
Forecasting on US Electricity data using ARIMA

Problem Background: The file titled **US Electricity.csv** includes a time series index compiled by the US Federal Reserve representing total fossil-fuel US electricity generation by all utilities from January 1939 through October 2021.

In the following code box we read the CSV file and set up the data as a *tsibble* and then we plot it and subset it to examine it.

```
library(fpp3)

## Warning: package 'fpp3' was built under R version 4.1.1

## -- Attaching packages ----- fpp3 0.4.0 --

## v tibble      3.1.2      v tsibble      1.0.1
## v dplyr       1.0.7      v tsibbledata 0.3.0
## v tidyr       1.1.3      v feasts      0.2.2
## v lubridate   1.7.10     v fable       0.3.1
## v ggplot2     3.3.5

## Warning: package 'tsibble' was built under R version 4.1.1

## Warning: package 'tsibbledata' was built under R version 4.1.1

## Warning: package 'feasts' was built under R version 4.1.1

## Warning: package 'fabletools' was built under R version 4.1.1

## Warning: package 'fable' was built under R version 4.1.1

## -- Conflicts ----- fpp3_conflicts --
## x lubridate::date()      masks base::date()
## x dplyr::filter()        masks stats::filter()
## x tsibble::intersect()   masks base::intersect()
## x tsibble::interval()   masks lubridate::interval()
## x dplyr::lag()           masks stats::lag()
## x tsibble::setdiff()     masks base::setdiff()
## x tsibble::union()       masks base::union()

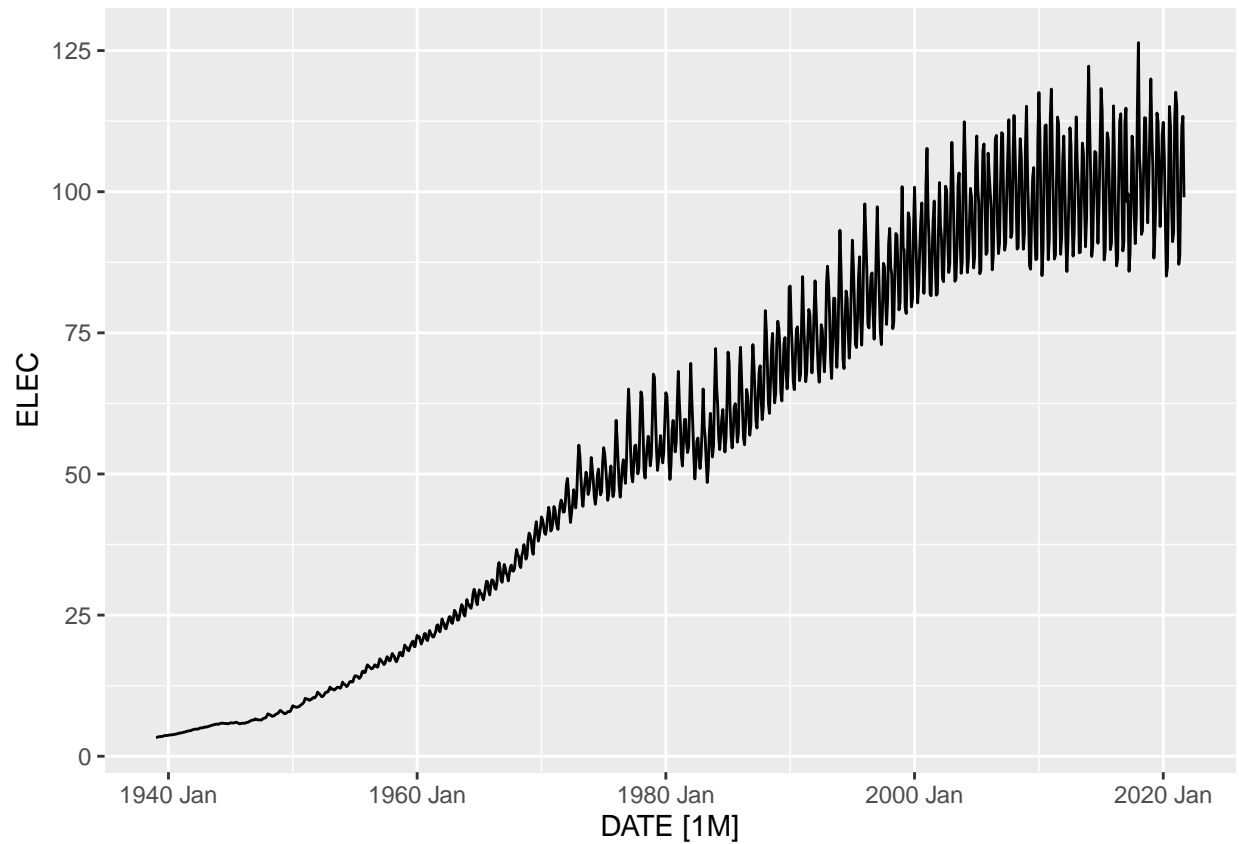
library(tseries)

## Warning: package 'tseries' was built under R version 4.1.1

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

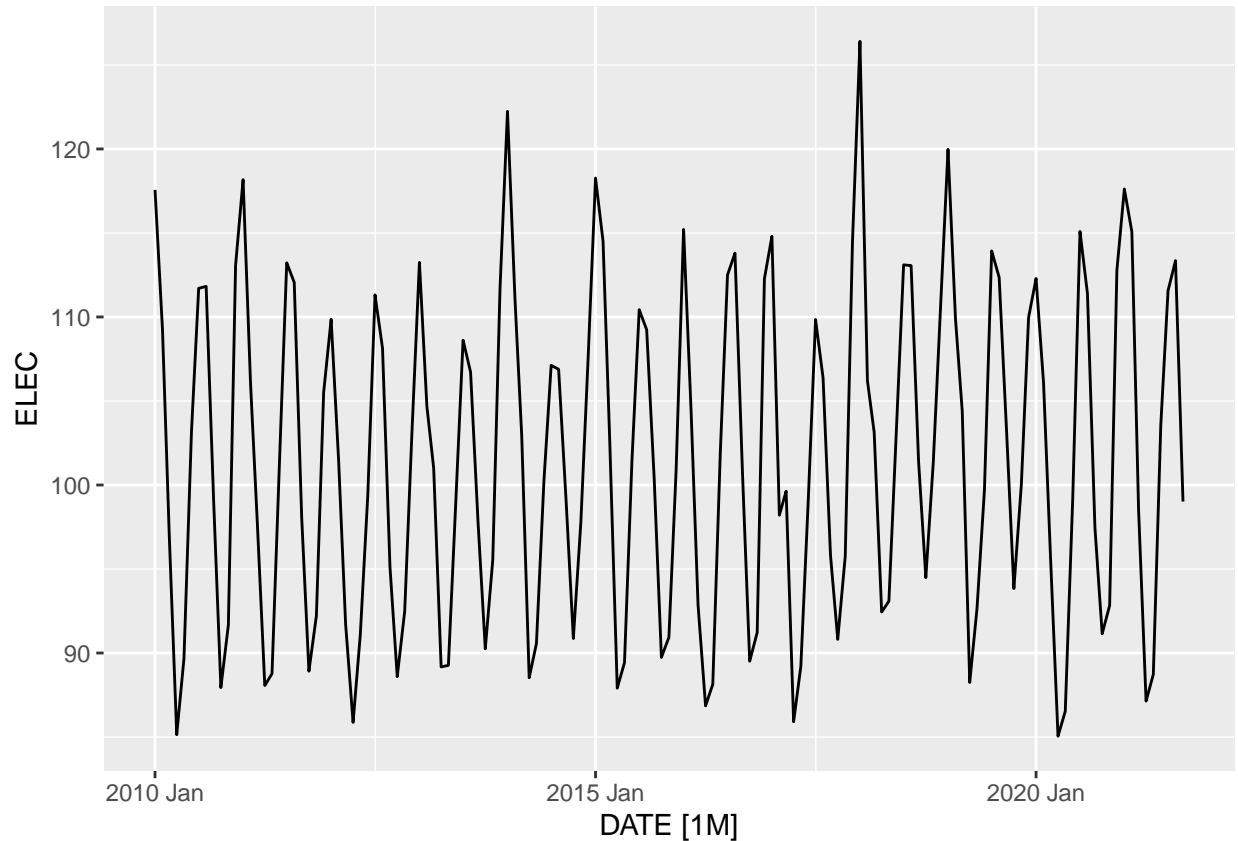
```
D <- read.csv("US Electricity.csv") %>%
  mutate(
    DATE = yearmonth(DATE)
  ) %>%
  as_tsibble(index = DATE)

D %>% autoplot(ELEC)
```



```
DR <- D %>% filter(
  DATE >= yearmonth("2010 Jan")
)

DR %>% autoplot(ELEC)
```



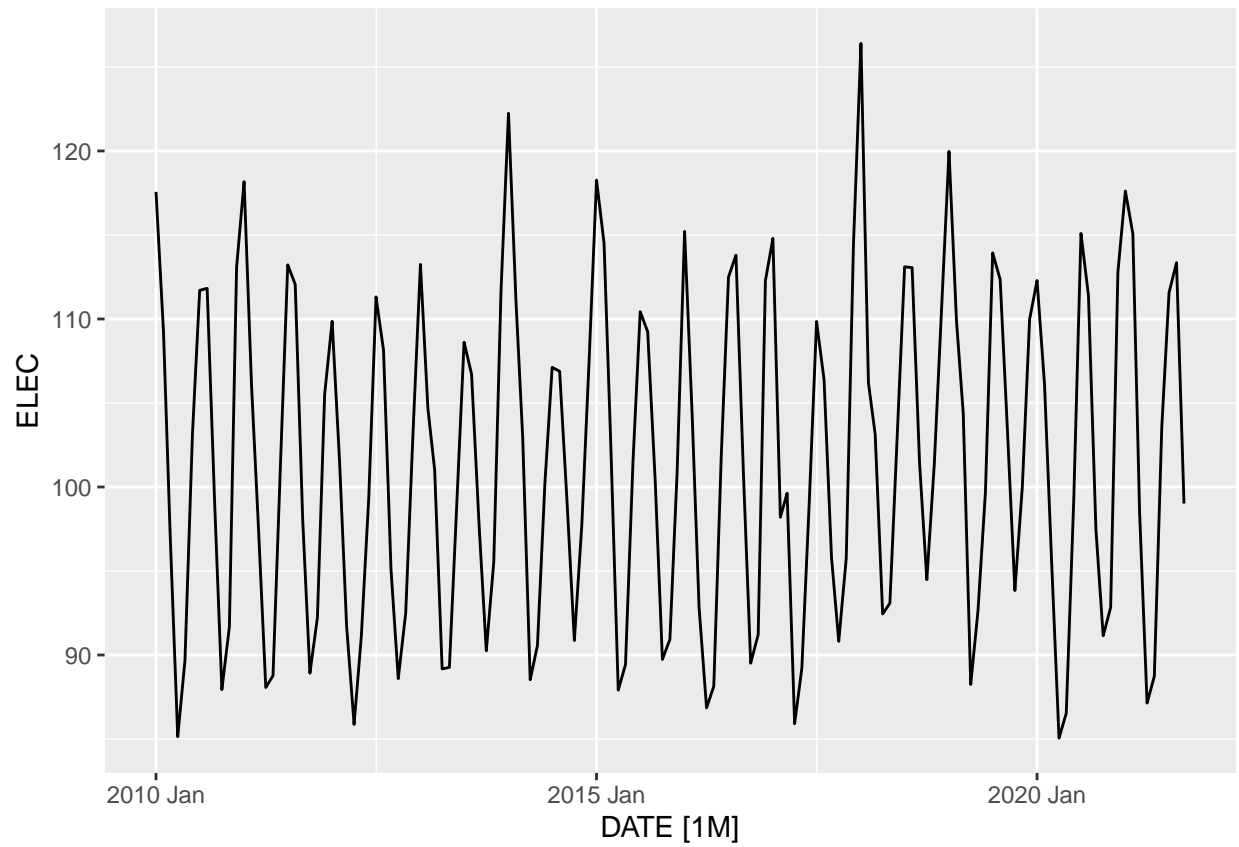
We are interested in developing a two-year long monthly forecast (24 months) for the national electricity production requirements.

1. First, we examine the stationarity of the **ELEC** time series in the reduced **DR** data, and also examine the corresponding ACF and PACF diagrams and then propose three plausible ARIMA models to fit the data.

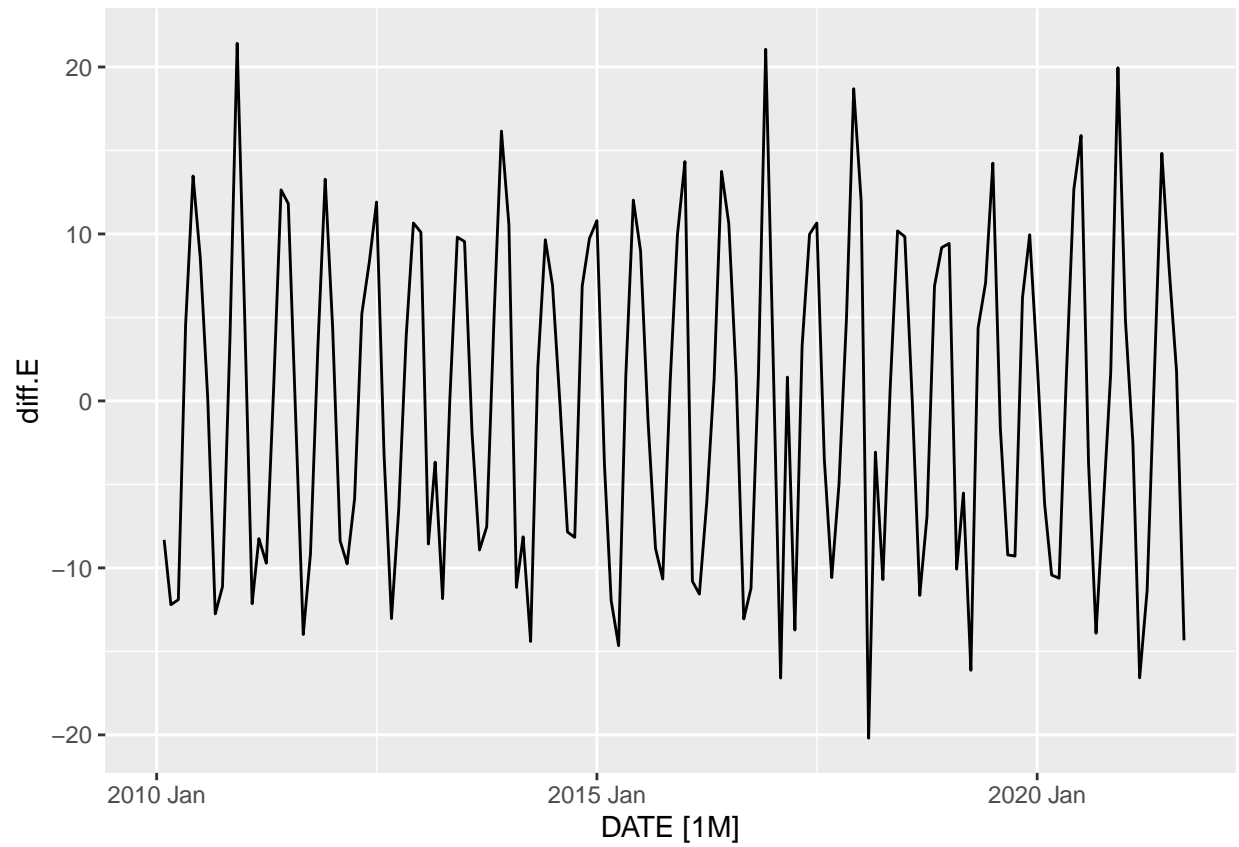
Solution:

```
DR %>%
  mutate(diff.E = difference(ELEC),
         diff2.E = difference(diff.E)) -> DR

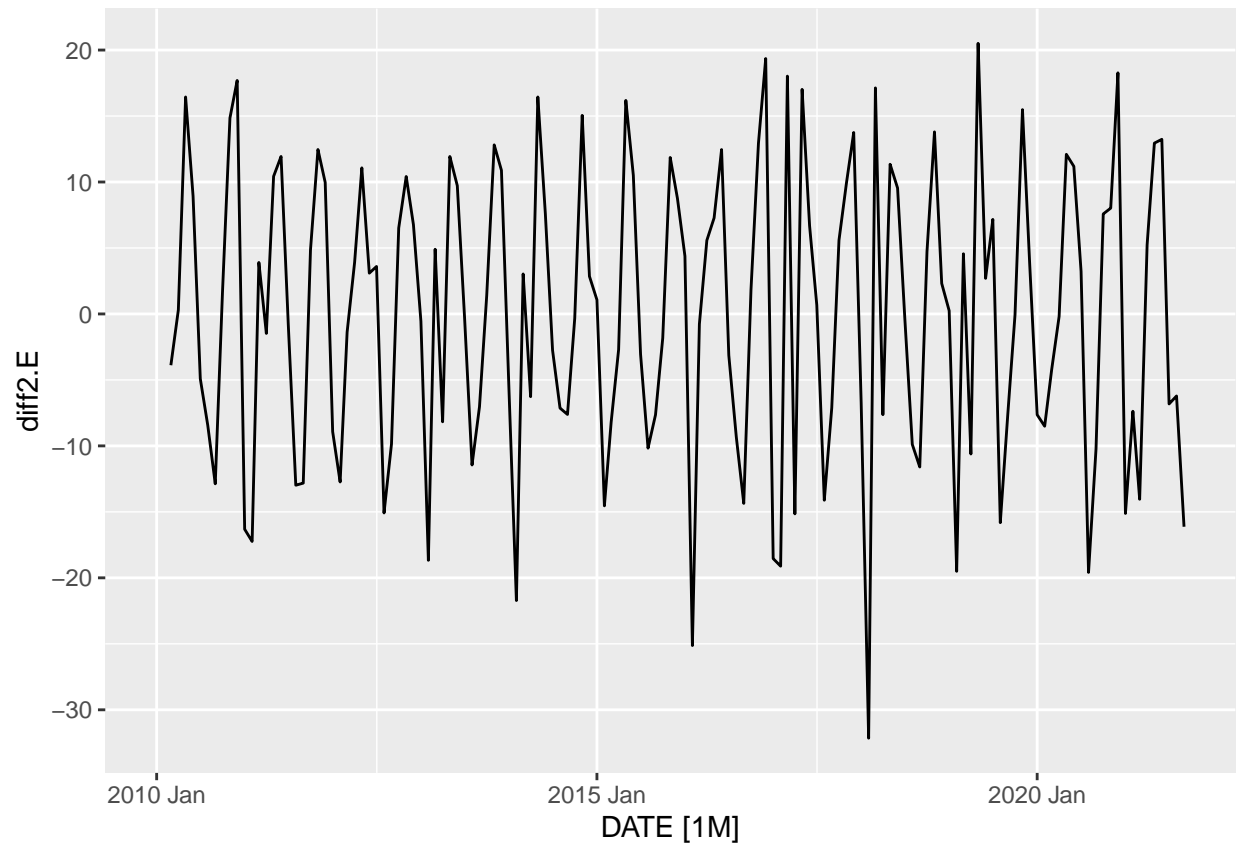
DR %>% autoplot(.vars = ELEC)
```



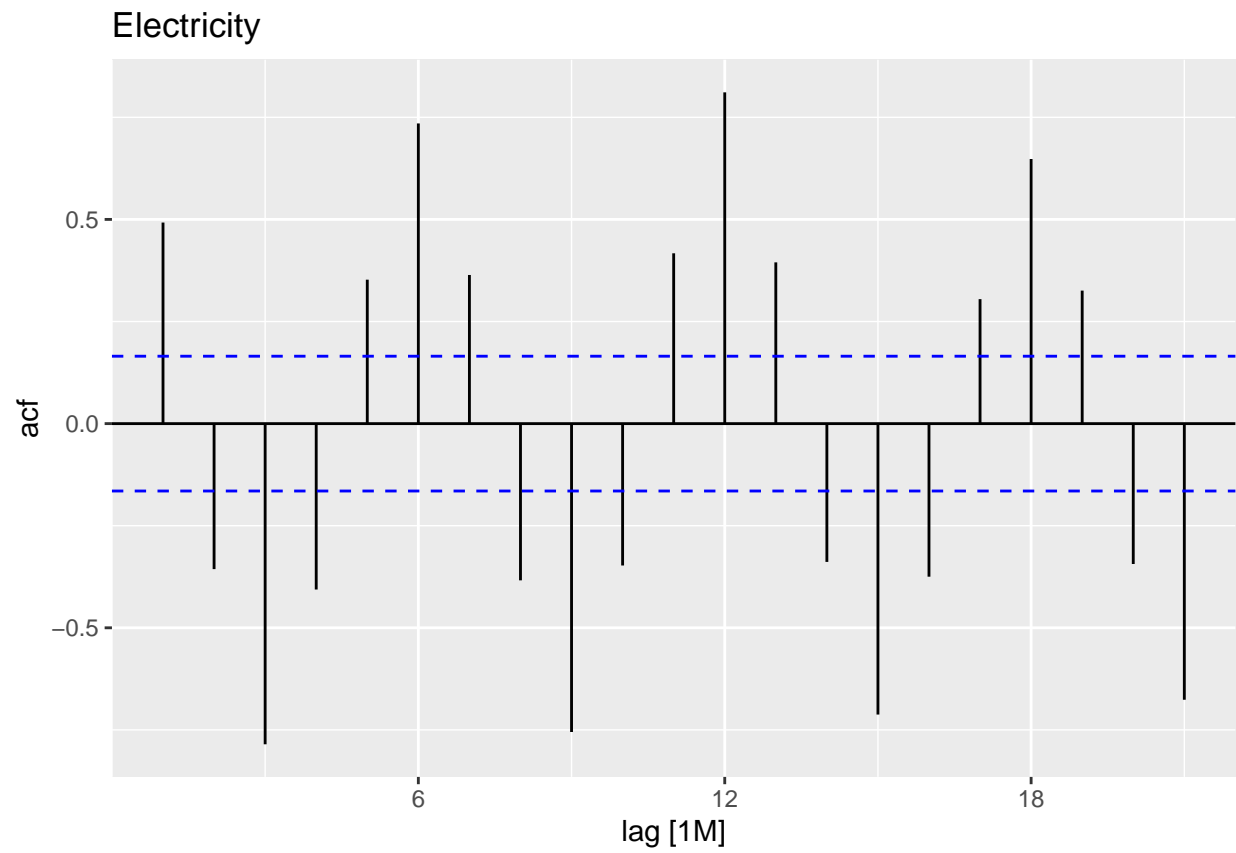
```
DR %>% autoplot(.vars = diff.E)
```



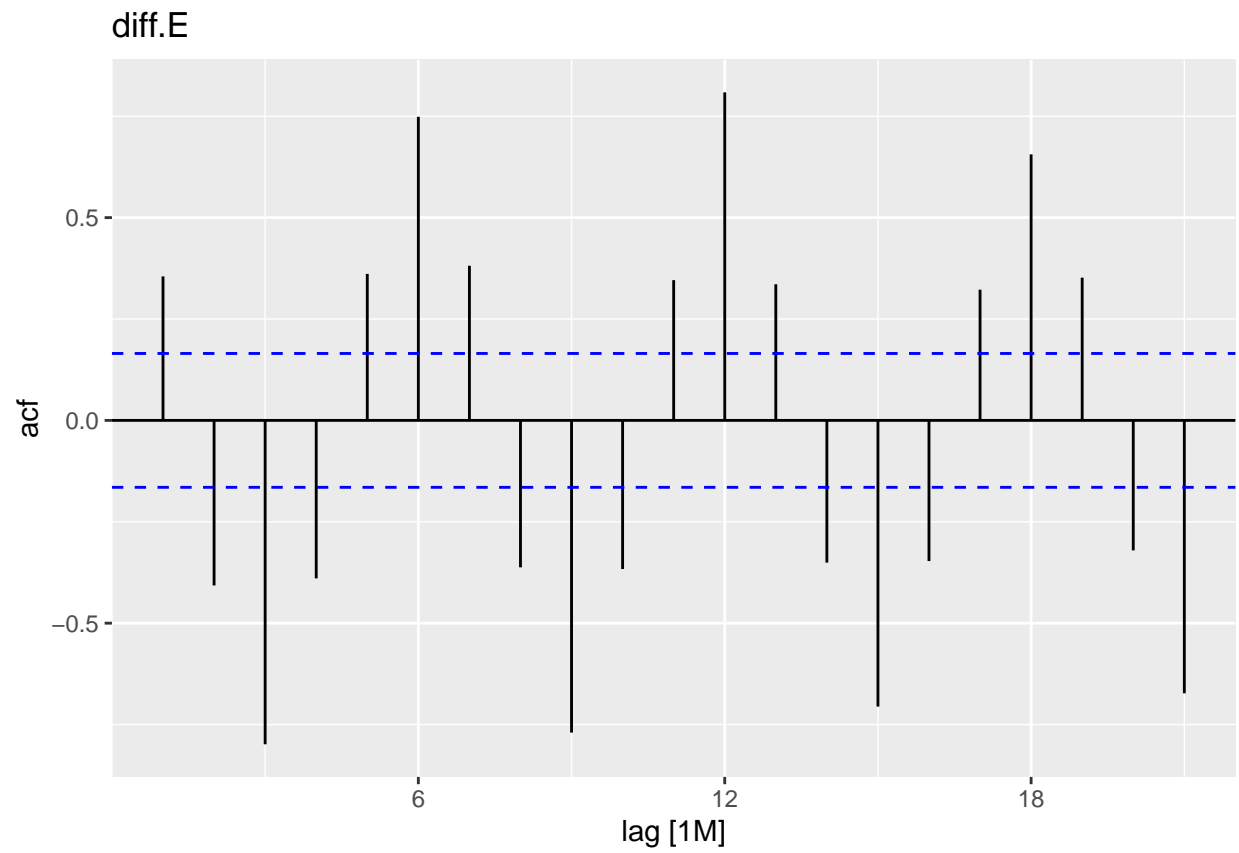
```
DR %>% autoplot(.vars = diff2.E)
```



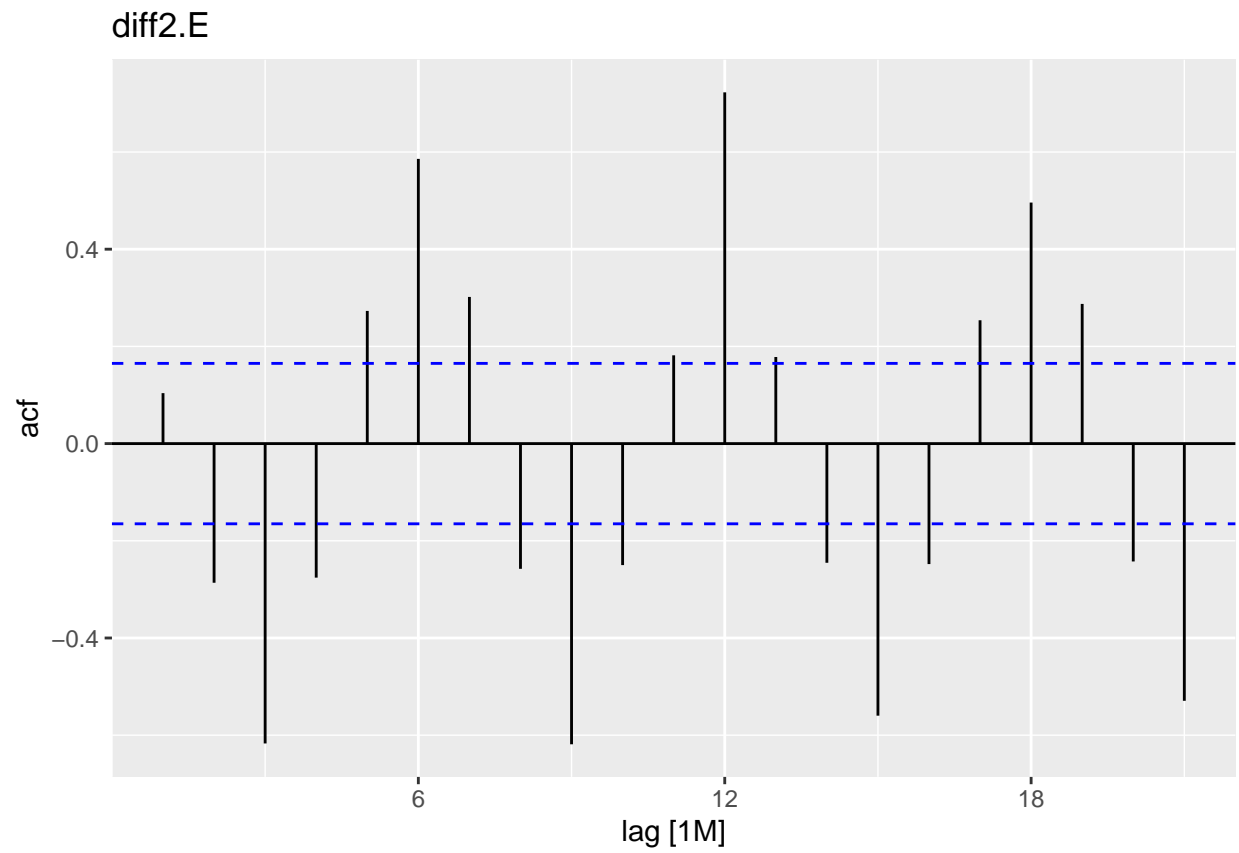
```
# Examine Stationarity Visually  
DR %>% ACF(ELEC) %>%  
  autoplot() +  
  labs(title = "Electricity")
```



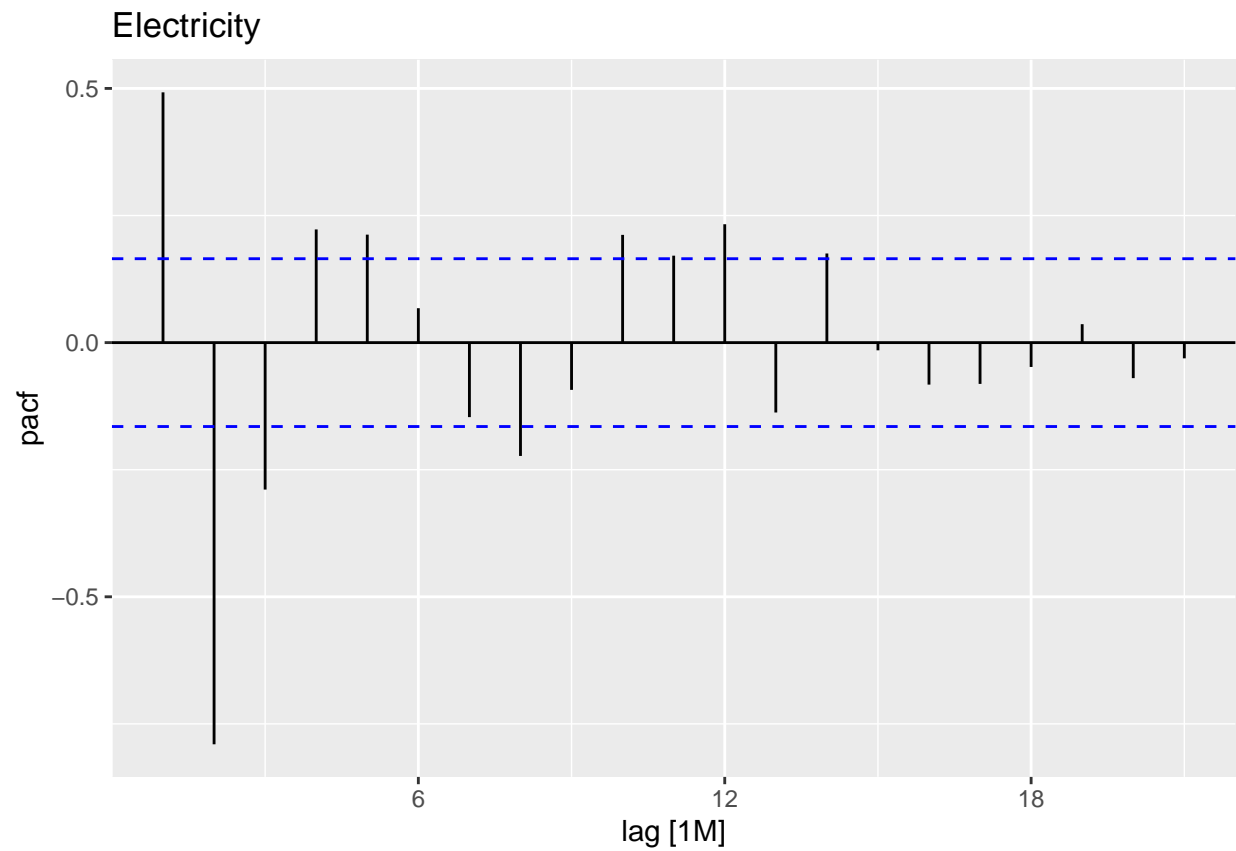
```
DR %>% ACF(diff.E) %>%  
  autoplot() +  
  labs(title = "diff.E")
```



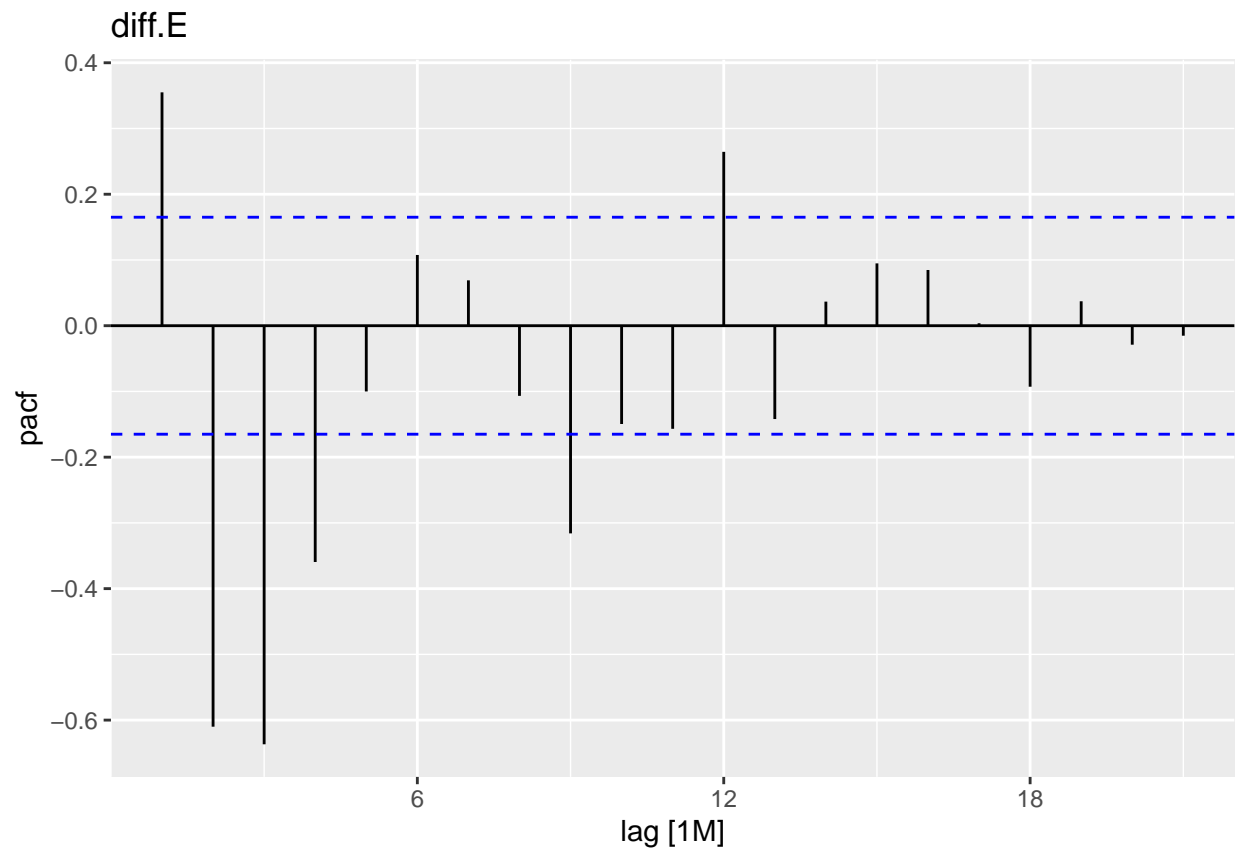
```
DR %>% ACF(diff2.E) %>%  
  autoplot() +  
  labs(title = "diff2.E")
```

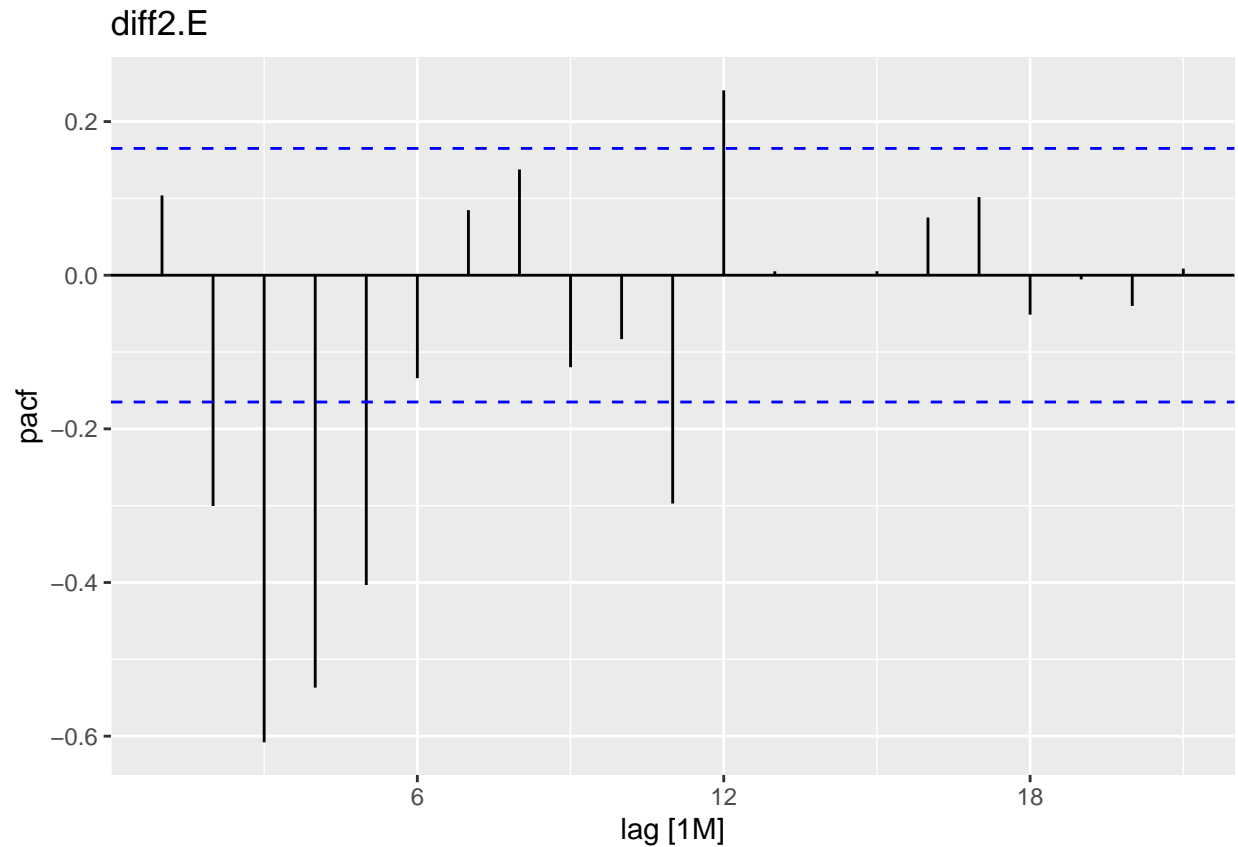
```
# PACF
DR %>% PACF(ELEC) %>%
  autoplot() +
  labs(title = "Electricity")
```



```
DR %>% PACF(diff.E) %>%  
  autoplot() +  
  labs(title = "diff.E")
```



```
DR %>% PACF(diff2.E) %>%  
  autoplot() +  
  labs(title = "diff2.E")
```



Based on the above graphs, the time series looks stationary. The ACF and PACF show seasonality. We further conduct KPSS and ADF tests to confirm our initial understanding:

```
DR %>% features(ELEC, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##   <dbl>     <dbl>
## 1      0.222         0.1
```

```
DR %>% features(diff.E, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##   <dbl>     <dbl>
## 1      0.0472         0.1
```

```
DR %>% features(diff2.E, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##   <dbl>     <dbl>
## 1      0.0607         0.1
```

```
DR %>% features(ELEC, unitroot_ndiffs)
```

```
## # A tibble: 1 x 1
##   ndiffs
##   <int>
## 1     0
```

```
DR$ELEC %>% adf.test()
```

```
##
## Augmented Dickey-Fuller Test
##
## data: .
## Dickey-Fuller = -3.8581, Lag order = 5, p-value = 0.01823
## alternative hypothesis: stationary
```

```
DR$diff.E %>%
  na.omit() %>%
  adf.test()
```

```
##
## Augmented Dickey-Fuller Test
##
## data: .
## Dickey-Fuller = -7.3944, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
DR$diff2.E %>%
  na.omit() %>%
  adf.test()
```

```
##
## Augmented Dickey-Fuller Test
##
## data: .
## Dickey-Fuller = -14.314, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

Looking at the KPSS test p-value, it is greater than the threshold, hence we **fail to reject** the hypothesis, indicating that the time series is stationary. `## Null - stationary`

According to the ADF test p-value, it is below the threshold, hence, we can **reject** the hypothesis. This also confirms that the time series is stationary. `## Null - Not stationary`

Seasonality in data with 3 strong PACF terms, suggesting the 3 ARIMA models with $m=12$: **ARIMA1** : `pdq(3,0,0)(1,1,0)[12]` **ARIMA2** : `pdq(2,0,0)(1,1,0)[12]` **ARIMA3** : `pdq(1,0,0)(1,1,0)[12]`

2. Using **fable**, we fit the following five models to the **DR** data: (i)-(iii) the three models we proposed in (1), (iv) the automatically selected model by the `ARIMA()` function, and (v) the automatically selected model by the `ETS()` function.

```
m <- DR %>% model(m1 = ARIMA(ELEC ~ pdq(3,0,0) + PDQ(1,1,0)),
  m2 = ARIMA(ELEC ~ pdq(2,0,0) + PDQ(1,1,0)),
  m3 = ARIMA(ELEC ~ pdq(1,0,0) + PDQ(1,1,0)),
  m4 = ARIMA(ELEC),
  m5 = ETS(ELEC))

m %>% glance() %>%
  select(.model, AICc, BIC)
```

```
## # A tibble: 5 x 3
##   .model AICc  BIC
##   <chr> <dbl> <dbl>
## 1 m1     706.  720.
## 2 m2     704.  715.
## 3 m3     703.  711.
## 4 m4     684.  695.
## 5 m5    1023. 1063.
```

Based on the above table, **m4** that is automatic ARIMA has the best model which is using ARIMA(1,0,0)(2,1,0)[12] as shown below:

```
m %>% select(m4)%>%report()
```

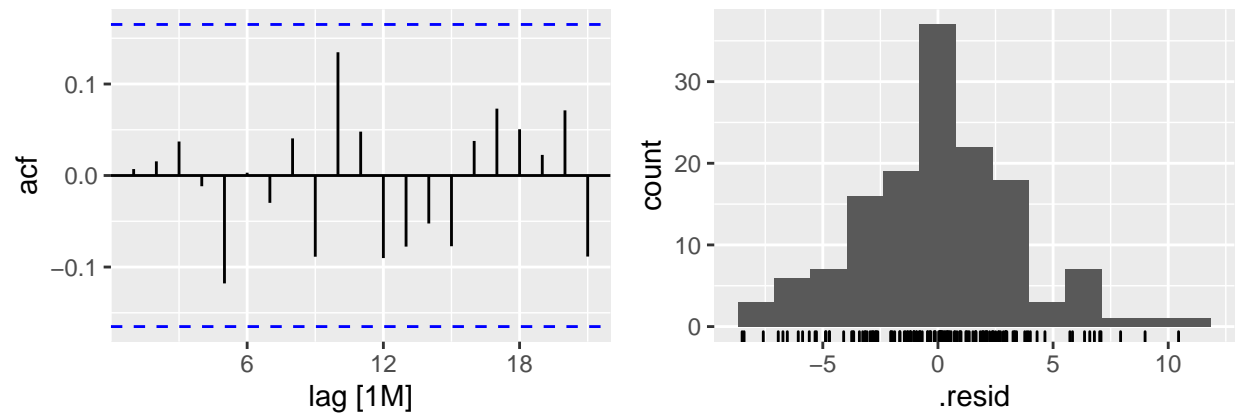
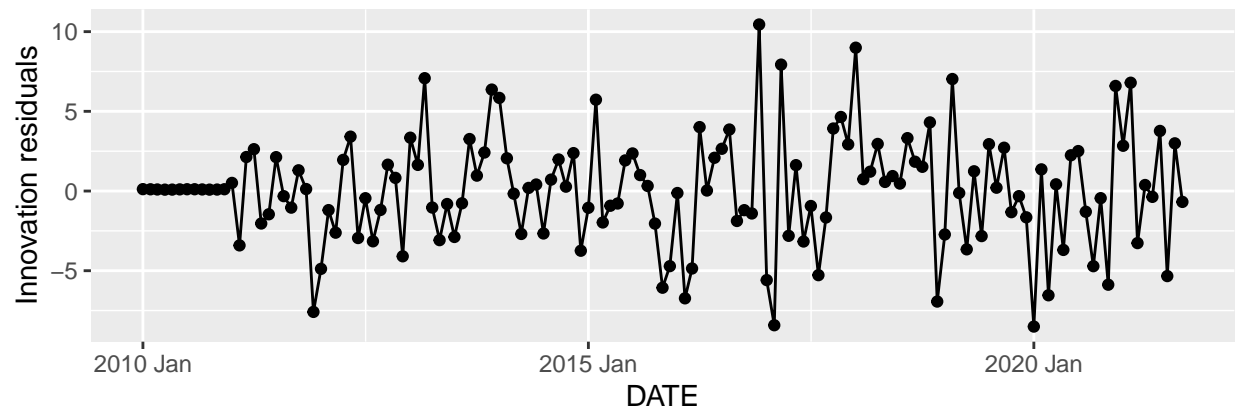
```
## Series: ELEC
## Model: ARIMA(1,0,0)(2,1,0)[12]
##
## Coefficients:
##          ar1      sar1      sar2
##      0.3757 -0.3375 -0.4386
## s.e.  0.0877  0.0834  0.0870
##
## sigma^2 estimated as 10.76: log likelihood=-337.76
## AIC=683.53  AICc=683.85  BIC=694.97
```

3. We examine the residuals of all the models using the Ljung-Box test and the `gg_tsresiduals()` function.

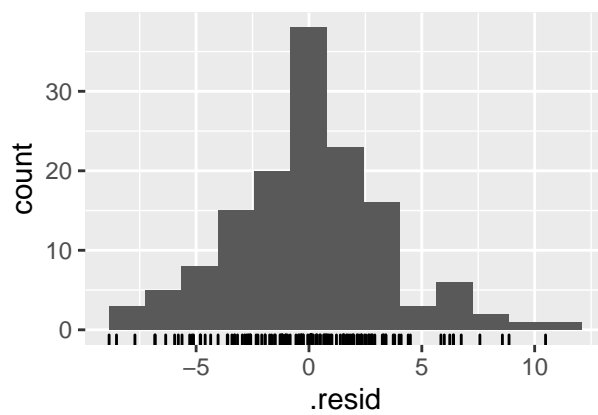
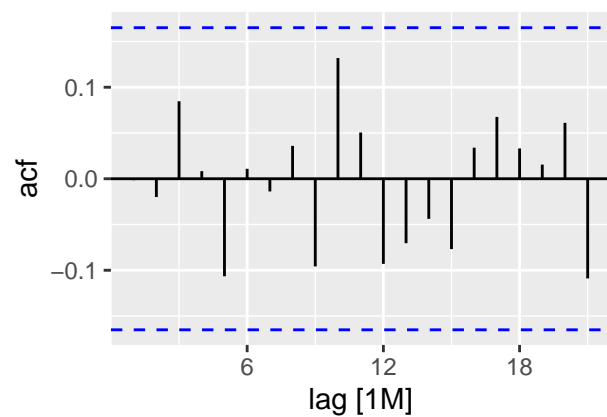
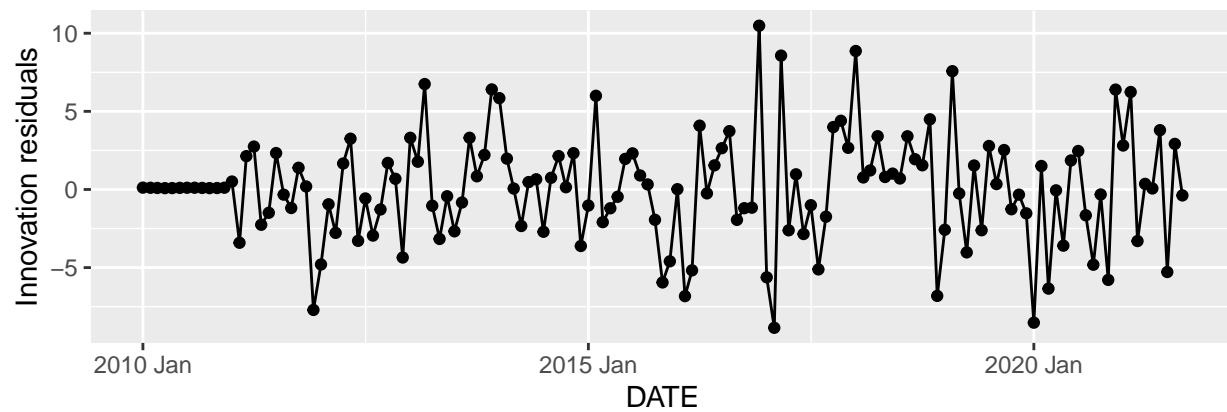
```
m %>% augment() %>%
  features(.resid, ljung_box, lag = 10)
```

```
## # A tibble: 5 x 3
##   .model lb_stat lb_pvalue
##   <chr>   <dbl>   <dbl>
## 1 m1      6.70    0.753
## 2 m2      7.12    0.714
## 3 m3      7.94    0.635
## 4 m4      9.00    0.532
## 5 m5     20.4    0.0256
```

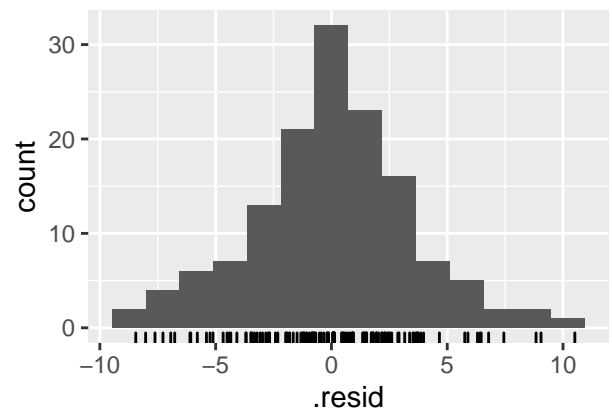
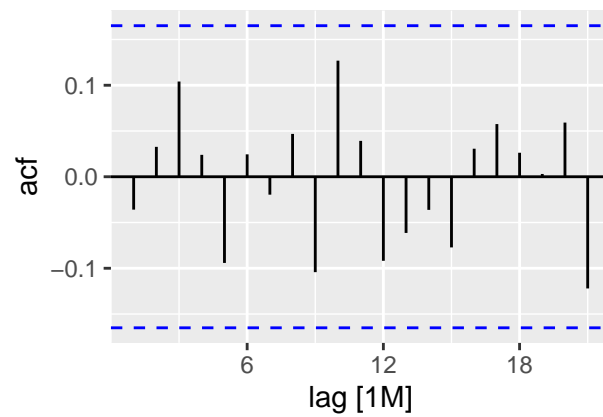
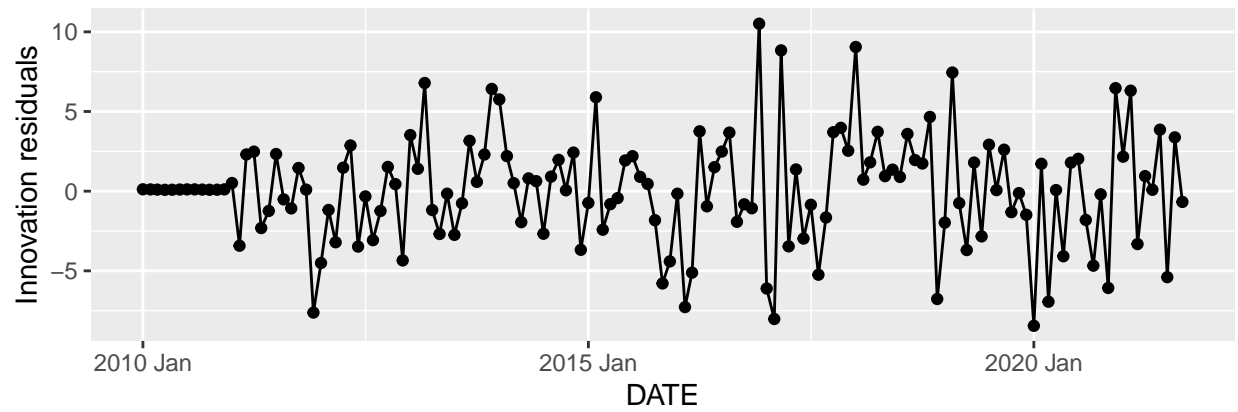
```
m %>% select(m1) %>% gg_tsresiduals()
```



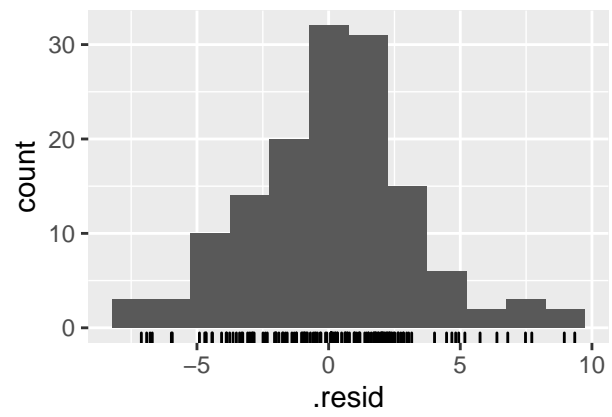
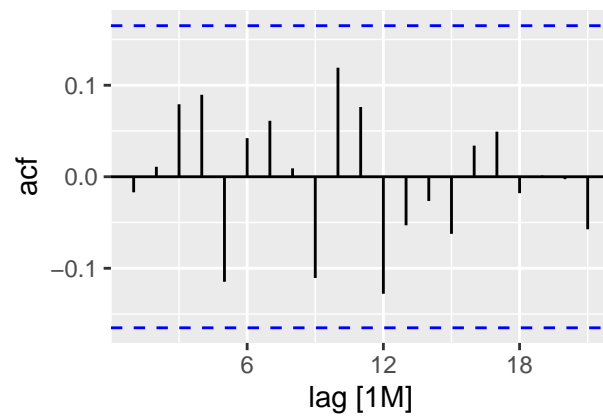
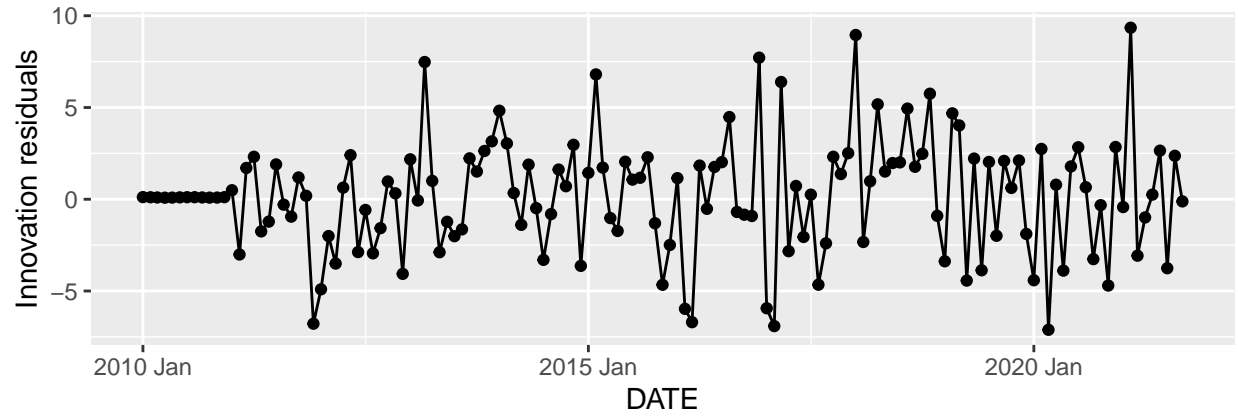
```
m %>% select(m2) %>% gg_tsresiduals()
```



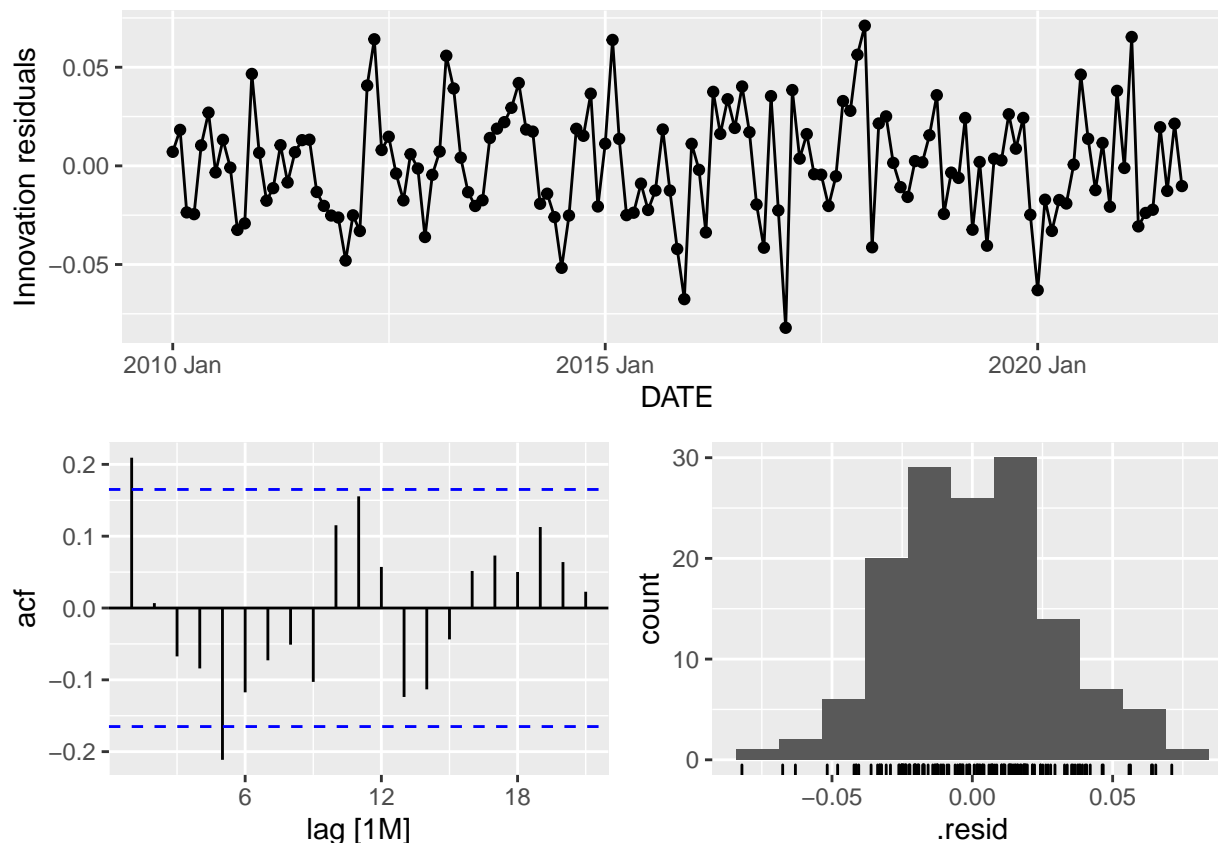
```
m %>% select(m3) %>% gg_tsresiduals()
```

```
m %>% select(m4) %>% gg_tsresiduals()
```



```
m %>% select(m5) %>% gg_tsresiduals()
```



As shown above, we have high p values for the ARIMA models indicating that the residuals are uncorrelated. Only the ETS model has low p value and the errors are correlated. This is also confirmed by the residual plots.

4. For the set of five models selected (automatically and/or manually), we examine the in-sample accuracy metrics. Based on a holistic analysis of the information criteria, we select the best two ARIMA models and the ETS model.

Solution:

```
m %>% glance()%>%select(.model,AICc,BIC)
```

```
## # A tibble: 5 x 3
##   .model AICc  BIC
##   <chr>  <dbl> <dbl>
## 1 m1      706.  720.
## 2 m2      704.  715.
## 3 m3      703.  711.
## 4 m4      684.  695.
## 5 m5     1023. 1063.
```

The two best ARIMA models are: 1. m4: ARIMA(1,0,0)(2,1,0)[12] AICc of 683.85 2. m3: ARIMA(1,0,0)(1,1,0)[12] AICc of 702.74 3. m2: ARIMA(2,0,0)(1,1,0)[12] AICc of 704.11 The ETS model is: 3. m5

m4:

```
m %>% select(m4)%>%report() # ETS
```

```
## Series: ELEC
## Model: ARIMA(1,0,0)(2,1,0)[12]
##
## Coefficients:
##          ar1      sar1      sar2
##      0.3757 -0.3375 -0.4386
## s.e.  0.0877  0.0834  0.0870
##
## sigma^2 estimated as 10.76:  log likelihood=-337.76
## AIC=683.53  AICc=683.85  BIC=694.97
```

m3:

```
m %>% select(m3)%>%report() # ETS
```

```
## Series: ELEC
## Model: ARIMA(1,0,0)(1,1,0)[12]
##
## Coefficients:
##          ar1      sar1
##      0.5054 -0.2361
## s.e.  0.0754  0.0878
##
## sigma^2 estimated as 13.06:  log likelihood=-348.28
## AIC=702.56  AICc=702.75  BIC=711.13
```

m2:

```
m %>% select(m2)%>%report() # ETS
```

```
## Series: ELEC
## Model: ARIMA(2,0,0)(1,1,0)[12]
##
## Coefficients:
##          ar1      ar2      sar1
##      0.4665  0.0767 -0.2400
## s.e.  0.0873  0.0877  0.0879
##
## sigma^2 estimated as 13.09:  log likelihood=-347.9
## AIC=703.79  AICc=704.12  BIC=715.23
```

ETS:

```
m %>% select(m5)%>%report() # ETS
```

```
## Series: ELEC
## Model: ETS(M,N,A)
## Smoothing parameters:
```

```
##      alpha = 0.2924178
##      gamma = 0.0001000089
##
##      Initial states:
##      l[0]      s[0]      s[-1]      s[-2]      s[-3]      s[-4]      s[-5]      s[-6]
## 101.0318 7.817185 -6.612555 -11.00125 -2.759789 8.873747 10.47588 -0.3751617
##      s[-7]      s[-8]      s[-9]      s[-10]      s[-11]
## -11.78396 -13.88839 -2.460821 6.012828 15.70228
##
##      sigma^2: 9e-04
##
##      AIC      AICc      BIC
## 1019.152 1022.992 1063.383
```

For model cross-validation purposes stretch the DR data as follows:

```
D.CV <- DR %>%
  filter(DATE >= yearmonth("2010 Jan")) %>%
  stretch_tsibble(.init = 36, .step = 1)
```

- Next, we fit cross-validation models for each of the time sub-series in the stretched data for each of the four model types selected in (4). In the case(s) where the models were automatically selected, we do not run the automatic selection under cross validation, instead we manually enter the model order/type when you call the ARIMA()/ETS() function.

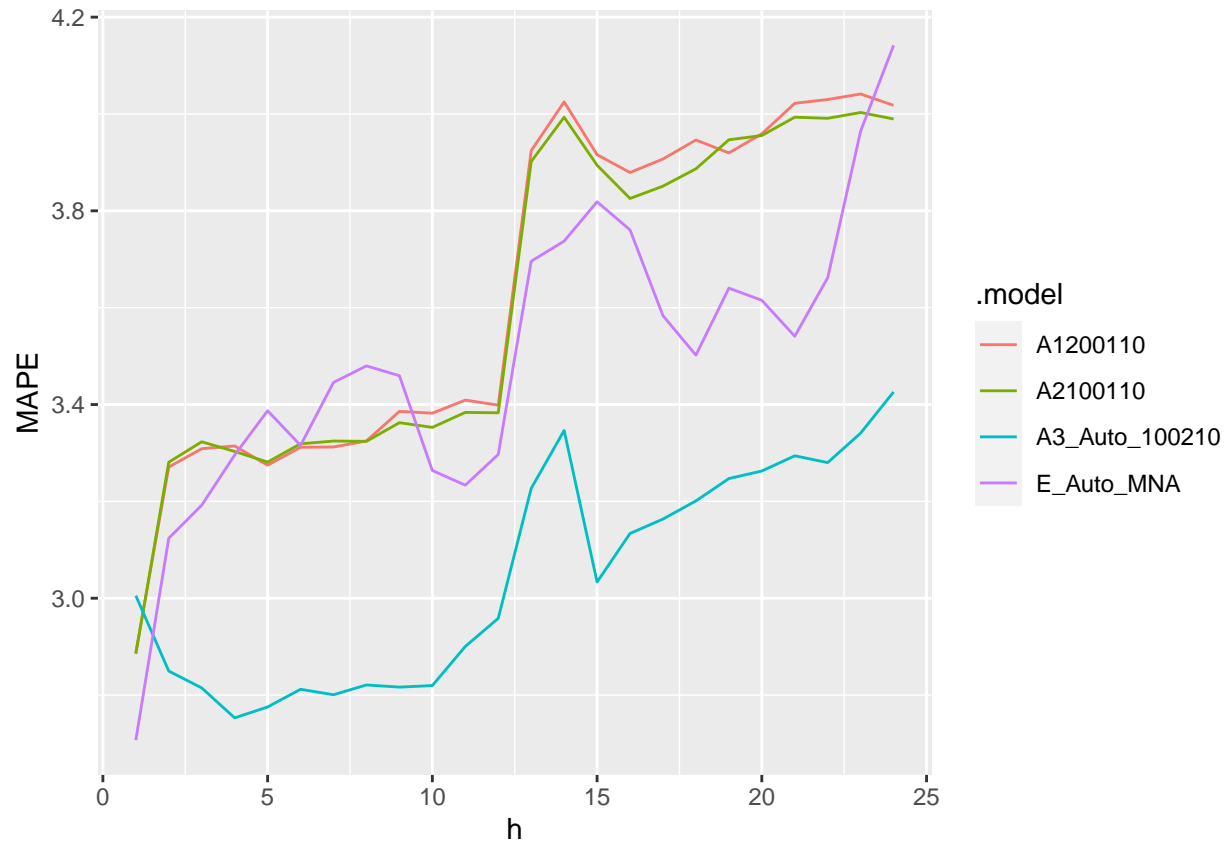
```
mC <- D.CV %>%model( A1200110 = ARIMA(ELEC ~ pdq(2,0,0) + PDQ(1,1,0)),
  A2100110 = ARIMA(ELEC ~ pdq(1,0,0) + PDQ(1,1,0)),
  A3_Auto_100210= ARIMA(ELEC ~ pdq(1,0,0)+ PDQ(2,1,0)),
  E_Auto_MNA = ETS(ELEC ~ error("M") + trend("N") + season("A")))
```

- Next, we prepare a 24-month ahead forecast for each of the models fitted in (5) and prepare a plot of MAPE vs months-ahead.

```
fc <- mC%>%forecast(h = 24)

fc2 <- fc%>%group_by(.id,.model) %>%
  mutate(h = row_number()) %>%
  ungroup()

fc2%>%accuracy(DR, by = c("h", ".model")) %>%
  ggplot(aes(x = h, y = MAPE,color = .model)) +
  geom_line()
```



We use cross-validation to validate the performance of our model. However, based on the above results, we see that ETS model performs poorly, hence, we discard it.

- Below, we examine the cross-validation residuals of the models we selected in (6), and based on their correlation (model vs. model), we discuss if it is advisable to prepare an ensemble forecast averaging the forecasts of two or more models.

In order to make ensemble forecast, we look at the correlation of residuals. If the residuals are highly correlated, we do not prepare any ensemble forecasts, but if the residuals are not correlated, we prepare ensemble forecasts.

```
x <- m %>% select(m4) %>% residuals()
y <- m %>% select(m3) %>% residuals()
z <- m %>% select(m2) %>% residuals()
cor(x$.resid,y$.resid)
```

```
## [1] 0.9125074
```

```
cor(x$.resid,z$.resid)
```

```
## [1] 0.9091801
```

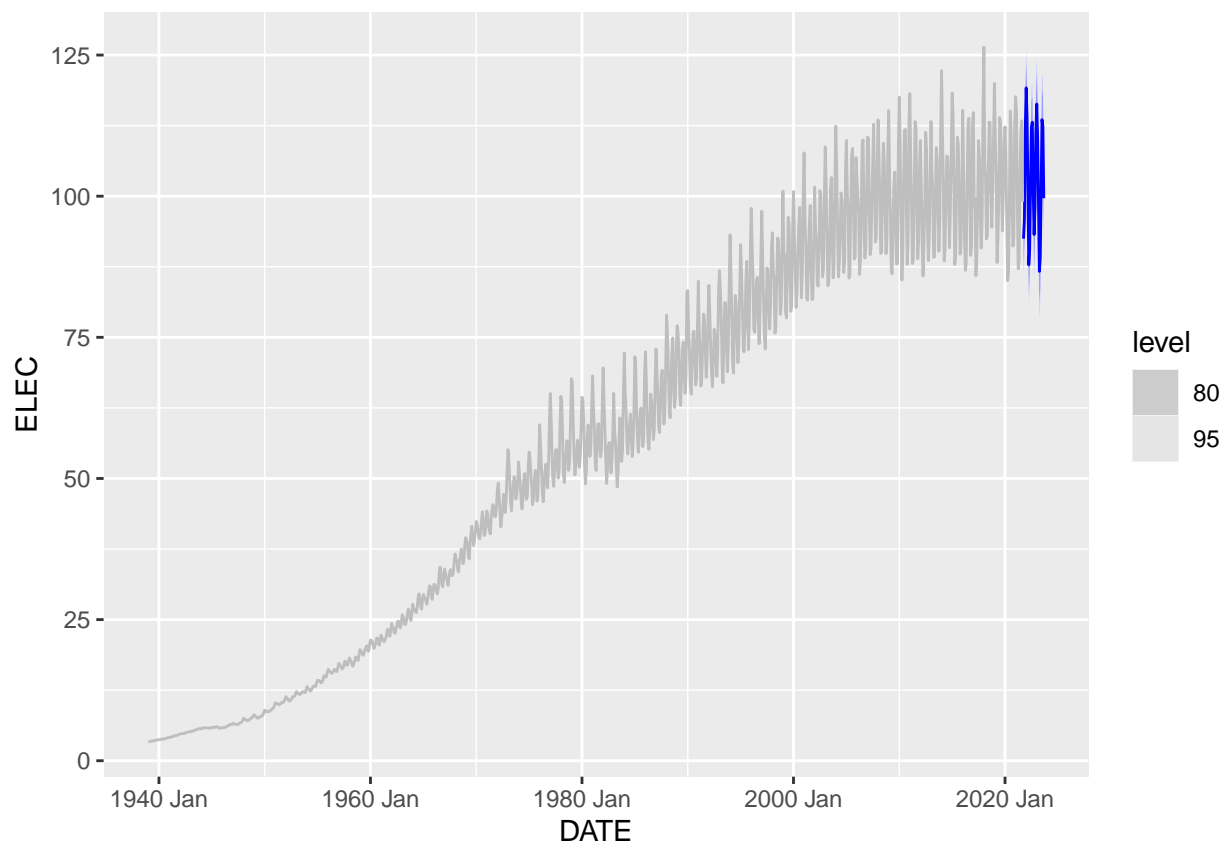
```
cor(y$.resid,z$.resid)
```

```
## [1] 0.9970743
```

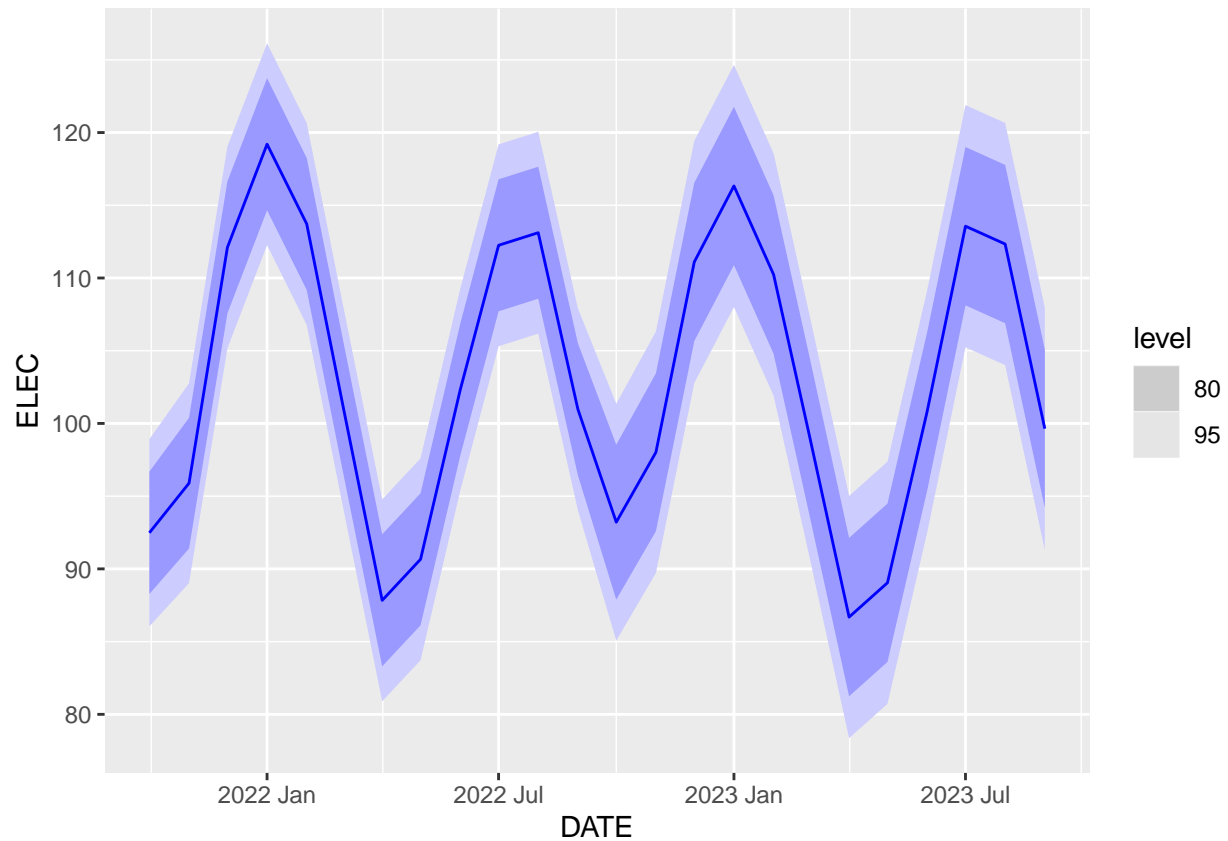
As shown above, the residuals are quite high. Hence we would be building ensemble forecasts.

8. The index is very useful for energy planning purpose as most of the variability and seasonality is produced by combined cycle natural gas plants and single cycle plants that also run on natural gas (i.e., nuclear and coal generation is fixed and relatively constant). For this purpose it is of interest to know what is the production index level that will not be separated with a probability (service-level) of 95%. For the best model in (6), we plot the 24-month ahead forecast and plot the forecast and the corresponding confidence interval to help us address the service level question. Below, we report numerically the month-by-month the index forecasts that meet the desired 95% service level.

```
m_best <- DR%>%model(arima_best = ARIMA(ELEC ~ pdq(1,0,0) + PDQ(2,1,0)))  
fc_best <- m_best%>%forecast(h=24)  
fc_best%>%autoplot()+geom_line(data = D, mapping = aes(y = ELEC), col = "grey")
```



```
fc_best%>%autoplot()
```



```
fc_best %>% hilo(level = c(90)) %>%
  unpack_hilo("90%") %>%
  select(ELEC, "90%_upper")
```

```
## # A tsibble: 24 x 3 [1M]
##       ELEC '90%_upper'    DATE
##       <dbl>      <dbl>    <mth>
## 1 N(92, 11)      97.9 2021 Oct
## 2 N(96, 12)     102. 2021 Nov
## 3 N(112, 12)    118. 2021 Dec
## 4 N(119, 13)    125. 2022 Jan
## 5 N(114, 13)    120. 2022 Feb
## 6 N(101, 13)    107. 2022 Mar
## 7 N(88, 13)     93.7 2022 Apr
## 8 N(91, 13)     96.5 2022 May
## 9 N(102, 13)    108. 2022 Jun
## 10 N(112, 13)   118. 2022 Jul
## # ... with 14 more rows
```