Businesses Regress to the Mean

Ravi Shankar

Tuesday, November 17, 2015

Introduction - It is hypothesized that all businesses regress to the mean with time, as studied for entities in existence for 10 years. The average ratings (quantitative, with star ratings, and qualitative, with sentiment polarity count determined with text mining) for the first two years (as there were fewer reviews) and last six months are used to compare. Practical utility: Companies can use this to identify the business lifespan and reinvent themselves to be more agile to the needs of the community.

Methods and Data - We saved yelpreview as an rds file. Two subsets were formed: sreview (5 Mbytes) and treview (400 Mbytes) for analyzing star rankings and text reviews, respectively. They are available at the Github site (see submission page). The commented out lines below can also be uncommented and executed to get the same result, without using saved data. Step 1: Use the date column of sreview to subset the data for the first 2 years ('st24') and the last six months ('en6').

```
library(lubridate, warn.conflicts = FALSE, quietly=TRUE)

#sreview <- yelpreview[,c("stars", "date", "business_id")]  --- rds at Github
#treview <- yelpreview[,c("date","text", "business_id")]  --- rds at Github
sreview <- readRDS("sreview.rds")
s.order <- sreview[order(sreview$date),]
start <- s.order$date[1]; dim <- dim(s.order); end <- s.order$date[dim[1]]
st <- as.Date(start); st24 <- st + 730; end <- as.Date(end); en6 <- end-180
# collect unique business_ids for first 24 months and last 6 months.
st24.reviews <- s.order[as.Date(s.order$date) <= st24,]
en.reviews <- s.order[as.Date(s.order$date) >= en6,]
s24.un <- unique(st24.reviews$business_id); e.un <- unique(en.reviews$business_id)
# Find the common subset. 1156 of the 1663 businesses that started in the first 2
years were stil around after 10 years
match <- s24.un %in% e.un; bus.list <- s24.un[match]
bus.list <- s24.un[match]; length(bus.list)

## [1] 1156
```

Step 2: Obtain the star ratings for 1156 businesses at start and end of the 10 year period

```
match.st <- st24.reviews$business_id %in% bus.list;
match.en <- en.reviews$business_id %in% bus.list;
reviews.st <- st24.reviews[match.st ,];reviews.en <- en.reviews[match.en,]
```

Find average of reviews for each business at start and end

```
# for first 2 years
rev.s.f <- factor(reviews.st$business_id)
stars.s.avg <- tapply(reviews.st$stars, rev.s.f, mean)
stars.s.cnt <- tapply(reviews.st$stars, rev.s.f, length)
stars.s.bus <- tapply(reviews.st$business_id, rev.s.f, unique)
```

```
st.bus <- data.frame(stars.s.avg, stars.s.cnt, stars.s.bus)

# for last six months
rev.e.f <- factor(reviews.en$business_id)
stars.e.avg <- tapply(reviews.en$stars, rev.e.f, mean)
stars.e.cnt <- tapply(reviews.en$stars, rev.e.f, length)
stars.e.bus <- tapply(reviews.en$business_id, rev.e.f, unique)
en.bus <- data.frame(stars.e.avg, stars.e.cnt, stars.e.bus)
# order businesses in alphabetical order for start and end
st.order <-  st.bus [order(st.bus$stars.s.bus),]
en.order <-  en.bus [order(en.bus$stars.e.bus),]
rm(sreview)
```

We then run paired t-test statistic and produce a violin plot. See 'Results'

Step 3: Obtain text reviews for the 1156 businesses at start and end of the 10 year period. The subset for text reviews, treview, is large at 400 Mbytes. We subset this for the 'st24' and 'en6' periods. There are intermediate steps needed. These steps are documented below commented out; corresponding initial and final data sets of the following steps are available at the Github site. You may also uncomment these and achieve the same results, without using the saved data.

```
library(tm); library(rJava);library(SnowballC)

# treview <- yelpreview[,c("date","text", "business_id")]
#treview <- readRDS("treview.rds")  -- rds saved at the Github site
# collect unique business_ids for the first 24 months
#t.order <- treview[order(treview$date),]
#st.reviewt <- t.order[as.Date(t.order$date) <= st24,]
#s.ut <- unique(st.reviewt$business_id)
# collect unique business_ids for the last 6 months
#en.reviewt <- t.order[as.Date(t.order$date) >= en6,]
#e.ut <- unique(en.reviewt$business_id)
# find the subset that is common to both st24 and en6
#match <- s.ut %in% e.ut
#bus.lis.t <- s.ut[match] -- saved at the Github site
#verify that this list is the same as of bus.list obtained for star ratings
#match.stt <- st.reviewt$business_id %in% bus.lis.t
#reviewt.stt <- st.reviewt[match.stt ,] -- saved at the Github site
#match.ent <- en.reviewt$business_id %in% bus.lis.t
#reviewt.ent <- en.reviewt[match.ent,] -- saved at the Github site
```

We then combine all the reviews for a given business. "tm.lexicon.GeneralInquirer" package is used to find positive and negative sentiment counts for reviews of each business.

```
reviewt.stt <- readRDS("reviewt.stt.rds")
reviewt.ent <- readRDS("reviewt.ent.rds")
bus.lis.t <- readRDS("bus.lis.t.rds")
rev.t.f <- factor(reviewt.stt$business_id)
# combine reviews for a given business, first for 'start' set of businesses
st.tx.all <- function(x){
  reviewt.st <- reviewt.stt[reviewt.stt$business_id ==x, ]
  text.st <- as.list(reviewt.st$text)
  st.tx.all <-  do.call(paste, c(text.st))
  }
reviews <- tapply(reviewt.stt$business_id, rev.t.f, st.tx.all)

# calculate positive and negative sentiment counts
require("tm.lexicon.GeneralInquirer")
st <- VCorpus(VectorSource(reviews), readerControl = list(language = "en"))
pos.st <- sapply(st, tm_term_score, terms_in_General_Inquirer_categories("Positiv"))
neg.st <- sapply(st, tm_term_score, terms_in_General_Inquirer_categories("Negativ"))

# repeat for ending text reviews
rev.e.f <- factor(reviewt.ent$business_id)
en.tx.all <- function(x){
  reviewt.en <- reviewt.ent[reviewt.ent$business_id ==x, ]
  text.en <- as.list(reviewt.en$text)
  en.tx.all <-  do.call(paste, c(text.en))
}
reviewe <- tapply(reviewt.ent$business_id, rev.e.f, en.tx.all)

#require("tm.lexicon.GeneralInquirer") -positive & negative sentiment counts
en <- VCorpus(VectorSource(reviewe), readerControl = list(language = "en"))
pos.en <- sapply(en, tm_term_score, terms_in_General_Inquirer_categories("Positiv"))
neg.en <- sapply(en, tm_term_score, terms_in_General_Inquirer_categories("Negativ"))
```

Compute the ratio of the positive to negative counts (after adding a '1' to ensure no errors and a minimum value) as the quantitative equivalent for the textual reviews. Also, order them alphabetically so paired comparisions can be made for start and end reviews.

```
txt.st.rt <- (pos.st+1)/(neg.st+1)
bus.t.un <- tapply(reviewt.stt$business_id, rev.t.f, unique)
st.bus.t.rt <- data.frame(txt.st.rt, bus.t.un)
st.order.t.rt <-  st.bus.t.rt[order(st.bus.t.rt$bus.t.un),]
# order them for end
txt.en.rt <- (pos.en+1)/(neg.en+1)
bus.e.un <- tapply(reviewt.ent$business_id, rev.e.f, unique)
#head(en.order$stars.e.bus); tail(en.order$stars.e.bus)
en.bus.t.rt <- data.frame(txt.en.rt, bus.e.un)
en.order.t.rt <-  en.bus.t.rt[order(en.bus.t.rt$bus.e.un),]
```

Results - We document here the results for both star and textual review ratings. The mean and sd values are listed first. Results for a paired sample t-test statistic are provided next for 1156 businesses; this compares the start value with the end value, as averaged over 24 and 6 months respectively. Violin Plots are created next. The violin plots use black colored boxes overlapping the vertical line of symmetry to show 1 SD above and below the mean.

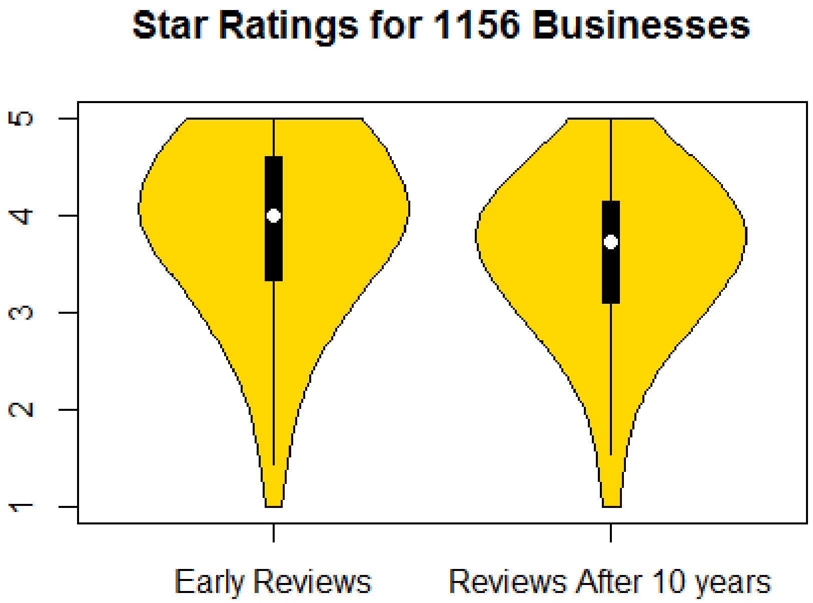
```
# t-test statistic for paired samples for star ratings
y1<- st.order$stars.s.avg; mean(y1); sd(y1)

## [1] 3.895769

## [1] 0.9257539

y2<- en.order$stars.e.avg; mean(y2); sd(y2)

## [1] 3.58098
## [1] 0.8250258
```

```
t.test(y1,y2, paired=TRUE)
```

```
## 
##  Paired t-test
## 
## data:  y1 and y2
## t = 9.745, df = 1155, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2514108 0.3781677
## sample estimates:
## mean of the differences 
##               0.3147892
```

```
library(vioplot)
```

```
vioplot(y1,y2, names=c("Early Reviews", "Reviews After 10 years"),col="gold")
title("Star Ratings for 1156 Businesses")
```



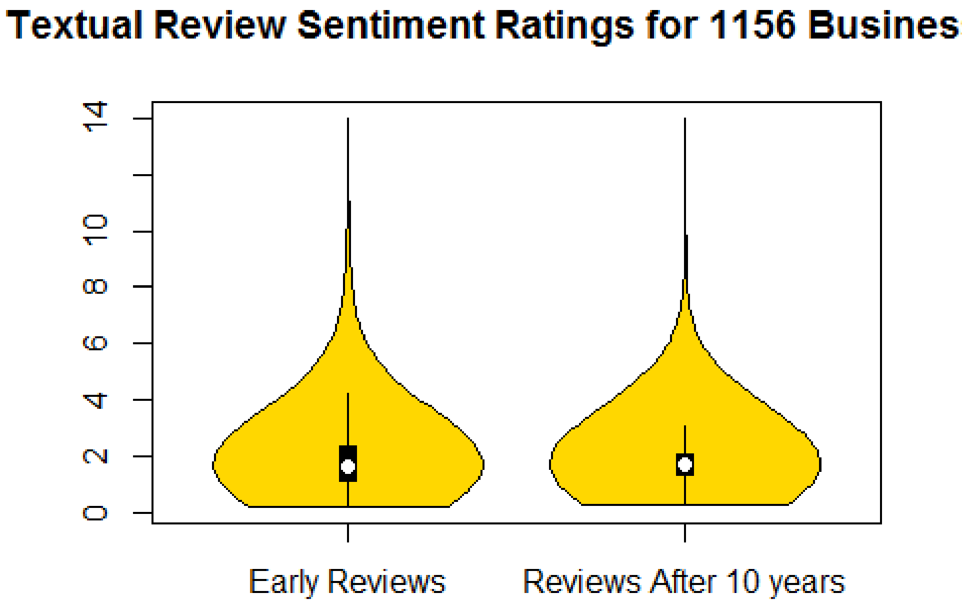
```
# t-test statistic for paired samples for textual review sentiment polarity ratings
y1.t.rt<- st.order.t.rt$txt.st.rt; mean(y1.t.rt);sd(y1.t.rt)
```

```
## [1] 1.969881
```

```
## [1] 1.34316
```

```
y2.t.rt<- en.order.t.rt$txt.en.rt; mean(y2.t.rt);sd(y2.t.rt)
```

```
## [1] 1.805578
## [1] 0.8547988

t.test(y1.t.rt,y2.t.rt, paired=TRUE)

##
##  Paired t-test
##
## data:  y1.t.rt and y2.t.rt
## t = 3.6141, df = 1155, p-value = 0.0003144
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.07510607 0.25350002
## sample estimates:
## mean of the differences
##                0.164303

vioplot(y1.t.rt,y2.t.rt, names=c("Early Reviews", "Reviews After 10
years"),col="gold")
title("Textual Review Sentiment Ratings for 1156 Businesses")
```



Discussion - The t-test statistics and plots show that the businesses did regress in both mean and sd values, with regard to both quantitative and qualitative ratings. Both pairs of distributions were shown to be significantly different at p <0.001, with the end reviews lower in both mean and sd. The 'regression to the mean' was first noticed by Sir Galton when he compared the heights of parents and their children. Our study explored and validated the same principle for reviews of businesses. One limitation of our study may be noted: Despite using two years of data at the start, there were significantly fewer reviews for all the businesses at the start relative to the end six months of review, presumably because Yelp was also starting ten years ago. The two sets of distributions seem inverted relative to each other.