Yelpers in The Great Recession: A Case Study

2/7/2017

outline

intro data set explantions yelp data fred analysis yelp users time series yelp reviews time series yelp review stars time series fred data for recession analysis yelp stock data conclusion

Abstract

Introduction

This is a case study into Yelp restaurantgoers' consumer behavior during The Great Recession based off the Yelp Challenge 9.

Data Sets

Yelp Challenge 9:

FRED:

Yahoo Finance:

BEA:

Challenges and Data Issues Addressed

Early Yelp data is not very good Most recent Yelp data does not include entire month

subset relevant data from begining of 2006 to end of 2016

Recession only happened in US, data from worldwide

Motivation

If restaurant Yelpers' behavior during and aroud The Great Recession period can be modeled,then there can be conclusions about Yelpers' behavior. If recessionary behavior can be understood, then restaurants can react accordingly.

The Process

After gathering the data, I examined all the potentially relevant information.

Then gathered any additional data that may help

Then test to see if assumptions about the inner workings and dynamics were true

Finally, try to model the consumer's behavior

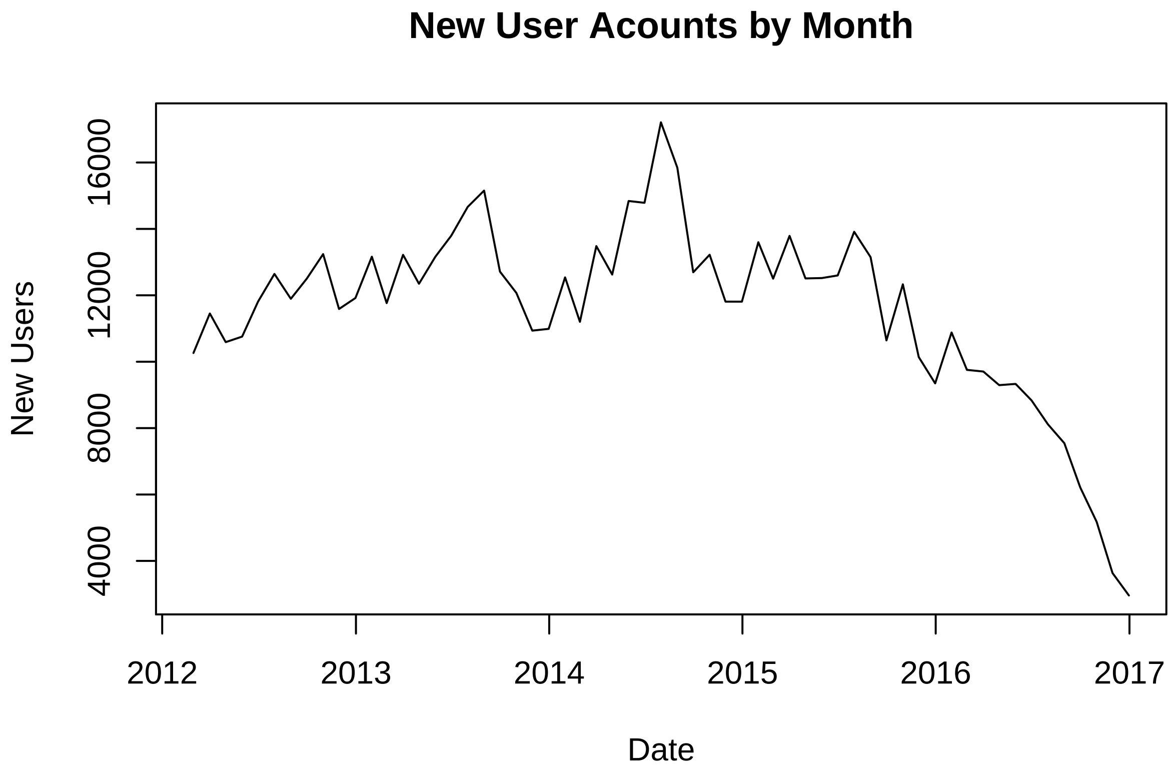
```r
# SETUP
setwd("C:/cygwin64/home/Lester/yelp_challenge_9")

# load data example json_file =
# file('yelp_academic_dataset_checkin.json') json_data =
# jsonlite::stream_in(json_file) head(json_data)
# length(json_data$business_id)

load_json = function(filename) {
    json_file = file(filename)
    json_data = jsonlite::stream_in(json_file)
    return(json_data)
}

# business = load_json('yelp_academic_dataset_business.json')
# review = load_json('yelp_academic_dataset_review.json')
# checkin = load_json('yelp_academic_dataset_checkin.json')
# tip = load_json('yelp_academic_dataset_tip.json') user =
# load_json('yelp_academic_dataset_user.json')

remove_lists_from_df = function(df) {
    i = 1
    while (i <= length(df)) {
        if (class(df[, i]) == "list") {
            df[i] = sapply(df[, i], paste, collapse = "|")
        }
        i = i + 1
    }

    return(df)
}


add_recession_dummy = function(l) {
    rec = c()
    for (i in 1:length(l)) {

        if (l[i] >= as.Date("2007-12-01") & l[i] <= as.Date("2009-07-01")) {
            rec = c(rec, 1)
        } else {
            rec = c(rec, 0)
        }
    }

    return(rec)

}
```

user data

Users, Stocks, & Reviews

First, I will explore the Yelp stock data to see if it is of any relevance in an attempt to handle any cases of endogeneity.

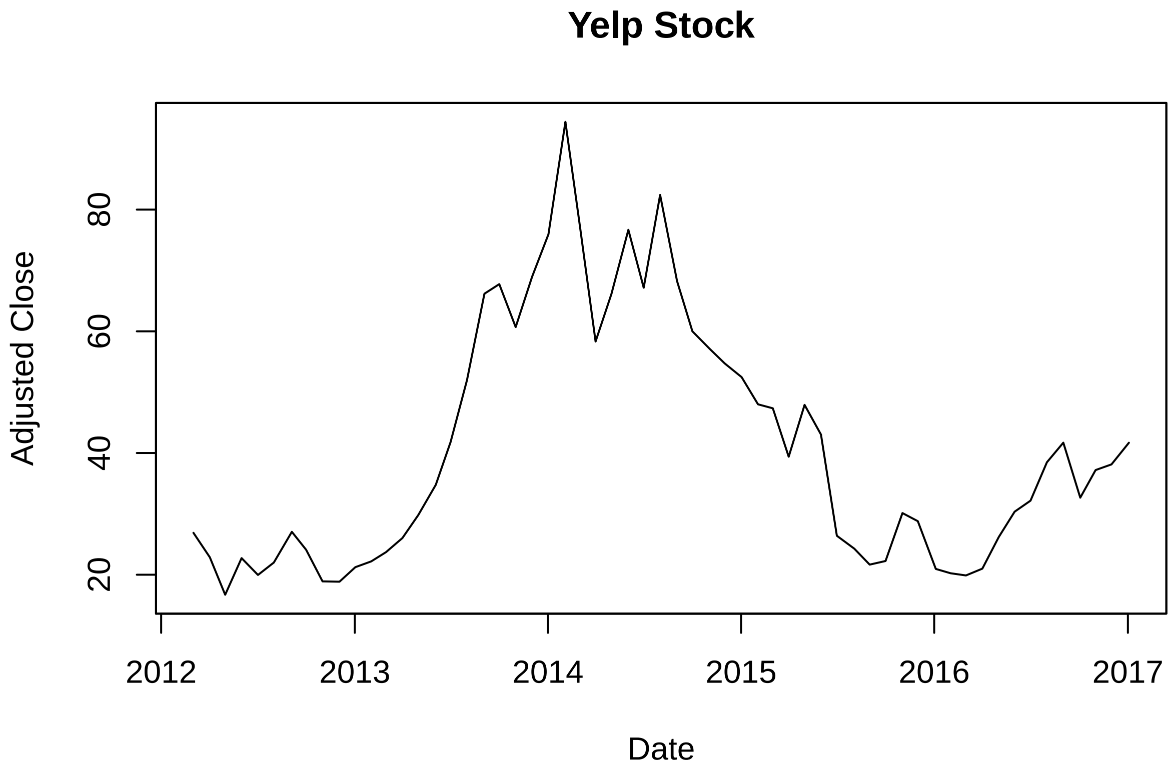
```
plot(stock_users, type = "l", ylab = "New Users", xlab = "Date",
    main = "New User Acounts by Month")
```



```
yelp_stock = get.hist.quote("YELP", end = "2017-01-30", quote = "AdjClose",
    compression = "m")
```

```
## time series starts 2012-03-02
## time series ends   2017-01-03
```

```
# yelp_stock

plot(yelp_stock, ylab = "Adjusted Close", xlab = "Date", main = "Yelp Stock")
```



```
test_stationary = function(t) {
    print(kpss.test(t))
    print(adf.test(t))
}

test_cointegration = function(resid) {
    print(test_stationary(resid))
}

test_stationary(ts(stock_users$coredata.ts_m., start = c(2012,
    3), freq = 12))
```

```
## Warning in kpss.test(t): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 1.1995, Truncation lag parameter = 1, p-value = 0.01
##
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -1.0302, Lag order = 3, p-value = 0.9257
## alternative hypothesis: stationary
```

```r
test_stationary(yelp_stock)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 0.53728, Truncation lag parameter = 1, p-value =
## 0.03327
##
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -1.4945, Lag order = 3, p-value = 0.7789
## alternative hypothesis: stationary
```

```r
log_user_growth = as.data.frame(diff(log(stock_users$coredata.ts_m.)))

log_yelp_growth = as.data.frame(diff(log(yelp_stock)))

# log_user_growth log_yelp_growth




ts_users = ts(log_user_growth, start = c(2012, 4), freq = 12)
test_stationary(ts_users)
```

```
## Warning in kpss.test(t): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 0.8162, Truncation lag parameter = 1, p-value = 0.01
##
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -2.5657, Lag order = 3, p-value = 0.3463
## alternative hypothesis: stationary
```

```r
ts_yelp = ts(log_yelp_growth, start = c(2012, 4), freq = 12)
test_stationary(ts_yelp)
```

```
## Warning in kpss.test(t): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 0.13073, Truncation lag parameter = 1, p-value = 0.1
##
##
##  Augmented Dickey-Fuller Test
```

```
##
## data:  t
## Dickey-Fuller = -3.9873, Lag order = 3, p-value = 0.01641
## alternative hypothesis: stationary
```

```r
combined = cbind(ts_yelp, ts_users)
select = VARselect(combined, lag.max = 12, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
vm = VAR(combined, p = select$select[1])
# plot(vm$y)
summary(vm)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: ts_yelp, ts_users
## Deterministic variables: const
## Sample size: 46
## Log Likelihood: 104.197
## Roots of the characteristic polynomial:
## 1.121 0.9968 0.9968 0.9859 0.9859 0.9739 0.9739 0.9716 0.9716 0.9556 0.9556 0.9471 0.9423 0.9423 0.93
## Call:
## VAR(y = combined, p = select$select[1])
##
##
## Estimation results for equation ts_yelp:
## ========================================
## ts_yelp = ts_yelp.l1 + ts_users.l1 + ts_yelp.l2 + ts_users.l2 + ts_yelp.l3 + ts_users.l3 + ts_yelp.l4
##
##              Estimate Std. Error t value Pr(>|t|)
## ts_yelp.l1    0.22443    0.24350   0.922    0.367
## ts_users.l1   0.27950    0.43228   0.647    0.525
## ts_yelp.l2   -0.02723    0.26692  -0.102    0.920
## ts_users.l2  -0.12729    0.44781  -0.284    0.779
## ts_yelp.l3    0.06446    0.25133   0.256    0.800
## ts_users.l3  -0.11399    0.47105  -0.242    0.811
## ts_yelp.l4   -0.15175    0.26027  -0.583    0.566
## ts_users.l4   0.41380    0.39727   1.042    0.309
## ts_yelp.l5    0.34786    0.24875   1.398    0.177
## ts_users.l5  -0.04790    0.41595  -0.115    0.909
## ts_yelp.l6    0.23889    0.29288   0.816    0.424
## ts_users.l6  -0.01589    0.38874  -0.041    0.968
## ts_yelp.l7   -0.07162    0.29937  -0.239    0.813
## ts_users.l7  -0.33711    0.38189  -0.883    0.387
## ts_yelp.l8    0.01820    0.29149   0.062    0.951
## ts_users.l8  -0.22491    0.40855  -0.551    0.588
## ts_yelp.l9    0.07345    0.27828   0.264    0.794
## ts_users.l9   0.40564    0.37089   1.094    0.286
## ts_yelp.l10  -0.15597    0.24322  -0.641    0.528
## ts_users.l10  0.31981    0.38672   0.827    0.418
## ts_yelp.l11  -0.16642    0.22890  -0.727    0.475
## ts_users.l11 -0.26574    0.41547  -0.640    0.529
## ts_yelp.l12  -0.13732    0.23143  -0.593    0.559
## ts_users.l12  0.05773    0.44119   0.131    0.897
## const         0.01366    0.03194   0.428    0.673
```

```
##
##
## Residual standard error: 0.1904 on 21 degrees of freedom
## Multiple R-Squared: 0.411,   Adjusted R-squared: -0.2621
## F-statistic: 0.6106 on 24 and 21 DF,  p-value: 0.8779
##
##
## Estimation results for equation ts_users:
## =========================================
## ts_users = ts_yelp.l1 + ts_users.l1 + ts_yelp.l2 + ts_users.l2 + ts_yelp.l3 + ts_users.l3 + ts_yelp.l
##
##               Estimate Std. Error t value Pr(>|t|)
## ts_yelp.l1   -0.158567   0.103539  -1.531  0.14058
## ts_users.l1   0.062517   0.183814   0.340  0.73715
## ts_yelp.l2   -0.034134   0.113501  -0.301  0.76657
## ts_users.l2   0.353135   0.190419   1.855  0.07776 .
## ts_yelp.l3    0.004447   0.106872   0.042  0.96720
## ts_users.l3   0.185054   0.200299   0.924  0.36604
## ts_yelp.l4   -0.072499   0.110674  -0.655  0.51954
## ts_users.l4  -0.073652   0.168926  -0.436  0.66728
## ts_yelp.l5   -0.163585   0.105773  -1.547  0.13691
## ts_users.l5   0.041078   0.176871   0.232  0.81859
## ts_yelp.l6    0.235357   0.124538   1.890  0.07266 .
## ts_users.l6   0.140393   0.165300   0.849  0.40528
## ts_yelp.l7   -0.105272   0.127298  -0.827  0.41755
## ts_users.l7   0.428628   0.162388   2.640  0.01533 *
## ts_yelp.l8    0.053978   0.123949   0.435  0.66765
## ts_users.l8   0.031861   0.173722   0.183  0.85624
## ts_yelp.l9   -0.015866   0.118330  -0.134  0.89462
## ts_users.l9  -0.097292   0.157710  -0.617  0.54393
## ts_yelp.l10   0.053363   0.103420   0.516  0.61126
## ts_users.l10  0.264388   0.164440   1.608  0.12281
## ts_yelp.l11   0.092298   0.097331   0.948  0.35377
## ts_users.l11  0.439468   0.176665   2.488  0.02134 *
## ts_yelp.l12  -0.029449   0.098410  -0.299  0.76769
## ts_users.l12  0.662858   0.187602   3.533  0.00197 **
## const        -0.009345   0.013580  -0.688  0.49886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.08095 on 21 degrees of freedom
## Multiple R-Squared: 0.798,   Adjusted R-squared: 0.5672
## F-statistic: 3.457 on 24 and 21 DF,  p-value: 0.002768
##
##
##
## Covariance matrix of residuals:
##           ts_yelp ts_users
## ts_yelp  0.036241 0.007759
## ts_users 0.007759 0.006553
##
## Correlation matrix of residuals:
##          ts_yelp ts_users
## ts_yelp   1.0000   0.5035
## ts_users  0.5035   1.0000
```

```
grangertest(ts_users ~ ts_yelp)
```

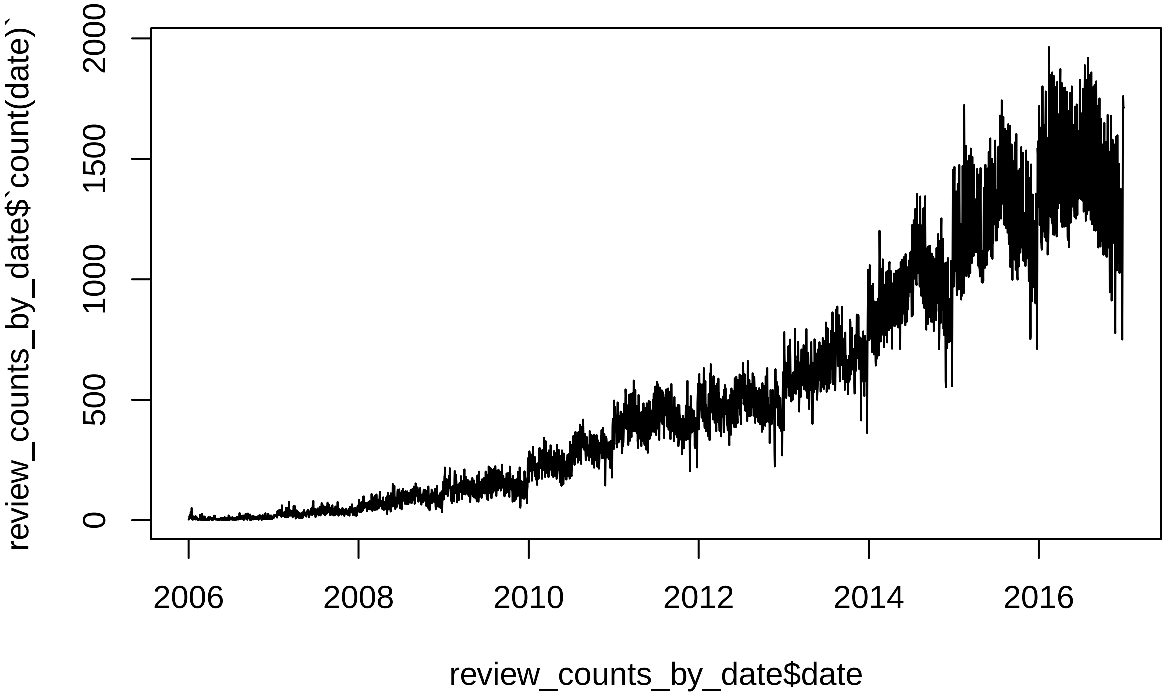
```
## Granger causality test
##
## Model 1: ts_users ~ Lags(ts_users, 1:1) + Lags(ts_yelp, 1:1)
## Model 2: ts_users ~ Lags(ts_users, 1:1)
##   Res.Df Df      F Pr(>F)
## 1     54
## 2     55 -1 0.2396 0.6265
```

```
# users and stock no effects

# sp500 =
# get.hist.quote('^GSPC',quote='AdjClose',compression='m',start=as.Date('2012-03-01','%Y-%m-%d'),end=as
# # sp500 test_stationary(sp500) log_sp500_growth =
# as.data.frame(diff(log(sp500))) ts_sp500 =
# ts(log_sp500_growth,start=c(2012,4),freq=12) # ts_sp500
# test_stationary(ts_sp500) # ts_yelp # ts_sp500
# plot(ts_users,col='blue') lines(ts_yelp,col='red') combined
# = cbind(ts_yelp,ts_sp500)
# select=VARselect(combined,lag.max=12,type=c('const','trend','both','none'),season=NULL,exogen=NULL)
# vm=VAR(combined,p=select$select[1]) plot(vm$y) summary(vm)
```

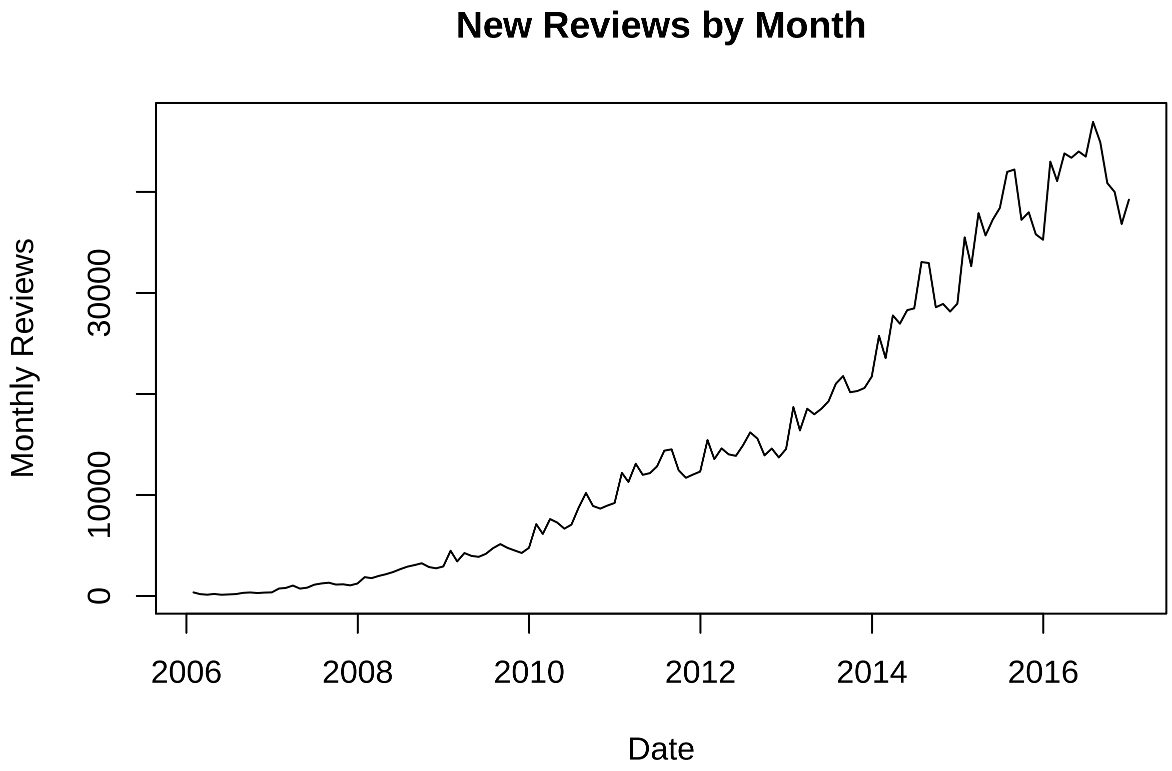
Growth rate of users do not have an effect on the stock value of Yelp - no correlation to stock market. good! This elinates the possibility that effects are due to yelp as a company doing good or bad. Eliminate stock data from our study

review scores and gdp

```
plot(review_counts_by_date$date, review_counts_by_date$`count(date)`,
    type = "l")
```



```
# convert to monthly
reviews_by_date = xts(review_counts_by_date$`count(date)`, as.Date(review_counts_by_date$date,
    "%Y-%m-%d"))
df_rev_m = apply.monthly(reviews_by_date, sum)
df_rev_count = data.frame(date = index(df_rev_m), coredata(df_rev_m))
# df_rev_count

plot(df_rev_count$date, df_rev_count$coredata.df_rev_m., type = "l",
    xlab = "Date", ylab = "Monthly Reviews", main = "New Reviews by Month")
```



```
# review growth rates/new user growth rates
log_rev_count = diff(log(df_rev_count$coredata.df_rev_m.))
# log_rev_count[1]=NA
log_rev_count = na.omit(log_rev_count)
log_rev_count = ts(log_rev_count, start = c(2006, 2), freq = 12)

# ts_m
log_user_count = diff(log(ts_m[, 1]))
log_user_count = na.omit(log_user_count)
log_user_count = ts(log_user_count, start = c(2006, 2), freq = 12)

# plot(log_rev_count,type='l', main='growth rate of user
# reviews and accounts') lines(log_user_count[,1],col='red')

# do a var model between growth rate of users revs and
# accounts

# create var of growth rates
rates_combined = cbind(log_rev_count, log_user_count)
select = VARselect(rates_combined, lag.max = 12, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
vm_rates = VAR(rates_combined, select$select[1])
# plot(vm_rates$y)
summary(vm_rates)
```

```
##
```

```
## VAR Estimation Results:
## =========================
## Endogenous variables: log_rev_count, log_user_count
## Deterministic variables: const
## Sample size: 119
## Log Likelihood: 297.714
## Roots of the characteristic polynomial:
## 1.048 0.9737 0.9737 0.9716 0.9716  0.97  0.97 0.9687 0.9581 0.9581 0.9101 0.9101 0.8859 0.8859 0.857
## Call:
## VAR(y = rates_combined, p = select$select[1])
##
##
## Estimation results for equation log_rev_count:
## ==============================================
## log_rev_count = log_rev_count.l1 + log_user_count.l1 + log_rev_count.l2 + log_user_count.l2 + log_rev
##
##                     Estimate Std. Error t value Pr(>|t|)
## log_rev_count.l1   -0.447591   0.102482  -4.367 3.23e-05 ***
## log_user_count.l1   0.073934   0.092299   0.801 0.425138
## log_rev_count.l2   -0.140889   0.109751  -1.284 0.202398
## log_user_count.l2   0.208054   0.097639   2.131 0.035711 *
## log_rev_count.l3    0.024924   0.105031   0.237 0.812936
## log_user_count.l3  -0.053311   0.095380  -0.559 0.577540
## log_rev_count.l4    0.210029   0.098349   2.136 0.035316 *
## log_user_count.l4  -0.405138   0.096096  -4.216 5.72e-05 ***
## log_rev_count.l5    0.005835   0.095989   0.061 0.951655
## log_user_count.l5  -0.071718   0.102943  -0.697 0.487721
## log_rev_count.l6    0.113966   0.094900   1.201 0.232807
## log_user_count.l6   0.108670   0.101916   1.066 0.289033
## log_rev_count.l7    0.115462   0.094805   1.218 0.226318
## log_user_count.l7   0.124041   0.103038   1.204 0.231678
## log_rev_count.l8   -0.084966   0.095834  -0.887 0.377561
## log_user_count.l8   0.003598   0.103881   0.035 0.972444
## log_rev_count.l9   -0.243305   0.090552  -2.687 0.008528 **
## log_user_count.l9   0.122566   0.101162   1.212 0.228713
## log_rev_count.l10  -0.184751   0.095714  -1.930 0.056592 .
## log_user_count.l10  0.299530   0.100755   2.973 0.003748 **
## log_rev_count.l11  -0.132124   0.086128  -1.534 0.128376
## log_user_count.l11  0.214219   0.101571   2.109 0.037598 *
## log_rev_count.l12   0.092019   0.066103   1.392 0.167189
## log_user_count.l12  0.373805   0.096160   3.887 0.000189 ***
## const               0.032366   0.013296   2.434 0.016811 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.07862 on 94 degrees of freedom
## Multiple R-Squared: 0.6877,  Adjusted R-squared: 0.6079
## F-statistic: 8.623 on 24 and 94 DF,  p-value: 7.305e-15
##
##
## Estimation results for equation log_user_count:
## ===============================================
## log_user_count = log_rev_count.l1 + log_user_count.l1 + log_rev_count.l2 + log_user_count.l2 + log_r
```

```
##
##                     Estimate Std. Error t value Pr(>|t|)
## log_rev_count.l1   -0.228340   0.108927  -2.096  0.03874 *
## log_user_count.l1   0.085515   0.098103   0.872  0.38560
## log_rev_count.l2    0.109457   0.116653   0.938  0.35049
## log_user_count.l2   0.032154   0.103779   0.310  0.75738
## log_rev_count.l3    0.155007   0.111635   1.389  0.16826
## log_user_count.l3   0.086143   0.101378   0.850  0.39764
## log_rev_count.l4    0.212001   0.104533   2.028  0.04538 *
## log_user_count.l4  -0.139470   0.102139  -1.365  0.17536
## log_rev_count.l5   -0.003095   0.102025  -0.030  0.97586
## log_user_count.l5   0.064438   0.109416   0.589  0.55732
## log_rev_count.l6   -0.001041   0.100868  -0.010  0.99179
## log_user_count.l6   0.187274   0.108324   1.729  0.08712 .
## log_rev_count.l7   -0.085882   0.100767  -0.852  0.39622
## log_user_count.l7   0.260097   0.109517   2.375  0.01958 *
## log_rev_count.l8   -0.212682   0.101861  -2.088  0.03951 *
## log_user_count.l8   0.012833   0.110413   0.116  0.90772
## log_rev_count.l9   -0.310263   0.096246  -3.224  0.00174 **
## log_user_count.l9   0.237115   0.107524   2.205  0.02988 *
## log_rev_count.l10  -0.282063   0.101732  -2.773  0.00671 **
## log_user_count.l10  0.349787   0.107091   3.266  0.00152 **
## log_rev_count.l11  -0.161645   0.091543  -1.766  0.08068 .
## log_user_count.l11  0.258147   0.107957   2.391  0.01879 *
## log_rev_count.l12   0.256980   0.070259   3.658  0.00042 ***
## log_user_count.l12  0.257381   0.102207   2.518  0.01348 *
## const              -0.008062   0.014132  -0.570  0.56971
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.08356 on 94 degrees of freedom
## Multiple R-Squared: 0.618,   Adjusted R-squared: 0.5204
## F-statistic: 6.336 on 24 and 94 DF,  p-value: 3.029e-11
##
##
##
## Covariance matrix of residuals:
##                log_rev_count log_user_count
## log_rev_count       0.006181       0.002504
## log_user_count      0.002504       0.006982
##
## Correlation matrix of residuals:
##                log_rev_count log_user_count
## log_rev_count         1.0000         0.3811
## log_user_count        0.3811         1.0000
```

```
# do granger causality test
grangertest(log_rev_count ~ log_user_count, order = select$select[1])
```

```
## Granger causality test
##
## Model 1: log_rev_count ~ Lags(log_rev_count, 1:12) + Lags(log_user_count, 1:12)
## Model 2: log_rev_count ~ Lags(log_rev_count, 1:12)
##   Res.Df  Df      F    Pr(>F)
## 1     94
## 2    106 -12 5.9487 1.318e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
grangertest(log_user_count ~ log_rev_count, order = select$select[1])
```

```
## Granger causality test
##
## Model 1: log_user_count ~ Lags(log_user_count, 1:12) + Lags(log_rev_count, 1:12)
## Model 2: log_user_count ~ Lags(log_user_count, 1:12)
##   Res.Df  Df      F   Pr(>F)
## 1     94
## 2    106 -12 3.4167 0.000354 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
user_rev_lm = lm(log_rev_count ~ log_user_count)
summary(user_rev_lm)
```

```
##
## Call:
## lm(formula = log_rev_count ~ log_user_count)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68573 -0.04008  0.00958  0.04946  0.41635
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.02131    0.01183   1.801   0.0741 .
## log_user_count  0.77640    0.08763   8.860 5.44e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1341 on 129 degrees of freedom
## Multiple R-squared:  0.3783, Adjusted R-squared:  0.3735
## F-statistic: 78.51 on 1 and 129 DF,  p-value: 5.444e-15
```

```
test_cointegration(resid(user_rev_lm))
```

```
## Warning in kpss.test(t): p-value greater than printed p-value
```

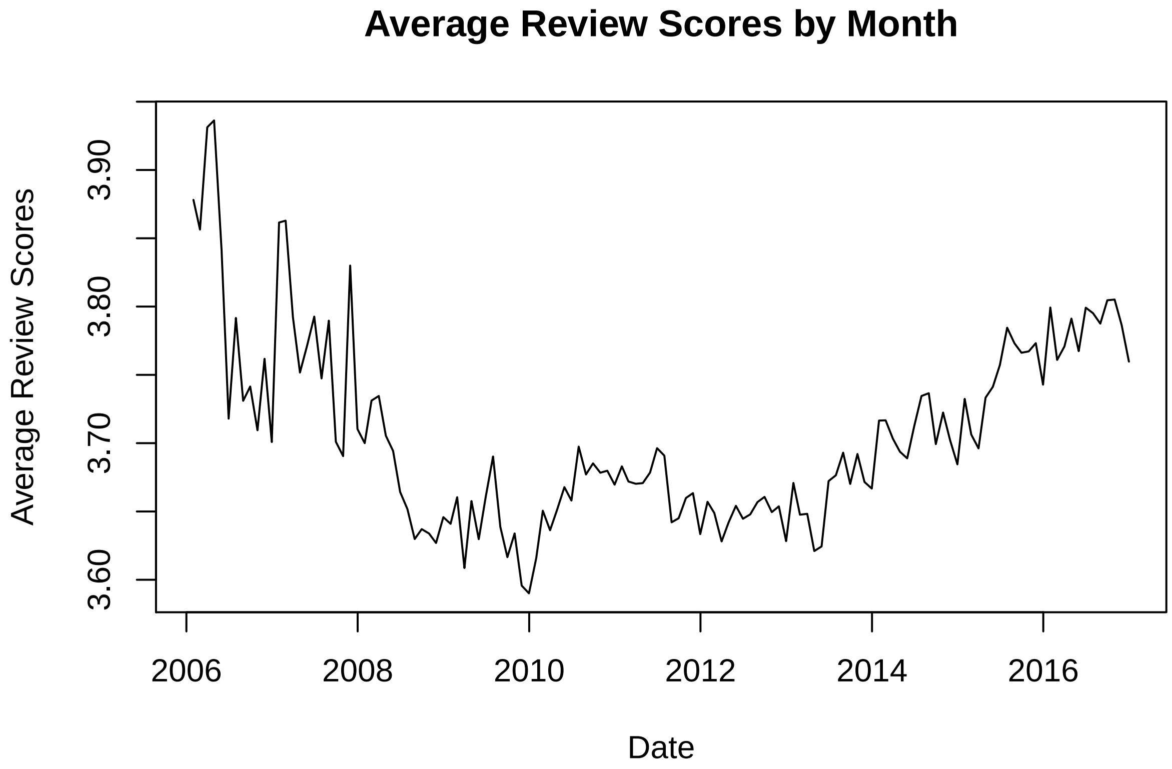
```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 0.27805, Truncation lag parameter = 2, p-value = 0.1
```

```
## Warning in adf.test(t): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -4.91, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
##
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -4.91, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
# users and num reviews granger cause each other and are
# cointegrated, just use one

# stars by month
stars_by_date = xts(review_date_star$stars, as.Date(review_date_star$date,
    "%Y-%m-%d"))
df_stars_m = apply.monthly(stars_by_date, sum)
df_stars = data.frame(date = index(df_stars_m), coredata(df_stars_m))


df_stars$avg = df_stars$coredata.df_stars_m./df_rev_count$coredata.df_rev_m.
# head(df_stars)

plot(df_stars$date, df_stars$avg, type = "l", xlab = "Date",
    ylab = "Average Review Scores", main = "Average Review Scores by Month")
```



```
# evidence of recession in stars
stars_recession_dummy = add_recession_dummy(df_stars$date)
stars_avg = df_stars$avg
stars_lm = lm(stars_avg ~ stars_recession_dummy)
```

```r
summary(stars_lm)
```

```
##
## Call:
## lm(formula = stars_avg ~ stars_recession_dummy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12291 -0.04541 -0.01606  0.04614  0.22330
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.712971   0.006261 593.052  < 2e-16 ***
## stars_recession_dummy -0.048566   0.016502  -2.943  0.00385 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06655 on 130 degrees of freedom
## Multiple R-squared:  0.06246,    Adjusted R-squared:  0.05525
## F-statistic: 8.661 on 1 and 130 DF,  p-value: 0.003851
```

```r
test_stationary(df_stars$avg)
```

```
## Warning in kpss.test(t): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 0.84648, Truncation lag parameter = 2, p-value = 0.01
##
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -1.9674, Lag order = 5, p-value = 0.5901
## alternative hypothesis: stationary
```

```r
# convert to growth rates
df_stars_diff_log = as.data.frame(diff(log(df_stars$avg)))
df_stars_diff_log$date = df_stars$date[2:length(df_stars$date)]

df_stars_diff_log
```

```
##    diff(log(df_stars$avg))       date
## 1            -5.619865e-03 2006-02-28
## 2             1.923989e-02 2006-03-31
## 3             1.265143e-03 2006-04-29
## 4            -2.443172e-02 2006-05-31
## 5            -3.263090e-02 2006-06-30
## 6             1.963358e-02 2006-07-31
## 7            -1.612599e-02 2006-08-31
## 8             2.800839e-03 2006-09-30
## 9            -8.594358e-03 2006-10-31
## 10            1.400202e-02 2006-11-30
## 11           -1.633079e-02 2006-12-31
## 12             4.251363e-02 2007-01-31
## 13             3.453816e-04 2007-02-28
## 14            -1.845148e-02 2007-03-31
## 15            -1.075444e-02 2007-04-30
## 16             5.325714e-03 2007-05-31
## 17             5.534589e-03 2007-06-30
## 18            -1.201243e-02 2007-07-31
## 19             1.122151e-02 2007-08-31
## 20            -2.366109e-02 2007-09-30
## 21            -2.852157e-03 2007-10-31
## 22             3.710066e-02 2007-11-30
## 23            -3.173946e-02 2007-12-31
## 24            -2.795006e-03 2008-01-31
## 25             8.392467e-03 2008-02-29
## 26             8.993568e-04 2008-03-31
## 27            -7.813701e-03 2008-04-30
## 28            -3.015577e-03 2008-05-31
## 29            -8.163412e-03 2008-06-30
## 30            -3.424082e-03 2008-07-31
## 31            -6.027338e-03 2008-08-31
## 32             1.996352e-03 2008-09-30
## 33            -8.728518e-04 2008-10-31
## 34            -1.921703e-03 2008-11-30
## 35             5.202632e-03 2008-12-31
## 36            -1.341539e-03 2009-01-31
## 37             5.339377e-03 2009-02-28
## 38            -1.426190e-02 2009-03-31
## 39             1.348717e-02 2009-04-30
## 40            -7.656413e-03 2009-05-31
## 41             8.582641e-03 2009-06-30
## 42             7.958529e-03 2009-07-31
## 43            -1.406197e-02 2009-08-31
## 44            -6.107051e-03 2009-09-30
## 45             4.783419e-03 2009-10-31
## 46            -1.056028e-02 2009-11-30
## 47            -1.577807e-03 2009-12-31
## 48             7.166652e-03 2010-01-31
## 49             9.542976e-03 2010-02-28
## 50            -3.927418e-03 2010-03-31
## 51             4.125844e-03 2010-04-30
## 52             4.519455e-03 2010-05-31
## 53            -2.684434e-03 2010-06-30
## 54             1.074447e-02 2010-07-31
## 55            -5.534875e-03 2010-08-31
## 56             2.208555e-03 2010-09-30
## 57            -1.857518e-03 2010-10-31
## 58             3.900998e-04 2010-11-30
## 59            -2.770717e-03 2010-12-31
## 60             3.639439e-03 2011-01-31
## 61            -3.016042e-03 2011-02-28
## 62            -4.471740e-04 2011-03-31
## 63             1.281053e-04 2011-04-30
## 64             2.094403e-03 2011-05-31
## 65             4.849414e-03 2011-06-30
## 66           -1.482645e-03 2011-07-31
## 67           -1.330958e-02 2011-08-31
## 68            8.154915e-04 2011-09-30
## 69            4.060745e-03 2011-10-31
## 70            9.731713e-04 2011-11-30
## 71           -8.223230e-03 2011-12-31
## 72            6.484596e-03 2012-01-31
## 73           -2.251225e-03 2012-02-29
## 74           -5.706624e-03 2012-03-31
## 75            3.849061e-03 2012-04-30
## 76            3.306098e-03 2012-05-31
## 77           -2.592167e-03 2012-06-30
## 78            8.750600e-04 2012-07-31
## 79            2.467858e-03 2012-08-31
## 80            1.041843e-03 2012-09-30
## 81           -3.053989e-03 2012-10-31
## 82            1.161721e-03 2012-11-30
## 83           -6.995426e-03 2012-12-31
## 84            1.166825e-02 2013-01-31
## 85           -6.350921e-03 2013-02-28
## 86            1.749019e-04 2013-03-31
## 87           -7.492477e-03 2013-04-30
## 88            9.204614e-04 2013-05-31
## 89            1.312812e-02 2013-06-30
## 90            1.147936e-03 2013-07-31
## 91            4.490404e-03 2013-08-31
## 92           -6.214124e-03 2013-09-30
## 93            5.951240e-03 2013-10-31
## 94           -5.569303e-03 2013-11-30
## 95           -1.303337e-03 2013-12-31
## 96            1.348184e-02 2014-01-31
## 97            4.870529e-05 2014-02-28
## 98           -3.615021e-03 2014-03-31
## 99           -2.585917e-03 2014-04-30
## 100          -1.311054e-03 2014-05-31
## 101           6.378572e-03 2014-06-30
## 102           5.917766e-03 2014-07-31
## 103           5.419283e-04 2014-08-31
## 104          -1.002166e-02 2014-09-30
## 105           6.225047e-03 2014-10-31
## 106          -5.479281e-03 2014-11-30
## 107          -4.762909e-03 2014-12-31
## 108           1.292571e-02 2015-01-31
## 109          -7.020857e-03 2015-02-28
## 110          -2.731912e-03 2015-03-31
## 111           1.000075e-02 2015-04-30
## 112           2.110902e-03 2015-05-31
## 113           4.274085e-03 2015-06-30
## 114           7.234081e-03 2015-07-31
## 115          -3.011319e-03 2015-08-31
## 116          -1.848002e-03 2015-09-30
## 117           2.730000e-04 2015-10-31
## 118           1.595003e-03 2015-11-30
## 119          -8.085294e-03 2015-12-31
## 120            1.497098e-02 2016-01-31
## 121           -1.014415e-02 2016-02-29
## 122            2.609374e-03 2016-03-31
## 123            5.401924e-03 2016-04-30
## 124           -6.305894e-03 2016-05-31
## 125            8.415438e-03 2016-06-30
## 126           -1.068921e-03 2016-07-31
## 127           -2.013443e-03 2016-08-31
## 128            4.508334e-03 2016-09-30
## 129            1.314815e-04 2016-10-31
## 130           -4.937163e-03 2016-11-30
## 131           -7.075452e-03 2016-12-31
```

```r
colnames(df_stars_diff_log) = c("avg", "date")

test_stationary(df_stars_diff_log$avg)
```

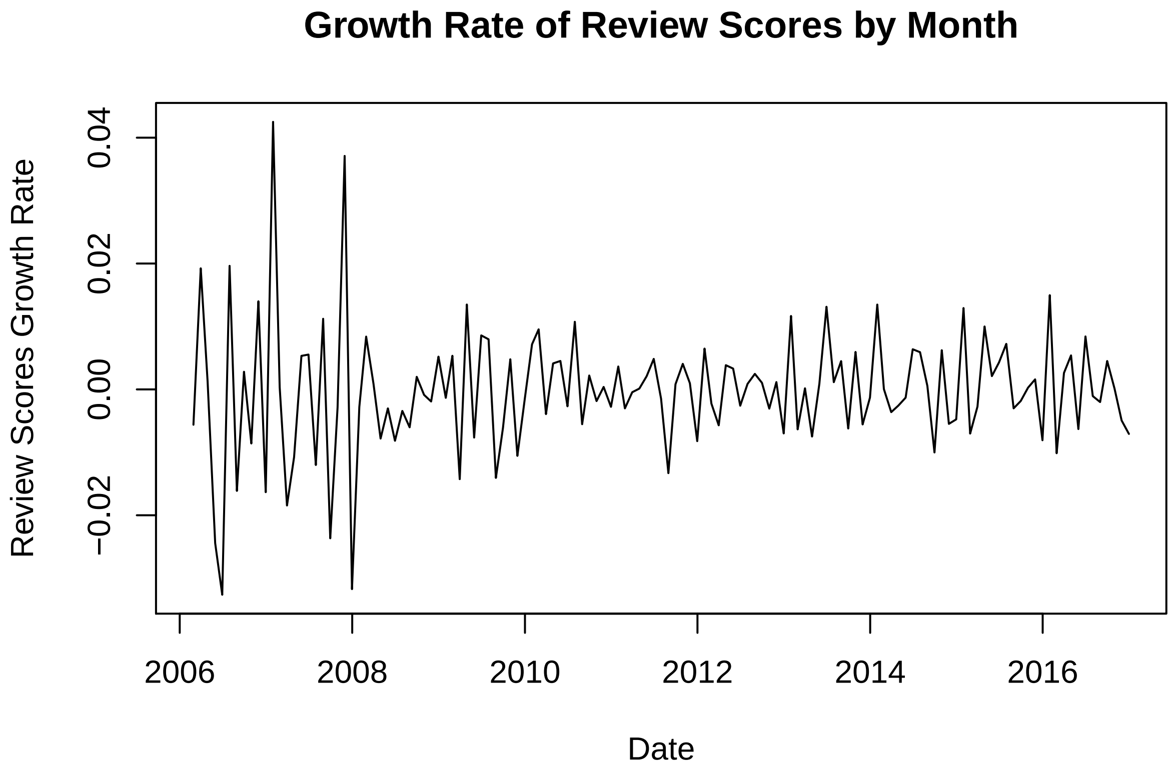
```
## Warning in kpss.test(t): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 0.20563, Truncation lag parameter = 2, p-value = 0.1
```

```
## Warning in adf.test(t): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -5.5779, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```r
stars_recession_diff_log_dummy = add_recession_dummy(df_stars_diff_log$date)

# stars_reg = lm(df_stars_diff_log$avg ~
# stars_recession_diff_log_dummy) summary(stars_reg)

plot(df_stars_diff_log$date, df_stars_diff_log$avg, type = "l",
    main = "Growth Rate of Review Scores by Month", xlab = "Date",
    ylab = "Review Scores Growth Rate")
```



```
stars_diff_log_avg = df_stars_diff_log$avg
stars_diff_log_lm = lm(stars_diff_log_avg ~ stars_recession_diff_log_dummy)
summary(stars_diff_log_lm)
```

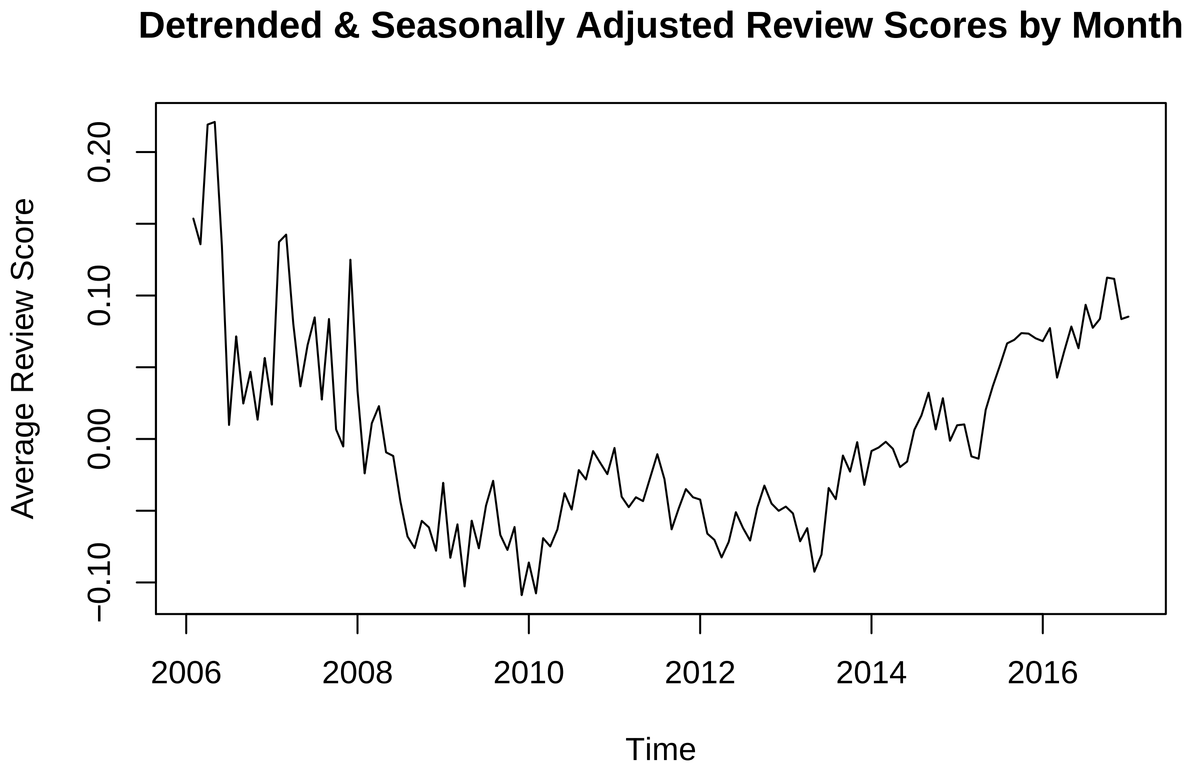
```
##
## Call:
## lm(formula = stars_diff_log_avg ~ stars_recession_diff_log_dummy)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.032757 -0.005360  0.000147  0.004525  0.042388
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    0.0001261  0.0009446   0.133    0.894
## stars_recession_diff_log_dummy -0.0025015  0.0024803  -1.009    0.315
##
## Residual standard error: 0.009997 on 129 degrees of freedom
## Multiple R-squared:  0.007823,   Adjusted R-squared:  0.0001321
## F-statistic: 1.017 on 1 and 129 DF,  p-value: 0.3151
```

```
# however, intuitively we should be looking at level, not
# growth rates so lets detrend the data and season

ts_stars = ts(df_stars$avg, start = c(2006, 2), freq = 12)

stars_tslm = tslm(ts_stars ~ trend + season)
```

```
summary(stars_tslm)
```

```
##
## Call:
## tslm(formula = ts_stars ~ trend + season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.10883 -0.04940 -0.01198  0.04378  0.22099
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.677e+00  2.426e-02 151.563   <2e-16 ***
## trend       -2.068e-05  1.619e-04  -0.128    0.899
## season2      4.736e-02  3.015e-02   1.571    0.119
## season3      4.357e-02  3.014e-02   1.446    0.151
## season4      3.505e-02  3.013e-02   1.163    0.247
## season5      3.822e-02  3.012e-02   1.269    0.207
## season6      2.959e-02  3.011e-02   0.983    0.328
## season7      3.114e-02  3.011e-02   1.034    0.303
## season8      4.315e-02  3.010e-02   1.433    0.154
## season9      2.931e-02  3.010e-02   0.974    0.332
## season10     1.768e-02  3.010e-02   0.587    0.558
## season11     1.906e-02  3.009e-02   0.633    0.528
## season12     2.838e-02  3.009e-02   0.943    0.347
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07057 on 119 degrees of freedom
## Multiple R-squared:  0.03497,    Adjusted R-squared:  -0.06235
## F-statistic: 0.3593 on 12 and 119 DF,  p-value: 0.9748
```

```
# plot(ts_stars) lines(stars_tslm$fitted.values,col='red')

detrend_stars = resid(stars_tslm)
plot(detrend_stars, main = "Detrended & Seasonally Adjusted Review Scores by Month",
    ylab = "Average Review Score")
```



```
detrend_stars_lm = lm(detrend_stars ~ stars_recession_dummy)
summary(detrend_stars_lm)
```

```
##
## Call:
## lm(formula = detrend_stars ~ stars_recession_dummy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.11605 -0.04819 -0.01401  0.05040  0.21376
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.007226   0.006128   1.179  0.24049
## stars_recession_dummy -0.050203   0.016153  -3.108  0.00231 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06514 on 130 degrees of freedom
## Multiple R-squared:  0.06917,    Adjusted R-squared:  0.06201
## F-statistic:  9.66 on 1 and 130 DF,  p-value: 0.002314
```

the growth rates plot shows that even during recession we don't see a dip in either. if there was constant account creation but a dip in reviews, that would mean the recession has an effect on num of reviews

descriptive stats of reviews and stars

```
sd(df_rev_count$coredata.df_rev_m.)
```

```
## [1] 14205.68
```

```
mean(df_rev_count$coredata.df_rev_m.)
```

```
## [1] 15550.11
```

```
sd(log_rev_count)
```

```
## [1] 0.1694621
```

```
mean(log_rev_count)
```

```
## [1] 0.0357887
```

```
sd(df_stars$avg)
```

```
## [1] 0.06847152
```

```
mean(df_stars$avg)
```

```
## [1] 3.705981
```

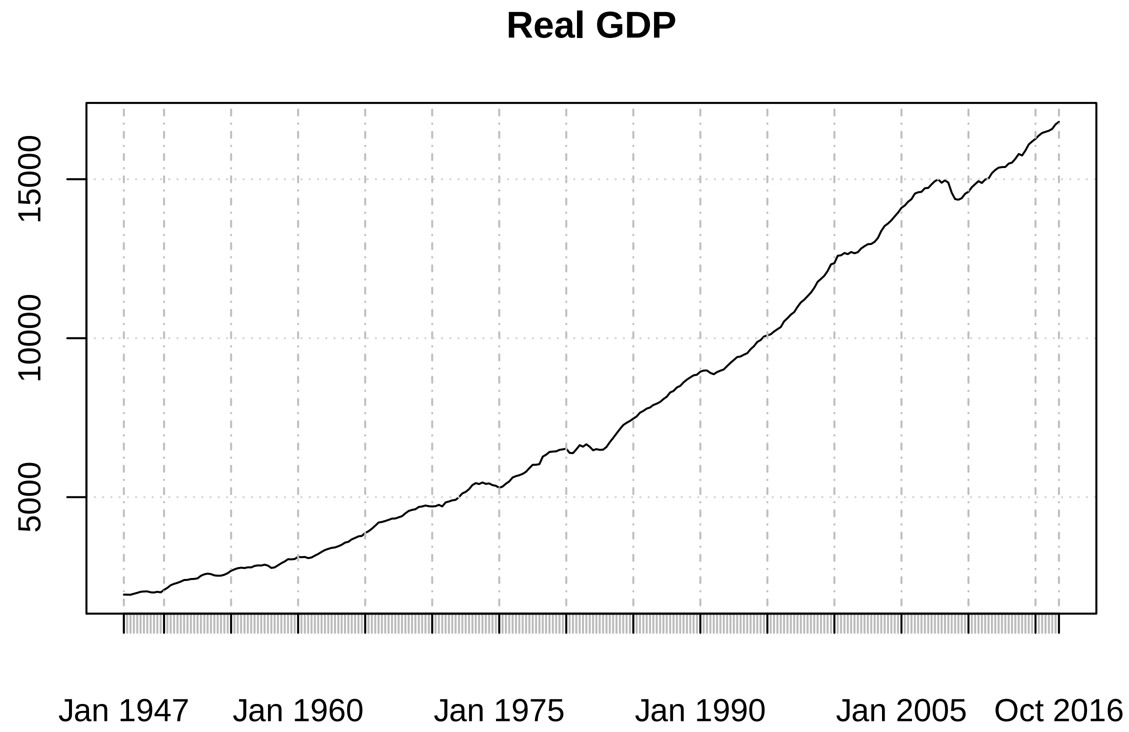
fred data

```
# real gdp
getSymbols("GDPC96", src = "FRED")
```

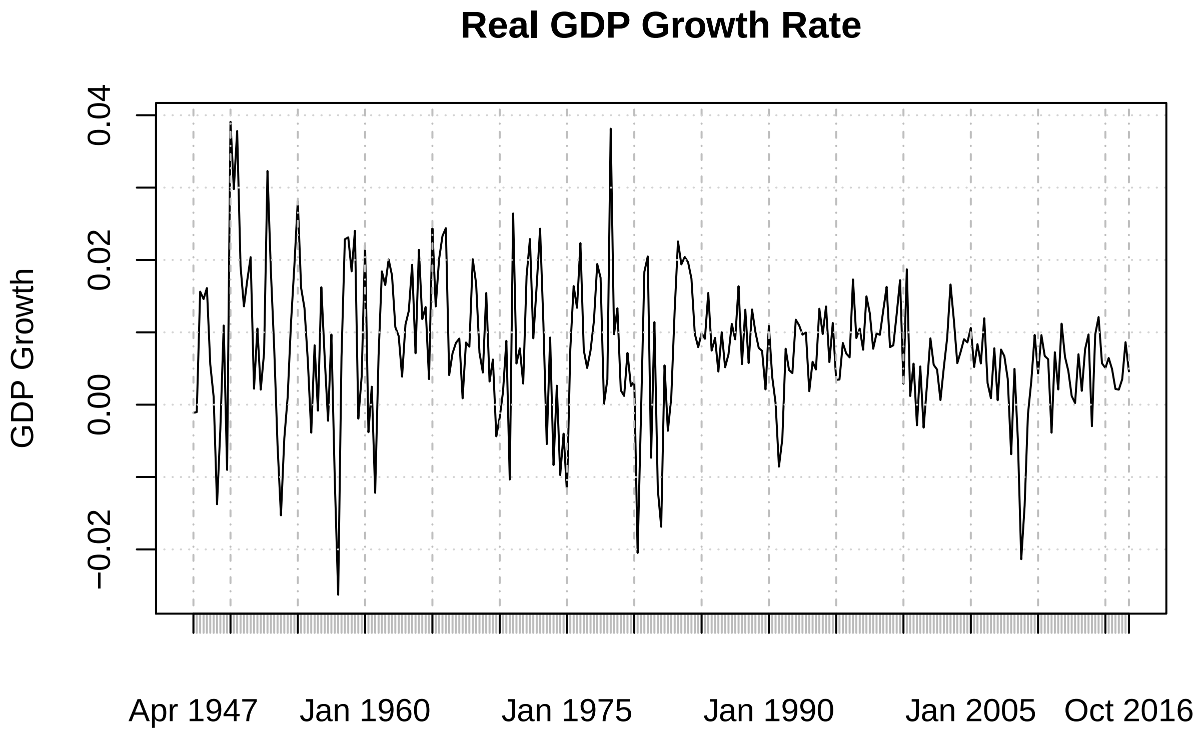
```
##     As of 0.4-0, 'getSymbols' uses env=parent.frame() and
##  auto.assign=TRUE by default.
##
##  This  behavior  will be  phased out in 0.5-0  when the call  will
##  default to use auto.assign=FALSE. getOption("getSymbols.env") and
##  getOptions("getSymbols.auto.assign") are now checked for alternate defaults
##
##  This message is shown once per session and may be disabled by setting
##  options("getSymbols.warning4.0"=FALSE). See ?getSymbols for more details.
```

```
## [1] "GDPC96"
```

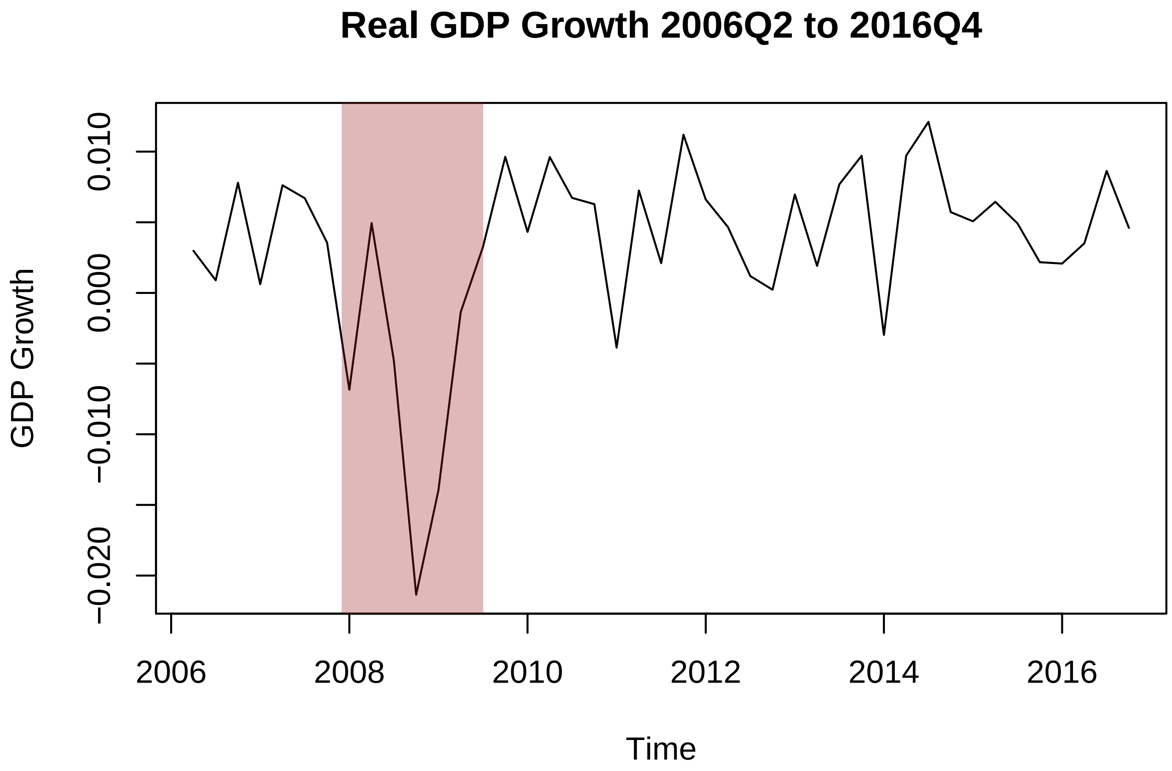
```
gdp = GDPC96

plot(gdp, main = "Real GDP")
```



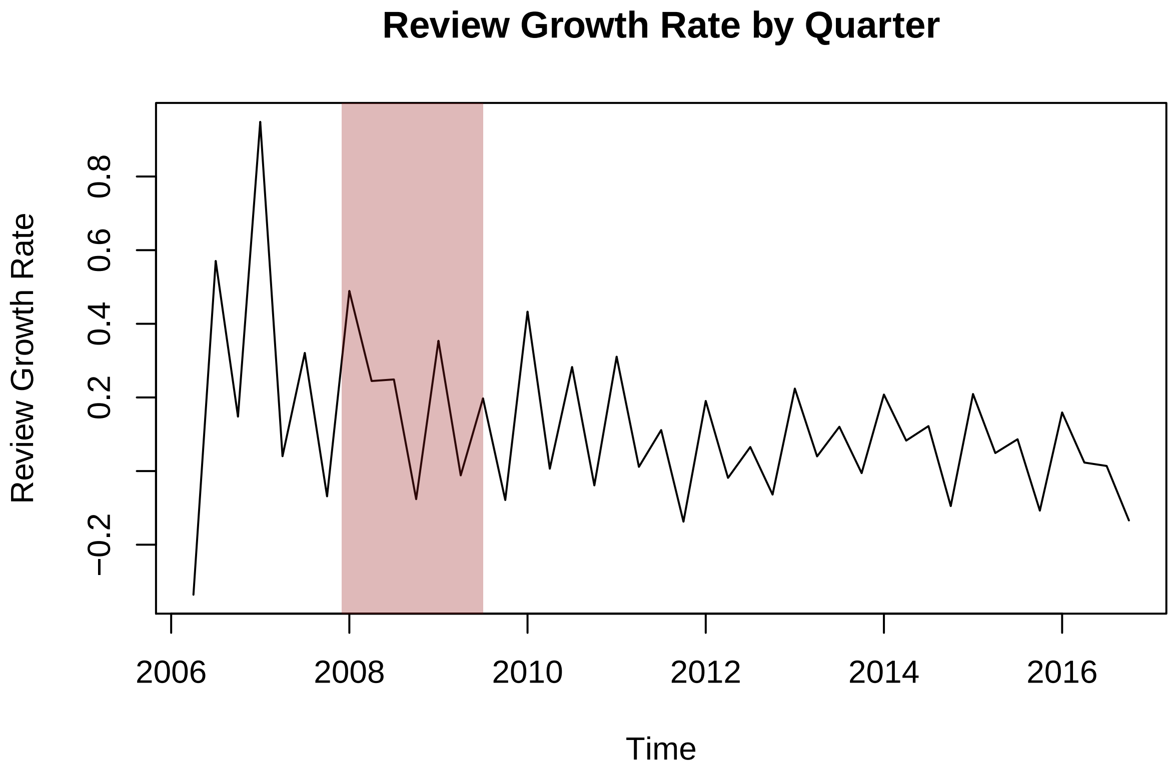
```
gdp_growth = na.omit(diff(log(gdp)))
plot(gdp_growth, ylab = "GDP Growth", main = "Real GDP Growth Rate")
```



```
gdp_growth_subset = with(gdp_growth, gdp_growth[index(gdp_growth) >=
    "2006-04-01" & index(gdp_growth) < "2016-12-30", ])
gdp_growth_subset = ts(gdp_growth_subset, start = c(2006, 2),
    frequency = 4)
plot(gdp_growth_subset, ylab = "GDP Growth", main = "Real GDP Growth 2006Q2 to 2016Q4")
rect(2007.9166667, -1, 2009.5, 1, col = rgb(red = 150/255, green = 25/255,
    blue = 25/255, alpha = 0.3), border = NA)
```



```
# create quarterly review growth rate
reviews_by_quarter = xts(review_counts_by_date$`count(date)`,
    as.Date(review_counts_by_date$date, "%Y-%m-%d"))
df_rev_m_quarter = apply.quarterly(reviews_by_quarter, sum)
df_rev_quarter = data.frame(date = index(df_rev_m_quarter), coredata(df_rev_m_quarter))
# df_rev_quarter
# plot(df_rev_quarter$date,df_rev_quarter$coredata.df_rev_m_quarter.,type='l')

log_rev_quarter = diff(log(df_rev_quarter$coredata.df_rev_m_quarter.))
log_rev_quarter = na.omit(log_rev_quarter)
log_rev_quarter = ts(log_rev_quarter, start = c(2006, 2), freq = 4)
plot(log_rev_quarter, type = "l", ylab = "Review Growth Rate",
    main = "Review Growth Rate by Quarter")
rect(2007.9166667, -1, 2009.5, 1, col = rgb(red = 150/255, green = 25/255,
    blue = 25/255, alpha = 0.3), border = NA)
```



```
# length(log_rev_quarter) length(gdp_growth_subset)

# var of gdp and user reviews growth rates
gdp_combined = cbind(log_rev_quarter, gdp_growth_subset)
select = VARselect(gdp_combined, lag.max = 12, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
vm_gdp = VAR(gdp_combined, p = 12)
# plot(vm_gdp$y)
summary(vm_gdp)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: log_rev_quarter, gdp_growth_subset
## Deterministic variables: const
## Sample size: 31
## Log Likelihood: 238.784
## Roots of the characteristic polynomial:
## 0.9987 0.9987 0.9921 0.9921 0.9906 0.9795 0.9795 0.9699 0.9612 0.9612 0.9576 0.9576 0.9327 0.9327 0.9
## Call:
## VAR(y = gdp_combined, p = 12)
##
##
## Estimation results for equation log_rev_quarter:
## ================================================
## log_rev_quarter = log_rev_quarter.l1 + gdp_growth_subset.l1 + log_rev_quarter.l2 + gdp_growth_subset
```

```
##
##                       Estimate Std. Error t value Pr(>|t|)
## log_rev_quarter.l1     0.41495    0.36251   1.145   0.2960
## gdp_growth_subset.l1  -0.69319    3.48625  -0.199   0.8490
## log_rev_quarter.l2     0.28843    0.37771   0.764   0.4740
## gdp_growth_subset.l2   4.10864    3.15933   1.300   0.2412
## log_rev_quarter.l3     0.27193    0.32528   0.836   0.4352
## gdp_growth_subset.l3   1.55990    3.26731   0.477   0.6500
## log_rev_quarter.l4     0.63886    0.20809   3.070   0.0219 *
## gdp_growth_subset.l4   1.46361    2.96026   0.494   0.6386
## log_rev_quarter.l5    -0.21532    0.23826  -0.904   0.4010
## gdp_growth_subset.l5  -0.43820    3.45264  -0.127   0.9032
## log_rev_quarter.l6    -0.16586    0.22996  -0.721   0.4979
## gdp_growth_subset.l6   4.69612    3.45845   1.358   0.2233
## log_rev_quarter.l7    -0.13165    0.24321  -0.541   0.6078
## gdp_growth_subset.l7  -4.04228    2.95912  -1.366   0.2209
## log_rev_quarter.l8     0.05838    0.24925   0.234   0.8226
## gdp_growth_subset.l8   5.04951    3.49119   1.446   0.1982
## log_rev_quarter.l9     0.11847    0.20277   0.584   0.5803
## gdp_growth_subset.l9   0.05217    3.36605   0.016   0.9881
## log_rev_quarter.l10    0.11208    0.13776   0.814   0.4470
## gdp_growth_subset.l10  6.34267    3.22213   1.968   0.0966 .
## log_rev_quarter.l11   -0.08093    0.15120  -0.535   0.6117
## gdp_growth_subset.l11 -1.75022    2.11397  -0.828   0.4394
## log_rev_quarter.l12    0.43781    0.17667   2.478   0.0479 *
## gdp_growth_subset.l12  2.53313    2.12131   1.194   0.2775
## const                 -0.16460    0.20382  -0.808   0.4502
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.04107 on 6 degrees of freedom
## Multiple R-Squared: 0.9823,  Adjusted R-squared: 0.9117
## F-statistic:  13.9 on 24 and 6 DF,  p-value: 0.001734
##
##
## Estimation results for equation gdp_growth_subset:
## ==================================================
## gdp_growth_subset = log_rev_quarter.l1 + gdp_growth_subset.l1 + log_rev_quarter.l2 + gdp_growth_subs
##
##                        Estimate Std. Error t value Pr(>|t|)
## log_rev_quarter.l1    -0.036465   0.038485  -0.947   0.3800
## gdp_growth_subset.l1  -0.473658   0.370109  -1.280   0.2479
## log_rev_quarter.l2    -0.042996   0.040098  -1.072   0.3248
## gdp_growth_subset.l2  -0.403692   0.335402  -1.204   0.2741
## log_rev_quarter.l3    -0.005267   0.034533  -0.153   0.8838
## gdp_growth_subset.l3  -0.410470   0.346866  -1.183   0.2814
## log_rev_quarter.l4    -0.034836   0.022091  -1.577   0.1659
## gdp_growth_subset.l4  -0.611076   0.314269  -1.944   0.0998 .
## log_rev_quarter.l5    -0.004935   0.025294  -0.195   0.8518
## gdp_growth_subset.l5  -0.259610   0.366540  -0.708   0.5053
## log_rev_quarter.l6     0.026056   0.024413   1.067   0.3269
## gdp_growth_subset.l6  -0.470368   0.367157  -1.281   0.2474
## log_rev_quarter.l7    -0.014837   0.025820  -0.575   0.5864
## gdp_growth_subset.l7  -0.179878   0.314147  -0.573   0.5877
## log_rev_quarter.l8    -0.028504   0.026461  -1.077   0.3228
## gdp_growth_subset.l8  -0.567262   0.370633  -1.531   0.1768
## log_rev_quarter.l9    -0.002516   0.021527  -0.117   0.9108
## gdp_growth_subset.l9  -0.410645   0.357347  -1.149   0.2942
## log_rev_quarter.l10   -0.021819   0.014625  -1.492   0.1863
## gdp_growth_subset.l10 -0.189002   0.342069  -0.553   0.6006
## log_rev_quarter.l11   -0.034597   0.016052  -2.155   0.0746 .
## gdp_growth_subset.l11  0.090608   0.224424   0.404   0.7004
## log_rev_quarter.l12   -0.014337   0.018756  -0.764   0.4736
## gdp_growth_subset.l12 -0.408396   0.225203  -1.813   0.1197
## const                  0.043626   0.021638   2.016   0.0904 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.00436 on 6 degrees of freedom
## Multiple R-Squared: 0.7602,  Adjusted R-squared: -0.1992
## F-statistic: 0.7924 on 24 and 6 DF,  p-value: 0.6886
##
##
##
## Covariance matrix of residuals:
##                   log_rev_quarter gdp_growth_subset
## log_rev_quarter         0.0016865         1.157e-04
## gdp_growth_subset       0.0001157         1.901e-05
##
## Correlation matrix of residuals:
##                   log_rev_quarter gdp_growth_subset
## log_rev_quarter            1.0000            0.6464
## gdp_growth_subset          0.6464            1.0000
```

```
# do a var on quarterly gdp growth and quarterly avg review
# score

recession_dummy_reviews_q = add_recession_dummy(df_rev_quarter$date)

reg_reviews = lm(df_rev_quarter$coredata.df_rev_m_quarter. ~
    recession_dummy_reviews_q)
summary(reg_reviews)
```

```
##
## Call:
## lm(formula = df_rev_quarter$coredata.df_rev_m_quarter. ~ recession_dummy_reviews_q)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -53417 -28547  -3776  24826  78800
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  53903       6553   8.225 2.72e-10 ***
## recession_dummy_reviews_q   -45589      16430  -2.775  0.00821 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 39860 on 42 degrees of freedom
## Multiple R-squared:  0.1549, Adjusted R-squared:  0.1348
## F-statistic: 7.699 on 1 and 42 DF,  p-value: 0.008211
```

```
# first reg is misleading because trend



# look at growth rates

rec_dummy_rev_growth_q = recession_dummy_reviews_q[2:length(recession_dummy_reviews_q)]

reg_log_reviews = lm(log_rev_quarter ~ rec_dummy_rev_growth_q)
summary(reg_log_reviews)
```

```
##
## Call:
## lm(formula = log_rev_quarter ~ rec_dummy_rev_growth_q)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44588 -0.13871 -0.02718  0.09270  0.83847
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.10999    0.03739   2.941  0.00535 **
## rec_dummy_rev_growth_q  0.05857    0.09268   0.632  0.53093
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2244 on 41 degrees of freedom
## Multiple R-squared:  0.009647,   Adjusted R-squared:  -0.01451
## F-statistic: 0.3994 on 1 and 41 DF,  p-value: 0.5309
```

```
# use lm
log_reviews_lm = lm(log_rev_quarter ~ rec_dummy_rev_growth_q)
summary(log_reviews_lm)
```

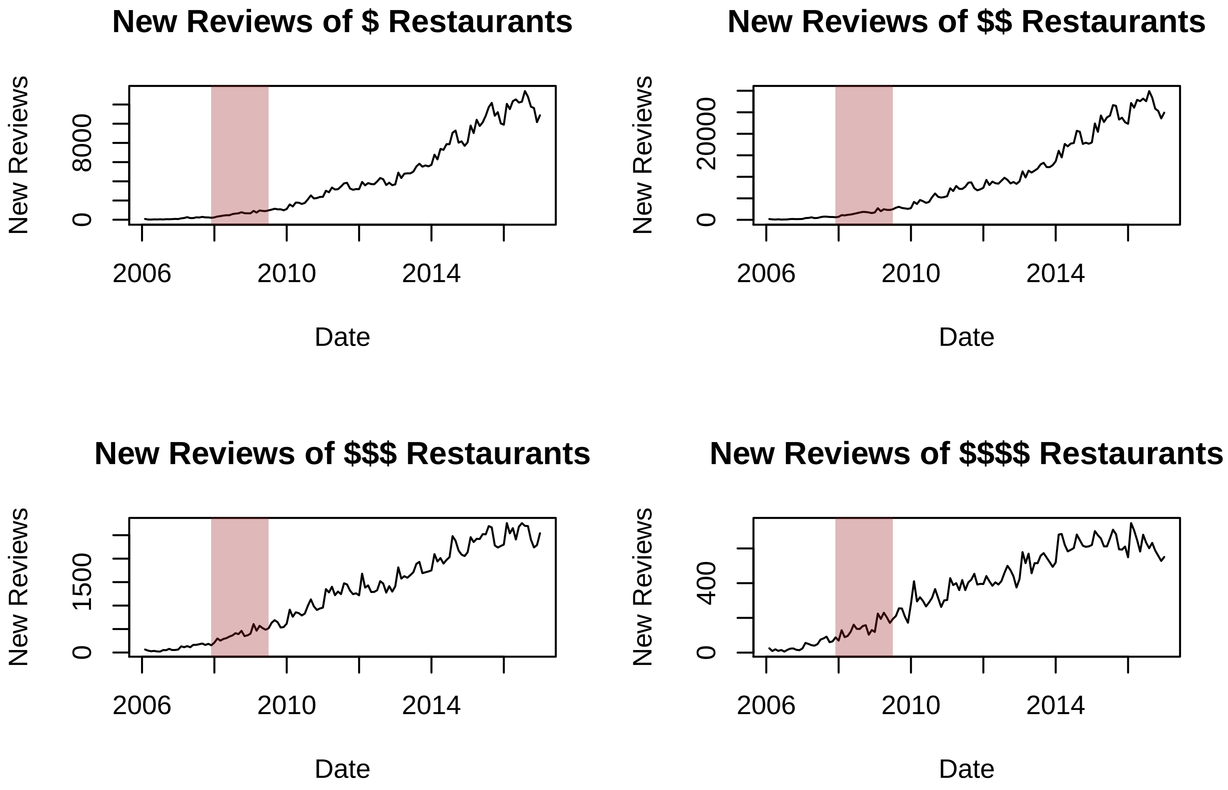
```
##
## Call:
## lm(formula = log_rev_quarter ~ rec_dummy_rev_growth_q)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44588 -0.13871 -0.02718  0.09270  0.83847
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.10999    0.03739   2.941  0.00535 **
## rec_dummy_rev_growth_q  0.05857    0.09268   0.632  0.53093
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2244 on 41 degrees of freedom
## Multiple R-squared:  0.009647,   Adjusted R-squared:  -0.01451
## F-statistic: 0.3994 on 1 and 41 DF,  p-value: 0.5309
```

examine social dynamics during recessions through dollar signs as well as review texts

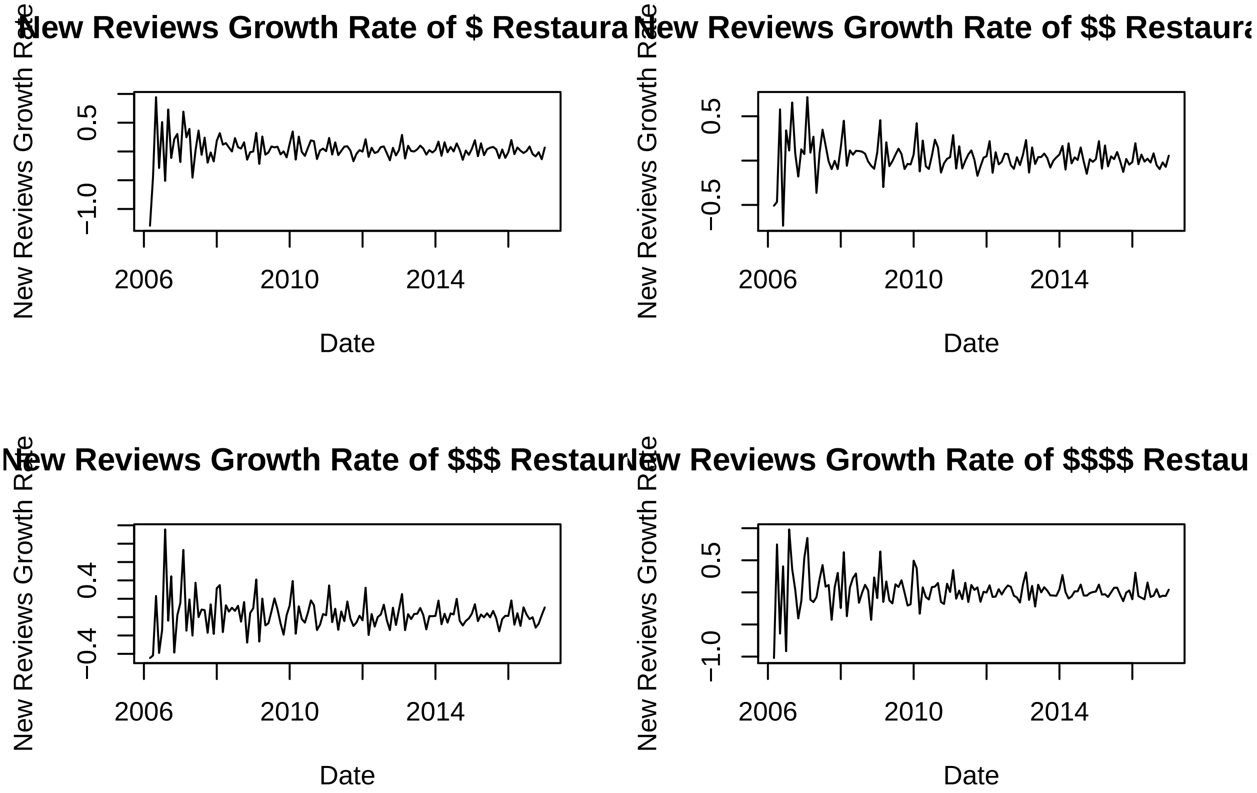
do people eat at cheaper places? do people care more about overpriced food?

```
dollars_1_xts = xts(dollars_gbd_1$`count(date)`, as.Date(dollars_gbd_1$date,
    "%Y-%m-%d"))
df_d_1 = apply.monthly(dollars_1_xts, sum)
df_dollars_1 = data.frame(date = index(df_d_1), coredata(df_d_1))
# df_dollars_1

dollars_2_xts = xts(dollars_gbd_2$`count(date)`, as.Date(dollars_gbd_2$date,
    "%Y-%m-%d"))
df_d_2 = apply.monthly(dollars_2_xts, sum)
df_dollars_2 = data.frame(date = index(df_d_2), coredata(df_d_2))
# df_dollars_2

dollars_3_xts = xts(dollars_gbd_3$`count(date)`, as.Date(dollars_gbd_3$date,
    "%Y-%m-%d"))
df_d_3 = apply.monthly(dollars_3_xts, sum)
df_dollars_3 = data.frame(date = index(df_d_3), coredata(df_d_3))
# df_dollars_3

dollars_4_xts = xts(dollars_gbd_4$`count(date)`, as.Date(dollars_gbd_4$date,
    "%Y-%m-%d"))
df_d_4 = apply.monthly(dollars_4_xts, sum)
df_dollars_4 = data.frame(date = index(df_d_4), coredata(df_d_4))
# df_dollars_4


recession_dummy_dollars_m = add_recession_dummy(df_dollars_1$date)
# par(mfrow=c(2,2))
# plot(df_dollars_1,type='l',xlab='Date',ylab='New Reviews',
# main='New Reviews of $ Restaurants')
# plot(df_dollars_2,type='l',xlab='Date',ylab='New Reviews',
# main='New Reviews of $$ Restaurants')
# plot(df_dollars_3,type='l',xlab='Date',ylab='New Reviews',
# main='New Reviews of $$$ Restaurants')
# plot(df_dollars_4,type='l',xlab='Date',ylab='New Reviews',
# main='New Reviews of $$$$ Restaurants') highly seasonal
```

```
df_dollars_1_dlog = as.data.frame(diff(log(df_dollars_1$coredata.df_d_1.)))
df_dollars_2_dlog = as.data.frame(diff(log(df_dollars_2$coredata.df_d_2.)))
df_dollars_3_dlog = as.data.frame(diff(log(df_dollars_3$coredata.df_d_3.)))
df_dollars_4_dlog = as.data.frame(diff(log(df_dollars_4$coredata.df_d_4.)))



# ts of dlogs
ts_dollar_1 = ts(df_dollars_1$coredata.df_d_1., start = c(2006,
    2), freq = 12)
ts_dollar_2 = ts(df_dollars_2$coredata.df_d_2., start = c(2006,
    2), freq = 12)
ts_dollar_3 = ts(df_dollars_3$coredata.df_d_3., start = c(2006,
    2), freq = 12)
ts_dollar_4 = ts(df_dollars_4$coredata.df_d_4., start = c(2006,
    2), freq = 12)


par(mfrow = c(2, 2))
plot(ts_dollar_1, type = "l", xlab = "Date", ylab = "New Reviews",
    main = "New Reviews of $ Restaurants")
rect(2007.9166667, -1000, 2009.5, 20000, col = rgb(red = 150/255,
    green = 25/255, blue = 25/255, alpha = 0.3), border = NA)
plot(ts_dollar_2, type = "l", xlab = "Date", ylab = "New Reviews",
    main = "New Reviews of $$ Restaurants")
rect(2007.9166667, -3000, 2009.5, 40000, col = rgb(red = 150/255,
    green = 25/255, blue = 25/255, alpha = 0.3), border = NA)
plot(ts_dollar_3, type = "l", xlab = "Date", ylab = "New Reviews",
    main = "New Reviews of $$$ Restaurants")
rect(2007.9166667, -1000, 2009.5, 7000, col = rgb(red = 150/255,
    green = 25/255, blue = 25/255, alpha = 0.3), border = NA)
plot(ts_dollar_4, type = "l", xlab = "Date", ylab = "New Reviews",
    main = "New Reviews of $$$$ Restaurants")
rect(2007.9166667, -1000, 2009.5, 2000, col = rgb(red = 150/255,
    green = 25/255, blue = 25/255, alpha = 0.3), border = NA)
```



```r
dollars_1_dlog = diff(log(ts_dollar_1))
dollars_2_dlog = diff(log(ts_dollar_2))
dollars_3_dlog = diff(log(ts_dollar_3))
dollars_4_dlog = diff(log(ts_dollar_4))

par(mfrow = c(2, 2))
plot(dollars_1_dlog, xlab = "Date", ylab = "New Reviews Growth Rate",
    main = "New Reviews Growth Rate of $ Restaurants")
plot(dollars_2_dlog, xlab = "Date", ylab = "New Reviews Growth Rate",
    main = "New Reviews Growth Rate of $$ Restaurants")
plot(dollars_3_dlog, xlab = "Date", ylab = "New Reviews Growth Rate",
    main = "New Reviews Growth Rate of $$$ Restaurants")
plot(dollars_4_dlog, xlab = "Date", ylab = "New Reviews Growth Rate",
    main = "New Reviews Growth Rate of $$$$ Restaurants")
```



```r
# we can see highly seasonal

# lm_dollars_recession_dummy =
# recession_dummy_dollars_m[2:length(recession_dummy_dollars_m)]
```

seasonally adjust the data

```r
# seasonally adjust data and remove trend
```

```r
par(mfrow = c(2, 2))

# levels
tslm_d1 = tslm(ts_dollar_1 ~ trend + season)
# summary(tslm_d1)
tslm_d1_resid = resid(tslm_d1)
plot(tslm_d1_resid, xlab = "Date", ylab = "New Reviews", main = "Adjusted Reviews, $")
rect(2007.9166667, -3000, 2009.5, 3000, col = rgb(red = 150/255,
    green = 25/255, blue = 25/255, alpha = 0.3), border = NA)
lm_d1_adj = lm(tslm_d1_resid ~ recession_dummy_dollars_m)
summary(lm_d1_adj)
```

```
##
## Call:
## lm(formula = tslm_d1_resid ~ recession_dummy_dollars_m)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1932.8 -1098.0  -162.2   975.3  2613.1
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  33.99     120.76   0.281    0.779
## recession_dummy_dollars_m  -236.11     318.30  -0.742    0.460
##
## Residual standard error: 1284 on 130 degrees of freedom
## Multiple R-squared:  0.004215,   Adjusted R-squared:  -0.003445
## F-statistic: 0.5502 on 1 and 130 DF,  p-value: 0.4596
```

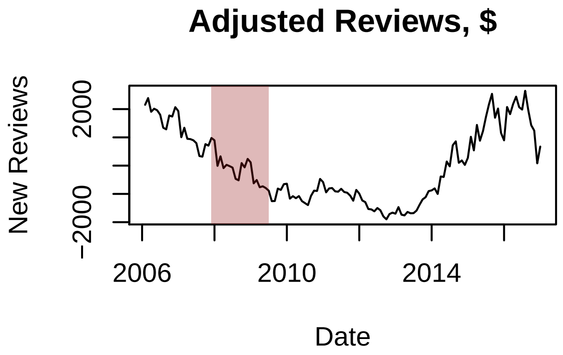
```r
tslm_d2 = tslm(ts_dollar_2 ~ trend + season)
# summary(tslm_d1)
tslm_d2_resid = resid(tslm_d2)
plot(tslm_d2_resid, xlab = "Date", ylab = "New Reviews", main = "Adjusted Reviews, $$")
rect(2007.9166667, -7000, 2009.5, 7000, col = rgb(red = 150/255,
    green = 25/255, blue = 25/255, alpha = 0.3), border = NA)
lm_d2_adj = lm(tslm_d2_resid ~ recession_dummy_dollars_m)
summary(lm_d2_adj)
```

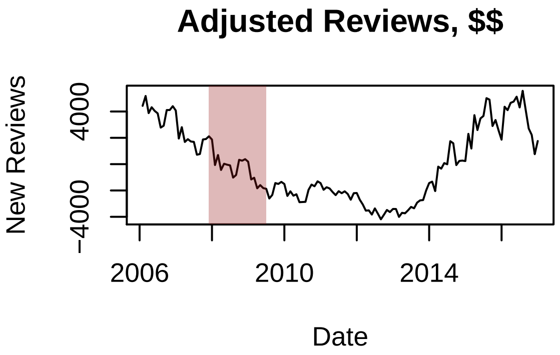
```
##
## Call:
## lm(formula = tslm_d2_resid ~ recession_dummy_dollars_m)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4262.1 -2293.1  -201.7  2170.2  5499.9
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  71.32     259.47   0.275    0.784
## recession_dummy_dollars_m  -495.52     683.92  -0.725    0.470
##
## Residual standard error: 2758 on 130 degrees of freedom
## Multiple R-squared:  0.004022,   Adjusted R-squared:  -0.00364
## F-statistic: 0.5249 on 1 and 130 DF,  p-value: 0.47
```

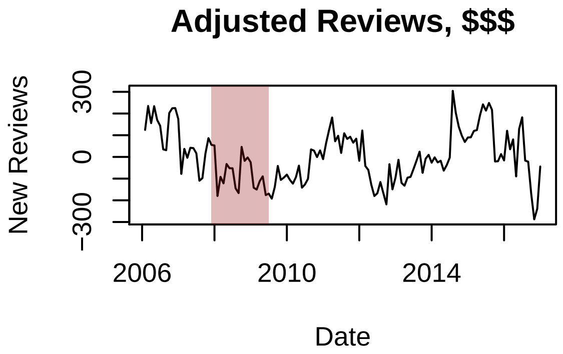
```r
tslm_d3 = tslm(ts_dollar_3 ~ trend + season)
# summary(tslm_d1)
tslm_d3_resid = resid(tslm_d3)
plot(tslm_d3_resid, xlab = "Date", ylab = "New Reviews", main = "Adjusted Reviews, $$$")
rect(2007.9166667, -500, 2009.5, 500, col = rgb(red = 150/255,
    green = 25/255, blue = 25/255, alpha = 0.3), border = NA)
lm_d3_adj = lm(tslm_d3_resid ~ recession_dummy_dollars_m)
summary(lm_d3_adj)
```

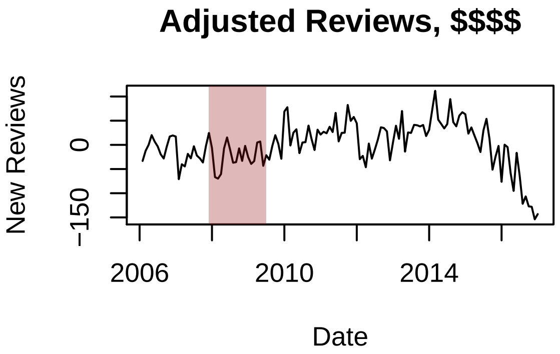
```
##
## Call:
## lm(formula = tslm_d3_resid ~ recession_dummy_dollars_m)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -301.93  -84.56  -10.65   76.69  290.26
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  14.40      10.99   1.310  0.19253
## recession_dummy_dollars_m  -100.01      28.97  -3.453  0.00075 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 116.8 on 130 degrees of freedom
## Multiple R-squared:  0.084,  Adjusted R-squared:  0.07695
## F-statistic: 11.92 on 1 and 130 DF,  p-value: 0.0007496
```

```r
tslm_d4 = tslm(ts_dollar_4 ~ trend + season)
# summary(tslm_d1)
tslm_d4_resid = resid(tslm_d4)
plot(tslm_d4_resid, xlab = "Date", ylab = "New Reviews", main = "Adjusted Reviews, $$$$")
rect(2007.9166667, -500, 2009.5, 500, col = rgb(red = 150/255,
    green = 25/255, blue = 25/255, alpha = 0.3), border = NA)
```









```r
lm_d4_adj = lm(tslm_d4_resid ~ recession_dummy_dollars_m)
summary(lm_d4_adj)
```

```
##
## Call:
## lm(formula = tslm_d4_resid ~ recession_dummy_dollars_m)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -158.13  -21.63    5.18   29.74  107.62
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  4.168      4.353   0.958   0.3400
## recession_dummy_dollars_m  -28.960     11.472  -2.524   0.0128 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 46.27 on 130 degrees of freedom
## Multiple R-squared:  0.04673,    Adjusted R-squared:  0.03939
## F-statistic: 6.372 on 1 and 130 DF,  p-value: 0.0128
```

introduce num of reviews into model

control for number of new reviews, to make sure that the recession dummy doesn't accidentally capture an effect from new review numbers, in other words if the number of reviews dropped it would make sense that the number of reviews for a certain amount of dollar signs drops too

```
# levels
lm_d1_adj_rev = lm(tslm_d1_resid ~ recession_dummy_dollars_m +
    df_rev_count$coredata.df_rev_m.)
summary(lm_d1_adj_rev)
```

```
##
## Call:
## lm(formula = tslm_d1_resid ~ recession_dummy_dollars_m + df_rev_count$coredata.df_rev_m.)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1829.6  -917.4  -307.4  1123.4  2837.2
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -4.521e+02  1.852e+02  -2.441 0.016016 *
## recession_dummy_dollars_m        1.689e+02  3.291e+02   0.513 0.608659
## df_rev_count$coredata.df_rev_m.  2.751e-02  8.162e-03   3.370 0.000991 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1235 on 129 degrees of freedom
## Multiple R-squared:  0.0848, Adjusted R-squared:  0.07061
## F-statistic: 5.976 on 2 and 129 DF,  p-value: 0.003295
```

```
lm_d2_adj_rev = lm(tslm_d2_resid ~ recession_dummy_dollars_m +
    df_rev_count$coredata.df_rev_m.)
summary(lm_d2_adj_rev)
```

```
##
## Call:
## lm(formula = tslm_d2_resid ~ recession_dummy_dollars_m + df_rev_count$coredata.df_rev_m.)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4138.1 -1943.8  -620.9  2367.0  6147.7
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -977.68395  397.81376  -2.458 0.015313 *
## recession_dummy_dollars_m        378.55976  706.74989   0.536 0.593133
## df_rev_count$coredata.df_rev_m.    0.05937    0.01753   3.387 0.000938 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2653 on 129 degrees of freedom
## Multiple R-squared:  0.08534,    Adjusted R-squared:  0.07116
## F-statistic: 6.018 on 2 and 129 DF,  p-value: 0.003171
```

```
lm_d3_adj_rev = lm(tslm_d3_resid ~ recession_dummy_dollars_m +
    df_rev_count$coredata.df_rev_m.)
summary(lm_d3_adj_rev)
```

```
##
## Call:
## lm(formula = tslm_d3_resid ~ recession_dummy_dollars_m + df_rev_count$coredata.df_rev_m.)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -315.824  -83.622   -8.919   77.962  280.690
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       3.403e+00  1.754e+01   0.194  0.84646
## recession_dummy_dollars_m        -9.085e+01  3.116e+01  -2.916  0.00418 **
## df_rev_count$coredata.df_rev_m.  6.221e-04  7.729e-04   0.805  0.42231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 117 on 129 degrees of freedom
## Multiple R-squared:  0.08857,    Adjusted R-squared:  0.07444
## F-statistic: 6.268 on 2 and 129 DF,  p-value: 0.002523
```

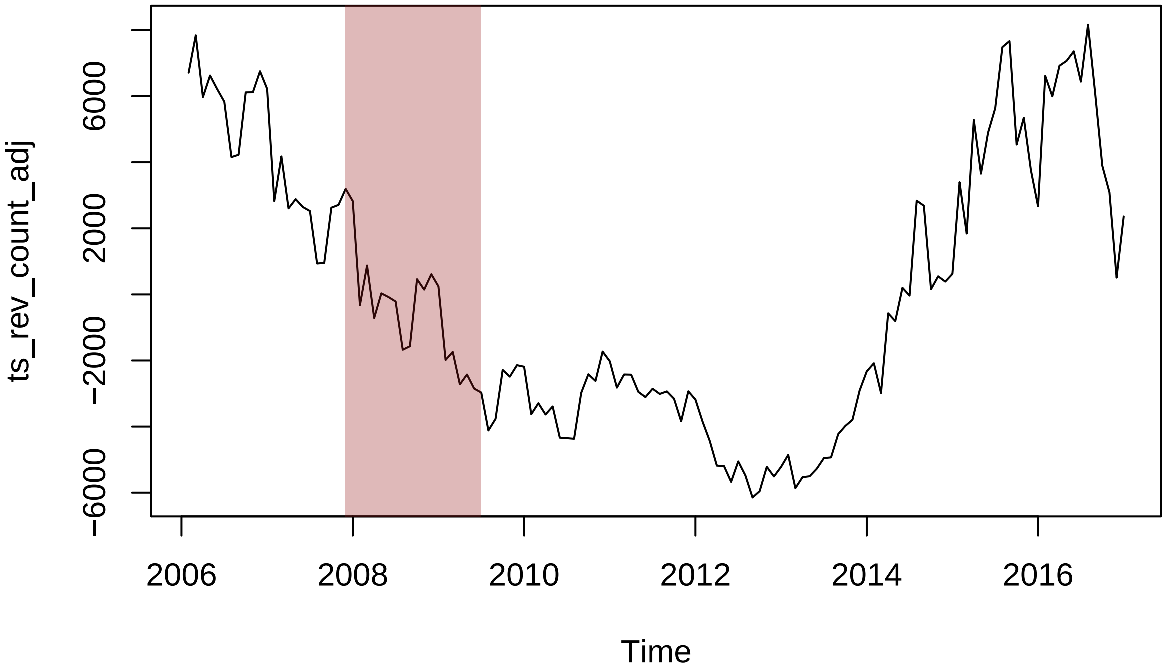
```
lm_d4_adj_rev = lm(tslm_d4_resid ~ recession_dummy_dollars_m +
    df_rev_count$coredata.df_rev_m.)
summary(lm_d4_adj_rev)
```

```
##
## Call:
## lm(formula = tslm_d4_resid ~ recession_dummy_dollars_m + df_rev_count$coredata.df_rev_m.)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -145.400  -28.213    4.314   30.616  111.525
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       1.591e+01  6.836e+00   2.327  0.02152 *
## recession_dummy_dollars_m        -3.874e+01  1.214e+01  -3.190  0.00179 **
## df_rev_count$coredata.df_rev_m. -6.644e-04  3.012e-04  -2.206  0.02919 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 45.6 on 129 degrees of freedom
## Multiple R-squared:  0.08137,    Adjusted R-squared:  0.06712
## F-statistic: 5.713 on 2 and 129 DF,  p-value: 0.004195
```

```
# detrend review counts

ts_rev_count = ts(df_rev_count$coredata.df_rev_m., start = c(2006,
    2), freq = 12)
ts_rev_count_tslm = tslm(ts_rev_count ~ trend + season)
summary(ts_rev_count_tslm)
```

```
##
## Call:
## tslm(formula = ts_rev_count ~ trend + season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6146.8 -3161.7  -642.9  3115.8  8167.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10135.277   1480.253  -6.847 3.52e-10 ***
## trend          356.093      9.876  36.056  < 2e-16 ***
## season2       3427.756   1839.239   1.864   0.0648 .
## season3       1767.844   1838.682   0.961   0.3383
## season4       3223.751   1838.178   1.754   0.0820 .
## season5       2287.839   1837.727   1.245   0.2156
## season6       2263.018   1837.329   1.232   0.2205
## season7       2320.834   1836.984   1.263   0.2089
## season8       3676.922   1836.692   2.002   0.0476 *
## season9       3374.556   1836.453   1.838   0.0686 .
## season10      1166.008   1836.267   0.635   0.5267
## season11       750.823   1836.134   0.409   0.6833
## season12      -197.361   1836.054  -0.107   0.9146
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4306 on 119 degrees of freedom
## Multiple R-squared:  0.9165, Adjusted R-squared:  0.9081
## F-statistic: 108.9 on 12 and 119 DF,  p-value: < 2.2e-16
```

```r
ts_rev_count_adj = resid(ts_rev_count_tslm, ylab = "New Reviews",
    main = "Adjusted Reviews, Total")
plot(ts_rev_count_adj)
rect(2007.9166667, -9000, 2009.5, 9000, col = rgb(red = 150/255,
    green = 25/255, blue = 25/255, alpha = 0.3), border = NA)
```



```
lm_d1_adj_rev_adj = lm(tslm_d1_resid ~ recession_dummy_dollars_m +
    ts_rev_count_adj)
summary(lm_d1_adj_rev_adj)
```

```
##
## Call:
## lm(formula = tslm_d1_resid ~ recession_dummy_dollars_m + ts_rev_count_adj)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -303.05  -50.07   -3.11   56.55  359.65
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -4.796724   9.627037  -0.498    0.619
## recession_dummy_dollars_m 33.324607  25.434935   1.310    0.192
## ts_rev_count_adj           0.311485   0.002184 142.630   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 102.3 on 129 degrees of freedom
## Multiple R-squared:  0.9937, Adjusted R-squared:  0.9936
## F-statistic: 1.021e+04 on 2 and 129 DF,  p-value: < 2.2e-16
```

```
lm_d2_adj_rev_adj = lm(tslm_d2_resid ~ recession_dummy_dollars_m +
    ts_rev_count_adj)
summary(lm_d2_adj_rev_adj)
```

```
##
## Call:
## lm(formula = tslm_d2_resid ~ recession_dummy_dollars_m + ts_rev_count_adj)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -257.51  -74.84   -1.27   80.55  428.91
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -12.196577  10.777970  -1.132  0.25989
## recession_dummy_dollars_m  84.734117  28.475734   2.976  0.00349 **
## ts_rev_count_adj            0.670808   0.002445 274.365  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 114.5 on 129 degrees of freedom
## Multiple R-squared:  0.9983, Adjusted R-squared:  0.9983
## F-statistic: 3.779e+04 on 2 and 129 DF,  p-value: < 2.2e-16
```

```
lm_d3_adj_rev_adj = lm(tslm_d3_resid ~ recession_dummy_dollars_m +
    ts_rev_count_adj)
summary(lm_d3_adj_rev_adj)
```

```
##
## Call:
## lm(formula = tslm_d3_resid ~ recession_dummy_dollars_m + ts_rev_count_adj)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -351.65  -55.54   -0.17   58.72  244.76
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                12.307561   8.912974   1.381 0.169710
## recession_dummy_dollars_m -85.505163  23.548357  -3.631 0.000406 ***
## ts_rev_count_adj            0.016770   0.002022   8.294 1.25e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 94.71 on 129 degrees of freedom
## Multiple R-squared:  0.4026, Adjusted R-squared:  0.3933
## F-statistic: 43.47 on 2 and 129 DF,  p-value: 3.712e-15
```

```
lm_d4_adj_rev_adj = lm(tslm_d4_resid ~ recession_dummy_dollars_m +
    ts_rev_count_adj)
summary(lm_d4_adj_rev_adj)
```

```
##
## Call:
## lm(formula = tslm_d4_resid ~ recession_dummy_dollars_m + ts_rev_count_adj)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -156.602  -22.164    6.943   27.480  101.189
```

```
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 4.662e+00  4.092e+00   1.139  0.25668
## recession_dummy_dollars_m -3.239e+01  1.081e+01  -2.996  0.00328 **
## ts_rev_count_adj          -3.962e-03  9.282e-04  -4.269 3.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.48 on 129 degrees of freedom
## Multiple R-squared:  0.1647, Adjusted R-squared:  0.1518
## F-statistic: 12.72 on 2 and 129 DF,  p-value: 9.082e-06
```

```
vif(lm_d1_adj_rev_adj)
```

```
## recession_dummy_dollars_m          ts_rev_count_adj
##                  1.005547                  1.005547
```

```
vif(lm_d2_adj_rev_adj)
```

```
## recession_dummy_dollars_m          ts_rev_count_adj
##                  1.005547                  1.005547
```

```
vif(lm_d3_adj_rev_adj)
```

```
## recession_dummy_dollars_m          ts_rev_count_adj
##                  1.005547                  1.005547
```

```
vif(lm_d4_adj_rev_adj)
```

```
## recession_dummy_dollars_m          ts_rev_count_adj
##                  1.005547                  1.005547
```

re examine stars, but subset by dollar signs

```
par(mfrow = c(2, 2))

dollars_1_star_xts = xts(dollars_obd_1_star$stars, as.Date(dollars_obd_1_star$date,
    "%Y-%m-%d"))
df_d_1_star = apply.monthly(dollars_1_star_xts, sum)
df_dollars_1_star = data.frame(date = index(df_d_1_star), coredata(df_d_1_star))
df_dollars_1_star$avg = df_dollars_1_star$coredata.df_d_1_star./df_dollars_1$coredata.df_d_1.
plot(df_dollars_1_star$date, df_dollars_1_star$avg, xlab = "Date",
    ylab = "Average Star Rating", main = "Average Rating for $ Restaurants")
d1_star_lm = lm(df_dollars_1_star$avg ~ recession_dummy_dollars_m)
summary(d1_star_lm)
```

```
##
## Call:
## lm(formula = df_dollars_1_star$avg ~ recession_dummy_dollars_m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26498 -0.03895 -0.01530  0.01052  0.53502
##
```

```
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                3.750694   0.009036 415.069   <2e-16 ***
## recession_dummy_dollars_m -0.044273   0.023818  -1.859   0.0653 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09606 on 130 degrees of freedom
## Multiple R-squared:  0.02589,    Adjusted R-squared:  0.0184 
## F-statistic: 3.455 on 1 and 130 DF,  p-value: 0.06532
```

```
dollars_2_star_xts = xts(dollars_obd_2_star$stars, as.Date(dollars_obd_2_star$date,
    "%Y-%m-%d"))
df_d_2_star = apply.monthly(dollars_2_star_xts, sum)
df_dollars_2_star = data.frame(date = index(df_d_2_star), coredata(df_d_2_star))
df_dollars_2_star$avg = df_dollars_2_star$coredata.df_d_2_star./df_dollars_2$coredata.df_d_2.
plot(df_dollars_2_star$date, df_dollars_2_star$avg, xlab = "Date",
    ylab = "Average Star Rating", main = "Average Rating for $$ Restaurants")
d2_star_lm = lm(df_dollars_2_star$avg ~ recession_dummy_dollars_m)
summary(d2_star_lm)
```

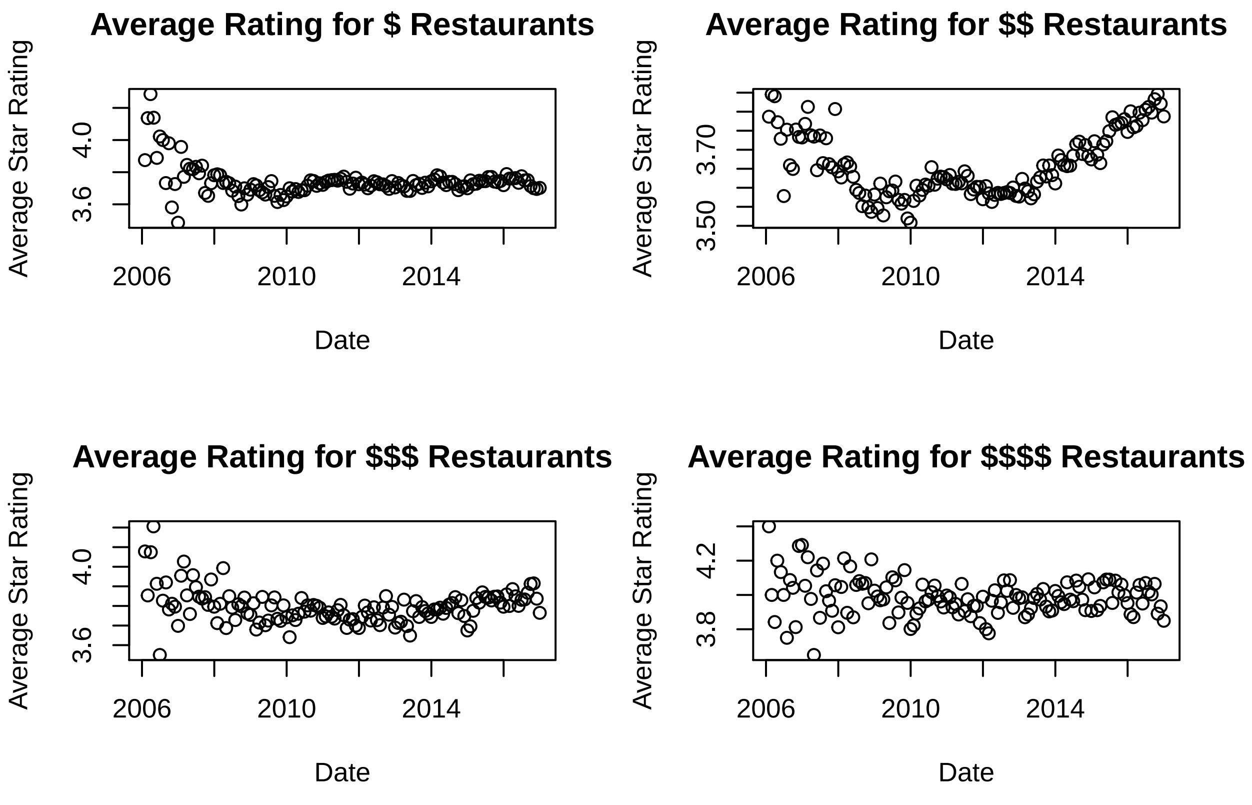
```
##
## Call:
## lm(formula = df_dollars_2_star$avg ~ recession_dummy_dollars_m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.16317 -0.06408 -0.01288  0.06248  0.17477 
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                3.671420   0.007485 490.516  < 2e-16 ***
## recession_dummy_dollars_m -0.076331   0.019728  -3.869 0.000172 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07956 on 130 degrees of freedom
## Multiple R-squared:  0.1033, Adjusted R-squared:  0.09637 
## F-statistic: 14.97 on 1 and 130 DF,  p-value: 0.000172
```

```
dollars_3_star_xts = xts(dollars_obd_3_star$stars, as.Date(dollars_obd_3_star$date,
    "%Y-%m-%d"))
df_d_3_star = apply.monthly(dollars_3_star_xts, sum)
df_dollars_3_star = data.frame(date = index(df_d_3_star), coredata(df_d_3_star))
df_dollars_3_star$avg = df_dollars_3_star$coredata.df_d_3_star./df_dollars_3$coredata.df_d_3.
plot(df_dollars_3_star$date, df_dollars_3_star$avg, xlab = "Date",
    ylab = "Average Star Rating", main = "Average Rating for $$$ Restaurants")
d3_star_lm = lm(df_dollars_3_star$avg ~ recession_dummy_dollars_m)
summary(d3_star_lm)
```

```
##
## Call:
## lm(formula = df_dollars_3_star$avg ~ recession_dummy_dollars_m)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.25080 -0.05478 -0.00522  0.03954  0.40509
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.800796   0.008157 465.945   <2e-16 ***
## recession_dummy_dollars_m -0.020898   0.021501  -0.972    0.333
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08671 on 130 degrees of freedom
## Multiple R-squared:  0.007215,   Adjusted R-squared:  -0.000422
## F-statistic: 0.9447 on 1 and 130 DF,  p-value: 0.3329
```

```
dollars_4_star_xts = xts(dollars_obd_4_star$stars, as.Date(dollars_obd_4_star$date,
    "%Y-%m-%d"))
df_d_4_star = apply.monthly(dollars_4_star_xts, sum)
df_dollars_4_star = data.frame(date = index(df_d_4_star), coredata(df_d_4_star))
df_dollars_4_star$avg = df_dollars_4_star$coredata.df_d_4_star./df_dollars_4$coredata.df_d_4.
plot(df_dollars_4_star$date, df_dollars_4_star$avg, xlab = "Date",
    ylab = "Average Star Rating", main = "Average Rating for $$$$ Restaurants")
```



```
d4_star_lm = lm(df_dollars_4_star$avg ~ recession_dummy_dollars_m)
summary(d4_star_lm)
```

```
##
## Call:
```

```
## lm(formula = df_dollars_4_star$avg ~ recession_dummy_dollars_m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33494 -0.06462 -0.00311  0.06433  0.41506
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.98494    0.01040  383.33   <2e-16 ***
## recession_dummy_dollars_m  0.03506    0.02740    1.28    0.203
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1105 on 130 degrees of freedom
## Multiple R-squared:  0.01244,    Adjusted R-squared:  0.004841
## F-statistic: 1.637 on 1 and 130 DF,  p-value: 0.203
```

```
# looks like the stars dropping during a recession was only
# in 1 and 2 dollar signs restaurants
```

how bout use stars in reg for dolla dolla

```
lm_d1_adj_rev_adj_star = lm(tslm_d1_resid ~ recession_dummy_dollars_m +
    ts_rev_count_adj + df_dollars_1_star$avg)
summary(lm_d1_adj_rev_adj_star)
```

```
##
## Call:
## lm(formula = tslm_d1_resid ~ recession_dummy_dollars_m + ts_rev_count_adj +
##     df_dollars_1_star$avg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -298.43  -50.98    0.42   55.13  365.21
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -351.19798  380.16876  -0.924    0.357
## recession_dummy_dollars_m   36.69179   25.71834   1.427    0.156
## ts_rev_count_adj             0.31065    0.00237 131.083   <2e-16 ***
## df_dollars_1_star$avg       92.38432  101.35747   0.911    0.364
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 102.4 on 128 degrees of freedom
## Multiple R-squared:  0.9938, Adjusted R-squared:  0.9936
## F-statistic:  6801 on 3 and 128 DF,  p-value: < 2.2e-16
```

```
lm_d2_adj_rev_adj_star = lm(tslm_d2_resid ~ recession_dummy_dollars_m +
    ts_rev_count_adj + df_dollars_2_star$avg)
summary(lm_d2_adj_rev_adj_star)
```

```
##
```

```
## Call:
## lm(formula = tslm_d2_resid ~ recession_dummy_dollars_m + ts_rev_count_adj +
##     df_dollars_2_star$avg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -255.96  -76.18   -4.78   72.52  402.90
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -6.772e+02  8.070e+02  -0.839  0.40291
## recession_dummy_dollars_m  9.608e+01  3.166e+01   3.035  0.00292 **
## ts_rev_count_adj           6.679e-01  4.259e-03 156.841  < 2e-16 ***
## df_dollars_2_star$avg      1.812e+02  2.199e+02   0.824  0.41137
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 114.7 on 128 degrees of freedom
## Multiple R-squared:  0.9983, Adjusted R-squared:  0.9983
## F-statistic: 2.513e+04 on 3 and 128 DF,  p-value: < 2.2e-16
```

```
lm_d3_adj_rev_adj_star = lm(tslm_d3_resid ~ recession_dummy_dollars_m +
    ts_rev_count_adj + df_dollars_3_star$avg)
summary(lm_d3_adj_rev_adj_star)
```

```
##
## Call:
## lm(formula = tslm_d3_resid ~ recession_dummy_dollars_m + ts_rev_count_adj +
##     df_dollars_3_star$avg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -333.28  -58.64   -0.01   64.36  241.80
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                8.423e+02  4.110e+02   2.049 0.042474 *
## recession_dummy_dollars_m -8.814e+01  2.331e+01  -3.781 0.000238 ***
## ts_rev_count_adj           1.900e-02  2.283e-03   8.323 1.11e-13 ***
## df_dollars_3_star$avg     -2.184e+02  1.082e+02  -2.020 0.045488 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93.6 on 128 degrees of freedom
## Multiple R-squared:  0.421,  Adjusted R-squared:  0.4075
## F-statistic: 31.03 on 3 and 128 DF,  p-value: 3.839e-15
```

```
lm_d4_adj_rev_adj_star = lm(tslm_d4_resid ~ recession_dummy_dollars_m +
    ts_rev_count_adj + df_dollars_4_star$avg)
summary(lm_d4_adj_rev_adj_star)
```

```
##
## Call:
## lm(formula = tslm_d4_resid ~ recession_dummy_dollars_m + ts_rev_count_adj +
##     df_dollars_4_star$avg)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -155.685  -23.921    6.389   25.495   99.837
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -6.299e+01  1.430e+02  -0.440  0.66034
## recession_dummy_dollars_m -3.309e+01  1.094e+01  -3.023  0.00302 **
## ts_rev_count_adj          -4.083e-03  9.652e-04  -4.230 4.42e-05 ***
## df_dollars_4_star$avg      1.698e+01  3.588e+01   0.473  0.63684
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.61 on 128 degrees of freedom
## Multiple R-squared:  0.1662, Adjusted R-squared:  0.1466
## F-statistic: 8.503 on 3 and 128 DF,  p-value: 3.411e-05
```

```
vif(lm_d1_adj_rev_adj_star)
```

```
## recession_dummy_dollars_m          ts_rev_count_adj
##                  1.026731                  1.182581
##     df_dollars_1_star$avg
##                  1.207316
```

```
vif(lm_d2_adj_rev_adj_star)
```

```
## recession_dummy_dollars_m          ts_rev_count_adj
##                  1.240119                  3.043228
##     df_dollars_2_star$avg
##                  3.374949
```

```
vif(lm_d3_adj_rev_adj_star)
```

```
## recession_dummy_dollars_m          ts_rev_count_adj
##                  1.008711                  1.312308
##     df_dollars_3_star$avg
##                  1.314554
```

```
vif(lm_d4_adj_rev_adj_star)
```

```
## recession_dummy_dollars_m          ts_rev_count_adj
##                  1.024237                  1.080757
##     df_dollars_4_star$avg
##                  1.088332
```

```
# not really a difference, star ratings dynamics not really
# linked to review numbers, but both effected by rec in dif
# ways
```

comment out for now, cuz takes too long to process

analyze word usages

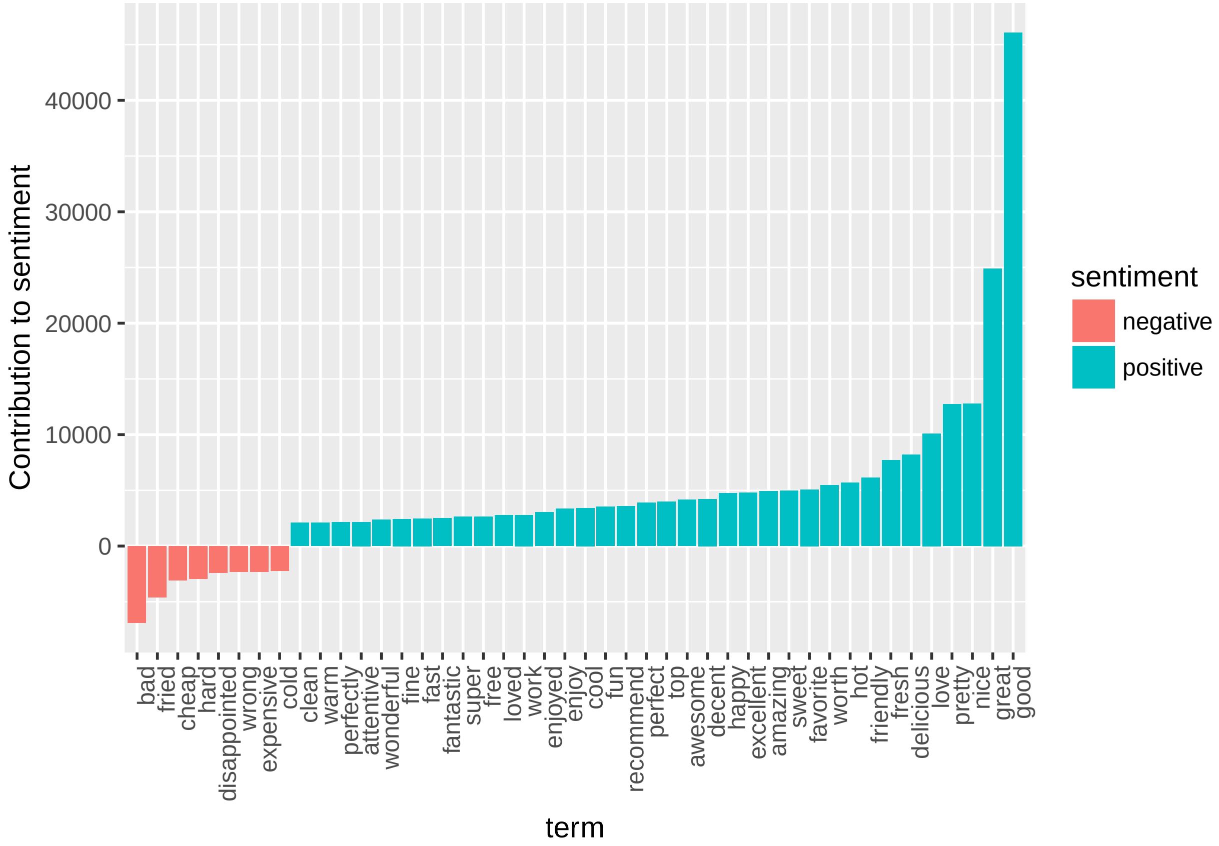
```
buildCorpus = function(data, stem) {
    corpus = Corpus(VectorSource(data))
    corpus = tm_map(corpus, content_transformer(tolower))
    corpus = tm_map(corpus, PlainTextDocument)
    corpus = tm_map(corpus, removePunctuation)
    corpus = tm_map(corpus, removeWords, stopWords)
    if (stem == 1)
        corpus = tm_map(corpus, stemDocument)
    return(corpus)
}

buildWordCloud = function(corpus, pal, val, name) {
    wordcloud(corpus, max.words = 75, random.order = FALSE, colors = brewer.pal(val,
        pal), main = name)
}
stopWords = removePunctuation(stopwords("SMART"))


restaurant_reviews_rec = with(restaurant_reviews, restaurant_reviews[(restaurant_reviews$date >=
    "2007-12" & restaurant_reviews$date <= "2009-06"), ])

restaurant_reviews_norec = with(restaurant_reviews, restaurant_reviews[(restaurant_reviews$date >
    "2009-06" & restaurant_reviews$date <= "2011-12"), ])

# create corupses
corpus_reviews_rec = buildCorpus(restaurant_reviews_rec$text,
    0)
dtm_rec = DocumentTermMatrix(corpus_reviews_rec)
tidy_rec = tidy(dtm_rec)
ap_sentiments <- tidy_rec %>% inner_join(get_sentiments("bing"),
    by = c(term = "word"))

ap_sentiments
```

```
## # A tibble: 436,171 × 4
##         document            term count sentiment
##            <chr>           <chr> <dbl>     <chr>
## 1  character(0)            died     1  negative
## 2  character(0)      enthusiasm     1  positive
## 3  character(0)       fantastic     2  positive
## 4  character(0)            good     1  positive
## 5  character(0)        horrible     1  negative
## 6  character(0)            love     1  positive
## 7  character(0) recommendations     1  positive
## 8  character(0)            good     1  positive
## 9  character(0)            nice     1  positive
## 10 character(0)            pure     1  positive
## # ... with 436,161 more rows
```

```
ap_sentiments %>% count(document, sentiment, wt = count) %>%
    ungroup() %>% spread(sentiment, n, fill = 0) %>% mutate(sentiment = positive -
    negative) %>% arrange(sentiment)
```

```
## # A tibble: 1 × 4
##       document negative positive sentiment
##          <chr>    <dbl>    <dbl>     <dbl>
## 1 character(0)    158714    346991     188277
```

```
ap_sentiments %>% count(sentiment, term, wt = count) %>% ungroup() %>%
    filter(n >= 2000) %>% mutate(n = ifelse(sentiment == "negative",
    -n, n)) %>% mutate(term = reorder(term, n)) %>% ggplot(aes(term,
    n, fill = sentiment)) + geom_bar(stat = "identity") + theme(axis.text.x = element_text(angle = 90,
    hjust = 1)) + ylab("Contribution to sentiment")
```



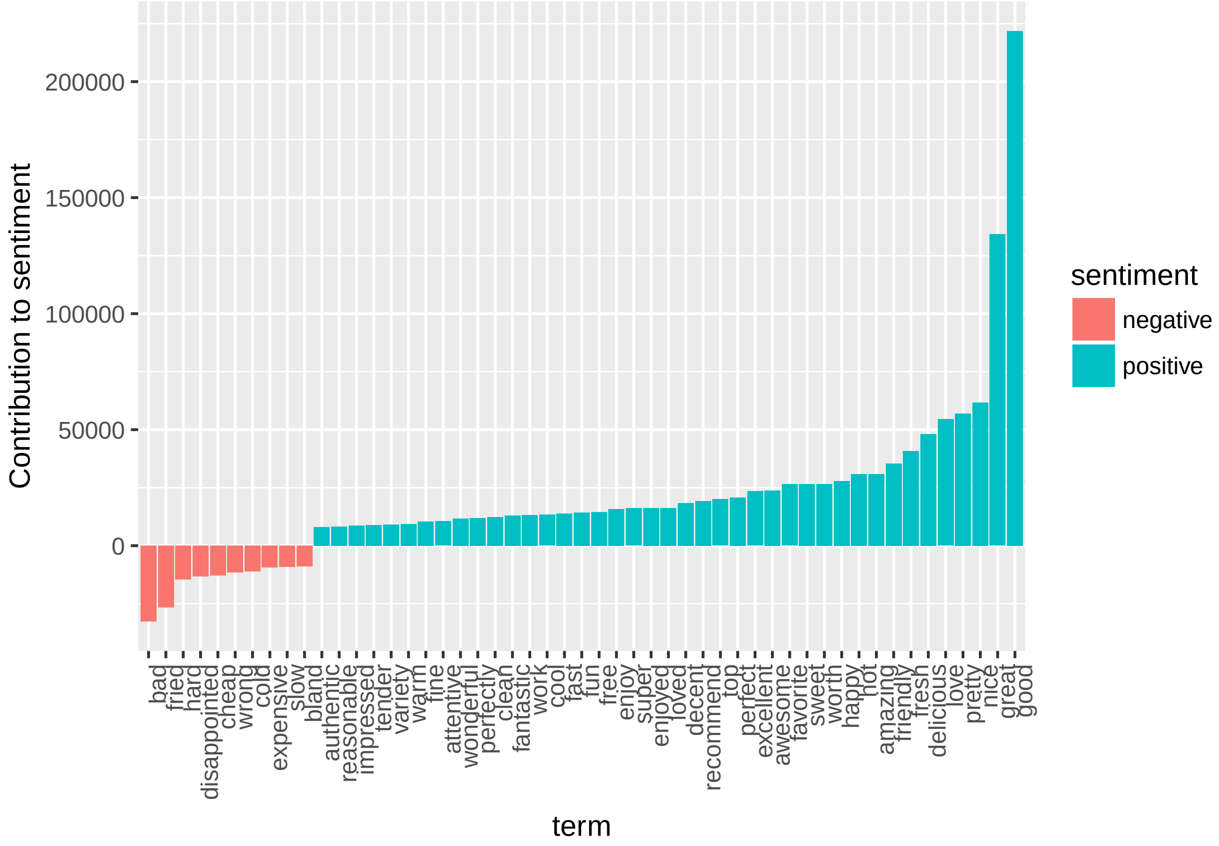
```
# negative: 158714 positive: 346991 percent negative: 31.4%
# cheap: 3rd highest negative word expensive: 7th



corpus_reviews_norec = buildCorpus(restaurant_reviews_norec$text,
    0)
dtm_norec = DocumentTermMatrix(corpus_reviews_norec)
tidy_norec = tidy(dtm_norec)
ap_sentiments <- tidy_norec %>% inner_join(get_sentiments("bing"),
    by = c(term = "word"))

ap_sentiments
```

```
## # A tibble: 2,121,102 × 4
##         document      term count sentiment
##            <chr>     <chr> <dbl>     <chr>
## 1  character(0) amazingly     1  positive
## 2  character(0)   awesome     1  positive
## 3  character(0)      fast     2  positive
## 4  character(0)   fucking     1  negative
## 5  character(0)     great     1  positive
## 6  character(0)      holy     1  positive
## 7  character(0)      nice     1  positive
## 8  character(0)      shit     1  negative
## 9  character(0)      weak     1  negative
## 10 character(0)      work     1  positive
## # ... with 2,121,092 more rows
```

```
ap_sentiments %>% count(document, sentiment, wt = count) %>%
    ungroup() %>% spread(sentiment, n, fill = 0) %>% mutate(sentiment = positive -
    negative) %>% arrange(sentiment)
```

```
## # A tibble: 1 × 4
##       document negative positive sentiment
##          <chr>    <dbl>    <dbl>     <dbl>
## 1 character(0)   733329  1731938    998609
```

```
ap_sentiments %>% count(sentiment, term, wt = count) %>% ungroup() %>%
    filter(n >= 7500) %>% mutate(n = ifelse(sentiment == "negative",
    -n, n)) %>% mutate(term = reorder(term, n)) %>% ggplot(aes(term,
    n, fill = sentiment)) + geom_bar(stat = "identity") + theme(axis.text.x = element_text(angle = 90,
    hjust = 1)) + ylab("Contribution to sentiment")
```



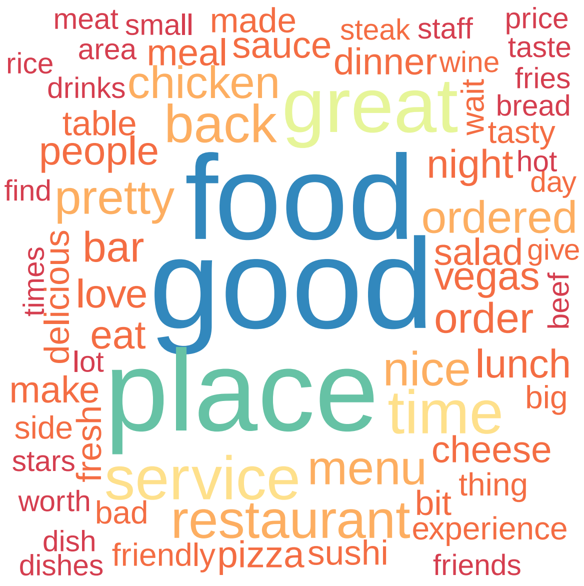
```
# negative: 733329 positive: 1731938 percent negative: 29.75%
# cheap: 5th expensive: 8th
par(mfrow = c(1, 2))
```

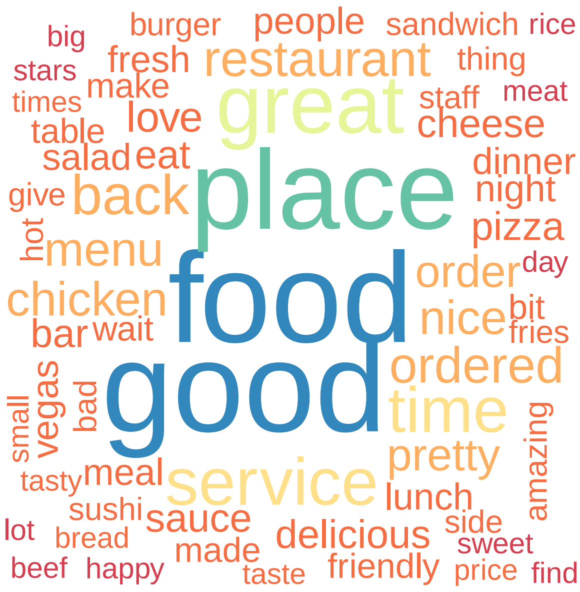
```r
buildWordCloud(corpus_reviews_rec, "Spectral", 8, "test")
```

```
## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors
## = brewer.pal(val, : atmosphere could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors
## = brewer.pal(val, : favorite could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : burger could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : places could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : buffet could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : sweet could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : long could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : amazing could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors
## = brewer.pal(val, : sandwich could not be fit on page. It will not be
## plotted.
```

```r
buildWordCloud(corpus_reviews_norec, "Spectral", 8, "test2")
```

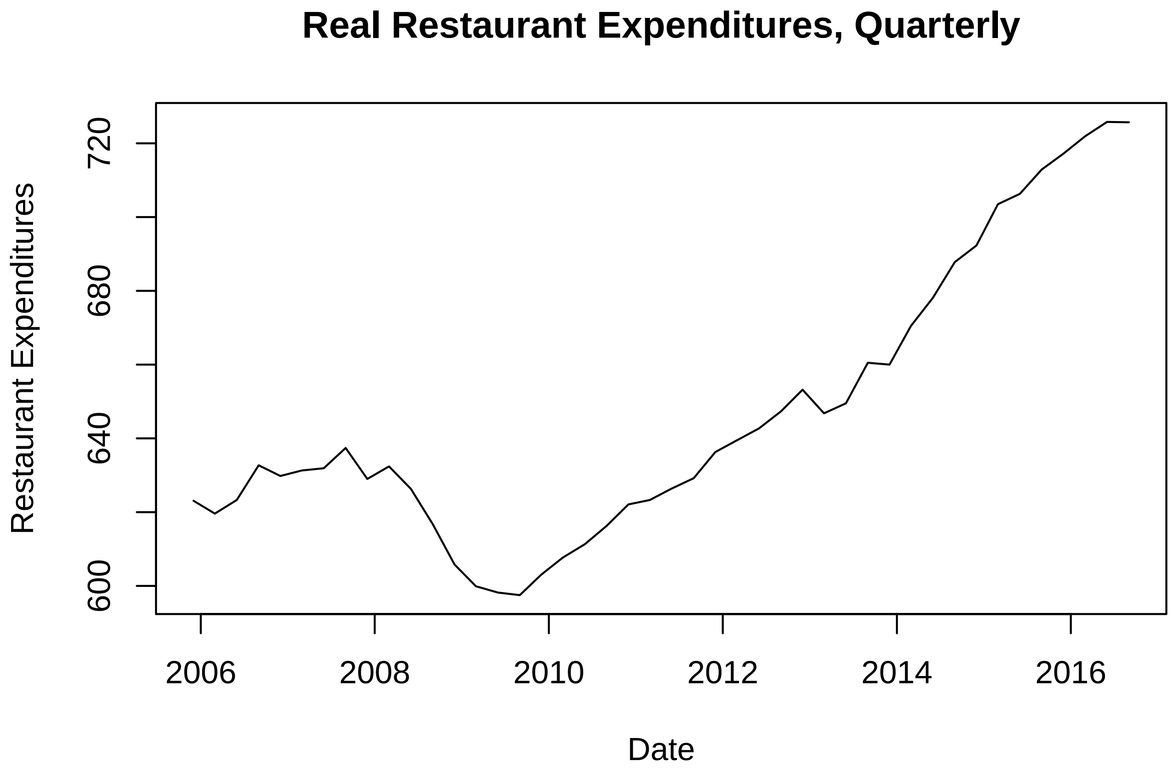
```
## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors
## = brewer.pal(val, : experience could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : drinks could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : dish could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : worth could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : fried could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors
## = brewer.pal(val, : favorite could not be fit on page. It will not be
## plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : steak could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : flavor could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : area could not be fit on page. It will not be plotted.
## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : minutes could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : drink could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : special could not be fit on page. It will not be plotted.

## Warning in wordcloud(corpus, max.words = 75, random.order = FALSE, colors =
## brewer.pal(val, : friends could not be fit on page. It will not be plotted.
```





connect with restaurants

```r
# add x axis with dates
rest_exp_dates = seq(as.Date("2005/12/01"), by = "quarter", length.out = 44)

plot(rest_exp_dates, restaurant_expenditures$real_exp, type = "l",
    xlab = "Date", ylab = "Restaurant Expenditures", main = "Real Restaurant Expenditures, Quarterly")
```



```
test_stationary(restaurant_expenditures$real_exp)
```

```
## Warning in kpss.test(t): p-value smaller than printed p-value
```

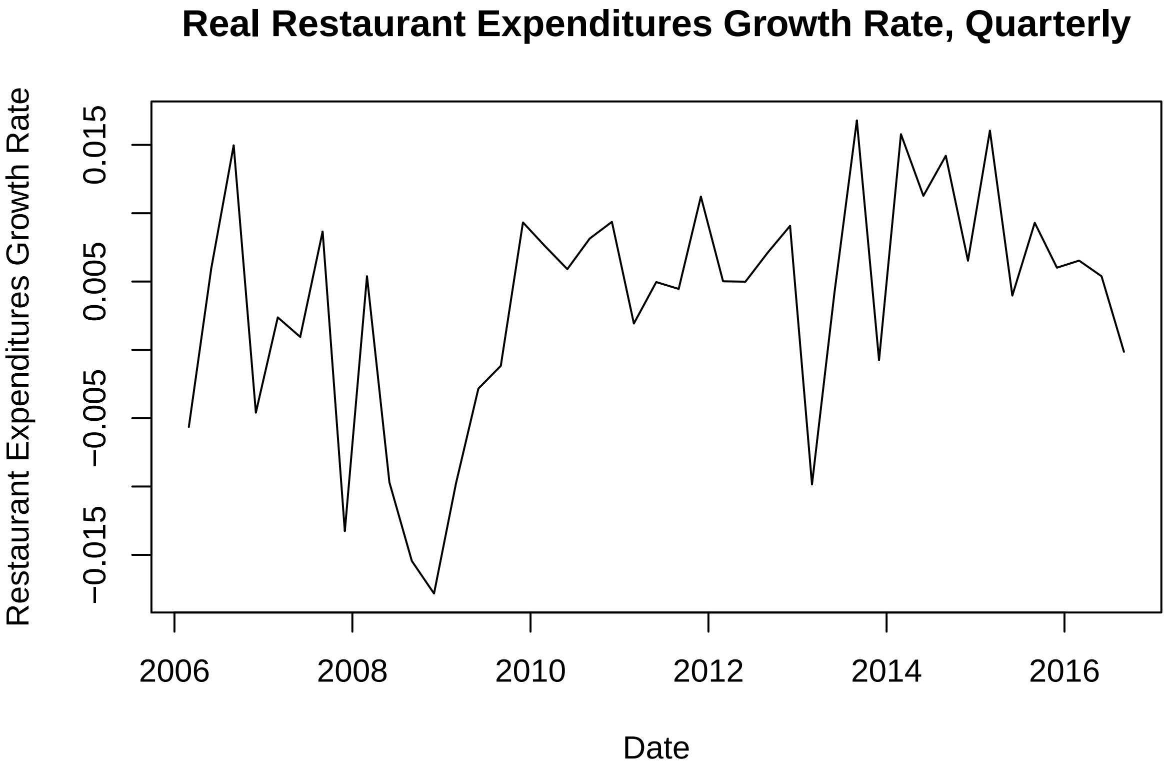
```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 1.7047, Truncation lag parameter = 1, p-value = 0.01
##
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -2.4338, Lag order = 3, p-value = 0.4021
## alternative hypothesis: stationary
```

```
rest_real_exp_diff_log = diff(log(restaurant_expenditures$real_exp))

test_stationary(rest_real_exp_diff_log)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 0.69776, Truncation lag parameter = 1, p-value =
## 0.01375
##
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -2.2003, Lag order = 3, p-value = 0.4946
## alternative hypothesis: stationary
```

```
rest_exp_dates_diff = rest_exp_dates[2:length(rest_exp_dates)]
plot(rest_exp_dates_diff, rest_real_exp_diff_log, type = "l",
    xlab = "Date", ylab = "Restaurant Expenditures Growth Rate",
    main = "Real Restaurant Expenditures Growth Rate, Quarterly")
```



```
# create var + granger causality for real exp and gdp
gdp_exp_combined = cbind(rest_real_exp_diff_log, gdp_growth_subset)
select = VARselect(gdp_exp_combined, lag.max = 4, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
vm_gdp_exp = VAR(gdp_exp_combined, select$select[1])
# plot(vm_gdp_exp$y)
summary(vm_gdp_exp)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: rest_real_exp_diff_log, gdp_growth_subset
## Deterministic variables: const
## Sample size: 40
## Log Likelihood: 309.26
## Roots of the characteristic polynomial:
## 0.829 0.7485 0.7485 0.7006 0.5467 0.5467
## Call:
## VAR(y = gdp_exp_combined, p = select$select[1])
##
##
## Estimation results for equation rest_real_exp_diff_log:
## =========================================================
## rest_real_exp_diff_log = rest_real_exp_diff_log.l1 + gdp_growth_subset.l1 + rest_real_exp_diff_log.l2
##
##                            Estimate Std. Error t value Pr(>|t|)
## rest_real_exp_diff_log.l1 -0.3061642  0.2190439  -1.398  0.17152
## gdp_growth_subset.l1       0.7168516  0.2564126   2.796  0.00857 **
## rest_real_exp_diff_log.l2  0.1645218  0.1996566   0.824  0.41584
## gdp_growth_subset.l2       0.1951121  0.2662260   0.733  0.46880
## rest_real_exp_diff_log.l3  0.1923951  0.1868020   1.030  0.31053
## gdp_growth_subset.l3       0.1265239  0.2690216   0.470  0.64123
## const                     -0.0001735  0.0013498  -0.129  0.89852
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.007011 on 33 degrees of freedom
## Multiple R-Squared: 0.4144,  Adjusted R-squared: 0.3079
## F-statistic: 3.892 on 6 and 33 DF,  p-value: 0.004777
##
##
## Estimation results for equation gdp_growth_subset:
## ====================================================
## gdp_growth_subset = rest_real_exp_diff_log.l1 + gdp_growth_subset.l1 + rest_real_exp_diff_log.l2 + gd
##
##                           Estimate Std. Error t value Pr(>|t|)
## rest_real_exp_diff_log.l1 -0.095202   0.191103  -0.498   0.6217
## gdp_growth_subset.l1       0.469106   0.223705   2.097   0.0437 *
## rest_real_exp_diff_log.l2  0.070094   0.174189   0.402   0.6900
## gdp_growth_subset.l2      -0.008452   0.232267  -0.036   0.9712
## rest_real_exp_diff_log.l3  0.323437   0.162974   1.985   0.0556 .
## gdp_growth_subset.l3      -0.292756   0.234706  -1.247   0.2211
## const                      0.001641   0.001178   1.393   0.1729
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.006116 on 33 degrees of freedom
## Multiple R-Squared: 0.2667,  Adjusted R-squared: 0.1333
## F-statistic:     2 on 6 and 33 DF,  p-value: 0.09384
##
##
##
## Covariance matrix of residuals:
##                        rest_real_exp_diff_log gdp_growth_subset
## rest_real_exp_diff_log              4.915e-05         2.948e-05
## gdp_growth_subset                   2.948e-05         3.741e-05
##
```

```
## Correlation matrix of residuals:
##                        rest_real_exp_diff_log gdp_growth_subset
## rest_real_exp_diff_log                 1.0000            0.6874
## gdp_growth_subset                      0.6874            1.0000
```

```
grangertest(rest_real_exp_diff_log ~ gdp_growth_subset[1:length(gdp_growth_subset)],
    order = select$select[1])
```

```
## Granger causality test
##
## Model 1: rest_real_exp_diff_log ~ Lags(rest_real_exp_diff_log, 1:3) + Lags(gdp_growth_subset[1:lengt
## Model 2: rest_real_exp_diff_log ~ Lags(rest_real_exp_diff_log, 1:3)
##   Res.Df Df      F  Pr(>F)
## 1     33
## 2     36 -3 2.8912 0.05002 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
grangertest(gdp_growth_subset[1:length(gdp_growth_subset)] ~
    rest_real_exp_diff_log, order = select$select[1])
```

```
## Granger causality test
##
## Model 1: gdp_growth_subset[1:length(gdp_growth_subset)] ~ Lags(gdp_growth_subset[1:length(gdp_growth_
## Model 2: gdp_growth_subset[1:length(gdp_growth_subset)] ~ Lags(gdp_growth_subset[1:length(gdp_growth_
##   Res.Df Df      F Pr(>F)
## 1     33
## 2     36 -3 1.3863 0.2642
```

```
rec_exp_diff_log_dummy = add_recession_dummy(rest_exp_dates_diff)
lm_rest_real_exp_diff_log = lm(rest_real_exp_diff_log ~ rec_exp_diff_log_dummy)
summary(lm_rest_real_exp_diff_log)
```

```
##
## Call:
## lm(formula = rest_real_exp_diff_log ~ rec_exp_diff_log_dummy)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0158435 -0.0038443 -0.0000908  0.0033176  0.0144583
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.005997   0.001061   5.654 1.34e-06 ***
## rec_exp_diff_log_dummy -0.015065   0.002629  -5.730 1.05e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006364 on 41 degrees of freedom
## Multiple R-squared:  0.4447, Adjusted R-squared:  0.4312
## F-statistic: 32.84 on 1 and 41 DF,  p-value: 1.046e-06
```

yes gdp granger causes restaurant expenditures and recession dummy

connect num of reviews with restaurants

need to convert number of reviews to be on same scale as rest expenditure growth rates

```
rev_exp_combined = cbind(rest_real_exp_diff_log, log_rev_quarter)
select = VARselect(rev_exp_combined, lag.max = 4, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
vm_rev_exp = VAR(rev_exp_combined, select$select[1])
# plot(vm_rev_exp$y)
summary(vm_rev_exp)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: rest_real_exp_diff_log, log_rev_quarter
## Deterministic variables: const
## Sample size: 39
## Log Likelihood: 190.353
## Roots of the characteristic polynomial:
## 0.9262 0.9126 0.868 0.868 0.5991 0.5991 0.5268 0.5268
## Call:
## VAR(y = rev_exp_combined, p = select$select[1])
##
##
## Estimation results for equation rest_real_exp_diff_log:
## =======================================================
## rest_real_exp_diff_log = rest_real_exp_diff_log.l1 + log_rev_quarter.l1 + rest_real_exp_diff_log.l2 +
##
##                            Estimate Std. Error t value Pr(>|t|)
## rest_real_exp_diff_log.l1  0.2010747  0.1805061   1.114   0.2741
## log_rev_quarter.l1        -0.0077785  0.0088976  -0.874   0.3889
## rest_real_exp_diff_log.l2  0.2249583  0.1755076   1.282   0.2097
## log_rev_quarter.l2        -0.0041271  0.0076756  -0.538   0.5948
## rest_real_exp_diff_log.l3  0.1365037  0.1726607   0.791   0.4354
## log_rev_quarter.l3         0.0005648  0.0077776   0.073   0.9426
## rest_real_exp_diff_log.l4 -0.0140669  0.1650931  -0.085   0.9327
## log_rev_quarter.l4        -0.0107478  0.0078840  -1.363   0.1830
## const                      0.0044418  0.0025252   1.759   0.0888 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.007607 on 30 degrees of freedom
## Multiple R-Squared: 0.3578,  Adjusted R-squared: 0.1866
## F-statistic:  2.09 on 8 and 30 DF,  p-value: 0.06888
##
##
## Estimation results for equation log_rev_quarter:
## ================================================
## log_rev_quarter = rest_real_exp_diff_log.l1 + log_rev_quarter.l1 + rest_real_exp_diff_log.l2 + log_r
##
##                          Estimate Std. Error t value Pr(>|t|)
## rest_real_exp_diff_log.l1 -0.87806    1.83748  -0.478  0.63621
## log_rev_quarter.l1         0.12723    0.09057   1.405  0.17040
## rest_real_exp_diff_log.l2  3.28692    1.78660   1.840  0.07572 .  
## log_rev_quarter.l2         0.07675    0.07813   0.982  0.33380    
## rest_real_exp_diff_log.l3  0.73014    1.75762   0.415  0.68080    
## log_rev_quarter.l3        -0.16649    0.07917  -2.103  0.04397 *  
## rest_real_exp_diff_log.l4 -5.18564    1.68058  -3.086  0.00434 ** 
## log_rev_quarter.l4         0.54755    0.08026   6.823 1.44e-07 ***
## const                      0.03019    0.02571   1.174  0.24945    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## 
## Residual standard error: 0.07744 on 30 degrees of freedom
## Multiple R-Squared: 0.8139,  Adjusted R-squared: 0.7642 
## F-statistic:  16.4 on 8 and 30 DF,  p-value: 5.127e-09
## 
## 
## 
## Covariance matrix of residuals:
##                        rest_real_exp_diff_log log_rev_quarter
## rest_real_exp_diff_log              5.787e-05       0.0001152
## log_rev_quarter                     1.152e-04       0.0059970
## 
## Correlation matrix of residuals:
##                        rest_real_exp_diff_log log_rev_quarter
## rest_real_exp_diff_log                 1.0000          0.1955
## log_rev_quarter                        0.1955          1.0000
```

```
grangertest(rest_real_exp_diff_log ~ log_rev_quarter, order = select$select[1])
```

```
## Granger causality test
## 
## Model 1: rest_real_exp_diff_log ~ Lags(rest_real_exp_diff_log, 1:4) + Lags(log_rev_quarter, 1:4)
## Model 2: rest_real_exp_diff_log ~ Lags(rest_real_exp_diff_log, 1:4)
##   Res.Df Df      F Pr(>F)
## 1     30                 
## 2     34 -4 0.9082 0.4718
```

```
grangertest(log_rev_quarter ~ rest_real_exp_diff_log, order = select$select[1])
```

```
## Granger causality test
## 
## Model 1: log_rev_quarter ~ Lags(log_rev_quarter, 1:4) + Lags(rest_real_exp_diff_log, 1:4)
## Model 2: log_rev_quarter ~ Lags(log_rev_quarter, 1:4)
##   Res.Df Df      F  Pr(>F)  
## 1     30                    
## 2     34 -4 2.8674 0.04005 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# try to seasonally adjust leave trend in because the
# original seasonally adjusted rest exp had a trend

log_rev_quarter_tslm = tslm(log_rev_quarter ~ season)
summary(log_rev_quarter_tslm)
```

```
## 
```

```
## Call:
## tslm(formula = log_rev_quarter ~ season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34782 -0.07594 -0.00878  0.04565  0.59583
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.35264    0.05057   6.973 2.34e-08 ***
## season2     -0.34070    0.06988  -4.876 1.85e-05 ***
## season3     -0.15807    0.06988  -2.262   0.0293 *
## season4     -0.41248    0.06988  -5.903 7.06e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1599 on 39 degrees of freedom
## Multiple R-squared:  0.5214, Adjusted R-squared:  0.4846
## F-statistic: 14.16 on 3 and 39 DF,  p-value: 2.155e-06
```

```
log_rev_quarter_adj = resid(log_rev_quarter_tslm)
# plot(log_rev_quarter_adj)


rev_exp_adj_combined = cbind(rest_real_exp_diff_log, log_rev_quarter_adj)
select = VARselect(rev_exp_adj_combined, lag.max = 4, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
vm_rev_adj_exp = VAR(rev_exp_adj_combined, select$select[1])
# plot(vm_rev_adj_exp$y)
summary(vm_rev_adj_exp)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: rest_real_exp_diff_log, log_rev_quarter_adj
## Deterministic variables: const
## Sample size: 39
## Log Likelihood: 196.578
## Roots of the characteristic polynomial:
## 0.9055 0.828 0.721 0.721 0.5383 0.5383 0.4846 0.4846
## Call:
## VAR(y = rev_exp_adj_combined, p = select$select[1])
##
##
## Estimation results for equation rest_real_exp_diff_log:
## =======================================================
## rest_real_exp_diff_log = rest_real_exp_diff_log.l1 + log_rev_quarter_adj.l1 + rest_real_exp_diff_log
##
##                            Estimate Std. Error t value Pr(>|t|)
## rest_real_exp_diff_log.l1  0.180897   0.180663   1.001    0.325
## log_rev_quarter_adj.l1    -0.010709   0.014668  -0.730    0.471
## rest_real_exp_diff_log.l2  0.239154   0.178751   1.338    0.191
## log_rev_quarter_adj.l2    -0.004026   0.011875  -0.339    0.737
## rest_real_exp_diff_log.l3  0.099466   0.168370   0.591    0.559
## log_rev_quarter_adj.l3     0.003002   0.012594   0.238    0.813
## rest_real_exp_diff_log.l4  0.026576   0.166231   0.160    0.874
## log_rev_quarter_adj.l4    -0.011961   0.011398  -1.049    0.302
## const                      0.001700   0.001486   1.144    0.262
##
##
## Residual standard error: 0.007706 on 30 degrees of freedom
## Multiple R-Squared: 0.3411,  Adjusted R-squared: 0.1654
## F-statistic: 1.941 on 8 and 30 DF,  p-value: 0.09013
##
##
## Estimation results for equation log_rev_quarter_adj:
## ====================================================
## log_rev_quarter_adj = rest_real_exp_diff_log.l1 + log_rev_quarter_adj.l1 + rest_real_exp_diff_log.l2
##
##                           Estimate Std. Error t value Pr(>|t|)
## rest_real_exp_diff_log.l1 -2.47221    1.57285  -1.572  0.12649
## log_rev_quarter_adj.l1     0.15930    0.12770   1.247  0.22186
## rest_real_exp_diff_log.l2  2.79746    1.55620   1.798  0.08231 .
## log_rev_quarter_adj.l2     0.22813    0.10338   2.207  0.03513 *
## rest_real_exp_diff_log.l3 -0.09956    1.46582  -0.068  0.94630
## log_rev_quarter_adj.l3    -0.15938    0.10964  -1.454  0.15643
## rest_real_exp_diff_log.l4 -2.85185    1.44720  -1.971  0.05806 .
## log_rev_quarter_adj.l4     0.28534    0.09923   2.875  0.00735 **
## const                     -0.01439    0.01293  -1.112  0.27477
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.06709 on 30 degrees of freedom
## Multiple R-Squared: 0.5769,  Adjusted R-squared: 0.4641
## F-statistic: 5.114 on 8 and 30 DF,  p-value: 0.0004486
##
##
##
## Covariance matrix of residuals:
##                        rest_real_exp_diff_log log_rev_quarter_adj
## rest_real_exp_diff_log              5.938e-05            0.000157
## log_rev_quarter_adj                 1.570e-04            0.004501
##
## Correlation matrix of residuals:
##                        rest_real_exp_diff_log log_rev_quarter_adj
## rest_real_exp_diff_log                 1.0000              0.3038
## log_rev_quarter_adj                    0.3038              1.0000
```

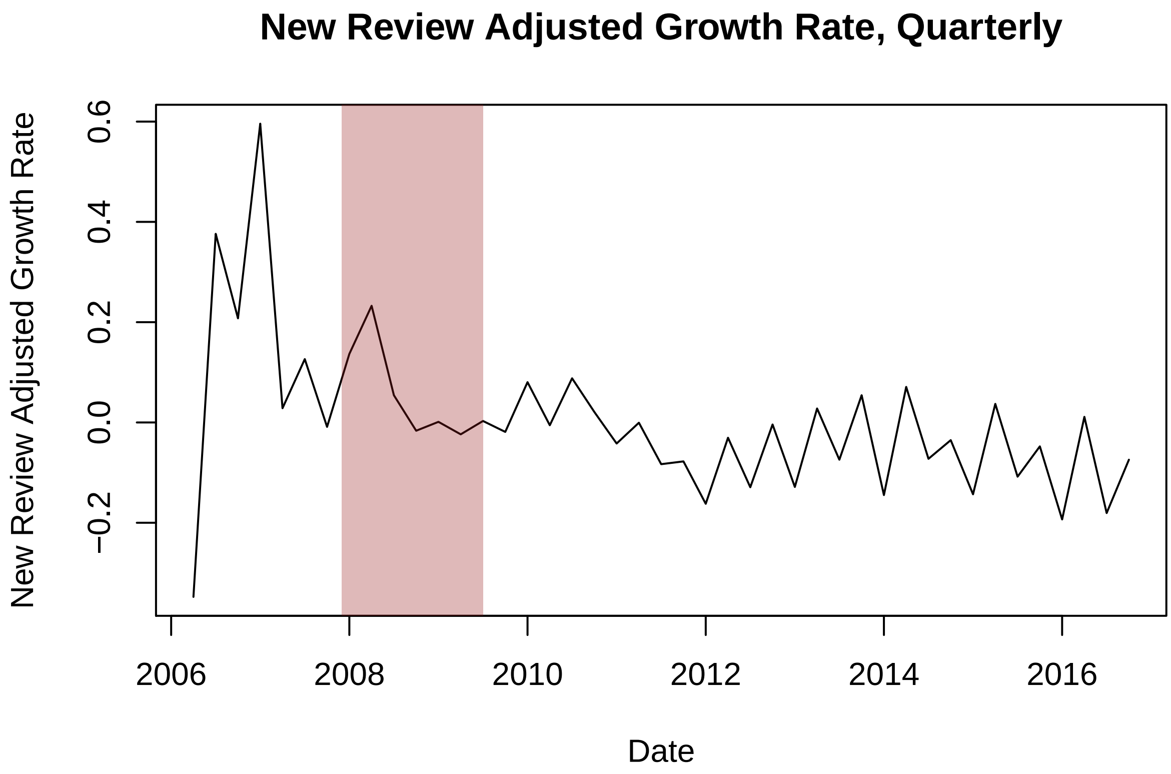
```
grangertest(rest_real_exp_diff_log ~ log_rev_quarter_adj, order = select$select[1])
```

```
## Granger causality test
##
## Model 1: rest_real_exp_diff_log ~ Lags(rest_real_exp_diff_log, 1:4) + Lags(log_rev_quarter_adj, 1:4)
## Model 2: rest_real_exp_diff_log ~ Lags(rest_real_exp_diff_log, 1:4)
##   Res.Df Df      F Pr(>F)
## 1     30
## 2     34 -4 0.6949 0.6014
```

```
grangertest(log_rev_quarter_adj ~ rest_real_exp_diff_log, order = select$select[1])
```

```
## Granger causality test
##
## Model 1: log_rev_quarter_adj ~ Lags(log_rev_quarter_adj, 1:4) + Lags(rest_real_exp_diff_log, 1:4)
## Model 2: log_rev_quarter_adj ~ Lags(log_rev_quarter_adj, 1:4)
##   Res.Df Df      F  Pr(>F)
## 1     30
## 2     34 -4 2.2997 0.08183 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(log_rev_quarter_adj, xlab = "Date", ylab = "New Review Adjusted Growth Rate",
    main = "New Review Adjusted Growth Rate, Quarterly")
rect(2007.9166667, -1, 2009.5, 1, col = rgb(red = 150/255, green = 25/255,
    blue = 25/255, alpha = 0.3), border = NA)
```



```
# plot(rest_real_exp_diff_log,type='l')


# only the lags
lm_temp = lm(log_rev_quarter_adj ~ rest_real_exp_diff_log)

summary(lm_temp)
```

```
##
## Call:
```

```
## lm(formula = log_rev_quarter_adj ~ rest_real_exp_diff_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35555 -0.07711 -0.01678  0.04534  0.58897
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.002987   0.025818   0.116    0.908
## rest_real_exp_diff_log -0.842576   2.849128  -0.296    0.769
##
## Residual standard error: 0.1558 on 41 degrees of freedom
## Multiple R-squared:  0.002129,   Adjusted R-squared:  -0.02221
## F-statistic: 0.08746 on 1 and 41 DF,  p-value: 0.7689
```

build rest exp into the dollar signs models

```
# convert to quarterly add in rest exp try var model with
# adding in rest exp

# adjusted num new rev by dollar signs
d1_quarterly = apply.quarterly(as.xts(tslm_d1_resid), FUN = sum)
ts_d1_quarterly = ts(d1_quarterly, start = c(2006, 1), freq = 4)

d2_quarterly = apply.quarterly(as.xts(tslm_d2_resid), FUN = sum)
ts_d2_quarterly = ts(d2_quarterly, start = c(2006, 1), freq = 4)

d3_quarterly = apply.quarterly(as.xts(tslm_d3_resid), FUN = sum)
ts_d3_quarterly = ts(d3_quarterly, start = c(2006, 1), freq = 4)

d4_quarterly = apply.quarterly(as.xts(tslm_d4_resid), FUN = sum)
ts_d4_quarterly = ts(d4_quarterly, start = c(2006, 1), freq = 4)

# review counts by quarter
rev_count_adj_q = apply.quarterly(as.xts(ts_rev_count_adj), FUN = sum)
ts_rev_count_adj_q = ts(rev_count_adj_q, start = c(2006, 1),
    freq = 4)

# dolla dolla stars
d1_star_quarterly = apply.quarterly(xts(df_dollars_1_star$avg,
    as.Date(df_dollars_1_star$date, "%Y-%m-%d")), FUN = sum)
ts_d1_star_quarterly = ts(d1_star_quarterly, start = c(2006,
    1), freq = 4)

d2_star_quarterly = apply.quarterly(xts(df_dollars_2_star$avg,
    as.Date(df_dollars_2_star$date, "%Y-%m-%d")), FUN = sum)
ts_d2_star_quarterly = ts(d2_star_quarterly, start = c(2006,
    1), freq = 4)

d3_star_quarterly = apply.quarterly(xts(df_dollars_3_star$avg,
    as.Date(df_dollars_3_star$date, "%Y-%m-%d")), FUN = sum)
ts_d3_star_quarterly = ts(d3_star_quarterly, start = c(2006,
```

```r
    1), freq = 4)

d4_quarterly = apply.quarterly(xts(df_dollars_4_star$avg, as.Date(df_dollars_4_star$date,
    "%Y-%m-%d")), FUN = sum)
ts_d4_star_quarterly = ts(d4_quarterly, start = c(2006, 1), freq = 4)

# rec quarter
rec_q = add_recession_dummy(index(d1_star_quarterly))

# since in levels, use trend adjusted level of expenditures
# (detrend)

rest_exp_q = ts(restaurant_expenditures$real_exp, start = c(2006,
    1), freq = 4)
rest_exp_q_tslm = tslm(rest_exp_q ~ trend)
rest_exp_q_adj = resid(rest_exp_q_tslm)






lm_d1_full = lm(ts_d1_quarterly ~ rec_q + ts_rev_count_adj_q +
    ts_d1_star_quarterly + rest_exp_q_adj)
summary(lm_d1_full)
```

```
##
## Call:
## lm(formula = ts_d1_quarterly ~ rec_q + ts_rev_count_adj_q + ts_d1_star_quarterly +
##     rest_exp_q_adj)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -569.29  -97.37  -17.66  121.28  408.08
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -8.777e+02  1.536e+03  -0.572    0.571
## rec_q                 5.358e+01  8.628e+01   0.621    0.538
## ts_rev_count_adj_q    3.034e-01  5.399e-03  56.207   <2e-16 ***
## ts_d1_star_quarterly  7.738e+01  1.364e+02   0.567    0.574
## rest_exp_q_adj        5.007e+00  3.017e+00   1.660    0.105
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201.6 on 39 degrees of freedom
## Multiple R-squared:  0.9975, Adjusted R-squared:  0.9972
## F-statistic:  3851 on 4 and 39 DF,  p-value: < 2.2e-16
```

```r
lm_d2_full = lm(ts_d2_quarterly ~ rec_q + ts_rev_count_adj_q +
    ts_d2_star_quarterly + rest_exp_q_adj)
summary(lm_d2_full)
```

```
##
```

```
## Call:
## lm(formula = ts_d2_quarterly ~ rec_q + ts_rev_count_adj_q + ts_d2_star_quarterly + 
##     rest_exp_q_adj)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -502.56 -211.31    6.58  205.47  656.43 
## 
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           -2.595e+03  4.914e+03  -0.528   0.6005    
## rec_q                  2.731e+02  1.440e+02   1.896   0.0654 .  
## ts_rev_count_adj_q     6.637e-01  8.055e-03  82.387   <2e-16 ***
## ts_d2_star_quarterly   2.323e+02  4.462e+02   0.521   0.6056    
## rest_exp_q_adj         2.197e+00  4.883e+00   0.450   0.6552    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 277.9 on 39 degrees of freedom
## Multiple R-squared:  0.999,  Adjusted R-squared:  0.9989 
## F-statistic:  9367 on 4 and 39 DF,  p-value: < 2.2e-16
```

```
lm_d3_full = lm(ts_d3_quarterly ~ rec_q + ts_rev_count_adj_q +
    ts_d3_star_quarterly + rest_exp_q_adj)
summary(lm_d3_full)
```

```
## 
## Call:
## lm(formula = ts_d3_quarterly ~ rec_q + ts_rev_count_adj_q + ts_d3_star_quarterly + 
##     rest_exp_q_adj)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -568.78 -164.74    9.84  174.17  465.42 
## 
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            4.254e+03  3.553e+03   1.197   0.2384    
## rec_q                 -2.089e+02  1.039e+02  -2.011   0.0513 .  
## ts_rev_count_adj_q     2.963e-02  6.739e-03   4.397 8.22e-05 ***
## ts_d3_star_quarterly -3.705e+02  3.116e+02  -1.189   0.2418    
## rest_exp_q_adj        -6.025e+00  3.829e+00  -1.573   0.1237    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 247.2 on 39 degrees of freedom
## Multiple R-squared:  0.5027, Adjusted R-squared:  0.4517 
## F-statistic: 9.856 on 4 and 39 DF,  p-value: 1.311e-05
```

```
lm_d4_full = lm(ts_d4_quarterly ~ rec_q + ts_rev_count_adj_q +
    ts_d4_star_quarterly + rest_exp_q_adj)
summary(lm_d4_full)
```

```
## 
## Call:
```

```
## lm(formula = ts_d4_quarterly ~ rec_q + ts_rev_count_adj_q + ts_d4_star_quarterly +
##     rest_exp_q_adj)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -316.46  -67.73   15.81   65.89  215.63
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -9.689e+02  1.208e+03  -0.802   0.4273
## rec_q                -7.340e+01  4.909e+01  -1.495   0.1429
## ts_rev_count_adj_q    1.116e-03  3.115e-03   0.358   0.7221
## ts_d4_star_quarterly  8.192e+01  1.010e+02   0.811   0.4224
## rest_exp_q_adj       -3.740e+00  1.710e+00  -2.187   0.0348 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 114.6 on 39 degrees of freedom
## Multiple R-squared:  0.2986, Adjusted R-squared:  0.2267
## F-statistic: 4.152 on 4 and 39 DF,  p-value: 0.006758
```

```r
vif(lm_d1_full)
```

```
##                rec_q   ts_rev_count_adj_q ts_d1_star_quarterly
##             1.078443             4.614419             1.318413
##       rest_exp_q_adj
##             4.514264
```

```r
vif(lm_d2_full)
```

```
##                rec_q   ts_rev_count_adj_q ts_d2_star_quarterly
##             1.581756             5.405495             6.411727
##       rest_exp_q_adj
##             6.224056
```

```r
vif(lm_d3_full)
```

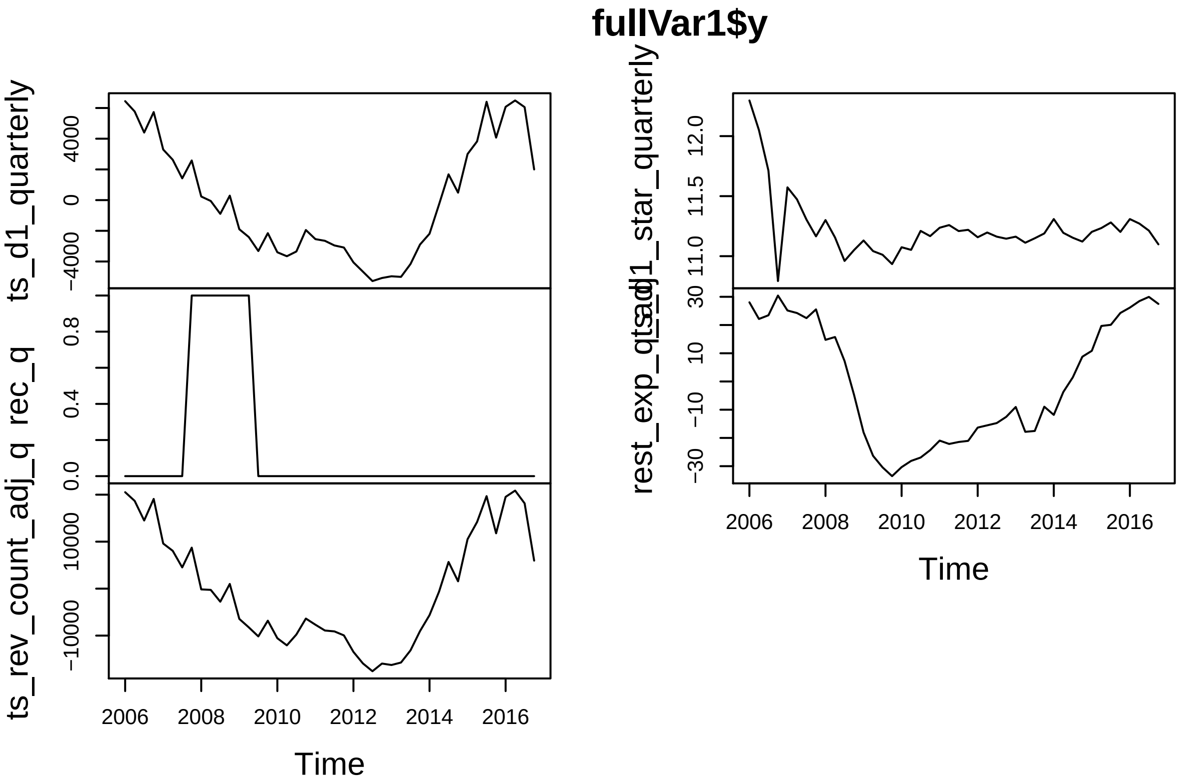
```
##                rec_q   ts_rev_count_adj_q ts_d3_star_quarterly
##             1.039758             4.779719             2.219843
##       rest_exp_q_adj
##             4.836175
```

```r
vif(lm_d4_full)
```

```
##                rec_q   ts_rev_count_adj_q ts_d4_star_quarterly
##             1.079835             4.752153             1.268115
##       rest_exp_q_adj
##             4.487196
```

```r
# lots of insignificance and multicollinearity how about a
# var

combinedFull1 = cbind(ts_d1_quarterly, rec_q, ts_rev_count_adj_q,
    ts_d1_star_quarterly, rest_exp_q_adj)
select = VARselect(combinedFull1, lag.max = 4, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
fullVar1 = VAR(combinedFull1, p = select$select[1])
```

```r
plot(fullVar1$y)
```

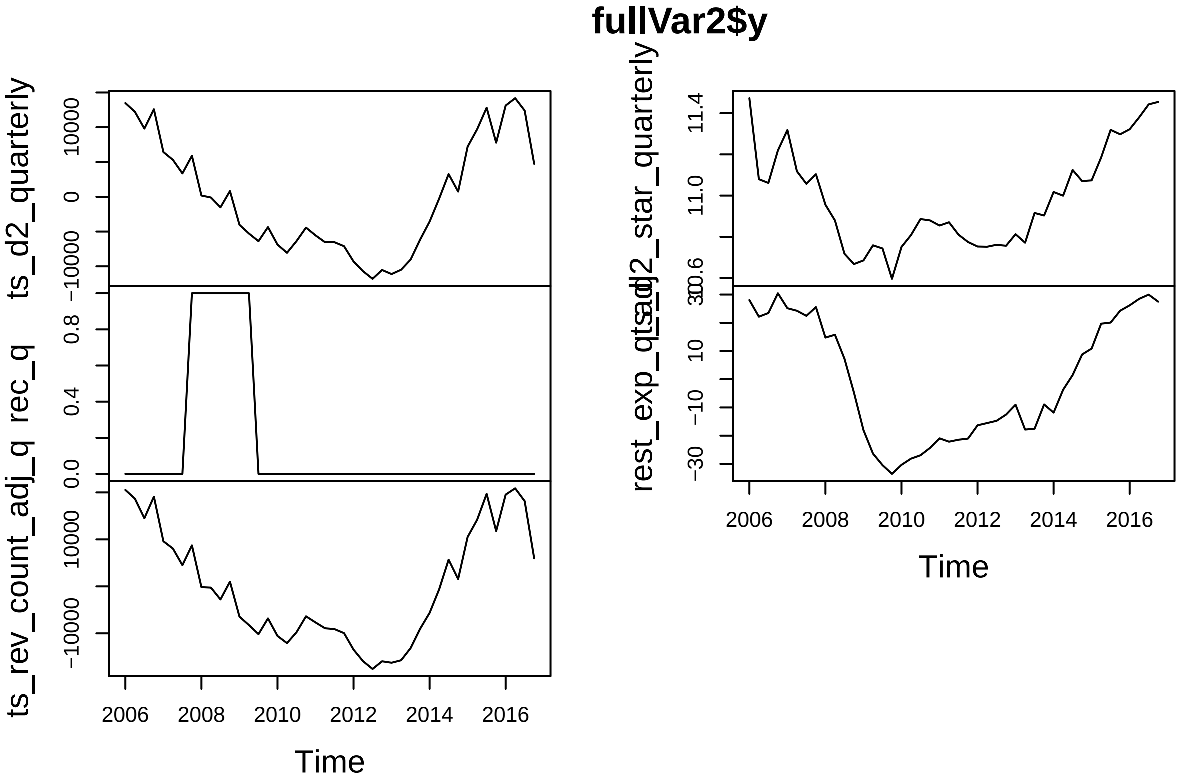


```r
summary(fullVar1)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: ts_d1_quarterly, rec_q, ts_rev_count_adj_q, ts_d1_star_quarterly, rest_exp_q_ad
## Deterministic variables: const
## Sample size: 40
## Log Likelihood: -607.768
## Roots of the characteristic polynomial:
## 0.9619 0.9619 0.9122 0.9122 0.9091 0.9091 0.8924 0.8924 0.8686 0.8686 0.8103 0.8103 0.7115 0.7048 0.
## Call:
## VAR(y = combinedFull1, p = select$select[1])
##
##
## Estimation results for equation ts_d1_quarterly:
## ================================================
## ts_d1_quarterly = ts_d1_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d1_star_quarterly.l1 +
##
##                          Estimate Std. Error t value Pr(>|t|)
## ts_d1_quarterly.l1         -2.4505     0.9469  -2.588  0.01805 *
## rec_q.l1                -1296.1353   674.3661  -1.922  0.06973 .
## ts_rev_count_adj_q.l1       0.9371     0.2587   3.622  0.00181 **
## ts_d1_star_quarterly.l1 -2428.5334  1747.9646  -1.389  0.18080
## rest_exp_q_adj.l1          45.8319    42.1551   1.087  0.29054
## ts_d1_quarterly.l2         -0.2245     0.9752  -0.230  0.82043
## rec_q.l2                  812.1205   876.3552   0.927  0.36570
## ts_rev_count_adj_q.l2       0.1788     0.2901   0.616  0.54505
## ts_d1_star_quarterly.l2 -2021.2192  1317.7635  -1.534  0.14156
## rest_exp_q_adj.l2         -19.0230    46.6645  -0.408  0.68808
## ts_d1_quarterly.l3         -0.7753     0.9386  -0.826  0.41907
## rec_q.l3                 -409.2881   883.2202  -0.463  0.64834
## ts_rev_count_adj_q.l3       0.1171     0.2809   0.417  0.68131
## ts_d1_star_quarterly.l3 -1187.8833  1081.3469  -1.099  0.28570
## rest_exp_q_adj.l3          74.6138    47.7373   1.563  0.13455
## ts_d1_quarterly.l4         -4.1616     0.7263  -5.730  1.6e-05 ***
## rec_q.l4                 -278.5479   931.0872  -0.299  0.76806
## ts_rev_count_adj_q.l4       1.3468     0.2403   5.604  2.1e-05 ***
## ts_d1_star_quarterly.l4 -1297.1193   691.6660  -1.875  0.07620 .
## rest_exp_q_adj.l4         -31.6419    38.1129  -0.830  0.41673
## const                   77721.5970 37132.9713   2.093  0.05000 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 758.6 on 19 degrees of freedom
## Multiple R-Squared: 0.9776,  Adjusted R-squared: 0.954
## F-statistic: 41.43 on 20 and 19 DF,  p-value: 1.2e-11
##
##
## Estimation results for equation rec_q:
## ======================================
## rec_q = ts_d1_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d1_star_quarterly.l1 + rest_exp_q
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d1_quarterly.l1      -1.461e-04  2.642e-04  -0.553 0.586741
## rec_q.l1                 7.662e-01  1.881e-01   4.073 0.000649 ***
## ts_rev_count_adj_q.l1    2.935e-05  7.217e-05   0.407 0.688747
## ts_d1_star_quarterly.l1 -7.929e-02  4.876e-01  -0.163 0.872542
## rest_exp_q_adj.l1       -1.228e-03  1.176e-02  -0.104 0.917929
## ts_d1_quarterly.l2      -3.517e-04  2.721e-04  -1.293 0.211625
## rec_q.l2                -2.050e-01  2.445e-01  -0.838 0.412266
## ts_rev_count_adj_q.l2    1.005e-04  8.093e-05   1.242 0.229475
## ts_d1_star_quarterly.l2  9.548e-02  3.676e-01   0.260 0.797879
## rest_exp_q_adj.l2        1.161e-02  1.302e-02   0.892 0.383461
## ts_d1_quarterly.l3      -2.358e-04  2.618e-04  -0.901 0.379108
## rec_q.l3                 9.077e-02  2.464e-01   0.368 0.716660
## ts_rev_count_adj_q.l3    6.173e-05  7.835e-05   0.788 0.440474
## ts_d1_star_quarterly.l3  1.678e-01  3.017e-01   0.556 0.584481
## rest_exp_q_adj.l3        3.998e-03  1.332e-02   0.300 0.767293
## ts_d1_quarterly.l4      -7.021e-05  2.026e-04  -0.346 0.732780
## rec_q.l4                -3.850e-02  2.597e-01  -0.148 0.883743
## ts_rev_count_adj_q.l4    4.796e-05  6.704e-05   0.715 0.483041
## ts_d1_star_quarterly.l4 -4.711e-01  1.930e-01  -2.442 0.024575 *
## rest_exp_q_adj.l4       -2.287e-03  1.063e-02  -0.215 0.831996
## const                    3.305e+00  1.036e+01   0.319 0.753154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##
## Residual standard error: 0.2116 on 19 degrees of freedom
## Multiple R-Squared: 0.8526,  Adjusted R-squared: 0.6975
## F-statistic: 5.497 on 20 and 19 DF,  p-value: 0.0002359
##
##
## Estimation results for equation ts_rev_count_adj_q:
## ===================================================
## ts_rev_count_adj_q = ts_d1_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d1_star_quarterly.l1
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d1_quarterly.l1      -8.486e+00  3.495e+00  -2.428  0.02530 *
## rec_q.l1                -4.701e+03  2.489e+03  -1.889  0.07431 .
## ts_rev_count_adj_q.l1    3.162e+00  9.550e-01   3.311  0.00367 **
## ts_d1_star_quarterly.l1 -9.240e+03  6.452e+03  -1.432  0.16840
## rest_exp_q_adj.l1        1.595e+02  1.556e+02   1.025  0.31811
## ts_d1_quarterly.l2      -1.134e+00  3.600e+00  -0.315  0.75613
## rec_q.l2                 2.728e+03  3.235e+03   0.843  0.40958
## ts_rev_count_adj_q.l2    6.697e-01  1.071e+00   0.625  0.53920
## ts_d1_star_quarterly.l2 -8.071e+03  4.864e+03  -1.659  0.11351
## rest_exp_q_adj.l2       -5.756e+01  1.723e+02  -0.334  0.74190
## ts_d1_quarterly.l3      -2.651e+00  3.465e+00  -0.765  0.45355
## rec_q.l3                -1.436e+03  3.260e+03  -0.440  0.66462
## ts_rev_count_adj_q.l3    4.921e-01  1.037e+00   0.475  0.64042
## ts_d1_star_quarterly.l3 -4.684e+03  3.992e+03  -1.173  0.25511
## rest_exp_q_adj.l3        2.175e+02  1.762e+02   1.234  0.23209
## ts_d1_quarterly.l4      -1.452e+01  2.681e+00  -5.416 3.16e-05 ***
## rec_q.l4                -1.388e+03  3.437e+03  -0.404  0.69088
## ts_rev_count_adj_q.l4    4.648e+00  8.871e-01   5.239 4.67e-05 ***
## ts_d1_star_quarterly.l4 -4.557e+03  2.553e+03  -1.785  0.09026 .
## rest_exp_q_adj.l4       -6.438e+01  1.407e+02  -0.458  0.65241
## const                    2.977e+05  1.371e+05   2.172  0.04275 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 2800 on 19 degrees of freedom
## Multiple R-Squared: 0.9699,  Adjusted R-squared: 0.9383
## F-statistic: 30.64 on 20 and 19 DF,  p-value: 1.836e-10
##
##
## Estimation results for equation ts_d1_star_quarterly:
## =====================================================
## ts_d1_star_quarterly = ts_d1_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d1_star_quarterly.l
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d1_quarterly.l1      -2.754e-05  8.599e-05  -0.320 0.752245
## rec_q.l1                 5.655e-02  6.124e-02   0.923 0.367365
## ts_rev_count_adj_q.l1    1.374e-05  2.349e-05   0.585 0.565476
## ts_d1_star_quarterly.l1 -8.605e-02  1.587e-01  -0.542 0.594077
## rest_exp_q_adj.l1       -4.349e-03  3.828e-03  -1.136 0.270056
## ts_d1_quarterly.l2      -2.109e-05  8.856e-05  -0.238 0.814304
## rec_q.l2                -1.626e-01  7.958e-02  -2.044 0.055116 .
## ts_rev_count_adj_q.l2    4.982e-06  2.635e-05   0.189 0.852019
## ts_d1_star_quarterly.l2 -7.464e-02  1.197e-01  -0.624 0.540234
## rest_exp_q_adj.l2        2.949e-04  4.238e-03   0.070 0.945253
## ts_d1_quarterly.l3      -1.122e-04  8.524e-05  -1.316 0.203875
## rec_q.l3                -9.149e-02  8.021e-02  -1.141 0.268188
## ts_rev_count_adj_q.l3    2.878e-05  2.551e-05   1.129 0.273137
## ts_d1_star_quarterly.l3 -6.640e-03  9.820e-02  -0.068 0.946796
## rest_exp_q_adj.l3        2.915e-04  4.335e-03   0.067 0.947097
## ts_d1_quarterly.l4      -1.281e-04  6.596e-05  -1.943 0.067027 .
## rec_q.l4                -9.021e-02  8.455e-02  -1.067 0.299373
## ts_rev_count_adj_q.l4    4.595e-05  2.182e-05   2.106 0.048775 *
## ts_d1_star_quarterly.l4  2.471e-01  6.281e-02   3.934 0.000891 ***
## rest_exp_q_adj.l4        2.977e-03  3.461e-03   0.860 0.400484
## const                    1.033e+01  3.372e+00   3.064 0.006392 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.06889 on 19 degrees of freedom
## Multiple R-Squared: 0.8397,  Adjusted R-squared: 0.671
## F-statistic: 4.977 on 20 and 19 DF,  p-value: 0.0004646
##
##
## Estimation results for equation rest_exp_q_adj:
## ===============================================
## rest_exp_q_adj = ts_d1_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d1_star_quarterly.l1 + re
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d1_quarterly.l1       6.031e-05  5.307e-03   0.011   0.9911
## rec_q.l1                -6.766e+00  3.779e+00  -1.790   0.0894 .
## ts_rev_count_adj_q.l1    3.180e-04  1.450e-03   0.219   0.8287
## ts_d1_star_quarterly.l1 -1.046e+00  9.796e+00  -0.107   0.9161
## rest_exp_q_adj.l1        6.094e-01  2.362e-01   2.580   0.0184 *
## ts_d1_quarterly.l2      -9.275e-04  5.465e-03  -0.170   0.8670
## rec_q.l2                 8.373e-01  4.911e+00   0.170   0.8664
## ts_rev_count_adj_q.l2    2.247e-04  1.626e-03   0.138   0.8915
## ts_d1_star_quarterly.l2 -2.740e+00  7.385e+00  -0.371   0.7147
## rest_exp_q_adj.l2        1.506e-01  2.615e-01   0.576   0.5715
## ts_d1_quarterly.l3      -1.452e-03  5.260e-03  -0.276   0.7855
## rec_q.l3                -5.699e+00  4.950e+00  -1.151   0.2639
## ts_rev_count_adj_q.l3    3.304e-04  1.574e-03   0.210   0.8360
## ts_d1_star_quarterly.l3  6.560e-01  6.060e+00   0.108   0.9149
## rest_exp_q_adj.l3        1.604e-01  2.675e-01   0.600   0.5558
## ts_d1_quarterly.l4      -7.497e-03  4.070e-03  -1.842   0.0812 .
## rec_q.l4                -3.198e+00  5.218e+00  -0.613   0.5473
## ts_rev_count_adj_q.l4    2.293e-03  1.347e-03   1.703   0.1050
## ts_d1_star_quarterly.l4 -3.813e+00  3.876e+00  -0.984   0.3377
## rest_exp_q_adj.l4        9.428e-03  2.136e-01   0.044   0.9653
## const                    8.026e+01  2.081e+02   0.386   0.7040
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 4.251 on 19 degrees of freedom
## Multiple R-Squared:  0.98,   Adjusted R-squared: 0.9588 
## F-statistic: 46.43 on 20 and 19 DF,  p-value: 4.238e-12
## 
## 
## 
## Covariance matrix of residuals:
##                      ts_d1_quarterly      rec_q ts_rev_count_adj_q
## ts_d1_quarterly            575512.07 -6.170e+01         2067622.85
## rec_q                         -61.70  4.479e-02            -183.88
## ts_rev_count_adj_q        2067622.85 -1.839e+02         7842058.98
## ts_d1_star_quarterly           18.06 -1.698e-03              67.87
## rest_exp_q_adj                244.78 -1.052e-01            1110.43
##                      ts_d1_star_quarterly rest_exp_q_adj
## ts_d1_quarterly                 18.059377       244.7838
## rec_q                           -0.001698        -0.1052
## ts_rev_count_adj_q              67.874425      1110.4343
## ts_d1_star_quarterly             0.004746        -0.1275
## rest_exp_q_adj                  -0.127510        18.0752
## 
## Correlation matrix of residuals:
##                      ts_d1_quarterly   rec_q ts_rev_count_adj_q
## ts_d1_quarterly               1.0000 -0.3843            0.97326
## rec_q                        -0.3843  1.0000           -0.31026
## ts_rev_count_adj_q            0.9733 -0.3103            1.00000
## ts_d1_star_quarterly          0.3455 -0.1165            0.35182
## rest_exp_q_adj                0.0759 -0.1169            0.09327
##                      ts_d1_star_quarterly rest_exp_q_adj
## ts_d1_quarterly                    0.3455        0.07590
## rec_q                             -0.1165       -0.11693
## ts_rev_count_adj_q                 0.3518        0.09327
## ts_d1_star_quarterly               1.0000       -0.43534
## rest_exp_q_adj                    -0.4353        1.00000
```

```r
combinedFull2 = cbind(ts_d2_quarterly, rec_q, ts_rev_count_adj_q,
    ts_d2_star_quarterly, rest_exp_q_adj)
select = VARselect(combinedFull2, lag.max = 4, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
fullVar2 = VAR(combinedFull2, p = select$select[1])
plot(fullVar2$y)
```



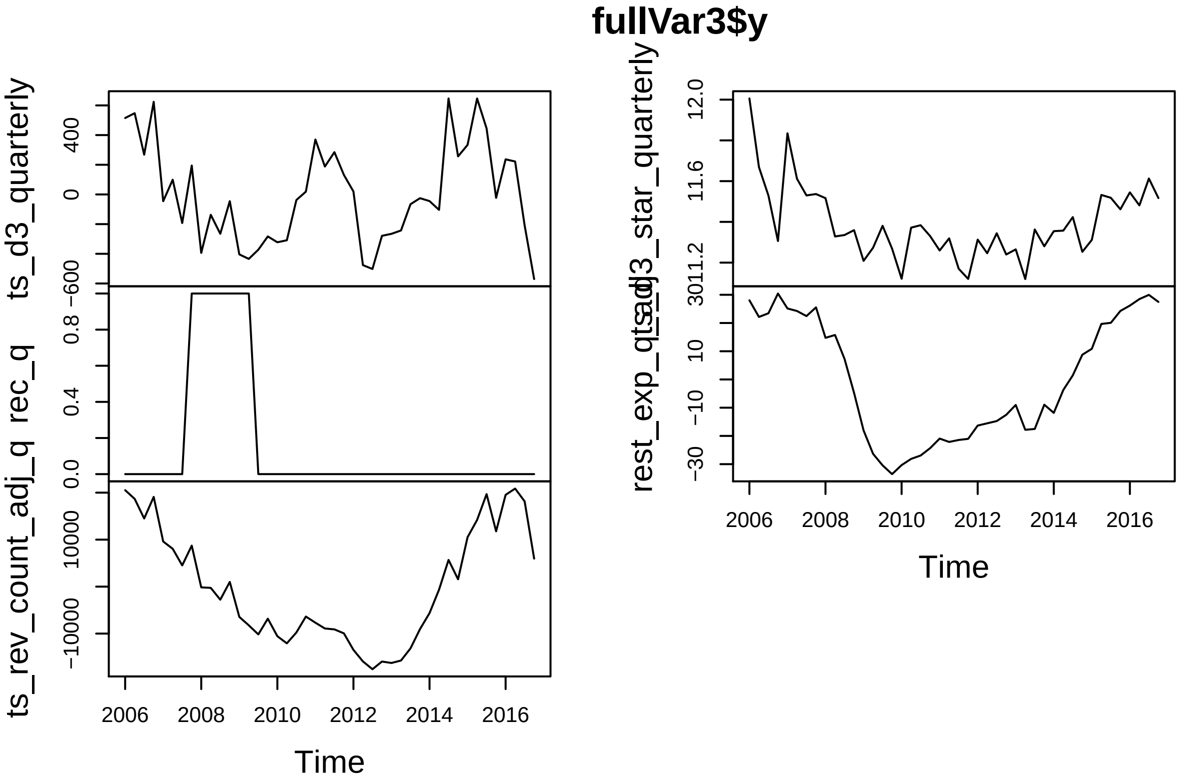
```r
summary(fullVar2)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: ts_d2_quarterly, rec_q, ts_rev_count_adj_q, ts_d2_star_quarterly, rest_exp_q_ad
## Deterministic variables: const
## Sample size: 40
## Log Likelihood: -591.043
## Roots of the characteristic polynomial:
## 1.032 1.032 0.9771 0.9771 0.9426 0.9426 0.9144 0.9066 0.9066 0.8902 0.8632 0.8632 0.8596 0.8596 0.74
## Call:
## VAR(y = combinedFull2, p = select$select[1])
##
##
## Estimation results for equation ts_d2_quarterly:
## ================================================
## ts_d2_quarterly = ts_d2_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d2_star_quarterly.l1 + r
##
##                          Estimate Std. Error t value Pr(>|t|)
## ts_d2_quarterly.l1       5.866e+00  1.926e+00   3.046  0.00664 **
## rec_q.l1                -1.835e+03  2.172e+03  -0.845  0.40860
## ts_rev_count_adj_q.l1   -3.220e+00  1.270e+00  -2.536  0.02016 *
## ts_d2_star_quarterly.l1 -3.106e+03  7.528e+03  -0.413  0.68453
## rest_exp_q_adj.l1        3.470e+02  1.159e+02   2.994  0.00747 **
## ts_d2_quarterly.l2      -7.587e+00  2.501e+00  -3.034  0.00683 **
## rec_q.l2                 2.477e+03  2.368e+03   1.046  0.30869
## ts_rev_count_adj_q.l2    5.279e+00  1.583e+00   3.335  0.00348 **
## ts_d2_star_quarterly.l2  3.096e+03  7.095e+03   0.436  0.66751
## rest_exp_q_adj.l2       -2.398e+02  1.226e+02  -1.956  0.06537 .
## ts_d2_quarterly.l3      -1.920e+00  3.062e+00  -0.627  0.53810
## rec_q.l3                -1.003e+02  2.543e+03  -0.039  0.96896
## ts_rev_count_adj_q.l3    8.223e-01  1.975e+00   0.416  0.68176
## ts_d2_star_quarterly.l3 -5.025e+03  5.855e+03  -0.858  0.40145
## rest_exp_q_adj.l3        4.949e+01  1.331e+02   0.372  0.71414
## ts_d2_quarterly.l4       7.118e+00  2.402e+00   2.963  0.00799 **
## rec_q.l4                -7.759e+02  2.670e+03  -0.291  0.77449
## ts_rev_count_adj_q.l4   -4.632e+00  1.601e+00  -2.894  0.00930 **
## ts_d2_star_quarterly.l4 -1.049e+04  6.487e+03  -1.618  0.12223
## rest_exp_q_adj.l4        1.223e+01  1.154e+02   0.106  0.91674
## const                    1.701e+05  1.448e+05   1.174  0.25480
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 2052 on 19 degrees of freedom
## Multiple R-Squared: 0.9645,  Adjusted R-squared: 0.9271
## F-statistic: 25.79 on 20 and 19 DF,  p-value: 8.543e-10
##
##
## Estimation results for equation rec_q:
## ======================================
## rec_q = ts_d2_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d2_star_quarterly.l1 + rest_exp_q
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d2_quarterly.l1       1.495e-04  1.457e-04   1.026  0.31765
## rec_q.l1                 4.687e-01  1.643e-01   2.853  0.01018 *
## ts_rev_count_adj_q.l1   -9.164e-05  9.608e-05  -0.954  0.35213
## ts_d2_star_quarterly.l1 -1.530e+00  5.694e-01  -2.686  0.01462 *
## rest_exp_q_adj.l1       -3.825e-03  8.768e-03  -0.436  0.66758
## ts_d2_quarterly.l2      -1.589e-05  1.892e-04  -0.084  0.93393
## rec_q.l2                -6.470e-02  1.791e-01  -0.361  0.72191
## ts_rev_count_adj_q.l2    2.719e-05  1.197e-04   0.227  0.82282
## ts_d2_star_quarterly.l2 -9.971e-01  5.367e-01  -1.858  0.07875 .
## rest_exp_q_adj.l2        3.266e-03  9.274e-03   0.352  0.72857
## ts_d2_quarterly.l3       3.026e-04  2.317e-04   1.306  0.20705
## rec_q.l3                -1.012e-02  1.924e-01  -0.053  0.95860
## ts_rev_count_adj_q.l3   -2.244e-04  1.494e-04  -1.502  0.14947
## ts_d2_star_quarterly.l3  1.385e+00  4.429e-01   3.127  0.00555 **
## rest_exp_q_adj.l3       -1.509e-03  1.007e-02  -0.150  0.88243
## ts_d2_quarterly.l4       1.522e-04  1.817e-04   0.838  0.41266
## rec_q.l4                -5.374e-01  2.020e-01  -2.661  0.01544 *
## ts_rev_count_adj_q.l4   -9.552e-05  1.211e-04  -0.789  0.43987
## ts_d2_star_quarterly.l4  1.614e-01  4.907e-01   0.329  0.74586
## rest_exp_q_adj.l4        8.811e-03  8.732e-03   1.009  0.32564
## const                    1.100e+01  1.096e+01   1.004  0.32805
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
```

```
## Residual standard error: 0.1552 on 19 degrees of freedom
## Multiple R-Squared: 0.9207,  Adjusted R-squared: 0.8372
## F-statistic: 11.03 on 20 and 19 DF,  p-value: 1.206e-06
##
##
## Estimation results for equation ts_rev_count_adj_q:
## ===================================================
## ts_rev_count_adj_q = ts_d2_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d2_star_quarterly.l1
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d2_quarterly.l1       8.178e+00  3.026e+00   2.702  0.01412 *
## rec_q.l1                -2.870e+03  3.414e+03  -0.841  0.41099
## ts_rev_count_adj_q.l1   -4.364e+00  1.996e+00  -2.186  0.04154 *
## ts_d2_star_quarterly.l1 -5.636e+03  1.183e+04  -0.476  0.63926
## rest_exp_q_adj.l1        4.980e+02  1.822e+02   2.734  0.01319 *
## ts_d2_quarterly.l2      -1.158e+01  3.931e+00  -2.947  0.00828 **
## rec_q.l2                 3.999e+03  3.721e+03   1.075  0.29602
## ts_rev_count_adj_q.l2    8.035e+00  2.488e+00   3.229  0.00441 **
## ts_d2_star_quarterly.l2  4.293e+03  1.115e+04   0.385  0.70455
## rest_exp_q_adj.l2       -3.583e+02  1.927e+02  -1.860  0.07847 .
## ts_d2_quarterly.l3      -2.231e+00  4.814e+00  -0.464  0.64823
## rec_q.l3                -4.709e+02  3.997e+03  -0.118  0.90746
## ts_rev_count_adj_q.l3    7.634e-01  3.104e+00   0.246  0.80835
## ts_d2_star_quarterly.l3 -7.344e+03  9.202e+03  -0.798  0.43469
## rest_exp_q_adj.l3        1.062e+02  2.092e+02   0.508  0.61762
## ts_d2_quarterly.l4       1.021e+01  3.776e+00   2.703  0.01409 *
## rec_q.l4                -1.596e+03  4.196e+03  -0.380  0.70792
## ts_rev_count_adj_q.l4   -6.600e+00  2.515e+00  -2.624  0.01672 *
## ts_d2_star_quarterly.l4 -1.575e+04  1.020e+04  -1.544  0.13897
## rest_exp_q_adj.l4       -9.019e+00  1.814e+02  -0.050  0.96087
## const                    2.677e+05  2.277e+05   1.176  0.25414
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 3226 on 19 degrees of freedom
## Multiple R-Squared: 0.9601,  Adjusted R-squared: 0.9181
## F-statistic: 22.86 on 20 and 19 DF,  p-value: 2.479e-09
##
##
## Estimation results for equation ts_d2_star_quarterly:
## =====================================================
## ts_d2_star_quarterly = ts_d2_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d2_star_quarterly.l
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d2_quarterly.l1       2.511e-04  5.336e-05   4.706 0.000154 ***
## rec_q.l1                -1.318e-01  6.019e-02  -2.190 0.041202 *
## ts_rev_count_adj_q.l1   -1.512e-04  3.520e-05  -4.296 0.000390 ***
## ts_d2_star_quarterly.l1  3.043e-01  2.086e-01   1.459 0.160954
## rest_exp_q_adj.l1       -2.271e-03  3.212e-03  -0.707 0.488183
## ts_d2_quarterly.l2      -1.416e-04  6.931e-05  -2.043 0.055191 .
## rec_q.l2                -1.136e-01  6.562e-02  -1.731 0.099602 .
## ts_rev_count_adj_q.l2    9.729e-05  4.387e-05   2.218 0.038960 *
## ts_d2_star_quarterly.l2 -1.106e-01  1.966e-01  -0.563 0.580260
## rest_exp_q_adj.l2       -5.119e-03  3.397e-03  -1.507 0.148340
## ts_d2_quarterly.l3       8.447e-05  8.487e-05   0.995 0.332151
## rec_q.l3                -5.572e-02  7.048e-02  -0.791 0.438943
## ts_rev_count_adj_q.l3   -5.479e-05  5.473e-05  -1.001 0.329327
## ts_d2_star_quarterly.l3 -1.873e-01  1.623e-01  -1.154 0.262702
## rest_exp_q_adj.l3        5.113e-03  3.689e-03   1.386 0.181722
## ts_d2_quarterly.l4      -1.665e-05  6.658e-05  -0.250 0.805216
## rec_q.l4                -1.116e-01  7.399e-02  -1.508 0.147899
## ts_rev_count_adj_q.l4   -8.959e-08  4.436e-05  -0.002 0.998409
## ts_d2_star_quarterly.l4  1.987e-01  1.798e-01   1.106 0.282726
## rest_exp_q_adj.l4        3.444e-03  3.199e-03   1.076 0.295201
## const                    8.803e+00  4.014e+00   2.193 0.040966 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.05687 on 19 degrees of freedom
## Multiple R-Squared: 0.9714,  Adjusted R-squared: 0.9414
## F-statistic: 32.31 on 20 and 19 DF,  p-value: 1.141e-10
##
##
## Estimation results for equation rest_exp_q_adj:
## ===============================================
## rest_exp_q_adj = ts_d2_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d2_star_quarterly.l1 + re
##
##                            Estimate Std. Error t value Pr(>|t|)
## ts_d2_quarterly.l1       5.163e-03  3.809e-03   1.356  0.19114
## rec_q.l1                -4.483e+00  4.296e+00  -1.043  0.30983
## ts_rev_count_adj_q.l1   -3.128e-03  2.512e-03  -1.245  0.22819
## ts_d2_star_quarterly.l1 -5.651e+00  1.489e+01  -0.380  0.70849
## rest_exp_q_adj.l1        7.596e-01  2.292e-01   3.313  0.00365 **
## ts_d2_quarterly.l2      -1.459e-03  4.947e-03  -0.295  0.77128
## rec_q.l2                 9.946e-01  4.683e+00   0.212  0.83407
## ts_rev_count_adj_q.l2    1.293e-03  3.131e-03   0.413  0.68430
## ts_d2_star_quarterly.l2 -7.451e+00  1.403e+01  -0.531  0.60158
## rest_exp_q_adj.l2        3.495e-02  2.425e-01   0.144  0.88690
## ts_d2_quarterly.l3      -2.842e-03  6.057e-03  -0.469  0.64430
## rec_q.l3                -5.639e+00  5.030e+00  -1.121  0.27628
## ts_rev_count_adj_q.l3    1.844e-03  3.906e-03   0.472  0.64221
## ts_d2_star_quarterly.l3  1.995e+01  1.158e+01   1.723  0.10116
## rest_exp_q_adj.l3       -1.055e-01  2.633e-01  -0.401  0.69312
## ts_d2_quarterly.l4      -3.932e-04  4.751e-03  -0.083  0.93491
## rec_q.l4                -5.516e+00  5.281e+00  -1.045  0.30931
## ts_rev_count_adj_q.l4    4.214e-05  3.166e-03   0.013  0.98952
## ts_d2_star_quarterly.l4 -2.054e+01  1.283e+01  -1.601  0.12583
## rest_exp_q_adj.l4        1.787e-01  2.283e-01   0.783  0.44354
## const                    1.527e+02  2.865e+02   0.533  0.60032
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 4.059 on 19 degrees of freedom
## Multiple R-Squared: 0.9817,  Adjusted R-squared: 0.9625
## F-statistic: 51.03 on 20 and 19 DF,  p-value: 1.784e-12
```

```
##
##
##
## Covariance matrix of residuals:
##                      ts_d2_quarterly     rec_q ts_rev_count_adj_q
## ts_d2_quarterly            4.212e+06 50.210696          6.590e+06
## rec_q                      5.021e+01  0.024100          7.471e+01
## ts_rev_count_adj_q         6.590e+06 74.712202          1.040e+07
## ts_d2_star_quarterly       4.282e+01 -0.003881          6.338e+01
## rest_exp_q_adj             3.502e+03 -0.057205          5.437e+03
##                      ts_d2_star_quarterly rest_exp_q_adj
## ts_d2_quarterly                 42.824546      3.502e+03
## rec_q                           -0.003881     -5.720e-02
## ts_rev_count_adj_q              63.384622      5.437e+03
## ts_d2_star_quarterly             0.003235      9.767e-03
## rest_exp_q_adj                   0.009767      1.648e+01
##
## Correlation matrix of residuals:
##                      ts_d2_quarterly    rec_q ts_rev_count_adj_q
## ts_d2_quarterly               1.0000  0.15760             0.9955
## rec_q                         0.1576  1.00000             0.1492
## ts_rev_count_adj_q            0.9955  0.14920             1.0000
## ts_d2_star_quarterly          0.3669 -0.43952             0.3455
## rest_exp_q_adj                0.4204 -0.09078             0.4153
##                      ts_d2_star_quarterly rest_exp_q_adj
## ts_d2_quarterly                   0.36690        0.42036
## rec_q                            -0.43952       -0.09078
## ts_rev_count_adj_q                0.34551        0.41526
## ts_d2_star_quarterly              1.00000        0.04231
## rest_exp_q_adj                    0.04231        1.00000
```

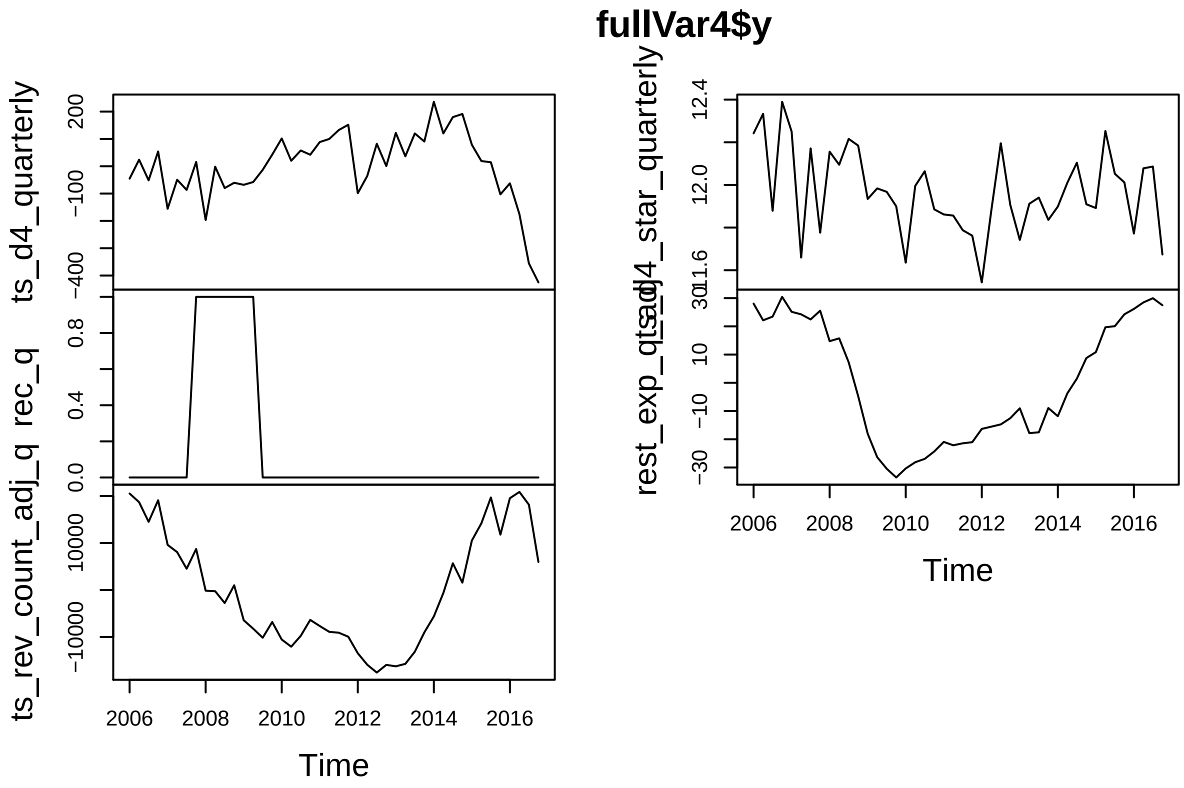
```r
combinedFull3 = cbind(ts_d3_quarterly, rec_q, ts_rev_count_adj_q,
    ts_d3_star_quarterly, rest_exp_q_adj)
select = VARselect(combinedFull3, lag.max = 4, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
fullVar3 = VAR(combinedFull3, p = select$select[1])
plot(fullVar3$y)
```



```r
summary(fullVar3)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: ts_d3_quarterly, rec_q, ts_rev_count_adj_q, ts_d3_star_quarterly, rest_exp_q_a
## Deterministic variables: const
## Sample size: 43
## Log Likelihood: -763.689
## Roots of the characteristic polynomial:
## 0.8855 0.8855 0.7869 0.4815 0.04915
## Call:
## VAR(y = combinedFull3, p = select$select[1])
##
##
## Estimation results for equation ts_d3_quarterly:
## ================================================
## ts_d3_quarterly = ts_d3_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d3_star_quarterly.l1 +
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d3_quarterly.l1       5.397e-01  1.883e-01   2.866  0.00682 **
## rec_q.l1                -1.817e+02  1.217e+02  -1.493  0.14392
## ts_rev_count_adj_q.l1   -1.043e-03  8.931e-03  -0.117  0.90762
## ts_d3_star_quarterly.l1  2.586e+02  3.446e+02   0.750  0.45772
## rest_exp_q_adj.l1       -7.448e-01  4.360e+00  -0.171  0.86531
## const                   -2.936e+03  3.931e+03  -0.747  0.45979
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 267.6 on 37 degrees of freedom
## Multiple R-Squared: 0.4141,  Adjusted R-squared: 0.3349
## F-statistic:  5.23 on 5 and 37 DF,  p-value: 0.0009912
##
##
## Estimation results for equation rec_q:
## ======================================
## rec_q = ts_d3_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d3_star_quarterly.l1 + rest_exp_q
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d3_quarterly.l1      -3.473e-05  1.428e-04  -0.243   0.8091
## rec_q.l1                 7.808e-01  9.229e-02   8.460 3.55e-10 ***
## ts_rev_count_adj_q.l1   -1.001e-05  6.771e-06  -1.478   0.1478
## ts_d3_star_quarterly.l1 -2.555e-02  2.613e-01  -0.098   0.9226
## rest_exp_q_adj.l1        8.216e-03  3.306e-03   2.485   0.0176 *
## const                    3.311e-01  2.980e+00   0.111   0.9121
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.2029 on 37 degrees of freedom
## Multiple R-Squared: 0.7402,  Adjusted R-squared: 0.7051
## F-statistic: 21.08 on 5 and 37 DF,  p-value: 6.405e-10
##
##
## Estimation results for equation ts_rev_count_adj_q:
## ===================================================
## ts_rev_count_adj_q = ts_d3_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d3_star_quarterly.l1
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d3_quarterly.l1         -0.7711     3.2127  -0.240   0.8116
## rec_q.l1                -3535.7649  2076.9402  -1.702   0.0971 .
## ts_rev_count_adj_q.l1       0.7922     0.1524   5.199 7.62e-06 ***
## ts_d3_star_quarterly.l1  -295.3277  5880.1673  -0.050   0.9602
## rest_exp_q_adj.l1          76.6078    74.3943   1.030   0.3098
## const                    3631.0451 67065.0351   0.054   0.9571
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 4565 on 37 degrees of freedom
## Multiple R-Squared: 0.8715,  Adjusted R-squared: 0.8541
## F-statistic: 50.19 on 5 and 37 DF,  p-value: 1.775e-15
##
##
## Estimation results for equation ts_d3_star_quarterly:
## =====================================================
## ts_d3_star_quarterly = ts_d3_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d3_star_quarterly.l
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d3_quarterly.l1      -7.828e-05  7.221e-05  -1.084   0.2854
## rec_q.l1                -6.431e-02  4.668e-02  -1.378   0.1766
## ts_rev_count_adj_q.l1    6.607e-06  3.425e-06   1.929   0.0614 .
## ts_d3_star_quarterly.l1 -2.741e-04  1.322e-01  -0.002   0.9984
## rest_exp_q_adj.l1        2.662e-03  1.672e-03   1.592   0.1199
## const                    1.140e+01  1.507e+00   7.560 5.15e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.1026 on 37 degrees of freedom
## Multiple R-Squared: 0.6154,  Adjusted R-squared: 0.5634
## F-statistic: 11.84 on 5 and 37 DF,  p-value: 7.024e-07
##
##
## Estimation results for equation rest_exp_q_adj:
## ===============================================
## rest_exp_q_adj = ts_d3_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d3_star_quarterly.l1 + re
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d3_quarterly.l1      -0.0011843  0.0029922  -0.396    0.695
## rec_q.l1                -9.6317060  1.9344305  -4.979 1.50e-05 ***
## ts_rev_count_adj_q.l1    0.0001337  0.0001419   0.942    0.352
## ts_d3_star_quarterly.l1 -4.0970909  5.4766986  -0.748    0.459
## rest_exp_q_adj.l1        0.9421511  0.0692897  13.597 5.76e-16 ***
## const                   48.2197370 62.4633550   0.772    0.445
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 4.252 on 37 degrees of freedom
## Multiple R-Squared: 0.9654,  Adjusted R-squared: 0.9608
## F-statistic: 206.7 on 5 and 37 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##                      ts_d3_quarterly      rec_q ts_rev_count_adj_q
## ts_d3_quarterly            71595.326  6.4368767          9.309e+05
## rec_q                          6.437  0.0411518          4.811e+01
## ts_rev_count_adj_q        930881.022 48.1078324          2.084e+07
## ts_d3_star_quarterly          -1.661 -0.0028005         -1.866e+01
## rest_exp_q_adj               441.690  0.0006136          6.659e+03
##                      ts_d3_star_quarterly rest_exp_q_adj
## ts_d3_quarterly                  -1.66131      4.417e+02
## rec_q                            -0.00280      6.136e-04
## ts_rev_count_adj_q              -18.66098      6.659e+03
## ts_d3_star_quarterly              0.01053     -6.391e-02
## rest_exp_q_adj                   -0.06391      1.808e+01
##
## Correlation matrix of residuals:
##                      ts_d3_quarterly      rec_q ts_rev_count_adj_q
## ts_d3_quarterly              1.00000  0.1185874            0.76207
## rec_q                        0.11859  1.0000000            0.05195
## ts_rev_count_adj_q           0.76207  0.0519472            1.00000
## ts_d3_star_quarterly        -0.06051 -0.1345389           -0.03984
## rest_exp_q_adj               0.38823  0.0007114            0.34308
##                      ts_d3_star_quarterly rest_exp_q_adj
## ts_d3_quarterly                  -0.06051      0.3882274
## rec_q                            -0.13454      0.0007114
## ts_rev_count_adj_q               -0.03984      0.3430760
## ts_d3_star_quarterly              1.00000     -0.1464891
## rest_exp_q_adj                   -0.14649      1.0000000
```

```
combinedFull4 = cbind(ts_d4_quarterly, rec_q, ts_rev_count_adj_q,
    ts_d4_star_quarterly, rest_exp_q_adj)
select = VARselect(combinedFull4, lag.max = 4, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
fullVar4 = VAR(combinedFull4, p = select$select[1])
plot(fullVar4$y)
```



```
summary(fullVar4)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: ts_d4_quarterly, rec_q, ts_rev_count_adj_q, ts_d4_star_quarterly, rest_exp_q_ad
## Deterministic variables: const
## Sample size: 43
## Log Likelihood: -752.05
## Roots of the characteristic polynomial:
```

```
## 0.912 0.912 0.7136 0.7136 0.2047
## Call:
## VAR(y = combinedFull4, p = select$select[1])
##
##
## Estimation results for equation ts_d4_quarterly:
## ================================================
## ts_d4_quarterly = ts_d4_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d4_star_quarterly.l1 + r
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d4_quarterly.l1       5.565e-01  1.441e-01   3.863 0.000436 ***
## rec_q.l1                -4.811e+01  4.026e+01  -1.195 0.239684
## ts_rev_count_adj_q.l1   -4.586e-03  2.473e-03  -1.854 0.071677 .
## ts_d4_star_quarterly.l1  1.110e+02  8.287e+01   1.340 0.188415
## rest_exp_q_adj.l1       -2.196e-01  1.435e+00  -0.153 0.879257
## const                   -1.327e+03  9.916e+02  -1.339 0.188873
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 89.85 on 37 degrees of freedom
## Multiple R-Squared: 0.5899,  Adjusted R-squared: 0.5344
## F-statistic: 10.64 on 5 and 37 DF,  p-value: 2.177e-06
##
##
## Estimation results for equation rec_q:
## ======================================
## rec_q = ts_d4_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d4_star_quarterly.l1 + rest_exp_q_
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d4_quarterly.l1       4.582e-06  3.234e-04   0.014   0.9888
## rec_q.l1                 7.781e-01  9.036e-02   8.611 2.29e-10 ***
## ts_rev_count_adj_q.l1   -1.168e-05  5.552e-06  -2.103   0.0423 *
## ts_d4_star_quarterly.l1  1.314e-01  1.860e-01   0.706   0.4844
## rest_exp_q_adj.l1        8.151e-03  3.222e-03   2.530   0.0158 *
## const                   -1.534e+00  2.226e+00  -0.689   0.4950
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.2017 on 37 degrees of freedom
## Multiple R-Squared: 0.7432,  Adjusted R-squared: 0.7085
## F-statistic: 21.42 on 5 and 37 DF,  p-value: 5.185e-10
##
##
## Estimation results for equation ts_rev_count_adj_q:
## ===================================================
## ts_rev_count_adj_q = ts_d4_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d4_star_quarterly.l1
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d4_quarterly.l1       1.023e+01  7.103e+00   1.440    0.158
## rec_q.l1                -2.720e+03  1.985e+03  -1.370    0.179
## ts_rev_count_adj_q.l1    7.646e-01  1.219e-01   6.270 2.71e-07 ***
## ts_d4_star_quarterly.l1  2.322e+03  4.086e+03   0.568    0.573
## rest_exp_q_adj.l1        9.852e+01  7.077e+01   1.392    0.172
## const                   -2.777e+04  4.889e+04  -0.568    0.573
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 4430 on 37 degrees of freedom
## Multiple R-Squared: 0.879,   Adjusted R-squared: 0.8627
## F-statistic: 53.77 on 5 and 37 DF,  p-value: 5.887e-16
##
##
## Estimation results for equation ts_d4_star_quarterly:
## =====================================================
## ts_d4_star_quarterly = ts_d4_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d4_star_quarterly.l
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d4_quarterly.l1       4.346e-04  2.809e-04   1.547   0.1303
## rec_q.l1                 1.744e-01  7.849e-02   2.222   0.0325 *
## ts_rev_count_adj_q.l1    2.520e-06  4.822e-06   0.523   0.6044
## ts_d4_star_quarterly.l1 -1.073e-01  1.616e-01  -0.664   0.5108
## rest_exp_q_adj.l1        3.546e-03  2.799e-03   1.267   0.2131
## const                    1.322e+01  1.933e+00   6.837 4.68e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.1752 on 37 degrees of freedom
## Multiple R-Squared: 0.2702,  Adjusted R-squared: 0.1715
## F-statistic: 2.739 on 5 and 37 DF,  p-value: 0.03337
##
##
## Estimation results for equation rest_exp_q_adj:
## ===============================================
## rest_exp_q_adj = ts_d4_quarterly.l1 + rec_q.l1 + ts_rev_count_adj_q.l1 + ts_d4_star_quarterly.l1 + re
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d4_quarterly.l1       0.0060711  0.0067898   0.894    0.377
## rec_q.l1                -8.6075302  1.8973409  -4.537 5.84e-05 ***
## ts_rev_count_adj_q.l1    0.0000934  0.0001166   0.801    0.428
## ts_d4_star_quarterly.l1 -1.3425796  3.9056055  -0.344    0.733
## rest_exp_q_adj.l1        0.9484601  0.0676524  14.020 2.22e-16 ***
## const                   17.3877254 46.7346891   0.372    0.712
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 4.234 on 37 degrees of freedom
## Multiple R-Squared: 0.9657,  Adjusted R-squared: 0.9611
## F-statistic: 208.4 on 5 and 37 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##                      ts_d4_quarterly     rec_q ts_rev_count_adj_q
## ts_d4_quarterly            8.073e+03  1.223753          1.755e+05
## rec_q                      1.224e+00  0.040672          3.994e+01
## ts_rev_count_adj_q         1.755e+05 39.938485          1.962e+07
## ts_d4_star_quarterly       3.323e+00 -0.005123          1.579e+02
## rest_exp_q_adj             8.474e+01  0.009275          6.156e+03
##                      ts_d4_star_quarterly rest_exp_q_adj
## ts_d4_quarterly                  3.323017      8.474e+01
## rec_q                           -0.005123      9.275e-03
## ts_rev_count_adj_q             157.891513      6.156e+03
## ts_d4_star_quarterly             0.030685     -1.964e-02
## rest_exp_q_adj                  -0.019635      1.793e+01
##
## Correlation matrix of residuals:
##                      ts_d4_quarterly    rec_q ts_rev_count_adj_q
## ts_d4_quarterly              1.00000  0.06754            0.44106
## rec_q                        0.06754  1.00000            0.04471
## ts_rev_count_adj_q           0.44106  0.04471            1.00000
## ts_d4_star_quarterly         0.21113 -0.14503            0.20349
## rest_exp_q_adj               0.22274  0.01086            0.32818
##                      ts_d4_star_quarterly rest_exp_q_adj
## ts_d4_quarterly                   0.21113        0.22274
## rec_q                            -0.14503        0.01086
## ts_rev_count_adj_q                0.20349        0.32818
## ts_d4_star_quarterly              1.00000       -0.02647
## rest_exp_q_adj                   -0.02647        1.00000
```

old stuff

sources: fred yelp https://www.bea.gov/iTable/iTable.cfm?reqid=9&step=1&acrdn=2#reqid=9&step=1&isuri=1&904=2004&903=64&906=q&905=2016&910=x&911=0 https://cran.r-project.org/web/packages/tidytext/vignettes/tidying__casting.html