Project

2022-11-29

```r
library(forecast)
```

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

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(zoo)
```

```
##
## Attaching package: 'zoo'

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

```r
library(readxl)
library(astsa)
```

```
##
## Attaching package: 'astsa'

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

```
library(ggplot2)
```

```
cpi = read.csv("CPI(ALL ITEMS).csv")
cpi.gas = read.csv("cpi(gasoline).csv")
crudeoil = read.csv("crude-oil-price.csv")
gasoline.retail = read.csv("gasoline retail price.csv")
production = read.csv("OPEC_Crude_Oil.csv",header=FALSE)
```

Data Pre-processing

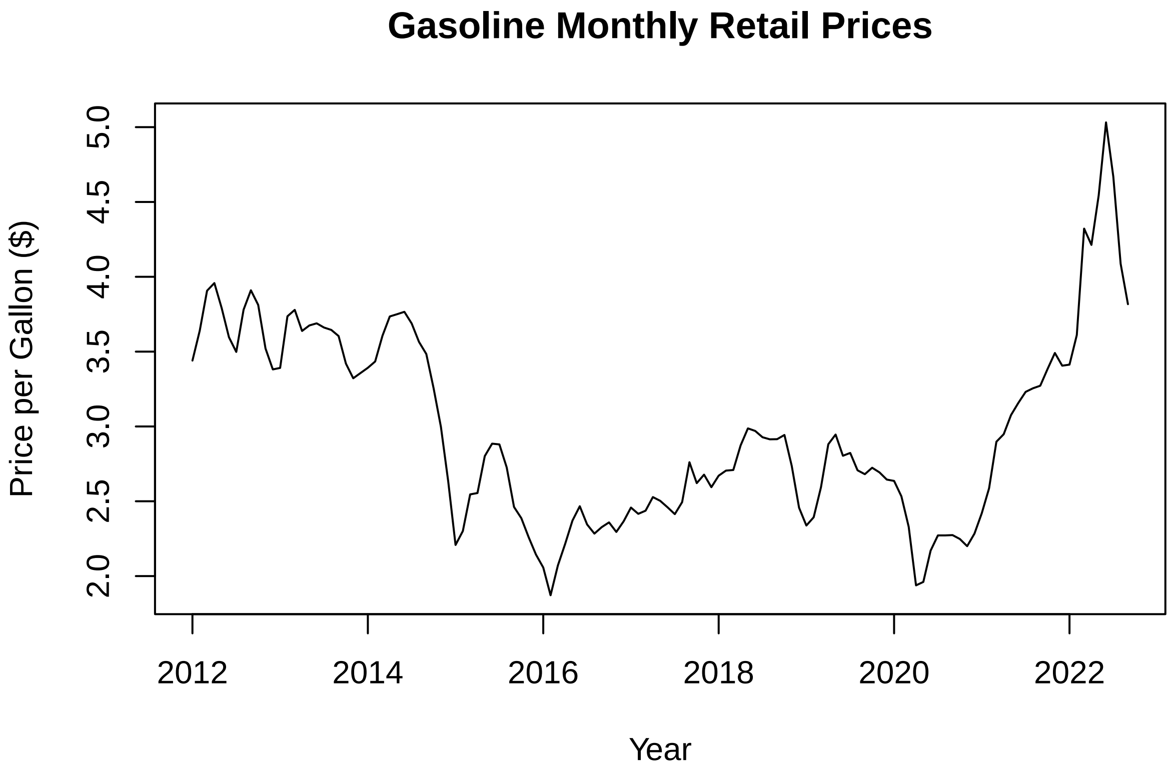
```
crudeoil$date <- as.Date(crudeoil$date)
```

Arrange from least recent to most recent

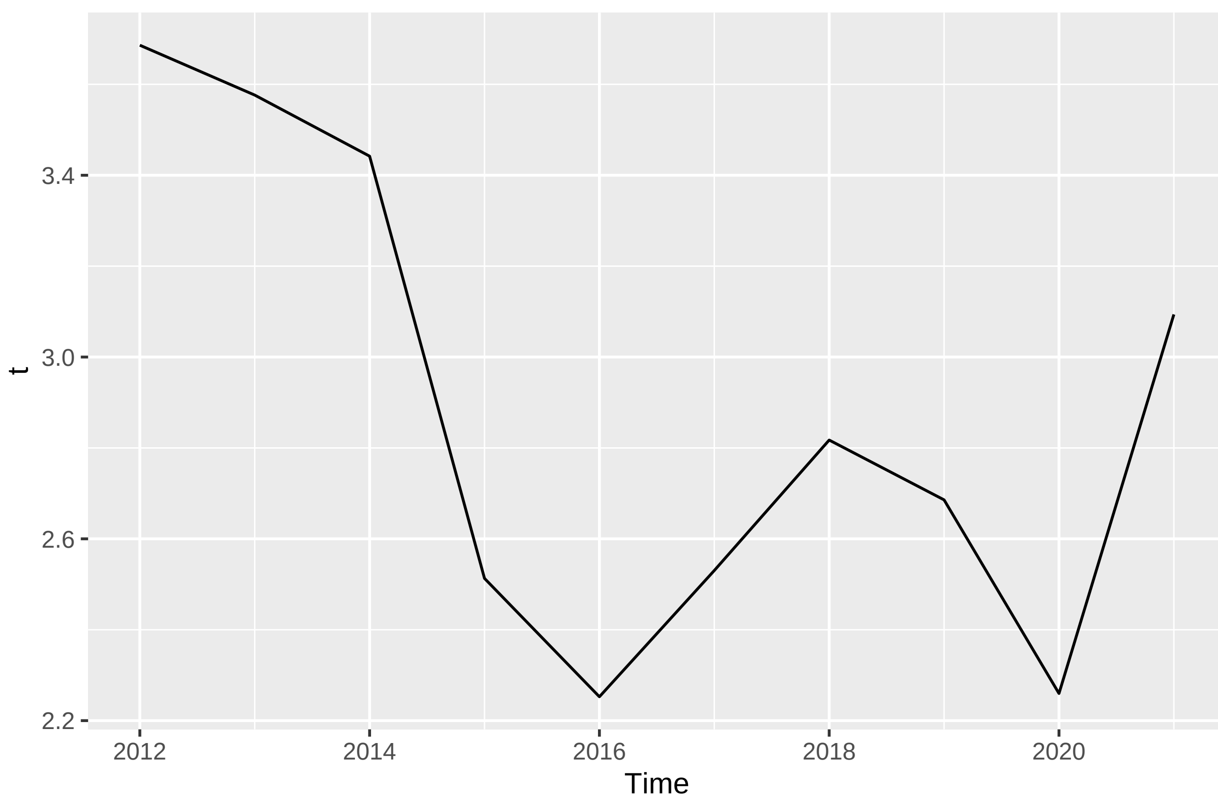
```
cpi.gas <- cpi.gas %>% arrange(mdy(cpi.gas$DATE))
crudeoil <- crudeoil[order(crudeoil$date),]
gasoline.retail <- gasoline.retail %>% arrange(my(gasoline.retail$Date))
```

```
cpi.ts = ts(cpi$CPIAUCSL, start=c(2012,1), end=c(2022,9),frequency = 12)
cpi.gas.ts = ts(cpi.gas$CUUR0000SETB01, start=c(2012,1), end=c(2022,9),frequency=12)
crudeoil.ts = ts(crudeoil$price.barrel.42.gal., start=c(2012,1), end=c(2022,9),frequency=12)
gasoline.ts = ts(gasoline.retail$U.S..All.Grades.All.Formulations.Retail.Gasoline.Prices..Dollars.per.Ga
production.ts = ts(production$V2,start=c(2012,1),  end=c(2022,9),frequency=12)
```

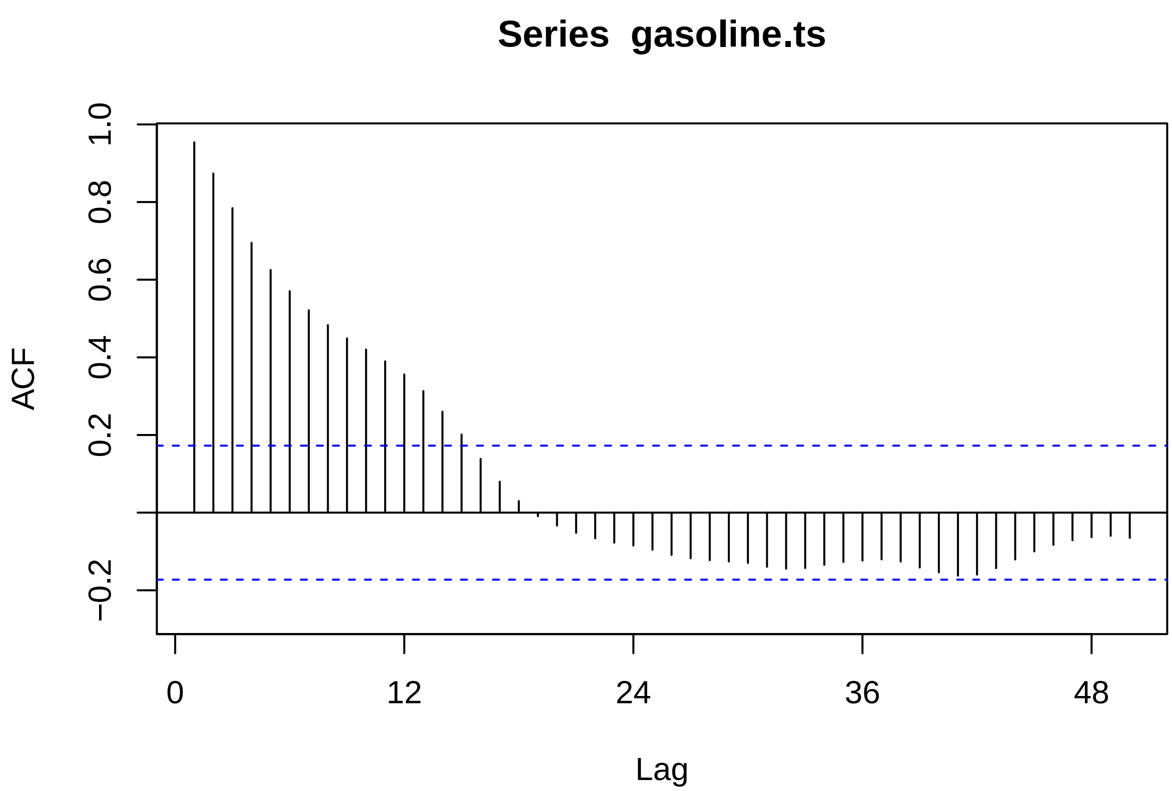
```
plot(gasoline.ts, ylab="Price per Gallon ($)", xlab = "Year", main = "Gasoline Monthly Retail Prices")
```



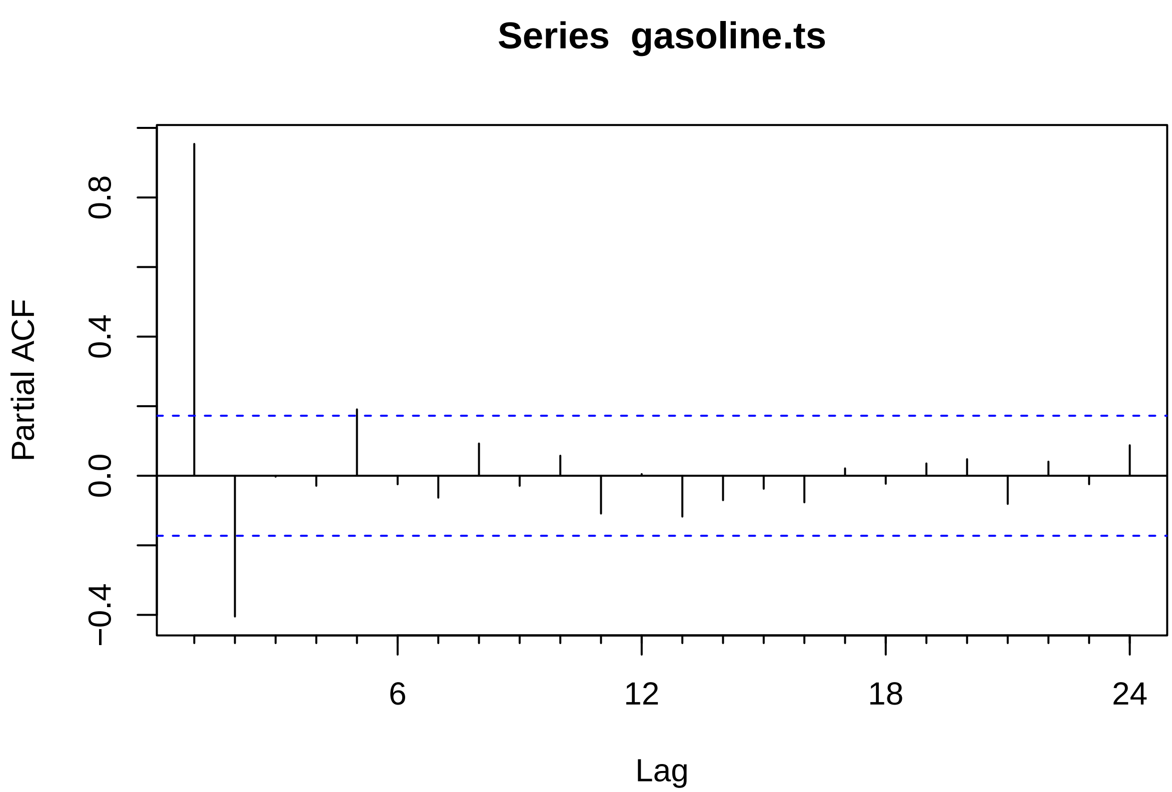

```
t = aggregate(gasoline.ts,FUN=mean)
autoplot(t)
```



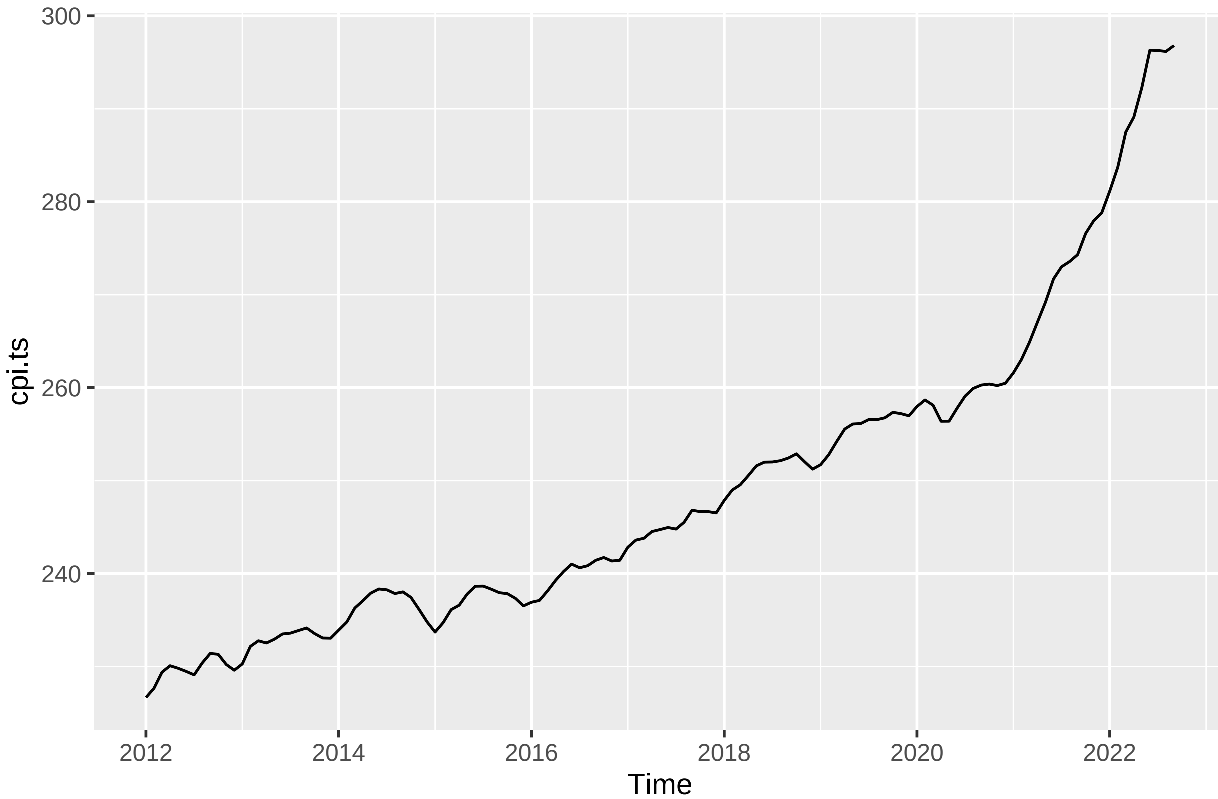
```r
Acf(gasoline.ts,50)
```



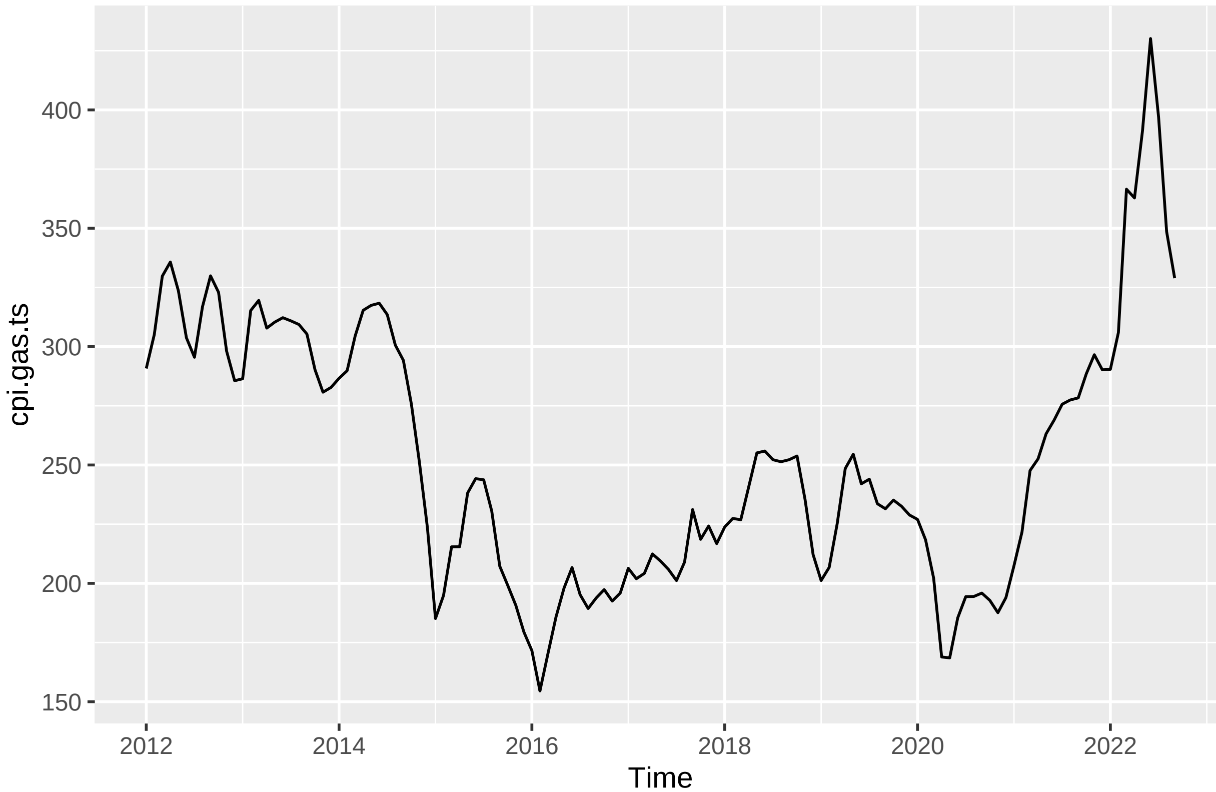

```r
Pacf(gasoline.ts)
```



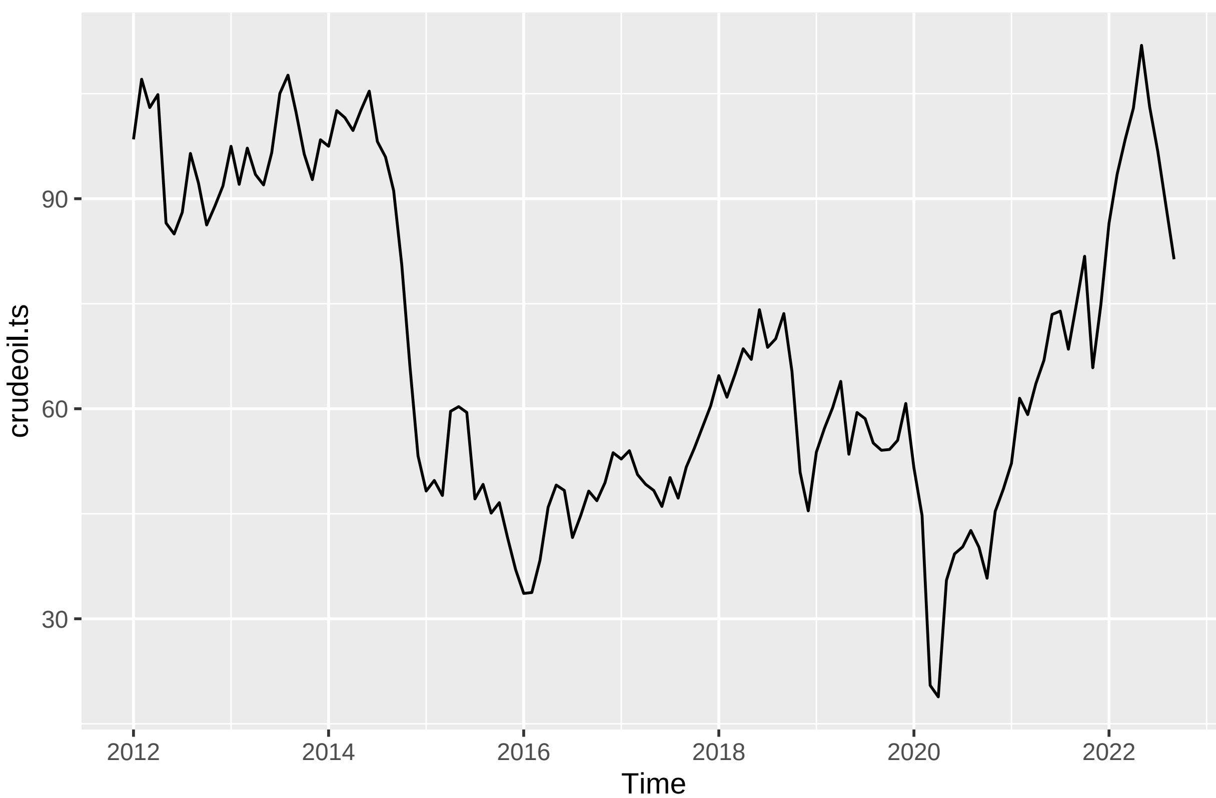
```
autoplot(cpi.ts)
```



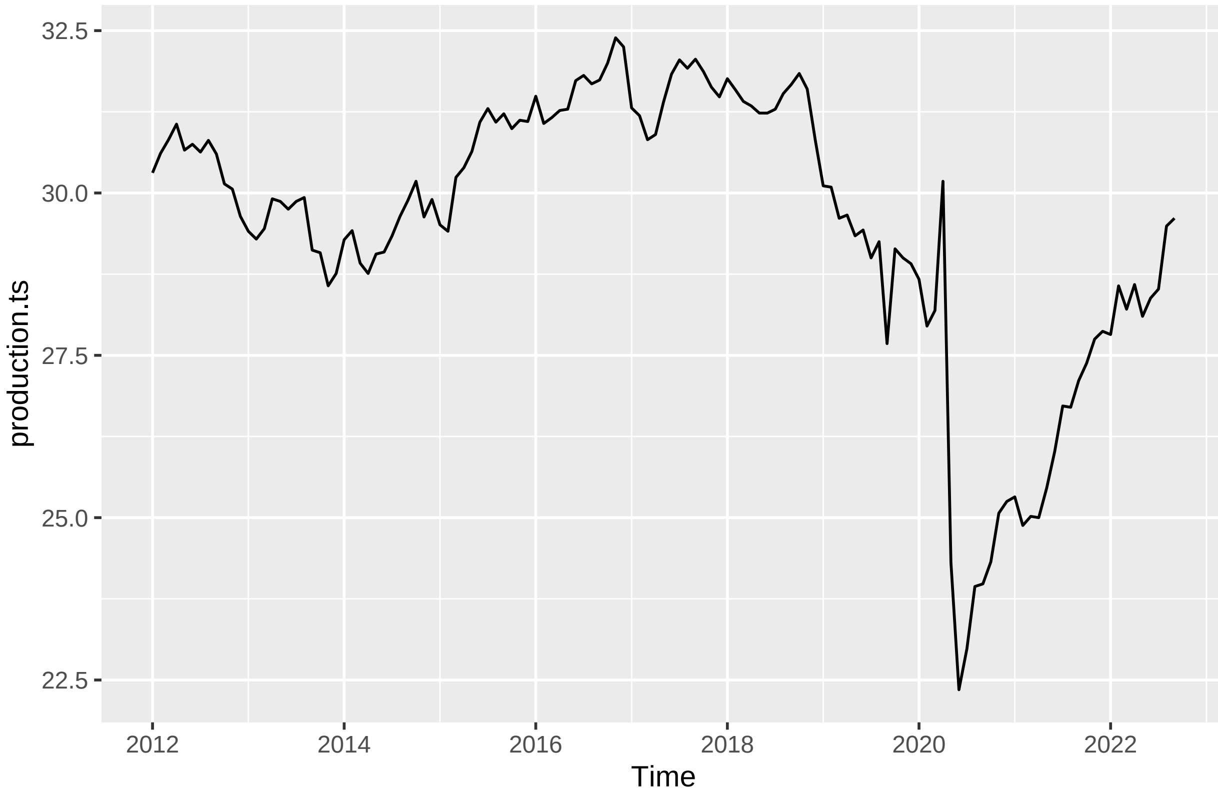

```
autoplot(cpi.gas.ts)
```




```
autoplot(crudeoil.ts)
```



```
autoplot(production.ts)
```

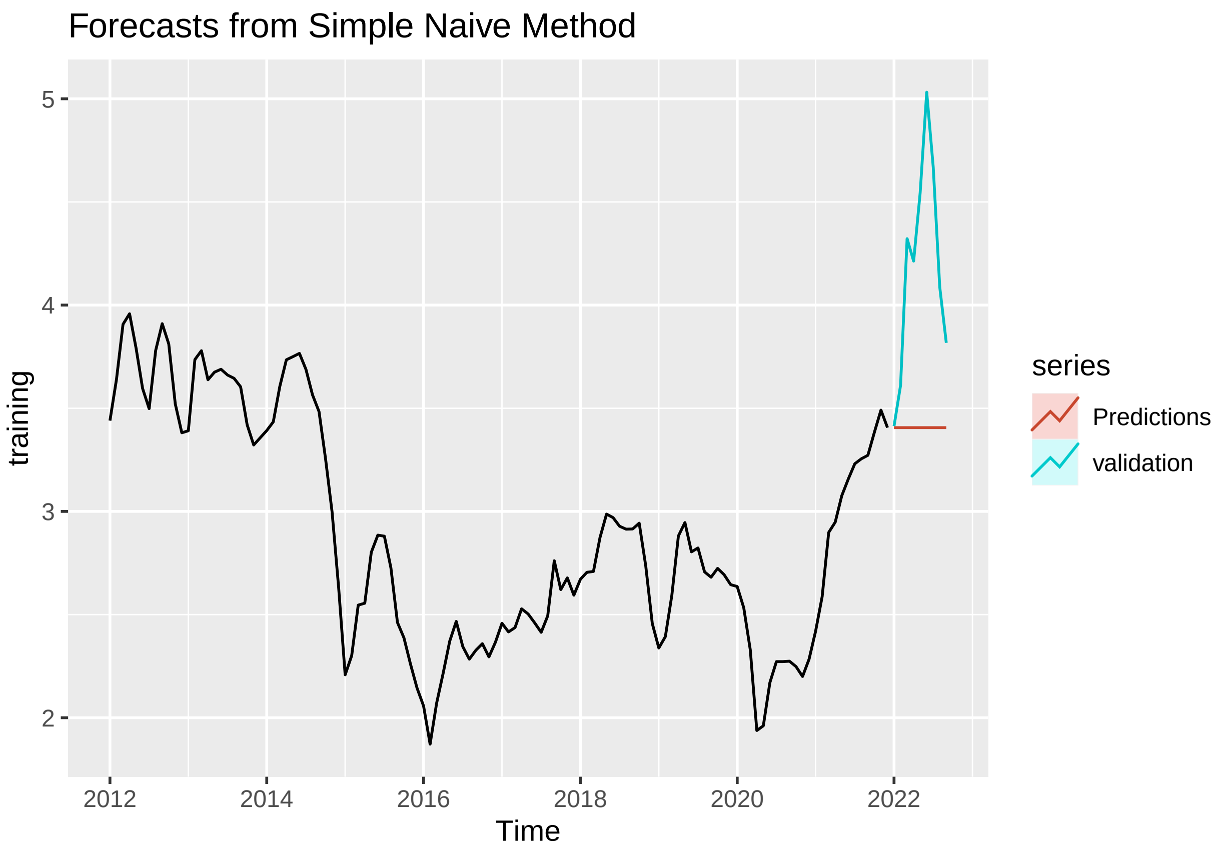



```
df1 = gasoline.ts
df2 = cpi.ts
df3 = cpi.gas.ts
df4 = crudeoil.ts
df5 = production.ts

for (i in 2:5) {
  df1 <- cbind(df1, eval(parse(text=paste("df", i, sep = ''))))
}

colnames(df1) <- c('Gasoline_Retail_Price','CPI(ALL)',
                   'CPI(Gas)',
                   'Crude_Oil_Price',
                   'Crude_Oil_Production')
df1 = as.data.frame(df1)
```

Simple Naive & Seasonal Naive

```
gasoline.ts <- ts(df1$Gasoline_Retail_Price, start=c(2012,1),end=c(2022,9),frequency = 12)

n=length(gasoline.ts)
stepsAhead = 9
nTraining= n-stepsAhead
```

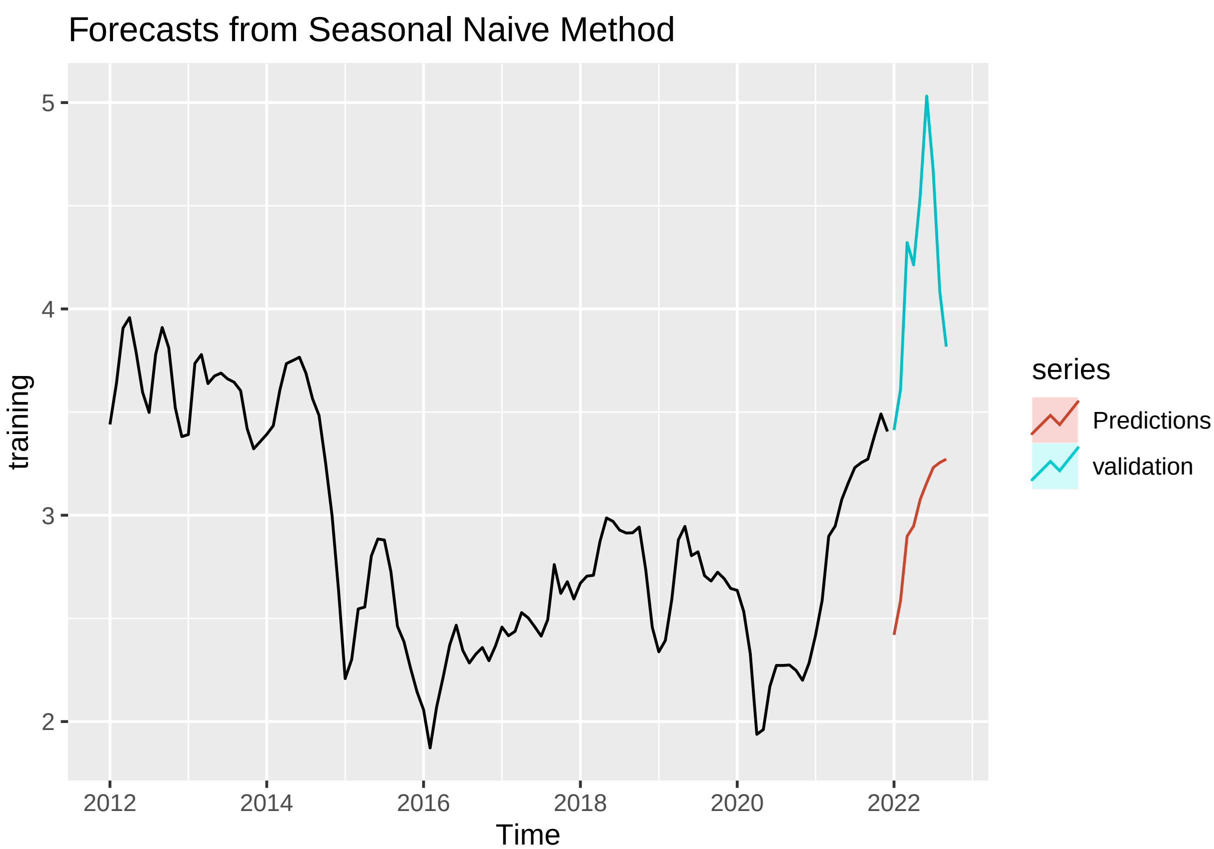
```r
training <- window(gasoline.ts, start=c(2012,1), end=c(2012,nTraining))
validation <- window(gasoline.ts, start=c(2012,nTraining+1), end=c(2012,nTraining+stepsAhead))

naive = naive(training, h= stepsAhead)

autoplot(training, main = "Forecasts from Simple Naive Method") + autolayer(validation) + autolayer(nai
```




```r
accuracy(naive, validation)
```

```
##                         ME      RMSE       MAE        MPE      MAPE      MASE
## Training set -0.0002857143 0.1447208 0.1106723 -0.1529087  4.018176 0.2689545
## Test set      0.7837777778 0.9248965 0.7837778 17.5633473 17.563347 1.9047276
##                   ACF1 Theil's U
## Training set 0.4451151        NA
## Test set     0.4955834   2.15743
```

```r
snaive = snaive(training, h= stepsAhead)
autoplot(training, main = "Forecasts from Seasonal Naive Method") + autolayer(validation) + autolayer(sn
```



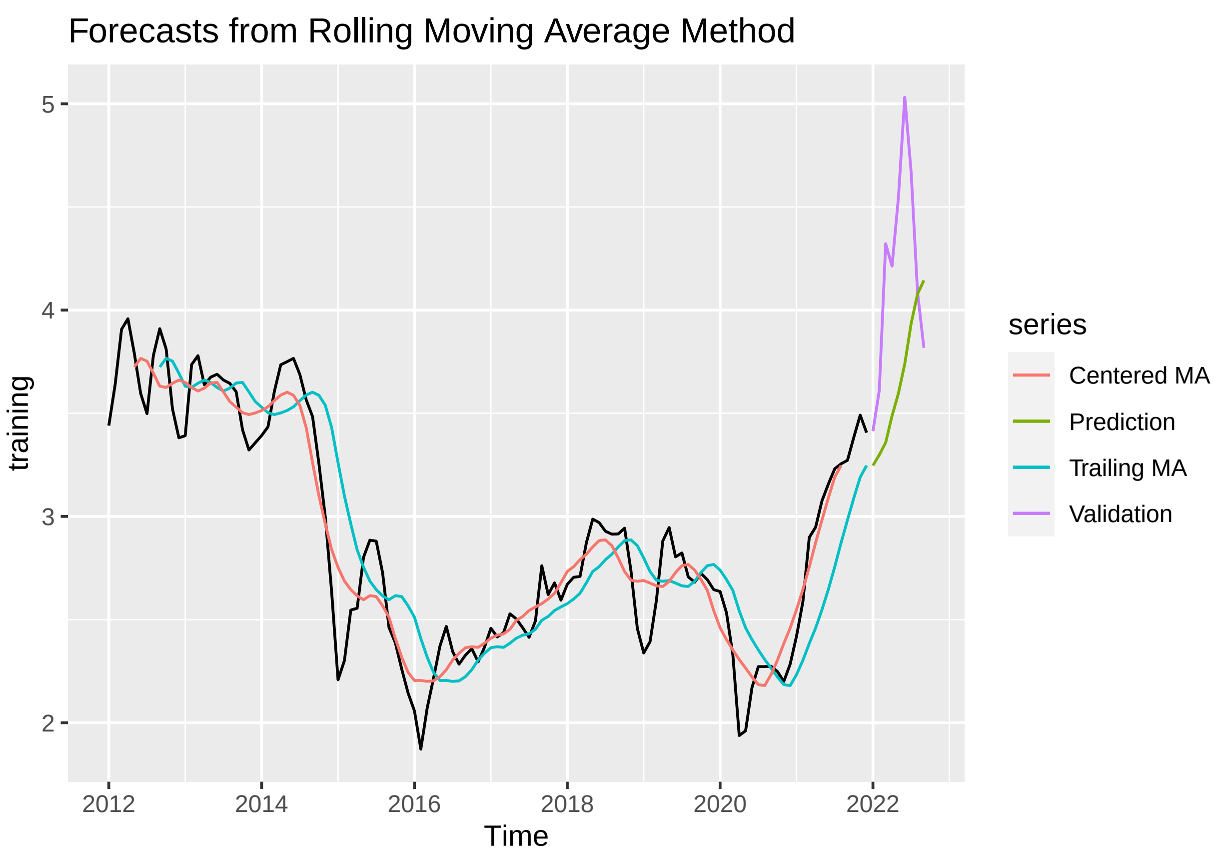

```
accuracy(snaive, validation)
```

```
##                        ME      RMSE       MAE       MPE     MAPE     MASE
## Training set -0.06582407 0.5361608 0.4114907 -3.963977 15.54517 1.000000
## Test set      1.20711111 1.2644907 1.2071111 28.380948 28.38095 2.933507
##                   ACF1 Theil's U
## Training set 0.9144277        NA
## Test set     0.4056736  2.968893
```

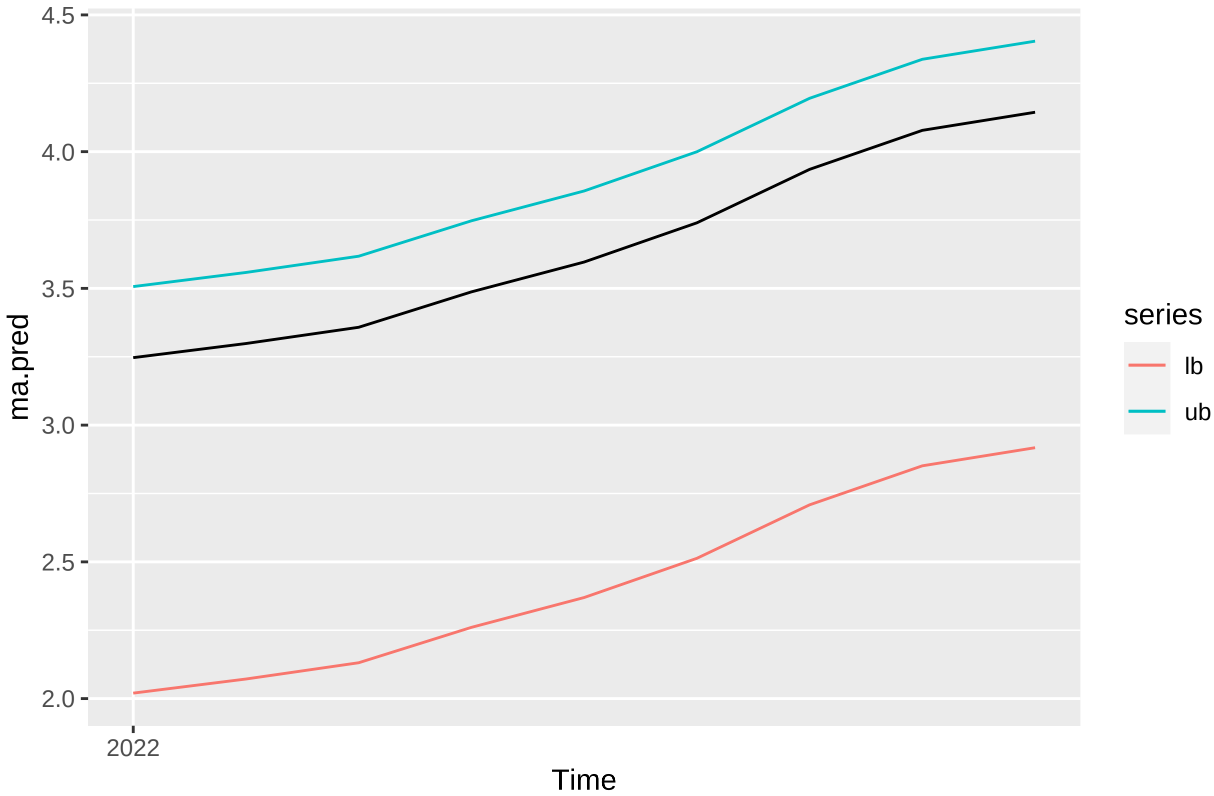
Moving Average Model

```
gasoline.ts <- ts(df1$Gasoline_Retail_Price, start=c(2012,1),end=c(2022,9),frequency = 12)

n = length(gasoline.ts)

w = 9
nValid = w

ma.trail.pred= rep(NA, nValid)

for (i in 1:nValid) {
  nTraining= n-nValid + (i - 1)
  training1=window(gasoline.ts, start=c(2012, 1),
                   end=c(2012, nTraining))
```

```
  ma.trailing.tmp = rollmean(training1, k=w, align="right")
  last.ma=tail(ma.trailing.tmp,1)

  ma.trail.pred[i] = last.ma
}

ma.pred= ts(ma.trail.pred,start=c(2012,length(gasoline.ts)-nValid+1), frequency = 12)

validation <- window(gasoline.ts, start=c(2022,1))

ma.trailing.right = rollmean(training, k=9, align="right")
ma.trailing.center = rollmean(training, k=9, align="center")

autoplot(training, main = "Forecasts from Rolling Moving Average Method") +
  autolayer(validation, series ="Validation")+autolayer(ma.pred, series = "Prediction") +
  autolayer(ma.trailing.right, series = "Trailing MA") +
  autolayer(ma.trailing.center, series = "Centered MA")
```




```
e <- ma.pred - validation

m <- quantile(e, probs = c(0.025,0.975))
lb=ma.pred+m[1]
ub=ma.pred+m[2]

autoplot(ma.pred) +
  autolayer(lb)+ autolayer(ub)
```




```
accuracy(ma.trailing.right, training)
```

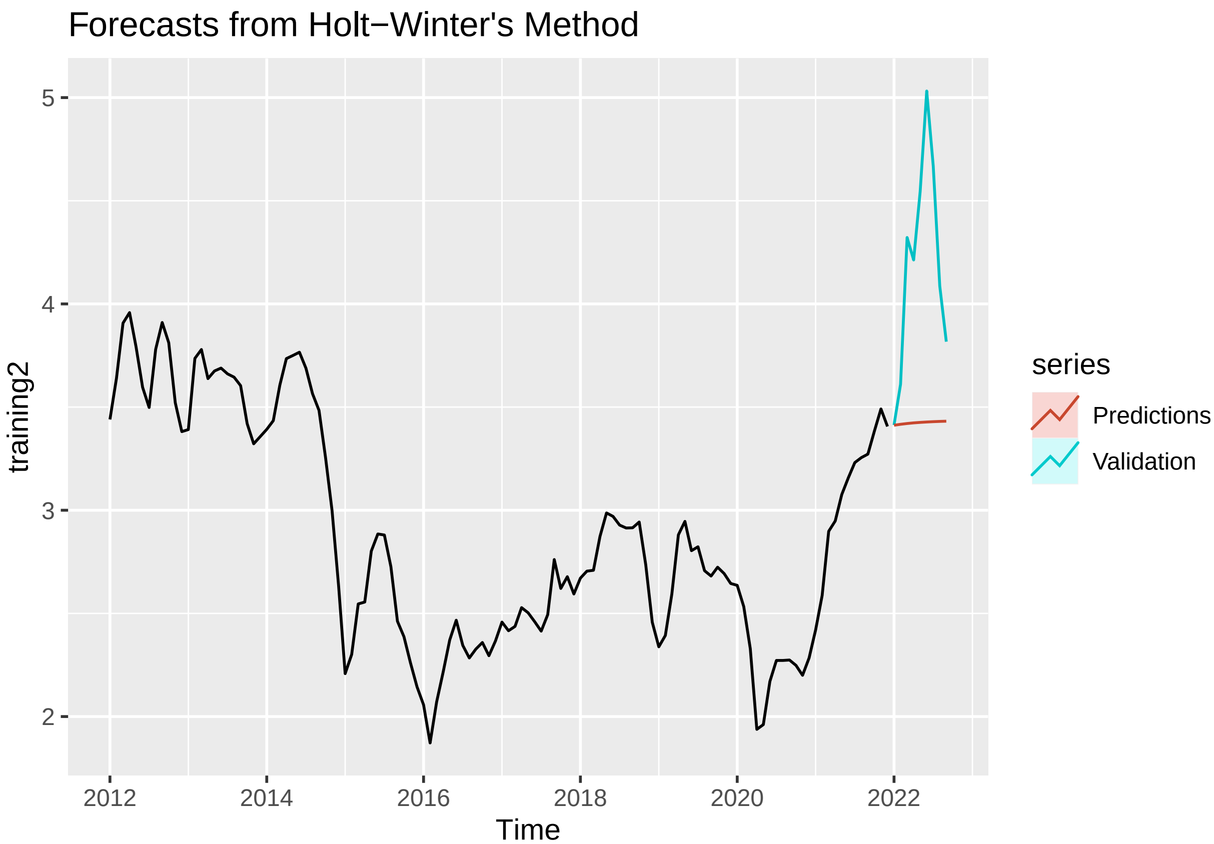
```
##                   ME      RMSE       MAE       MPE     MAPE      ACF1 Theil's U
## Test set -0.01297421 0.2771212 0.2055516 -1.234417 7.744419 0.8731434  2.016943
```

```
accuracy(ma.pred, validation)
```

```
##                 ME     RMSE       MAE      MPE     MAPE      ACF1 Theil's U
## Test set 0.5360864 0.729996 0.6087778 11.88611 13.79052 0.4337649  1.766839
```

Holt-Winter's model

```
gasoline.ts <- ts(df1$Gasoline_Retail_Price, start=c(2012,1),end=c(2022,9),frequency = 12)

n=length(gasoline.ts)
stepsAhead = 9

nTraining=n-stepsAhead
training2 <- window(gasoline.ts, start=c(2012,1), end=c(2012,nTraining))

zzz=ets(training2,model='ZZZ')
zzz.pred=forecast(zzz,h=stepsAhead,level=0)
autoplot(training2, main = "Forecasts from Holt-Winter's Method")+autolayer(validation, series = "Valid
```

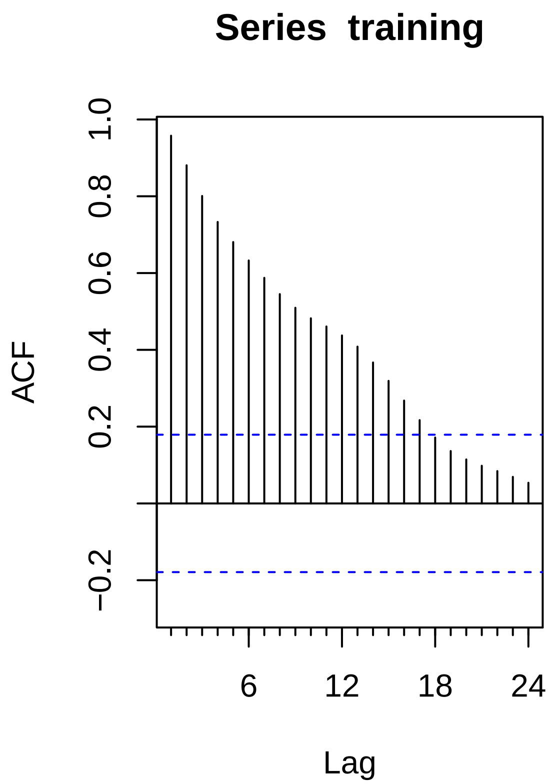


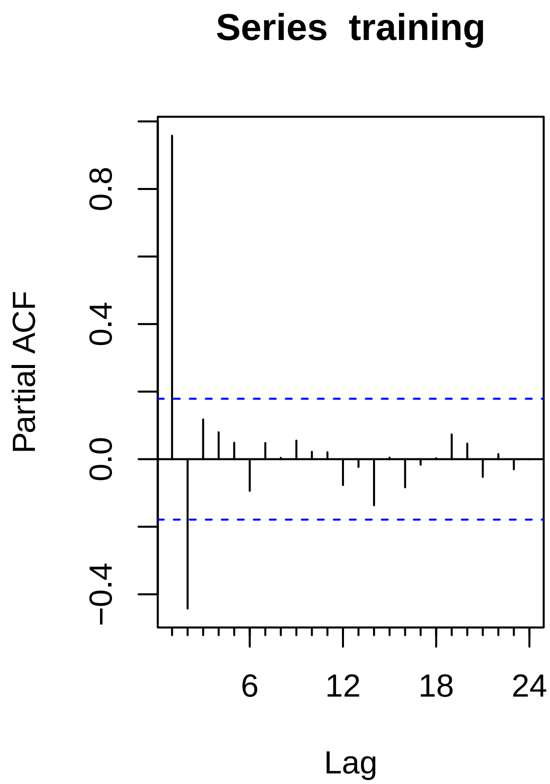

```
accuracy(zzz.pred,validation)
```

```
##                         ME      RMSE       MAE         MPE      MAPE      MASE
## Training set -0.002509267 0.1388332 0.1057644 -0.07218818  3.850549 0.2570273
## Test set      0.765629507 0.9076277 0.7656295 17.13538435 17.135384 1.8606239
##                   ACF1 Theil's U
## Training set 0.2896091        NA
## Test set     0.4928217  2.116923
```

ARIMA Model

```
par(mfrow=c(1,2))
Acf(training)
Pacf(training)
```






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

```
auto.arima(gasoline.ts)
```

```
## Series: gasoline.ts
## ARIMA(0,1,1)(1,0,0)[12]
##
## Coefficients:
##          ma1    sar1
##       0.4781  0.1935
## s.e.  0.0814  0.1125
##
## sigma^2 = 0.02482:  log likelihood = 55.59
## AIC=-105.17   AICc=-104.98   BIC=-96.62
```

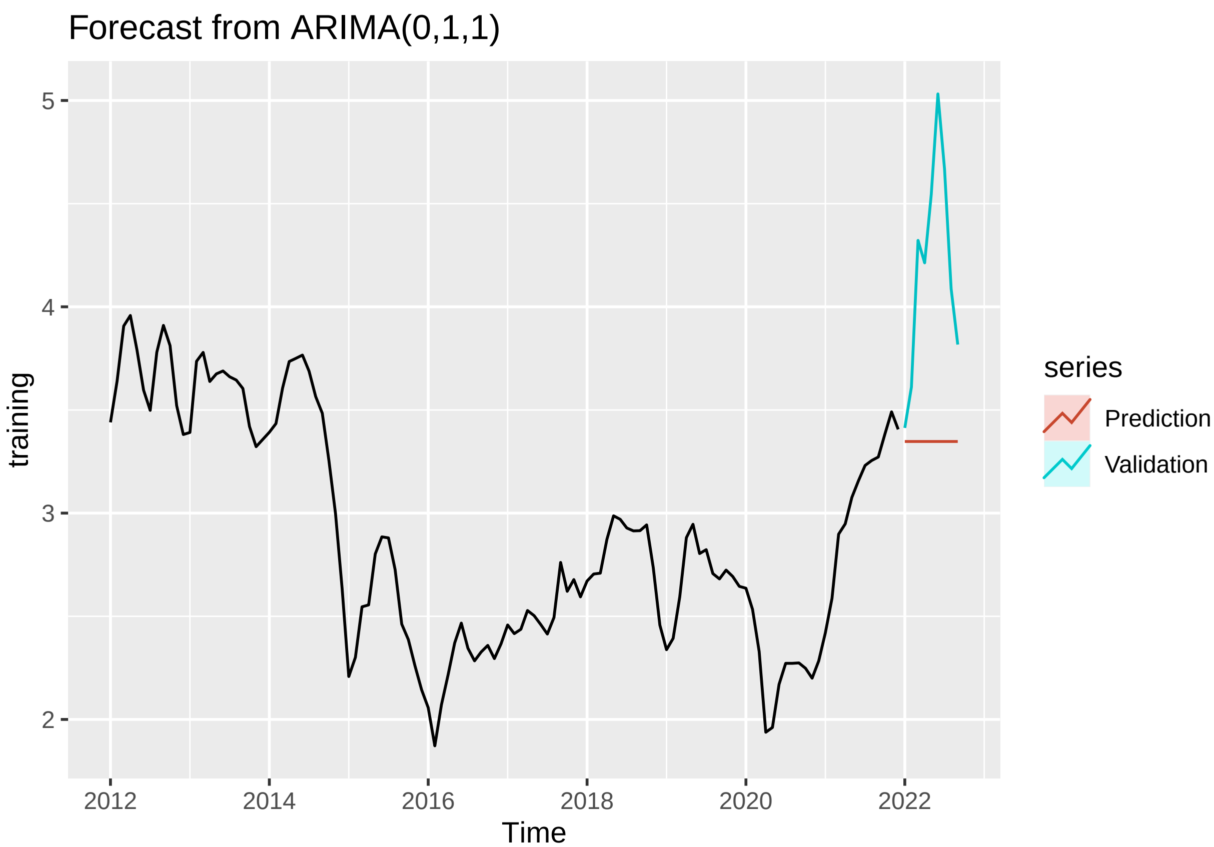
```
m1 = arima(training,order=c(0,1,1),seasonal=list(order=c(1,0,0),period=12))
m1.pred = forecast(m1,h=9)
accuracy(m1.pred, validation)
```

```
##                         ME      RMSE        MAE         MPE      MAPE     MASE
## Training set -0.0002289429 0.1260164 0.09519429 -0.05755609  3.483083 0.231340
## Test set      0.8055312476 0.9381094 0.80553125 18.12202008 18.122020 1.957593
##                    ACF1 Theil's U
## Training set 0.03154213        NA
## Test set     0.49005360  2.192487
```

```r
m2 = arima(training,order=c(0,1,1),seasonal=list(order=c(0,0,0),period=12))
m2.pred = forecast(m2,h=9)
accuracy(m2.pred, validation)
```

```
##                         ME      RMSE        MAE         MPE     MAPE      MASE
## Training set -0.0005597263 0.1262053 0.09554792 -0.07101427  3.49596 0.2321994
## Test set      0.8425950930 0.9752397 0.84259509 18.98692411 18.98692 2.0476648
##                    ACF1 Theil's U
## Training set 0.03531323        NA
## Test set     0.49558335  2.279334
```

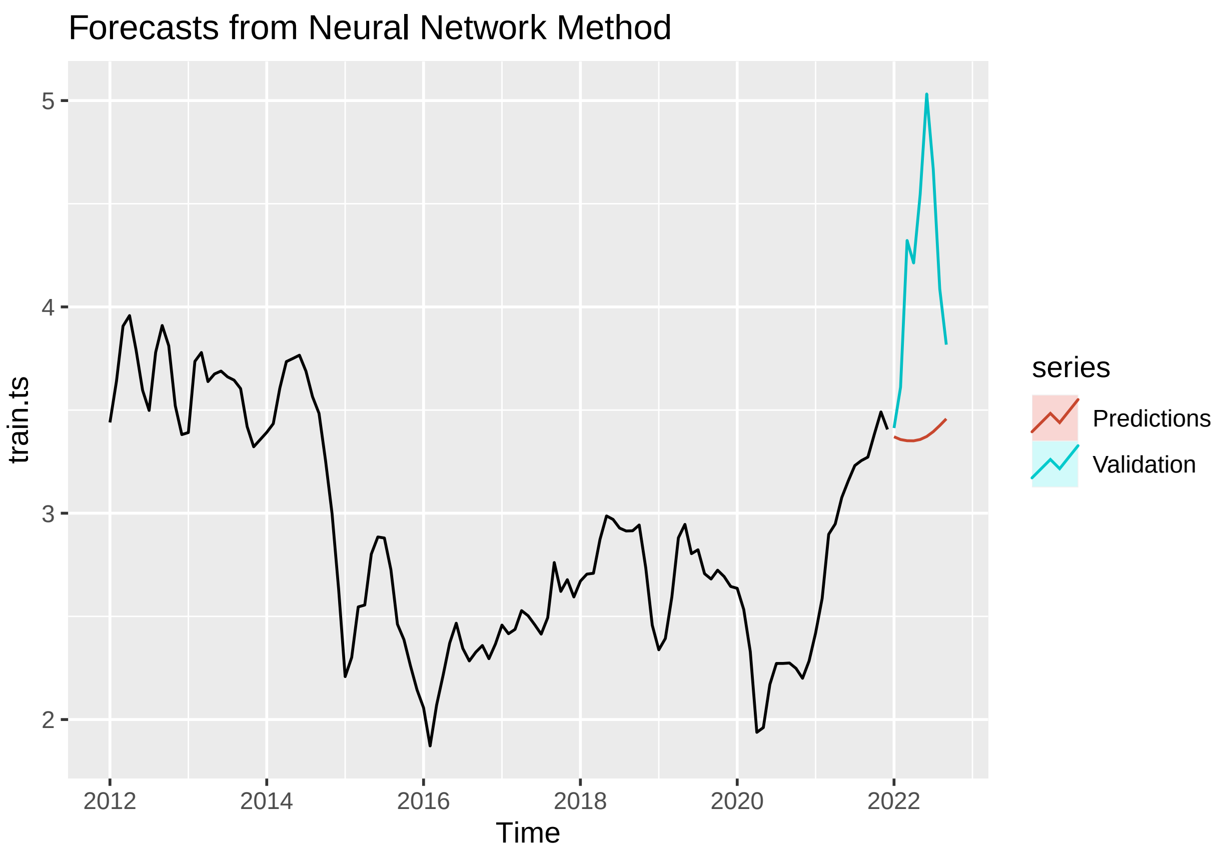
```r
autoplot(training, main = "Forecast from ARIMA(0,1,1)") +
  autolayer(validation, series="Validation")+
  autolayer(m2.pred, series="Prediction",PI=FALSE)
```



Neural Network

```r
gasoline.ts <- ts(df1$Gasoline_Retail_Price, start=c(2012,1),end=c(2022,9),frequency = 12)

nValid = 9
nTrain <- length(gasoline.ts) - nValid
train.ts <- window(gasoline.ts, start = c(2012, 1), end = c(2012, nTrain))
valid.ts <- window(gasoline.ts, start = c(2012, nTrain + 1))
```

```r
gasolineprice.nnetar.opt <- nnetar(train.ts)

gasolineprice.nnetar.opt.pred <- forecast(gasolineprice.nnetar.opt, h = 9)

autoplot(train.ts, main = "Forecasts from Neural Network Method") +
  autolayer(valid.ts, series = "Validation") +
  autolayer(gasolineprice.nnetar.opt.pred, series = "Predictions")
```

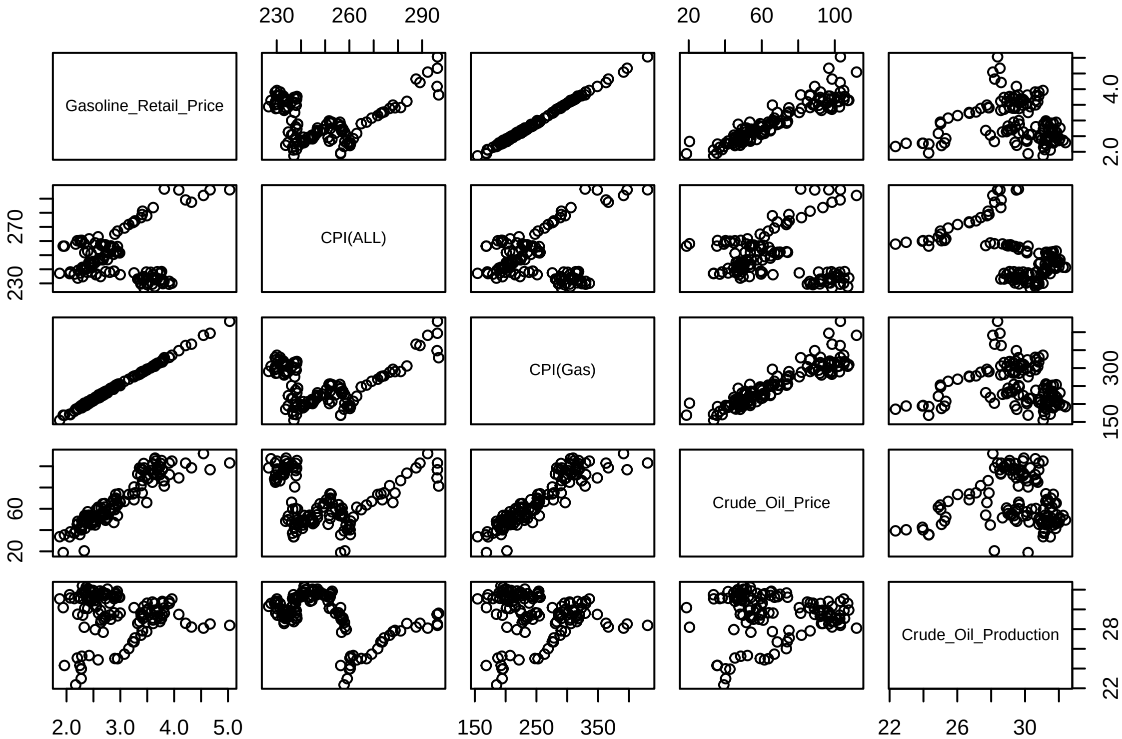



```r
accuracy(gasolineprice.nnetar.opt.pred, valid.ts)
```

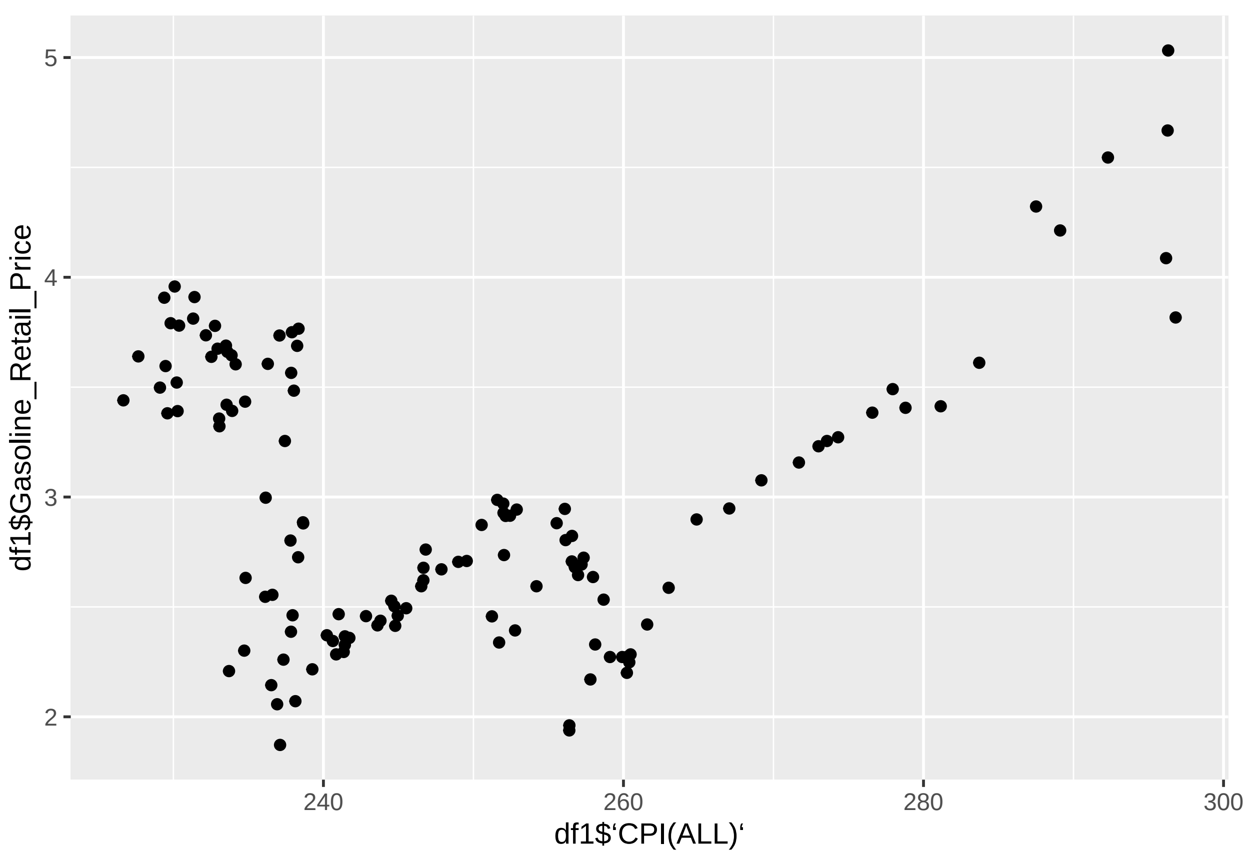
```
##                         ME      RMSE        MAE        MPE      MAPE      MASE
## Training set -3.203377e-05 0.1075198 0.08402676 -0.1810773  3.175958 0.2042008
## Test set      8.079017e-01 0.9490161 0.80790165 18.1332493 18.133249 1.9633532
##                    ACF1 Theil's U
## Training set 0.04612772        NA
## Test set     0.48438110  2.221744
```

Regression with external factors

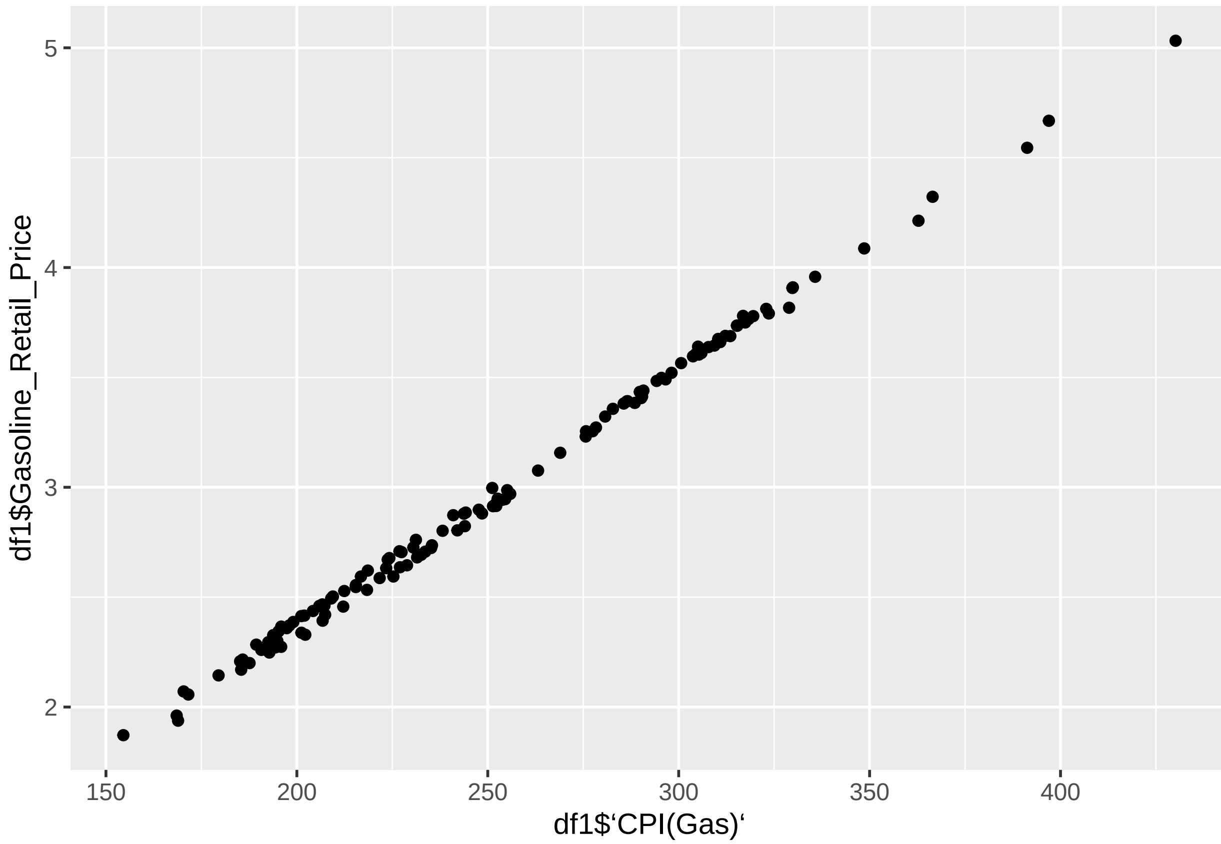
```r
pairs(df1)
```



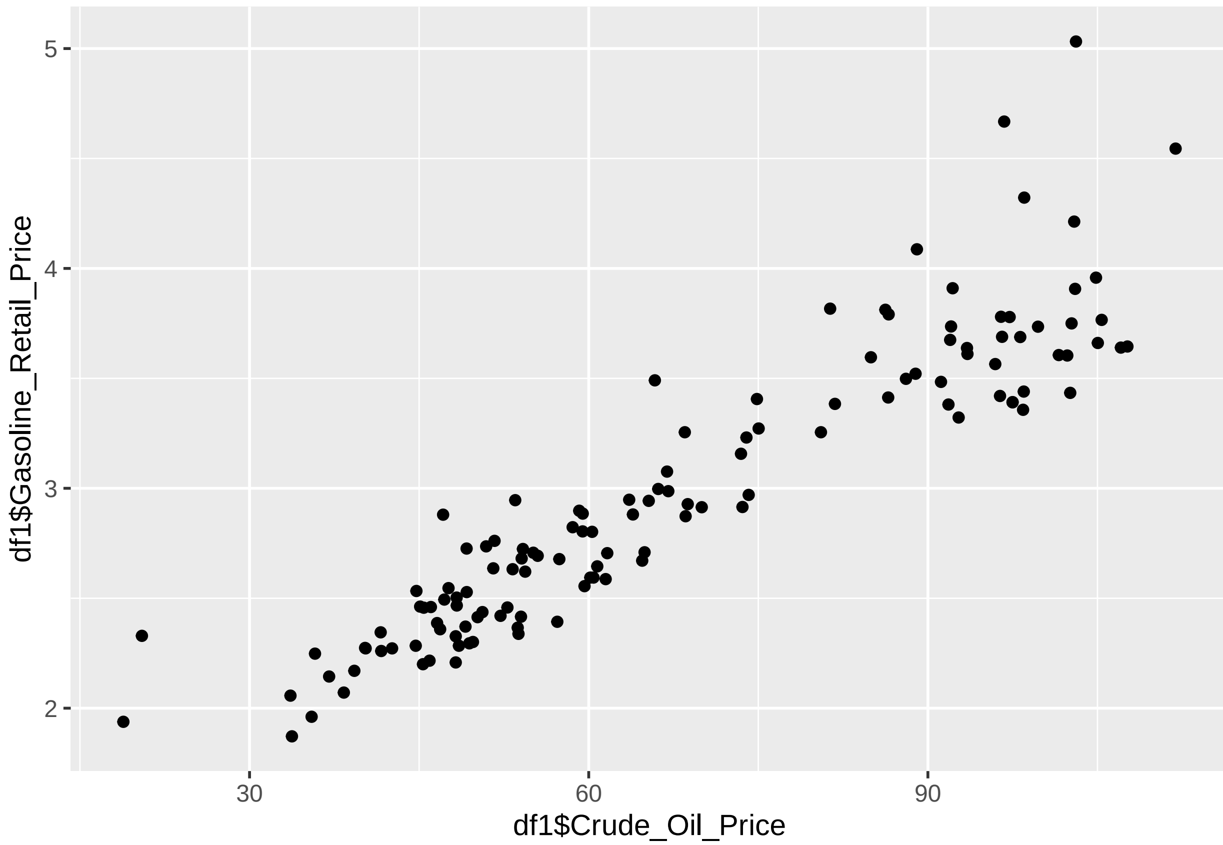

```
qplot(df1$`CPI(ALL)`,df1$Gasoline_Retail_Price)
```



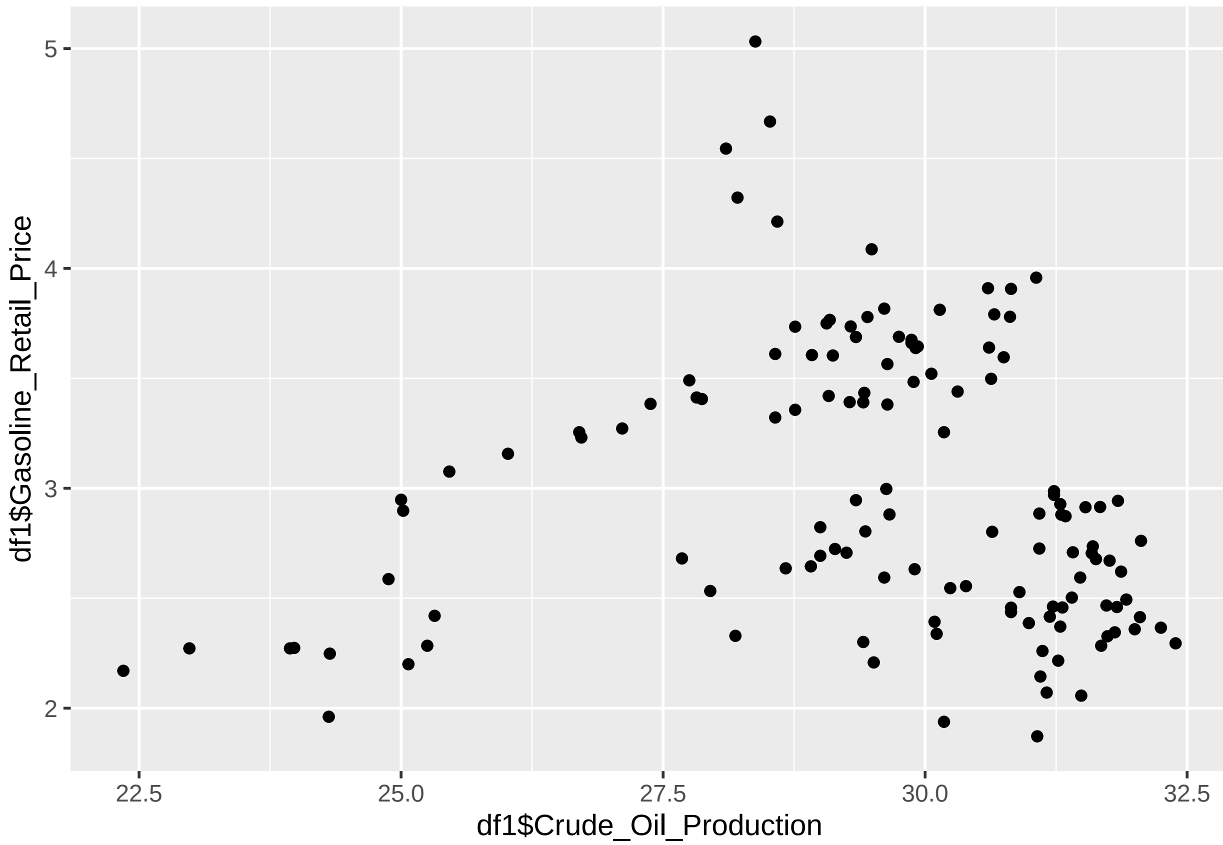
```
qplot(df1$`CPI(Gas)`,df1$Gasoline_Retail_Price)
```




```
qplot(df1$Crude_Oil_Price, df1$Gasoline_Retail_Price)
```

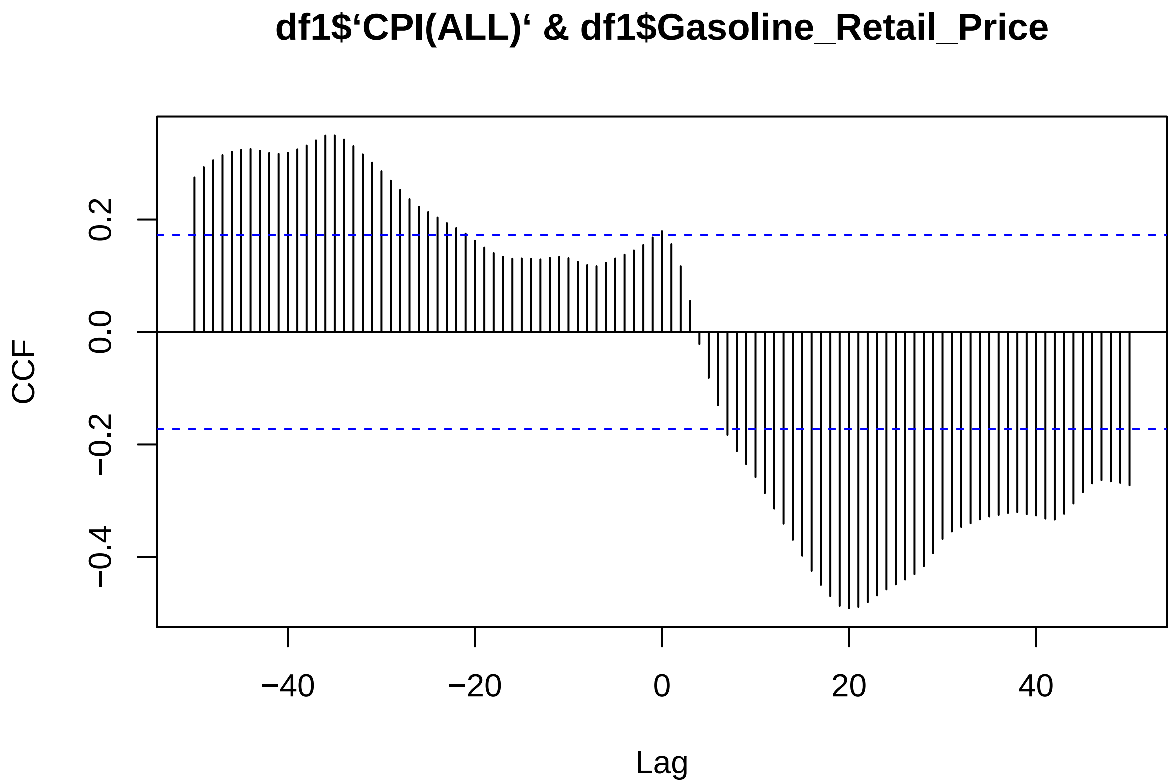



```
qplot(df1$Crude_Oil_Production,df1$Gasoline_Retail_Price)
```

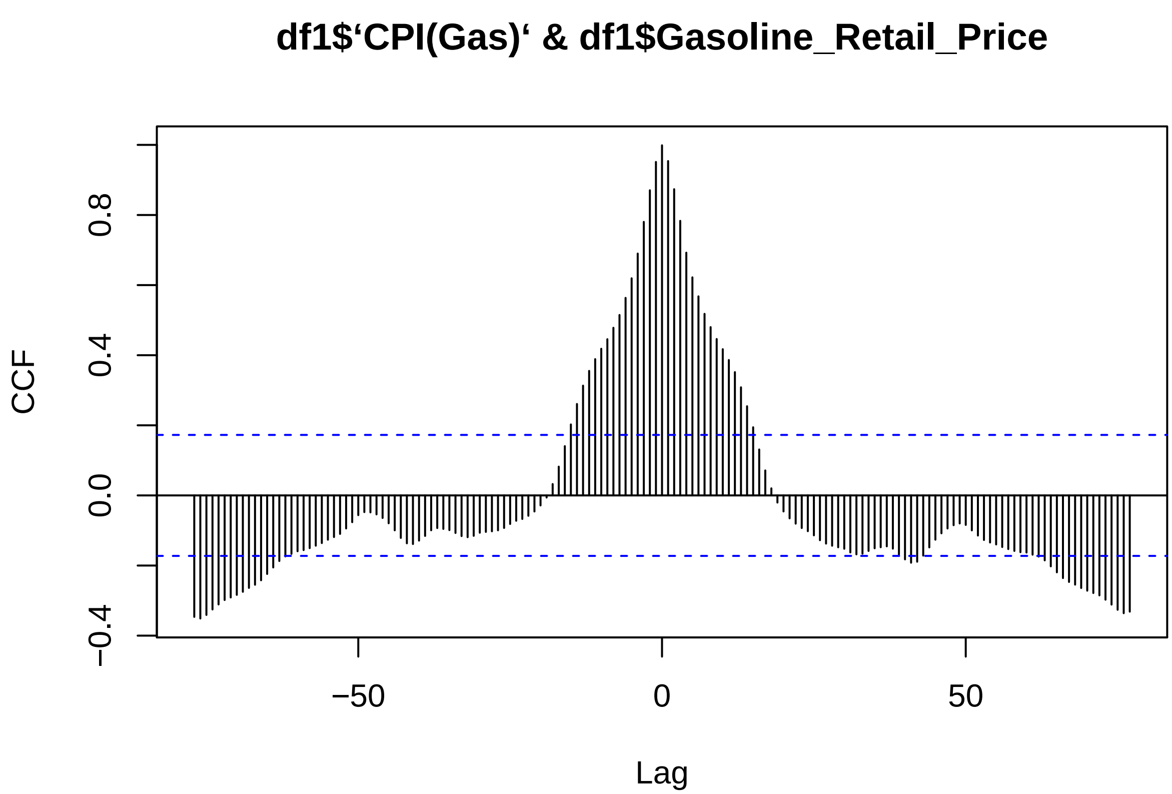


Study the cross-correlations between the each variable and gasoline retail price

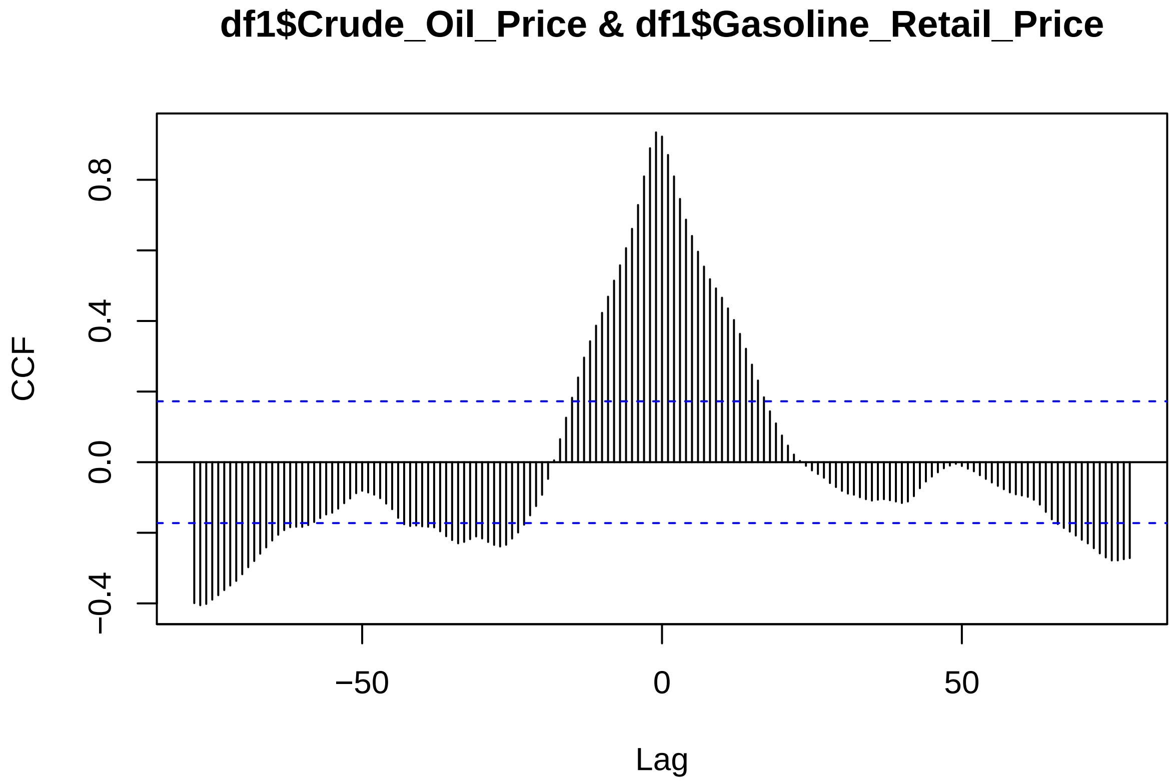
```
Ccf(df1$`CPI(ALL)`,df1$Gasoline_Retail_Price, 50)
```



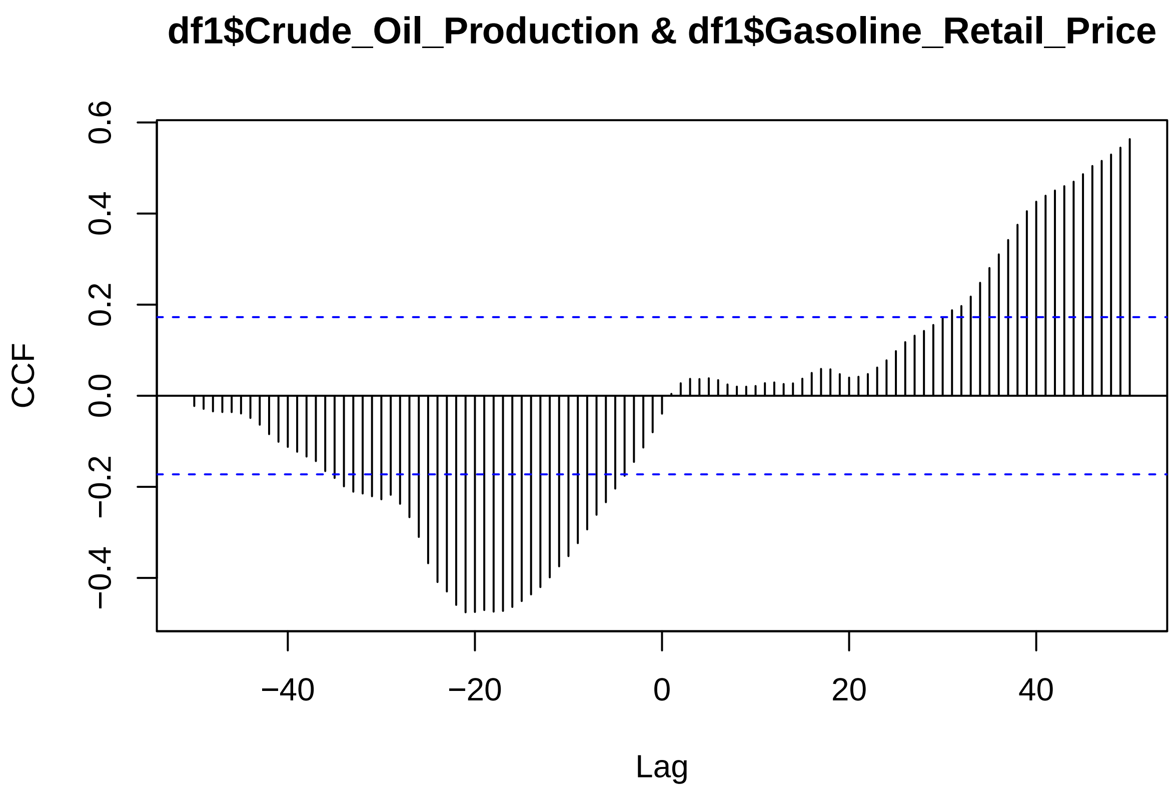

```
Ccf(df1$`CPI(Gas)`,df1$Gasoline_Retail_Price, 77)
```



```r
Ccf(df1$Crude_Oil_Price, df1$Gasoline_Retail_Price, 78)
```




```r
Ccf(df1$Crude_Oil_Production,df1$Gasoline_Retail_Price, 50)
```



1. CPI(ALL) 34 months ago affect the gasoline retail price now
2. CPI(GAS) 76 months ago affect the gasoline retail price now
3. Spot price of crude oil contracts 77 months ago affect the gasoline retail price now
4. Crude oil production 20 months ago affect the gasoline retail price now

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8     v purrr   0.3.4
## v tidyr   1.2.1     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
lag <- stats::lag

newdata <- ts.intersect(retail_price = gasoline.ts,
                        leadcpi=lag(cpi.ts,-34),
                        leadcpigas=lag(cpi.gas.ts,-76),
                        leadcrudepri=lag(crudeoil.ts,-77),
                        leadcrudepro=lag(production.ts,-20))


m3 = tslm(retail_price~leadcpi + leadcpigas + leadcrudepri + leadcrudepro,data=newdata)
summary(m3)
```

```
##
## Call:
## tslm(formula = retail_price ~ leadcpi + leadcpigas + leadcrudepri +
##     leadcrudepro, data = newdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62154 -0.17550  0.05613  0.19456  0.74604
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  21.872333   4.985988   4.387 6.45e-05 ***
## leadcpi      -0.052004   0.016643  -3.125  0.00305 **
## leadcpigas   -0.004140   0.003710  -1.116  0.27005
## leadcrudepri -0.015050   0.006668  -2.257  0.02870 *
## leadcrudepro -0.127399   0.036732  -3.468  0.00113 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.3246 on 47 degrees of freedom
## Multiple R-squared:  0.8024, Adjusted R-squared:  0.7856
## F-statistic: 47.72 on 4 and 47 DF,  p-value: 5.59e-16
```

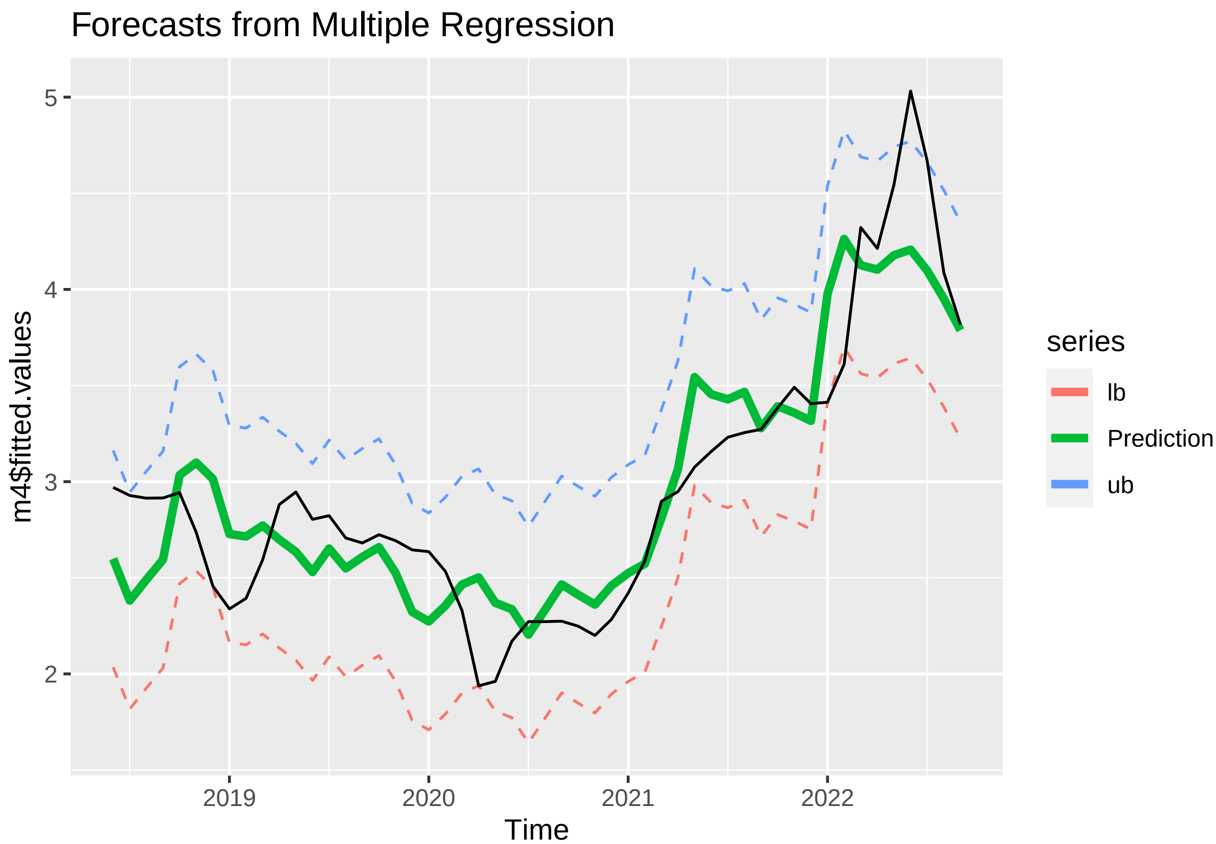
```r
m4 = tslm(retail_price~leadcpi + leadcrudepri + leadcrudepro,data=newdata)
summary(m4)
```

```
##
## Call:
## tslm(formula = retail_price ~ leadcpi + leadcrudepri + leadcrudepro,
##     data = newdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65134 -0.18108  0.02195  0.17904  0.82529
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.930601   4.243225   4.461 4.91e-05 ***
## leadcpi      -0.042256   0.014203  -2.975 0.004574 **
## leadcrudepri -0.021247   0.003702  -5.739 6.27e-07 ***
## leadcrudepro -0.130281   0.036735  -3.547 0.000883 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3254 on 48 degrees of freedom
## Multiple R-squared:  0.7972, Adjusted R-squared:  0.7845
## F-statistic: 62.89 on 3 and 48 DF,  p-value: < 2.2e-16
```

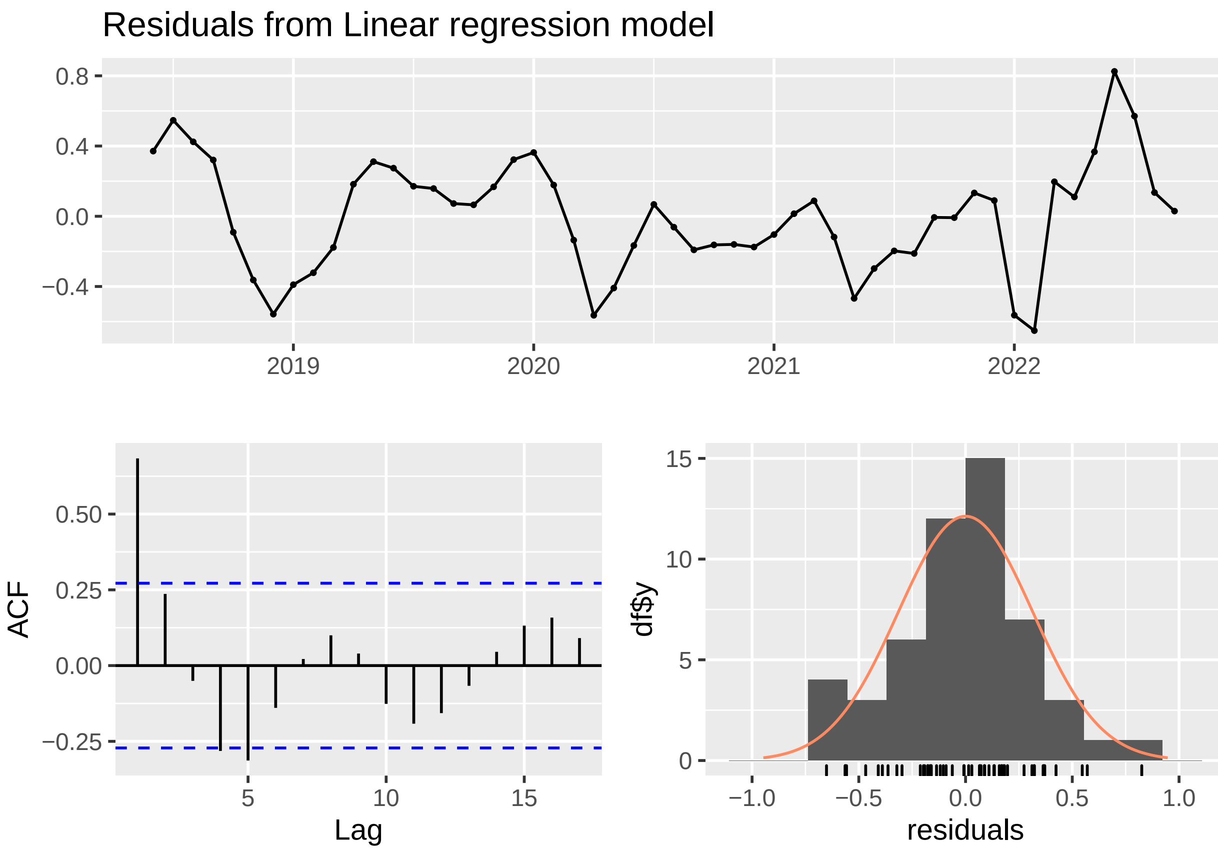
```r
accuracy(m4$fitted.values, gasoline.ts)
```

```
##                     ME      RMSE       MAE       MPE     MAPE      ACF1
## Test set -6.645325e-17 0.3126671 0.2520754 -1.016327 8.728708 0.6837357
##          Theil's U
## Test set  1.633076
```

```r
e <- m4$fitted.values - gasoline.ts
m <- quantile(e, probs = c(0.025,0.975))
lb=m4$fitted.values+m[1]
ub=m4$fitted.values+m[2]

reg.ts <- window(gasoline.ts,start=c(2018,6), frequency=12)
autoplot(m4$fitted.values, series = "Prediction", lwd=1.5, main="Forecasts from Multiple Regression") +
  autolayer(lb,lty="dashed")+ autolayer(ub,lty="dashed") + autolayer(reg.ts, series="Observed",lwd=0.5,
```

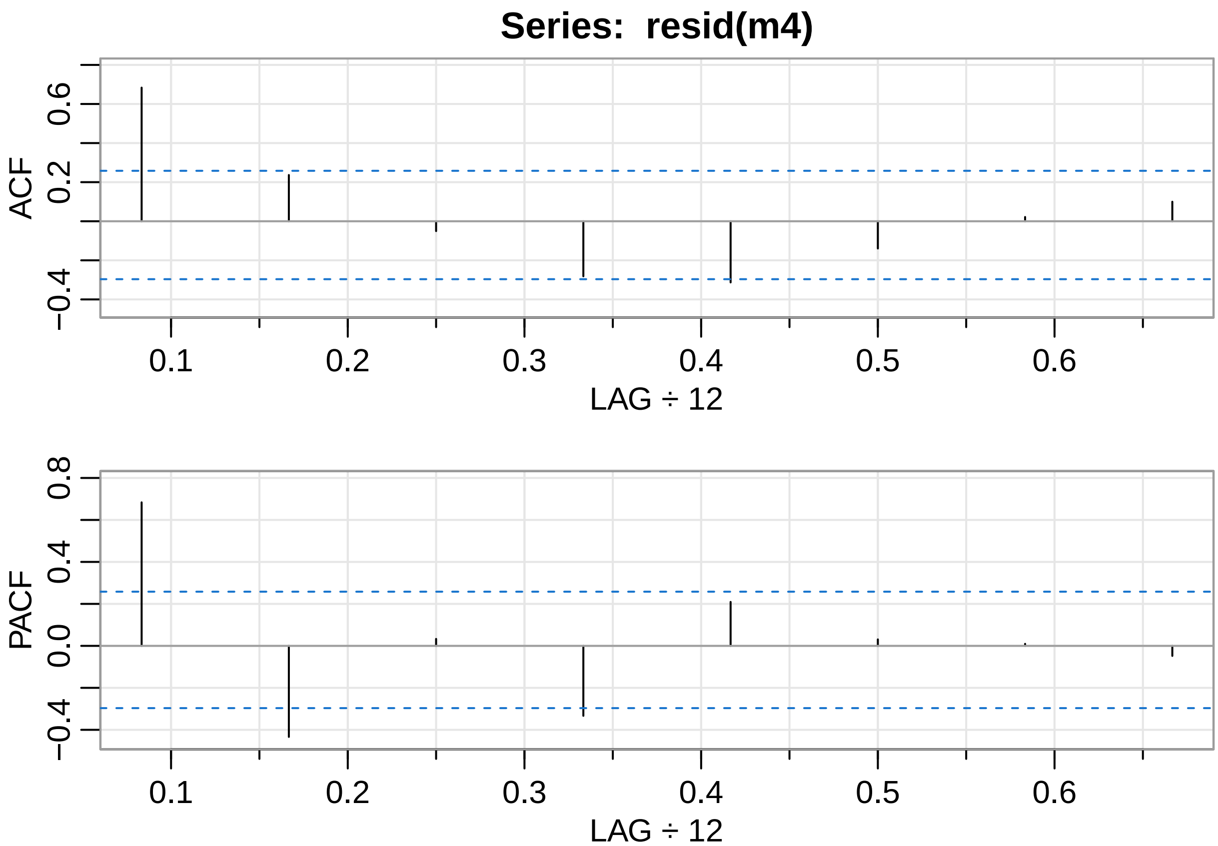



```
data = as.data.frame(newdata)
attach(data)
checkresiduals(m4)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals from Linear regression model
## LM test = 36.35, df = 10, p-value = 7.329e-05
```

```
acf2(resid(m4))
```




```
##      [,1]  [,2]  [,3]  [,4]  [,5]  [,6] [,7]  [,8]
## ACF  0.68  0.24 -0.05 -0.28 -0.31 -0.14 0.02  0.10
## PACF 0.68 -0.43  0.03 -0.33  0.21  0.03 0.01 -0.05
```

```
m5 = Arima(retail_price,order=c(2,0,0), xreg = cbind(leadcpi,leadcrudepri,leadcrudepro))
m5.pred = forecast(m5, xreg = cbind(leadcpi,leadcrudepri,leadcrudepro))
accuracy(m5.pred,retail_price)
```

```
##                        ME     RMSE       MAE        MPE      MAPE      MASE
## Training set -0.005773133 0.183258 0.1229021 -0.6900423  3.951394 0.8615814
## Test set     -0.029574300 0.433012 0.3332058 -3.7222658 11.789820 2.3358760
##                    ACF1
## Training set 0.05097853
## Test set             NA
```

Compare all models

```
accuracy(naive, validation)
```

```
##                         ME      RMSE       MAE        MPE      MAPE      MASE
## Training set -0.0002857143 0.1447208 0.1106723 -0.1529087  4.018176 0.2689545
## Test set      0.7837777778 0.9248965 0.7837778 17.5633473 17.563347 1.9047276
##                   ACF1 Theil's U
## Training set 0.4451151        NA
## Test set     0.4955834   2.15743
```

```r
accuracy(snaive, validation)
```

```
##                       ME      RMSE       MAE       MPE     MAPE     MASE
## Training set -0.06582407 0.5361608 0.4114907 -3.963977 15.54517 1.000000
## Test set      1.20711111 1.2644907 1.2071111 28.380948 28.38095 2.933507
##                   ACF1 Theil's U
## Training set 0.9144277        NA
## Test set     0.4056736  2.968893
```

```r
accuracy(ma.pred, validation)
```

```
##                 ME     RMSE       MAE      MPE     MAPE      ACF1 Theil's U
## Test set 0.5360864 0.729996 0.6087778 11.88611 13.79052 0.4337649  1.766839
```

```r
accuracy(zzz.pred,validation)
```

```
##                        ME      RMSE       MAE         MPE      MAPE      MASE
## Training set -0.002509267 0.1388332 0.1057644 -0.07218818  3.850549 0.2570273
## Test set      0.765629507 0.9076277 0.7656295 17.13538435 17.135384 1.8606239
##                   ACF1 Theil's U
## Training set 0.2896091        NA
## Test set     0.4928217  2.116923
```

```r
accuracy(m1.pred, validation) #ARIMA(0,1,1)(1,0,0)12
```

```
##                         ME      RMSE        MAE         MPE      MAPE     MASE
## Training set -0.0002289429 0.1260164 0.09519429 -0.05755609  3.483083 0.231340
## Test set      0.8055312476 0.9381094 0.80553125 18.12202008 18.122020 1.957593
##                    ACF1 Theil's U
## Training set 0.03154213        NA
## Test set     0.49005360  2.192487
```

```r
accuracy(m2.pred, validation) #ARIMA(0,1,1)(0,0,0)12
```

```
##                         ME      RMSE        MAE         MPE     MAPE      MASE
## Training set -0.0005597263 0.1262053 0.09554792 -0.07101427  3.49596 0.2321994
## Test set      0.8425950930 0.9752397 0.84259509 18.98692411 18.98692 2.0476648
##                    ACF1 Theil's U
## Training set 0.03531323        NA
## Test set     0.49558335  2.279334
```

```r
accuracy(gasolineprice.nnetar.opt.pred, valid.ts)
```

```
##                         ME      RMSE        MAE        MPE      MAPE      MASE
## Training set -3.203377e-05 0.1075198 0.08402676 -0.1810773  3.175958 0.2042008
## Test set      8.079017e-01 0.9490161 0.80790165 18.1332493 18.133249 1.9633532
##                    ACF1 Theil's U
## Training set 0.04612772        NA
## Test set     0.48438110  2.221744
```

```
accuracy(m4$fitted.values, gasoline.ts)
```

```
##                      ME      RMSE       MAE       MPE     MAPE      ACF1
## Test set -6.645325e-17 0.3126671 0.2520754 -1.016327 8.728708 0.6837357
##          Theil's U
## Test set  1.633076
```

```
AIC(m4)
```

```
## [1] 36.6575
```

```
m5$AIC
```

```
## NULL
```

Revaluate model accuracy with new released data

```
actual.gasoline <- c(3.935,3.799)
actual.ts <- ts(actual.gasoline, start=c(2022,10),end=c(2022,11), frequency = 12)

#1
naive.pred = naive(training, h = 11)
accuracy(naive.pred, actual.ts)
```

```
##                         ME      RMSE       MAE        MPE      MAPE      MASE
## Training set -0.0002857143 0.1447208 0.1106723 -0.1529087  4.018176 0.2689545
## Test set      0.4610000000 0.4659882 0.4610000 11.8941419 11.894142 1.1203168
##                    ACF1 Theil's U
## Training set  0.4451151        NA
## Test set     -0.5000000  2.889706
```

```
accuracy(naive, validation)
```

```
##                         ME      RMSE       MAE        MPE      MAPE      MASE
## Training set -0.0002857143 0.1447208 0.1106723 -0.1529087  4.018176 0.2689545
## Test set      0.7837777778 0.9248965 0.7837778 17.5633473 17.563347 1.9047276
##                   ACF1 Theil's U
## Training set 0.4451151        NA
## Test set     0.4955834   2.15743
```

```
#2
snaive.pred = snaive(training, h = 11)
accuracy(snaive.pred, actual.ts)
```

```
##                       ME      RMSE       MAE       MPE     MAPE     MASE
## Training set -0.06582407 0.5361608 0.4114907 -3.963977 15.54517 1.000000
## Test set      0.42950000 0.4463547 0.4295000 11.054969 11.05497 1.043766
##                    ACF1 Theil's U
## Training set  0.9144277        NA
## Test set     -0.5000000  2.264706
```

```r
accuracy(snaive, validation)
```

```
##                        ME      RMSE       MAE       MPE     MAPE     MASE
## Training set -0.06582407 0.5361608 0.4114907 -3.963977 15.54517 1.000000
## Test set      1.20711111 1.2644907 1.2071111 28.380948 28.38095 2.933507
##                   ACF1 Theil's U
## Training set 0.9144277        NA
## Test set     0.4056736  2.968893
```

```r
#3
n = length(gasoline.ts)

w = 9
nValid = w

ma.trail.pred1= rep(NA, nValid)

for (i in 1:11) {
  nTraining= n-nValid + (i - 1)
  training3=window(gasoline.ts, start=c(2012, 1),
                    end=c(2012, nTraining))

  ma.trailing.tmp1 = rollmean(training3, k=w, align="right")
  last.ma1=tail(ma.trailing.tmp1,1)

  ma.trail.pred1[i] = last.ma1
}
```

```
## Warning in window.default(x, ...): 'end' value not changed
```

```r
ma.pred1= ts(ma.trail.pred1,start=c(2012,length(gasoline.ts)-nValid+1), frequency = 12)

validation <- window(gasoline.ts, start=c(2022,1))

accuracy(ma.pred1, actual.ts)
```

```
##                  ME      RMSE       MAE       MPE     MAPE ACF1 Theil's U
## Test set -0.3227778 0.3298628 0.3227778 -8.380495 8.380495 -0.5  2.873366
```

```r
#4
zzz.pred1=forecast(zzz,h=11,level=0)
accuracy(zzz.pred1, actual.ts)
```

```
##                         ME      RMSE       MAE         MPE      MAPE      MASE
## Training set -0.002509267 0.1388332 0.1057644 -0.07218818  3.850549 0.2570273
## Test set      0.434357592 0.4396972 0.4343576 11.20481639 11.204816 1.0555708
##                    ACF1 Theil's U
## Training set  0.2896091        NA
## Test set     -0.5000000  2.691479
```

```r
#5
m1.pred1 = forecast(m1,h=11)
accuracy(m1.pred1, actual.ts)
```

```
##                         ME      RMSE        MAE         MPE      MAPE     MASE
## Training set -0.0002289429 0.1260164 0.09519429 -0.05755609  3.483083 0.231340
## Test set      0.4571546212 0.4626369 0.45715462 11.79330056 11.793301 1.110972
##                     ACF1 Theil's U
## Training set  0.03154213        NA
## Test set     -0.50000000  2.839291
```

```r
#5
m2.pred1 = forecast(m2,h=11)
accuracy(m2.pred1, actual.ts)
```

```
##                         ME      RMSE        MAE         MPE     MAPE      MASE
## Training set -0.0005597263 0.1262053 0.09554792 -0.07101427  3.49596 0.2321994
## Test set      0.5198173152 0.5242462 0.51981732 13.41561869 13.41562 1.2632540
##                     ACF1 Theil's U
## Training set  0.03531323        NA
## Test set     -0.50000000  3.322186
```

```r
#6
gasolineprice.nnetar.opt.pred1 <- forecast(gasolineprice.nnetar.opt, h = 11)
accuracy(gasolineprice.nnetar.opt.pred1, actual.ts)
```

```
##                         ME      RMSE        MAE        MPE     MAPE      MASE
## Training set -3.203377e-05 0.1075198 0.08402676 -0.1810773 3.175958 0.2042008
## Test set      3.546054e-01 0.3650583 0.35460543  9.1334233 9.133423 0.8617580
##                     ACF1 Theil's U
## Training set  0.04612772        NA
## Test set     -0.50000000  1.969652
```

```r
#7
summary(m4)
```

```
##
## Call:
## tslm(formula = retail_price ~ leadcpi + leadcrudepri + leadcrudepro,
##     data = newdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65134 -0.18108  0.02195  0.17904  0.82529
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.930601   4.243225   4.461 4.91e-05 ***
## leadcpi      -0.042256   0.014203  -2.975 0.004574 **
## leadcrudepri -0.021247   0.003702  -5.739 6.27e-07 ***
## leadcrudepro -0.130281   0.036735  -3.547 0.000883 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3254 on 48 degrees of freedom
## Multiple R-squared:  0.7972, Adjusted R-squared:  0.7845
## F-statistic: 62.89 on 3 and 48 DF,  p-value: < 2.2e-16
```

```
actual.gasoline <- c(3.935,3.799)


variable.data <- as.data.frame(ts.intersect(leadcpi1=lag(cpi.ts,-34),
                        leadcrudepri1=lag(crudeoil.ts,-77),
                        leadcrudepro1=lag(production.ts,-20)))

attach(variable.data)
reg.pred <- rep(NA, length(leadcpi1))
n = length(reg.pred)

for (i in 1:n) {
  reg.pred[i] = m4$coefficients[1] + m4$coefficients[2] * leadcpi1[i] +
                m4$coefficients[3]* leadcrudepri1[i] +
                m4$coefficients[4]* leadcrudepro1[i]
}
m4$fitted.values
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2018                                             2.598925 2.381283 2.490093
## 2019 2.727574 2.714782 2.771635 2.698502 2.634915 2.529861 2.652244 2.549009
## 2020 2.272634 2.355115 2.464885 2.501999 2.369795 2.336433 2.204409 2.334079
## 2021 2.524140 2.572481 2.810111 3.066336 3.543817 3.454562 3.428125 3.466897
## 2022 3.976544 4.262342 4.125407 4.103141 4.177792 4.206709 4.097743 3.951966
##           Sep      Oct      Nov      Dec
## 2018 2.594463 3.034031 3.099071 3.014543
## 2019 2.608519 2.658986 2.525460 2.322336
## 2020 2.465419 2.411049 2.360318 2.459285
## 2021 3.278357 3.391941 3.358067 3.316245
## 2022 3.787627
```

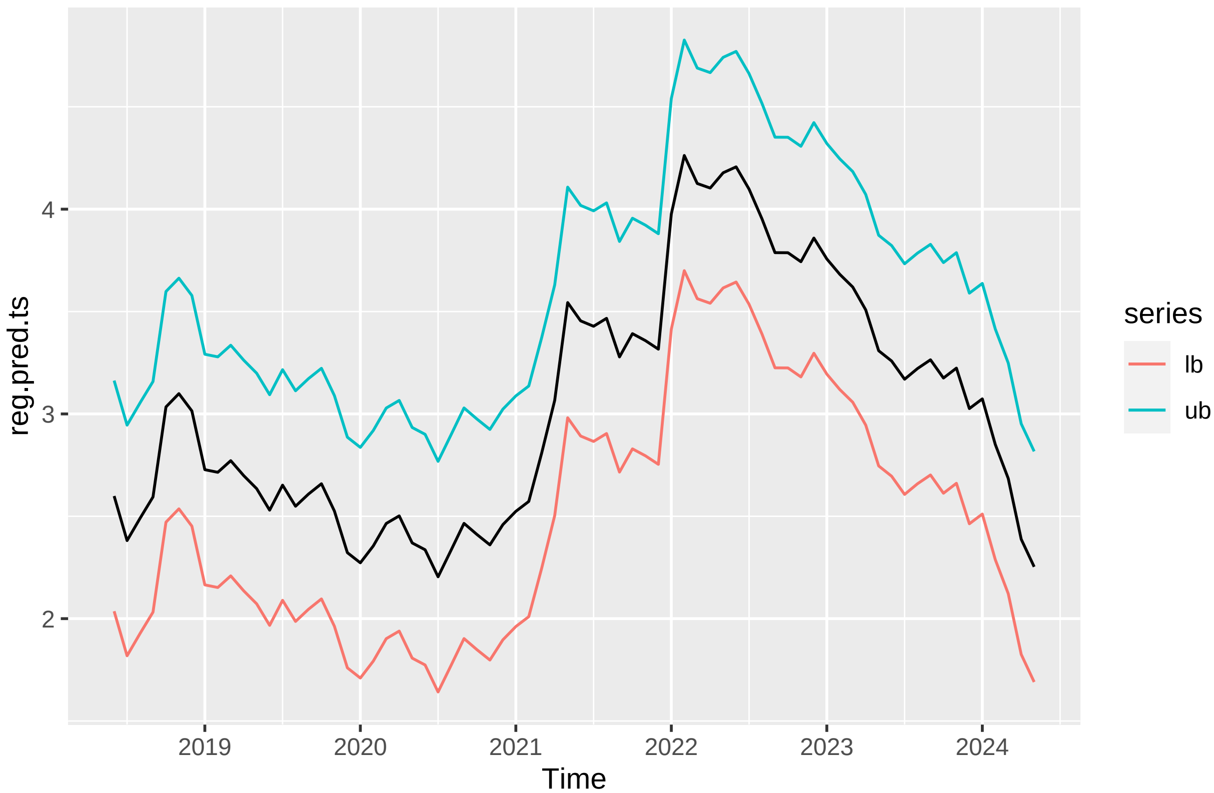
```
reg.pred.ts <- ts(reg.pred, start=c(2018,6),frequency=12)
gasoline.ts <- ts(append(as.numeric(gasoline.ts), actual.gasoline),start=c(2012,1),frequency = 12)
accuracy(reg.pred.ts, gasoline.ts)
```

```
##                   ME      RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set 0.003767786 0.3075737 0.246507 -0.881996 8.502112 0.6831247  1.627235
```

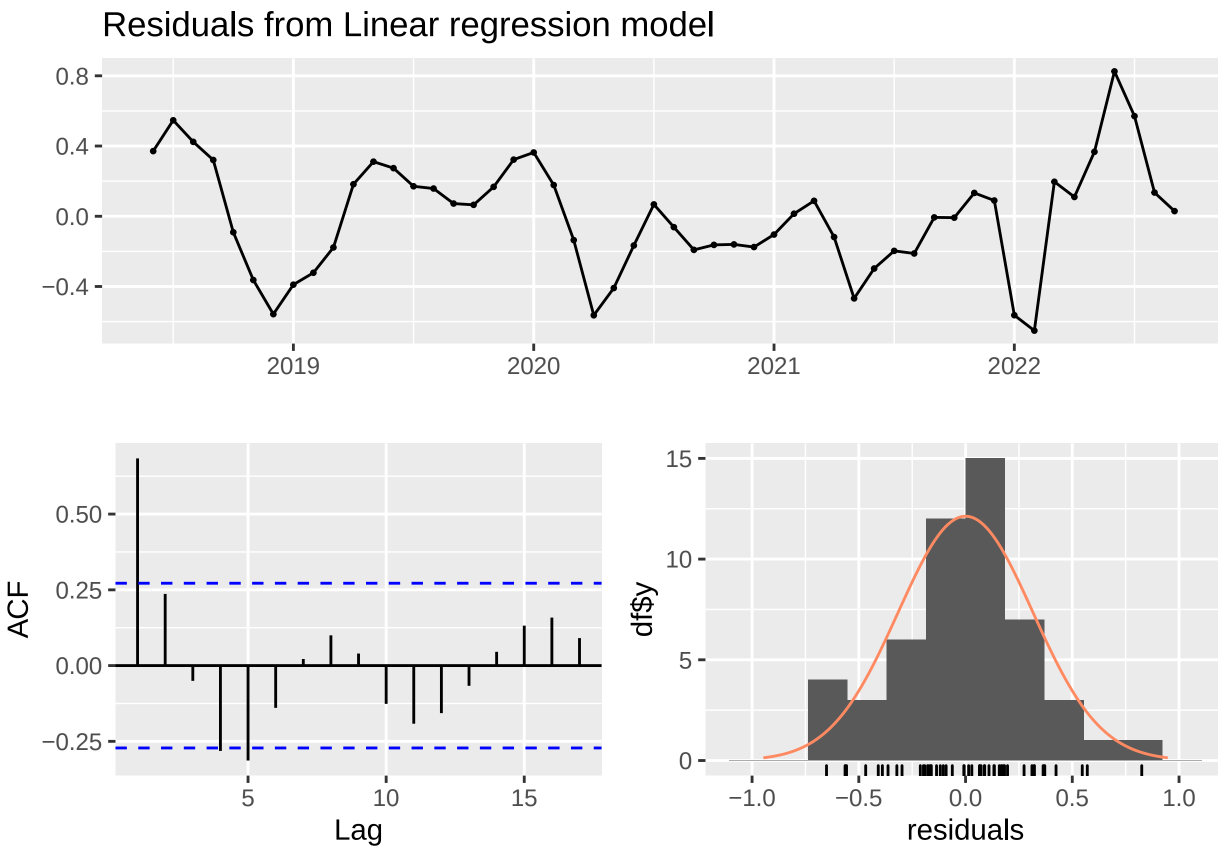
```
accuracy(reg.pred.ts, actual.ts)
```

```
##                 ME      RMSE       MAE      MPE     MAPE ACF1 Theil's U
## Test set 0.1017302 0.1116451 0.1017302 2.610618 2.610618 -0.5 0.4098129
```

```r
e <- reg.pred.ts - gasoline.ts
m <- quantile(e, probs = c(0.025,0.975))
lb=reg.pred.ts+m[1]
ub=reg.pred.ts+m[2]

autoplot(reg.pred.ts) +
  autolayer(lb)+ autolayer(ub)
```

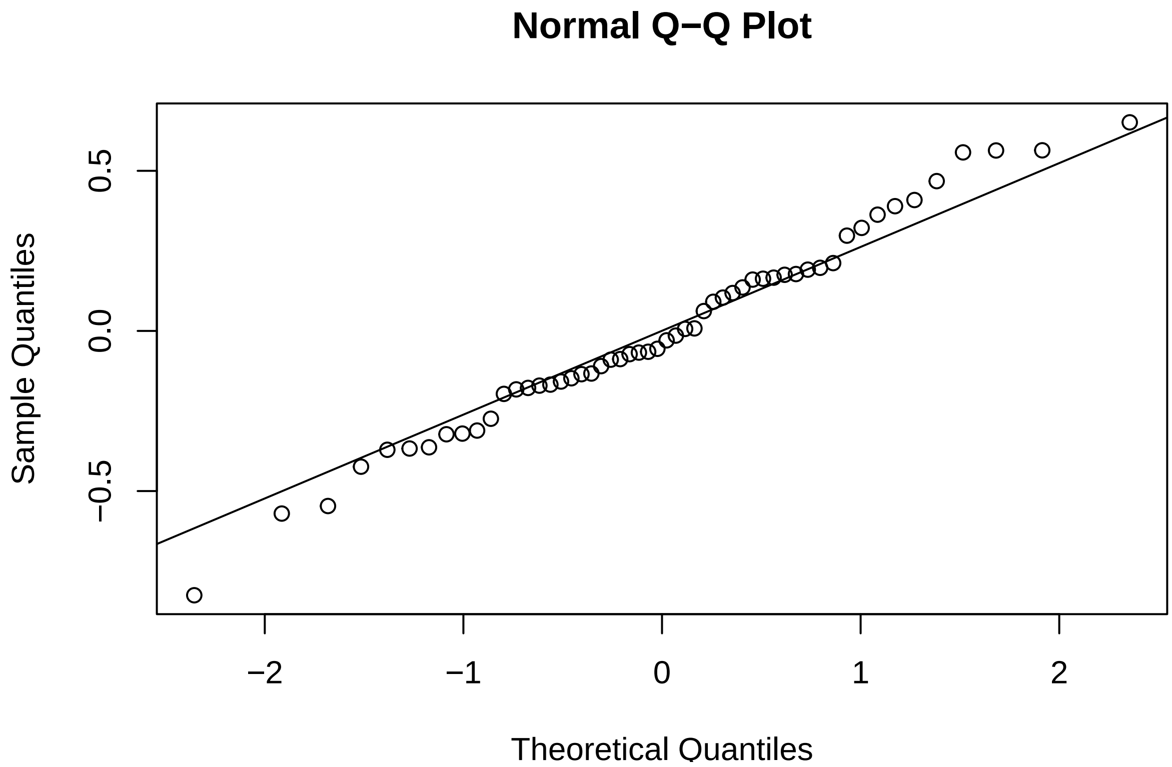



```r
checkresiduals(m4)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals from Linear regression model
## LM test = 36.35, df = 10, p-value = 7.329e-05
```

```
qqnorm(e)
qqline(e)
```


Normal Q−Q Plot
Sample Quantiles
0.5
0.0
−0.5
−2
−1
0
1
2
Theoretical Quantiles