Econ 104 Project 2 Marcus Young, Geoffrey Penarubia, Kyle Almon

2023-02-16

```
library(AER)
```

```
## Loading required package: car

## Loading required package: carData

## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

## Loading required package: sandwich

## Loading required package: survival
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::recode() masks car::recode()
## x purrr::some()   masks car::some()
```

```
library(readr)
library(knitr)
library(xtable)
library(effects)
```

```
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```
library(broom)
library(jtools)
library(leaps)
library(car)
library(Boruta)
library(lmtest)
library(AICcmodavg)
library(flexmix)
```

```
## Loading required package: lattice
```

```
library(caret)
```

```
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
##
## The following object is masked from 'package:survival':
##
##     cluster
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(RColorBrewer)
library(ggplot2)
library(rlang)
```

```
##
## Attaching package: 'rlang'
##
## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice
```

```
library(base)
library(xfun)
```

```
##
## Attaching package: 'xfun'
##
## The following objects are masked from 'package:base':
##
##     attr, isFALSE
```

```
library(tinytex)
```

```
##
## Attaching package: 'tinytex'
##
## The following object is masked from 'package:rlang':
##
##     check_installed
```

```
library(stats)
library(TSA)
```

```
##
## Attaching package: 'TSA'
##
## The following object is masked from 'package:readr':
##
```

```
##     spec
##
## The following objects are masked from 'package:stats':
##
##     acf, arima
##
## The following object is masked from 'package:utils':
##
##     tar
```

```
library(timeSeries)
```

```
## Loading required package: timeDate
##
## Attaching package: 'timeDate'
##
## The following objects are masked from 'package:TSA':
##
##     kurtosis, skewness
##
## The following object is masked from 'package:xtable':
##
##     align
##
##
## Attaching package: 'timeSeries'
##
## The following object is masked from 'package:zoo':
##
##     time<-
```

```
library(fUnitRoots)
library(fBasics)
```

```
##
## Attaching package: 'fBasics'
##
## The following objects are masked from 'package:TSA':
##
##     kurtosis, skewness
##
## The following object is masked from 'package:flexmix':
##
##     getModel
##
## The following object is masked from 'package:car':
##
##     densityPlot
```

```
library(tseries)
```

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

```
library(timsac)
```

```r
library(TTR)
```

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

```r
library(fpp)
```

```
## Loading required package: forecast
## Registered S3 methods overwritten by 'forecast':
##   method       from
##   fitted.Arima TSA
##   plot.Arima   TSA
## Loading required package: fma
## Loading required package: expsmooth
```

```r
library(strucchange)
```

```
##
## Attaching package: 'strucchange'
##
## The following object is masked from 'package:stringr':
##
##     boundary
```

```r
library(lattice)
library(foreign)
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following objects are masked from 'package:fma':
##
##     cement, housing, petrol
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(car)
require(stats)
require(stats4)
```

```
## Loading required package: stats4
```

```r
library(KernSmooth)
```

```
## KernSmooth 2.23 loaded
## Copyright M. P. Wand 1997-2009
```

```r
library(fastICA)
library(cluster)
library(leaps)
library(mgcv)
```

```
## Loading required package: nlme
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:forecast':
##
##     getResponse
##
## The following object is masked from 'package:dplyr':
##
##     collapse
##
## This is mgcv 1.8-41. For overview type 'help("mgcv-package")'.
```

```
library(rpart)
library(pan)
library(mgcv)
library(DAAG)
```

```
##
## Attaching package: 'DAAG'
##
## The following object is masked from 'package:MASS':
##
##     hills
##
## The following objects are masked from 'package:fma':
##
##     milk, ozone
##
## The following object is masked from 'package:survival':
##
##     lung
##
## The following object is masked from 'package:car':
##
##     vif
```

```
library(TTR)
library(tis)
```

```
##
## Attaching package: 'tis'
##
## The following object is masked from 'package:mgcv':
##
##     ti
##
## The following object is masked from 'package:forecast':
##
##     easter
##
## The following object is masked from 'package:TTR':
##
##     lags
```

```
##
## The following objects are masked from 'package:timeSeries':
##
##     description, interpNA
##
## The following objects are masked from 'package:timeDate':
##
##     dayOfWeek, dayOfYear, isHoliday
##
## The following object is masked from 'package:dplyr':
##
##     between
```

```
require(graphics)
library(forecast)
library(xtable)
library(dynlm)
library(vars)
```

```
## Loading required package: urca
##
## Attaching package: 'urca'
##
## The following objects are masked from 'package:fUnitRoots':
##
##     punitroot, qunitroot, unitrootTable
```

```
library(ARDL)
```

```
## To cite the ARDL package in publications:
##
## Use this reference to refer to the validity of the ARDL package.
##
##   Natsiopoulos, Kleanthis, and Tzeremes, Nickolaos G. (2022). ARDL
##   bounds test for cointegration: Replicating the Pesaran et al. (2001)
##   results for the UK earnings equation using R. Journal of Applied
##   Econometrics, 37(5), 1079-1090. https://doi.org/10.1002/jae.2919
##
## Use this reference to cite this specific version of the ARDL package.
##
##   Kleanthis Natsiopoulos and Nickolaos Tzeremes (2023). ARDL: ARDL, ECM
##   and Bounds-Test for Cointegration. R package version 0.2.2.
##   https://CRAN.R-project.org/package=ARDL
```

```
HDLD <- read_csv("Home Depot Lowes Walmart Data.csv")
```

```
## New names:
## Rows: 61 Columns: 14
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (2): Date, Inflation_Rate dbl (8): Walmart_%Change, Walmart_Close,
## Home_Depot_%Change, Home_Depot_Clos... lgl (4): ...11, ...12, ...13, ...14
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...11`
## * `` -> `...12`
```

```
## * `` -> `...13`
## * `` -> `...14`
```

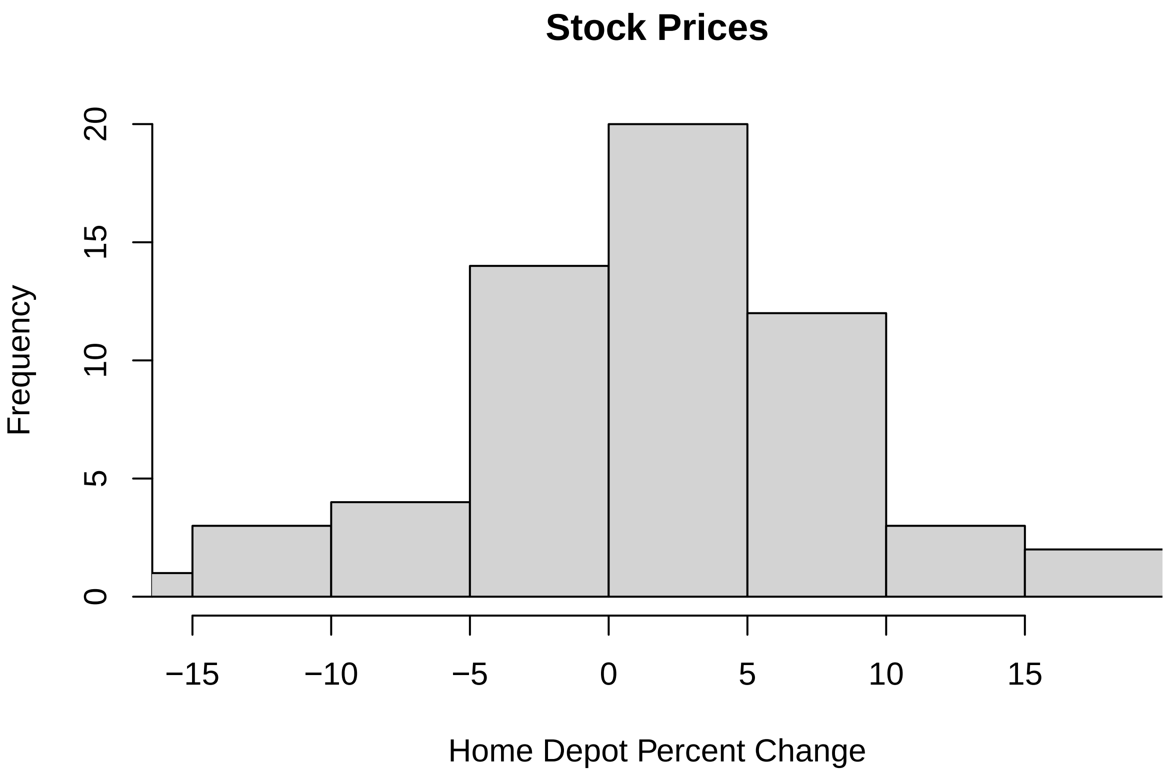
```r
HDPC <- HDLD$`Home_Depot_%Change`

LowesPC <- HDLD$`Lowes_%Change`

WPC <- HDLD$`Walmart_%Change`

InflationPC <- HDLD$`Inflation_Rate_%Change`

#For this project, our group decided that we would like to look at Stocks,
#and their change in percent of their stocks. Specifically, we decided to
#look at two of the biggest home improvement corporations in the US,
#Home Depot and Lowes, with a shared dynamic of being part of the S&P 500.
#In addition, we would like to see the effect that Inflation rates have on
#the stock price of one of the biggest retail stores in the nation, Walmart.
```

```r
#1
#Five Number summary
#Summary Comments- Something that caught our eye in the summary
#of our data was looking at the inflation rate change, with a minimum
#percent change of -78.623% and with a max of 447.458 %. Visually looking
#at the scatterplot, boxplots as well, this was definitely an outlier
#in the data.
summary(HDLD)
```

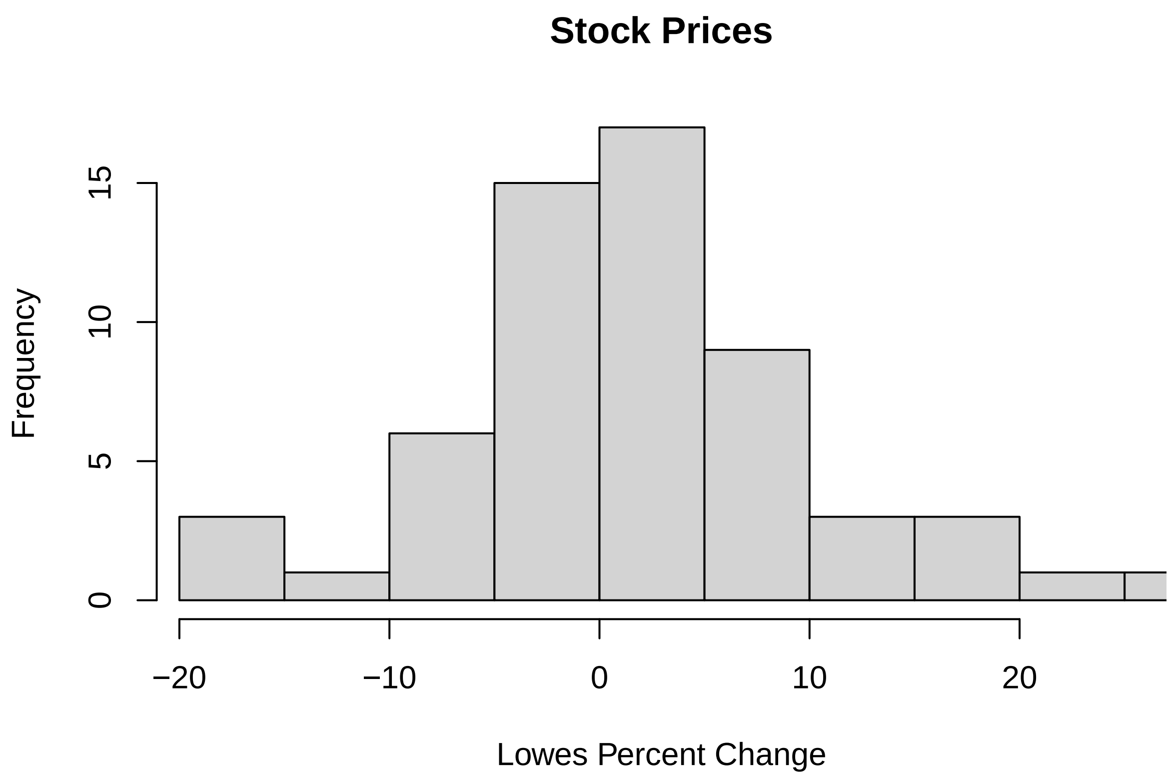
```
##      Date           Walmart_%Change   Walmart_Close    Home_Depot_%Change
##  Length:61          Min.   :-15.923   Min.   : 75.78   Min.   :-15.095
##  Class :character   1st Qu.: -2.403   1st Qu.: 99.62   1st Qu.: -3.013
##  Mode  :character   Median :  1.178   Median :124.43   Median :  1.645
##                     Mean   :  1.115   Mean   :118.66   Mean   :  1.463
##                     3rd Qu.:  4.485   3rd Qu.:137.71   3rd Qu.:  5.891
##                     Max.   : 10.181   Max.   :151.85   Max.   : 18.526
##                     NA's   :2         NA's   :2        NA's   :2
##  Home_Depot_Close Lowes_%Change      Lowes_Close     Inflation_Rate_%Change
##  Min.   :156.1    Min.   :-19.255   Min.   : 75.38   Min.   :-78.623
##  1st Qu.:186.8    1st Qu.: -2.168   1st Qu.: 99.98   1st Qu.: -6.008
##  Median :251.0    Median :  1.939   Median :150.04   Median :  2.790
##  Mean   :246.9    Mean   :  1.981   Mean   :145.92   Mean   :  9.892
##  3rd Qu.:297.1    3rd Qu.:  5.652   3rd Qu.:190.87   3rd Qu.: 10.325
##  Max.   :403.2    Max.   : 25.163   Max.   :252.46   Max.   :447.458
##  NA's   :2        NA's   :2         NA's   :2        NA's   :2
##  Inflation_Rate       Inflation       ...11           ...12
##  Length:61          Min.   :0.118   Mode:logical   Mode:logical
##  Class :character   1st Qu.:1.694   NA's:61        NA's:61
##  Mode  :character   Median :2.436
##                     Mean   :3.707
##                     3rd Qu.:5.806
##                     Max.   :9.060
##                     NA's   :2
##   ...13          ...14
##  Mode:logical   Mode:logical
##  NA's:61        NA's:61
##
```

```
##
##
##
##
```

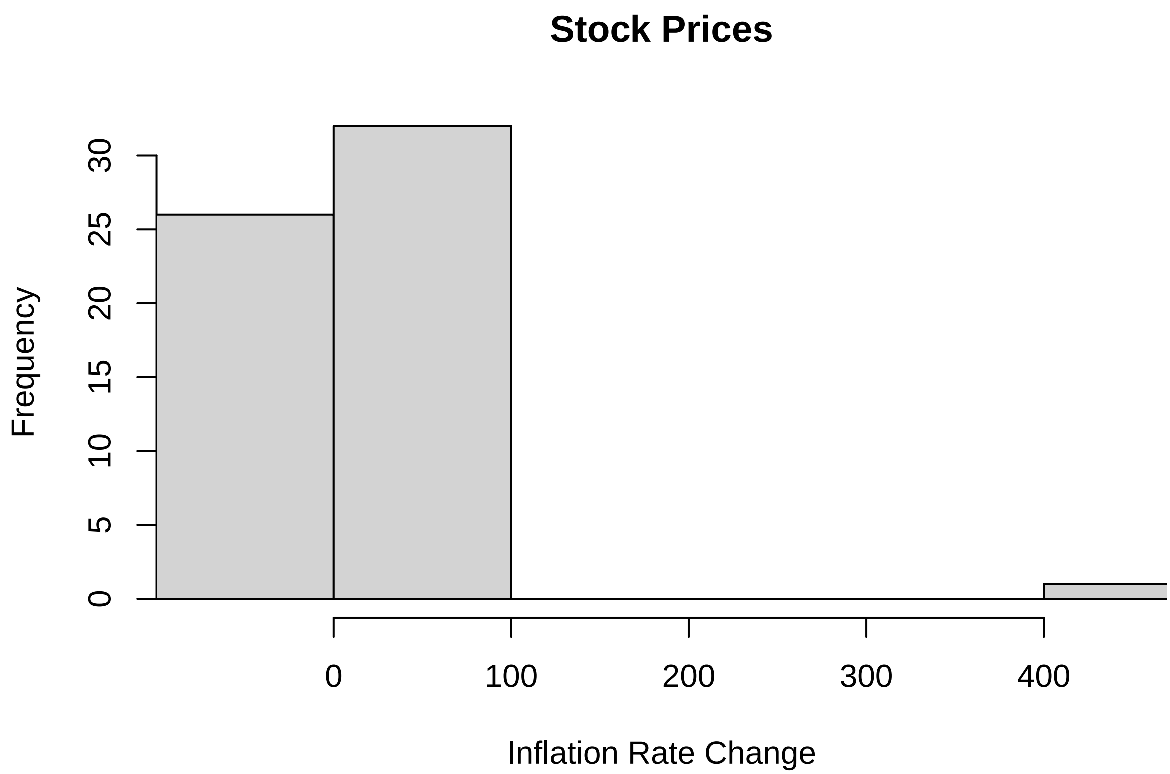
```r
#Histogram Comments
#When looking at the histogram for Home Depot's % Change in Stock Price,
#it seemed to resemble a normal distribution, being unimodal, close to being
#symmetric for the most part, and display of a "bell curve".
#When looking at the histogram for Inflation percent changes, the graph is
#biased towards the left side and is a right skewing distribution, with almost
#no data on the right side.
hist(HDPC, main="Stock Prices", xlab= "Home Depot Percent Change",
     xlim=c(-15.1,18.6))
```



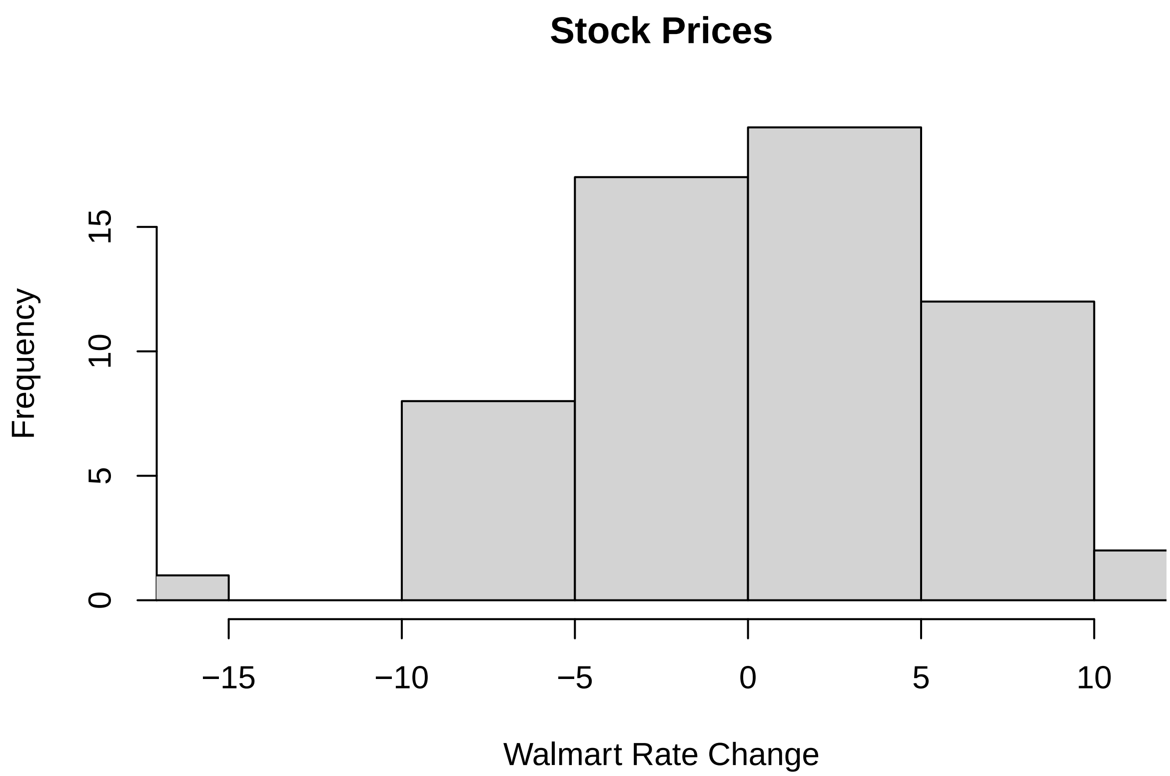

```r
hist(LowesPC, main="Stock Prices", xlab= "Lowes Percent Change",
     xlim=c(-19.3,25.2))
```



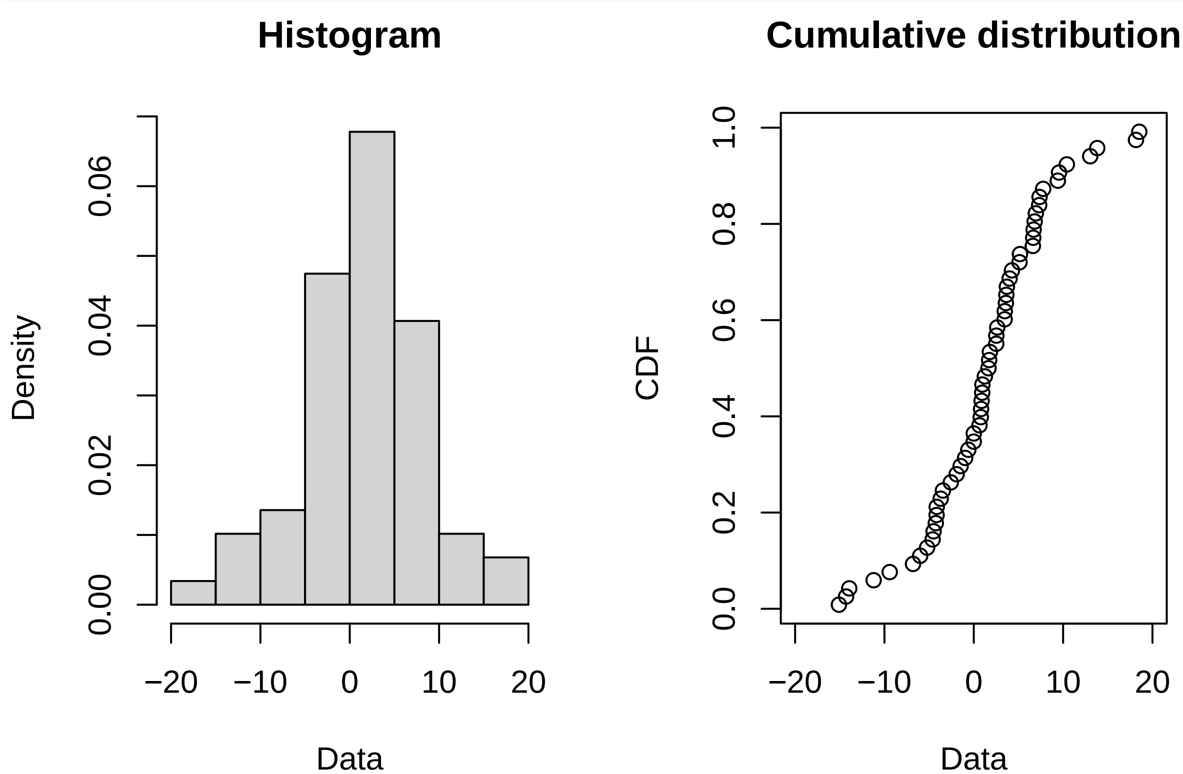

```r
hist(InflationPC, main="Stock Prices", xlab="Inflation Rate Change",
     xlim=c(-78.7,448))
```



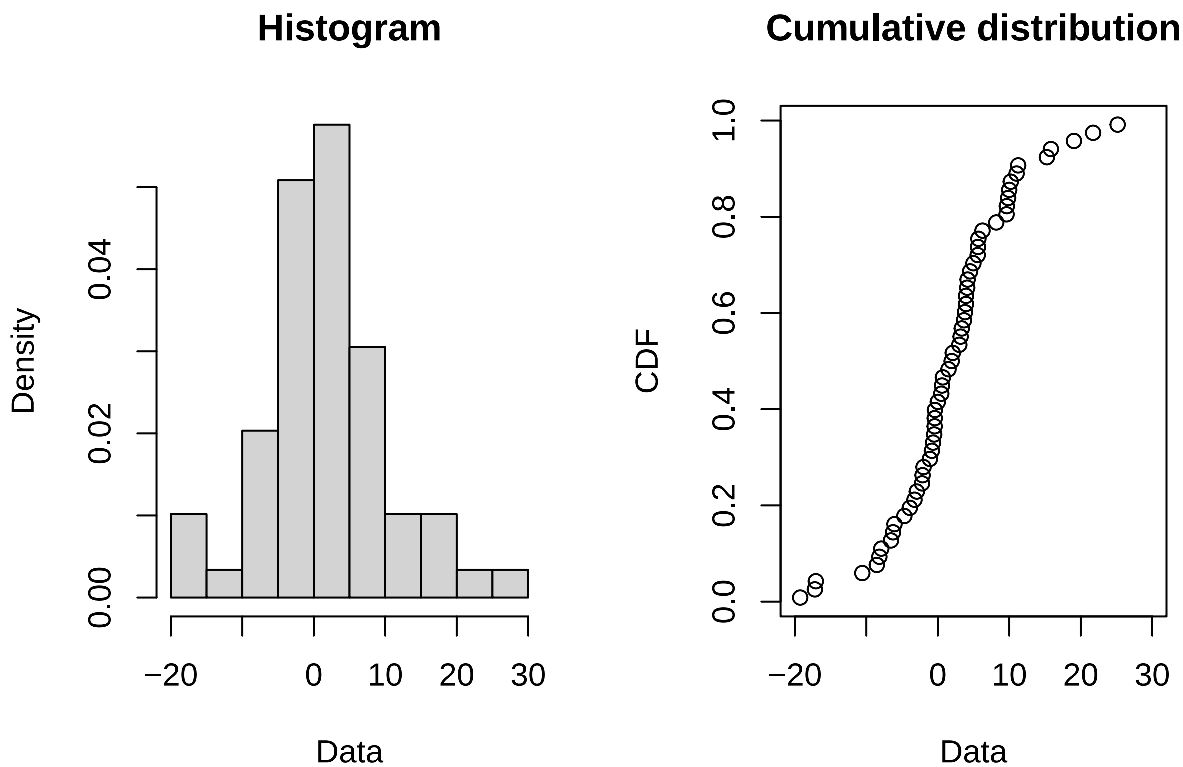

```r
hist(WPC, main="Stock Prices", xlab="Walmart Rate Change", xlim=c(-16,11))
```



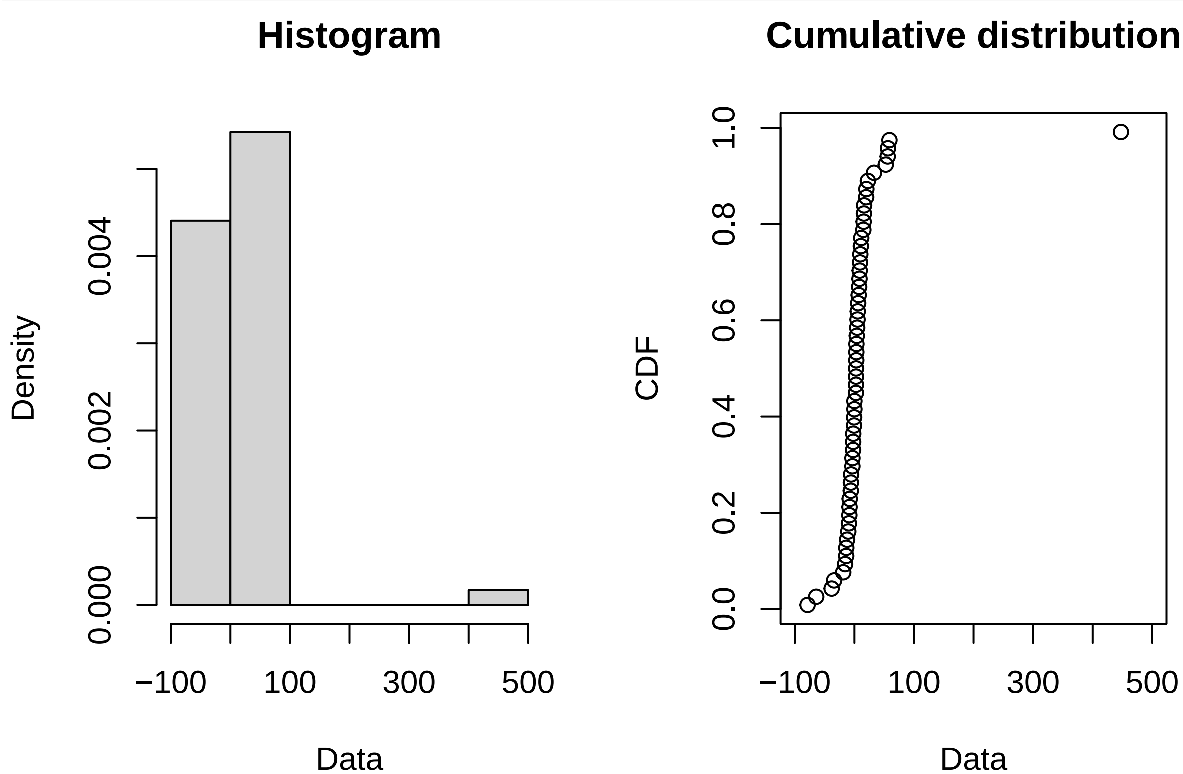

```
library(fitdistrplus)
#Fitted Distributions
plotdist(HDPC, histo= TRUE,)
```



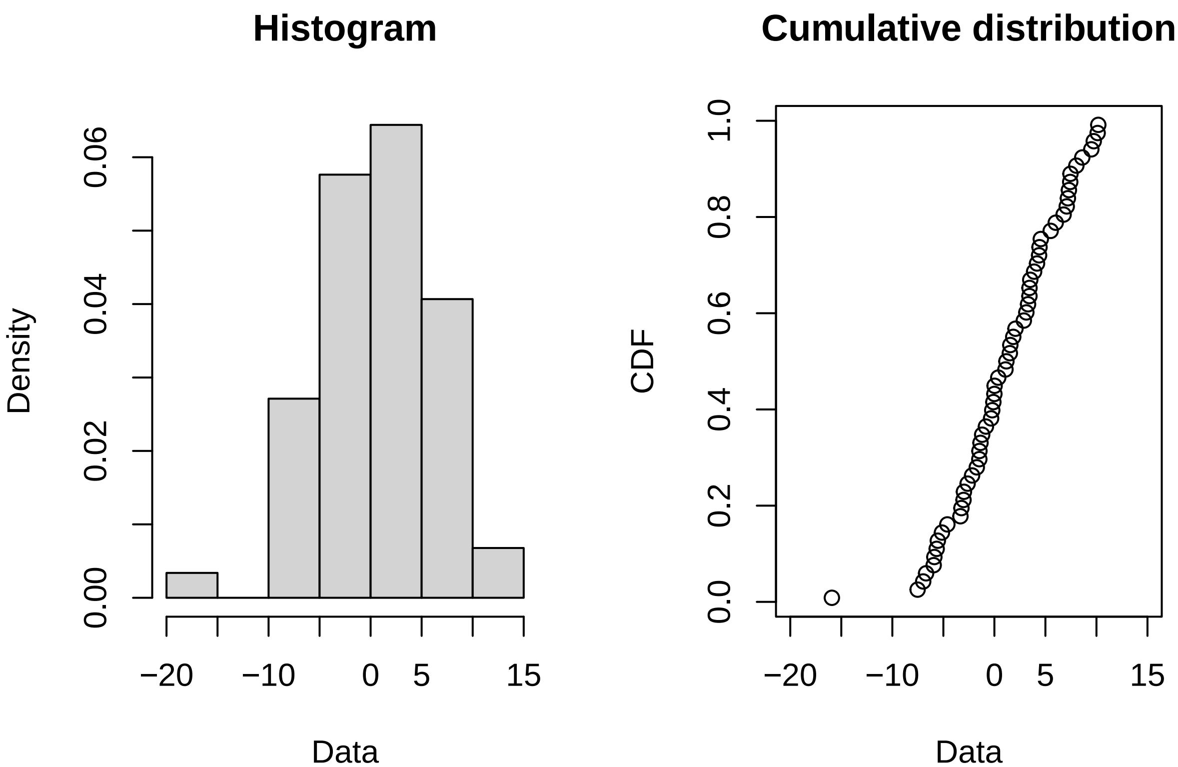
```
plotdist(LowesPC, histo= TRUE,)
```



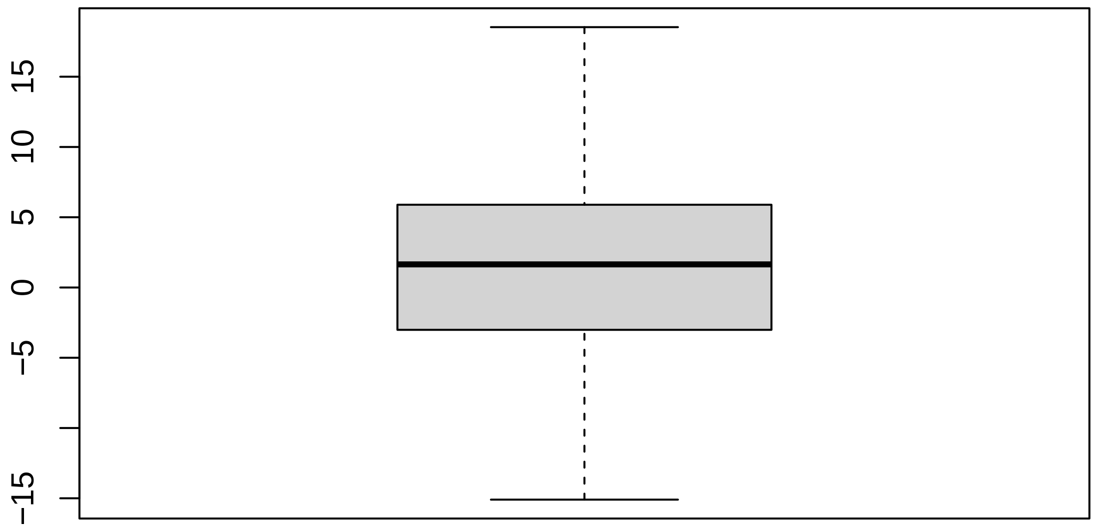

```
plotdist(InflationPC, histo=TRUE,)
```



```r
plotdist(WPC, histo=TRUE,)
```

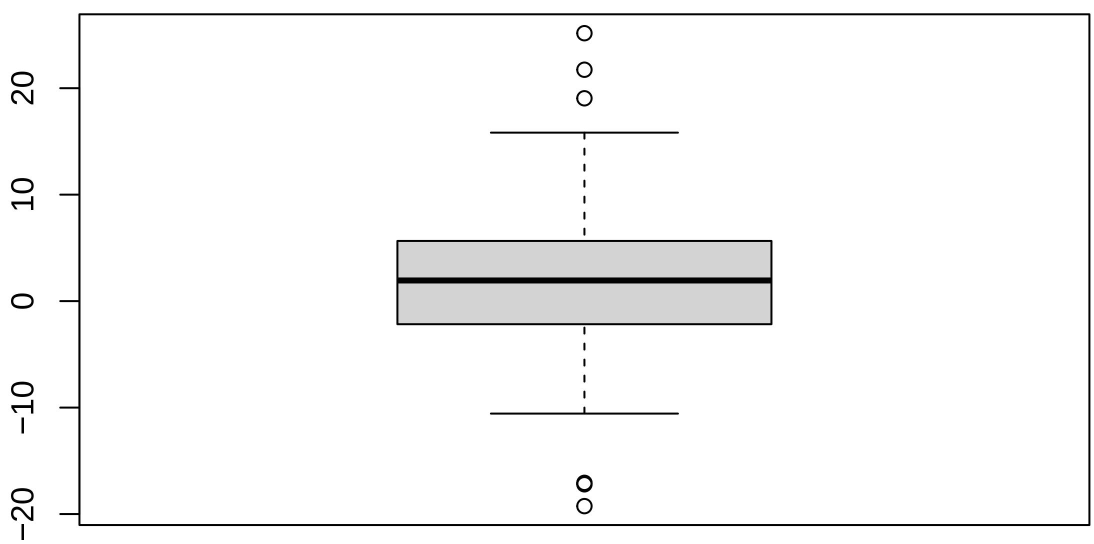



```r
#The boxplot for Inflation %Change was interesting to look at, as there were
#many months that were considered outliers, with the highest rate coming close
#to 450% and at one point going far into the negatives of 78%, compared to the
#mean of 9.892, and median of 2.79%. These outliers were obviously during the
#COVID-19 Pandemic.
#For Home Depot's boxplot, it seems to have a normal distribution,
#with no real outliers in the data.
#The boxplot for Inflation looks very compact with the outlier that
#makes the margins super wide.
#Boxplots
boxplot(HDPC, xlab= "Home Depot % Change")
```



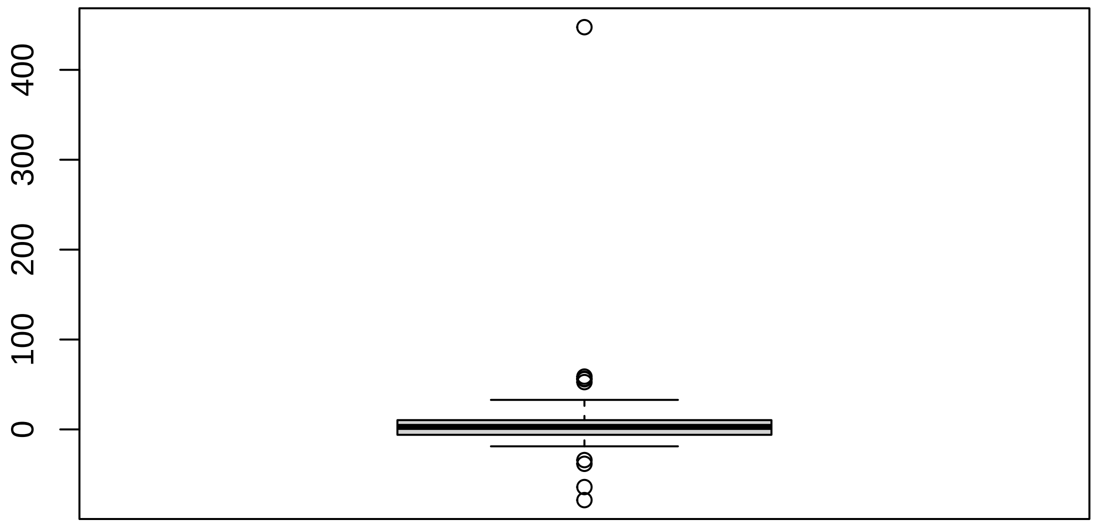
Home Depot % Change

```
boxplot(LowesPC,xlab="Lowes % Change")
```



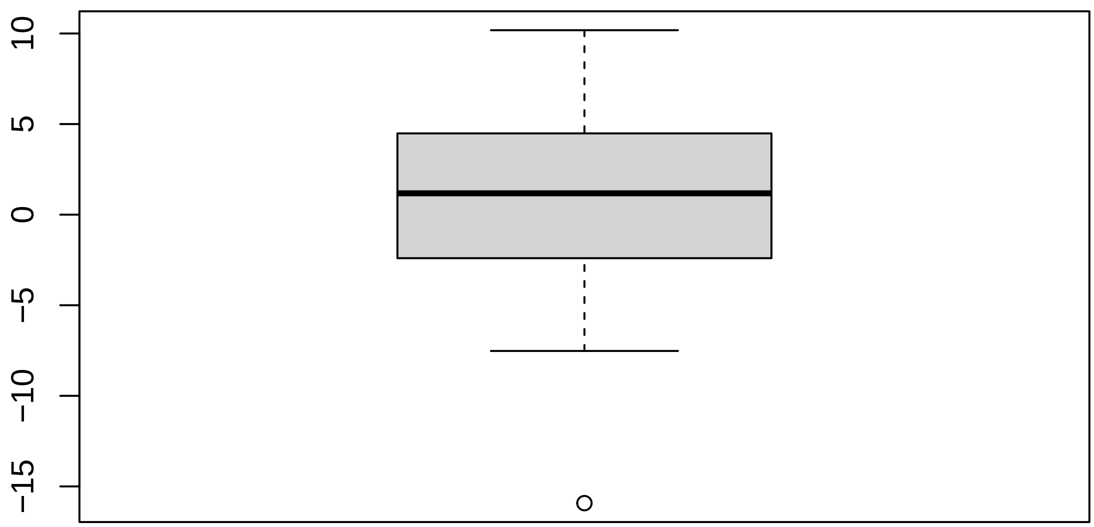
Lowes % Change

```
boxplot(InflationPC, xlab= "Inflation Rate % Change")
```



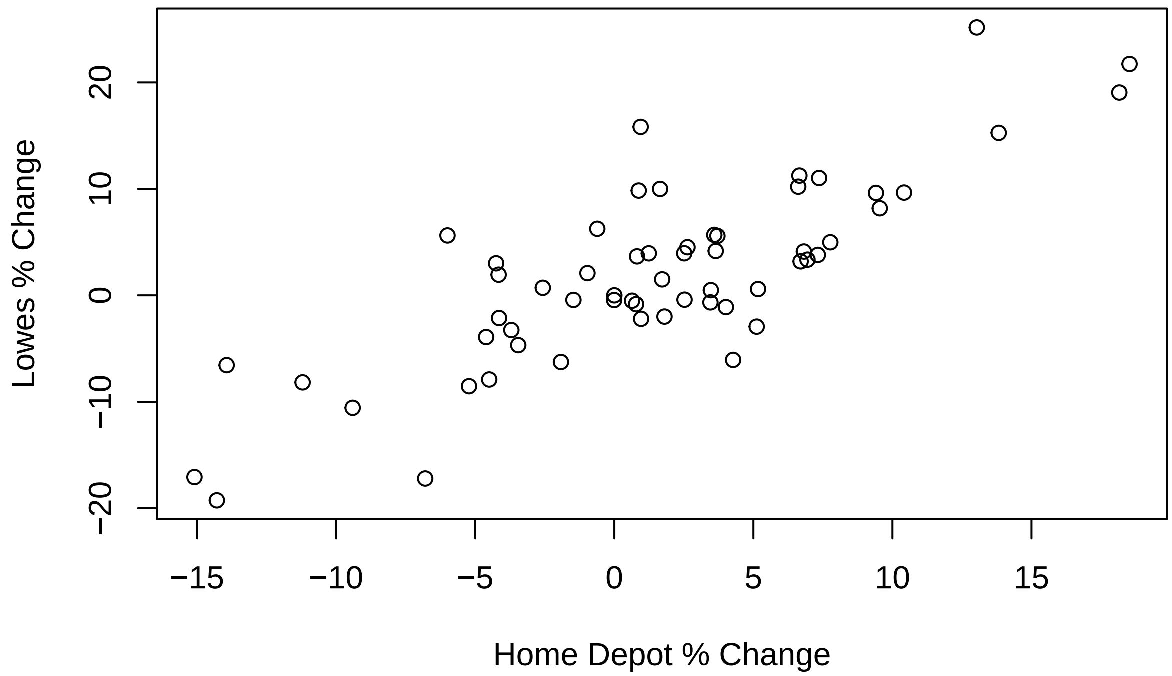
Inflation Rate % Change

```
boxplot(WPC, xlab="Walmart % Change")
```

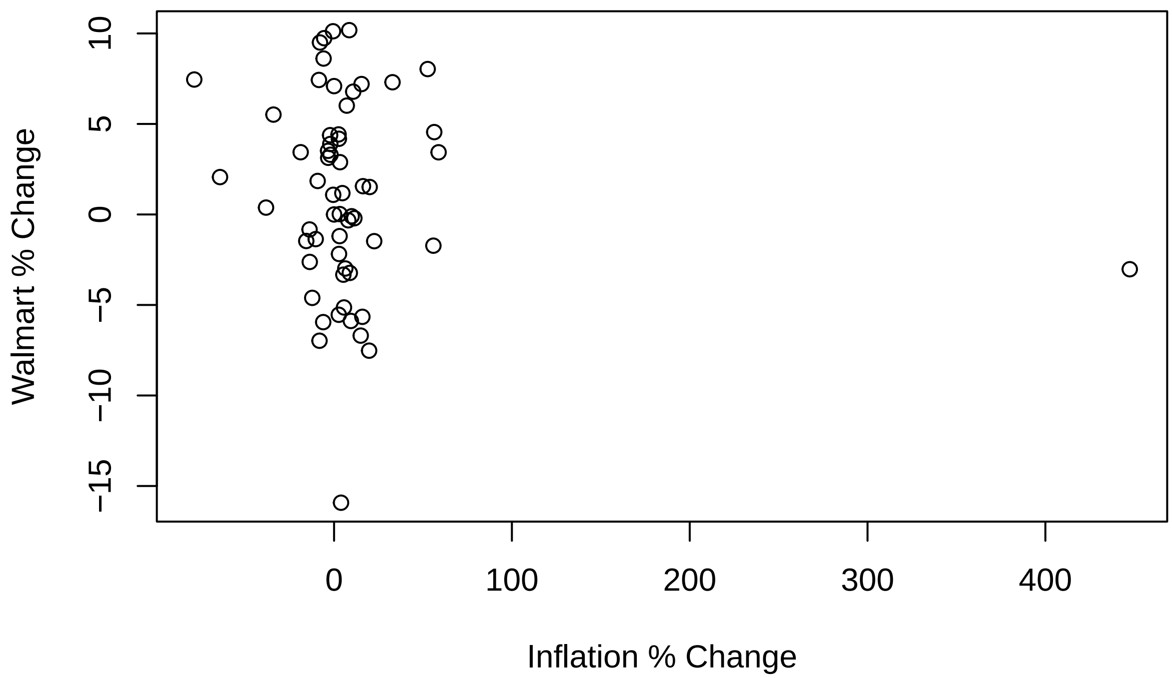


Walmart % Change

```
#Comments: For almost all of the data, the inflation change percentages are all
#under the 100% change, with an outlier being the max % change mentioned earlier
#in the summary.

#For the scatterplot between Home Depot Percent Changes and Lowes, there seems
#to be a positive linear relationship. As the percent change of Home Depot
#increases, there is also an increase in the percent change for Lowes.
#Scatterplot
plot(HDPC, LowesPC, xlab="Home Depot % Change", ylab= "Lowes % Change")
```

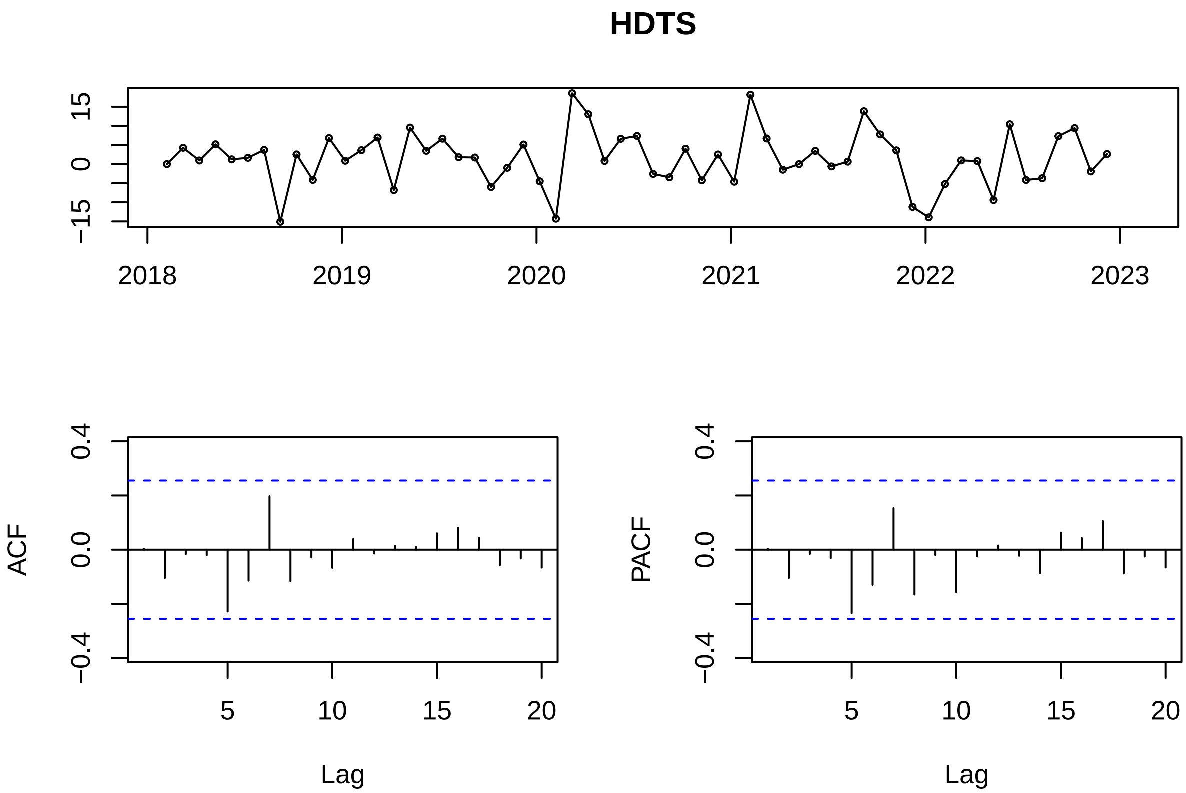



```
plot(InflationPC, WPC, xlab="Inflation % Change", ylab="Walmart % Change")
```

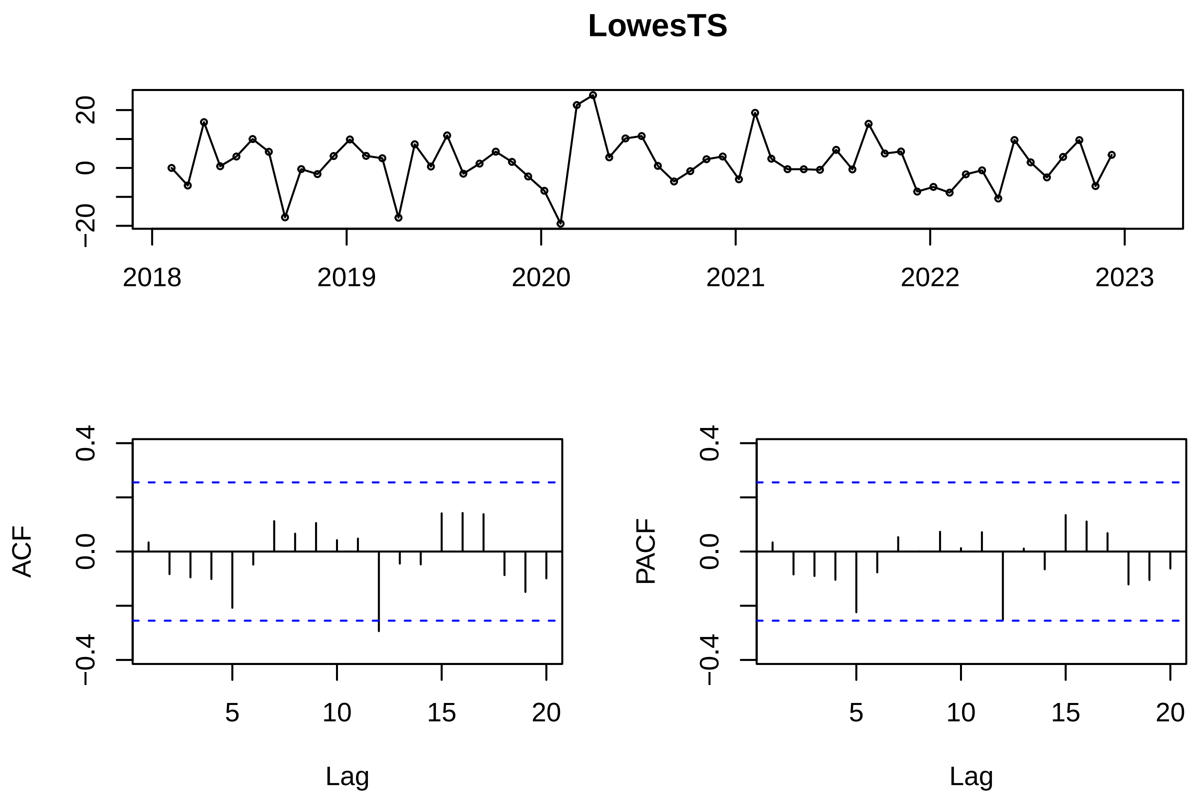



```
#2
HDTS <- ts(HDLD[,4],start=2018.1,freq=12)
LowesTS <- ts(HDLD[,6],start=2018.1, freq=12)
WTS <- ts(HDLD[,2], start=2018.1, freq=12)
InflationTS <- ts(HDLD[,8], start=2018.1, freq=12)


#After looking at the time series for Home Depot, we found that almost all of
#the lags were insignificant, and that the only close one that we would consider
```

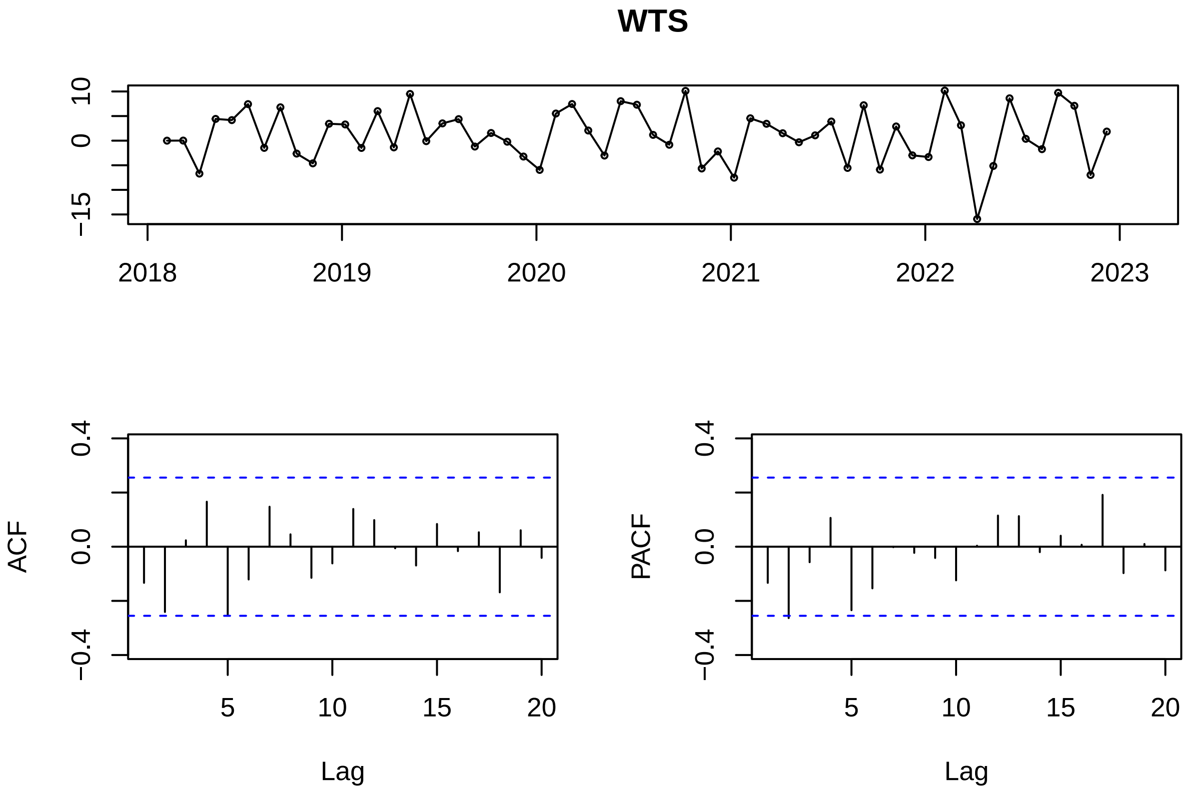
```
#would be Lag 5 in the ACF and PACF model.
tsdisplay(HDTS)
```



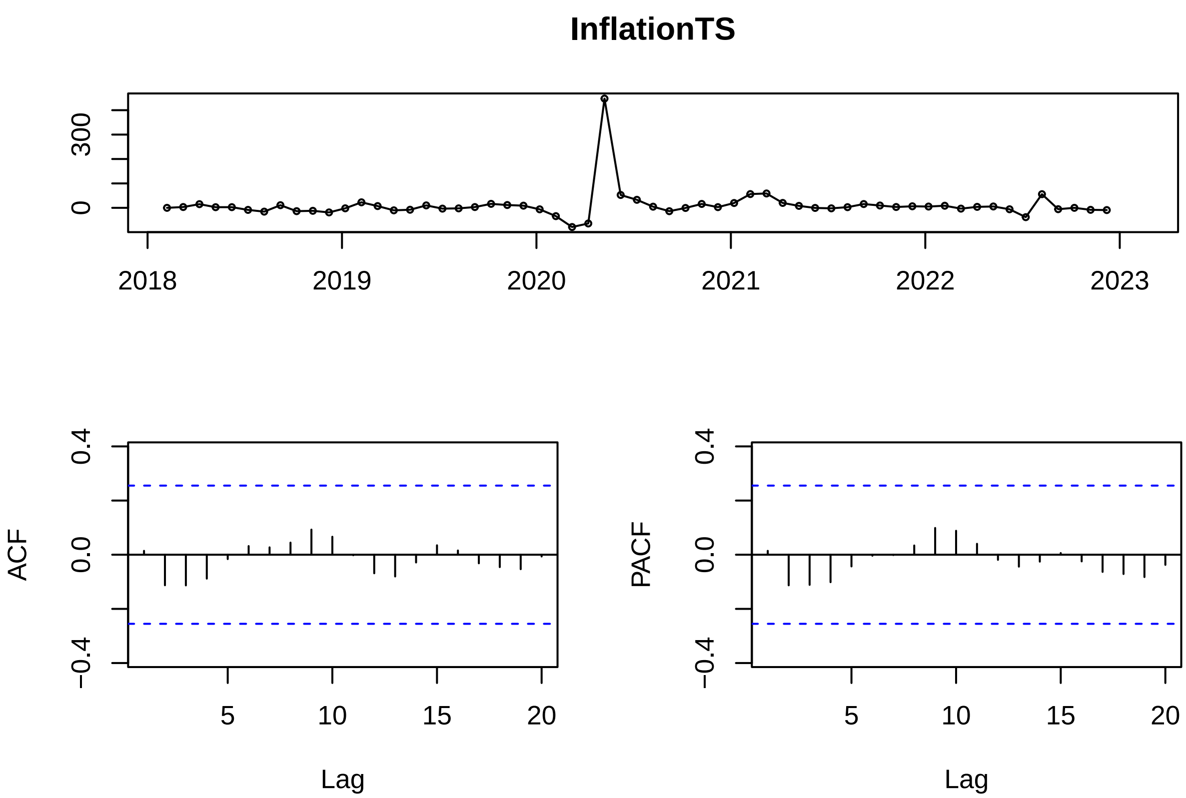

```
#Similarly, for the Lowes time series, the only signifcant lags that we may
#consider, by looking at the ACF and PACF model would be Lag 12.
tsdisplay(LowesTS)
```



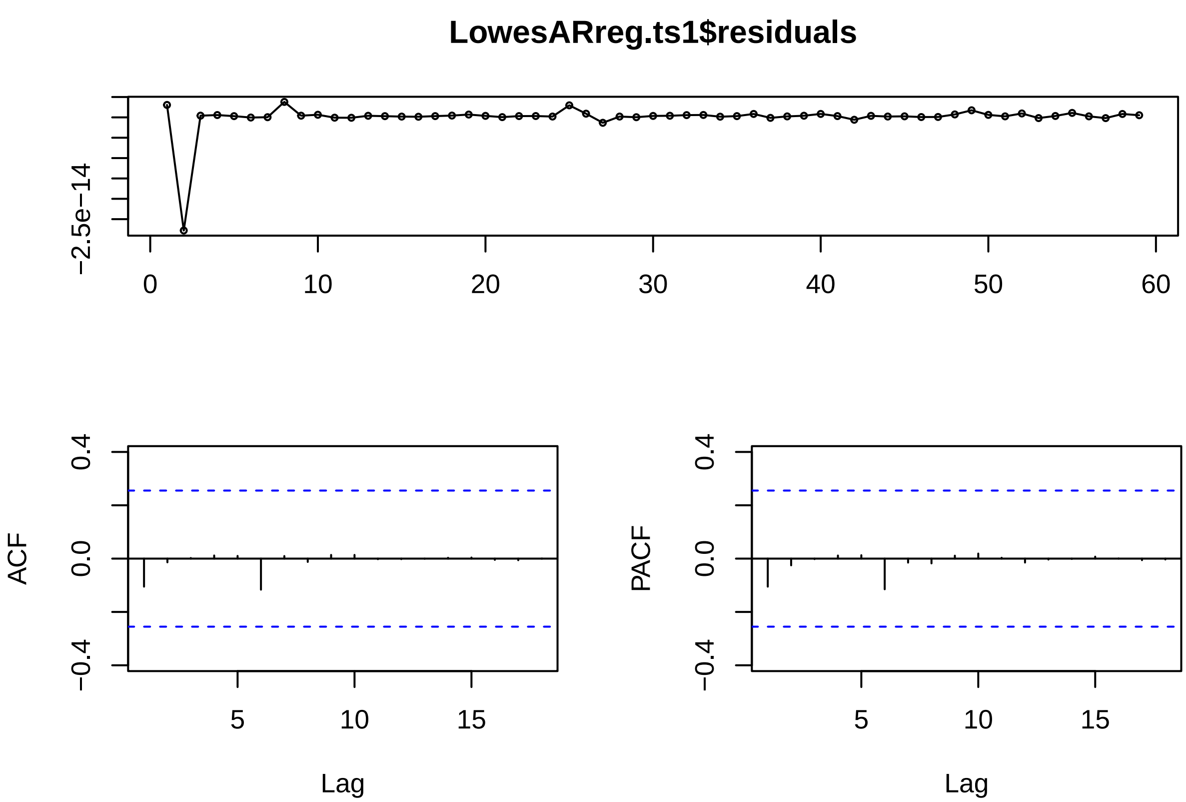

```
#For Walmart's time series, we found that there were only 2 significant lags,
#from the ACF model, and seeing that it would be closer significance for Lag 2,
#in the PACF.
tsdisplay(WTS)
```



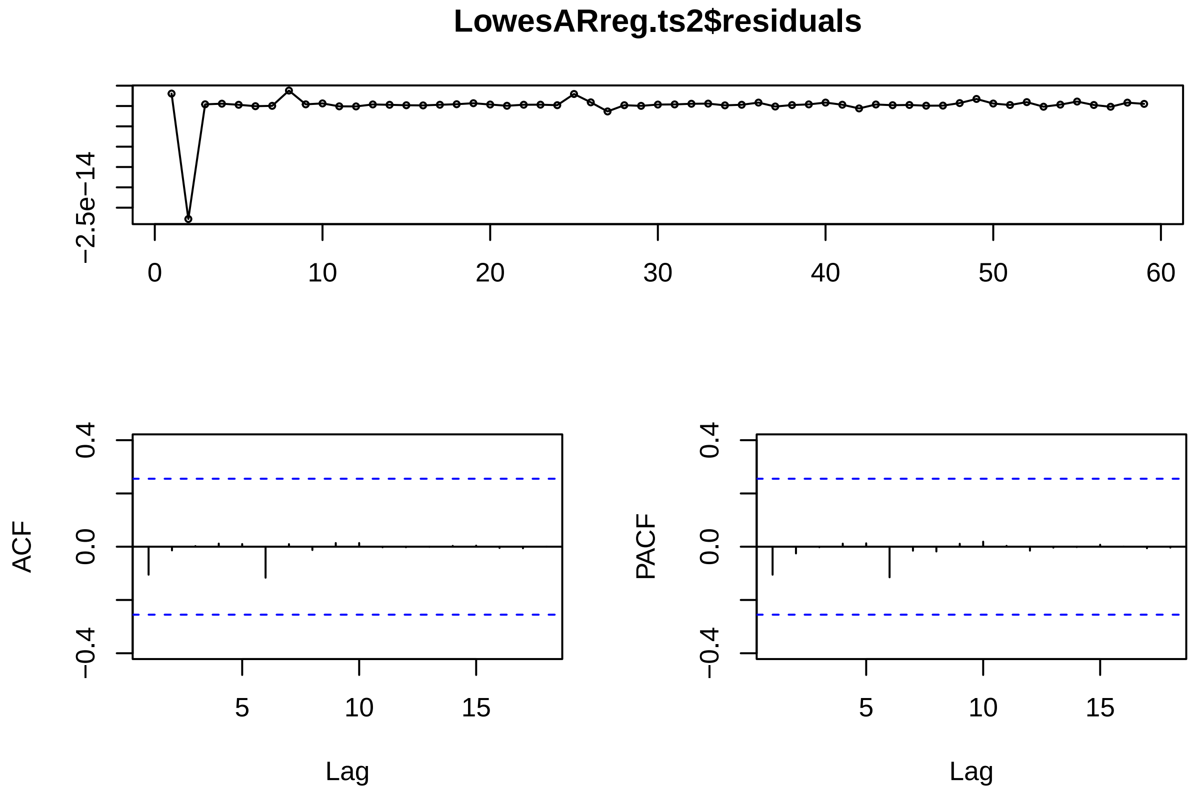

```
#For the Inflation time series, there were no significant lags that were present.
tsdisplay(InflationTS)
```



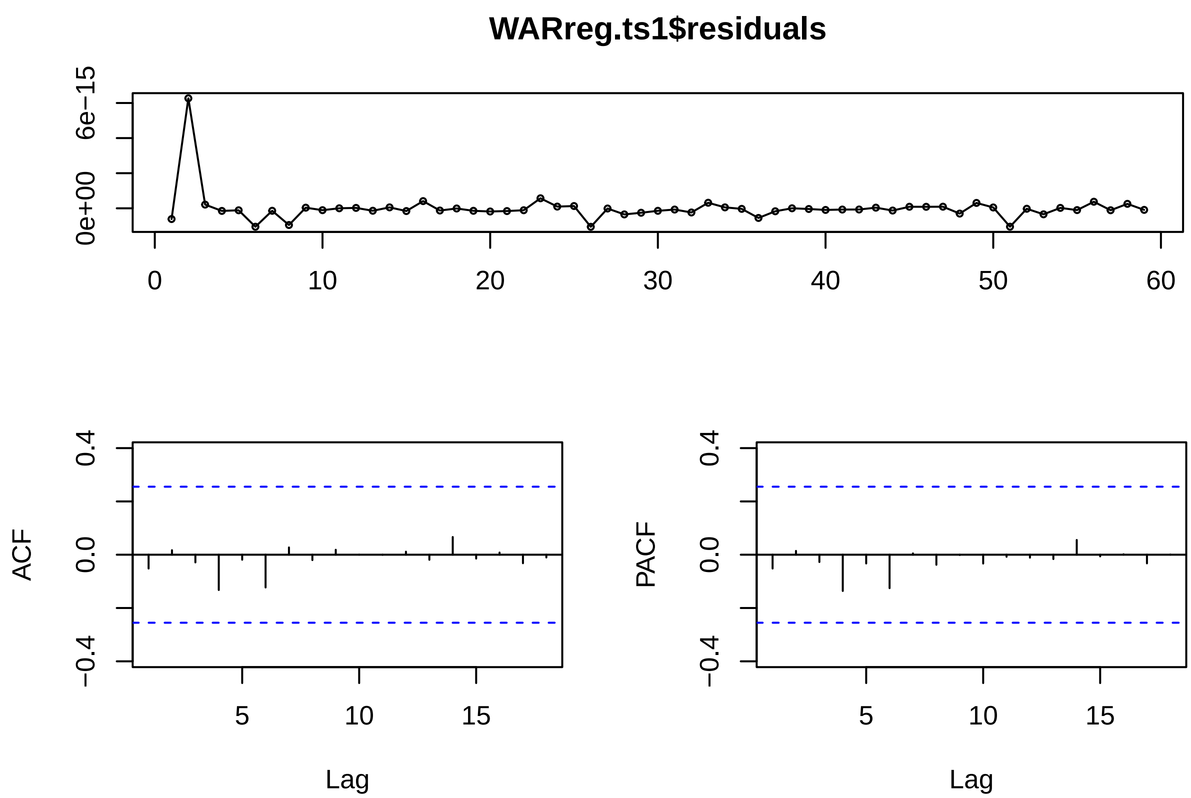
```
#3
#AR Models
#It was very useful to check the residuals in the time series models for both
#our AR models because it helped us check whether our model has really captured
#the data. For our forecasting to be a good model, we were looking for two main
#things: whether the residuals were uncorrelated, and whether they had zero
#mean. In our LowesAR Model #1, the ACF and PACF of the residuals were applied
#to Lowes Percent Change for its stock. By looking at it,the only lags that
#were apparent were @Lag 1 and 6 but all other lags were not really there.
#With, checking the Breusch-Godfrey test for serial correlation, there was a
#lack of correlation with a p-value of 0.9987 at 5% significance level to show
#that the forecast was good and there is little to no information that is
#missing in the residuals which should be used in computing forecasts.
#The same can be said for our Lowes ARModel #2 which only included Lag12.
LowesARreg.ts1 = dynlm(LowesPC~L(LowesPC,12)+L(LowesPC,5))
LowesARreg.ts2 = dynlm(LowesPC~L(LowesPC,12))
WARreg.ts1 = dynlm(WPC~L(WPC,5)+L(WPC,2))
WARreg.ts2 = dynlm(WPC~L(WPC,2))

tsdisplay(LowesARreg.ts1$residuals)
```



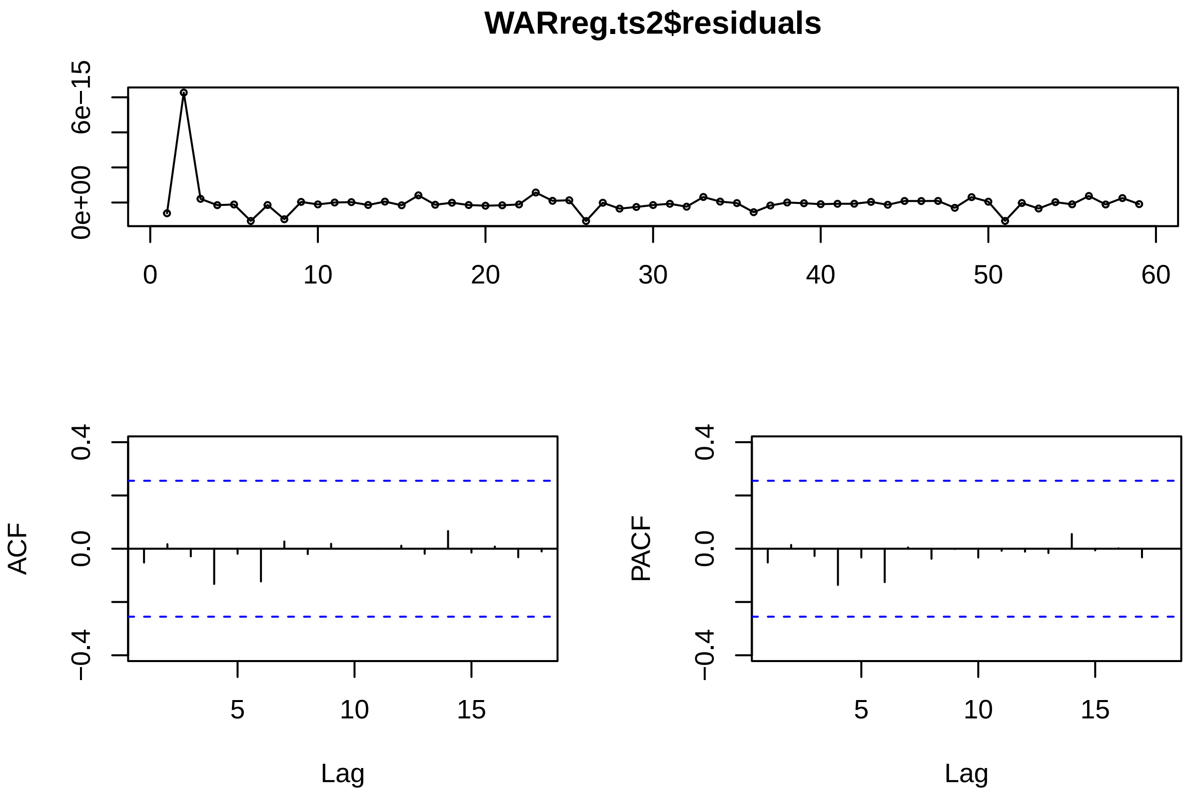

```
tsdisplay(LowesARreg.ts2$residuals)
```




```
tsdisplay(WARreg.ts1$residuals)
```



```
tsdisplay(WARreg.ts2$residuals)
```




```
summary(LowesARreg.ts1)
```

```
## Warning in summary.lm(LowesARreg.ts1): essentially perfect fit: summary may be
## unreliable

##
## Time series regression with "numeric" data:
## Start = 1, End = 59
##
## Call:
## dynlm(formula = LowesPC ~ L(LowesPC, 12) + L(LowesPC, 5))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.778e-14  1.591e-16  3.150e-16  5.805e-16  3.794e-15
##
## Coefficients: (1 not defined because of singularities)
##                  Estimate Std. Error    t value Pr(>|t|)
## (Intercept)    -9.250e-16  5.070e-16 -1.825e+00   0.0733 .
## L(LowesPC, 12)  1.000e+00  5.785e-17  1.729e+16   <2e-16 ***
## L(LowesPC, 5)          NA         NA         NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.793e-15 on 57 degrees of freedom
##   (0 observations deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:      1
```

```
## F-statistic: 2.988e+32 on 1 and 57 DF,  p-value: < 2.2e-16
```

```
summary(LowesARreg.ts2)
```

```
## Warning in summary.lm(LowesARreg.ts2): essentially perfect fit: summary may be
## unreliable
```

```
##
## Time series regression with "numeric" data:
## Start = 1, End = 59
##
## Call:
## dynlm(formula = LowesPC ~ L(LowesPC, 12))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.778e-14  1.591e-16  3.150e-16  5.805e-16  3.794e-15
##
## Coefficients:
##                  Estimate Std. Error    t value Pr(>|t|)
## (Intercept)    -9.250e-16  5.070e-16 -1.825e+00   0.0733 .
## L(LowesPC, 12)  1.000e+00  5.785e-17  1.729e+16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.793e-15 on 57 degrees of freedom
##   (0 observations deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.988e+32 on 1 and 57 DF,  p-value: < 2.2e-16
```

```
summary(WARreg.ts1)
```

```
## Warning in summary.lm(WARreg.ts1): essentially perfect fit: summary may be
## unreliable
```

```
##
## Time series regression with "numeric" data:
## Start = 1, End = 59
##
## Call:
## dynlm(formula = WPC ~ L(WPC, 5) + L(WPC, 2))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.055e-15 -1.533e-16 -7.190e-17  4.980e-17  6.268e-15
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept) 4.625e-16  1.195e-16 3.871e+00 0.000282 ***
## L(WPC, 5)   1.000e+00  2.208e-17 4.528e+16  < 2e-16 ***
## L(WPC, 2)          NA         NA        NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.981e-16 on 57 degrees of freedom
##   (0 observations deleted due to missingness)
```

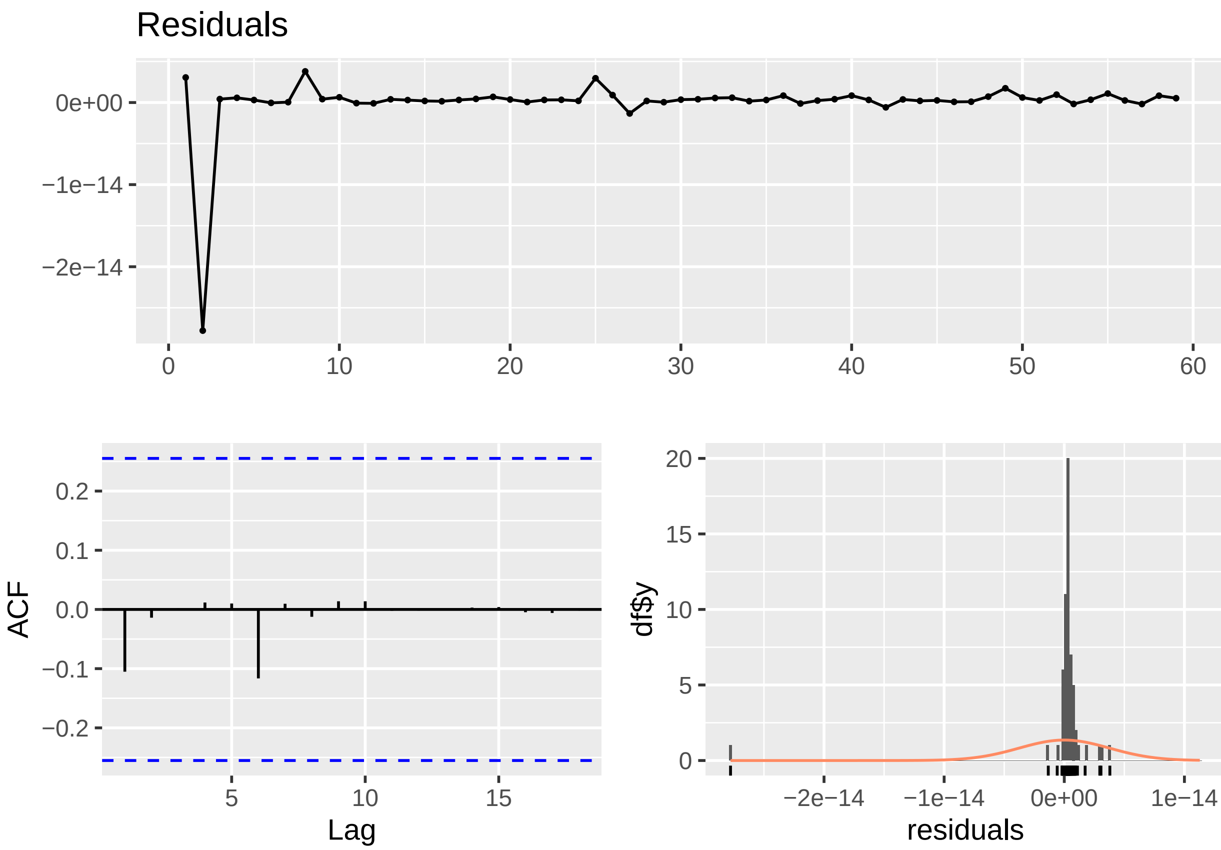
```
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.05e+33 on 1 and 57 DF,  p-value: < 2.2e-16
```

```r
summary(WARreg.ts2)
```

```
## Warning in summary.lm(WARreg.ts2): essentially perfect fit: summary may be
## unreliable
```

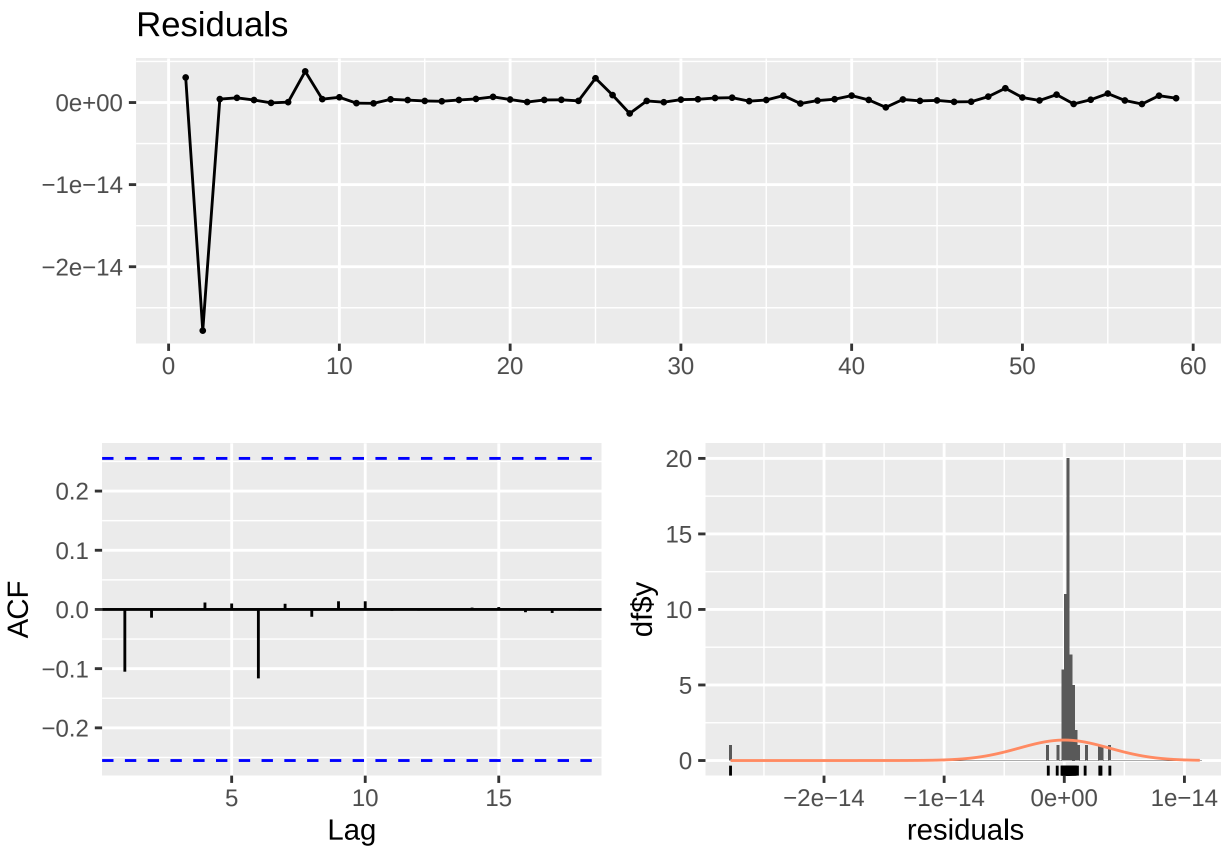
```
##
## Time series regression with "numeric" data:
## Start = 1, End = 59
##
## Call:
## dynlm(formula = WPC ~ L(WPC, 2))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.055e-15 -1.533e-16 -7.190e-17  4.980e-17  6.268e-15
##
## Coefficients:
##             Estimate Std. Error   t value Pr(>|t|)
## (Intercept) 4.625e-16  1.195e-16 3.871e+00 0.000282 ***
## L(WPC, 2)   1.000e+00  2.208e-17 4.528e+16  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.981e-16 on 57 degrees of freedom
##   (0 observations deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.05e+33 on 1 and 57 DF,  p-value: < 2.2e-16
```

```r
checkresiduals(LowesARreg.ts1)
```



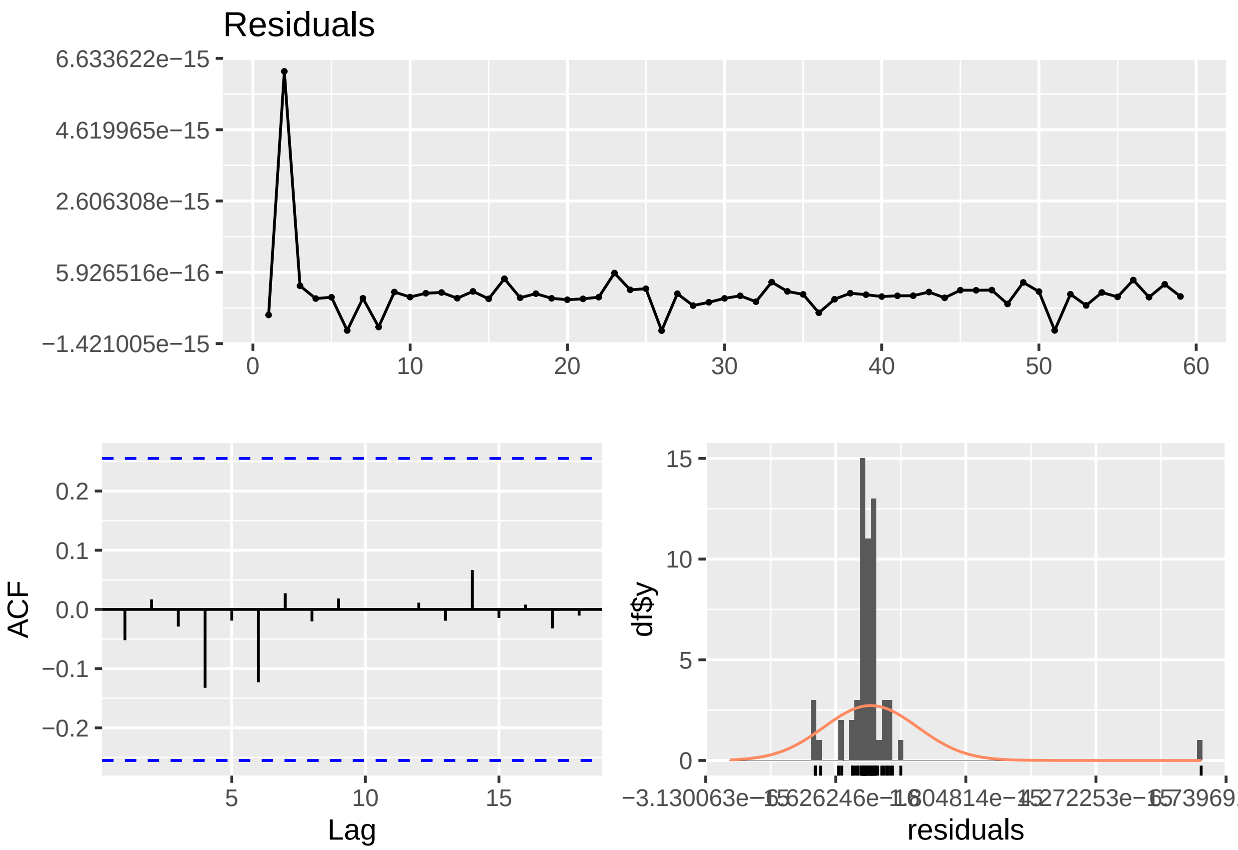

```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 1.5616, df = 10, p-value = 0.9987
```

```
checkresiduals(LowesARreg.ts2)
```



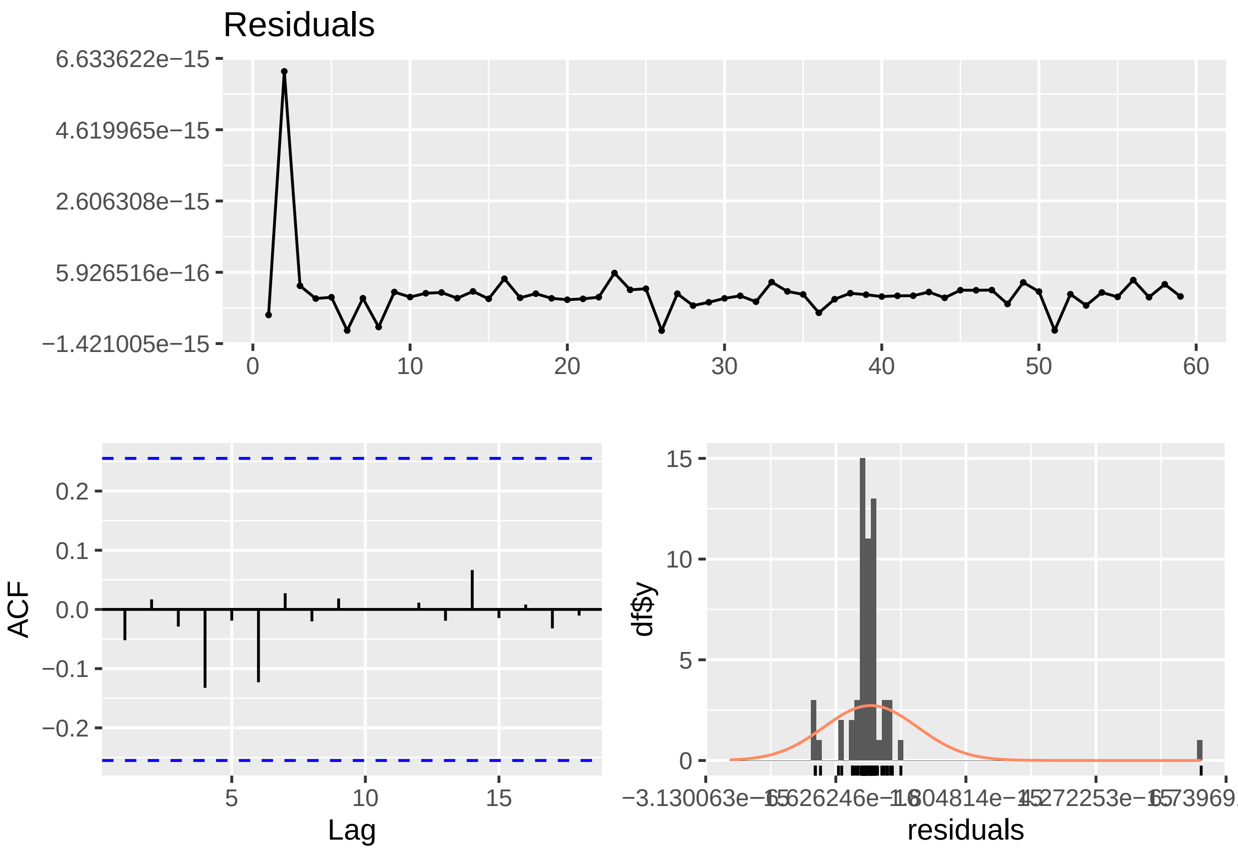

```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 1.5616, df = 10, p-value = 0.9987
```

```
checkresiduals(WARreg.ts1)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 2.4632, df = 10, p-value = 0.9914
```

```
checkresiduals(WARreg.ts2)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 2.4632, df = 10, p-value = 0.9914
```

```
#Training/Testing/MSE
#For the Lowes AR Models that were different in that time series 1 included
#significant lags of 12 and 5 vs time series 2 that only displayed a significant
#lag of 12. Because we did our in monthly periods, our AIC and BIC for the
#models that we had models for had the same values. We are not entirely sure
#that we did this part right or if it has something to do within the monthly
#percent changes, and that it may have been different if we had used daily
#data of these stock percent changes.
#evaluate the training/testing performance by splitting subset
vec_x <- 1:61
train_obs <- sample(vec_x, 61*0.67)
train <- HDLD[train_obs,]
test<- HDLD[-train_obs,]

#make into dataframe for AR model/ test and train
test_df <- as.data.frame(test)
train_df <- as.data.frame(train)

dim(test_df)
```

```
## [1] 21 14
```

```r
dim(train_df)
```

```
## [1] 40 14
```

```r
testLowesARreg.ts1 = dynlm(`Lowes_%Change`~L(`Lowes_%Change`,12)+
                              L(`Lowes_%Change`,5), data = test_df)
trainLowesARreg.ts1 = dynlm(`Lowes_%Change`~L(`Lowes_%Change`,12)+
                              L(`Lowes_%Change`,5), data = train_df)
testLowesARreg.ts2 = dynlm(`Lowes_%Change`~
                              L(`Lowes_%Change`,12), data = test_df)
trainLowesARreg.ts2 = dynlm(`Lowes_%Change`~
                              L(`Lowes_%Change`,12), data = train_df)

sqrt(sum(testLowesARreg.ts1$residuals^2))
```

```
## [1] 2.455074e-15
```

```r
sqrt(sum(trainLowesARreg.ts1$residuals^2))
```

```
## [1] 8.881784e-16
```

```r
sqrt(sum(testLowesARreg.ts2$residuals^2))
```

```
## [1] 2.455074e-15
```

```r
sqrt(sum(trainLowesARreg.ts2$residuals^2))
```

```
## [1] 8.881784e-16
```

```r
testWARreg.ts1 = dynlm(`Walmart_%Change`~ L(`Walmart_%Change`,5)+
                         L(`Walmart_%Change`,2), data=test_df)
trainWARreg.ts1 = dynlm(`Walmart_%Change` ~ L(`Walmart_%Change`,5) +
                          L(`Walmart_%Change`,2), data=train_df)
testWARreg.ts2 = dynlm(`Walmart_%Change`~ L(`Walmart_%Change`,5),
                       data=test_df)
trainWARreg.ts2 = dynlm(`Walmart_%Change` ~ L(`Walmart_%Change`,5),
                        data=train_df)

sqrt(sum(testWARreg.ts1$residuals^2))
```

```
## [1] 3.398897e-15
```

```r
sqrt(sum(trainWARreg.ts1$residuals^2))
```

```
## [1] 6.155796e-15
```

```r
sqrt(sum(testWARreg.ts2$residuals^2))
```

```
## [1] 3.398897e-15
```

```r
sqrt(sum(trainWARreg.ts2$residuals^2))
```

```
## [1] 6.155796e-15
```

```r
#AIC and BIC
AIC(LowesARreg.ts1,LowesARreg.ts2)
```

```
##                df       AIC
## LowesARreg.ts1  3 -3746.848
## LowesARreg.ts2  3 -3746.848
```

```
BIC(LowesARreg.ts1,LowesARreg.ts2)
```

```
##                 df       BIC
## LowesARreg.ts1  3 -3740.615
## LowesARreg.ts2  3 -3740.615
```

```
AIC(WARreg.ts1,WARreg.ts2)
```

```
##             df       AIC
## WARreg.ts1  3 -3916.856
## WARreg.ts2  3 -3916.856
```

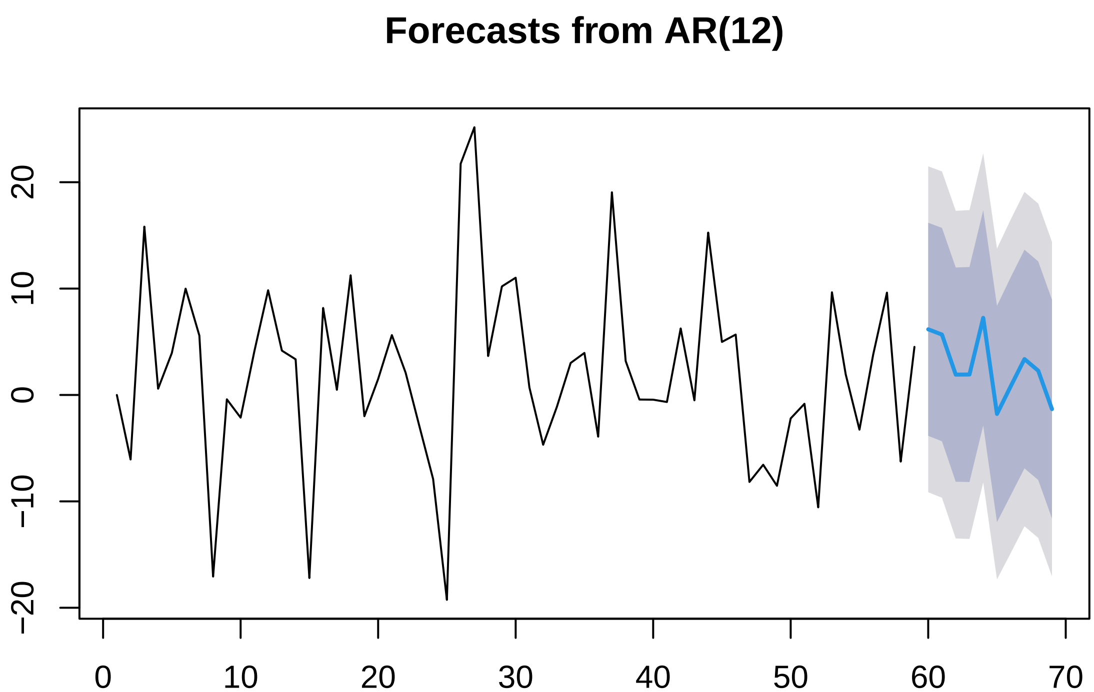
```
BIC(WARreg.ts1,WARreg.ts2)
```

```
##             df       BIC
## WARreg.ts1  3 -3910.623
## WARreg.ts2  3 -3910.623
```

```
#10-step-ahead forecast
Lowes.TS = LowesTS[-c(60,61),]
ARLowesForecast1 = ar(Lowes.TS, aic=FALSE, order.max=12, method="ols")
forecast(ARLowesForecast1,10)
```

```
##    Point Forecast      Lo 80     Hi 80      Lo 95    Hi 95
## 60      6.1728916  -3.841982 16.187765  -9.143536 21.48932
## 61      5.6768796  -4.351763 15.705522  -9.660605 21.01436
## 62      1.9129652  -8.156082 11.982013 -13.486314 17.31224
## 63      1.9223360  -8.183362 12.028034 -13.532995 17.37767
## 64      7.2483835  -2.863967 17.360734  -8.217123 22.71389
## 65     -1.7862790 -11.951463  8.378905 -17.332586 13.76003
## 66      0.8179653  -9.432655 11.068585 -14.859005 16.49494
## 67      3.3756336  -6.898426 13.649693 -12.337184 19.08845
## 68      2.2796796  -7.994573 12.553933 -13.433434 17.99279
## 69     -1.3290752 -11.606667  8.948517 -17.047296 14.38915
```

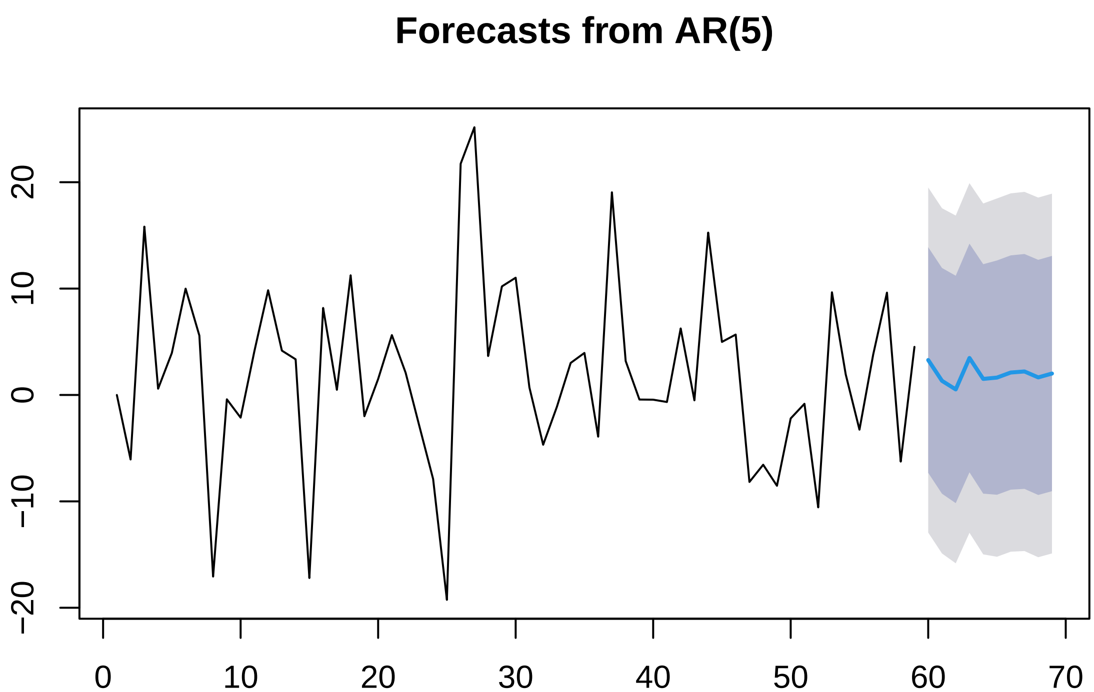
```
plot(forecast(ARLowesForecast1,10))
```




```
ARLowesForecast2 = ar(Lowes.TS, aic=FALSE, order.max=5, method="ols")
forecast(ARLowesForecast2,10)
```

```
##    Point Forecast       Lo 80    Hi 80     Lo 95    Hi 95
## 60      3.2854055  -7.318663 13.88947 -12.93212 19.50293
## 61      1.3300231  -9.279147 11.93919 -14.89530 17.55535
## 62      0.5211109 -10.164927 11.20715 -15.82177 16.86400
## 63      3.4773121  -7.267046 14.22167 -12.95477 19.90939
## 64      1.5080789  -9.274319 12.29048 -14.98218 17.99833
## 65      1.6327767  -9.376416 12.64197 -15.20433 18.46988
## 66      2.1097374  -8.901185 13.12066 -14.73002 18.94949
## 67      2.2101984  -8.827088 13.24748 -14.66987 19.09027
## 68      1.6514556  -9.399931 12.70284 -15.25018 18.55309
## 69      2.0139373  -9.042104 13.06998 -14.89482 18.92269
```

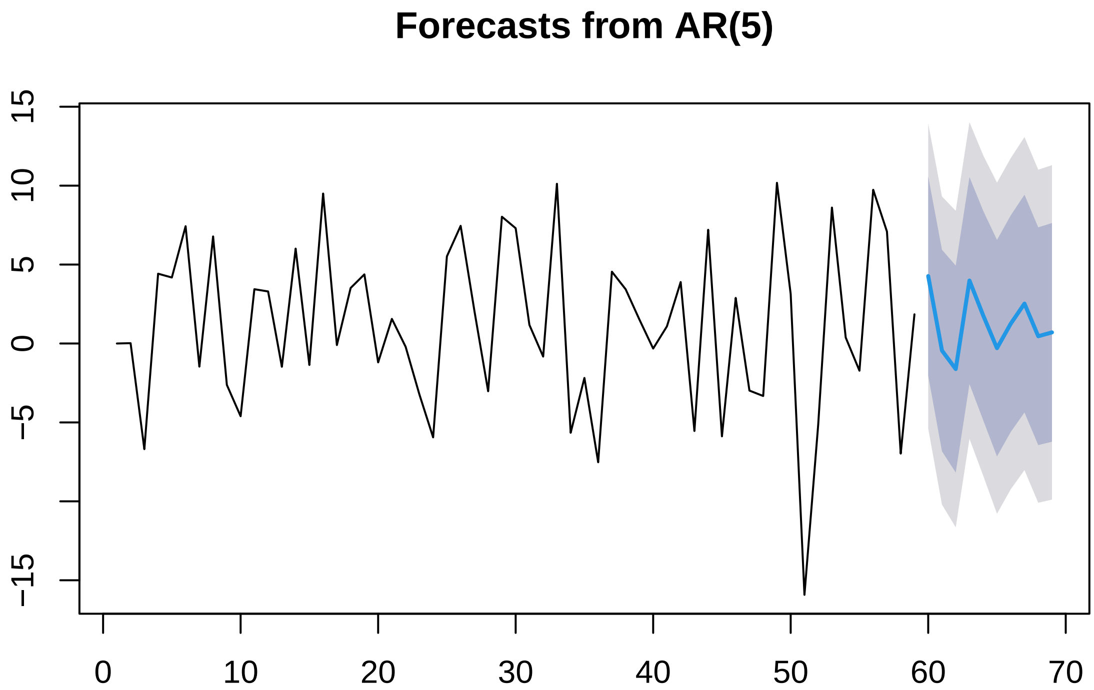
```
plot(forecast(ARLowesForecast2,10))
```




```
W.TS = WTS[-c(60,61),]
ARWalmartForecast1 = ar(W.TS, aic=FALSE, order.max=5, method="ols")
forecast(ARWalmartForecast1,10)
```

```
##    Point Forecast     Lo 80     Hi 80      Lo 95     Hi 95
## 60      4.2711423 -2.045089 10.587374  -5.388700 13.930984
## 61     -0.4466853 -6.830142  5.936772 -10.209340  9.315970
## 62     -1.6194638 -8.175515  4.936588 -11.646079  8.407152
## 63      3.9882966 -2.567890 10.544483  -6.038526 14.015119
## 64      1.7702584 -4.864130  8.404647  -8.376163 11.916680
## 65     -0.2978193 -7.155415  6.559777 -10.785607 10.189969
## 66      1.2529268 -5.605458  8.111311  -9.236067 11.741921
## 67      2.5276242 -4.370082  9.425330  -8.021507 13.076755
## 68      0.4592741 -6.439189  7.357737 -10.091014 11.009562
## 69      0.7042643 -6.222652  7.631181  -9.889541 11.298069
```

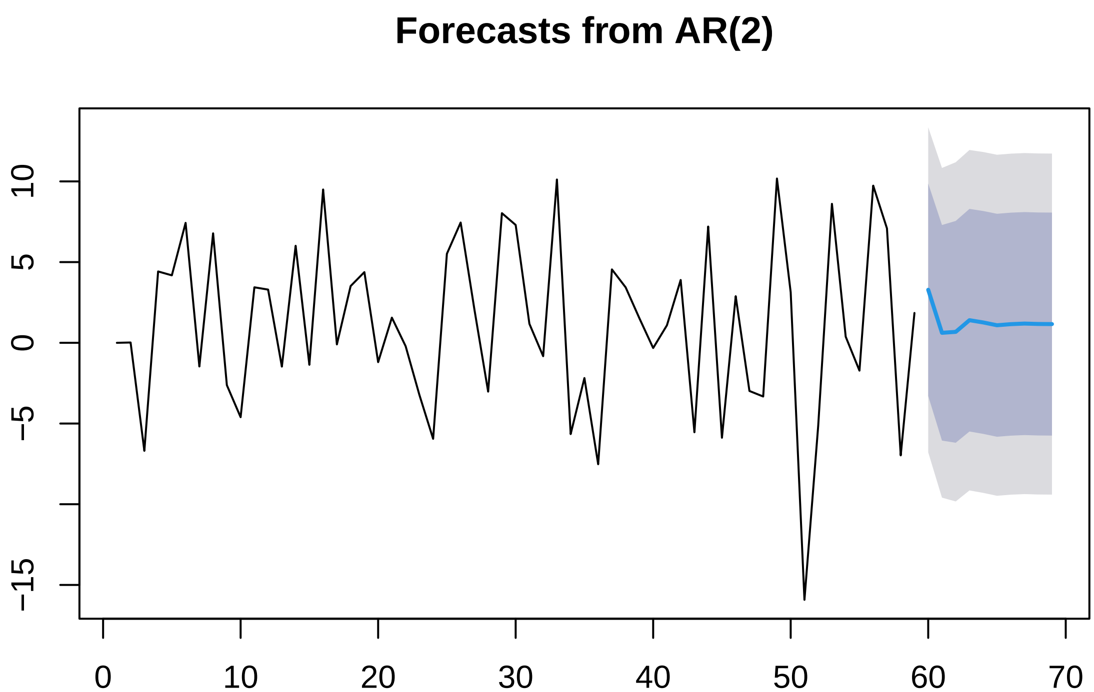
```
plot(forecast(ARWalmartForecast1,10))
```



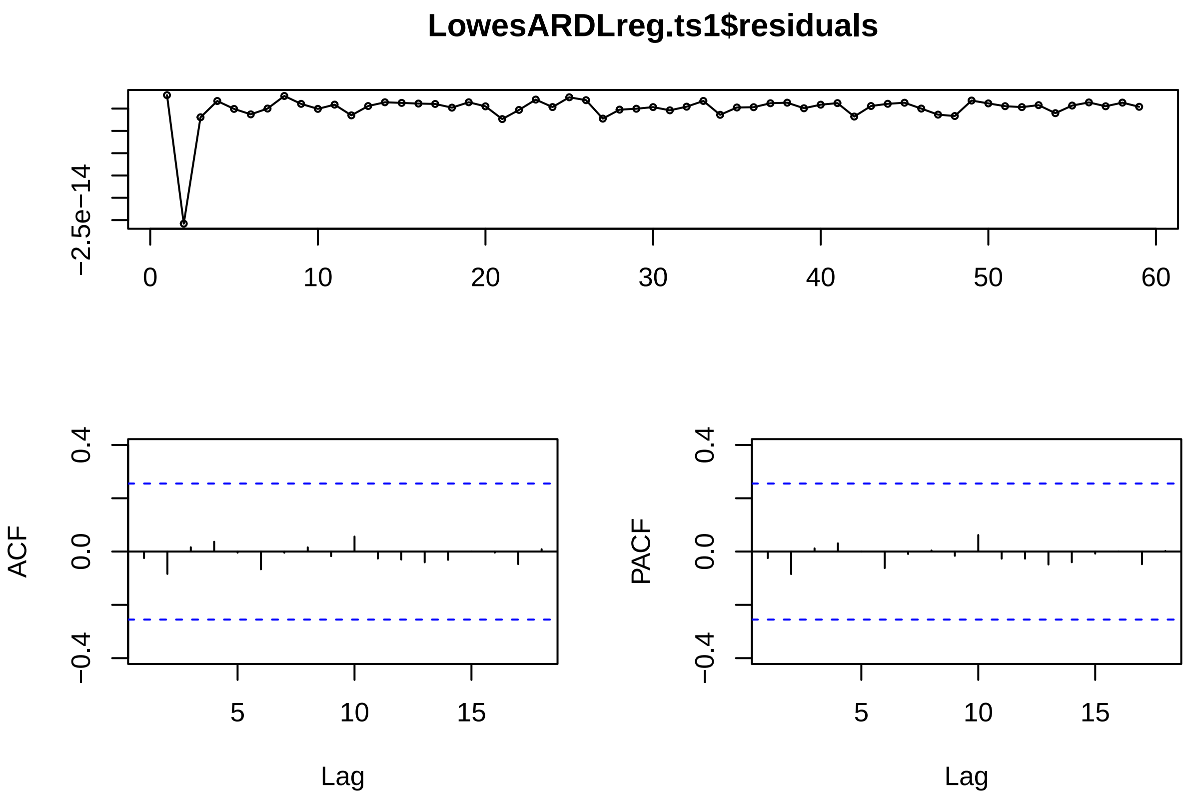

```
ARWalmartForecast2 = ar(W.TS, aic=FALSE, order.max=2, method="ols")
forecast(ARWalmartForecast2,10)
```

```
##    Point Forecast      Lo 80    Hi 80     Lo 95    Hi 95
## 60      3.2861833 -3.296765 9.869131 -6.781567 13.35393
## 61      0.6199794 -6.057576 7.297535 -9.592461 10.83242
## 62      0.6793117 -6.191045 7.549668 -9.827992 11.18662
## 63      1.4009842 -5.494013 8.295982 -9.144005 11.94597
## 64      1.2619098 -5.641688 8.165508 -9.296232 11.82005
## 65      1.0875022 -5.819551 7.994556 -9.475925 11.65093
## 66      1.1553475 -5.751944 8.062639 -9.408444 11.71914
## 67      1.1916719 -5.715972 8.099316 -9.372657 11.75600
## 68      1.1668706 -5.740774 8.074515 -9.397460 11.73120
## 69      1.1611208 -5.746552 8.068794 -9.403254 11.72550
```

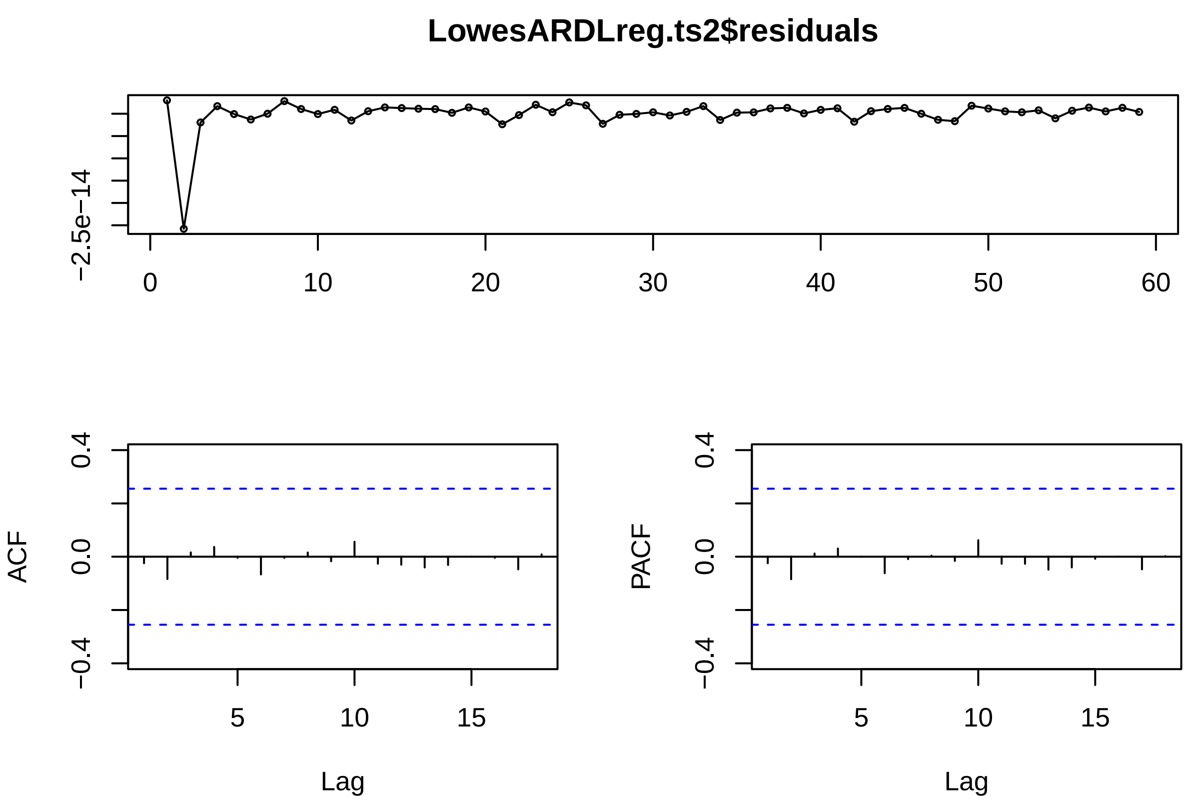
```
plot(forecast(ARWalmartForecast2,10))
```



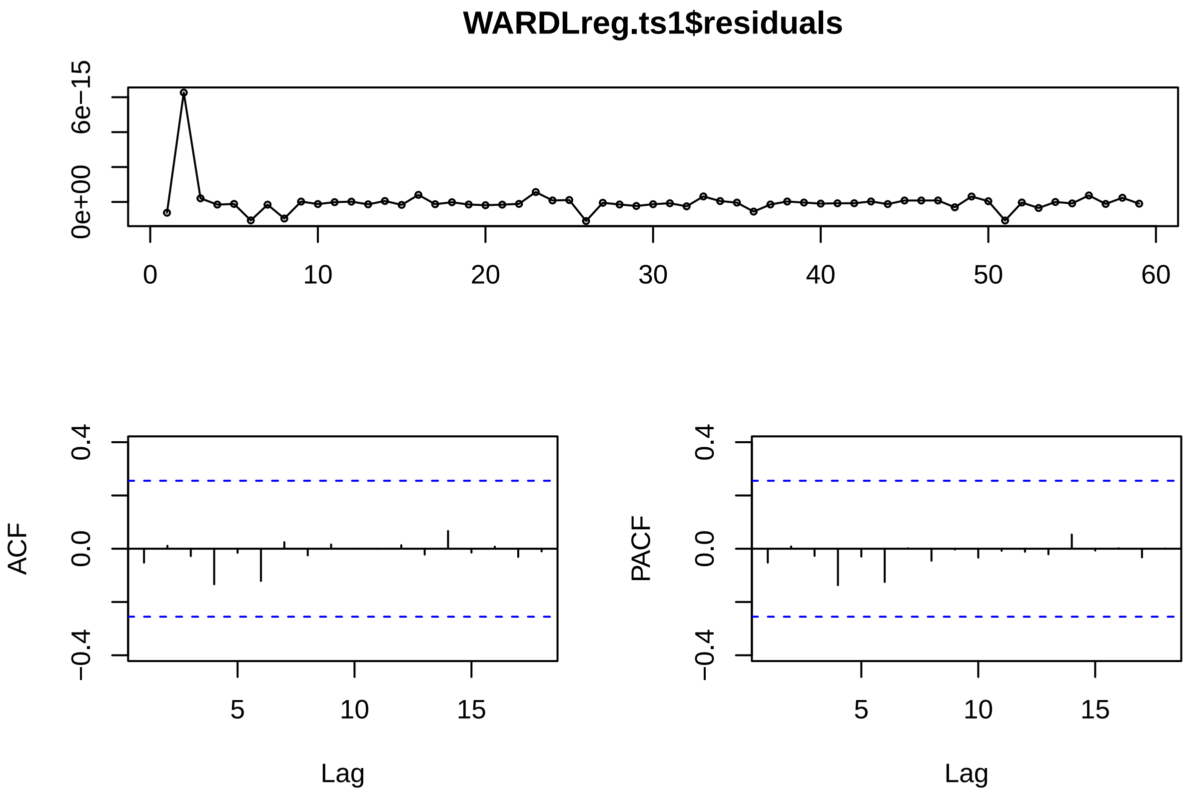

```
#4
#ARDL Models
#When running our ARDL model for Lowes residuals with the addition of
#Home Depot, we were able to find that the same result of the p-value when
#testing serial correlation was very close to 1, and is greater than the 5%
#significance level, meaning we cannot conclude that there is correlation
#between the two, which means that our forecast is good.

#The same can be said for the ARDL models for our Walmart model, as
#our p-value was 0.9915, greater than the 5% significance level.
#The only ones that remotely are close are @lags4 and 6 and those aren't
#really close.
LowesARDLreg.ts1 = dynlm(LowesPC~L(LowesPC,12)+L(LowesPC,5)+L(HDPC,5))
LowesARDLreg.ts2 = dynlm(LowesPC~L(LowesPC,12)+L(HDPC,5))
WARDLreg.ts1 = dynlm(WPC~L(WPC,5)+L(WPC,2)+L(InflationPC,3))
WARDLreg.ts2 = dynlm(WPC~L(WPC,2)+L(InflationPC,3))

tsdisplay(LowesARDLreg.ts1$residuals)
```



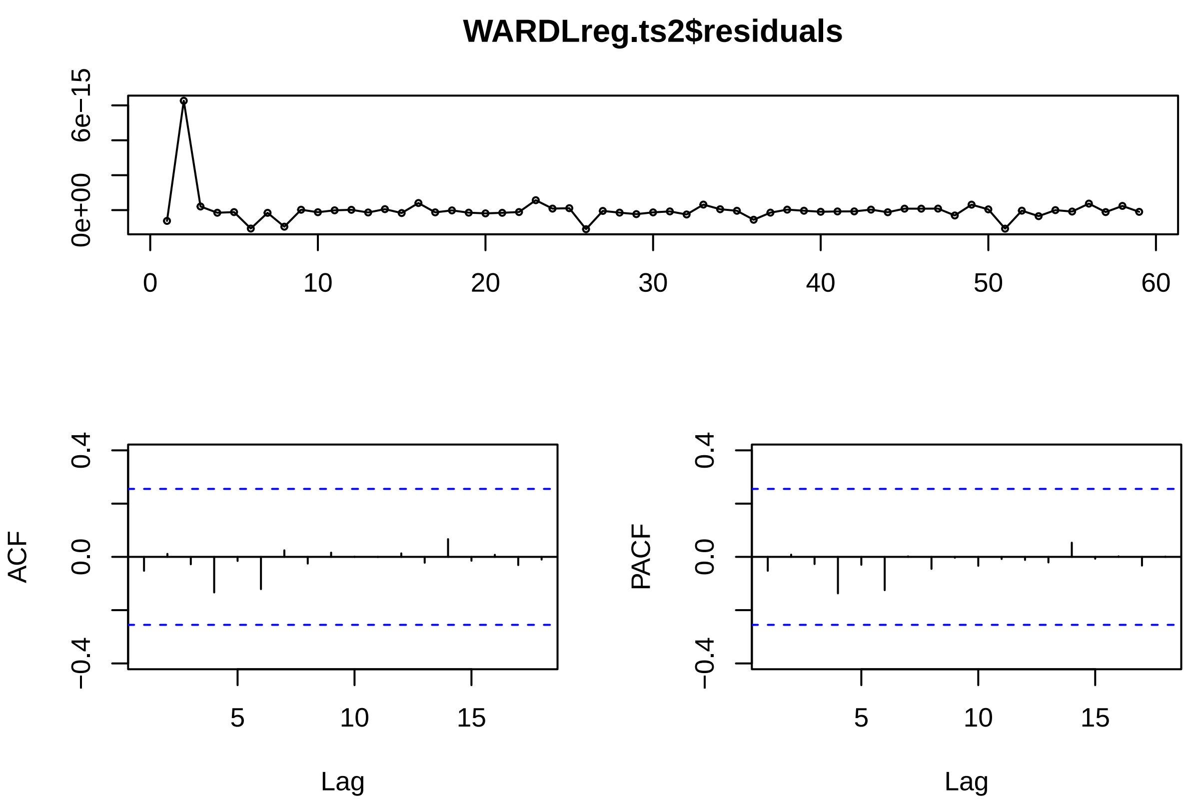

```
tsdisplay(LowesARDLreg.ts2$residuals)
```



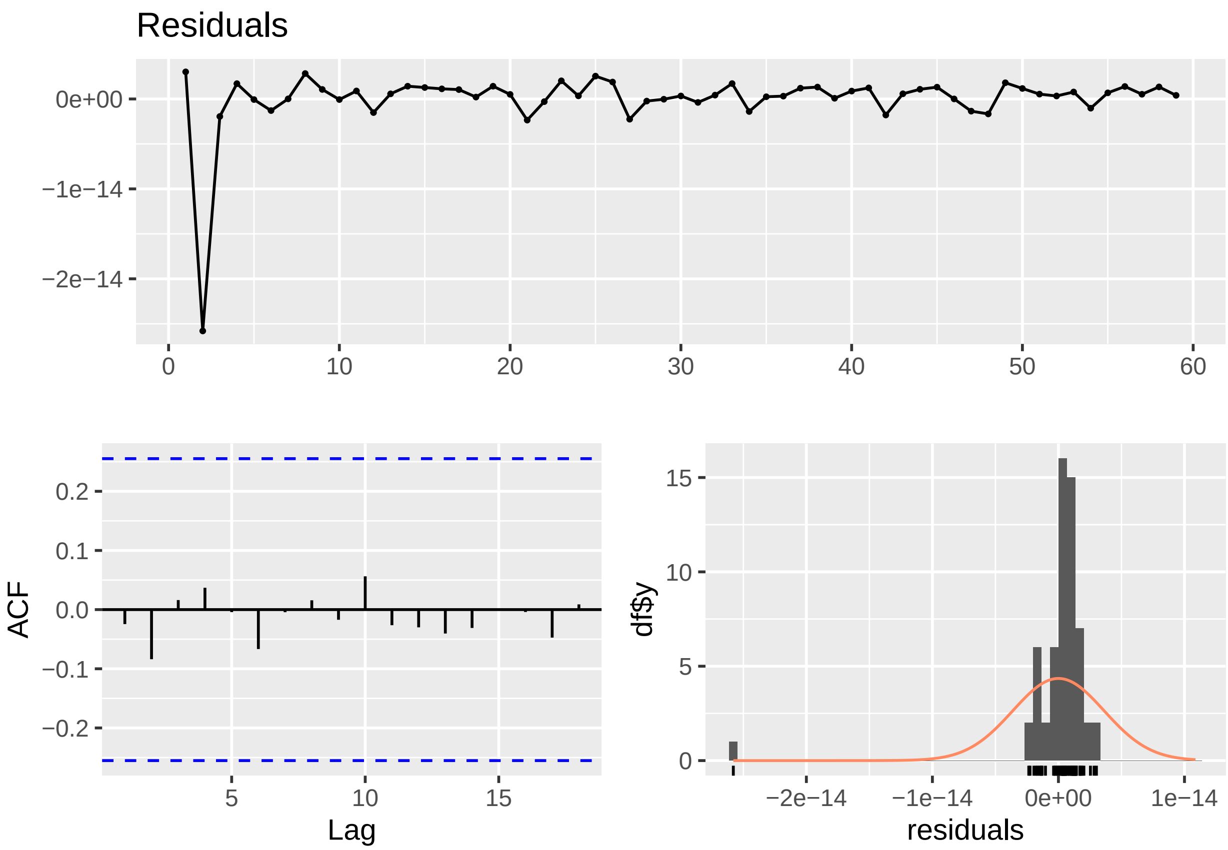
```
tsdisplay(WARDLreg.ts1$residuals)
```




```
tsdisplay(WARDLreg.ts2$residuals)
```

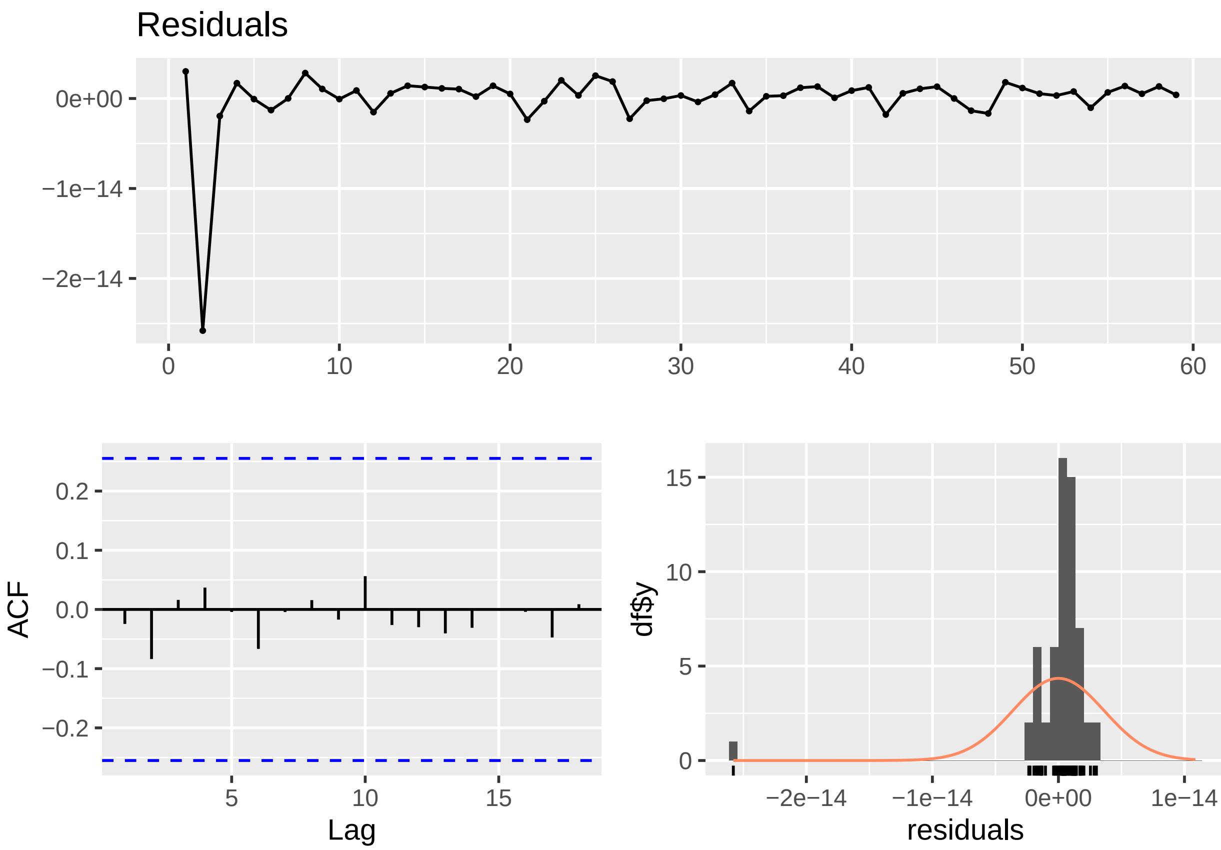


```
checkresiduals(LowesARDLreg.ts1)
```



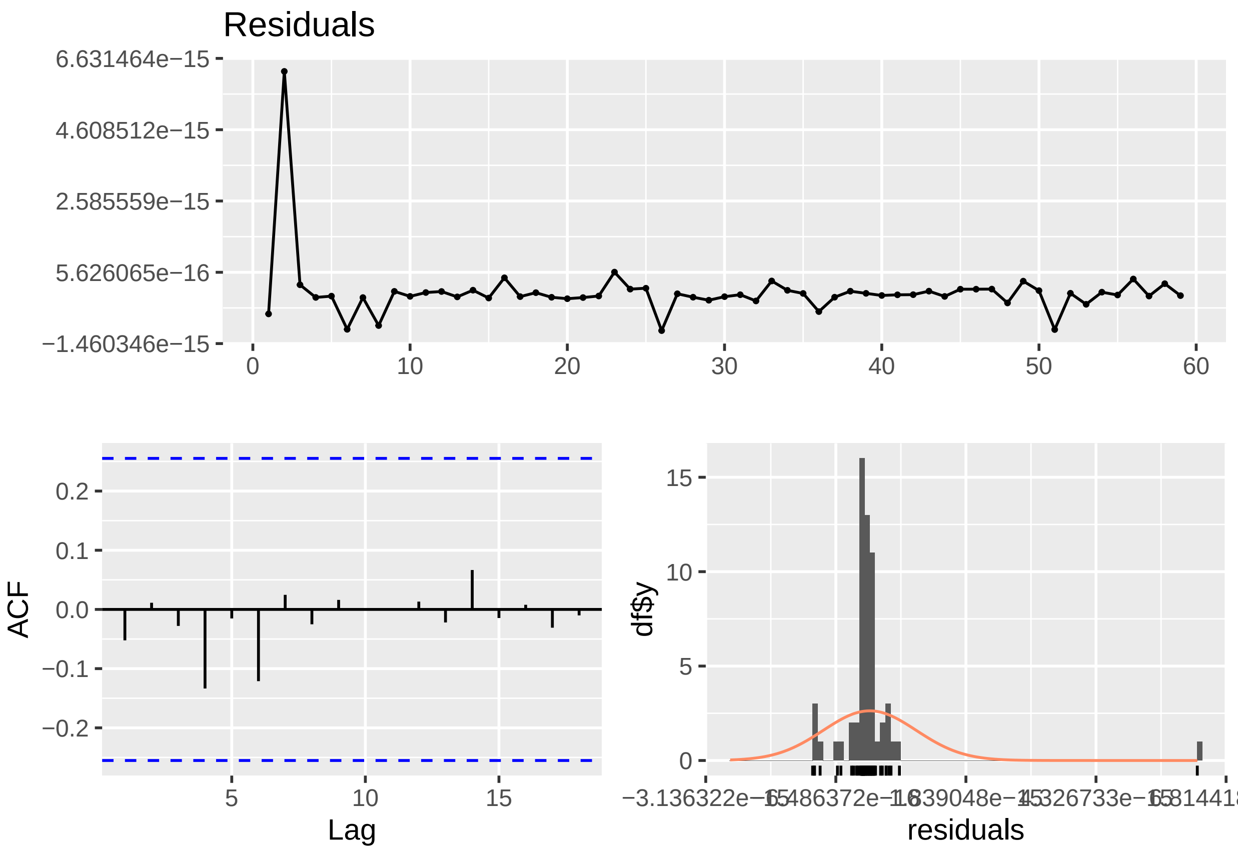

```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 1.0485, df = 10, p-value = 0.9998
```

```
checkresiduals(LowesARDLreg.ts2)
```



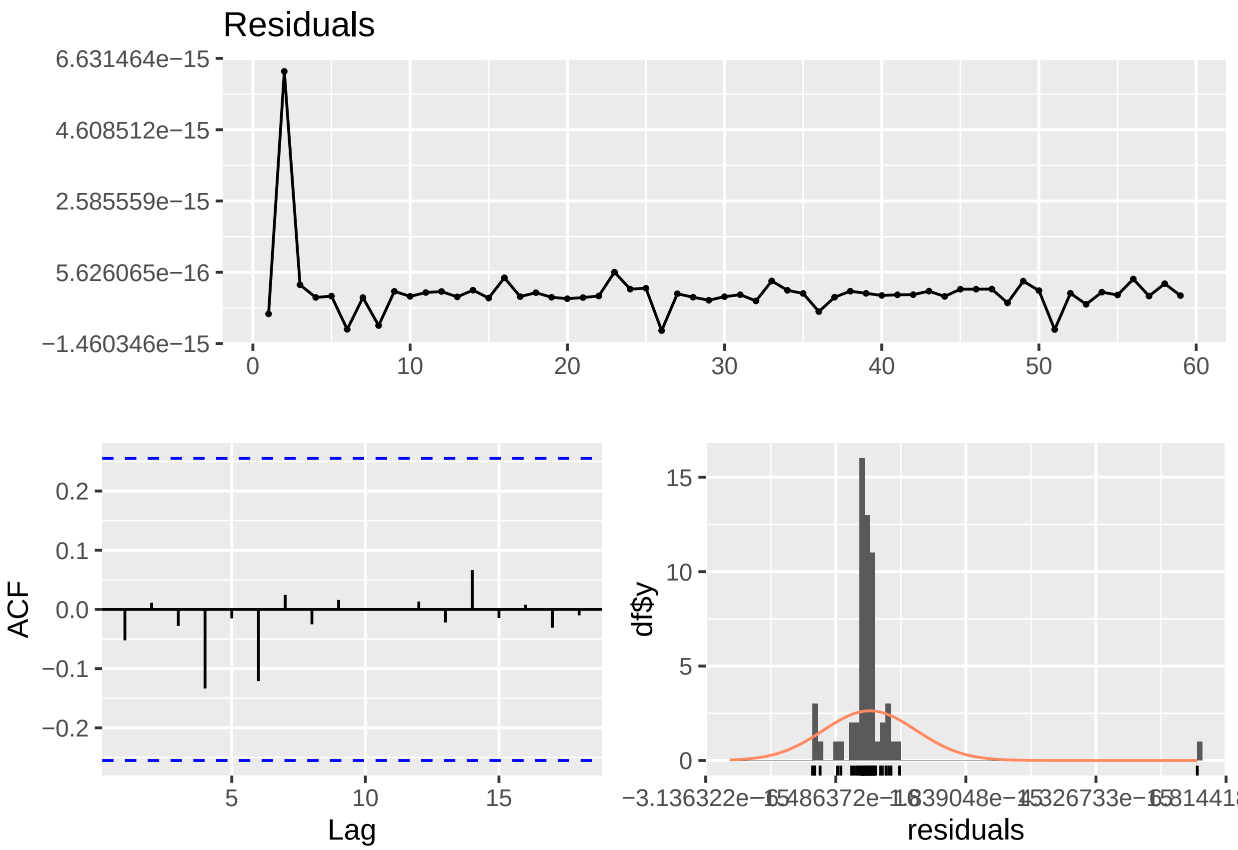

```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 1.0485, df = 10, p-value = 0.9998
```

```
checkresiduals(WARDLreg.ts1)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 2.4831, df = 10, p-value = 0.9911
```

```
checkresiduals(WARDLreg.ts2)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 2.4831, df = 10, p-value = 0.9911
```

```
#Training/Testing
#Similar to our AR models from before, when we created our two ardl models for
#Lowes and Walmart, the AIC and BIC were the same. The reason for this comes
#from the lack of significant lags, which goes back to the earlier point of
#using daily data instead of monthly data. This could help in finding lags
#that would be significant to test for.
testLowesARDLreg.ts1 = dynlm(`Lowes_%Change`~L(`Lowes_%Change`,12)+
                              L(`Lowes_%Change`,5)+L(`Home_Depot_%Change`,5),
                            data = test_df)
trainLowesARDLreg.ts1 = dynlm(`Lowes_%Change`~L(`Lowes_%Change`,12)+
                               L(`Lowes_%Change`,5)+L(`Home_Depot_%Change`,5),
                             data = train_df)
testLowesARDLreg.ts2 = dynlm(`Lowes_%Change`~L(`Lowes_%Change`,12)+
                              L(`Home_Depot_%Change`,5), data = test_df)
trainLowesARDLreg.ts2 = dynlm(`Lowes_%Change`~L(`Lowes_%Change`,12)+
                               L(`Home_Depot_%Change`,5), data = train_df)

sqrt(sum(testLowesARDLreg.ts1$residuals^2))
```

```
## [1] 2.450025e-15
```

```r
sqrt(sum(trainLowesARDLreg.ts1$residuals^2))
```

```
## [1] 8.505028e-16
```

```r
sqrt(sum(testLowesARDLreg.ts2$residuals^2))
```

```
## [1] 2.450025e-15
```

```r
sqrt(sum(trainLowesARDLreg.ts2$residuals^2))
```

```
## [1] 8.505028e-16
```

```r
testWARDLreg.ts1 = dynlm(`Walmart_%Change`~L(`Walmart_%Change`,5)+
                           L(`Walmart_%Change`,2)+L(`Inflation_Rate_%Change`,3),
                      data=test_df)
trainWARDLreg.ts1 = dynlm(`Walmart_%Change` ~L(`Walmart_%Change`,5) +
                            L(`Walmart_%Change`,2)+L(`Inflation_Rate_%Change`,3),
                      data=train_df)
testWARDLreg.ts2 = dynlm(`Walmart_%Change`~ L(`Walmart_%Change`,5)+
                           L(`Inflation_Rate_%Change`,3), data=test_df)
trainWARDLreg.ts2 = dynlm(`Walmart_%Change` ~ L(`Walmart_%Change`,5)+
                            L(`Inflation_Rate_%Change`,3), data=train_df)

sqrt(sum(testWARDLreg.ts1$residuals^2))
```

```
## [1] 3.33803e-15
```

```r
sqrt(sum(trainWARDLreg.ts1$residuals^2))
```

```
## [1] 6.155553e-15
```

```r
sqrt(sum(testWARDLreg.ts2$residuals^2))
```

```
## [1] 3.33803e-15
```

```r
sqrt(sum(trainWARDLreg.ts2$residuals^2))
```

```
## [1] 6.155553e-15
```

```r
#AIC and BIC
AIC(LowesARDLreg.ts1,LowesARDLreg.ts2)
```

```
##                  df       AIC
## LowesARDLreg.ts1  4 -3749.212
## LowesARDLreg.ts2  4 -3749.212
```

```r
BIC(LowesARDLreg.ts1,LowesARDLreg.ts2)
```

```
##                  df       BIC
## LowesARDLreg.ts1  4 -3740.902
## LowesARDLreg.ts2  4 -3740.902
```

```r
AIC(WARDLreg.ts1,WARDLreg.ts2)
```

```
##              df       AIC
## WARDLreg.ts1  4 -3914.919
## WARDLreg.ts2  4 -3914.919
```

```r
BIC(WARDLreg.ts1,WARDLreg.ts2)
```

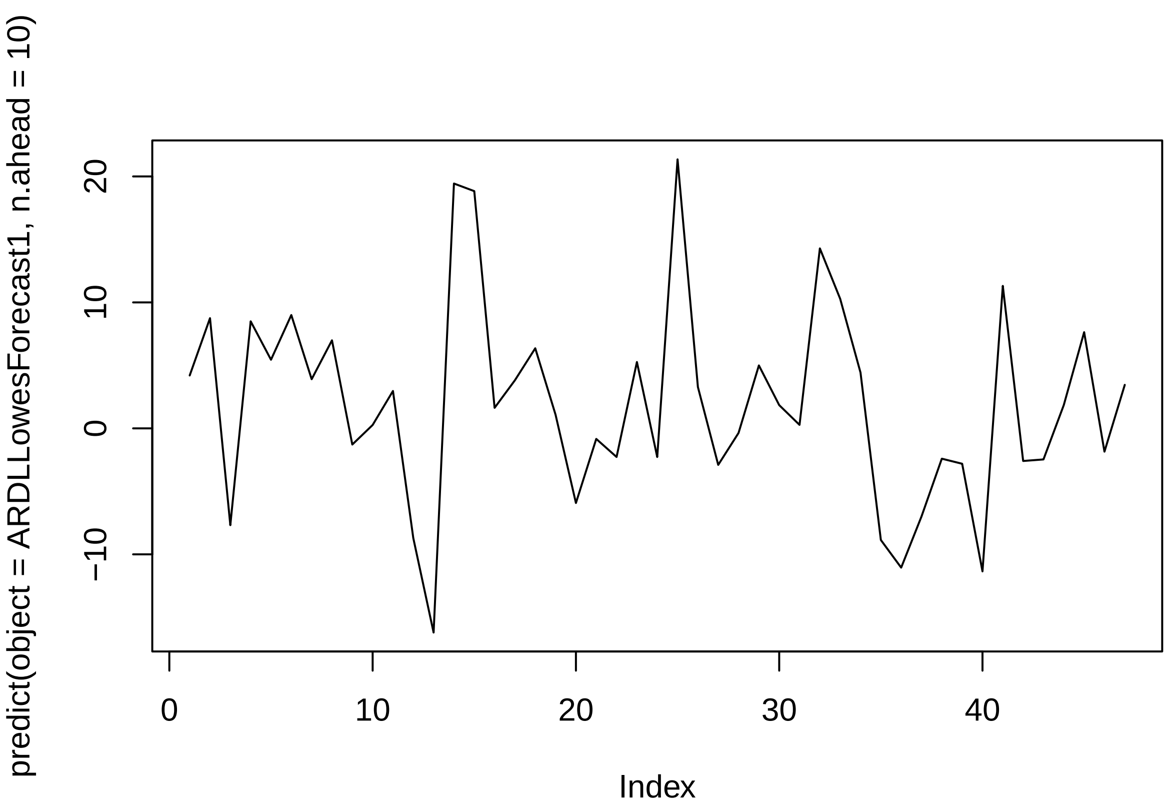
```
##              df       BIC
```

```
## WARDLreg.ts1  4 -3906.609
## WARDLreg.ts2  4 -3906.609
```

```
#10-step-ahead forecast
LowesHD = cbind(LowesTS,HDTS)
Lowes_HD = data.frame(LowesHD)

ARDLLowesForecast1 <- ardl(LowesTS ~ HDTS, data=Lowes_HD, order= c(12,5))
predict(object=ARDLLowesForecast1,n.ahead=10)
```

```
##          13          14          15          16          17          18
##   4.1976330   8.7475771  -7.6825899   8.4990549   5.4471156   8.9950959
##          19          20          21          22          23          24
##   3.9040807   6.9906403  -1.2834134   0.2687059   2.9673942  -8.7047138
##          25          26          27          28          29          30
## -16.2071046  19.4359988  18.8322808   1.6315705   3.8210985   6.3587449
##          31          32          33          34          35          36
##   1.0782411  -5.9246353  -0.8360724  -2.2714577   5.2678590  -2.2725746
##          37          38          39          40          41          42
##  21.3546723   3.2839348  -2.8973619  -0.3641723   4.9970264   1.8461106
##          43          44          45          46          47          48
##   0.2769080  14.2878019  10.2810327   4.4272246  -8.8589361 -11.0548223
##          49          50          51          52          53          54
##  -6.9934623  -2.4067290  -2.8080269 -11.3526526  11.3081097  -2.5887610
##          55          56          57          58          59
##  -2.4649054   1.8517320   7.6357723  -1.8460852   3.4543602
```

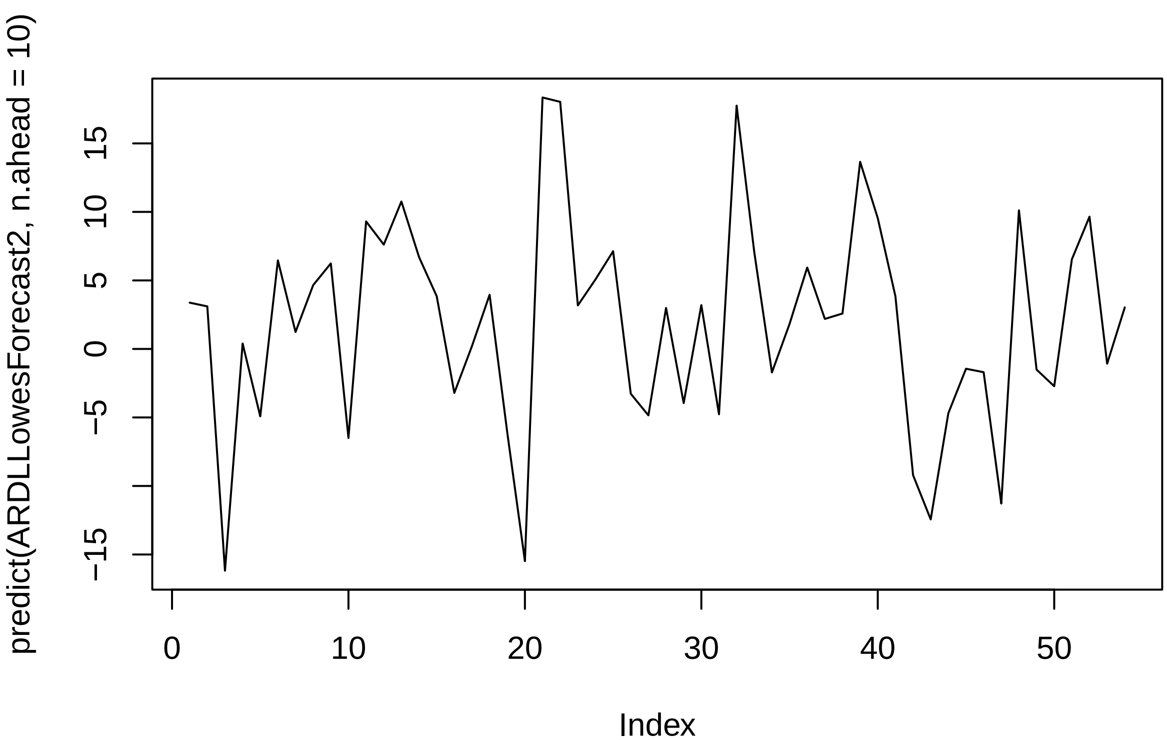
```
plot(predict(object=ARDLLowesForecast1,n.ahead=10), type="l")
```



```
ARDLLowesForecast2 = ardl(LowesTS ~ HDTS, data=Lowes_HD, order= c(5,5))
predict(ARDLLowesForecast2,n.ahead=10)
```

```
##           6           7           8           9          10          11
##   3.3782190   3.1044154 -16.1841181   0.3955455  -4.9164853   6.4605888
##          12          13          14          15          16          17
##   1.2385307   4.6556004   6.2412857  -6.4963344   9.3061135   7.6135122
##          18          19          20          21          22          23
##  10.7590243   6.7029111   3.8486037  -3.2101329   0.2103124   3.9536814
##          24          25          26          27          28          29
##  -6.0950030 -15.4828505  18.3483554  18.0322848   3.1774758   5.0726132
##          30          31          32          33          34          35
##   7.1311738  -3.2706370  -4.8468856   2.9922863  -3.9496954   3.1971807
##          36          37          38          39          40          41
##  -4.7718607  17.7545220   7.0614939  -1.7136227   1.8243503   5.9410501
##          42          43          44          45          46          47
##   2.1909086   2.5855001  13.6553202   9.5458753   3.8398546  -9.2086523
##          48          49          50          51          52          53
## -12.4392060  -4.6872902  -1.4456851  -1.6970115 -11.2822624  10.1157802
##          54          55          56          57          58          59
##  -1.5095336  -2.7217992   6.5368483   9.6525068  -1.0729198   3.0327609
```

```
plot(predict(ARDLLowesForecast2,n.ahead=10), type="l")
```

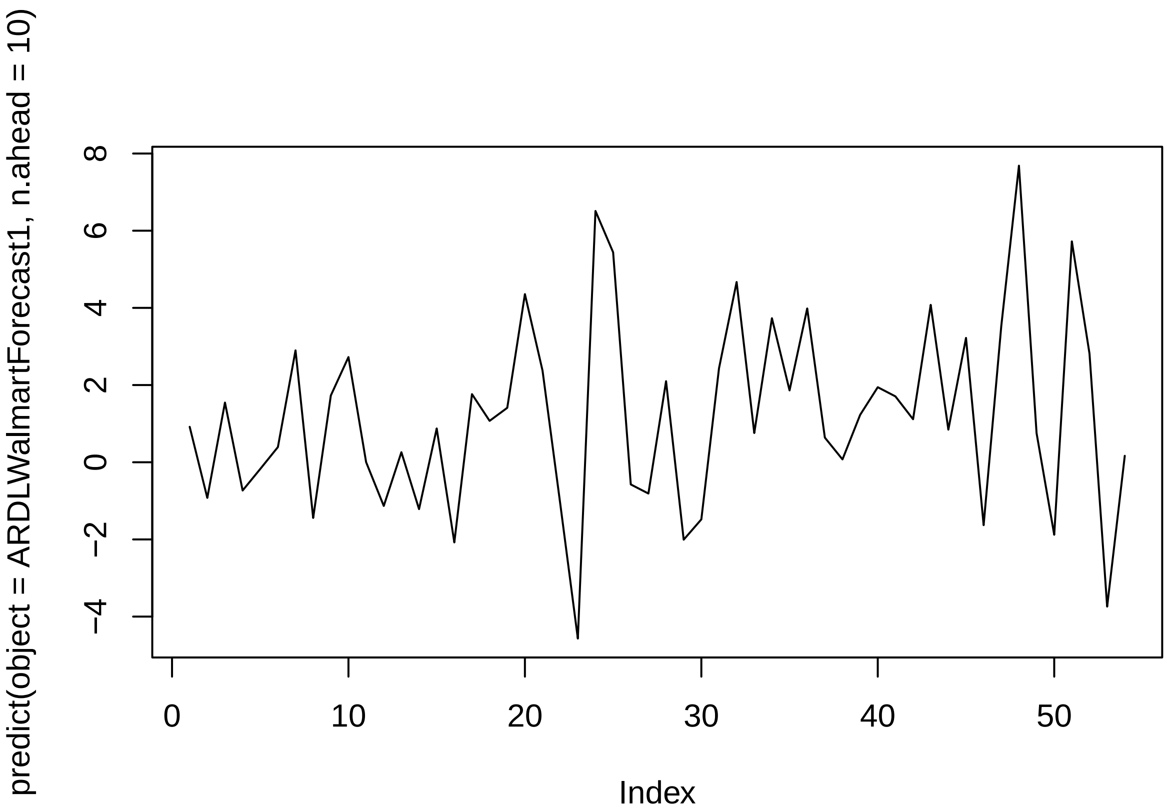



```
WalmartInflation = cbind(WTS,InflationTS)
Walmart_Inflation = data.frame(WalmartInflation)
ARDLWalmartForecast1 = ardl(WTS ~ InflationTS, data=Walmart_Inflation,
                             order = c(5,3))
predict(object=ARDLWalmartForecast1,n.ahead=10)
```

```
##            6             7             8             9            10            11
```

```
##  0.917995067 -0.923368577  1.546426972 -0.732474135 -0.174453732  0.392567972
##           12           13           14           15           16           17
##  2.898969933 -1.443044801  1.729649093  2.725376869  0.002450912 -1.131893440
##           18           19           20           21           22           23
##  0.259307350 -1.215320480  0.875124795 -2.077223842  1.764676951  1.072323556
##           24           25           26           27           28           29
##  1.410445951  4.356722930  2.372602271 -1.065687549 -4.569225071  6.511321728
##           30           31           32           33           34           35
##  5.439704615 -0.575047351 -0.811036910  2.100313709 -2.007369706 -1.480838107
##           36           37           38           39           40           41
##  2.432936010  4.670410411  0.757114521  3.732180712  1.861230571  3.984906157
##           42           43           44           45           46           47
##  0.637973856  0.073896978  1.227969157  1.944389930  1.706561264  1.114276237
##           48           49           50           51           52           53
##  4.078220746  0.844796839  3.221770797 -1.630375935  3.531198741  7.685216964
##           54           55           56           57           58           59
##  0.754506668 -1.879393972  5.723254948  2.814008134 -3.740827050  0.167180344
```

```r
plot(predict(object=ARDLWalmartForecast1,n.ahead=10), type="l")
```

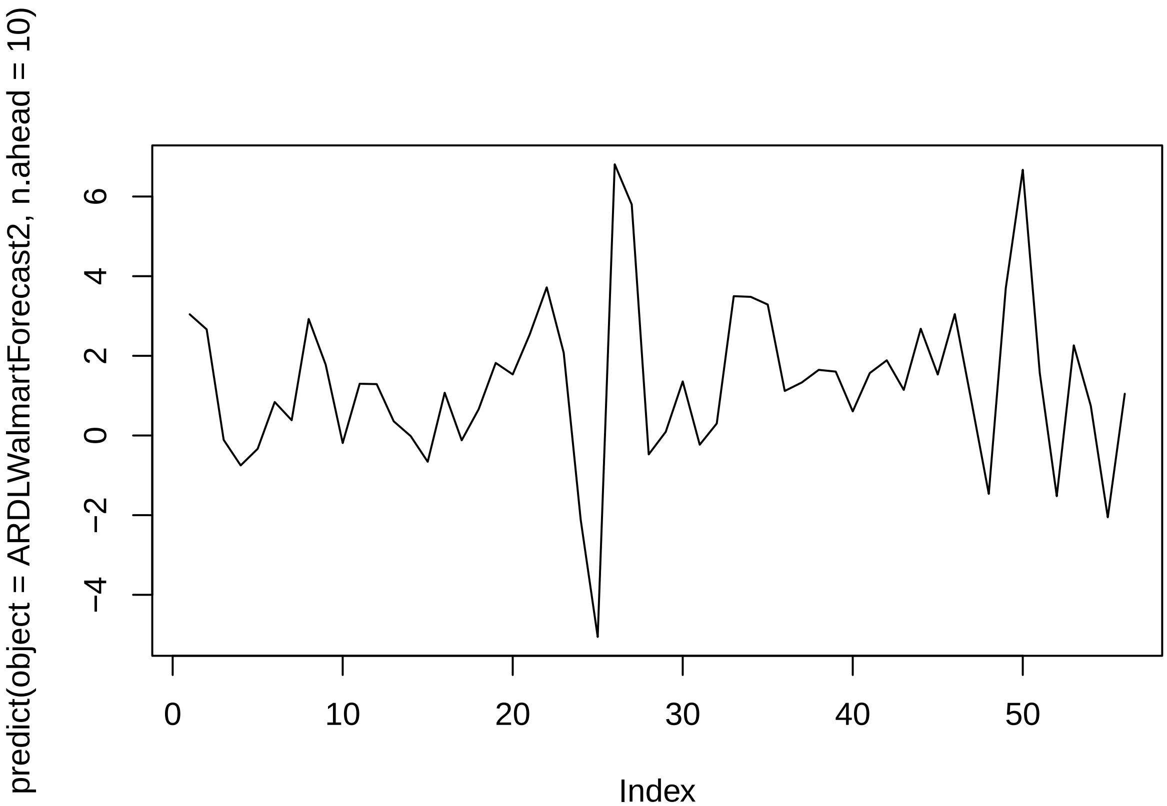



```r
ARDLWalmartForecast2 = ardl(WTS ~ InflationTS, data=Walmart_Inflation,
                            order = c(2,3))
predict(object=ARDLWalmartForecast2,n.ahead=10)
```

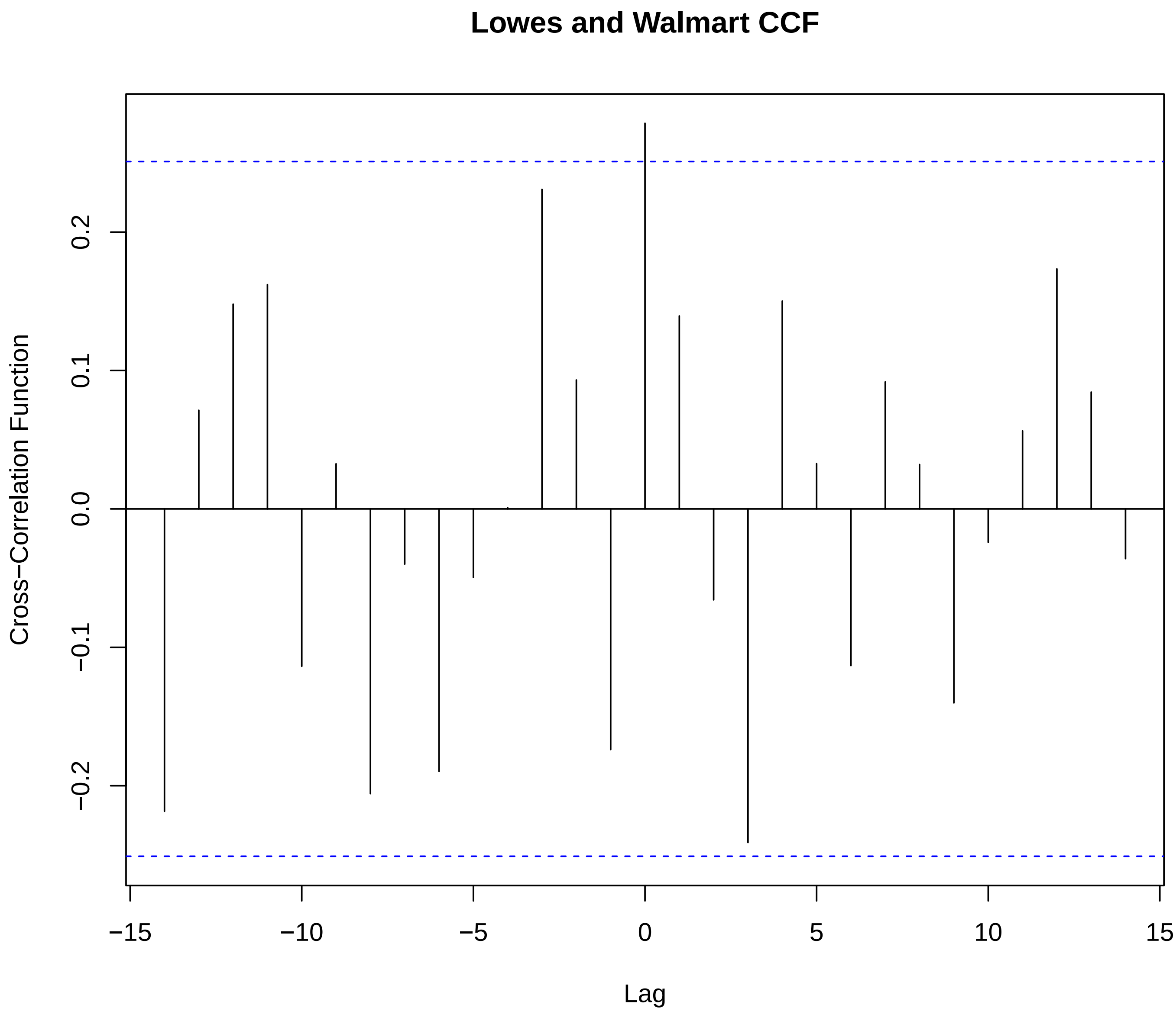
```
##           4           5           6           7           8           9
##  3.04335529  2.66363814 -0.11016664 -0.75138445 -0.33350319  0.84077031
##          10          11          12          13          14          15
##  0.38320530  2.92580826  1.77777400 -0.18970815  1.29884453  1.29005135
##          16          17          18          19          20          21
```

```
##   0.35690077 -0.01555300 -0.65955998  1.07315538 -0.12088899  0.66140895
##           22          23          24          25          26          27
##   1.82181212  1.53361091  2.53674873  3.71884334  2.07669899 -2.12094865
##           28          29          30          31          32          33
## -5.05737043  6.80907461  5.79991419 -0.47464875  0.09139935  1.35699670
##           34          35          36          37          38          39
## -0.23153024  0.30037106  3.49821209  3.48078639  3.28815692  1.11828283
##           40          41          42          43          44          45
##   1.33005977  1.64742703  1.60382618  0.60461527  1.56660045  1.88746408
##           46          47          48          49          50          51
##   1.14424347  2.67977000  1.53098153  3.04823510  0.81285862 -1.46643509
##           52          53          54          55          56          57
##   3.69797814  6.67089914  1.56695394 -1.52376382  2.26201902  0.74599885
##           58          59
## -2.05399169  1.04750197
```

```
plot(predict(object=ARDLWalmartForecast2,n.ahead=10), type="l")
```




```
#5
#CCF
#Our Cross-Validation Function model determined that predicting Lowes and
#Walmart future stock price is relatively accurate. All lags stay inside the
#border determining optimal predictions.
ccf(as.ts(LowesPC), as.ts(WPC), na.action = na.pass,
    ylab= "Cross-Correlation Function", main = "Lowes and Walmart CCF")
```




```
LowesW = cbind(LowesTS,WTS)
LowesWtot = data.frame(LowesW)

LowesW_tot <- LowesWtot[-c(60,61),]

VARselect(LowesW_tot, lag.max=22)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     18     18     18     19
##
## $criteria
##                   1           2           3           4           5           6
## AIC(n)     7.939442    7.983642    8.096001    8.013328    8.080622    8.237311
## HQ(n)      8.031538    8.137135    8.310891    8.289615    8.418307    8.636392
## SC(n)      8.200672    8.419025    8.705537    8.797017    9.038465    9.369307
## FPE(n) 2807.796579 2942.365388 3311.738294 3081.674645 3353.104335 4023.243185
```

```
##                 7           8           9          10          11          12
## AIC(n)    8.336271    8.320857    8.504020    8.139463    8.209840    8.143801
## HQ(n)     8.796749    8.842733    9.087294    8.784134    8.915908    8.911266
## SC(n)     9.642420    9.801160   10.158477    9.968073   10.212603   10.320717
## FPE(n) 4605.706877 4767.240473 6123.727512 4651.298009 5619.106785 6158.288385
##                13          14          15           16          17   18   19
## AIC(n)    8.220771    7.990158    7.689599     7.670732    3.181268 -Inf -Inf
## HQ(n)     9.049633    8.880417    8.641256     8.683786    4.255719 -Inf -Inf
## SC(n)    10.571841   10.515381   10.388975    10.544261    6.228950 -Inf -Inf
## FPE(n) 8221.008954 8738.112333 9835.408943 18537.781118 709.472606  NaN    0
##          20   21   22
## AIC(n) -Inf -Inf -Inf
## HQ(n)  -Inf -Inf -Inf
## SC(n)  -Inf -Inf -Inf
## FPE(n)    0    0    0
```

```
LowesW_model = VAR(LowesW_tot,p=18)
summary(LowesW_model)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: LowesTS, WTS
## Deterministic variables: const
## Sample size: 41
## Log Likelihood: -175.6
## Roots of the characteristic polynomial:
## 1.034 1.034 1.033 1.033 1.026 1.026 1.021 1.021 1.019 1.019 1.019 1.019 0.9968 0.9968 0.9951 0.9951 (
## Call:
## VAR(y = LowesW_tot, p = 18)
##
##
## Estimation results for equation LowesTS:
## ========================================
## LowesTS = LowesTS.l1 + WTS.l1 + LowesTS.l2 + WTS.l2 + LowesTS.l3 + WTS.l3 + LowesTS.l4 + WTS.l4 + Lo
##
##              Estimate Std. Error t value Pr(>|t|)
## LowesTS.l1   0.012832   0.551714   0.023    0.983
## WTS.l1       0.088055   0.830022   0.106    0.921
## LowesTS.l2  -0.204891   0.543248  -0.377    0.725
## WTS.l2       0.350784   0.887562   0.395    0.713
## LowesTS.l3   0.576320   0.528823   1.090    0.337
## WTS.l3      -0.785375   0.931286  -0.843    0.447
## LowesTS.l4   0.237126   0.545302   0.435    0.686
## WTS.l4      -0.509948   0.982426  -0.519    0.631
## LowesTS.l5   0.267088   0.550689   0.485    0.653
## WTS.l5      -1.150697   0.959452  -1.199    0.297
## LowesTS.l6   0.005958   0.527312   0.011    0.992
## WTS.l6      -0.663770   0.907738  -0.731    0.505
## LowesTS.l7  -0.318014   0.490651  -0.648    0.552
## WTS.l7       0.386606   0.922767   0.419    0.697
## LowesTS.l8   0.308689   0.513471   0.601    0.580
## WTS.l8      -0.167740   0.929797  -0.180    0.866
## LowesTS.l9  -0.153964   0.519436  -0.296    0.782
## WTS.l9       0.838706   0.842163   0.996    0.376
```

```
## LowesTS.l10  0.097108   0.469914   0.207    0.846
## WTS.l10     -0.217479   0.885799  -0.246    0.818
## LowesTS.l11 -0.637707   0.520870  -1.224    0.288
## WTS.l11      1.128037   0.810886   1.391    0.237
## LowesTS.l12 -0.345399   0.502286  -0.688    0.529
## WTS.l12      0.463890   1.001323   0.463    0.667
## LowesTS.l13 -0.426804   0.491780  -0.868    0.434
## WTS.l13      0.293127   0.840468   0.349    0.745
## LowesTS.l14 -0.498031   0.517954  -0.962    0.391
## WTS.l14     -0.104248   0.932734  -0.112    0.916
## LowesTS.l15  0.223718   0.574281   0.390    0.717
## WTS.l15      0.256853   1.110445   0.231    0.828
## LowesTS.l16  0.184070   0.610240   0.302    0.778
## WTS.l16      0.004140   0.932805   0.004    0.997
## LowesTS.l17  0.599697   0.579490   1.035    0.359
## WTS.l17     -0.639942   0.966234  -0.662    0.544
## LowesTS.l18 -0.212258   0.515775  -0.412    0.702
## WTS.l18      0.018774   0.965039   0.019    0.985
## const        3.051586   7.552972   0.404    0.707
##
##
## Residual standard error: 12.17 on 4 degrees of freedom
## Multiple R-Squared: 0.8027,  Adjusted R-squared: -0.9728
## F-statistic: 0.4521 on 36 and 4 DF,  p-value: 0.9128
##
##
## Estimation results for equation WTS:
## ====================================
## WTS = LowesTS.l1 + WTS.l1 + LowesTS.l2 + WTS.l2 + LowesTS.l3 + WTS.l3 + LowesTS.l4 + WTS.l4 + LowesTS
##
##             Estimate Std. Error t value Pr(>|t|)
## LowesTS.l1  -0.02336    0.25675  -0.091   0.9319
## WTS.l1      -0.08039    0.38627  -0.208   0.8453
## LowesTS.l2   0.18429    0.25281   0.729   0.5064
## WTS.l2      -0.46181    0.41305  -1.118   0.3262
## LowesTS.l3   0.63256    0.24610   2.570   0.0620 .
## WTS.l3      -0.69932    0.43340  -1.614   0.1819
## LowesTS.l4   0.22306    0.25377   0.879   0.4290
## WTS.l4      -0.56224    0.45720  -1.230   0.2862
## LowesTS.l5   0.30220    0.25628   1.179   0.3037
## WTS.l5      -0.82735    0.44650  -1.853   0.1375
## LowesTS.l6  -0.16265    0.24540  -0.663   0.5437
## WTS.l6      -0.02160    0.42244  -0.051   0.9617
## LowesTS.l7  -0.15158    0.22834  -0.664   0.5431
## WTS.l7       0.41123    0.42943   0.958   0.3925
## LowesTS.l8   0.14047    0.23896   0.588   0.5882
## WTS.l8      -0.15602    0.43270  -0.361   0.7367
## LowesTS.l9  -0.30683    0.24173  -1.269   0.2732
## WTS.l9       0.11633    0.39192   0.297   0.7814
## LowesTS.l10 -0.28883    0.21869  -1.321   0.2571
## WTS.l10     -0.37838    0.41223  -0.918   0.4106
## LowesTS.l11 -0.45769    0.24240  -1.888   0.1320
## WTS.l11      0.95170    0.37736   2.522   0.0652 .
## LowesTS.l12 -0.07566    0.23375  -0.324   0.7624
```

```
## WTS.l12     -0.21925    0.46599  -0.471   0.6625
## LowesTS.l13 -0.26496    0.22886  -1.158   0.3114
## WTS.l13     -0.23577    0.39113  -0.603   0.5791
## LowesTS.l14 -0.29190    0.24104  -1.211   0.2925
## WTS.l14     -0.14196    0.43407  -0.327   0.7600
## LowesTS.l15  0.19168    0.26726   0.717   0.5129
## WTS.l15     -0.42477    0.51677  -0.822   0.4573
## LowesTS.l16  0.04304    0.28399   0.152   0.8869
## WTS.l16     -0.16772    0.43410  -0.386   0.7189
## LowesTS.l17  0.39902    0.26968   1.480   0.2131
## WTS.l17     -0.34373    0.44966  -0.764   0.4872
## LowesTS.l18 -0.26422    0.24003  -1.101   0.3328
## WTS.l18     -0.75569    0.44910  -1.683   0.1677
## const        6.62504    3.51495   1.885   0.1325
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 5.664 on 4 degrees of freedom
## Multiple R-Squared: 0.9026,  Adjusted R-squared: 0.02597
## F-statistic:  1.03 on 36 and 4 DF,  p-value: 0.5647
##
##
##
## Covariance matrix of residuals:
##         LowesTS   WTS
## LowesTS  148.15 53.51
## WTS       53.51 32.09
##
## Correlation matrix of residuals:
##         LowesTS    WTS
## LowesTS  1.0000 0.7761
## WTS      0.7761 1.0000
```

```r
#ACF,PACF,data,Fitted Values
# For Lowes, our IRF model shows that an initial postive shock in the stock
#price does not have a severe response in the long run. As a matter of fact,
#it returns back to normal prior to period 1 as the confidence interval line
#hits 0 before period 1.
#Similar to Lowes, an initial positive shock to Walmart's stock price does not
#have a significant impact to price in the long run. There is some volatility
#throughout the 10 periods but has very constant volatility expressing it is
#not significant.

plot(LowesW_model)
```

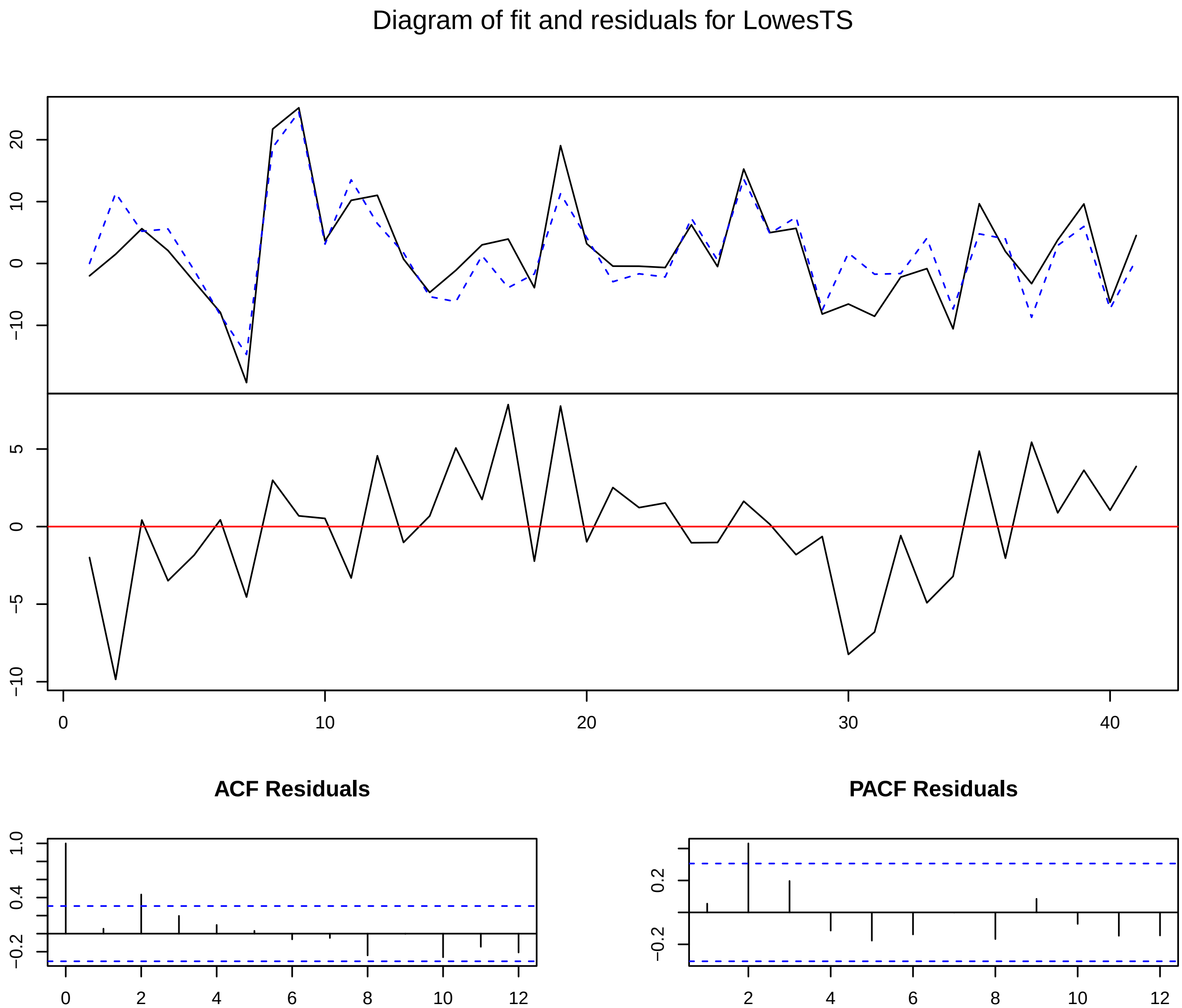
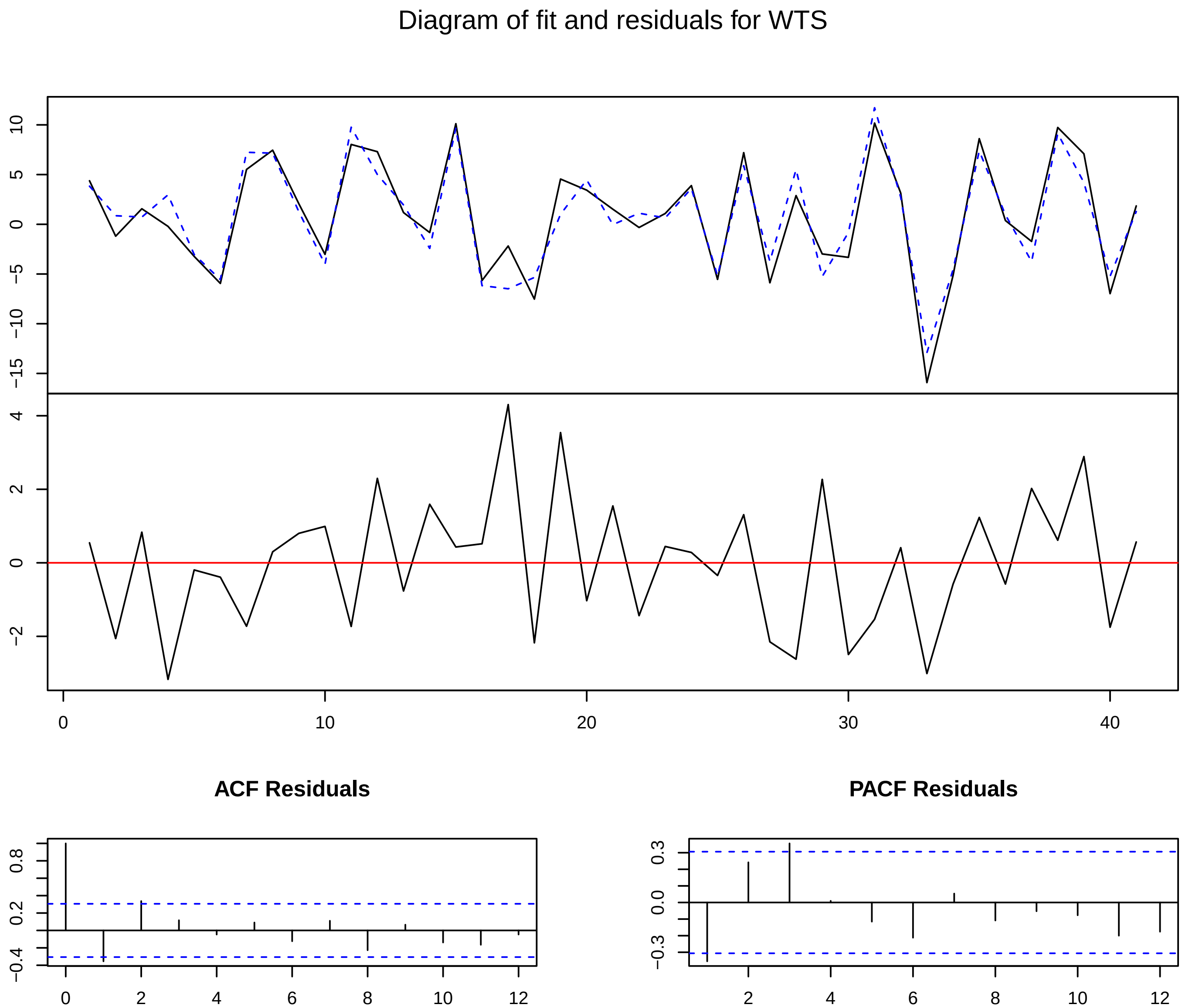


Diagram of fit and residuals for LowesTS
ACF Residuals
PACF Residuals



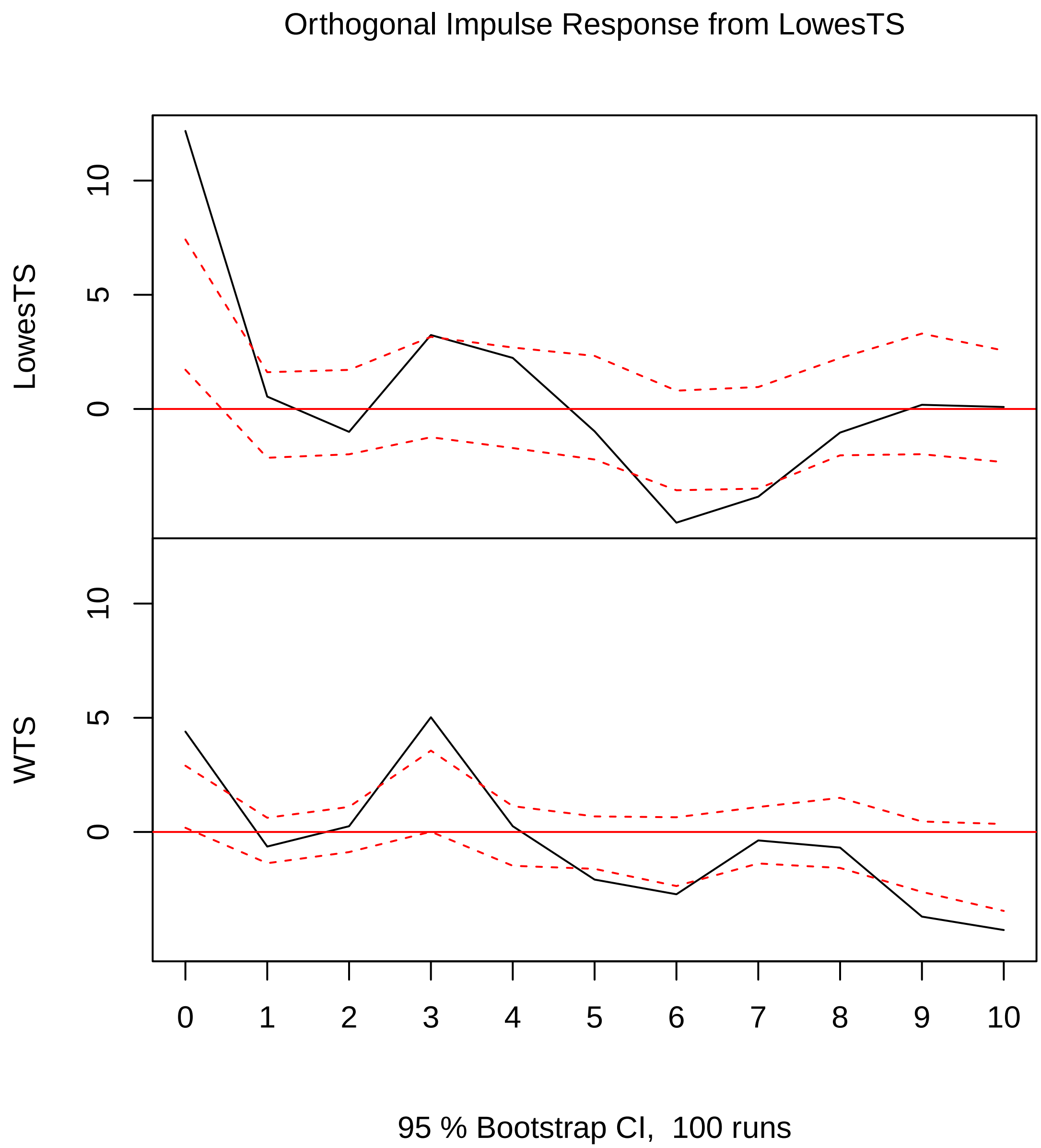

```
#IRF
# For Lowes, our IRF model shows that an initial postive shock in the stock
#price does not have a severe response in the long run. As a matter of fact,
#it returns back to normal prior to period 1 as the confidence interval line
#hits 0 before period 1.

#Similar to Lowes, an initial positive shock to Walmarts stock price does not
#have a significant impact to price in the long run. There is some volatility
#throughout the 10 periods but has very constant volatility expressing it is
#not significant.
irf(LowesW_model)
```

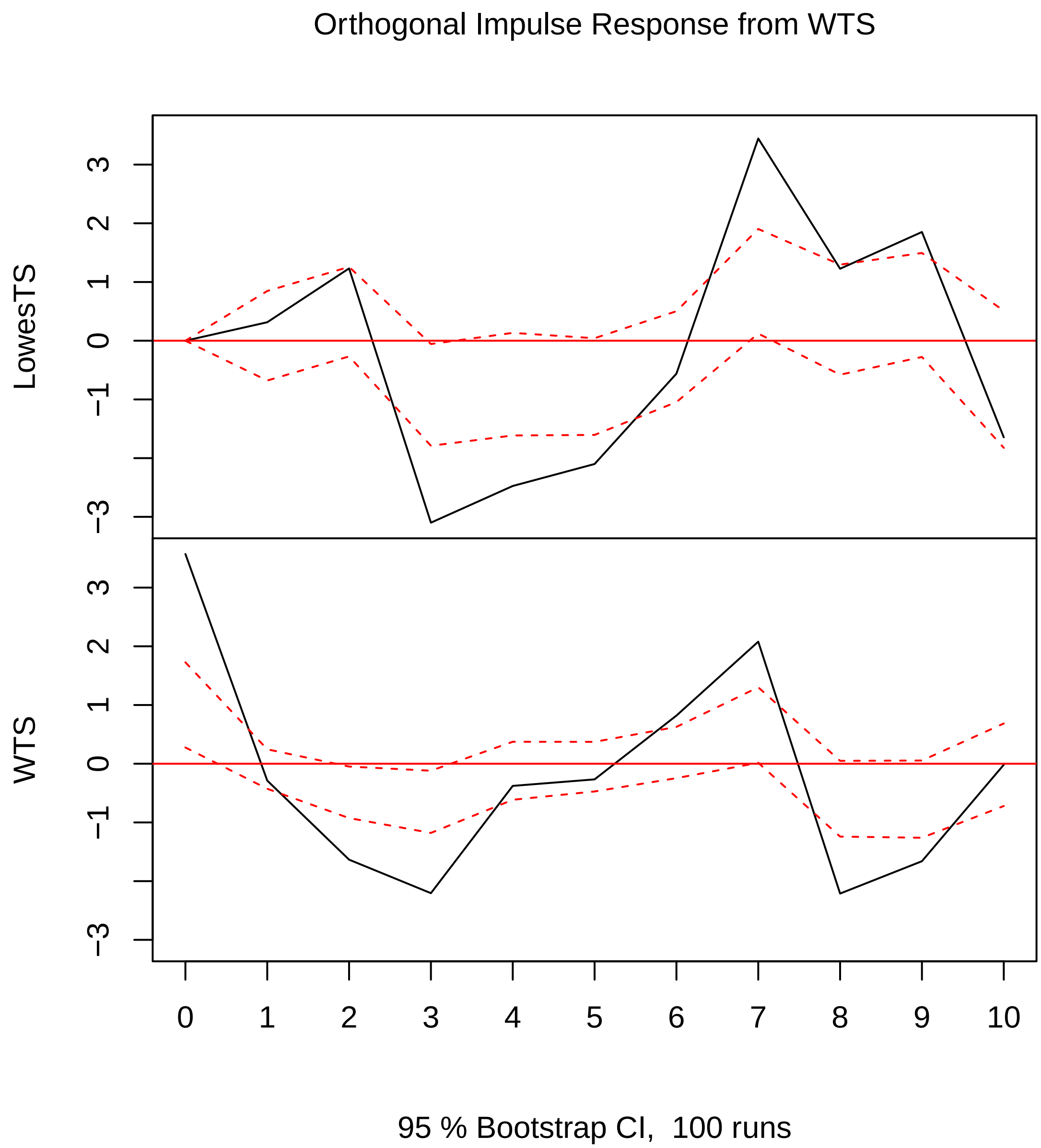
```
##
## Impulse response coefficients
## $LowesTS
##            LowesTS        WTS
##  [1,] 12.17167547  4.3958962
##  [2,]  0.54326495 -0.6376961
```

```
##  [3,] -1.00103563  0.2516443
##  [4,]  3.23666033  5.0229454
##  [5,]  2.23567334  0.2529921
##  [6,] -0.97820652 -2.0839017
##  [7,] -4.97706205 -2.7273630
##  [8,] -3.84045699 -0.3697283
##  [9,] -1.03260981 -0.6841929
## [10,]  0.18239790 -3.7062982
## [11,]  0.08613286 -4.2933781
##
## $WTS
##          LowesTS         WTS
##  [1,]  0.0000000  3.57229188
##  [2,]  0.3145568 -0.28718545
##  [3,]  1.2318506 -1.63397187
##  [4,] -3.0988513 -2.20500618
##  [5,] -2.4743393 -0.37742970
##  [6,] -2.0999064 -0.26666665
##  [7,] -0.5613834  0.81956452
##  [8,]  3.4444938  2.07887720
##  [9,]  1.2259931 -2.21142830
## [10,]  1.8518572 -1.66077936
## [11,] -1.6450943 -0.01758041
##
##
## Lower Band, CI= 0.95
## $LowesTS
##         LowesTS        WTS
##  [1,]  1.156617  0.1726512
##  [2,] -2.753204 -1.4374702
##  [3,] -2.452944 -0.9112523
##  [4,] -1.386110  0.1576801
##  [5,] -1.683711 -1.8846616
##  [6,] -2.643126 -2.3133758
##  [7,] -4.941233 -2.5017713
##  [8,] -4.431865 -1.6242937
##  [9,] -2.807662 -2.0176420
## [10,] -1.836833 -3.1023781
## [11,] -1.948787 -2.9768056
##
## $WTS
##           LowesTS         WTS
##  [1,]  0.00000000  0.31433452
##  [2,] -0.90547425 -0.52733129
##  [3,] -0.72831595 -1.04214564
##  [4,] -2.21322091 -1.39410802
##  [5,] -2.09673143 -0.99706983
##  [6,] -1.65246194 -0.76844318
##  [7,] -1.03030101 -0.17773611
##  [8,] -0.07388068 -0.02287288
##  [9,] -0.52974267 -1.37478355
## [10,] -0.43108404 -1.14548922
## [11,] -1.85058509 -1.18793924
##
```

```
##
## Upper Band, CI= 0.95
## $LowesTS
##         LowesTS       WTS
##  [1,] 6.8360045 2.6146113
##  [2,] 2.3705877 0.7389175
##  [3,] 1.7805584 1.2794463
##  [4,] 3.4360372 3.5323968
##  [5,] 3.1414038 1.5708402
##  [6,] 2.3278953 0.6682328
##  [7,] 0.8915001 0.7300118
##  [8,] 1.7066708 1.4145756
##  [9,] 3.3244643 2.3465361
## [10,] 4.1613499 0.6280361
## [11,] 5.1457618 1.4463797
##
## $WTS
##           LowesTS        WTS
##  [1,]  0.00000000  1.9403635
##  [2,]  0.54651300  0.2549789
##  [3,]  1.18404557  0.0505537
##  [4,] -0.16077243 -0.2184105
##  [5,] -0.06246189  0.1841670
##  [6,]  0.31686416  0.3915424
##  [7,]  0.81731343  0.8551959
##  [8,]  2.02563229  1.3626456
##  [9,]  1.49988406  0.1182953
## [10,]  2.10959913  0.3194450
## [11,]  0.46616560  0.8768049
```

```
plot(irf(LowesW_model))
```


Orthogonal Impulse Response from LowesTS
LowesTS
WTS
10
5
0
0 1 2 3 4 5 6 7 8 9 10
95 % Bootstrap CI, 100 runs




```r
#Granger-Causality
#Our p value was not significant, which means that Lowes and Walmart are not
#great predictor of each others stock price.
grangertest(LowesTS~WTS, order=17)
```

```
## Granger causality test
##
## Model 1: LowesTS ~ Lags(LowesTS, 1:17) + Lags(WTS, 1:17)
## Model 2: LowesTS ~ Lags(LowesTS, 1:17)
##   Res.Df  Df      F Pr(>F)
## 1      7
## 2     24 -17 0.8375 0.6426
```

```r
#AIC and BIC
#Looking at the AIC and BIC of both companies, Walmart has a lower AIC and BIC
#which indicates that Walmart stock price has a larger effect on Lowes than
#Lowes does on Walmart. Walmart is a better fit with this model.
```

```r
#Lowes
LowesW.ts = dynlm(LowesPC~L(WPC,5)+L(WPC,11)+L(LowesPC,11))
#Walmart
WLowes.ts = dynlm(WPC~L(LowesPC,3)+L(WPC,5)+L(LowesPC,11)+L(WPC,11))

AIC(LowesW.ts,WLowes.ts)
```

```
##           df       AIC
## LowesW.ts  4 -3787.051
## WLowes.ts  4 -3910.696
```

```r
BIC(LowesW.ts,WLowes.ts)
```

```
##           df       BIC
## LowesW.ts  4 -3778.741
## WLowes.ts  4 -3902.386
```

```r
#Training/Testing
testLowesW.ts = dynlm(`Lowes_%Change`~L(`Walmart_%Change`,5)+
                        L(`Walmart_%Change`,11)+L(`Lowes_%Change`,11),
                      data=test_df)
trainLowesW.ts = dynlm(`Lowes_%Change`~L(`Walmart_%Change`,5)+
                         L(`Walmart_%Change`,11)+L(`Lowes_%Change`,11),
                       data=train_df)

testWLowes.ts = dynlm(`Walmart_%Change`~L(`Lowes_%Change`,3)+
                        L(`Walmart_%Change`,5)+L(`Lowes_%Change`,11)+
                        L(`Walmart_%Change`,11), data=train_df)
trainWLowes.ts = dynlm(`Walmart_%Change`~L(`Lowes_%Change`,3)+
                         L(`Walmart_%Change`,5)+L(`Lowes_%Change`,11)+
                         L(`Walmart_%Change`,11), data=train_df)


sqrt(sum(testLowesW.ts$residuals^2))
```

```
## [1] 3.544029e-15
```

```r
sqrt(sum(trainLowesW.ts$residuals^2))
```

```
## [1] 1.991017e-14
```

```r
sqrt(sum(testWLowes.ts$residuals^2))
```

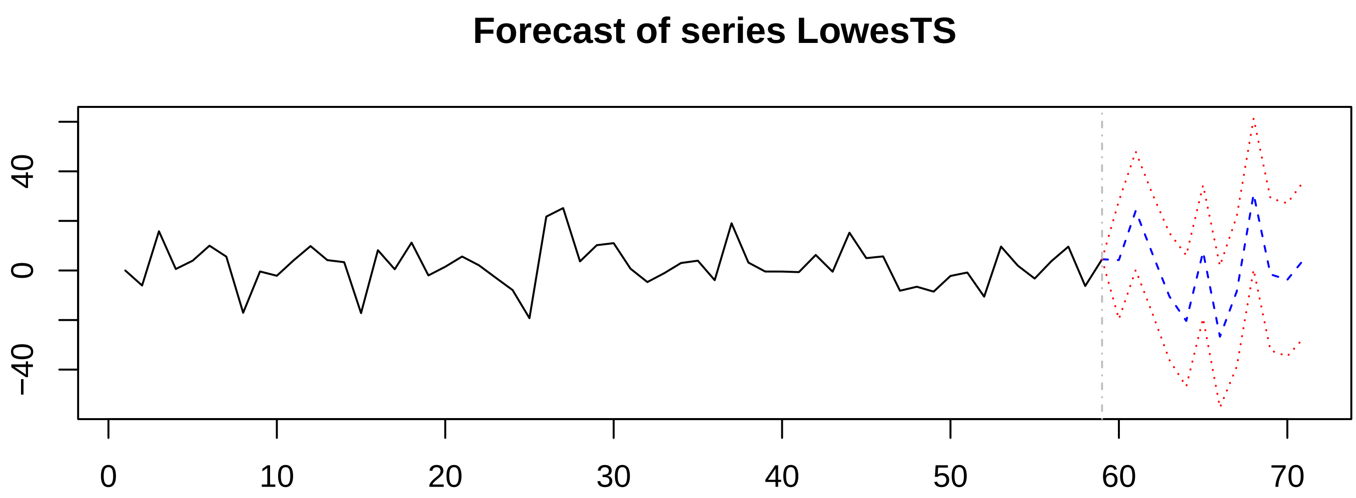
```
## [1] 1.176281e-15
```

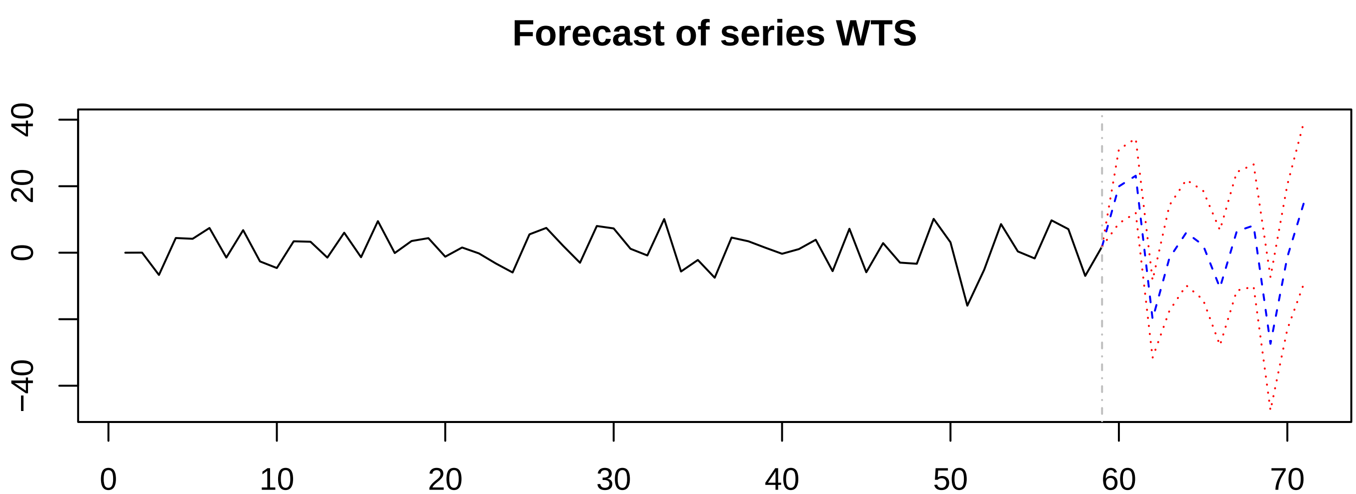
```r
sqrt(sum(trainWLowes.ts$residuals^2))
```

```
## [1] 1.176281e-15
```

```r
#N-step-ahead forecast
# Our forecast for the stock price of Lowes for the next 10 months express
#constant volatility of positive periods and negative periods but by the last
#month, it returns to the previous price prior to the prediction.
#Our Walmart forecast expresses relatively constant volatility as well but
#by the end of the predicted 10 months, the stock price of Walmart is expected
#to by slightly higher than it was 10 months prior. Of course, this model does
#omit external factors which could affect this prediction.
LowesW_model_predict = predict(object=LowesW_model, n.ahead=12)
```

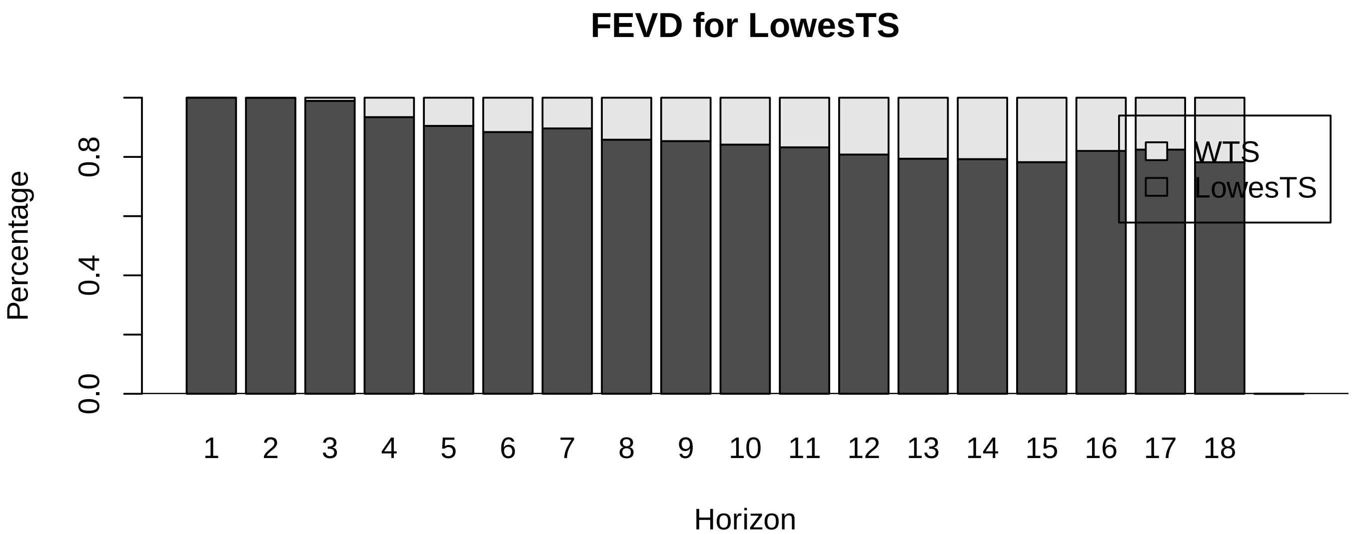
```
plot(LowesW_model_predict)
```

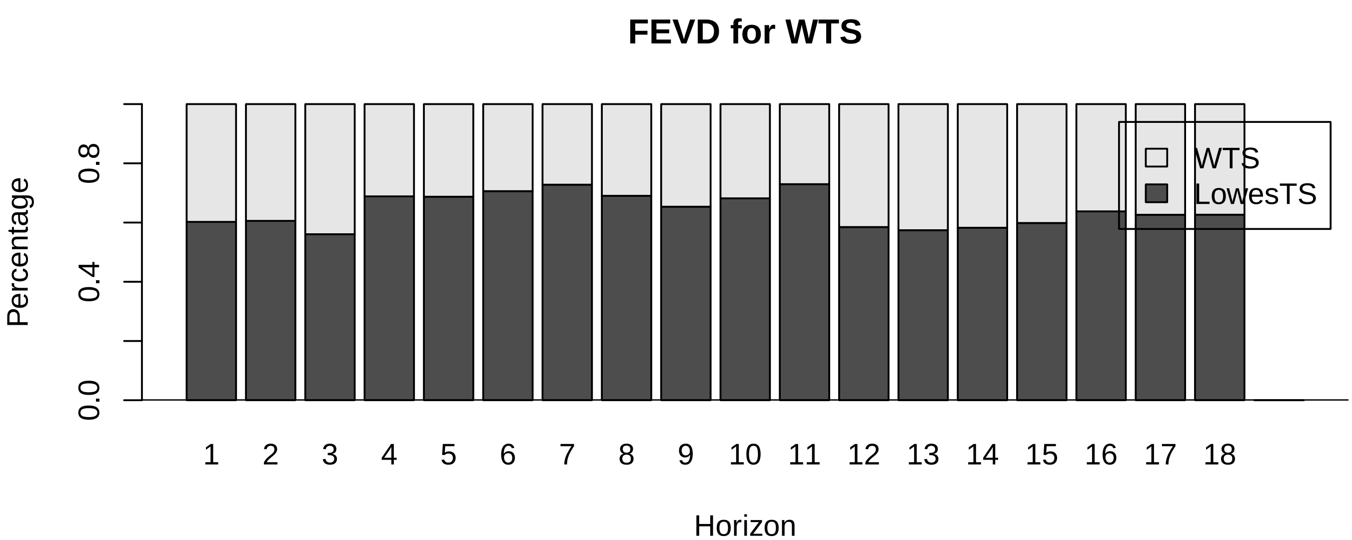




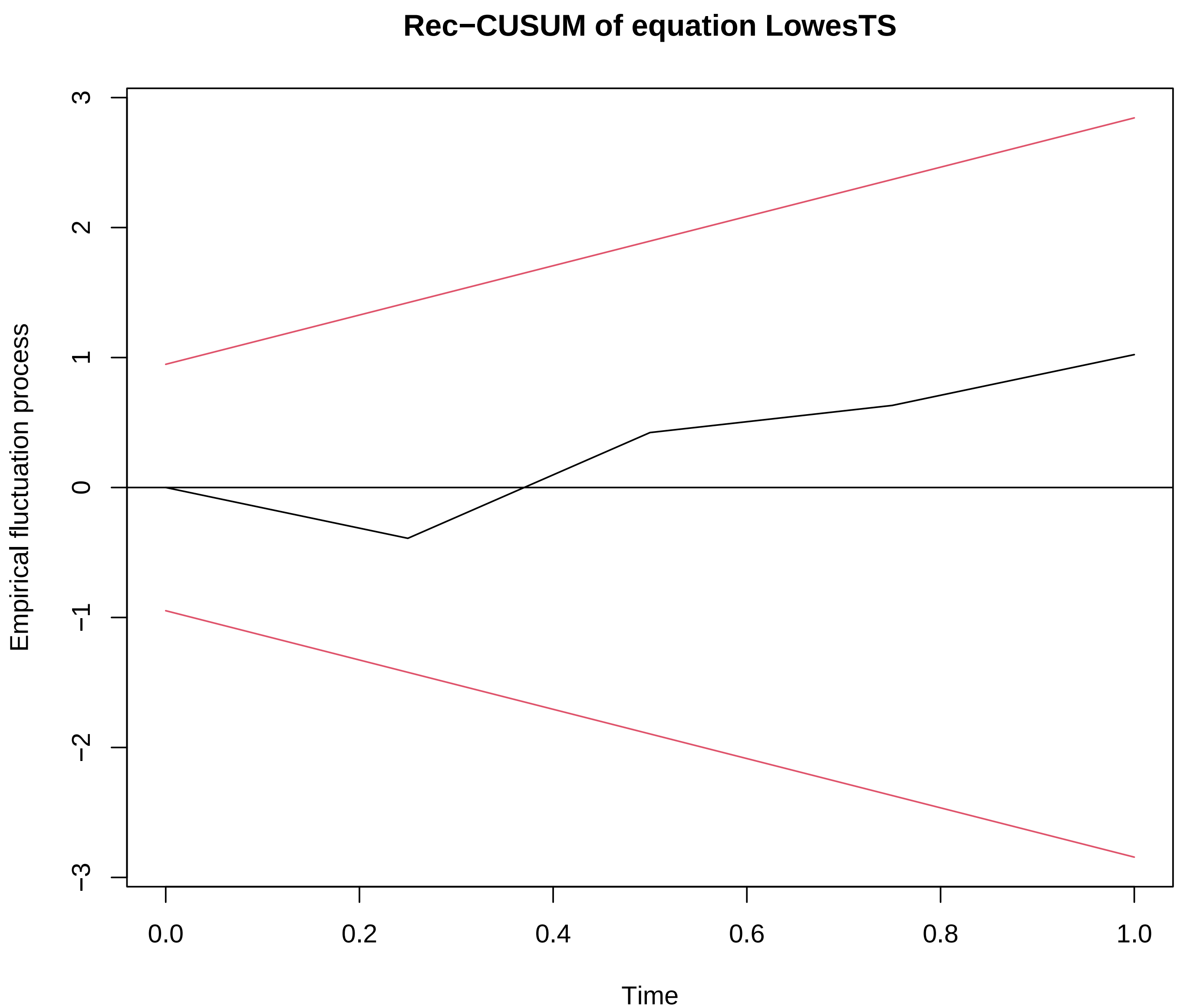

```
#FEVD
#For the FEVD plot for Lowes Time Series, in the initial first 5 periods,
#approximately 90% of the variation in Lowe's Percent Change in Stock Price
#is from shocks from Lowes itself and the remaining 10% comes from Walmart's
#Time Series. The contribution that Walmart has on Lowes variation changes
#slightly, about 5-10% after the 5th period and seems to converge around 80-85%.
#The system seems to become stable after about 15 periods.


#For the FEVD plot for Walmart's time series, in the initial first period,
#approximately 60% of the variation in Walmart's Percent change in stock price
#comes from shocks from Lowes and the remaining 40% comes from Walmart itself.
#The contribution that Lowes remains fairly high, across the 18 periods that we
#had, at 60-70% . In conclusion, we found that Lowes had bigger shocks on the
#variation in Lowes percent change in stocks than vice versa.
```

```r
plot(fevd(LowesW_model,n.ahead = 18))
```

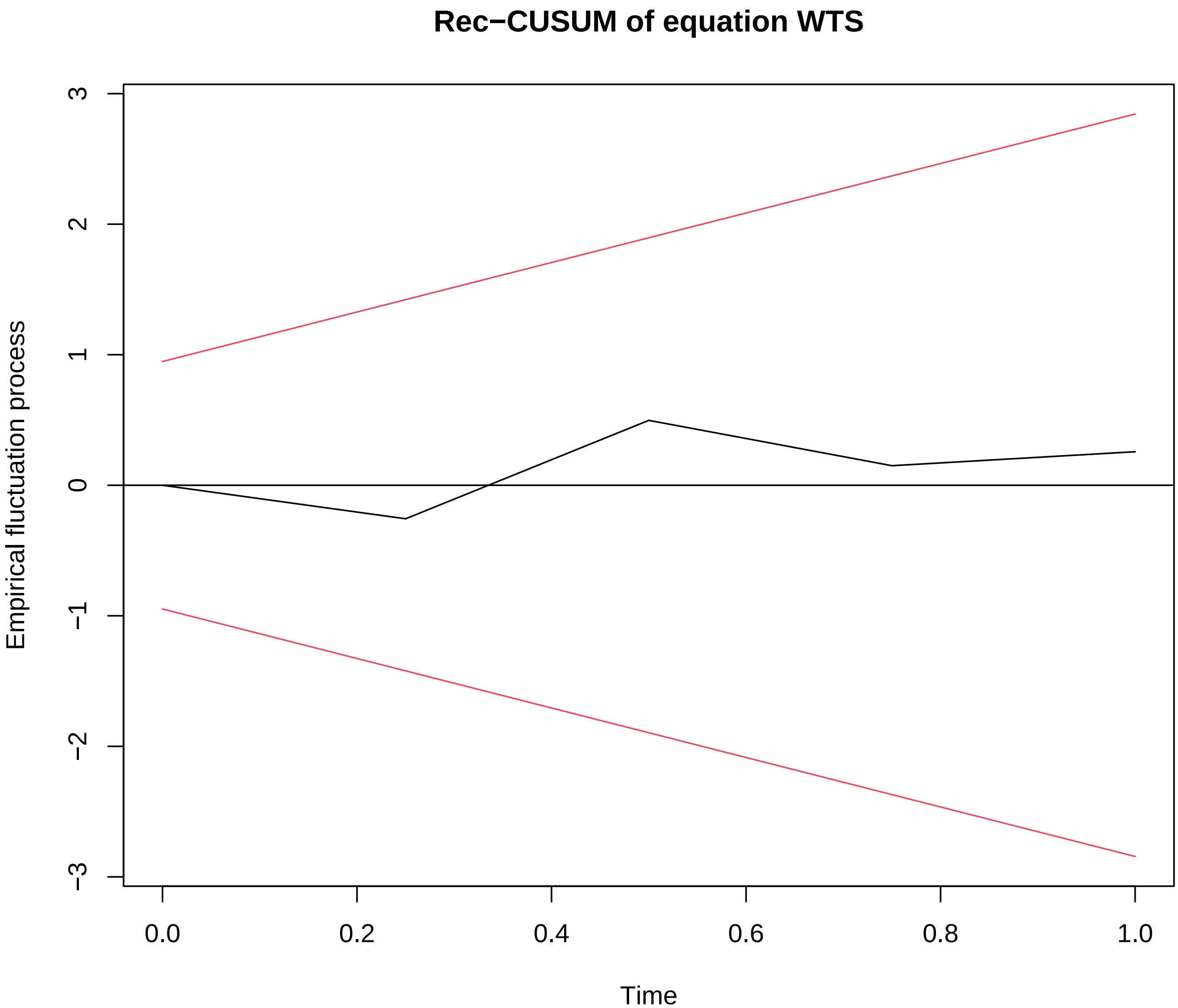





```r
#CUSUM
#Our CUSUM model for Lowes determined that there is no significant shifts that
#would alter a normal price change that has been predicted. We determined this
#since the plotted line stays within the border.
plot(stability(LowesW_model, type= "Rec-CUSUM"), plot.type="single")
```



Rec−CUSUM of equation LowesTS
Empirical fluctuation process
3
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
Time




```
#6
#Conclusion: In conclusion, there were a lot of key findings that we found
#through the many graphs, models and tests that we used. One finding that
#we think we can improve or maybe make our results look better is to switch
#our data from monthly to daily. This change might have been beneficial when
#looking at if there were more significant lags than the two that we had for
#the AR model. When we were looking at the AR models, specifically the lags
#that were shown in the ACF and PACF, the lags were barely considered lags.
#So when creating our new models, it was difficult to include more than one lag.
#Therefore, our AIC and BICs were different. On the other hand, according to
#plotting our residuals of the AR and ARDL models and the Breusch Godfrey Test,
#it showed that our forecast on the stock price changes were unbiased, and that
#there was no correlation between the residuals. When we created the VAR model
#and ran the VARselect function it resulted in one significant lag for a
#lag max of ten. We decided to increase the lag max to see if we could find
#more significant lags. However, when the lags are over ten, the significance
#of the lags are not as impactful and significant on the overall model, as
```

```
#Professor Rojas mentioned in lecture. Therefore, we prefer to use the Lowes
#AR model, and we would use the Walmart ARDL model, because of the lower AIC
#and BICs that they had in comparison to the other models, making them the
#better fits.
```