Yelpers in The Great Recession

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

Introduction

Hypothesis: yelpers

Data Sets

yelp fred yahoo finance bea

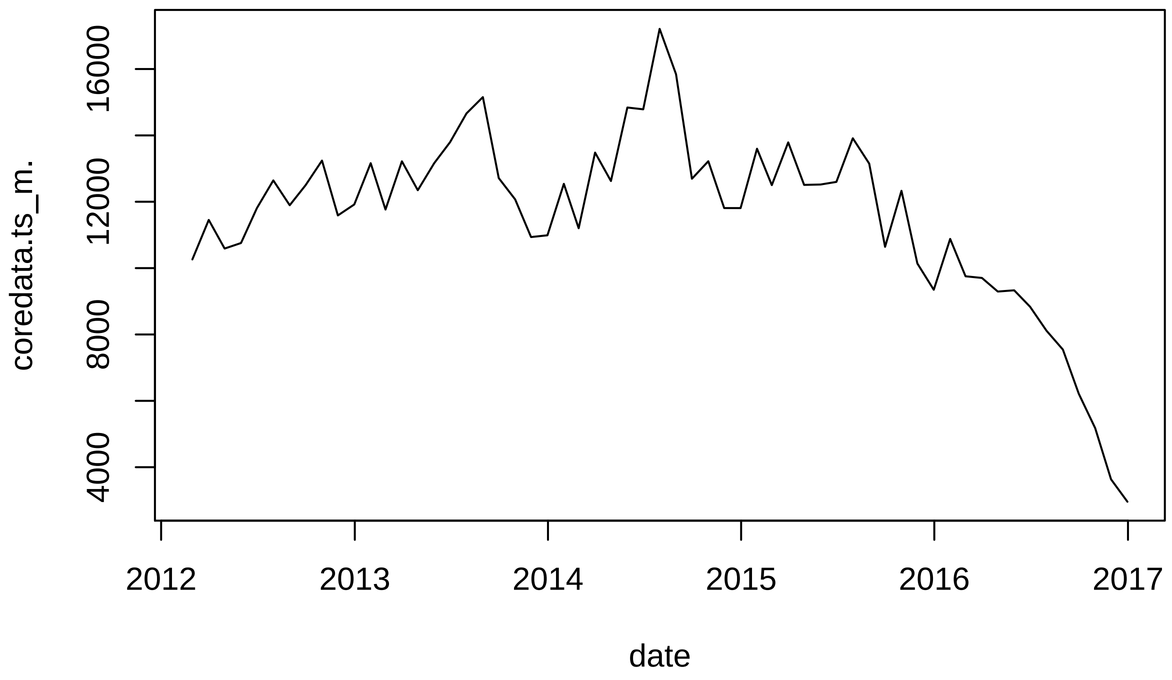
Motivation

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

```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

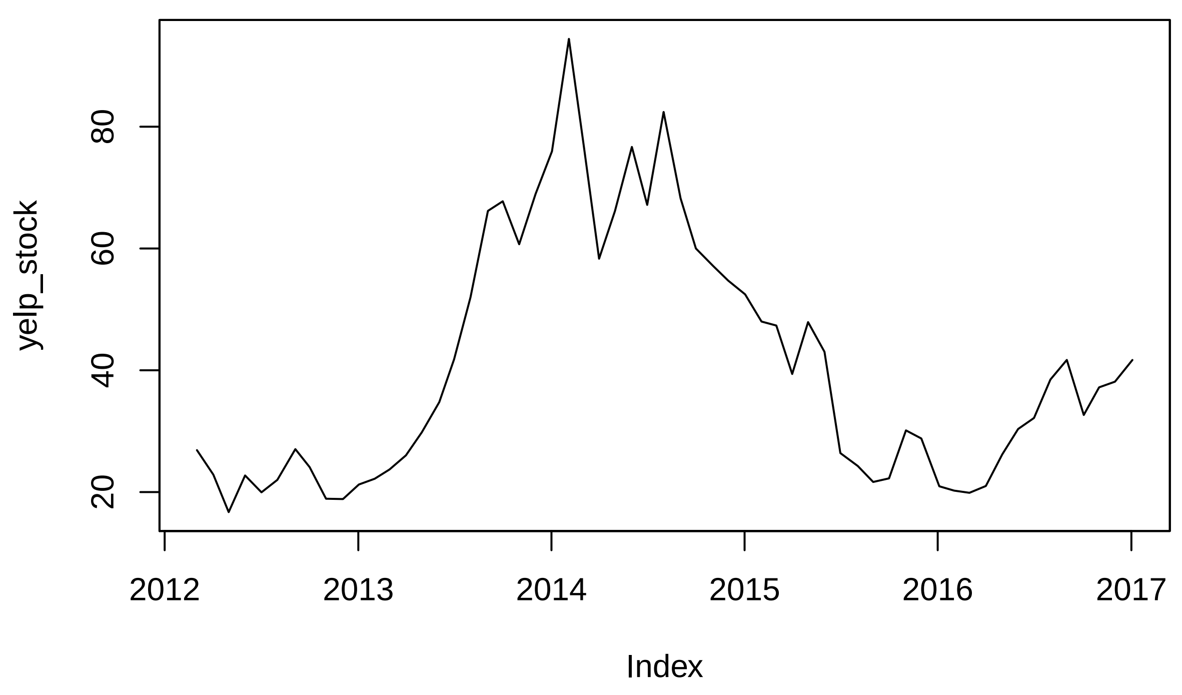
```
plot(stock_users, type = "l")
```



```r
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
```

```r
# yelp_stock

plot(yelp_stock)
```



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

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
```

```
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
```

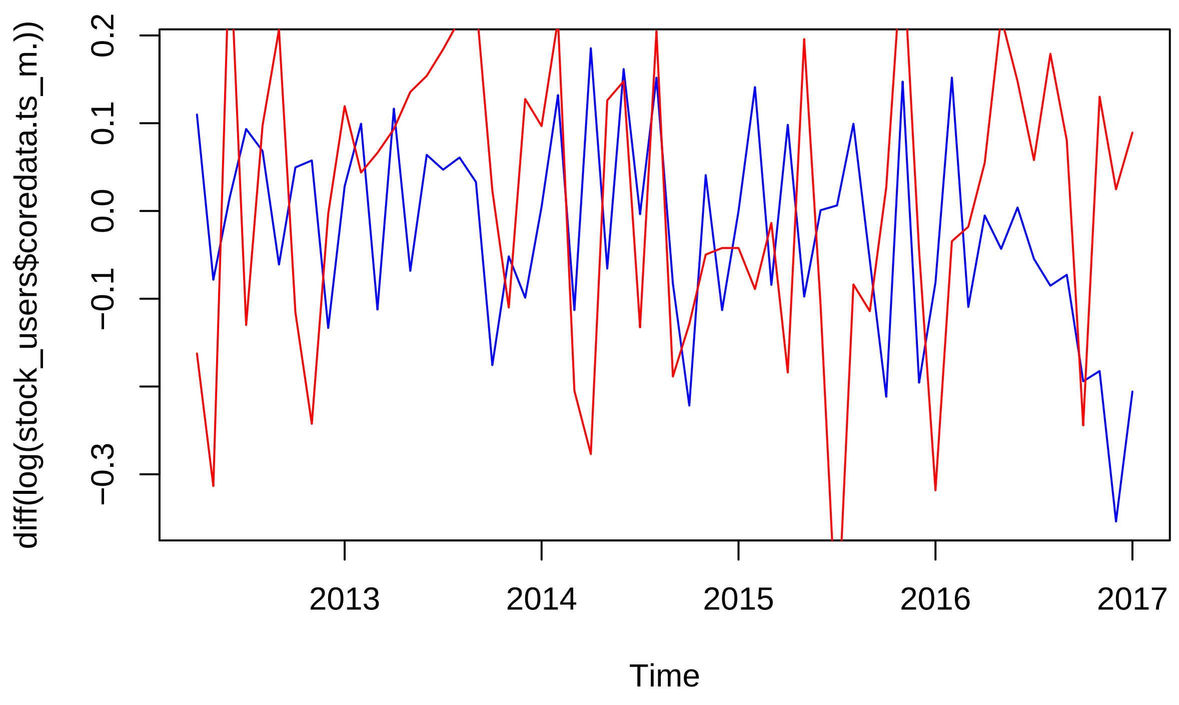
```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 0.8162, Truncation lag parameter = 1, p-value = 0.01
##
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -2.5657, Lag order = 3, p-value = 0.3463
## alternative hypothesis: stationary
```

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

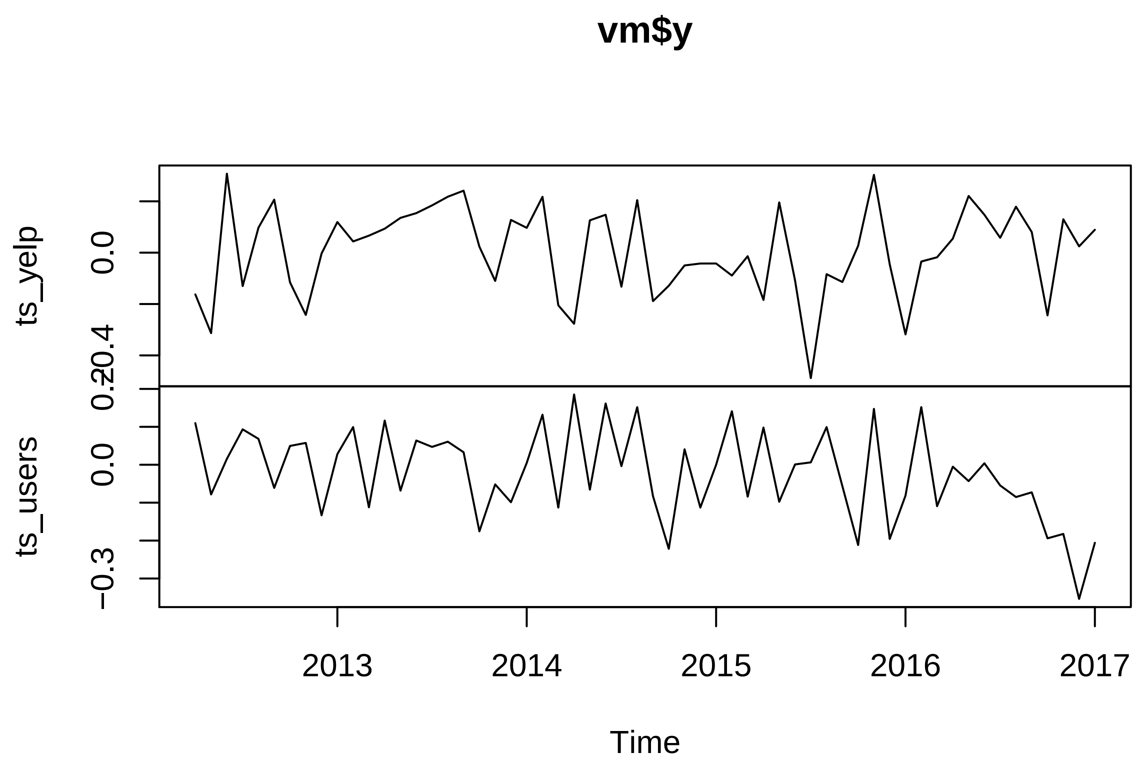
```
## Warning in kpss.test(t): p-value greater than printed p-value
```

```
##
##  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
plot(ts_users, col = "blue")
lines(ts_yelp, col = "red")
```



```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
```

```
sp500 = get.hist.quote("^GSPC", quote = "AdjClose", compression = "m",
    start = as.Date("2012-03-01", "%Y-%m-%d"), end = as.Date("2017-01-30",
        "%Y-%m-%d"))
```

```
## time series ends   2017-01-03
```

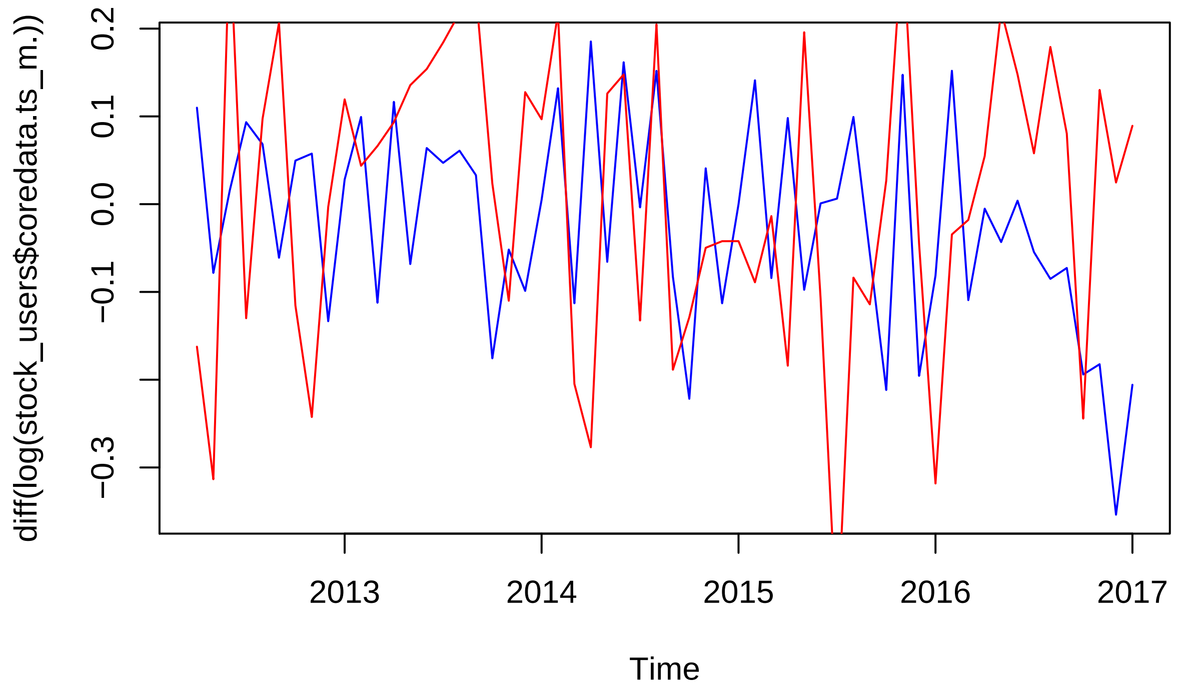
```
# sp500
test_stationary(sp500)
```

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

##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 2.7456, Truncation lag parameter = 1, p-value = 0.01
##
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -1.3383, Lag order = 3, p-value = 0.842
## alternative hypothesis: stationary
```

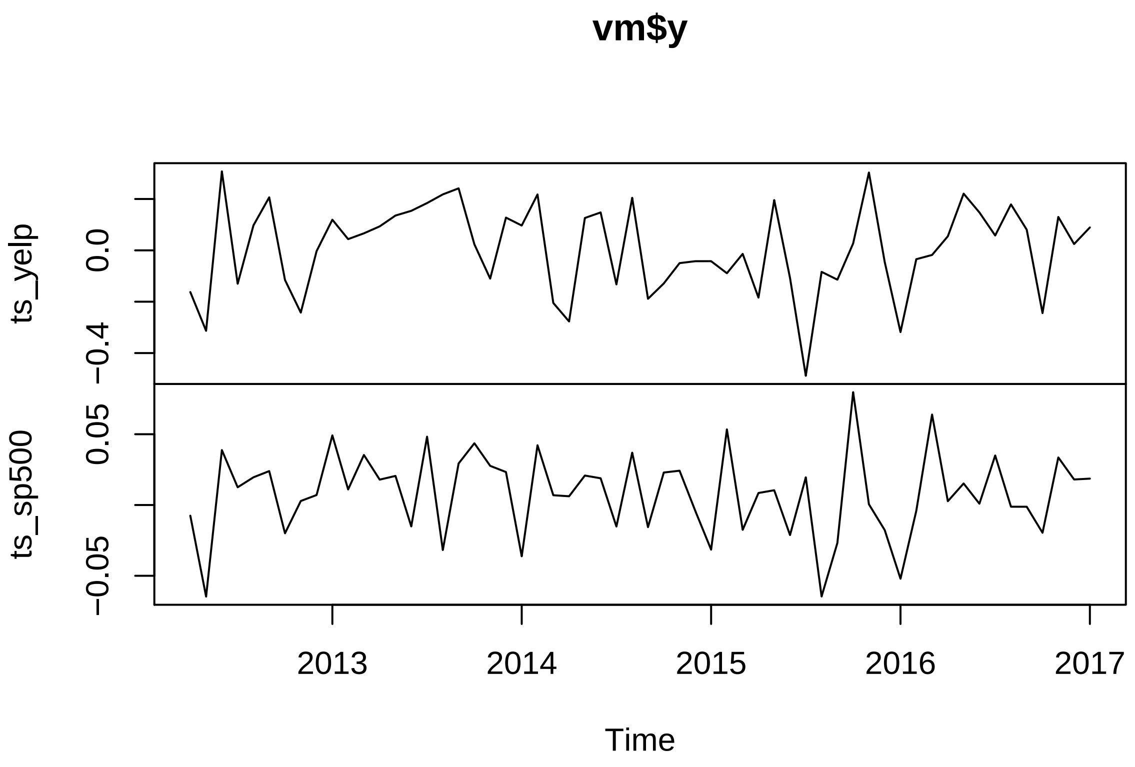
```
log_sp500_growth = as.data.frame(diff(log(sp500)))
ts_sp500 = ts(log_sp500_growth, start = c(2012, 4), freq = 12)
# ts_sp500
test_stationary(ts_sp500)
```

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

##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 0.096794, Truncation lag parameter = 1, p-value = 0.1

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

##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -5.2647, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
# ts_yelp ts_sp500


plot(ts_users, col = "blue")
lines(ts_yelp, col = "red")
```



```r
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)
```



```r
summary(vm)
```

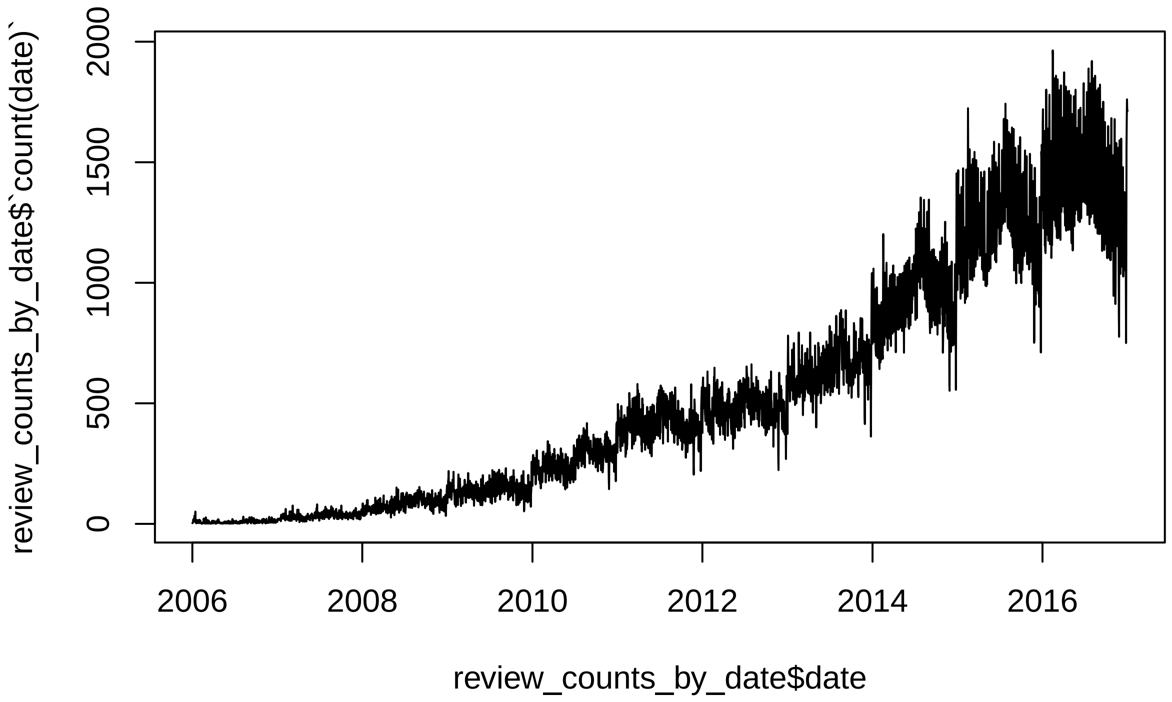
```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: ts_yelp, ts_sp500
## Deterministic variables: const
## Sample size: 57
## Log Likelihood: 146.967
## Roots of the characteristic polynomial:
## 0.261 0.09083
## Call:
## VAR(y = combined, p = select$select[1])
##
##
## Estimation results for equation ts_yelp:
## ========================================
## ts_yelp = ts_yelp.l1 + ts_sp500.l1 + const
##
##             Estimate Std. Error t value Pr(>|t|)
## ts_yelp.l1  0.062844   0.144375   0.435    0.665
## ts_sp500.l1 0.524613   0.838397   0.626    0.534
## const       0.005891   0.023856   0.247    0.806
##
##
## Residual standard error: 0.1732 on 54 degrees of freedom
## Multiple R-Squared: 0.01655, Adjusted R-squared: -0.01987
## F-statistic: 0.4545 on 2 and 54 DF,  p-value: 0.6372
##
##
## Estimation results for equation ts_sp500:
## =========================================
## ts_sp500 = ts_yelp.l1 + ts_sp500.l1 + const
##
##              Estimate Std. Error t value Pr(>|t|)
## ts_yelp.l1   0.017279   0.024603   0.702   0.4855
## ts_sp500.l1 -0.233048   0.142871  -1.631   0.1087
## const        0.010378   0.004065   2.553   0.0135 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.02952 on 54 degrees of freedom
## Multiple R-Squared: 0.04714, Adjusted R-squared: 0.01185
## F-statistic: 1.336 on 2 and 54 DF,  p-value: 0.2715
##
##
##
## Covariance matrix of residuals:
##           ts_yelp  ts_sp500
## ts_yelp  0.030002 0.0020338
## ts_sp500 0.002034 0.0008712
##
## Correlation matrix of residuals:
##          ts_yelp ts_sp500
## ts_yelp   1.0000   0.3978
## ts_sp500  0.3978   1.0000
```

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.

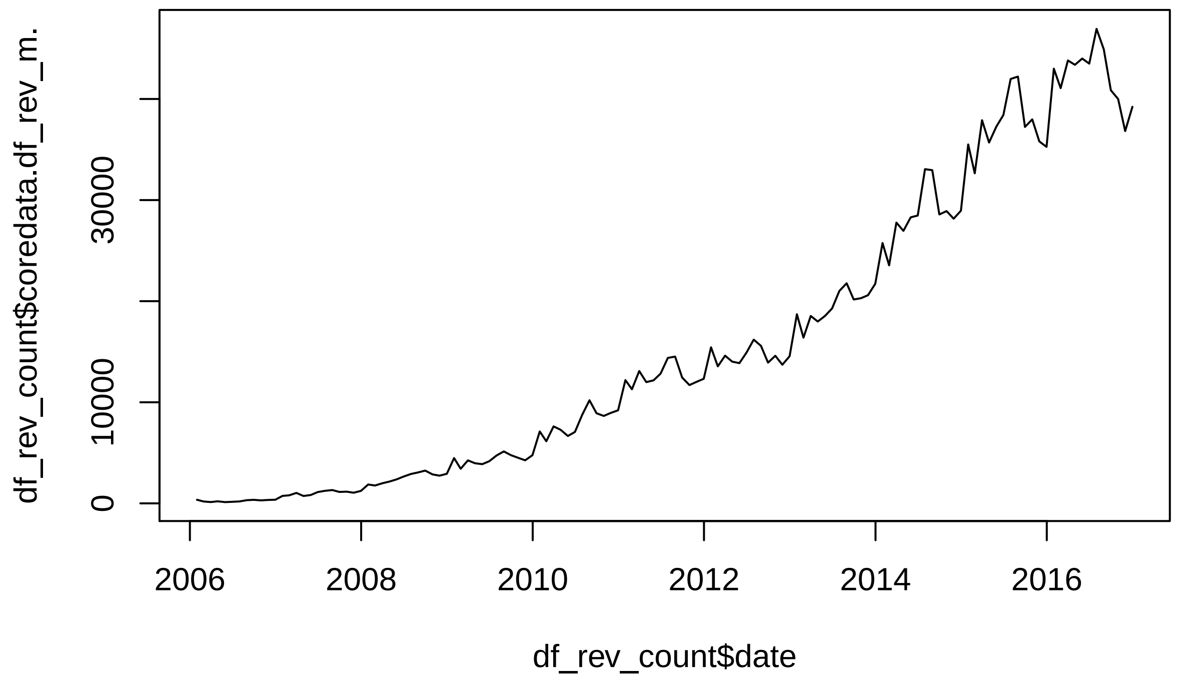
should i examine review scores + users + etc and its relation to s and p500 during recession? or just move on to gdp and recession

review scores and gdp

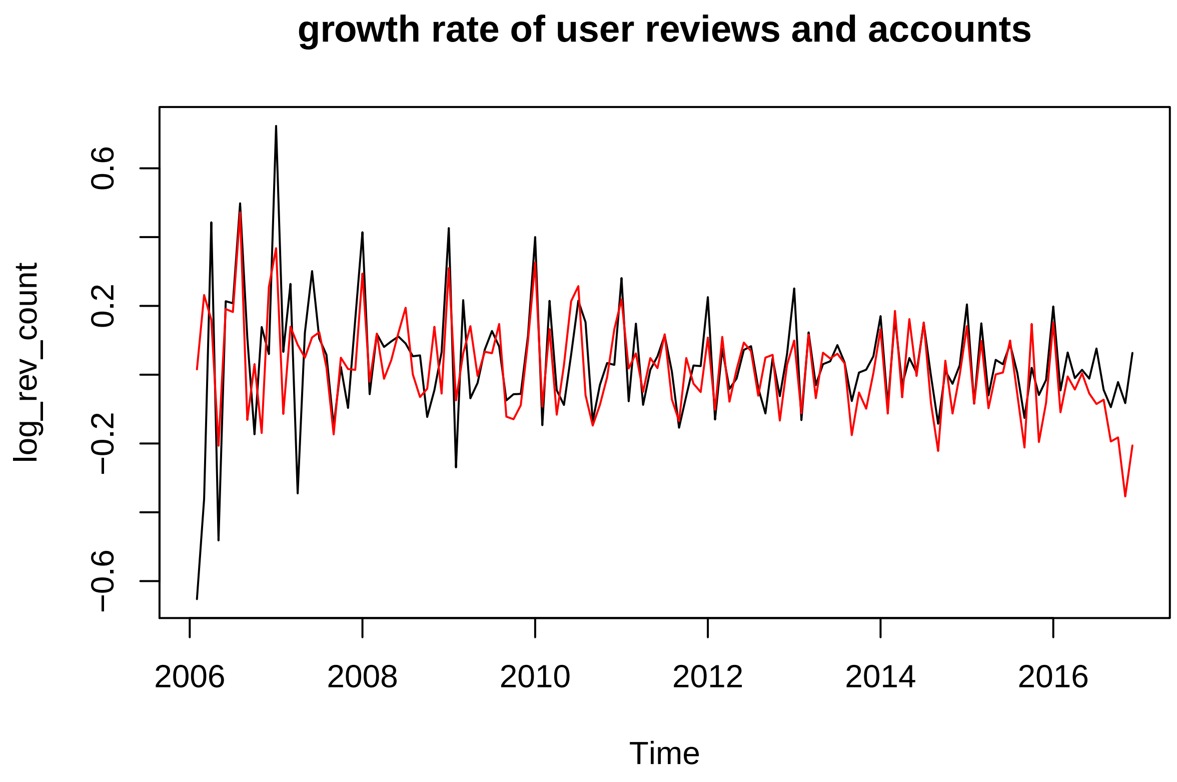
```
plot(review_counts_by_date$date, review_counts_by_date$`count(date)`,
    type = "l")
```



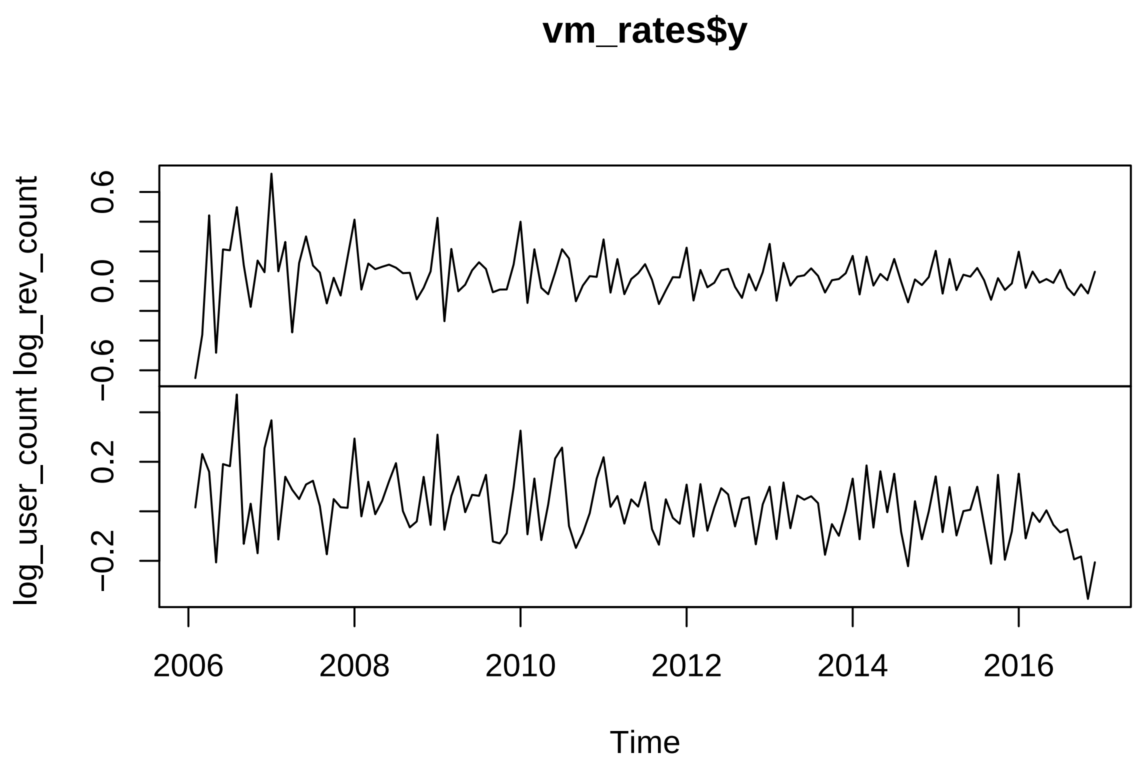
```r
# convert to monthly
reviews_by_date = xts(review_counts_by_date$`count(date)`, as.Date(review_counts_by_date$date,
    "%Y-%m-%d"))
df_rev_m = apply.monthly(reviews_by_date, sum)
df_rev_count = data.frame(date = index(df_rev_m), coredata(df_rev_m))
# df_rev_count

plot(df_rev_count$date, df_rev_count$coredata.df_rev_m., type = "l")
```



```r
# 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")
```



```r
# 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_re
##
##                     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_re
##
##                     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])
```

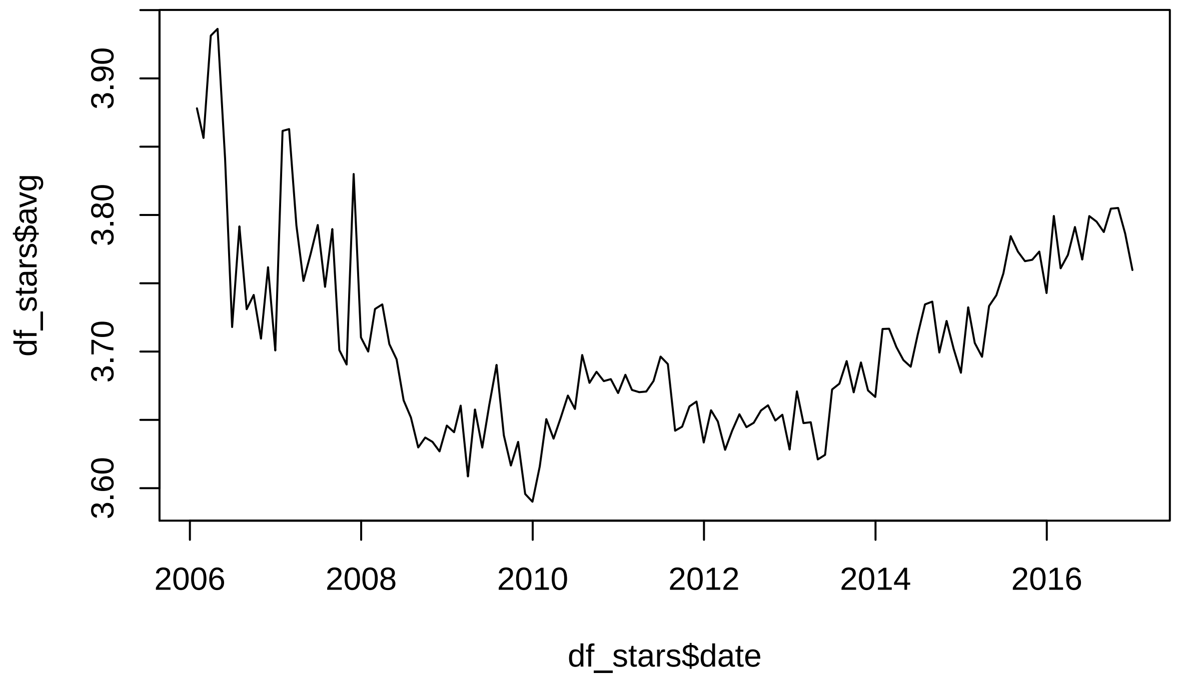
```
## Granger causality test
##
## Model 1: log_user_count ~ Lags(log_user_count, 1:12) + Lags(log_rev_count, 1:12)
## Model 2: log_user_count ~ Lags(log_user_count, 1:12)
##   Res.Df  Df      F   Pr(>F)
## 1     94
## 2    106 -12 3.4167 0.000354 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01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

```r
# users and review#s granger cause each other, 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")
```



```r
# 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)
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
```

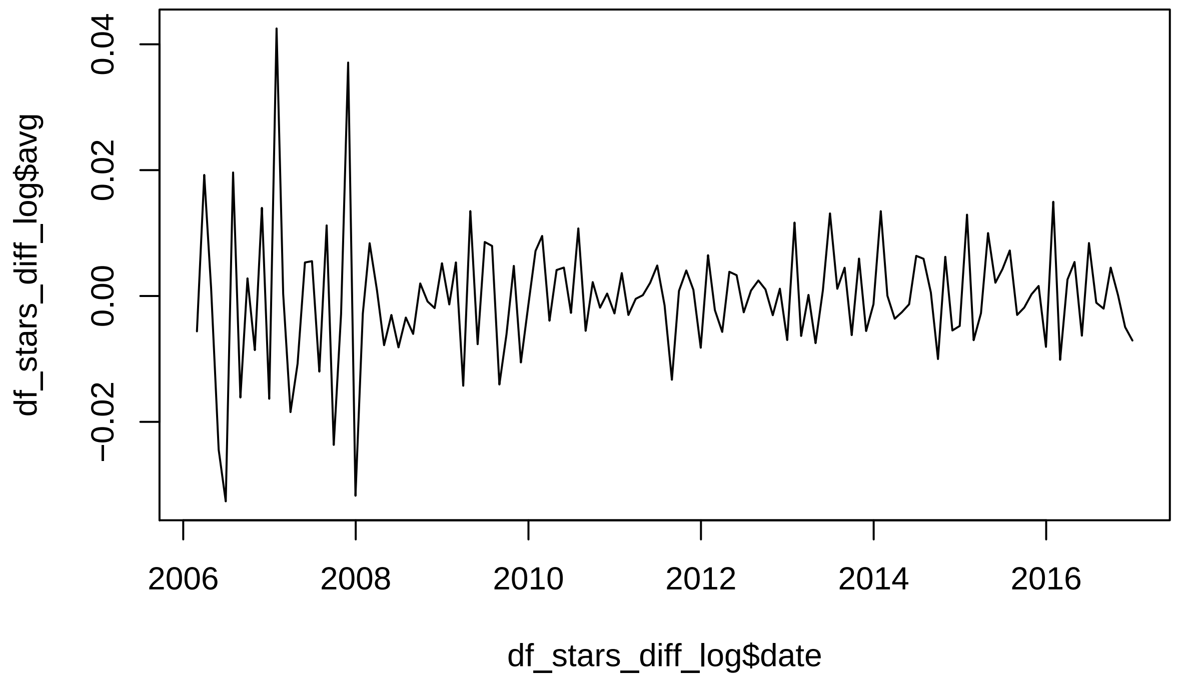
```
##
##  KPSS Test for Level Stationarity
```

```
##
## 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
```

```
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")
```



```
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)
```

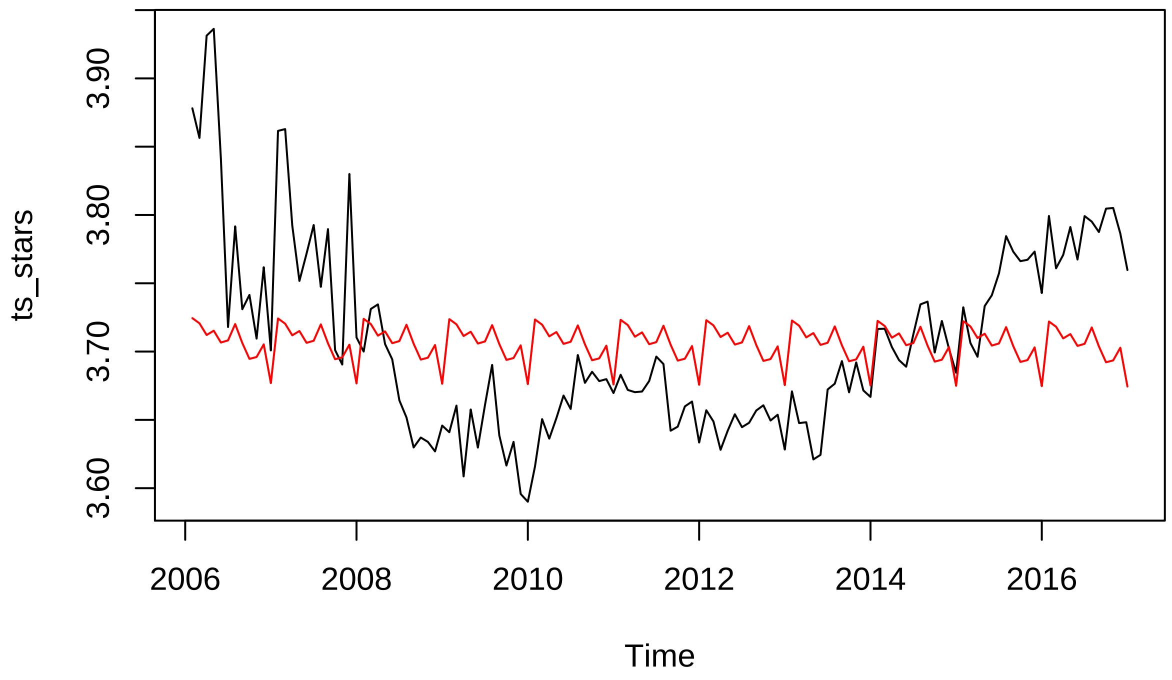
```
##
## Call:
## lm(formula = stars_diff_log_avg ~ stars_recession_diff_log_dummy)
##
```

```
## 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
```

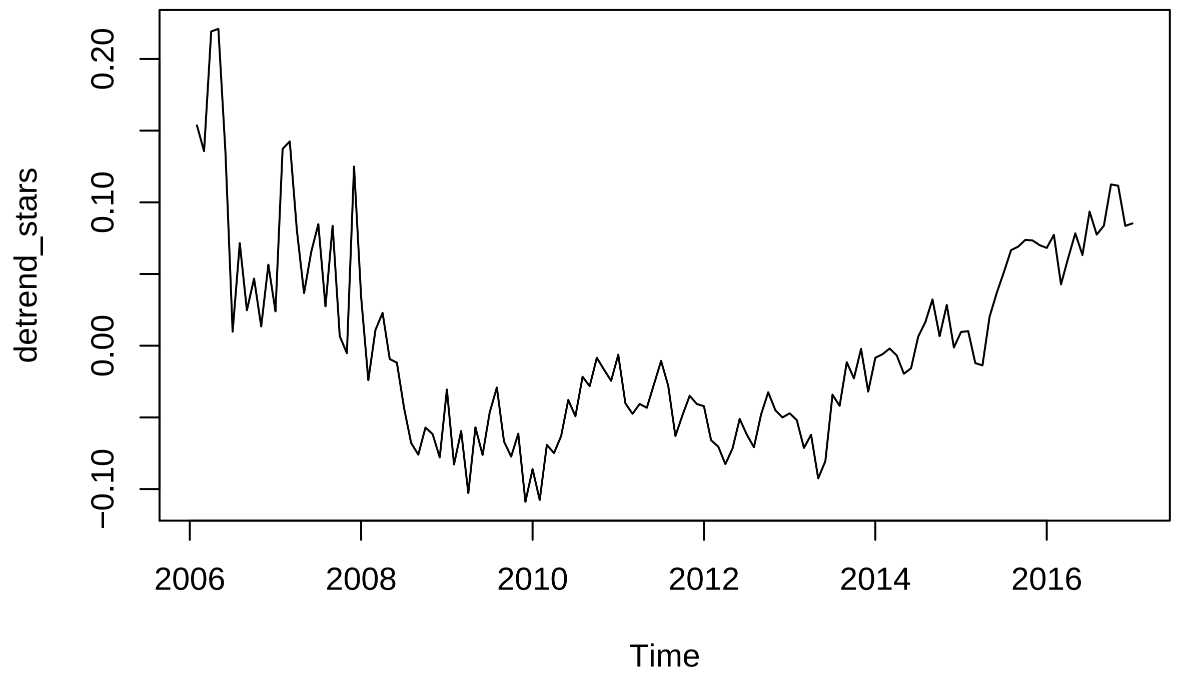
```
# however, intuitively we should be looking at level, not
# growth rates so lets detrend the data and season

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)
```



```
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
```

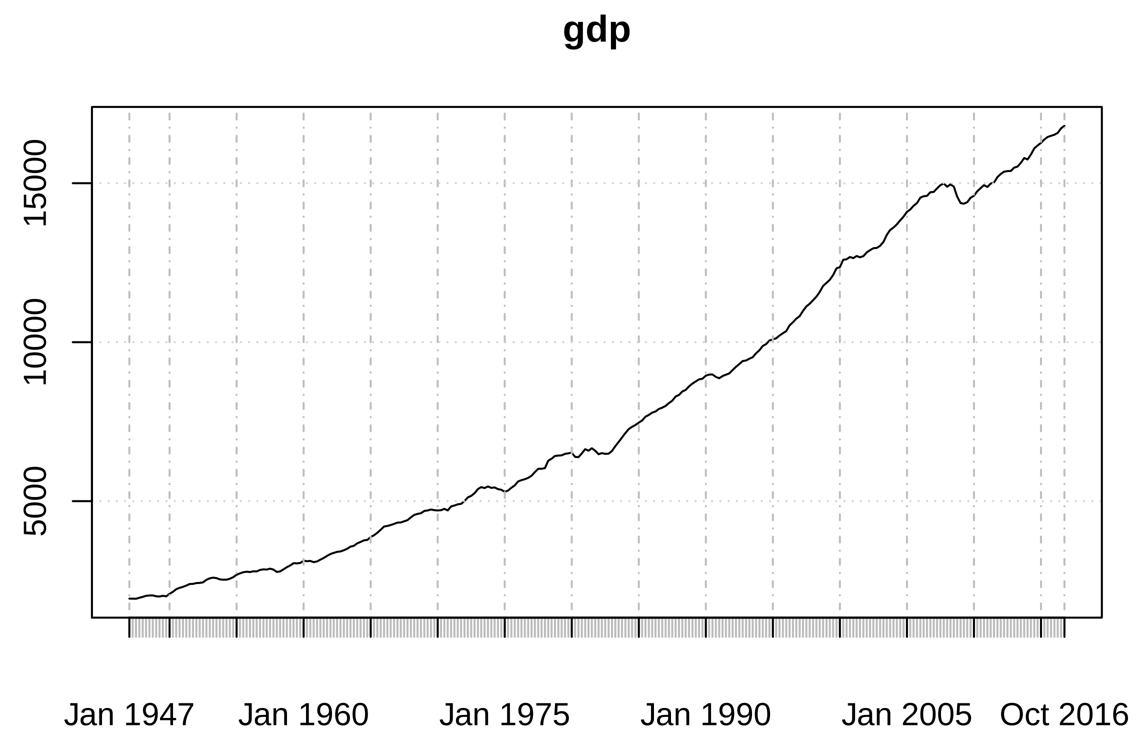
fred data

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

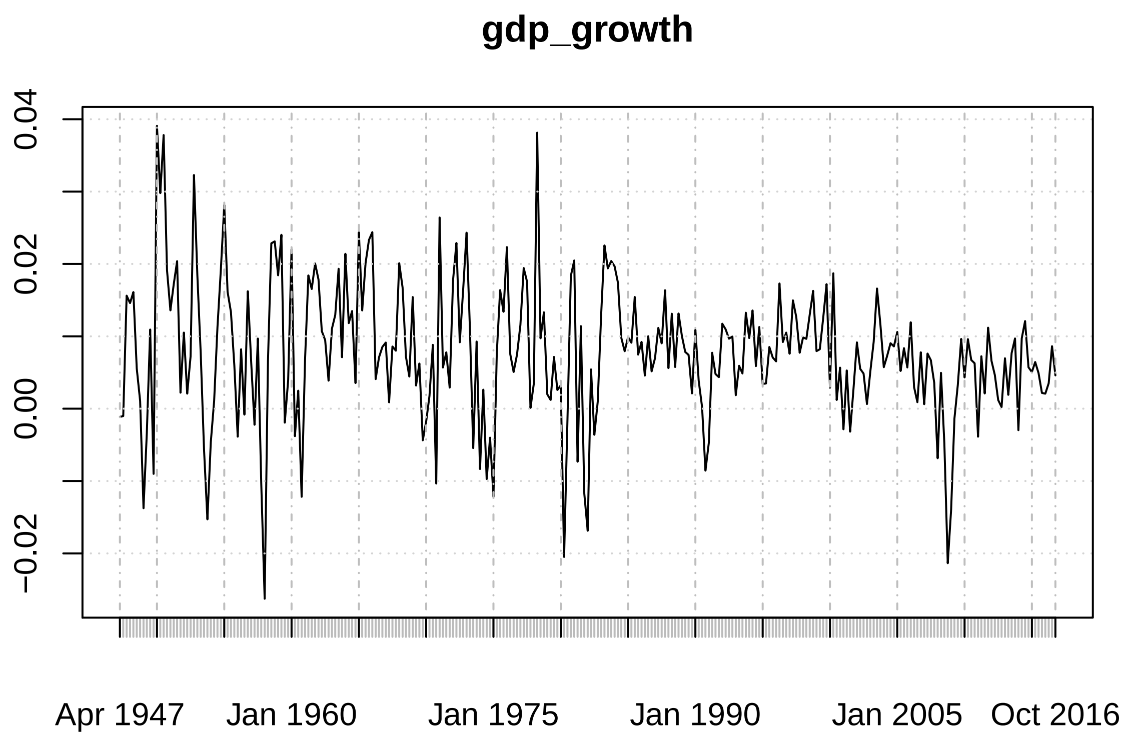
```
##     As of 0.4-0, 'getSymbols' uses env=parent.frame() and
##  auto.assign=TRUE by default.
##
##  This  behavior  will be  phased out in 0.5-0  when the call  will
##  default to use auto.assign=FALSE. getOption("getSymbols.env") and
##  getOptions("getSymbols.auto.assign") are now checked for 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"
```

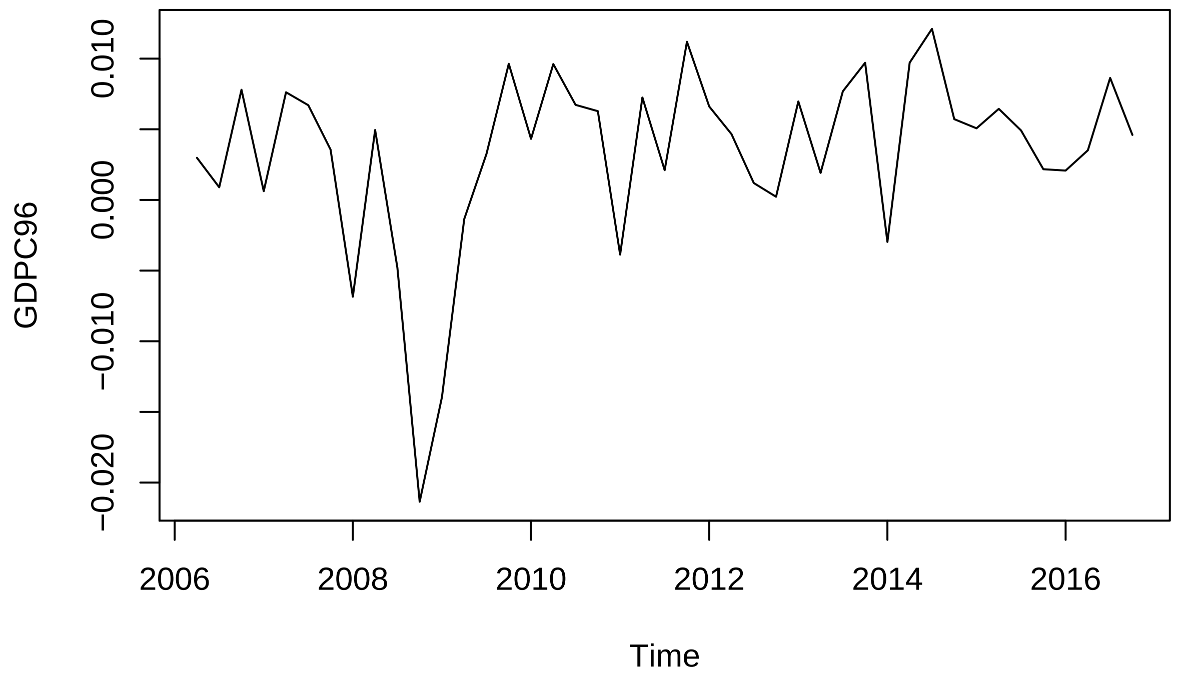
```
gdp = GDPC96

plot(gdp)
```



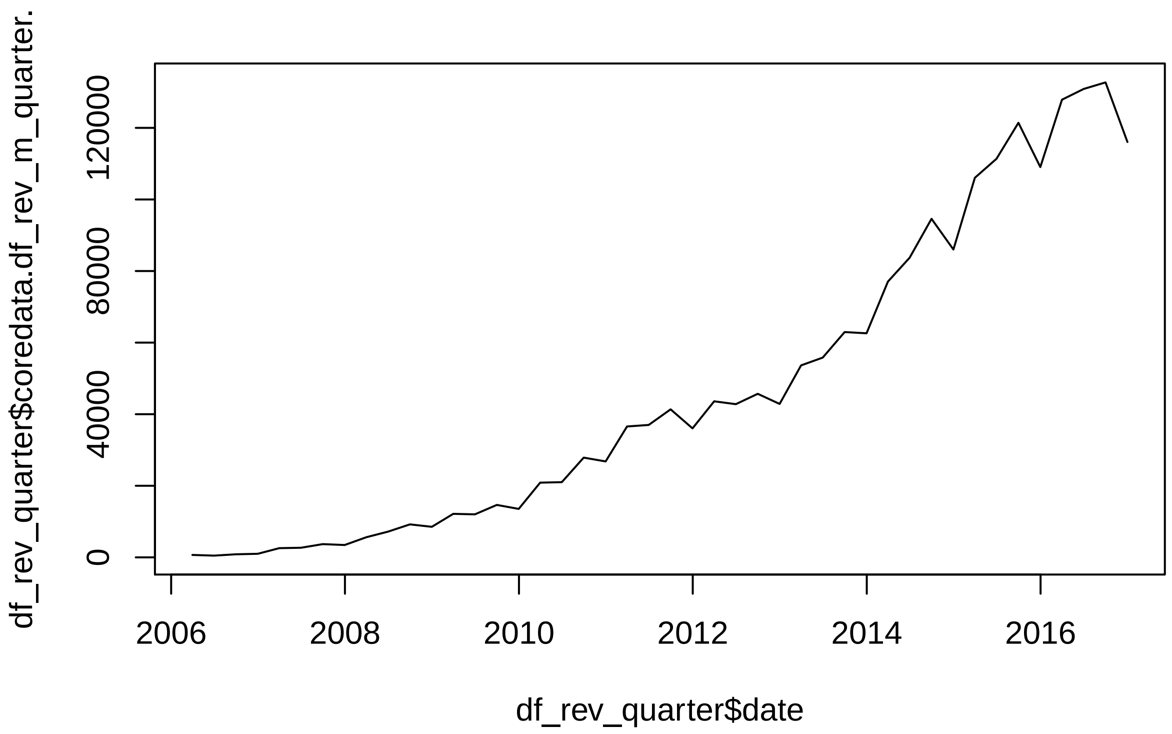
```
gdp_growth = na.omit(diff(log(gdp)))
plot(gdp_growth)
```



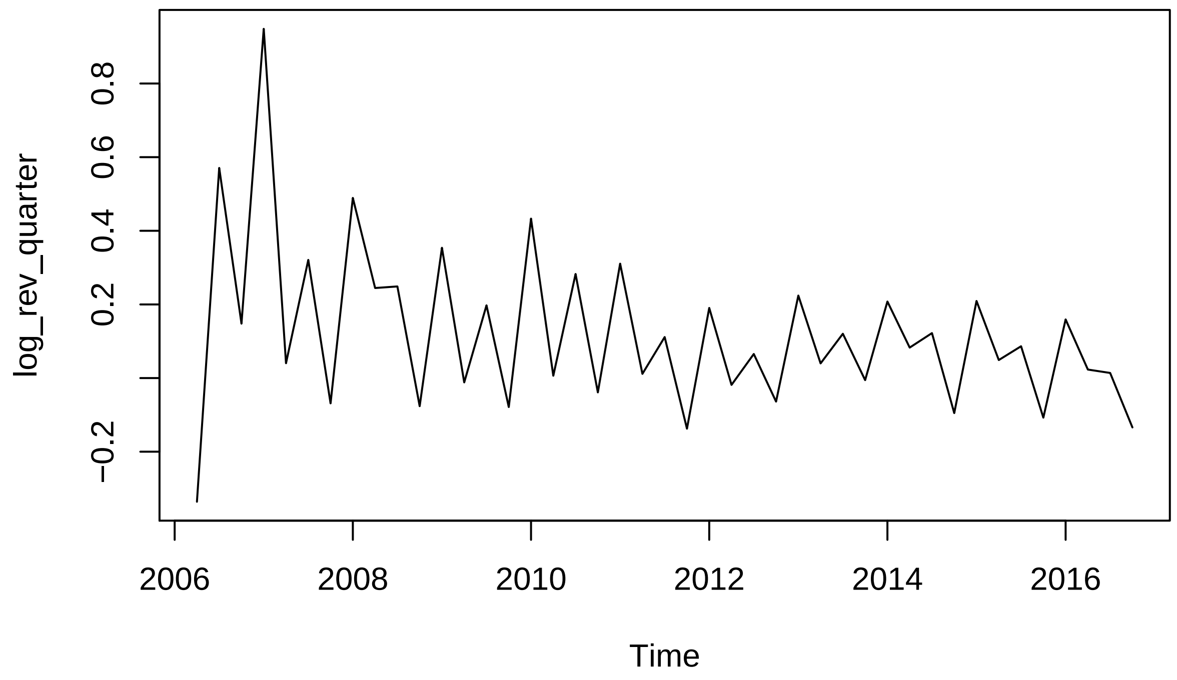
```
gdp_growth_subset = with(gdp_growth, gdp_growth[index(gdp_growth) >=
    "2006-04-01" & index(gdp_growth) < "2016-12-30", ])
gdp_growth_subset = ts(gdp_growth_subset, start = c(2006, 2),
    frequency = 4)
plot(gdp_growth_subset)
```



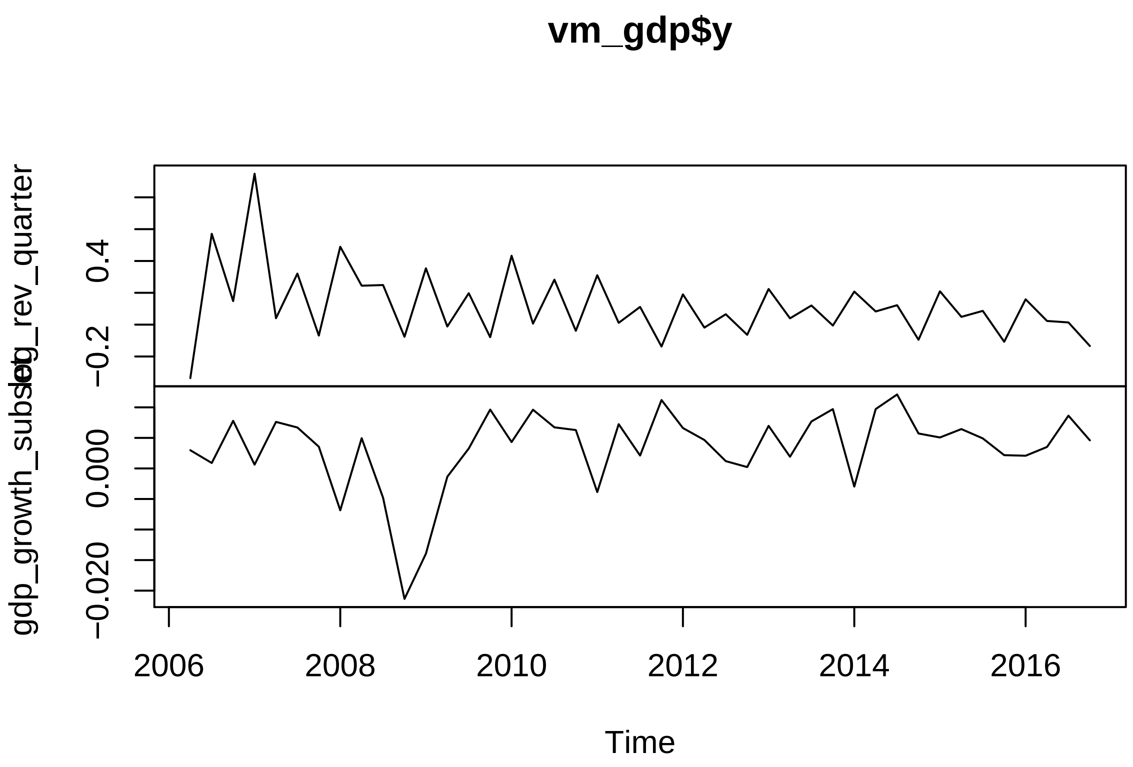
```r
# create quarterly review growth rate
reviews_by_quarter = xts(review_counts_by_date$`count(date)`,
    as.Date(review_counts_by_date$date, "%Y-%m-%d"))
df_rev_m_quarter = apply.quarterly(reviews_by_quarter, sum)
df_rev_quarter = data.frame(date = index(df_rev_m_quarter), coredata(df_rev_m_quarter))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



```
# 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)
```



```r
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.
## 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
```

```r
# 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
```

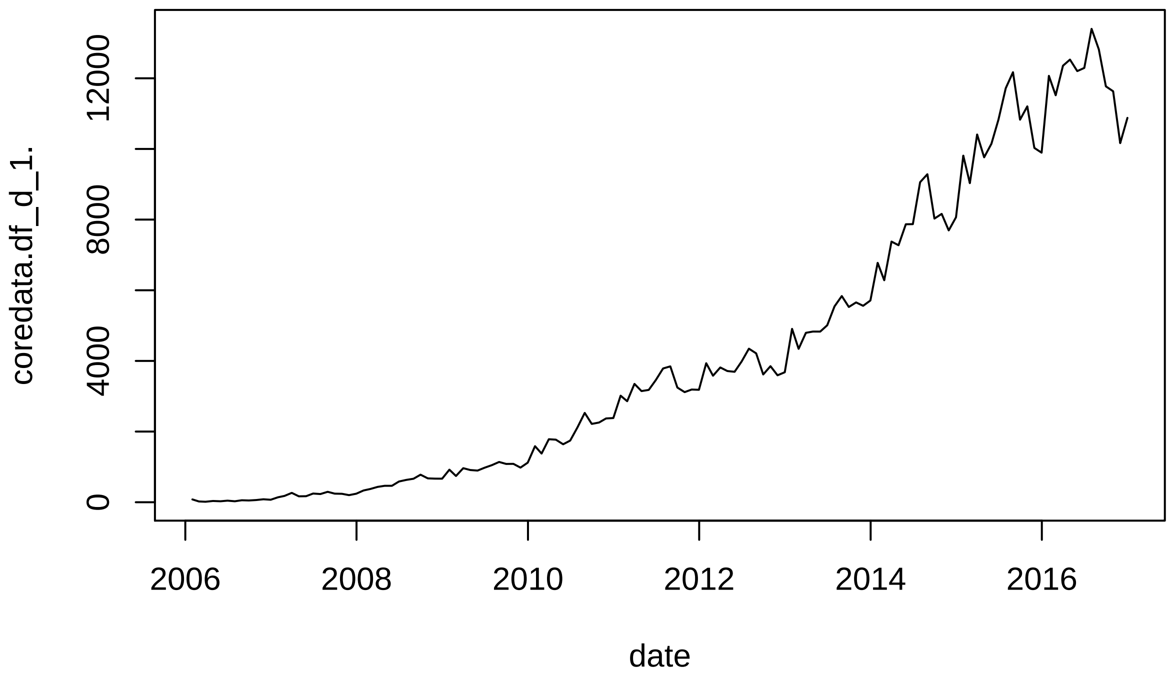
```r
# use lm
log_reviews_lm = lm(log_rev_quarter ~ rec_dummy_rev_growth_q)
summary(log_reviews_lm)
```

```
##
## 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
```

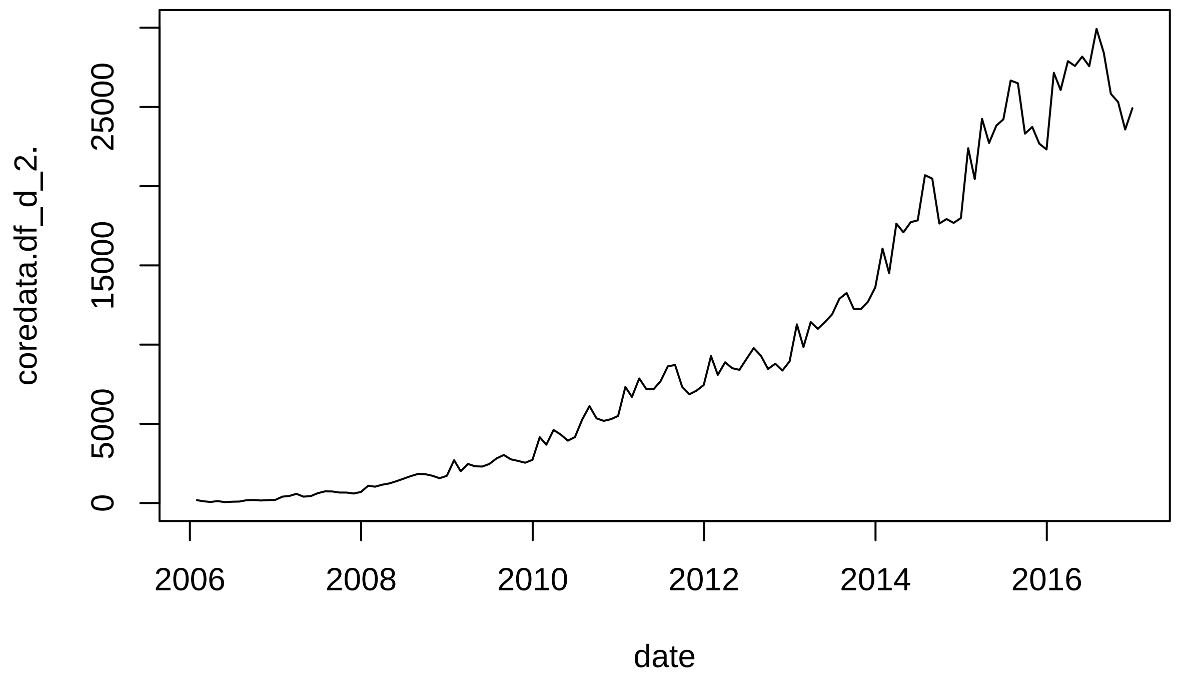
examine social dynamics during recessions through dollar signs as well as review texts

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

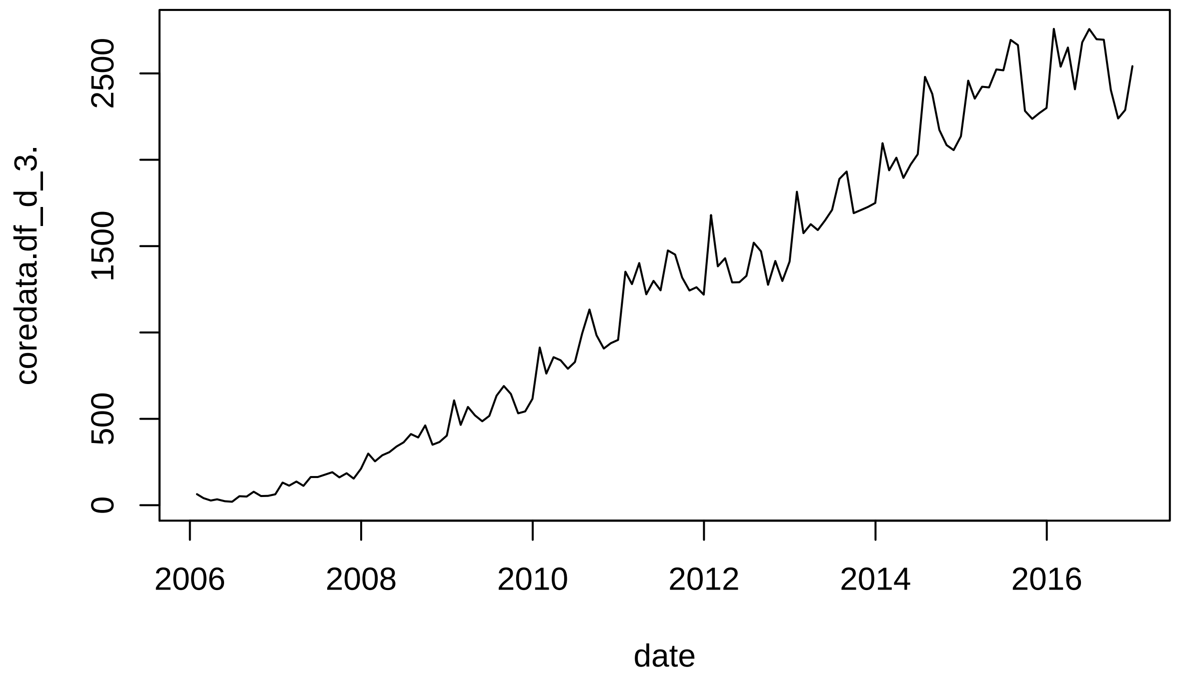
```
dollars_1_xts = xts(dollars_gbd_1$`count(date)`, as.Date(dollars_gbd_1$date,
    "%Y-%m-%d"))
df_d_1 = apply.monthly(dollars_1_xts, sum)
df_dollars_1 = data.frame(date = index(df_d_1), coredata(df_d_1))
# df_dollars_1

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)


plot(df_dollars_1, type = "l")
```



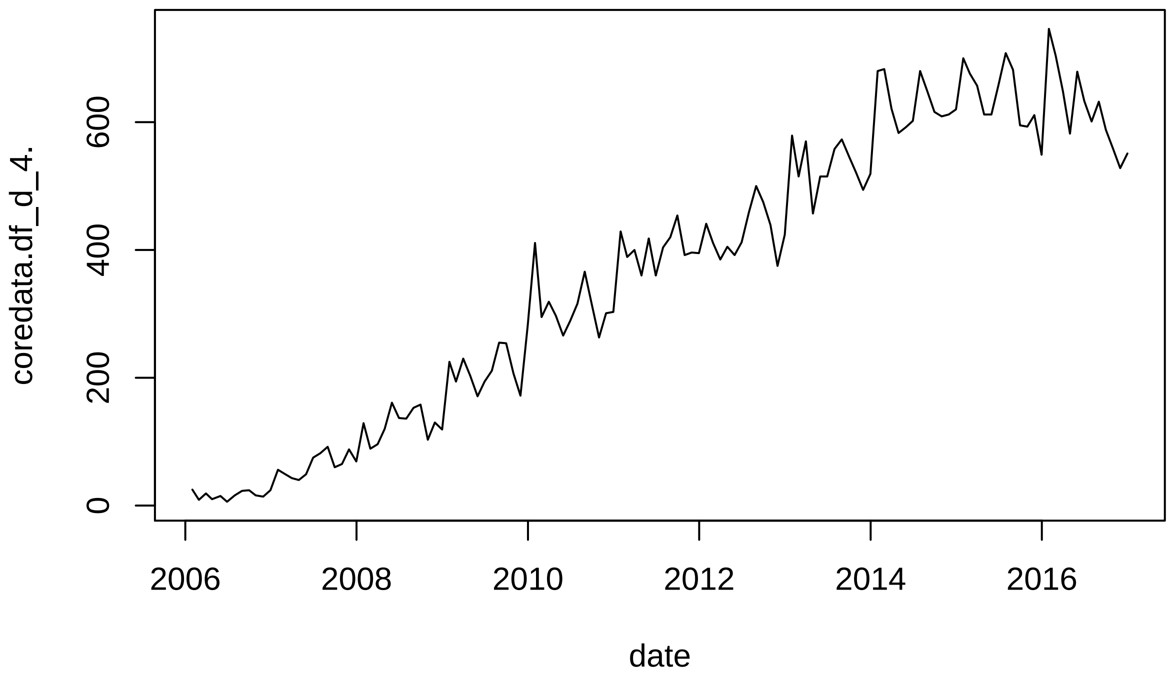
```
plot(df_dollars_2, type = "l")
```



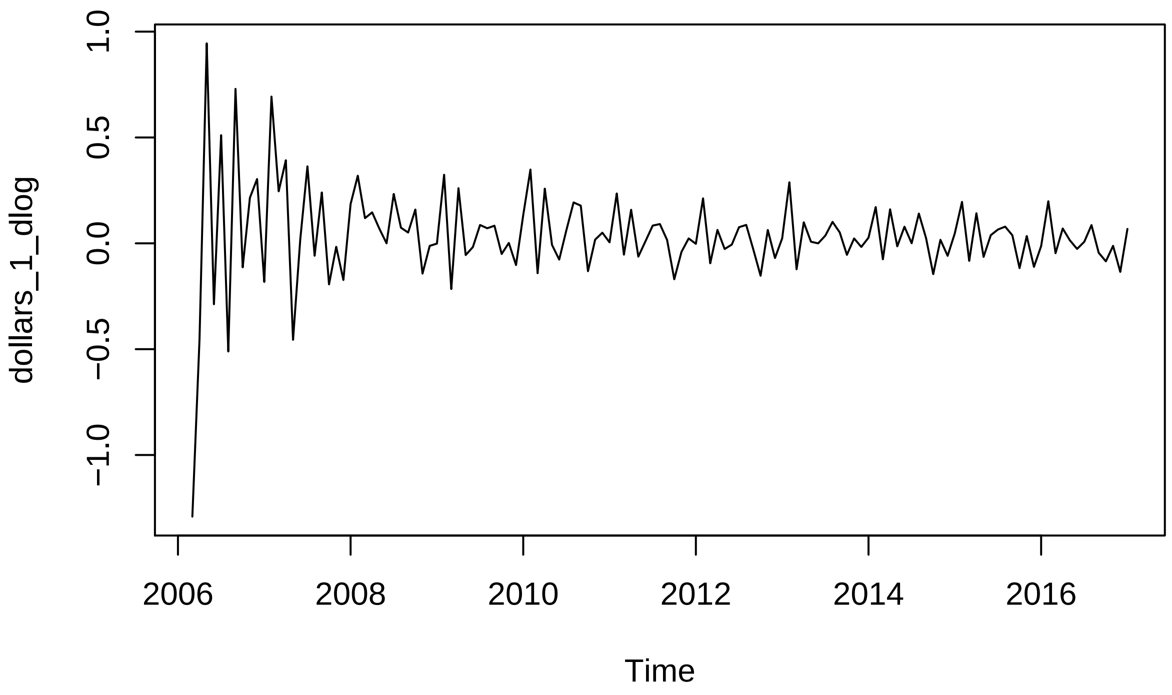
```
plot(df_dollars_3, type = "l")
```



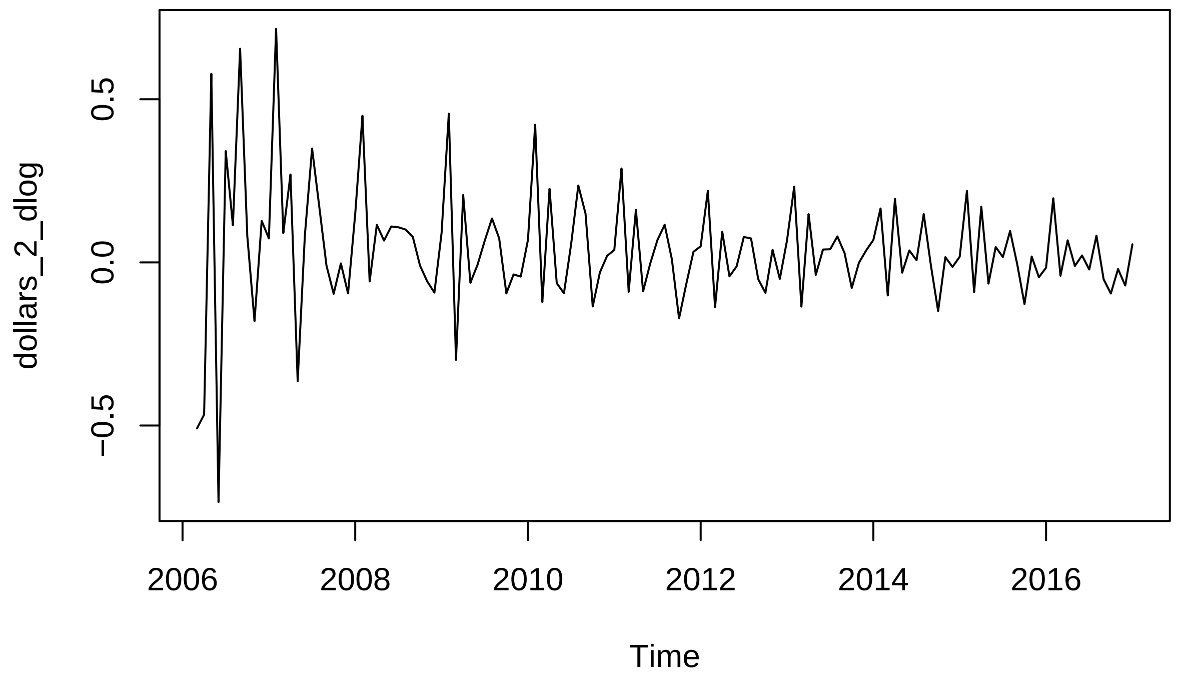
```r
plot(df_dollars_4, type = "l")
```



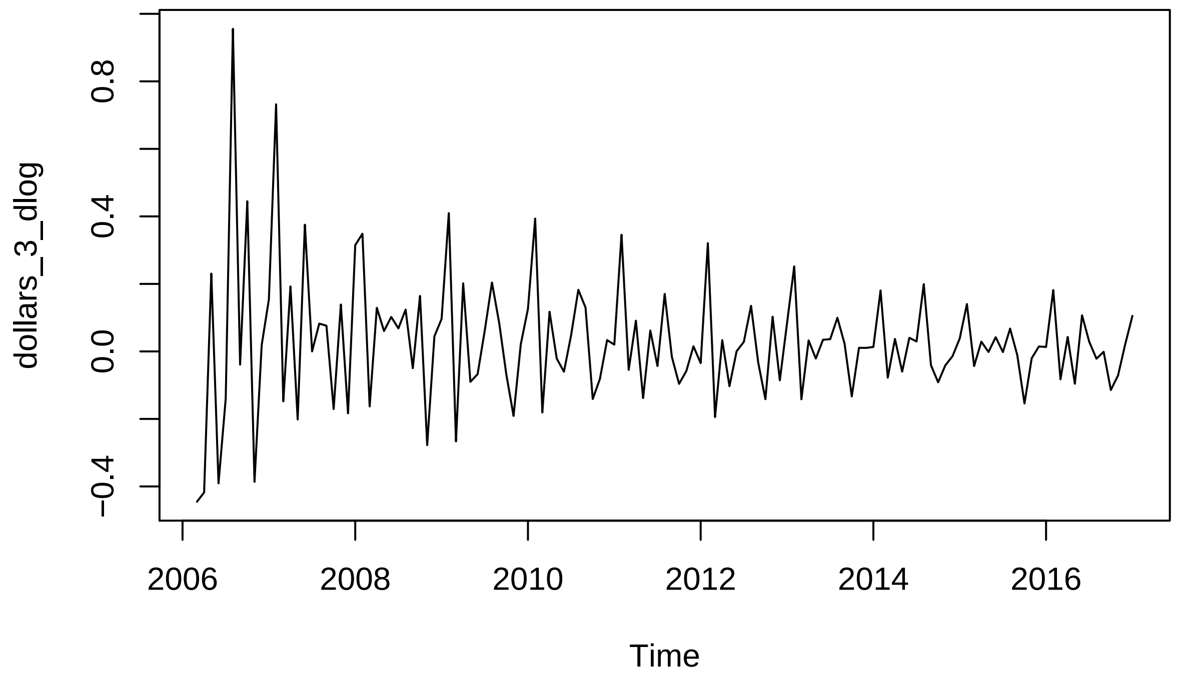
```
df_dollars_1_dlog = as.data.frame(diff(log(df_dollars_1$coredata.df_d_1.)))
df_dollars_2_dlog = as.data.frame(diff(log(df_dollars_2$coredata.df_d_2.)))
df_dollars_3_dlog = as.data.frame(diff(log(df_dollars_3$coredata.df_d_3.)))
df_dollars_4_dlog = as.data.frame(diff(log(df_dollars_4$coredata.df_d_4.)))


# 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)


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))

plot(dollars_1_dlog)
```



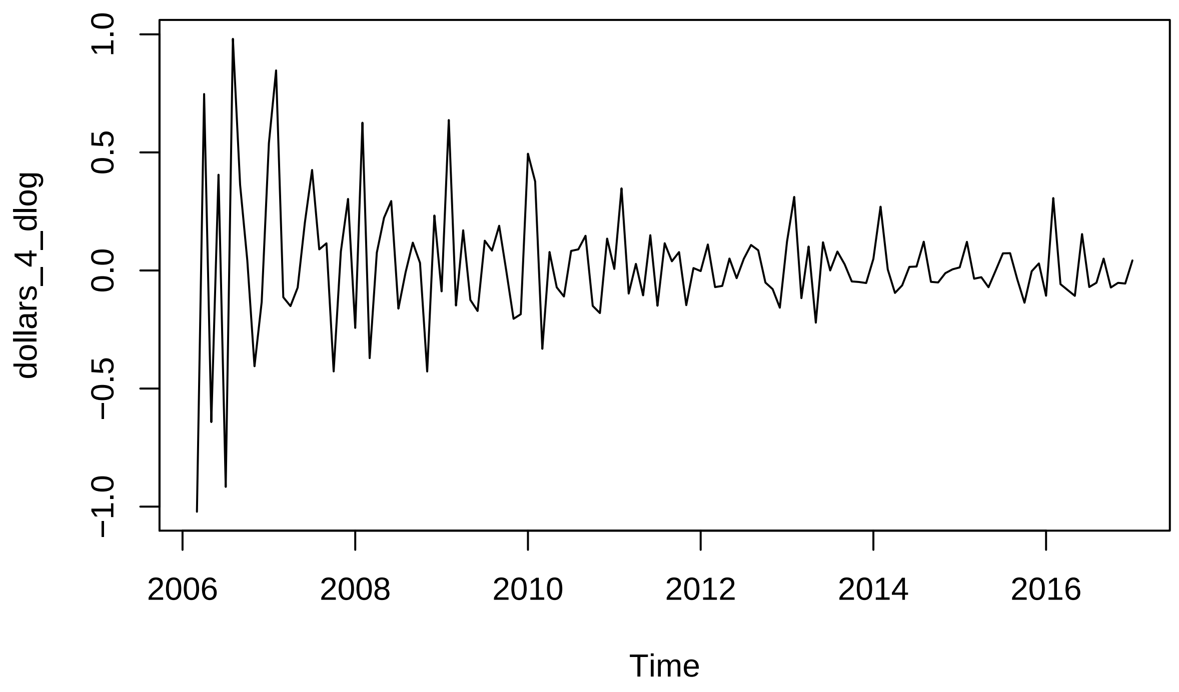
```r
plot(dollars_2_dlog)
```



```
plot(dollars_3_dlog)
```



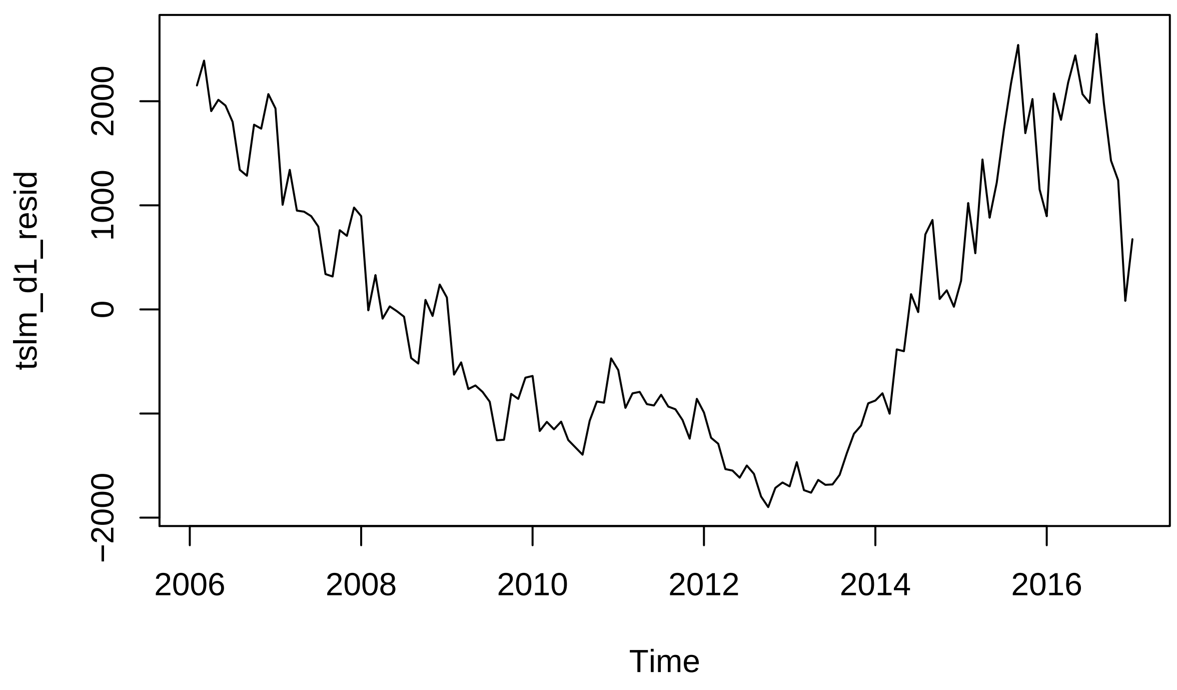
```r
plot(dollars_4_dlog)
```



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

seasonally adjust the data

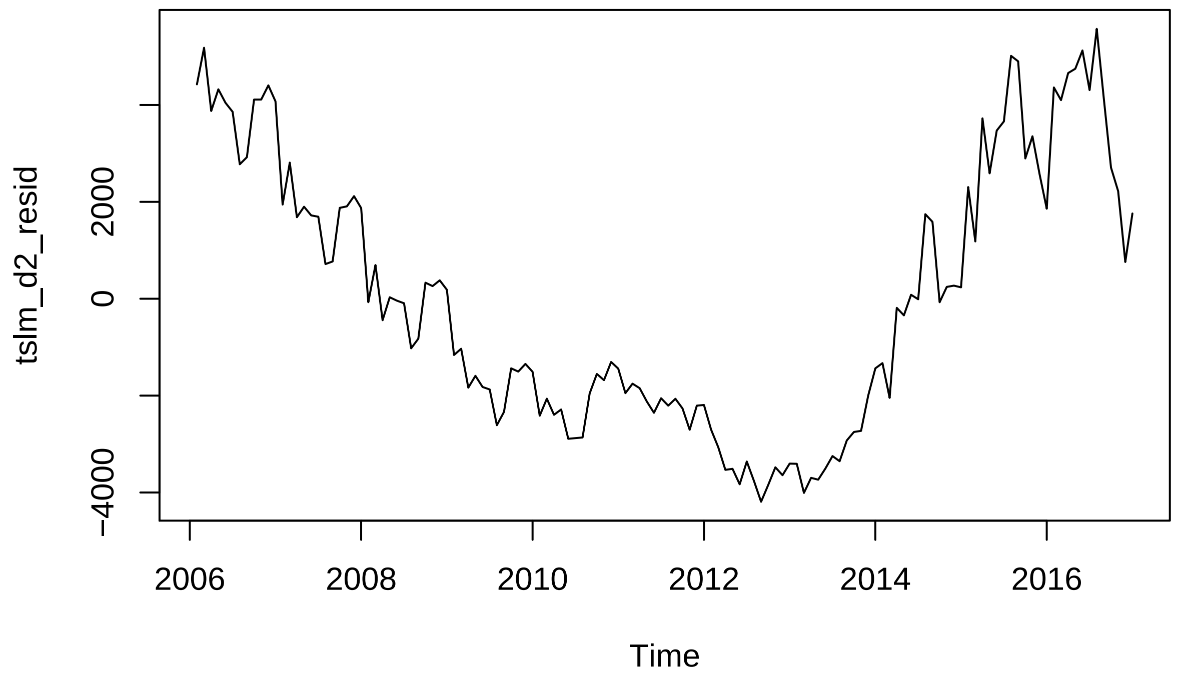
```
# seasonally adjust data

# levels
tslm_d1 = tslm(ts_dollar_1 ~ trend + season)
# summary(tslm_d1)
tslm_d1_resid = resid(tslm_d1)
plot(tslm_d1_resid)
```



```
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
```

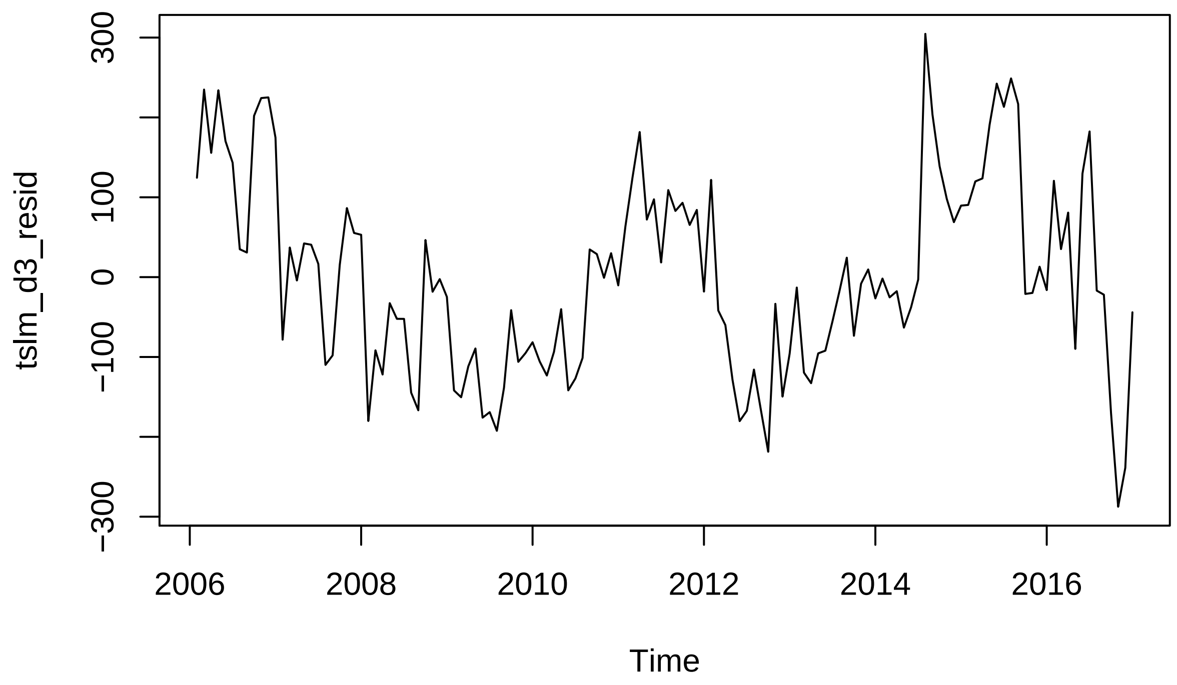
```
tslm_d2 = tslm(ts_dollar_2 ~ trend + season)
# summary(tslm_d1)
tslm_d2_resid = resid(tslm_d2)
plot(tslm_d2_resid)
```



```r
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
```

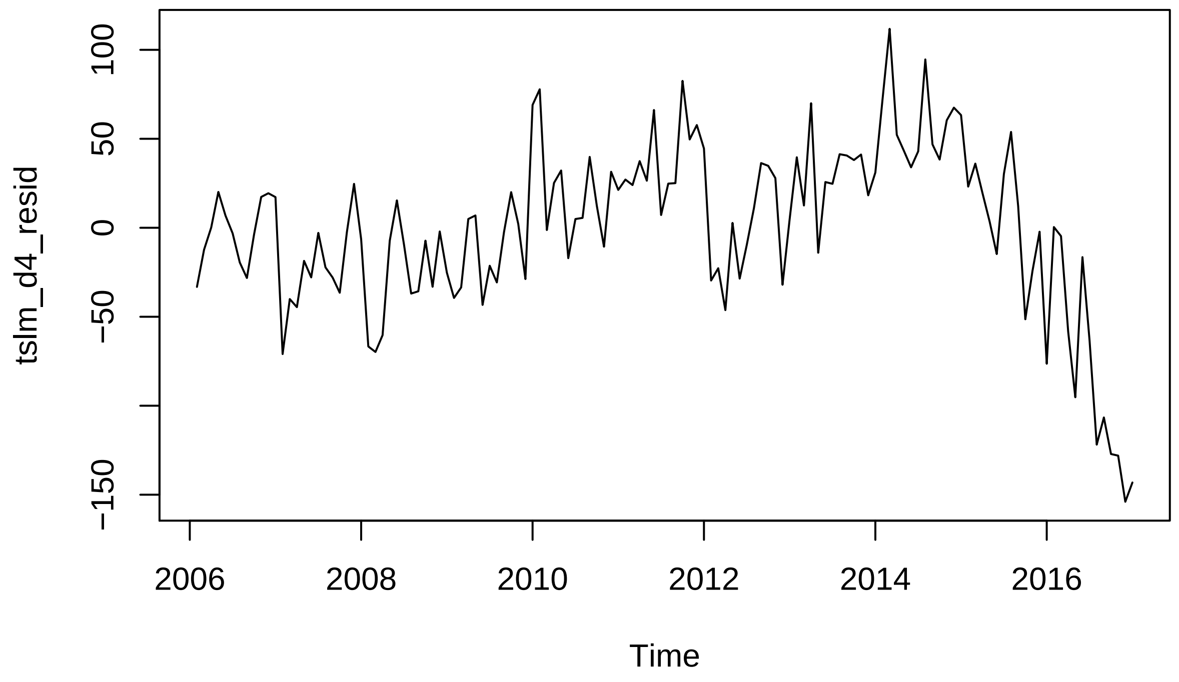
```r
tslm_d3 = tslm(ts_dollar_3 ~ trend + season)
# summary(tslm_d1)
tslm_d3_resid = resid(tslm_d3)
plot(tslm_d3_resid)
```



```
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
```

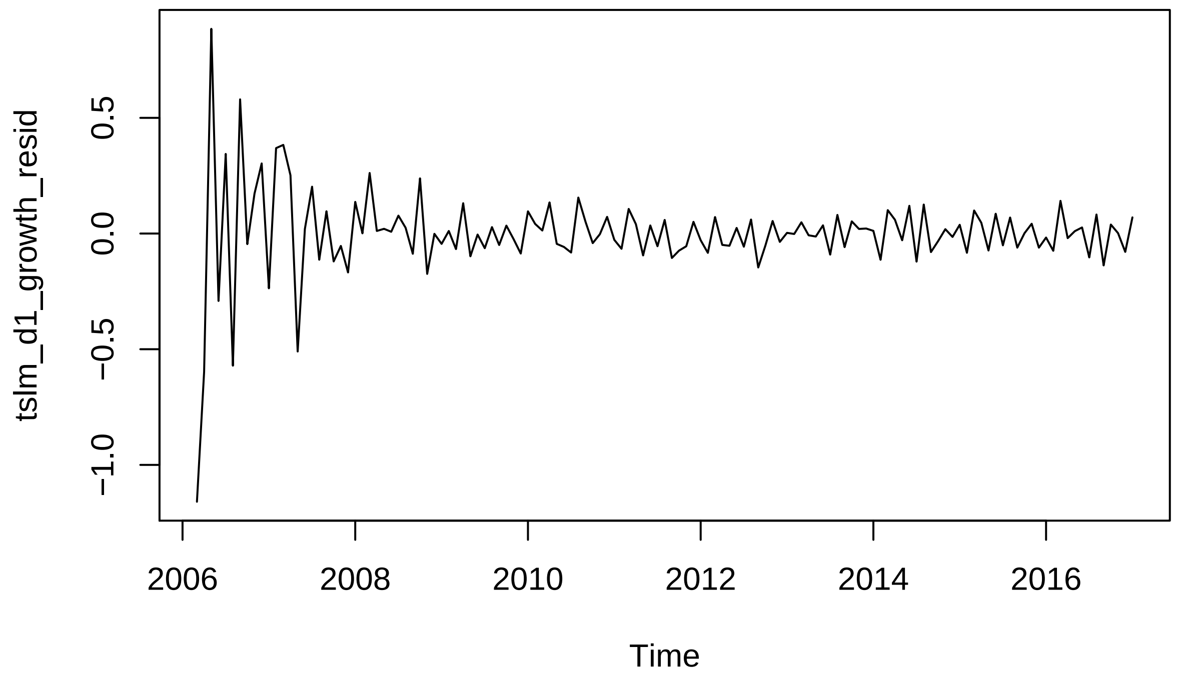
```
tslm_d4 = tslm(ts_dollar_4 ~ trend + season)
# summary(tslm_d1)
tslm_d4_resid = resid(tslm_d4)
plot(tslm_d4_resid)
```



```
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
```

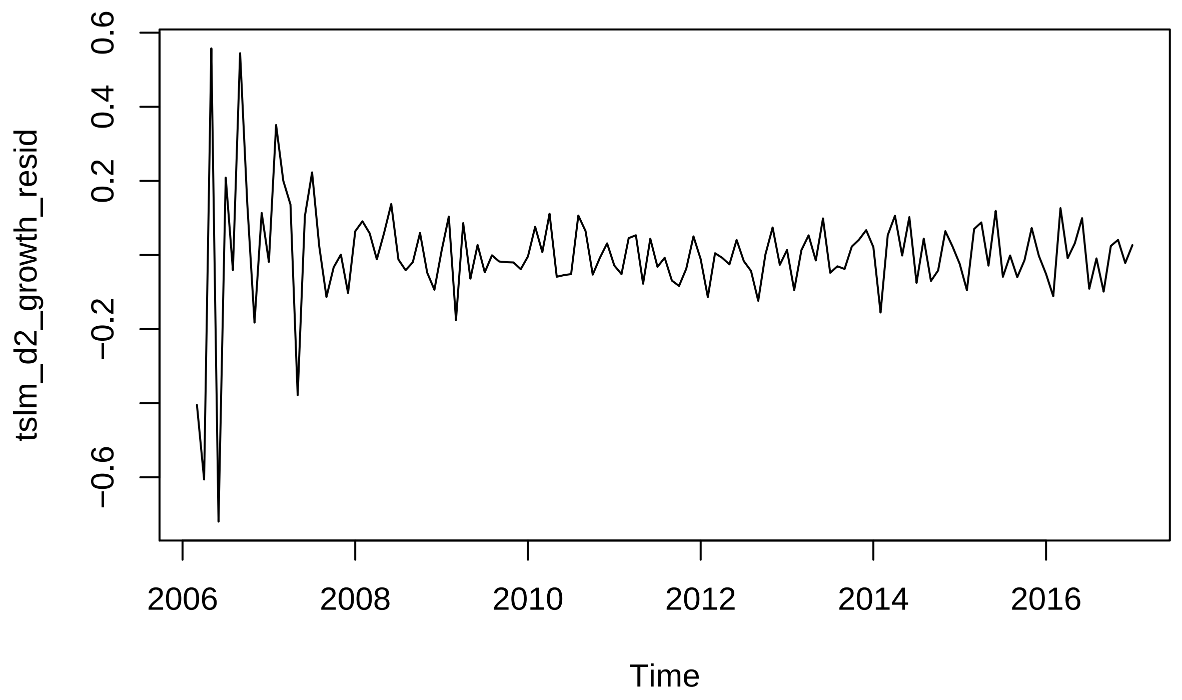
```
# growth
tslm_d1_growth = tslm(dollars_1_dlog ~ trend + season)
# summary(tslm_d1)
tslm_d1_growth_resid = resid(tslm_d1_growth)
plot(tslm_d1_growth_resid)
```



```
lm_d1_growth_adj = lm(tslm_d1_growth_resid ~ lm_dollars_recession_dummy)
summary(lm_d1_growth_adj)
```

```
##
## Call:
## lm(formula = tslm_d1_growth_resid ~ lm_dollars_recession_dummy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.15582 -0.06009 -0.00420  0.05984  0.88806
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -0.003396   0.018404  -0.185    0.854
## lm_dollars_recession_dummy  0.023417   0.048325   0.485    0.629
##
## Residual standard error: 0.1948 on 129 degrees of freedom
## Multiple R-squared:  0.001817,   Adjusted R-squared:  -0.005921
## F-statistic: 0.2348 on 1 and 129 DF,  p-value: 0.6288
```

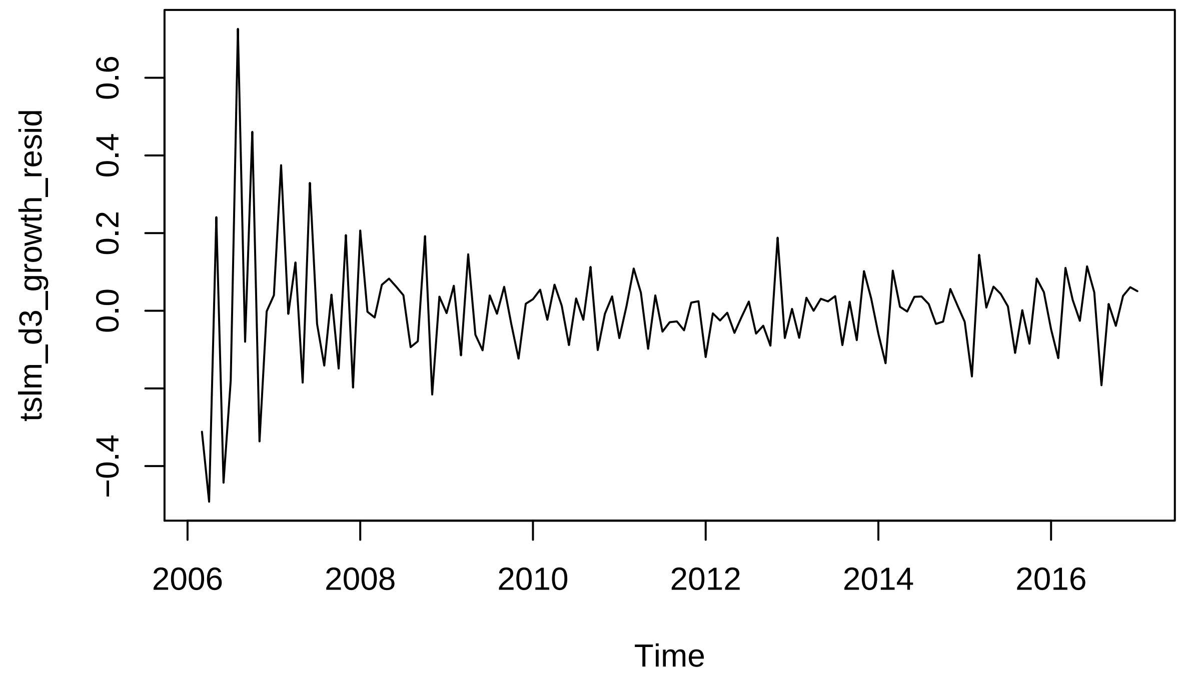
```
tslm_d2_growth = tslm(dollars_2_dlog ~ trend + season)
# summary(tslm_d1)
tslm_d2_growth_resid = resid(tslm_d2_growth)
plot(tslm_d2_growth_resid)
```



```r
lm_d2_growth_adj = lm(tslm_d2_growth_resid ~ lm_dollars_recession_dummy)
summary(lm_d2_growth_adj)
```

```
##
## Call:
## lm(formula = tslm_d2_growth_resid ~ lm_dollars_recession_dummy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.71790 -0.04976 -0.00577  0.05448  0.55929
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -0.001668   0.013474  -0.124    0.902
## lm_dollars_recession_dummy  0.011500   0.035381   0.325    0.746
##
## Residual standard error: 0.1426 on 129 degrees of freedom
## Multiple R-squared:  0.0008183,  Adjusted R-squared:  -0.006927
## F-statistic: 0.1056 on 1 and 129 DF,  p-value: 0.7457
```

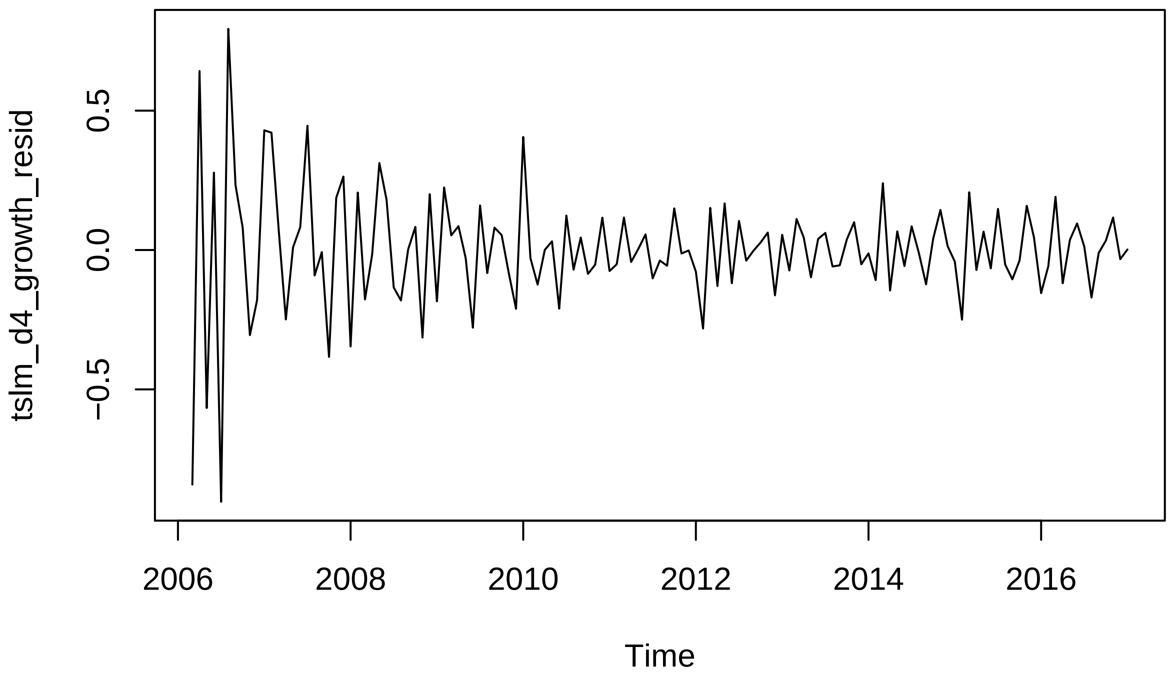
```r
tslm_d3_growth = tslm(dollars_3_dlog ~ trend + season)
# summary(tslm_d1)
tslm_d3_growth_resid = resid(tslm_d3_growth)
plot(tslm_d3_growth_resid)
```



```
lm_d3_growth_adj = lm(tslm_d3_growth_resid ~ lm_dollars_recession_dummy)
summary(lm_d3_growth_adj)
```

```
##
## Call:
## lm(formula = tslm_d3_growth_resid ~ lm_dollars_recession_dummy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48967 -0.06243  0.00702  0.04723  0.72817
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -0.002172   0.013268  -0.164    0.870
## lm_dollars_recession_dummy  0.014978   0.034838   0.430    0.668
##
## Residual standard error: 0.1404 on 129 degrees of freedom
## Multiple R-squared:  0.001431,   Adjusted R-squared:  -0.00631
## F-statistic: 0.1848 on 1 and 129 DF,  p-value: 0.668
```

```
tslm_d4_growth = tslm(dollars_4_dlog ~ trend + season)
# summary(tslm_d1)
tslm_d4_growth_resid = resid(tslm_d4_growth)
plot(tslm_d4_growth_resid)
```

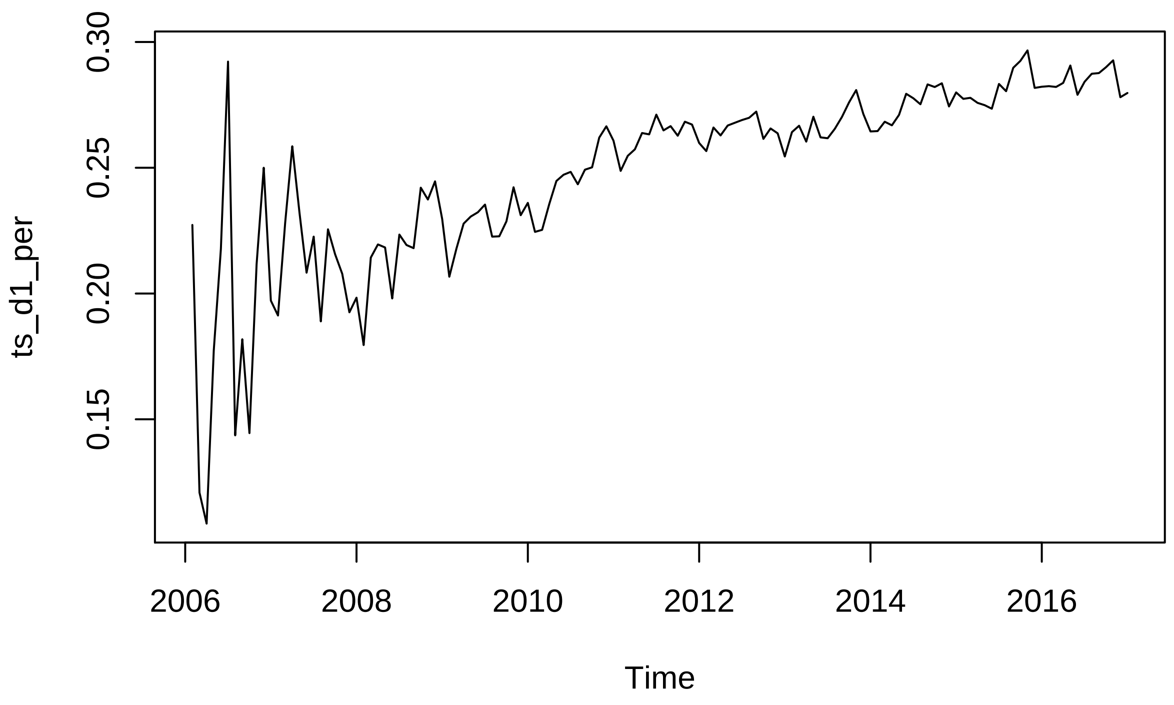


```
lm_d4_growth_adj = lm(tslm_d4_growth_resid ~ lm_dollars_recession_dummy)
summary(lm_d4_growth_adj)
```

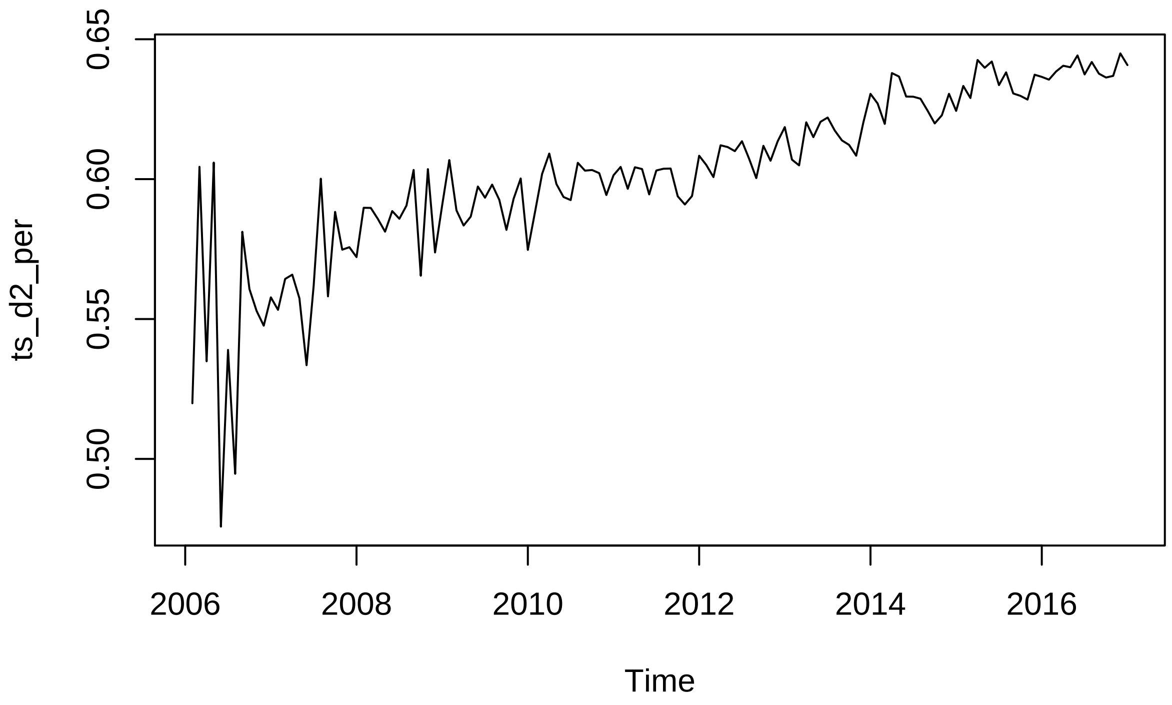
```
##
## Call:
## lm(formula = tslm_d4_growth_resid ~ lm_dollars_recession_dummy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90447 -0.08604 -0.00312  0.09386  0.79209
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 0.001376   0.020201   0.068    0.946
## lm_dollars_recession_dummy -0.009485   0.053043  -0.179    0.858
##
## Residual standard error: 0.2138 on 129 degrees of freedom
## Multiple R-squared:  0.0002478,  Adjusted R-squared:  -0.007502
## F-statistic: 0.03198 on 1 and 129 DF,  p-value: 0.8584
```

```
# percentages
ts_dollar_total = ts_dollar_1 + ts_dollar_2 + ts_dollar_3 + ts_dollar_4
ts_d1_per = ts_dollar_1/ts_dollar_total
ts_d2_per = ts_dollar_2/ts_dollar_total
ts_d3_per = ts_dollar_3/ts_dollar_total
ts_d4_per = ts_dollar_4/ts_dollar_total
```

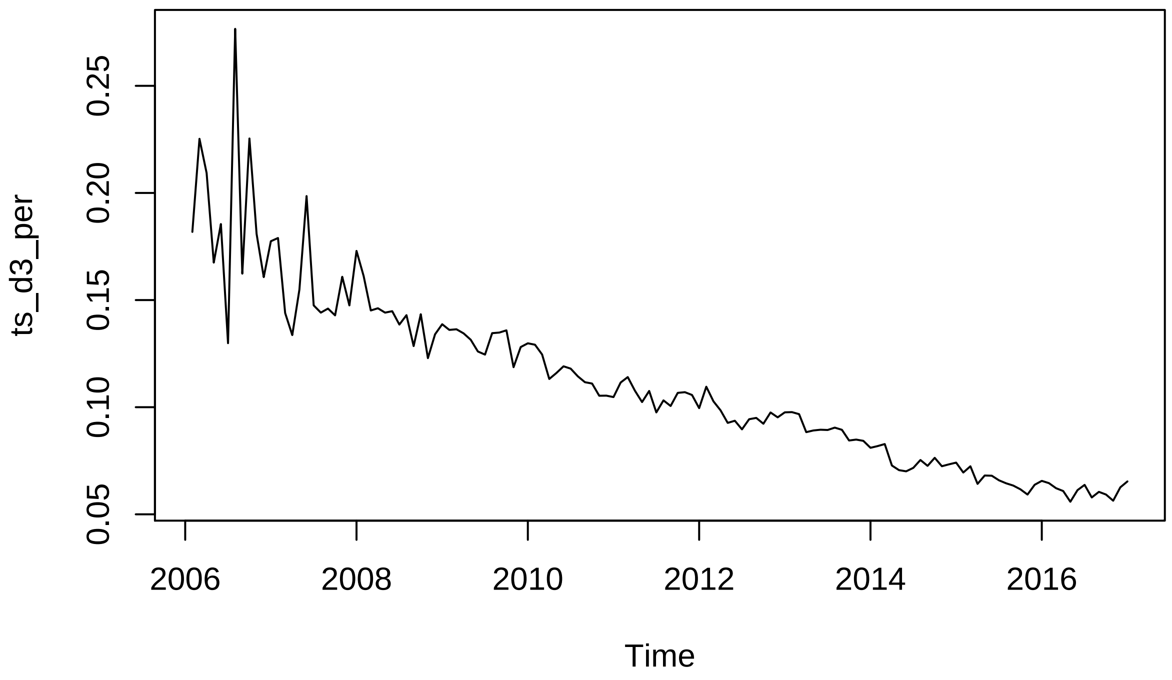
```
plot(ts_d1_per)
```



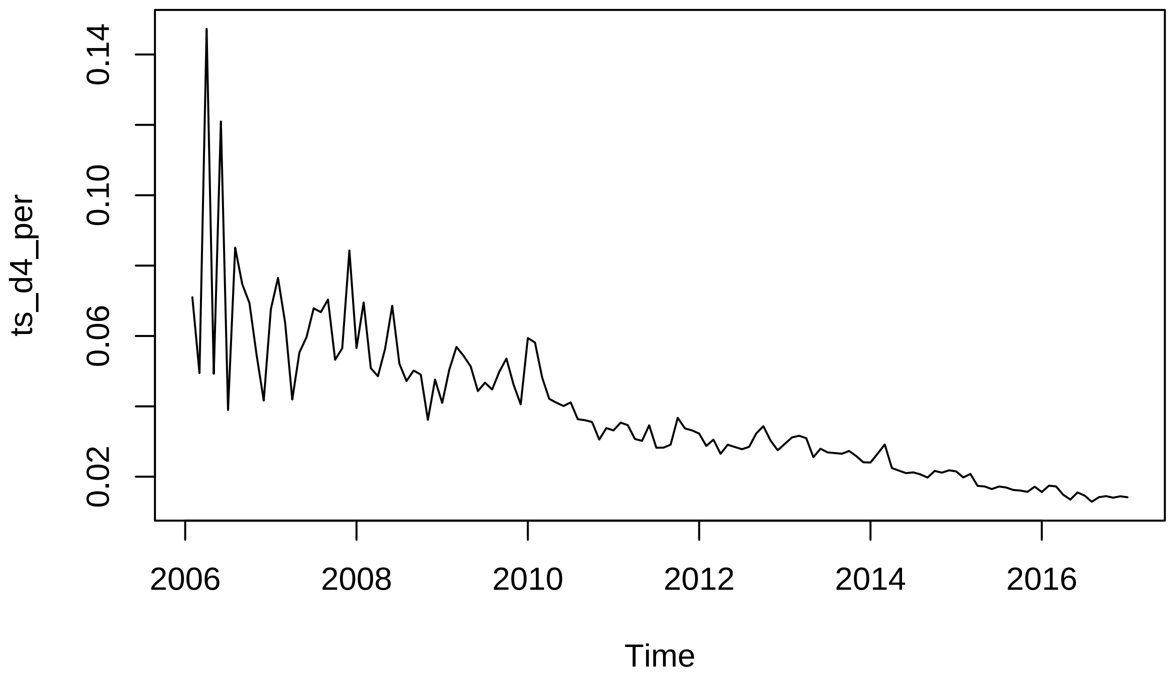
```
plot(ts_d2_per)
```



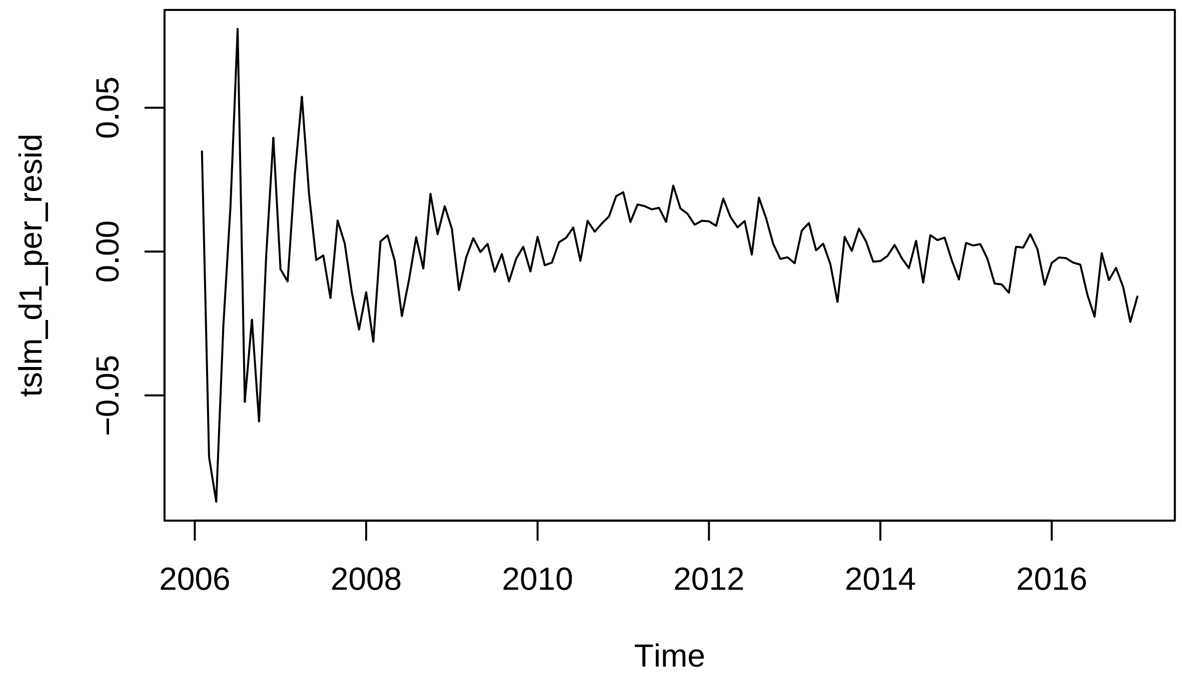
```r
plot(ts_d3_per)
```



```r
plot(ts_d4_per)
```



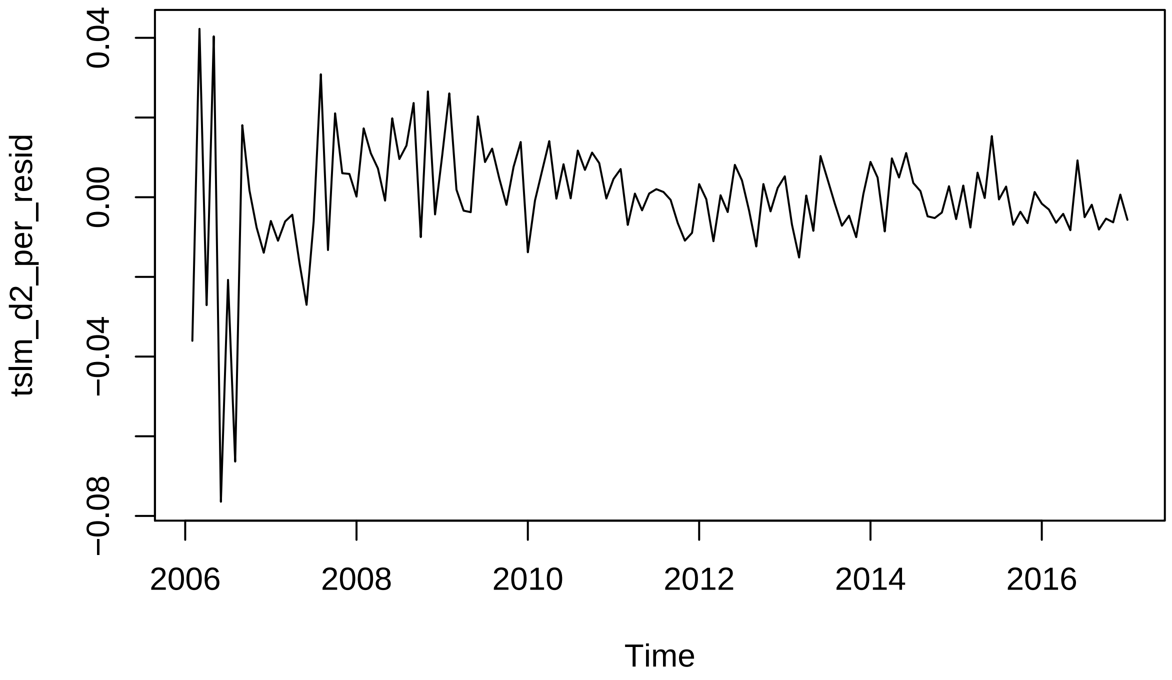
```
tslm_d1_per = tslm(ts_d1_per ~ trend + season)
tslm_d1_per_resid = resid(tslm_d1_per)
plot(tslm_d1_per_resid)
```



```
lm_d1_per_adj = lm(tslm_d1_per_resid ~ recession_dummy_dollars_m)
summary(lm_d1_per_adj)
```

```
##
## Call:
## lm(formula = tslm_d1_per_resid ~ recession_dummy_dollars_m)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.087316 -0.005327  0.001185  0.009100  0.077087
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.0003379  0.0017737   0.191    0.849
## recession_dummy_dollars_m -0.0023478  0.0046751  -0.502    0.616
##
## Residual standard error: 0.01885 on 130 degrees of freedom
## Multiple R-squared:  0.001936,   Adjusted R-squared:  -0.005741
## F-statistic: 0.2522 on 1 and 130 DF,  p-value: 0.6164
```

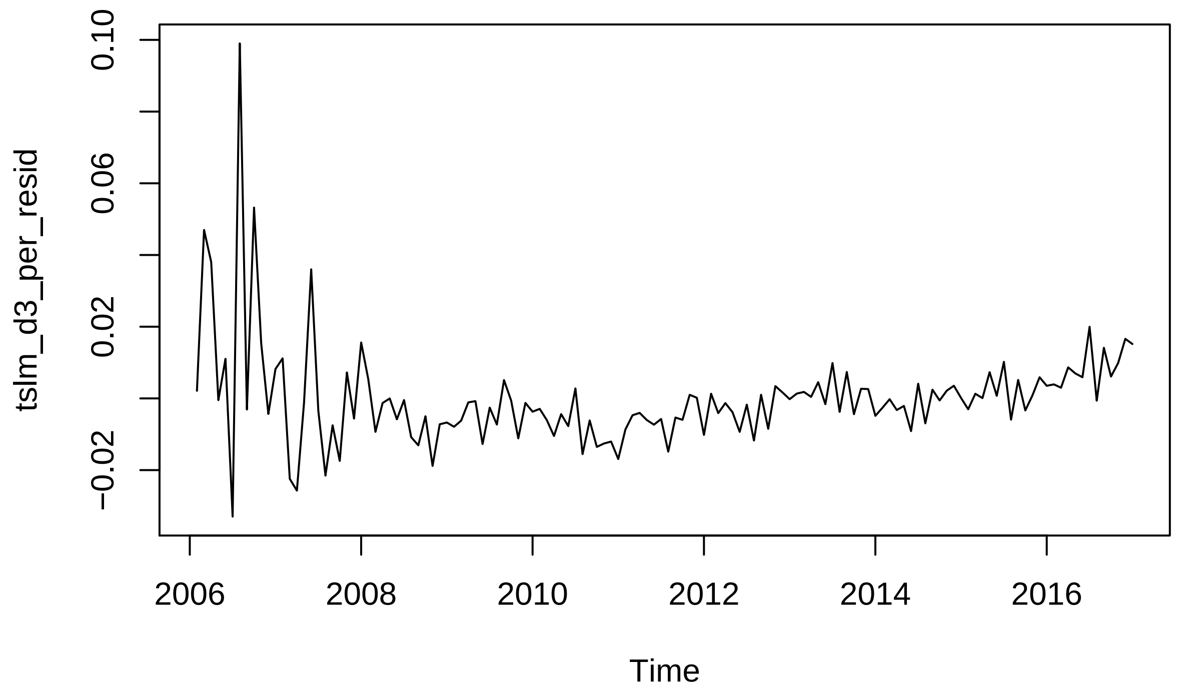
```
tslm_d2_per = tslm(ts_d2_per ~ trend + season)
tslm_d2_per_resid = resid(tslm_d2_per)
plot(tslm_d2_per_resid)
```



```
lm_d2_per_adj = lm(tslm_d2_per_resid ~ recession_dummy_dollars_m)
summary(lm_d2_per_adj)
```

```
##
## Call:
## lm(formula = tslm_d2_per_resid ~ recession_dummy_dollars_m)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.074900 -0.005344  0.000920  0.006557  0.043791
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -0.001535   0.001324  -1.159  0.24858
## recession_dummy_dollars_m  0.010663   0.003491   3.055  0.00273 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01408 on 130 degrees of freedom
## Multiple R-squared:  0.06698,    Adjusted R-squared:  0.0598
## F-statistic: 9.332 on 1 and 130 DF,  p-value: 0.002732
```

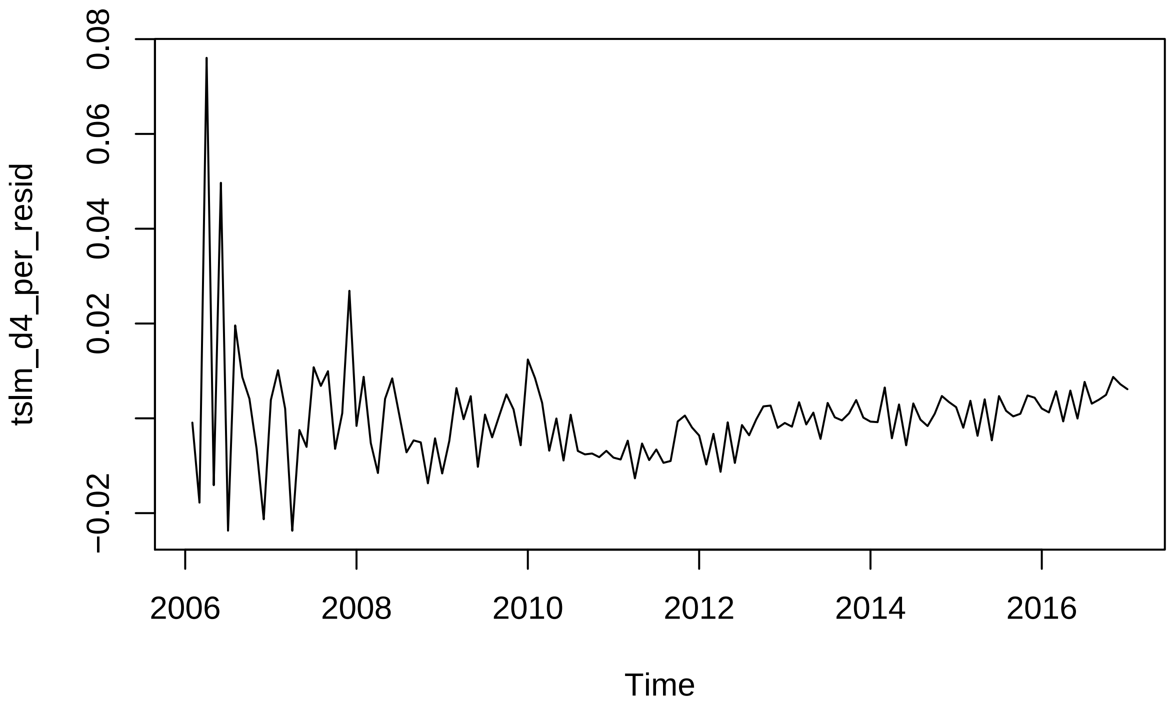
```
tslm_d3_per = tslm(ts_d3_per ~ trend + season)
tslm_d3_per_resid = resid(tslm_d3_per)
plot(tslm_d3_per_resid)
```



```
lm_d3_per_adj = lm(tslm_d3_per_resid ~ recession_dummy_dollars_m)
summary(lm_d3_per_adj)
```

```
##
## Call:
## lm(formula = tslm_d3_per_resid ~ recession_dummy_dollars_m)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.033760 -0.006447 -0.001380  0.003443  0.098237
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.0007874  0.0013541   0.581    0.562
## recession_dummy_dollars_m -0.0054702  0.0035690  -1.533    0.128
##
## Residual standard error: 0.01439 on 130 degrees of freedom
## Multiple R-squared:  0.01775,    Adjusted R-squared:  0.01019
## F-statistic: 2.349 on 1 and 130 DF,  p-value: 0.1278
```

```
tslm_d4_per = tslm(ts_d4_per ~ trend + season)
tslm_d4_per_resid = resid(tslm_d4_per)
plot(tslm_d4_per_resid)
```



```
lm_d4_per_adj = lm(tslm_d4_per_resid ~ recession_dummy_dollars_m)
summary(lm_d4_per_adj)
```

```
##
## Call:
## lm(formula = tslm_d4_per_resid ~ recession_dummy_dollars_m)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.024128 -0.005074 -0.000451  0.003447  0.075637
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.0004095  0.0010212   0.401    0.689
## recession_dummy_dollars_m -0.0028453  0.0026917  -1.057    0.292
##
## Residual standard error: 0.01086 on 130 degrees of freedom
## Multiple R-squared:  0.008522,   Adjusted R-squared:  0.0008948
## F-statistic: 1.117 on 1 and 130 DF,  p-value: 0.2925
```

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:
```

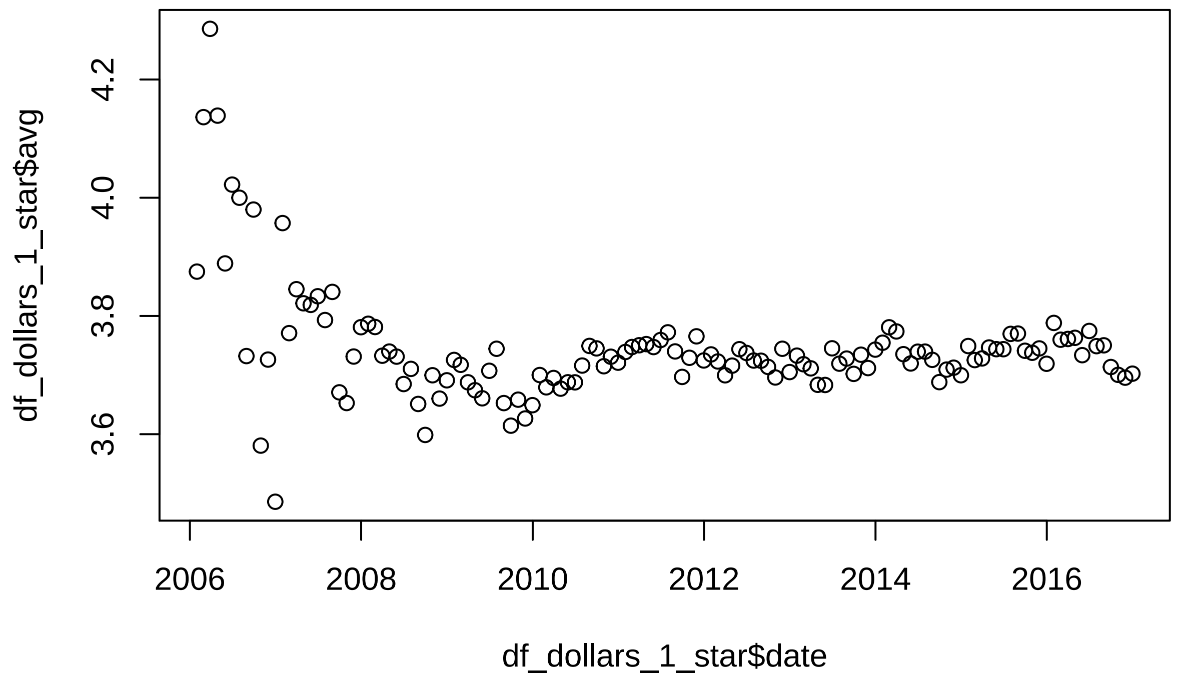
```
## lm(formula = tslm_d3_resid ~ recession_dummy_dollars_m + df_rev_count$coredata.df_rev_m.)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -315.824  -83.622   -8.919   77.962  280.690
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     3.403e+00  1.754e+01   0.194  0.84646
## recession_dummy_dollars_m      -9.085e+01  3.116e+01  -2.916  0.00418 **
## df_rev_count$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
```

```r
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
```

re examine stars, but subset by dollar signs

```r
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)
```

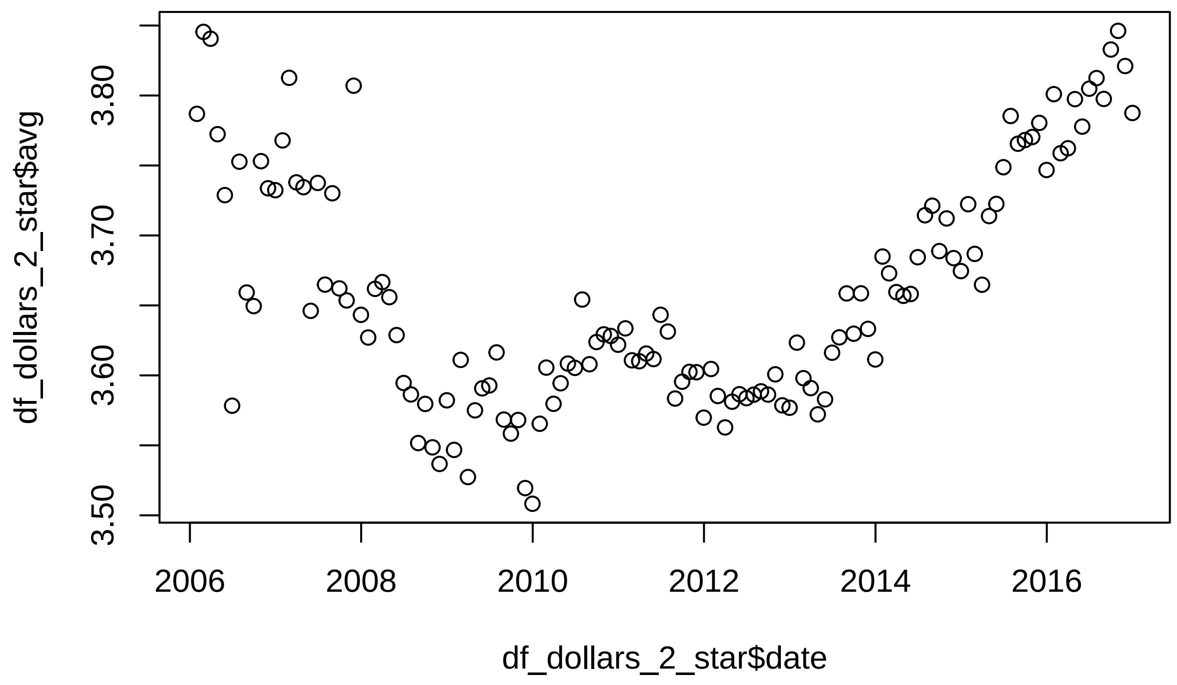


```
d1_star_lm = lm(df_dollars_1_star$avg ~ recession_dummy_dollars_m)
summary(d1_star_lm)
```

```
##
## 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.
```

```r
plot(df_dollars_2_star$date, df_dollars_2_star$avg)
```

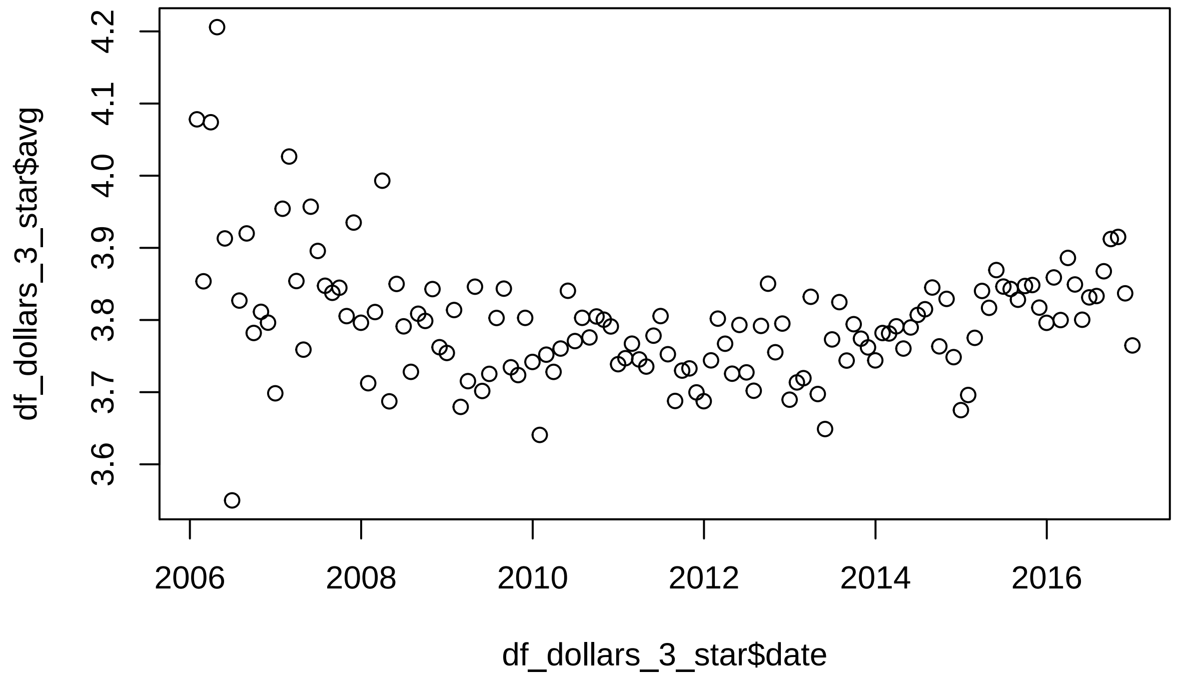


```r
d2_star_lm = lm(df_dollars_2_star$avg ~ recession_dummy_dollars_m)
summary(d2_star_lm)
```

```
##
## 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
```

```r
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)
```

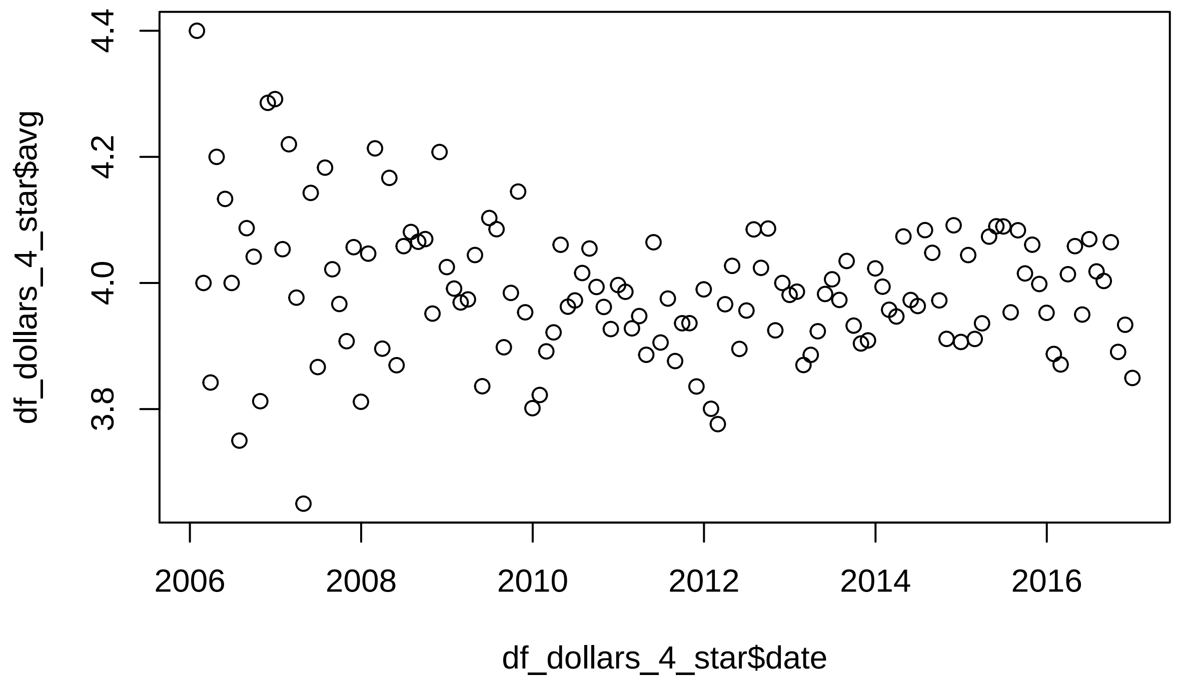
```r
df_dollars_3_star = data.frame(date = index(df_d_3_star), coredata(df_d_3_star))
df_dollars_3_star$avg = df_dollars_3_star$coredata.df_d_3_star./df_dollars_3$coredata.df_d_3.
plot(df_dollars_3_star$date, df_dollars_3_star$avg)
```



```r
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
```

```r
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)
```



```r
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
```

comment out for now, cuz takes too long to process

analyze word usages

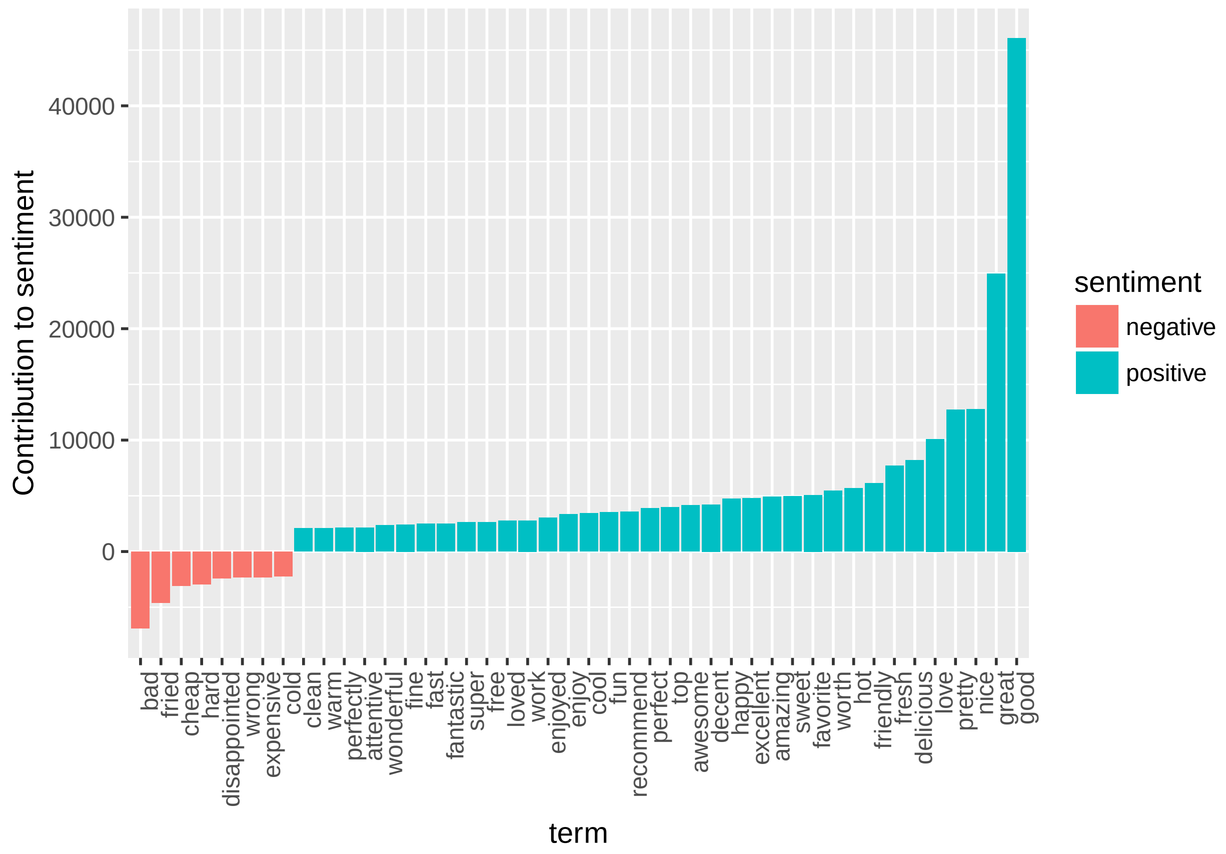
```
buildCorpus = function(data, stem) {
    corpus = Corpus(VectorSource(data))
    corpus = tm_map(corpus, content_transformer(tolower))
    corpus = tm_map(corpus, PlainTextDocument)
    corpus = tm_map(corpus, removePunctuation)
    corpus = tm_map(corpus, removeWords, stopWords)
    if (stem ==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

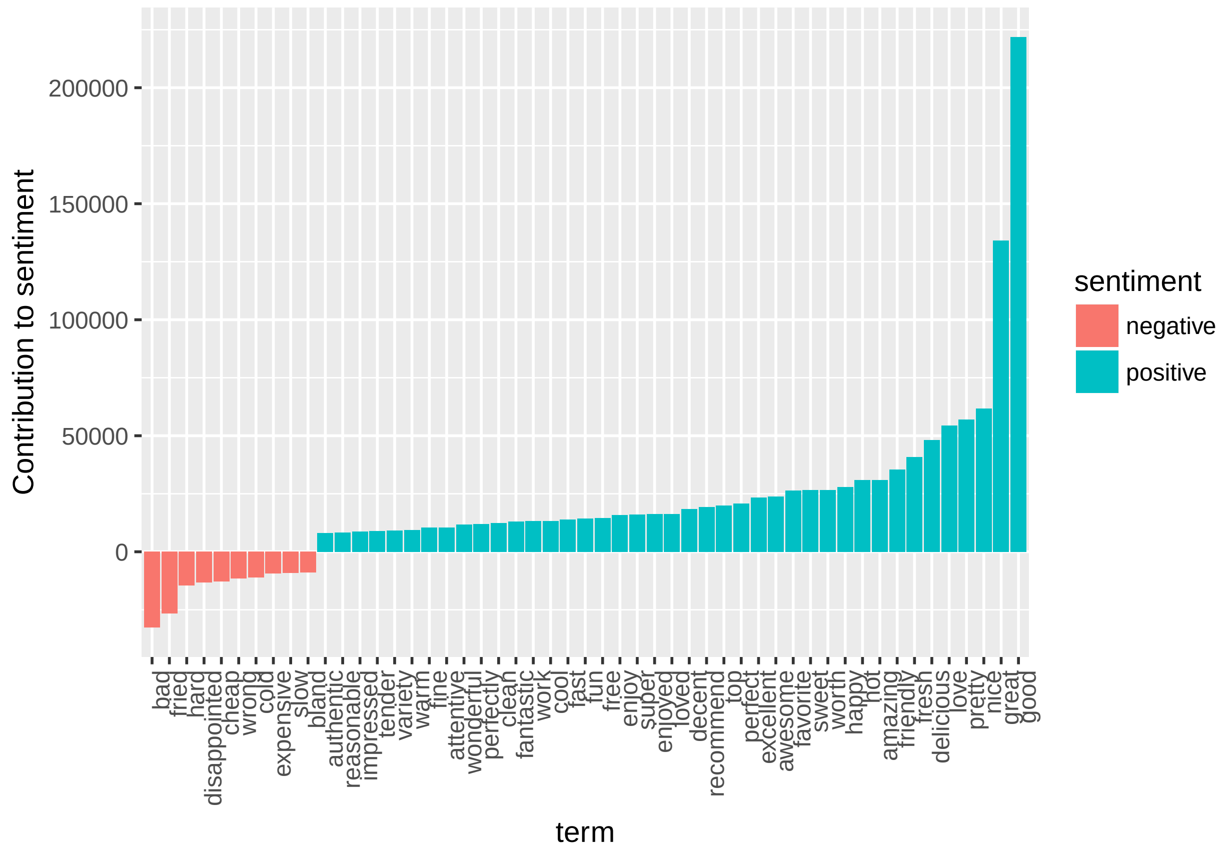
```r
dtm_norec = DocumentTermMatrix(corpus_reviews_norec)
tidy_norec = tidy(dtm_norec)
ap_sentiments <- tidy_norec %>% inner_join(get_sentiments("bing"),
    by = c(term = "word"))

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
```

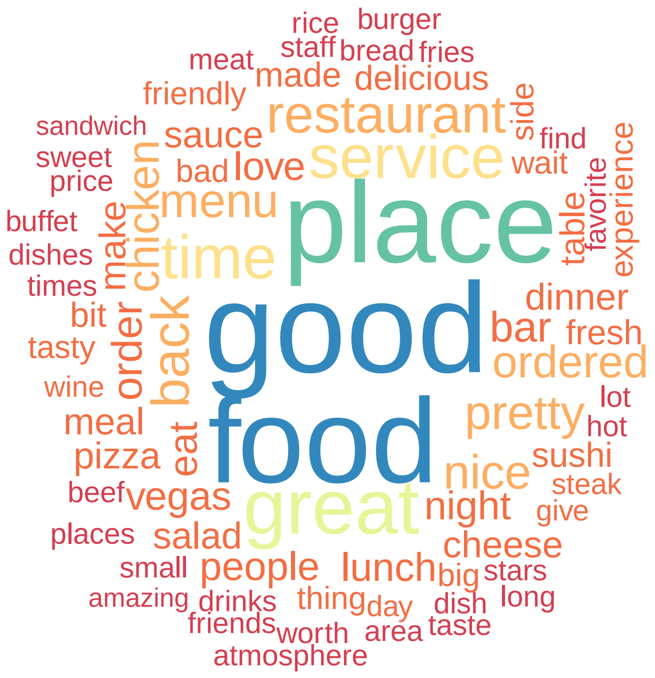
```r
ap_sentiments %>% count(document, sentiment, wt = count) %>%
    ungroup() %>% spread(sentiment, n, fill = 0) %>% mutate(sentiment = positive -
    negative) %>% arrange(sentiment)
```

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

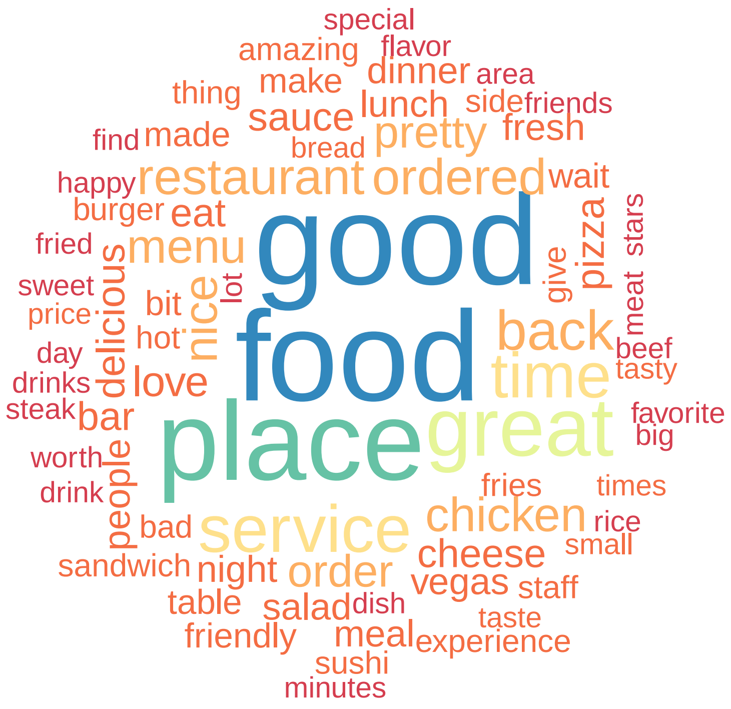
```r
ap_sentiments %>% count(sentiment, term, wt = count) %>% ungroup() %>%
    filter(n >= 7500) %>% mutate(n = ifelse(sentiment == "negative",
    -n, n)) %>% mutate(term = reorder(term, n)) %>% ggplot(aes(term,
    n, fill = sentiment)) + geom_bar(stat = "identity") + theme(axis.text.x = element_text(angle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



```
# negative: 733329 positive: 1731938 percent negative: 29.75%
# cheap: 5th expensive: 8th

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

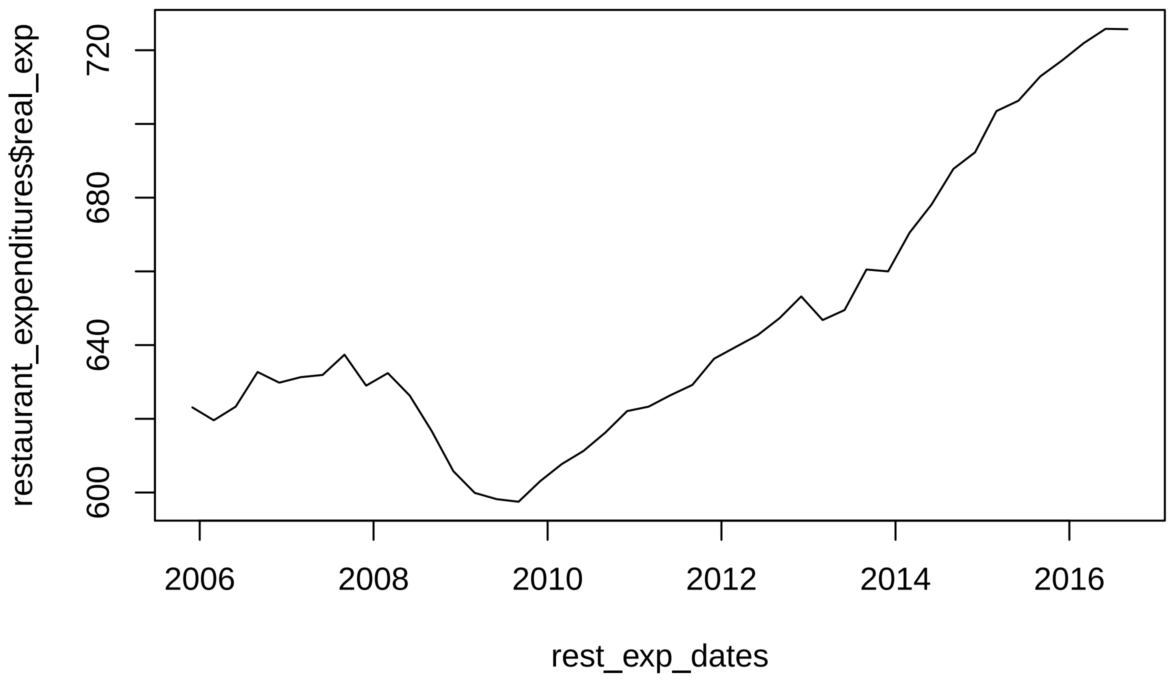


```r
buildWordCloud(corpus_reviews_norec, "Spectral", 8, "test2")
```



connect with restaurants

```
# 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")
```



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

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

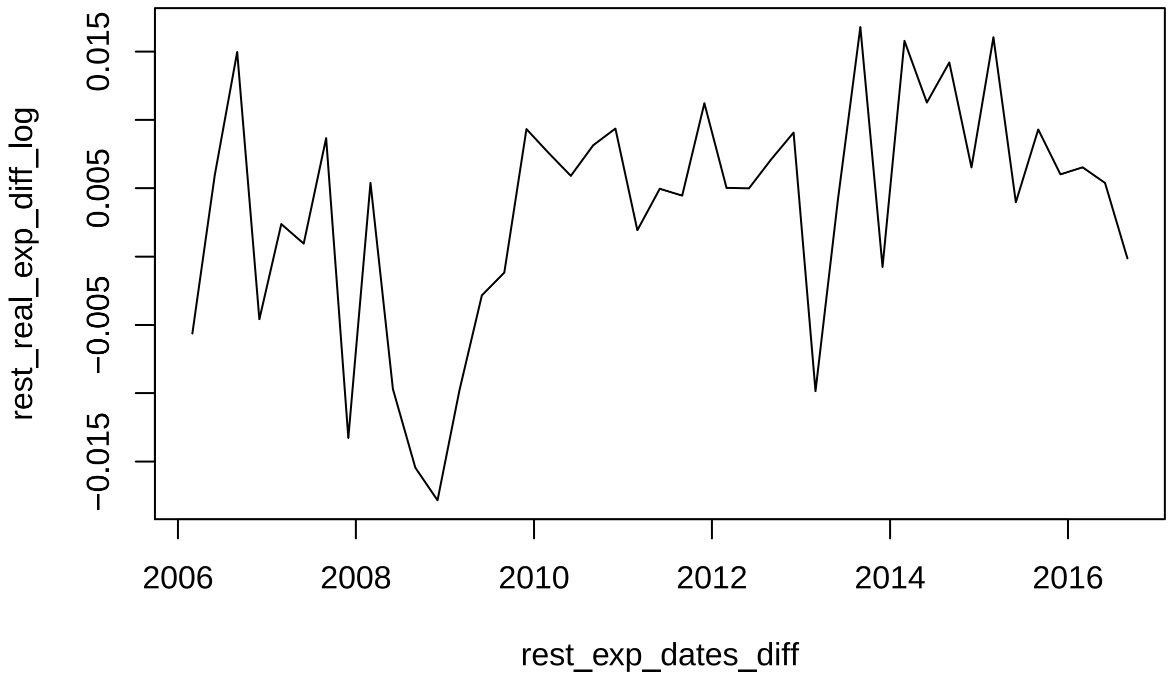
```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 1.7047, Truncation lag parameter = 1, p-value = 0.01
##
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -2.4338, Lag order = 3, p-value = 0.4021
## alternative hypothesis: stationary
```

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

test_stationary(rest_real_exp_diff_log)
```

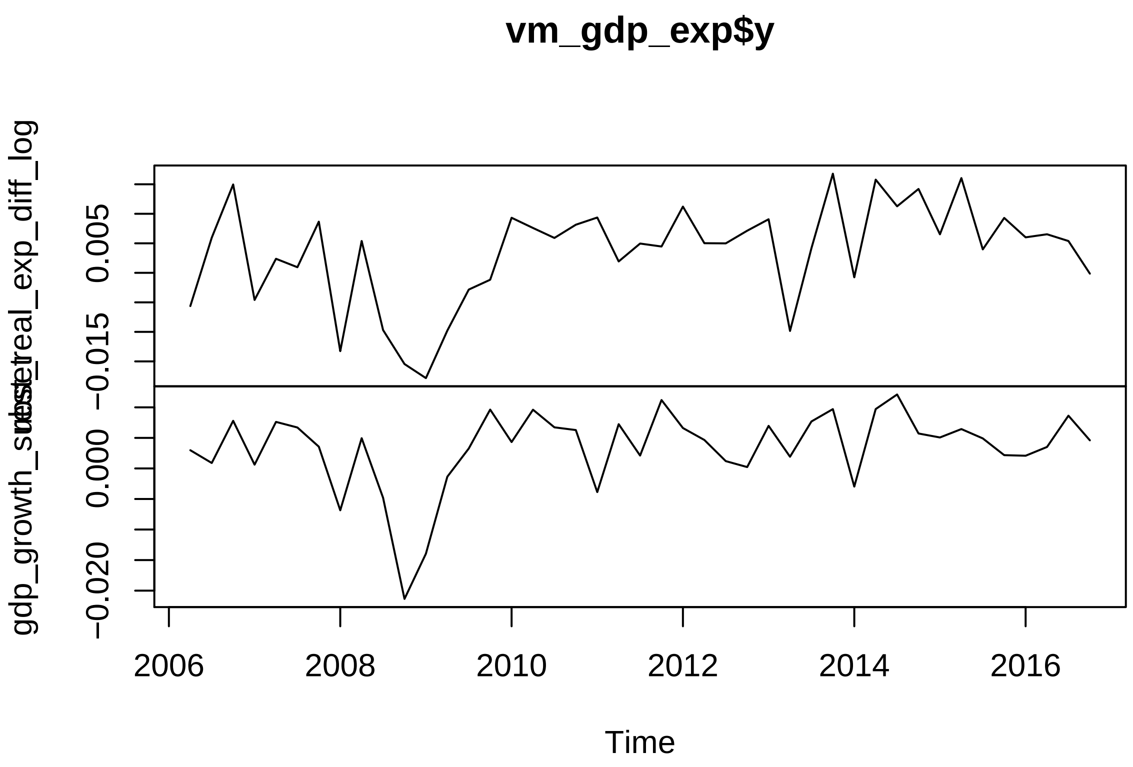
```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 0.69776, Truncation lag parameter = 1, p-value =
## 0.01375
##
```

```
##
##  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")
```



```
# 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)
```



```r
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 + g
##
##                             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
```

```r
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
```

```r
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
```

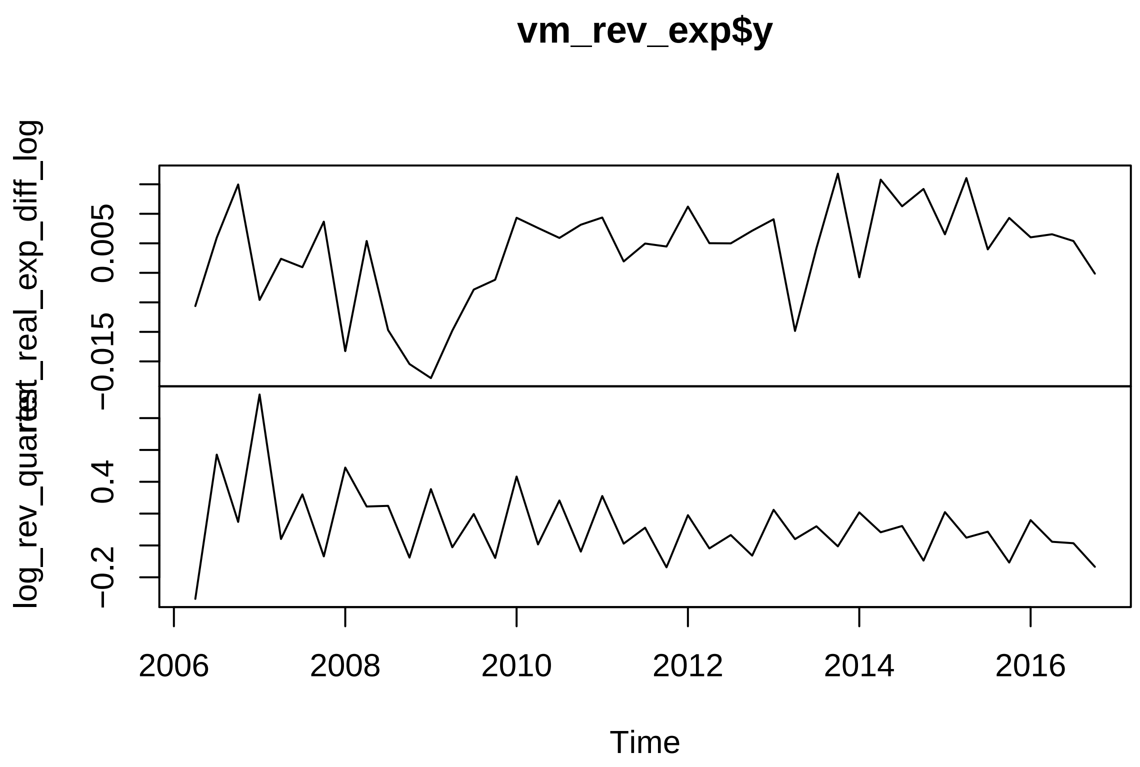
```r
rec_exp_diff_log_dummy = add_recession_dummy(rest_exp_dates_diff)
lm_rest_real_exp_diff_log = lm(rest_real_exp_diff_log ~ rec_exp_diff_log_dummy)
summary(lm_rest_real_exp_diff_log)
```

```
##
## 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

```r
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)
```



```r
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_re
##
##                            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
```

```r
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
```

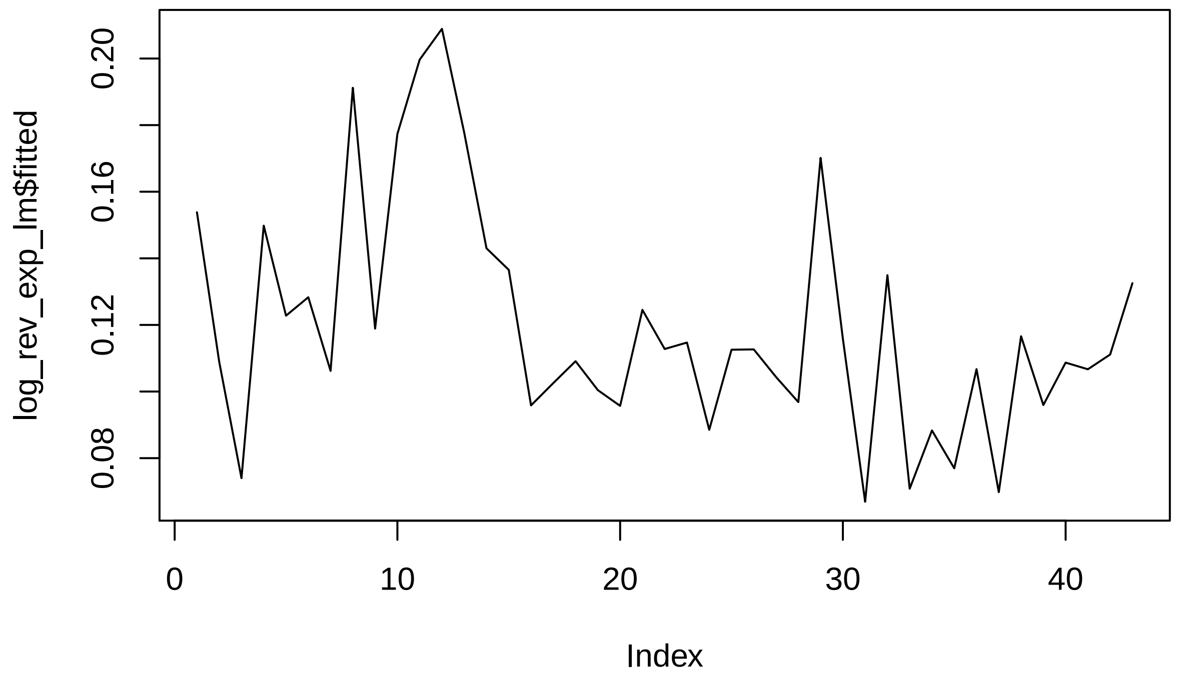
```r
grangertest(log_rev_quarter ~ rest_real_exp_diff_log, order = select$select[1])
```

```
## 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
```

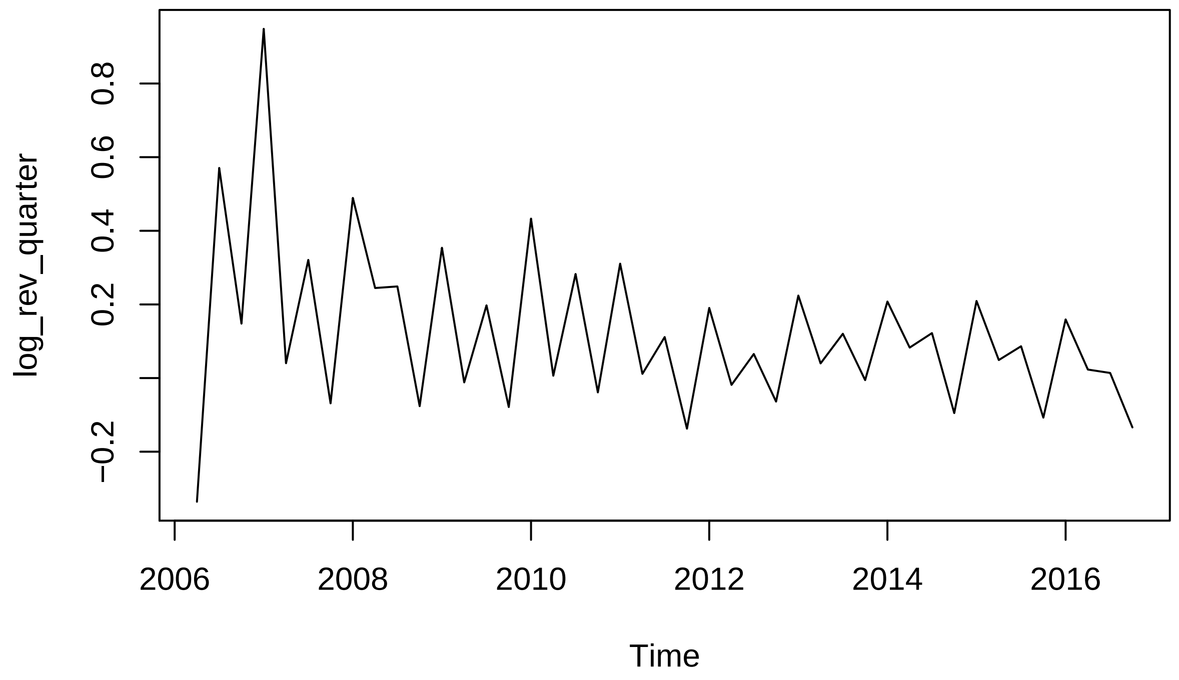
```r
# looks like log_rev_quarter is granger caused by
# rest_real_exp_diff_log
log_rev_exp_lm = lm(log_rev_quarter ~ rec_dummy_rev_growth_q +
    rest_real_exp_diff_log)
summary(log_rev_exp_lm)
```

```
##
## Call:
## lm(formula = log_rev_quarter ~ rec_dummy_rev_growth_q + rest_real_exp_diff_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48972 -0.13521 -0.02083  0.10232  0.79866
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.131999   0.047314   2.790  0.00804 **
## rec_dummy_rev_growth_q  0.007791   0.114349   0.068  0.94602
## rest_real_exp_diff_log -3.875810   5.061899  -0.766  0.44836
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2255 on 40 degrees of freedom
## Multiple R-squared:  0.02395,    Adjusted R-squared:  -0.02485
## F-statistic: 0.4908 on 2 and 40 DF,  p-value: 0.6158
```

```r
plot(log_rev_exp_lm$fitted, type = "l")
```



```
plot(log_rev_quarter)
```

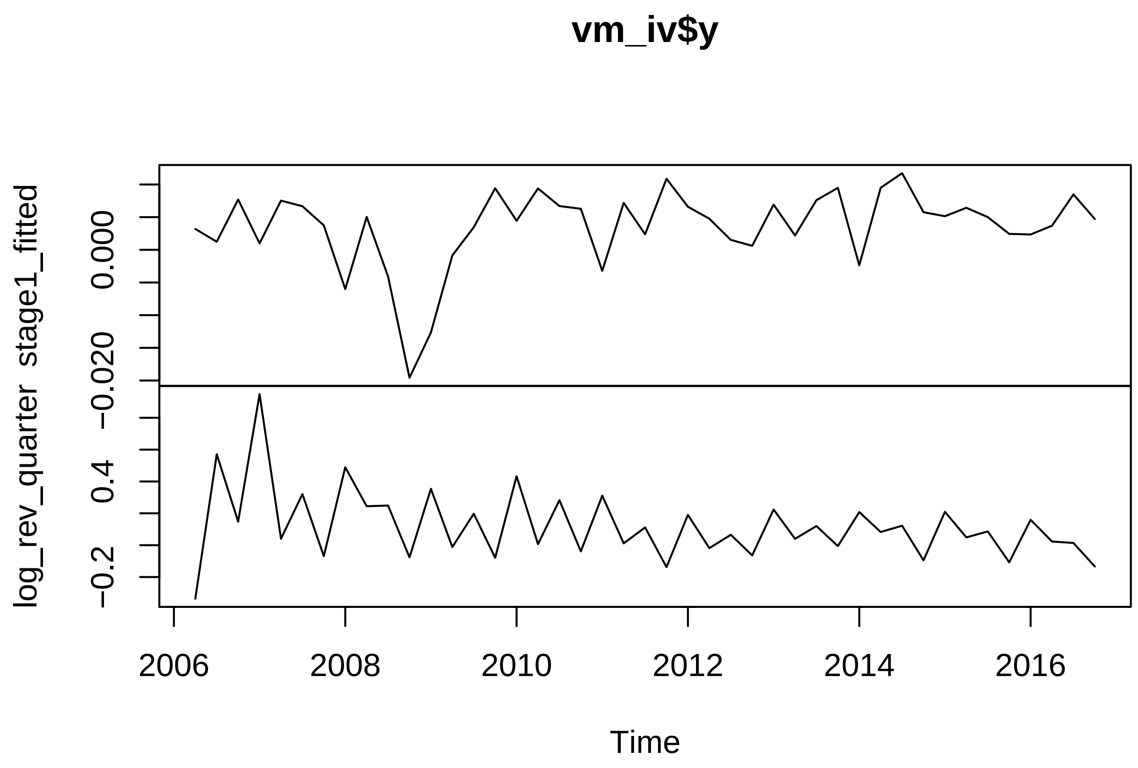


```
# gdp IV

# two stage regression
stage1 = lm(rest_real_exp_diff_log ~ gdp_growth_subset)
summary(stage1)
```

```
##
## Call:
## lm(formula = rest_real_exp_diff_log ~ gdp_growth_subset)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0120421 -0.0052213  0.0003529  0.0046706  0.0125832
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.0004049  0.0010428   0.388      0.7
## gdp_growth_subset 0.9357058  0.1460001   6.409 1.13e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006036 on 41 degrees of freedom
## Multiple R-squared:  0.5005, Adjusted R-squared:  0.4883
## F-statistic: 41.07 on 1 and 41 DF,  p-value: 1.134e-07
```

```
stage1_fitted = fitted(stage1)
```

```
# var for possible iv
iv_combined = cbind(stage1_fitted, log_rev_quarter)
select = VARselect(iv_combined, lag.max = 4, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
vm_iv = VAR(iv_combined, select$select[1])
plot(vm_iv$y)
```



```
summary(vm_iv)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: stage1_fitted, log_rev_quarter
## Deterministic variables: const
## Sample size: 39
## Log Likelihood: 202.729
## Roots of the characteristic polynomial:
##  0.91 0.8766 0.8609 0.8609 0.7678 0.7678 0.6853 0.6853
## Call:
## VAR(y = iv_combined, p = select$select[1])
##
##
## Estimation results for equation stage1_fitted:
## ==============================================
## stage1_fitted = stage1_fitted.l1 + log_rev_quarter.l1 + stage1_fitted.l2 + log_rev_quarter.l2 + stag
##
```

```
##                     Estimate Std. Error t value Pr(>|t|)
## stage1_fitted.l1    0.487351   0.175025   2.784   0.0092 **
## log_rev_quarter.l1  0.007386   0.006431   1.148   0.2599
## stage1_fitted.l2   -0.140392   0.197013  -0.713   0.4816
## log_rev_quarter.l2 -0.005684   0.005587  -1.017   0.3171
## stage1_fitted.l3    0.203458   0.193332   1.052   0.3010
## log_rev_quarter.l3 -0.005723   0.005737  -0.998   0.3265
## stage1_fitted.l4   -0.226631   0.167025  -1.357   0.1849
## log_rev_quarter.l4 -0.007732   0.006001  -1.288   0.2074
## const               0.004077   0.001809   2.253   0.0317 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.005443 on 30 degrees of freedom
## Multiple R-Squared: 0.3944,  Adjusted R-squared: 0.2329
## F-statistic: 2.442 on 8 and 30 DF,  p-value: 0.03638
##
##
## Estimation results for equation log_rev_quarter:
## ================================================
## log_rev_quarter = stage1_fitted.l1 + log_rev_quarter.l1 + stage1_fitted.l2 + log_rev_quarter.l2 + sta
##
##                    Estimate Std. Error t value Pr(>|t|)
## stage1_fitted.l1   -3.07255    2.48728  -1.235  0.22630
## log_rev_quarter.l1  0.23450    0.09139   2.566  0.01553 *
## stage1_fitted.l2    4.00838    2.79976   1.432  0.16257
## log_rev_quarter.l2  0.11229    0.07940   1.414  0.16757
## stage1_fitted.l3    3.81243    2.74744   1.388  0.17547
## log_rev_quarter.l3 -0.24877    0.08153  -3.051  0.00474 **
## stage1_fitted.l4   -6.21185    2.37360  -2.617  0.01376 *
## log_rev_quarter.l4  0.51038    0.08528   5.985 1.46e-06 ***
## const               0.02798    0.02571   1.088  0.28519
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.07735 on 30 degrees of freedom
## Multiple R-Squared: 0.8143,  Adjusted R-squared: 0.7648
## F-statistic: 16.44 on 8 and 30 DF,  p-value: 4.95e-09
##
##
##
## Covariance matrix of residuals:
##                 stage1_fitted log_rev_quarter
## stage1_fitted       2.962e-05      -1.602e-05
## log_rev_quarter    -1.602e-05       5.982e-03
##
## Correlation matrix of residuals:
##                 stage1_fitted log_rev_quarter
## stage1_fitted         1.00000        -0.03807
## log_rev_quarter      -0.03807         1.00000
```

```
grangertest(stage1_fitted ~ log_rev_quarter, order = select$select[1])
```

```
## Granger causality test
##
## Model 1: stage1_fitted ~ Lags(stage1_fitted, 1:4) + Lags(log_rev_quarter, 1:4)
## Model 2: stage1_fitted ~ Lags(stage1_fitted, 1:4)
##   Res.Df Df      F Pr(>F)
## 1     30
## 2     34 -4 2.5951 0.0563 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
grangertest(log_rev_quarter ~ stage1_fitted, order = select$select[1])
```

```
## Granger causality test
##
## Model 1: log_rev_quarter ~ Lags(log_rev_quarter, 1:4) + Lags(stage1_fitted, 1:4)
## Model 2: log_rev_quarter ~ Lags(log_rev_quarter, 1:4)
##   Res.Df Df      F  Pr(>F)
## 1     30
## 2     34 -4 2.8927 0.03881 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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