HW03

2023-11-08

Levi Johnson and Logan Rayburn

packages

```
require(quantmod)
```

```
## Loading required package: quantmod

## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

## Loading required package: TTR

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
require(forecast)
```

```
## Loading required package: forecast
```

```
require(fBasics)
```

```
## Loading required package: fBasics

##
## Attaching package: 'fBasics'

## The following object is masked from 'package:TTR':
##
##     volatility
```

```r
require(CADFtest)
```

```
## Loading required package: CADFtest

## Loading required package: dynlm

## Loading required package: sandwich

## Loading required package: tseries

## Loading required package: urca

## Registered S3 methods overwritten by 'CADFtest':
##   method      from
##   bread.mlm   sandwich
##   estfun.mlm sandwich
```

```r
require(urca)
# install.packages("sandwich")
require(sandwich)
# install.packages("lmtest")
require(lmtest)
```

```
## Loading required package: lmtest
```

```r
require(nlme)
```

```
## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:forecast':
##
##     getResponse
```

```r
#install.packages("MTS")
require(MTS)
```

```
## Loading required package: MTS

##
## Attaching package: 'MTS'

## The following object is masked from 'package:TTR':
##
##     VMA
```

```r
require(car)
```

```
## Loading required package: car

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:fBasics':
##
##     densityPlot
```

```r
# install.packages("strucchange")
require(strucchange)
```

```
## Loading required package: strucchange

## Warning: package 'strucchange' was built under R version 4.3.2
```

```r
# install.packages("vars")
require(vars)
```

```
## Loading required package: vars

## Warning: package 'vars' was built under R version 4.3.2

## Loading required package: MASS

##
## Attaching package: 'vars'

## The following object is masked from 'package:MTS':
##
##     VAR
```

```r
require(forecast)
```

Note

Question 1 is a little jank it took me a little to get everything up and running after there it gets better though

Loading in data sources

Going to model Vail Resorts with

Daily crude oil price Weekly

I am having lots of issues with the CrudeOil Price not being the same length or having missing values that don't occur in the other ones

```r
#load in vail resorts
getSymbols("MTN")
```

```
## [1] "MTN"
```

```r
#National Average weekly gass pri
getSymbols("GASREGW", src = "FRED")
```

```
## [1] "GASREGW"
```

```r
getSymbols("DCOILWTICO", src = "FRED")
```

```
## [1] "DCOILWTICO"
```

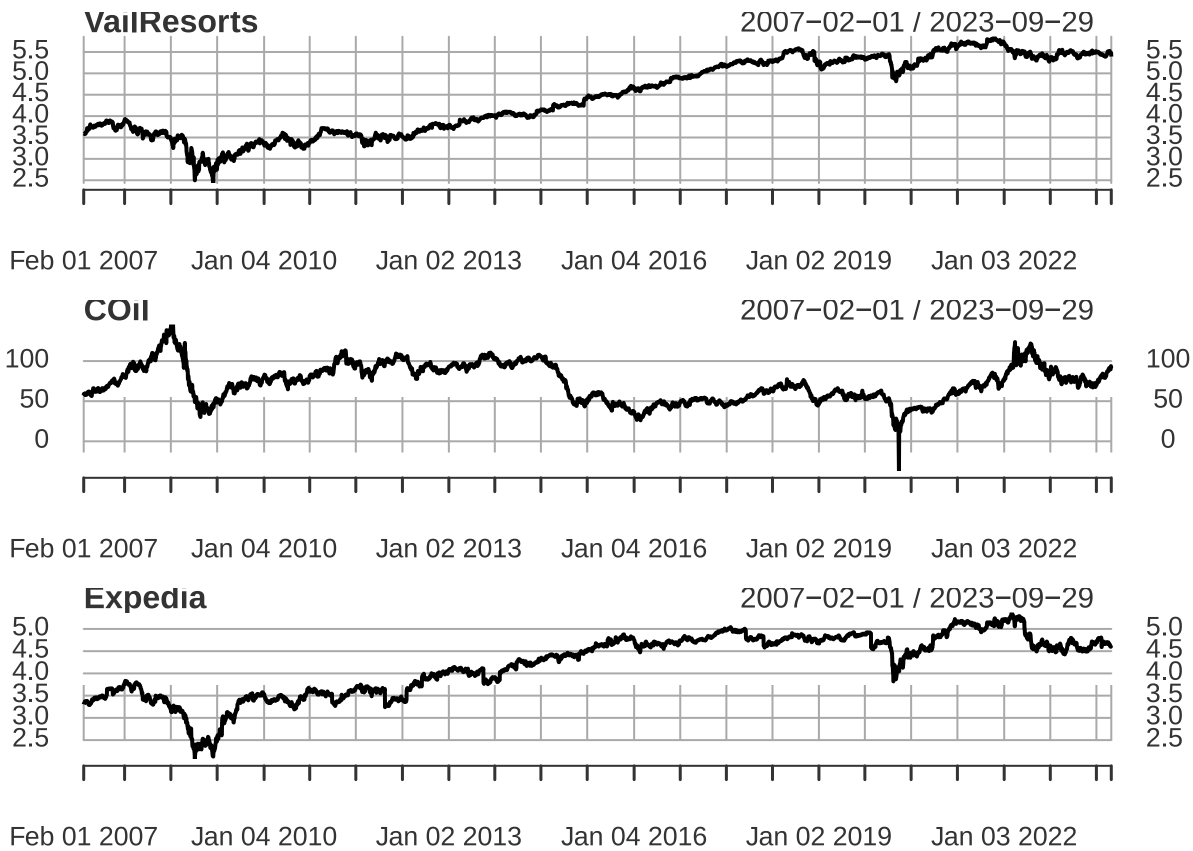
```r
#adding Expedia as it is a travel company, going to be used to model travel
getSymbols("EXPE")
```

```
## [1] "EXPE"
```

formatting and getting rid of NA's

```r
startdate <- "2007-02-01"
enddate <- "2023-9-29"

VailResorts <- na.omit(window(MTN, start = startdate, end = enddate))
COil <- na.omit(window(DCOILWTICO, start = startdate, end = enddate))
Expedia <- na.omit(window(EXPE, start = startdate, end = enddate))
```

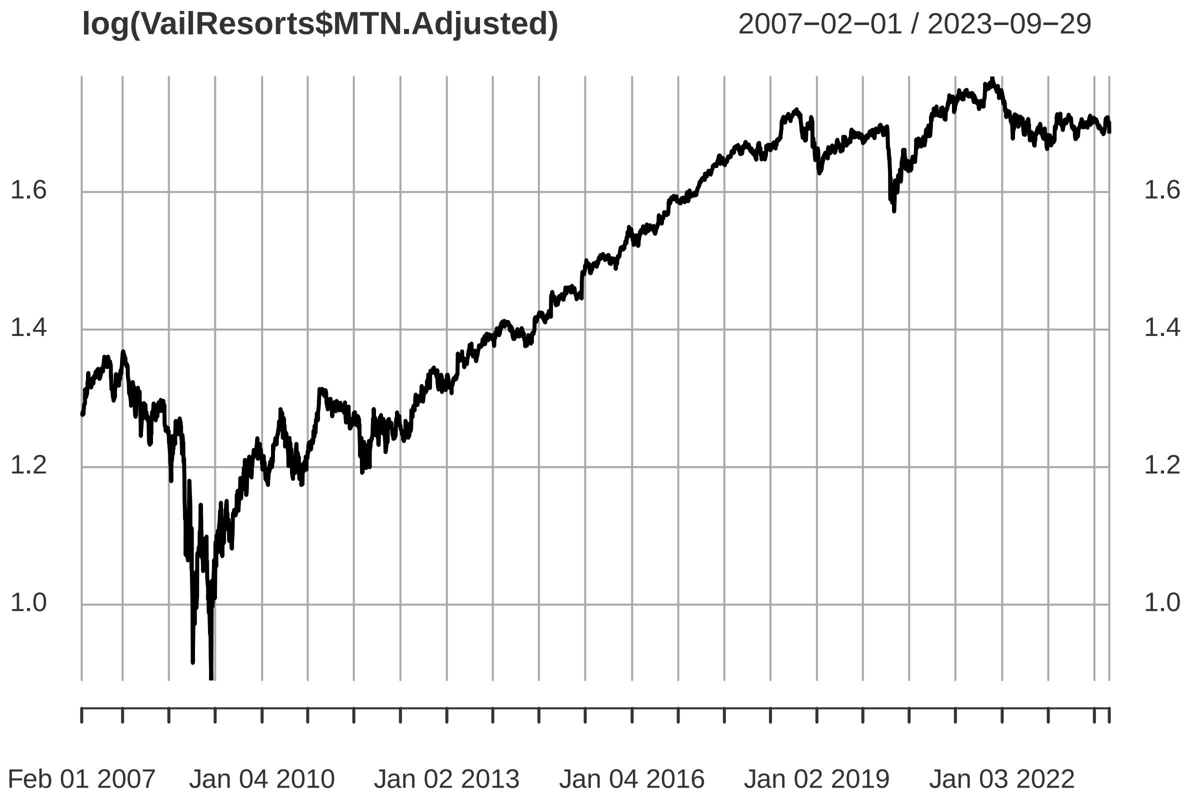
```r
VailResorts <- log(VailResorts$MTN.Adjusted)
Expedia <- log(Expedia$EXPE.Adjusted)
COil <- COil
#crude Oil went negative one day so there will be a missing value within coil
par(mfrow = c(3, 1))
plot(VailResorts)
plot(COil)
plot(Expedia)
```



vailResort:

seems as though there is a trend in the data so case 4.

```
plot(log(VailResorts$MTN.Adjusted))
```



fail to reject the null hypothesis so go to case 2.

```
VailTest <- CADFtest(VailResorts)
summary(VailTest)
```

```
## Augmented DF test
##                                               ADF test
## t-test statistic:                           -2.5873828
## p-value:                                     0.2861699
## Max lag of the diff. dependent variable:     1.0000000
##
## Call:
## dynlm(formula = formula(model), start = obs.1, end = obs.T)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.195792 -0.010275  0.000425  0.010609  0.174189
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.694e-03  3.678e-03   2.636  0.00842 **
## trnd         2.119e-06  8.632e-07   2.455  0.01412 *
## L(y, 1)     -3.080e-03  1.190e-03  -2.587  0.28617
## L(d(y), 1)   1.294e-03  1.547e-02   0.084  0.93336
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.02373 on 4189 degrees of freedom
## Multiple R-squared:  0.001598,   Adjusted R-squared:  0.0008826
## F-statistic:    NA on NA and NA DF,  p-value: NA
```

So according to our DF test we have a random walk with no drift. Im going to assume that Expedia is going to be very similar.

```
vail_df <- ur.df(VailResorts,type="trend", lags=1)
summary(vail_df)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.195792 -0.010275  0.000425  0.010609  0.174189
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.696e-03  3.678e-03   2.636  0.00842 **
## z.lag.1     -3.080e-03  1.190e-03  -2.587  0.00970 **
## tt           2.119e-06  8.632e-07   2.455  0.01412 *
## z.diff.lag   1.294e-03  1.547e-02   0.084  0.93336
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02373 on 4189 degrees of freedom
## Multiple R-squared:  0.001598,   Adjusted R-squared:  0.0008826
## F-statistic: 2.234 on 3 and 4189 DF,  p-value: 0.08215
##
##
## Value of test-statistic is: -2.5874 2.6895 3.3515
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.96 -3.41 -3.12
## phi2  6.09  4.68  4.03
## phi3  8.27  6.25  5.34
```

```
vail_df <- ur.df(VailResorts,type="drift", lags=1)
summary(vail_df)
```

```
##
```

```
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.196737 -0.010366  0.000297  0.010669  0.174936
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.956e-03  1.896e-03   1.031    0.302
## z.lag.1     -3.430e-04  4.178e-04  -0.821    0.412
## z.diff.lag   6.082e-05  1.547e-02   0.004    0.997
##
## Residual standard error: 0.02375 on 4190 degrees of freedom
## Multiple R-squared:  0.0001608,  Adjusted R-squared:  -0.0003164
## F-statistic: 0.337 on 2 and 4190 DF,  p-value: 0.7139
##
##
## Value of test-statistic is: -0.8209 1.0188
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

According to the dicky fuller test there is a RW with no drift. However, that just looks crazy and after looking at the regression with just an intercept there is a statistcally significant value greater than 0. It is probably just that the RW is so powerful it is overpowering the drift.

```
summary(lm(VailResorts ~ 1))
```

```
##
## Call:
## lm(formula = VailResorts ~ 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.01768 -0.82294  0.00912  0.88279  1.40366
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45342    0.01356   328.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.878 on 4194 degrees of freedom
```

Expedia:

```r
plot(Expedia)
```


```r
Expedia_cadf <- CADFtest(Expedia, criterion = c("BIC"), type = "trend")
summary(Expedia_cadf)
```

```
## Augmented DF test
##                                              ADF test
## t-test statistic:                          -2.4028571
## p-value:                                    0.3779248
## Max lag of the diff. dependent variable:    0.0000000
##
## Call:
## dynlm(formula = formula(model), start = obs.1, end = obs.T)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40967 -0.01199  0.00052  0.01232  0.24783
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.033e-02  4.132e-03   2.501   0.0124 *
## trnd         1.324e-06  7.168e-07   1.847   0.0649 .
## L(y, 1)     -3.042e-03  1.266e-03  -2.403   0.3779
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02877 on 4190 degrees of freedom
## Multiple R-squared:  0.001419,   Adjusted R-squared:  0.0009423
## F-statistic:    NA on NA and NA DF,  p-value: NA
```

gotta go to case 2 Same results as vail resorts. As well these stocks both follow really similar patterns.

```r
expedia_df <- ur.df(Expedia,type="drift", lags=1)
summary(expedia_df)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40996 -0.01185  0.00028  0.01210  0.24874
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0046839  0.0027651   1.694   0.0904 .
## z.lag.1     -0.0010395  0.0006487  -1.602   0.1092
## z.diff.lag   0.0074763  0.0154452   0.484   0.6284
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02878 on 4190 degrees of freedom
## Multiple R-squared:  0.0006621,  Adjusted R-squared:  0.0001851
## F-statistic: 1.388 on 2 and 4190 DF,  p-value: 0.2497
##
##
## Value of test-statistic is: -1.6023 1.5282
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

Expedia is has basically the same the same result as vail resorts. They most likely share the same unit root. I am also going to treat this as a random walk with drift.

```r
summary(lm(Expedia ~ 1))
```

```
##
```

```
## Call:
## lm(formula = Expedia ~ 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1314 -0.6017  0.2285  0.5581  1.1580
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.20705    0.01058   397.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6853 on 4194 degrees of freedom
```

Run the cOil Test gonna find a unit root.

didn't log it this time going to test it before we log it. Taking care of the na value after logging. Just replacing it with 0.

```r
anyNA(COil)
```

```
## [1] FALSE
```

```r
COil[is.na(COil)] <- 0
anyNA(COil)
```

```
## [1] FALSE
```

```r
plot(COil)
```

looking at it it doesn't really have a trend so we are just going to do case 2. I logged it initially first and it sort of broke everything. This makes sense as there isn't really a need to log it variable to the data. We accept the null hypothesis with 1 lag so we then go to the second DF test.

```
COil_cadf <- CADFtest(COil, criterion = c("BIC"), type = "drift")
summary(COil_cadf)
```

```
## Augmented DF test
##                                                ADF test
## t-test statistic:                            -2.4842630
## p-value:                                      0.1193794
## Max lag of the diff. dependent variable:      1.0000000
##
## Call:
## dynlm(formula = formula(model), start = obs.1, end = obs.T)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.700  -0.847   0.052   0.911  37.609
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.256735   0.104671   2.453   0.0142 *
## L(y, 1)     -0.003403   0.001370  -2.484   0.1194
## L(d(y), 1)  -0.142859   0.015299  -9.338   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.039 on 4185 degrees of freedom
## Multiple R-squared:  0.02234,    Adjusted R-squared:  0.02188
## F-statistic:    NA on NA and NA DF,  p-value: NA
```

we reject both tau2 and phi1 so we have a RW with no drift which is what we would expect.

```
COil_df <- ur.df(COil, type="drift", lags=1)
summary(COil_df)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.700  -0.847   0.052   0.911  37.609
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.256735   0.104671   2.453   0.0142 *
## z.lag.1     -0.003403   0.001370  -2.484   0.0130 *
## z.diff.lag  -0.142859   0.015299  -9.338   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.039 on 4185 degrees of freedom
## Multiple R-squared:  0.02234,    Adjusted R-squared:  0.02188
## F-statistic: 47.82 on 2 and 4185 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -2.4843 3.1245
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

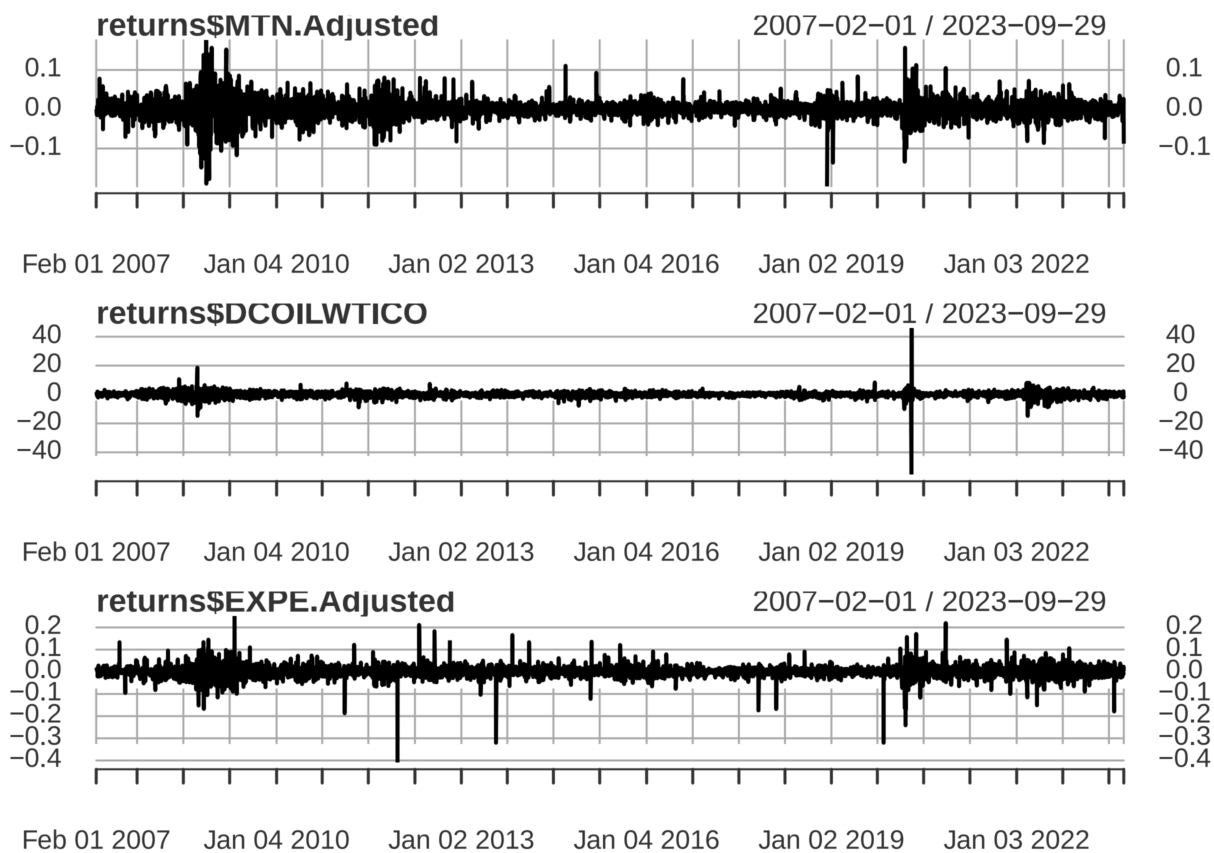
Most of my Data Points contain a unit root so we are just going to work with daily returns.

if we want to say how to variables that share a unit root long term interact we want to do a cointergration. If we difference a log then we get the return. Which a change in the log is a percent change. So that way whe get the returns.

this gives us the returns

```r
#Note COil is missing a value so maybe worth going back and changing how I log it.
data <- merge.xts(VailResorts, COil, Expedia)
returns <- diff(data)
```

```r
par(mfrow = c(3, 1))

plot(returns$MTN.Adjusted)
plot(returns$DCOILWTICO)
plot(returns$EXPE.Adjusted)
```



auto.arima with seasonality = TRUE still returns no season elements. This is what we would expect as people are hedging there stocks based on this so the seasonality should be no longer reflected in the stock price. We do get a c(2, 0, 0) so two auto regressive terms.

```r
Vail1 <- auto.arima(returns$MTN.Adjusted, seasonal=TRUE)
summary(Vail1)
```

```
## Series: returns$MTN.Adjusted
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##          ar1      ar2
##       0.0010  -0.0407
```

```
## s.e.  0.0156   0.0156
##
## sigma^2 = 0.0005633:  log likelihood = 9734.45
## AIC=-19462.9   AICc=-19462.89   BIC=-19443.87
##
## Training set error measures:
##                        ME       RMSE       MAE MPE MAPE      MASE          ACF1
## Training set 0.0004443783 0.02372825 0.0156305 NaN  Inf 0.6982738 -0.0002092407
```

#Note came back later and edited it to make knit work it was just easier to reload the data and do everything than figure out why it was broken.

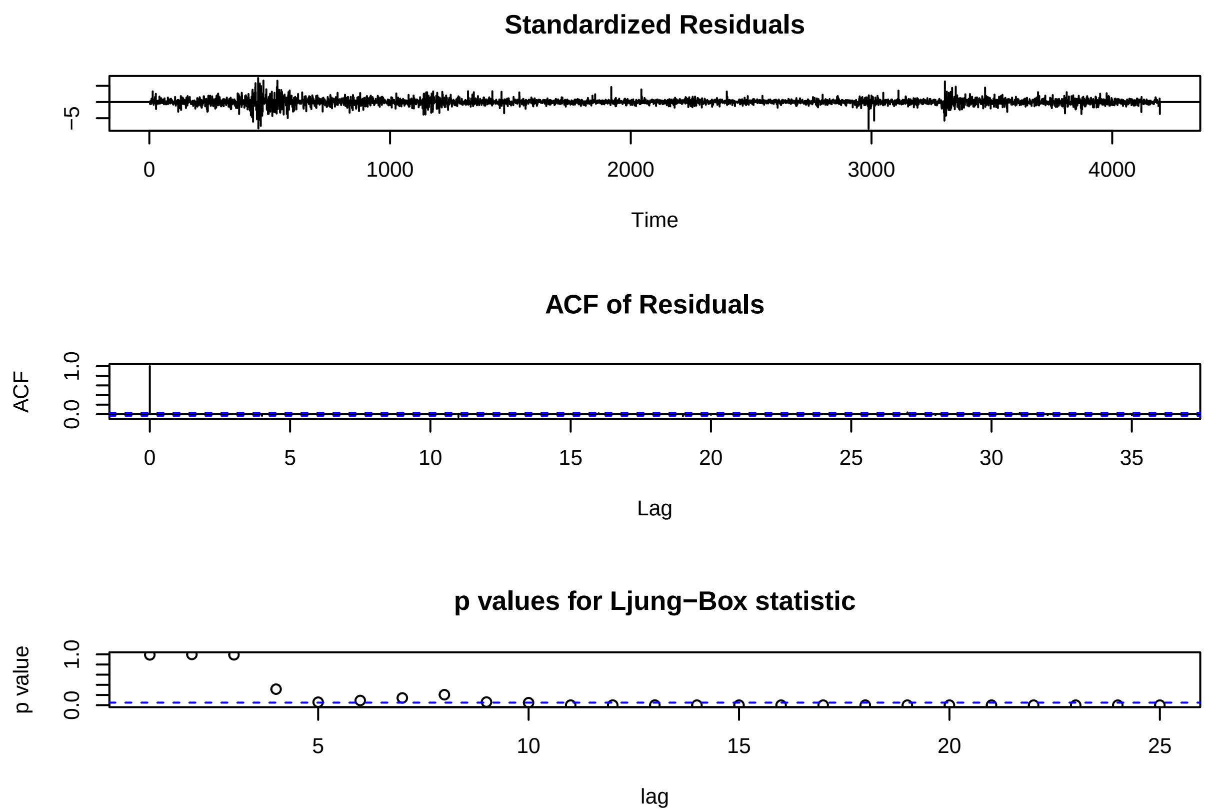
From what we can see in fitting our fourier terms none of them are statistically significant so we can conclude with pretty solid confidence that there is no season term in our Vail Resorts data. This makes sense as with how people know Vail Resorts brings in a large amount of its income during the winter people hedge on that when buying the stock. So as a result this sort of strips the stock itself of its seasonality.

```
startdate <- "2007-02-01"
enddate <- "2023-10-01"

MTN <- window(MTN, start = startdate, end = enddate)
temp <-  ts(na.omit(diff(log(MTN$MTN.Adjusted))),freq=252, start = 2007)

auto.arima(temp, xreg=fourier(temp,K=c(4)), seasonal=FALSE)
```

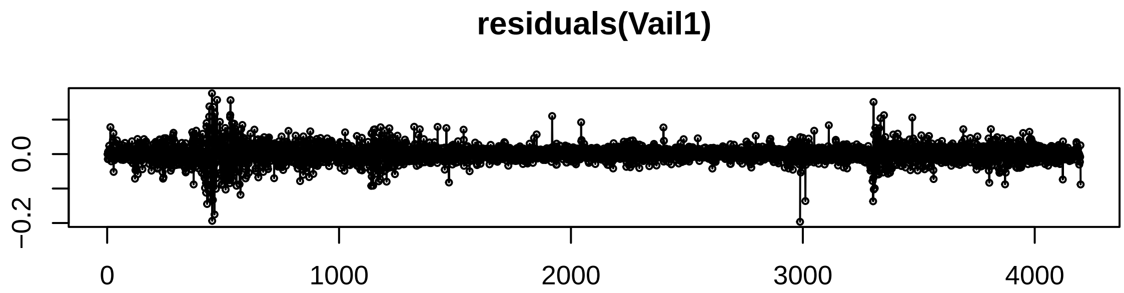
```
## Series: temp
## Regression with ARIMA(3,0,3) errors
##
## Coefficients:
##           ar1      ar2     ar3     ma1     ma2      ma3  S1-252  C1-252  S2-252
##       -0.0804  -0.0407  0.8746  0.0744  0.0006  -0.8799   4e-04  -4e-04   5e-04
## s.e.   0.0605   0.0560  0.0552  0.0564  0.0514   0.0500   4e-04   4e-04   4e-04
##       C2-252  S3-252  C3-252  S4-252  C4-252
##       -6e-04   0e+00  -7e-04   2e-04   1e-04
## s.e.   4e-04   5e-04   5e-04   5e-04   5e-04
##
## sigma^2 = 0.0005602:  log likelihood = 9756.61
## AIC=-19483.21   AICc=-19483.1   BIC=-19388.09
```

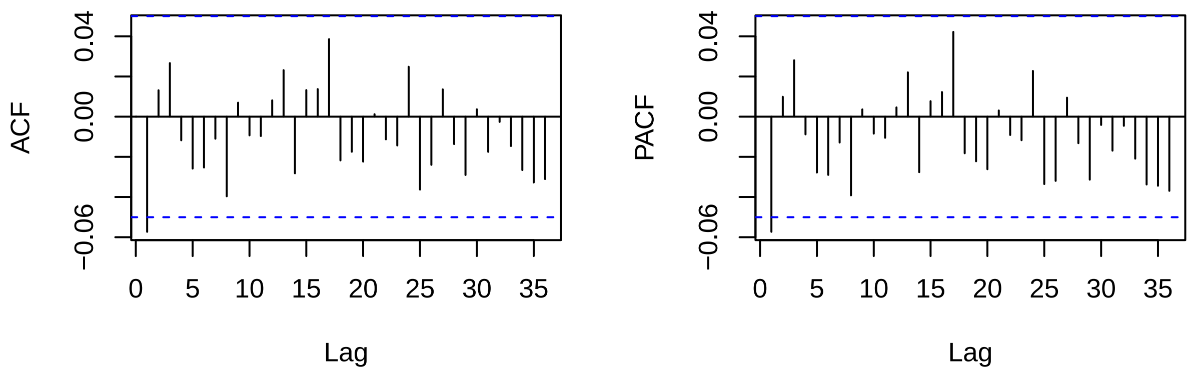
Theres auto correlation errors with the ARMA(2,0) Especially with the LJung have multiple significant values. Going to have to check over a larger space.

```
tsdiag(Vail1,gof=25)
```




```
tsdisplay(residuals(Vail1))
```

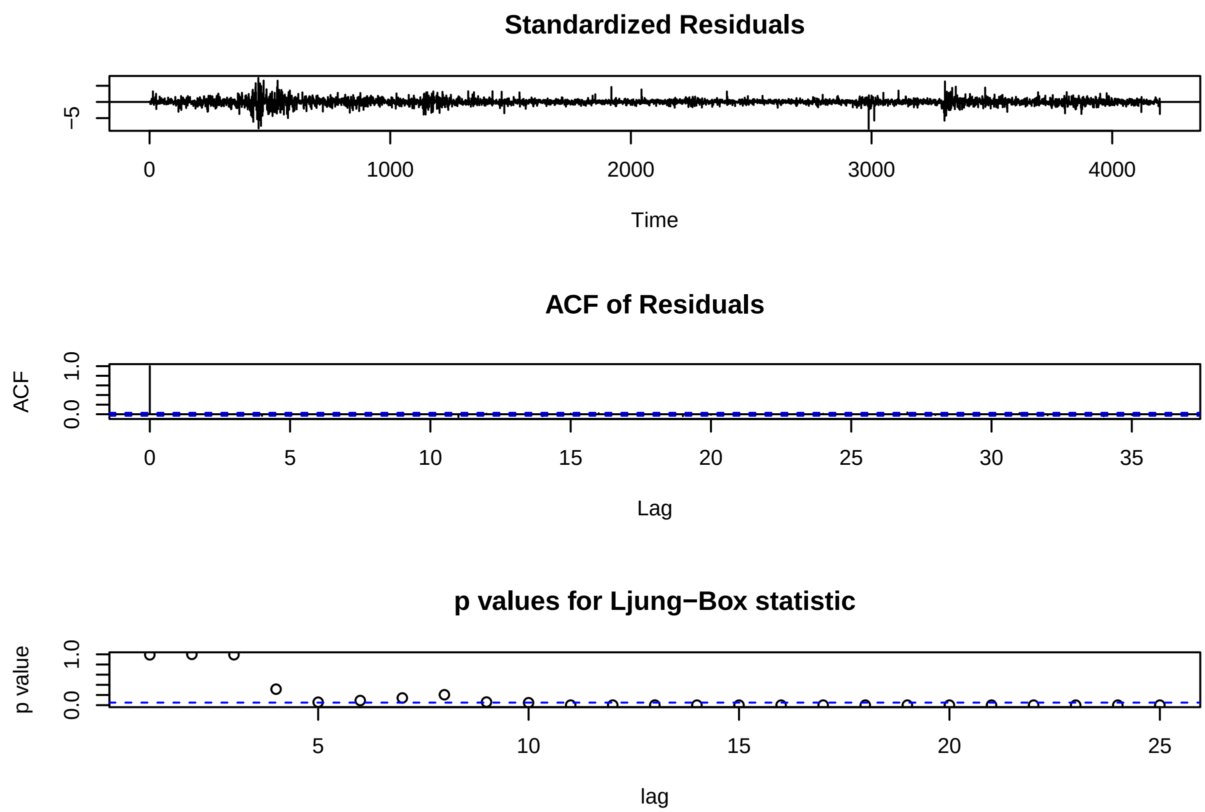




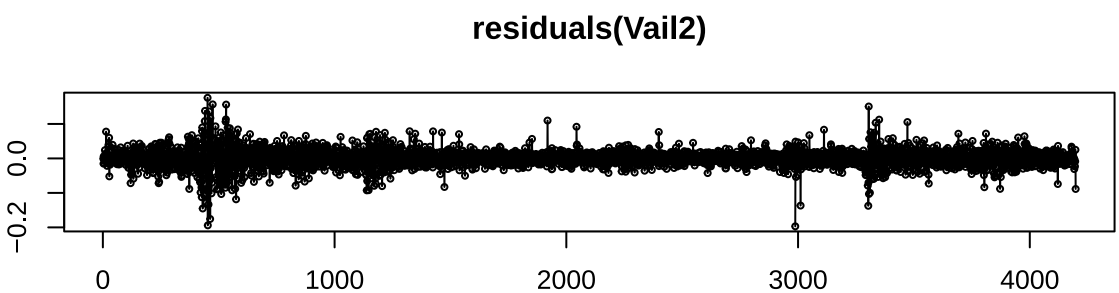

```r
Vail2 <- Arima(returns$MTN.Adjusted, order = c(2,0,0), include.constant = T)
summary(Vail2)
```

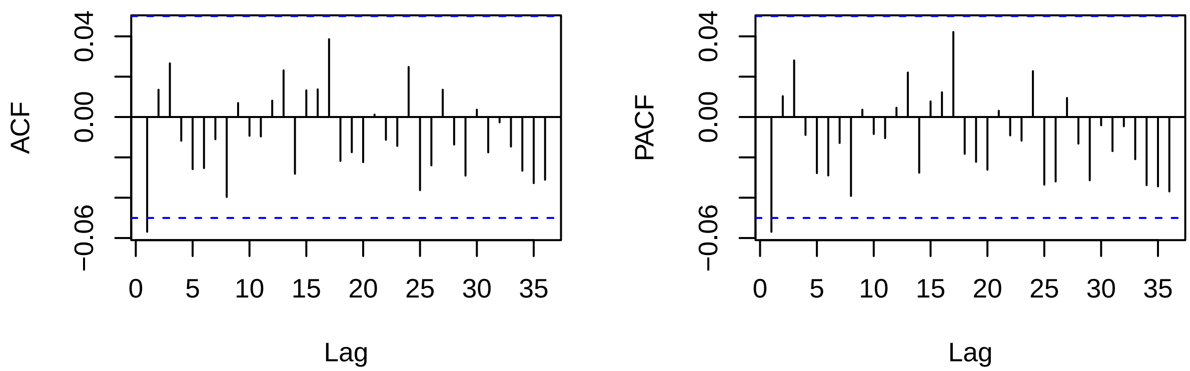
```
## Series: returns$MTN.Adjusted
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2   mean
##       0.0006  -0.0411  4e-04
## s.e.  0.0156   0.0156  4e-04
##
## sigma^2 = 0.0005632:  log likelihood = 9735.19
## AIC=-19462.38   AICc=-19462.37   BIC=-19437.01
##
## Training set error measures:
##                        ME       RMSE        MAE MPE MAPE      MASE         ACF1
## Training set 8.901913e-07 0.02372405 0.01561959 NaN  Inf 0.6971041 0.0002002092
```

```r
tsdiag(Vail2,gof=25)
```




```
tsdisplay(residuals(Vail2))
```





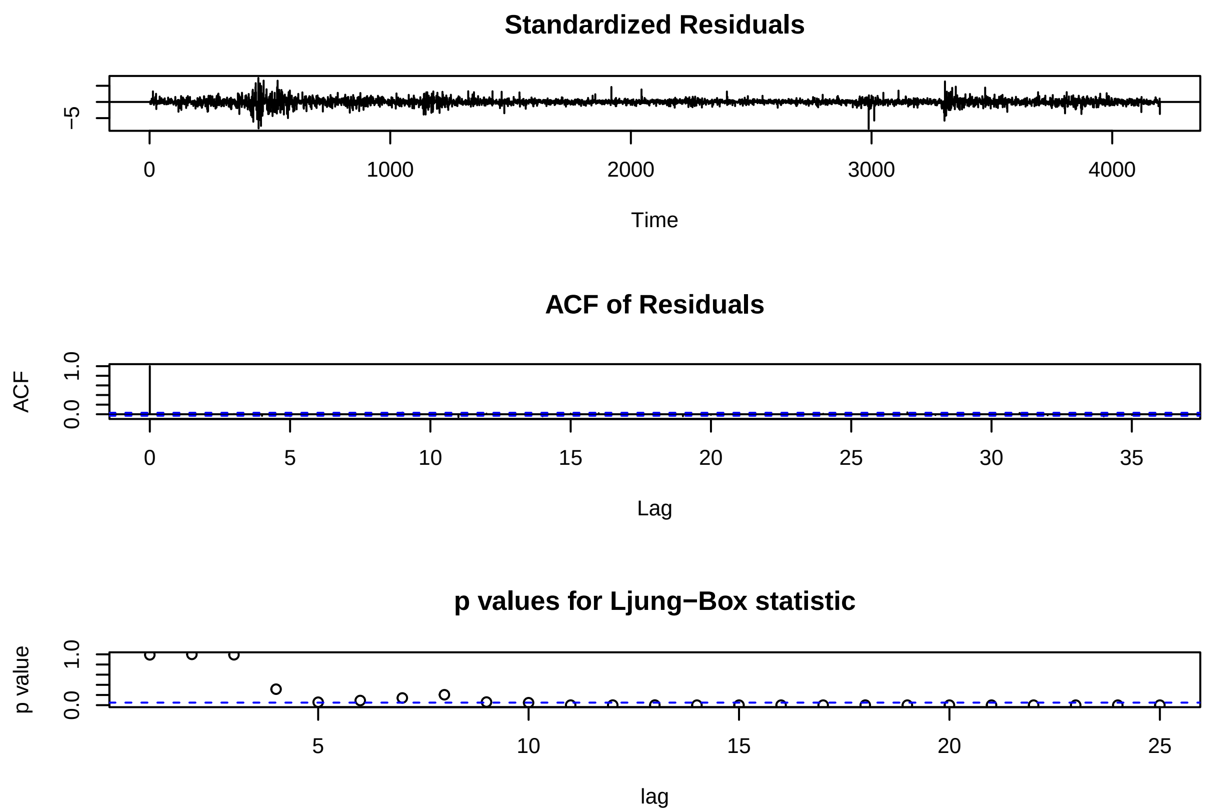
just gonna check a large potential size with there being auto cor still in the model. arima(13,0,5) is the final model. Bigger than I like but lets check the residuals.

```
auto.arima(returns$MTN.Adjusted,max.p=15,max.order=100,stepwise=F,trace=T,approximation=F)
```

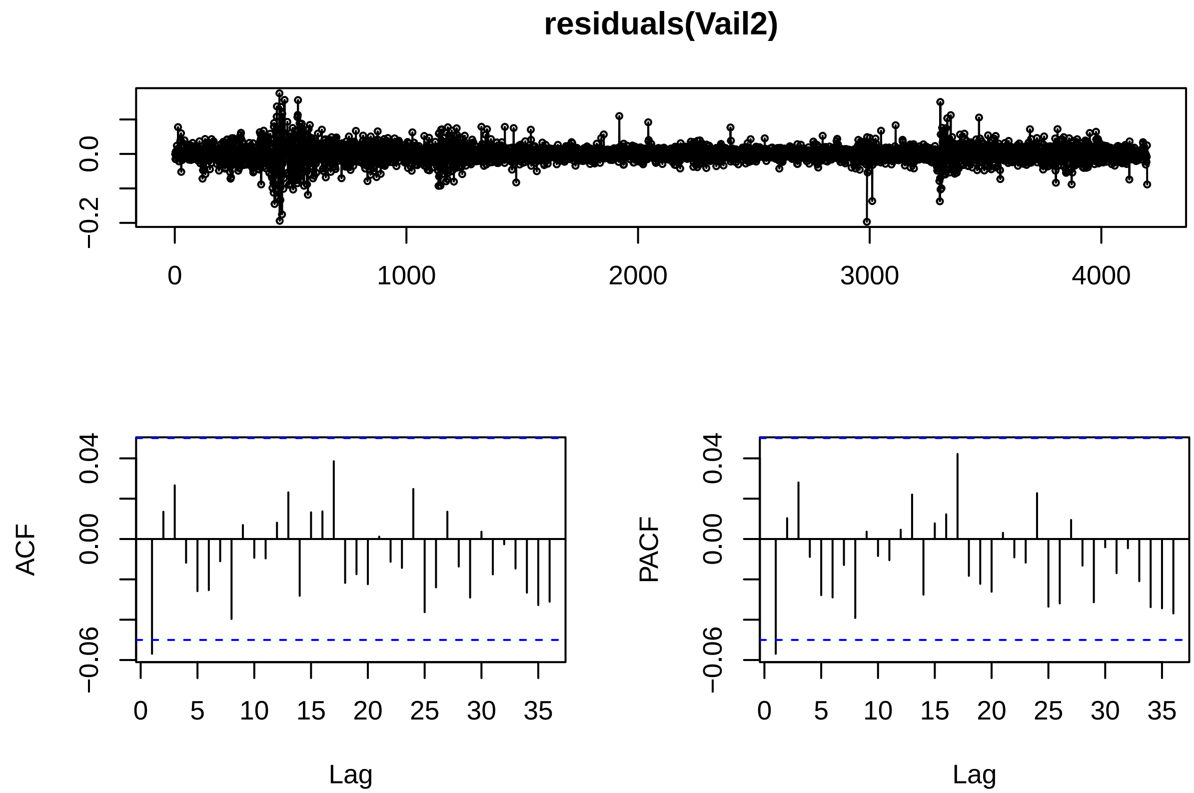
large model test from Auto.ARIMA basically has the same auto cor results so were just going to go back to the initial arima(2,0,0).

```
vail2 <- Arima(returns$MTN.Adjusted, order = c(13,0,5), include.constant = T)
```

```
tsdiag(Vail2,gof=25)
```




```r
tsdisplay(residuals(Vail2))
```



I'm just going to have to go with the base arima(2,0,0) and potential in the future get some more explanatory variables. Most of the auto is gone there is still some though. Some of these plots are concerning however the Unit root is really strong as well as theres probably issues with the structural breaks.

Expedia

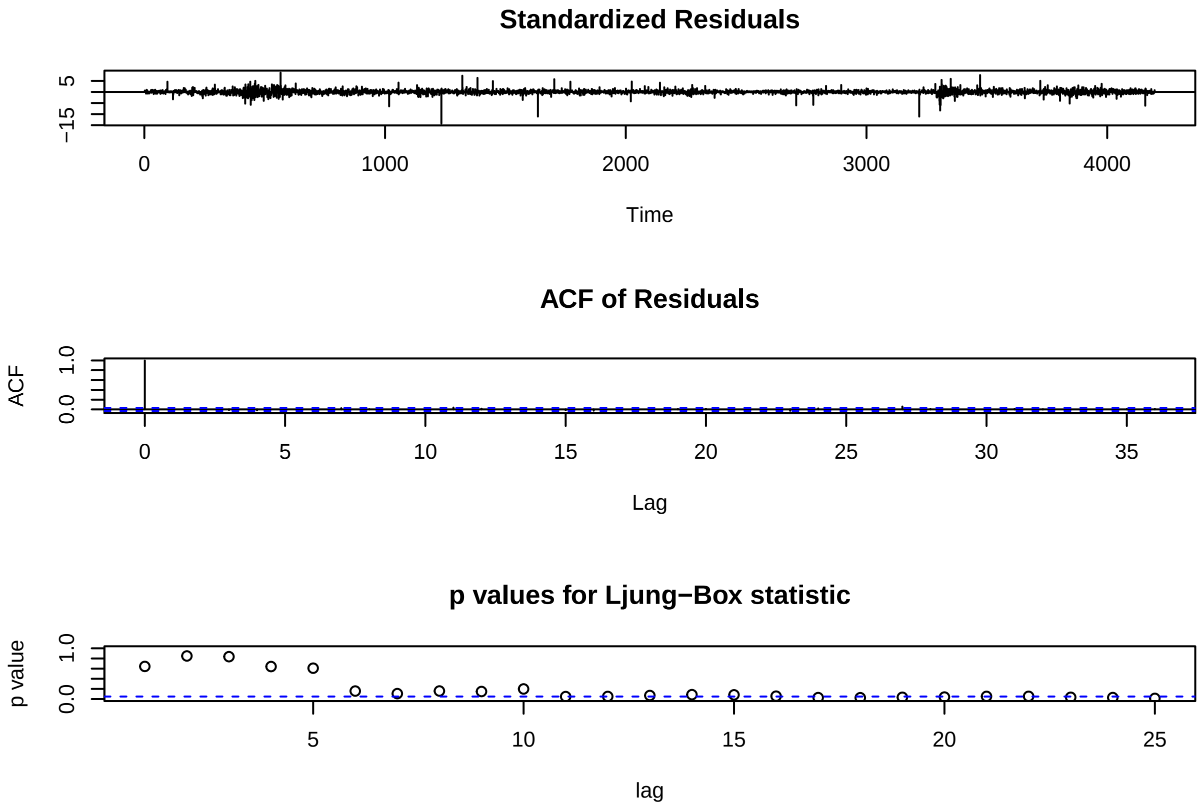
likes the arima(0,0,0)

```r
exM1 <- auto.arima(returns$EXPE.Adjusted)
summary(exM1)
```

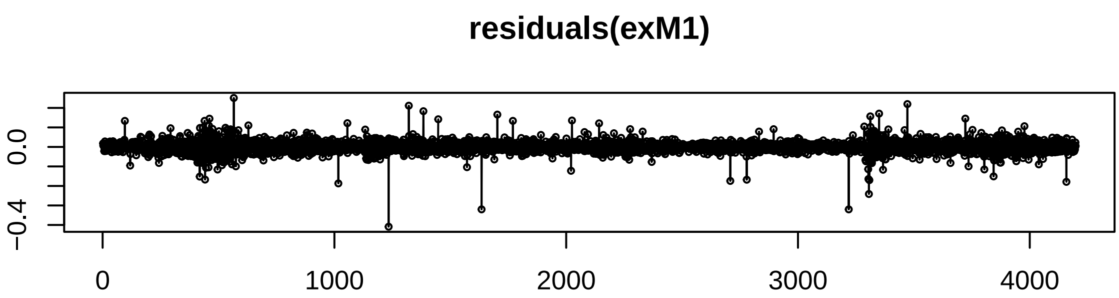
```
## Series: returns$EXPE.Adjusted
## ARIMA(0,0,0) with zero mean
##
## sigma^2 = 0.0008285:  log likelihood = 8924.85
## AIC=-17847.71   AICc=-17847.71   BIC=-17841.37
##
## Training set error measures:
##                        ME       RMSE        MAE MPE MAPE      MASE         ACF1
## Training set 0.0003133435 0.02878333 0.01813365 NaN  Inf 0.6926082 0.007163625
```

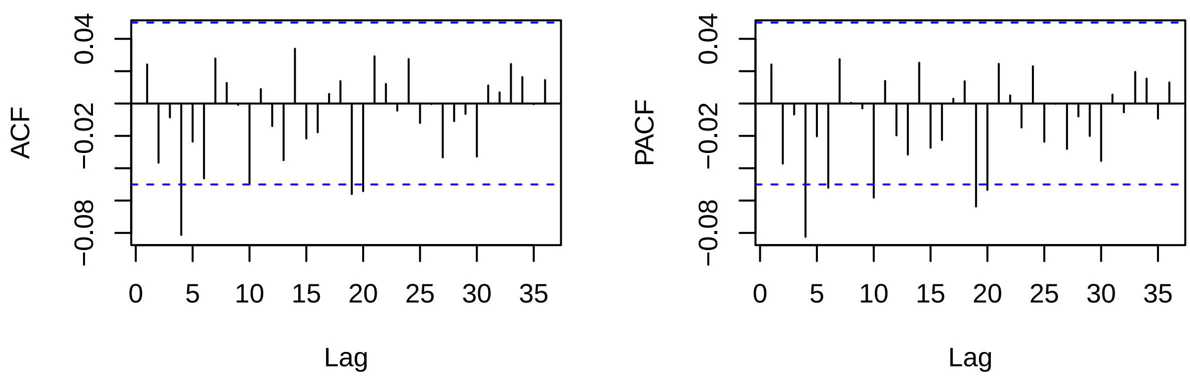
This ones got a bit more auto core were gonna look for a little more complex model.

```
tsdiag(exM1,gof=25)
```




```
tsdisplay(residuals(exM1))
```





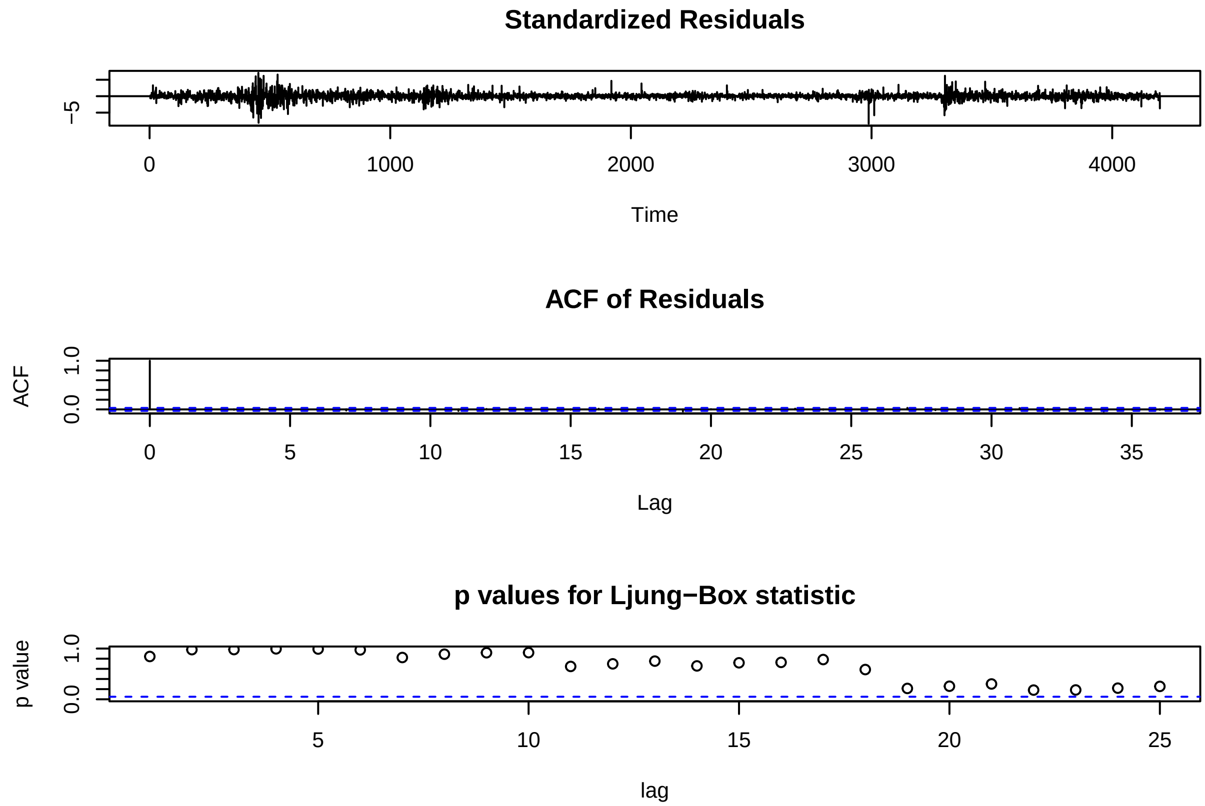
final model ARIMA(6,0,5) little more complex. Didn't let is search as far. See if there is issues with auto cor.

```
auto.arima(returns$MTN.Adjusted,max.p=7,max.order=100,stepwise=F,trace=T,approximation=F)
```

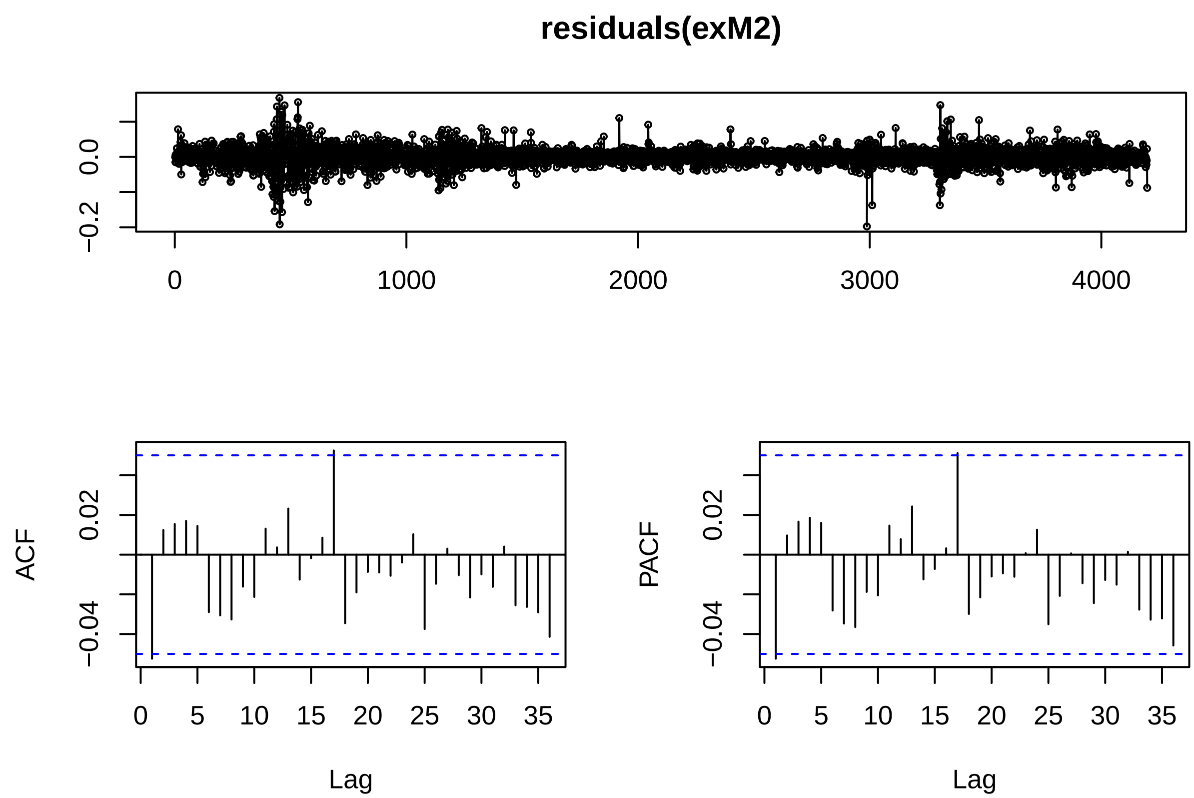
```
exM2 <- Arima(returns$MTN.Adjusted, order = c(6,0,5), include.constant = T)
```

The Residuals are much better here. A little sus with the third lag but looks pretty good.

```
tsdiag(exM2,gof=25)
```




```
tsdisplay(residuals(exM2))
```



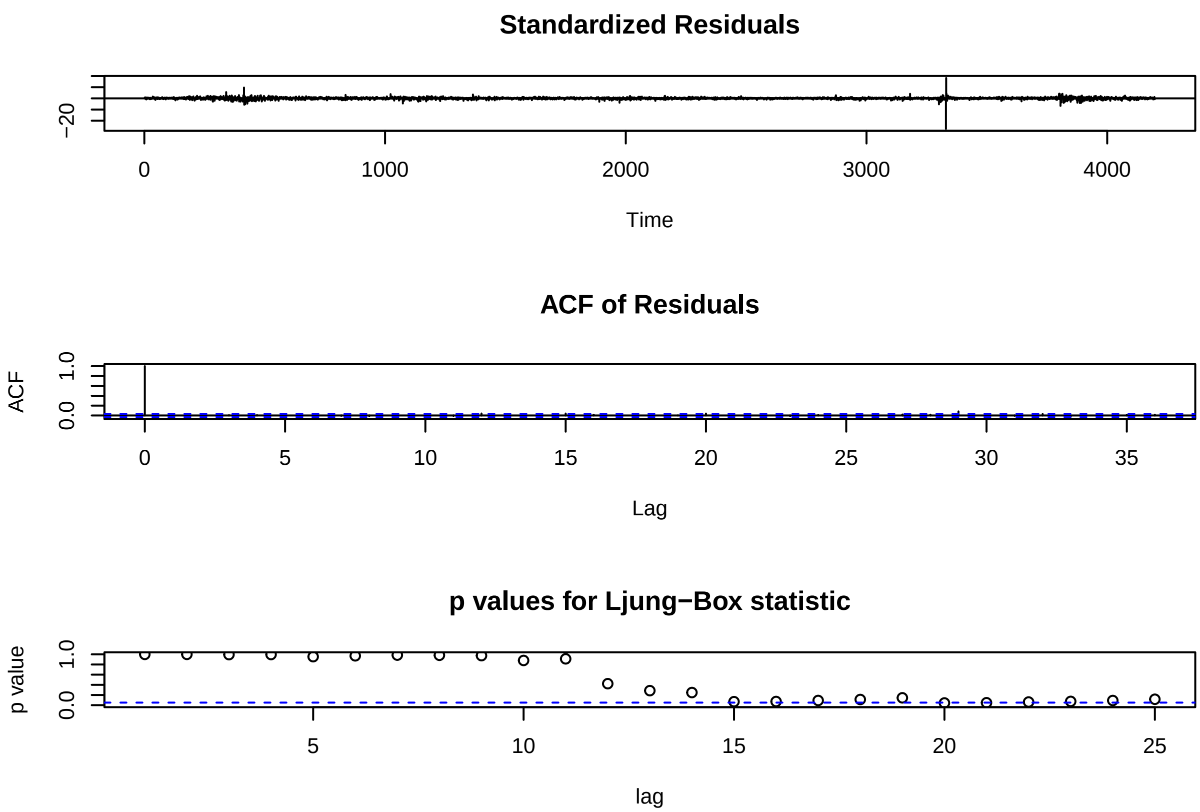
gonna roll with this model for Expedia

COil

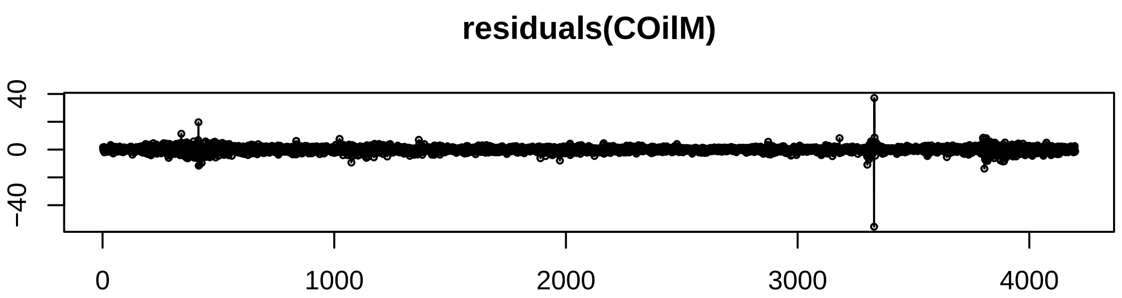
pretty long equation for base auto.arima there is definitely auto core issues however I feel it mostly stems from the 1 value that goes negative so I feel as though it stems from a structural break from covid. So I am just going to test all of my answers for structural breaks by doing question 2. These Standard errors seem as though they are seasonal.

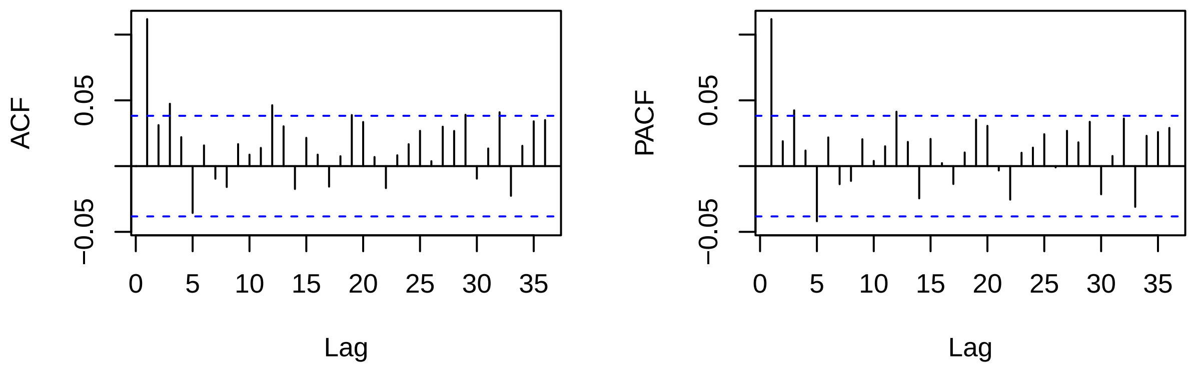
```
COilM <- auto.arima(returns$DCOILWTICO)
```

```
tsdiag(COilM,gof=25)
```




```
tsdisplay(residuals(COilM))
```





Question 2

My data and everything has become so messed up I am just going to reload everything and resubset it knowing what we know now.

```
rm(list = ls())
```

There was way to much going on in my other notebook so I'm just going to do question 2 from a clean slate.

```
#load in vail resorts
getSymbols("MTN")
```

```
## [1] "MTN"
```

```
#National Average weekly gass pri
#getSymbols("GASREGW", src = "FRED")
getSymbols("DCOILWTICO", src = "FRED")
```

```
## [1] "DCOILWTICO"
```

```
#adding Expedia as it is a travel company, going to be used to model travel
getSymbols("EXPE")
```

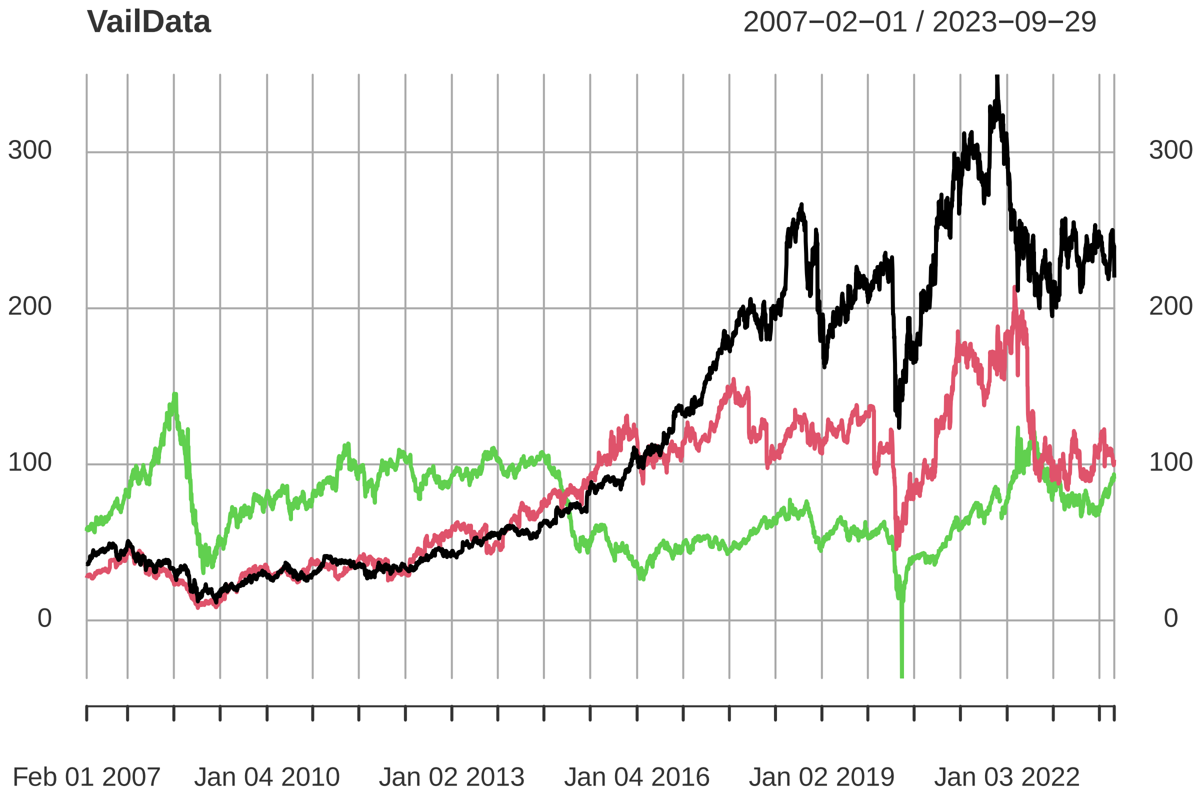
```
## [1] "EXPE"
```

```r
startdate <- "2007-02-01"
enddate <- "2023-10-01"

MTN <- window(MTN, start = startdate, end = enddate)
DCOILWTICO <- window(DCOILWTICO, start = startdate, end = enddate)
EXPE <- window(EXPE, start = startdate, end = enddate)
```

```r
mtn <- merge.xts(MTN$MTN.Adjusted,EXPE$EXPE.Adjusted, DCOILWTICO,join="inner")
```

```
## Warning in merge.xts(MTN$MTN.Adjusted, EXPE$EXPE.Adjusted, DCOILWTICO, join =
## "inner"): 'join' only applicable to two object merges
```

```r
VailData <- mtn
plot(VailData)
```




```r
vailData <- na.omit(VailData)
```

```r
temp <- ts(na.omit(diff(log(VailData))),freq=252, start = 2007)#so they stay the same size remove colum
```

```
## Warning in log(VailData): NaNs produced
```

```
vResorts = temp[,1]
Expedia = temp[,2]
COil = temp[,3]
```

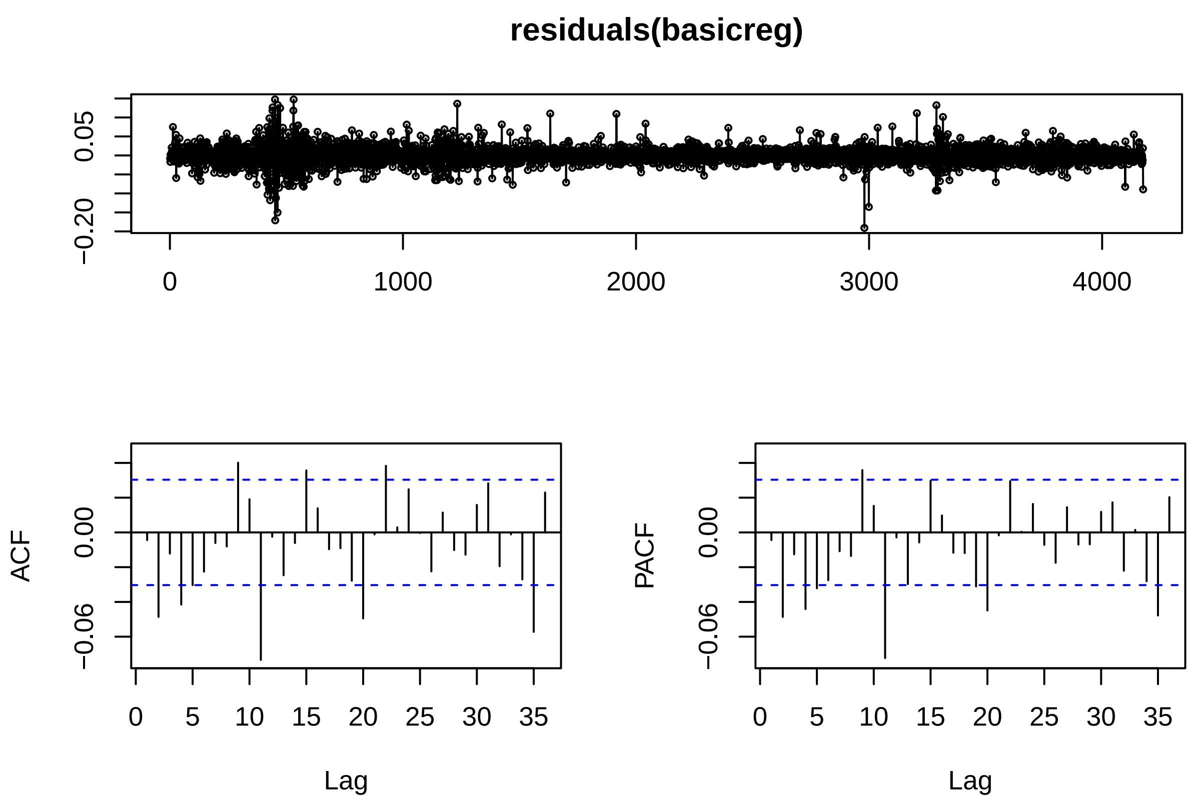
different lengths probably because of the NA values from logging COil. Going to need to go in and make some changes especially to variable length.

both predictors are significant

```
basicreg <- lm(temp[,1] ~ temp[,2] + temp[,3])
#summary(basicreg)
```

There is definite auto correlation within the residuals you might be able to say that it is at season values but I wouldn't be completely confident on that.

```
tsdisplay(residuals(basicreg))
```



Because there is no lagged dependent variables were just gonna use HAC standard errors.

Both expedia and Crude Oil are still significant even after using HAC standard errors.

```
coeftest(basicreg,vcov=vcovHAC(basicreg))
```

```
##
## t test of coefficients:
```

```
##
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.00031462 0.00032828  0.9584  0.33792
## temp[, 2]   0.34925626 0.03173382 11.0058  < 2e-16 ***
## temp[, 3]   0.04033196 0.01939329  2.0797  0.03762 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question:

I will be looking for a break in the regression coefficients of vail Resorts. I am first going to look for breaks then fit a dynamic model with breaks if they exist.

Bai and Perron test:

```
bp_vResorts = breakpoints(vResorts ~ Expedia + COil)
breakpoints(bp_vResorts)
```

```
##
##   Optimal 3-segment partition:
##
## Call:
## breakpoints.breakpointsfull(obj = bp_vResorts)
##
## Breakpoints at observation number:
## 1232 3274
##
## Corresponding to breakdates:
## 2011(224) 2019(250)
```

```
summary(bp_vResorts)
```

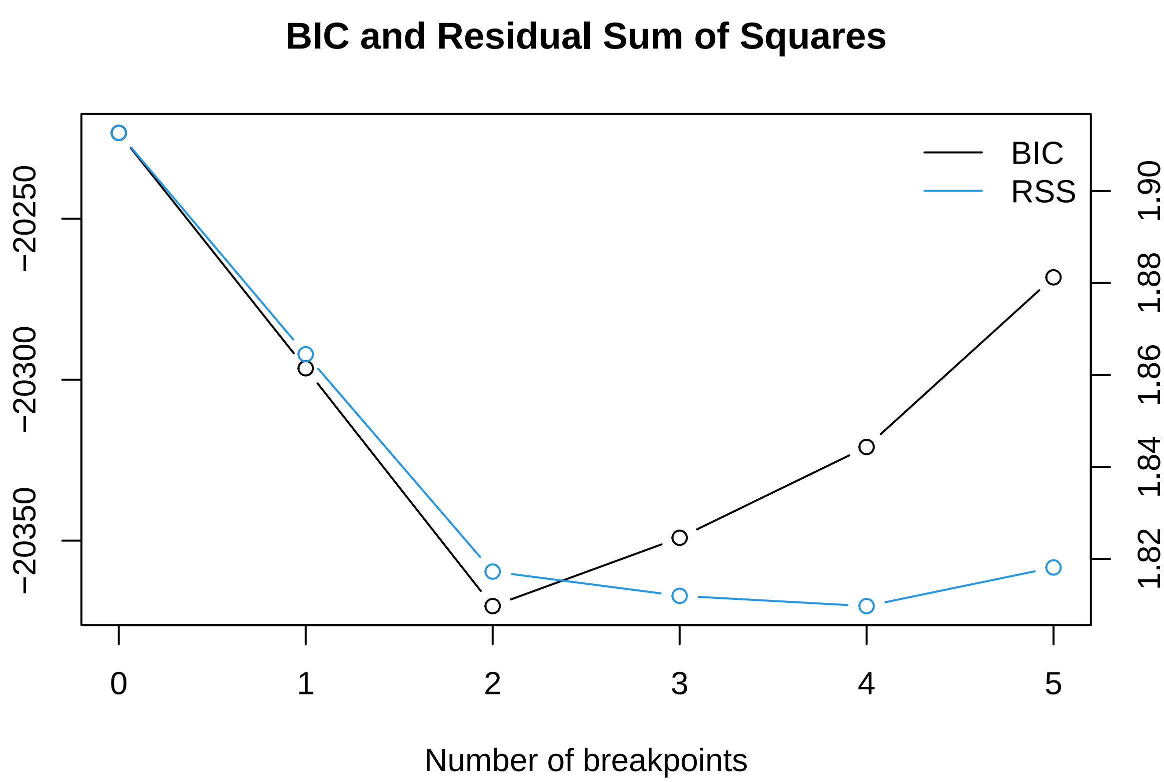
```
##
##   Optimal (m+1)-segment partition:
##
## Call:
## breakpoints.formula(formula = vResorts ~ Expedia + COil)
##
## Breakpoints at observation number:
##
## m = 1       1232
## m = 2       1232           3274
## m = 3       1232 1916      3274
## m = 4       1232 1916 2638 3274
## m = 5   754 1380 2022 2648 3274
##
## Corresponding to breakdates:
##
## m = 1             2011(224)
## m = 2             2011(224)                     2019(250)
## m = 3             2011(224) 2014(152)           2019(250)
## m = 4             2011(224) 2014(152) 2017(118) 2019(250)
```

```
## m = 5   2009(250) 2012(120) 2015(6)   2017(128) 2019(250)
##
## Fit:
##
## m   0          1          2          3          4          5
## RSS      1.913      1.865      1.817      1.812      1.810      1.818
## BIC -20223.361 -20296.464 -20370.326 -20349.120 -20320.907 -20268.215
```

```r
bp_vResorts
```

```
##
##   Optimal 3-segment partition:
##
## Call:
## breakpoints.formula(formula = vResorts ~ Expedia + COil)
##
## Breakpoints at observation number:
## 1232 3274
##
## Corresponding to breakdates:
## 2011(224) 2019(250)
```

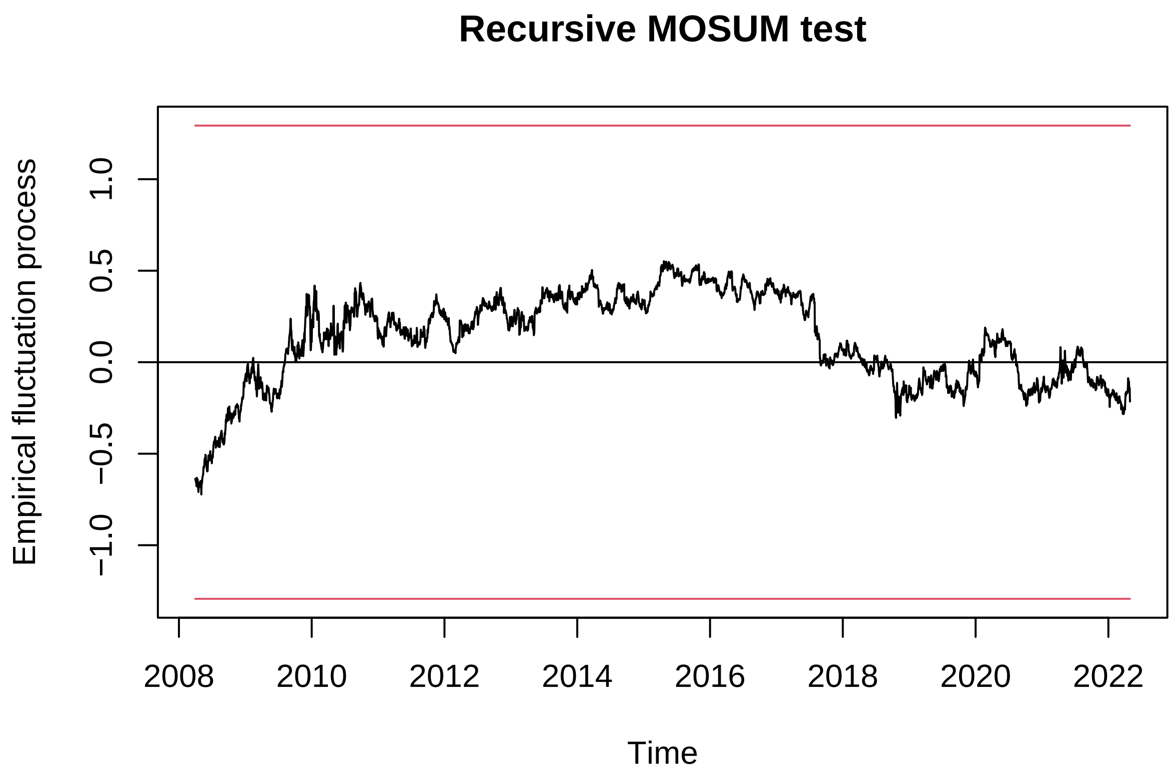
```r
plot(bp_vResorts)
```



The Bai and Perron test detects 2 structural breaks.

The Recursive estimate does not detect any structural break however the residual sum test does detect a structural break:

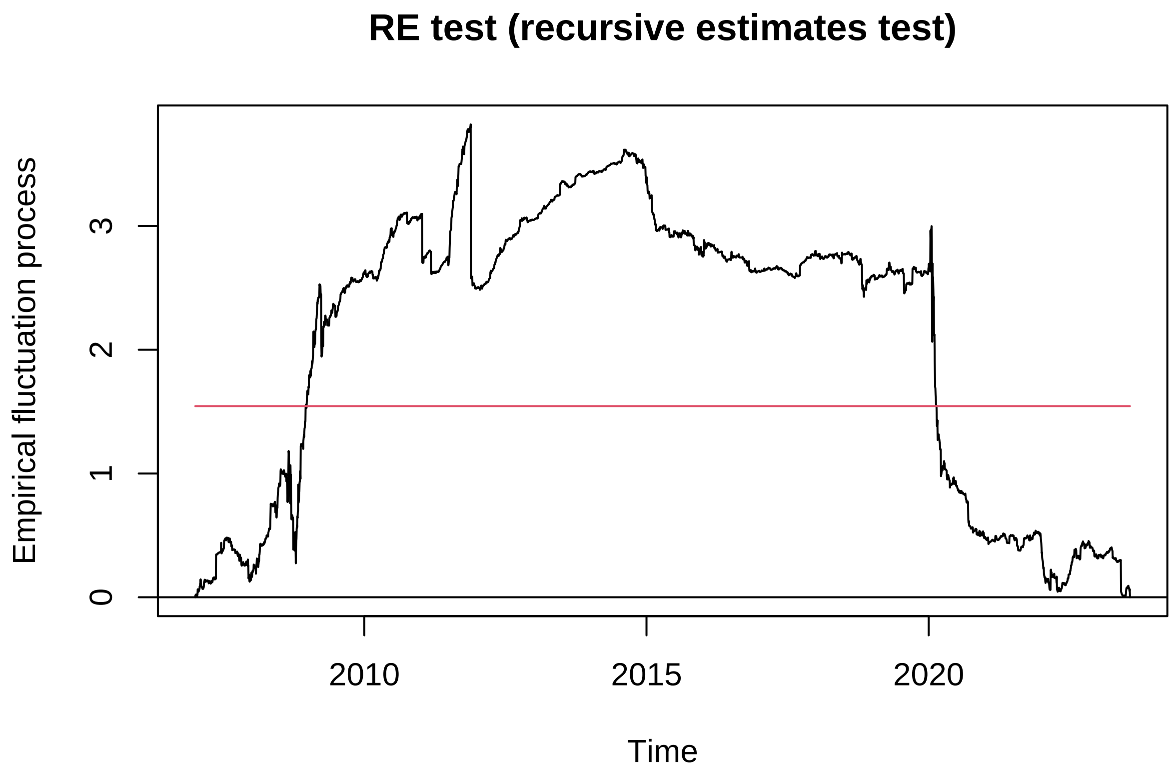
```
efptest.sum <- efp(temp[,1]~temp[,2] + temp[,3],type="Rec-MOSUM")
plot(efptest.sum)
```



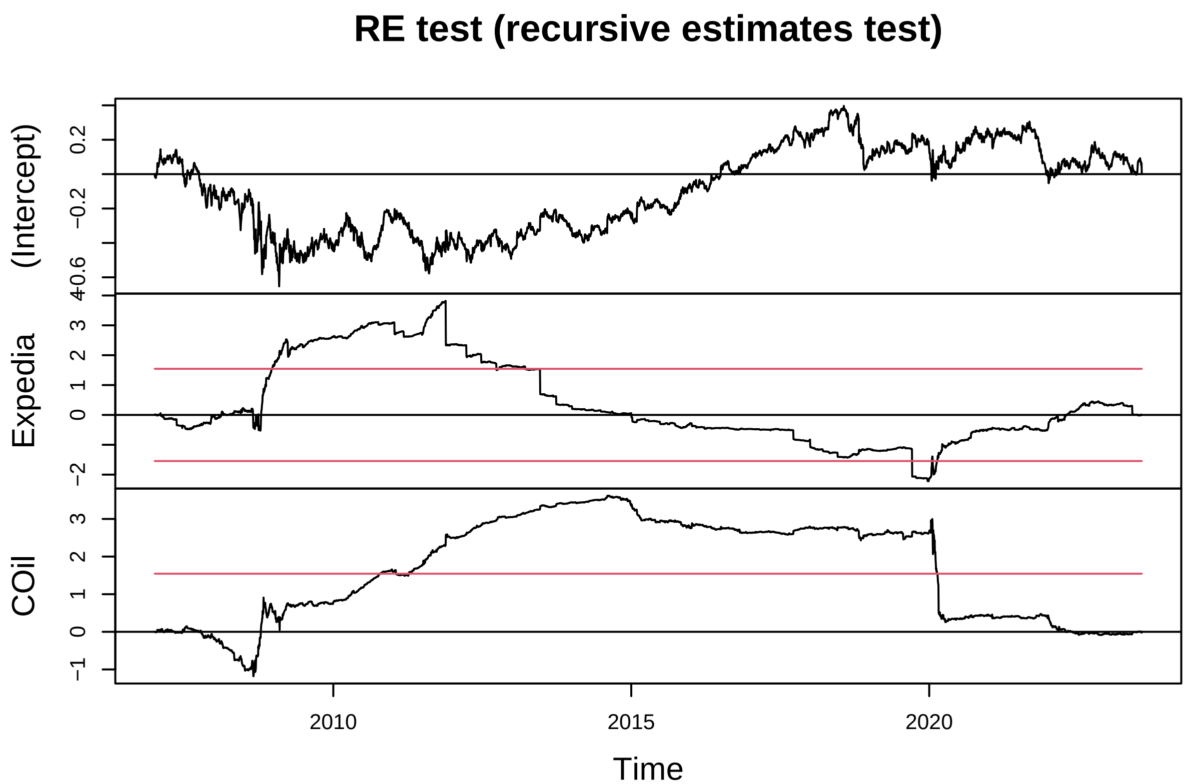

```
sctest(efptest.sum)
```

```
##
##  Recursive MOSUM test
##
## data:  efptest.sum
## M = 0.72297, p-value = 0.4721
```

```
efptest.est <- efp(vResorts~ Expedia + COil,type="RE",rescale=TRUE)
plot(efptest.est)
```




```r
plot(efptest.est,functional=NULL)
```



The RE test more closely agrees with what I assumed would occur in our data set. Where we thought that there would be structural breaks for

```
efptest.est
```

```
##
## Empirical Fluctuation Process: RE test (recursive estimates test)
##
## Call: efp(formula = vResorts ~ Expedia + COil, type = "RE", rescale = TRUE)
```

We are going to go with two structural breaks which is what the Bai and Perron test says as well as the RE Test

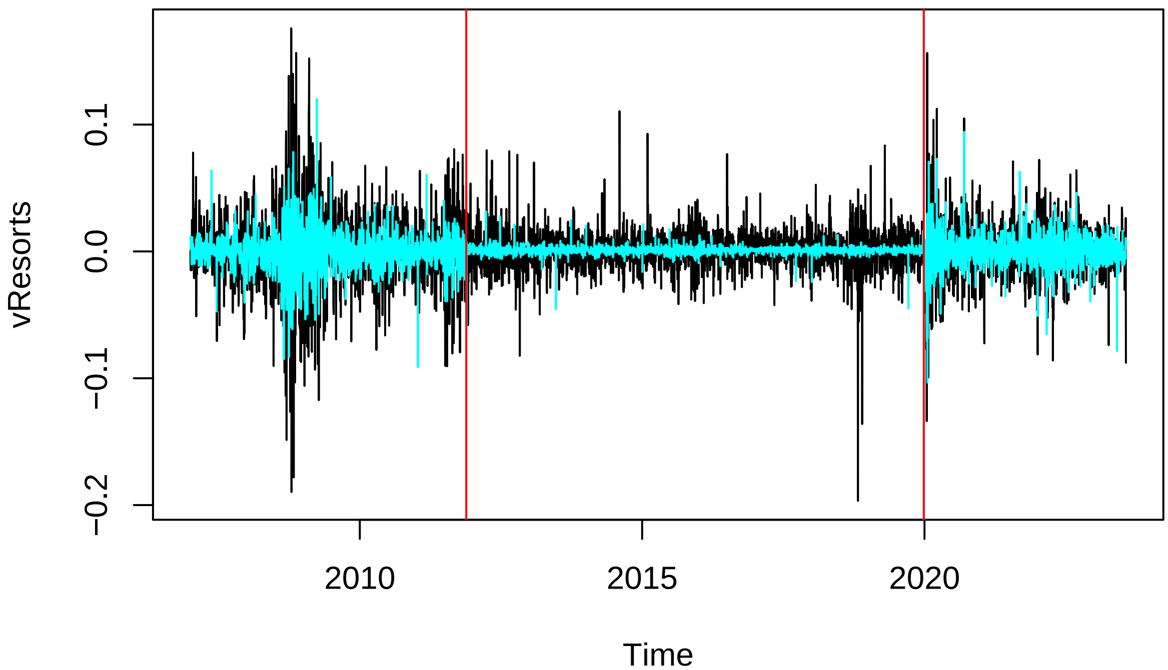
```
structural.model <- dynlm(vResorts ~ breakfactor(bp_vResorts, breaks = 2)/(Expedia + COil))
summary(structural.model)
```

```
##
## Time series regression with "ts" data:
## Start = 2007(1), End = 2023(144)
##
## Call:
## dynlm(formula = vResorts ~ breakfactor(bp_vResorts, breaks = 2)/(Expedia +
##     COil))
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max
## -0.195148 -0.009390  0.000069  0.009184  0.150666
##
## Coefficients:
##                                                   Estimate Std. Error
## (Intercept)                                     -0.0002336  0.0005950
## breakfactor(bp_vResorts, breaks = 2)segment2     0.0010775  0.0007535
## breakfactor(bp_vResorts, breaks = 2)segment3     0.0003484  0.0009155
## breakfactor(bp_vResorts, breaks = 2)segment1:Expedia  0.4784287  0.0197166
## breakfactor(bp_vResorts, breaks = 2)segment2:Expedia  0.1443975  0.0188873
## breakfactor(bp_vResorts, breaks = 2)segment3:Expedia  0.4367710  0.0209520
## breakfactor(bp_vResorts, breaks = 2)segment1:COil     0.1166553  0.0221154
## breakfactor(bp_vResorts, breaks = 2)segment2:COil     0.0236716  0.0219543
## breakfactor(bp_vResorts, breaks = 2)segment3:COil    -0.0210847  0.0167505
##                                                 t value Pr(>|t|)
## (Intercept)                                      -0.393    0.695
## breakfactor(bp_vResorts, breaks = 2)segment2      1.430    0.153
## breakfactor(bp_vResorts, breaks = 2)segment3      0.381    0.704
## breakfactor(bp_vResorts, breaks = 2)segment1:Expedia  24.265  < 2e-16 ***
## breakfactor(bp_vResorts, breaks = 2)segment2:Expedia   7.645 2.57e-14 ***
## breakfactor(bp_vResorts, breaks = 2)segment3:Expedia  20.846  < 2e-16 ***
## breakfactor(bp_vResorts, breaks = 2)segment1:COil      5.275 1.40e-07 ***
## breakfactor(bp_vResorts, breaks = 2)segment2:COil      1.078    0.281
## breakfactor(bp_vResorts, breaks = 2)segment3:COil     -1.259    0.208
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02088 on 4167 degrees of freedom
## Multiple R-squared:  0.2283, Adjusted R-squared:  0.2268
## F-statistic: 154.1 on 8 and 4167 DF,  p-value: < 2.2e-16
```

There are two break points in this model. The first occurs in 2011 and the second ones occurs in 2019. I would have to some more research on why the structural break occured in 2011 my inital guess would be something with global warming however I would not be overly confident in that.Looking at the model however it seems to stabelize more after that point so maybe it is a increase in stabilization within the stock. For the second it is at the start of covid and all resorts and vacation type events were stopped due to covid. This caused a increase in volatility in the stock which you can see in the returns.

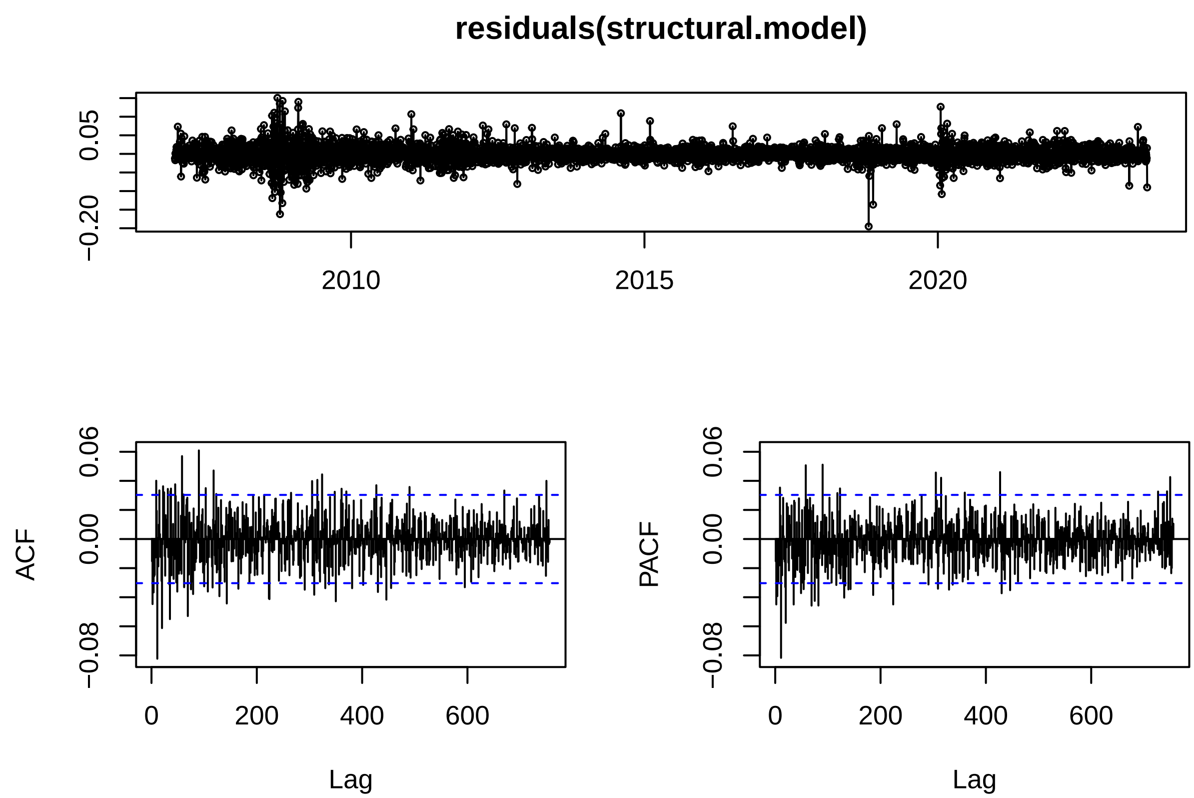
```r
plot(vResorts)
lines(ts(fitted(structural.model), frequency = 252, start = 2007), col = "cyan")
lines(bp_vResorts, breaks= 2, col = "red",lty = 1 )
```



testing the residuals of the final model:

There is definitely some auto correlation still in the model. Especially at the initial residuals. We don't have any lags in our model currently however as I mainly focused on the structural breaks for this part. The residuals are not overly seasonal however which shows that our intial differensing in seasonality is still correct taking into our tests for seasonality. In the interest of getting this assignment done at a reasonable time I am going to move on as this being my current model. I understand that I should look for some lags. I will test a simple model with 2 ar terms just to see but I am not going to search any more at the moment.

```
tsdisplay(residuals(structural.model))
```



dynamic model creation 2lags

```
dynamic.model <- dynlm(vResorts ~ L(vResorts, 1:2) + breakfactor(bp_vResorts, breaks = 2)/(Expedia + CO
summary(dynamic.model)
```

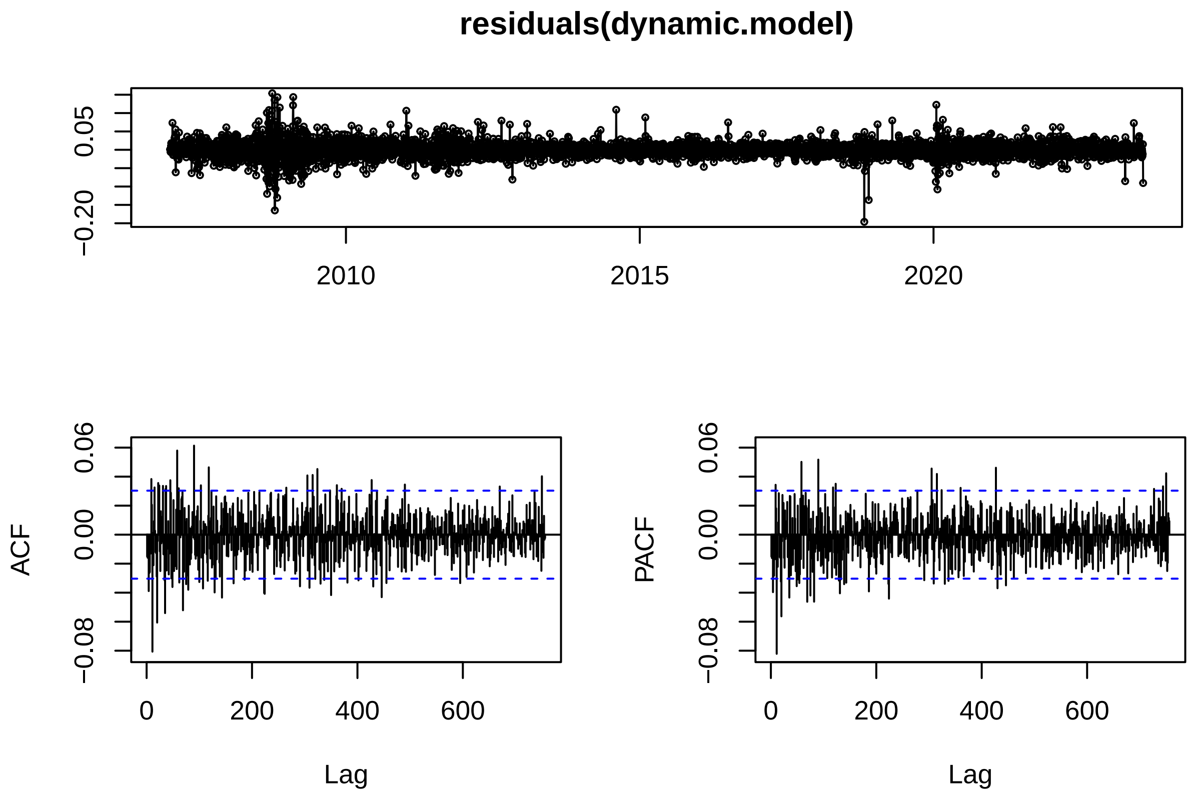
```
##
## Time series regression with "ts" data:
## Start = 2007(3), End = 2023(144)
##
## Call:
## dynlm(formula = vResorts ~ L(vResorts, 1:2) + breakfactor(bp_vResorts,
##     breaks = 2)/(Expedia + COil))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.196081 -0.009380  0.000136  0.009131  0.153814
##
## Coefficients:
##                                                Estimate Std. Error
## (Intercept)                                  -0.0002116  0.0005954
## L(vResorts, 1:2)1                            -0.0002249  0.0136592
## L(vResorts, 1:2)2                            -0.0303758  0.0136601
## breakfactor(bp_vResorts, breaks = 2)segment2          0.0010836  0.0007540
## breakfactor(bp_vResorts, breaks = 2)segment3          0.0003298  0.0009156
## breakfactor(bp_vResorts, breaks = 2)segment1:Expedia  0.4774389  0.0197211
## breakfactor(bp_vResorts, breaks = 2)segment2:Expedia  0.1445343  0.0188849
```

```
## breakfactor(bp_vResorts, breaks = 2)segment3:Expedia  0.4375517  0.0209559
## breakfactor(bp_vResorts, breaks = 2)segment1:COil     0.1155955  0.0221596
## breakfactor(bp_vResorts, breaks = 2)segment2:COil     0.0230580  0.0219509
## breakfactor(bp_vResorts, breaks = 2)segment3:COil    -0.0197111  0.0167585
##                                                      t value Pr(>|t|)
## (Intercept)                                           -0.355   0.7223
## L(vResorts, 1:2)1                                     -0.016   0.9869
## L(vResorts, 1:2)2                                     -2.224   0.0262 *
## breakfactor(bp_vResorts, breaks = 2)segment2           1.437   0.1507
## breakfactor(bp_vResorts, breaks = 2)segment3           0.360   0.7187
## breakfactor(bp_vResorts, breaks = 2)segment1:Expedia  24.209  < 2e-16 ***
## breakfactor(bp_vResorts, breaks = 2)segment2:Expedia   7.653 2.42e-14 ***
## breakfactor(bp_vResorts, breaks = 2)segment3:Expedia  20.880  < 2e-16 ***
## breakfactor(bp_vResorts, breaks = 2)segment1:COil      5.216 1.91e-07 ***
## breakfactor(bp_vResorts, breaks = 2)segment2:COil      1.050   0.2936
## breakfactor(bp_vResorts, breaks = 2)segment3:COil     -1.176   0.2396
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02088 on 4163 degrees of freedom
## Multiple R-squared:  0.2293, Adjusted R-squared:  0.2274
## F-statistic: 123.9 on 10 and 4163 DF,  p-value: < 2.2e-16
```

checking for auto cor.

There is still auto cor which I have really been struggling to get rid of in these data sets. I am just going to move on for now in the interest of time. For my final project I will find a much better way to eliminate auto cor.

```
tsdisplay(residuals(dynamic.model))
```

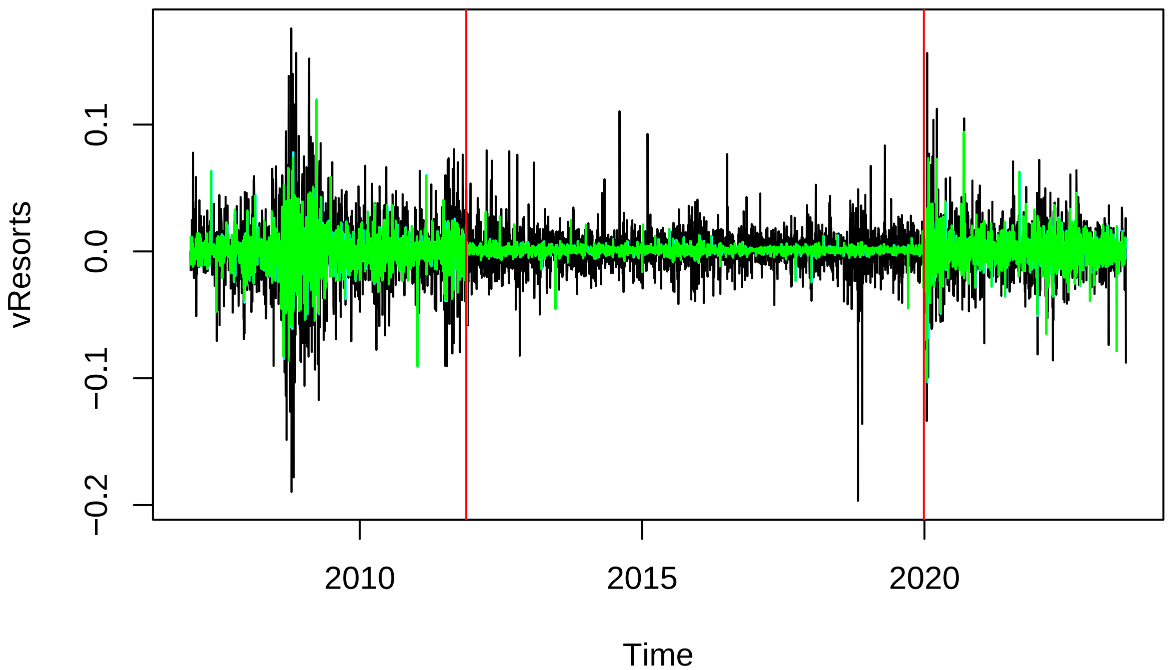


Final model equation: I didn't change the dates into the actual dates or the numbers in t that they would have to be I just left them as the values from the breakpoints test.

$$(1-L)(1-L^2)log(VailResorts) = -.0002116 - 0.0002251(1-L) - 0.0002251(1-L^2) + (t < (2011) + 224)(0.4774390 * Expedia + 0.$$

the dynamic model still doesn't catch the dip near around the start of 2019 which is unfortunate.

```
plot(vResorts)
lines(ts(fitted(structural.model), frequency = 252, start = 2007), col = "cyan")
lines(ts(fitted(dynamic.model), frequency = 252, start = 2007), col = "green")
lines(bp_vResorts, breaks= 2, col = "red",lty = 1 )
```



Question 4

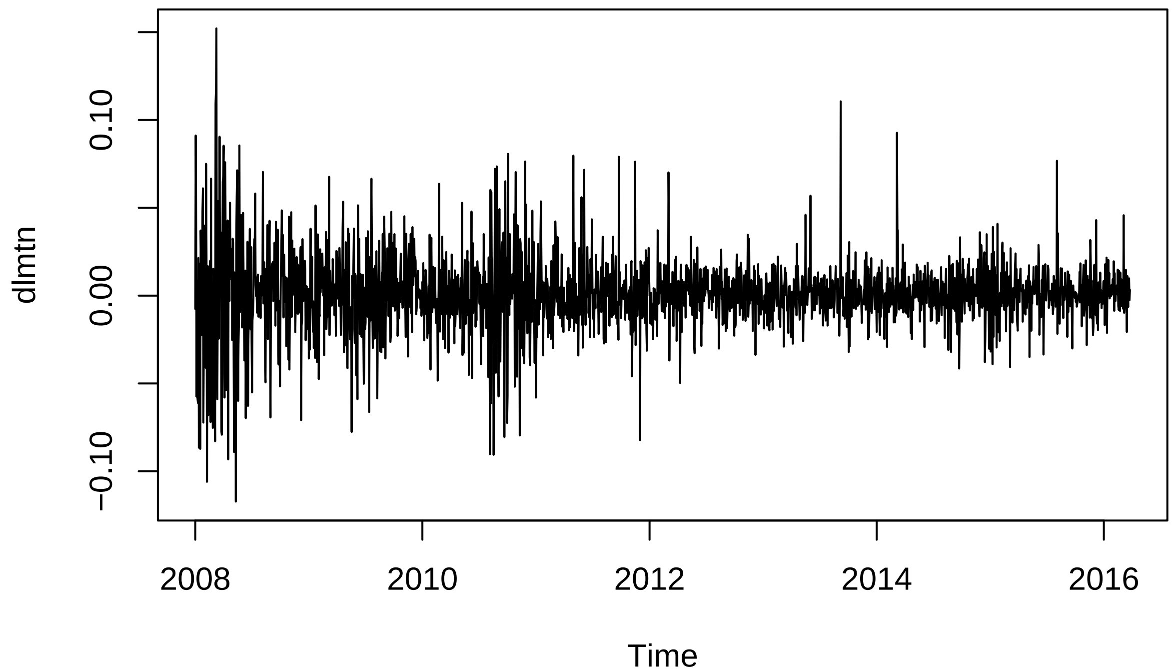
Based on how we were unable to eliminate auto cor in our old model as well as the prevalence of break points we are going to change the indexing of our final model. For my final project I have some snow data that I am going to compare these variables too. That data starts in 2008-12-31 and ends at 2017-03-30. So these will be our new start and end dates. I feel most of the issues that would arise from this model especially in relation to break points probably comes from what happened around covid as well so I feel this smaller section will be much better to work with.

```
startdate = "2008-12-31"
enddate = "2017-03-30"
```

calculate new first difference and returns

```
mtn <- window(mtn, start = startdate, end = enddate)
dlmtn <- ts(na.omit(diff(log(mtn$MTN.Adjusted))), freq=252, start = 2008)
dlexp <- ts(na.omit(diff(log(mtn$EXPE.Adjusted))), freq=252, start = 2008)
dcOil <- ts(na.omit(diff(log(mtn$DCOILWTICO))), freq=252, start = 2008)#we are going to logcOil as if y
          #which is easier to work with than a random walk with drift
```

```
ts.plot(dlmtn)
```



now we get zero break points which is what we want.

```
bp_mtn = breakpoints(dlmtn~1)
breakpoints(bp_mtn)
```

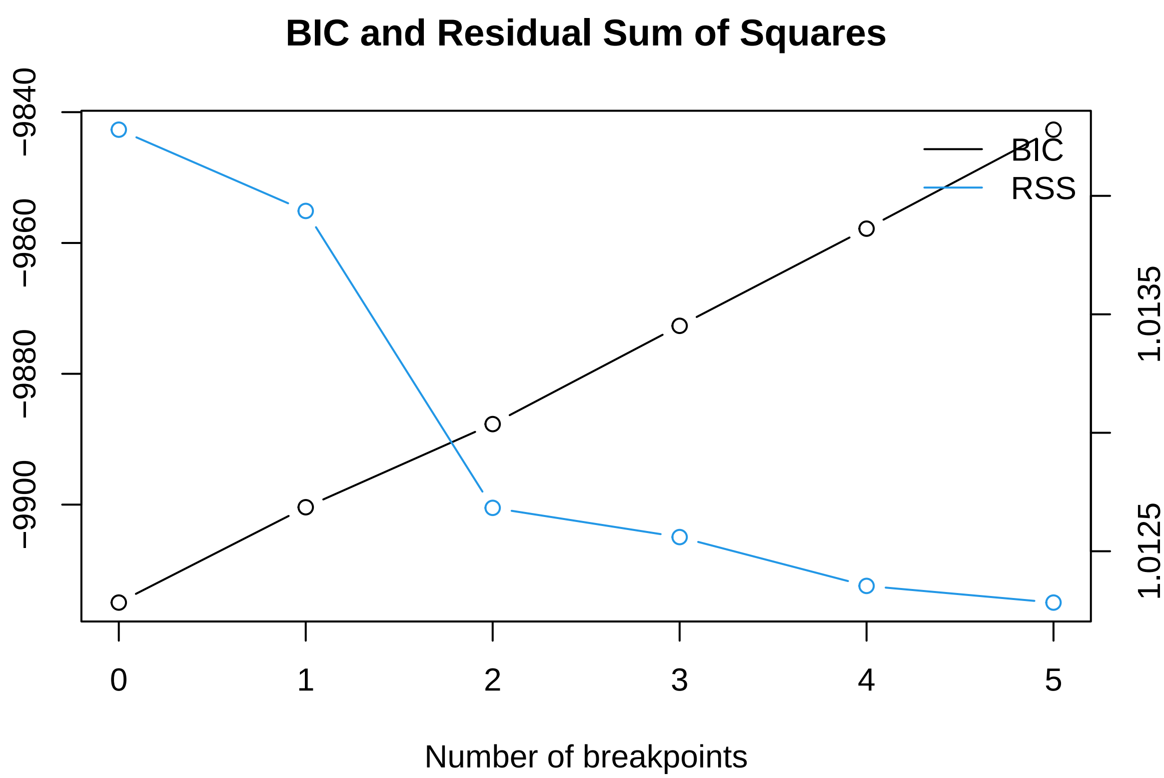
```
##
##   Optimal 1-segment partition:
##
## Call:
## breakpoints.breakpointsfull(obj = bp_mtn)
##
## Breakpoints at observation number:
## NA
##
## Corresponding to breakdates:
## NA
```

```
summary(bp_mtn)
```

```
##
##   Optimal (m+1)-segment partition:
##
## Call:
## breakpoints.formula(formula = dlmtn ~ 1)
##
```

```
## Breakpoints at observation number:
##
## m = 1       664
## m = 2   329 664
## m = 3   329 664          1697
## m = 4   329 664 977 1351
## m = 5   329 664 977 1351 1752
##
## Corresponding to breakdates:
##
## m = 1            2010(160)
## m = 2   2009(77) 2010(160)
## m = 3   2009(77) 2010(160)                     2014(185)
## m = 4   2009(77) 2010(160) 2011(221) 2013(91)
## m = 5   2009(77) 2010(160) 2011(221) 2013(91) 2014(240)
##
## Fit:
##
## m   0         1         2         3         4         5
## RSS     1.014     1.014     1.013     1.013     1.012     1.012
## BIC -9914.971 -9900.398 -9887.688 -9872.666 -9857.813 -9842.681
```

```r
plot(bp_mtn)
```



with the other variables taken into consideration we have 1 break point. Probably the 2011 break point that we had in the more complex model.

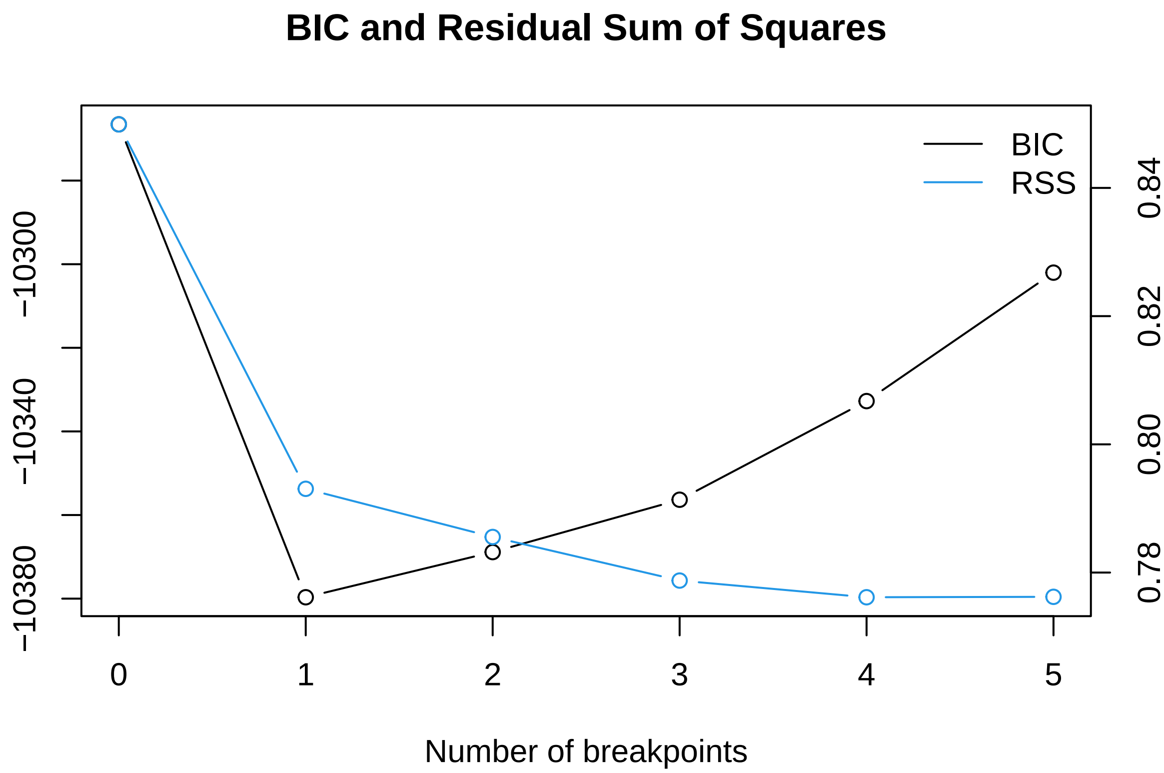
```r
bp_mtn = breakpoints(dlmtn~dlexp + dcOil)
breakpoints(bp_mtn)
```

```
##
##   Optimal 2-segment partition:
##
## Call:
## breakpoints.breakpointsfull(obj = bp_mtn)
##
## Breakpoints at observation number:
## 749
##
## Corresponding to breakdates:
## 2010(245)
```

```r
summary(bp_mtn)
```

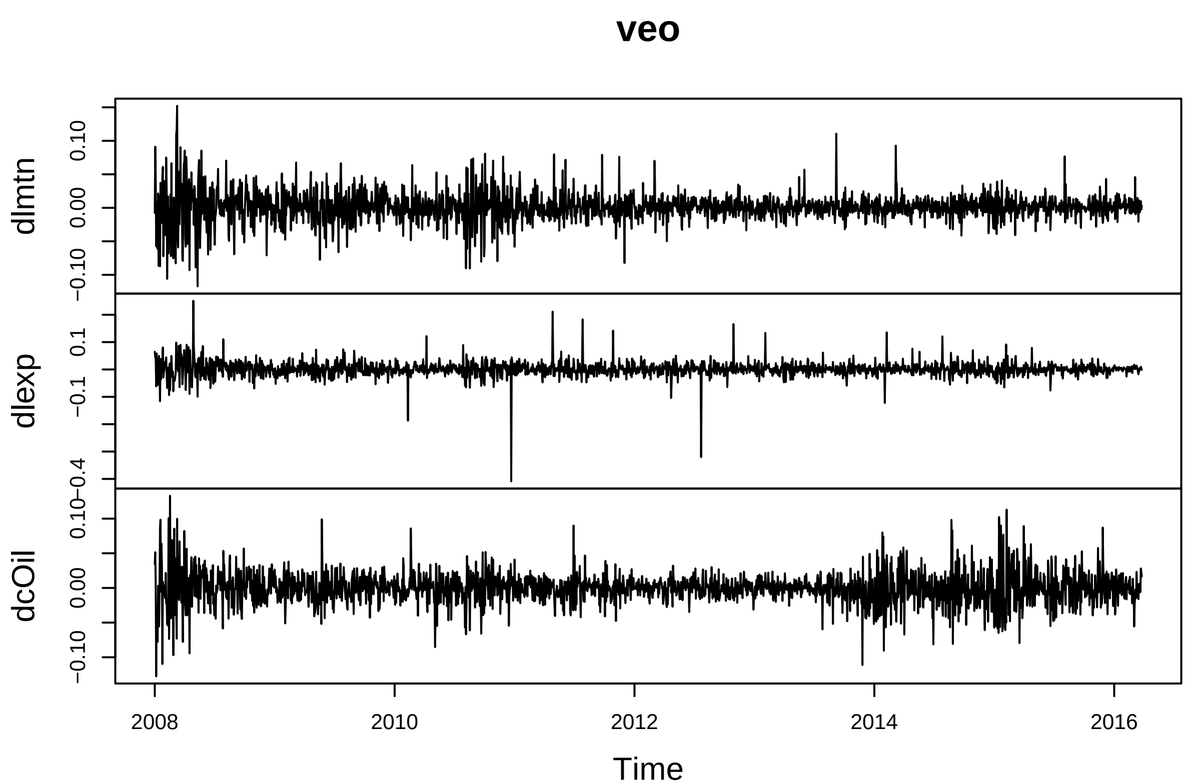
```
##
##   Optimal (m+1)-segment partition:
##
## Call:
## breakpoints.formula(formula = dlmtn ~ dlexp + dcOil)
##
## Breakpoints at observation number:
##
## m = 1       749
## m = 2       749 1231
## m = 3   313 749 1231
## m = 4   313 749 1157      1656
## m = 5   313 749 1125 1437 1750
##
## Corresponding to breakdates:
##
## m = 1             2010(245)
## m = 2             2010(245) 2012(223)
## m = 3   2009(61)  2010(245) 2012(223)
## m = 4   2009(61)  2010(245) 2012(149)           2014(144)
## m = 5   2009(61)  2010(245) 2012(117) 2013(177) 2014(238)
##
## Fit:
##
## m   0          1          2          3          4          5
## RSS  8.499e-01  7.931e-01  7.855e-01  7.788e-01  7.761e-01  7.762e-01
## BIC -1.027e+04 -1.038e+04 -1.037e+04 -1.036e+04 -1.033e+04 -1.030e+04
```

```r
plot(bp_mtn)
```



Now we are going to conduct a var on the data

```
veo <- cbind(dlmtn, dlexp, dcOil)
plot(veo)
```




```r
summary(veo)
```

```
##      dlmtn                 dlexp                 dcOil
##  Min.   :-0.1172727   Min.   :-0.4088060   Min.   :-1.274e-01
##  1st Qu.:-0.0085995   1st Qu.:-0.0105160   1st Qu.:-1.211e-02
##  Median : 0.0007734   Median : 0.0009216   Median : 0.000e+00
##  Mean   : 0.0010006   Mean   : 0.0011792   Mean   : 5.796e-05
##  3rd Qu.: 0.0110192   3rd Qu.: 0.0124622   3rd Qu.: 1.163e-02
##  Max.   : 0.1521477   Max.   : 0.2508945   Max.   : 1.330e-01
```

selecting lag length

I'm going to go with the larger number of lags just because I feel like there should be more lags in the data based on my earlier analysis. So we are going to use AIC which thinks 3 lags. AIC and FPE = 3, HQ and SC = 1

```r
VARselect(veo, lag.max = 15, type = "none")
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      3      1      1      3
##
## $criteria
##                     1             2             3             4             5
## AIC(n) -2.259872e+01 -2.260742e+01 -2.260898e+01 -2.260698e+01 -2.260106e+01
```

```
## HQ(n)  -2.258970e+01 -2.258938e+01 -2.258193e+01 -2.257091e+01 -2.255597e+01
## SC(n)  -2.257412e+01 -2.255822e+01 -2.253519e+01 -2.250858e+01 -2.247807e+01
## FPE(n)  1.532860e-10  1.519576e-10  1.517201e-10  1.520244e-10  1.529269e-10
##                    6             7             8             9            10
## AIC(n) -2.259653e+01 -2.259380e+01 -2.259077e+01 -2.258582e+01 -2.258035e+01
## HQ(n)  -2.254241e+01 -2.253067e+01 -2.251862e+01 -2.250466e+01 -2.249016e+01
## SC(n)  -2.244893e+01 -2.242160e+01 -2.239398e+01 -2.236443e+01 -2.233436e+01
## FPE(n)  1.536223e-10  1.540419e-10  1.545096e-10  1.552755e-10  1.561286e-10
##                   11            12            13            14            15
## AIC(n) -2.257320e+01 -2.256789e+01 -2.256695e+01 -2.256673e+01 -2.256356e+01
## HQ(n)  -2.247400e+01 -2.245967e+01 -2.244971e+01 -2.244047e+01 -2.242828e+01
## SC(n)  -2.230261e+01 -2.227270e+01 -2.224716e+01 -2.222235e+01 -2.219458e+01
## FPE(n)  1.572482e-10  1.580861e-10  1.582355e-10  1.582702e-10  1.587737e-10
```

creating the VAR model with 3 lags

```r
varvoe <- VAR(veo, lag.max = 13, type ="none",ic="FPE")
varvoe
```

```
##
## VAR Estimation Results:
## =======================
##
## Estimated coefficients for equation dlmtn:
## ==========================================
## Call:
## dlmtn = dlmtn.l1 + dlexp.l1 + dcOil.l1 + dlmtn.l2 + dlexp.l2 + dcOil.l2 + dlmtn.l3 + dlexp.l3 + dcOil
##
##     dlmtn.l1     dlexp.l1     dcOil.l1     dlmtn.l2     dlexp.l2     dcOil.l2
## -0.010528119  0.023307124  0.012858709 -0.115374388  0.003691191  0.037082864
##     dlmtn.l3     dlexp.l3     dcOil.l3
## -0.047581855  0.016463837 -0.008653411
##
##
## Estimated coefficients for equation dlexp:
## ==========================================
## Call:
## dlexp = dlmtn.l1 + dlexp.l1 + dcOil.l1 + dlmtn.l2 + dlexp.l2 + dcOil.l2 + dlmtn.l3 + dlexp.l3 + dcOil
##
##    dlmtn.l1    dlexp.l1    dcOil.l1    dlmtn.l2    dlexp.l2    dcOil.l2
## -0.05044044  0.02582342 -0.02524802 -0.03656146 -0.02224428  0.04949987
##    dlmtn.l3    dlexp.l3    dcOil.l3
##  0.05095687 -0.02166700 -0.04600411
##
##
## Estimated coefficients for equation dcOil:
## ==========================================
## Call:
## dcOil = dlmtn.l1 + dlexp.l1 + dcOil.l1 + dlmtn.l2 + dlexp.l2 + dcOil.l2 + dlmtn.l3 + dlexp.l3 + dcOil
##
##      dlmtn.l1      dlexp.l1      dcOil.l1      dlmtn.l2      dlexp.l2
##  0.0122566052  0.0001836115 -0.0417598160 -0.0376626153  0.0389319213
##      dcOil.l2      dlmtn.l3      dlexp.l3      dcOil.l3
## -0.0256570515 -0.0382748378 -0.0252778041  0.0032641968
```

Estimations of var in a vector system

$mtn_t = -0.010529mtn_{t-1} + 0.023307exp_{t-1} + 0.012859COil_{t-1} - 0.115375mtn_{t-2} + 0.003692exp_{t-2} + 0.037083COil_{t-2} - 0.04758$

$exp_t = -0.05044mtn_{t-1} + 0.02582exp_{t-1} - 0.02525COil_{t-1} - 0.03656mtn_{t-2} - 0.02224exp_{t-2} + 0.04950COil_{t-2} + 0.05096mtn_{t-3}$

$COil_t = 0.0122560mtn_{t-1} + 0.0001837exp_{t-1} - 0.0417598COil_{t-1} - 0.0376620mtn_{t-2} + 0.0389318exp_{t-2} + -0.0256571COil_{t-2} -$

my kniting was breaking doing the fancy matrix for these variables. However this should be in a matrix together so

$$y_t = matrix(mtn_t, exp_t, COil_t)$$

```r
summary(varvoe)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: dlmtn, dlexp, dcOil
## Deterministic variables: none
## Sample size: 2072
## Log Likelihood: 14558.451
## Roots of the characteristic polynomial:
## 0.438 0.438 0.3919 0.3919 0.3842 0.342 0.342 0.303 0.2574
## Call:
## VAR(y = veo, type = "none", lag.max = 13, ic = "FPE")
##
##
## Estimation results for equation dlmtn:
## ======================================
## dlmtn = dlmtn.l1 + dlexp.l1 + dcOil.l1 + dlmtn.l2 + dlexp.l2 + dcOil.l2 + dlmtn.l3 + dlexp.l3 + dcOil
##
##           Estimate Std. Error t value Pr(>|t|)
## dlmtn.l1 -0.010528   0.023999  -0.439   0.6609
## dlexp.l1  0.023307   0.019095   1.221   0.2224
## dcOil.l1  0.012859   0.021123   0.609   0.5428
## dlmtn.l2 -0.115374   0.023818  -4.844 1.37e-06 ***
## dlexp.l2  0.003691   0.019084   0.193   0.8467
## dcOil.l2  0.037083   0.021112   1.756   0.0792 .
## dlmtn.l3 -0.047582   0.023922  -1.989   0.0468 *
## dlexp.l3  0.016464   0.019060   0.864   0.3878
## dcOil.l3 -0.008653   0.021130  -0.410   0.6822
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.02191 on 2063 degrees of freedom
## Multiple R-Squared: 0.01629, Adjusted R-squared: 0.012
## F-statistic: 3.797 on 9 and 2063 DF,  p-value: 9.116e-05
##
##
## Estimation results for equation dlexp:
## ======================================
```

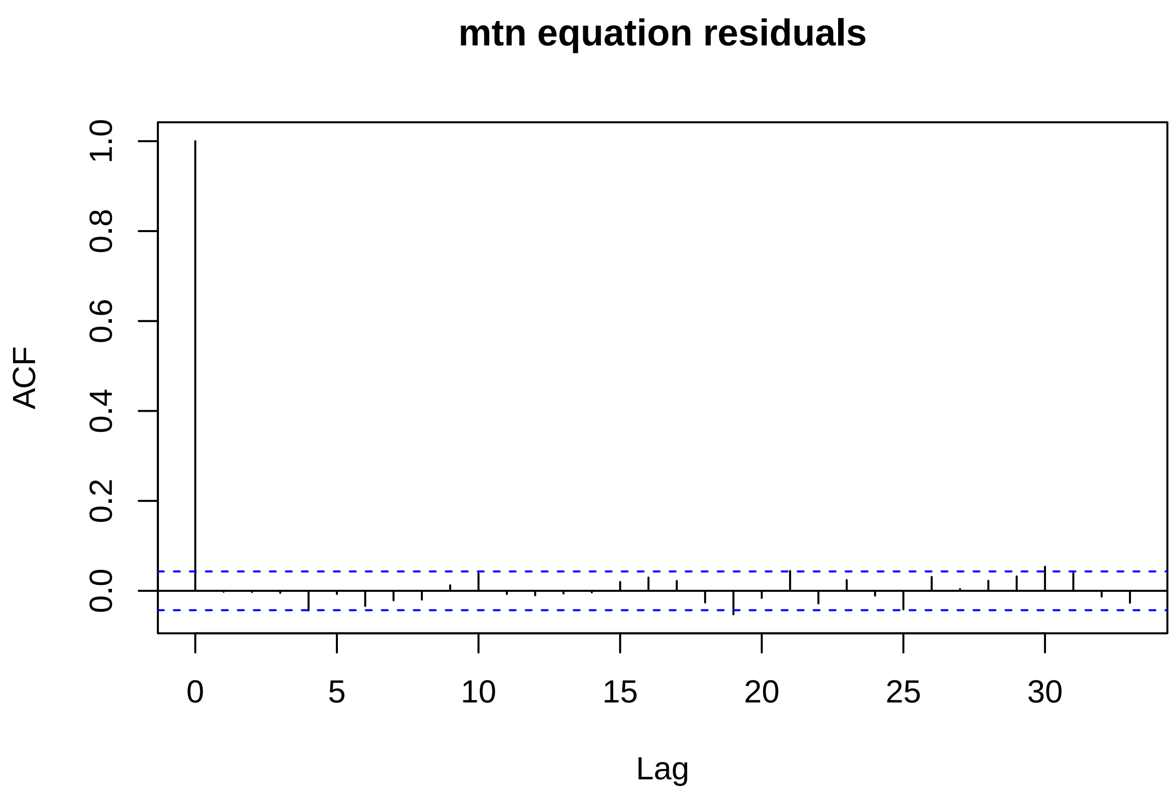
```
## dlexp = dlmtn.l1 + dlexp.l1 + dcOil.l1 + dlmtn.l2 + dlexp.l2 + dcOil.l2 + dlmtn.l3 + dlexp.l3 + dcOil
##
##          Estimate Std. Error t value Pr(>|t|)
## dlmtn.l1 -0.05044    0.03014  -1.674   0.0944 .
## dlexp.l1  0.02582    0.02398   1.077   0.2817
## dcOil.l1 -0.02525    0.02653  -0.952   0.3413
## dlmtn.l2 -0.03656    0.02991  -1.222   0.2217
## dlexp.l2 -0.02224    0.02397  -0.928   0.3535
## dcOil.l2  0.04950    0.02651   1.867   0.0620 .
## dlmtn.l3  0.05096    0.03004   1.696   0.0900 .
## dlexp.l3 -0.02167    0.02394  -0.905   0.3655
## dcOil.l3 -0.04600    0.02654  -1.734   0.0831 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.02751 on 2063 degrees of freedom
## Multiple R-Squared: 0.008288,    Adjusted R-squared: 0.003961
## F-statistic: 1.916 on 9 and 2063 DF,  p-value: 0.04571
##
##
## Estimation results for equation dcOil:
## ======================================
## dcOil = dlmtn.l1 + dlexp.l1 + dcOil.l1 + dlmtn.l2 + dlexp.l2 + dcOil.l2 + dlmtn.l3 + dlexp.l3 + dcOil
##
##            Estimate Std. Error t value Pr(>|t|)
## dlmtn.l1  0.0122566  0.0255075   0.481   0.6309
## dlexp.l1  0.0001836  0.0202953   0.009   0.9928
## dcOil.l1 -0.0417598  0.0224511  -1.860   0.0630 .
## dlmtn.l2 -0.0376626  0.0253160  -1.488   0.1370
## dlexp.l2  0.0389319  0.0202840   1.919   0.0551 .
## dcOil.l2 -0.0256571  0.0224393  -1.143   0.2530
## dlmtn.l3 -0.0382748  0.0254266  -1.505   0.1324
## dlexp.l3 -0.0252778  0.0202580  -1.248   0.2122
## dcOil.l3  0.0032642  0.0224587   0.145   0.8845
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.02329 on 2063 degrees of freedom
## Multiple R-Squared: 0.007712,    Adjusted R-squared: 0.003383
## F-statistic: 1.782 on 9 and 2063 DF,  p-value: 0.06689
##
##
##
## Covariance matrix of residuals:
##           dlmtn     dlexp     dcOil
## dlmtn 4.788e-04 0.0002328 8.622e-05
## dlexp 2.328e-04 0.0007556 1.076e-04
## dcOil 8.622e-05 0.0001076 5.422e-04
##
## Correlation matrix of residuals:
##        dlmtn  dlexp  dcOil
## dlmtn 1.0000 0.3870 0.1692
```

```
## dlexp 0.3870 1.0000 0.1682
## dcOil 0.1692 0.1682 1.0000
```

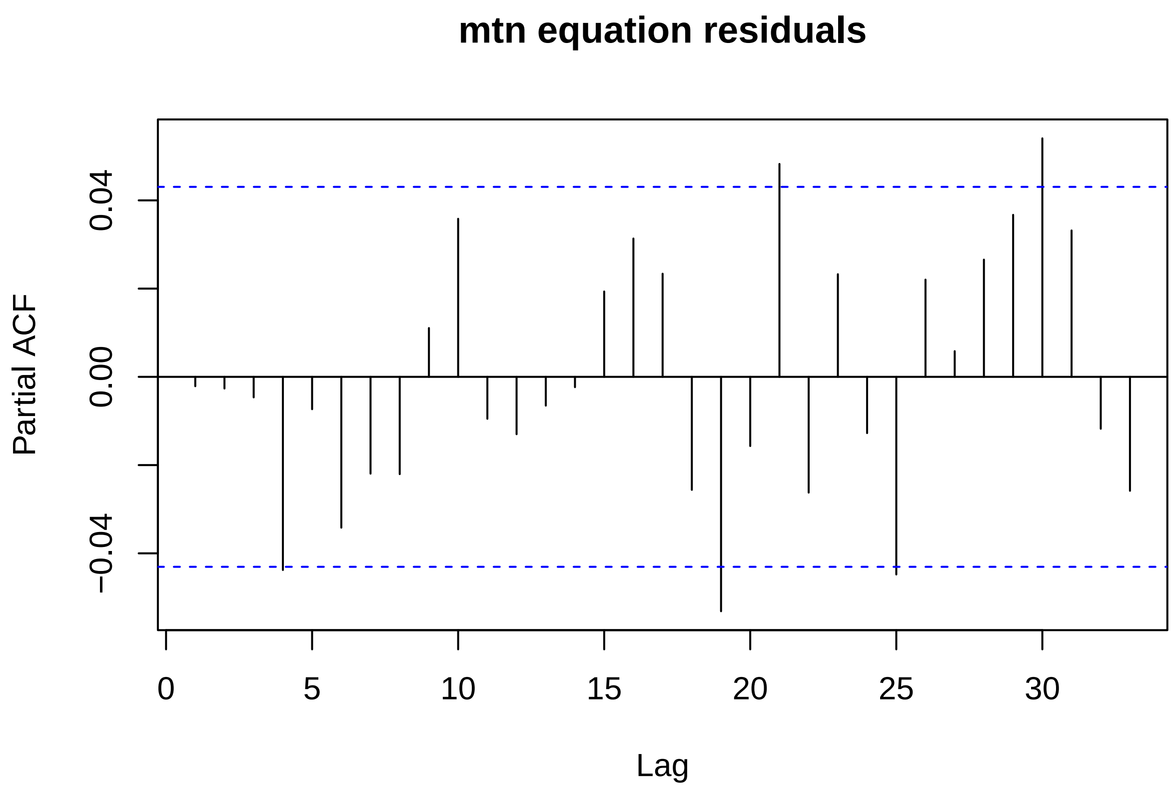
cannot plot in a markdown to big but you can just plot in terminal so this plot isn't in the markdown itself but I looked at in in the terminal. Bellow is my attempt to plot 1 of them maybe it will be complete on the markdown not entierly sure. dlmtn - really good no probs with auto dlexp - same no issue dcOil - same all residuals and auto looks good.

A couple sightly significant lags for PACF and seasonal looking residuals auto cor for afc however neither to concerning really.

```r
#cannot plot this it breaks my markdown
#plot(varvoe, name = "dlmtn")
acf(varvoe$varresult$dlmtn$residuals,main="mtn equation residuals")
```




```r
pacf(varvoe$varresult$dlmtn$residuals,main="mtn equation residuals")
```



No issues with auto cor so we are just going to go with this model

Granger Causality

investigating relationship testing for causality of oil on the other markets. This would be interesting as this is our economic variable so we would expect expensive oil to hurt both of the other markets.

I do not reject the null of there being no causality of crude oil on the other variables (mtn and exp)

```r
roots(varvoe)
```

```
## [1] 0.4379678 0.4379678 0.3919093 0.3919093 0.3841527 0.3420270 0.3420270
## [8] 0.3030131 0.2574168
```

```r
causality(varvoe, cause="dcOil")
```

```
## $Granger
##
##  Granger causality H0: dcOil do not Granger-cause dlmtn dlexp
##
## data:  VAR object varvoe
## F-Test = 1.6736, df1 = 6, df2 = 6189, p-value = 0.1231
##
##
```

```
## $Instant
##
##  H0: No instantaneous causality between: dcOil and dlmtn dlexp
##
## data:  VAR object varvoe
## Chi-squared = 81.554, df = 2, p-value < 2.2e-16
```

```r
#causality(varvoe, cause="dlexp")
```

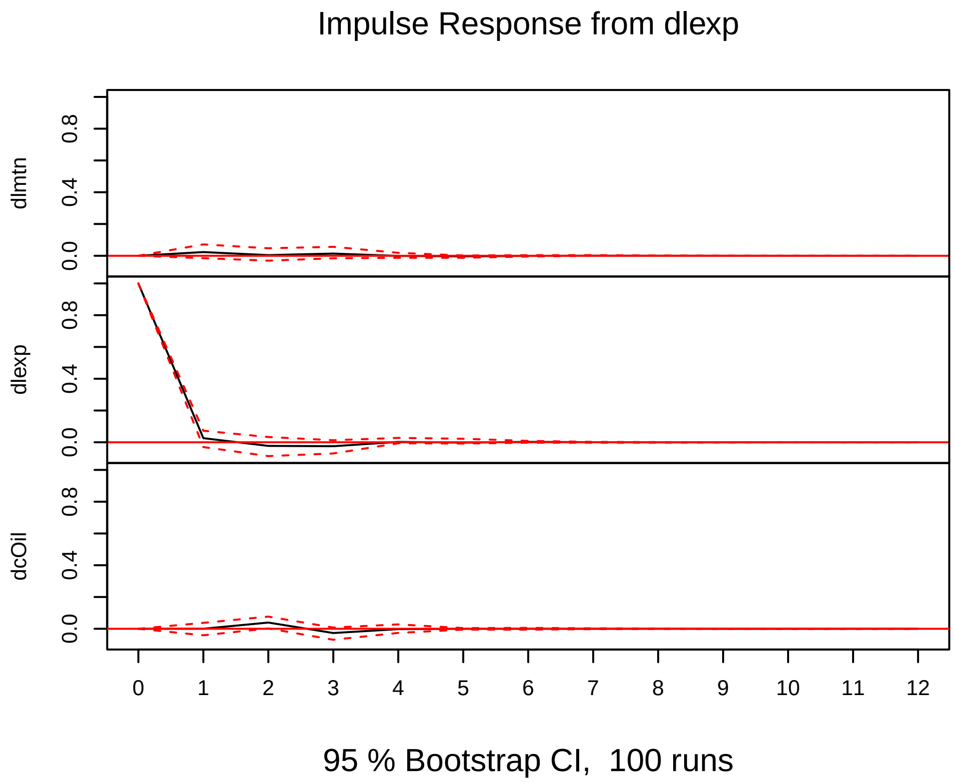
4b)

```r
serial.test(varvoe, lags.pt = 16)
```

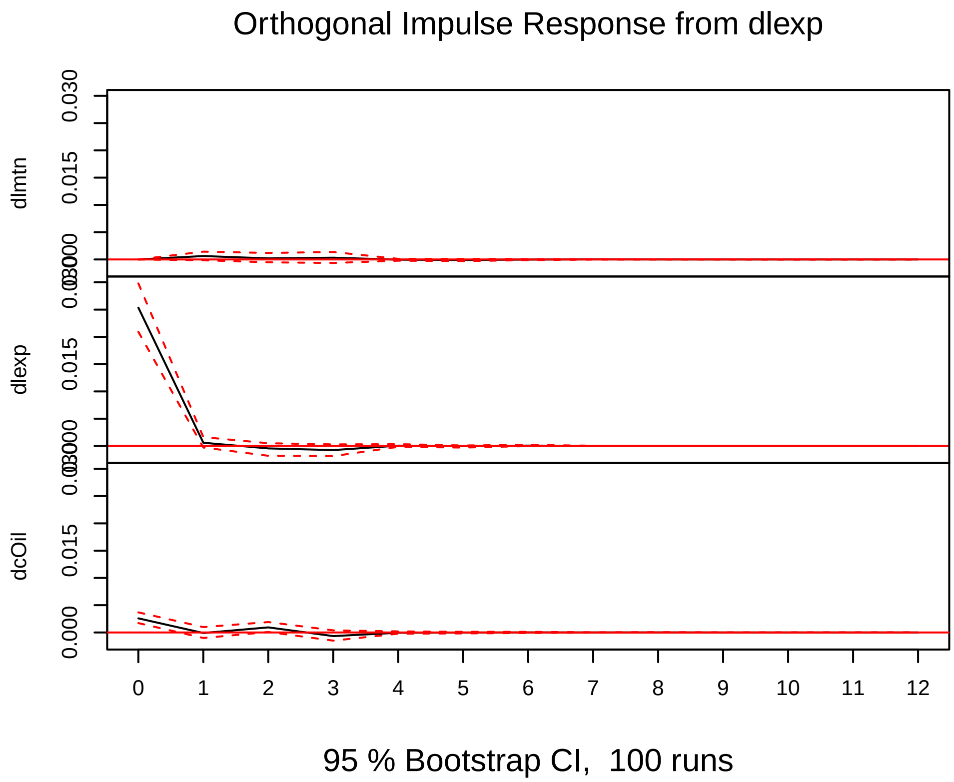
```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object varvoe
## Chi-squared = 136.72, df = 117, p-value = 0.1028
```

From the plot we cannot really determine any impulse responses or causal relationships within the data. All that I really can see is that there is an negative response in Expedia stock 2 days after the initial shock then dying out. We cannot conclude any causal relationships between these variables.There is a positive relationship for the current time period for an Expedia shock and crude oil.

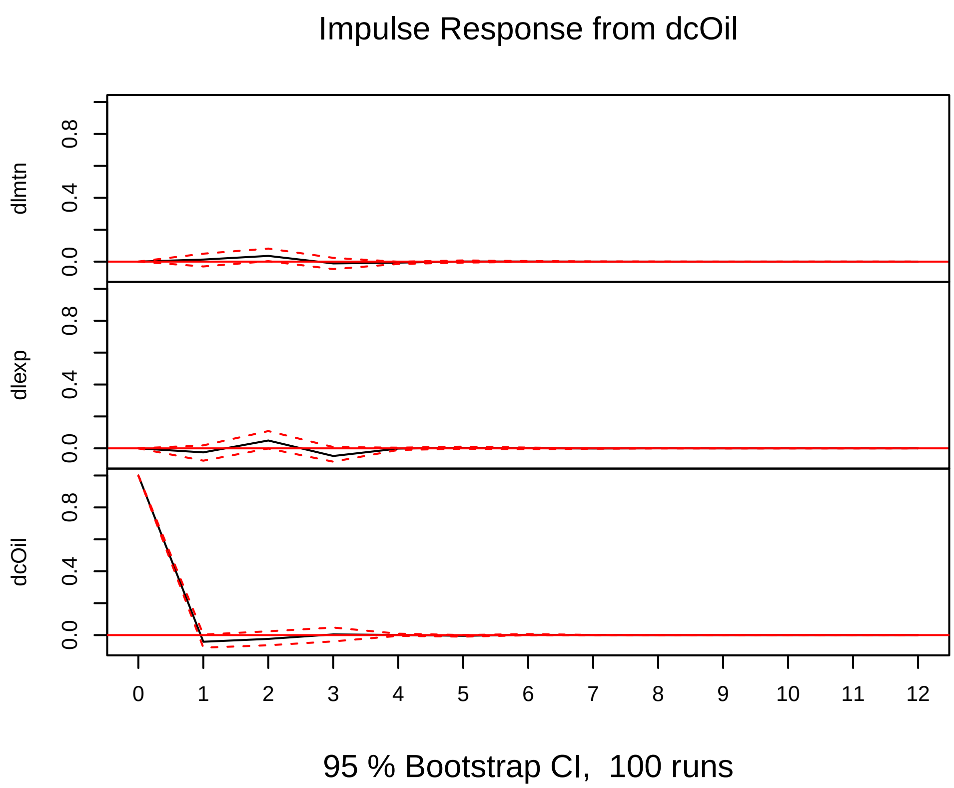
```r
plot(irf(varvoe,n.ahead=12,ortho=F,impulse = "dlexp"))
```



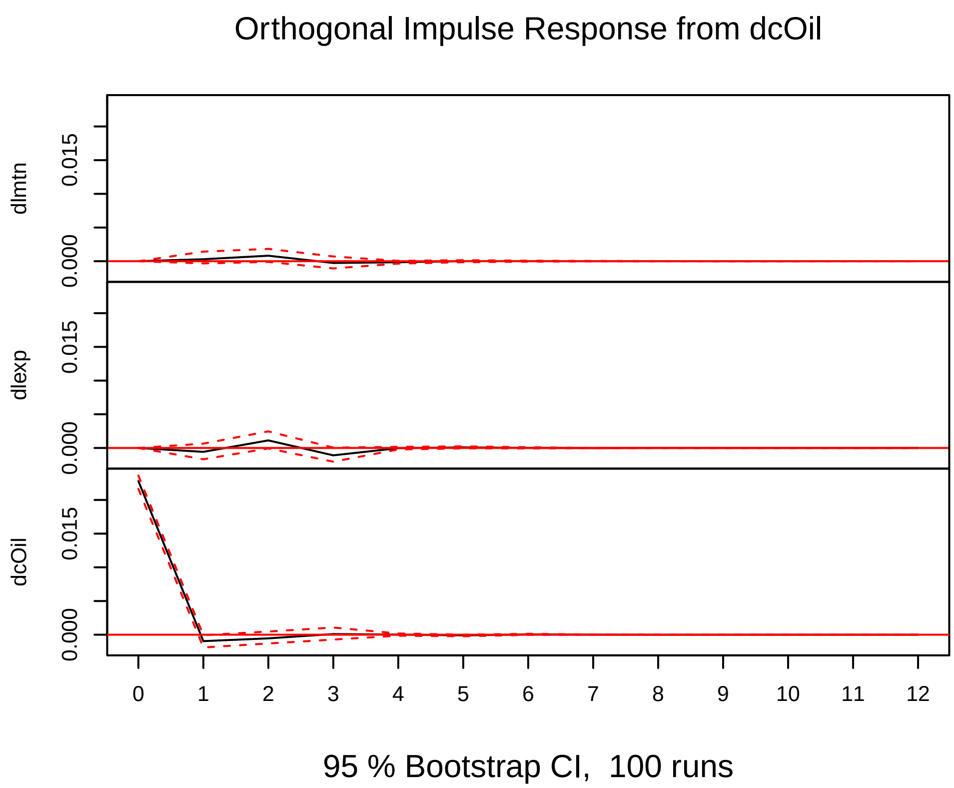

```r
plot(irf(varvoe,n.ahead=12,ortho=T,impulse = "dlexp"))
```




```
plot(irf(varvoe,n.ahead=12,ortho=F,impulse = "dcOil"))
```




```r
plot(irf(varvoe,n.ahead=12,ortho=T,impulse = "dcOil"))
```



The non-orthogonalized response of differenced crude oil says that for a one-time, one unit increase in the real crude oil predicts that we wont see any change in either returns for Expedia or returns for Vail resorts, in the second period. In this case two days later. For non of the variables do we see a causal relationship in any period except for one case where there is one in the current time period. From this however we can conclude if there is a one percent increase in Expedia stock there is normally a 0.002588979 increase in differenced dcOil occurring at the same time which then dies out after that period.

```
temp <- irf(varvoe,n.ahead=12,ortho=T,impulse = "dlexp",response="dcOil")
temp$irf$dlexp[1,]
```

```
##       dcOil
## 0.002588974
```