Homework 4. Time series (100 points)

The submitted files must include pdf-s with your answers along with all R scripts. For example:

- Student A submitted:
  - Homework4.pdf - final report containing all answers
  - Homework4.Rmd - R-markdown files with student solutions

No pdf report - no grade. If you experience difficulties with knitting, combine your answers in Word and any other editor and produce pdf-file for grading.

No R scripts - 50 % reduction in grade if relative code present in pdf- report, 100% reduction if no such code present.

Reports longer than 40 pages are not going to be graded.

Question1

1. The plastics data set (see plastics.csv) consists of the monthly sales (in thousands) of product A for a plastics manufacturer for five years. (Total 32 points)

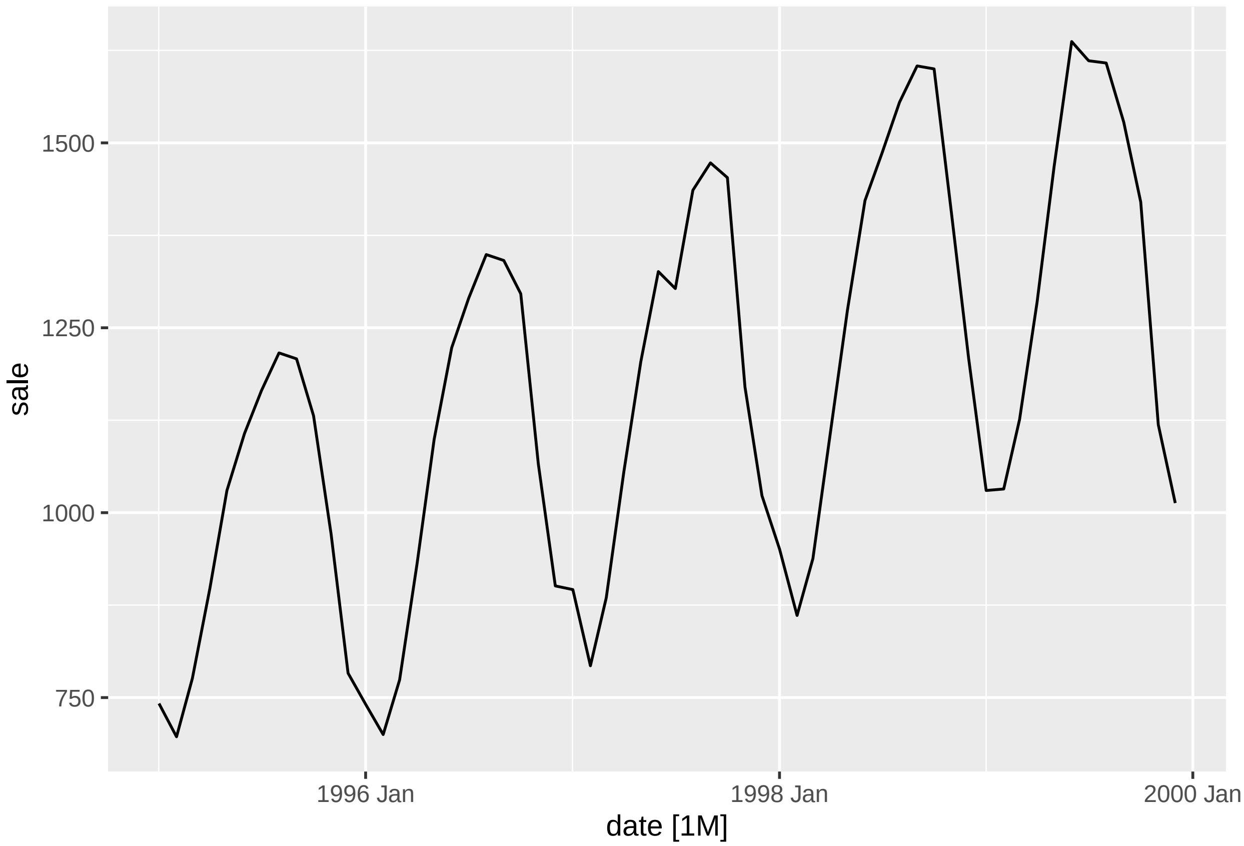
1.1 Read csv file and convert to tsible with proper index (2 points)

```r
plastics <- readr::read_csv("plastics.csv") %>%
  mutate(date = yearmonth(date)) %>%
  as_tsibble(
    index = date
  )
```

```
## Rows: 60 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): date
## dbl (1): sale
##
## 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.
```

1.2 Plot the time series of sales of product A. Can you identify seasonal fluctuations and/or a trend-cycle? (2 points)

```r
plastics %>% autoplot(sale)
```



The time-series plot exhibits a discernible upward trend, indicating that there is a positive increment in the data points over time. Alongside the rising trend, there is a prominent seasonal pattern with a periodicity of one year.

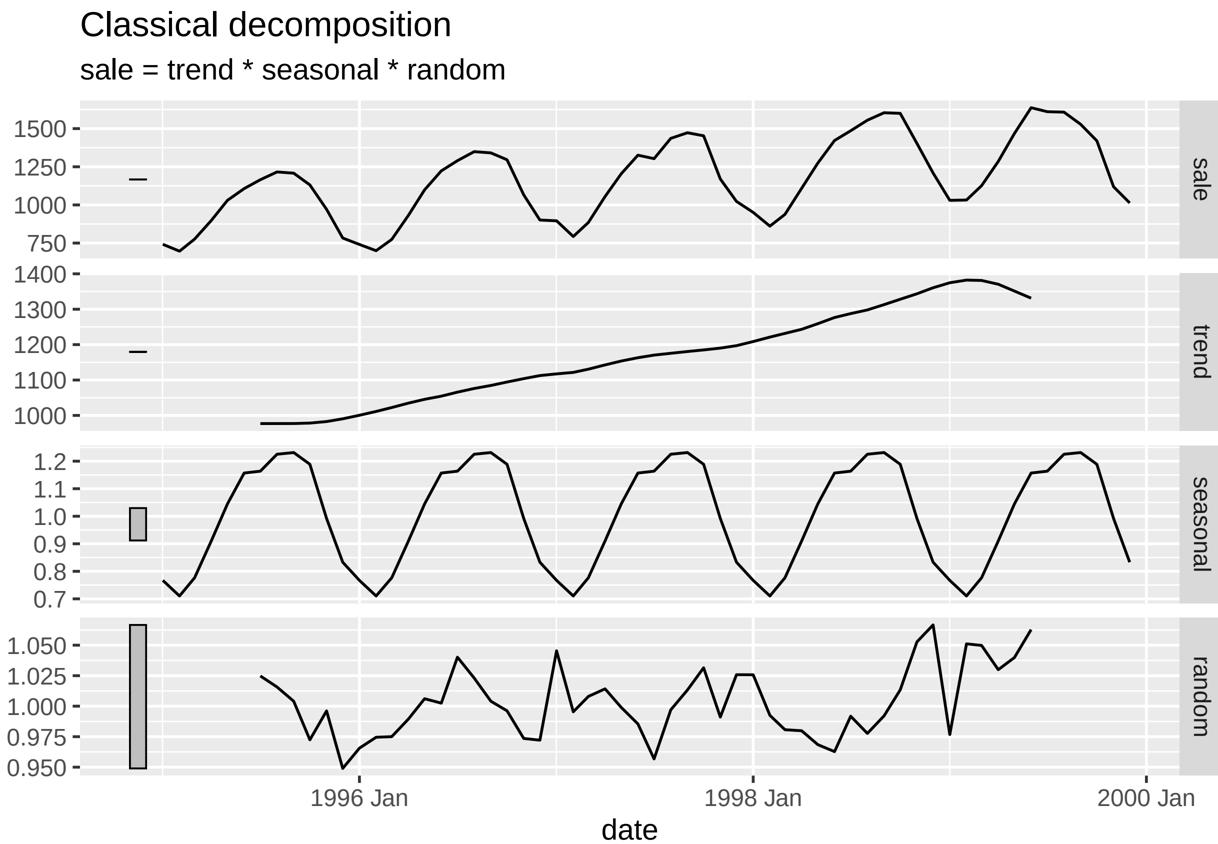
1.3) Use a classical multiplicative decomposition to calculate the trend-cycle and seasonal components. Plot these components. (4 points)

```
dcmp = plastics %>% model(classical_decomposition(type='m'))
```

```
## Model not specified, defaulting to automatic modelling of the `sale` variable.
## Override this using the model formula.
```

```
components(dcmp) %>% autoplot()
```

```
## Warning: Removed 6 rows containing missing values (`geom_line()`).
```

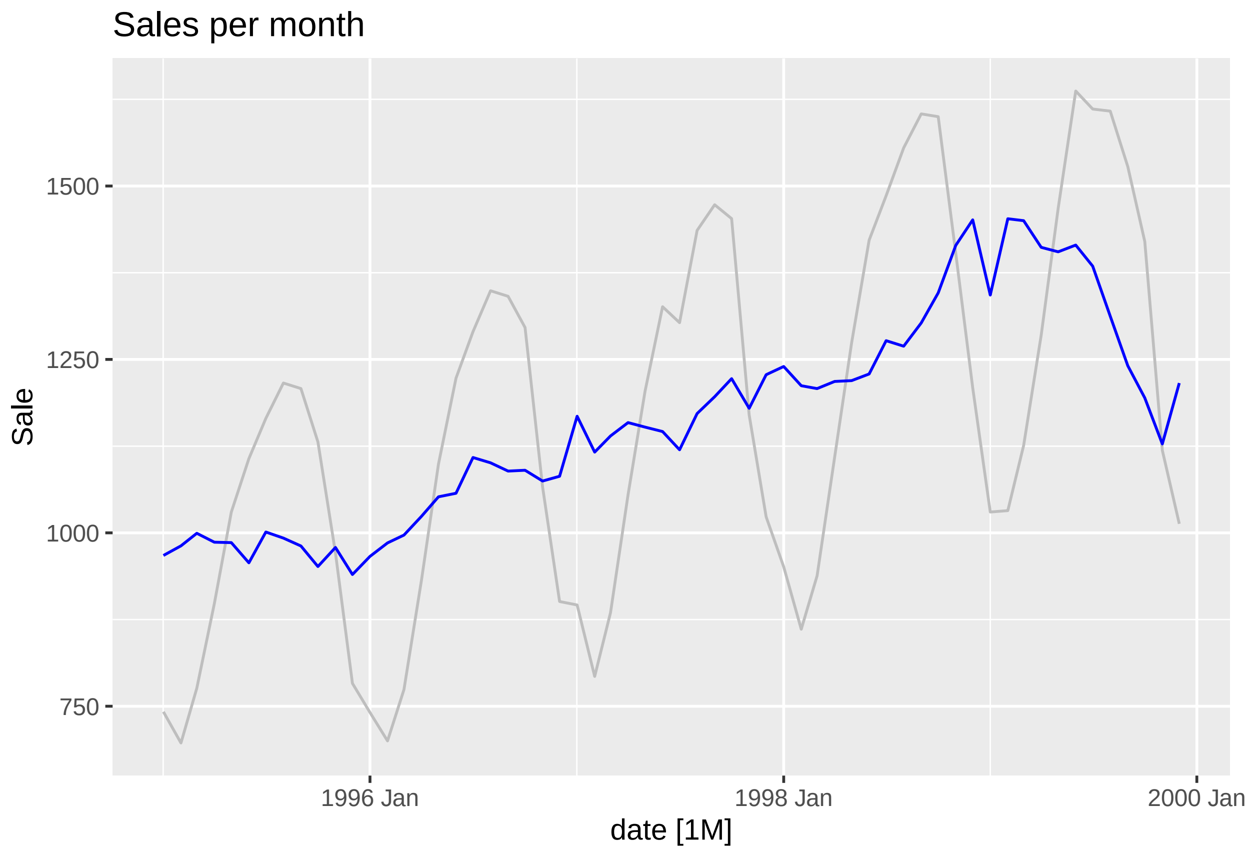


1.4 Do the results support the graphical interpretation from part a? (2 points)

Yes the result support the graphical interpretation we made earlier.

1.5 Compute and plot the seasonally adjusted data. (2 points)

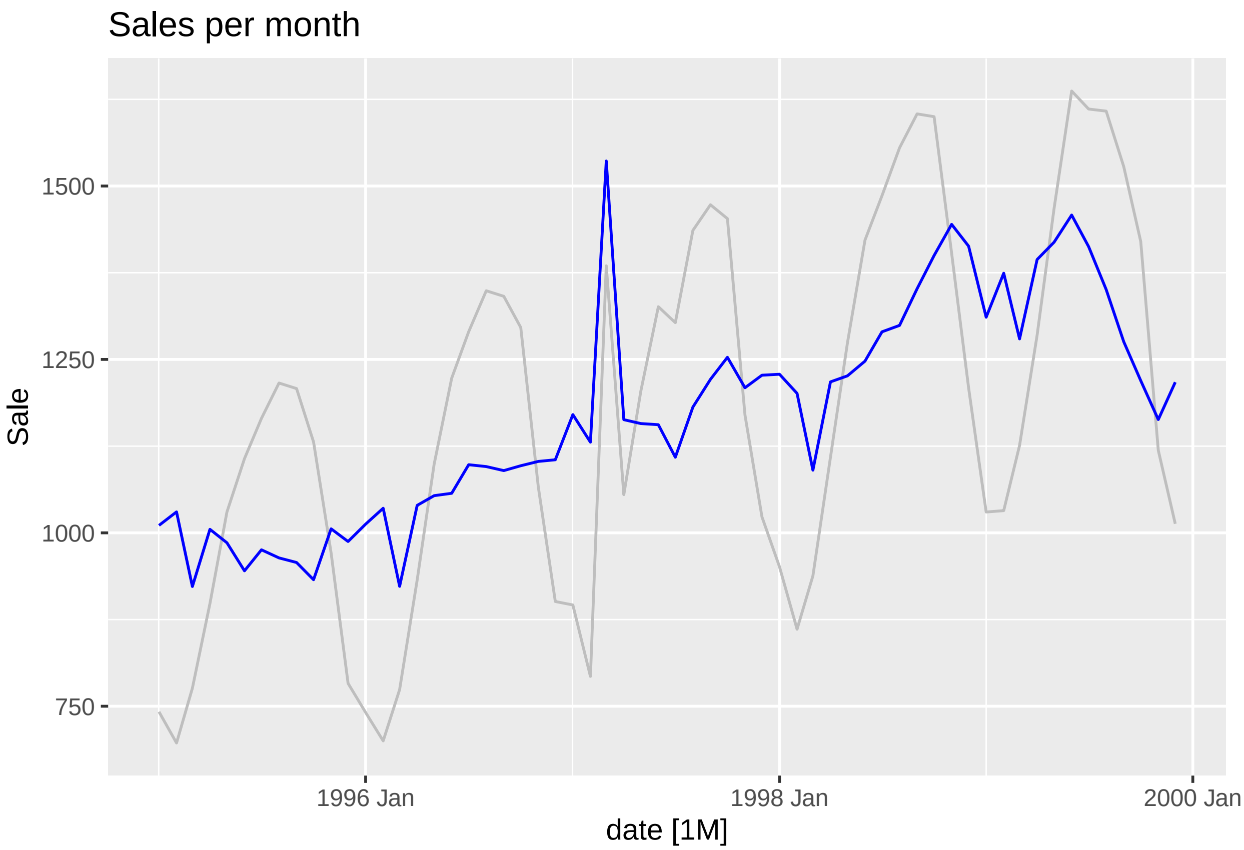
```
plastics %>%
  autoplot(sale, color = "gray") +
  autolayer(components(dcmp), season_adjust, color = "blue") +
  labs(
    y = "Sale",
    title = "Sales per month"
  )
```



1.6 Change one observation to be an outlier (e.g., add 500 to one observation), and recompute the seasonally adjusted data. What is the effect of the outlier? (2 points)

tip: use autoplot to plot original and add outlier plot with autolayer

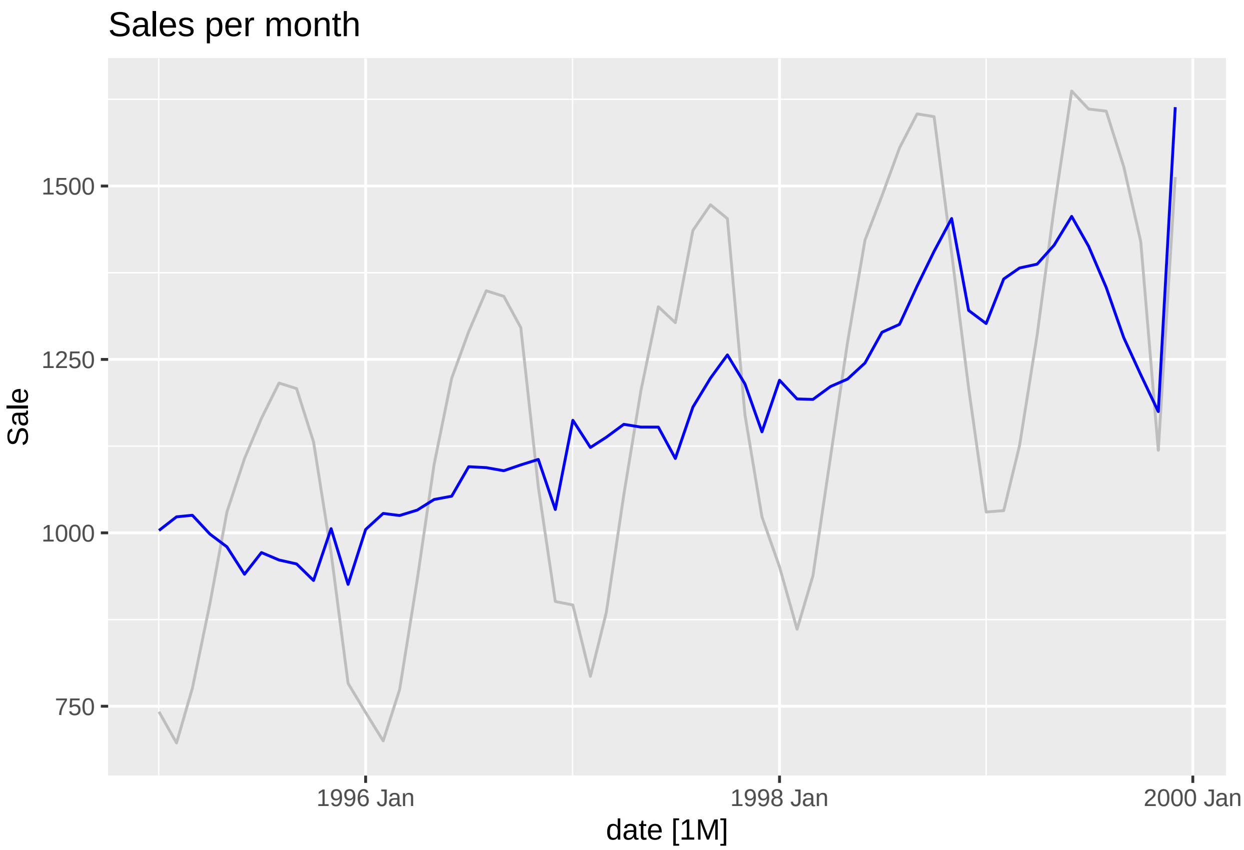
```
plastics_outlier <- plastics %>%
  mutate(sale = if_else(date == yearmonth("1997-03"), sale + 500, sale))
dcmp <- plastics_outlier %>% model(stl = STL(sale))
plastics_outlier %>%
  autoplot(sale, color = "gray") +
  autolayer(components(dcmp), season_adjust, color = "blue") +
  labs(
    y = "Sale",
    title = "Sales per month"
  )
```



We can see that the outlier skewed the estimation of the seasonal factor for the period it occurs in, leading to incorrect seasonal adjustments not only for that point but potentially for the entire series.

1.7 Does it make any difference if the outlier is near the end rather than in the middle of the time series? (2 points)

```
plastics_outlier <- plastics %>%
  mutate(sale = if_else(date == yearmonth("1999-12"), sale + 500, sale))
dcmp <- plastics_outlier %>% model(stl = STL(sale))
plastics_outlier %>%
  autoplot(sale, color = "gray") +
  autolayer(components(dcmp), season_adjust, color = "blue") +
  labs(
    y = "Sale",
    title = "Sales per month"
  )
```



An outlier in the middle of a time series may have less impact on seasonal adjustment due to the presence of more data points to stabilize the seasonal pattern. In contrast, an outlier near the end can disproportionately affect the seasonal adjustment, as there are fewer data points to mitigate its effect, potentially leading to skewed seasonal factors and less reliable adjustments.

1.8 Let's do some accuracy estimation. Split the data into training and testing. Let all points up to the end of 1998 (including) are training set. (2 points)

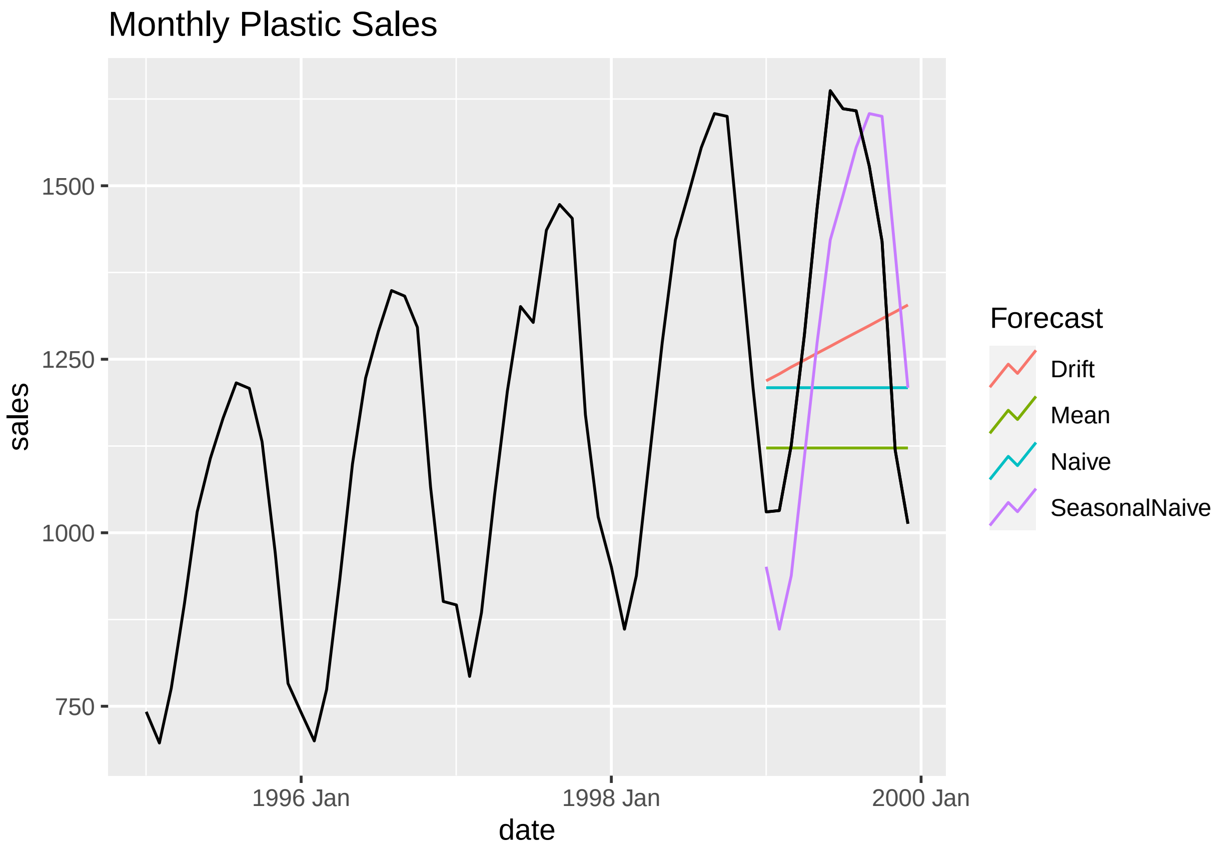
```r
training_set <- plastics %>%
  filter(date <= yearmonth("1998-12"))

testing_set <- plastics %>%
  filter(date > yearmonth("1998-12"))
```

1.9 Using training set create a fit for mean, naive, seasonal naive and drift methods. Forecast next year (in training set). Plot forecasts and actual data. Which model performs the best. (4 points)

```r
# Specify, estimate and forecast
training_set %>%
  model(
    Mean = MEAN(sale),
    Naive = NAIVE(sale),
    SeasonalNaive = SNAIVE(sale),
    Drift = RW(sale ~ drift())
  ) %>%
  forecast(h = 12) %>%
  autoplot(plastics, level = NULL) +
  labs(
    title = "Monthly Plastic Sales",
```

```r
    y = "sales"
  ) +
  guides(colour = guide_legend(title = "Forecast"))+
  geom_line(data = ungroup(testing_set), aes(x = date, y = sale), colour = "black")
```



We can clearly see Seasonly naive performed better

1.10 Repeat 1.9 for appropriate EST. Report the model. Check residuals. Plot forecasts and actual data. (4 points)

```r
fit <- training_set %>%
  model(additive = ETS(sale ~ error("A") + trend("A") + season("A")))

fit %>%
  select(additive) %>%
  report()
```

```
## Series: sale
## Model: ETS(A,A,A)
##   Smoothing parameters:
##     alpha = 0.874131
##     beta  = 0.002090183
##     gamma = 0.00257488
##
##   Initial states:
##       l[0]      b[0]       s[0]      s[-1]     s[-2]     s[-3]     s[-4]     s[-5]
##   954.9065 6.943258 -183.3809 -5.131477 213.1458 259.1886 264.8406 191.2004
##     s[-6]     s[-7]      s[-8]   s[-9]     s[-10]     s[-11]
##   162.2698 44.62585 -103.1951 -257.1 -328.7508 -257.7128
```

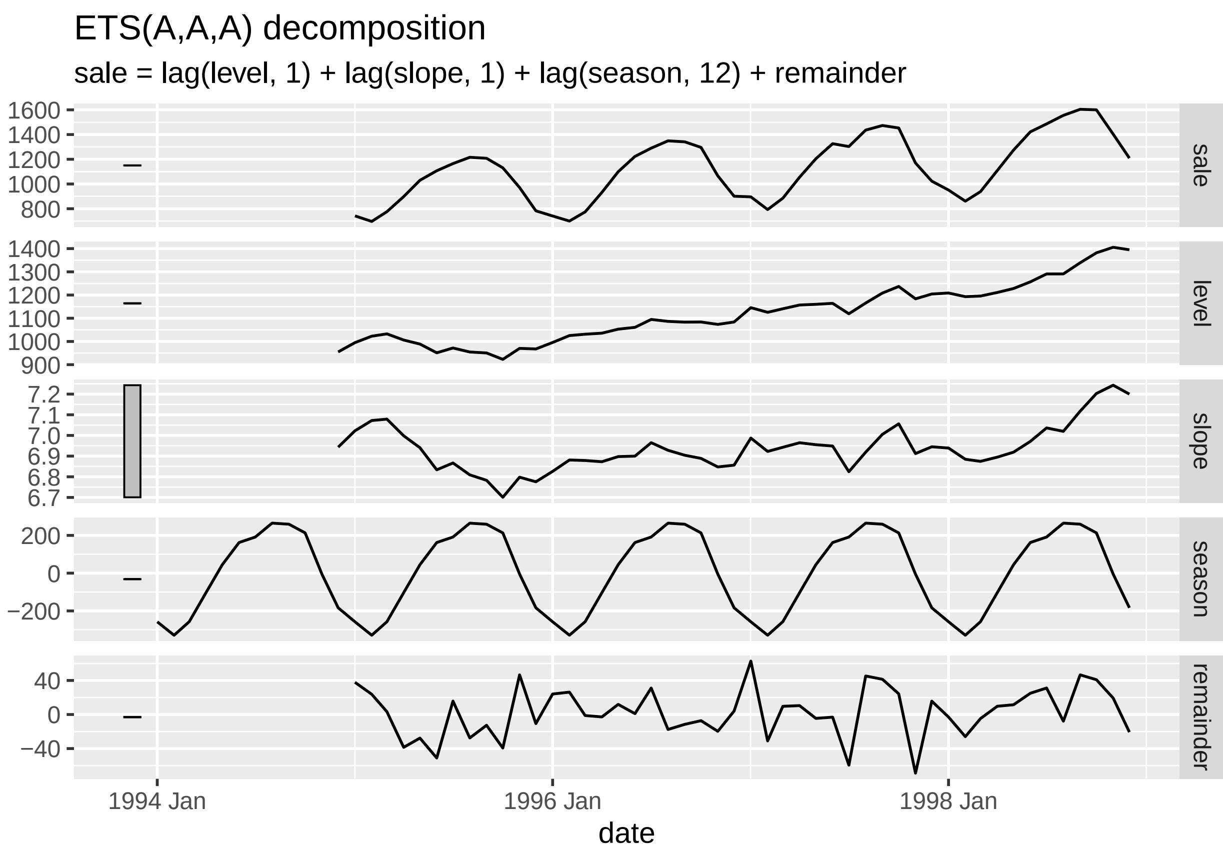
```
##
##   sigma^2:  1263.433
##
##      AIC     AICc      BIC
## 543.1515 563.5515 574.9619
```

```
accuracy(fit %>% forecast(h = 12), testing_set)
```

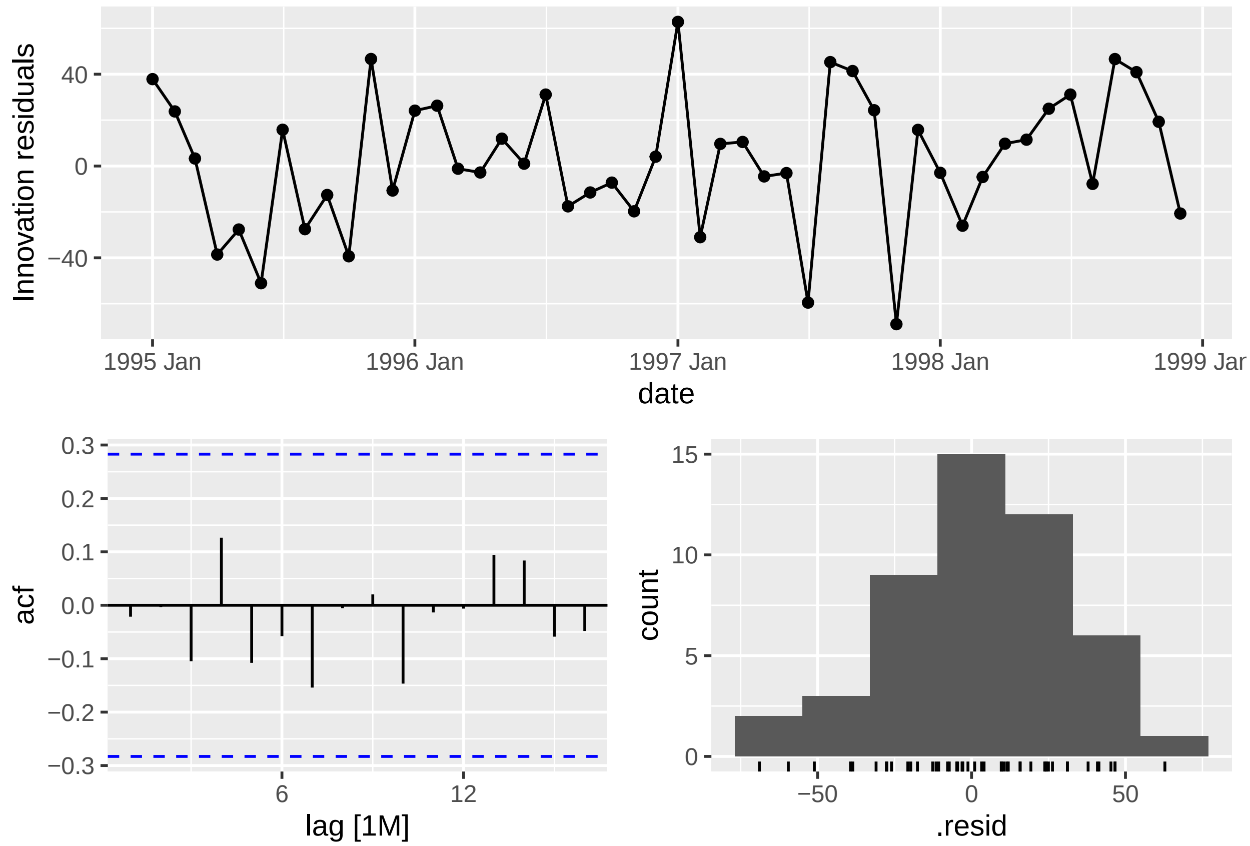
```
## # A tibble: 1 x 10
##   .model   .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>    <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 additive Test  -119.  168.  125. -9.87  10.2   NaN   NaN 0.796
```

```
components(fit) %>% autoplot()
```

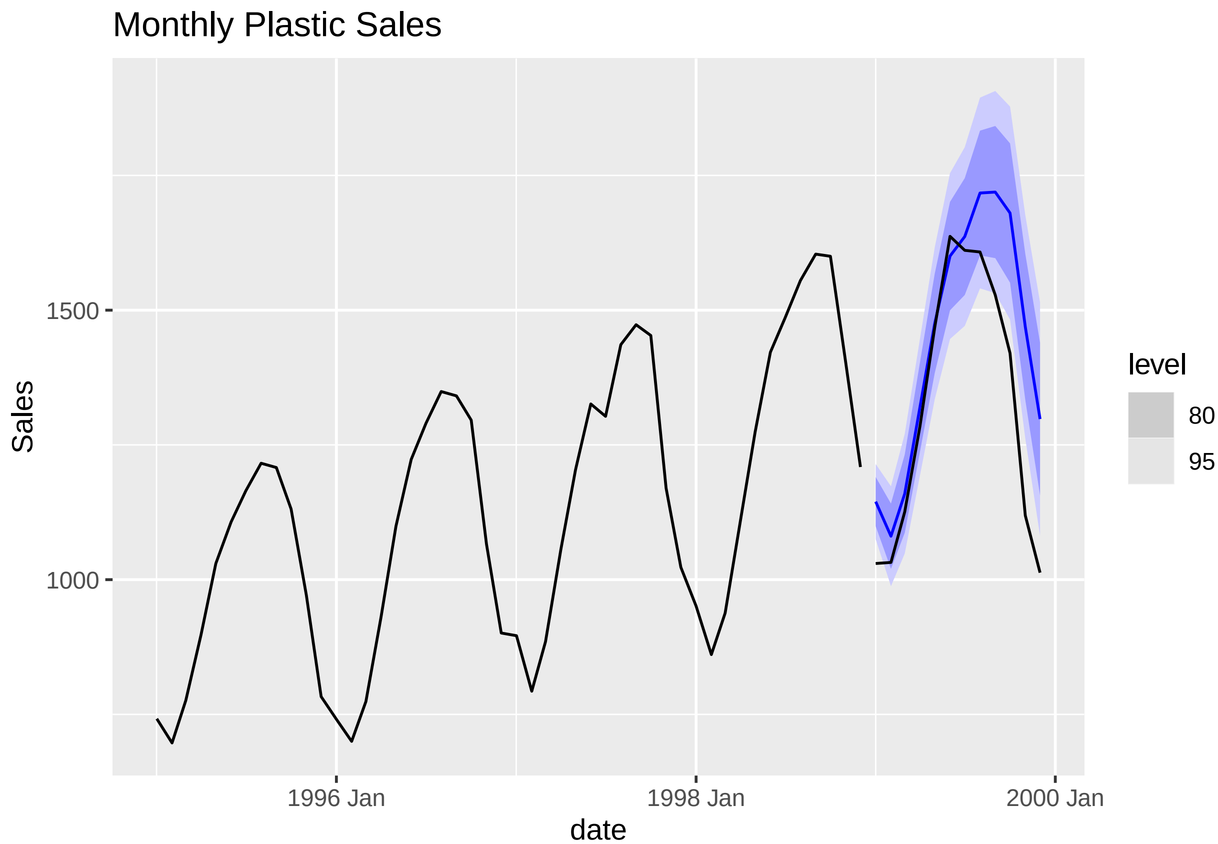
```
## Warning: Removed 12 rows containing missing values (`geom_line()`).
```




```
# Extract the residuals
fit %>%
gg_tsresiduals()
```




```r
fc <- fit %>% forecast(h=12)
fc %>% autoplot(training_set) + labs(y = "Sales", title = "Monthly Plastic Sales")+
  geom_line(data = ungroup(testing_set), aes(x = date, y = sale), colour = "black")
```



1.11 Repeat 1.9 for appropriate ARIMA. Report the model. Check residuals. Plot forecasts and actual data. (4 points)

```
training_set %>% features(sale, unitroot_nsdiffs)
```

```
## # A tibble: 1 x 1
##   nsdiffs
##     <int>
## 1       1
```

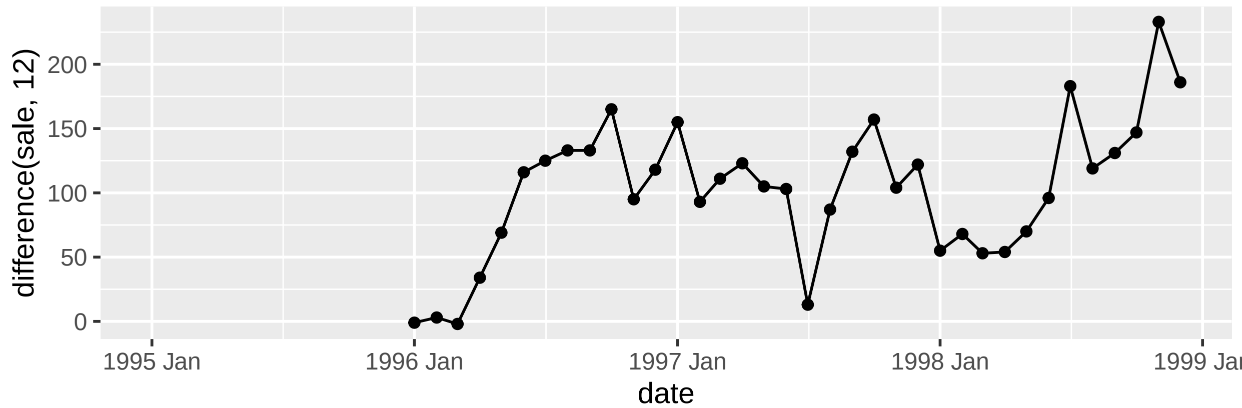
```
training_set %>% mutate(d_sale = difference(sale, 12)) %>%features(d_sale, unitroot_ndiffs)
```

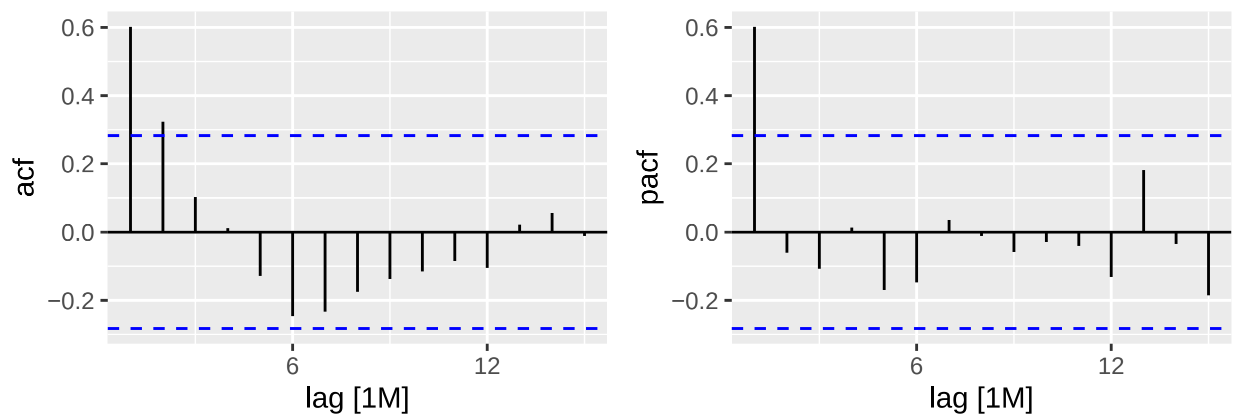
```
## # A tibble: 1 x 1
##   ndiffs
##    <int>
## 1      0
```

```
training_set %>% gg_tsdisplay(difference(sale,12), plot_type='partial')
```

```
## Warning: Removed 12 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 12 rows containing missing values (`geom_point()`).
```

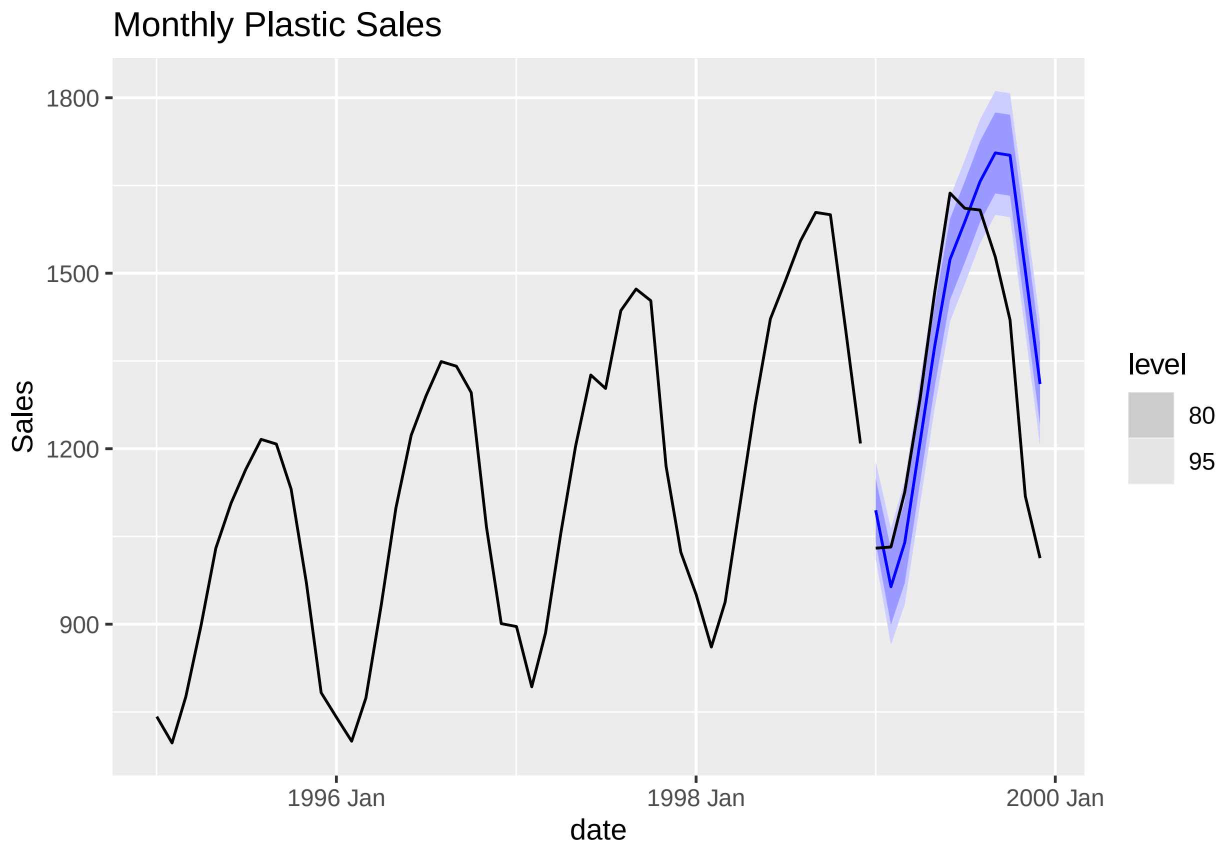





```r
fit = training_set %>% model(arima = ARIMA(sale ~ pdq(0,0,2)+PDQ(0,1,0)))
report(fit)
```

```
## Series: sale
## Model: ARIMA(0,0,2)(0,1,0)[12] w/ drift
##
## Coefficients:
##          ma1     ma2  constant
##       0.6606  0.4435  101.6507
## s.e.  0.1425  0.1583   13.9116
##
## sigma^2 estimated as 1786:  log likelihood=-184.63
## AIC=377.26   AICc=378.55   BIC=383.6
```

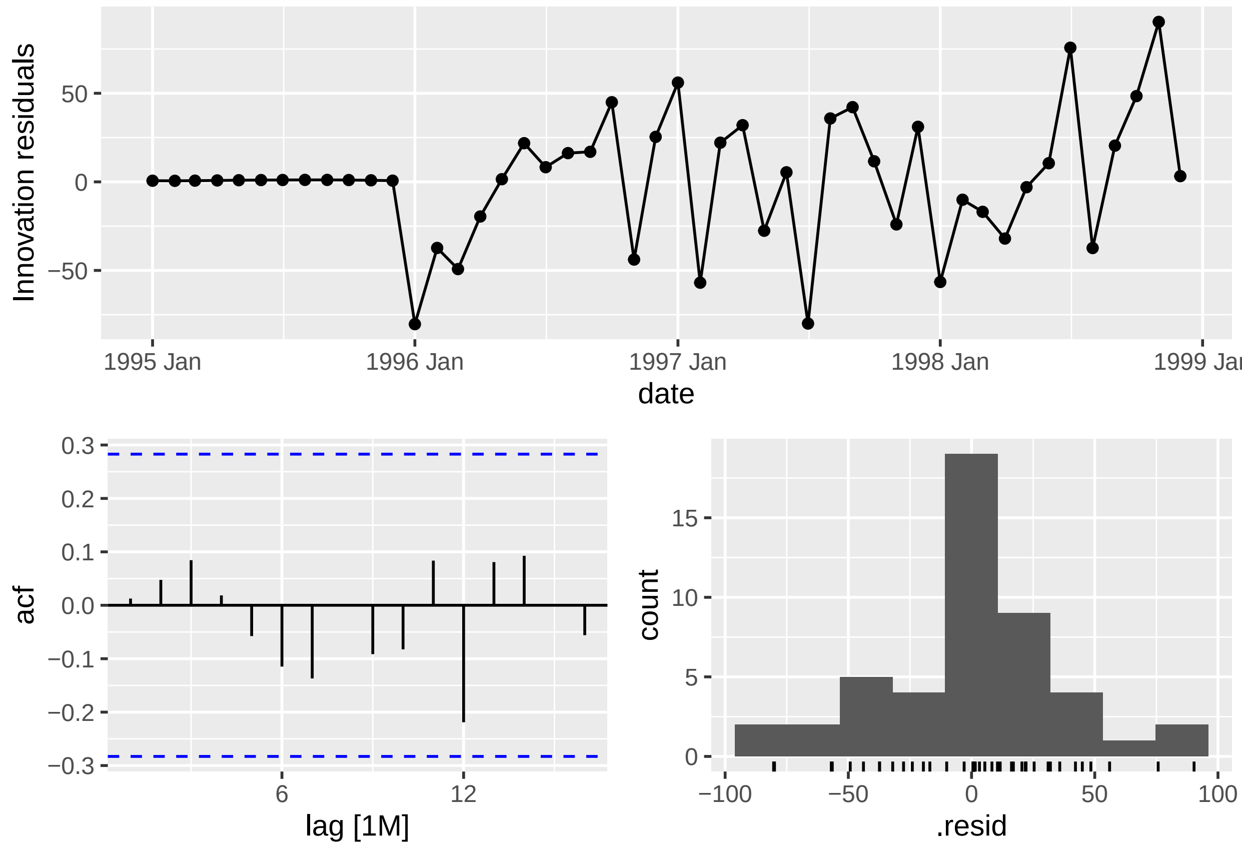
```r
fc <- fit %>% forecast(h=12)
fc %>% autoplot(training_set) + labs(y = "Sales", title = "Monthly Plastic Sales")+
  geom_line(data = ungroup(testing_set), aes(x = date, y = sale), colour = "black")
```




```r
augment(fit %>% dplyr::select(arima)) %>%
  features(.resid, ljung_box, lag=24, dof=4)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 arima     15.2     0.766
```

```r
fit %>%
  gg_tsresiduals()
```




```
accuracy(fit %>% forecast(h = 12), testing_set)
```

```
## # A tibble: 1 x 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arima  Test  -66.5  181.  143. -5.83  11.6   NaN   NaN 0.807
```

1.12 Which model has best performance? (2 points)

- When it comes to AIC ARIMA model has lowest AIC value hence it better model when compared with ETS(A,A,A)
- So ARIMA model is better model.

Question 2

2 For this exercise use data set visitors (visitors.csv), the monthly Australian short-term overseas visitors data (thousands of people per month), May 1985–April 2005. (Total 32 points)

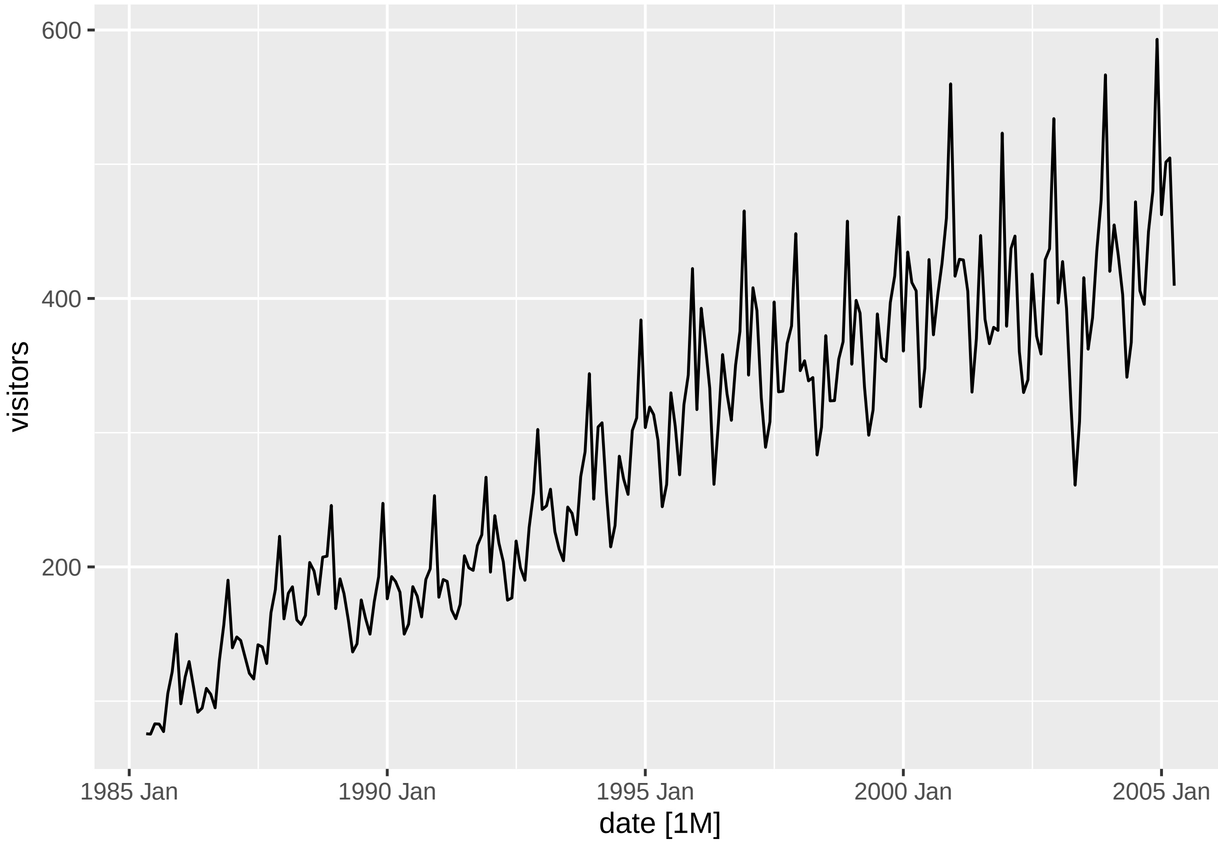
2.1 Make a time plot of your data and describe the main features of the series. (6 points)

```
visitors <- readr::read_csv("visitors.csv") %>%
  mutate(date = yearmonth(date)) %>%
  as_tsibble(
    index = date
  )
```

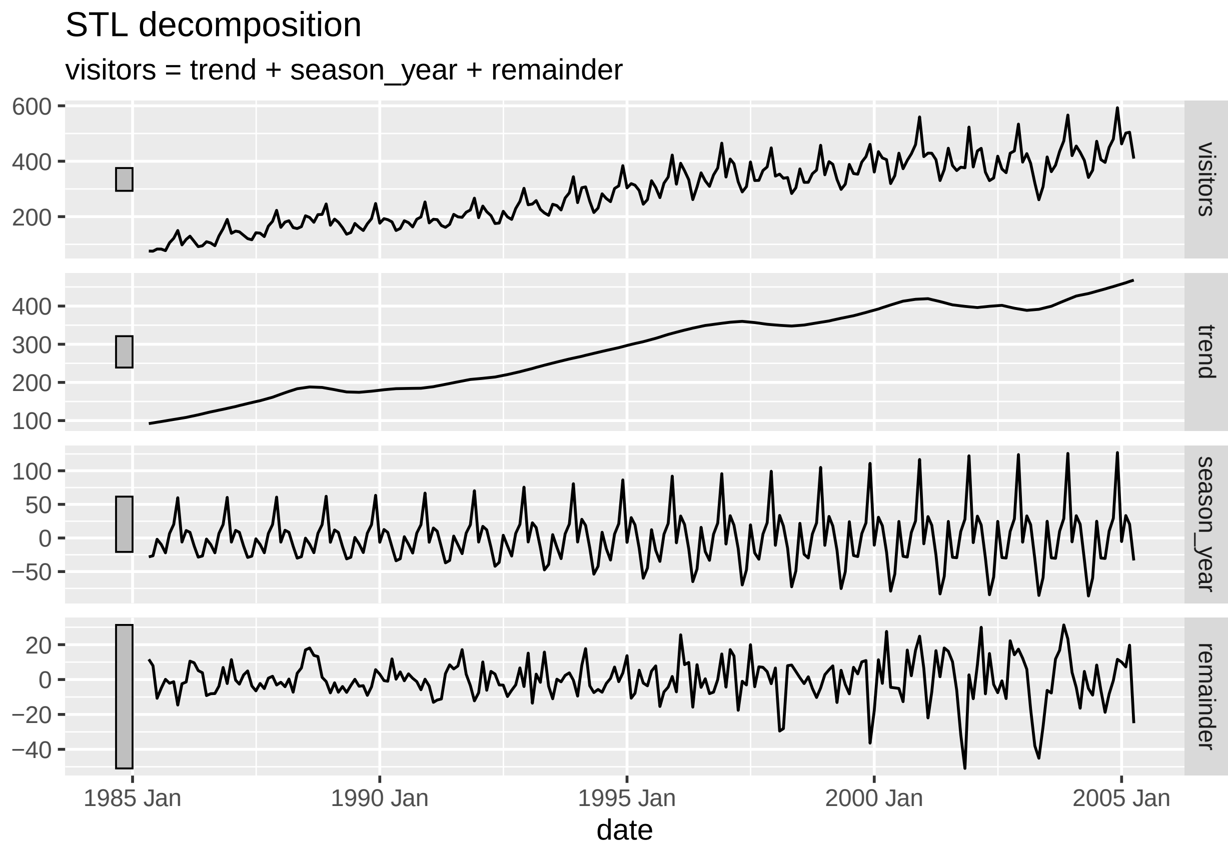
```
## Rows: 240 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## chr (1): date
## dbl (1): visitors
##
## 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.
```

```r
visitors %>% autoplot(visitors)
```




```r
dcmp <- visitors %>% model(stl = STL(visitors))
components(dcmp) %>% autoplot()
```



- The time series plot exhibits a discernible positive trend, indicating a consistent increase in values over time.
- An annual seasonal pattern is evident, characterized by regular fluctuations that repeat every year.
- There is increasing variability in the seasonal component, with the amplitude of seasonal fluctuations becoming more pronounced as time progresses.

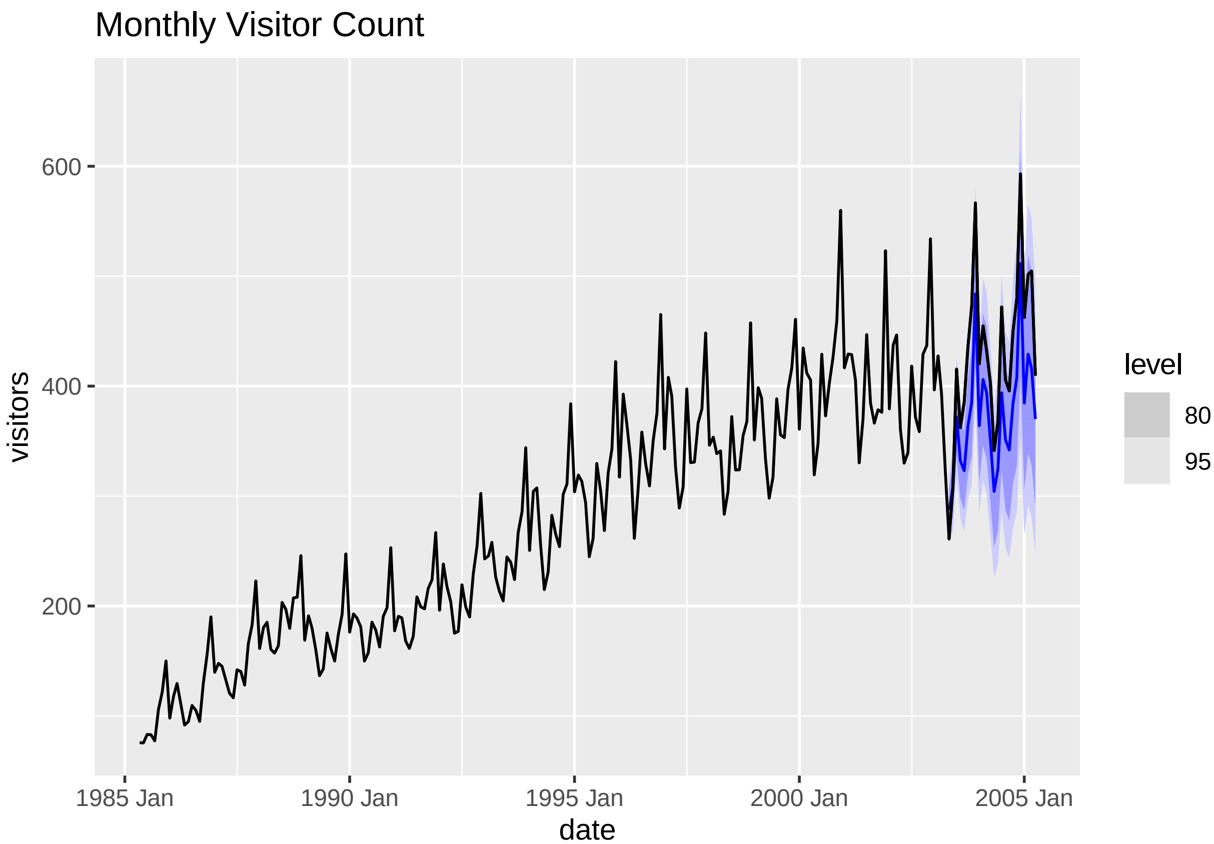
2.2 Split your data into a training set and a test set comprising the last two years of available data. Forecast the test set using Holt-Winters' multiplicative method. (6 points)

```
train <- visitors %>% filter(date < yearmonth("2003-05"))
test <- visitors %>% filter(date >= yearmonth("2003-05"))
```

```
fit <- train %>%
  model(multiplicative = ETS(visitors ~ error("M") + trend("A") + season("M")))


forecast <- fit %>%
  forecast(h = nrow(test))

forecast %>% autoplot(visitors) + labs(y = "visitors", title = "Monthly Visitor Count")+
  geom_line(data = ungroup(test), aes(x = date, y = visitors), colour = "black")
```



2.3. Why is multiplicative seasonality necessary here? (6 points)

As the seasonal fluctuations are proportional to the level of the time series, multiplicative seasonality is necessary here.

2.4. Forecast the two-year test set using each of the following methods: (8 points)

I. an ETS model;
II. an additive ETS model applied to a Box-Cox transformed series;
III. a seasonal naïve method;

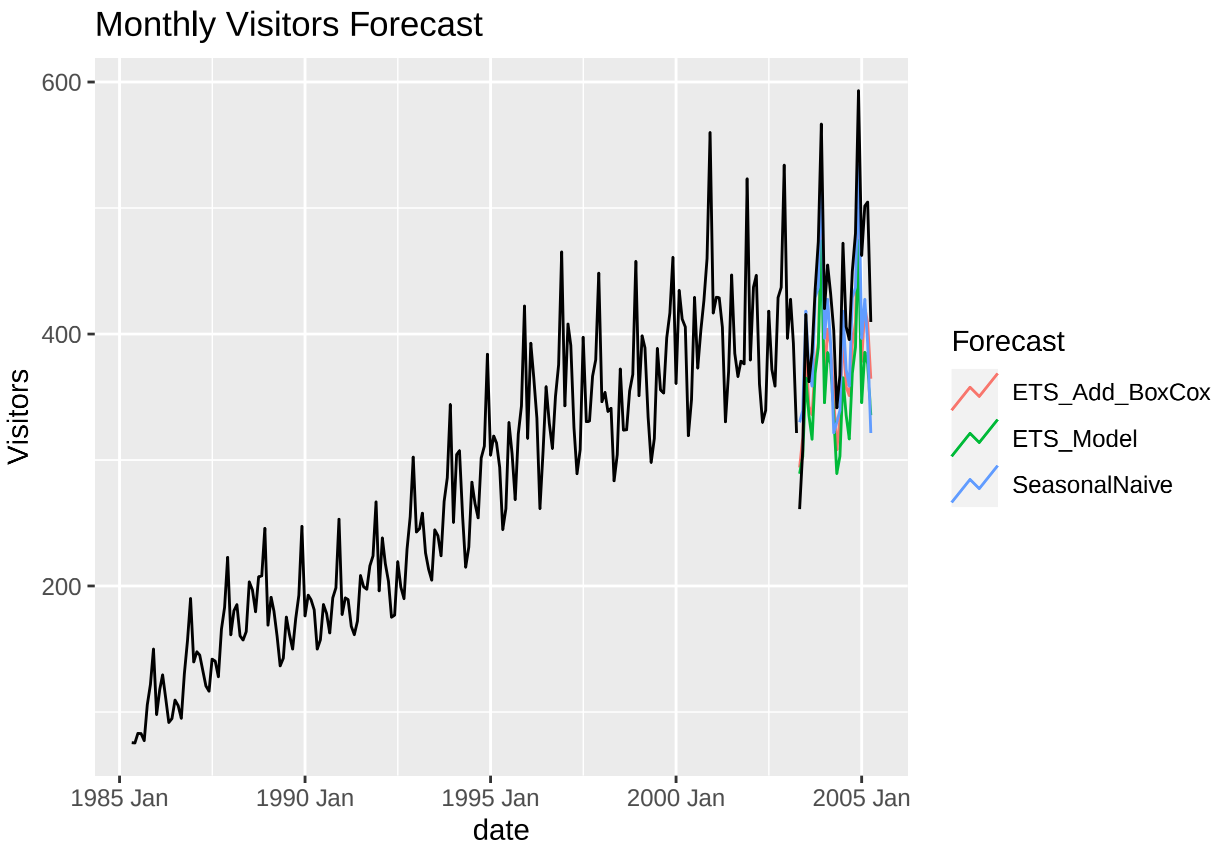
```
train %>%
  features(visitors, features = guerrero)
```

```
## # A tibble: 1 x 1
##   lambda_guerrero
##             <dbl>
## 1           0.362
```

```
fit = train %>%
  model(
    ETS_Model = ETS(visitors),
    ETS_Add_BoxCox = ETS(box_cox(visitors, 0.3624893) ~ error("A") + trend("A") + season("A")),
    SeasonalNaive = SNAIVE(visitors)
  )

fit %>% forecast(h = 24) %>%
  autoplot(train, level = NULL) +
  labs(
    title = "Monthly Visitors Forecast",
```

```r
    y = "Visitors"
  ) +
  guides(colour = guide_legend(title = "Forecast")) +
  geom_line(data = ungroup(test), aes(x = date, y = visitors), colour = "black")
```



2.5. Which method gives the best forecasts? Does it pass the residual tests? (6 points)

```r
report(fit)
```

```
## Warning in report.mdl_df(fit): Model reporting is only supported for individual
## models, so a glance will be shown. To see the report for a specific model, use
## `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 3 x 9
##   .model            sigma2 log_lik   AIC  AICc   BIC     MSE    AMSE     MAE
##   <chr>              <dbl>   <dbl> <dbl> <dbl> <dbl>   <dbl>   <dbl>   <dbl>
## 1 ETS_Model        0.00291  -1132. 2300. 2304. 2361. 211.    283.     0.0402
## 2 ETS_Add_BoxCox   0.161     -375.  784.  787.  841.   0.149   0.200  0.299
## 3 SeasonalNaive  675.          NA     NA    NA    NA  NA      NA     NA
```

```r
accuracy(fit %>% forecast(h = 24), test)
```

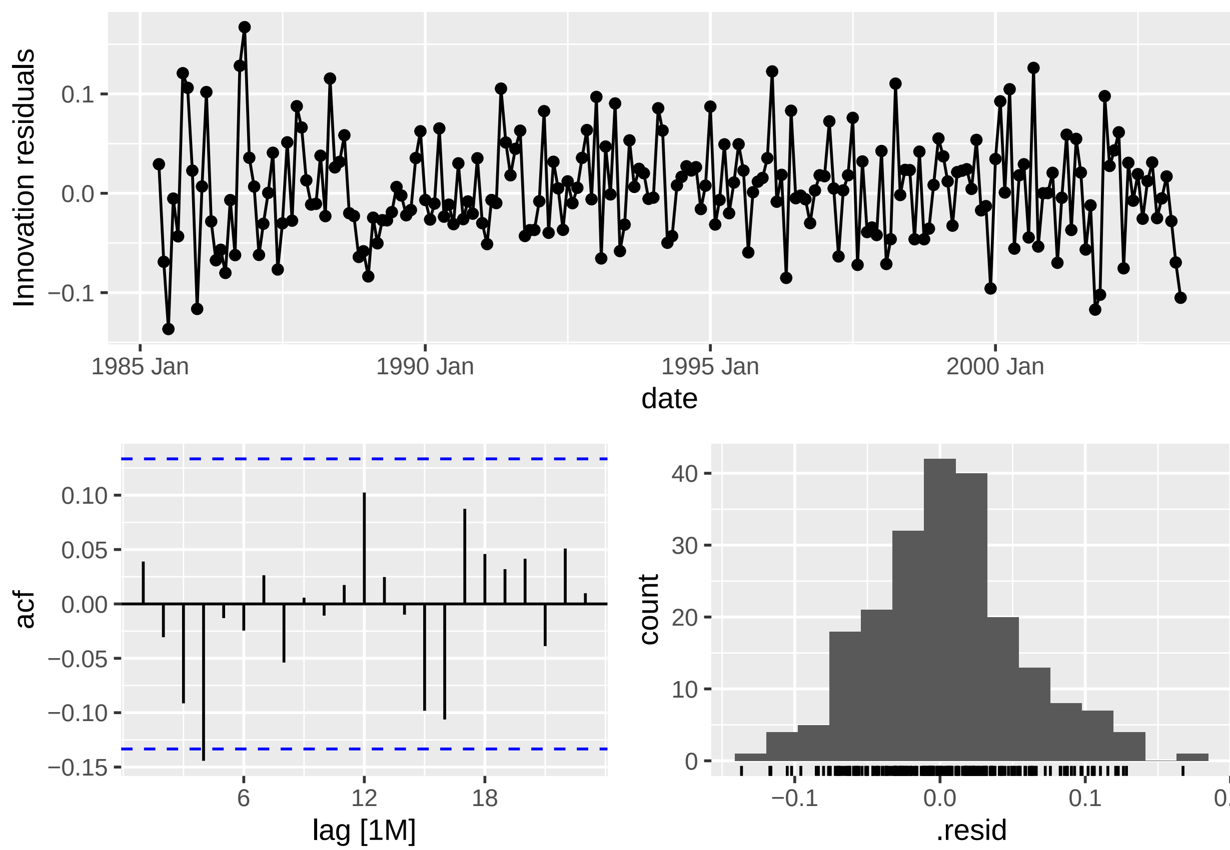
```
## # A tibble: 3 x 10
##   .model         .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>          <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ETS_Add_BoxCox Test   51.5  59.5  55.1 11.1  12.4    NaN   NaN 0.591
## 2 ETS_Model      Test   72.2  80.2  74.6 15.9  16.8    NaN   NaN 0.587
## 3 SeasonalNaive  Test   32.9  50.3  42.2  6.64  9.96   NaN   NaN 0.573
```

RMSE of Seasonal naive is lower so we can say it is the best model out of three.

```r
report(fit%>%dplyr::select(ETS_Model))
```

```
## Series: visitors
## Model: ETS(M,Ad,M)
##   Smoothing parameters:
##     alpha = 0.6396059
##     beta  = 0.0008087781
##     gamma = 0.000113317
##     phi   = 0.9798906
##
##   Initial states:
##      l[0]     b[0]      s[0]    s[-1]    s[-2]     s[-3]    s[-4]    s[-5]
##  87.76017 3.063938 0.9391967 1.053846 1.079178 0.9674827 1.327517 1.091004
##     s[-6]     s[-7]     s[-8]    s[-9]    s[-10]    s[-11]
##  1.032901 0.8867273 0.9399195 1.023092 0.8488214 0.8103151
##
##   sigma^2:  0.0029
##
##      AIC     AICc      BIC
## 2300.157 2303.629 2360.912
```

```r
fit %>%
  dplyr::select(ETS_Model) %>%
  gg_tsresiduals()
```



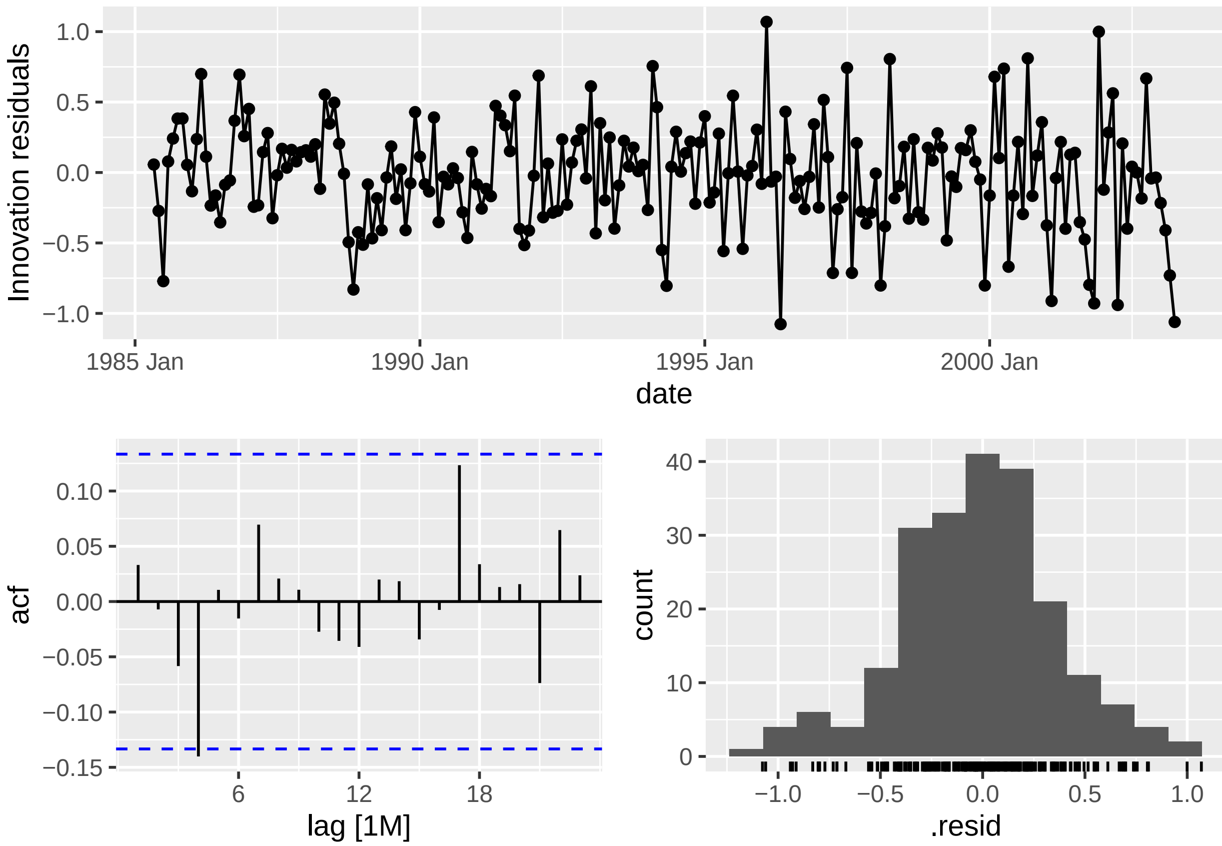
```r
augment(fit %>% dplyr::select(ETS_Model)) %>%
  features(.resid, ljung_box, lag=24, dof=16)
```

```
## # A tibble: 1 x 3
##   .model    lb_stat lb_pvalue
##   <chr>       <dbl>     <dbl>
## 1 ETS_Model    23.2   0.00315
```

```r
report(fit%>%dplyr::select(ETS_Add_BoxCox))
```

```
## Series: visitors
## Model: ETS(A,A,A)
## Transformation: box_cox(visitors, 0.3624893)
##   Smoothing parameters:
##     alpha = 0.5587622
##     beta  = 0.005073895
##     gamma = 0.1547005
##
##   Initial states:
##      l[0]       b[0]       s[0]     s[-1]     s[-2]      s[-3]    s[-4]
##  11.56647 0.05343262 -0.4152995 0.3406862 0.4568056 -0.2062445 2.116354
##      s[-5]     s[-6]      s[-7]      s[-8]     s[-9]     s[-10]   s[-11]
##  0.8264984 0.3002555 -0.9754815 -0.3773534 0.1014559 -0.9714562 -1.19622
##
##   sigma^2:  0.1611
##
##      AIC     AICc      BIC
## 784.1161 787.2070 841.4959
```

```r
fit %>%
  dplyr::select(ETS_Add_BoxCox) %>%
  gg_tsresiduals()
```




```
augment(fit %>% dplyr::select(ETS_Add_BoxCox)) %>%
  features(.resid, ljung_box, lag=24, dof=15)
```

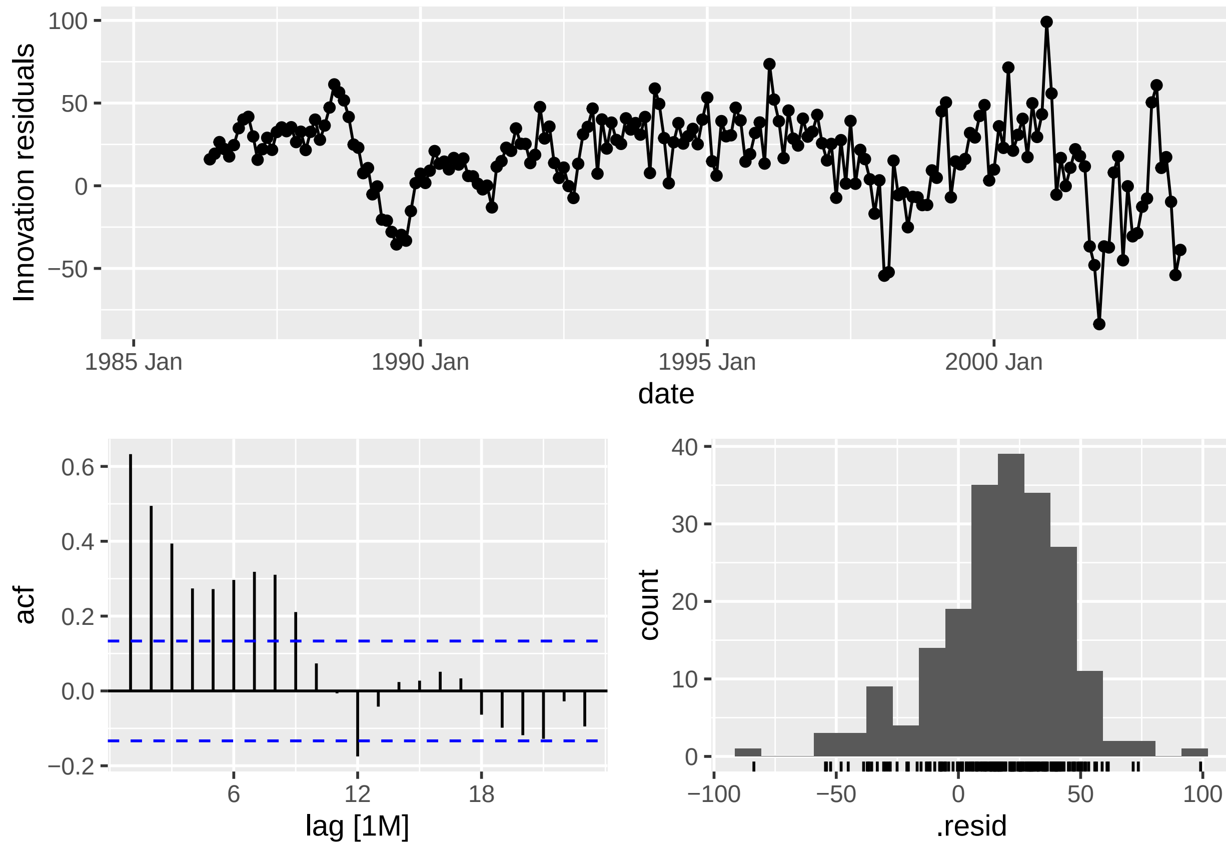
```
## # A tibble: 1 x 3
##   .model         lb_stat lb_pvalue
##   <chr>            <dbl>     <dbl>
## 1 ETS_Add_BoxCox    25.4   0.00252
```

```
report(fit%>%dplyr::select(SeasonalNaive))
```

```
## Series: visitors
## Model: SNAIVE
##
## sigma^2: 674.9436
```

```
fit %>%
  dplyr::select(SeasonalNaive) %>%
  gg_tsresiduals()
```

```
## Warning: Removed 12 rows containing missing values (`geom_line()`).
## Warning: Removed 12 rows containing missing values (`geom_point()`).
## Warning: Removed 12 rows containing non-finite values (`stat_bin()`).
```




```r
augment(fit %>% dplyr::select(SeasonalNaive)) %>%
  features(.resid, ljung_box, lag=24, dof=0)
```

```
## # A tibble: 1 x 3
##   .model        lb_stat lb_pvalue
##   <chr>           <dbl>     <dbl>
## 1 SeasonalNaive    295.         0
```

All three models failed Residual tests because of the fact theri P-Value is less than 0.05.

Question 3

3. Consider usmelec (usmelec.csv), the total net generation of electricity (in billion kilowatt hours) by the U.S. electric industry (monthly for the period January 1973 – June 2013). In general there are two peaks per year: in mid-summer and mid-winter. (Total 36 points)

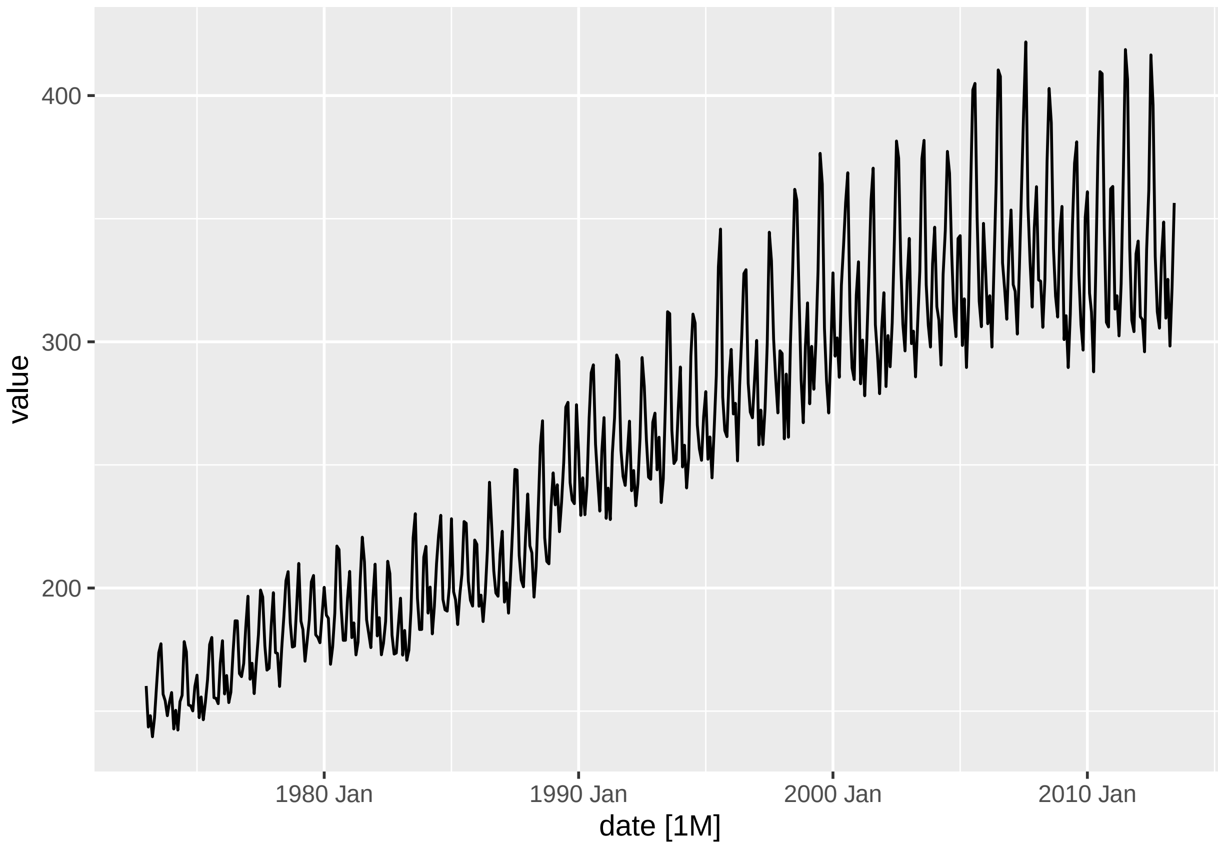
3.1 Examine the 12-month moving average of this series to see what kind of trend is involved. (4 points)

```r
usmelec <- readr::read_csv("usmelec.csv") %>%
  mutate(date = yearmonth(index)) %>%
  dplyr::select(-index) %>%
  as_tsibble(
    index = date
  )
```

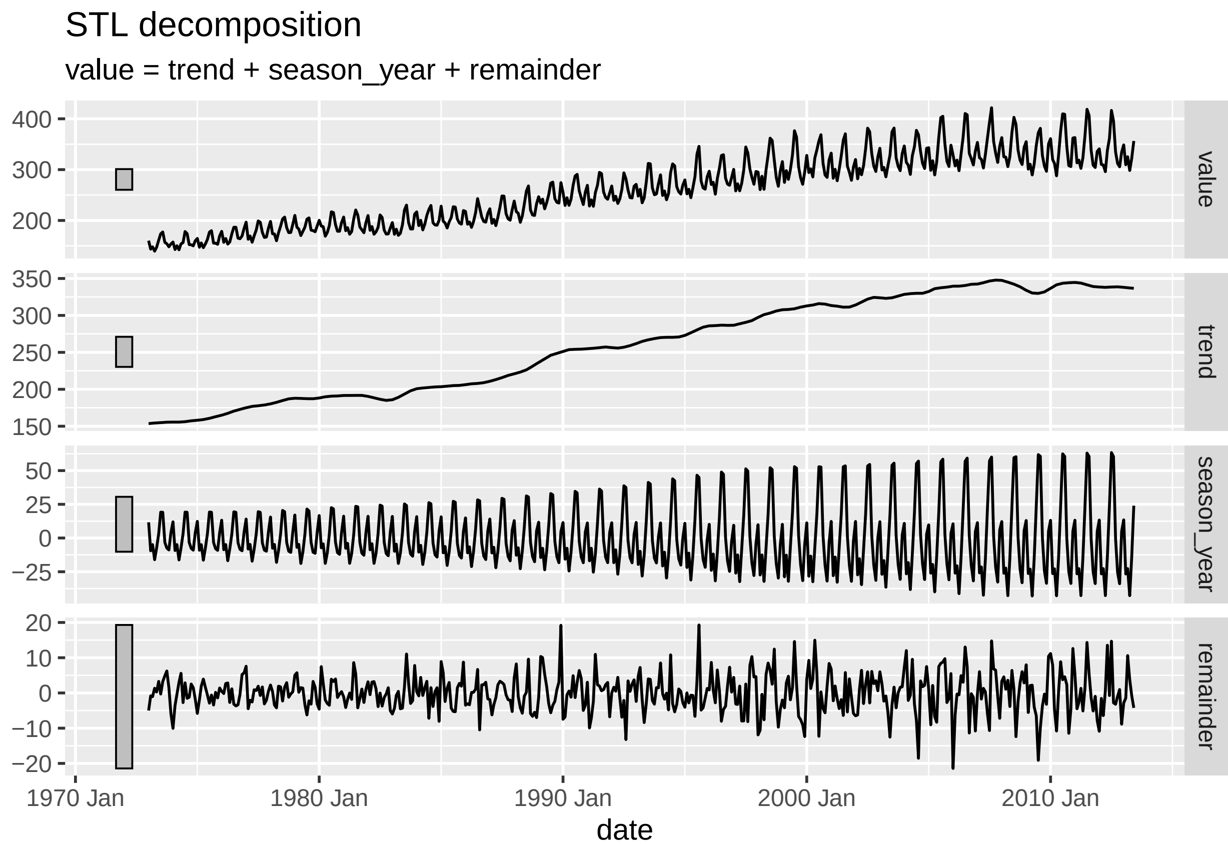
```
## Rows: 486 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): index
```

```
## dbl (1): value
##
## 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.
```

```
usmelec %>% autoplot(value)
```




```
dcmp <- usmelec %>% model(stl = STL(value))
components(dcmp) %>% autoplot()
```

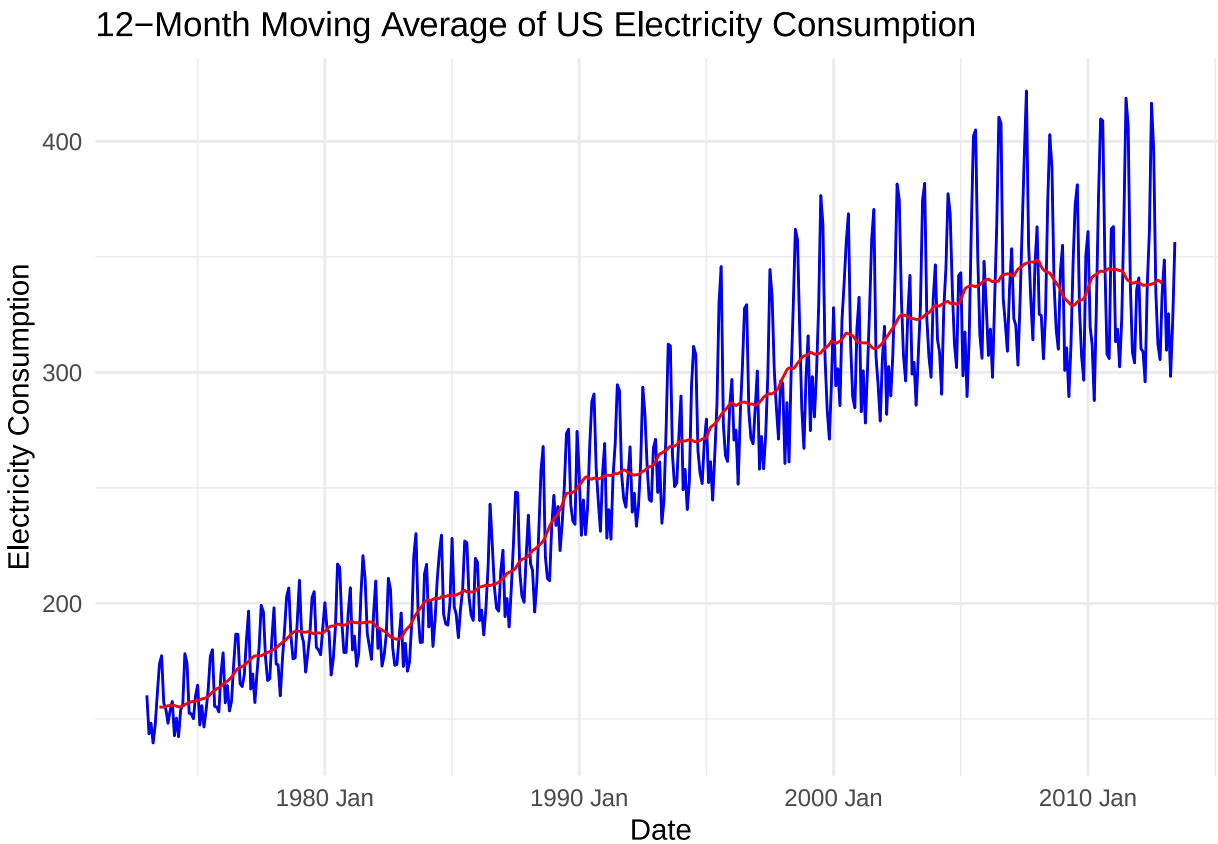



```
library(slider)


usmelec_ma <- usmelec %>%
  mutate(moving_avg = 1/2*((slide_dbl(value, mean, .before = 5, .after = 6, .complete = TRUE)) +(slide_

ggplot(usmelec_ma, aes(x = date)) +
  geom_line(aes(y = value), color = "blue") +
  geom_line(aes(y = moving_avg), color = "red") +
  labs(title = "12-Month Moving Average of US Electricity Consumption",
       x = "Date",
       y = "Electricity Consumption") +
  theme_minimal()
```

```
## Warning: Removed 12 rows containing missing values (`geom_line()`).
```



Even after doing 12 month moving average we can see an upward trend in the date.

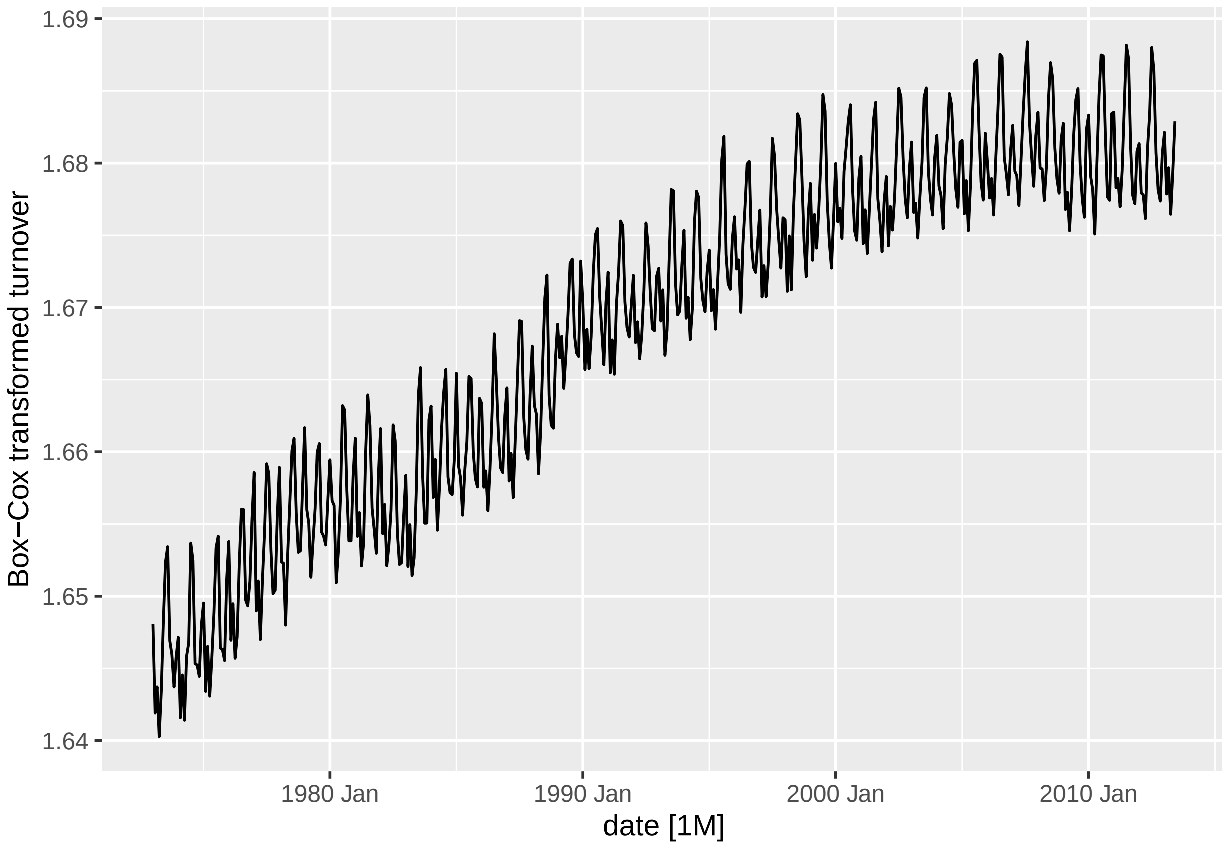
3.2 Do the data need transforming? If so, find a suitable transformation. (4 points)

The impact of seasonal fluctuations, which are proportionate to the level of the time series, can be mitigated by applying a Box-Cox transformation. To determine the appropriate lambda parameter for this transformation, the Guerrero method can be employed.

```
usmelec %>%
  features(value, features = guerrero)
```

```
## # A tibble: 1 x 1
##   lambda_guerrero
##             <dbl>
## 1          -0.574
```

```
usmelec %>% autoplot(box_cox(value, -0.5738168)) +
  labs(y = "Box-Cox transformed turnover")
```



3.3 Are the data stationary? If not, find an appropriate differencing which yields stationary data. (4 points)

```
usmelec_transformed = usmelec %>% mutate(value = box_cox(value, -0.5738168))
```

```
library(tseries)
```

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

```
# Extract the time series data
ts_data <- as.numeric(usmelec_transformed$value)

# Perform the KPSS Test
kpss_result <- kpss.test(ts_data)
```

```
## Warning in kpss.test(ts_data): p-value smaller than printed p-value
```

```
# Print the results
print(kpss_result)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  ts_data
## KPSS Level = 7.8337, Truncation lag parameter = 5, p-value = 0.01
```

For KPSS test we can see that P-value is less than alpha which is 0.05. So we reject the null hypothesis suggestung that timeseries is not stationary.

```r
usmelec_transformed %>% features(value, unitroot_nsdiffs)
```

```
## # A tibble: 1 x 1
##   nsdiffs
##     <int>
## 1       1
```

Shows it has a seasonal difference of 1.

```r
usmelec_transformed %>% mutate(d_log_value = difference(value, 12)) %>%features(d_log_value, unitroot_nc
```

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

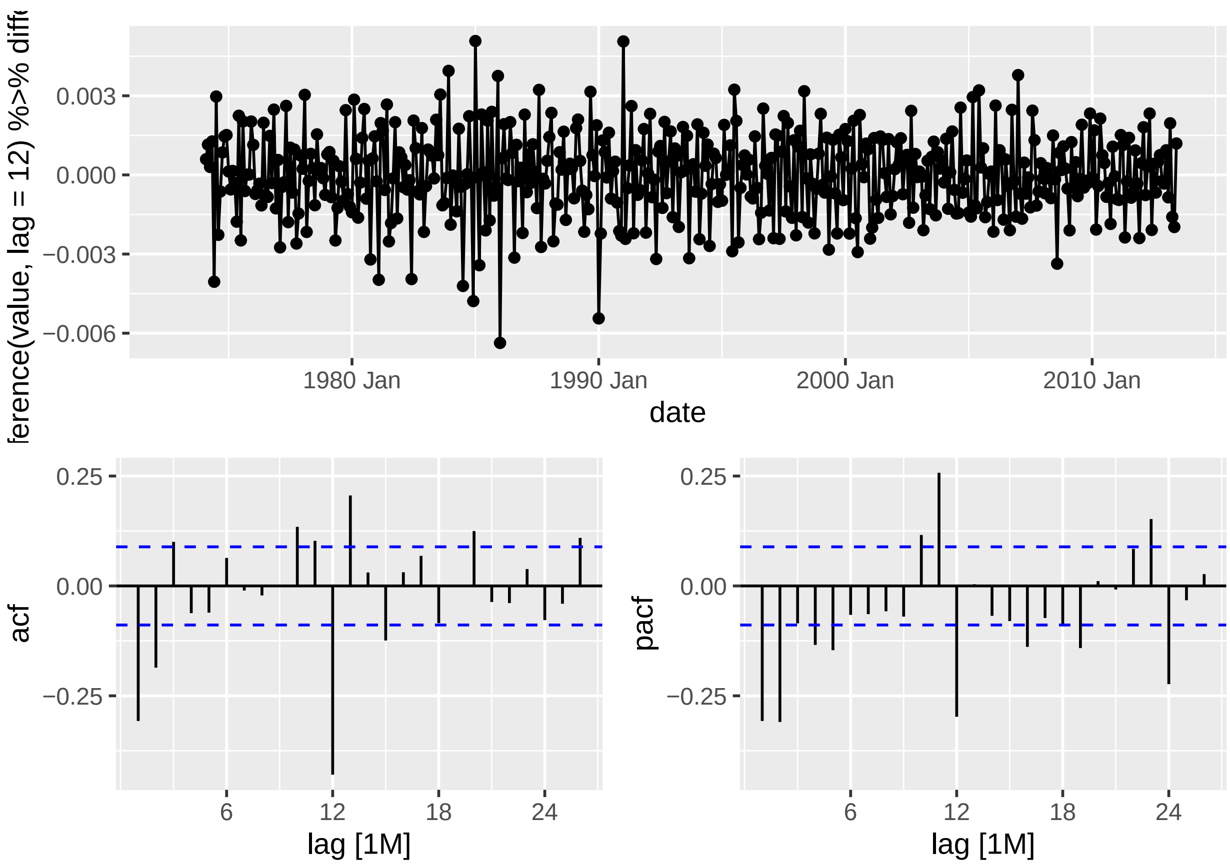
There is a difference of 1 required in non-seasonal aspect.

3.4 Identify a couple of ARIMA models that might be useful in describing the time series. Which of your models is the best according to their AIC values? (6 points)

```r
usmelec_transformed %>% gg_tsdisplay(difference(value, lag=12) %>% difference(), plot_type='partial')
```

```
## Warning: Removed 13 rows containing missing values (`geom_line()`).

## Warning: Removed 13 rows containing missing values (`geom_point()`).
```




```r
fit_arima = usmelec_transformed %>% model(
  arima_211 = ARIMA(value ~ pdq(2,1,1)+PDQ(2,1,1)),
  arima_112 = ARIMA(value ~ pdq(1,1,2)+PDQ(2,1,1)),
```

```r
  arima_111 = ARIMA(value ~ pdq(1,1,1)+PDQ(2,1,1))
  )
report(fit_arima)
```

```
## Warning in report.mdl_df(fit_arima): Model reporting is only supported for
## individual models, so a glance will be shown. To see the report for a specific
## model, use `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 3 x 8
##   .model        sigma2 log_lik    AIC   AICc    BIC ar_roots   ma_roots
##   <chr>          <dbl>   <dbl>  <dbl>  <dbl>  <dbl> <list>     <list>
## 1 arima_211 0.00000128   2548. -5081. -5081. -5052. <cpl [26]> <cpl [13]>
## 2 arima_112 0.00000128   2548. -5081. -5081. -5052. <cpl [25]> <cpl [14]>
## 3 arima_111 0.00000127   2547. -5083. -5082. -5058. <cpl [25]> <cpl [13]>
```

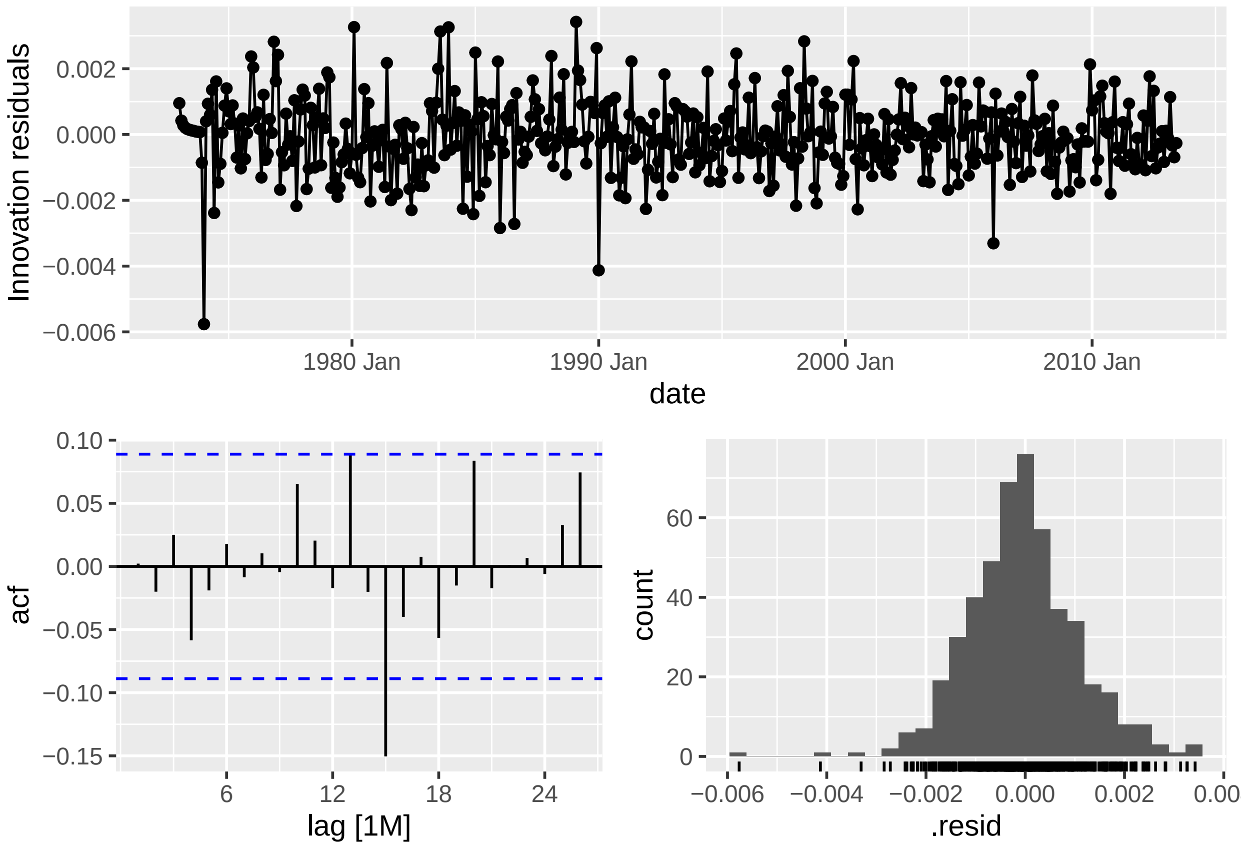
`ARIMA(211,211) has lowest AIC value hence it might be better value.`

‘ 3.5 Estimate the parameters of your best model and do diagnostic testing on the residuals. Do the residuals resemble white noise? If not, try to find another ARIMA model which fits better. (4 points)

```r
report(fit_arima %>% dplyr::select(arima_211))
```

```
## Series: value
## Model: ARIMA(2,1,1)(2,1,1)[12]
##
## Coefficients:
##          ar1      ar2      ma1    sar1     sar2     sma1
##       0.3815  -0.0401  -0.8080  0.0374  -0.0947  -0.8463
## s.e.  0.0658   0.0546   0.0482  0.0557   0.0532   0.0342
##
## sigma^2 estimated as 1.276e-06:  log likelihood=2547.58
## AIC=-5081.17   AICc=-5080.93   BIC=-5052.06
```

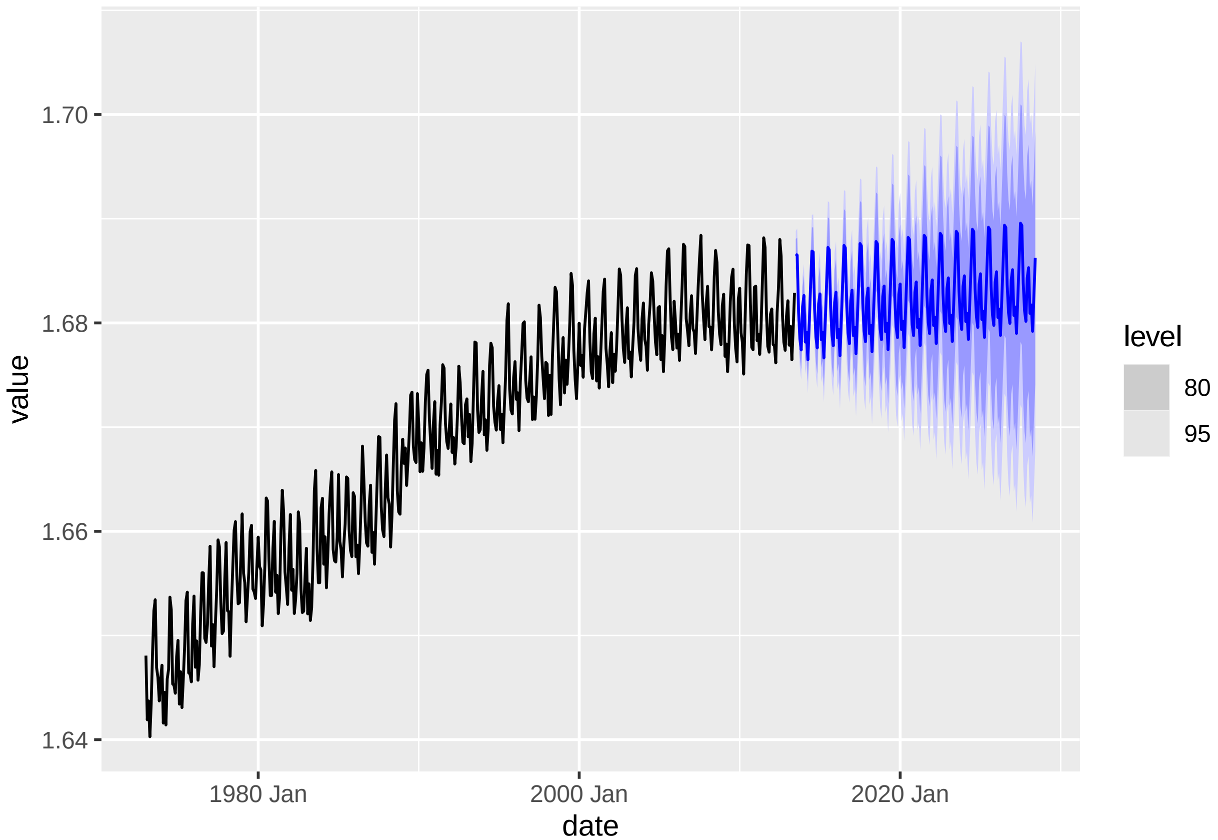
```r
fit_arima %>% dplyr::select(arima_211) %>% gg_tsresiduals()
```

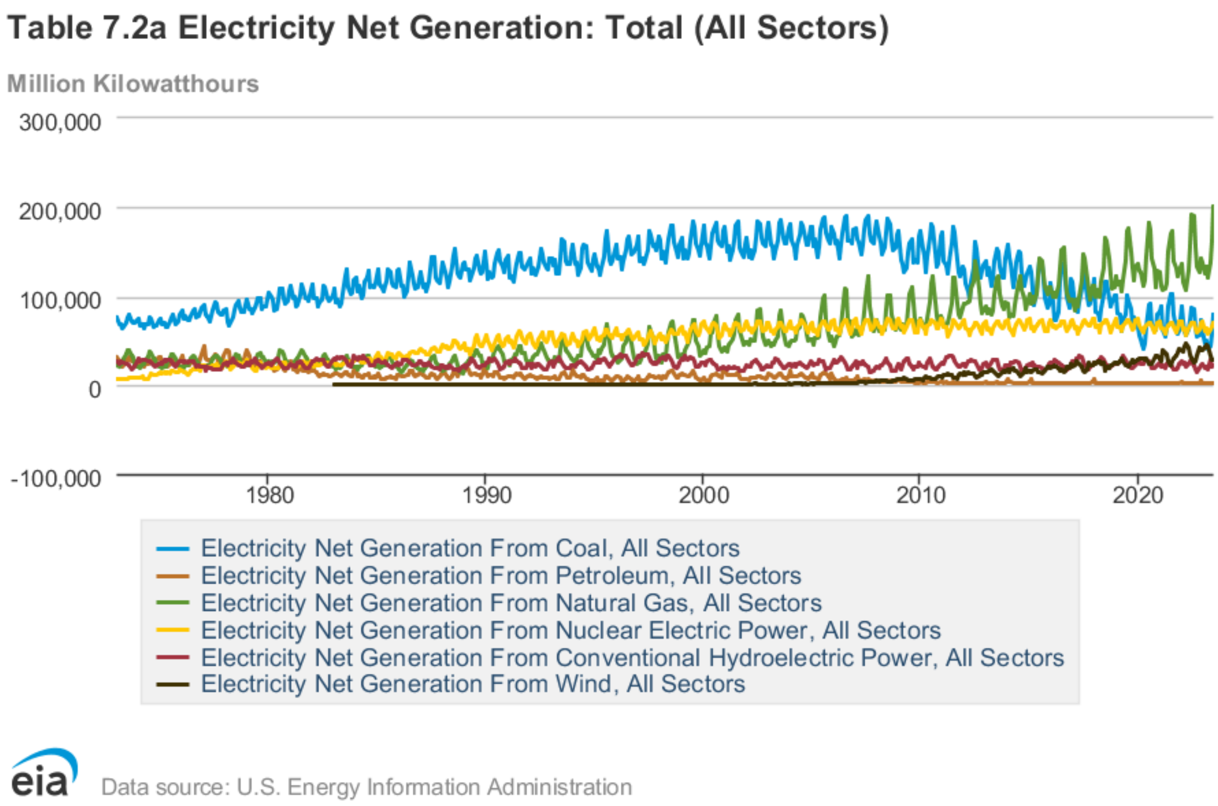


The errors actually resembles white noise. We can see other than at lag 15 there is no auto-correlation between residuals and residuals are normally distributed and has mean zero and almost constant variance. Hence we can say it is very close to white noise.

3.6 Forecast the next 15 years of electricity generation by the U.S. electric industry. Get the latest figures from the EIA (https://www.eia.gov/totalenergy/data/monthly/#electricity) to check the accuracy of your forecasts. (8 points)

```
fit_arima %>% dplyr::select(arima_211) %>% forecast(h = 180) %>% autoplot(usmelec_transformed)
```





If we add all the values from the different sectors it is near to what we have forecasted.

3.7. Eventually, the prediction intervals are so wide that the forecasts are not particularly useful. How many years of forecasts do you think are sufficiently accurate to be usable? (6 points)

Forecasting beyond two years can often lead to less reliable predictions as the confidence intervals tend to widen exponentially, reflecting increasing uncertainty over time.