Modeling Restaurant-Goer's Behavior in The Great Recession: An Econometric Case Study

Lester Pi

3/18/2017

Abstract

This is a case study into Yelp restaurant-goers' consumer behavior during The Great Recession based off data from the Yelp Dataset Challenge 9. By manipulating and transforming the dataset, I rebuilt the relevant data into an econometric framework. Combined with a very light semantic analysis, we are able to see how consumer behavior changed during The Great Recession. With this information and econometric models, we can effectively determine how restaurant-goers' behavior will change in the event of a future recession.

Introduction

Yelp is a platform where users can review businesses based off a star system, with one star being the lowest and five stars being the highest. Along with the review, users write their thoughts about it which typically include why they feel the business earned the score they gave it. Another key feature of Yelp is business information that includes attributes such as price, business type, and location.

The Great Recession hit in December 2007 and lasted until June 2009 and was related to the financial crisis of 2007-08 and subprime mortgage crisis of 2007-09. One of the key aspects from a consumer standpoint is the Consumer Price Index (CPI), which is an indexed measure of prices and purchasing power. According to the Federal Reserve Economic Data of St. Louis (FRED), the CPI for urban food and beverages increases during the most of the recession, but declines towards the end. In other words, purchasing power was weaker during most of the recession.

Motivation

I wanted to focus on the restaurant industry for three main reasons: Yelp is very well-known for their restaurant reviews, I majoritively use Yelp for restaurant reviews, and I have a personal interest in the food and restaurant world. The Great Recession sets the stage for a great case study in that it was recent enough to occur after Yelp's conception and this also allows to account for systematic time differences. It was also a very interesting recession since it had huge ramifications both domestically and worldwide.

If restaurant Yelpers' behavior during and around The Great Recession period can be modeled, then we can apply this model to a future recession. Depending on the results, this can have important insights for restaurants who are looking to survive, or perhaps even take advantage of, a recession.

Datasets

The following are the datasets which I used in this case study.

Yelp Dataset Challenge 9: Contains a selective subset of Yelp data covering reviews, users, businesses, tips, and check-ins. Core datasets that will be examined.

FRED GDP: U.S. Real GDP data pulled from FRED. Used to examine The Great Recession.

Yahoo Finance Yelp Stock: Yelp's adjusted closure stock price. Used to measure Yelp company success and performance.

BEA Restaurant Expenditures: Seasonally adjusted real restaurant expenditures from The BEA. Used to connect Yelp restaurant data with a generalized restaurant industry.

Data Work Oveview

The Yelp data came in large Json files that needed to be converted into R-workable dataframes. Once in a workable format, I explored the data and extracted the relevant information. Regular expressions were used to analyze large groups of text when only a small specific portion was needed. By using SQL queries, I subsetted the interesting data into the needed date ranges and the associated categories, such as by geography and business type.

The other data sets were pulled from their sources, either manually or through R, and formatted and transformed accordingly.

Most of the data is either transformed into growth rates, detrended and/or seasonally adjusted, or kept in original levels.

Challenges and Data Issues Addressed

1. Yelp Data has a disproportionate amount of observations during its early days as well as the latest month due to not having a complete month's worth of data. This can cause statistical insignificance and heteroskedasticity. In order to prevent this, I omitted some of the earliest and latest data.
2. Almost everything is a affected by endogeneity. I tried to prove or disprove what I thought could be a potential instrumental variable that could be used to reduce endogeneity.
3. The Yelp data is a Yelp-decided subset of their data. This can cause large selection bias. I examined the data and saw that it includes small, medium, and large cities alike. There is no way to obtain the unreleased portion of data.
4. Level sets vs. growth rates. Growth rates allow our data to be transformed into stationary (or nearly stationary) data. However, they do not always make intuitive sense for this case study. Therefore, I used the levels for creating linear models, but used growth rates for determining Granger causality.

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



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



remove_lists_from_df = function(df) {
    i = 1
    while (i <= length(df)) {
```

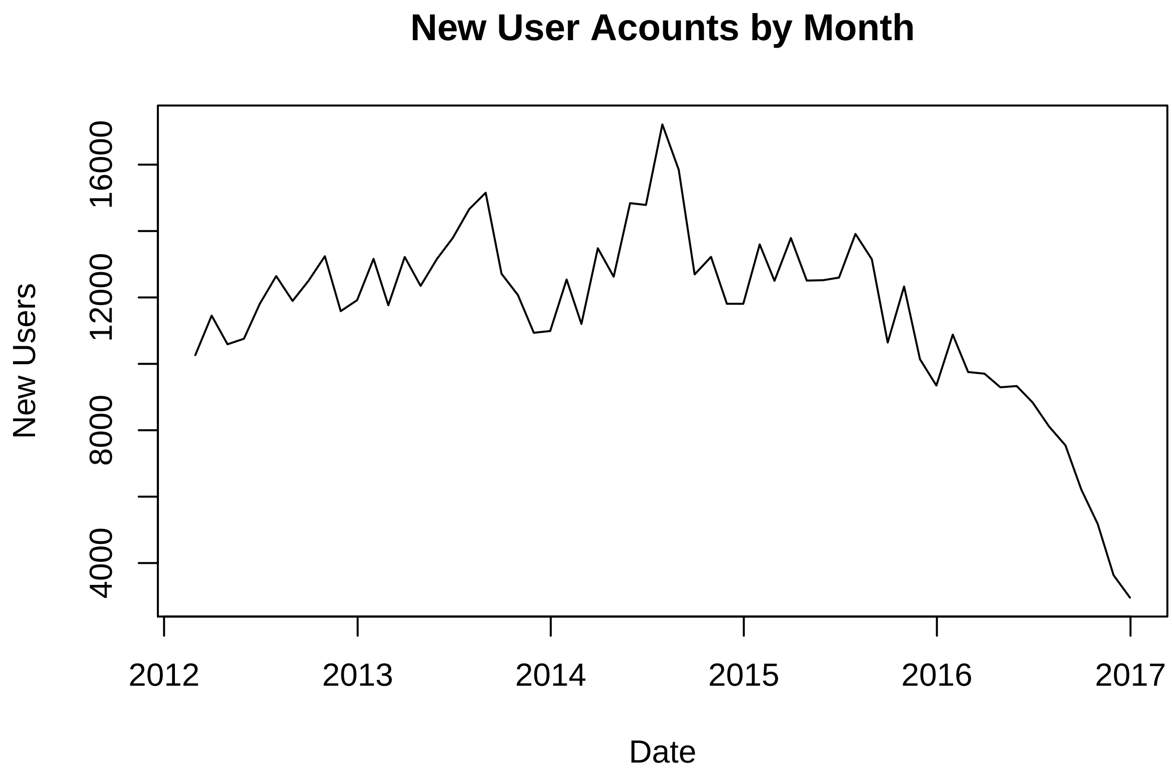
```
        if (class(df[, i]) == "list") {
            df[i] = sapply(df[, i], paste, collapse = "|")
        }
        i = i + 1
    }

    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)

}


descriptive_stats = function(lm_mod, name) {
    par(mfrow = c(3, 2))
    rec = recresid(resid(lm_mod) ~ 1, col = "skyblue3")
    plot(resid(lm_mod))
    truehist(resid(lm_mod))
    print(resettest(lm_mod))
    acf(resid(lm_mod))
    pacf(resid(lm_mod))
    jarque.bera.test(resid(lm_mod))
    plot(efp(resid(lm_mod) ~ 1, type = "Rec-CUSUM"))
    plot(rec, pch = 16)
    abline(h = 0, col = "red")
    title(paste("Descriptive Statistics:", name, sep = " "),
        outer = TRUE)
}

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

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

# dollar sign extractor
get_dollar_signs = function(s) {
```

```r
    temp = str_extract(s, "RestaurantsPriceRange2(.*?)[0-9]")
    return(substr(temp, nchar(temp), nchar(temp)))
}

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)
}
```

Does Yelp Performance Affect the Userbase?

First, I will explore the Yelp stock data to see if it is of any relevance in an attempt to handle any cases of endogeneity. In this case, the stock data will be a representation of the company performance. Intuitively, there is a chance it may be an instrumental variable such that the better the company is doing, the more they can advertise to, accumulate, and support a larger userbase.

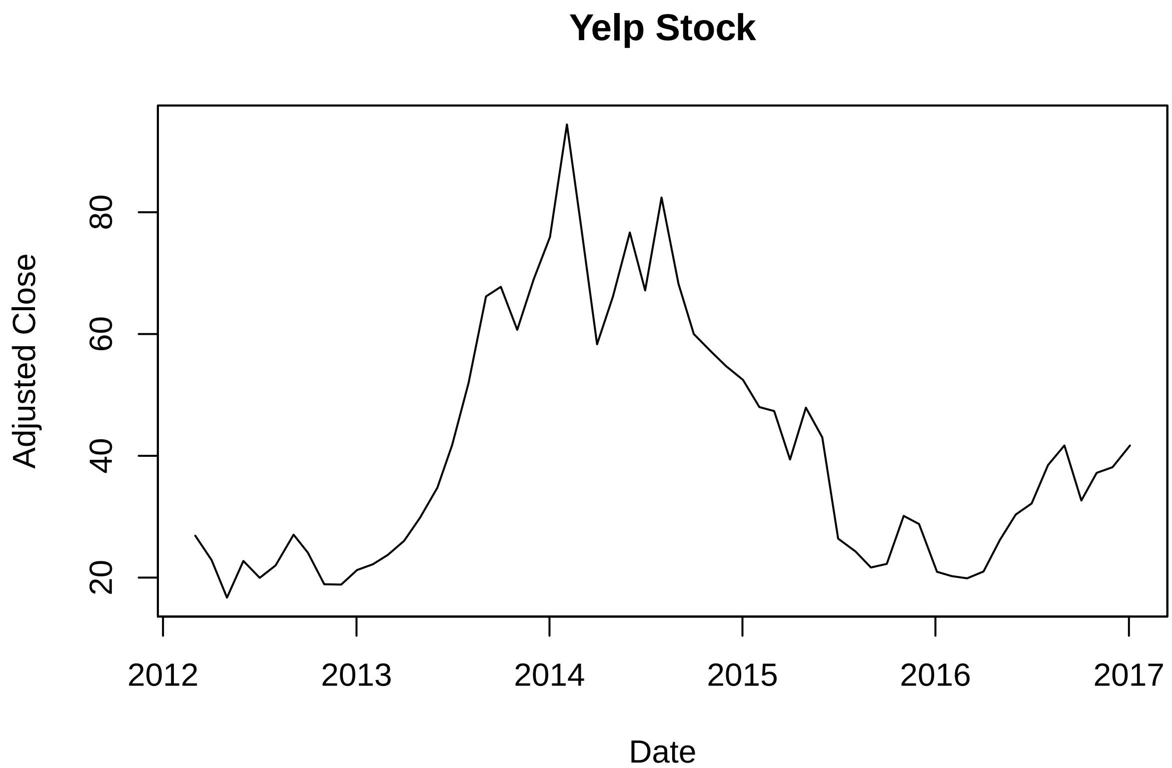
```r
plot(stock_users, type = "l", ylab = "New Users", xlab = "Date",
    main = "New User Acounts by Month")
```




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




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

```
##
##  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)))

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

```
##
##  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)
```

```
##
##  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
```

```
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, order = select$select[1])
```

```
## Granger causality test
##
## Model 1: ts_users ~ Lags(ts_users, 1:12) + Lags(ts_yelp, 1:12)
## Model 2: ts_users ~ Lags(ts_users, 1:12)
##   Res.Df  Df     F Pr(>F)
## 1     21
## 2     33 -12 1.071 0.4288
```

```
grangertest(ts_yelp ~ ts_users, order = select$select[1])
```

```
## Granger causality test
##
## Model 1: ts_yelp ~ Lags(ts_yelp, 1:12) + Lags(ts_users, 1:12)
## Model 2: ts_yelp ~ Lags(ts_yelp, 1:12)
##   Res.Df  Df      F Pr(>F)
```

```
## 1      21
## 2      33 -12 0.5091 0.8856
```

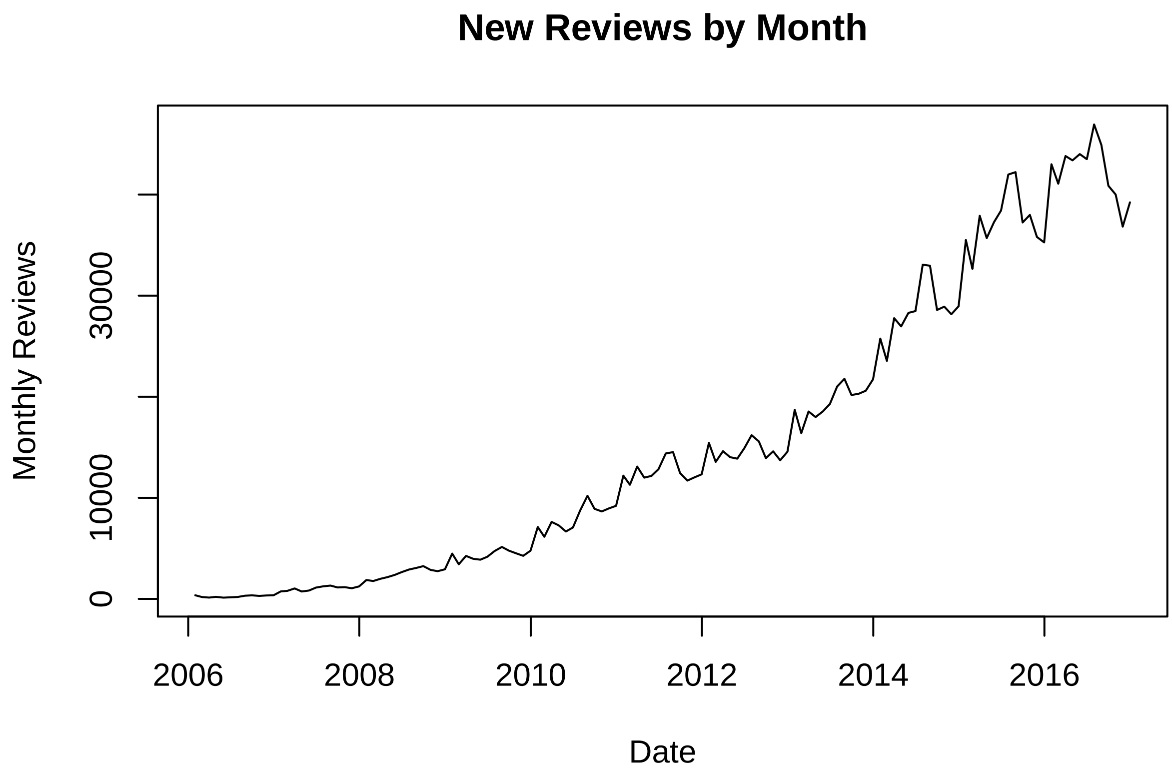
The number of new users do not have an effect on the stock value of Yelp and vice versa. This allows to eliminate the stock value as an instrumental variable.

```r
# remove user data to save memory

rm(user)
rm(subset_users)
rm(users_by_date_all)
rm(users_by_date)
rm(users_by_date_xts)
rm(users_df)
rm(stock_users)
```

Connecting Users and Reviews

We will attempt to connect new users with new reviews.

```r
# plot(review_counts_by_date$date,review_counts_by_date$`count(date)`,type='l')

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

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




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

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

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

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

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

```
##
```

```
## VAR Estimation Results:
## =========================
## Endogenous variables: log_rev_count, log_user_count
## Deterministic variables: const
## Sample size: 119
## Log Likelihood: 297.714
## Roots of the characteristic polynomial:
## 1.048 0.9737 0.9737 0.9716 0.9716  0.97  0.97 0.9687 0.9581 0.9581 0.9101 0.9101 0.8859 0.8859 0.857
## Call:
## VAR(y = rates_combined, p = select$select[1])
##
##
## Estimation results for equation log_rev_count:
## ==============================================
## log_rev_count = log_rev_count.l1 + log_user_count.l1 + log_rev_count.l2 + log_user_count.l2 + log_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_r
```

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

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

```
## Granger causality test
##
## Model 1: log_rev_count ~ Lags(log_rev_count, 1:12) + Lags(log_user_count, 1:12)
## Model 2: log_rev_count ~ Lags(log_rev_count, 1:12)
##   Res.Df  Df      F    Pr(>F)
```

```
## 1     94
## 2    106 -12 5.9487 1.318e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

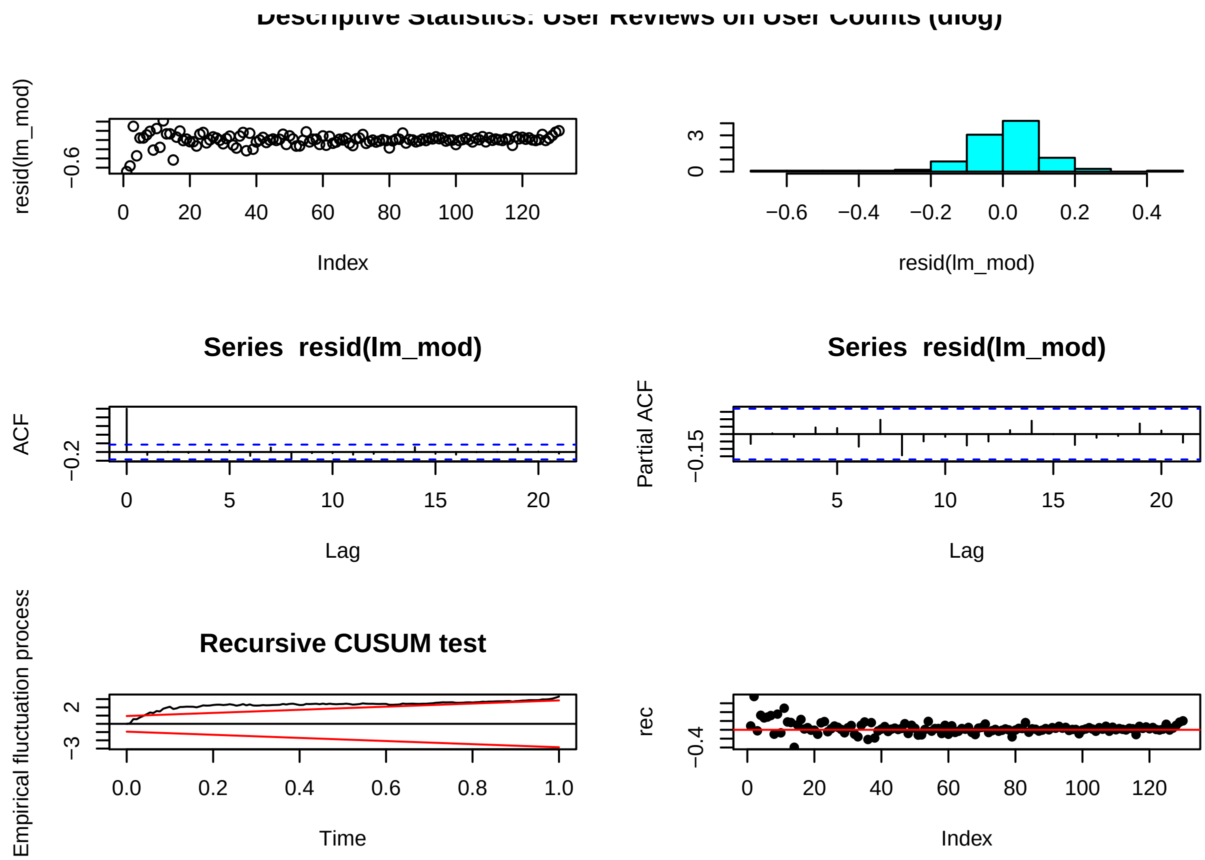
```
## Granger causality test
##
## Model 1: log_user_count ~ Lags(log_user_count, 1:12) + Lags(log_rev_count, 1:12)
## Model 2: log_user_count ~ Lags(log_user_count, 1:12)
##   Res.Df  Df      F   Pr(>F)
## 1     94
## 2    106 -12 3.4167 0.000354 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01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

```r
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
descriptive_stats(user_rev_lm, "User Reviews on User Counts (dlog)")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 4.2704, df1 = 2, df2 = 127, p-value = 0.01604
```




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

```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 0.27805, Truncation lag parameter = 2, p-value = 0.1
##
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -4.91, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
##
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -4.91, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

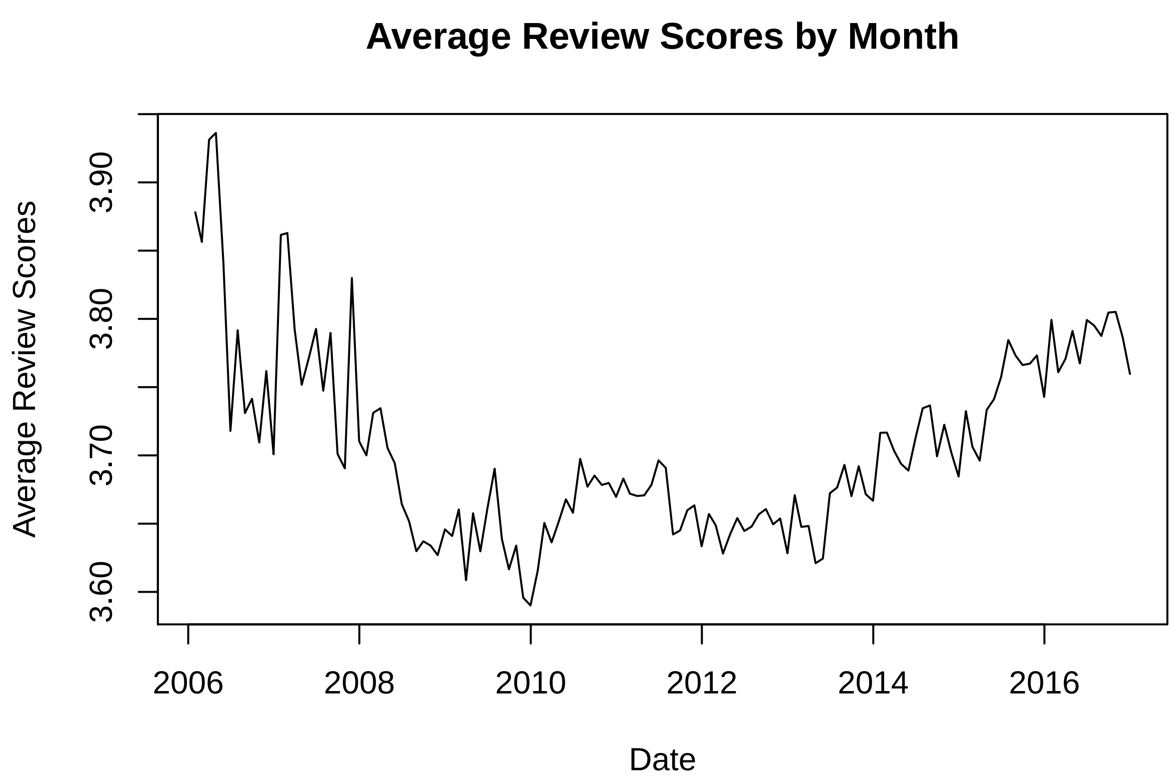
```r
# users and num reviews granger cause each other and are
# cointegrated, just use one
```

Through the VAR model results, Granger causality test, and cointegration test, we can conclude that users and reviews can be used interchangably. The error descriptive statistics do not look perfect, but we just want to see if reviews can be used in place of users. From here on out, we will focus on reviews since reviews contain more valuable information than the user data.

Do Review Scores Change?

By examining review scores, we can see if people become more or less critical during the recession, and many other insights.

```
# 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)

par(mfrow = c(1, 1))
plot(df_stars$date, df_stars$avg, type = "l", xlab = "Date",
    ylab = "Average Review Scores", main = "Average Review Scores by Month")
```



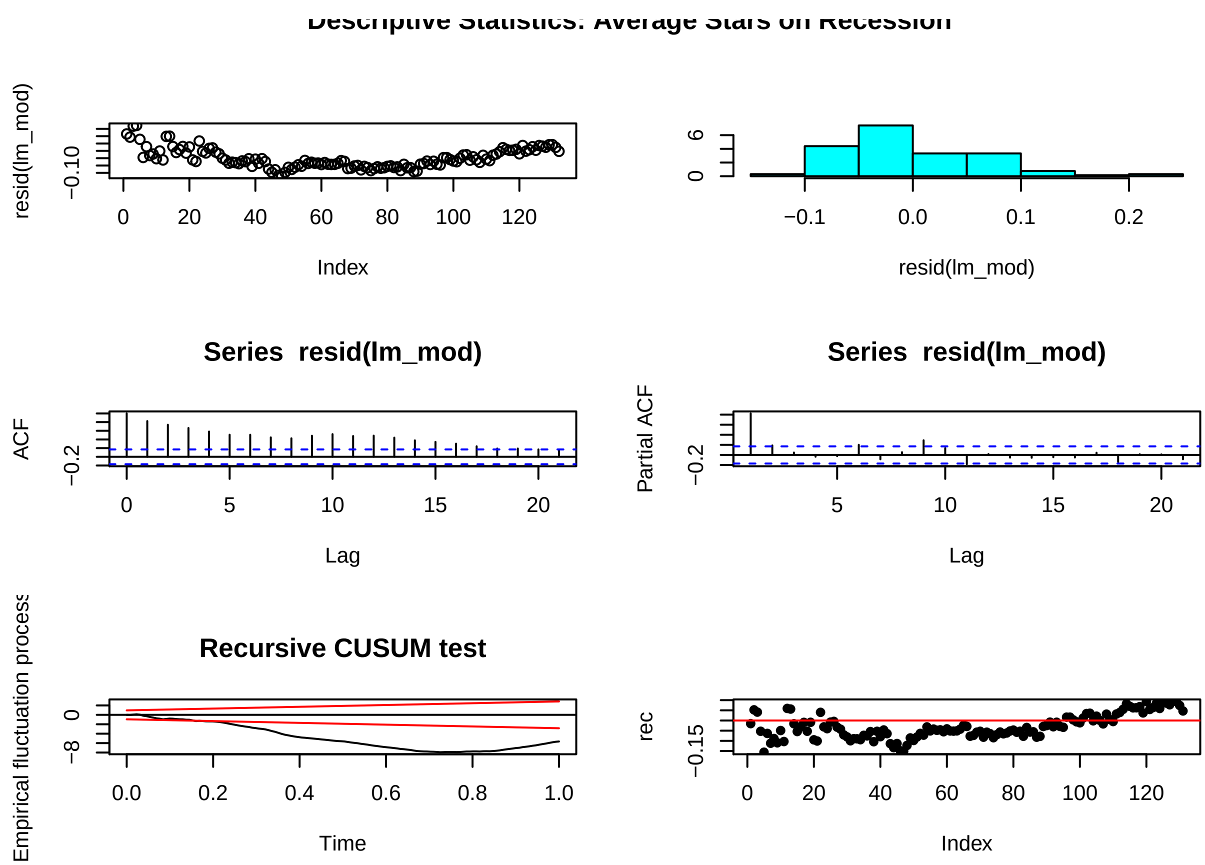

```
# evidence of recession in stars
stars_recession_dummy = add_recession_dummy(df_stars$date)
stars_avg = df_stars$avg
```

```r
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
descriptive_stats(stars_lm, "Average Stars on Recession")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 0, df1 = 2, df2 = 128, p-value = 1
```




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

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

test_stationary(df_stars_diff_log$avg)
```

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

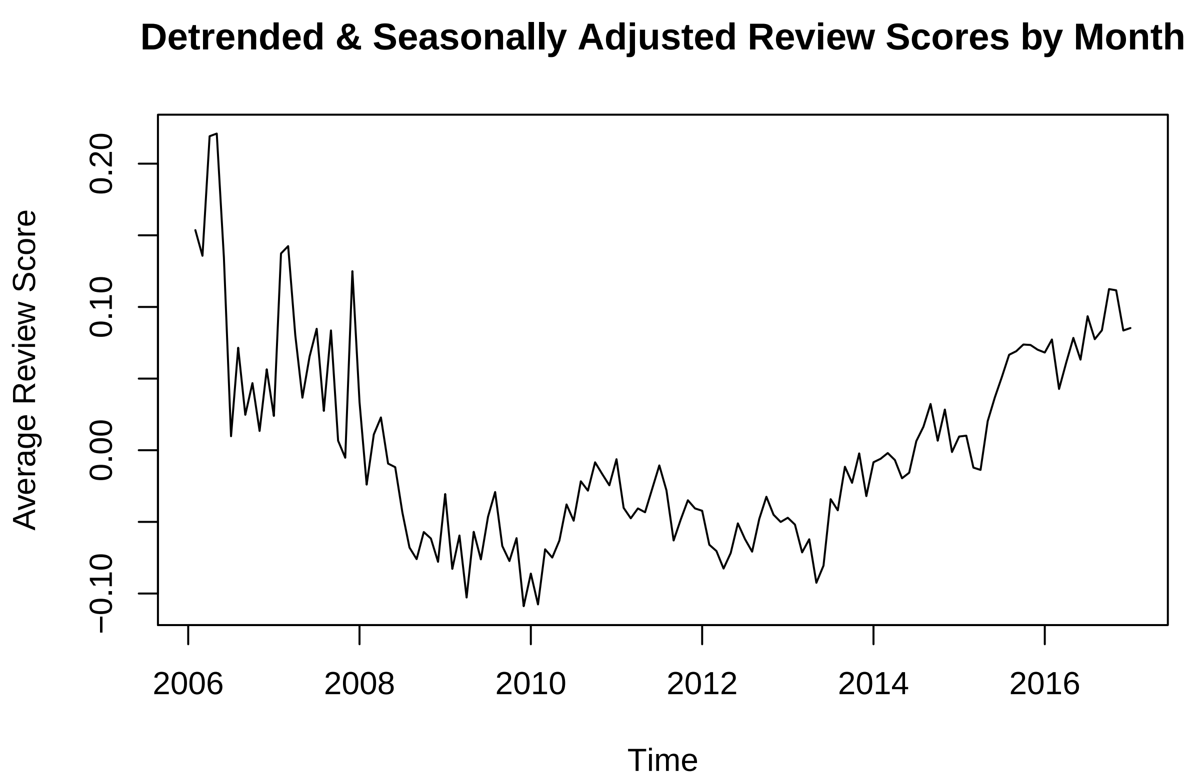
```
## data:  t
## KPSS Level = 0.20563, Truncation lag parameter = 2, p-value = 0.1
##
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -5.5779, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
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',main='Growth
# Rate of Review Scores by Month',xlab='Date',ylab='Review
# Scores Growth Rate') stars_diff_log_avg =
# df_stars_diff_log$avg stars_diff_log_lm =
# lm(stars_diff_log_avg~stars_recession_diff_log_dummy)
# summary(stars_diff_log_lm)
# descriptive_stats(stars_diff_log_lm)

# 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
```

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

detrend_stars = resid(stars_tslm)
par(mfrow = c(1, 1))
plot(detrend_stars, main = "Detrended & Seasonally Adjusted Review Scores by Month",
    ylab = "Average Review Score")
```



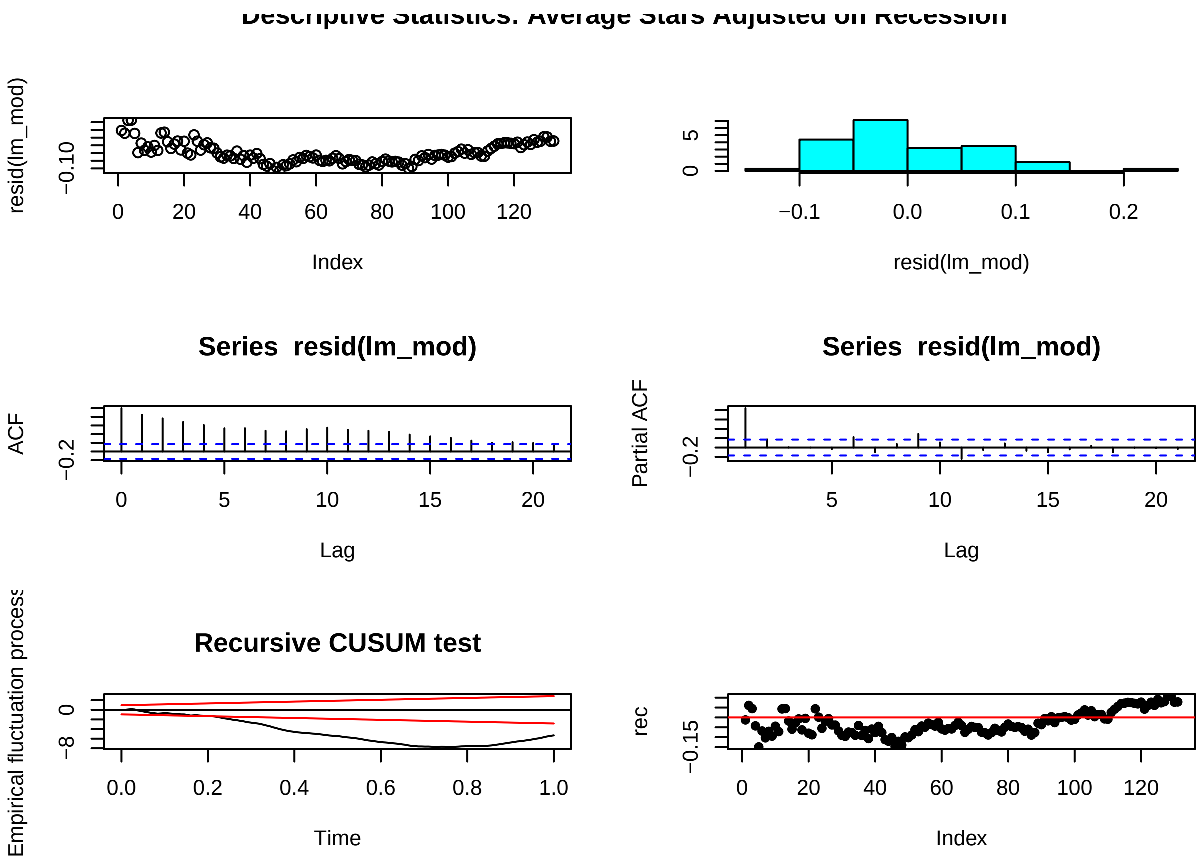

```r
detrend_stars_lm = lm(detrend_stars ~ stars_recession_dummy)
summary(detrend_stars_lm)
```

```
##
## 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
```

```r
descriptive_stats(detrend_stars_lm, "Average Stars Adjusted on Recession")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 0, df1 = 2, df2 = 128, p-value = 1
```



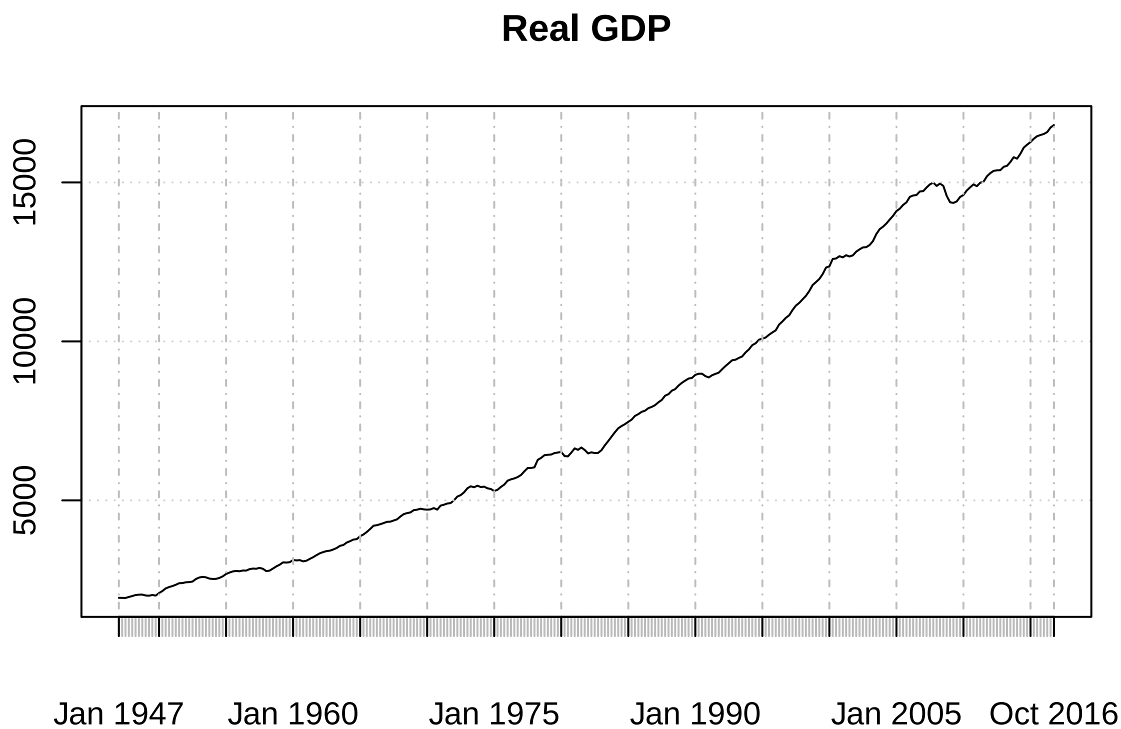
The regression results cannot be trusted because the error statistics are all over the place. No conclusion can be made yet.

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

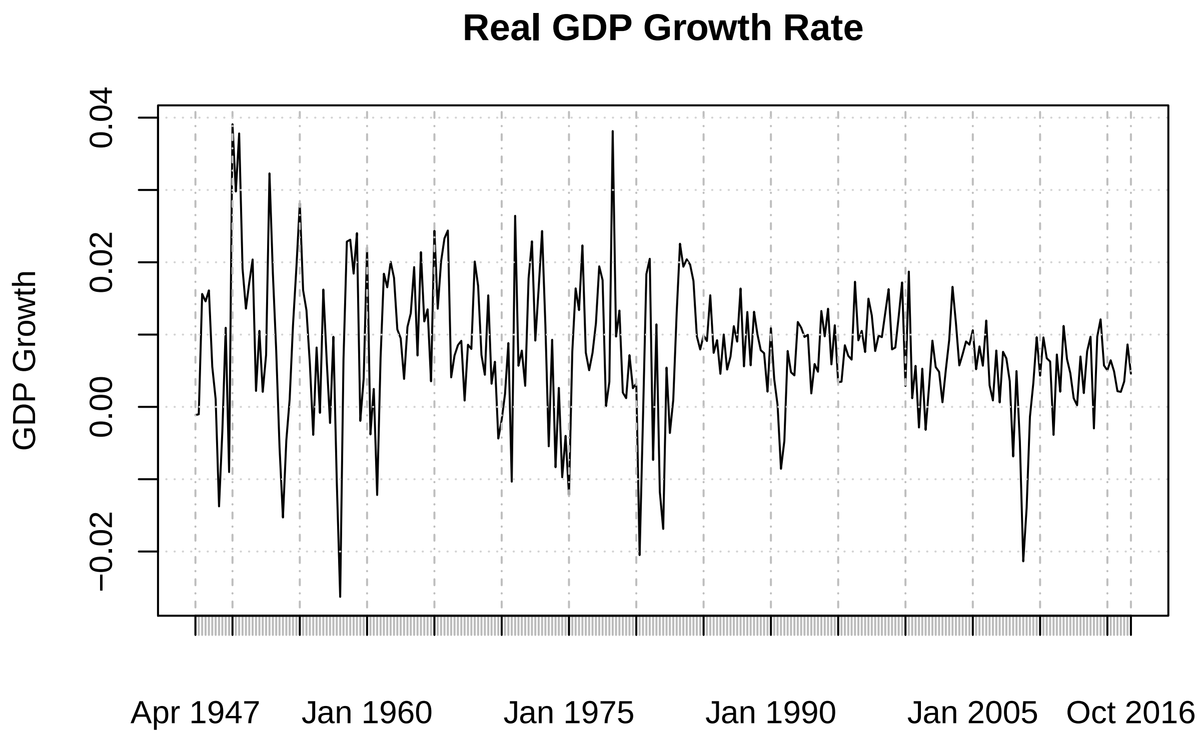
```
##     As of 0.4-0, 'getSymbols' uses env=parent.frame() and
##  auto.assign=TRUE by default.
##
##  This  behavior  will be  phased out in 0.5-0  when the call  will
##  default to use auto.assign=FALSE. getOption("getSymbols.env") and
##  getOptions("getSymbols.auto.assign") are now checked for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

```r
gdp = GDPC96

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

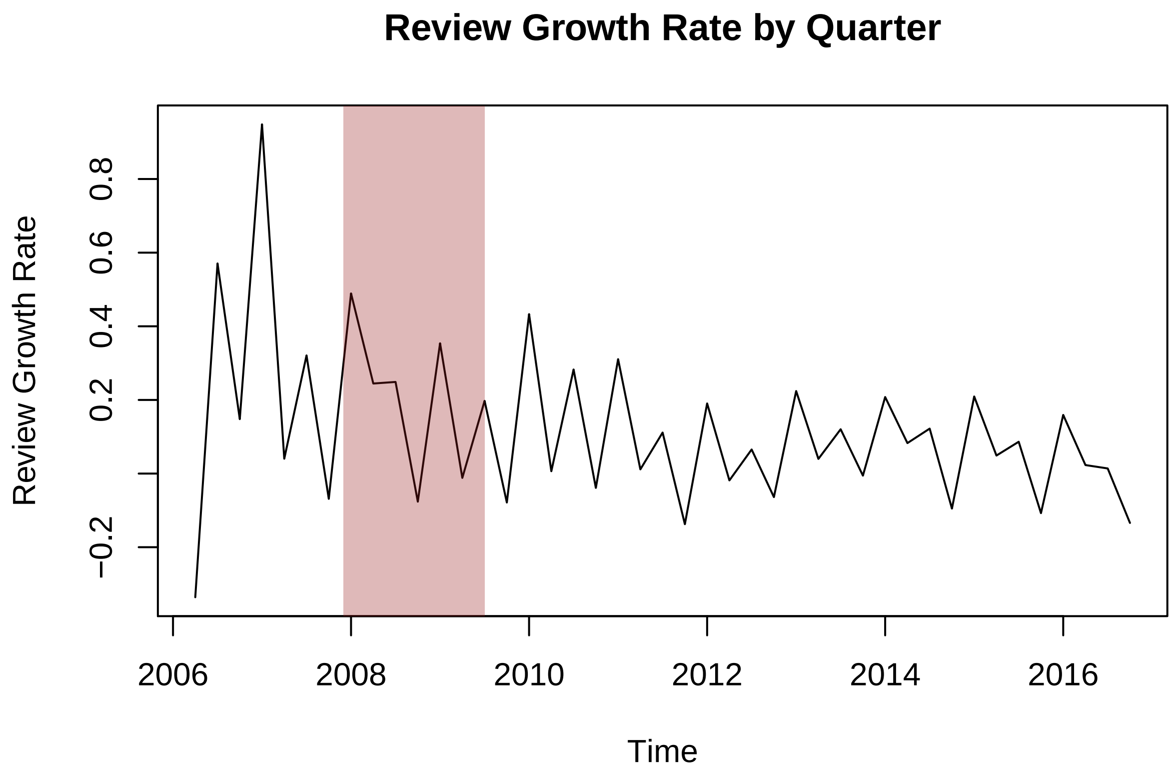



```r
gdp_growth = na.omit(diff(log(gdp)))
plot(gdp_growth, ylab = "GDP Growth", main = "Real GDP Growth Rate")
```




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


```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", ylab = "Review Growth Rate",
    main = "Review Growth Rate by Quarter")
rect(2007.9166667, -1, 2009.5, 1, col = rgb(red = 150/255, green = 25/255,
    blue = 25/255, alpha = 0.3), border = NA)
```




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

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

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

```
##
##                        Estimate Std. Error t value Pr(>|t|)
## log_rev_quarter.l1     0.41495    0.36251   1.145   0.2960
## gdp_growth_subset.l1  -0.69319    3.48625  -0.199   0.8490
## log_rev_quarter.l2     0.28843    0.37771   0.764   0.4740
## gdp_growth_subset.l2   4.10864    3.15933   1.300   0.2412
## log_rev_quarter.l3     0.27193    0.32528   0.836   0.4352
## gdp_growth_subset.l3   1.55990    3.26731   0.477   0.6500
## log_rev_quarter.l4     0.63886    0.20809   3.070   0.0219 *
## gdp_growth_subset.l4   1.46361    2.96026   0.494   0.6386
## log_rev_quarter.l5    -0.21532    0.23826  -0.904   0.4010
## gdp_growth_subset.l5  -0.43820    3.45264  -0.127   0.9032
## log_rev_quarter.l6    -0.16586    0.22996  -0.721   0.4979
## gdp_growth_subset.l6   4.69612    3.45845   1.358   0.2233
## log_rev_quarter.l7    -0.13165    0.24321  -0.541   0.6078
## gdp_growth_subset.l7  -4.04228    2.95912  -1.366   0.2209
## log_rev_quarter.l8     0.05838    0.24925   0.234   0.8226
## gdp_growth_subset.l8   5.04951    3.49119   1.446   0.1982
## log_rev_quarter.l9     0.11847    0.20277   0.584   0.5803
## gdp_growth_subset.l9   0.05217    3.36605   0.016   0.9881
## log_rev_quarter.l10    0.11208    0.13776   0.814   0.4470
## gdp_growth_subset.l10  6.34267    3.22213   1.968   0.0966 .
## log_rev_quarter.l11   -0.08093    0.15120  -0.535   0.6117
## gdp_growth_subset.l11 -1.75022    2.11397  -0.828   0.4394
## log_rev_quarter.l12    0.43781    0.17667   2.478   0.0479 *
## gdp_growth_subset.l12  2.53313    2.12131   1.194   0.2775
## const                 -0.16460    0.20382  -0.808   0.4502
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.04107 on 6 degrees of freedom
## Multiple R-Squared: 0.9823,  Adjusted R-squared: 0.9117
## F-statistic:  13.9 on 24 and 6 DF,  p-value: 0.001734
##
##
## Estimation results for equation gdp_growth_subset:
## ==================================================
## gdp_growth_subset = log_rev_quarter.l1 + gdp_growth_subset.l1 + log_rev_quarter.l2 + gdp_growth_subs
##
##                         Estimate Std. Error t value Pr(>|t|)
## log_rev_quarter.l1    -0.036465   0.038485  -0.947   0.3800
## gdp_growth_subset.l1  -0.473658   0.370109  -1.280   0.2479
## log_rev_quarter.l2    -0.042996   0.040098  -1.072   0.3248
## gdp_growth_subset.l2  -0.403692   0.335402  -1.204   0.2741
## log_rev_quarter.l3    -0.005267   0.034533  -0.153   0.8838
## gdp_growth_subset.l3  -0.410470   0.346866  -1.183   0.2814
## log_rev_quarter.l4    -0.034836   0.022091  -1.577   0.1659
## gdp_growth_subset.l4  -0.611076   0.314269  -1.944   0.0998 .
## log_rev_quarter.l5    -0.004935   0.025294  -0.195   0.8518
## gdp_growth_subset.l5  -0.259610   0.366540  -0.708   0.5053
## log_rev_quarter.l6     0.026056   0.024413   1.067   0.3269
## gdp_growth_subset.l6  -0.470368   0.367157  -1.281   0.2474
## log_rev_quarter.l7    -0.014837   0.025820  -0.575   0.5864
```

```
## gdp_growth_subset.l7  -0.179878   0.314147  -0.573   0.5877
## log_rev_quarter.l8    -0.028504   0.026461  -1.077   0.3228
## gdp_growth_subset.l8  -0.567262   0.370633  -1.531   0.1768
## log_rev_quarter.l9    -0.002516   0.021527  -0.117   0.9108
## gdp_growth_subset.l9  -0.410645   0.357347  -1.149   0.2942
## log_rev_quarter.l10   -0.021819   0.014625  -1.492   0.1863
## gdp_growth_subset.l10 -0.189002   0.342069  -0.553   0.6006
## log_rev_quarter.l11   -0.034597   0.016052  -2.155   0.0746 .
## gdp_growth_subset.l11  0.090608   0.224424   0.404   0.7004
## log_rev_quarter.l12   -0.014337   0.018756  -0.764   0.4736
## gdp_growth_subset.l12 -0.408396   0.225203  -1.813   0.1197
## const                  0.043626   0.021638   2.016   0.0904 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.00436 on 6 degrees of freedom
## Multiple R-Squared: 0.7602,  Adjusted R-squared: -0.1992
## F-statistic: 0.7924 on 24 and 6 DF,  p-value: 0.6886
##
##
##
## Covariance matrix of residuals:
##                   log_rev_quarter gdp_growth_subset
## log_rev_quarter         0.0016865         1.157e-04
## gdp_growth_subset       0.0001157         1.901e-05
##
## Correlation matrix of residuals:
##                   log_rev_quarter gdp_growth_subset
## log_rev_quarter            1.0000            0.6464
## gdp_growth_subset          0.6464            1.0000
```

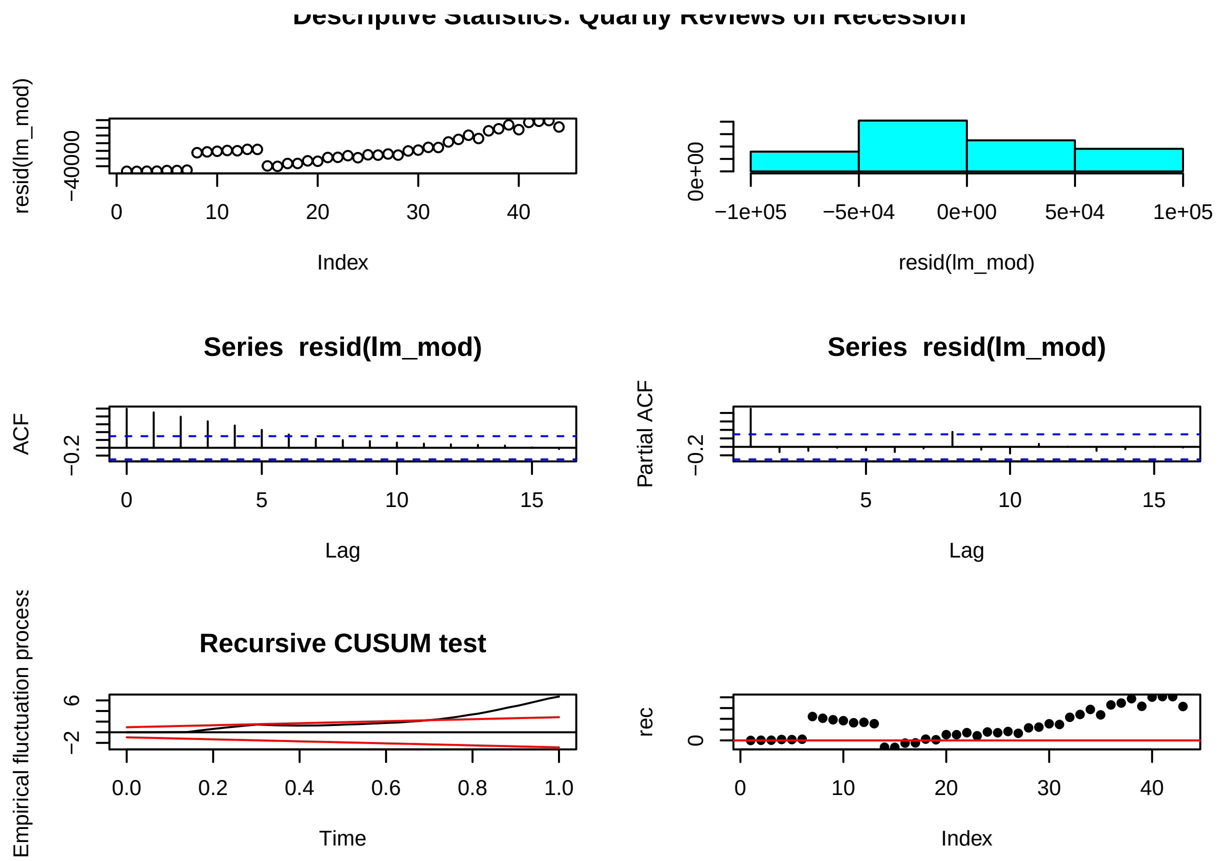
```
recession_dummy_reviews_q = add_recession_dummy(df_rev_quarter$date)

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
```

```r
descriptive_stats(reg_reviews, "Quartly Reviews on Recession")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 0, df1 = 2, df2 = 40, p-value = 1
```



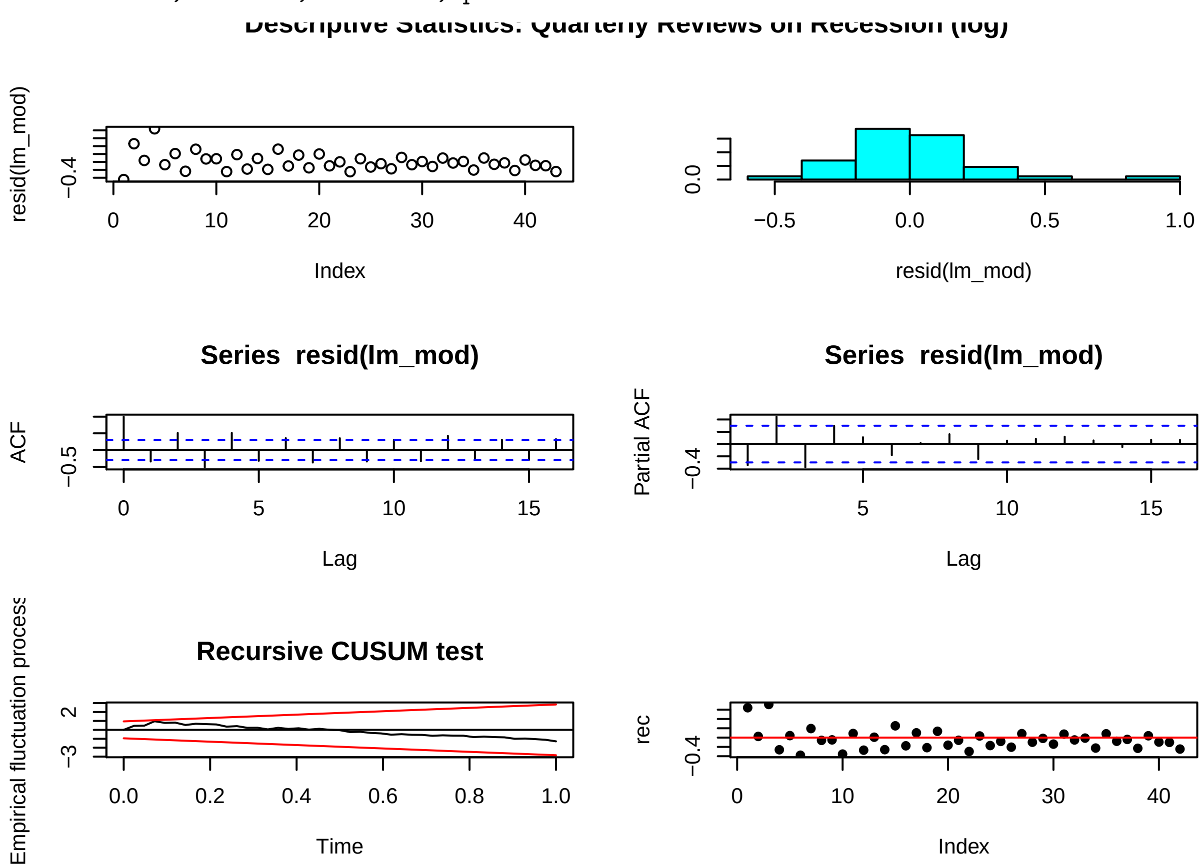

```r
# first reg is misleading because trend



# look at growth rates

rec_dummy_rev_growth_q = recession_dummy_reviews_q[2:length(recession_dummy_reviews_q)]

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

```
##
## Call:
## lm(formula = log_rev_quarter ~ rec_dummy_rev_growth_q)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44588 -0.13871 -0.02718  0.09270  0.83847
```

```
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.10999    0.03739   2.941  0.00535 **
## rec_dummy_rev_growth_q    0.05857    0.09268   0.632  0.53093
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2244 on 41 degrees of freedom
## Multiple R-squared:  0.009647,   Adjusted R-squared:  -0.01451
## F-statistic: 0.3994 on 1 and 41 DF,  p-value: 0.5309
```

```r
descriptive_stats(reg_log_reviews, "Quarterly Reviews on Recession (log)")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 0, df1 = 2, df2 = 39, p-value = 1
```



It looks that using the growth rates are better suited based off the error descriptive statistics, but we want to examine the social dynamics and not reviews as a whole.

Let's continue by splitting the restaurants by dollar signs ($).

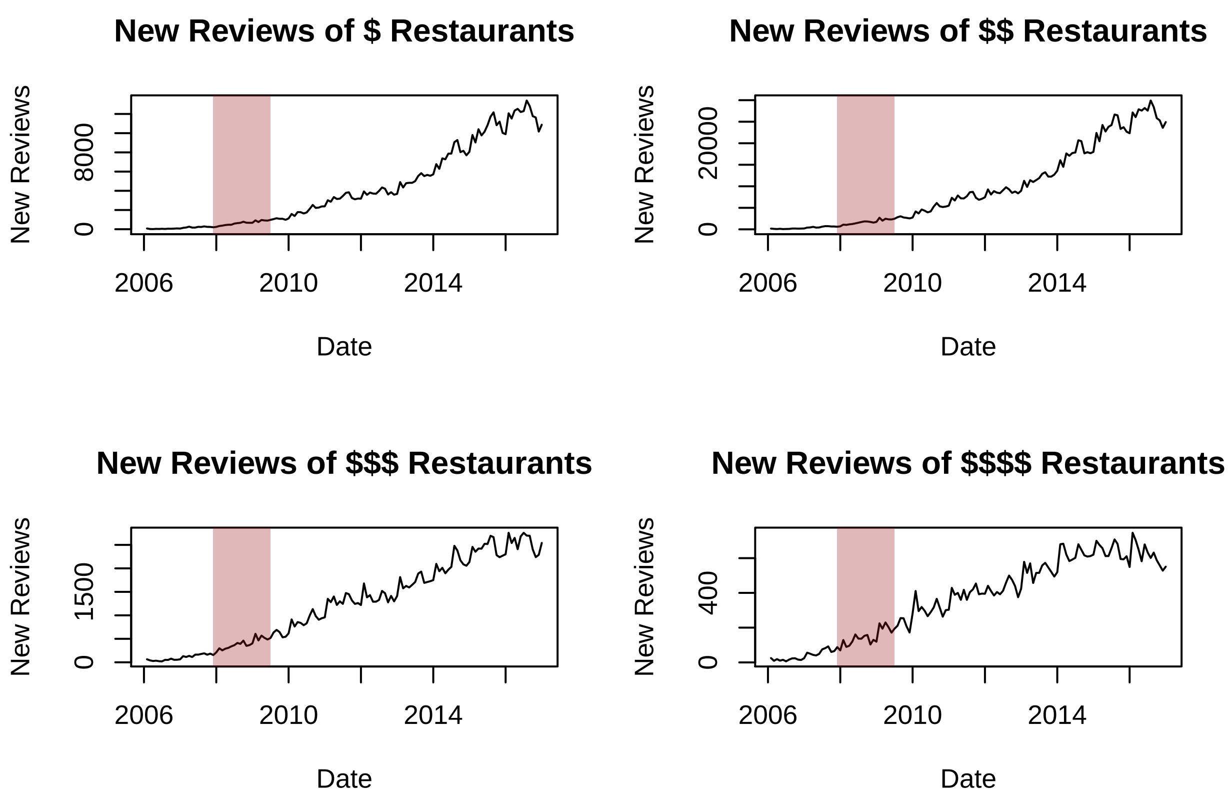
Examining Restaurants by Prices

How do people react to prices during a recession? To examine this, we will subset our data into 4 categories by price, one for each price level. This will be refered to as how many dollar signs ($) a business is.

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

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

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

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


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

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


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

```r
ts_dollar_4 = ts(df_dollars_4$coredata.df_d_4., start = c(2006,
    2), freq = 12)


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




```r
# dollars_1_dlog=diff(log(ts_dollar_1))
# dollars_2_dlog=diff(log(ts_dollar_2))
# dollars_3_dlog=diff(log(ts_dollar_3))
# dollars_4_dlog=diff(log(ts_dollar_4)) par(mfrow=c(2,2))
```

```r
# plot(dollars_1_dlog,xlab='Date',ylab='New Reviews Growth
# Rate', main='New Reviews Growth Rate of $ Restaurants')
# plot(dollars_2_dlog,xlab='Date',ylab='New Reviews Growth
# Rate', main='New Reviews Growth Rate of $$ Restaurants')
# plot(dollars_3_dlog,xlab='Date',ylab='New Reviews Growth
# Rate', main='New Reviews Growth Rate of $$$ Restaurants')
# plot(dollars_4_dlog,xlab='Date',ylab='New Reviews Growth
# Rate', main='New Reviews Growth Rate of $$$$ Restaurants')

# we can see highly seasonal

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

There is a strong and obvious trend along with seasonality. This is something that should be wiped out as much as possible to see the true effect of the recession.

```r
# seasonally adjust data and remove trend

par(mfrow = c(2, 2))

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

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

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

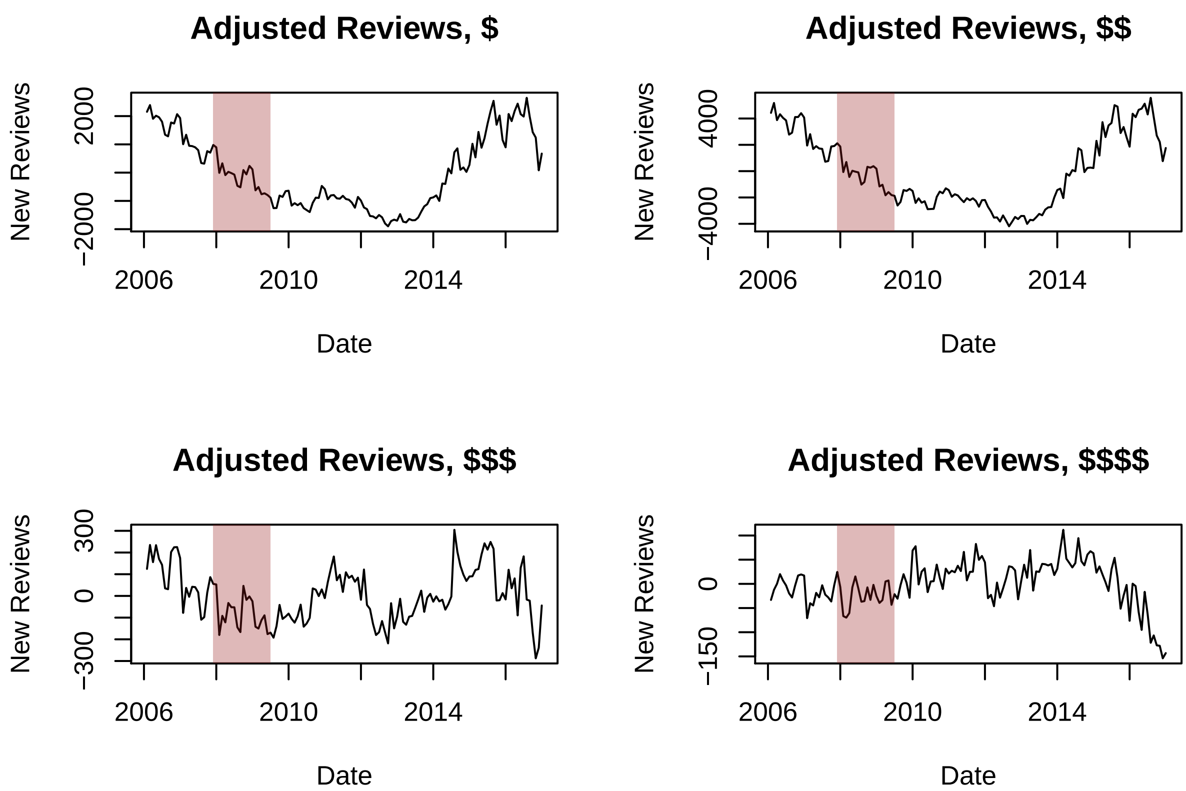
```r
summary(lm_d2_adj)
```

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

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

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

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



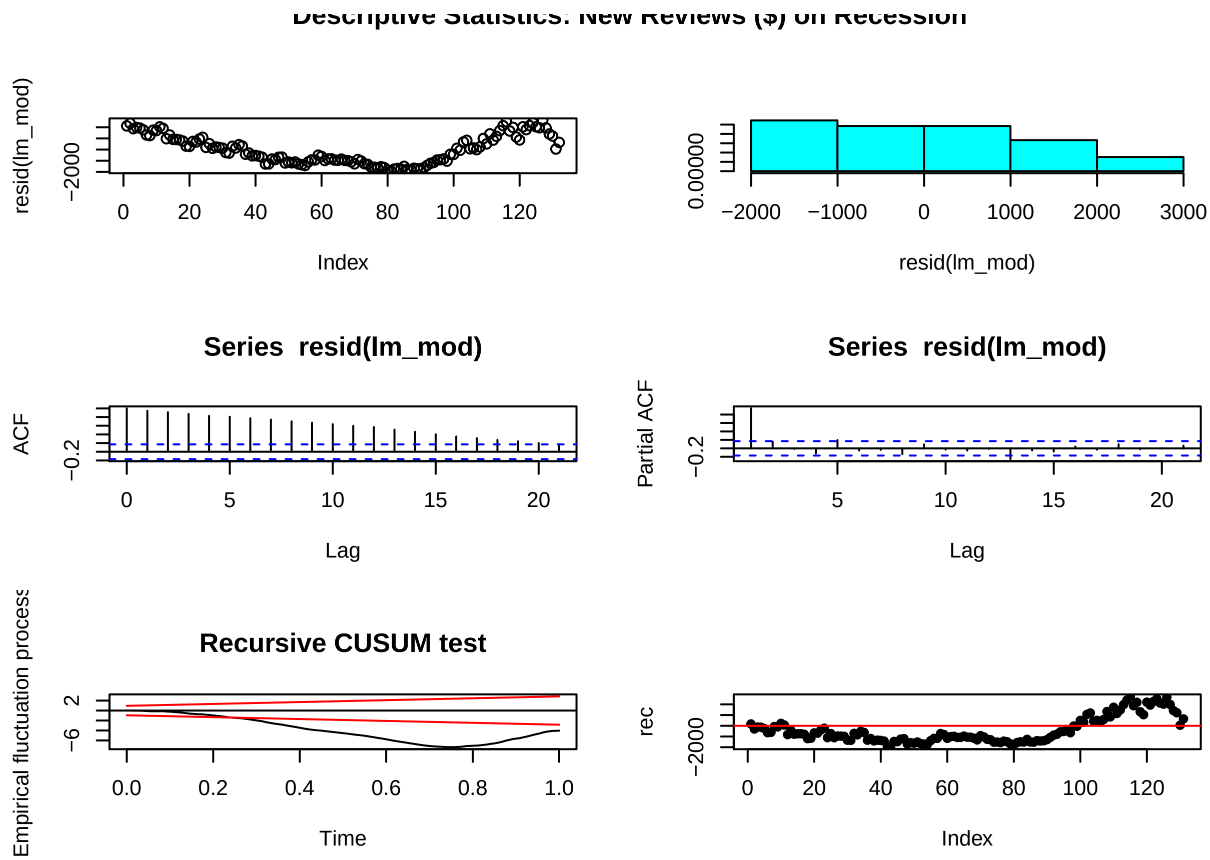

```r
lm_d4_adj = lm(tslm_d4_resid ~ recession_dummy_dollars_m)
summary(lm_d4_adj)
```

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

```r
descriptive_stats(lm_d1_adj, "New Reviews ($) on Recession")
```

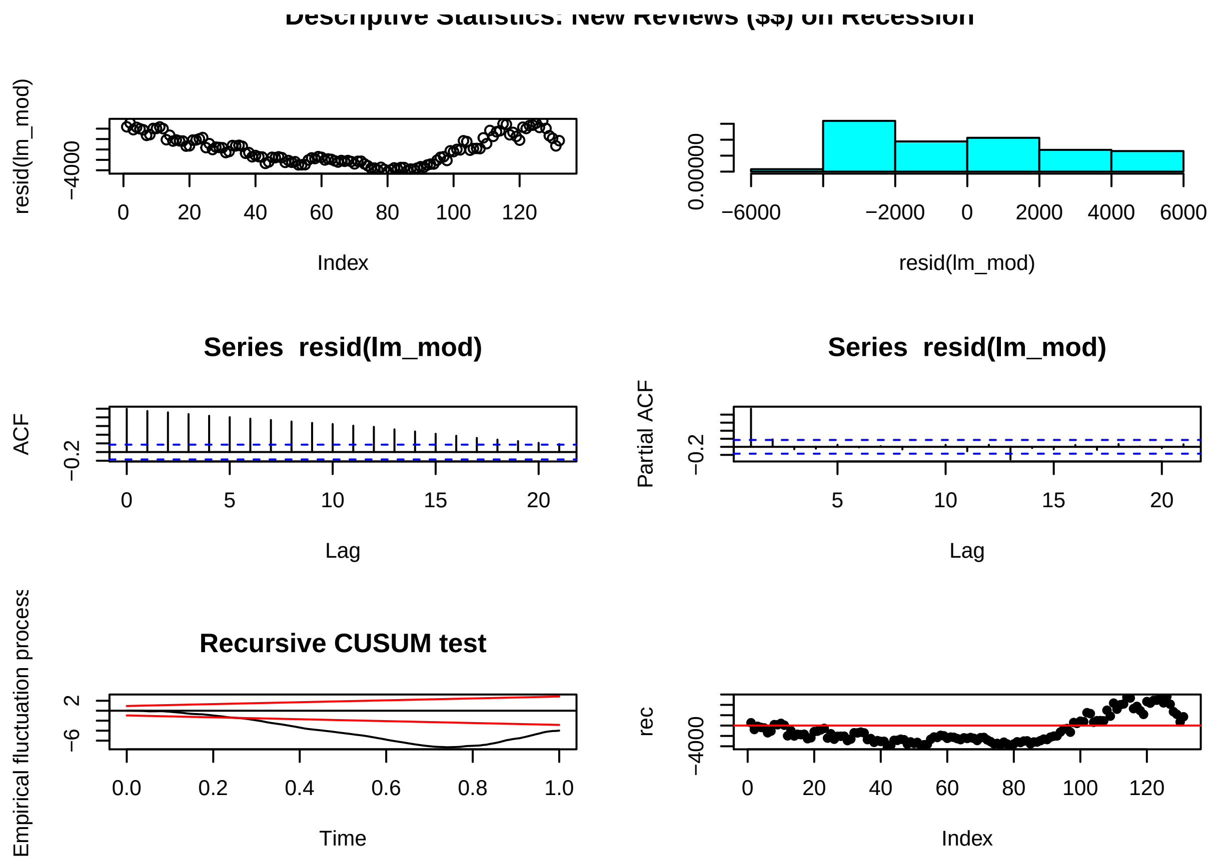
```
##
##  RESET test
##
```

```
## data:  lm_mod
## RESET = 0, df1 = 2, df2 = 128, p-value = 1
```



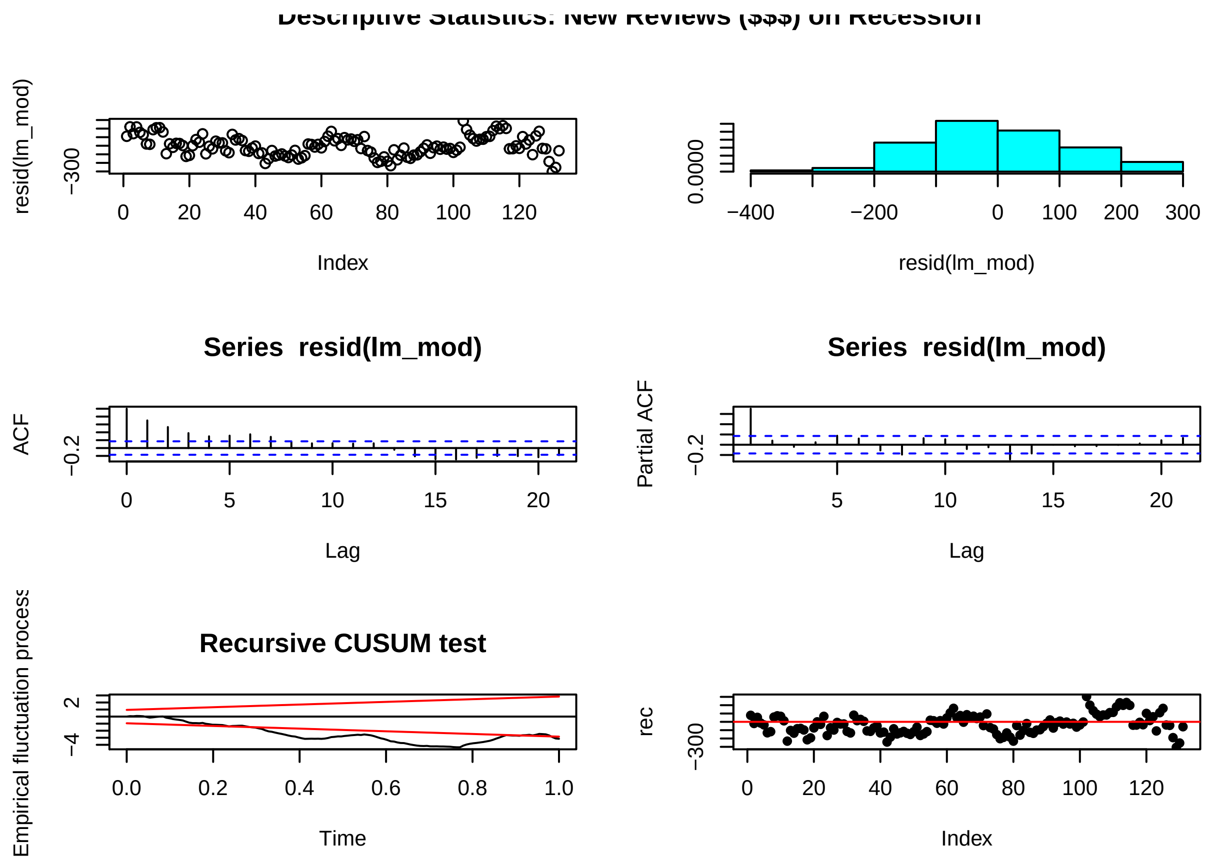

```r
descriptive_stats(lm_d2_adj, "New Reviews ($$) on Recession")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 0, df1 = 2, df2 = 128, p-value = 1
```



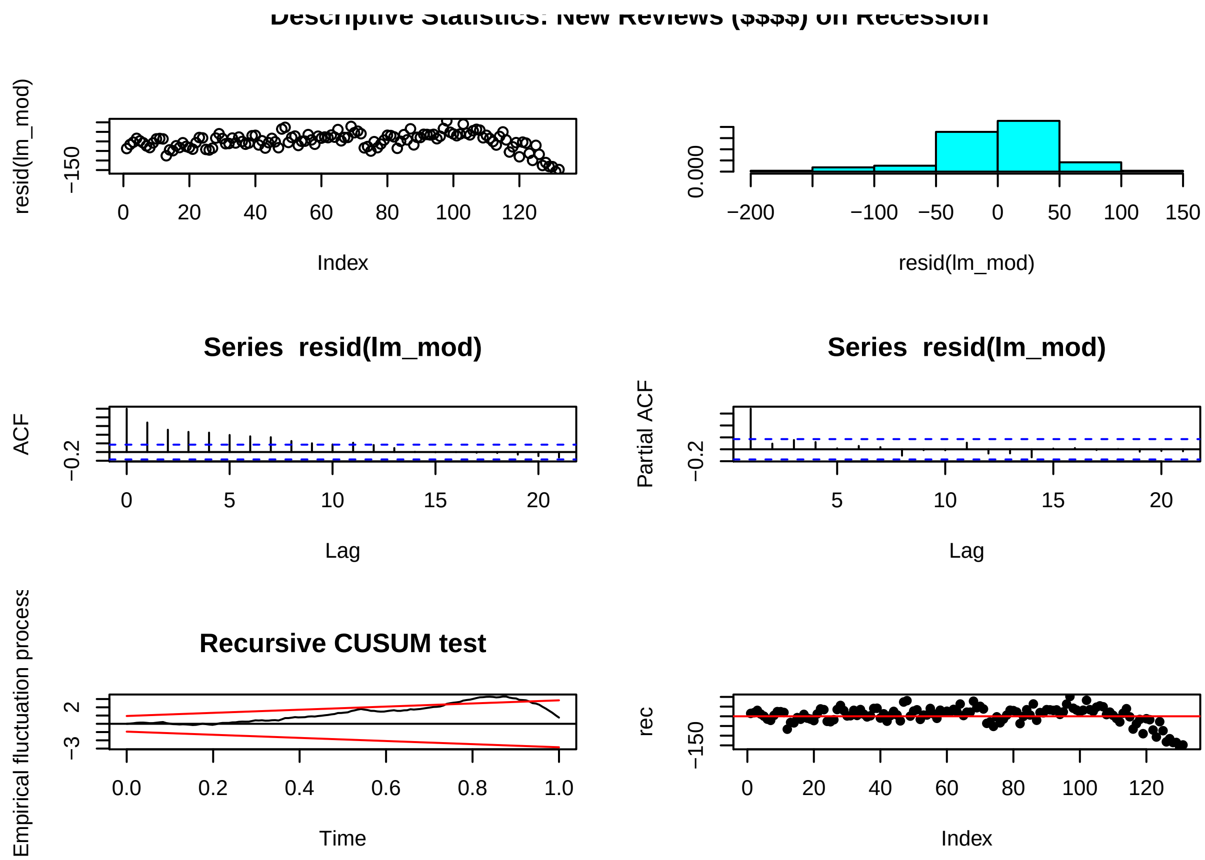

```r
descriptive_stats(lm_d3_adj, "New Reviews ($$$) on Recession")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 0, df1 = 2, df2 = 128, p-value = 1
```




```r
descriptive_stats(lm_d4_adj, "New Reviews ($$$$) on Recession")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 0, df1 = 2, df2 = 128, p-value = 1
```



The regression results and error descriptive statistics do not look too promising, but could be leading in the right direction.

Let's try to make it better.

Making a Better Model

By controlling for the aggegated number of reviews between dollar signs, we can eliminate the effect that a drop in aggregated reviews (again, by dollar signs) can potentially have.

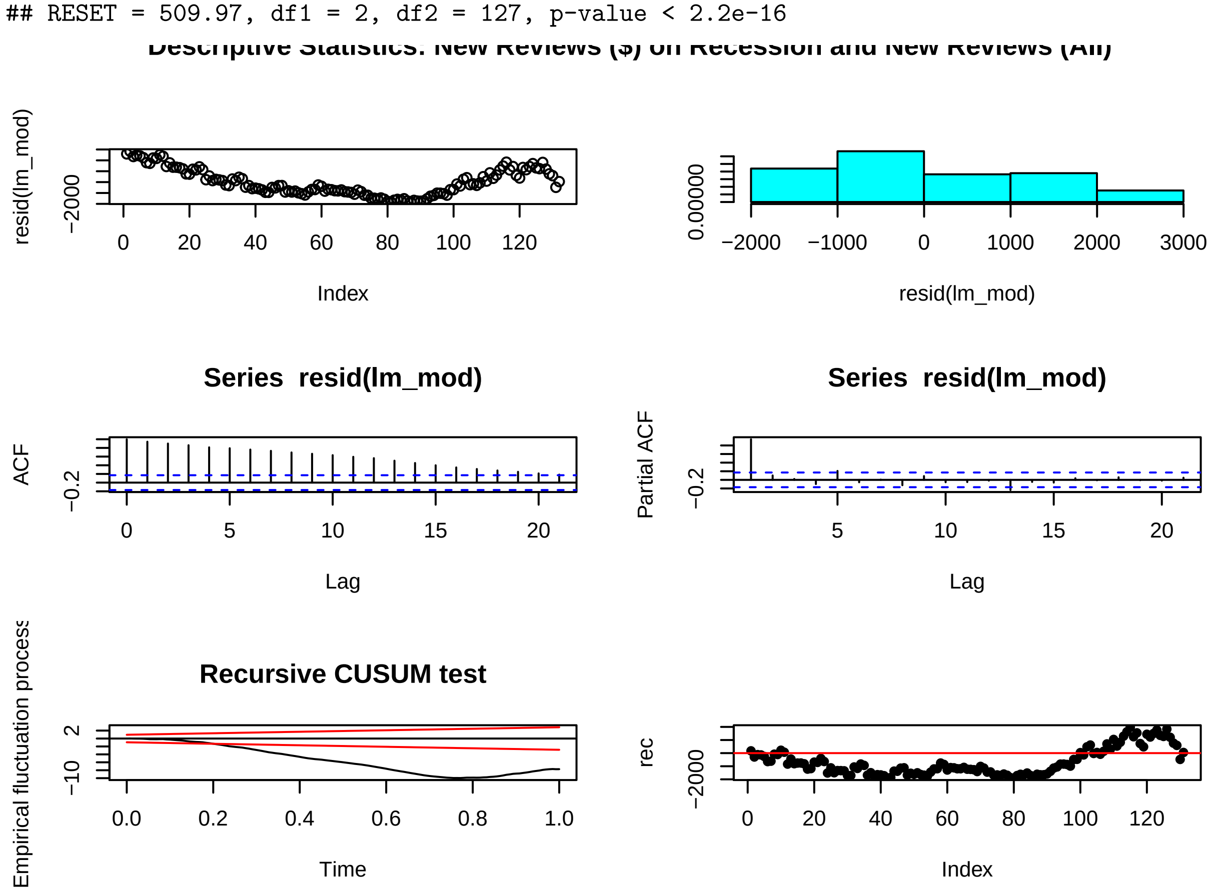
```
# levels
lm_d1_adj_rev = lm(tslm_d1_resid ~ recession_dummy_dollars_m +
    df_rev_count$coredata.df_rev_m.)
summary(lm_d1_adj_rev)
```

```
##
## 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
```

```
descriptive_stats(lm_d1_adj_rev, "New Reviews ($) on Recession and New Reviews (All)")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 509.97, df1 = 2, df2 = 127, p-value < 2.2e-16
```



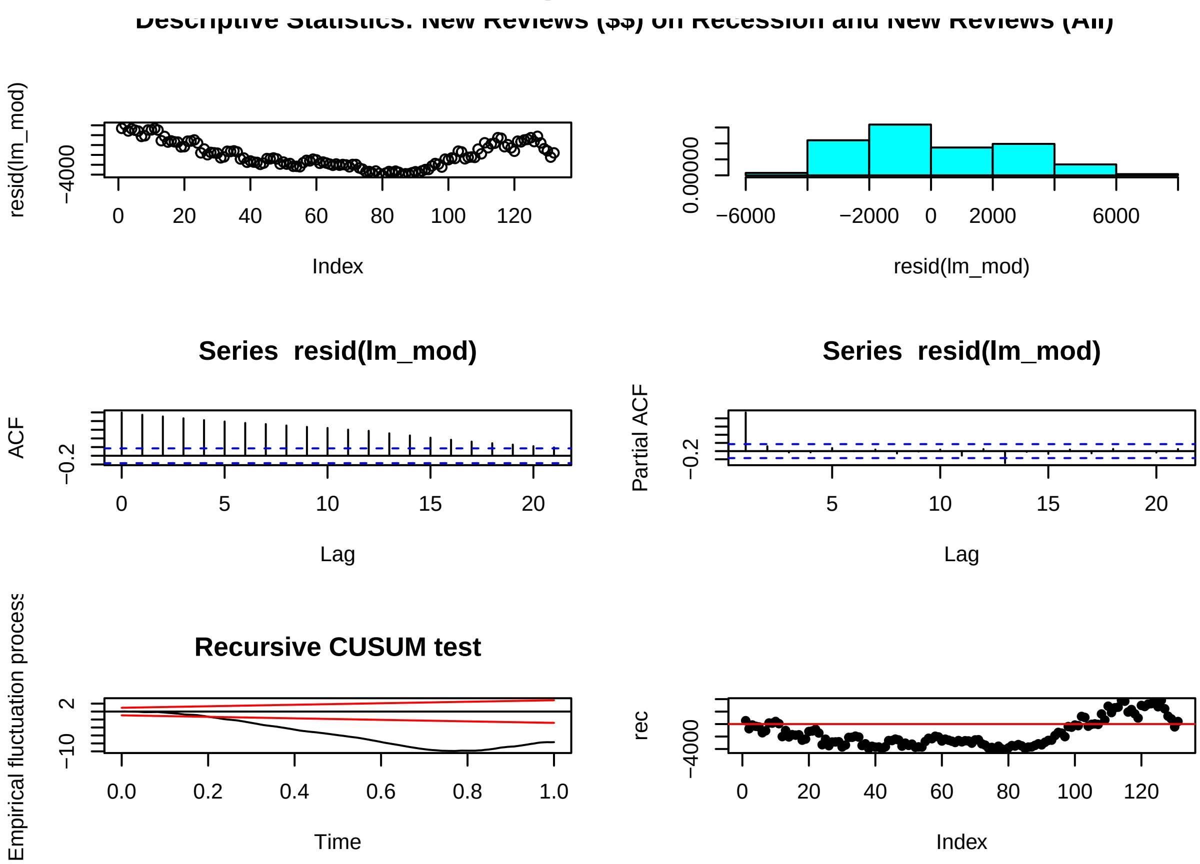

```
lm_d2_adj_rev = lm(tslm_d2_resid ~ recession_dummy_dollars_m +
    df_rev_count$coredata.df_rev_m.)
summary(lm_d2_adj_rev)
```

```
##
## 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
```

```
descriptive_stats(lm_d2_adj_rev, "New Reviews ($$) on Recession and New Reviews (All)")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 579.39, df1 = 2, df2 = 127, p-value < 2.2e-16
```



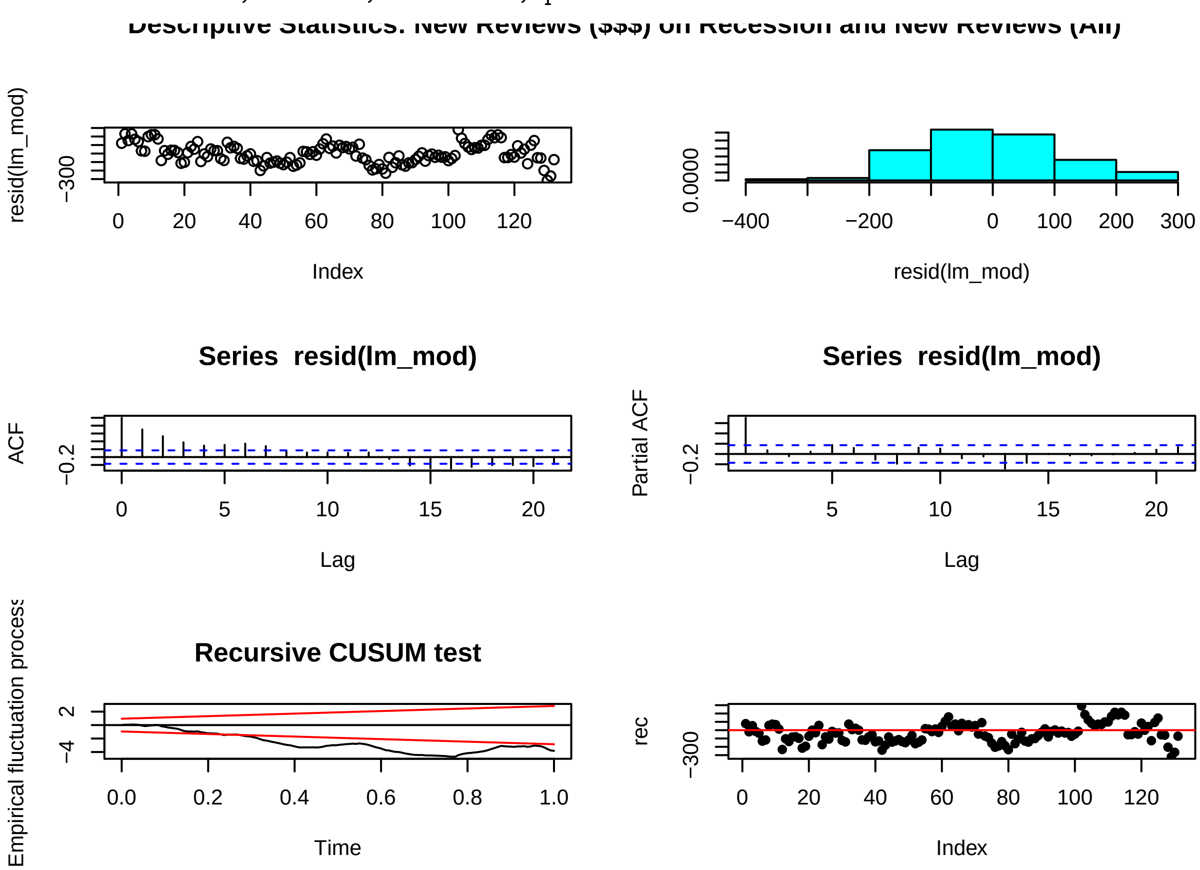

```
lm_d3_adj_rev = lm(tslm_d3_resid ~ recession_dummy_dollars_m +
    df_rev_count$coredata.df_rev_m.)
summary(lm_d3_adj_rev)
```

```
##
```

```
## 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
```

```
descriptive_stats(lm_d3_adj_rev, "New Reviews ($$$) on Recession and New Reviews (All)")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 5.8453, df1 = 2, df2 = 127, p-value = 0.003729
```

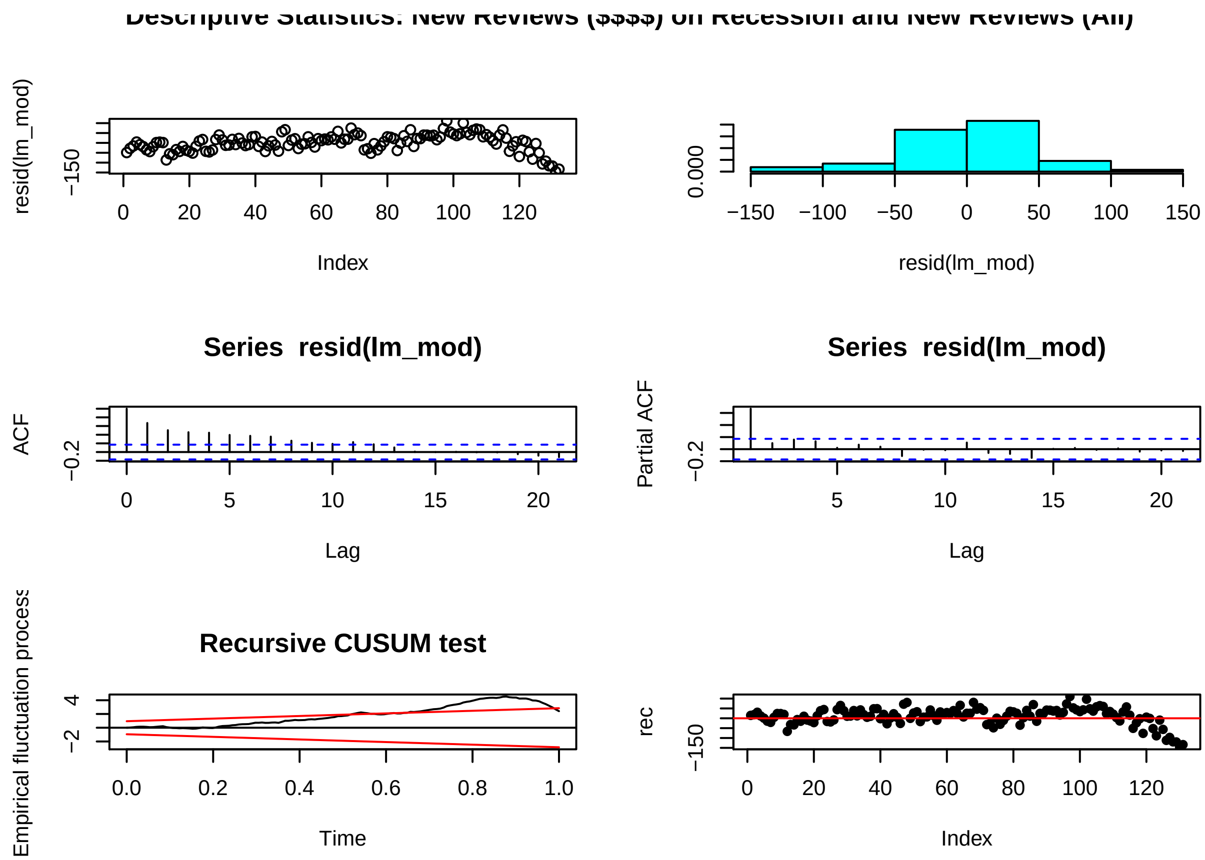


```r
lm_d4_adj_rev = lm(tslm_d4_resid ~ recession_dummy_dollars_m +
    df_rev_count$coredata.df_rev_m.)
summary(lm_d4_adj_rev)
```

```
##
## 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
```

```r
descriptive_stats(lm_d4_adj_rev, "New Reviews ($$$$) on Recession and New Reviews (All)")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 24.981, df1 = 2, df2 = 127, p-value = 7.096e-10
```



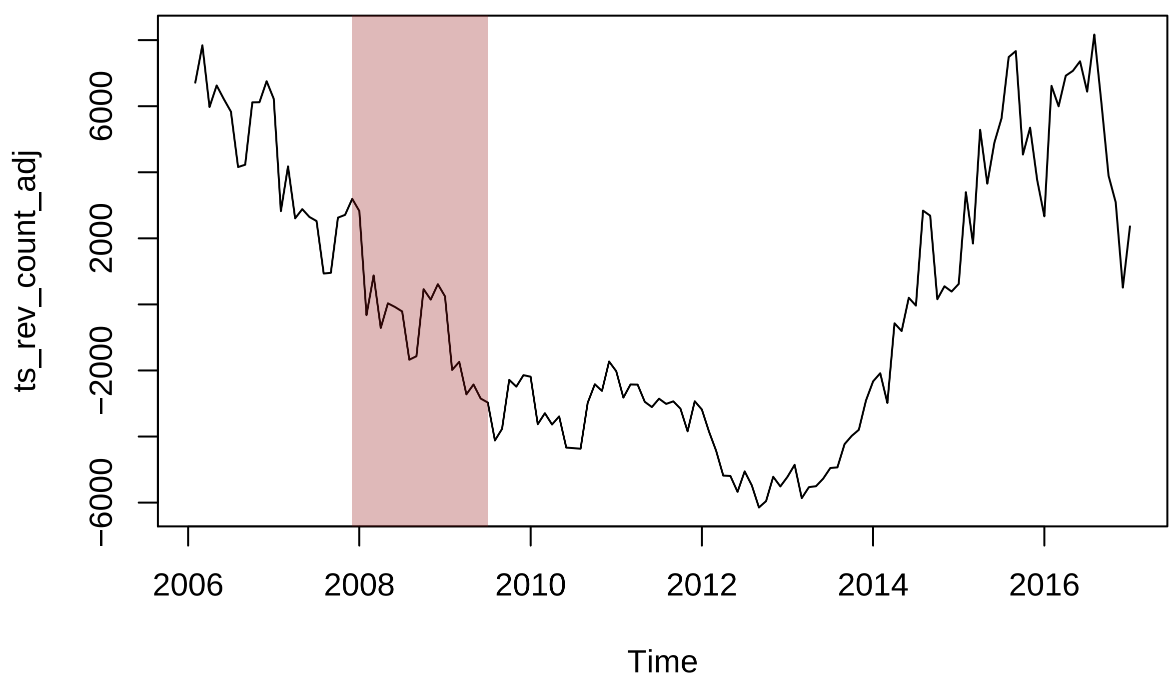
Adding new reviews in doesn't help much, but what about if we detrend and seasonally adjust?

```
# detrend review counts

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

```
##
## Call:
## tslm(formula = ts_rev_count ~ trend + season)
##
## Residuals:
##     Min       1Q  Median      3Q     Max
## -6146.8 -3161.7  -642.9  3115.8  8167.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10135.277   1480.253  -6.847 3.52e-10 ***
## trend          356.093      9.876  36.056  < 2e-16 ***
## season2       3427.756   1839.239   1.864   0.0648 .
## season3       1767.844   1838.682   0.961   0.3383
## season4       3223.751   1838.178   1.754   0.0820 .
## season5       2287.839   1837.727   1.245   0.2156
## season6       2263.018   1837.329   1.232   0.2205
## season7       2320.834   1836.984   1.263   0.2089
```

```
## season8       3676.922   1836.692   2.002   0.0476 *
## season9       3374.556   1836.453   1.838   0.0686 .
## season10      1166.008   1836.267   0.635   0.5267
## season11       750.823   1836.134   0.409   0.6833
## season12      -197.361   1836.054  -0.107   0.9146
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4306 on 119 degrees of freedom
## Multiple R-squared:  0.9165, Adjusted R-squared:  0.9081
## F-statistic: 108.9 on 12 and 119 DF,  p-value: < 2.2e-16
```

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



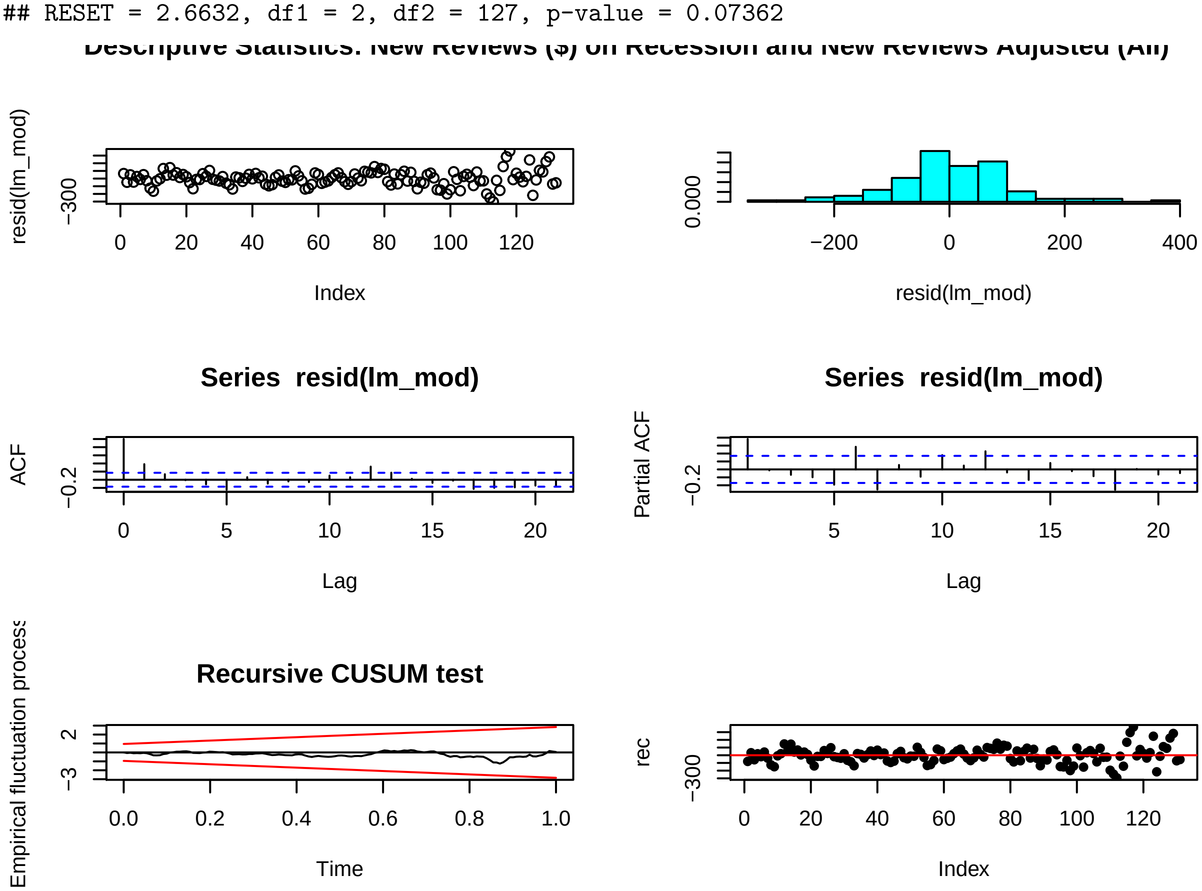

```
lm_d1_adj_rev_adj = lm(tslm_d1_resid ~ recession_dummy_dollars_m +
    ts_rev_count_adj)
summary(lm_d1_adj_rev_adj)
```

```
##
## Call:
## lm(formula = tslm_d1_resid ~ recession_dummy_dollars_m + ts_rev_count_adj)
##
## Residuals:
```

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

```
descriptive_stats(lm_d1_adj_rev_adj, "New Reviews ($) on Recession and New Reviews Adjusted (All)")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 2.6632, df1 = 2, df2 = 127, p-value = 0.07362
```

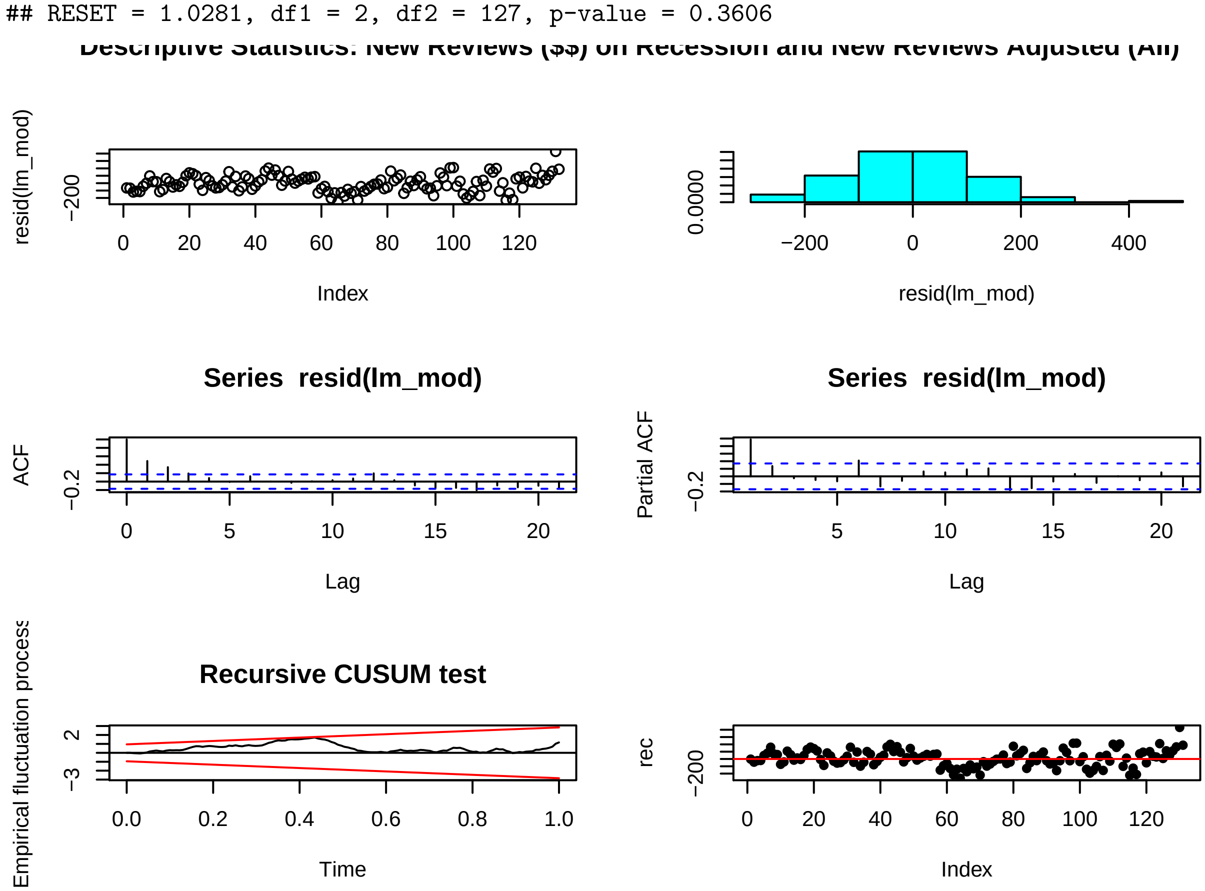



```
lm_d2_adj_rev_adj = lm(tslm_d2_resid ~ recession_dummy_dollars_m +
    ts_rev_count_adj)
summary(lm_d2_adj_rev_adj)
```

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

```r
descriptive_stats(lm_d2_adj_rev_adj, "New Reviews ($$) on Recession and New Reviews Adjusted (All)")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 1.0281, df1 = 2, df2 = 127, p-value = 0.3606
```

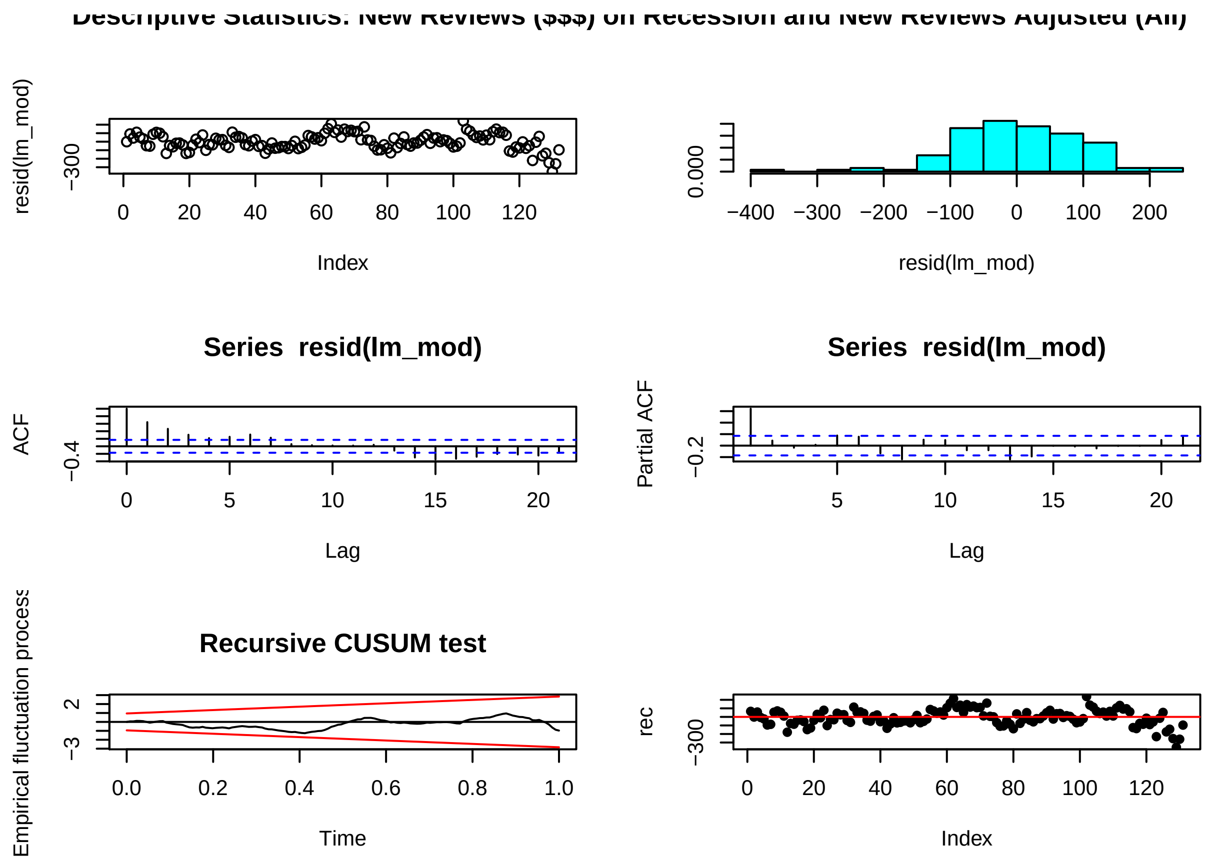


```r
lm_d3_adj_rev_adj = lm(tslm_d3_resid ~ recession_dummy_dollars_m +
    ts_rev_count_adj)
summary(lm_d3_adj_rev_adj)
```

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

```r
descriptive_stats(lm_d3_adj_rev_adj, "New Reviews ($$$) on Recession and New Reviews Adjusted (All)")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 4.1441, df1 = 2, df2 = 127, p-value = 0.01805
```



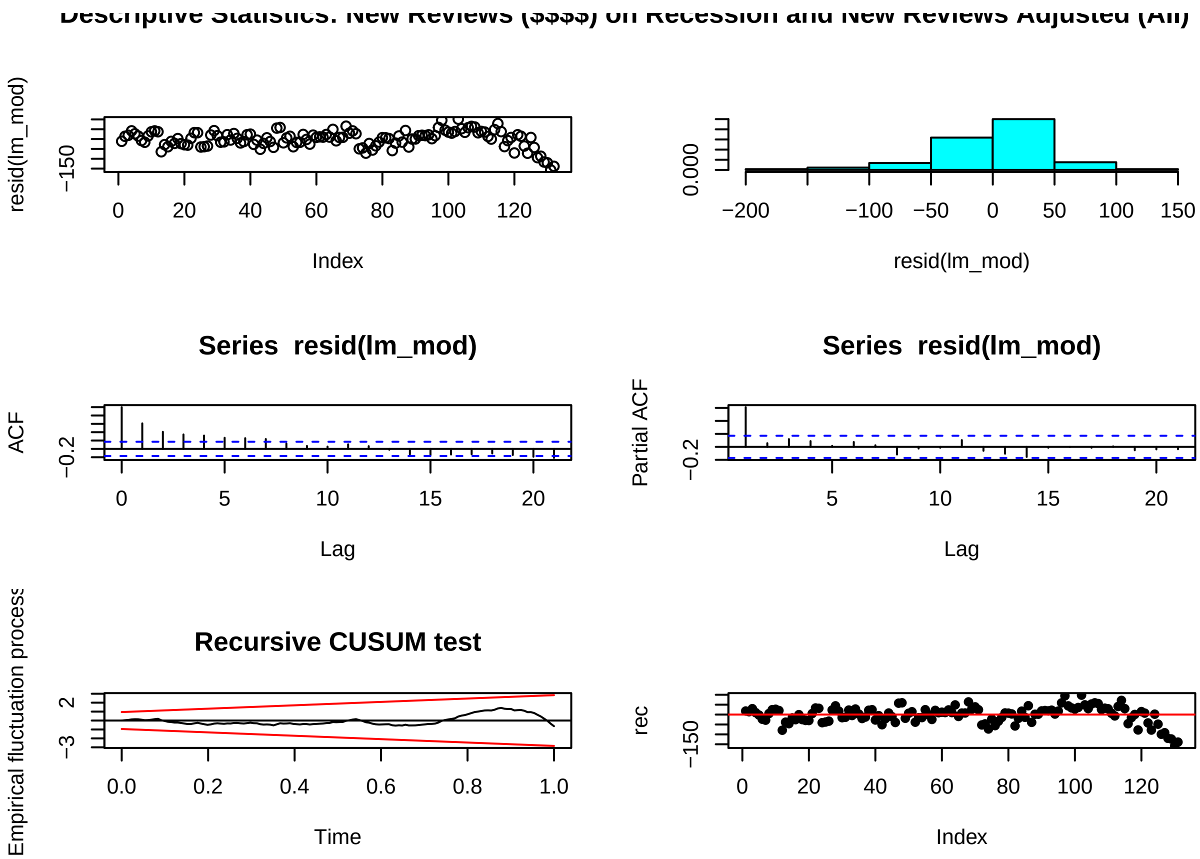

```
lm_d4_adj_rev_adj = lm(tslm_d4_resid ~ recession_dummy_dollars_m +
    ts_rev_count_adj)
summary(lm_d4_adj_rev_adj)
```

```
##
## Call:
## lm(formula = tslm_d4_resid ~ recession_dummy_dollars_m + ts_rev_count_adj)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -156.602  -22.164    6.943   27.480  101.189
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                4.662e+00  4.092e+00   1.139  0.25668
## recession_dummy_dollars_m -3.239e+01  1.081e+01  -2.996  0.00328 **
## ts_rev_count_adj          -3.962e-03  9.282e-04  -4.269 3.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.48 on 129 degrees of freedom
## Multiple R-squared:  0.1647, Adjusted R-squared:  0.1518
## F-statistic: 12.72 on 2 and 129 DF,  p-value: 9.082e-06
```

```
descriptive_stats(lm_d4_adj_rev_adj, "New Reviews ($$$$) on Recession and New Reviews Adjusted (All)")
```

```
##
```

```
##  RESET test
##
## data:  lm_mod
## RESET = 3.1303, df1 = 2, df2 = 127, p-value = 0.0471
```




```r
vif(lm_d1_adj_rev_adj)
```

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

```r
vif(lm_d2_adj_rev_adj)
```

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

```r
vif(lm_d3_adj_rev_adj)
```

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

```r
vif(lm_d4_adj_rev_adj)
```

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

Although not perfect, using detrended and seasonally adjusted data for the new reviews improves our results drastically. We finally have results that can be trustworthy.

So what do our new results tell us?

We see a decrease in ($$$$) and ($$$) reviews, but an increase in ($$) reviews. Not only do people prefer ($$) restaurants over the more expensive ($$$) and ($$$$) restaurants, it seems that customers who would originally have dined at the more expensive eateries are now choosing the less expensive ($$) restaurants. A surprising results is that the least expensive ($) restaurants do not see a significant change. This could be because inexpensive ($) restauarants are not substitutes for the others while the ($$) restaurants can be substitutes for ($$$) and ($$$$) restaurants.

Since reviews can be modeled when broken down by prices, this gives more promise to the previous review score analysis as long as it is also broken down in the same way.

Breaking Down Review Scores by Price

After splitting our reviews by prices, a re-examination of review scores is due.

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

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

```
##
## Call:
## lm(formula = df_dollars_1_star$avg ~ recession_dummy_dollars_m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26498 -0.03895 -0.01530  0.01052  0.53502
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.750694   0.009036 415.069   <2e-16 ***
## recession_dummy_dollars_m -0.044273   0.023818  -1.859   0.0653 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09606 on 130 degrees of freedom
## Multiple R-squared:  0.02589,    Adjusted R-squared:  0.0184
## F-statistic: 3.455 on 1 and 130 DF,  p-value: 0.06532
```

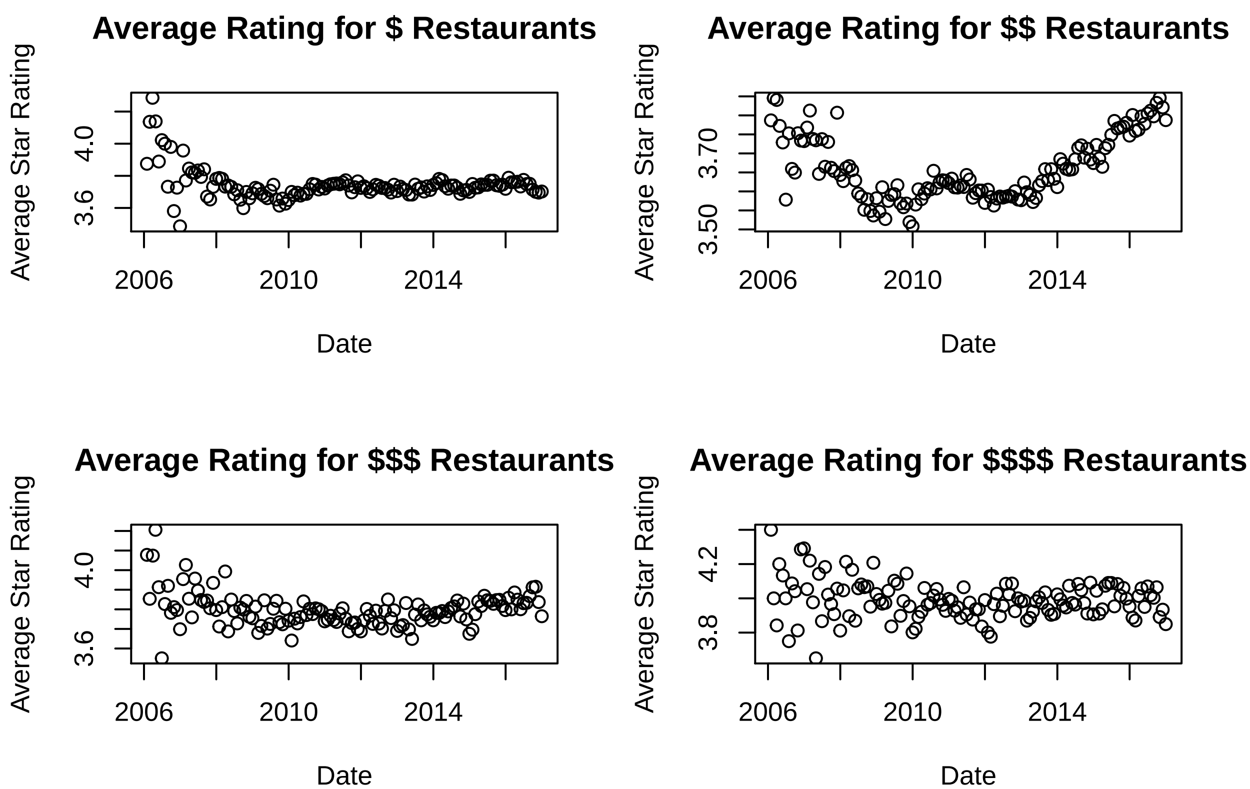
```
dollars_2_star_xts = xts(dollars_obd_2_star$stars, as.Date(dollars_obd_2_star$date,
    "%Y-%m-%d"))
df_d_2_star = apply.monthly(dollars_2_star_xts, sum)
df_dollars_2_star = data.frame(date = index(df_d_2_star), coredata(df_d_2_star))
df_dollars_2_star$avg = df_dollars_2_star$coredata.df_d_2_star./df_dollars_2$coredata.df_d_2.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

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

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

```
##
## Call:
## lm(formula = df_dollars_3_star$avg ~ recession_dummy_dollars_m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25080 -0.05478 -0.00522  0.03954  0.40509
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3.800796   0.008157 465.945   <2e-16 ***
## recession_dummy_dollars_m -0.020898   0.021501  -0.972    0.333
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08671 on 130 degrees of freedom
## Multiple R-squared:  0.007215,   Adjusted R-squared:  -0.000422
## F-statistic: 0.9447 on 1 and 130 DF,  p-value: 0.3329
```

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



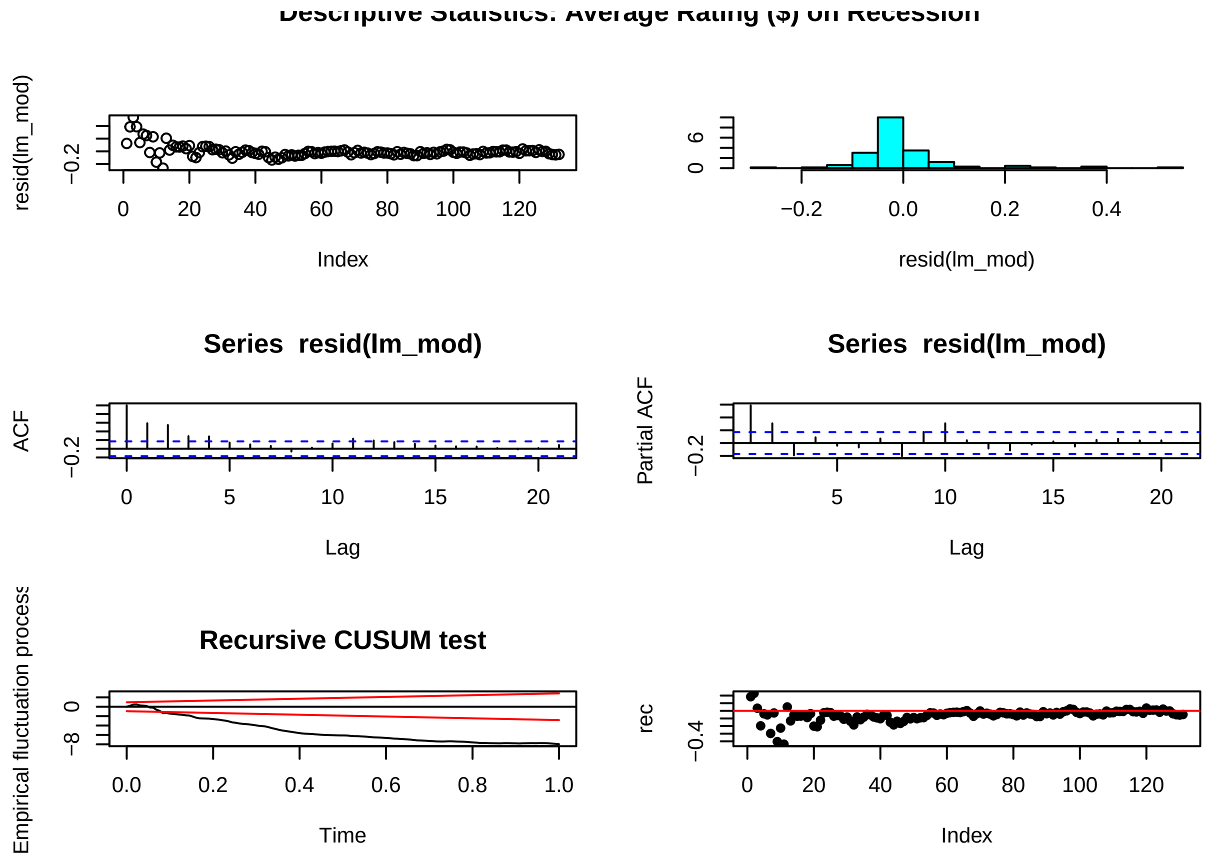

```r
d4_star_lm = lm(df_dollars_4_star$avg ~ recession_dummy_dollars_m)
summary(d4_star_lm)
```

```
##
## 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
```

```r
descriptive_stats(d1_star_lm, "Average Rating ($) on Recession")
```

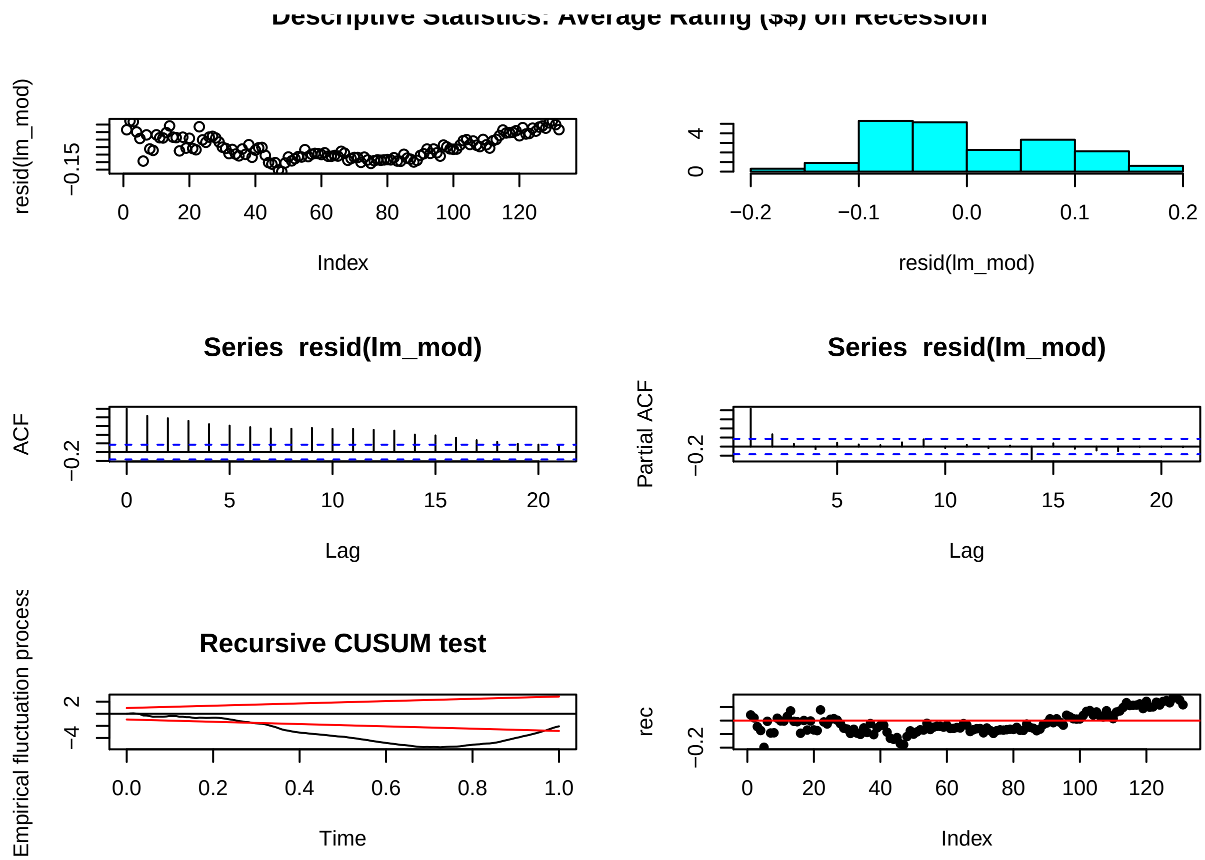
```
##
##  RESET test
##
```

```
## data:  lm_mod
## RESET = 0, df1 = 2, df2 = 128, p-value = 1
```



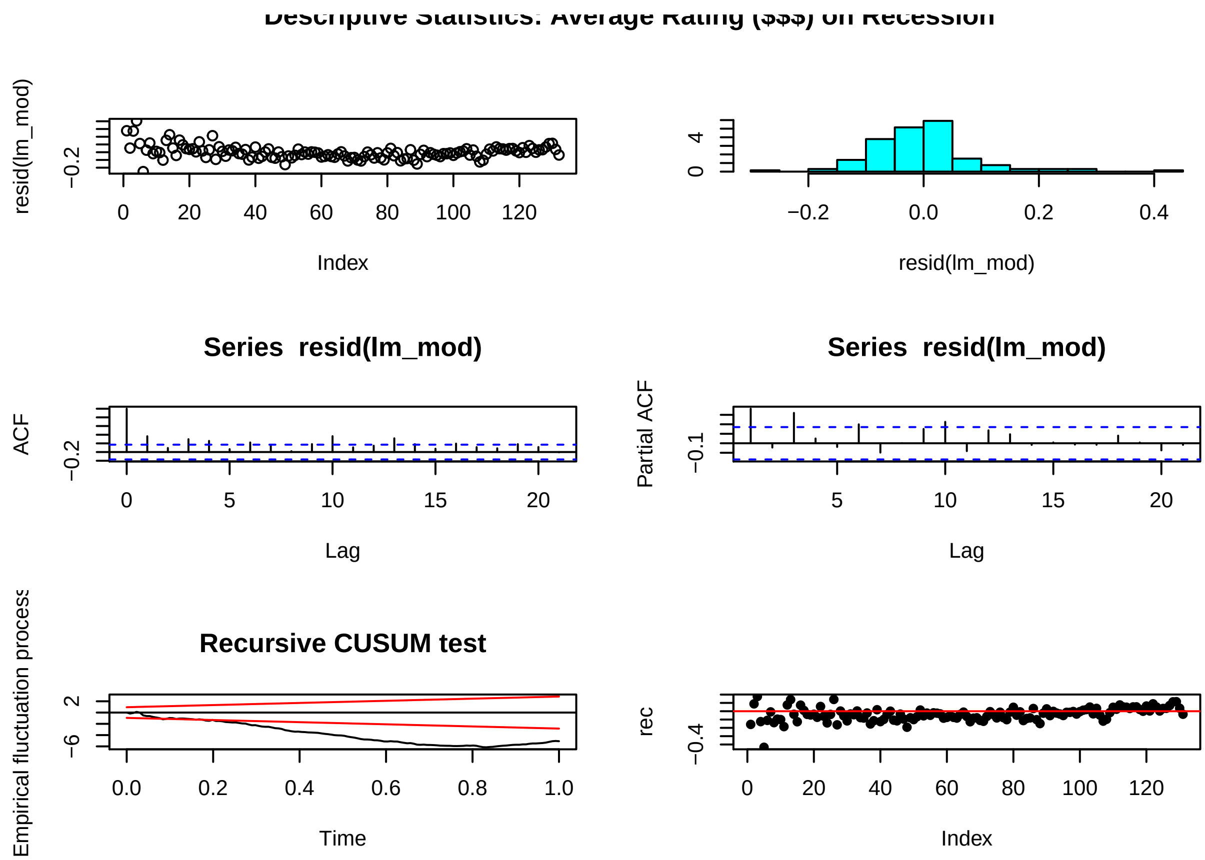

```r
descriptive_stats(d2_star_lm, "Average Rating ($$) on Recession")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 0, df1 = 2, df2 = 128, p-value = 1
```



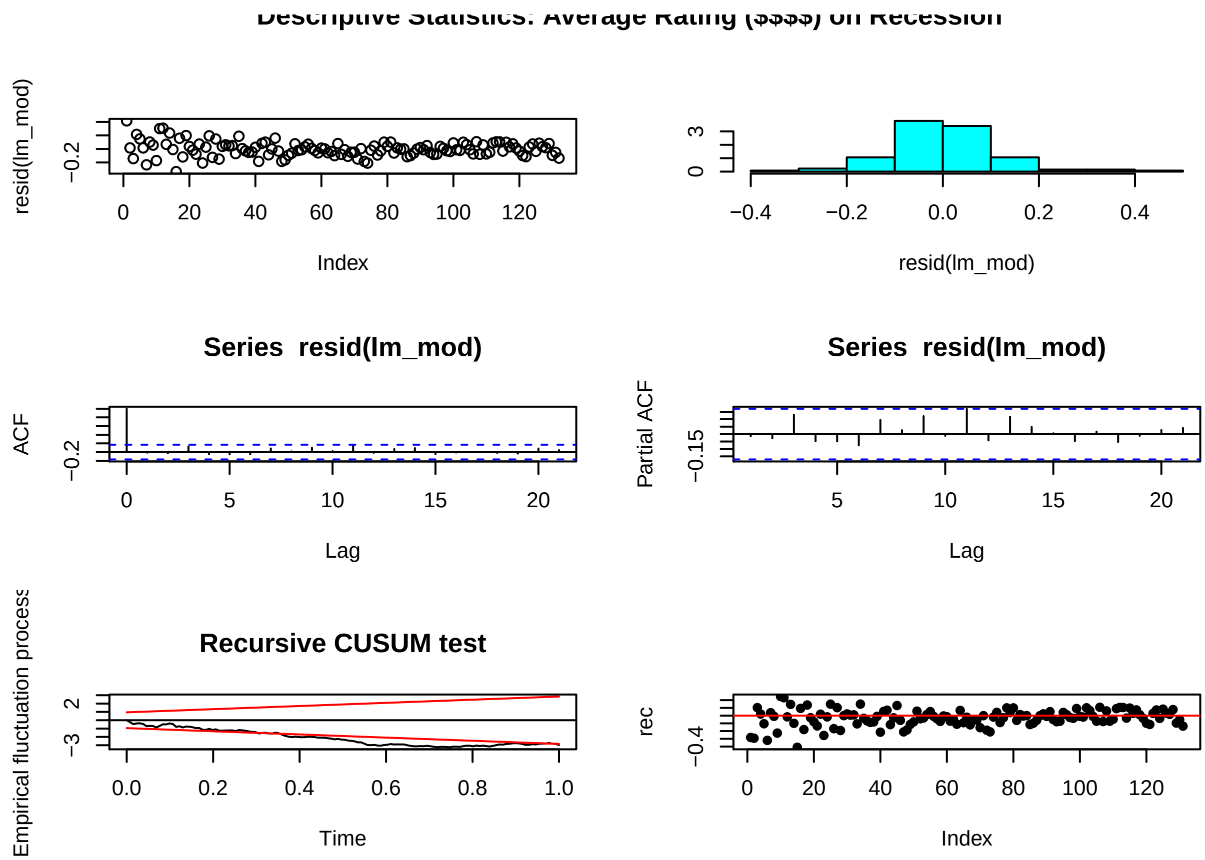

```r
descriptive_stats(d3_star_lm, "Average Rating ($$$) on Recession")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 0, df1 = 2, df2 = 128, p-value = 1
```




```
descriptive_stats(d4_star_lm, "Average Rating ($$$$) on Recession")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 0, df1 = 2, df2 = 128, p-value = 1
```




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

The results look better, but still are not that great. If we are to interpret the regression anyways, however, it looks as if the lower priced restaurants have lower review scores. This could be a sign of many things, such as the substituters from ($$$) and ($$$$) restaurants having higher expectations or people just want more bang for their buck. The higher priced restaurants do not see a change, which could be due to the type of people who still dine there during recessions. They could be part of the group unaffected by the recession.

Sentiment Analysis

By doing a sentiment analysis on the text in the reviews, we can see the association with words during the recession and the period of time (of equal length) directly after the recession.

```
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)


corpus_reviews_norec = buildCorpus(restaurant_reviews_norec$text,
    0)
dtm_norec = DocumentTermMatrix(corpus_reviews_norec)
tidy_norec = tidy(dtm_norec)
```

Word Clouds:

```
par(mfrow = c(1, 2))
buildWordCloud(corpus_reviews_rec, "Spectral", 8, "test")
buildWordCloud(corpus_reviews_norec, "Spectral", 8, "test2")
```

It is hard to tell which word cloud is from the recession and which is from the period after, but the one on the left is from the recession.

Sentiment Visualization:

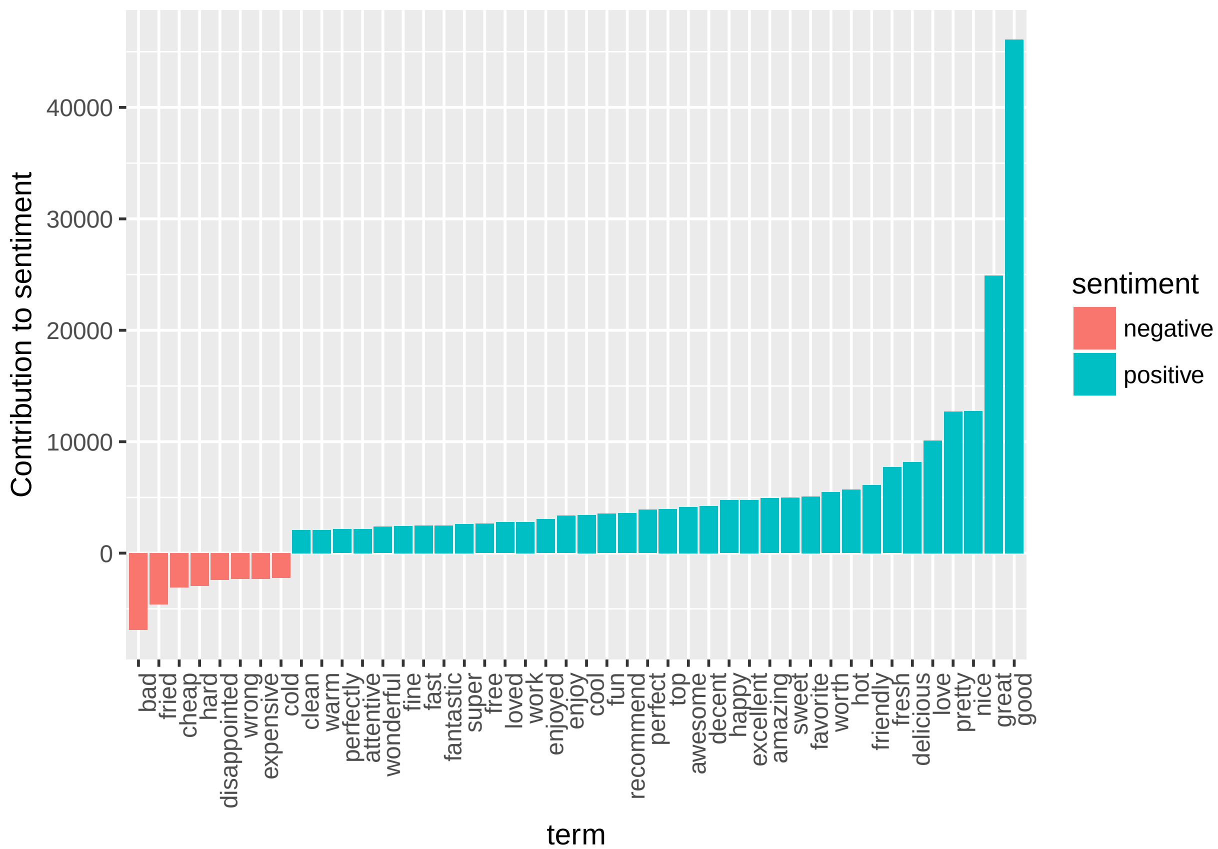
```
# reference:
# https://cran.r-project.org/web/packages/tidytext/vignettes/tidying_casting.html
ap_sentiments <- tidy_rec %>% inner_join(get_sentiments("bing"),
    by = c(term = "word"))

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")
```



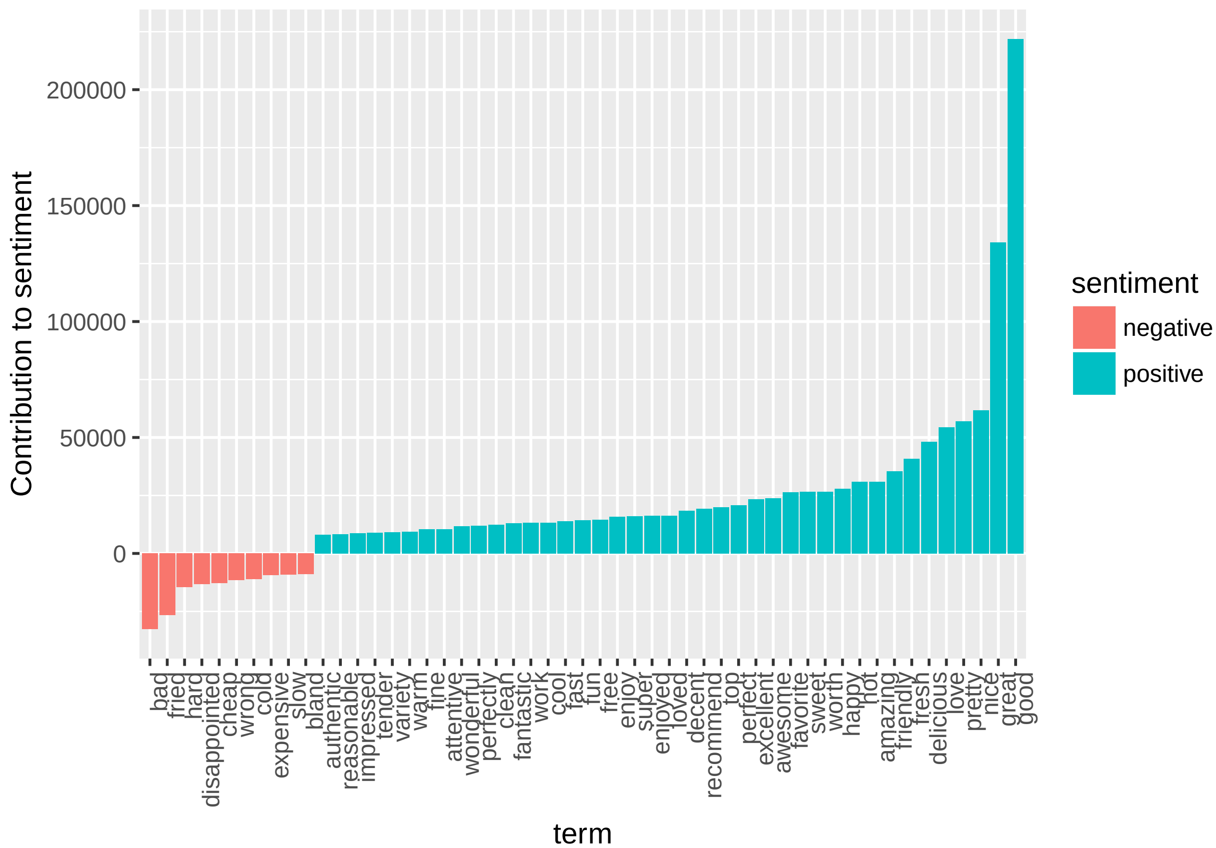

```r
# negative: 158714 positive: 346991 percent negative: 31.4%
# cheap: 3rd highest negative word expensive: 7th




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")
```




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

Again, it is difficult to distinguish.

The Sentiment visualization on the top is from the recession. On closer examination of the sentiments, the following statistics are extracted.

Recession:
negative: 158714
positive: 346991
percent negative: 31.4%
cheap negative word rank: 3rd

expensive negative word rank: 7th

Post-Recession:
negative: 733329
positive: 1731938
percent negative: 29.75%
cheap negative word rank: 5th
expensive negative word rank: 8th

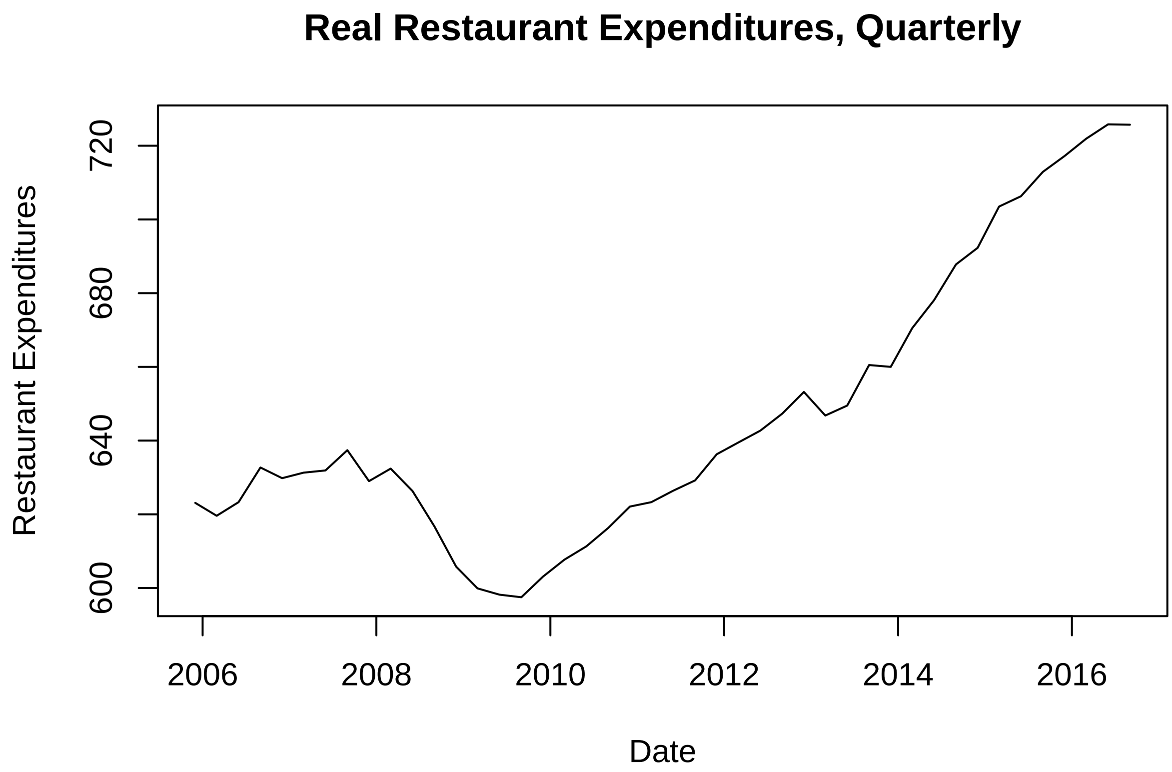
There is a higher percentage of negative sentiment during the recession as well as having the words "cheap" and "expensive" rank higher for negative words. The word "cheap", however is not always used in a negative way, but it is still related to price. This shows that reviews are more concerned with price during the recession compared to the period directly following it. It should be noted that there is a large difference in the sample size of words from the two periods. This leads to a soft conclusion that reviewers are more concerned with prices during the recession, as it follows recessionary thinking.

Connecting Yelp Reviews with the Restaurant Industry

By using restaurant expenditures, there can be a connection made between the review data and the actual restaurant industry.

But first, lets confirm that GDP and the recession can be linked to restaurant expenditures.

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

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




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

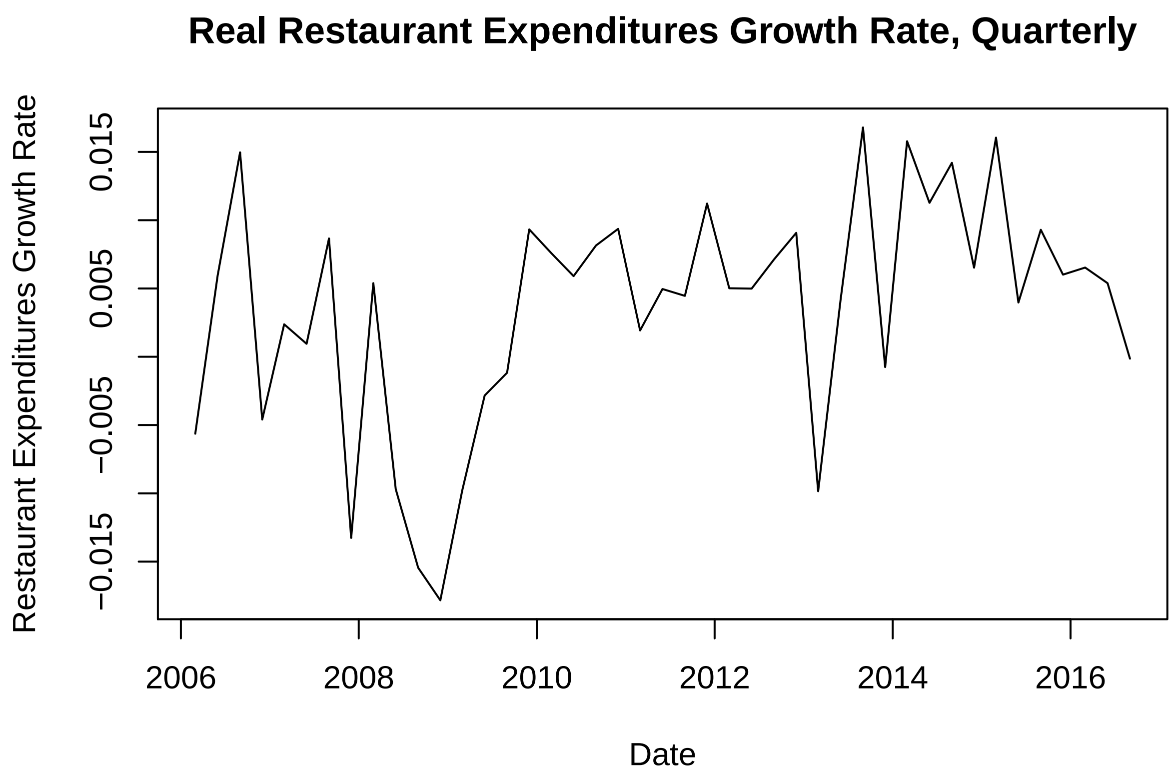
```
##
##  KPSS Test for Level Stationarity
##
## data:  t
## KPSS Level = 1.7047, Truncation lag parameter = 1, p-value = 0.01
##
##
##  Augmented Dickey-Fuller Test
##
## data:  t
## Dickey-Fuller = -2.4338, Lag order = 3, p-value = 0.4021
## alternative hypothesis: stationary
```

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

test_stationary(rest_real_exp_diff_log)
```

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

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

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




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

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: rest_real_exp_diff_log, gdp_growth_subset
## Deterministic variables: const
## Sample size: 40
## Log Likelihood: 309.26
## Roots of the characteristic polynomial:
```

```
## 0.829 0.7485 0.7485 0.7006 0.5467 0.5467
## Call:
## VAR(y = gdp_exp_combined, p = select$select[1])
##
##
## Estimation results for equation rest_real_exp_diff_log:
## =======================================================
## rest_real_exp_diff_log = rest_real_exp_diff_log.l1 + gdp_growth_subset.l1 + rest_real_exp_diff_log.l2
##
##                            Estimate Std. Error t value Pr(>|t|)
## rest_real_exp_diff_log.l1 -0.3061642  0.2190439  -1.398  0.17152
## gdp_growth_subset.l1       0.7168516  0.2564126   2.796  0.00857 **
## rest_real_exp_diff_log.l2  0.1645218  0.1996566   0.824  0.41584
## gdp_growth_subset.l2       0.1951121  0.2662260   0.733  0.46880
## rest_real_exp_diff_log.l3  0.1923951  0.1868020   1.030  0.31053
## gdp_growth_subset.l3       0.1265239  0.2690216   0.470  0.64123
## const                     -0.0001735  0.0013498  -0.129  0.89852
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.007011 on 33 degrees of freedom
## Multiple R-Squared: 0.4144,  Adjusted R-squared: 0.3079
## F-statistic: 3.892 on 6 and 33 DF,  p-value: 0.004777
##
##
## Estimation results for equation gdp_growth_subset:
## ==================================================
## gdp_growth_subset = rest_real_exp_diff_log.l1 + gdp_growth_subset.l1 + rest_real_exp_diff_log.l2 + 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
```

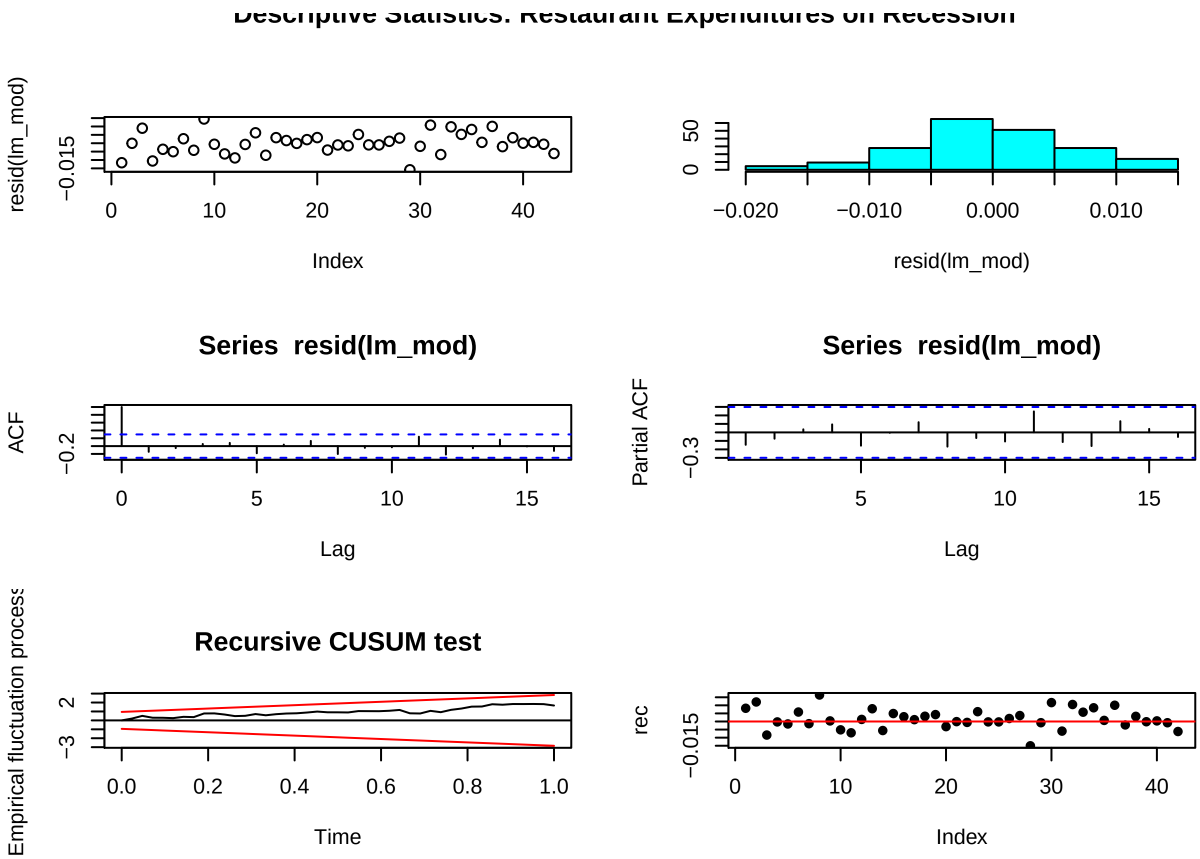
```r
rec_exp_diff_log_dummy = add_recession_dummy(rest_exp_dates_diff)
lm_rest_real_exp_diff_log = lm(rest_real_exp_diff_log ~ rec_exp_diff_log_dummy)
summary(lm_rest_real_exp_diff_log)
```

```
##
## 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
```

```r
descriptive_stats(lm_rest_real_exp_diff_log, "Restaurant Expenditures on Recession")
```

```
##
##  RESET test
##
## data:  lm_mod
```

```
## RESET = 0, df1 = 2, df2 = 39, p-value = 1
```



The assumption that restaurant expenditures are related to the recession and GDP is confirmed. GDP will not be needed in constructing a full model as restaurant expenditures will be used in its place.

By seasonally adjusting the growth rate in new reviews, a Granger causality test can be run between restaurant expenditures and new reviews.

```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=
# vm_rev_exp=VAR(rev_exp_combined,select$select[1]) #
# plot(vm_rev_exp$y) summary(vm_rev_exp)
# grangertest(rest_real_exp_diff_log~log_rev_quarter,
# order=select$select[1])
# grangertest(log_rev_quarter~rest_real_exp_diff_log,
# order=select$select[1]) #try to seasonally adjust #leave
# trend in because the original seasonally adjusted rest exp
# had a trend

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

```
##
## Call:
## tslm(formula = log_rev_quarter ~ season)
##
## Residuals:
```

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

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


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

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: rest_real_exp_diff_log, log_rev_quarter_adj
## Deterministic variables: const
## Sample size: 39
## Log Likelihood: 196.578
## Roots of the characteristic polynomial:
## 0.9055 0.828 0.721 0.721 0.5383 0.5383 0.4846 0.4846
## Call:
## VAR(y = rev_exp_adj_combined, p = select$select[1])
##
##
## Estimation results for equation rest_real_exp_diff_log:
## ==========================================================
## rest_real_exp_diff_log = rest_real_exp_diff_log.l1 + log_rev_quarter_adj.l1 + rest_real_exp_diff_log
##
##                            Estimate Std. Error t value Pr(>|t|)
## rest_real_exp_diff_log.l1  0.180897   0.180663   1.001    0.325
## log_rev_quarter_adj.l1    -0.010709   0.014668  -0.730    0.471
## rest_real_exp_diff_log.l2  0.239154   0.178751   1.338    0.191
## log_rev_quarter_adj.l2    -0.004026   0.011875  -0.339    0.737
## rest_real_exp_diff_log.l3  0.099466   0.168370   0.591    0.559
## log_rev_quarter_adj.l3     0.003002   0.012594   0.238    0.813
## rest_real_exp_diff_log.l4  0.026576   0.166231   0.160    0.874
## log_rev_quarter_adj.l4    -0.011961   0.011398  -1.049    0.302
## const                      0.001700   0.001486   1.144    0.262
##
```

```
##
## Residual standard error: 0.007706 on 30 degrees of freedom
## Multiple R-Squared: 0.3411,  Adjusted R-squared: 0.1654
## F-statistic: 1.941 on 8 and 30 DF,  p-value: 0.09013
##
##
## Estimation results for equation log_rev_quarter_adj:
## ====================================================
## log_rev_quarter_adj = rest_real_exp_diff_log.l1 + log_rev_quarter_adj.l1 + rest_real_exp_diff_log.l2
##
##                           Estimate Std. Error t value Pr(>|t|)
## rest_real_exp_diff_log.l1 -2.47221    1.57285  -1.572  0.12649
## log_rev_quarter_adj.l1     0.15930    0.12770   1.247  0.22186
## rest_real_exp_diff_log.l2  2.79746    1.55620   1.798  0.08231 .
## log_rev_quarter_adj.l2     0.22813    0.10338   2.207  0.03513 *
## rest_real_exp_diff_log.l3 -0.09956    1.46582  -0.068  0.94630
## log_rev_quarter_adj.l3    -0.15938    0.10964  -1.454  0.15643
## rest_real_exp_diff_log.l4 -2.85185    1.44720  -1.971  0.05806 .
## log_rev_quarter_adj.l4     0.28534    0.09923   2.875  0.00735 **
## const                     -0.01439    0.01293  -1.112  0.27477
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.06709 on 30 degrees of freedom
## Multiple R-Squared: 0.5769,  Adjusted R-squared: 0.4641
## F-statistic: 5.114 on 8 and 30 DF,  p-value: 0.0004486
##
##
##
## Covariance matrix of residuals:
##                        rest_real_exp_diff_log log_rev_quarter_adj
## rest_real_exp_diff_log              5.938e-05            0.000157
## log_rev_quarter_adj                 1.570e-04            0.004501
##
## Correlation matrix of residuals:
##                        rest_real_exp_diff_log log_rev_quarter_adj
## rest_real_exp_diff_log                 1.0000              0.3038
## log_rev_quarter_adj                    0.3038              1.0000
```

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

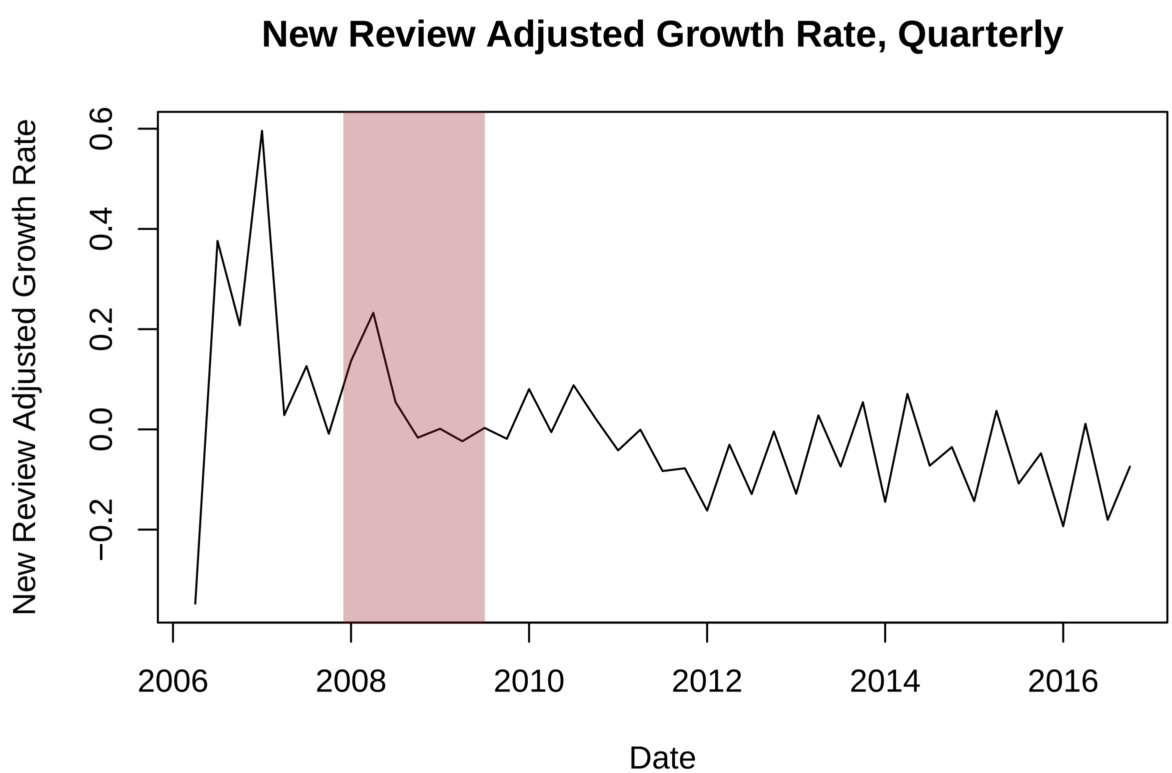
```
## Granger causality test
##
## Model 1: rest_real_exp_diff_log ~ Lags(rest_real_exp_diff_log, 1:4) + Lags(log_rev_quarter_adj, 1:4)
## Model 2: rest_real_exp_diff_log ~ Lags(rest_real_exp_diff_log, 1:4)
##   Res.Df Df      F Pr(>F)
## 1     30
## 2     34 -4 0.6949 0.6014
```

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

```
## Granger causality test
##
## Model 1: log_rev_quarter_adj ~ Lags(log_rev_quarter_adj, 1:4) + Lags(rest_real_exp_diff_log, 1:4)
```

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

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




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

New reviews is Granger caused by restaurant expenditures. This makes sense as reviews are typically made after people visit a restaurant. This shows that the number of reviews can in fact be related to the previous restaurant expenditures.

Building Full Models

A full model that incorporates all the previous results will be useful to show the real effect of the recession and determine if there is a causal inference that can be made.

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

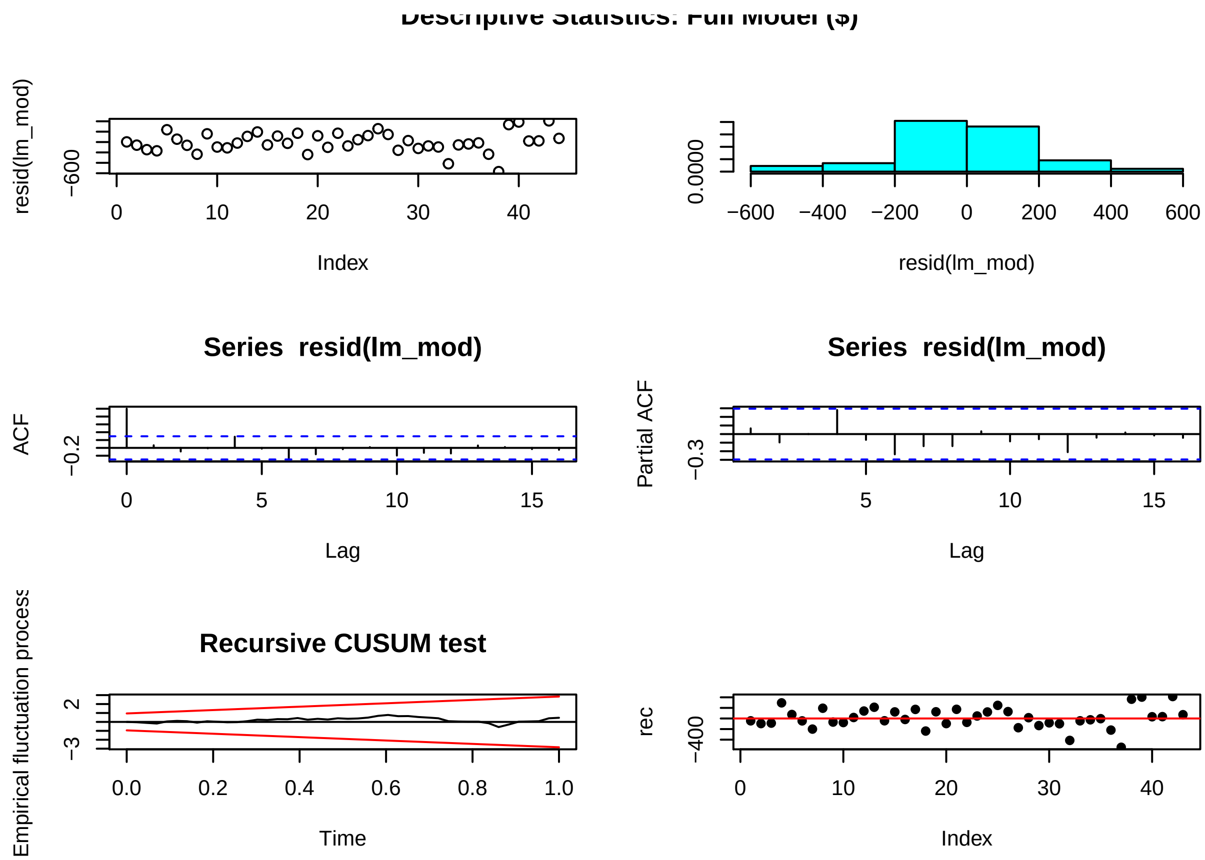
```r
# adjusted num new rev by dollar signs
d1_quarterly = apply.quarterly(as.xts(tslm_d1_resid), FUN = sum)
ts_d1_quarterly = ts(d1_quarterly, start = c(2006, 1), freq = 4)

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

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

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

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

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

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

d3_star_quarterly = apply.quarterly(xts(df_dollars_3_star$avg,
    as.Date(df_dollars_3_star$date, "%Y-%m-%d")), FUN = sum)
ts_d3_star_quarterly = ts(d3_star_quarterly, start = c(2006,
    1), freq = 4)

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

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

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

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

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

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

```r
descriptive_stats(lm_d1_full, "Full Model ($)")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 0.58033, df1 = 2, df2 = 37, p-value = 0.5647
```

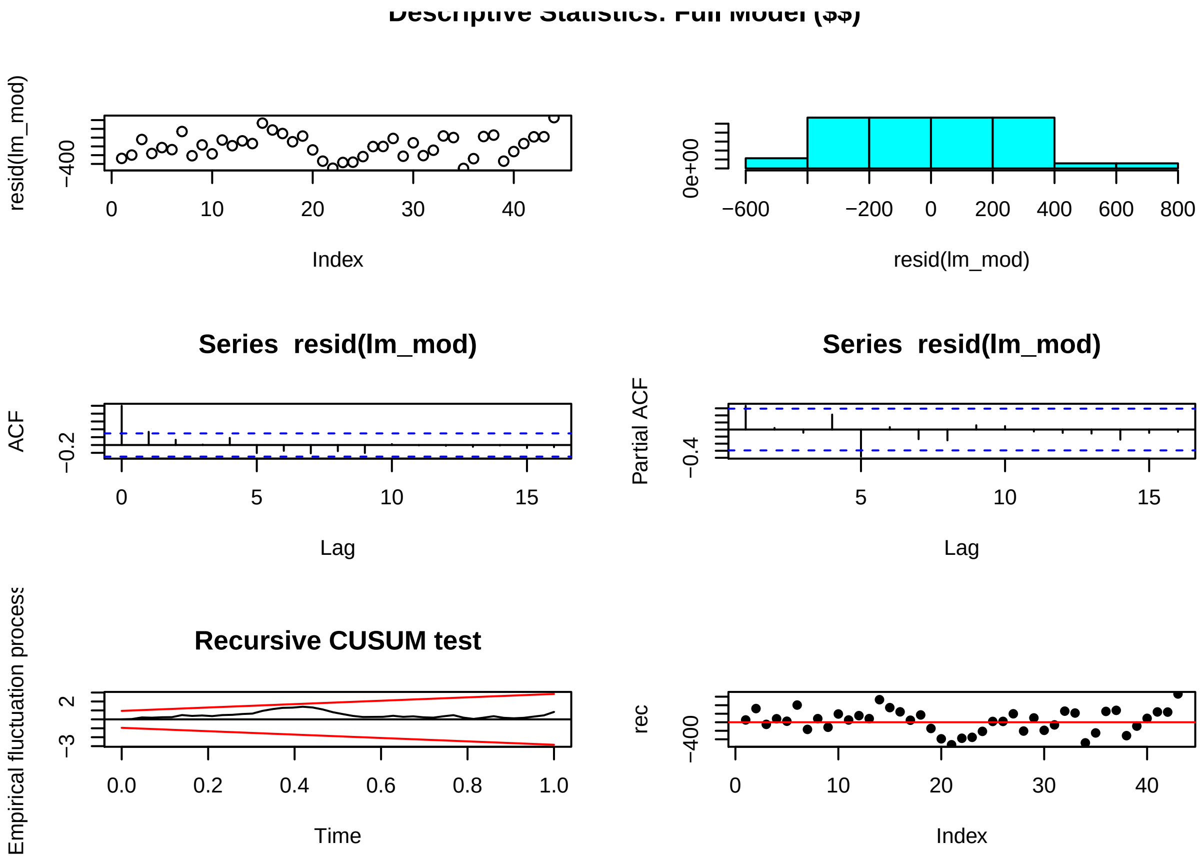



```r
lm_d2_full = lm(ts_d2_quarterly ~ rec_q + ts_rev_count_adj_q +
    ts_d2_star_quarterly + rest_exp_q_adj)
summary(lm_d2_full)
```

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

```r
descriptive_stats(lm_d2_full, "Full Model ($$)")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 0.40231, df1 = 2, df2 = 37, p-value = 0.6717
```



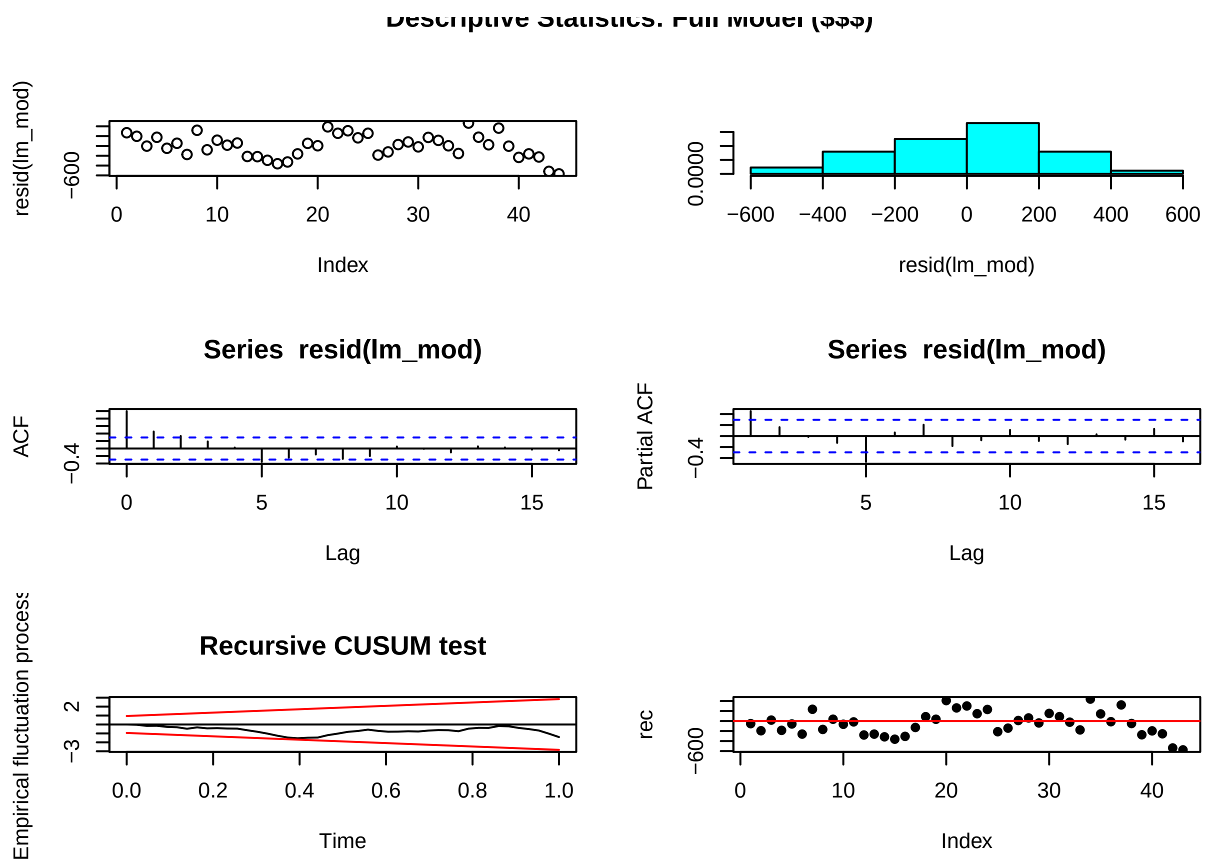

```r
lm_d3_full = lm(ts_d3_quarterly ~ rec_q + ts_rev_count_adj_q +
    ts_d3_star_quarterly + rest_exp_q_adj)
summary(lm_d3_full)
```

```
##
## Call:
## lm(formula = ts_d3_quarterly ~ rec_q + ts_rev_count_adj_q + ts_d3_star_quarterly +
##     rest_exp_q_adj)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -568.78 -164.74    9.84  174.17  465.42
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             4.254e+03  3.553e+03   1.197   0.2384
## rec_q                  -2.089e+02  1.039e+02  -2.011   0.0513 .
## ts_rev_count_adj_q      2.963e-02  6.739e-03   4.397 8.22e-05 ***
```

```
## ts_d3_star_quarterly -3.705e+02  3.116e+02  -1.189   0.2418
## rest_exp_q_adj       -6.025e+00  3.829e+00  -1.573   0.1237
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 247.2 on 39 degrees of freedom
## Multiple R-squared:  0.5027, Adjusted R-squared:  0.4517
## F-statistic: 9.856 on 4 and 39 DF,  p-value: 1.311e-05
```

```r
descriptive_stats(lm_d3_full, "Full Model ($$$)")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 0.031157, df1 = 2, df2 = 37, p-value = 0.9693
```



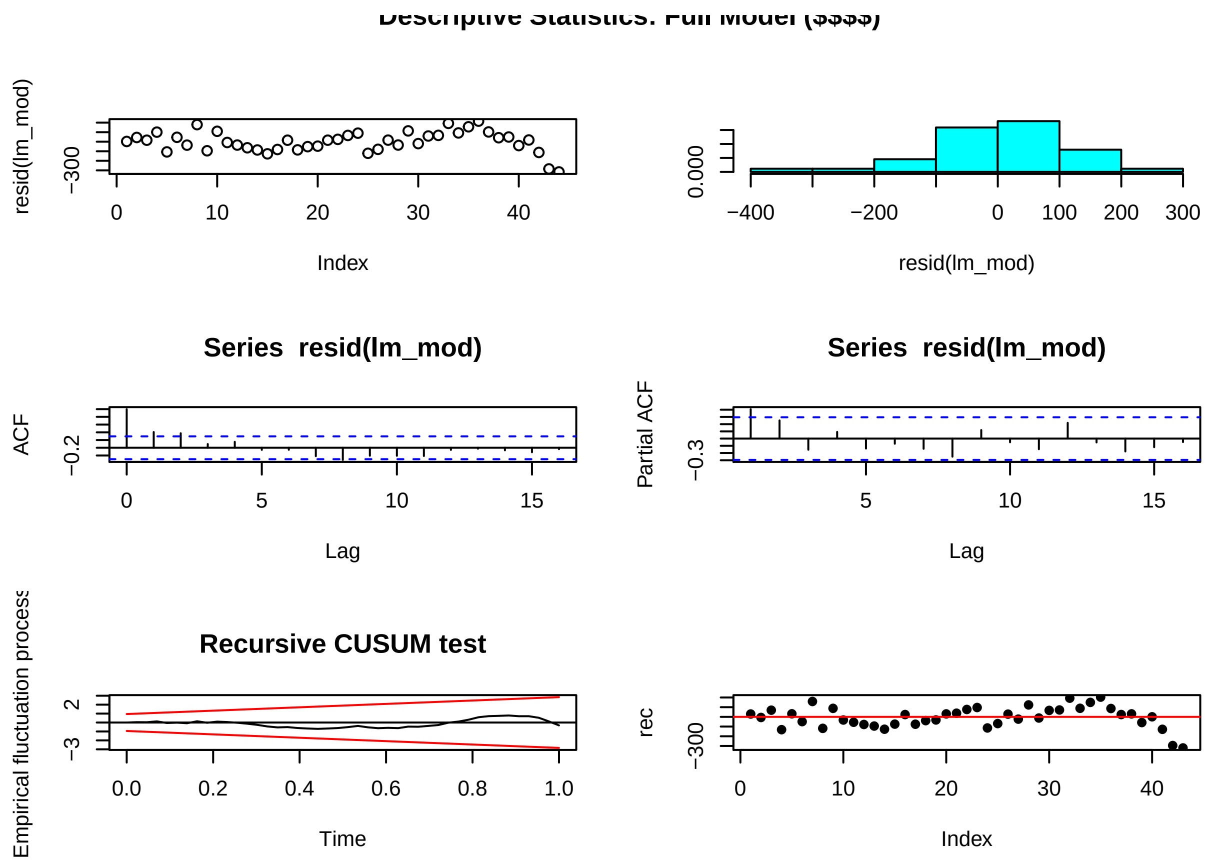

```r
lm_d4_full = lm(ts_d4_quarterly ~ rec_q + ts_rev_count_adj_q +
    ts_d4_star_quarterly + rest_exp_q_adj)
summary(lm_d4_full)
```

```
##
## Call:
## lm(formula = ts_d4_quarterly ~ rec_q + ts_rev_count_adj_q + ts_d4_star_quarterly +
##     rest_exp_q_adj)
##
## Residuals:
```

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

```r
descriptive_stats(lm_d4_full, "Full Model ($$$$)")
```

```
##
##  RESET test
##
## data:  lm_mod
## RESET = 6.0849, df1 = 2, df2 = 37, p-value = 0.005192
```




```r
vif(lm_d1_full)
```

```
##                  rec_q   ts_rev_count_adj_q ts_d1_star_quarterly
```

```
##             1.078443             4.614419             1.318413
##       rest_exp_q_adj
##             4.514264
```

```r
vif(lm_d2_full)
```

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

```r
vif(lm_d3_full)
```

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

```r
vif(lm_d4_full)
```

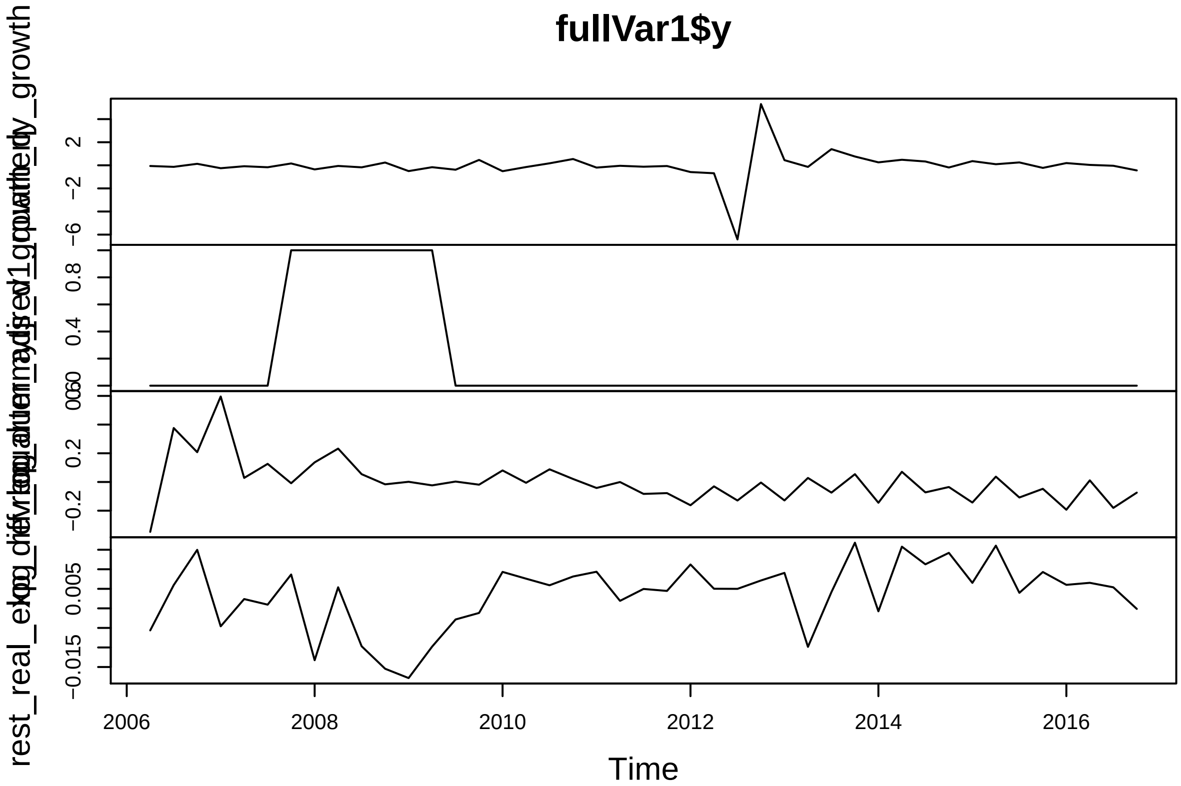
```
##                rec_q   ts_rev_count_adj_q ts_d4_star_quarterly
##             1.079835             4.752153             1.268115
##       rest_exp_q_adj
##             4.487196
```

```r
# lots of insignificance and multicollinearity
```

The regression results and error descriptive statistics are actually worse in this model. There is even multicolinearity occuring. The better model is actually in the form of Restaurant Review (dollars) regressed on Recession and Adjusted New Reviews.

Since new reviews is Granger caused by restaurant expenditures, can we find a causal inference through comprehensive VAR models that include the "Better" model, with restaurant expenditures added in?

```r
# how about a var standardize so min value is 1
ts_d1_quarterly_growth = diff(log(ts_d1_quarterly + 1 - min(ts_d1_quarterly)))
ts_d2_quarterly_growth = diff(log(ts_d2_quarterly + 1 - min(ts_d2_quarterly)))
ts_d3_quarterly_growth = diff(log(ts_d3_quarterly + 1 - min(ts_d3_quarterly)))
ts_d4_quarterly_growth = diff(log(ts_d4_quarterly + 1 - min(ts_d4_quarterly)))

# rec_dummy_rev_growth_q,rest_real_exp_diff_log,log_rev_quarter_adj


combinedFull1 = cbind(ts_d1_quarterly_growth, rec_dummy_rev_growth_q,
    log_rev_quarter_adj, rest_real_exp_diff_log)
select = VARselect(combinedFull1, lag.max = 4, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
fullVar1 = VAR(combinedFull1, p = select$select[1])
plot(fullVar1$y)
```




```r
summary(fullVar1)
```

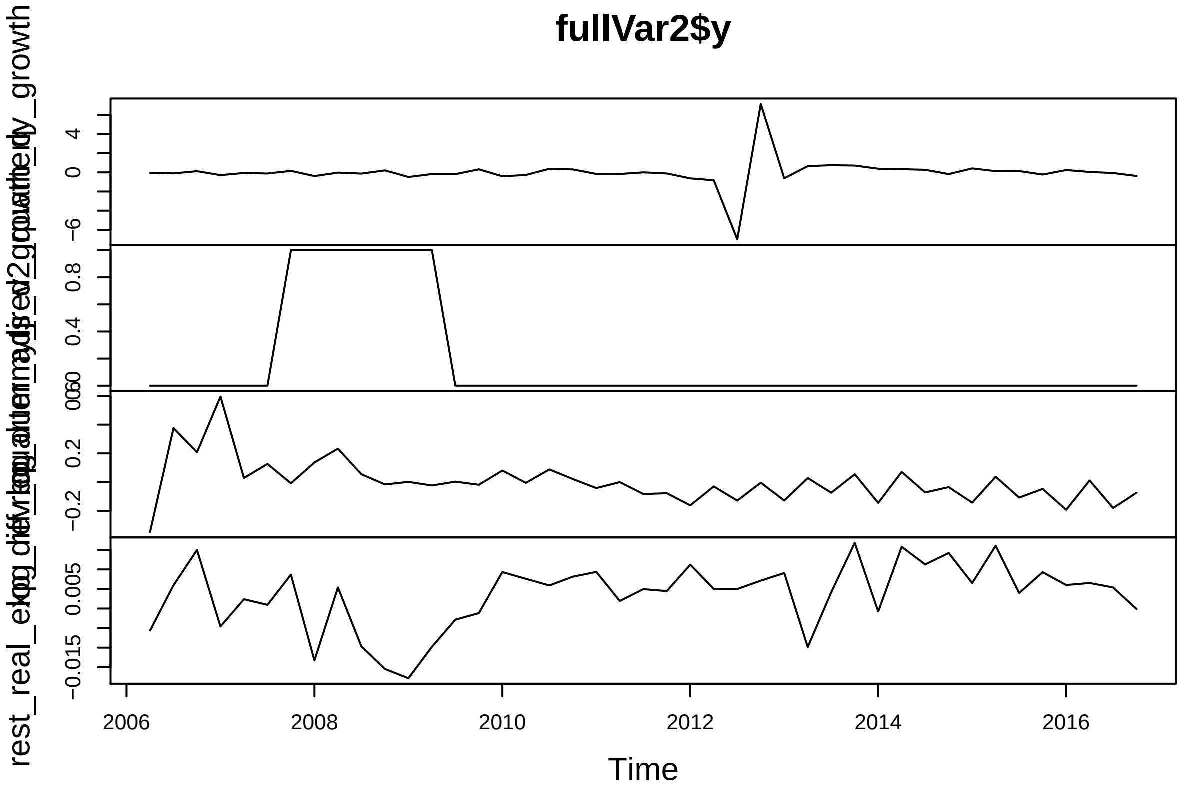
```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: ts_d1_quarterly_growth, rec_dummy_rev_growth_q, log_rev_quarter_adj, rest_real_
## Deterministic variables: const
## Sample size: 39
## Log Likelihood: 186.394
## Roots of the characteristic polynomial:
## 0.8951 0.8609 0.8609 0.8126 0.7882 0.7882 0.7858 0.7858 0.7761 0.7761 0.687 0.687 0.6853 0.6853 0.52
## Call:
## VAR(y = combinedFull1, p = select$select[1])
##
##
## Estimation results for equation ts_d1_quarterly_growth:
## =======================================================
## ts_d1_quarterly_growth = ts_d1_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d1_quarterly_growth.l1  -0.39237    0.21007  -1.868   0.0752 .
## rec_dummy_rev_growth_q.l1   0.20667    1.65046   0.125   0.9015
## log_rev_quarter_adj.l1     -1.99453    3.65864  -0.545   0.5911
## rest_real_exp_diff_log.l1   7.41961   54.75724   0.136   0.8934
## ts_d1_quarterly_growth.l2  -0.14804    0.22374  -0.662   0.5151
## rec_dummy_rev_growth_q.l2   0.01848    1.95556   0.009   0.9925
```

```
## log_rev_quarter_adj.l2      2.50226    2.75682   0.908   0.3739
## rest_real_exp_diff_log.l2 -50.59090   52.04671  -0.972   0.3416
## ts_d1_quarterly_growth.l3   0.02531    0.23000   0.110   0.9134
## rec_dummy_rev_growth_q.l3  -0.69983    1.79373  -0.390   0.7002
## log_rev_quarter_adj.l3      0.58370    2.87070   0.203   0.8407
## rest_real_exp_diff_log.l3   3.76477   45.88509   0.082   0.9354
## ts_d1_quarterly_growth.l4  -0.08937    0.22677  -0.394   0.6973
## rec_dummy_rev_growth_q.l4  -0.16531    1.57689  -0.105   0.9175
## log_rev_quarter_adj.l4     -2.96767    3.30451  -0.898   0.3789
## rest_real_exp_diff_log.l4   6.12182   40.77106   0.150   0.8820
## const                       0.24189    0.72519   0.334   0.7419
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.642 on 22 degrees of freedom
## Multiple R-Squared: 0.2136,  Adjusted R-squared: -0.3583
## F-statistic: 0.3735 on 16 and 22 DF,  p-value: 0.9761
##
##
## Estimation results for equation rec_dummy_rev_growth_q:
## =========================================================
## rec_dummy_rev_growth_q = ts_d1_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj
##
##                            Estimate Std. Error t value Pr(>|t|)
## ts_d1_quarterly_growth.l1 -0.014485   0.023514  -0.616  0.54420
## rec_dummy_rev_growth_q.l1  0.605706   0.184742   3.279  0.00343 **
## log_rev_quarter_adj.l1     0.150996   0.409524   0.369  0.71587
## rest_real_exp_diff_log.l1 -2.374127   6.129166  -0.387  0.70222
## ts_d1_quarterly_growth.l2 -0.002676   0.025044  -0.107  0.91587
## rec_dummy_rev_growth_q.l2  0.046828   0.218893   0.214  0.83257
## log_rev_quarter_adj.l2    -0.600262   0.308580  -1.945  0.06464 .
## rest_real_exp_diff_log.l2  8.419212   5.825767   1.445  0.16251
## ts_d1_quarterly_growth.l3 -0.018486   0.025745  -0.718  0.48029
## rec_dummy_rev_growth_q.l3  0.229900   0.200779   1.145  0.26450
## log_rev_quarter_adj.l3     0.947515   0.321327   2.949  0.00742 **
## rest_real_exp_diff_log.l3  0.605286   5.136076   0.118  0.90726
## ts_d1_quarterly_growth.l4  0.012198   0.025383   0.481  0.63559
## rec_dummy_rev_growth_q.l4 -0.106610   0.176507  -0.604  0.55202
## log_rev_quarter_adj.l4     0.829803   0.369885   2.243  0.03527 *
## rest_real_exp_diff_log.l4  6.325242   4.563646   1.386  0.17963
## const                     -0.026665   0.081173  -0.329  0.74564
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.1838 on 22 degrees of freedom
## Multiple R-Squared: 0.8706,  Adjusted R-squared: 0.7765
## F-statistic: 9.252 on 16 and 22 DF,  p-value: 2.243e-06
##
##
## Estimation results for equation log_rev_quarter_adj:
## ======================================================
## log_rev_quarter_adj = ts_d1_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj.l1
```

```
##
##                            Estimate Std. Error t value Pr(>|t|)
## ts_d1_quarterly_growth.l1 -0.0008878  0.0089122  -0.100   0.9216
## rec_dummy_rev_growth_q.l1 -0.0036114  0.0700200  -0.052   0.9593
## log_rev_quarter_adj.l1     0.1139951  0.1552159   0.734   0.4704
## rest_real_exp_diff_log.l1 -3.8560574  2.3230467  -1.660   0.1111
## ts_d1_quarterly_growth.l2  0.0013876  0.0094921   0.146   0.8851
## rec_dummy_rev_growth_q.l2  0.0517908  0.0829637   0.624   0.5389
## log_rev_quarter_adj.l2     0.2252633  0.1169567   1.926   0.0671 .
## rest_real_exp_diff_log.l2  1.9645377  2.2080541   0.890   0.3832
## ts_d1_quarterly_growth.l3 -0.0051530  0.0097578  -0.528   0.6027
## rec_dummy_rev_growth_q.l3 -0.0687385  0.0760981  -0.903   0.3762
## log_rev_quarter_adj.l3    -0.1498326  0.1217880  -1.230   0.2316
## rest_real_exp_diff_log.l3 -1.5544668  1.9466508  -0.799   0.4331
## ts_d1_quarterly_growth.l4  0.0103243  0.0096206   1.073   0.2948
## rec_dummy_rev_growth_q.l4 -0.0646803  0.0668987  -0.967   0.3441
## log_rev_quarter_adj.l4     0.2586797  0.1401921   1.845   0.0785 .
## rest_real_exp_diff_log.l4 -4.3203752  1.7296908  -2.498   0.0205 *
## const                      0.0194861  0.0307657   0.633   0.5330
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.06966 on 22 degrees of freedom
## Multiple R-Squared: 0.6655,  Adjusted R-squared: 0.4222
## F-statistic: 2.736 on 16 and 22 DF,  p-value: 0.01484
##
##
## Estimation results for equation rest_real_exp_diff_log:
## =======================================================
## rest_real_exp_diff_log = ts_d1_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj
##
##                            Estimate Std. Error t value Pr(>|t|)
## ts_d1_quarterly_growth.l1  3.677e-05  6.894e-04   0.053  0.95794
## rec_dummy_rev_growth_q.l1 -1.589e-02  5.416e-03  -2.933  0.00769 **
## log_rev_quarter_adj.l1     6.315e-03  1.201e-02   0.526  0.60419
## rest_real_exp_diff_log.l1 -2.285e-01  1.797e-01  -1.272  0.21676
## ts_d1_quarterly_growth.l2 -7.658e-04  7.342e-04  -1.043  0.30830
## rec_dummy_rev_growth_q.l2  2.434e-03  6.417e-03   0.379  0.70817
## log_rev_quarter_adj.l2    -1.092e-02  9.047e-03  -1.207  0.24027
## rest_real_exp_diff_log.l2  1.701e-01  1.708e-01   0.996  0.33006
## ts_d1_quarterly_growth.l3  1.750e-03  7.548e-04   2.318  0.03013 *
## rec_dummy_rev_growth_q.l3 -4.698e-03  5.886e-03  -0.798  0.43338
## log_rev_quarter_adj.l3    -8.656e-03  9.421e-03  -0.919  0.36812
## rest_real_exp_diff_log.l3 -5.106e-02  1.506e-01  -0.339  0.73777
## ts_d1_quarterly_growth.l4  2.348e-03  7.442e-04   3.155  0.00459 **
## rec_dummy_rev_growth_q.l4 -6.471e-04  5.175e-03  -0.125  0.90163
## log_rev_quarter_adj.l4     8.773e-03  1.084e-02   0.809  0.42716
## rest_real_exp_diff_log.l4 -2.895e-02  1.338e-01  -0.216  0.83068
## const                      7.602e-03  2.380e-03   3.195  0.00419 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
```

```
## Residual standard error: 0.005388 on 22 degrees of freedom
## Multiple R-Squared: 0.7637,  Adjusted R-squared: 0.5919
## F-statistic: 4.445 on 16 and 22 DF,  p-value: 0.0007467
##
##
##
## Covariance matrix of residuals:
##                         ts_d1_quarterly_growth rec_dummy_rev_growth_q
## ts_d1_quarterly_growth                2.695990              0.0492205
## rec_dummy_rev_growth_q                0.049221              0.0337783
## log_rev_quarter_adj                   0.034703              0.0023058
## rest_real_exp_diff_log                0.001365             -0.0001474
##                         log_rev_quarter_adj rest_real_exp_diff_log
## ts_d1_quarterly_growth            0.0347028              1.365e-03
## rec_dummy_rev_growth_q            0.0023058             -1.474e-04
## log_rev_quarter_adj               0.0048523              1.705e-04
## rest_real_exp_diff_log            0.0001705              2.903e-05
##
## Correlation matrix of residuals:
##                         ts_d1_quarterly_growth rec_dummy_rev_growth_q
## ts_d1_quarterly_growth                  1.0000                 0.1631
## rec_dummy_rev_growth_q                  0.1631                 1.0000
## log_rev_quarter_adj                     0.3034                 0.1801
## rest_real_exp_diff_log                  0.1543                -0.1488
##                         log_rev_quarter_adj rest_real_exp_diff_log
## ts_d1_quarterly_growth               0.3034                 0.1543
## rec_dummy_rev_growth_q               0.1801                -0.1488
## log_rev_quarter_adj                  1.0000                 0.4542
## rest_real_exp_diff_log               0.4542                 1.0000
```

```
combinedFull2 = cbind(ts_d2_quarterly_growth, rec_dummy_rev_growth_q,
    log_rev_quarter_adj, rest_real_exp_diff_log)
select = VARselect(combinedFull2, lag.max = 4, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
fullVar2 = VAR(combinedFull2, p = select$select[1])
plot(fullVar2$y)
```




```r
summary(fullVar2)
```

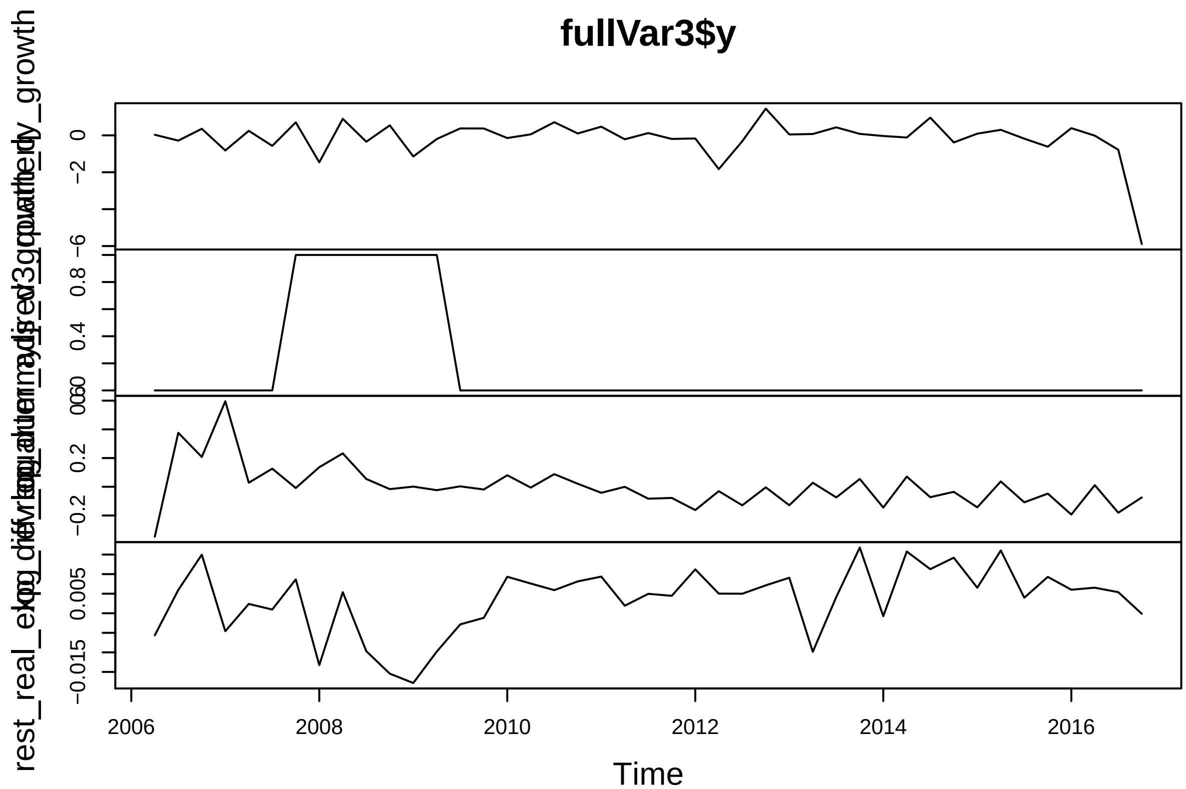
```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: ts_d2_quarterly_growth, rec_dummy_rev_growth_q, log_rev_quarter_adj, rest_real_
## Deterministic variables: const
## Sample size: 39
## Log Likelihood: 181.982
## Roots of the characteristic polynomial:
## 0.894 0.8626 0.8626 0.8132 0.791 0.791 0.7742 0.7742 0.7738 0.7738 0.7495 0.7495 0.6855 0.6855 0.5475
## Call:
## VAR(y = combinedFull2, p = select$select[1])
##
##
## Estimation results for equation ts_d2_quarterly_growth:
## =======================================================
## ts_d2_quarterly_growth = ts_d2_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj
##
##                            Estimate Std. Error t value Pr(>|t|)
## ts_d2_quarterly_growth.l1  -0.52271    0.20862  -2.506   0.0201 *
## rec_dummy_rev_growth_q.l1   0.39807    1.84446   0.216   0.8311
## log_rev_quarter_adj.l1     -2.57580    4.10628  -0.627   0.5369
## rest_real_exp_diff_log.l1   3.78776   58.98050   0.064   0.9494
## ts_d2_quarterly_growth.l2  -0.19242    0.23281  -0.827   0.4174
## rec_dummy_rev_growth_q.l2  -0.21258    2.17134  -0.098   0.9229
```

```
## log_rev_quarter_adj.l2      3.07380    3.09641   0.993   0.3316
## rest_real_exp_diff_log.l2 -57.99457   58.16599  -0.997   0.3296
## ts_d2_quarterly_growth.l3  -0.07908    0.24067  -0.329   0.7456
## rec_dummy_rev_growth_q.l3  -0.62020    1.99799  -0.310   0.7592
## log_rev_quarter_adj.l3      0.55518    3.21460   0.173   0.8645
## rest_real_exp_diff_log.l3   3.26463   51.12624   0.064   0.9497
## ts_d2_quarterly_growth.l4  -0.11251    0.21840  -0.515   0.6116
## rec_dummy_rev_growth_q.l4  -0.35629    1.75776  -0.203   0.8412
## log_rev_quarter_adj.l4     -3.55627    3.69006  -0.964   0.3457
## rest_real_exp_diff_log.l4   5.03728   45.52213   0.111   0.9129
## const                       0.31792    0.81106   0.392   0.6988
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.837 on 22 degrees of freedom
## Multiple R-Squared: 0.2937,  Adjusted R-squared: -0.2201
## F-statistic: 0.5716 on 16 and 22 DF,  p-value: 0.8726
##
##
## Estimation results for equation rec_dummy_rev_growth_q:
## ==========================================================
## rec_dummy_rev_growth_q = ts_d2_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj
##
##                            Estimate Std. Error t value Pr(>|t|)
## ts_d2_quarterly_growth.l1 -0.012174   0.020840  -0.584  0.56505
## rec_dummy_rev_growth_q.l1  0.607623   0.184253   3.298  0.00328 **
## log_rev_quarter_adj.l1     0.161916   0.410197   0.395  0.69684
## rest_real_exp_diff_log.l1 -1.902341   5.891866  -0.323  0.74984
## ts_d2_quarterly_growth.l2 -0.002184   0.023256  -0.094  0.92602
## rec_dummy_rev_growth_q.l2  0.052259   0.216906   0.241  0.81184
## log_rev_quarter_adj.l2    -0.612937   0.309316  -1.982  0.06016 .
## rest_real_exp_diff_log.l2  8.566444   5.810500   1.474  0.15457
## ts_d2_quarterly_growth.l3 -0.013764   0.024042  -0.572  0.57280
## rec_dummy_rev_growth_q.l3  0.227376   0.199589   1.139  0.26686
## log_rev_quarter_adj.l3     0.958293   0.321123   2.984  0.00684 **
## rest_real_exp_diff_log.l3  0.448569   5.107264   0.088  0.93081
## ts_d2_quarterly_growth.l4  0.010759   0.021817   0.493  0.62679
## rec_dummy_rev_growth_q.l4 -0.098404   0.175592  -0.560  0.58086
## log_rev_quarter_adj.l4     0.827238   0.368619   2.244  0.03521 *
## rest_real_exp_diff_log.l4  6.553441   4.547439   1.441  0.16363
## const                     -0.031452   0.081021  -0.388  0.70161
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.1835 on 22 degrees of freedom
## Multiple R-Squared: 0.8711,  Adjusted R-squared: 0.7773
## F-statistic: 9.291 on 16 and 22 DF,  p-value: 2.163e-06
##
##
## Estimation results for equation log_rev_quarter_adj:
## =======================================================
## log_rev_quarter_adj = ts_d2_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj.l1
```

```
##
##                            Estimate Std. Error t value Pr(>|t|)
## ts_d2_quarterly_growth.l1 -0.0009688  0.0078996  -0.123   0.9035
## rec_dummy_rev_growth_q.l1 -0.0018217  0.0698434  -0.026   0.9794
## log_rev_quarter_adj.l1     0.1184337  0.1554907   0.762   0.4543
## rest_real_exp_diff_log.l1 -3.4482735  2.2333922  -1.544   0.1369
## ts_d2_quarterly_growth.l2  0.0023880  0.0088156   0.271   0.7890
## rec_dummy_rev_growth_q.l2  0.0548158  0.0822213   0.667   0.5119
## log_rev_quarter_adj.l2     0.2218344  0.1172505   1.892   0.0717 .
## rest_real_exp_diff_log.l2  1.9037095  2.2025493   0.864   0.3967
## ts_d2_quarterly_growth.l3 -0.0036456  0.0091133  -0.400   0.6930
## rec_dummy_rev_growth_q.l3 -0.0705082  0.0756571  -0.932   0.3615
## log_rev_quarter_adj.l3    -0.1426758  0.1217262  -1.172   0.2537
## rest_real_exp_diff_log.l3 -1.6379576  1.9359779  -0.846   0.4066
## ts_d2_quarterly_growth.l4  0.0077853  0.0082701   0.941   0.3567
## rec_dummy_rev_growth_q.l4 -0.0606317  0.0665606  -0.911   0.3722
## log_rev_quarter_adj.l4     0.2546272  0.1397302   1.822   0.0820 .
## rest_real_exp_diff_log.l4 -4.1448418  1.7237689  -2.405   0.0251 *
## const                      0.0166747  0.0307122   0.543   0.5926
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.06954 on 22 degrees of freedom
## Multiple R-Squared: 0.6666,  Adjusted R-squared: 0.4241
## F-statistic: 2.749 on 16 and 22 DF,  p-value: 0.01445
##
##
## Estimation results for equation rest_real_exp_diff_log:
## =======================================================
## rest_real_exp_diff_log = ts_d2_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj
##
##                            Estimate Std. Error t value Pr(>|t|)
## ts_d2_quarterly_growth.l1  2.745e-06  6.181e-04   0.004  0.99650
## rec_dummy_rev_growth_q.l1 -1.589e-02  5.464e-03  -2.908  0.00815 **
## log_rev_quarter_adj.l1     6.613e-03  1.217e-02   0.544  0.59221
## rest_real_exp_diff_log.l1 -1.965e-01  1.747e-01  -1.125  0.27287
## ts_d2_quarterly_growth.l2 -6.936e-04  6.897e-04  -1.006  0.32555
## rec_dummy_rev_growth_q.l2  2.583e-03  6.433e-03   0.401  0.69195
## log_rev_quarter_adj.l2    -1.204e-02  9.174e-03  -1.313  0.20278
## rest_real_exp_diff_log.l2  1.851e-01  1.723e-01   1.074  0.29434
## ts_d2_quarterly_growth.l3  1.321e-03  7.130e-04   1.853  0.07732 .
## rec_dummy_rev_growth_q.l3 -4.089e-03  5.919e-03  -0.691  0.49691
## log_rev_quarter_adj.l3    -7.687e-03  9.524e-03  -0.807  0.42823
## rest_real_exp_diff_log.l3 -4.114e-02  1.515e-01  -0.272  0.78843
## ts_d2_quarterly_growth.l4  2.234e-03  6.470e-04   3.453  0.00226 **
## rec_dummy_rev_growth_q.l4 -5.668e-04  5.208e-03  -0.109  0.91431
## log_rev_quarter_adj.l4     9.346e-03  1.093e-02   0.855  0.40180
## rest_real_exp_diff_log.l4 -1.979e-02  1.349e-01  -0.147  0.88470
## const                      7.202e-03  2.403e-03   2.997  0.00664 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
```

```
## Residual standard error: 0.005441 on 22 degrees of freedom
## Multiple R-Squared: 0.7591,  Adjusted R-squared: 0.5839
## F-statistic: 4.333 on 16 and 22 DF,  p-value: 0.0008908
##
##
##
## Covariance matrix of residuals:
##                        ts_d2_quarterly_growth rec_dummy_rev_growth_q
## ts_d2_quarterly_growth                3.37281              0.0512481
## rec_dummy_rev_growth_q                0.05125              0.0336574
## log_rev_quarter_adj                   0.03902              0.0022028
## rest_real_exp_diff_log                0.00171             -0.0001891
##                        log_rev_quarter_adj rest_real_exp_diff_log
## ts_d2_quarterly_growth           0.0390199              0.0017101
## rec_dummy_rev_growth_q           0.0022028             -0.0001891
## log_rev_quarter_adj              0.0048362              0.0001675
## rest_real_exp_diff_log           0.0001675              0.0000296
##
## Correlation matrix of residuals:
##                        ts_d2_quarterly_growth rec_dummy_rev_growth_q
## ts_d2_quarterly_growth                 1.0000                 0.1521
## rec_dummy_rev_growth_q                 0.1521                 1.0000
## log_rev_quarter_adj                    0.3055                 0.1727
## rest_real_exp_diff_log                 0.1711                -0.1894
##                        log_rev_quarter_adj rest_real_exp_diff_log
## ts_d2_quarterly_growth              0.3055                 0.1711
## rec_dummy_rev_growth_q              0.1727                -0.1894
## log_rev_quarter_adj                 1.0000                 0.4428
## rest_real_exp_diff_log              0.4428                 1.0000
```

```r
combinedFull3 = cbind(ts_d3_quarterly_growth, rec_dummy_rev_growth_q,
    log_rev_quarter_adj, rest_real_exp_diff_log)
select = VARselect(combinedFull3, lag.max = 4, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
fullVar3 = VAR(combinedFull3, p = select$select[1])
plot(fullVar3$y)
```




```r
summary(fullVar3)
```

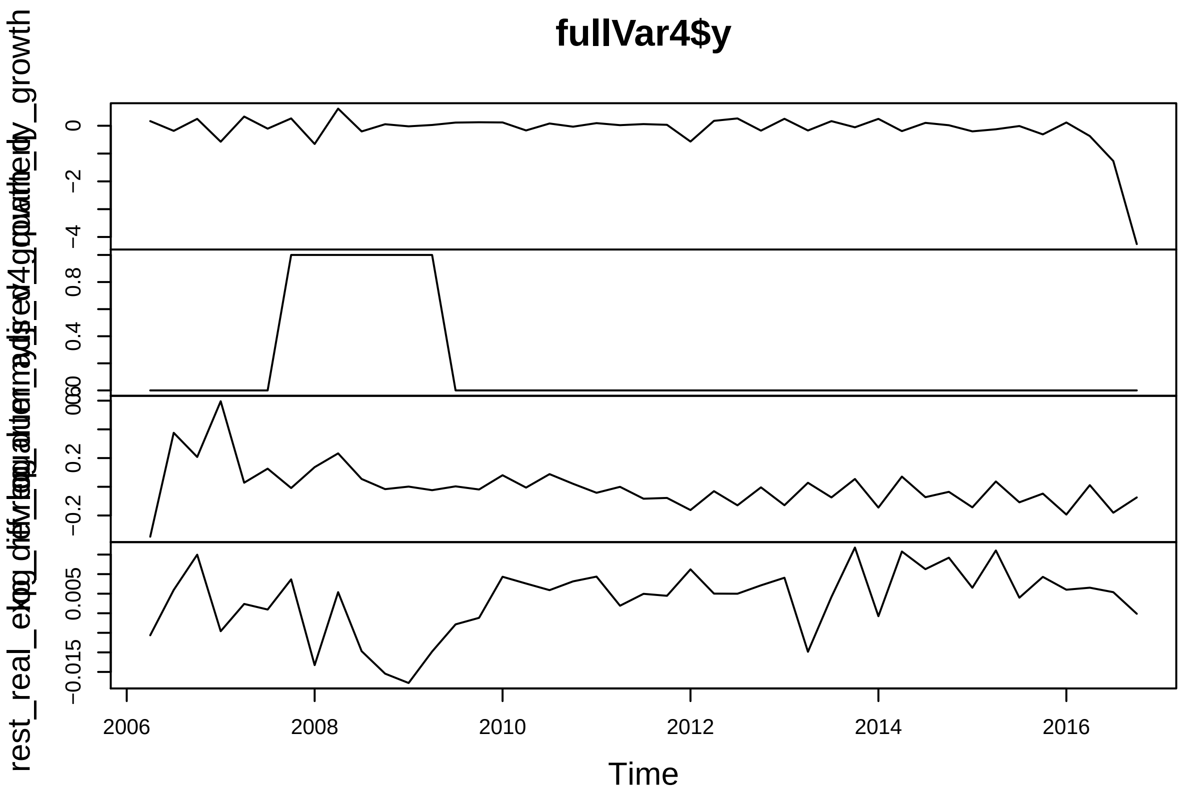
```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: ts_d3_quarterly_growth, rec_dummy_rev_growth_q, log_rev_quarter_adj, rest_real_
## Deterministic variables: const
## Sample size: 39
## Log Likelihood: 188.659
## Roots of the characteristic polynomial:
## 0.9017 0.9017 0.8734 0.8734 0.8291 0.8035 0.8035 0.7893 0.7664 0.7664 0.746 0.746 0.619 0.619 0.5303
## Call:
## VAR(y = combinedFull3, p = select$select[1])
##
##
## Estimation results for equation ts_d3_quarterly_growth:
## =======================================================
## ts_d3_quarterly_growth = ts_d3_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj
##
##                            Estimate Std. Error t value Pr(>|t|)
## ts_d3_quarterly_growth.l1  0.33280    0.40295   0.826    0.418
## rec_dummy_rev_growth_q.l1 -0.42115    1.37194  -0.307    0.762
## log_rev_quarter_adj.l1     1.37316    2.95476   0.465    0.647
## rest_real_exp_diff_log.l1 -6.31688   41.13913  -0.154    0.879
## ts_d3_quarterly_growth.l2 -0.18879    0.41359  -0.456    0.653
## rec_dummy_rev_growth_q.l2  0.75936    1.60170   0.474    0.640
```

```
## log_rev_quarter_adj.l2    -1.38866    2.15979  -0.643    0.527
## rest_real_exp_diff_log.l2 22.22430   41.59467   0.534    0.598
## ts_d3_quarterly_growth.l3 -0.07205    0.40761  -0.177    0.861
## rec_dummy_rev_growth_q.l3 -0.44422    1.49665  -0.297    0.769
## log_rev_quarter_adj.l3     1.62349    2.31810   0.700    0.491
## rest_real_exp_diff_log.l3  4.96137   37.31844   0.133    0.895
## ts_d3_quarterly_growth.l4  0.48821    0.41504   1.176    0.252
## rec_dummy_rev_growth_q.l4  0.48682    1.30045   0.374    0.712
## log_rev_quarter_adj.l4    -0.05672    2.64174  -0.021    0.983
## rest_real_exp_diff_log.l4 -7.08428   34.78237  -0.204    0.840
## const                     -0.27696    0.58018  -0.477    0.638
##
##
## Residual standard error: 1.295 on 22 degrees of freedom
## Multiple R-Squared: 0.2399,  Adjusted R-squared: -0.3129
## F-statistic: 0.434 on 16 and 22 DF,  p-value: 0.9543
##
##
## Estimation results for equation rec_dummy_rev_growth_q:
## =======================================================
## rec_dummy_rev_growth_q = ts_d3_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj
##
##                           Estimate Std. Error t value Pr(>|t|)
## ts_d3_quarterly_growth.l1 -0.02021    0.05562  -0.363  0.71975
## rec_dummy_rev_growth_q.l1  0.59761    0.18937   3.156  0.00459 **
## log_rev_quarter_adj.l1     0.09656    0.40786   0.237  0.81505
## rest_real_exp_diff_log.l1 -0.93702    5.67861  -0.165  0.87044
## ts_d3_quarterly_growth.l2  0.03777    0.05709   0.662  0.51514
## rec_dummy_rev_growth_q.l2  0.07198    0.22109   0.326  0.74783
## log_rev_quarter_adj.l2    -0.49055    0.29812  -1.645  0.11409
## rest_real_exp_diff_log.l2  5.32726    5.74149   0.928  0.36355
## ts_d3_quarterly_growth.l3 -0.07109    0.05626  -1.264  0.21960
## rec_dummy_rev_growth_q.l3  0.21564    0.20659   1.044  0.30790
## log_rev_quarter_adj.l3     0.85442    0.31998   2.670  0.01398 *
## rest_real_exp_diff_log.l3  2.60211    5.15122   0.505  0.61848
## ts_d3_quarterly_growth.l4 -0.03130    0.05729  -0.546  0.59028
## rec_dummy_rev_growth_q.l4 -0.13219    0.17951  -0.736  0.46925
## log_rev_quarter_adj.l4     0.86377    0.36465   2.369  0.02705 *
## rest_real_exp_diff_log.l4  5.36441    4.80116   1.117  0.27591
## const                     -0.02181    0.08008  -0.272  0.78790
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.1788 on 22 degrees of freedom
## Multiple R-Squared: 0.8775,  Adjusted R-squared: 0.7885
## F-statistic: 9.853 on 16 and 22 DF,  p-value: 1.288e-06
##
##
## Estimation results for equation log_rev_quarter_adj:
## ====================================================
## log_rev_quarter_adj = ts_d3_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj.l1
##
##                             Estimate Std. Error t value Pr(>|t|)
```

```
## ts_d3_quarterly_growth.l1  1.371e-02  2.243e-02   0.611   0.5472
## rec_dummy_rev_growth_q.l1  1.911e-03  7.635e-02   0.025   0.9803
## log_rev_quarter_adj.l1     8.369e-02  1.644e-01   0.509   0.6159
## rest_real_exp_diff_log.l1 -2.852e+00  2.290e+00  -1.246   0.2259
## ts_d3_quarterly_growth.l2 -8.758e-04  2.302e-02  -0.038   0.9700
## rec_dummy_rev_growth_q.l2  7.818e-02  8.914e-02   0.877   0.3899
## log_rev_quarter_adj.l2     2.562e-01  1.202e-01   2.131   0.0445 *
## rest_real_exp_diff_log.l2  1.388e+00  2.315e+00   0.600   0.5549
## ts_d3_quarterly_growth.l3  1.295e-03  2.268e-02   0.057   0.9550
## rec_dummy_rev_growth_q.l3 -1.038e-01  8.329e-02  -1.246   0.2259
## log_rev_quarter_adj.l3    -1.293e-01  1.290e-01  -1.002   0.3270
## rest_real_exp_diff_log.l3 -1.747e+00  2.077e+00  -0.841   0.4094
## ts_d3_quarterly_growth.l4 -2.059e-05  2.310e-02  -0.001   0.9993
## rec_dummy_rev_growth_q.l4 -5.037e-02  7.237e-02  -0.696   0.4937
## log_rev_quarter_adj.l4     2.213e-01  1.470e-01   1.505   0.1465
## rest_real_exp_diff_log.l4 -4.173e+00  1.936e+00  -2.156   0.0423 *
## const                      1.650e-02  3.229e-02   0.511   0.6143
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.07209 on 22 degrees of freedom
## Multiple R-Squared: 0.6417,  Adjusted R-squared: 0.3812
## F-statistic: 2.463 on 16 and 22 DF,  p-value: 0.02544
##
##
## Estimation results for equation rest_real_exp_diff_log:
## =========================================================
## rest_real_exp_diff_log = ts_d3_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj
##
##                             Estimate Std. Error t value Pr(>|t|)
## ts_d3_quarterly_growth.l1  8.937e-04  1.981e-03   0.451   0.6563
## rec_dummy_rev_growth_q.l1 -1.182e-02  6.744e-03  -1.753   0.0936 .
## log_rev_quarter_adj.l1     4.619e-04  1.453e-02   0.032   0.9749
## rest_real_exp_diff_log.l1 -2.030e-01  2.022e-01  -1.004   0.3264
## ts_d3_quarterly_growth.l2 -1.789e-03  2.033e-03  -0.880   0.3885
## rec_dummy_rev_growth_q.l2 -1.776e-03  7.874e-03  -0.226   0.8236
## log_rev_quarter_adj.l2    -8.063e-03  1.062e-02  -0.759   0.4557
## rest_real_exp_diff_log.l2  1.295e-01  2.045e-01   0.633   0.5331
## ts_d3_quarterly_growth.l3  5.144e-04  2.004e-03   0.257   0.7998
## rec_dummy_rev_growth_q.l3 -4.034e-03  7.358e-03  -0.548   0.5890
## log_rev_quarter_adj.l3    -5.028e-03  1.140e-02  -0.441   0.6634
## rest_real_exp_diff_log.l3  3.484e-04  1.835e-01   0.002   0.9985
## ts_d3_quarterly_growth.l4  4.522e-03  2.040e-03   2.216   0.0373 *
## rec_dummy_rev_growth_q.l4 -5.889e-05  6.393e-03  -0.009   0.9927
## log_rev_quarter_adj.l4     2.537e-03  1.299e-02   0.195   0.8469
## rest_real_exp_diff_log.l4 -8.521e-02  1.710e-01  -0.498   0.6232
## const                      7.534e-03  2.852e-03   2.642   0.0149 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.006368 on 22 degrees of freedom
## Multiple R-Squared:  0.67,   Adjusted R-squared:  0.43
```

```
## F-statistic: 2.792 on 16 and 22 DF,  p-value: 0.01331
##
##
##
## Covariance matrix of residuals:
##                          ts_d3_quarterly_growth rec_dummy_rev_growth_q
## ts_d3_quarterly_growth                 1.677980             -3.447e-02
## rec_dummy_rev_growth_q                -0.034469              3.197e-02
## log_rev_quarter_adj                    0.026250              3.294e-03
## rest_real_exp_diff_log                 0.002096             -6.132e-05
##                          log_rev_quarter_adj rest_real_exp_diff_log
## ts_d3_quarterly_growth             0.0262505              2.096e-03
## rec_dummy_rev_growth_q             0.0032939             -6.132e-05
## log_rev_quarter_adj                0.0051971              1.876e-04
## rest_real_exp_diff_log             0.0001876              4.055e-05
##
## Correlation matrix of residuals:
##                          ts_d3_quarterly_growth rec_dummy_rev_growth_q
## ts_d3_quarterly_growth                   1.0000               -0.14882
## rec_dummy_rev_growth_q                  -0.1488                1.00000
## log_rev_quarter_adj                      0.2811                0.25553
## rest_real_exp_diff_log                   0.2541               -0.05385
##                          log_rev_quarter_adj rest_real_exp_diff_log
## ts_d3_quarterly_growth                0.2811                0.25413
## rec_dummy_rev_growth_q                0.2555               -0.05385
## log_rev_quarter_adj                   1.0000                0.40858
## rest_real_exp_diff_log                0.4086                1.00000
```

```
combinedFull4 = cbind(ts_d4_quarterly_growth, rec_dummy_rev_growth_q,
    log_rev_quarter_adj, rest_real_exp_diff_log)
select = VARselect(combinedFull4, lag.max = 4, type = c("const",
    "trend", "both", "none"), season = NULL, exogen = NULL)
fullVar4 = VAR(combinedFull4, p = select$select[1])
plot(fullVar4$y)
```




```r
summary(fullVar4)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: ts_d4_quarterly_growth, rec_dummy_rev_growth_q, log_rev_quarter_adj, rest_real_
## Deterministic variables: const
## Sample size: 40
## Log Likelihood: 195.241
## Roots of the characteristic polynomial:
##  2.39 0.8432 0.8176 0.8056 0.8056 0.7458 0.6579 0.6579 0.5647 0.5647 0.5301 0.5301
## Call:
## VAR(y = combinedFull4, p = select$select[1])
##
##
## Estimation results for equation ts_d4_quarterly_growth:
## =======================================================
## ts_d4_quarterly_growth = ts_d4_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj
##
##                            Estimate Std. Error t value Pr(>|t|)
## ts_d4_quarterly_growth.l1  1.68864    0.31811   5.308 1.33e-05 ***
## rec_dummy_rev_growth_q.l1 -0.51733    0.42778  -1.209   0.2370
## log_rev_quarter_adj.l1     0.96267    0.81316   1.184   0.2468
## rest_real_exp_diff_log.l1 -6.22896   16.16414  -0.385   0.7030
## ts_d4_quarterly_growth.l2  1.35918    0.51776   2.625   0.0141 *
## rec_dummy_rev_growth_q.l2  1.08276    0.60012   1.804   0.0824 .
```

```
## log_rev_quarter_adj.l2    -0.96652    0.72588  -1.332   0.1942
## rest_real_exp_diff_log.l2 12.05330   15.43433   0.781   0.4416
## ts_d4_quarterly_growth.l3  0.55051    0.46724   1.178   0.2490
## rec_dummy_rev_growth_q.l3 -0.53813    0.50558  -1.064   0.2966
## log_rev_quarter_adj.l3     1.09532    0.75325   1.454   0.1574
## rest_real_exp_diff_log.l3  4.10319   13.86476   0.296   0.7695
## const                     -0.09813    0.20383  -0.481   0.6341
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.5485 on 27 degrees of freedom
## Multiple R-Squared: 0.6174,  Adjusted R-squared: 0.4474
## F-statistic: 3.631 on 12 and 27 DF,  p-value: 0.002614
##
##
## Estimation results for equation rec_dummy_rev_growth_q:
## =======================================================
## rec_dummy_rev_growth_q = ts_d4_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj
##
##                             Estimate Std. Error t value Pr(>|t|)
## ts_d4_quarterly_growth.l1  0.048077   0.118906   0.404   0.6892
## rec_dummy_rev_growth_q.l1  0.891386   0.159900   5.575 6.53e-06 ***
## log_rev_quarter_adj.l1    -0.231499   0.303948  -0.762   0.4529
## rest_real_exp_diff_log.l1  3.716201   6.041919   0.615   0.5437
## ts_d4_quarterly_growth.l2 -0.120744   0.193530  -0.624   0.5379
## rec_dummy_rev_growth_q.l2 -0.005429   0.224318  -0.024   0.9809
## log_rev_quarter_adj.l2     0.052120   0.271322   0.192   0.8491
## rest_real_exp_diff_log.l2  3.500149   5.769127   0.607   0.5491
## ts_d4_quarterly_growth.l3 -0.180155   0.174648  -1.032   0.3114
## rec_dummy_rev_growth_q.l3 -0.075004   0.188980  -0.397   0.6946
## log_rev_quarter_adj.l3     0.914490   0.281555   3.248   0.0031 **
## rest_real_exp_diff_log.l3 -2.339785   5.182443  -0.451   0.6552
## const                      0.007739   0.076190   0.102   0.9198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.205 on 27 degrees of freedom
## Multiple R-Squared: 0.8035,  Adjusted R-squared: 0.7162
## F-statistic:   9.2 on 12 and 27 DF,  p-value: 1.054e-06
##
##
## Estimation results for equation log_rev_quarter_adj:
## ====================================================
## log_rev_quarter_adj = ts_d4_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj.l1
##
##                            Estimate Std. Error t value Pr(>|t|)
## ts_d4_quarterly_growth.l1  0.04453    0.05360   0.831 0.413430
## rec_dummy_rev_growth_q.l1  0.04661    0.07209   0.647 0.523317
## log_rev_quarter_adj.l1     0.25786    0.13702   1.882 0.070677 .
## rest_real_exp_diff_log.l1 -1.99196    2.72380  -0.731 0.470887
## ts_d4_quarterly_growth.l2  0.07808    0.08725   0.895 0.378736
## rec_dummy_rev_growth_q.l2  0.03687    0.10113   0.365 0.718248
```

```
## log_rev_quarter_adj.l2     0.52446    0.12232   4.288 0.000206 ***
## rest_real_exp_diff_log.l2 -0.34677    2.60082  -0.133 0.894922
## ts_d4_quarterly_growth.l3  0.10650    0.07873   1.353 0.187378
## rec_dummy_rev_growth_q.l3 -0.14361    0.08520  -1.686 0.103392
## log_rev_quarter_adj.l3    -0.35931    0.12693  -2.831 0.008662 **
## rest_real_exp_diff_log.l3 -3.50770    2.33633  -1.501 0.144866
## const                      0.02297    0.03435   0.669 0.509261
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.09242 on 27 degrees of freedom
## Multiple R-Squared: 0.666,   Adjusted R-squared: 0.5175
## F-statistic: 4.486 on 12 and 27 DF,  p-value: 0.0005842
##
##
## Estimation results for equation rest_real_exp_diff_log:
## =======================================================
## rest_real_exp_diff_log = ts_d4_quarterly_growth.l1 + rec_dummy_rev_growth_q.l1 + log_rev_quarter_adj
##
##                             Estimate Std. Error t value Pr(>|t|)
## ts_d4_quarterly_growth.l1  0.0026741  0.0038163   0.701   0.4895
## rec_dummy_rev_growth_q.l1 -0.0142418  0.0051320  -2.775   0.0099 **
## log_rev_quarter_adj.l1    -0.0030925  0.0097553  -0.317   0.7537
## rest_real_exp_diff_log.l1 -0.2117167  0.1939181  -1.092   0.2846
## ts_d4_quarterly_growth.l2  0.0043273  0.0062114   0.697   0.4920
## rec_dummy_rev_growth_q.l2  0.0024098  0.0071996   0.335   0.7404
## log_rev_quarter_adj.l2    -0.0094960  0.0087082  -1.090   0.2851
## rest_real_exp_diff_log.l2 -0.0177628  0.1851627  -0.096   0.9243
## ts_d4_quarterly_growth.l3  0.0007046  0.0056054   0.126   0.9009
## rec_dummy_rev_growth_q.l3 -0.0068884  0.0060654  -1.136   0.2661
## log_rev_quarter_adj.l3    -0.0021987  0.0090366  -0.243   0.8096
## rest_real_exp_diff_log.l3  0.0825638  0.1663328   0.496   0.6236
## const                      0.0076236  0.0024454   3.118   0.0043 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.00658 on 27 degrees of freedom
## Multiple R-Squared: 0.5779,  Adjusted R-squared: 0.3903
## F-statistic: 3.081 on 12 and 27 DF,  p-value: 0.007384
##
##
##
## Covariance matrix of residuals:
##                        ts_d4_quarterly_growth rec_dummy_rev_growth_q
## ts_d4_quarterly_growth              0.3008237             -0.0274882
## rec_dummy_rev_growth_q             -0.0274882              0.0420297
## log_rev_quarter_adj                 0.0037058              0.0076666
## rest_real_exp_diff_log              0.0008317             -0.0001533
##                        log_rev_quarter_adj rest_real_exp_diff_log
## ts_d4_quarterly_growth           3.706e-03              8.317e-04
## rec_dummy_rev_growth_q           7.667e-03             -1.533e-04
## log_rev_quarter_adj              8.542e-03              7.542e-05
```

```
## rest_real_exp_diff_log          7.542e-05             4.330e-05
##
## Correlation matrix of residuals:
##                        ts_d4_quarterly_growth rec_dummy_rev_growth_q
## ts_d4_quarterly_growth                 1.0000                -0.2445
## rec_dummy_rev_growth_q                -0.2445                 1.0000
## log_rev_quarter_adj                    0.0731                 0.4046
## rest_real_exp_diff_log                 0.2305                -0.1136
##                        log_rev_quarter_adj rest_real_exp_diff_log
## ts_d4_quarterly_growth              0.0731                 0.2305
## rec_dummy_rev_growth_q              0.4046                -0.1136
## log_rev_quarter_adj                 1.0000                 0.1240
## rest_real_exp_diff_log              0.1240                 1.0000
```

The results of the VAR models are not very assuring. They don't point to any real causal inferences. More research and work needs to be done to create a truley causal inference model.

Conclusion

The most effective model has been using the amount of reviews by dollar signs and regressing them on the recession and seasonally adjusted new reviews. These models showed that there is a drop in reviews for the higher priced ($$$) and ($$$$) restaurants while there is an increase in ($$) restaurants, with ($) restaurants being unaffected. This means that the ($$) restaurants are a substitute for the more expensive restaurants and that the lowest priced ($) restaurants are not interchangable with the other three categories.

Sentiment analysis on the review text showed that there was a possibility that reviews became more negative and price focused during the recession. However, the differences in sample size and potentially statistically insignificance of the results leave the sentiment analysis without a definite answer.

It was determined that the causality model was not valid and needs more research and work..

Future Work

1. If possible, examine all of Yelp's data, not just the Dataset Challenge.
2. In the event of a future recession, add the data in to create a more comprehensive model.
3. Do a stronger NLP analysis such as bigrams or creating a timeseries of words related to prices.
4. Improve the VAR models.

Sources

YELP: https://www.yelp.com/dataset_challenge
FRED: https://fred.stlouisfed.org/
BEA: 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
Yahoo Finance: https://finance.yahoo.com/quote/YELP?p=YELP