Project Proposal Data Mash ups in R

Adejare Windokun

Friday, December 12, 2014

Describes your motivation for performing this analysis.

Data Mashups whereby data is obtained from different sources (heterogeneous data) and merged together to provide to a more comprehensive view of the data is of huge advantage especially with the ease of accessing data over the internet, the ability to query and use third party tools including APIs and the advent of huge datasets made available by both the government and private entities. During the semester, while I have worked on single datasets, I haven't had the opportunity to work on multiple diverse datasets, with all the complexities that this entails - varied data formats, incomplete data, need for data transformations, and especially the need to work with third party APIs. This project will provide me the opportunity to do all these.

Data science workflow.

My datasets consists of:

1. A subset of the Amazon Reviews which is hosted at http://snap.stanford.edu/data/web-Amazon.html, specifically the dataset "Gourmet_Foods". This dataset is formatted as such:

   product/productId: asin, e.g. amazon.com/dp/B00006HAXW product/title: title of the product product/price: price of the product review/userId: id of the user, e.g. A1RSDE90N6RSZF review/profileName: name of the user review/helpfulness: fraction of users who found the review helpful review/score: rating of the product review/time: time of the review (unix time) review/summary: review summary review/text: text of the review

where by a single row of data (item) is spread across 10 lines with a space in between subsequent items. Data transformation which has to be performed include extracting the data, limiting the dataset to manageable number, transforming the data into a table structure, cleaning the data and removing the unwanted labels, transforming columns in the dataset (covert the time column from character to POSIXct() which R understands.

2. Reviewer data from Amazon.com using the R packages rvest and XML, which is returned in HTML format which has to be parsed to extract the relevant data which further has to be processed and converted to the required data type suitable for use in R.

3. Sentiment Analysis using AlchemyAPI (AlchemyAPI.com). Sentiment analysis was performed on both the summary and text fields which are reviews that the reviewer

made on each item. This data has to be sent to AlchemyAPI for analysis using HTML, and the result which is returned in the JSON format has to be parsed to extract the relevant data, followed by data transformation to the required data type in R for further processing.

Project includes data from at least two different types of data sources (e.g., two or more of these: relational, CSV, Neo4J, web page, MongoDB, etc.)

My project includes (a) data from a flat file (Amazon Reviews from snap.stanford.edu), (b) live from webpages (amazon.com - http://www.amazon.com/gp/cdp/member-reviews/userID) and (c) using an API from AlchemyAPI.com

Project includes at least one data transformation operation. [Examples: transforming from wide to long; converting columns to date format]

Each of the data sources required extensive data transformation before it could be used in R

Project includes at least one statistical analysis and at least one graphics that describes or validates your data.

Used linear regression in the corellation

Project includes at least one graphic that supports your conclusion(s).

plotted residuals vs the fitted from the linear regression

Project includes at least one statistical analysis that supports your conclusion(s).

My project looks at the correlation between "Helpfullness" and "Reviewer Scores" for items and the correlation between Sentiment Analysis on "Summary Text" and "Reviewes"

Project includes at least one feature that we did not cover in class! There are many examples:

Features not covered in class: 1. AlchemyAPI (http://www.alchemyapi.com/) 2. R packages a. Stringr b. Rvest c. RJSON d. Sqldf e. Rcurl

Presentation. Did you show (at least) one challenge you encountered in code and/or data, and what you did when you encountered that challenge? If you didn't encounter any challenges, your assignment was clearly too easy for you!

There were multiple unexpected challenges that occurred during this project. 1. Obtaining the data. While the Amazon Review dataset is supposed to be public at http://snap.stanford.edu, one has to request permission from Stanford and this caused a delay. 2. AlchemyAPI provides APIs to use their services, and even instructions on how these APIs work. However, they do not provide one for R, and therefore I was left to use the interface that they provide through REST. This unfortunately, was not well documented and required the manipulation of the URL, and much more manipulation of the return dataset which was in JSON format. The use of the API, for example makes submitting and extracting data a simple two line process, which however in R using the REST interface took multiple lines of code and the use of multiple R packages. 3. Limits on data extraction using the web interface: Querying Amazon.com for reviewers other reviews was a slow process due to the fact that the query had to be sent to amazon.com using the "post" service in HTML, and the return in HTLM had to be parsed to obtain the necessary information. Also, sending the requests too fact resulted in a server denial by the amazon servers - hence I had to build in a pause between each request. 4. Limits on allowable use AlchemyAPI. AlchemyAPI only allows 1000 request per day from a free account . Each of my rows of data required two requests, and therefore I was limited to only being able to obtain data on 500 rows per day. In addition, too frequent requests led to a denial of service, hence I had to insert a pause between requests. Additionally, using the REST interface led to unexpected errors (as compared to using the API) and this had to be taken into account and coded for. 5. Inconsistent data formats. The amazon.com website does not provide a consistent format and therefore parsing the HTML return data was a slow process and had to account for this inconsistency. This led to much more coding and error handling than should have been necessary.

Code

```
if(!require(stringr)) install.packages("stringr")
```

```
## Loading required package: stringr
```

```
library(stringr)
if(!require(RCurl)) install.packages ("RCurl")
```

```
## Loading required package: RCurl
## Loading required package: bitops
```

```
library(RCurl)
if(!require(rjson)) install.packages ("rjson")
```

```
## Loading required package: rjson
```

```
library(rjson)
if (!require(devtools)) install.packages('devtools')
```

```
## Loading required package: devtools
## WARNING: Rtools is required to build R packages, but is not currently 
installed.
## 
## Please download and install Rtools 3.1 from http://cran.r-
project.org/bin/windows/Rtools/ and then run find_rtools().
```

```
library(devtools)
if (!require(rvest)) install_github("hadley/rvest")
```

```
## Loading required package: rvest
```

```
library(rvest)
if (!require(XML)) install.packages('XML')
```

```
## Loading required package: XML
```

```
library(XML)
if(!require(sqldf)) install.packages ("sqldf")
```

```
## Loading required package: sqldf
## Loading required package: gsubfn
## Loading required package: proto
## Loading required package: RSQLite
## Loading required package: DBI
```

```
library(sqldf)

if(!require(ggplot2)) install.packages ("ggplot2")
```

```
## Loading required package: ggplot2
```

```
library(ggplot2)

x <-
"https://raw.githubusercontent.com/jwindokun/FinalProject/master/data/gourmet
Foods_test.txt"

try ({

line=readLines(x, n = -1L, ok = TRUE, warn = FALSE)

})
```

```
## Warning in file(con, "r"): unsupported URL scheme
```

```
try({
df <- data.frame(productID =character(), title = character(), price =
as.numeric(character()), userID =character(), profileName = character(),
                 helpfulness = character(), score = as.numeric(character()),
time = character(), summary = character(), text =
character(),stringsAsFactors=FALSE)


n = 1
d <- data.frame(productID =character(), title = character(), price =
as.numeric(character()), userID =character(), profileName = character(),
                helpfulness = character(), score = as.numeric(character()),
time = character(), summary = character(), text =
character(),stringsAsFactors=FALSE)

}, silent = TRUE)
#for (i in 1:length(line)){
# for the purpose of demonstration will only read in the first 10000 lines of
```

```
the file
for (i in 1:1000){

try({
 w = str_split_fixed(line[i], ":", 2)
    if (!is.na(w[1])){
      if (n < 11) {
        d[1, n] = str_trim(w[2])
        n = n + 1
        } else {
        df = rbind(df, d)
        n = 1
      }
    }

}, silent = TRUE)
  }

#clean the data

str(df)

## 'data.frame':    90 obs. of  10 variables:
##  $ productID  : chr  "B000EVS4TY" "B0000DF3IX" "B0002QF1LK" "B0002QF1LK"
...
##  $ title      : chr  "Arrowhead Mills Cookie Mix, Chocolate Chip, 12.9-
Ounce Units (Pack of 6)" "Paprika Hungarian Sweet" "Quaker Honey Graham Oh's
10.5 oz - (6 pack)" "Quaker Honey Graham Oh's 10.5 oz - (6 pack)" ...
##  $ price      : chr  "unknown" "unknown" "26.82" "26.82" ...
##  $ userID     : chr  "A2SRVDDDOQ8QJL" "A244MHL2UN2EYL" "A3FL7SXVYMC5NR"
"A12IDQSS4OW33B" ...
##  $ profileName: chr  "MJ23447" "P. J. Whiting \"book cook\"" "Brittany"
"Robin Goodfellow" ...
##  $ helpfulness: chr  "2/4" "0/0" "3/3" "3/3" ...
##  $ score      : chr  "4.0" "5.0" "5.0" "5.0" ...
##  $ time       : chr  "1206576000" "1127088000" "1138147200" "1118016000"
...
##  $ summary    : chr  "Delicious cookie mix" "Sweet Paprika: A sweet
ingredient!" "Best Cereal BY FAR" "Oh!" ...
##  $ text       : chr  "I thought it was funny that I bought this product
without knowing it was a mix. I read the header very quickly and just
thought"| __truncated__ "While in Hungary we were given a recipe for
Hungarian Goulash. It needs sweet paprika. This was terrific in that dish and
other"| __truncated__ "Without a doubt, I would recommend this wholesome and
sweet cereal treat to anyone. The crunchy o-shaped pieces are filled with"|
__truncated__ "This cereal is so sweet....yet so good for you! One
taste=ADDICTION!!!! I just tried this cereal out of curiousity and I was ho"|
__truncated__ ...
```

```r
df$time = as.POSIXct(as.numeric(df$time), origin="1970-01-01")
df$price[df$price == "unknown"] = 0
df$price = as.numeric(df$price)
df$score = as.numeric(df$score)

for (i in 1 :length(df$helpfulness)){
  try({

  e = df$helpfulness[i]
  e = strsplit(e, split = "/", fixed = TRUE)[[1]]
  e = as.numeric(e[1])/as.numeric(e[2])
  if (is.finite(e)) {
        df$helpfulness[i] = e
    } else {

        df$helpfulness[i] = 0
    }

}, silent = TRUE)
}

df$helpfulness = as.numeric(df$helpfulness)
str(df)
```

```
## 'data.frame':    90 obs. of  10 variables:
##  $ productID  : chr  "B000EVS4TY" "B0000DF3IX" "B0002QF1LK" "B0002QF1LK"
...
##  $ title      : chr  "Arrowhead Mills Cookie Mix, Chocolate Chip, 12.9-
Ounce Units (Pack of 6)" "Paprika Hungarian Sweet" "Quaker Honey Graham Oh's
10.5 oz - (6 pack)" "Quaker Honey Graham Oh's 10.5 oz - (6 pack)" ...
##  $ price      : num  0 0 26.8 26.8 26.8 ...
##  $ userID     : chr  "A2SRVDDDOQ8QJL" "A244MHL2UN2EYL" "A3FL7SXVYMC5NR"
"A12IDQSS4OW33B" ...
##  $ profileName: chr  "MJ23447" "P. J. Whiting \"book cook\"" "Brittany"
"Robin Goodfellow" ...
##  $ helpfulness: num  0.5 0 1 1 1 ...
##  $ score      : num  4 5 5 5 3 5 5 5 5 4 ...
##  $ time       : POSIXct, format: "2008-03-26 20:00:00" "2005-09-18
20:00:00" ...
##  $ summary    : chr  "Delicious cookie mix" "Sweet Paprika: A sweet
ingredient!" "Best Cereal BY FAR" "Oh!" ...
##  $ text       : chr  "I thought it was funny that I bought this product
without knowing it was a mix. I read the header very quickly and just
thought"| __truncated__ "While in Hungary we were given a recipe for
Hungarian Goulash. It needs sweet paprika. This was terrific in that dish and
other"| __truncated__ "Without a doubt, I would recommend this wholesome and
sweet cereal treat to anyone. The crunchy o-shaped pieces are filled with"|
__truncated__ "This cereal is so sweet....yet so good for you! One
taste=ADDICTION!!!! I just tried this cereal out of curiousity and I was ho"|
__truncated__ ...
```

```
df[1,]

##    productID
## 1 B000EVS4TY
##                                                                       title
## 1 Arrowhead Mills Cookie Mix, Chocolate Chip, 12.9-Ounce Units (Pack of 6)
##   price         userID profileName helpfulness score                time
## 1     0 A2SRVDDDOQ8QJL     MJ23447         0.5     4 2008-03-26 20:00:00
##                summary
## 1 Delicious cookie mix
##
text
## 1 I thought it was funny that I bought this product without knowing it was
a mix. I read the header very quickly and just thought it was packaged
cookies. But no, it is cookie MIX and I guess I should have noticed that
since it is right in the title.This is the first time I have ever tried
baking with a cookie mix. If you are used to the convenience of the cookie
dough that you buy wrapped up in plastic logs then you might be in for a bit
of a surprise. Mixing up the dough can get VERY messy (it is extremely
sticky). However, with a cookie mix like this you have a lot of flexibility
in the ratio of ingredients (I like to add some extra butter to make the
baked cookies more chewy). Also, this mix has really large chocolate chips in
it--I love that.I removed a star for the addition of 'natural flavors' in the
mix.

githubURL =
"https://github.com/jwindokun/FinalProject/raw/master/data/gourmetFoods.RData
"


try ({

  w = load(url(githubURL))

})

## Warning in load(url(githubURL)): unsupported URL scheme

api_key = "45319ace2f7c43fc55d05c2f973ba6261a5de4f0"



api =
paste("http://access.alchemyapi.com/calls/text/TextGetTextSentiment?apikey=",
api_key, "&outputMode=json", "&text=", sep = "")

gourmetFoodsAPI = gourmetFoods

gourmetFoodsAPI[1,]

##    productID
## 1 B000EVS4TY
##                                                                       title
## 1 Arrowhead Mills Cookie Mix, Chocolate Chip, 12.9-Ounce Units (Pack of 6)
##   price         userID profileName helpfulness score                time
## 1     0 A2SRVDDDOQ8QJL     MJ23447         0.5     4 2008-03-26 20:00:00
##              summary
## 1 Delicious cookie mix
##
text
## 1 I thought it was funny that I bought this product without knowing it was
a mix. I read the header very quickly and just thought it was packaged
cookies. But no, it is cookie MIX and I guess I should have noticed that
since it is right in the title.This is the first time I have ever tried
baking with a cookie mix. If you are used to the convenience of the cookie
dough that you buy wrapped up in plastic logs then you might be in for a bit
of a surprise. Mixing up the dough can get VERY messy (it is extremely
sticky). However, with a cookie mix like this you have a lot of flexibility
in the ratio of ingredients (I like to add some extra butter to make the
baked cookies more chewy). Also, this mix has really large chocolate chips in
it--I love that.I removed a star for the addition of 'natural flavors' in the
mix.
```

```
#for i in 1:nrow(gourmetfoodsAPI){
for (i in 1:2){

    if (length(gourmetFoodsAPI$text[i]) > 0) {

        phrase = URLencode(gourmetFoodsAPI$text[i])
        api_url = paste(api, phrase, sep="")
        result = getURI(api_url)
        r = fromJSON(result)
        gourmetFoodsAPI$apitextType[i] =
ifelse(!is.null(r$docSentiment$type), r$docSentiment$type, "")
        gourmetFoodsAPI$apitextScore[i] =
ifelse(!is.null(r$docSentiment$score), r$docSentiment$score, 0)
    }

    if (length(gourmetFoodsAPI$summary[i]) > 0) {
        phrase = URLencode(gourmetFoodsAPI$summary[i])
        api_url = paste(api, phrase, sep="")
        result = getURI(api_url)
        r = fromJSON(result)
        gourmetFoodsAPI$apisummaryType[i] =
ifelse(!is.null(r$docSentiment$type), r$docSentiment$type, "")
        gourmetFoodsAPI$apisummaryScore[i] =
ifelse(!is.null(r$docSentiment$score), r$docSentiment$score, 0)
    }

    Sys.sleep(2)
}
```

```
gourmetFoodsAPI[1,]

##    productID
## 1 B000EVS4TY
##                                                                       title
## 1 Arrowhead Mills Cookie Mix, Chocolate Chip, 12.9-Ounce Units (Pack of 6)
##   price         userID profileName helpfulness score                time
## 1     0 A2SRVDDDOQ8QJL     MJ23447         0.5     4 2008-03-26 20:00:00
##                 summary
## 1 Delicious cookie mix
##
text
## 1 I thought it was funny that I bought this product without knowing it was
a mix. I read the header very quickly and just thought it was packaged
cookies. But no, it is cookie MIX and I guess I should have noticed that
since it is right in the title.This is the first time I have ever tried
baking with a cookie mix. If you are used to the convenience of the cookie
dough that you buy wrapped up in plastic logs then you might be in for a bit
of a surprise. Mixing up the dough can get VERY messy (it is extremely
sticky). However, with a cookie mix like this you have a lot of flexibility
in the ratio of ingredients (I like to add some extra butter to make the
baked cookies more chewy). Also, this mix has really large chocolate chips in
it--I love that.I removed a star for the addition of 'natural flavors' in the
mix.
##   apitextType apitextScore apisummaryType apisummaryScore
## 1    positive     0.262573       positive        0.482862

githubURL =
"https://github.com/jwindokun/FinalProject/raw/master/data/gourmetFoods.RData
"


try ({

  load(url(githubURL))

})

## Warning in load(url(githubURL)): unsupported URL scheme

dfAmazon <- data.frame(userID = character(), reviewerName =  character(),
numReviews = numeric(), itemName = character(),
                       itemPrice = numeric(), date = character(), text =
character(),stringsAsFactors=FALSE)

# For demonstration purposes will only use the first 10 rows
#for (i in length(gourmetFoods)) {
for (i in 10) {

    try({
```

```r
    userID = gourmetFoods$userID[i]
    website = paste("http://www.amazon.com/gp/cdp/member-reviews/", userID,
sep ="")
    r_site <- html(website)
    r <- r_site %>%
    html_nodes(".small") %>%
    html_text() %>%
    gsub("[\t\n\r\f\v]", "", .)

    #Get the reviewer name
    df <- data.frame(userID = character(), reviewerName =  character(),
numReviews = numeric(), itemName = character(),
                     itemPrice = numeric(), date = character(), text =
character(),stringsAsFactors=FALSE)

    # First line contains reviewer information
    reviewerName = substr(str_trim(r[1]), start = 13, stop =
(nchar(str_trim(r[1])) -10))

    # Line 3 contains the total number of reviewsreviews
    numReviews = as.numeric(strsplit(str_trim(r[3]), split = " ", fixed =
TRUE)[[1]][3])


    # get the items reviewed

    # Line 16 and (with an interval of 8) contains information on the item
reviewed)
    n = 1
    for (i in seq(16,length(r),8)){

        df[n, "userID"] = ifelse(!is.null(userID), userID, "")

        df[n, "reviewerName"] = ifelse(!is.null(reviewerName), reviewerName,
"")
        df[n, "numReviews"] =ifelse(!is.null(numReviews), numReviews, 0)

        t = str_split_fixed(str_trim(r[i]), ":", 2)

        while (nchar(t[1]) < 4){

          t = str_split_fixed(str_trim(r[i + 1]), ":", 2)
          i = i+1

        }

        t = str_split_fixed(str_trim(r[i]), "Price:", 2)
        df[n, "itemName"] = ifelse(!is.null(t[1]), t[1], "")
        #df[n, "itemPrice"] =ifelse(!is.null(t[2]),
as.numeric(str_sub(str_trim(t[2]),2)), 0)
        p = as.numeric(str_sub(str_trim(t[2]),2))
        f <- function(x) is.numeric(x) & !is.na(x)
        #print (f(p))
        df[n, "itemPrice"] =ifelse(f(p), p, 0)
        n = n + 1

    }


    # Line 20 and (with an interval of 20 contains information on the date of
the review, and the text of the review)
    n = 1
    for (i in seq(20,length(r),8)){

        gDate <- "(([[:alpha:]]+)([[:space:]])([0-
9]{1,2})([,])([[:space:]])([0-9]{4}))"
        strings <- str_trim(r[i])

        date = str_extract(strings, gDate)

        while (is.na(date)){

          strings <- str_trim(r[i+1])
          date = str_extract(strings, gDate)
          i = i+1

        }


        #date = as.Date(str_extract(strings, gDate), "%B %d, %Y")
        #print (date)
        f = function (x) is.na(as.Date(as.character(x),format="%B/%m/%Y"))


        df[n, "date"] = ifelse(f(date), date, "")


        s = str_split_fixed(str_trim(r[i]), ":", 2)[2]
        s = str_trim(str_split_fixed(s, ")", 2)[2])
        df[n, "text"] = ifelse(!is.null(s), s, "")

        n = n + 1
    }

    dfAmazon = rbind(dfAmazon, df)
    Sys.sleep (2)

    }, silent = TRUE)

}
```

```
## Warning: NAs introduced by coercion
```

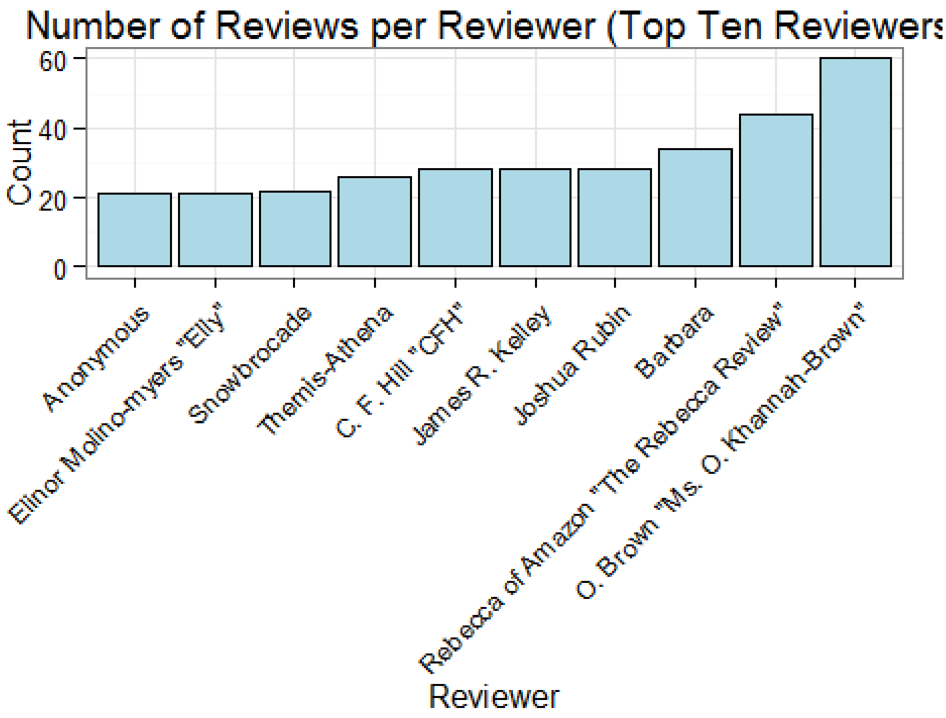
```
str(dfAmazon)
```

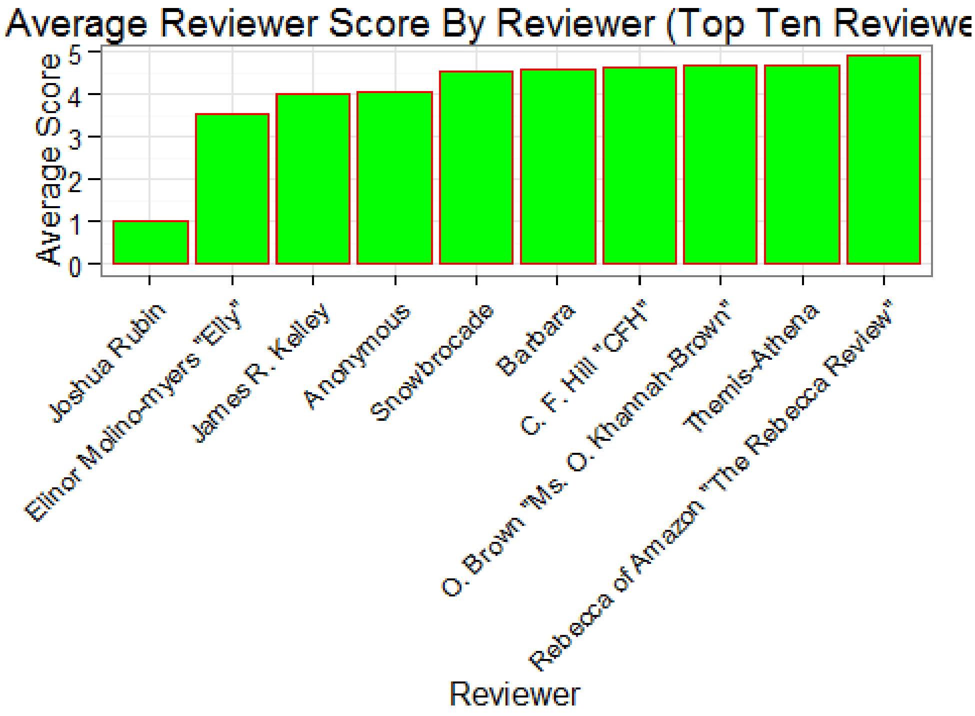
```
## 'data.frame':    10 obs. of  7 variables:
##  $ userID      : chr  "A3IY9HIAMJQ7HL" "A3IY9HIAMJQ7HL" "A3IY9HIAMJQ7HL"
"A3IY9HIAMJQ7HL" ...
##  $ reviewerName: chr  "Marcel Lee" "Marcel Lee" "Marcel Lee" "Marcel Lee"
...
##  $ numReviews  : num  289 289 289 289 289 289 289 289 289 289
##  $ itemName    : chr  "Sharp On All 4 Corners (Deluxe Edition) [Explicit]"
"PRhyme [Explicit]" "A Better Tomorrow [Explicit]" "ShadyXV [Explicit]
[+digital booklet]" ...
##  $ itemPrice   : num  17.49 8.99 11.49 15.49 2.99 ...
##  $ date        : chr  "December 11, 2014" "December 9, 2014" "December 2,
2014" "November 24, 2014" ...
##  $ text        : chr  "[Explicit] (MP3 Music)      Thereâ□□s a track here
entitled Heavy In The Game. The song itself, a duet with B-Legit, is
nothing"| __truncated__ "The music is provided by DJ Premier. That's a beat
check. Royce Da 5-9 also gets a verse check for his raps, though it must be
"| __truncated__ "The main problem I have with this album is the title. Wu-
Tang Clan (members) made a song called A Better Tomorrow back in 1997."|
__truncated__ "If there were any doubt, you need not go further than the
first song on this set to establish the fact that Eminem can still ra"|
__truncated__ ...
```

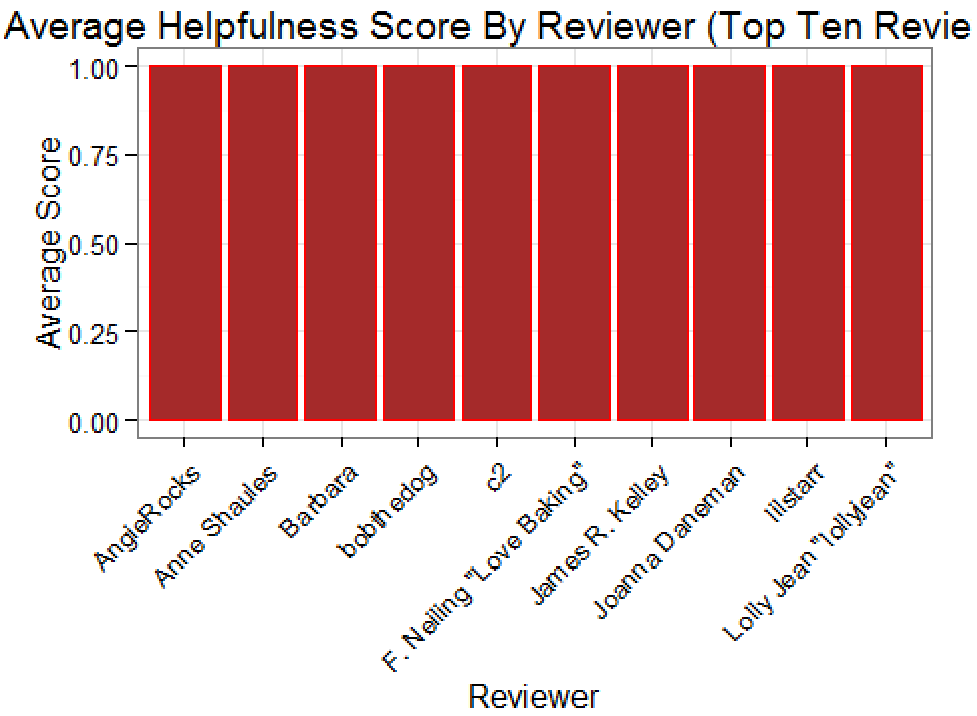
```
dfAmazon[1,]
```

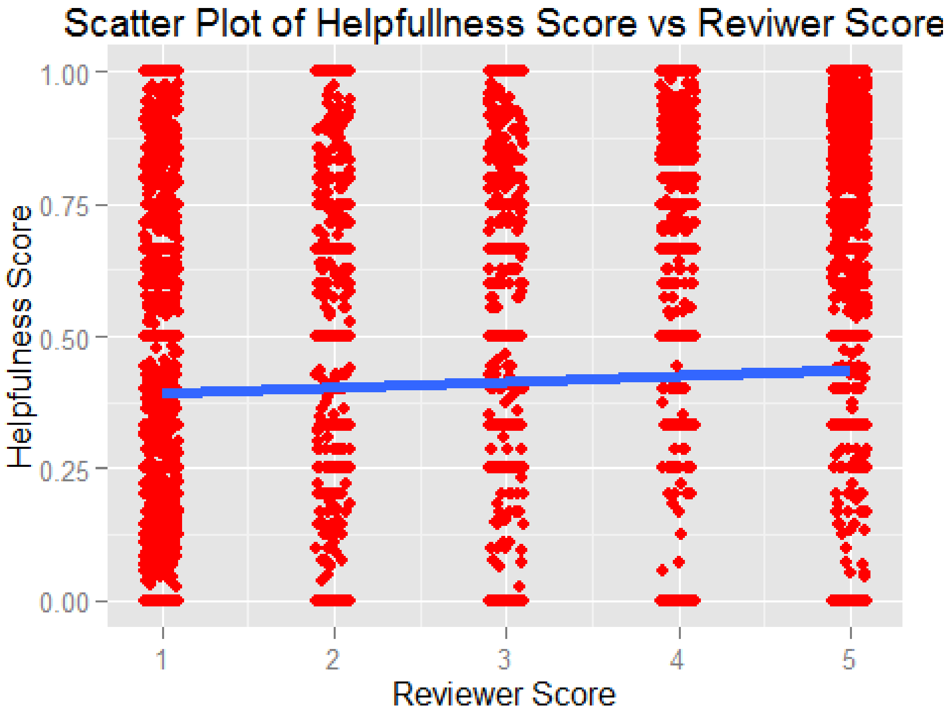
```
##           userID reviewerName numReviews
## 1 A3IY9HIAMJQ7HL   Marcel Lee        289
##                                               itemName itemPrice
## 1 Sharp On All 4 Corners (Deluxe Edition) [Explicit]     17.49
##                date
## 1 December 11, 2014
##
text
## 1 [Explicit] (MP3 Music)      Thereâ□□s a track here entitled Heavy In The
Game. The song itself, a duet with B-Legit, is nothing special, but the beat
is a slapper. Itâ□□s the type of music that should dominate E-40 albums
instead of standing as basically the sole highlight amongst a collection of
songs that, despite the set title, are comparatively dull.Sleep, a sex anthem
```

```
featuring Ludacris and Plies, is enticing, but almost every other track of
the 28 included here isnâ��t. And this is just the first two of another four-
album project, reportedly. So while E-40 is more prolific than ever, it is,
unfortunately for us long-time fans, an ongoing case of quantity over
quality.marcellee.com              Â Comment (1)Â |Â PermalinkÂ |Â Most
recent comment:Â Dec 12, 2014  6:57 AM PST

## Warning in load(url(githubURL)): unsupported URL scheme

## Loading required package: tcltk
```

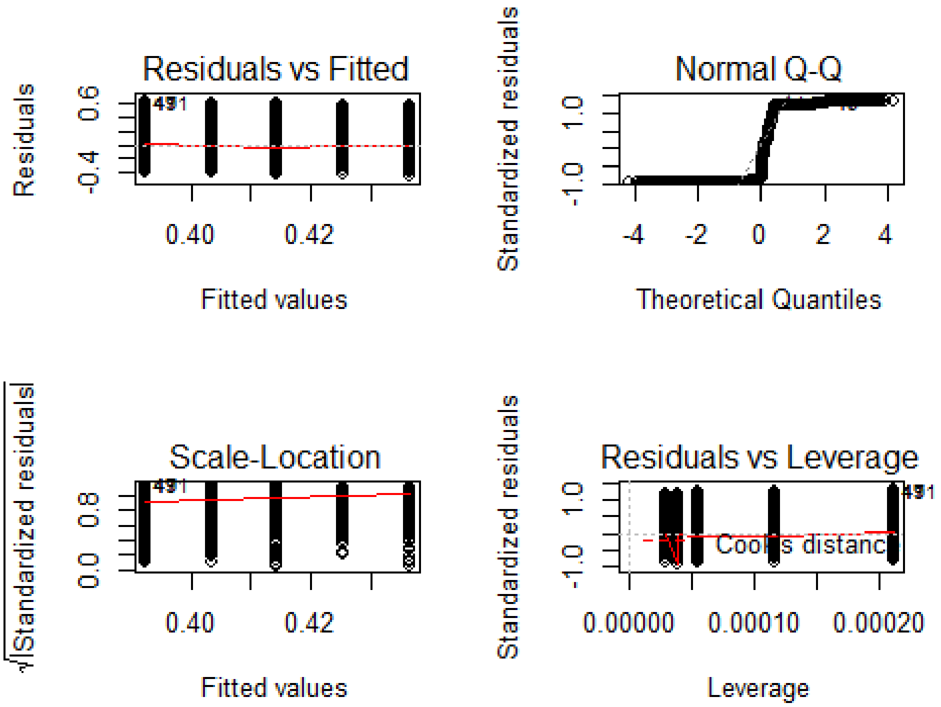

Number of Reviews per Reviewer (Top Ten Reviewers
Count
60
40
20
0
Anonymous
Elinor Molino-myers "Elly"
Snowbrocade
Themis-Athena
C. F. Hill "CFH"
James R. Kelley
Joshua Rubin
Barbara
Rebecca of Amazon "The Rebecca Review"
O. Brown "Ms. O. Khannah-Brown"
Reviewer


Average Reviewer Score By Reviewer (Top Ten Reviewe
Average Score
5
4
3
2
1
0
Joshua Rubin
Elinor Molino-myers "Elly"
James R. Kelley
Anonymous
Snowbrocade
Barbara
C. F. Hill "CFH"
O. Brown "Ms. O. Khannah-Brown"
Themis-Athena
Rebecca of Amazon "The Rebecca Review"
Reviewer

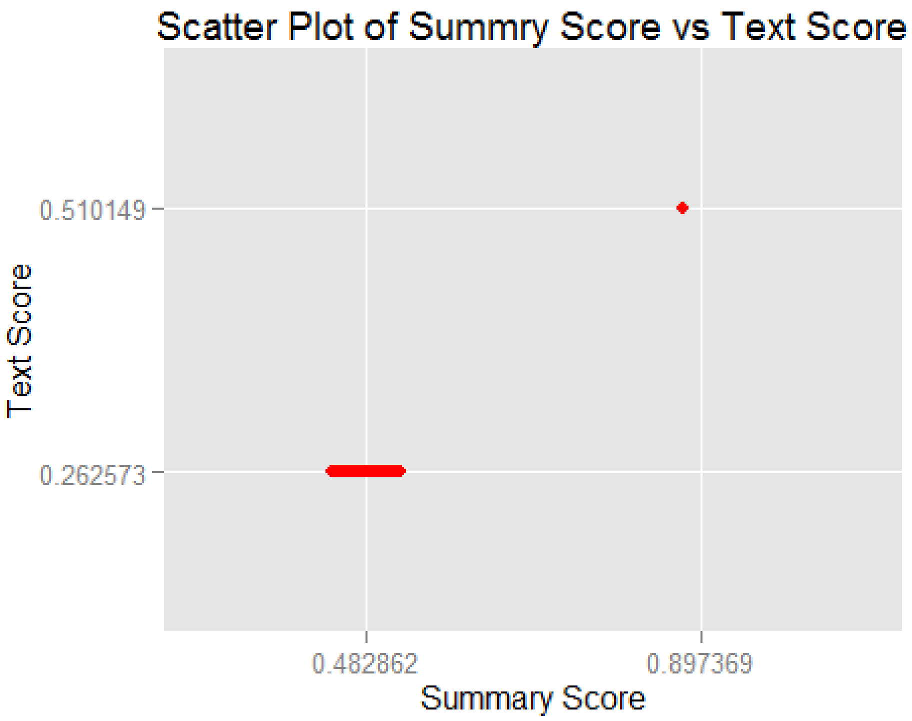




```
## (Intercept)       score
##  0.38132047  0.01101778
## Analysis of Variance Table
##
## Response: helpfulness
##              Df Sum Sq Mean Sq F value    Pr(>F)
## score         1    7.0  7.0360  32.662 1.105e-08 ***
## Residuals 35204 7583.6  0.2154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Call:
## lm(formula = helpfulness ~ score, data = gourmetFoods)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4364 -0.4364 -0.3923  0.5636  0.6077
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.381320   0.008559  44.553  < 2e-16 ***
## score       0.011018   0.001928   5.715 1.11e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4641 on 35204 degrees of freedom
## Multiple R-squared:  0.0009269,  Adjusted R-squared:  0.0008986
## F-statistic: 32.66 on 1 and 35204 DF,  p-value: 1.105e-08
```

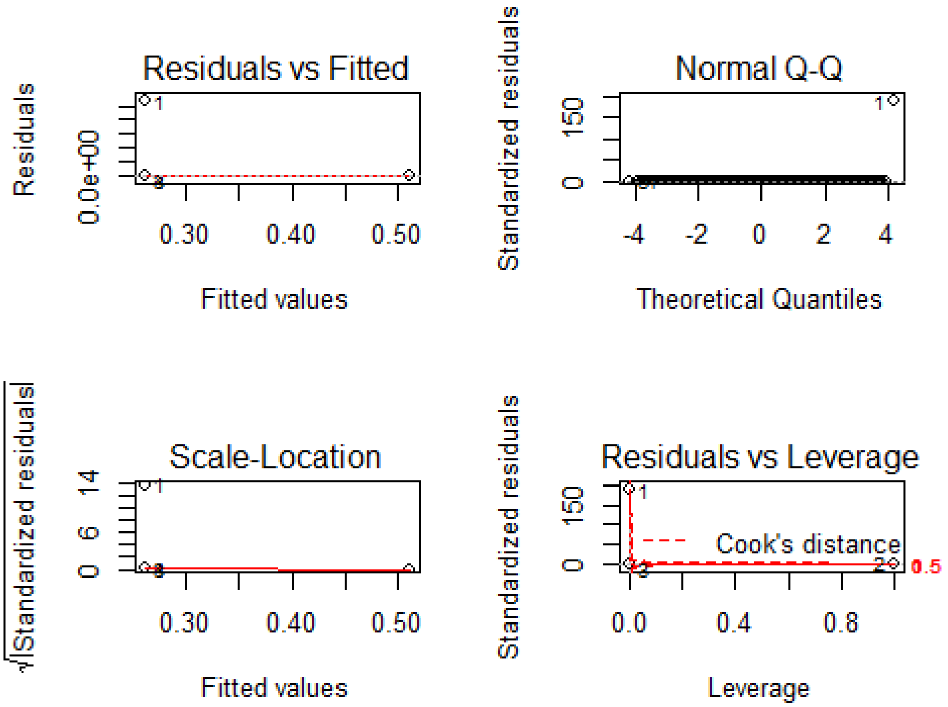


```
## Warning in load(url(githubURL)): unsupported URL scheme

## geom_smooth: Only one unique x value each group.Maybe you want aes(group =
1)?
```



```
##             (Intercept) apisummaryScore0.897369
##                0.262573                0.247576

## Warning in anova.lm(fit): ANOVA F-tests on an essentially perfect fit are
## unreliable

## Analysis of Variance Table
##
## Response: apitextScore
##                    Df   Sum Sq  Mean Sq    F value    Pr(>F)
## apisummaryScore     1 0.061292 0.061292 3.0591e+24 < 2.2e-16 ***
## Residuals       35204 0.000000 0.000000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Call:
## lm(formula = apitextScore ~ apisummaryScore, data = gourmetFoodsAPI)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -8.000e-16 -8.000e-16 -8.000e-16 -8.000e-16  2.656e-11
##
```

```
## Coefficients:
##                            Estimate Std. Error   t value Pr(>|t|)    
## (Intercept)               2.626e-01  7.544e-16 3.481e+14   <2e-16 ***
## apisummaryScore0.897369 2.476e-01  1.416e-13 1.749e+12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.415e-13 on 35204 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1 
## F-statistic: 3.059e+24 on 1 and 35204 DF,  p-value: < 2.2e-16

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```


Residuals vs Fitted
Residuals
0.0e+00
1
3
0.30
0.40
0.50
Fitted values
Normal Q-Q
Standardized residuals
0
150
1
-4
-2
0
2
4
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
0
6
14
1
3
0.30
0.40
0.50
Fitted values
Residuals vs Leverage
Standardized residuals
0
150
1
3
2
Cook's distance
0.5
0.0
0.4
0.8
Leverage