Forecasting Brent Crude Oil Appendix

Forecasting DC Oil Brent EU

Eirik Egge

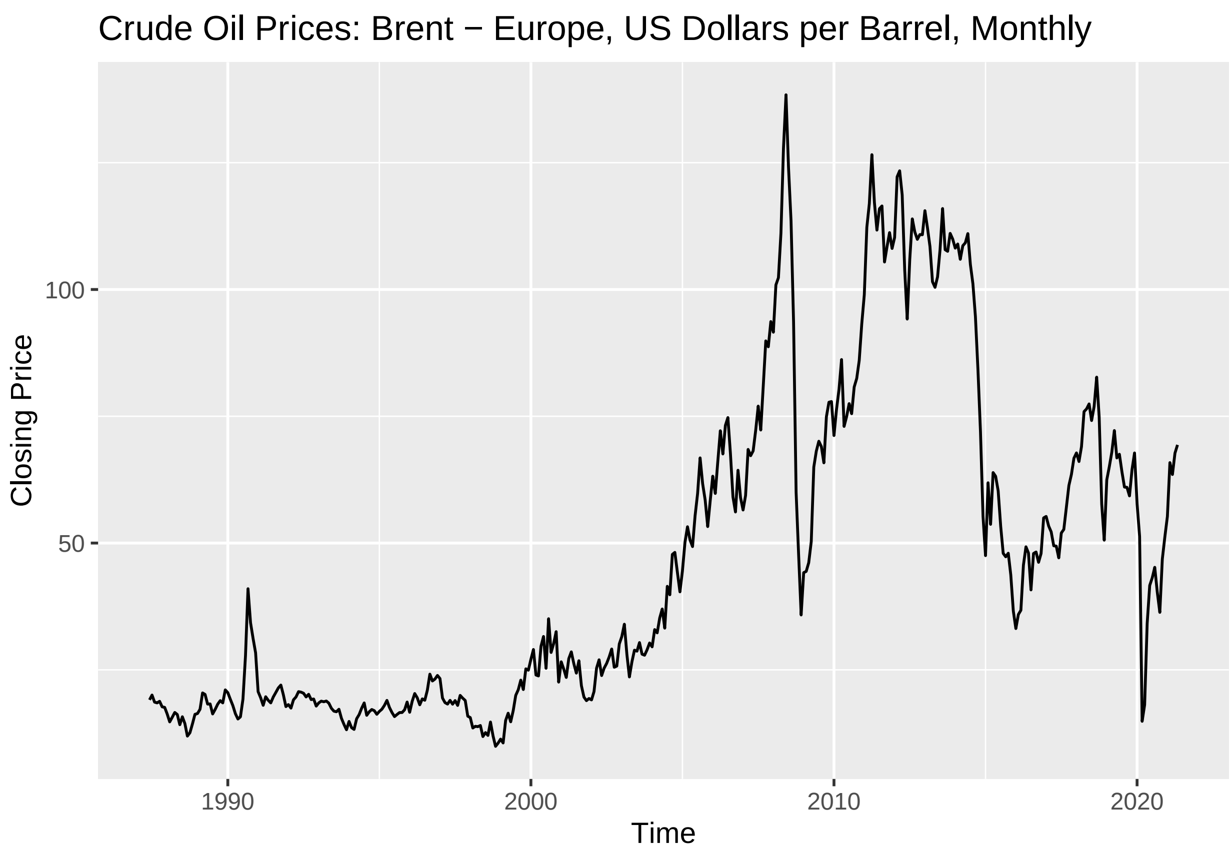
30/08/2021

1 Section 1.1

```
# LOAD DATA, INSPECT ACF & DECOMP. PLOT

#read data
df_oil <- read.csv("DCOILBRENTEU.csv", sep = ",")

#create xts and ts time series
oil$DATE <- as.Date(df_oil$DATE, "%Y-%m-%d")
```

```
## Warning in oil$DATE <- as.Date(df_oil$DATE, "%Y-%m-%d"): Coercing LHS to a list
```

```
oil.xts <- xts(df_oil[,2], order.by = oil$DATE)
oil.ts <- ts(df_oil[,2], start = c(1987,6), end = c(2021,5), frequency = 12)

#view data
autoplot(oil.ts) + ylab("Closing Price") + xlab("Time") +
  ggtitle("Crude Oil Prices: Brent - Europe, US Dollars per Barrel, Monthly")
```




```r
#check NAs
which(is.na(oil.ts)) #null na values
```

```
## integer(0)
```

```r
#check col names
names(df_oil)
```

```
## [1] "DATE"         "DCOILBRENTEU"
```

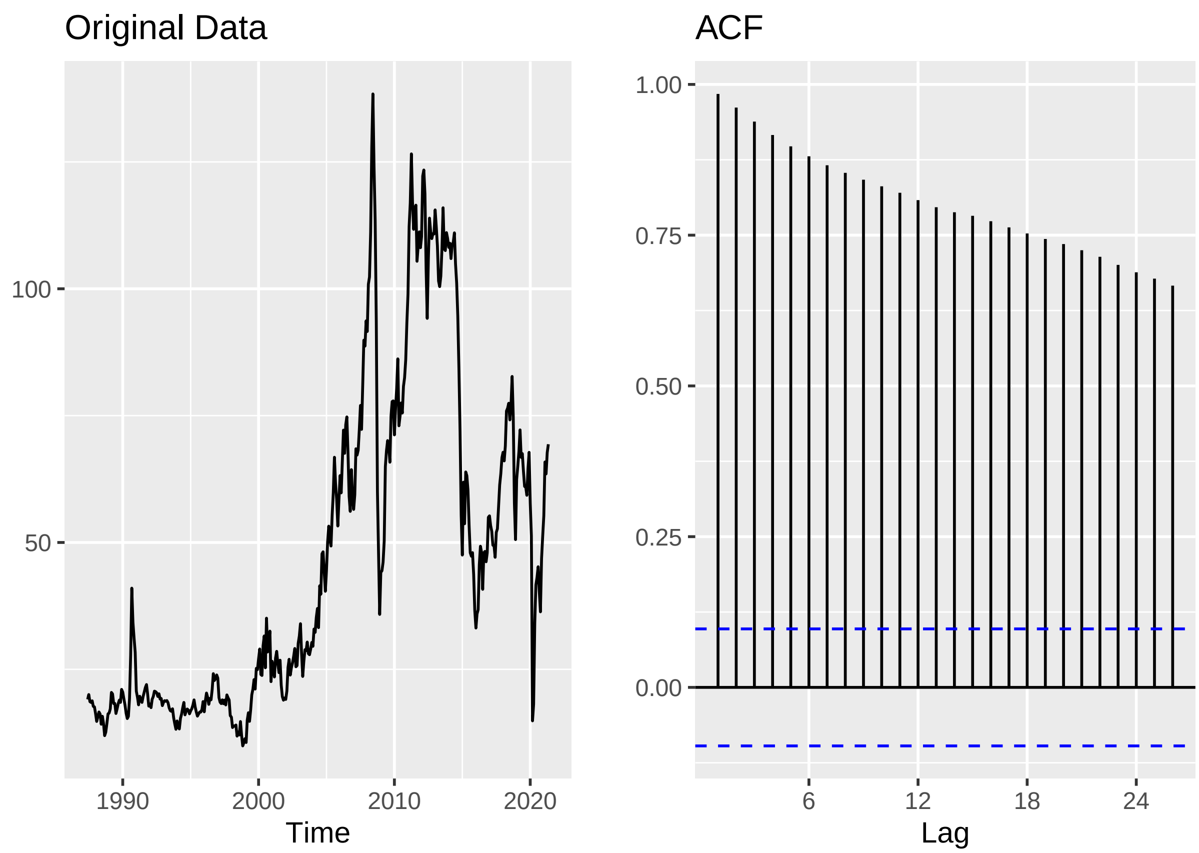
```r
#check class
class(df_oil)
```

```
## [1] "data.frame"
```

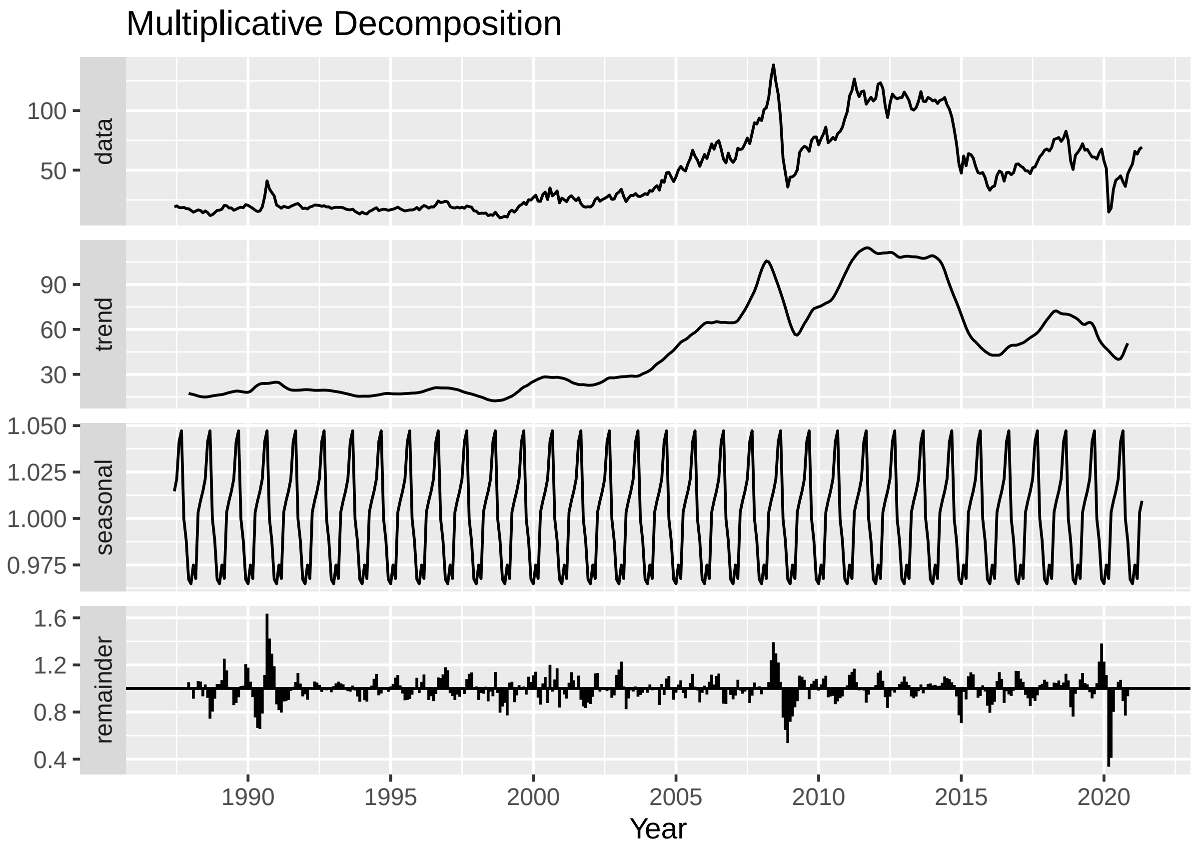
```r
# summary stats
summary(df_oil)
```

```
##      DATE            DCOILBRENTEU   
##  Length:408         Min.   :  9.91  
##  Class :character   1st Qu.: 18.98  
##  Mode  :character   Median : 34.23  
##                     Mean   : 46.77  
##                     3rd Qu.: 67.53  
##                     Max.   :138.40  
```

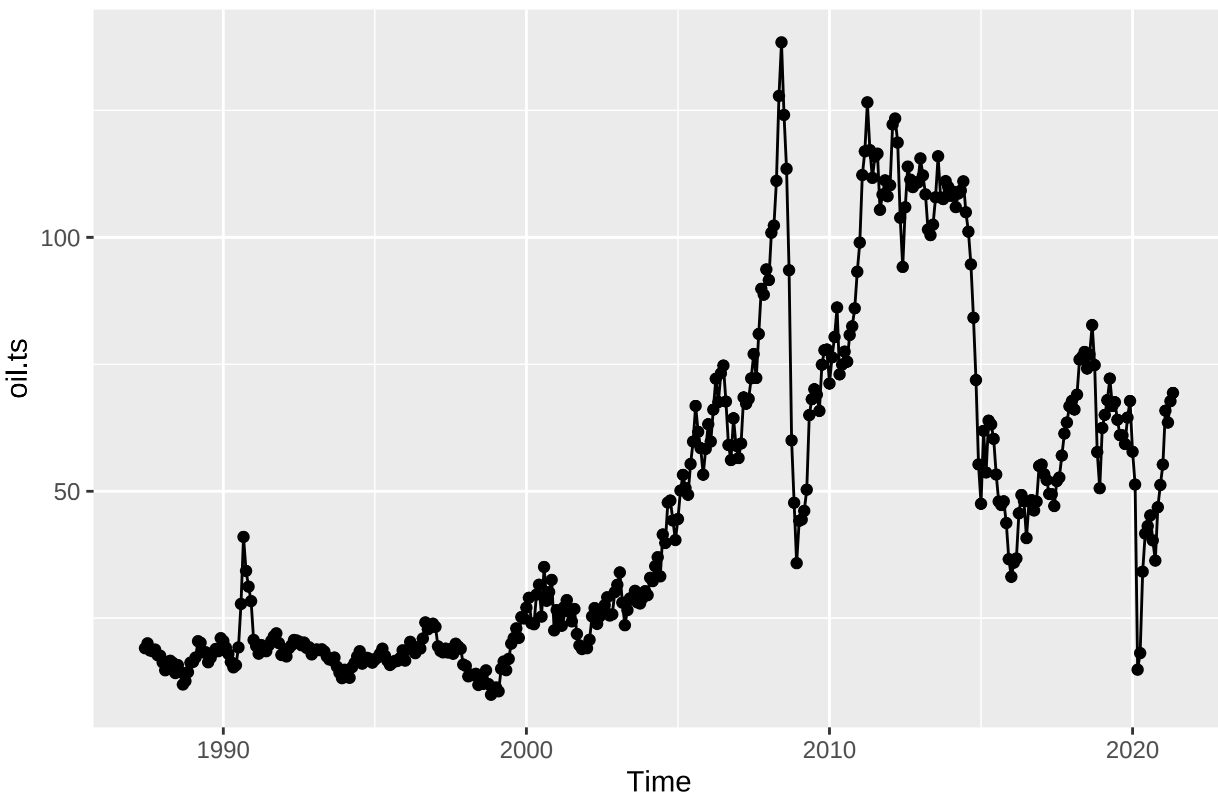
```r
#### TIME SERIES PATTERNS ####

grid.arrange((autoplot(oil.ts)
              + ylab("")
              + ggtitle("Original Data")),
             (ggAcf(oil.ts)
              + ylab("")
              + ggtitle("ACF")),
             nrow = 1)
```




```r
# decomposition plot (multiplicative)
(autoplot(decompose(oil.ts, type="multiplicative"))
+ xlab("Year")
+ ggtitle("Multiplicative Decomposition"))
```




```
#check variance for deciding upon transformation (unequal variance indicates that a transformation migh
autoplot(oil.ts) + geom_point()
```




```
# check for seasonality (test) # to be dealt with in stationarity section
summary(wo(oil.ts)) #p-valoue = 0.4955 (H0: Non-seasonal time series, HA: Seasonal time series) (https:/
```

```
## Test used:  WO
##
## Test statistic:  0
## P-value:  1 1 0.4996585
##
## The WO - test does not identify  seasonality
```

```
#conclusion: no seasonality
```

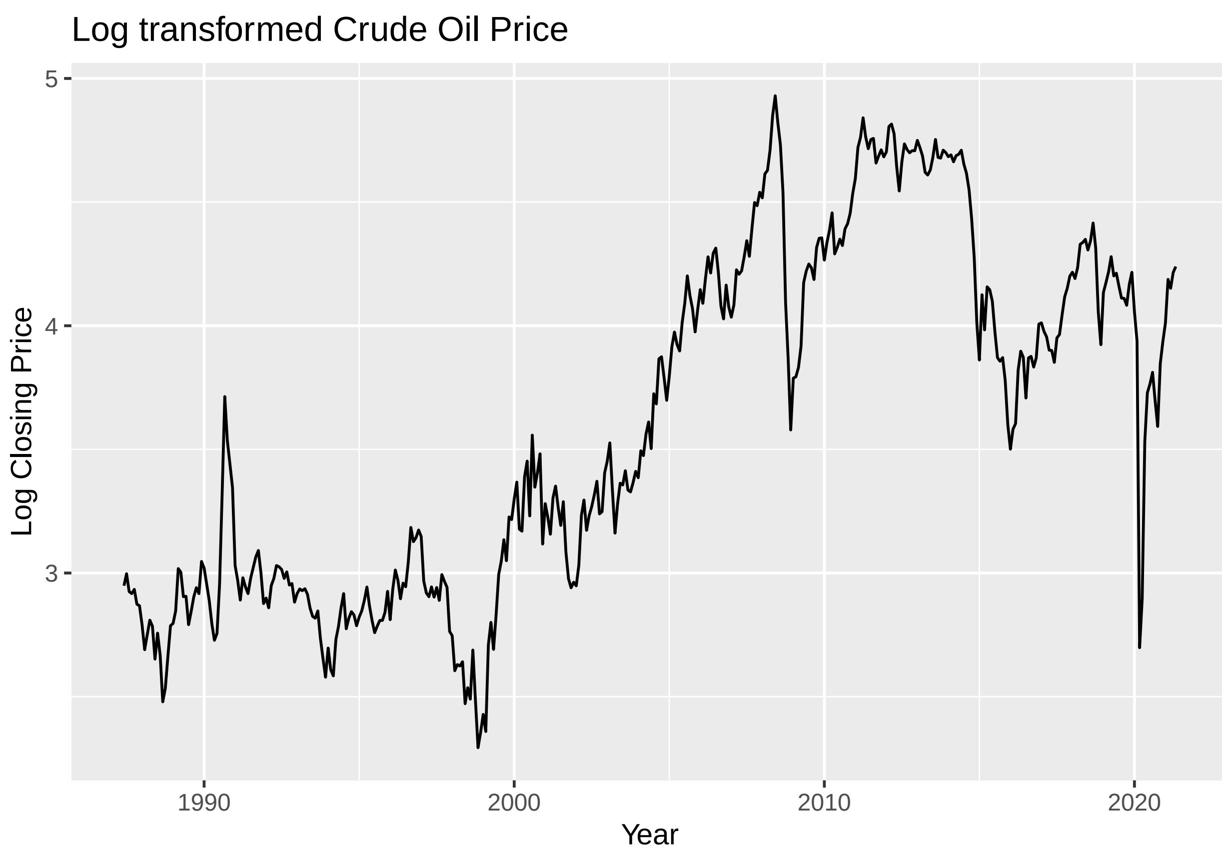
Comment Original Data: + Appears to be non stationary + Exponential increase from '02 until '08, followed by high fluctuation + Increasing variance

Comment ACF plot: + increasing trend until 2000, might be exponential + lot of fluctuations + several upswings and slumps, which might indicate a cyclic behavior

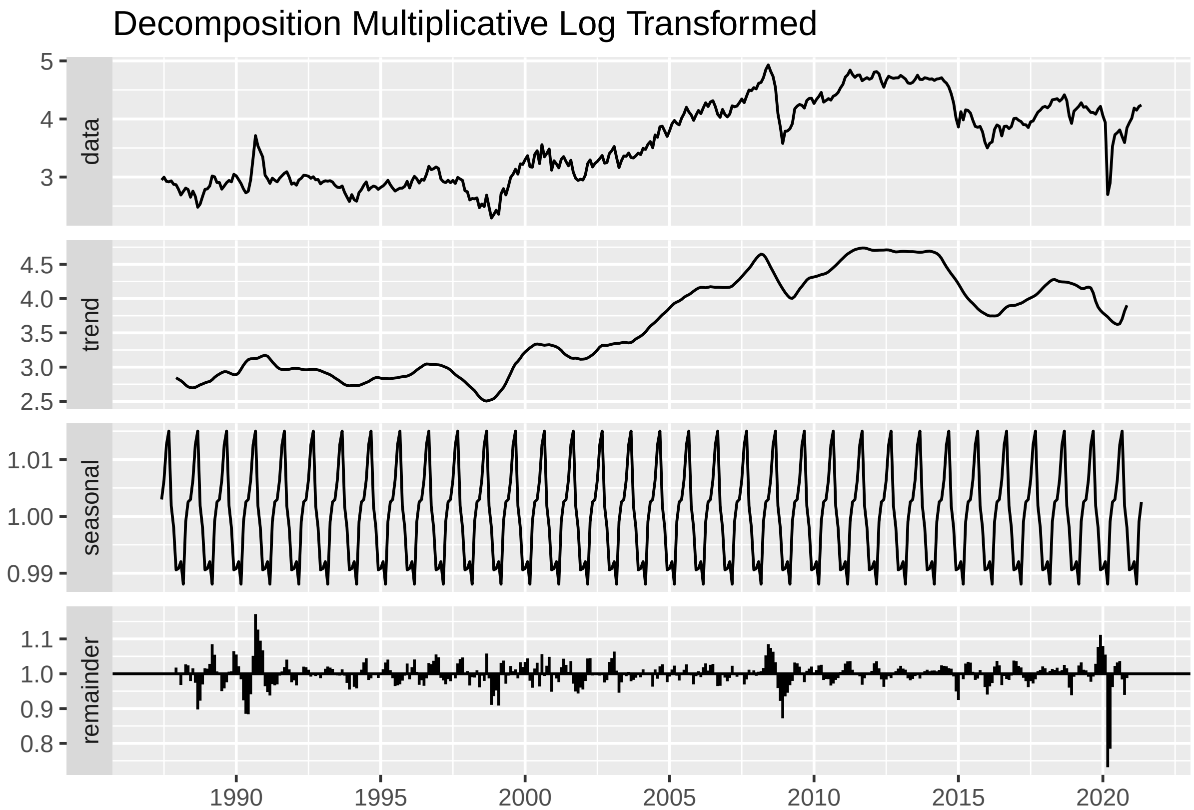
Comment Multiplicative Decomposition Plot: + trend is exponential + seasonality suggested, but cannot be confirmed as this plot assumes seasonal behavior that is the same each year + remainder/noise significant in '90, '07 and '20, otherwise stable (does not increase or decrease)

2 Section 1.2

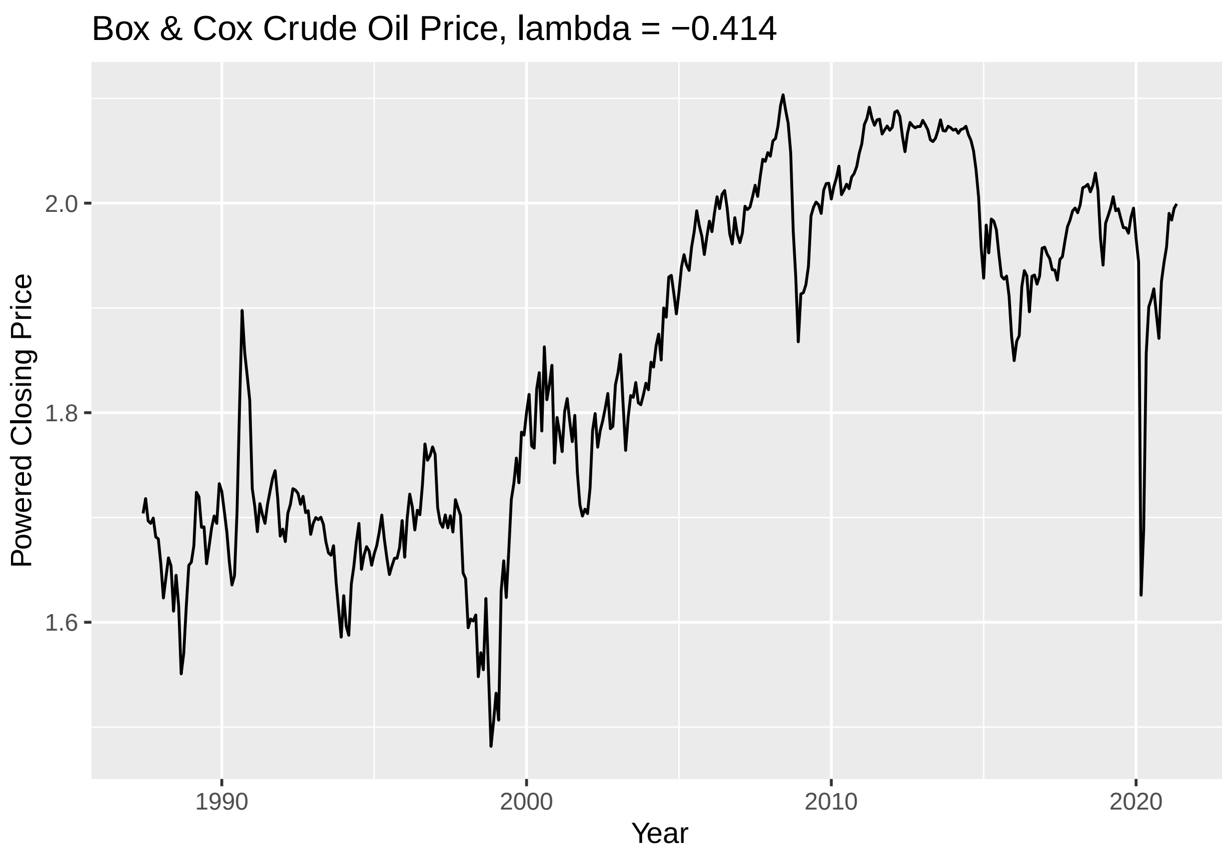
```r
# TRANSFORM

#Logarithmic transformation and decomposition plot
log_oil.ts <- log(oil.ts)

autoplot(log_oil.ts) + ylab("Log Closing Price") + xlab("Year") +
  ggtitle("Log transformed Crude Oil Price")
```



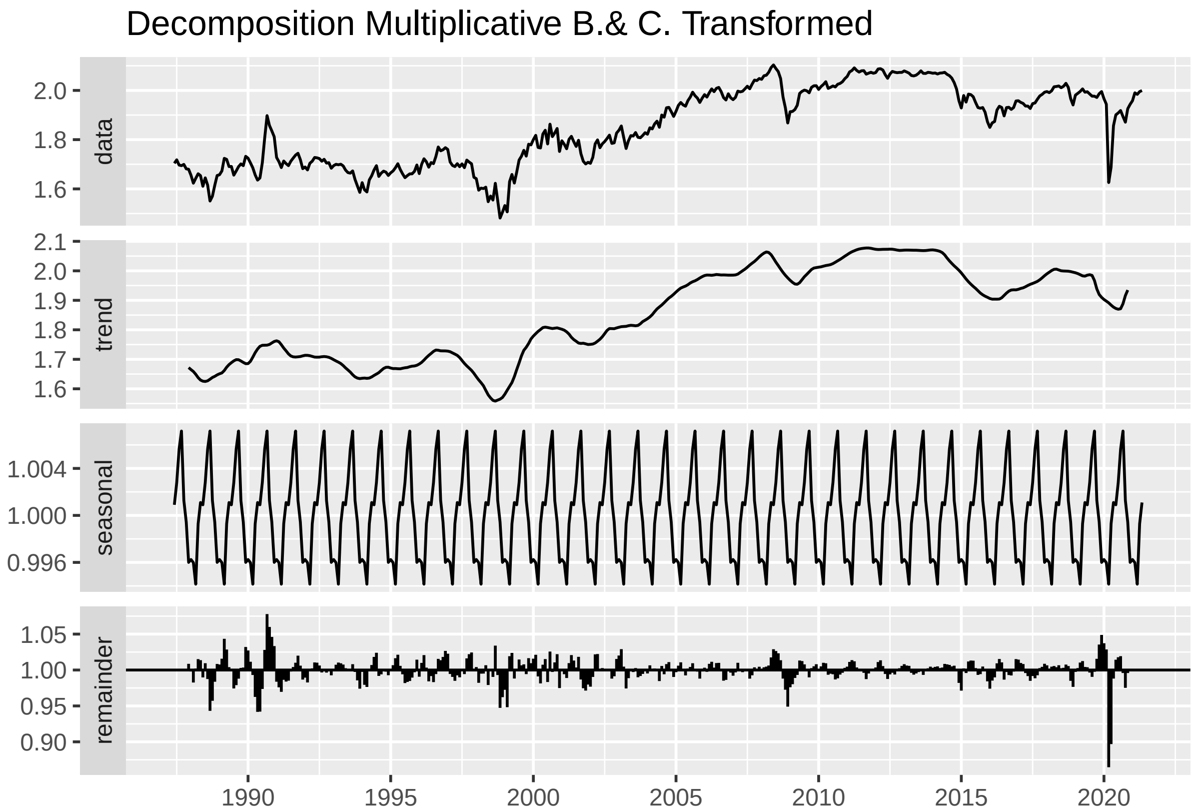

```r
dec_log <- decompose(log_oil.ts, type = c('multiplicative'),filter = NULL)
autoplot(dec_log) + ylab("") + xlab("") + ggtitle('Decomposition Multiplicative Log Transformed')
```




```
# Box & Cox transformation (power, lambda = -0.414) and decomposition plot
lambda_oil <- BoxCox.lambda(oil.ts)

bc_oil.ts <- BoxCox(oil.ts,lambda_oil)

autoplot(bc_oil.ts) + ylab("Powered Closing Price") + xlab("Year") +
  ggtitle("Box & Cox Crude Oil Price, lambda = -0.414")
```




```
dec_bc <- decompose(bc_oil.ts, type = c('multiplicative'),filter = NULL)
autoplot(dec_bc) + ylab("") + xlab("") + ggtitle('Decomposition Multiplicative B.& C. Transformed')
```



Comments regarding transformation (Box & Cox): + Time series now displays a more linear trend + Therefore, bc_oil.ts will be used for further analysis + The patterns in the historical data is now simplified + These factors can lead to a simpler forecasting task, and hereby more accurate forecasts

3 Section 2.1

```
# REMOVING NON STATIONARITY

# KPSS- H0: Data is stationary, HA: Data is not stationary

# ADF - H0: Data has unit root, HA: Stationary* (*different hypothesis)

summary(ur.kpss(bc_oil.ts, type = "tau"))
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: tau with 5 lags.
##
## Value of test-statistic is: 0.6583
##
```

```
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

```
summary(ur.df(bc_oil.ts, type= "trend", selectlags = "AIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32365 -0.01290  0.00237  0.01403  0.14146
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0967860  0.0270242   3.581 0.000384 ***
## z.lag.1     -0.0590891  0.0164844  -3.585 0.000379 ***
## tt           0.0000647  0.0000222   2.915 0.003758 **
## z.diff.lag   0.0702466  0.0497347   1.412 0.158598
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.031 on 402 degrees of freedom
## Multiple R-squared:  0.03258,    Adjusted R-squared:  0.02536
## F-statistic: 4.513 on 3 and 402 DF,  p-value: 0.003971
##
##
## Value of test-statistic is: -3.5845 4.3455 6.4253
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.98 -3.42 -3.13
## phi2  6.15  4.71  4.05
## phi3  8.34  6.30  5.36
```

Comment KPSS w/ tau: + The value of the test-statistic is greater than the critical value on 1% level. Therefore, we **reject** the null hypothesis that our data is stationary. (0.6583 > 0.216) 1pct + Conclusion: NOT STATIONARY

Comment ADF w/ trend: + GAMMA/TAU3 (gamma = 0 –> unit root): + 1pct: |T-stat| **accept** lower than critical value (unit root)

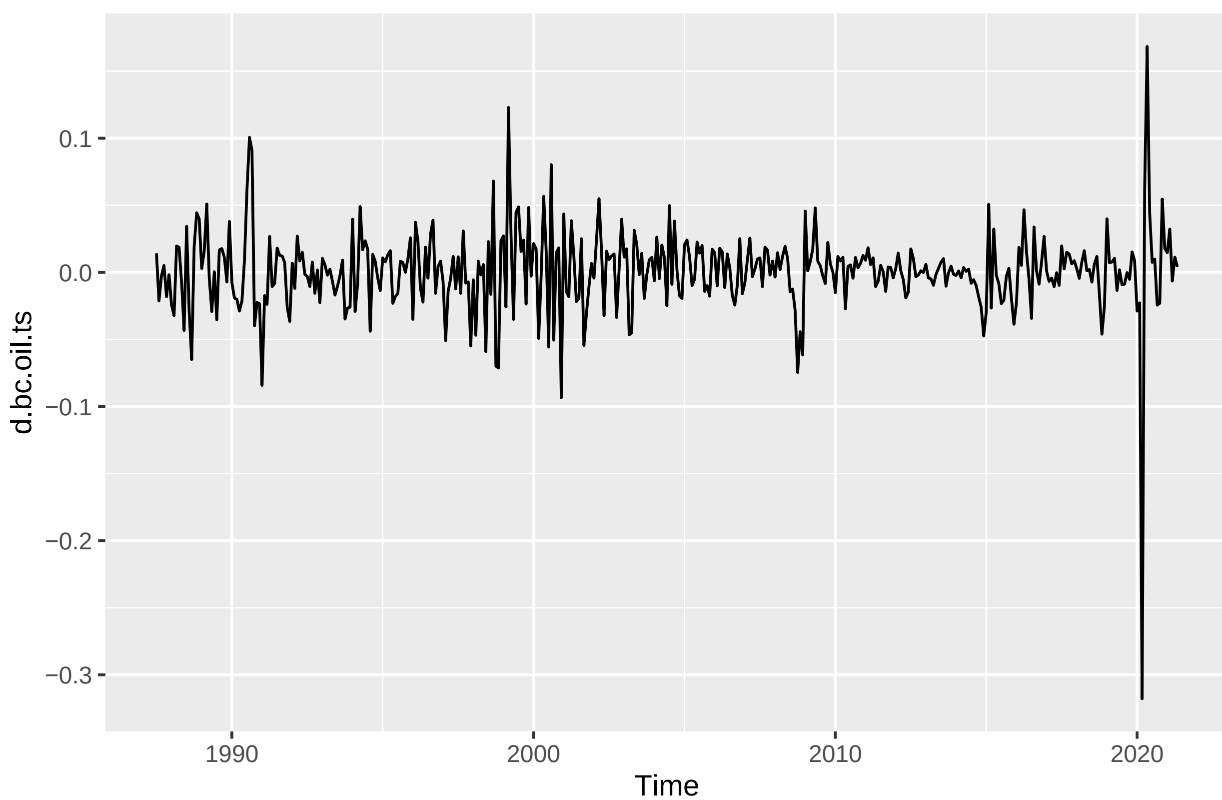
- PHI2 (alpha0 = alpha2 = gamma = 0, drift/alpha0)
  - 5pct & 1pct: |T-stat| **accept** lower than critical value (drift, trend, unit root)
- PHI3 (alpha2 = gamma = 0, trend/alpha2)

    - 5pct & 1pct: |T-stat| **accept** lower than critical value (trend)

- Conclusion 1pct: unit root, drift, trend

4 Section 2.2

```
# take the difference
d.bc.oil.ts <- diff(bc_oil.ts)

# inspect data
autoplot(d.bc.oil.ts)
```




```
summary(ur.kpss(d.bc.oil.ts, type = "tau"))
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: tau with 5 lags.
##
## Value of test-statistic is: 0.0354
##
```

```
## Critical value for a significance level of:
##                  10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

```r
summary(ur.df(d.bc.oil.ts, type="trend", selectlags = "AIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32138 -0.01324  0.00066  0.01322  0.12696
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.778e-04  3.122e-03   0.281  0.77869
## z.lag.1     -1.085e+00  6.852e-02 -15.840  < 2e-16 ***
## tt          -3.500e-07  1.327e-05  -0.026  0.97898
## z.diff.lag   1.322e-01  4.946e-02   2.672  0.00784 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03123 on 401 degrees of freedom
## Multiple R-squared:  0.4887, Adjusted R-squared:  0.4849
## F-statistic: 127.7 on 3 and 401 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -15.8401 83.6373 125.4552
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.98 -3.42 -3.13
## phi2  6.15  4.71  4.05
## phi3  8.34  6.30  5.36
```

```r
## no trend (0.0354 < 0.216)

# new test, type = "mu" and "drift"
summary(ur.kpss(d.bc.oil.ts, type = "mu"))
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
```

```
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 0.0354
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
summary(ur.df(d.bc.oil.ts, type="drift", selectlags = "AIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32145 -0.01325  0.00063  0.01319  0.12699
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0008064  0.0015508   0.520  0.60334
## z.lag.1     -1.0853683  0.0684349 -15.860  < 2e-16 ***
## z.diff.lag   0.1321749  0.0494020   2.675  0.00777 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03119 on 402 degrees of freedom
## Multiple R-squared:  0.4887, Adjusted R-squared:  0.4861
## F-statistic: 192.1 on 2 and 402 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -15.8599 125.7682
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.44 -2.87 -2.57
## phi1  6.47  4.61  3.79
```

Comment KPSS: + all pct: t-stat lower than critical values (0.0354 < 0.347 10pct) **accept** H0: data ia stationary

Comment ADF: + TAU2: + t-stat > critical value **reject** H0: hence, no drift. + PHI1: + t-stat > critical value **reject** H0: no unit root.

5 Section 2.3

```
summary(ur.df(d.bc.oil.ts, type="none", selectlags = "AIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32061 -0.01246  0.00141  0.01398  0.12779
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -1.08429    0.06834 -15.866  < 2e-16 ***
## z.diff.lag  0.13163    0.04935   2.668  0.00795 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03117 on 403 degrees of freedom
## Multiple R-squared:  0.4883, Adjusted R-squared:  0.4858
## F-statistic: 192.3 on 2 and 403 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -15.8657
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

Comment ADF: + T-stat > critical value at 1pct (-15.8657 > -2.58) + **Reject** H0: Data are stationary

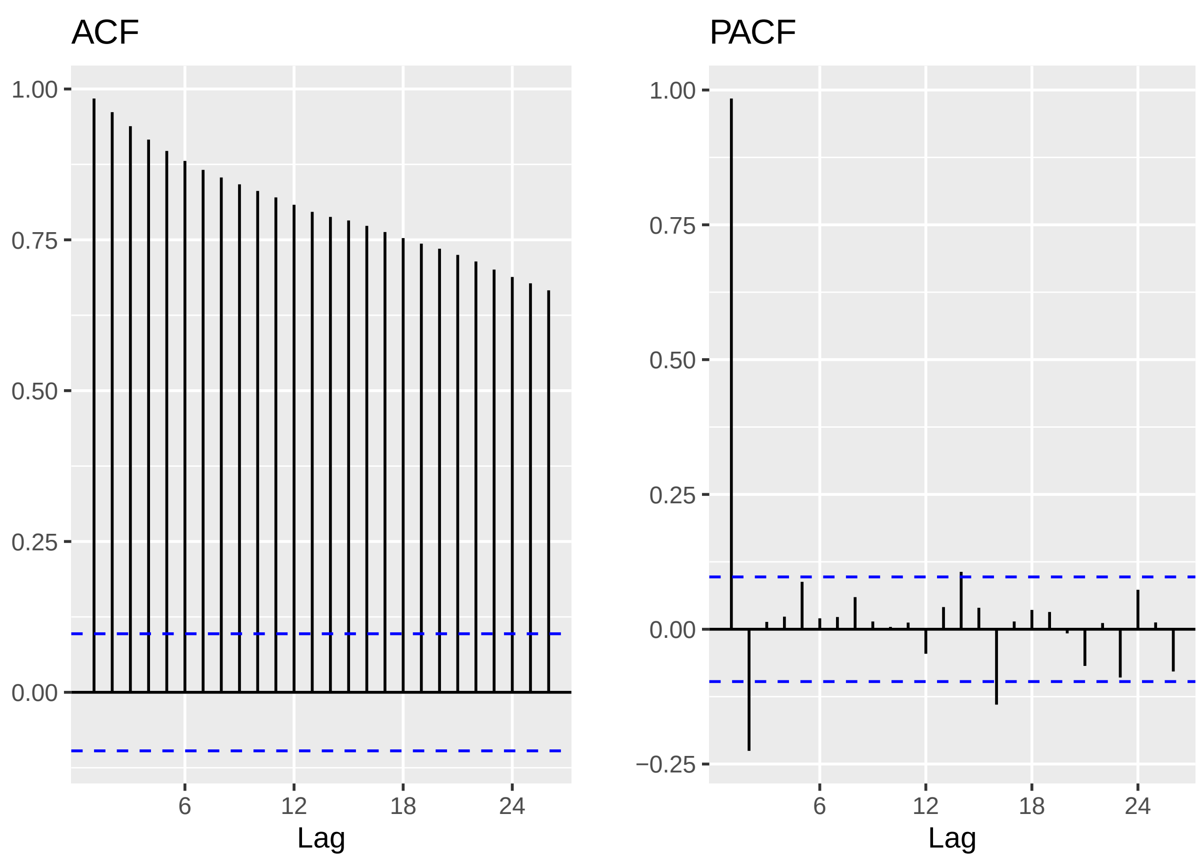
6 Section 2.4

```
###### DATA are now STATIONARY ######

# Box Pierce test for autocorrelation

Box.test(d.bc.oil.ts, lag = 10, fitdf = 0) #(check what fitdf is)
```

```
##
##  Box-Pierce test
```

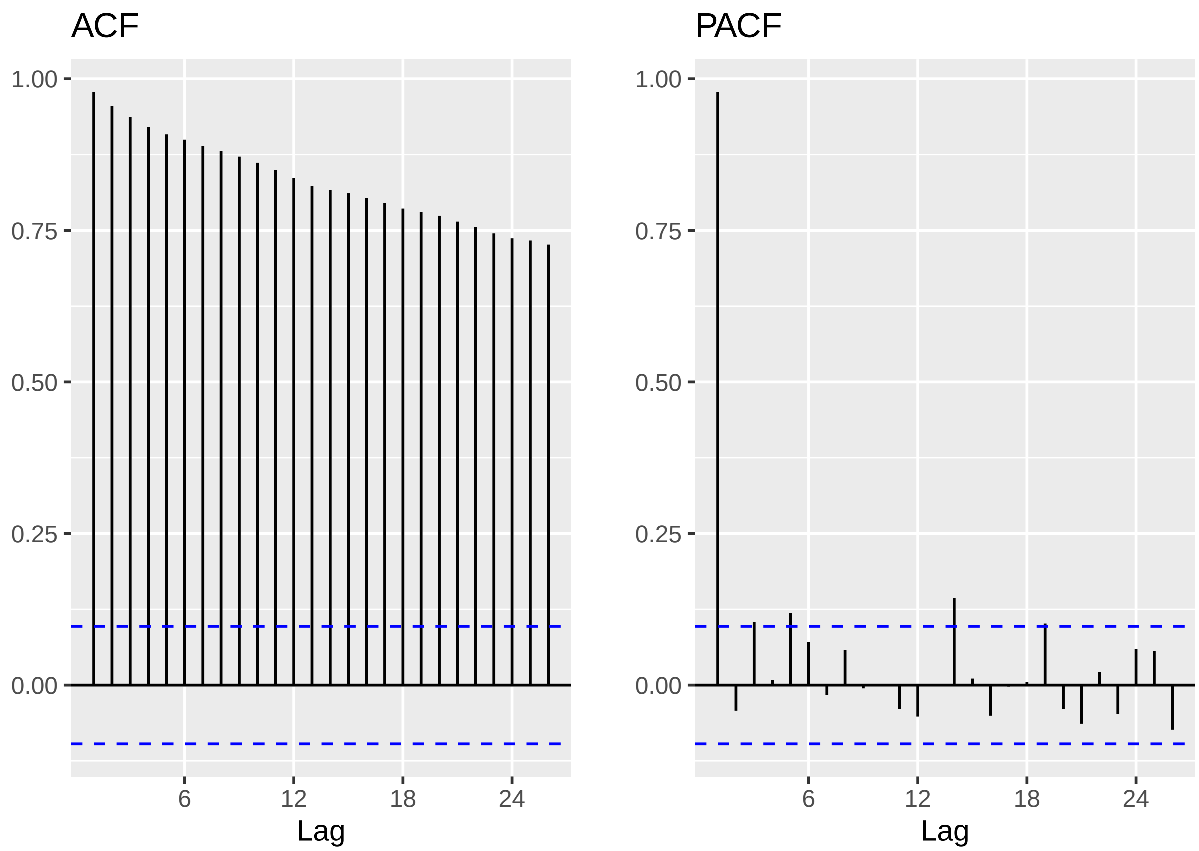
```
##
## data:  d.bc.oil.ts
## X-squared = 18.653, df = 10, p-value = 0.0449
```

```
# H0: Not random walk (double check)
# HA: Random walk
# P-value = 0.0449, therefore not random walk


# ACF & PACF PLOTS

# O.G. Data
grid.arrange((ggAcf(oil.ts)
              + ylab("")
              + ggtitle("ACF")),
             (ggPacf(oil.ts)
              + ylab("")
              + ggtitle("PACF")),
             nrow = 1)
```

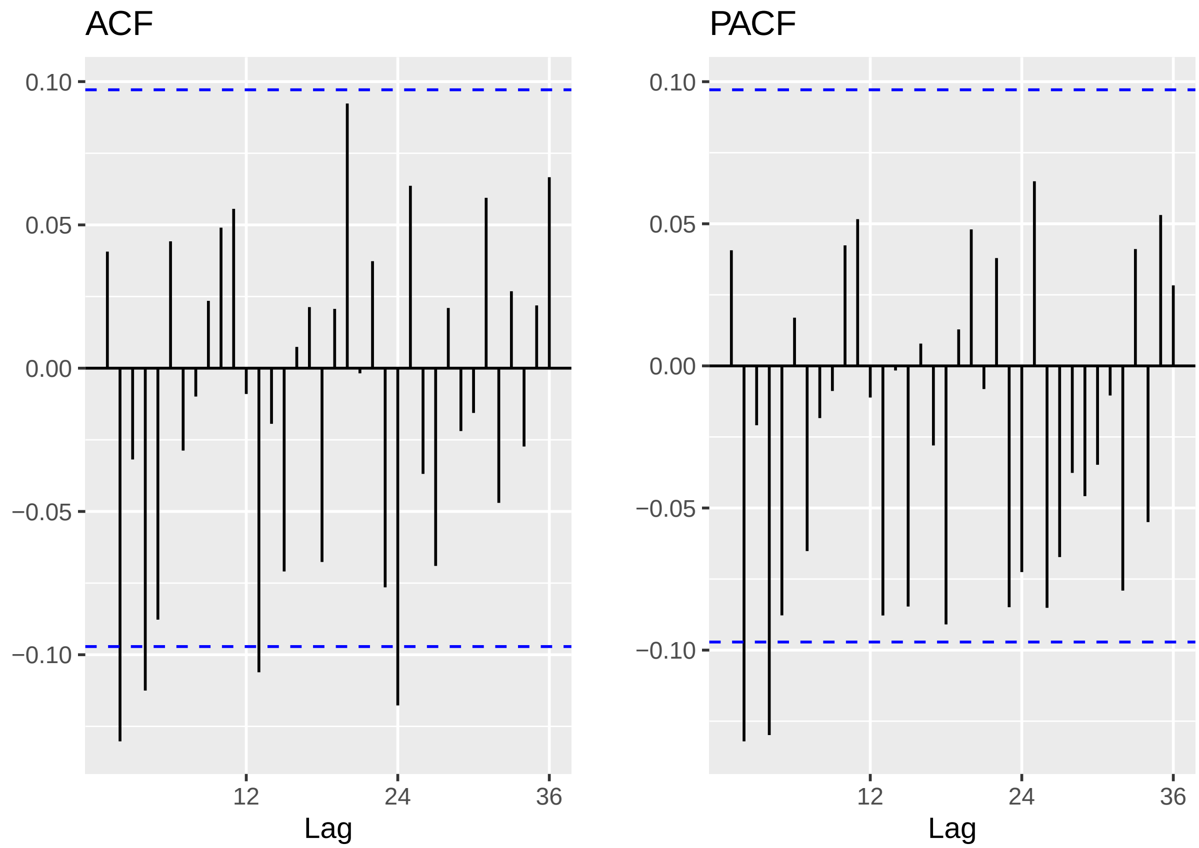



```
# B&C Transformed Oil
grid.arrange((ggAcf(bc_oil.ts)
              + ylab("")
              + ggtitle("ACF")),
             (ggPacf(bc_oil.ts)
```

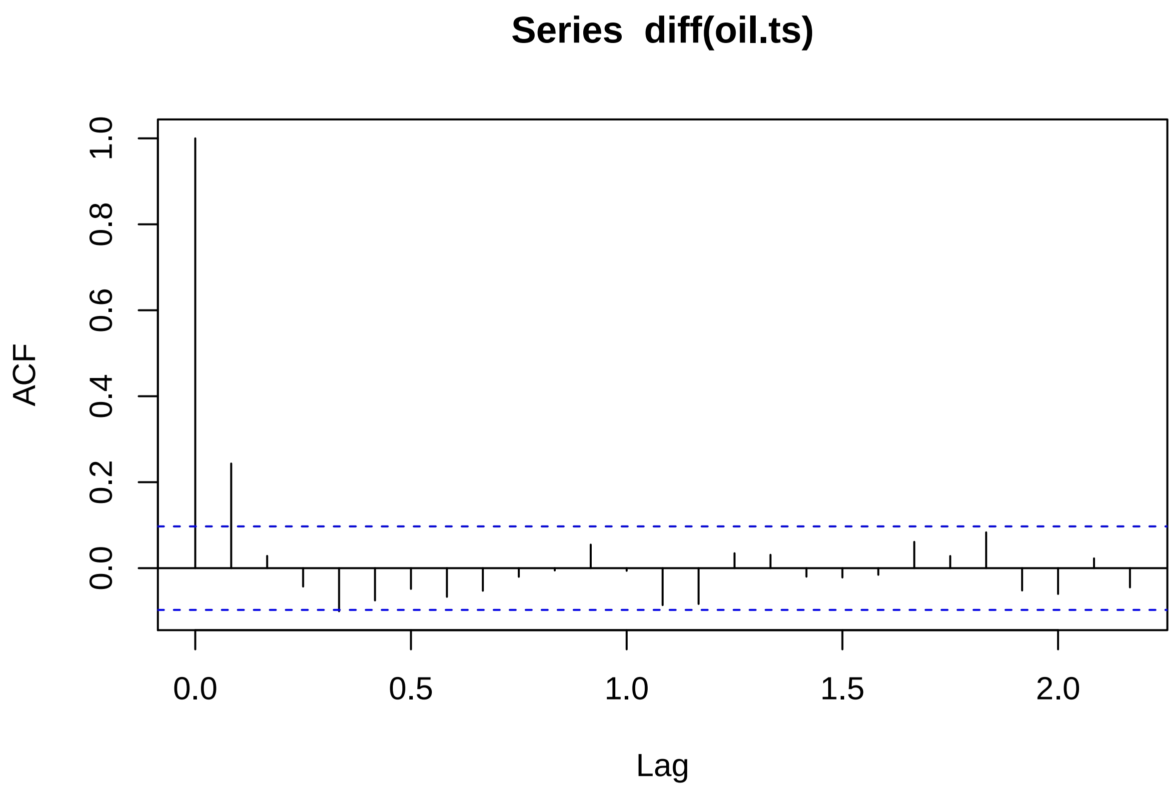
```
                 + ylab("")
                 + ggtitle("PACF")),
                nrow = 1)
```

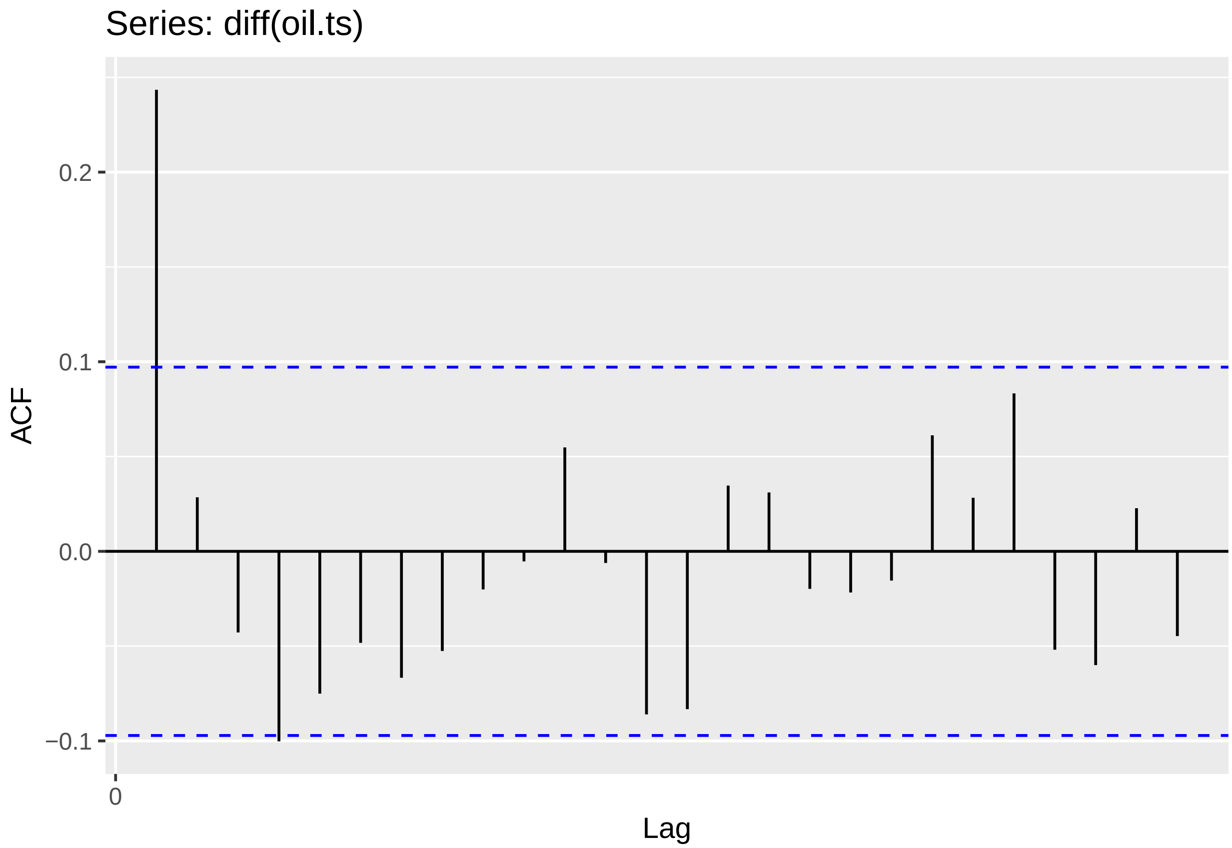



```
# B&C Transformed Oil DIFFERENTIATED
grid.arrange((ggAcf(d.bc.oil.ts, lag.max = 36)
                 + ylab("")
                 + ggtitle("ACF")),
                (ggPacf(d.bc.oil.ts, lag.max = 36)
                 + ylab("")
                 + ggtitle("PACF")),
                nrow = 1)
```

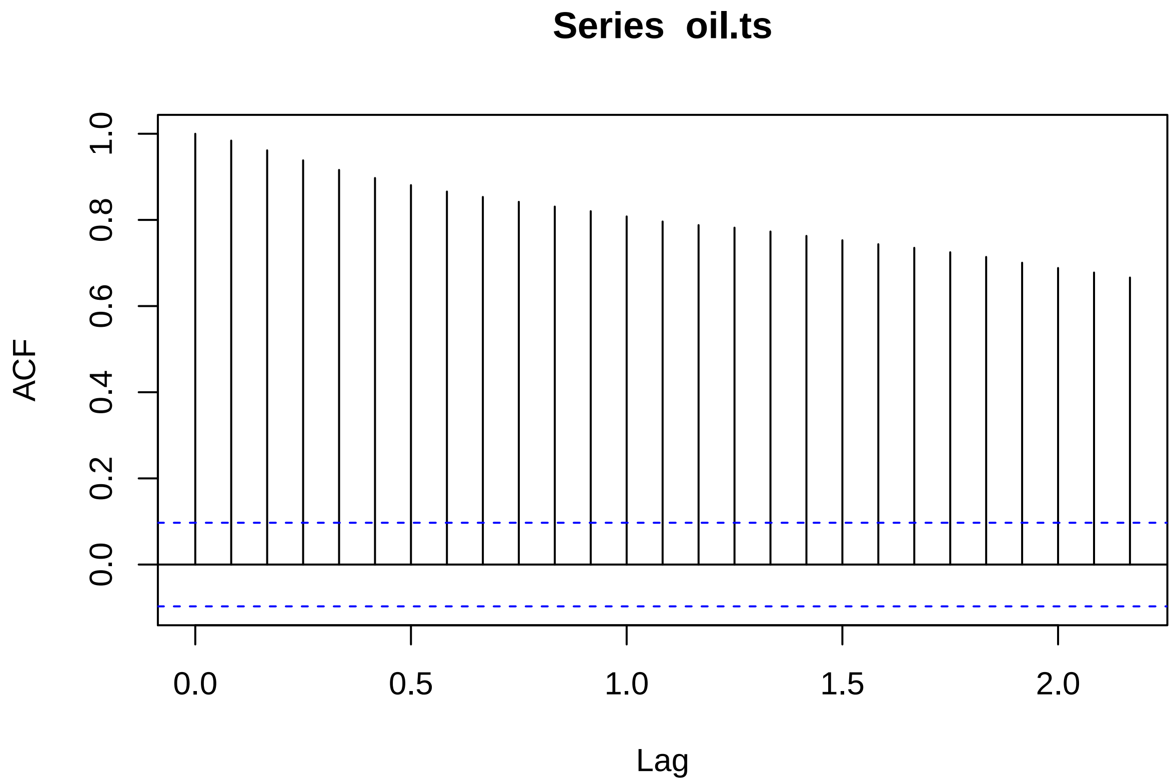


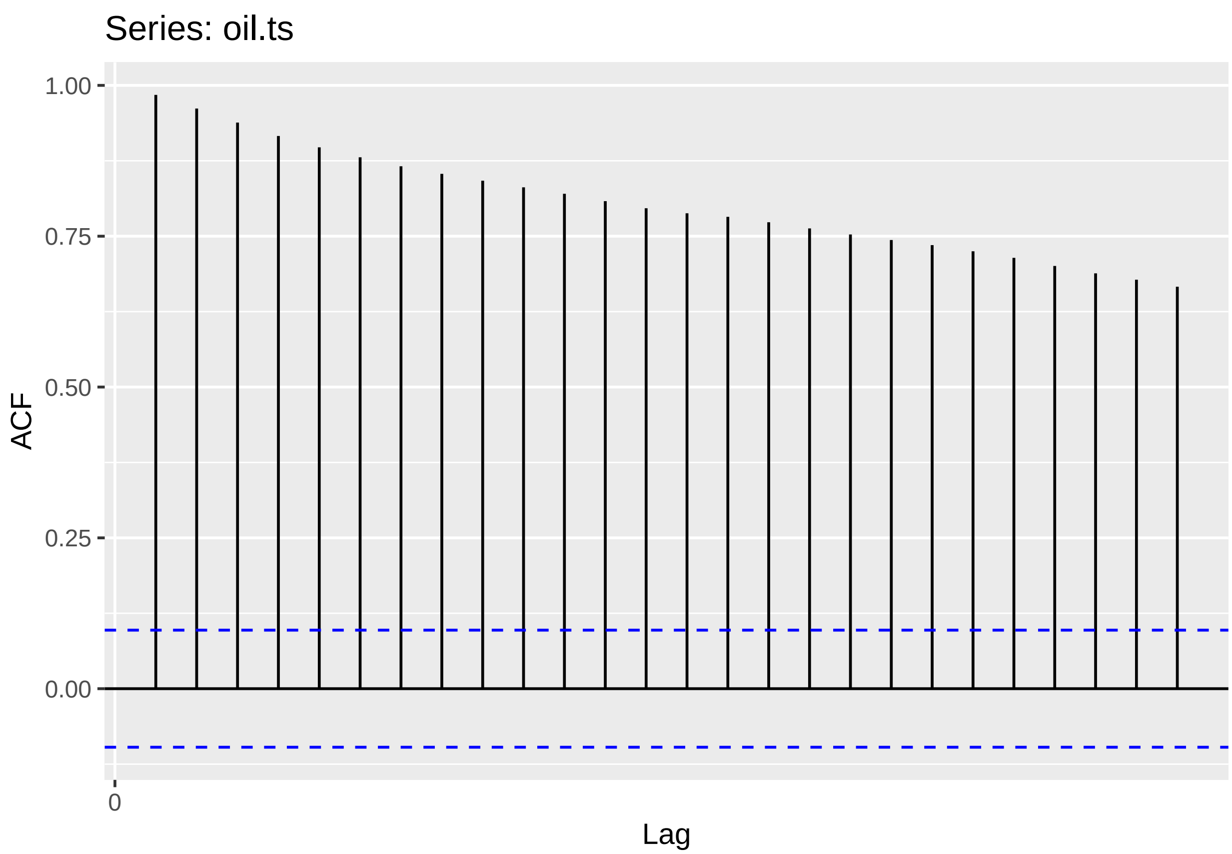

```
#inspection of diff vs. original (stationarity)
autoplot(acf(diff(oil.ts)))
```


Series diff(oil.ts)
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
Lag




```
autoplot(acf(oil.ts))
```


Series oil.ts
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
Lag



Comment ACF & PACF B&C TRANSFORMED ts: + ACF + Since the ACF spikes are all significant, the series is not stationary, therefore it should be differentiated.

Comment ACF & PACF 1 diff B&C ts: + ACF: + No clear pattern. + Significant spikes for lag = 2, 4, 13, 24 + No sharp drop after after q number of lags

- PACF:
  - No clear pattern.
  - Significant spikes for lag = 2, 4
  - no geometrical pattern, but it might be slightly decreasing
- Summary:
  - Many significant lags
  - 1st and 2nd lag changes sign
  - regular plots:
    - 1 lag is highly significant

Model Assumptions: + ARIMA(0,1,0) + Non Seasonal Part: + 0 AR - no geometric trend in ACF followed by q significant lags + 1 I due to non stationary process, 1 diff for Y variables to be mean variance stationary + 0 MA - no geometric trend in PACF followed by p significant lags + Seasonal Part: + 2 - Significant spikes for lag 24 and 36, therefore add seasonal order of 2 (check up what it's called)

- ETS (Error, Trend, Seasonal) (A = Additive, M = Multiplicative, Z = Automatic, Ad = Additive damped)
- ETS(A,N,N)

- Error: A - The series has been B&C transformed, therefore the error component is additive
- Trend: N - After Unit root handling there is no trend present (also what decomposition plot indicates)
- Seasonal: N - ref. WO test, no seasonality present

7 Section 3.1

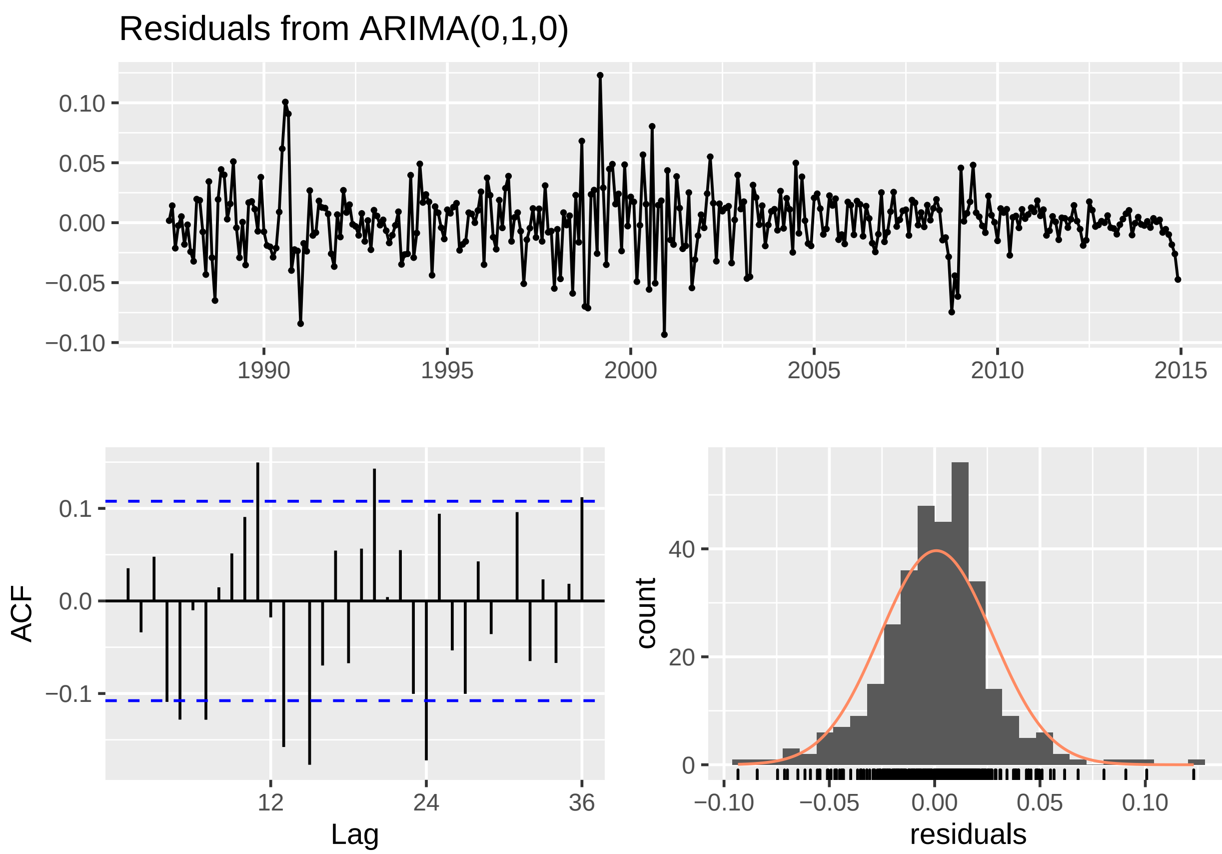
```
# split train test (~ 80/20 split)

train <- window(bc_oil.ts, end = c(2014, 12))
test <- window(bc_oil.ts, start = c(2015, 1))
h <- length(test)

train_og <- window(oil.ts, end = c(2014, 12))
train_log <- window(log_oil.ts, end = c(2014, 12))

coronaperiod <- window(bc_oil.ts, start = c(2019, 11))


#check autocorrelation for train necessary?
```

8 Section 3.2

```
#MODELS

#modelling procedure (https://otexts.com/fpp2/arima-r.html)

#hand model ARIMA
fit1.1 <- Arima(train, order = c(0, 1, 0))
summary(fit1.1)
```

```
## Series: train
## ARIMA(0,1,0)
##
## sigma^2 estimated as 0.0007124:  log likelihood=727.49
## AIC=-1452.97   AICc=-1452.96   BIC=-1449.17
##
## Training set error measures:
##                        ME       RMSE        MAE       MPE    MAPE      MASE
## Training set 0.0007732214 0.02665054 0.01930365 0.0305291 1.09186 0.3212321
##                    ACF1
## Training set 0.03536172
```

```
checkresiduals((fit1.1)) #Ljung Box Test - H0: Residuals are independent (Rejected 1%, they are correla
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,0)
## Q* = 75.977, df = 24, p-value = 2.626e-07
##
## Model df: 0.   Total lags used: 24
```

```
shapiro.test(residuals(fit1.1)) # H0: Normally Distributed (Rejected, not normally distributed)
```

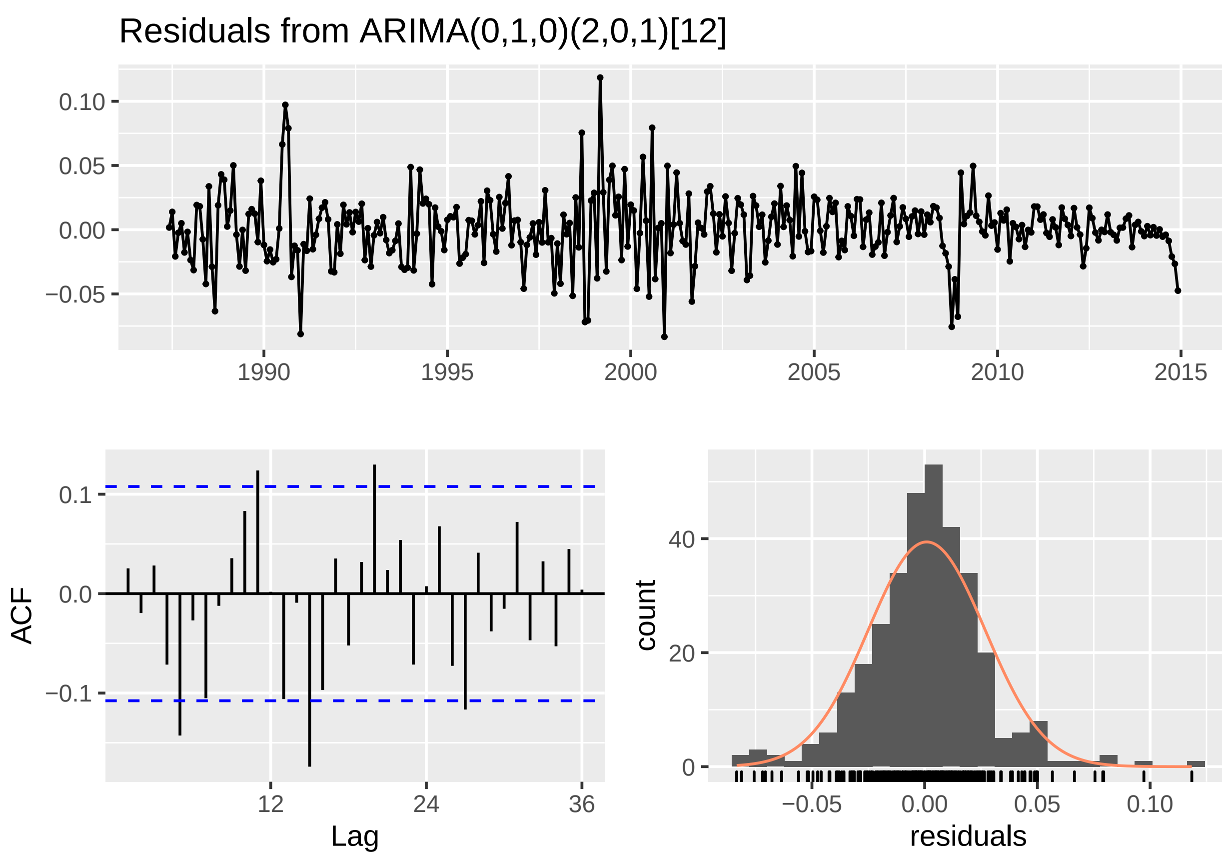
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit1.1)
## W = 0.96162, p-value = 1.224e-07
```

```
# significant lags ACF: 24 & 36 (+)
# Ljung-Box test (H0: Model is fine) rejects null hypothesis that the time series isn't auto correlated
  # Conclusion: Residuals are auto correlated
# Variance appears to reduce over time, which might indicate the presence of heteroskedacity

fit1.2 <- Arima(train, order = c(0,1,0), seasonal = c(2, 0, 1))
summary(fit1.2)
```

```
## Series: train
```

```
## ARIMA(0,1,0)(2,0,1)[12]
##
## Coefficients:
##          sar1     sar2    sma1
##       -0.5259  -0.1921  0.5322
## s.e.   0.1698   0.0574  0.1702
##
## sigma^2 estimated as 0.0006833:  log likelihood=735.18
## AIC=-1462.37   AICc=-1462.25   BIC=-1447.17
##
## Training set error measures:
##                        ME       RMSE        MAE        MPE     MAPE     MASE
## Training set 0.000901189 0.02598219 0.01881689 0.03731643 1.063242 0.313132
##                    ACF1
## Training set 0.02546586
```

```
checkresiduals((fit1.2))
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,0)(2,0,1)[12]
## Q* = 49.962, df = 21, p-value = 0.0003692
##
## Model df: 3.   Total lags used: 24
```

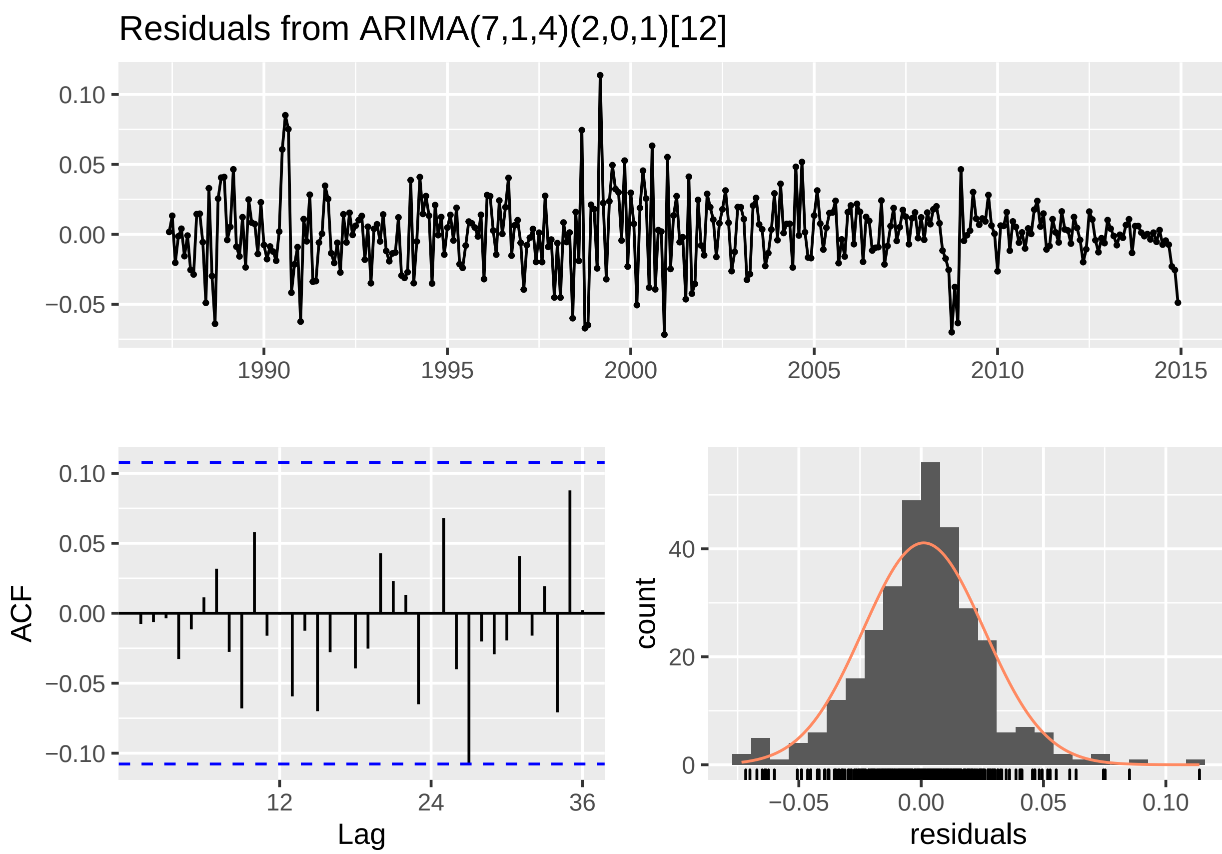
```r
shapiro.test(residuals(fit1.2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit1.2)
## W = 0.96502, p-value = 3.843e-07
```

```r
fit1.3 <- Arima(train, order = c(7,1,4), seasonal = c(2, 0, 1))
summary(fit1.3)
```

```
## Series: train
## ARIMA(7,1,4)(2,0,1)[12]
##
## Coefficients:
##          ar1      ar2     ar3      ar4      ar5      ar6      ar7      ma1
##       0.0673  -0.0571  0.1135  -0.7787  -0.0927  -0.0445  -0.1226  -0.0524
## s.e.  0.1149   0.1269  0.0936   0.0708   0.0710   0.0622   0.0691   0.1062
##          ma2      ma3     ma4      sar1     sar2    sma1
##       0.0265  -0.0860  0.7980  -0.6947  -0.1298  0.6579
## s.e.  0.1183   0.0866  0.0842   0.2554   0.0616  0.2479
##
## sigma^2 estimated as 0.0006447:  log likelihood=750
## AIC=-1469.99   AICc=-1468.46   BIC=-1413.01
##
## Training set error measures:
##                        ME       RMSE        MAE        MPE     MAPE     MASE
## Training set 0.001046399 0.02480936 0.0183064 0.04479473 1.033396 0.304637
##                      ACF1
## Training set -0.007606992
```

```r
checkresiduals(fit1.3)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(7,1,4)(2,0,1)[12]
## Q* = 10.363, df = 10, p-value = 0.4092
##
## Model df: 14.   Total lags used: 24
```

```
shapiro.test(residuals(fit1.3))
```

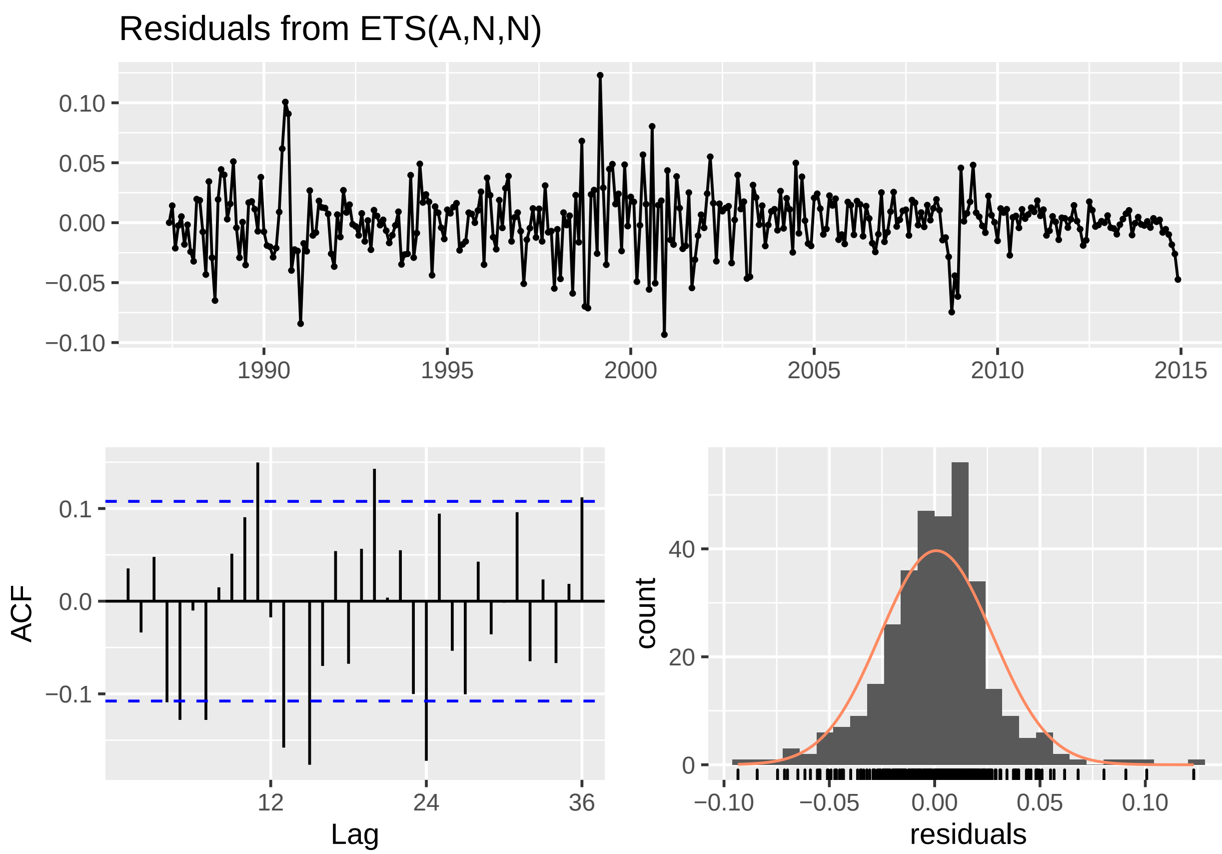
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit1.3)
## W = 0.97374, p-value = 9.933e-06
```

```
#auto ARIMA (what fit1.3 yielded)
#fit1.4 <- auto.arima(train, stepwise = F, approximation = F, max.p = 7,
#                    max.q = 7,
#                    max.P = 7,
#                    max.Q = 7,
#                    max.order = 7,
#                    max.d = 7,
#                    max.D = 7, ic = 'aic')
```

```r
#summary(fit1.4)
#checkresiduals((fit1.4))
#shapiro.test(residuals(fit1.4)) # H0: Normally Distributed (Rejected, not normally distributed)



# Selected: fit1.2 / ARIMA(0,1,0)(2,0,1)[12] (principle of parsimony)

#hand model ETS
fit2.1 <- ets(train, model = ("ANN"))
summary(fit2.1)
```

```
## ETS(A,N,N)
##
## Call:
##  ets(y = train, model = ("ANN"))
##
##   Smoothing parameters:
##     alpha = 0.9999
##
##   Initial states:
##     l = 1.7038
##
##   sigma:  0.0267
##
##       AIC      AICc       BIC
## -473.2148 -473.1414 -461.8084
##
## Training set error measures:
##                        ME       RMSE        MAE        MPE     MAPE     MASE
## Training set 0.0007681659 0.02665047 0.01929859 0.03023073 1.091562 0.321148
##                    ACF1
## Training set 0.03536078
```

```r
checkresiduals((fit2.1))
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,N,N)
## Q* = 75.893, df = 22, p-value = 7.583e-08
##
## Model df: 2.   Total lags used: 24
```

```
shapiro.test(residuals(fit2.1))
```

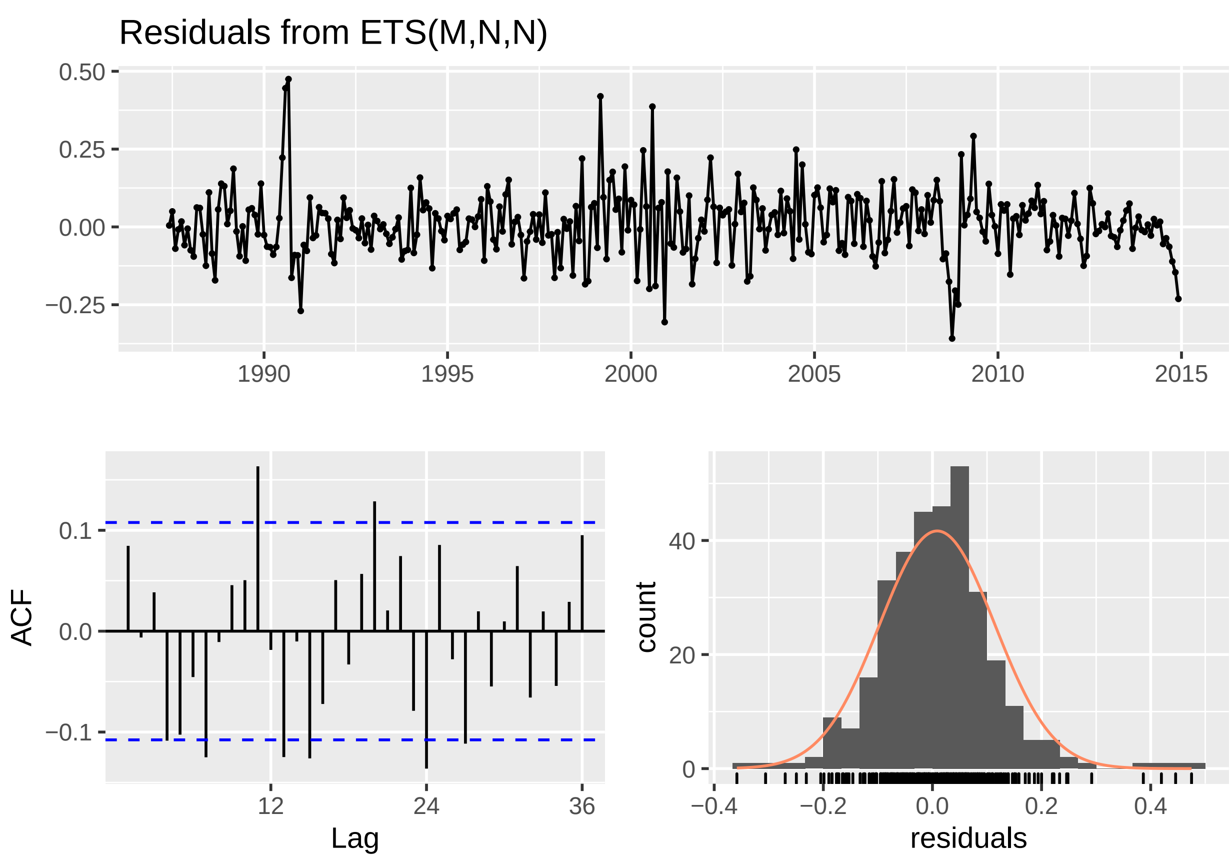
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit2.1)
## W = 0.96164, p-value = 1.231e-07
```

```
fit2.2 <- ets(train_og, model = ('MNN'))
summary(fit2.1)
```

```
## ETS(A,N,N)
##
## Call:
##  ets(y = train, model = ("ANN"))
##
```

```
##   Smoothing parameters:
##     alpha = 0.9999
##
##   Initial states:
##     l = 1.7038
##
##   sigma:  0.0267
##
##        AIC       AICc        BIC
## -473.2148 -473.1414 -461.8084
##
## Training set error measures:
##                        ME       RMSE        MAE        MPE     MAPE     MASE
## Training set 0.0007681659 0.02665047 0.01929859 0.03023073 1.091562 0.321148
##                    ACF1
## Training set 0.03536078
```

```
checkresiduals((fit2.2))
```

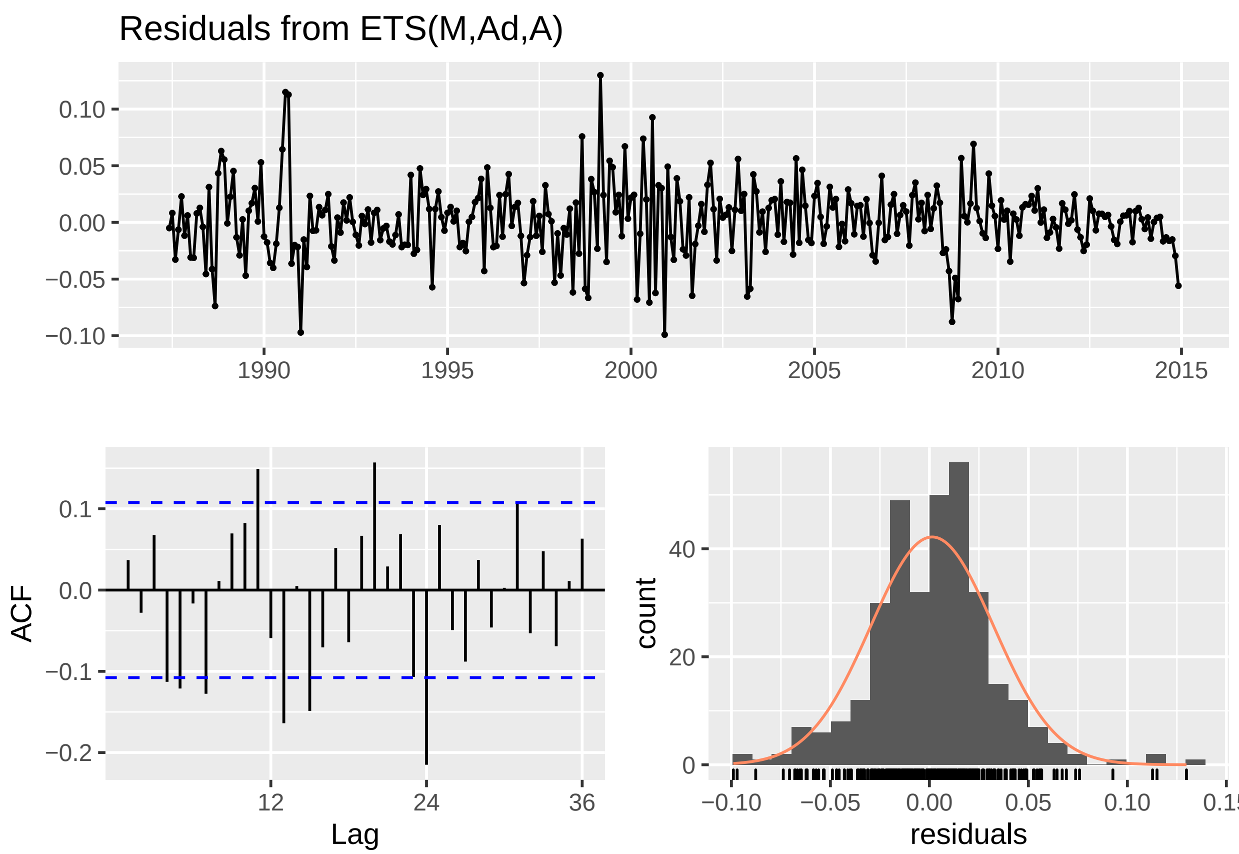



```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,N,N)
## Q* = 59.494, df = 22, p-value = 2.655e-05
##
## Model df: 2.   Total lags used: 24
```

```r
fit2.3 <- ets(train_log, model = 'MAA', damped = T)
summary(fit2.3)
```

```
## ETS(M,Ad,A)
##
## Call:
##  ets(y = train_log, model = "MAA", damped = T)
##
##   Smoothing parameters:
##     alpha = 0.9998
##     beta  = 0.0068
##     gamma = 2e-04
##     phi   = 0.8098
##
##   Initial states:
##     l = 2.9637
##     b = 0.0025
##     s = 0.0042 0.0174 -0.0063 -0.0458 -0.0328 -0.0434
##            -0.0199 0.0069 0.0558 0.0456 0.0203 -0.0022
##
##   sigma:  0.032
##
##      AIC     AICc      BIC
## 477.7184 479.9107 546.1565
##
## Training set error measures:
##                       ME      RMSE        MAE        MPE     MAPE      MASE
## Training set 0.003135537 0.1026466 0.07799246 0.04308142 2.320541 0.3139938
##                    ACF1
## Training set 0.06700029
```

```r
checkresiduals(fit2.3)
```



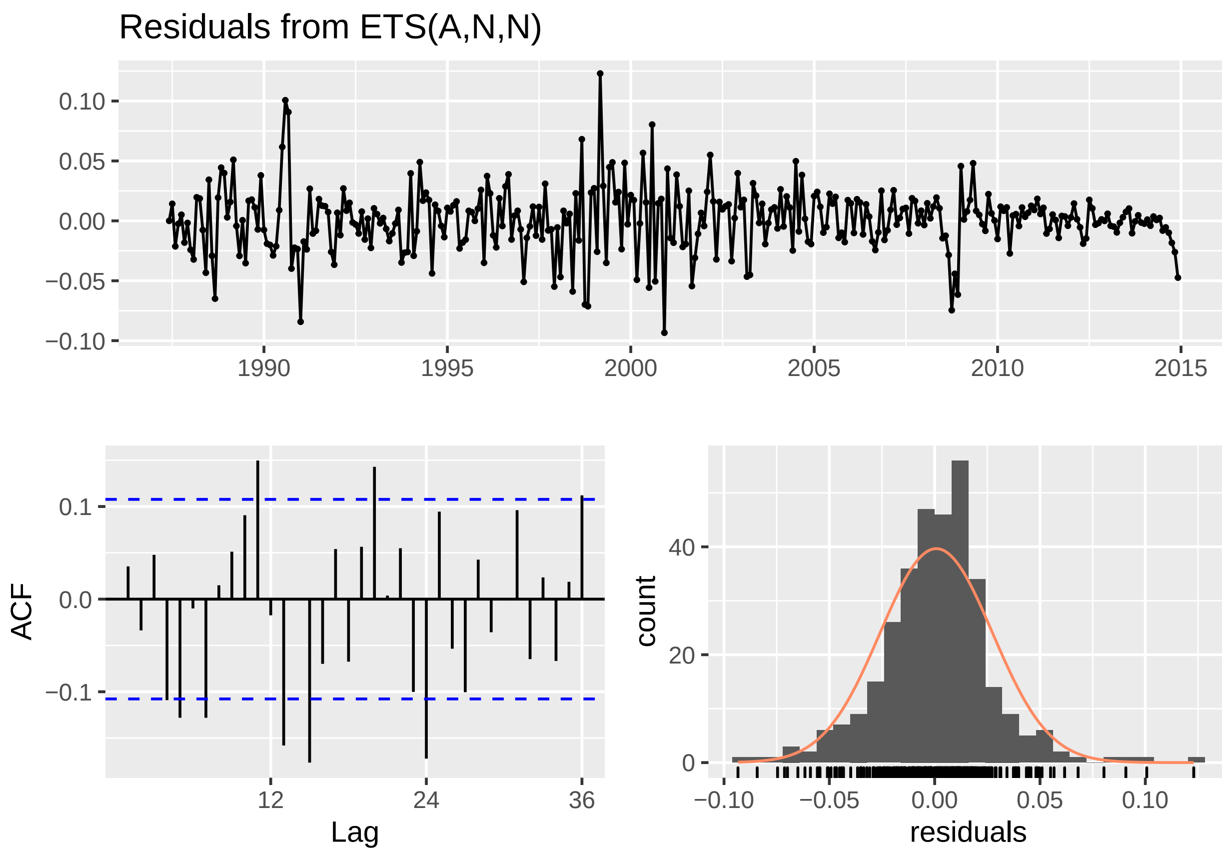

```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,Ad,A)
## Q* = 84.187, df = 7, p-value = 1.887e-15
##
## Model df: 17.   Total lags used: 24
```

```
#auto model ETS
fit2.4 <- ets(train, model = ("ZZZ"), ic = 'aic')
summary(fit2.4)
```

```
## ETS(A,N,N)
##
## Call:
##  ets(y = train, model = ("ZZZ"), ic = "aic")
##
##   Smoothing parameters:
##     alpha = 0.9999
##
##   Initial states:
##     l = 1.7038
##
##   sigma:  0.0267
##
```

```
##       AIC      AICc       BIC 
## -473.2148 -473.1414 -461.8084 
## 
## Training set error measures:
##                        ME       RMSE        MAE        MPE     MAPE     MASE
## Training set 0.0007681659 0.02665047 0.01929859 0.03023073 1.091562 0.321148
##                    ACF1
## Training set 0.03536078
```

```
checkresiduals((fit2.4))
```




```
## 
##  Ljung-Box test
## 
## data:  Residuals from ETS(A,N,N)
## Q* = 75.893, df = 22, p-value = 7.583e-08
## 
## Model df: 2.   Total lags used: 24
```

```
shapiro.test(residuals(fit2.4))
```

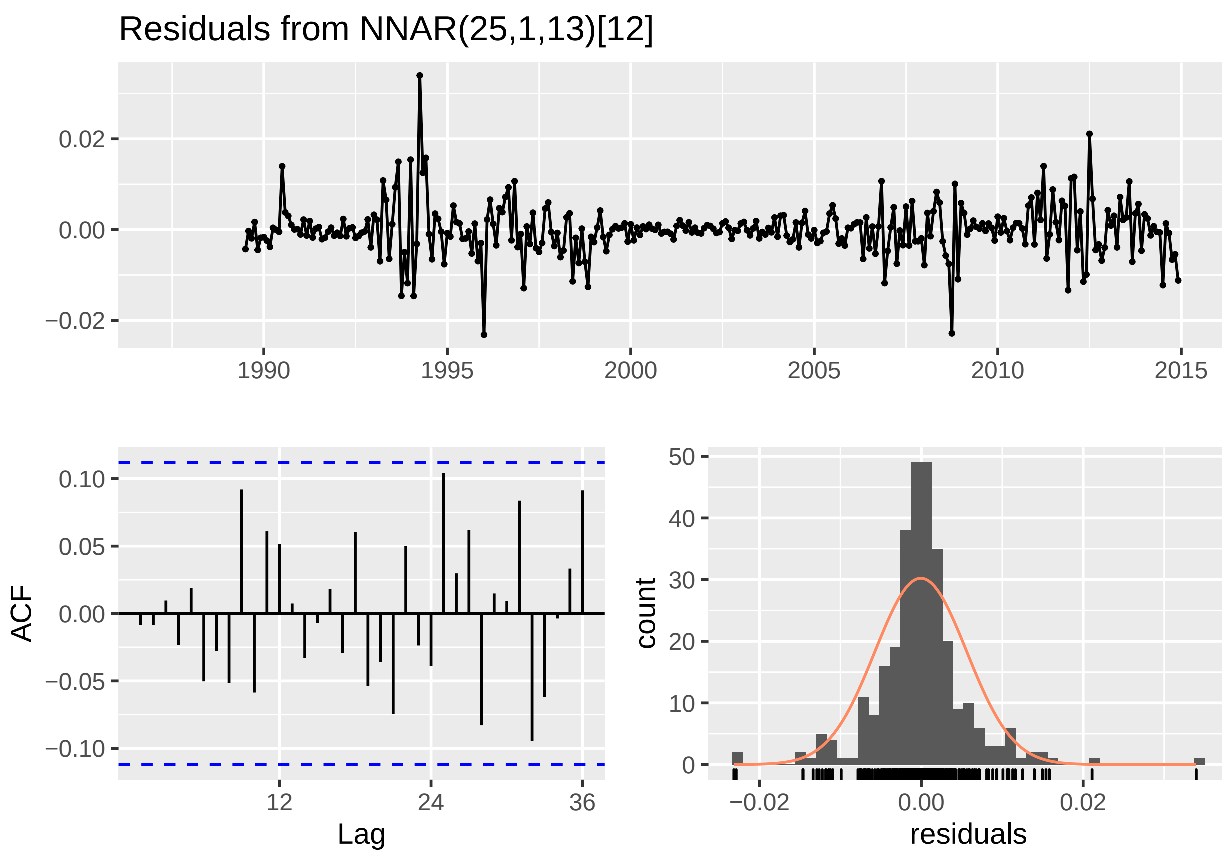
```
## 
##  Shapiro-Wilk normality test
## 
## data:  residuals(fit2.4)
## W = 0.96164, p-value = 1.231e-07
```

```r
# Selected: fit3.1 = fit3.4 / ETS(ANN) (lowest on all IC)

#auto model ANN
fit3.1 <- nnetar(train)
summary(fit3.1)
```

```
##           Length Class        Mode
## x         331    ts           numeric
## m           1    -none-       numeric
## p           1    -none-       numeric
## P           1    -none-       numeric
## scalex      2    -none-       list
## size        1    -none-       numeric
## subset    331    -none-       numeric
## model      20    nnetarmodels list
## nnetargs    0    -none-       list
## fitted    331    ts           numeric
## residuals 331    ts           numeric
## lags       25    -none-       numeric
## series      1    -none-       character
## method      1    -none-       character
## call        2    -none-       call
```

```r
checkresiduals((fit3.1))
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```




```
Box.test(residuals(fit3.1), lag = 25)
```

```
##
##  Box-Pierce test
##
## data:  residuals(fit3.1)
## X-squared = 17.323, df = 25, p-value = 0.8697
```

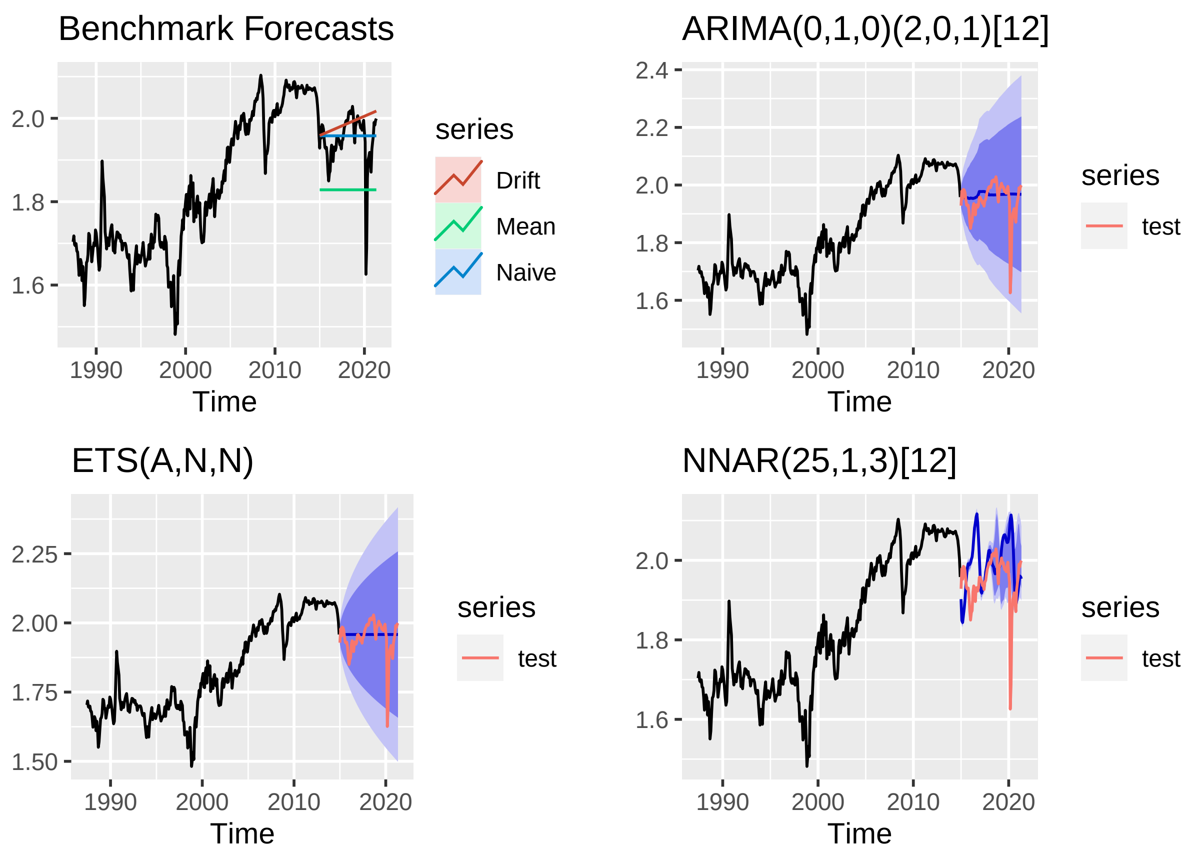
```
shapiro.test(residuals(fit3.1))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit3.1)
## W = 0.90638, p-value = 7.548e-13
```

Desired Properties of Residuals: 1. Uncorrelated (if sufficient model) 2. Zero Mean (if sufficient model) 3. Constant Variance (beneficial) 4. Normal Distribution (beneficial)

9 Section 4.1

```r
#FORECASTS

#benchmark forecasts
for.b1  <- meanf(train, h = h, level = c(80, 95))
for.b2  <- naive(train, PI = T, h = h)
for.b3  <- rwf(train, PI = T, h = h, drift = T)


#ARIMA, ETS, NNAR
for.autoar <- forecast(fit1.2, h = h)
for.autoets <- forecast(fit2.4, h = h)
for.nnetar <- forecast(fit3.1, h = h, PI = T)


#combined plot
grid.arrange((autoplot(bc_oil.ts)
              + ylab("")
              + autolayer(meanf(train, h = h), series = 'Mean', PI = F)
              + autolayer(naive(train, h = h), series = 'Naive', PI = F)
              + autolayer(rwf(train, h = h, drift = T), series = 'Drift', PI = F)
              + ggtitle("Benchmark Forecasts")),
             (autoplot(for.autoar)
              + ylab("")
              + autolayer(test)
              + ggtitle("ARIMA(0,1,0)(2,0,1)[12]")),
             (autoplot(for.autoets)
              + autolayer(test)
              + ylab("")
              + ggtitle("ETS(A,N,N)")),
             (autoplot(for.nnetar)
              + autolayer(test)
              + ylab("")

              + ggtitle("NNAR(25,1,3)[12]")),
             nrow = 2)
```



10 Section 4.2

```
#Accuracy measures [RMSE, MAE, MAPE, MASE] (https://otexts.com/fpp2/arima-ets.html)
accuracy(for.b1, test) #MEAN
```

```
##                        ME      RMSE       MAE        MPE     MAPE     MASE
## Training set 1.017070e-16 0.1654603 0.1476819 -0.8144646 8.081983 2.457576
## Test set     1.190878e-01 0.1348030 0.1279964  6.0058258 6.545479 2.129989
##                   ACF1 Theil's U
## Training set 0.9852447        NA
## Test set     0.7285703  2.783988
```

```
accuracy(for.b2, test) #NAIVE
```

```
##                         ME       RMSE        MAE         MPE     MAPE      MASE
## Training set  0.0007704014 0.02669072 0.01935698  0.03031858 1.094865 0.3221197
## Test set     -0.0104566944 0.06402558 0.04164963 -0.65327956 2.213835 0.6930918
##                    ACF1 Theil's U
## Training set 0.03530902        NA
## Test set     0.72857026  1.414839
```

```r
accuracy(for.b3, test) #DRIFT
```

```
##                         ME       RMSE        MAE         MPE     MAPE      MASE
## Training set  6.796015e-17 0.02667960 0.01931551 -0.01214836 1.092995 0.3214297
## Test set     -4.050235e-02 0.07672608 0.04808646 -2.19796825 2.576115 0.8002072
##                    ACF1 Theil's U
## Training set 0.03530902        NA
## Test set     0.74879319  1.707213
```

```r
accuracy(for.autoar,test) #ARIMA
```

```
##                        ME       RMSE        MAE         MPE     MAPE      MASE
## Training set  0.000901189 0.02598219 0.01881689  0.03731643 1.063242 0.3131320
## Test set     -0.017883000 0.06520805 0.04167171 -1.03368137 2.222554 0.6934593
##                    ACF1 Theil's U
## Training set 0.02546586        NA
## Test set     0.72558703  1.447245
```

```r
accuracy(for.autoets,test) #ETS
```

```
##                         ME       RMSE        MAE         MPE     MAPE      MASE
## Training set  0.0007681659 0.02665047 0.01929859  0.03023073 1.091562 0.3211480
## Test set     -0.0104614390 0.06402636 0.04164993 -0.65352345 2.213856 0.6930969
##                    ACF1 Theil's U
## Training set 0.03536078        NA
## Test set     0.72857026  1.414859
```

```r
accuracy(for.nnetar, test) #NNETAR
```

```
##                        ME        RMSE         MAE          MPE      MAPE
## Training set -6.08499e-05 0.005665004 0.003688737 -0.005818094 0.2017653
## Test set     -4.43420e-02 0.111306790 0.074754501 -2.427946796 3.9631471
##                    MASE         ACF1 Theil's U
## Training set 0.06138431 -0.008551557        NA
## Test set     1.24399031  0.853117541  2.459446
```

```r
#Accuracy measures corona period [2019(11) - CTD]
accuracy(for.b1, coronaperiod) #MEAN
```

```
##                        ME      RMSE       MAE        MPE     MAPE     MASE
## Training set 1.017070e-16 0.1654603 0.1476819 -0.8144646 8.081983 2.457576
## Test set     8.464857e-02 0.1295530 0.1207518  4.1460528 6.333068 2.009431
##                   ACF1 Theil's U
## Training set 0.9852447        NA
## Test set     0.5745000  1.391838
```

```r
accuracy(for.b2, coronaperiod) #NAIVE
```

```
##                        ME       RMSE        MAE         MPE     MAPE      MASE
## Training set  0.0007704014 0.02669072 0.01935698  0.03031858 1.094865 0.3221197
## Test set     -0.0448959165 0.10786216 0.06703195 -2.64480998 3.756314 1.1154792
##                    ACF1 Theil's U
## Training set 0.03530902        NA
## Test set     0.57450001  1.288377
```

```
accuracy(for.b3, coronaperiod) #DRIFT
```

```
##                        ME      RMSE        MAE         MPE     MAPE      MASE
## Training set  6.796015e-17 0.0266796 0.01931551 -0.01214836 1.092995 0.3214297
## Test set     -9.728321e-02 0.1372243 0.09728321 -5.38694703 5.386947 1.6188907
##                    ACF1 Theil's U
## Training set 0.03530902        NA
## Test set     0.56330047  1.636391
```

```
accuracy(for.autoar,coronaperiod) #ARIMA
```

```
##                       ME       RMSE        MAE         MPE     MAPE     MASE
## Training set  0.000901189 0.02598219 0.01881689  0.03731643 1.063242 0.313132
## Test set     -0.055186765 0.11273656 0.07016052 -3.18492542 3.936263 1.167542
##                    ACF1 Theil's U
## Training set 0.02546586        NA
## Test set     0.57615171  1.349242
```

```
accuracy(for.autoets,coronaperiod) #ETS
```

```
##                        ME       RMSE        MAE         MPE     MAPE     MASE
## Training set  0.0007681659 0.02665047 0.01929859  0.03023073 1.091562 0.321148
## Test set     -0.0449006611 0.10786413 0.06703319 -2.64505869 3.756387 1.115500
##                    ACF1 Theil's U
## Training set 0.03536078        NA
## Test set     0.57450001  1.288402
```

```
accuracy(for.nnetar, coronaperiod) #NNETAR
```

```
##                       ME        RMSE         MAE          MPE      MAPE
## Training set -6.08499e-05 0.005665004 0.003688737 -0.005818094 0.2017653
## Test set     -8.28992e-02 0.170562771 0.108973471 -4.743585121 6.0648305
##                    MASE         ACF1 Theil's U
## Training set 0.06138431 -0.008551557        NA
## Test set     1.81342850  0.771156106  2.052229
```