Assignment 1

Tom van de Looij - Mark Swaringen - Younes Moustaghfir, Group 30

13th of September

```r
library(fpp2)
```

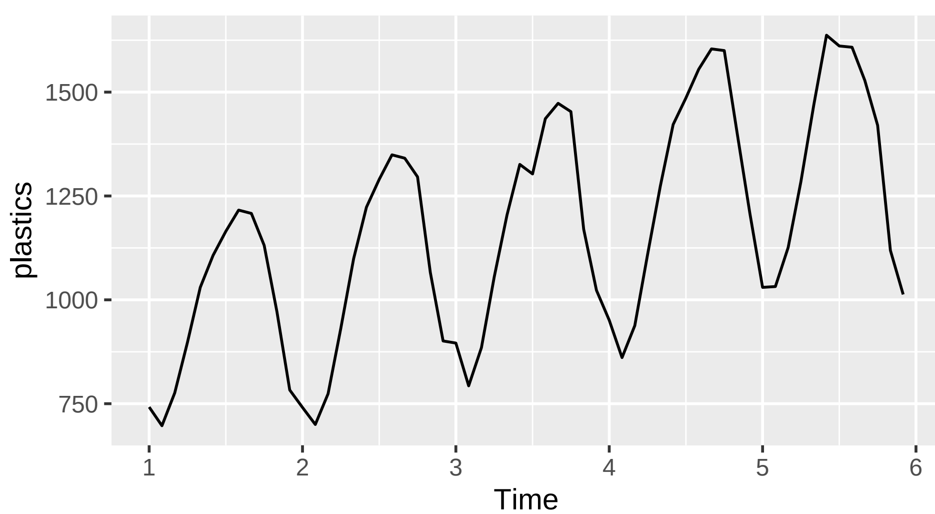
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## -- Attaching packages --------------------------------------------- fpp2 2.4 --

## v ggplot2   3.3.2     v fma       2.4
## v forecast  8.13      v expsmooth 2.3

##
```

```r
library(xlsx)
library(seasonal)
```

Exercise 1.3: Time series decomposition

a.1)

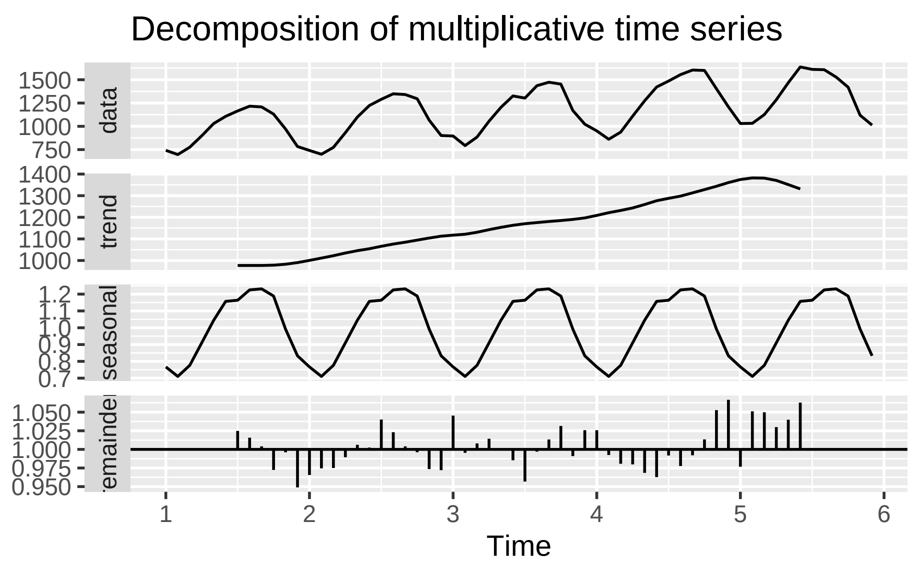
```r
autoplot(plastics)
```



There seems to be an upwards trend when looking at the graph created by the autoplot function. Next to that, we see seasonal fluctuations as well.

a.2)

```r
decomp <- decompose(plastics, type = "multiplicative")
autoplot(decomp)
```



Yes, we see indeed that the decomposition shows a clear upwards line in the trend graph. Next to that, we see clear a clear seasonal pattern as well

a.3)

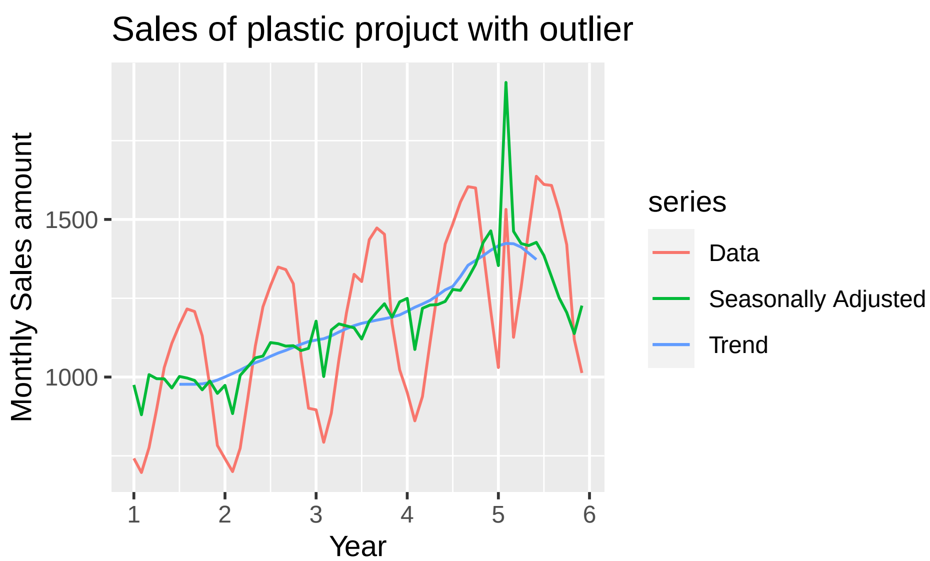
```r
autoplot(plastics, series="Data") +
  autolayer(trendcycle(decomp), series="Trend") +
  autolayer(seasadj(decomp), series="Seasonally Adjusted") +
  xlab("Year") + ylab("Monthly Sales amount")
```

```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```

a.4)

```r
plastics_new <- plastics
plastics_new[50] <- plastics_new[50] + 500
decomp_new <- decompose(plastics_new, type = "multiplicative")

autoplot(plastics_new, series = "Data") +
  autolayer(trendcycle(decomp_new),
            series = "Trend") +
  autolayer(seasadj(decomp_new),
            series = "Seasonally Adjusted") +
  xlab("Year") +
  ylab("Monthly Sales amount") +
  ggtitle("Sales of plastic projuct with outlier")
```

```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```



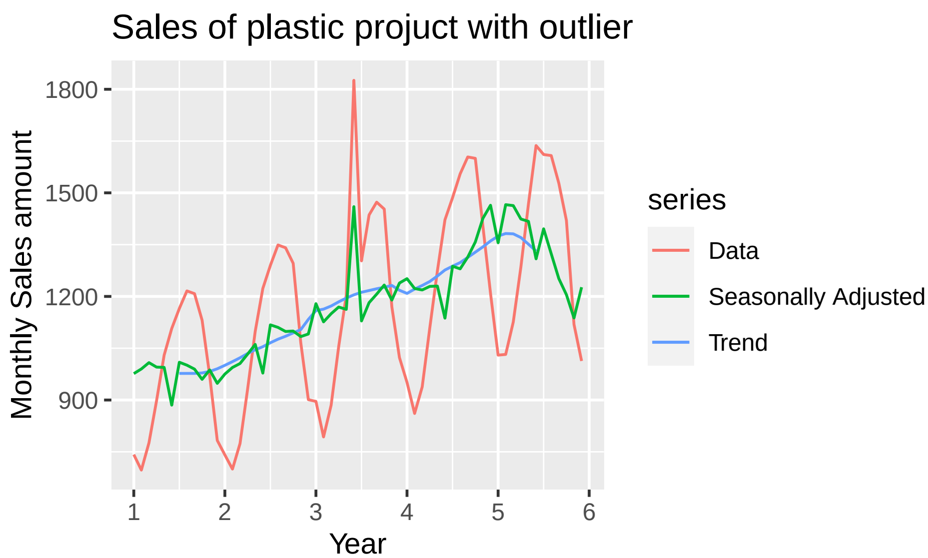
The outlier affects the seasonality, creating a new peak and breaking the seasonality cycle at the end. For the trend, it does not do much damage, because the trend seems pretty much the same except for the outlier value(still an upwards trend).

a.5)

```r
plastics_new <- plastics
plastics_new[30] <- plastics_new[30] + 500
decomp_new <- decompose(plastics_new, type = "multiplicative")

autoplot(plastics_new, series = "Data") +
  autolayer(trendcycle(decomp_new),
            series = "Trend") +
  autolayer(seasadj(decomp_new),
            series = "Seasonally Adjusted") +
  xlab("Year") +
  ylab("Monthly Sales amount") +
```

```r
  ggtitle("Sales of plastic projuct with outlier")
```

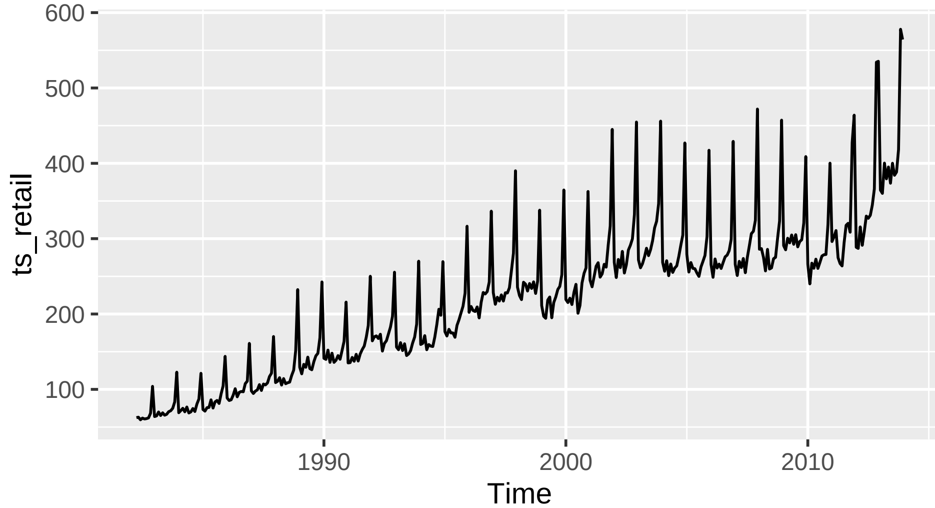
```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```



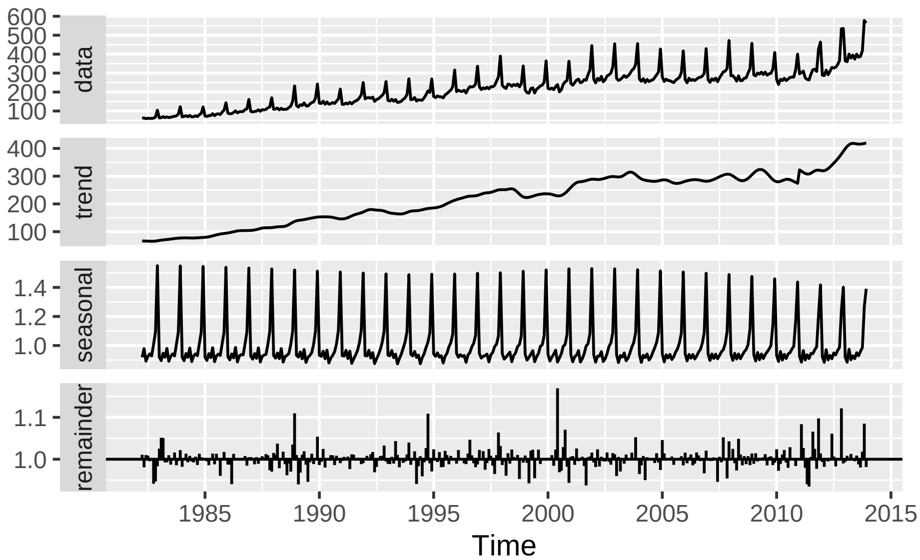
In the middle, the outlier seems to have a bigger effect on the seasonality, but the trend remains pretty much upwards.

b)

```r
retail <- read.xlsx("retail.xlsx",
                    sheetIndex = 1,
                    startRow = 2)


ts_retail <- ts(retail[,"A3349873A"],
                frequency=12,
                start=c(1982,4))

autoplot(ts_retail)
```



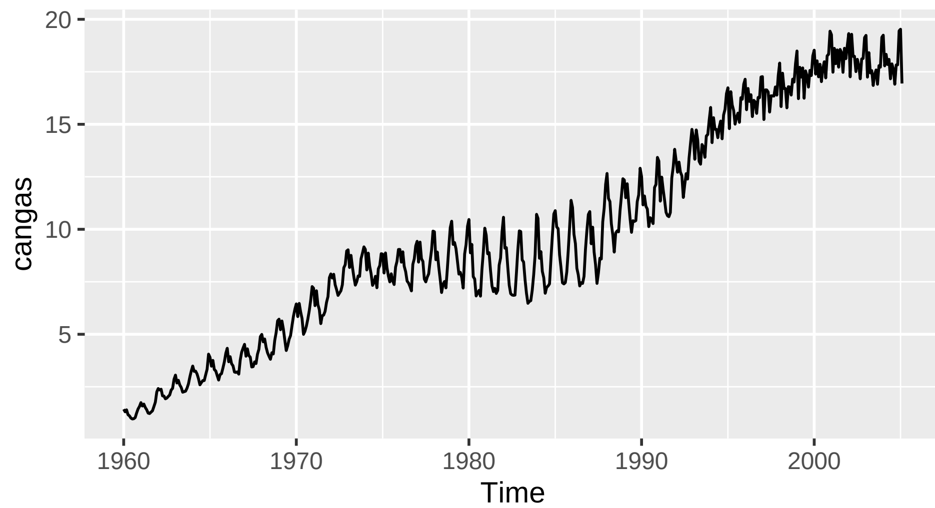

```r
x11_retail <- seas(ts_retail, x11 = "")
autoplot(x11_retail)
```



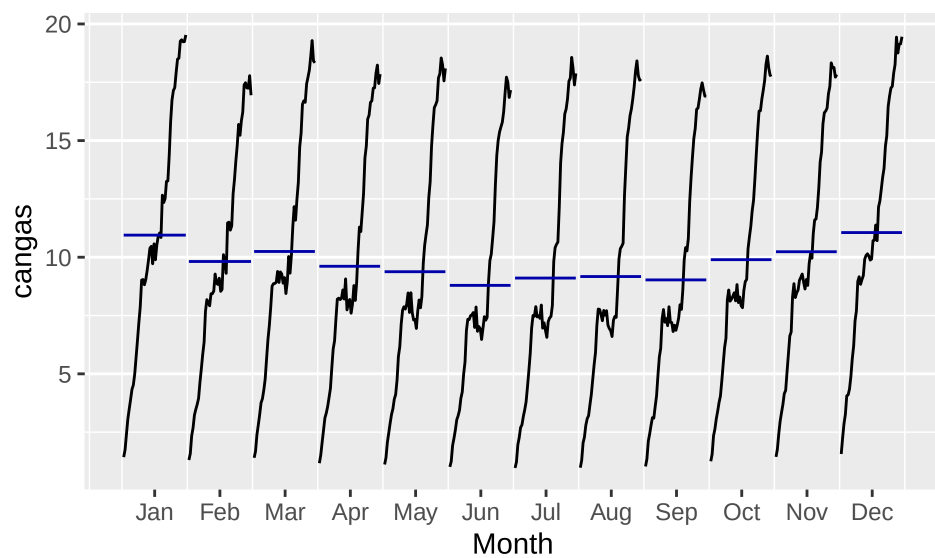
It seems to show quite an outlier just past 2000. We see some other outliers around 1995 and 1989 as well. Next to that, it seems that the seasonality slighly decreases over time.

c.1)

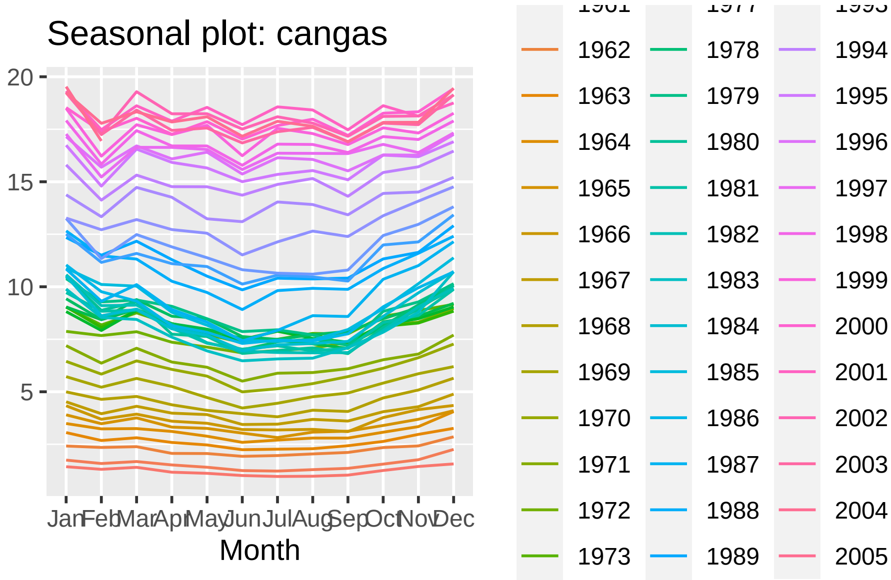
```r
autoplot(cangas)
```




```
ggsubseriesplot(cangas)
```



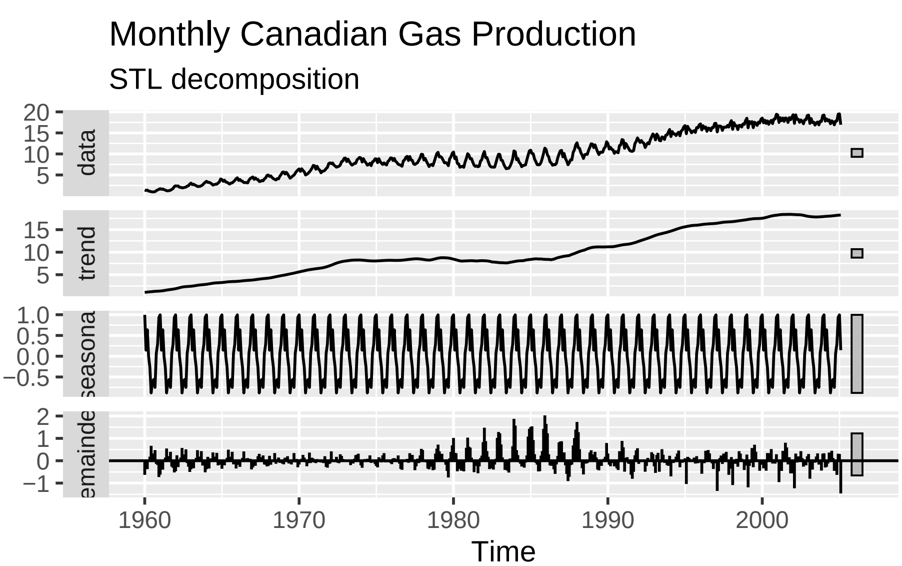

```
ggseasonplot(cangas)
```



There seems to be a slight decrease in February and the summer month, but an increase around the winter time again. This could be due to increase in gas demand at that point, due to the cold for example

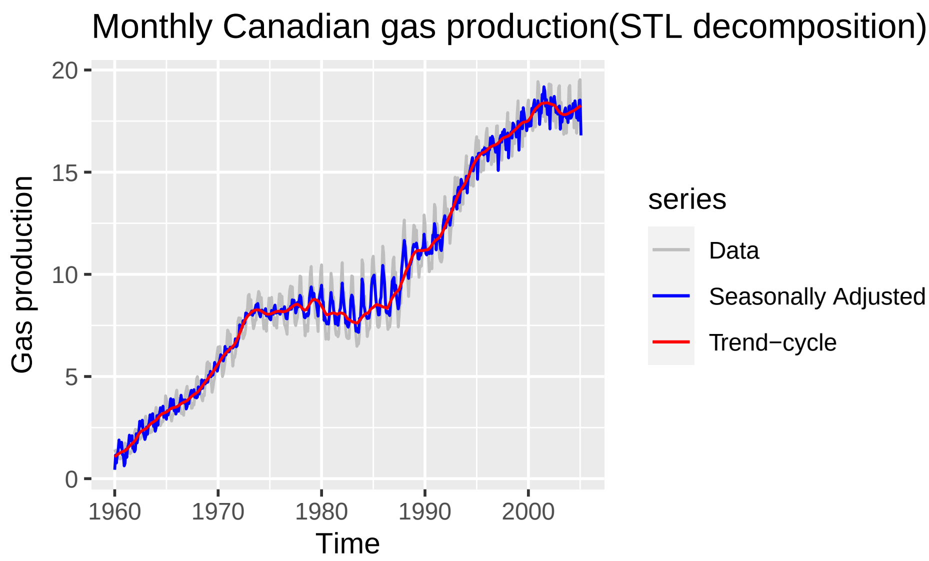
c.2)

```
stl_cangas <- stl(cangas, s.window = "periodic", robust = TRUE)

autoplot(stl_cangas) +
  ggtitle("Monthly Canadian Gas Production",
          subtitle = "STL decomposition")
```



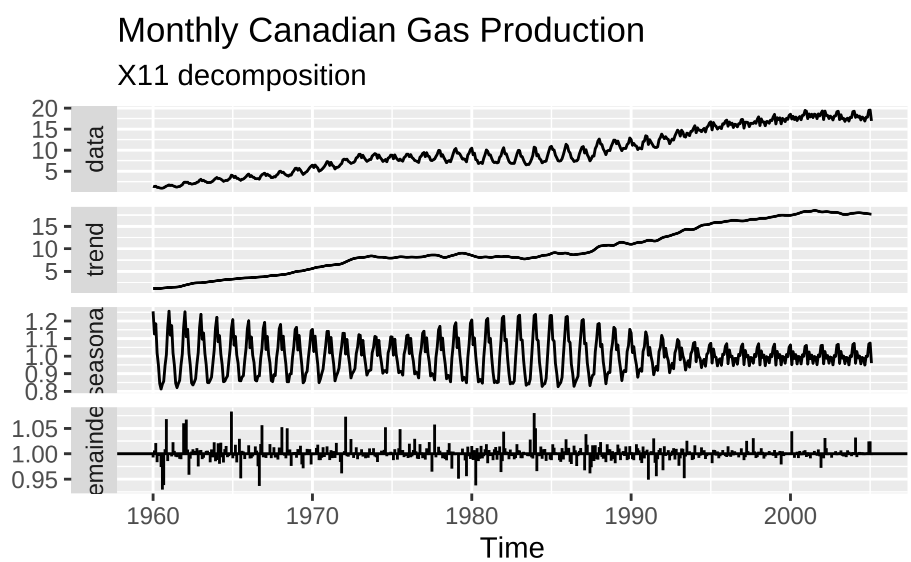

```
autoplot(cangas, series = "Data") +
  autolayer(seasadj(stl_cangas), series = "Seasonally Adjusted") +
  autolayer(trendcycle(stl_cangas), series = "Trend-cycle") +
  ggtitle("Monthly Canadian gas production(STL decomposition)") +
  ylab(expression(paste("Gas production"))) +
```

```
  scale_color_manual(values = c("gray", "blue", "red"),
                     breaks = c("Data", "Seasonally Adjusted", "Trend-cycle"))
```



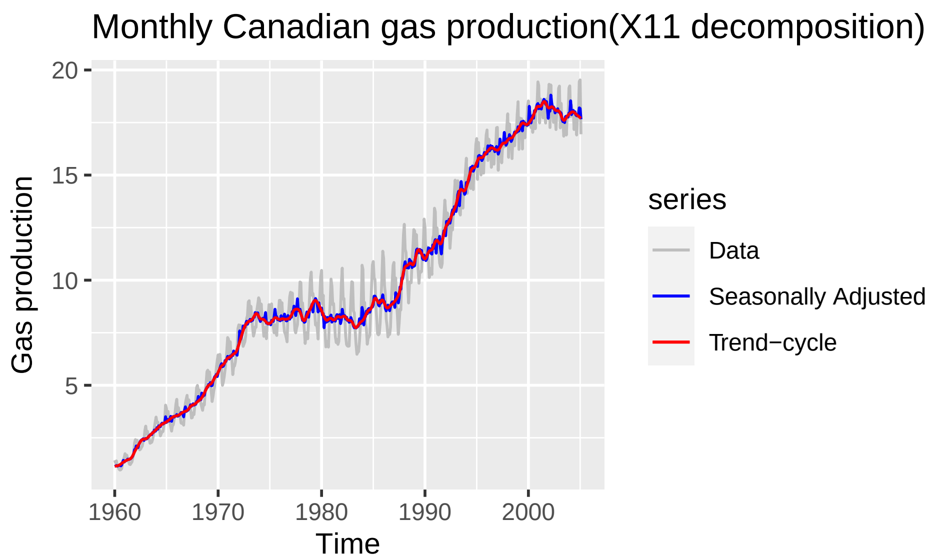
c.3)

```
x11_cangas <- seas(cangas, x11 = "")
seats_cangas <- seas(cangas)


autoplot(x11_cangas) +
  ggtitle("Monthly Canadian Gas Production",
          subtitle = "X11 decomposition")
```

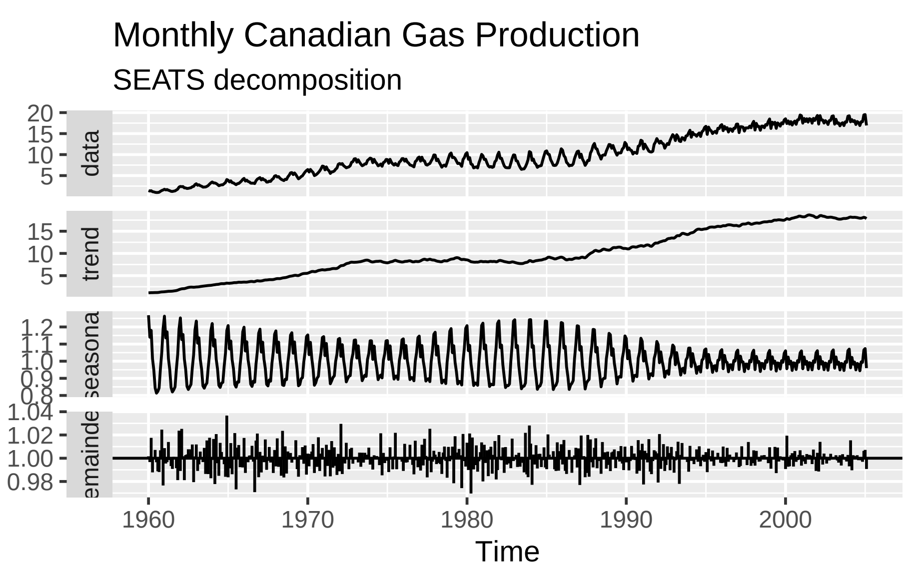



```
autoplot(cangas, series = "Data") +
  autolayer(seasadj(x11_cangas), series = "Seasonally Adjusted") +
  autolayer(trendcycle(x11_cangas), series = "Trend-cycle") +
```

```
  ggtitle("Monthly Canadian gas production(X11 decomposition)") +
  ylab(expression(paste("Gas production"))) +
  scale_color_manual(values = c("gray", "blue", "red"),
                     breaks = c("Data", "Seasonally Adjusted", "Trend-cycle"))
```

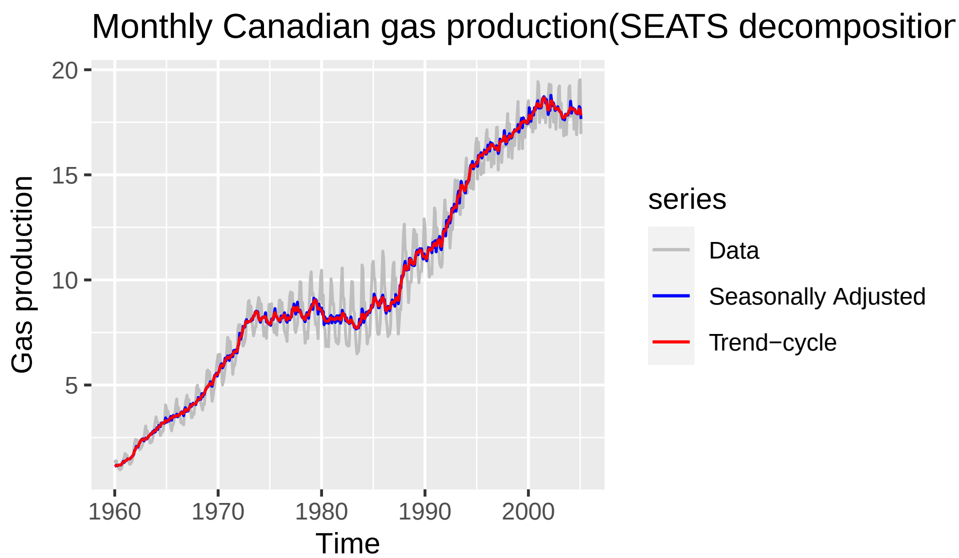



```
autoplot(seats_cangas) +
  ggtitle("Monthly Canadian Gas Production",
          subtitle = "SEATS decomposition")
```




```
autoplot(cangas, series = "Data") +
  autolayer(seasadj(seats_cangas), series = "Seasonally Adjusted") +
  autolayer(trendcycle(seats_cangas), series = "Trend-cycle") +
  ggtitle("Monthly Canadian gas production(SEATS decomposition)") +
  ylab(expression(paste("Gas production"))) +
```

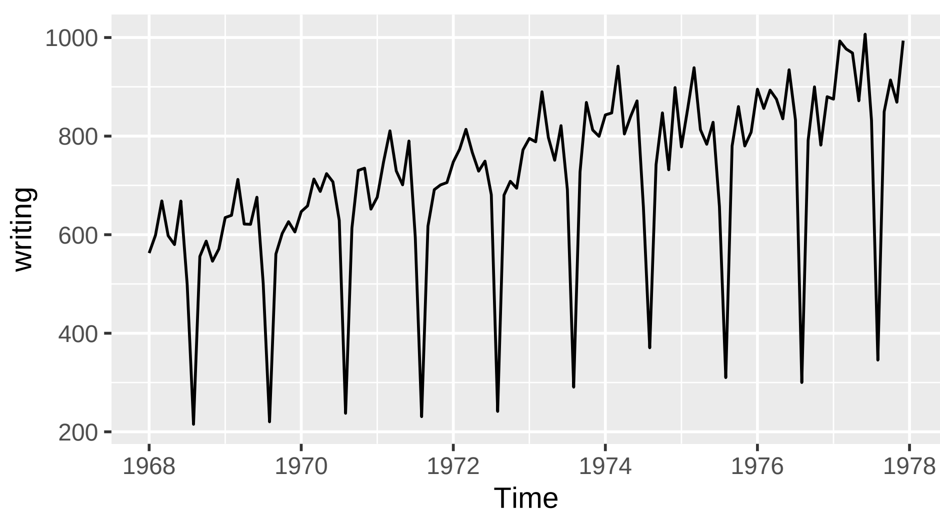
```
  scale_color_manual(values = c("gray", "blue", "red"),
                     breaks = c("Data", "Seasonally Adjusted", "Trend-cycle"))
```



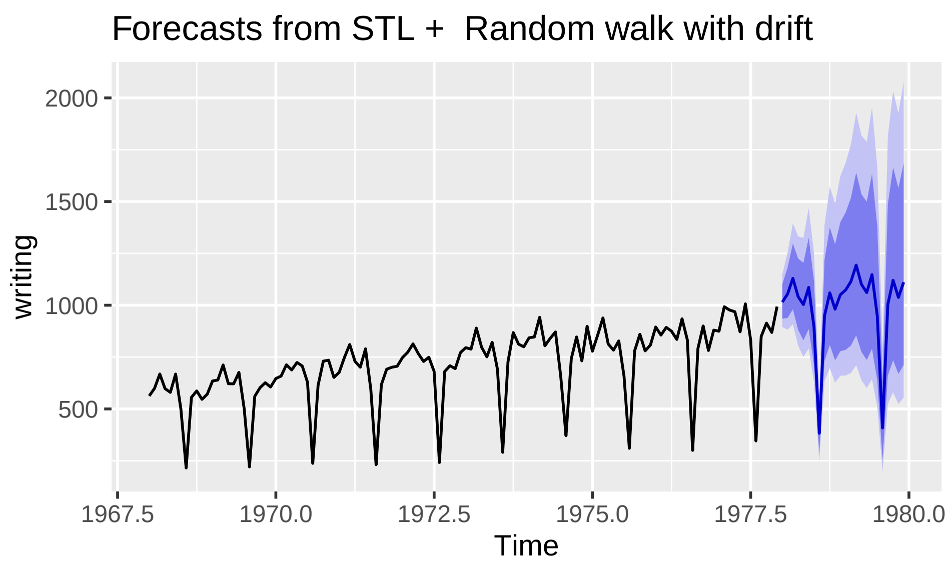
The mean for seasonal and remainder are around 1 for X11 and SEATS, for the STL we saw that to be around 0 instead.

d)

```
autoplot(writing)
```




```
stlf_writing <- stlf(writing,
                     s.window = 13,
                     robust = TRUE,
                     lambda = BoxCox.lambda(writing),
                     method = "rwdrift")
autoplot(stlf_writing)
```

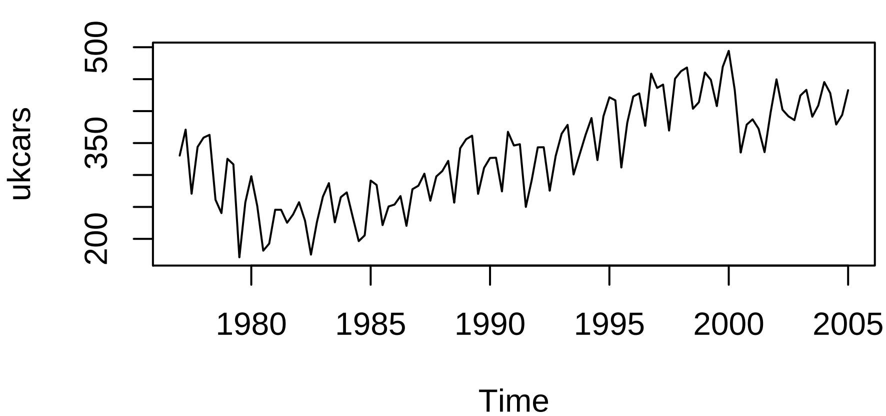


We see that there is an increasing trend in the writing data, so it would be better to use rwdrift to forecast. We applied a Box-CoX transformation with default values, in order to make the variance of the change due to seasonality equal per season.

Exercise 1.3: Exponential Smoothing

a.1)

```
plot(ukcars)
```



We clearly see seasonality within this series. We also first see a declining trend in the data up until around 1983, which is followed by a increasing trend afterwards. Around 2000 we see a slight drop.

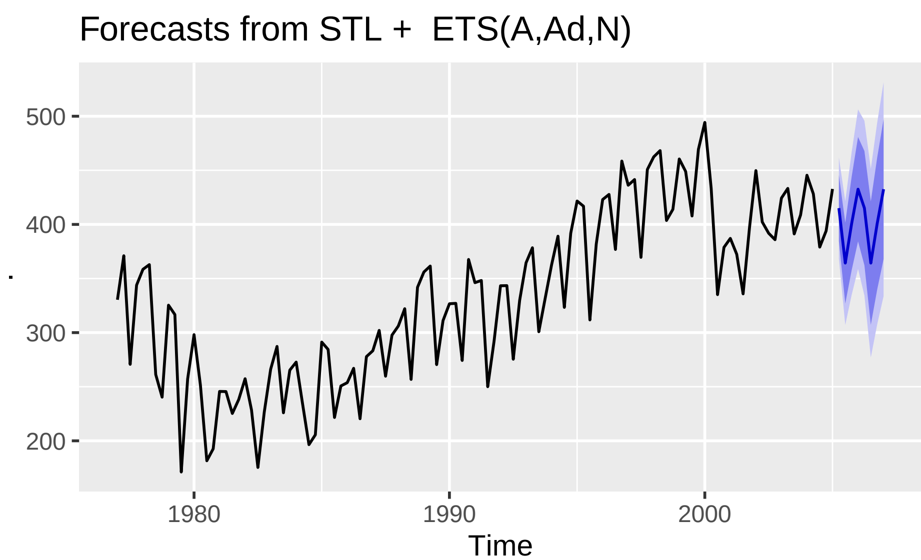
a.2)

```
stl_cars <- stl(ukcars, s.window = "periodic", robust = TRUE)
```

```
seasonal <- stl_cars$time.series[,1]
cars_sa <- ukcars - seasonal
```

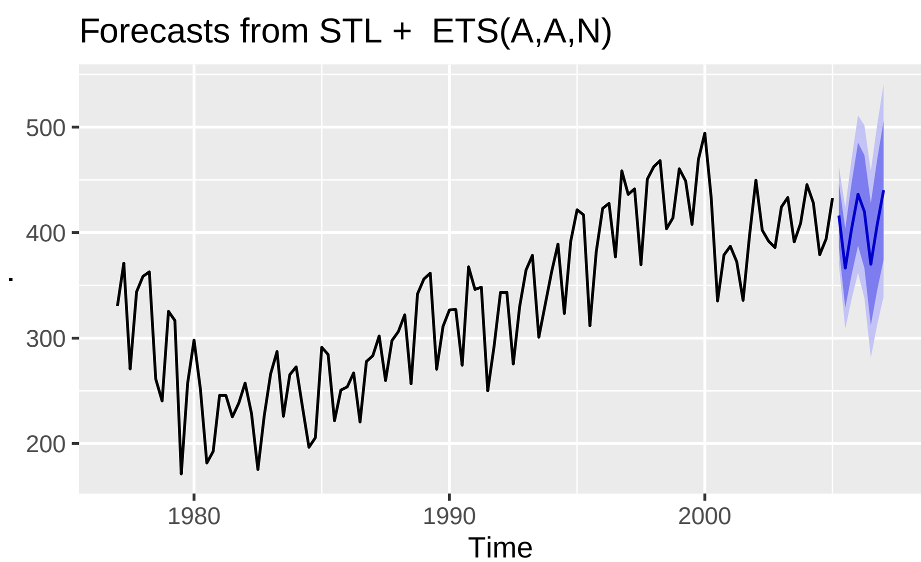
a.3)

```
stlf_ets_ukcars <- ukcars %>% stlf(h = 8, etsmodel = "AAN", damped = TRUE)
autoplot(stlf_ets_ukcars)
```



a.4)

```
stlf_ets_ukcars_holt <- ukcars %>% stlf(h = 8, etsmodel = "AAN", damped = FALSE)
autoplot(stlf_ets_ukcars_holt)
```

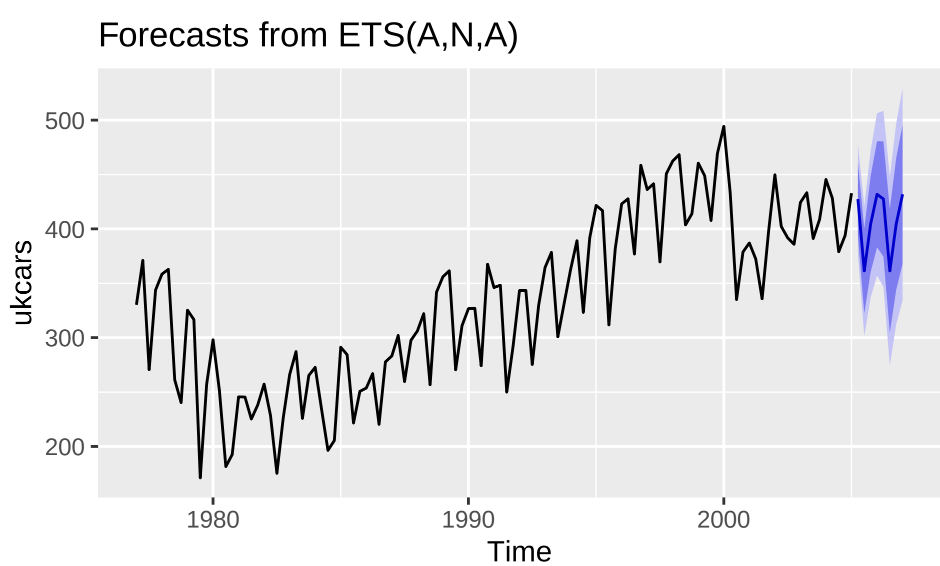


a.5)

```
ets_ukcars <- ets(ukcars)
summary(ets_ukcars)
```

```
## ETS(A,N,A)
##
## Call:
##  ets(y = ukcars)
##
##   Smoothing parameters:
##     alpha = 0.6199
##     gamma = 1e-04
##
##   Initial states:
##     l = 314.2568
##     s = -1.7579 -44.9601 21.1956 25.5223
##
##   sigma:  25.9302
##
##      AIC     AICc      BIC
## 1277.752 1278.819 1296.844
##
## Training set error measures:
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 1.313884 25.23244 20.17907 -0.1570979 6.629003 0.6576259
##                    ACF1
## Training set 0.02573334
```

```r
autoplot(forecast(ets_ukcars, h = 8))
```



a.6)

```r
print("Accuracy of stlf model")
```

```
## [1] "Accuracy of stlf model"
```

```r
accuracy(stlf_ets_ukcars)
```

```
##                    ME     RMSE      MAE        MPE     MAPE     MASE       ACF1
## Training set 1.551267 23.32113 18.48987 0.04121971 6.042764 0.602576 0.02262668
```

```r
print("Accuracy of stlf holt model")
```

```
## [1] "Accuracy of stlf holt model"
```

```r
accuracy(stlf_ets_ukcars_holt)
```

```
##                      ME   RMSE     MAE        MPE    MAPE      MASE       ACF1
## Training set -0.3412727 23.295 18.1605 -0.5970778 5.98018 0.5918418 0.02103582
```

```r
print("Accuracy of ETS(A, N, A) model")
```

```
## [1] "Accuracy of ETS(A, N, A) model"
```

```r
accuracy(ets_ukcars)
```

```
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 1.313884 25.23244 20.17907 -0.1570979 6.629003 0.6576259
##                    ACF1
## Training set 0.02573334
```

Using the Holt's linear method for the seasonally adjusted data resulted in the best model.

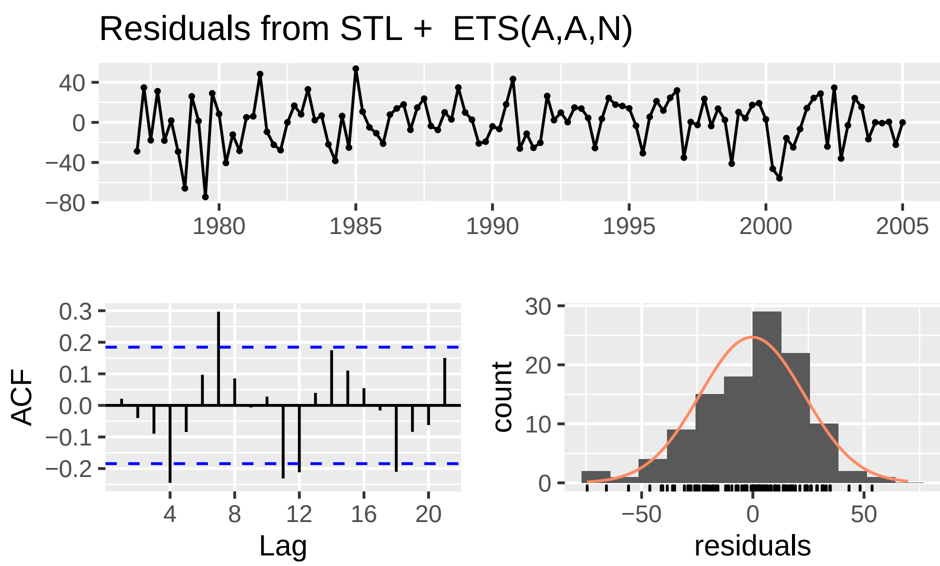
a.7)

Based on for example the RMSE, the answer would be same as for a.6.

a.8)

```r
checkresiduals(stlf_ets_ukcars_holt)
```

```
## Warning in checkresiduals(stlf_ets_ukcars_holt): The fitted degrees of freedom
## is based on the model used for the seasonally adjusted data.
```

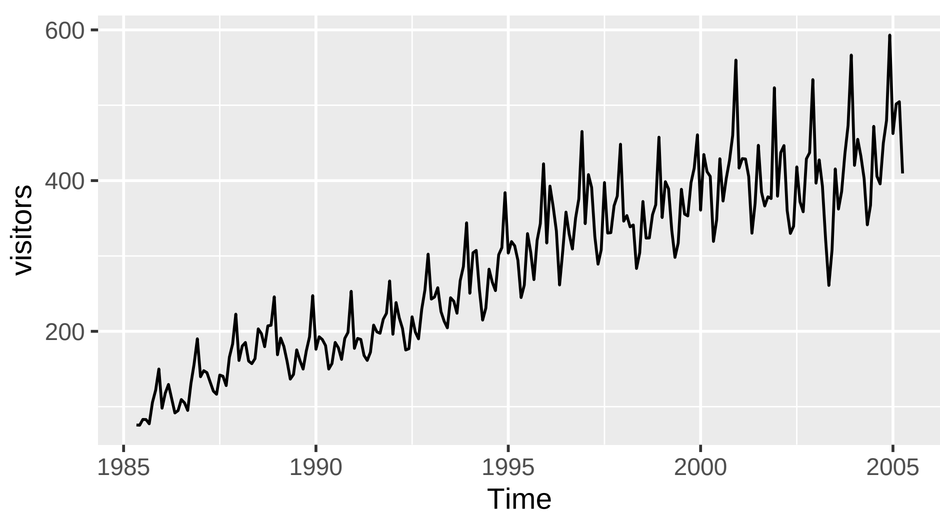



```
##
##  Ljung-Box test
##
## data:  Residuals from STL +  ETS(A,A,N)
## Q* = 22.061, df = 4, p-value = 0.0001949
##
## Model df: 4.   Total lags used: 8
```

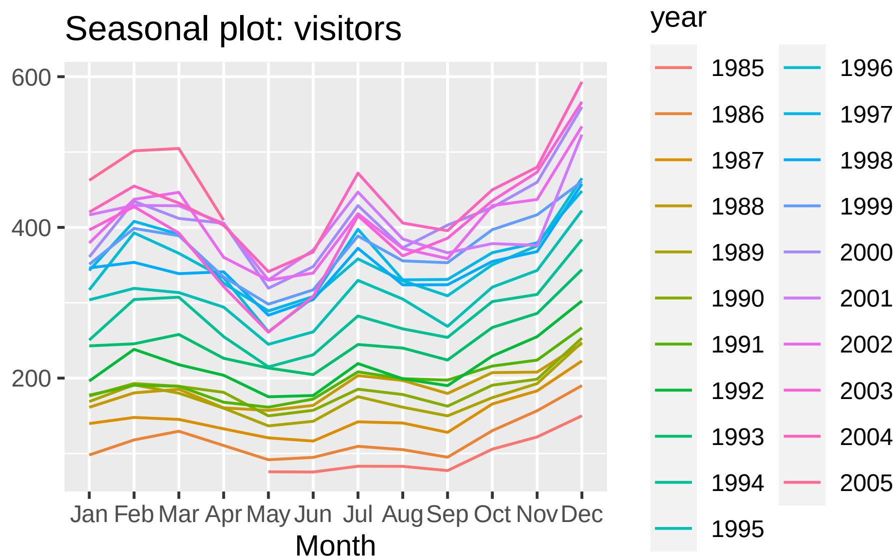
First we notice that the residuals seem not be fully normally distributed. When looking at the ACF plot, we also see some autocorrelation.

b.1)

```r
autoplot(visitors)
```



```r
ggseasonplot(visitors)
```



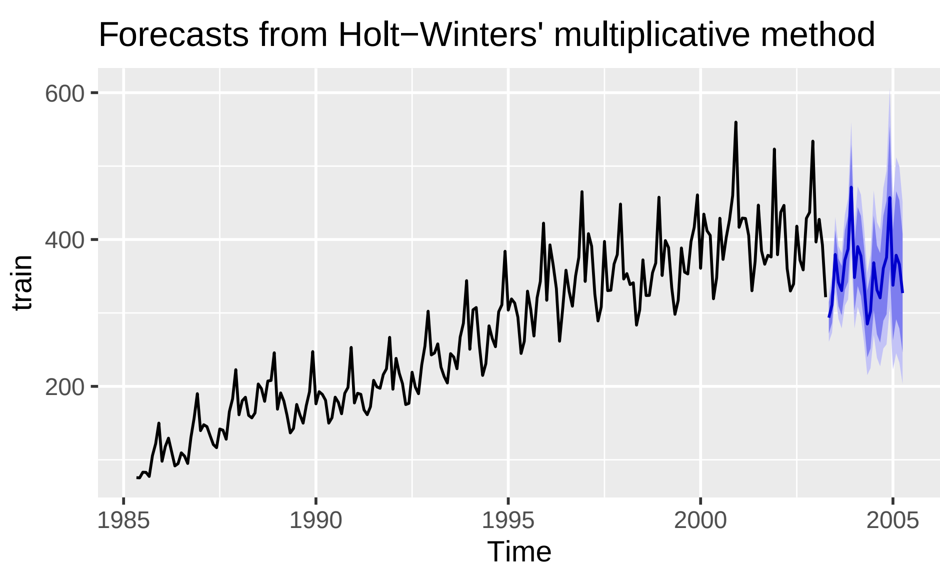
The data contains an increasing trend over time, with seasonality clearly visible. There also seems to be an outlier (or decrease) in 2003.

b.2)

```r
train <- subset(visitors, end = length(visitors) - 24)
test <- subset(visitors, start = length(visitors) - 23)
hw_mul_visitors_train <- hw(train,
                            h = 24,
                            seasonal = "multiplicative")
```

b.3)

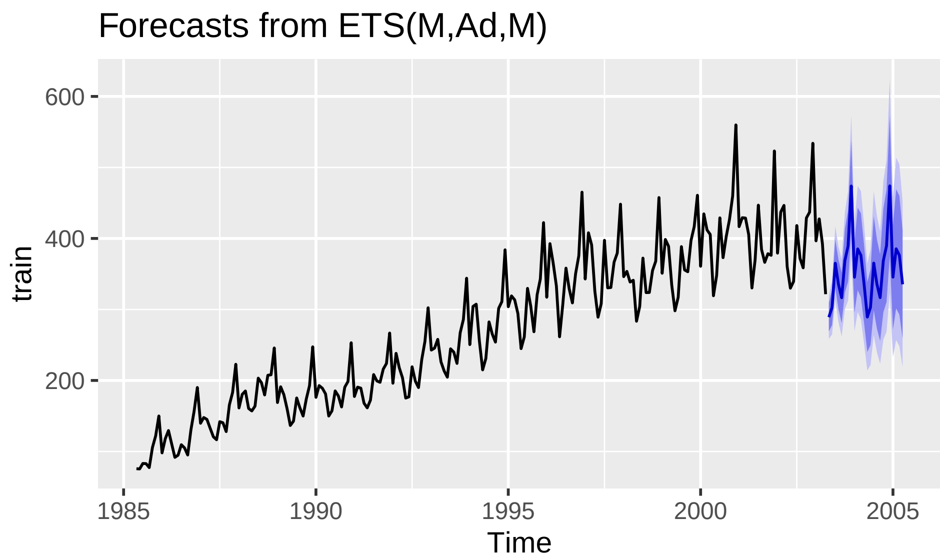
```r
autoplot(hw_mul_visitors_train)
```



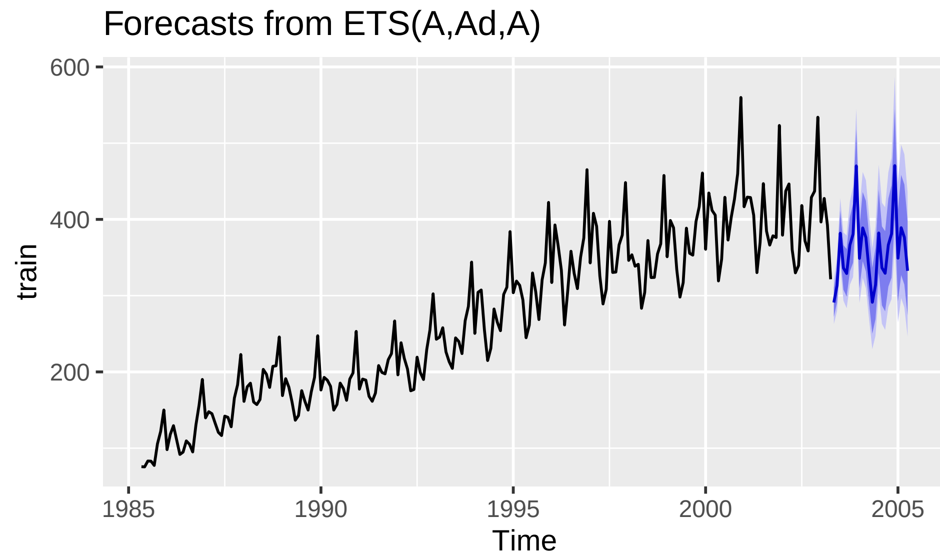
We can see that the variance of the seasonality increased over time, next to the fact that the amount of visitors increased. Multiplicative can handle that, but additive seasonality not.

b.4)

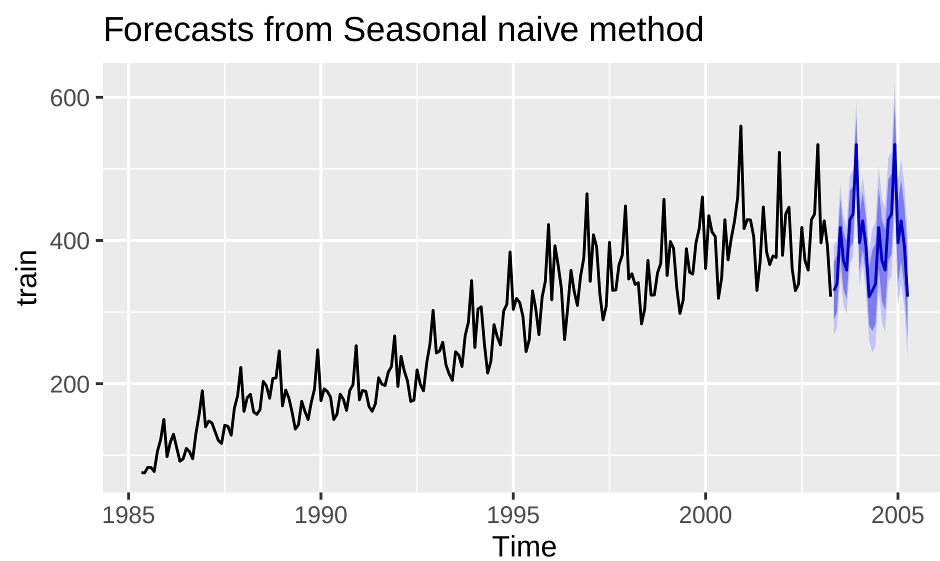
```
# b.4.1
ets_visitor_train <- forecast(ets(train), h = 24)
autoplot(ets_visitor_train)
```



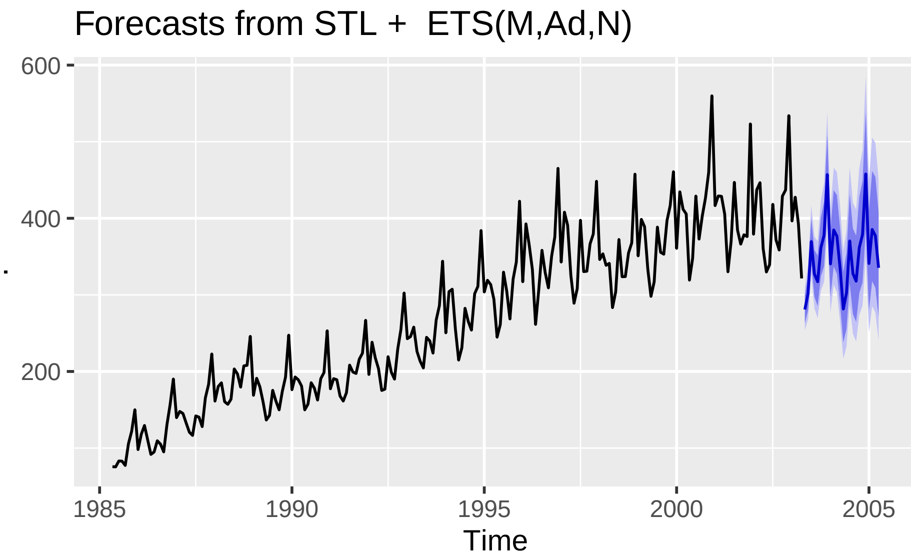

```
# b.4.2
ets_boxcox_visitor_train <- forecast(
  ets(train,
      lambda = BoxCox.lambda(train),
      additive.only = TRUE),
  h = 24
)
autoplot(ets_boxcox_visitor_train)
```




```
# b.4.3
snaive_visitor_train <- snaive(train, h = 24)
autoplot(snaive_visitor_train)
```




```
# b.4.4
boxcox_stl_ets_visitors_train <- train %>%
  stlm(
    lambda = BoxCox.lambda(train),
    s.window = 13,
    robust = TRUE,
    method = "ets"
  ) %>%
  forecast(h = 24)
autoplot(boxcox_stl_ets_visitors_train)
```



b.5)

```
accuracy(hw_mul_visitors_train, test)
```

```
##                      ME     RMSE      MAE        MPE      MAPE      MASE
## Training set -0.9749466 14.06539 10.35763 -0.5792169  4.223204 0.3970304
## Test set     72.9189889 83.23541 75.89673 15.9157249 17.041927 2.9092868
##                   ACF1 Theil's U
## Training set 0.1356528        NA
## Test set     0.6901318  1.151065
```

```
accuracy(ets_visitor_train, test)
```

```
##                      ME     RMSE      MAE        MPE      MAPE     MASE
## Training set  0.7640074 14.53480 10.57657  0.1048224  3.994788 0.405423
## Test set     72.1992664 80.23124 74.55285 15.9202832 16.822384 2.857773
##                     ACF1 Theil's U
## Training set -0.05311217        NA
## Test set      0.58716982  1.127269
```

```
accuracy(ets_boxcox_visitor_train, test)
```

```
##                     ME     RMSE      MAE        MPE      MAPE      MASE
## Training set  1.001363 14.97096 10.82396  0.1609336  3.974215 0.4149057
## Test set     69.458843 78.61032 72.41589 15.1662261 16.273089 2.7758586
##                     ACF1 Theil's U
## Training set -0.02535299        NA
## Test set      0.67684148  1.086953
```

```
accuracy(snaive_visitor_train, test)
```

```
##                    ME     RMSE      MAE      MPE      MAPE     MASE      ACF1
## Training set 17.29363 31.15613 26.08775 7.192445 10.285961 1.000000 0.6327669
## Test set     32.87083 50.30097 42.24583 6.640781  9.962647 1.619375 0.5725430
```

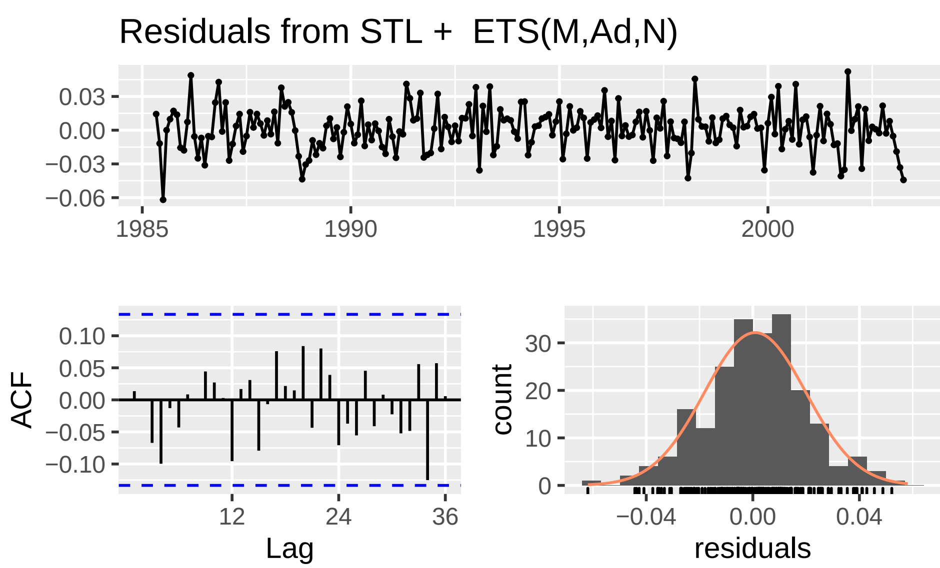
```
##              Theil's U
## Training set        NA
## Test set     0.6594016
```

```
accuracy(boxcox_stl_ets_visitors_train, test)
```

```
##                      ME     RMSE       MAE         MPE     MAPE      MASE
## Training set  0.5803348 13.36431  9.551391  0.08767744  3.51950 0.3661256
## Test set     76.3637263 84.24658 78.028992 16.87750474 17.51578 2.9910209
##                     ACF1 Theil's U
## Training set -0.05924203        NA
## Test set      0.64749552  1.178154
```

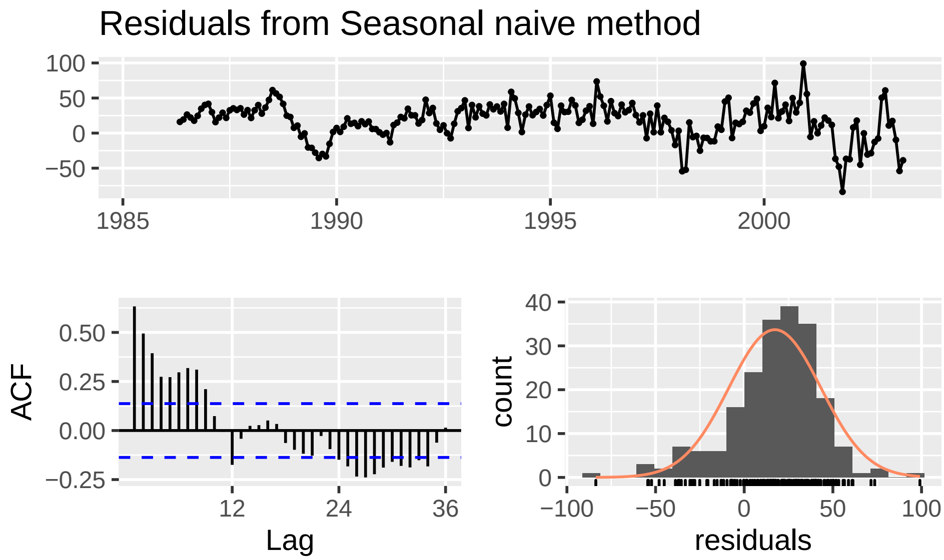
```
checkresiduals(boxcox_stl_ets_visitors_train)
```

```
## Warning in checkresiduals(boxcox_stl_ets_visitors_train): The fitted degrees of
## freedom is based on the model used for the seasonally adjusted data.
```




```
##
##  Ljung-Box test
##
## data:  Residuals from STL +  ETS(M,Ad,N)
## Q* = 15.032, df = 19, p-value = 0.7205
##
## Model df: 5.   Total lags used: 24
```

```
checkresiduals(snaive_visitor_train)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 295.02, df = 24, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 24
```

The STL decomposition applied to the Box-Cox transformed data followed by an ETS model applied to the seasonally adjusted data seems to be the best model when it comes to the training data, but if we look at the test set, we see that the seasonal naive model is best according to the RMSE. The ETS model seems to pass the residuals check, but the seasonal naive does not.

b.6)

```
forecastfunction_boxcox = function(x, h) forecast(ets(x, lambda = BoxCox.lambda(x), additive.o

forecastfunction_stlm = function(x, h) forecast(stlm(x, lambda = BoxCox.lambda(x), s.window = 

forecastfunction_ets = function(x, h) forecast(ets(x), h=h)

sqrt(mean(tsCV(visitors, snaive, h = 1)^2, na.rm = TRUE))
```

```
## [1] 32.78675
```

```
sqrt(mean(tsCV(visitors, forecastfunction_boxcox, h = 1)^2, na.rm = TRUE))
```

```
## [1] 18.86439
```

```
sqrt(mean(tsCV(visitors, forecastfunction_stlm, h = 1)^2, na.rm = TRUE))
```

```
## [1] 17.49642
```

```
sqrt(mean(tsCV(visitors, forecastfunction_ets, h = 1)^2, na.rm = TRUE))
```

```
## [1] 18.52985
```

```
sqrt(mean(tsCV(visitors, hw, h = 1, seasonal = "multiplicative")^2, na.rm = TRUE))
```

```
## [1] 19.62107
```

In this case, the STL followed by ETS seems to be the best model here, based on the RSME, as we've also seen when checking the score for the training data.