to extend this trend into the future by continuously improving the important aspects of customer experience. Another fairly positive outcome is one in which the ratings in the near future remain stable at a new normal value of around 4.1. The ratings for the month of March 2017 however suggest that we are facing the possibility of a new downward trend. Another possible way to interpret this lower average is as an outlier in the data. This is not unheard of as something similar happened in June 2014. Ultimately, the future trends of the restaurant's ratings will depend on the decisions that are made at RMT. It is up to the management and the staff at the restaurant whether this recent low becomes an outlier, or the beginning of a negative trend. In order to ensure a favourable outcome, such as a continuing upward trend or a period of stability around a new high, it is important to improve upon the restaurant's weaknesses and to effectively leverage its strengths. These different aspects of the restaurant can be hinted at by mining the text data from the customer reviews.

In Section 4.2, we performed a text and sentiment analysis of the customer review data in order to derive insights pertaining to the positive and negative aspects of the customer experience at RMT, as well as the restaurant's strengths and weaknesses. We began with a global analysis of the full set of review data, in which we found that the two most important aspects of a guest's experience at the restaurant are the food and the service. Positive or negative experiences related to these two aspects will make or break a customer's experience. Overall, the sentiment associated with these aspects of the restaurant was positive. The food was associated with words such as "good", "great", and "quality", and the service with words such as "great", "friendly" and "excellent". We did find however that the service was also strongly associated with the word "slow", suggesting that improvements can be made in this regard. Some positive sentiments were additionally associated with words like "atmosphere" and "decoration", implying that the customer

enjoys the physical aspects of RMT. After performing an analysis on the full data set, we subsequently divided the data into positive (> 3 stars) and negative (≤ 3 stars) reviews to mine for deeper insight. The positive data subset comprised 68% of the full data set, while the negative subset comprised 32%. In parsing the data in this way, we found that food and service were still the primary aspects of RMT that were being commented on. These were the aspects of the customers' experience that, when executed properly lead to positive reviews, and when executed poorly lead to negative reviews. During the analysis, we also found that positive sentiments were associated with special menus at RMT such as the Winterlicious menu or the more recent prix-fixe menu. In terms of the service, positive reviews contained associations to words such as "friendly", "attentive", "engage", "delightful". Negative reviews, though they still presented an association of service to "friendly", also resulted in associations to words like "chatting" and "milling". These results, combined with association of the word "slow" with service in both positive and negative reviews, suggest that customers have a positive experience when they feel they are being tended to and given positive attention, and a negative experience when they feel ignored, or when they feel the waitstaff is preoccupied with something other than their work. From the comparative analysis of positive and negative reviews, we also found that the word "food" was associated with a larger fraction of negative reviews, while the word "service" was associated with larger fraction of positive reviews. Additionally, a significant fraction of positive reviews mentioned words like "staff" and "friendly", while these words were not present in as significant a fraction of negative reviews. Finally, a significant fraction of positive reviews contained words expressing greater positive sentiments, such as "great", "excellent", "really", and "delicious". The abundance of positive reviews over negative reviews and the significant fractions of positive reviews containing words related to service suggest that the service at RMT is one of its greatest strengths. This is especially true in terms of how the waitstaff relate to the guests. The one aspect of service that has been tagged for clear improvement is the speed. Slow service appears to be a common complaint that is put forth by the customers. The other point of improvement related to service the work-orientation of the employees. When it comes to food, the reviews are mixed. Some of the customers have had positive experiences related to the food at RMT, while others have had negative experiences. A point of focus for the culinary aspects of RMT would be to ensure that the quality is consistent regardless of the kitchen staff and menu. In order to ensure that the ratings and review for RMT continue to exhibit a positive trend, it is important to make decisions that will have a positive impact on the restaurant. To that end I will make a number of recommendations based on the results obtained from the data analysis.

1. Ensure that the quality of the food is consistently great. One way of doing this could be by randomly sampling different items on the menu, taking detailed notes, and tracking the performance of the kitchen staff responsible for preparing those food items.
2. Address the slowness of service by ensuring that times associated with different steps of service are strictly met. Guests should not be waiting to be greeted or to place an order. The use of an expeditor in the kitchen will be useful for ensuring that bill times are met. This is especially important in down times, when staff tends to move more slowly.

3. Friendliness and approachability of the waitstaff is one of RMT's greatest strengths. Implement a program that focuses on developing these qualities in the staff.
4. Keep providing RMT's prix-fixe menu as this is associated with positive reviews.
5. RMT has strong brand recognition and trust from clientele. It might be beneficial to engage in below the line marketing campaigns to reach target customers, while providing a loyalty program to those who visit regularly.
6. Ensure that the waitstaff is performing as effectively as they could be by communicating effectively and providing critical feedback to ensure continued improvement. Keep track of how staff are performing, address concerns regarding poor performance as they arise, and maintain strict standards.

Using recommendations such as these, RMT will hopefully experience 2017 as a period of continued improvement.

This project has allowed me to develop the ground work for acquiring, cleaning, and analyzing customer reviews from a variety of websites. I have taken the time to analyze certain aspects of the review data, such as the evolution of the restaurant's ratings and common sentiments expressed in reviews over the restaurant's history. Given the volume of the available data, there are still a lot of ways in which we might analyze the data to mine for deeper insight. Future directions related to this project might involve a text analysis of specific periods of time in the restaurant's history in order to better understand trends in the time series data. For instance, we might strive to answer the following questions: Why were the ratings low in June 2014? Why were they low in 2015? What aspects of the customer experience were associated with the improvements in 2016? Why did the quality of the ratings fall in March 2017? In addition to this, it might be informative to perform a comparative analysis of the review data for RMT and that for its sister restaurant, REDS Wine Tavern, or one of its close competitors, such as Scaddabush. This could provide insight into what is working for these other restaurants that might also work for RMT. Another, more technical, direction might be to develop a logistic regression model to predict the numerical rating of a customer review based on the existence of certain words within the review.

Write closing remarks