American Marketing Association (AMA) membership information

This data analysis focuses on loading the AMA membership xlsx files from September 2013 to september 2016, missing the file from August, 2015, combining them together, and label the members who have renewed their membership by 1.

1. Note

This file needs to set the current working directory to where it contains the "New AMA membership" folder. To run this file, instead of press the "Knit HTML" button in R Studio, type "rmarkdown::render("AMAMembership.Rmd")" in the console. This will knit in the current session instead of a background session, so that you can play with the varialbes in console. This HTML will be updated, but you have to click open it.

```
rm(list=ls(all=TRUE));
```

```
library(knitr)
opts_chunk$set(echo = TRUE,tidy = TRUE);
# opts_chunk$set(cache.path = paste(getwd(), "/AMAMembership_cache/html", sep=""));
# echo = TRUE: shows the R code in the output document
# cache = TRUE: when evaluating code chunks, the cached chunks are skipped,
# but the objects created in these chunks are loaded from previously saved database
# autodep = TRUE: figure out the dependencies among chunks automatically
# by analyzing the global variables in the code (may not be reliable)
# include = FALSE: nothing will be written into the output document,
# but the code is still evaluated and plot files are generated if there are
# any plots in the chunk
```

2. Read in the data files (xlsx) into R studio.

```
library(xlsx)
```

```
## Loading required package: rJava
```

```
## Loading required package: xlsxjars
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```r
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
Sep2013 <- read.xlsx(paste(getwd(), "/New AMA Membership/2013 09_September AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Oct2013 <- read.xlsx(paste(getwd(), "/New AMA Membership/2013 10_October AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Nov2013 <- read.xlsx(paste(getwd(), "/New AMA Membership/2013 11_November AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Dec2013 <- read.xlsx(paste(getwd(), "/New AMA Membership/2013 12_December AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Jan2014 <- read.xlsx(paste(getwd(), "/New AMA Membership/2014 01_January AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Feb2014 <- read.xlsx(paste(getwd(), "/New AMA Membership/2014 02_February AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Mar2014 <- read.xlsx(paste(getwd(), "/New AMA Membership/2014 03_March AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Apr2014 <- read.xlsx(paste(getwd(), "/New AMA Membership/2014 04_April AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
May2014 <- read.xlsx(paste(getwd(), "/New AMA Membership/2014 05_May AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Jun2014 <- read.xlsx(paste(getwd(), "/New AMA Membership/2014 06_June AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Jul2014 <- read.xlsx(paste(getwd(), "/New AMA Membership/2014 07_July AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Aug2014 <- read.xlsx(paste(getwd(), "/New AMA Membership/2014 08_August AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Sep2014 <- read.xlsx(paste(getwd(), "/New AMA Membership/2014 09_September AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Oct2014 <- read.xlsx(paste(getwd(), "/New AMA Membership/2014 10_October AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Nov2014 <- read.xlsx(paste(getwd(), "/New AMA Membership/2014 11_November AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Dec2014 <- read.xlsx(paste(getwd(), "/New AMA Membership/2014 12_December AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Jan2015 <- read.xlsx(paste(getwd(), "/New AMA Membership/2015 01_January AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Feb2015 <- read.xlsx(paste(getwd(), "/New AMA Membership/2015 02_February AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Mar2015 <- read.xlsx(paste(getwd(), "/New AMA Membership/2015 03_March AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Apr2015 <- read.xlsx(paste(getwd(), "/New AMA Membership/2015 04_April AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
May2015 <- read.xlsx(paste(getwd(), "/New AMA Membership/2015 05_May AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Jun2015 <- read.xlsx(paste(getwd(), "/New AMA Membership/2015 06_June AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Jul2015 <- read.xlsx(paste(getwd(), "/New AMA Membership/2015 07_July AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Sep2015 <- read.xlsx(paste(getwd(), "/New AMA Membership/2015 09_September AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Oct2015 <- read.xlsx(paste(getwd(), "/New AMA Membership/2015 10_October AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Nov2015 <- read.xlsx(paste(getwd(), "/New AMA Membership/2015 11_November AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Dec2015 <- read.xlsx(paste(getwd(), "/New AMA Membership/2015 12_December AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Jan2016 <- read.xlsx(paste(getwd(), "/New AMA Membership/2016 01_January AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Feb2016 <- read.xlsx(paste(getwd(), "/New AMA Membership/2016 02_February AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Mar2016 <- read.xlsx(paste(getwd(), "/New AMA Membership/2016 03_March AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Apr2016 <- read.xlsx(paste(getwd(), "/New AMA Membership/2016 04_April AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
May2016 <- read.xlsx(paste(getwd(), "/New AMA Membership/2016 05_May AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Jun2016 <- read.xlsx(paste(getwd(), "/New AMA Membership/2016 06_June AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Jul2016 <- read.xlsx(paste(getwd(), "/New AMA Membership/2016 07_July AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Aug2016 <- read.xlsx(paste(getwd(), "/New AMA Membership/2016 08_August AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

```r
Sep2016 <- read.xlsx(paste(getwd(), "/New AMA Membership/2016 09_September AMA Membership_v2.xlsx",
    sep = ""), sheetIndex = 1)
```

3. Convert ID to numeric type.

```r
# Convert MEMBER.ID to numeric type and all the dates to date type.  NAMES2
# does not include Nov2013 file.
NAMES2 <- c("Sep2013", "Oct2013", "Dec2013", "Jan2014", "Feb2014", "Mar2014",
    "Apr2014", "May2014", "Jun2014", "Jul2014", "Aug2014", "Sep2014", "Oct2014",
    "Nov2014", "Dec2014", "Jan2015", "Feb2015", "Mar2015", "Apr2015", "May2015",
    "Jun2015", "Jul2015", "Sep2015", "Oct2015", "Nov2015", "Dec2015", "Jan2016",
    "Feb2016", "Mar2016", "Apr2016", "May2016", "Jun2016", "Jul2016", "Aug2016",
    "Sep2016")
# check if there is NA value before converting the MEMBER.ID
for (i in NAMES2) {
    j = get(i)
    print(dim(j[is.na(j$MEMBER.ID), ]))
    j$MEMBER.ID <- as.numeric(paste(j$MEMBER.ID))
    assign(i, j)
}
# check if there is NA value after converting the MEMBER.ID to numeric type.
for (i in NAMES2) {
    j = get(i)
    print(dim(j[is.na(j$MEMBER.ID), ]))
    str(j$MEMBER.ID)
}
```

4. Convert the date to date type.

```r
# These are the files with date format: '%Y-%m-%d'
NAMES3 <- c("Sep2013", "Oct2014", "Nov2014", "Dec2014", "Jan2015", "Feb2015",
    "Mar2015", "Apr2015", "May2015", "Jun2015", "Jul2015", "Sep2015", "Oct2015",
    "Nov2015", "Dec2015", "Jan2016", "Feb2016", "Mar2016", "Apr2016", "May2016",
    "Jun2016", "Jul2016", "Aug2016", "Sep2016")

# These are the files with date format: '%m/%d/%Y'
NAMES4 <- c("Oct2013", "Dec2013", "Jan2014", "Feb2014", "Mar2014", "Apr2014",
    "May2014", "Jun2014", "Jul2014", "Aug2014", "Sep2014")

# Check if there is NA value before converting MEMBER.SINCE.DATE
for (i in NAMES3) {
    j = get(i)
    print(dim(j[is.na(j$MEMBER.SINCE.DATE), ]))
    j$MEMBER.SINCE.DATE <- as.Date(j$MEMBER.SINCE.DATE, "%Y-%m-%d")
    assign(i, j)
}

# Check if there is NA value after converting MEMBER.SINCE.DATE
for (i in NAMES3) {
    j = get(i)
    print(dim(j[is.na(j$MEMBER.SINCE.DATE), ]))
    str(j$MEMBER.SINCE.DATE)
}
# Check if there is NA value before converting MEMBER.SINCE.DATE
for (i in NAMES4) {
    j = get(i)
    print(dim(j[is.na(j$MEMBER.SINCE.DATE), ]))
    j$MEMBER.SINCE.DATE <- as.Date(j$MEMBER.SINCE.DATE, "%m/%d/%Y")
    assign(i, j)
}
# Check if there is NA value after converting MEMBER.SINCE.DATE
for (i in NAMES4) {
    j = get(i)
    print(dim(j[is.na(j$MEMBER.SINCE.DATE), ]))
    str(j$MEMBER.SINCE.DATE)
}

# Check if there is NA value before converting EXPIRATION.DATE
```

```r
for (i in NAMES3) {
    j = get(i)
    print(dim(j[is.na(j$EXPIRATION.DATE), ]))
    j$EXPIRATION.DATE <- as.Date(j$EXPIRATION.DATE, "%Y-%m-%d")
    assign(i, j)
}
# Check if there is NA value after converting EXPIRATION.DATE
for (i in NAMES3) {
    j = get(i)
    print(dim(j[is.na(j$EXPIRATION.DATE), ]))
    str(j$EXPIRATION.DATE)
}
# Check if there is NA value before converting EXPIRATION.DATE
for (i in NAMES4) {
    j = get(i)
    print(dim(j[is.na(j$EXPIRATION.DATE), ]))
    j$EXPIRATION.DATE <- as.Date(j$EXPIRATION.DATE, "%m/%d/%Y")
    assign(i, j)
}
# Check if there is NA value after converting EXPIRATION.DATE
for (i in NAMES4) {
    j = get(i)
    print(dim(j[is.na(j$EXPIRATION.DATE), ]))
    str(j$EXPIRATION.DATE)
}

# Check if there is NA value before converting DATE.PULLED
for (i in NAMES3) {
    j = get(i)
    print(dim(j[is.na(j$DATE.PULLED), ]))
    j$DATE.PULLED <- as.Date(j$DATE.PULLED, "%Y-%m-%d")
    assign(i, j)
}
# Check if there is NA value after converting DATE.PULLED
for (i in NAMES3) {
    j = get(i)
    print(dim(j[is.na(j$DATE.PULLED), ]))
    str(j$DATE.PULLED)
}
# Check if there is NA value before converting DATE.PULLED
for (i in NAMES4) {
    j = get(i)
    print(dim(j[is.na(j$DATE.PULLED), ]))
    j$DATE.PULLED <- as.Date(j$DATE.PULLED, "%m/%d/%Y")
    assign(i, j)
}
# Check if there is NA value after converting DATE.PULLED
for (i in NAMES4) {
    j = get(i)
    print(dim(j[is.na(j$DATE.PULLED), ]))
    str(j$DATE.PULLED)
}
```

```
# The codes below are used to comapre names and class types of different
# files.  table(names(Sep2013) %in% names(Oct2013))['FALSE'];
# names(Sep2013)[which(sapply(Sep2013,class)!='factor')]; for (i in
# NAMES2){j=get(i);print(i);print(class(j$COMMUNICATION.CHANGE.DATE))}

# NAMES5 are the ones that these four columns need to be converted to date
# type:'ADDRESS.CHANGE.DATE', 'COMMUNICATION.CHANGE.DATE',
# 'STUDENT.GRADUATION.DATE', 'DATE.LAST.MODIFIED'.  It turns out that NAMES5
# is the same as NAMES4
NAMES5 <- c("Oct2013", "Dec2013", "Jan2014", "Feb2014", "Mar2014", "Apr2014",
    "May2014", "Jun2014", "Jul2014", "Aug2014", "Sep2014")

# Check the number of rows of the file and the number of empty values of
# 'ADDRESS.CHANGE.DATE' before converting.
for (i in NAMES5) {
    j = get(i)
    cat(paste(dim(j)[1], dim(j[j$ADDRESS.CHANGE.DATE == "", ])[1], sep = "\t"),
        "\n")
    j$ADDRESS.CHANGE.DATE <- as.Date(j$ADDRESS.CHANGE.DATE, "%m/%d/%Y")
    assign(i, j)
}
# Check the number of rows of the file and the number of NA values of
# 'ADDRESS.CHANGE.DATE' after converting.
for (i in NAMES5) {
    j = get(i)
    cat(paste(dim(j)[1], dim(j[is.na(j$ADDRESS.CHANGE.DATE), ])[1], sep = "\t"),
        "\n")
    print(head(unique(j$ADDRESS.CHANGE.DATE)))
}

# Check the number of rows of the file and the number of empty values of
# 'COMMUNICATION.CHANGE.DATE' before converting.
for (i in NAMES5) {
    j = get(i)
    cat(paste(dim(j)[1], dim(j[j$COMMUNICATION.CHANGE.DATE == "", ])[1], sep = "\t"),
        "\n")
    j$COMMUNICATION.CHANGE.DATE <- as.Date(j$COMMUNICATION.CHANGE.DATE, "%m/%d/%Y")
    assign(i, j)
}

# Check the number of rows of the file and the number of NA values of
# 'COMMUNICATION.CHANGE.DATE' after converting.
for (i in NAMES5) {
    j = get(i)
    cat(paste(dim(j)[1], dim(j[is.na(j$COMMUNICATION.CHANGE.DATE), ])[1], sep = "\t"),
        "\n")
    print(head(unique(j$COMMUNICATION.CHANGE.DATE)))
}

# NAMES6 are the ones whose COMMUNICATION.CHANGE.DATE are logic type.
NAMES6 <- c("May2015", "Jun2015", "Jul2015", "Sep2015", "Oct2015", "Dec2015",
    "Jan2016", "Mar2016", "Apr2016", "May2016", "Jul2016", "Aug2016")
```

```r
# Check the number of rows of the file and the number of empty values of
# 'COMMUNICATION.CHANGE.DATE' before converting.
for (i in NAMES6) {
    j = get(i)
    cat(paste(dim(j)[1], dim(j[j$COMMUNICATION.CHANGE.DATE == "", ])[1], sep = "\t"),
        "\n")
    j$COMMUNICATION.CHANGE.DATE <- as.Date(as.character(j$COMMUNICATION.CHANGE.DATE),
        "%m/%d/%Y")
    assign(i, j)
}
# Check the number of rows of the file and the number of NA values of
# 'COMMUNICATION.CHANGE.DATE' after converting.
for (i in NAMES6) {
    j = get(i)
    cat(paste(dim(j)[1], dim(j[is.na(j$COMMUNICATION.CHANGE.DATE), ])[1], sep = "\t"),
        "\n")
    print(head(unique(j$COMMUNICATION.CHANGE.DATE)))
}

# Check the number of rows of the file and the number of empty values of
# 'STUDENT.GRADUATE.DATE' before converting.
for (i in NAMES5) {
    j = get(i)
    cat(paste(dim(j)[1], dim(j[j$STUDENT.GRADUATION.DATE == "", ])[1], sep = "\t"),
        "\n")
    j$STUDENT.GRADUATION.DATE <- as.Date(j$STUDENT.GRADUATION.DATE, "%m/%d/%Y")
    assign(i, j)
}
# Check the number of rows of the file and the number of NA values of
# 'STUDENT.GRADUATE.DATE' after converting.
for (i in NAMES5) {
    j = get(i)
    cat(paste(dim(j)[1], dim(j[is.na(j$STUDENT.GRADUATION.DATE), ])[1], sep = "\t"),
        "\n")
    print(head(unique(j$STUDENT.GRADUATION.DATE)))
}

# Check the number of rows of the file and the number of empty values of
# 'DATE.LAST.MODIFIED' before converting.
for (i in NAMES5) {
    j = get(i)
    cat(paste(dim(j)[1], dim(j[is.na(j$DATE.LAST.MODIFIED), ])[1], sep = "\t"),
        "\n")
    j$DATE.LAST.MODIFIED <- as.Date(j$DATE.LAST.MODIFIED, "%m/%d/%Y")
    assign(i, j)
}
# Check the number of rows of the file and the number of empty values of
# 'DATE.LAST.MODIFIED' after converting.
for (i in NAMES5) {
    j = get(i)
    cat(paste(dim(j)[1], dim(j[is.na(j$DATE.LAST.MODIFIED), ])[1], sep = "\t"),
        "\n")
    print(head(unique(j$DATE.LAST.MODIFIED)))
}
```

5. Merge all the files.

```
# Merge all the files except Nov2013.

identical(Sep2015$COMPANY.NAME, Sep2015$COMPANY.NAME.1)
```

```
## [1] TRUE
```

```
Sep2015 <- subset(Sep2015, select = (names(Sep2015) != "COMPANY.NAME.1"))

All <- Sep2013
for (i in NAMES2[2:length(NAMES2)]) {
    j = get(i)
    All <- merge(All, j, all = TRUE)
    # the default of by is to use the columns with common names between the two
    # data frames.
}

# confirmed that after merging, the file has 48 columns.
ncol(All)
```

```
## [1] 48
```

```
# check the weekday of each file is pulled.
table(wday(All$DATE.PULLED, label = TRUE))
```

```
##
##   Sun   Mon  Tues   Wed Thurs   Fri   Sat
##  2652  2245  3299   980  5357  1665  2685
```

```
# DATE.PULLED2 are month and year of each DATE.PULLED.
All$DATE.PULLED2 <- as.yearmon(All$DATE.PULLED)

# EMP represents EXPIRATION.DATE-DATE.PULLED2 (in month).
All$EMP <- round((as.yearmon(All$EXPIRATION.DATE) - All$DATE.PULLED2) * 12,
    digits = 0)
# The smallest number of EMP is -1, which means that once the member has
# passed the expiration date by one month, he/she is dropped out.
table(All$EMP)
```

```
##
##   -1    0    1    2    3    4    5    6    7    8    9   10   11   12   13
##  533 1056 1279 1442 1482 1472 1481 1458 1421 1441 1455 1375 1355 1181  251
##   14   15   16   17   18   19   20   21   22   23   24   25   26   36
##   77   17   11   13   12   13   11   11   11   11   10    2    1    1
```

```r
# double chekc if All has any NA values.
dim(All[is.na(All$MEMBER.ID), ])
```

```
## [1]  0 50
```

```r
dim(All[is.na(All$MEMBER.SINCE.DATE), ])
```

```
## [1]  0 50
```

```r
dim(All[is.na(All$EXPIRATION.DATE), ])
```

```
## [1]  0 50
```

```r
dim(All[is.na(All$DATE.PULLED), ])
```

```
## [1]  0 50
```

```r
dim(All[is.na(All$DATE.PULLED2), ])
```

```
## [1]  0 50
```

```r
dim(All[is.na(All$STATUS), ])
```

```
## [1]  0 50
```

```r
dim(All[is.na(All$EMP), ])
```

```
## [1]  0 50
```

6. What if the member renews the membership earlier than the current expiration date?

- Does the expiration date get extended to one more year than the current expiration date? The answer is yes.

```r
AllDate <- subset(All, select = c("MEMBER.ID", "MEMBER.SINCE.DATE", "EXPIRATION.DATE",
    "DATE.PULLED2", "EMP"))

AllDate <- arrange(AllDate, MEMBER.ID, MEMBER.SINCE.DATE, EXPIRATION.DATE, DATE.PULLED2)

# For every MEMBER.ID, extract the earliest DATE.PULLED2 for every
# EXPIRATION.DATE. This DATE.PULLED2 is the time when the member joined or
# renew their membership.

ID <- split(AllDate, AllDate$MEMBER.ID)
t <- do.call(rbind, lapply(ID, function(x) (x[!duplicated(x$EXPIRATION.DATE),
    ])))
```

```
# Some of the EMP are larger than 12.  Check ID=638951, on 2014-01-31, the
# person renews the membership and the new expiration date is 2015-03-31,
# since the old expiration date is 2014-03-31. It confirms that if you renew
# your membership earlier than the current expiration date, your membership
# will be extended to one more year after your current expiration date.
# Same thing happened to ID=684951
t[t$MEMBER.ID == "638951", ]
```

```
##            MEMBER.ID MEMBER.SINCE.DATE EXPIRATION.DATE DATE.PULLED2 EMP
## 638951.170    638951        1977-04-01      2014-03-31     Sep 2013   6
## 638951.173    638951        1977-04-01      2015-03-31     Jan 2014  14
## 638951.186    638951        1977-04-01      2016-03-31     Feb 2015  13
```

```
t[t$MEMBER.ID == "684951", ]
```

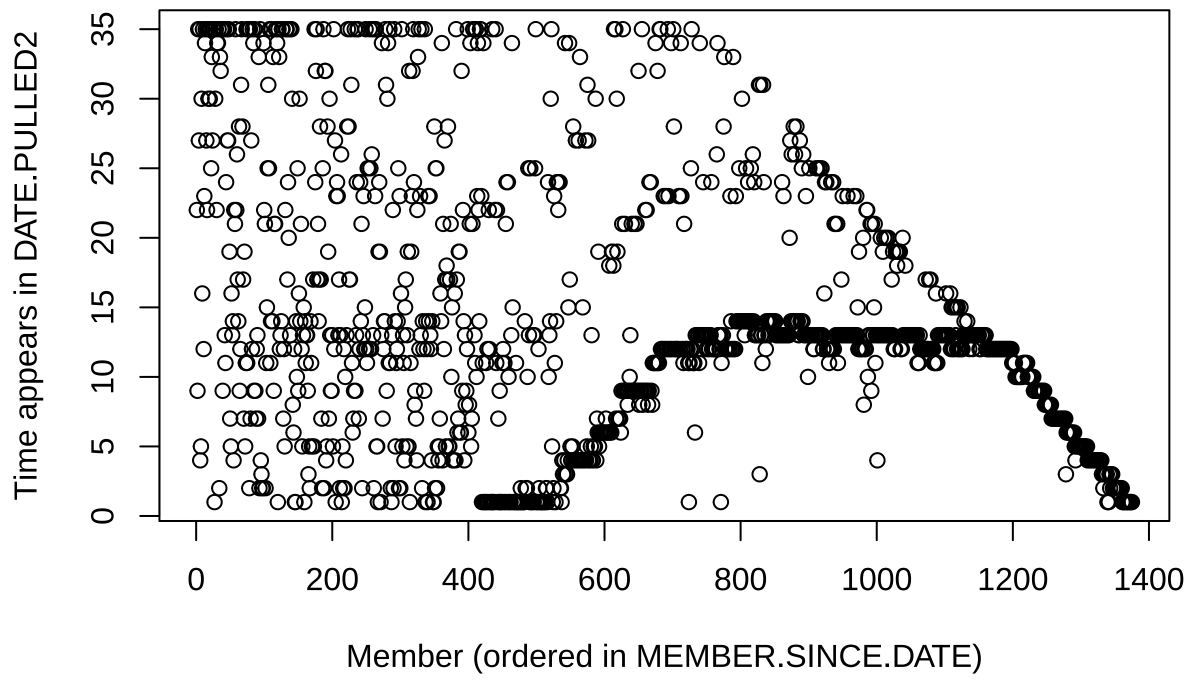
```
##            MEMBER.ID MEMBER.SINCE.DATE EXPIRATION.DATE DATE.PULLED2 EMP
## 684951.216    684951        1981-11-01      2014-06-30     Sep 2013   9
## 684951.224    684951        1981-11-01      2015-06-30     Jun 2014  12
## 684951.232    684951        1981-11-01      2016-06-30     Feb 2015  16
## 684951.244    684951        1981-11-01      2017-06-30     Mar 2016  15
```

7. Check how many times does each MEMBER.ID apprears in different DATE.PULLED2.

```
Freq <- AllDate[!duplicated(AllDate[c("MEMBER.ID", "DATE.PULLED2")]), ]
FreqID <- split(Freq, Freq$MEMBER.ID)

freq <- do.call(rbind, lapply(FreqID, function(x) data.frame(MEMBER.ID = unique(x$MEMBER.ID),
    MEMBER.SINCE.DATE = min(x$MEMBER.SINCE.DATE), frequency = nrow(x))))
freq <- arrange(freq, MEMBER.SINCE.DATE)

freq <- cbind(freq, lable = 1:nrow(freq))
plot(freq[, 4], freq[, 3], xlab = "Member (ordered in MEMBER.SINCE.DATE)", ylab = "Time appears in DATE
```



8. Define Renew == 1.

- For the member who has EXPIRATION.DATE 4 months after the DATE.PULLED2, and renewed the membership within 10 months after DATE.PULLED2, label the member at DATE.PULLED2 with Renew == 1. Otherwise Renew == 0.

```
bgyr = "2014"
bgmn = "01"
endyr = "2014"
endmn = "12"
BG <- as.yearmon(as.Date(paste(bgyr, bgmn, "01", sep = ""), "%Y%m%d"))
END <- as.yearmon(as.Date(paste(endyr, endmn, "01", sep = ""), "%Y%m%d"))

dates <- as.yearmon(seq(from = as.Date(BG), to = as.Date(END), by = "month"))

result <- All[FALSE, ]

for (i in 1:length(dates)) {
    test <- filter(All, DATE.PULLED2 == dates[i] & as.yearmon(EXPIRATION.DATE) ==
        dates[i] + 4/12)
    pullDate <- filter(All, (MEMBER.ID %in% test$MEMBER.ID) & (DATE.PULLED2 >
        dates[i]) & (DATE.PULLED2 <= dates[i] + 10/12))
    renewID <- pullDate[as.yearmon(pullDate$EXPIRATION.DATE) != (dates[i] +
        4/12), ]$MEMBER.ID
    test$Renew <- 0
    test[test$MEMBER.ID %in% renewID, ]$Renew <- 1
    result <- rbind(result, test)

}

table(duplicated(result$MEMBER.ID))
```

```
##
## FALSE  TRUE
##   564     1
```

```
which(duplicated(result$MEMBER.ID) == TRUE)
```

```
## [1] 388
```

```
result[result$MEMBER.ID == "3125514", ]
```

```
##     MEMBER.ID  MEMBER.TYPE MARKETING.ROLE MEMBER.SINCE.DATE
## 387   3125514 PROFESSIONAL      Marketers        2008-08-17
## 388   3125514 PROFESSIONAL      Marketers        2008-08-17
##     EXPIRATION.DATE  STATUS MARKETINGPOWER.STATUS PREFIX SUFFIX
## 387      2014-12-31 CURRENT             CONFIRMED     MS
## 388      2014-12-31 CURRENT             CONFIRMED     MS
##     ADDRESS.CHANGE.DATE COMMUNICATION.CHANGE.DATE            COMPANY.NAME
## 387                <NA>                      <NA> Teledyne Marine Systems
## 388                <NA>                      <NA> Teledyne Marine Systems
##             JOB.TITLE PRIMARY.ADDRESS.TYPE   PRIMARY.CITY PRIMARY.STATE
## 387 49 Edgerton Drive                  BUS North Falmouth            MA
## 388 49 Edgerton Drive                  BUS North Falmouth            MA
##     PRIMARY.POSTAL.CODE PRIMARY.COUNTRY PRIMARY.PHONE.LOCATION
## 387               02556   United States                   WORK
## 388               02556   United States                   WORK
##     SECONDARY.ADDRESS.TYPE SECONDARY.CITY SECONDARY.STATE
## 387
## 388
##     SECONDARY.POSTAL.CODE SECONDARY.COUNTRY GROUP MARKETING.RESPONSIBILITY
## 387                                            No Marketing Communications
## 388                                           Yes Marketing Communications
##     STUDENT.GRADUATION.DATE HIGHEST.DEGREE YEARS.IN.MARKETING
## 387                    <NA>        Masters           10 to 19
## 388                    <NA>        Masters           10 to 19
##        JOB.TITLE.CATEGORY INDUSTRY COMPANY.SALES.REVENUE COMPANY.SIZE
## 387 Director/Sr. Director    Other       0 to 50 Million    50 to 249
## 388 Director/Sr. Director    Other       0 to 50 Million    50 to 249
##     RESPONSIBILITY.TYPE YEARS.EMPLOYED.AT.COMPANY AGE.CATEGORY
## 387              B to B                               45 to 54
## 388              B to B                               45 to 54
##     NUMBER.OF.OTHER.ASSOCS PAID.BY.COMPANY       PRIMARY.TOPIC.AREA
## 387                 0 to 1               N Brand/Product Management
## 388                 0 to 1               N Brand/Product Management
##         SECONDARY.TOPIC.AREA TERTIARY.TOPIC.AREA DATE.LAST.MODIFIED
## 387 Marketing Communications Customer Engagement         2014-06-06
## 388 Marketing Communications Customer Engagement         2014-06-06
##     DATE.PULLED MEMBER.FOCUS AUTO.RENEWAL MARKET.CODE ALLOW.EMAIL.FLAG
## 387  2014-08-31     NATIONAL         <NA>        <NA>             <NA>
## 388  2014-08-31     NATIONAL         <NA>        <NA>             <NA>
##     ALLOW.PHONE.CALLS DATE.PULLED2 EMP Renew
## 387              <NA>     Aug 2014   4     1
## 388              <NA>     Aug 2014   4     1
```

```
AllDate[AllDate$MEMBER.ID == "3125514", ]
```

```
##      MEMBER.ID MEMBER.SINCE.DATE EXPIRATION.DATE DATE.PULLED2 EMP
## 3883   3125514        2008-08-17      2013-12-31     Sep 2013   3
## 3884   3125514        2008-08-17      2013-12-31     Oct 2013   2
## 3885   3125514        2008-08-17      2014-12-31     Dec 2013  12
## 3886   3125514        2008-08-17      2014-12-31     Jan 2014  11
## 3887   3125514        2008-08-17      2014-12-31     Feb 2014  10
## 3888   3125514        2008-08-17      2014-12-31     Mar 2014   9
## 3889   3125514        2008-08-17      2014-12-31     Apr 2014   8
## 3890   3125514        2008-08-17      2014-12-31     May 2014   7
## 3891   3125514        2008-08-17      2014-12-31     May 2014   7
## 3892   3125514        2008-08-17      2014-12-31     Jun 2014   6
## 3893   3125514        2008-08-17      2014-12-31     Jun 2014   6
## 3894   3125514        2008-08-17      2014-12-31     Jul 2014   5
## 3895   3125514        2008-08-17      2014-12-31     Jul 2014   5
## 3896   3125514        2008-08-17      2014-12-31     Aug 2014   4
## 3897   3125514        2008-08-17      2014-12-31     Aug 2014   4
## 3898   3125514        2008-08-17      2014-12-31     Sep 2014   3
## 3899   3125514        2008-08-17      2014-12-31     Sep 2014   3
## 3900   3125514        2008-08-17      2014-12-31     Oct 2014   2
## 3901   3125514        2008-08-17      2014-12-31     Oct 2014   2
## 3902   3125514        2008-08-17      2014-12-31     Nov 2014   1
## 3903   3125514        2008-08-17      2014-12-31     Nov 2014   1
## 3904   3125514        2008-08-17      2015-12-31     Dec 2014  12
## 3905   3125514        2008-08-17      2015-12-31     Jan 2015  11
## 3906   3125514        2008-08-17      2015-12-31     Feb 2015  10
## 3907   3125514        2008-08-17      2015-12-31     Mar 2015   9
## 3908   3125514        2008-08-17      2015-12-31     Apr 2015   8
## 3909   3125514        2008-08-17      2015-12-31     May 2015   7
## 3910   3125514        2008-08-17      2015-12-31     Jun 2015   6
## 3911   3125514        2008-08-17      2015-12-31     Jul 2015   5
## 3912   3125514        2008-08-17      2015-12-31     Sep 2015   3
## 3913   3125514        2008-08-17      2015-12-31     Oct 2015   2
## 3914   3125514        2008-08-17      2015-12-31     Nov 2015   1
## 3915   3125514        2008-08-17      2015-12-31     Dec 2015   0
## 3916   3125514        2012-12-10      2015-12-31     Jan 2016  -1
```

```
# result <- result[!duplicated(result$MEMBER.ID),];
dim(result[result$Renew == 1, ])
```

```
## [1] 236  51
```

```
write.xlsx(result, file = "result.xlsx")
```