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)

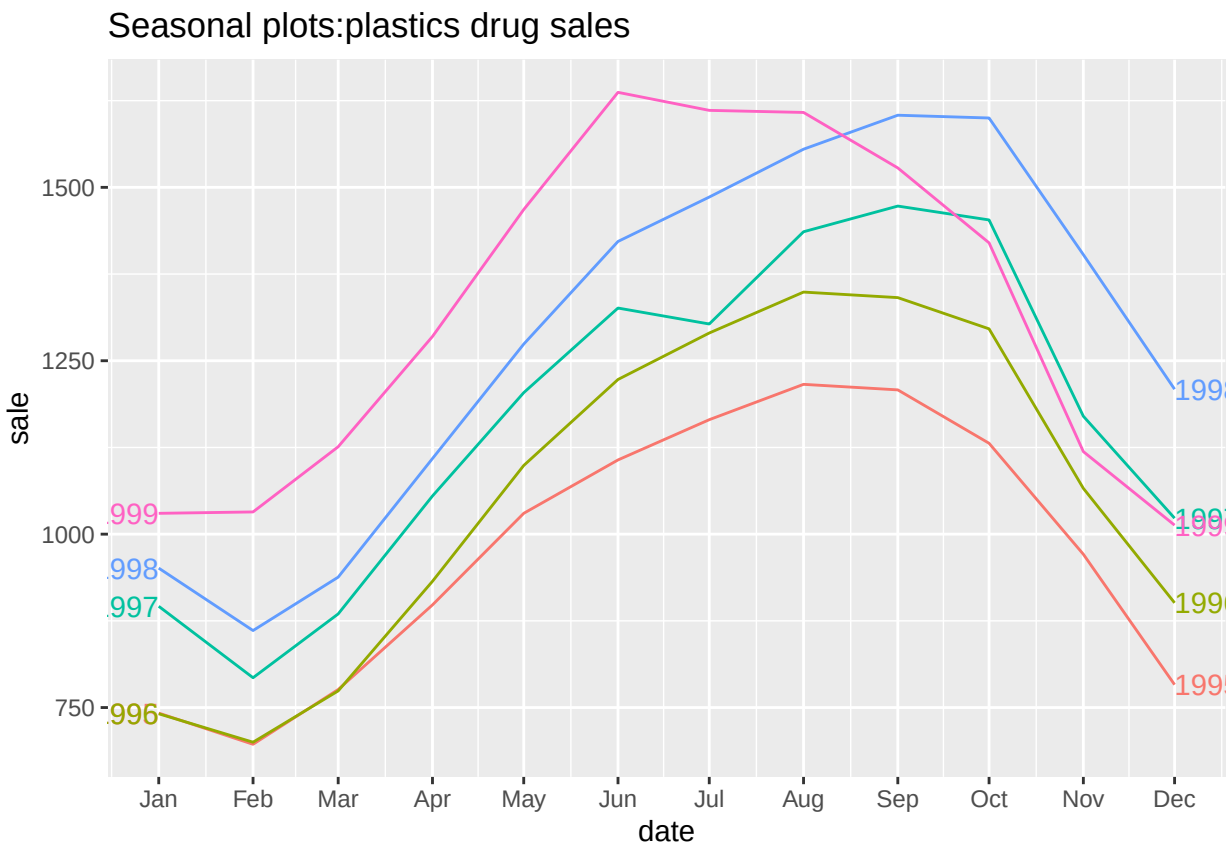
1.1 Read csv file and convert to tsible with proper index (2 points)

```
plastics_df<- read.csv("plastics.csv")

mutate(plastics_df,date=yearmonth(date)) %>%
tsibble(index=date) ->plastics_df

head(plastics_df)
```

```
## # A tsibble: 6 x 2 [1M]
##       date  sale
##      <mth> <int>
## 1 1995 Jan   742
## 2 1995 Feb   697
## 3 1995 Mar   776
## 4 1995 Apr   898
## 5 1995 May  1030
## 6 1995 Jun  1107
```

1.2 Plot the time series of sales of product A. Can you identify seasonal fluctuations and/or a trend-cycle? (2 points) #Yes, Sales exhibit a definite seasonal trend, increasing in the summer and falling off in the winter.

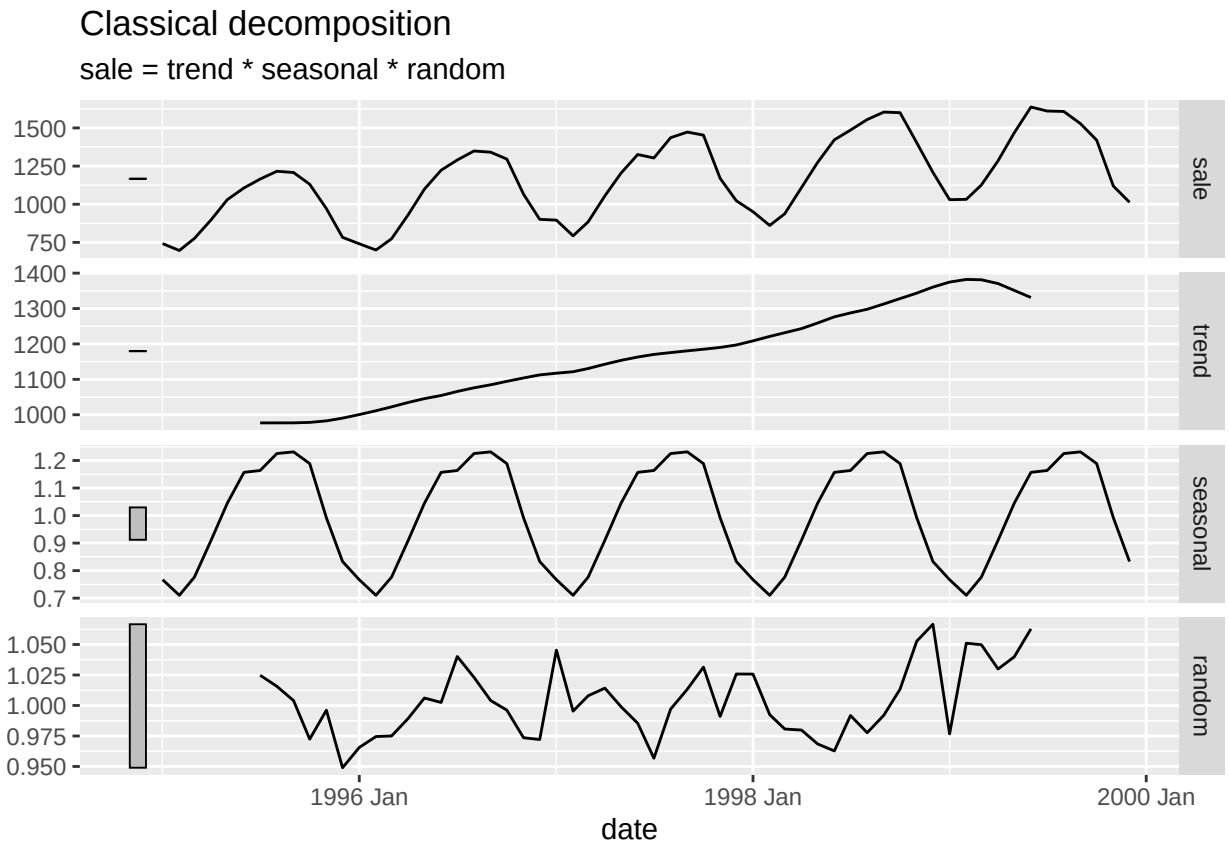
```
plastics_df %>% gg_season(sale, labels = "both") +
  labs(y = "sale", title = "Seasonal plots:plastics drug sales")
```



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

```
model = plastics_df %>% model(classical_decomposition(sale, type='m'))
components(model) %>% autoplot()
```

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

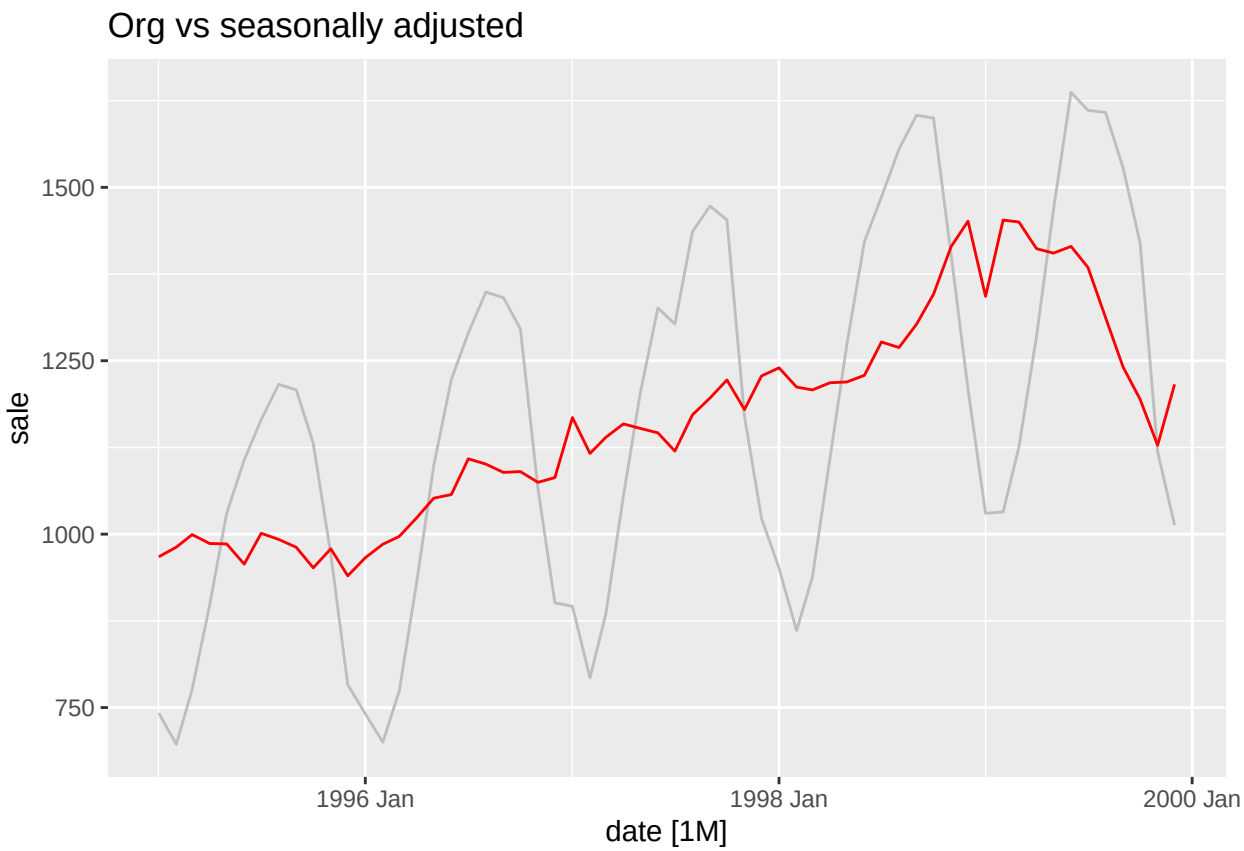


1.4 Do the results support the graphical interpretation from part a? (2 points) #Yes, it support the observations.

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

```
adjusted <- model %>% components() %>% select(season_adjust)
plastics_df %>%
  autoplot(sale, color = "gray") +
  autolayer(adjusted, series = "Seasonally Adjusted", color = "red") +
  labs(
    title = "Org vs seasonally adjusted",
  )
```

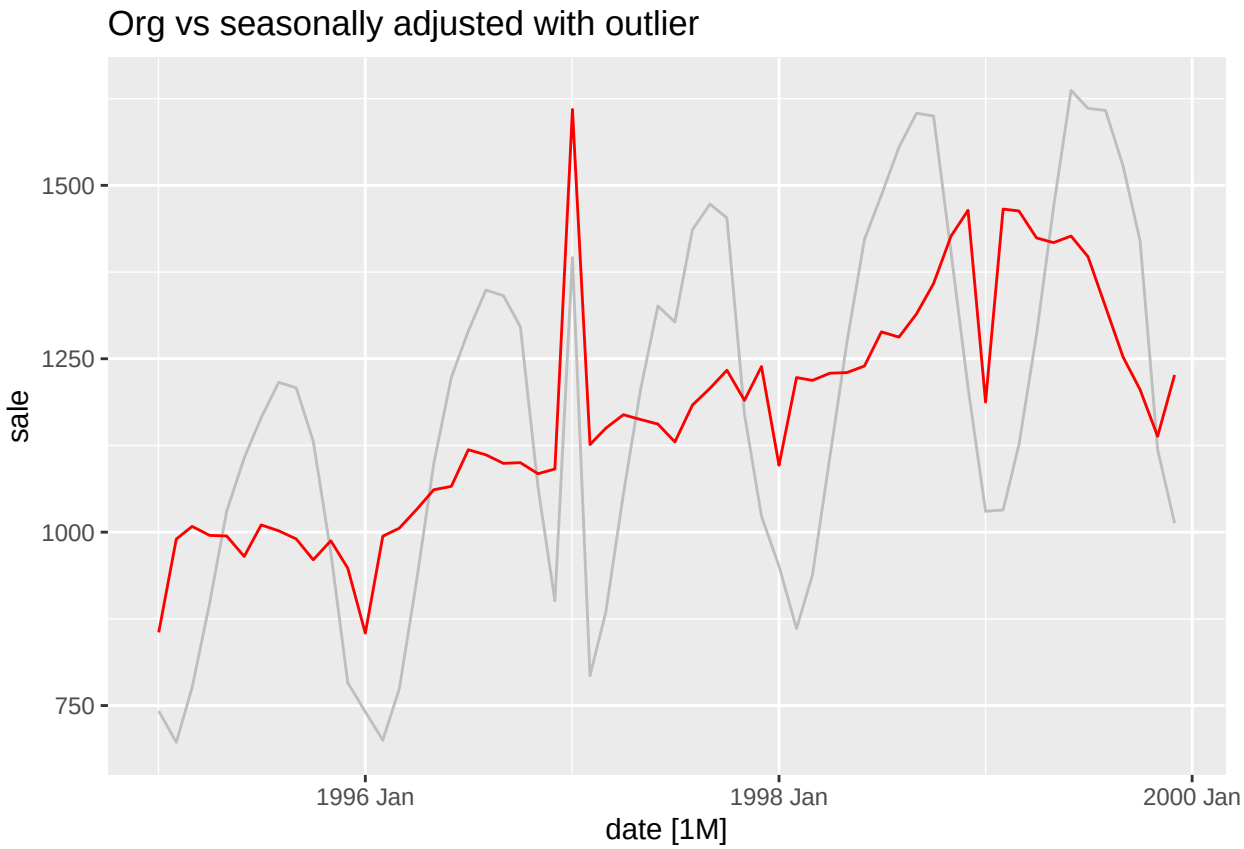
```
## Plot variable not specified, automatically selected `.vars = season_adjust`

## Warning in geom_line(eval_tidy(expr(aes(!!!aes_spec))), data = object, ..., :
## Ignoring unknown parameters: `series`
```



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) # The outlier has a good effect on the graph because we see could a spike. It may increase average value of the data.

```
plastics_df_new <- plastics_df

plastics_df_new$sale[25] <- plastics_df_new$sale[25] + 500
new_model <- plastics_df_new %>% model(classical_decomposition(sale, type='m'))

plastics_df_new %>%
  autoplot(sale, color="gray") +
  autolayer(select(components(new_model),season_adjust),color = "red") +
  labs(
    title = "Org vs seasonally adjusted with outlier",
  )
```

```
## Plot variable not specified, automatically selected `.vars = season_adjust`
```



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

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) #If the outlier is present at end then there is no significant impact, however if the outlier is at middle there could be impact towards the trend of the timeseries.

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)

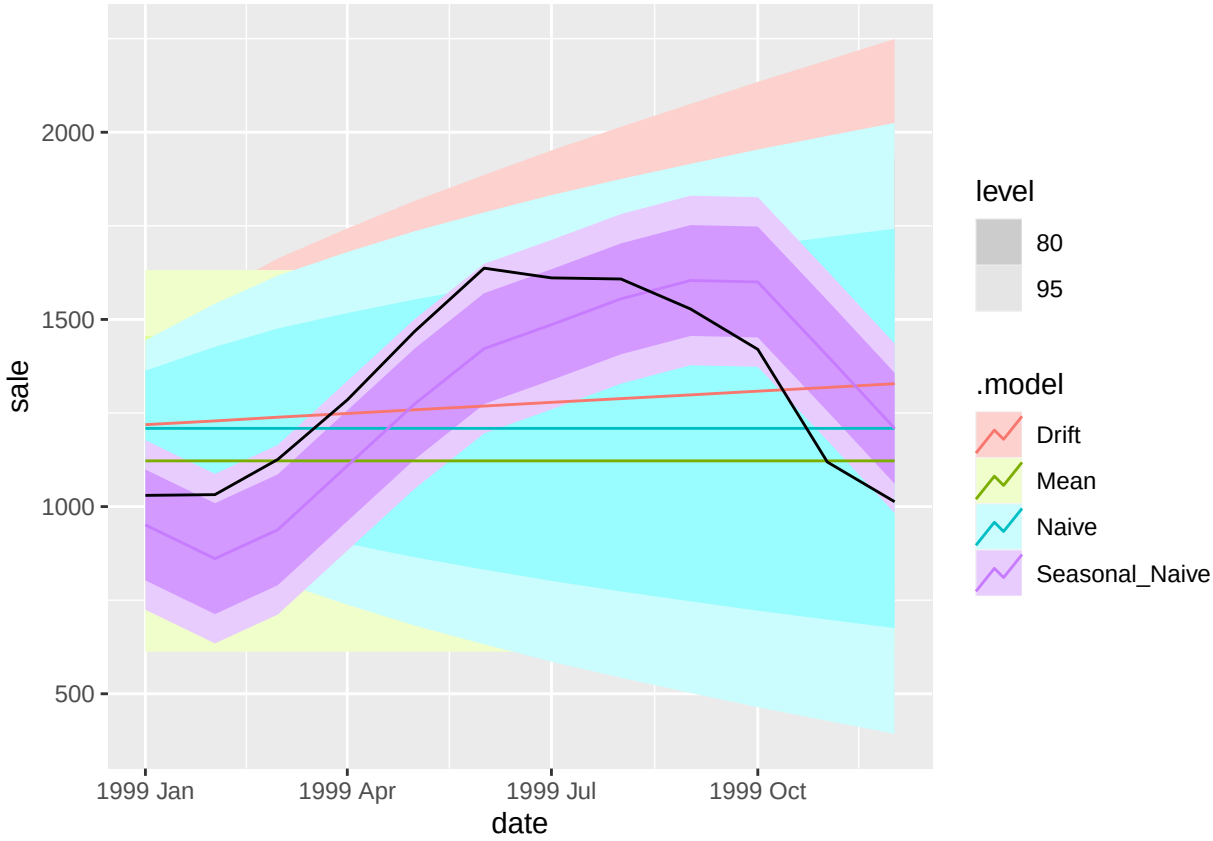
```
train <- plastics_df %>%
  filter(date <= yearmonth("1998-12"))

test <- plastics_df %>%
  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)

#Solution: As per the accuracy and other factor I feel Seasonal_Naive performs best.

```
fit <- train %>%
  model(
    Mean = MEAN(sale),
    Naive = NAIVE(sale),
    Seasonal_Naive = SNAIVE(sale),
    Drift = RW(sale ~ drift())
  )
```

```
pred <- forecast(fit,h = 12)
autoplot(pred,test)
```



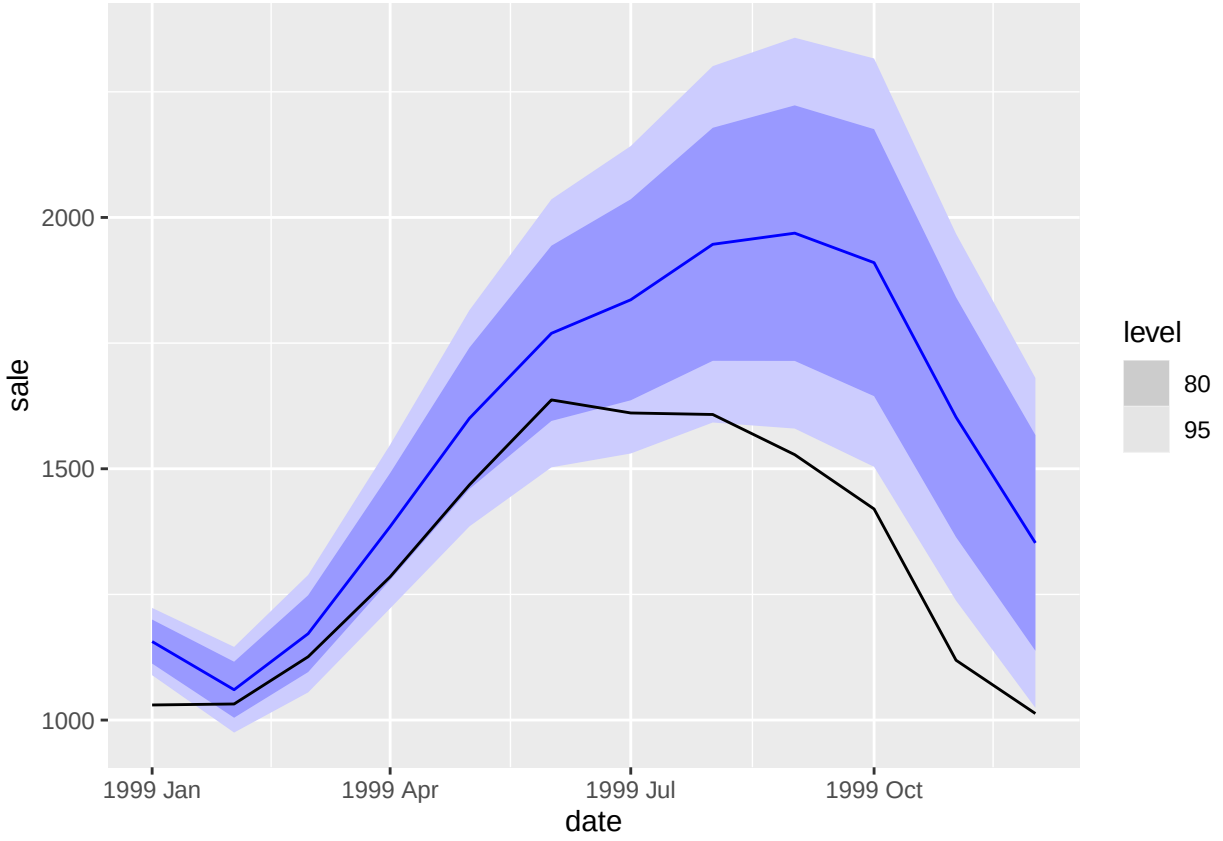
1.10 Repeat 1.9 for appropriate EST. Report the model. Check residuals. Plot forecasts and actual data. (4 points)

```
fit_ets <- train %>%
  model(ets_auto = ETS(sale))

report(fit_ets[1])
```

```
## Series: sale
## Model: ETS(M,Ad,M)
##   Smoothing parameters:
##     alpha = 0.9047366
##     beta  = 0.06929627
##     gamma = 0.0001231683
##     phi   = 0.9743239
##
##   Initial states:
##      l[0]     b[0]      s[0]     s[-1]    s[-2]    s[-3]    s[-4]    s[-5]
##  941.1877 10.06324 0.8274347 0.9880802 1.187251 1.233995 1.230671 1.171347
##     s[-6]    s[-7]     s[-8]     s[-9]    s[-10]    s[-11]
##  1.139274 1.040372 0.9093021 0.7774054 0.7109808 0.7838869
##
```

```
##   sigma^2:  9e-04
##
##      AIC     AICc      BIC
## 533.9618 557.5480 567.6434
```

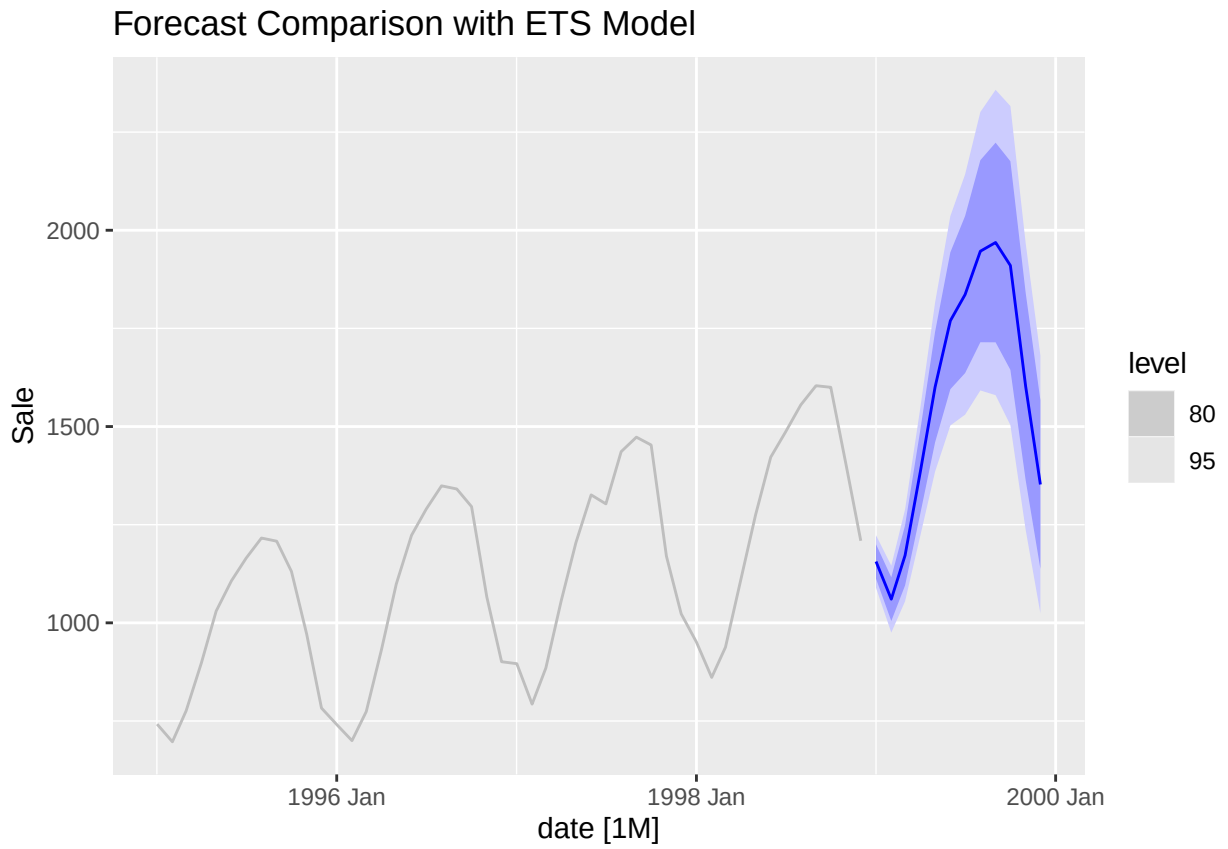
```r
fit_ets <- fit_ets[1]


pred_ets <- forecast(fit_ets, h = 12)

autoplot(pred_ets,test)
```



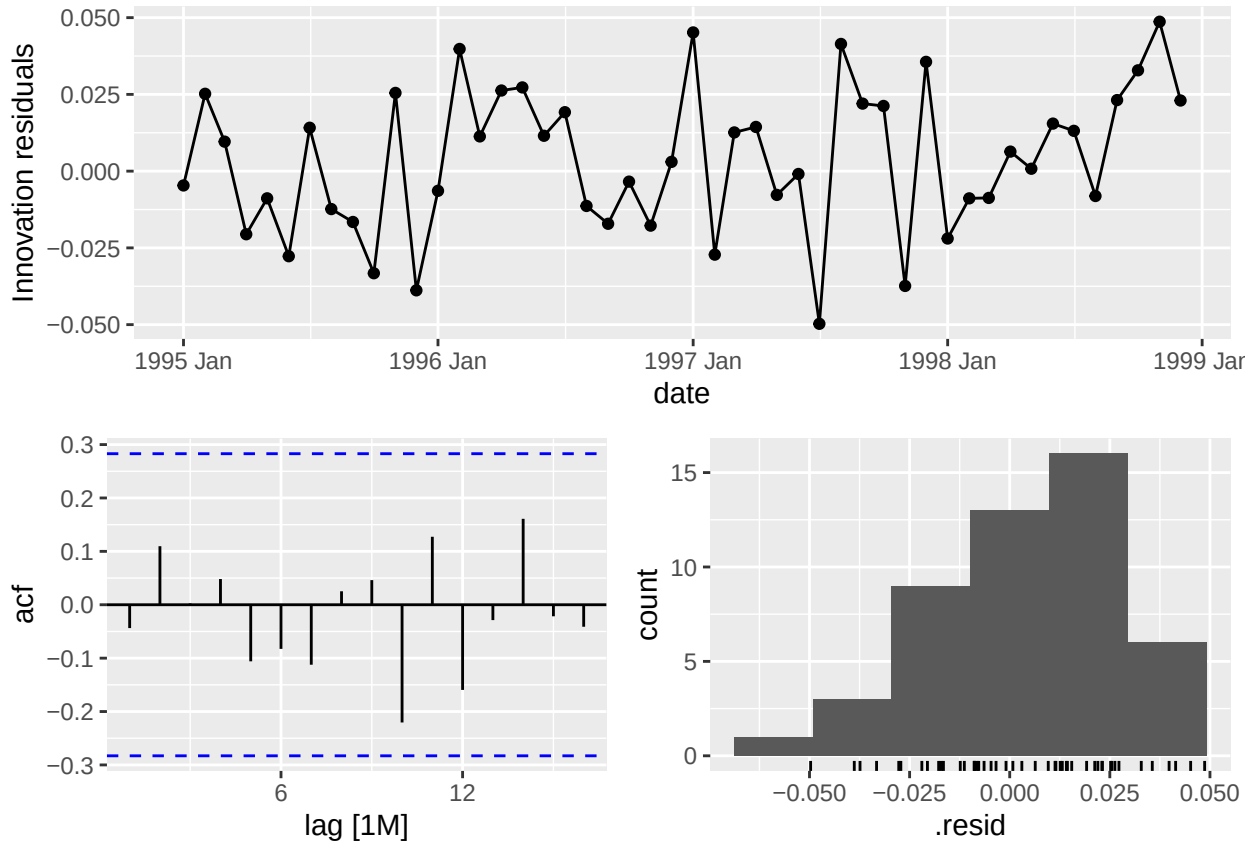

```r
autoplot(train, sale, color = "gray") +
  autolayer(pred_ets, series = "ETS", color = "blue") +
  labs(title = "Forecast Comparison with ETS Model",
       y = "Sale",
       color = "Forecast Method")
```

```
## Warning in distributional::geom_hilo_ribbon(intvl_mapping, data =
## dplyr::anti_join(interval_data, : Ignoring unknown parameters: 'series'

## Warning in distributional::geom_hilo_linerange(intvl_mapping, data =
## dplyr::semi_join(interval_data, : Ignoring unknown parameters: 'series'

## Warning in geom_line(mapping = without(mapping, "shape"), data =
## dplyr::anti_join(object, : Ignoring unknown parameters: 'series'
```

```
## Warning in ggplot2::geom_point(mapping = without(mapping, "linetype"), data =
## dplyr::semi_join(object, : Ignoring unknown parameters: ‘series‘
```




```
gg_tsresiduals(fit_ets)
```

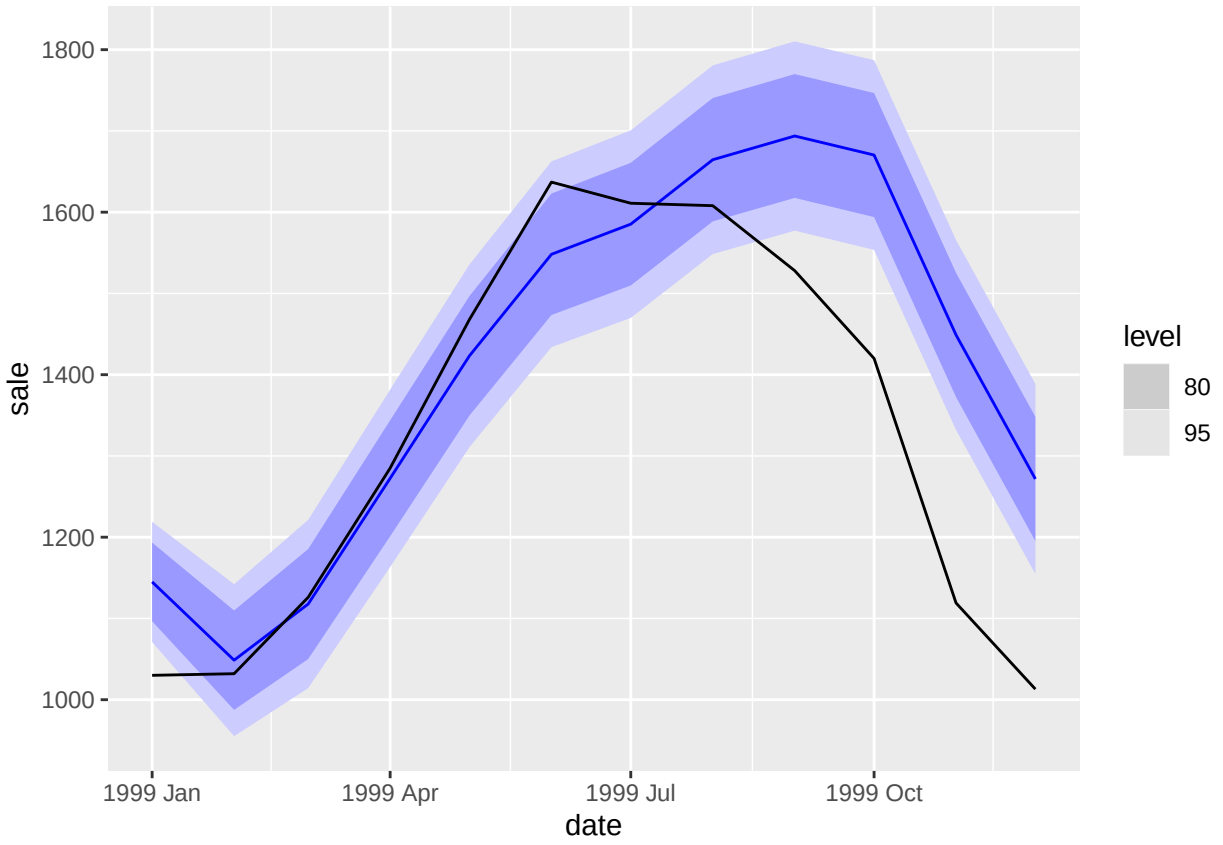


1.11 Repeat 1.9 for appropriate ARIMA. Report the model. Check residuals. Plot forecasts and actual data. (4 points)

```
fit_arima <- train %>%
  model(arima_auto = ARIMA(sale))

report(fit_arima[1])
```

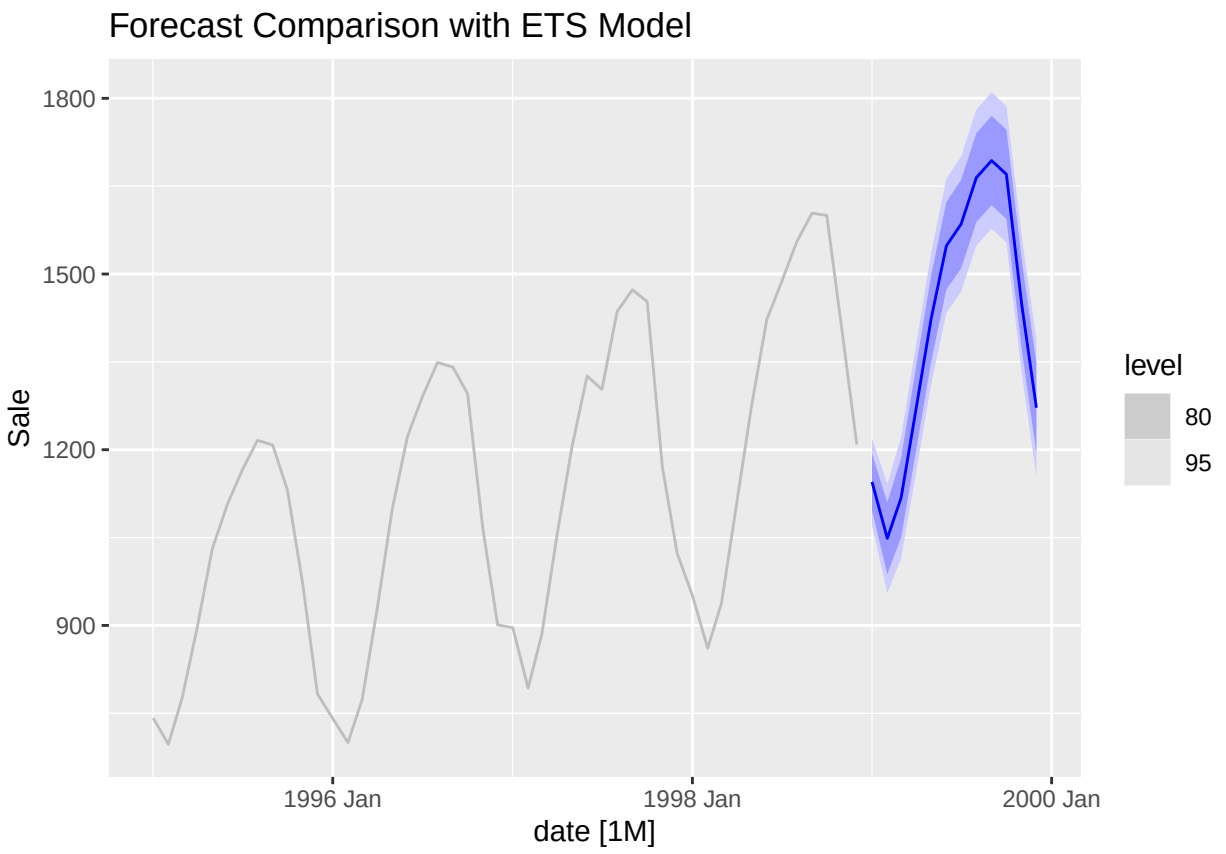
```
## Series: sale
## Model: ARIMA(1,0,0)(0,1,1)[12] w/ drift
##
## Coefficients:
##          ar1     sma1  constant
##       0.7769  -0.5223   23.1041
## s.e.  0.1237   0.3394    3.8041
##
## sigma^2 estimated as 1410:  log likelihood=-182.35
## AIC=372.7   AICc=373.99   BIC=379.03
```

```
fit_arima <- fit_arima[1]

pred_arima <- forecast(fit_arima, h = 12)

autoplot(pred_arima,test)
```

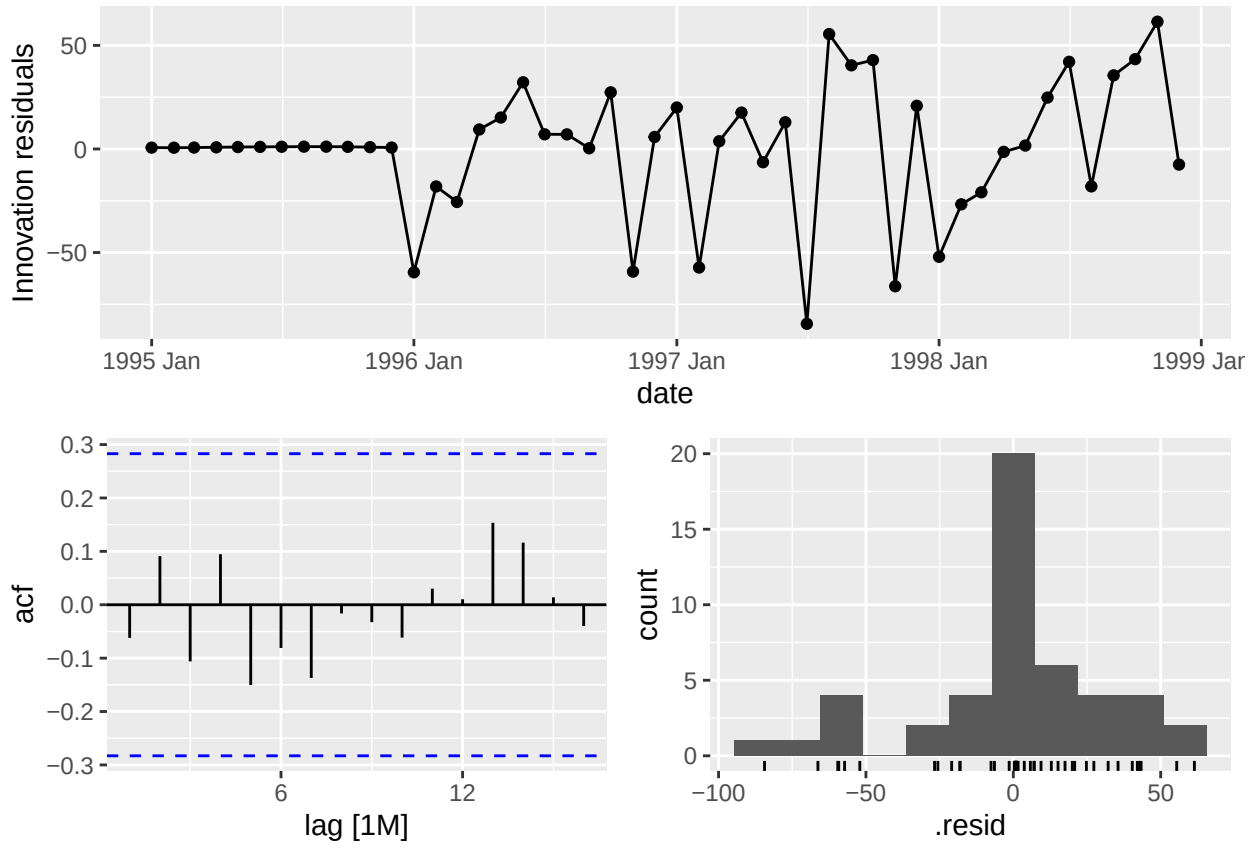



```
autoplot(train, sale, color = "gray") +
  autolayer(pred_arima, series = "ETS", color = "blue") +
  labs(title = "Forecast Comparison with ETS Model",
       y = "Sale",
       color = "Forecast Method")
```

```
## Warning in distributional::geom_hilo_ribbon(intvl_mapping, data =
## dplyr::anti_join(interval_data, : Ignoring unknown parameters: 'series'

## Warning in distributional::geom_hilo_linerange(intvl_mapping, data =
## dplyr::semi_join(interval_data, : Ignoring unknown parameters: 'series'

## Warning in geom_line(mapping = without(mapping, "shape"), data =
## dplyr::anti_join(object, : Ignoring unknown parameters: 'series'

## Warning in ggplot2::geom_point(mapping = without(mapping, "linetype"), data =
## dplyr::semi_join(object, : Ignoring unknown parameters: 'series'
```




```
gg_tsresiduals(fit_arima)
```



1.12 Which model has best performance? (2 points) #Solution: I feel Ets models performance best, by conisdering different paratmers and accuracy metrics.

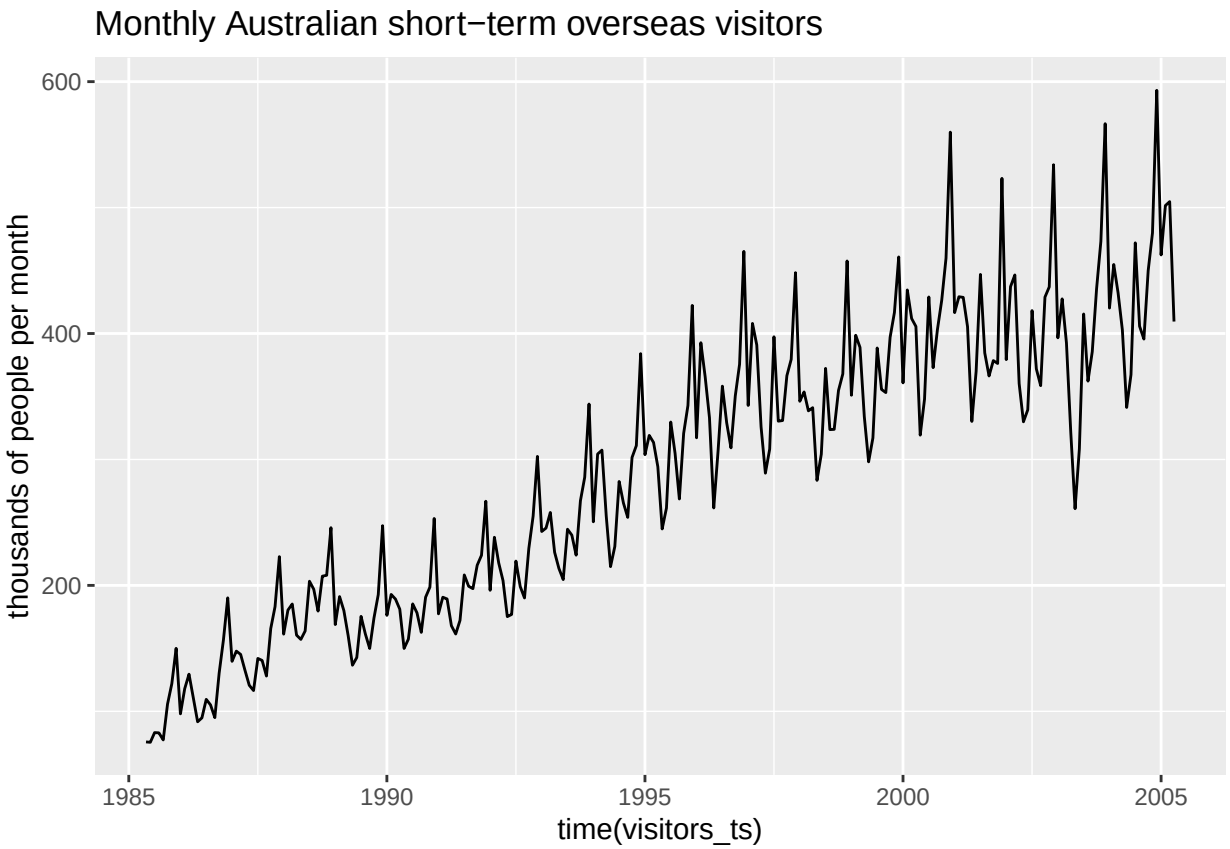
Question 2

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) #Soln: We could see that there's general upward trend over the time, which means increase of visitors overseas visitors. But I could see some decline in the visiitors near 2003 and 2004.

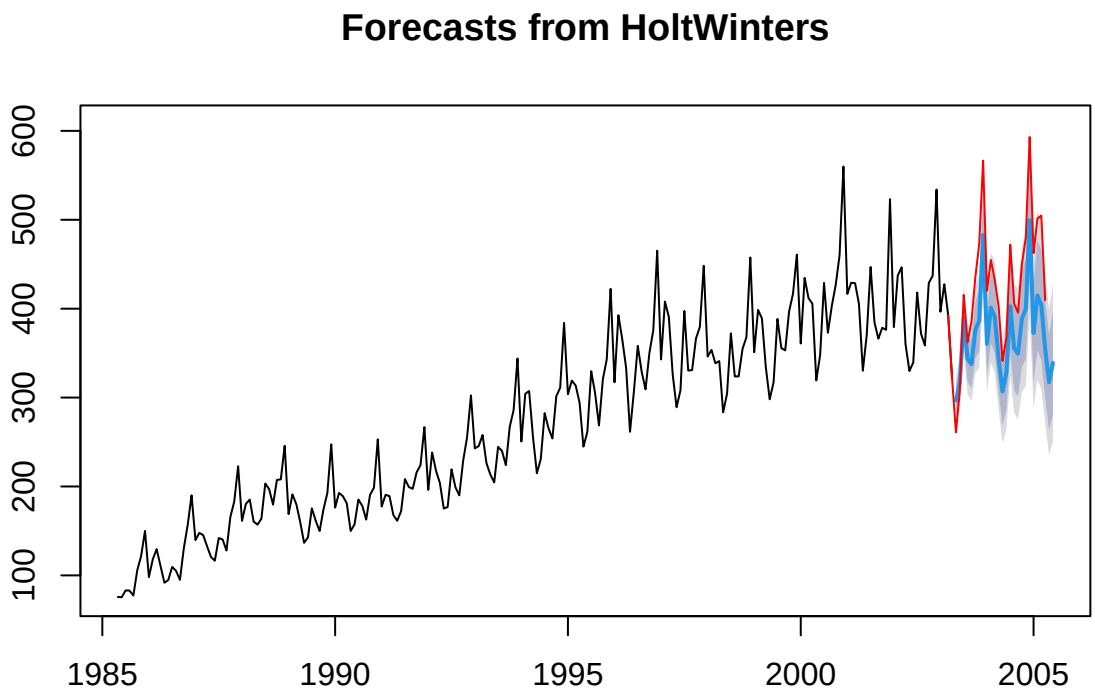
```
visitors_df <- fread("visitors.csv") %>%
  as.data.frame()
visitors_ts <- ts(visitors_df$visitors, start = c(1985, 5),frequency = 12)
temp <- data.frame(visitors_ts)
ggplot(data = temp, aes(x = time(visitors_ts),y =visitors_ts)) +
  geom_line()+
  labs(title = "Monthly Australian short-term overseas visitors", y = "thousands of people per month")
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```



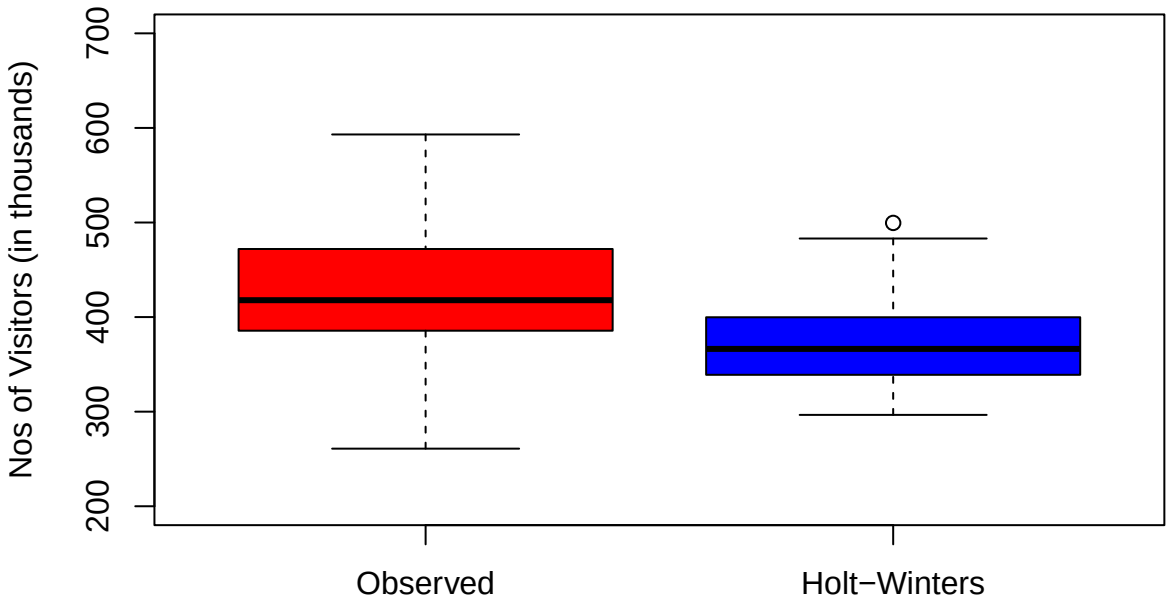
2.2 Split your data into a training set and a test set comprising the last two years of available data. Forecast the test set using Holt-Winters' multiplicative method. (6 points)

```
train <- window(visitors_ts, end = c(2003,4))
test <- window(visitors_ts, start = c(2003,3))
visitors_Holt_Winters <- HoltWinters(train, seasonal = "multiplicative")
HoltWinters_forecast <- forecast(visitors_Holt_Winters, h = length(test))
plot(HoltWinters_forecast)
lines(test, col ="red")
```



2.3. Why is multiplicative seasonality necessary here? (6 points) #Solution: By seeing the box plot we could see multiplicative seasonality necessary as we could see seasonal variation is proportional to level of series. We can also say that as the series increases we can see the peaks and trough of a season. We can see this that the median values are higher in the summers when the peak level is observerd in the series.

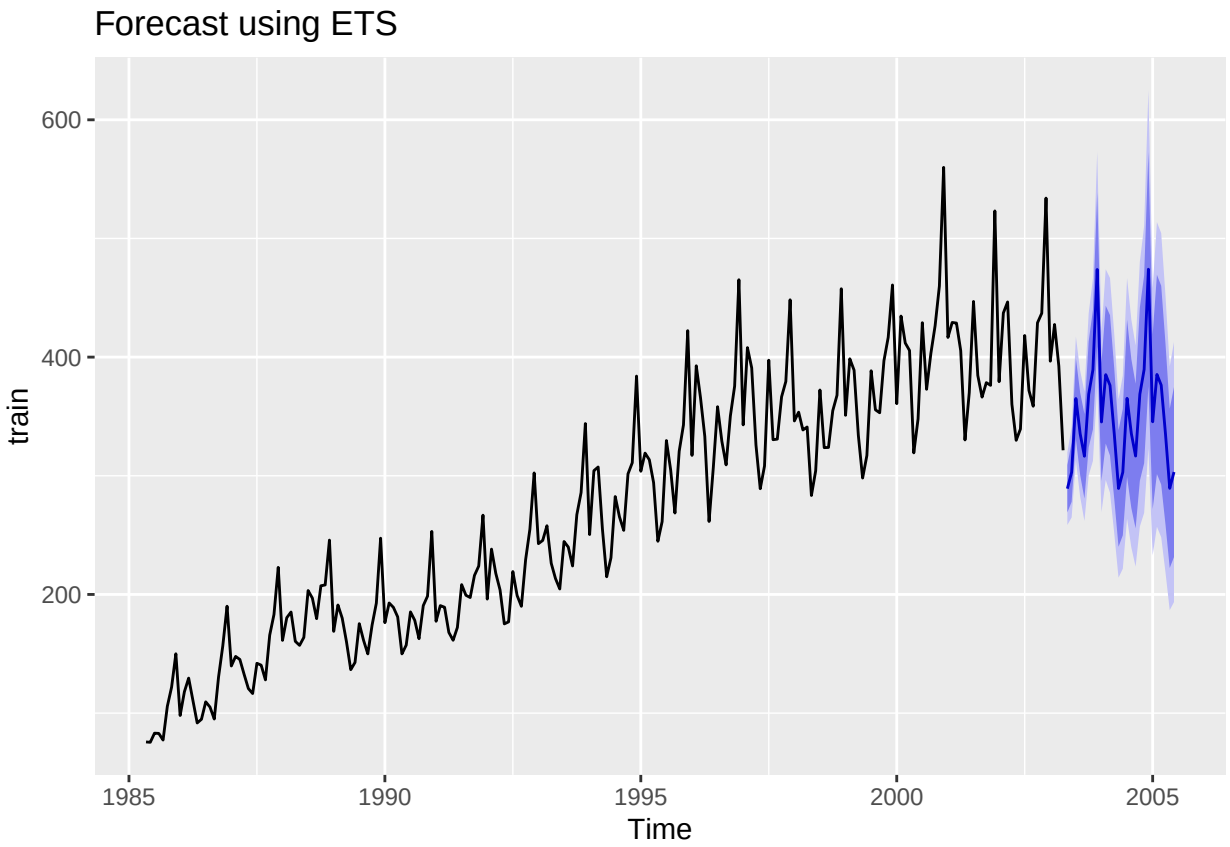
```
ylim_range <- c(200, 700)

# Create the boxplot
boxplot(test, HoltWinters_forecast$mean,
  names = c("Observed", "Holt-Winters"),
  col = c("red", "blue"),
  ylab = "Nos of Visitors (in thousands)",
  ylim = ylim_range)
```



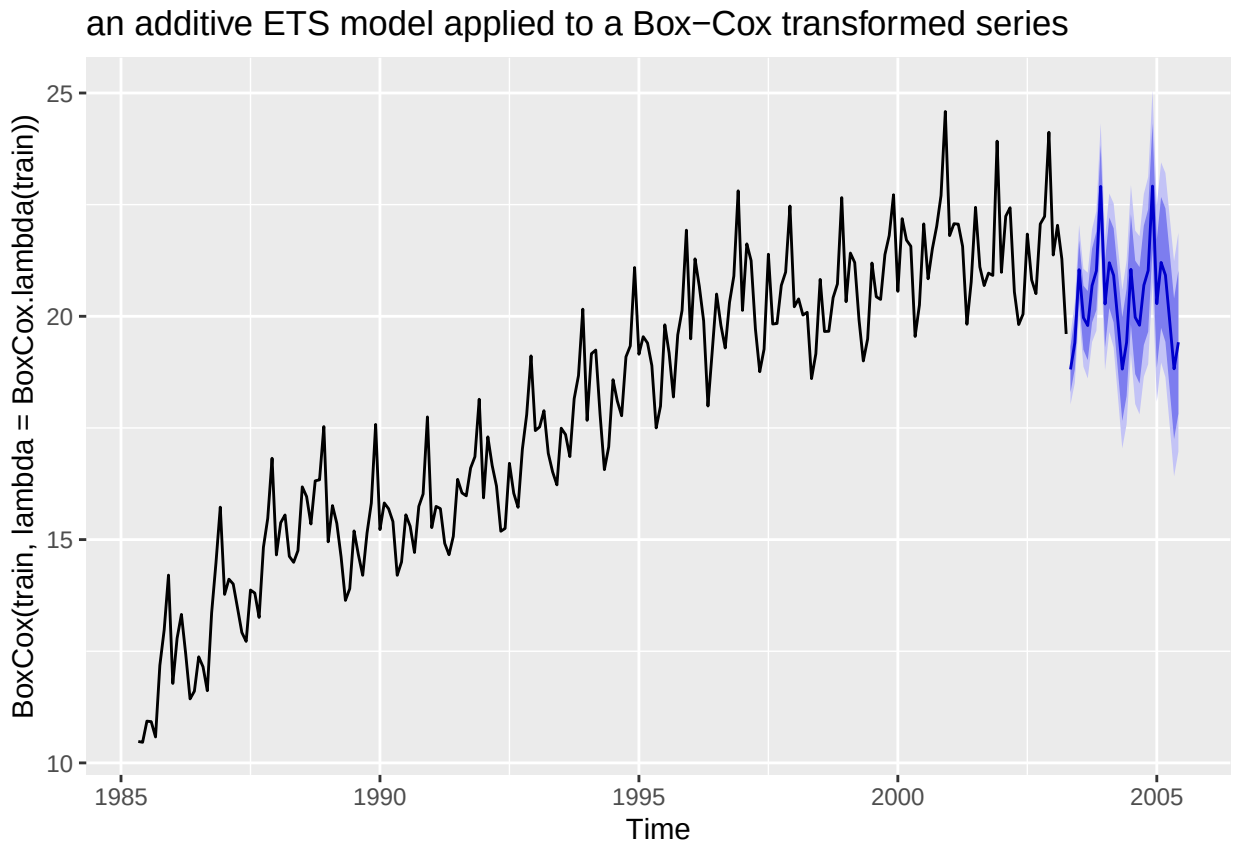
2.4. Forecast the two-year test set using each of the following methods: (8 points)

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

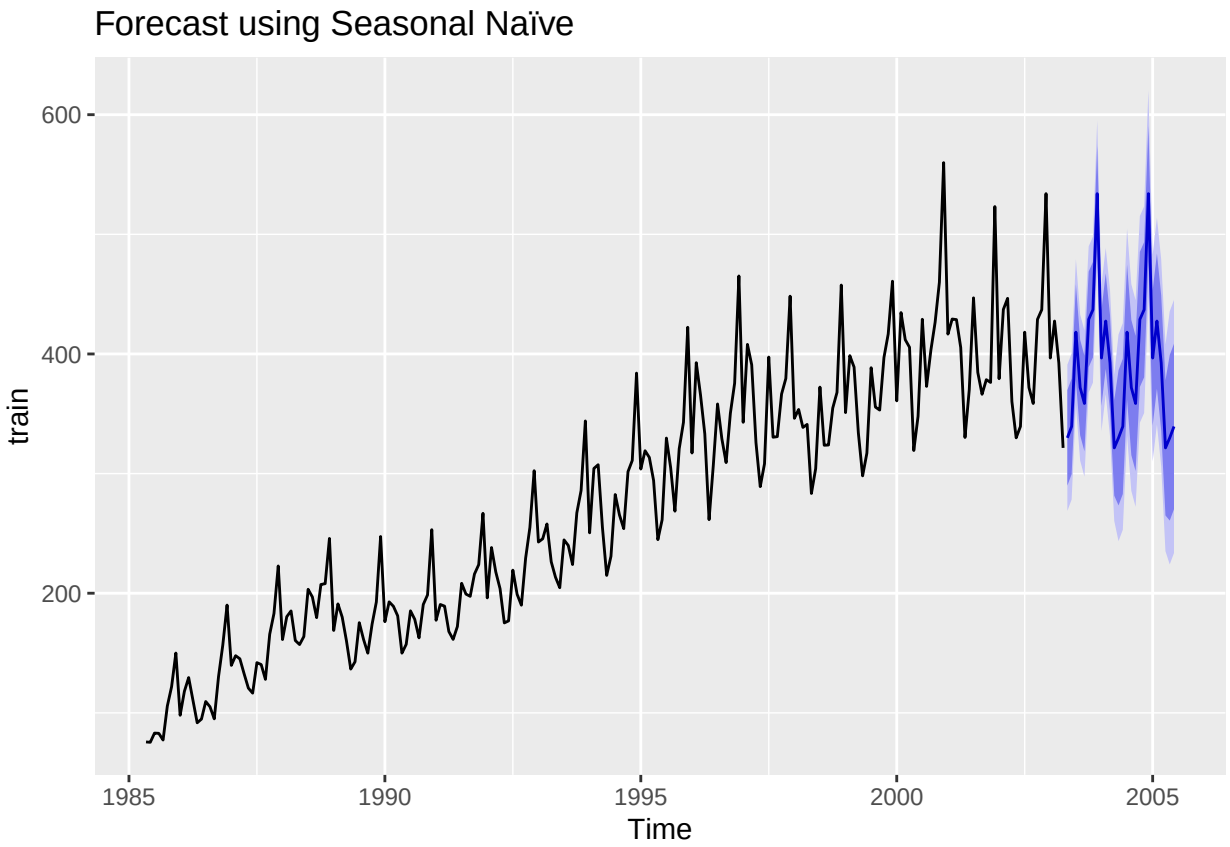
```
#I. an ETS model;
fit_ets1 <- ets(train)
forecast_ets <- forecast(fit_ets1, h = length(test))
autoplot(forecast_ets, main = "Forecast using ETS")
```




```
#
forecast_ets_box <- forecast(ets(BoxCox(train, lambda = BoxCox.lambda(train)), model = "AAA"), h = leng
autoplot(forecast_ets_box, main = "an additive ETS model applied to a Box-Cox transformed series")
```




```
#  III. a seasonal naïve method;
fit_seasonal_naive <- snaive(train,h = length(test))

autoplot(fit_seasonal_naive, main = "Forecast using Seasonal Naïve")
```

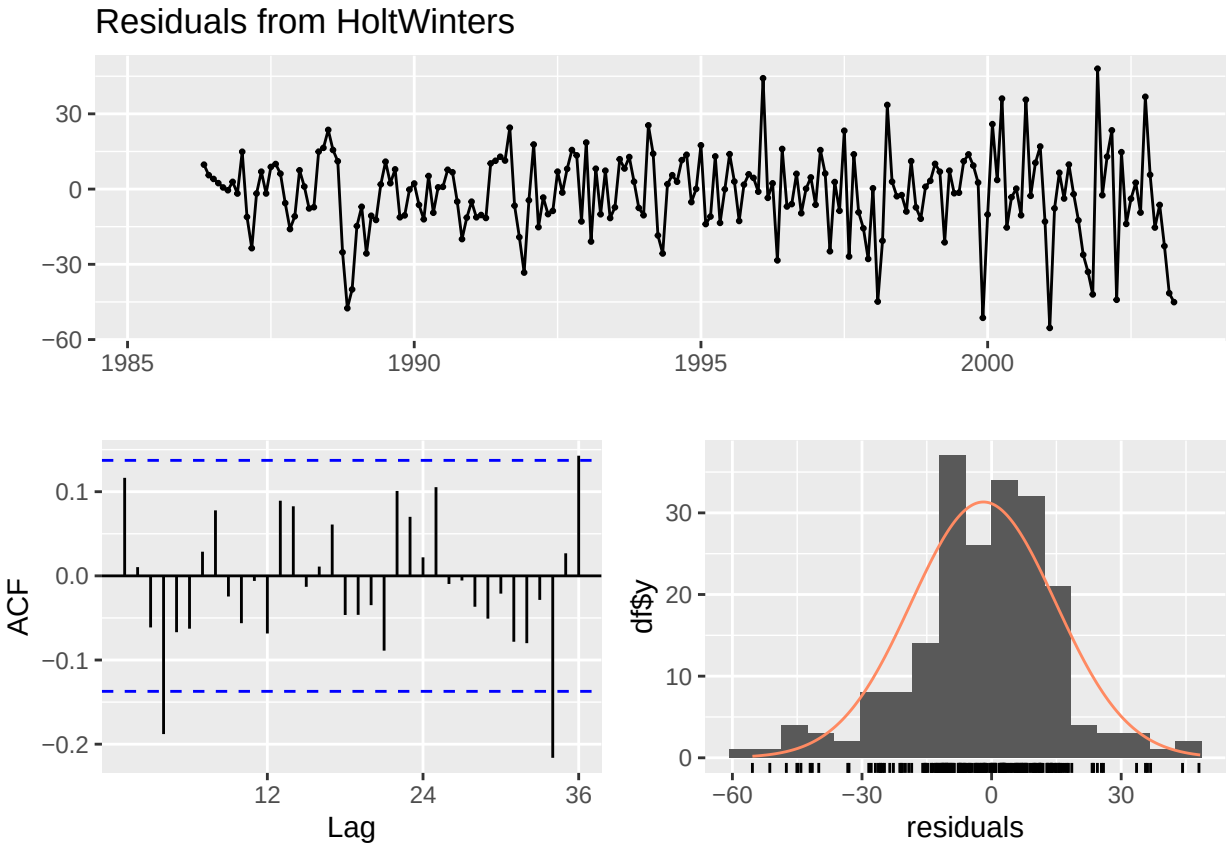


2.5. Which method gives the best forecasts? Does it pass the residual tests? (6 points) #By comparing the models I coild see ETS has best forecasts. Also they have passed residual test.

```
rmse(HoltWinters_forecast$mean,test)
```

```
## [1] 62.67992
```

```
checkresiduals(HoltWinters_forecast)
```

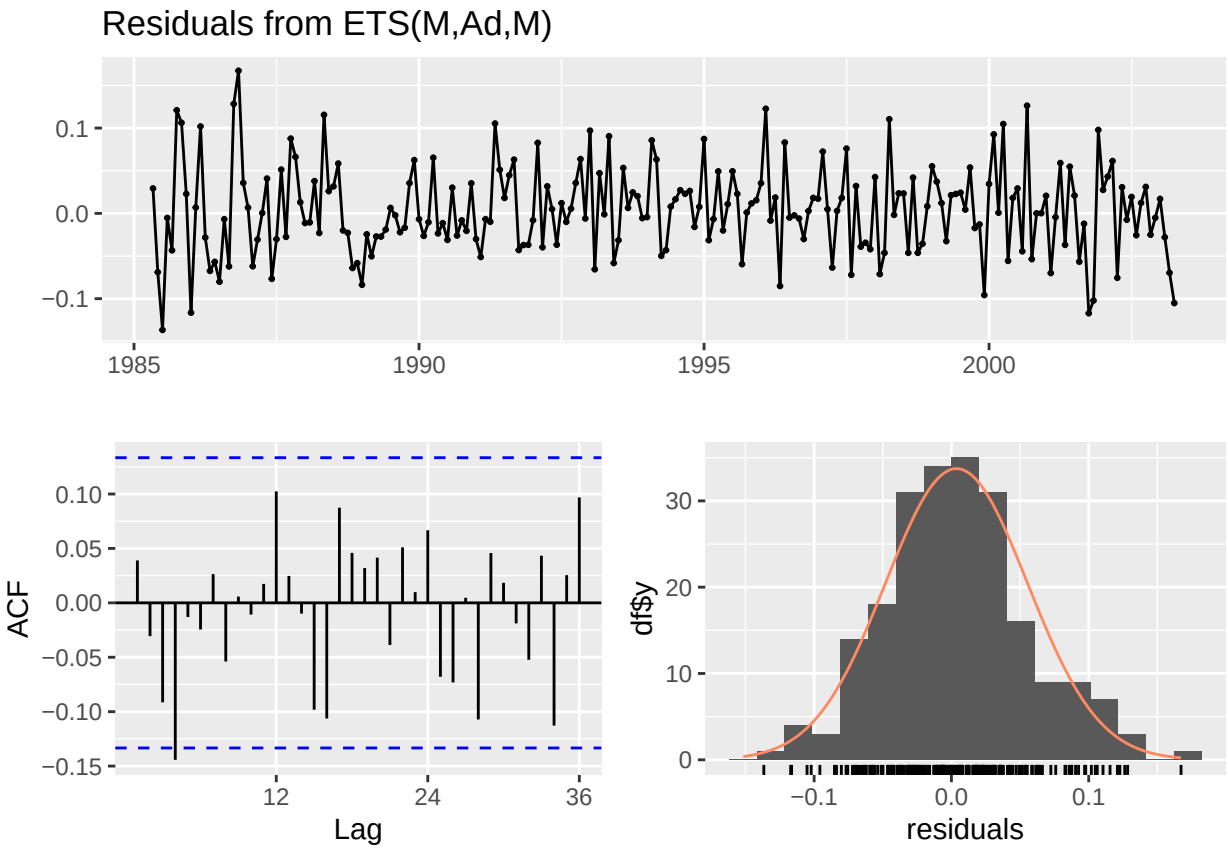



```
##
##  Ljung-Box test
##
## data:  Residuals from HoltWinters
## Q* = 26.984, df = 24, p-value = 0.3052
##
## Model df: 0.   Total lags used: 24
```

```
rmse(forecast_ets$mean,test)
```

```
## [1] 80.23124
```

```
checkresiduals(forecast_ets)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,Ad,M)
## Q* = 20.677, df = 24, p-value = 0.6577
##
## Model df: 0.   Total lags used: 24
```

```
rmse(fit_seasonal_naive$mean,test)
```

```
## [1] 50.30097
```

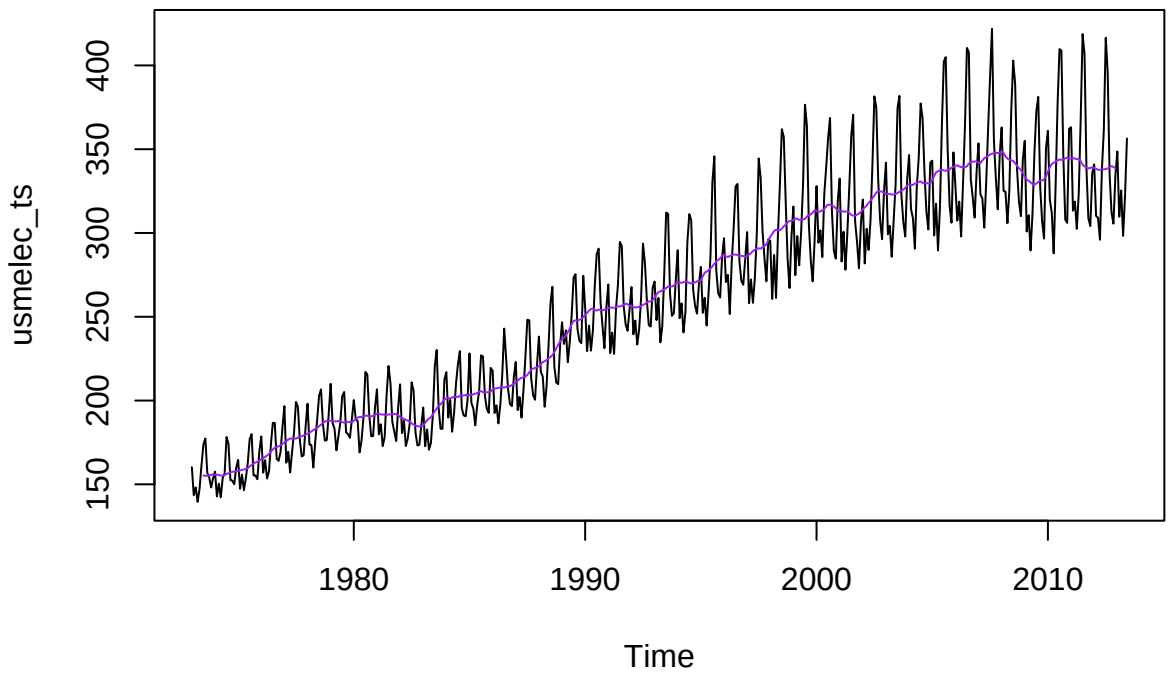
Question 3

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) #I could oberserve general upward trend with time.

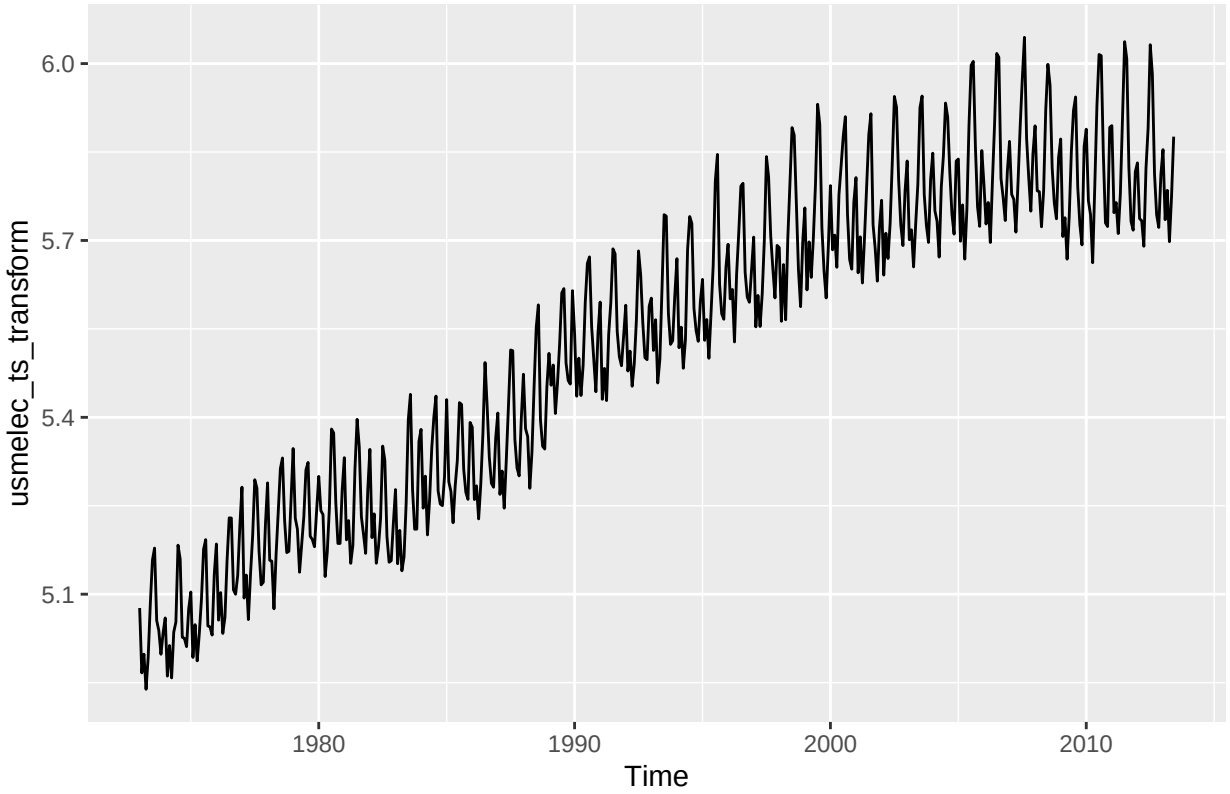
```
usmelec_df <- fread("usmelec.csv") %>%
  as.data.frame()
```

```
usmelec_ts <- ts(usmelec_df$value, start = c(1973, 1), end = c(2013,6), frequency = 12)
plot(usmelec_ts, col="black")
lines(ma(usmelec_ts,order = 12),col="purple")
```



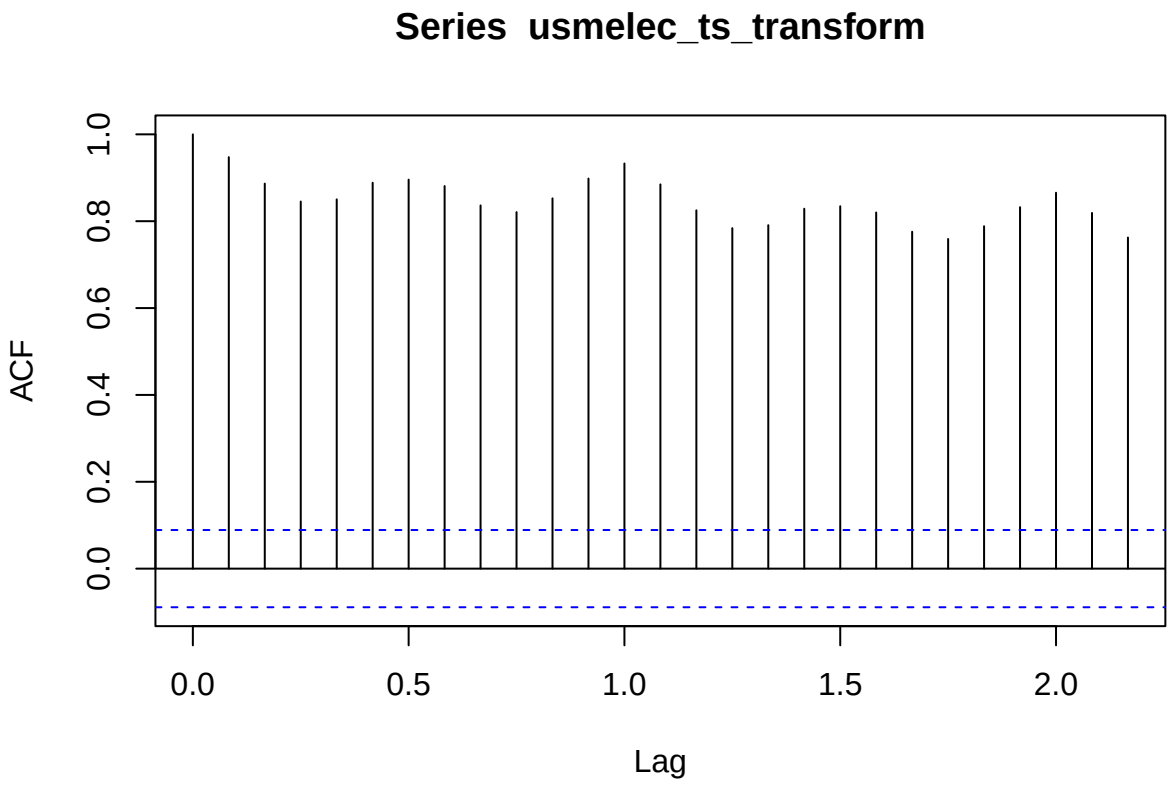
3.2 Do the data need transforming? If so, find a suitable transformation. (4 points) #Yes we need transformation, below is the transformed data. I did log transformation said by professor in lecture.

```
usmelec_ts_transform <- log(usmelec_ts)
autoplot(usmelec_ts_transform)
```

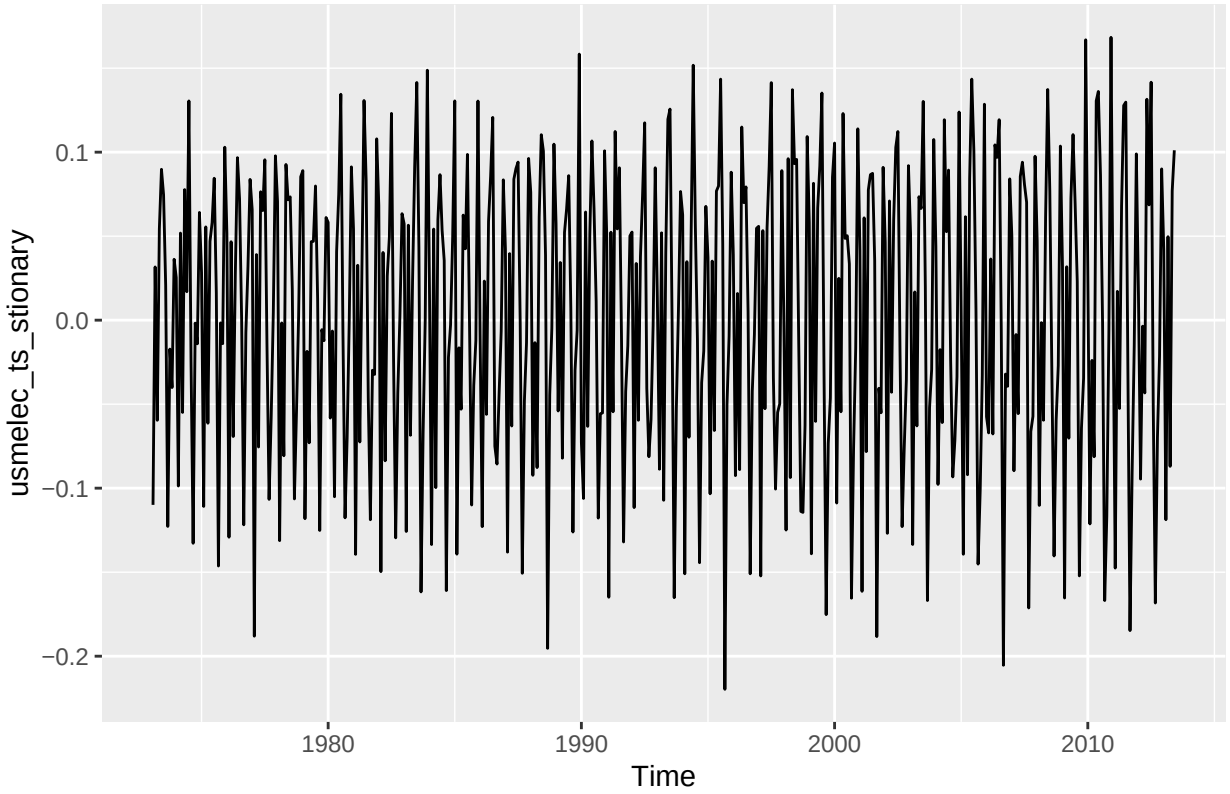


3.3 Are the data stationary? If not, find an appropriate differencing which yields stationary data. (4 points)
#The data is not stationary

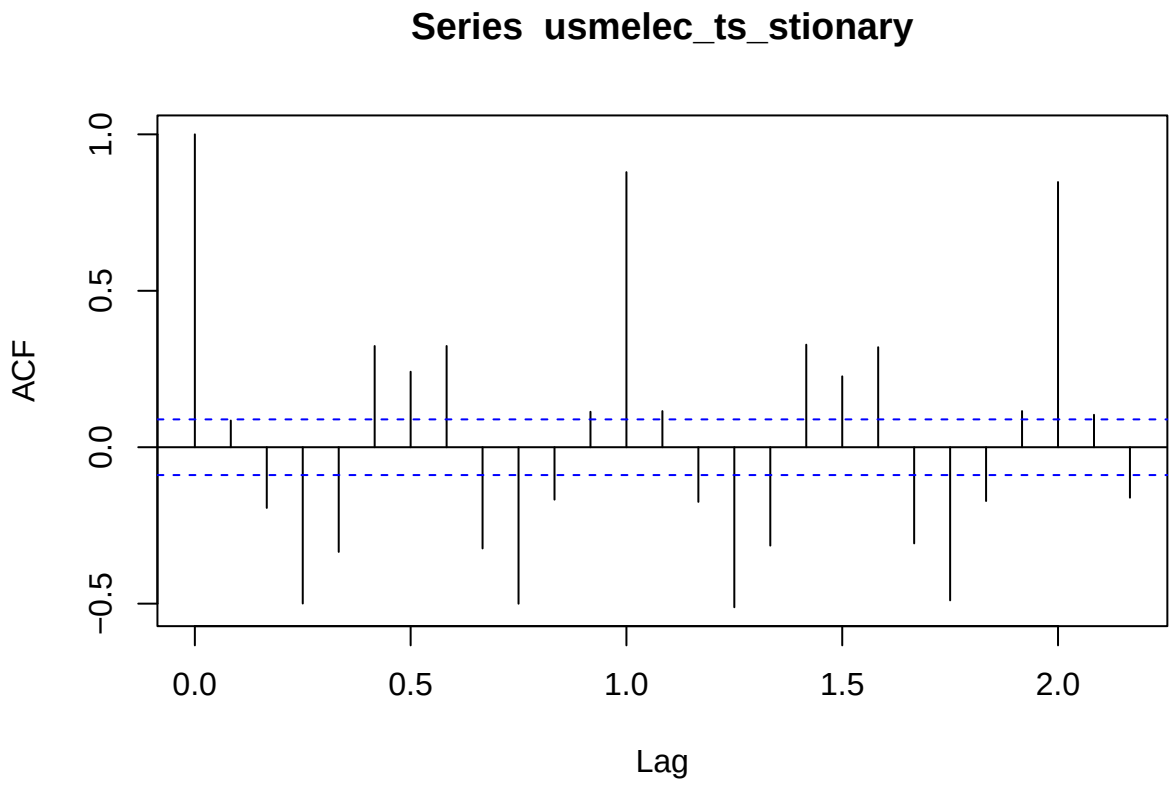
```
acf(usmelec_ts_transform)
```




```
usmelec_ts_stionary <- diff(usmelec_ts_transform)
autoplot(usmelec_ts_stionary)
```




```
acf(usmelec_ts_stionary)
```



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) #As per AIC values auto.arima (AIC=-2081.69) is best.

```
model1 <- Arima(usmelec_ts_stionary, order=c(4, 1, 0))
summary(model1)
```

```
## Series: usmelec_ts_stionary
## ARIMA(4,1,0)
##
## Coefficients:
##           ar1      ar2      ar3      ar4
##       -0.6510  -0.4148  -0.6149  -0.6471
## s.e.   0.0347   0.0360   0.0357   0.0347
##
## sigma^2 = 0.005968:  log likelihood = 553.27
## AIC=-1096.54   AICc=-1096.42   BIC=-1075.63
##
## Training set error measures:
##                       ME       RMSE        MAE      MPE     MAPE     MASE
## Training set 0.0004875977 0.07685262 0.06077432 145.9185 283.7756 2.019398
##                    ACF1
## Training set -0.1268654
```

```r
model2 <- Arima(usmelec_ts_stionary, order = c(2, 1, 0))
summary(model2)
```

```
## Series: usmelec_ts_stionary
## ARIMA(2,1,0)
##
## Coefficients:
##           ar1      ar2
##       -0.3872  -0.1188
## s.e.   0.0452   0.0452
##
## sigma^2 = 0.01154:  log likelihood = 393.91
## AIC=-781.81   AICc=-781.76   BIC=-769.26
##
## Training set error measures:
##                        ME      RMSE        MAE      MPE     MAPE  MASE
## Training set 0.0005584085 0.1071007 0.08835969 492.3506 692.0716 2.936
##                      ACF1
## Training set -0.04008904
```

```r
model3 <- Arima(usmelec_ts_stionary, order = c(0, 1, 1))
summary(model3)
```

```
## Series: usmelec_ts_stionary
## ARIMA(0,1,1)
##
## Coefficients:
##           ma1
##       -1.0000
## s.e.   0.0052
##
## sigma^2 = 0.007281:  log likelihood = 501.88
## AIC=-999.75   AICc=-999.73   BIC=-991.39
##
## Training set error measures:
##                         ME       RMSE       MAE      MPE     MAPE     MASE
## Training set 0.0005114272 0.08515432 0.0723278 98.41782 108.6833 2.403295
##                     ACF1
## Training set 0.08804448
```

```r
auto_model <- auto.arima(usmelec_ts_stionary)
summary(auto_model)
```

```
## Series: usmelec_ts_stionary
## ARIMA(1,0,1)(2,1,1)[12] with drift
##
## Coefficients:
##          ar1      ma1    sar1     sar2     sma1  drift
##       0.4196  -0.8569  0.0119  -0.1014  -0.8257      0
## s.e.  0.0619   0.0348  0.0565   0.0531   0.0359      0
##
## sigma^2 = 0.0006822:  log likelihood = 1047.85
```

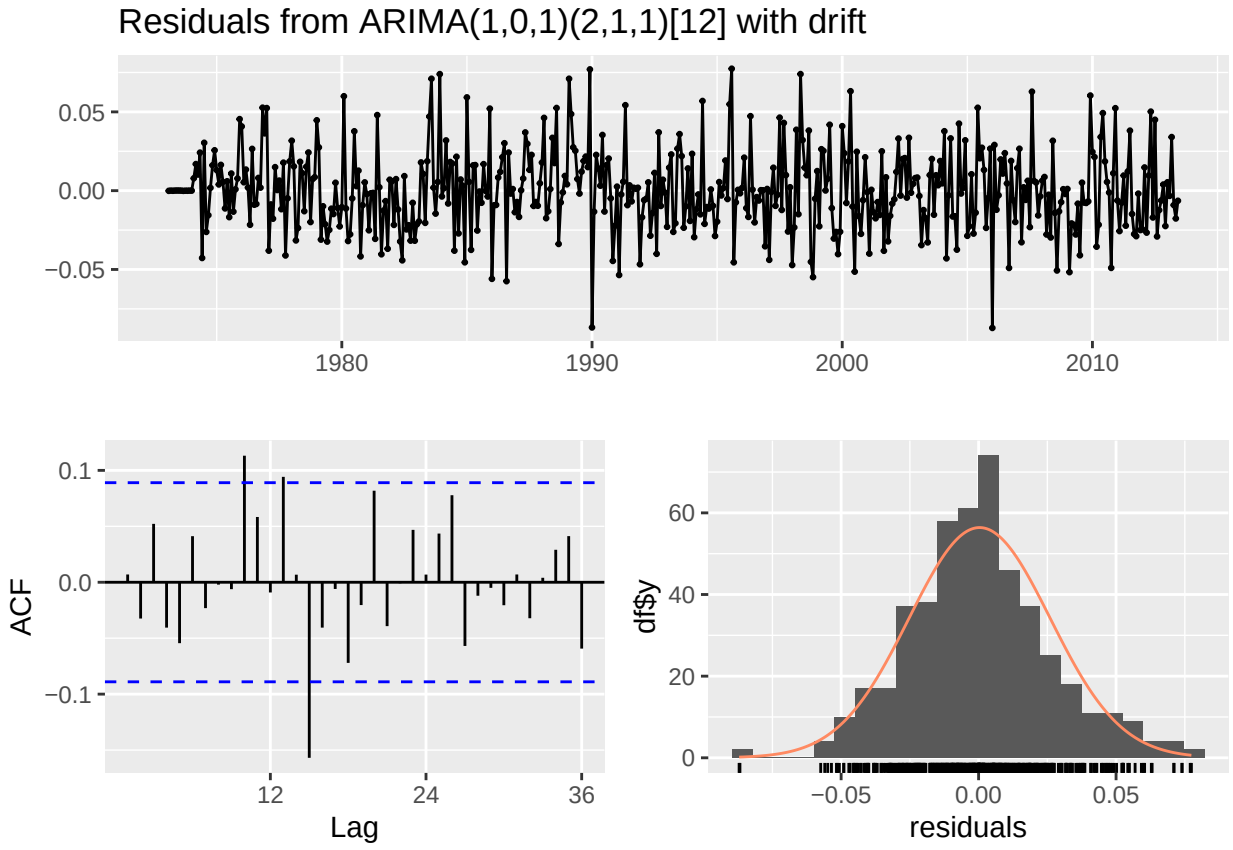
```
## AIC=-2081.69   AICc=-2081.45   BIC=-2052.58
##
## Training set error measures:
##                       ME       RMSE        MAE       MPE     MAPE      MASE
## Training set 0.0003631124 0.02563002 0.01952966 -7.924451 106.9543 0.6489281
##                     ACF1
## Training set 0.006946614
```

```
model5 <- Arima(usmelec_ts_stionary, order = c(2, 0, 2))
summary(model5)
```

```
## Series: usmelec_ts_stionary
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1     ma2    mean
##       1.0436  -0.6293  -1.3345  0.3923  0.0016
## s.e.  0.0686   0.0417   0.0870  0.0826  0.0003
##
## sigma^2 = 0.004448:  log likelihood = 626.3
## AIC=-1240.6   AICc=-1240.42   BIC=-1215.49
##
## Training set error measures:
##                          ME      RMSE        MAE      MPE     MAPE     MASE
## Training set -0.0001097452 0.0663514 0.05287293 6.665012 217.8532 1.756852
##                     ACF1
## Training set -0.03883998
```

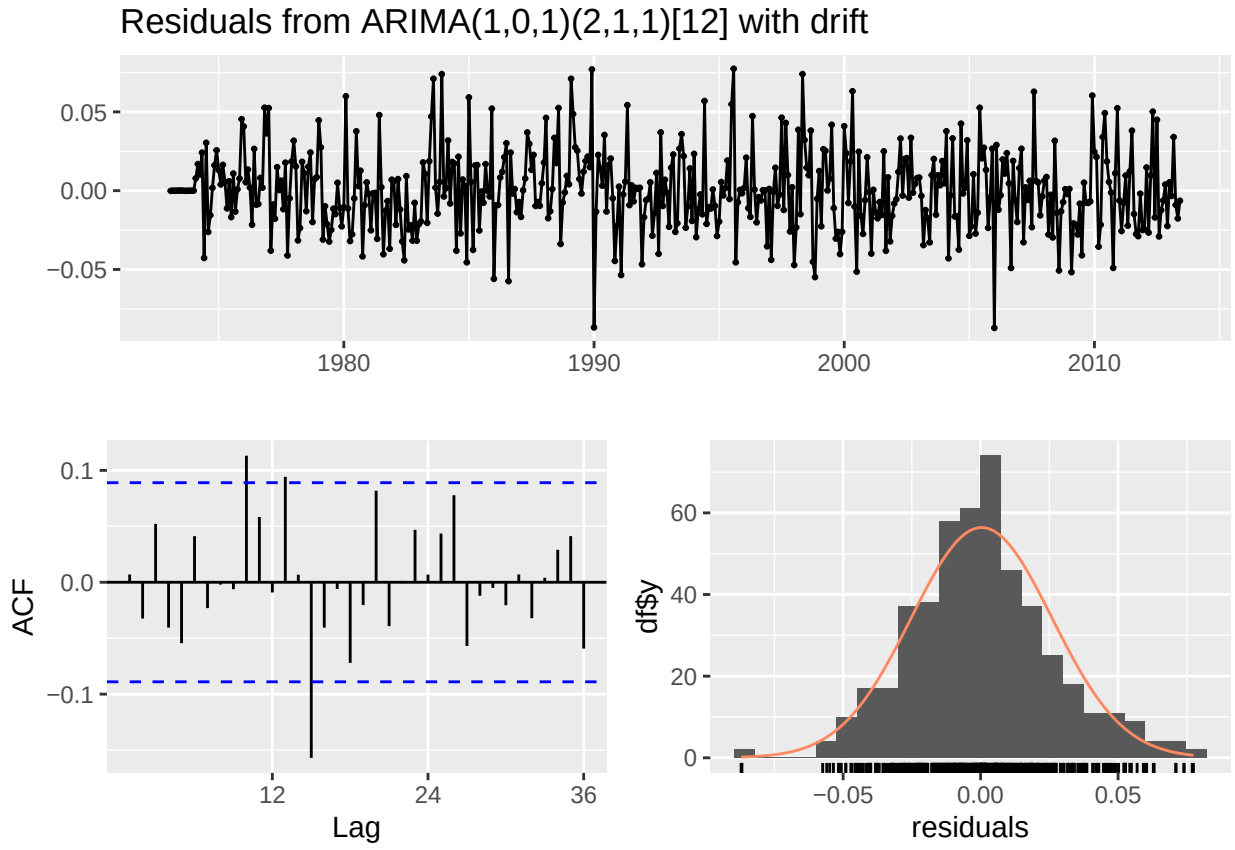
3.5 Estimate the parameters of your best model and do diagnostic testing on the residuals. Do the residuals resemble white noise? If not, try to find another ARIMA model which fits better. (4 points) #I could see that p-value = 0.004149 which means it doesn'y resemeble white noise. We will find the better model by adding seasonal adjustment. After I added the seasonal adjustm,en I could see ACF1 is almost zero which means that is no more has white noise.

```
best_model <- auto_model
checkresiduals(best_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1)(2,1,1)[12] with drift
## Q* = 39.21, df = 19, p-value = 0.004149
##
## Model df: 5.   Total lags used: 24
```

```
final_model <- Arima(usmelec_ts, order = c(1, 0, 1), seasonal = c(2, 1, 1), include.drift = TRUE)
final_model1 <- Arima(usmelec_ts_stionary, order = c(1, 0, 1), seasonal = c(2, 1, 1), include.drift = TR
checkresiduals(final_model1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1)(2,1,1)[12] with drift
## Q* = 39.21, df = 19, p-value = 0.004149
##
## Model df: 5.   Total lags used: 24
```

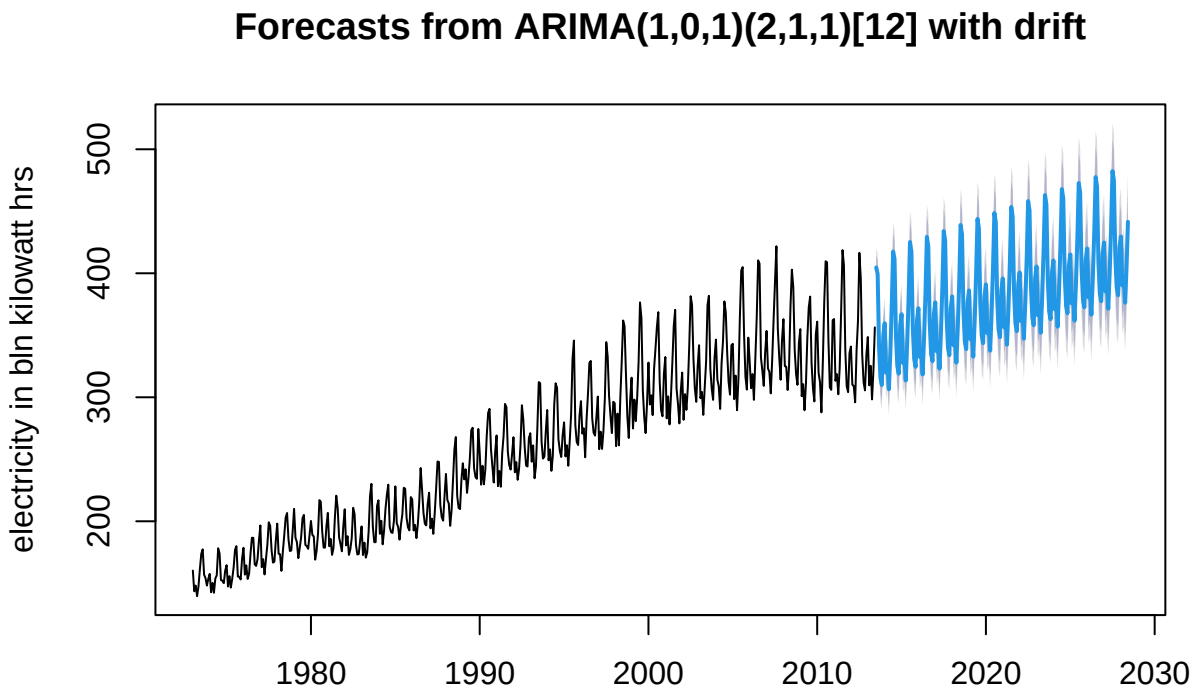
```
summary(final_model1)
```

```
## Series: usmelec_ts_stionary
## ARIMA(1,0,1)(2,1,1)[12] with drift
##
## Coefficients:
##          ar1      ma1    sar1     sar2     sma1  drift
##       0.4196  -0.8569  0.0119  -0.1014  -0.8257      0
## s.e.  0.0619   0.0348  0.0565   0.0531   0.0359      0
##
## sigma^2 = 0.0006822:  log likelihood = 1047.85
## AIC=-2081.69   AICc=-2081.45   BIC=-2052.58
##
## Training set error measures:
##                       ME       RMSE        MAE       MPE     MAPE      MASE
## Training set 0.0003631124 0.02563002 0.01952966 -7.924451 106.9543 0.6489281
##                      ACF1
```

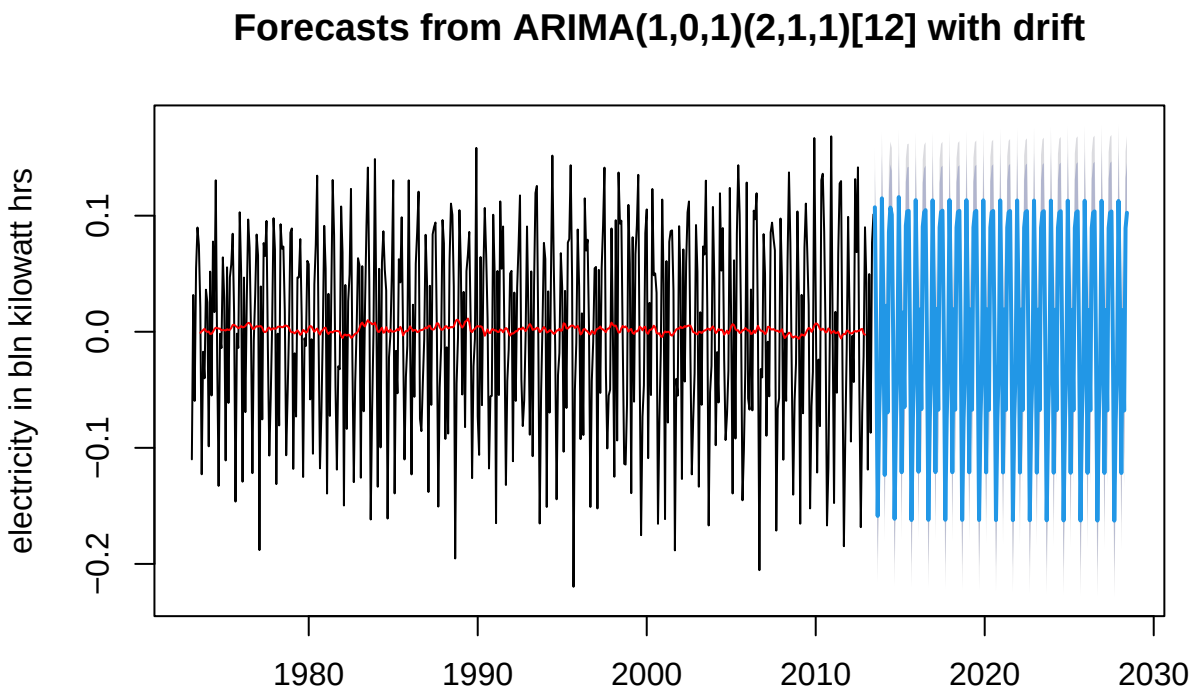
```
## Training set 0.006946614
```

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)

```
plot(forecast(final_model,h =180), ylab = "electricity in bln kilowatt hrs")
lines(ma(usmelec_ts_stionary, 12), col = "red")
```




```
plot(forecast(final_model1,h =180), ylab = "electricity in bln kilowatt hrs")
lines(ma(usmelec_ts_stionary, 12), col = "red")
```



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)

```
#I feel that forecast usually lose their practicality when the range increase more than 5 years. In our

#There can be several reasons for having the expansion f prediction intervals. One can be non-stationar
#Also more the complex is the model, more is the chance of wider prediction intervals, this may happen
# Also its important to transform the data this reduces the width.

#In our case wide prediction intervals are due to the seasonality and complexity of the data.
```