Week 5 Homework

Ben Arancibia

July 8, 2015

```r
library(fpp)
```

```
## Loading required package: forecast

## Warning: package 'forecast' was built under R version 3.1.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.1.3

##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Loading required package: timeDate
## This is forecast 6.1
##
## Loading required package: fma
## Loading required package: tseries
## Loading required package: expsmooth

## Warning: package 'expsmooth' was built under R version 3.1.3

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 3.1.3
```

HA 6.1

1) Show that a 3×5 MA is equivalent to a 7-term weighted moving average with weights of 0.067, 0.133, 0.200, 0.200, 0.200, 0.133, and 0.067.

First do the 7 term weighted moving average

```r
weights <- (c(0.067, 0.133, 0.200, 0.200, 0.200, 0.133, 0.067))

ma7 <- ma(weights, order=7)

ma7
```

```
## [1]        NA        NA        NA 0.1428571        NA        NA        NA
```

Next do the 3x5 MA

```r
ma5 <- ma(weights, order=5)
ma3x5 <- ma(ma5, order=3)
```

The reason that 3x5 is equivalent to the 7 term weight moving average is because combinations of moving averages result in weighted moving averages.

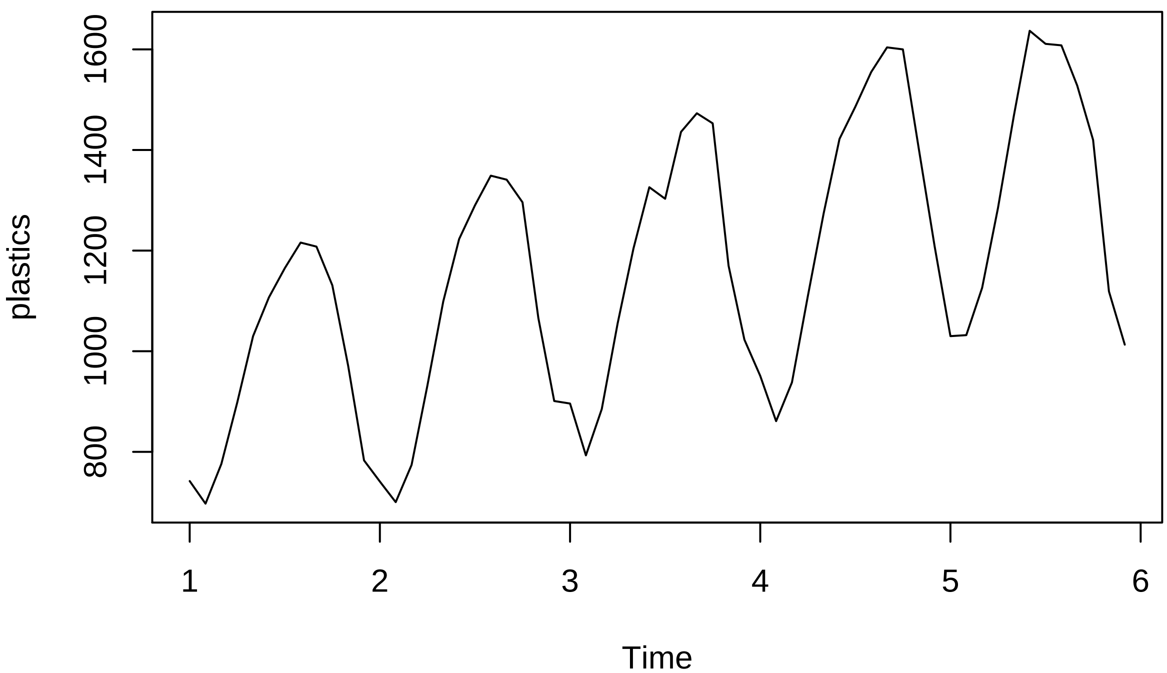
HA 6.2 The data below represent the monthly sales (in thousands) of product A for a plastics manufacturer for years 1 through 5 (data set plastics).

```r
plastics
```

```
##    Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1  742  697  776  898 1030 1107 1165 1216 1208 1131  971  783
## 2  741  700  774  932 1099 1223 1290 1349 1341 1296 1066  901
## 3  896  793  885 1055 1204 1326 1303 1436 1473 1453 1170 1023
## 4  951  861  938 1109 1274 1422 1486 1555 1604 1600 1403 1209
## 5 1030 1032 1126 1285 1468 1637 1611 1608 1528 1420 1119 1013
```

a) Plot the time series of sales of product A. Can you identify seasonal fluctuations and/or a trend?

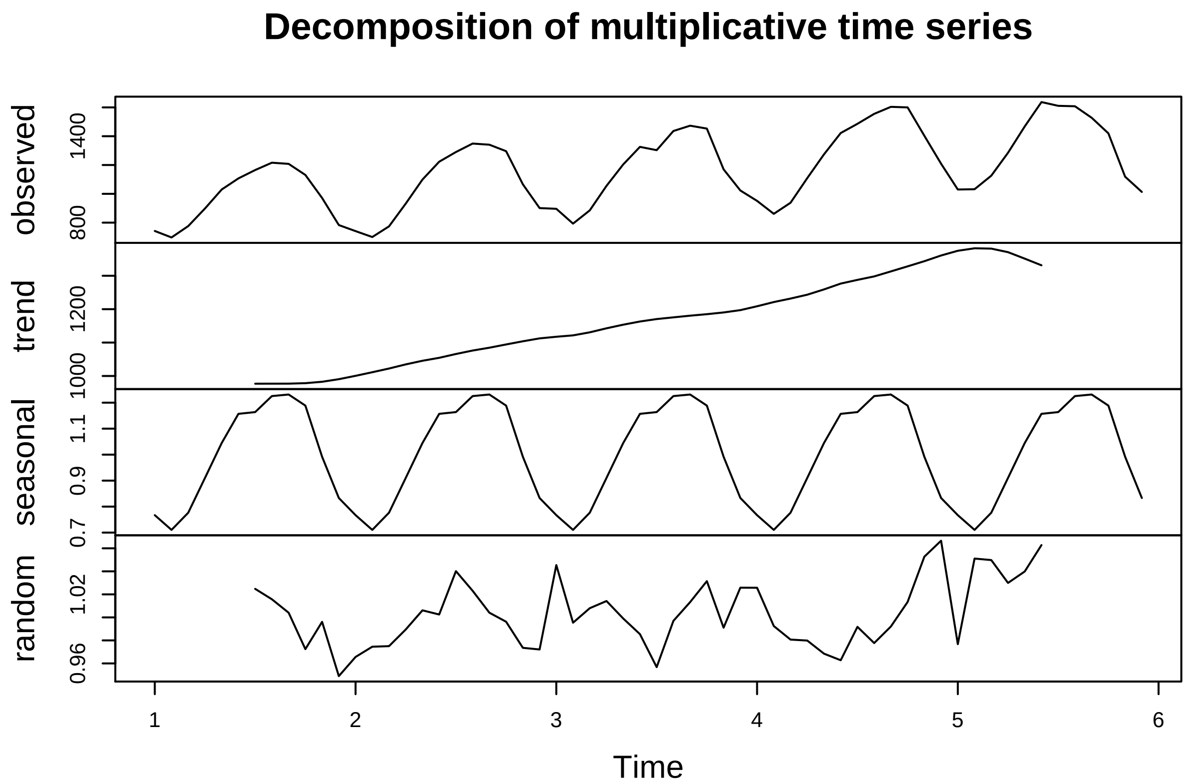
```r
plot(plastics)
```



There is seasonal and upward trend for the product A.

b) Use a classical multiplicative decomposition to calculate the trend-cycle and seasonal indices.

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

plot(decomp)
```

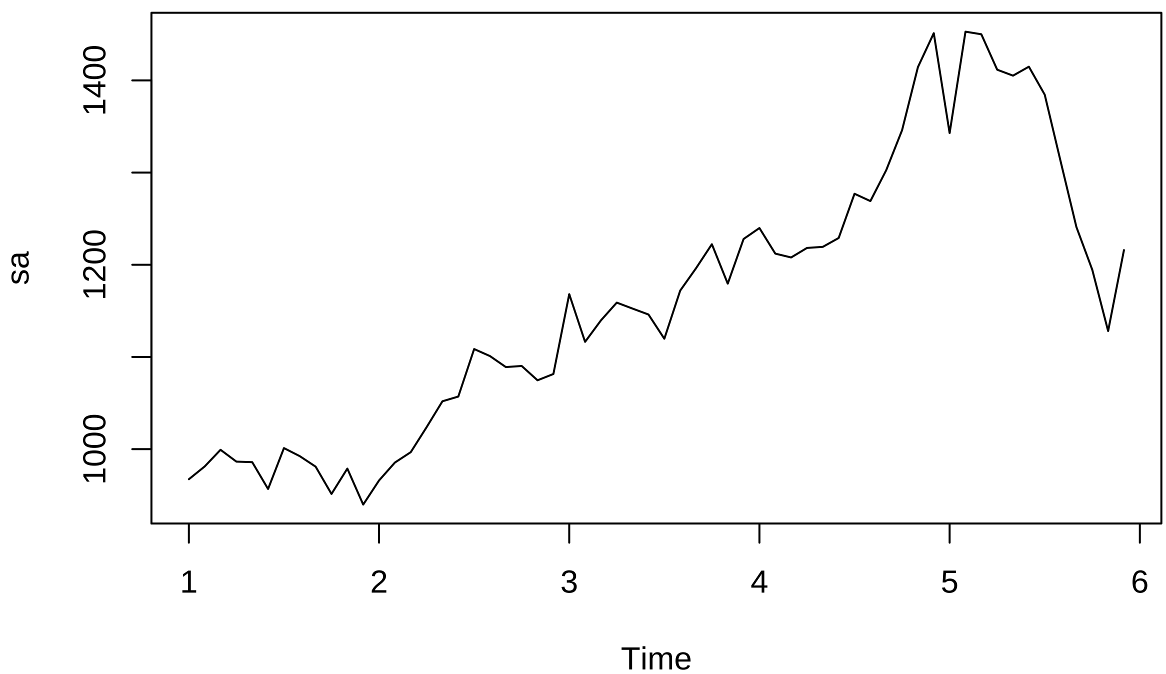


c) Do the results support the graphical interpretation from part (a)?

Yes the results show there is an unchanged seasonanility (third graphic), but there is a tip in the trend that was not noticed in the first plot.

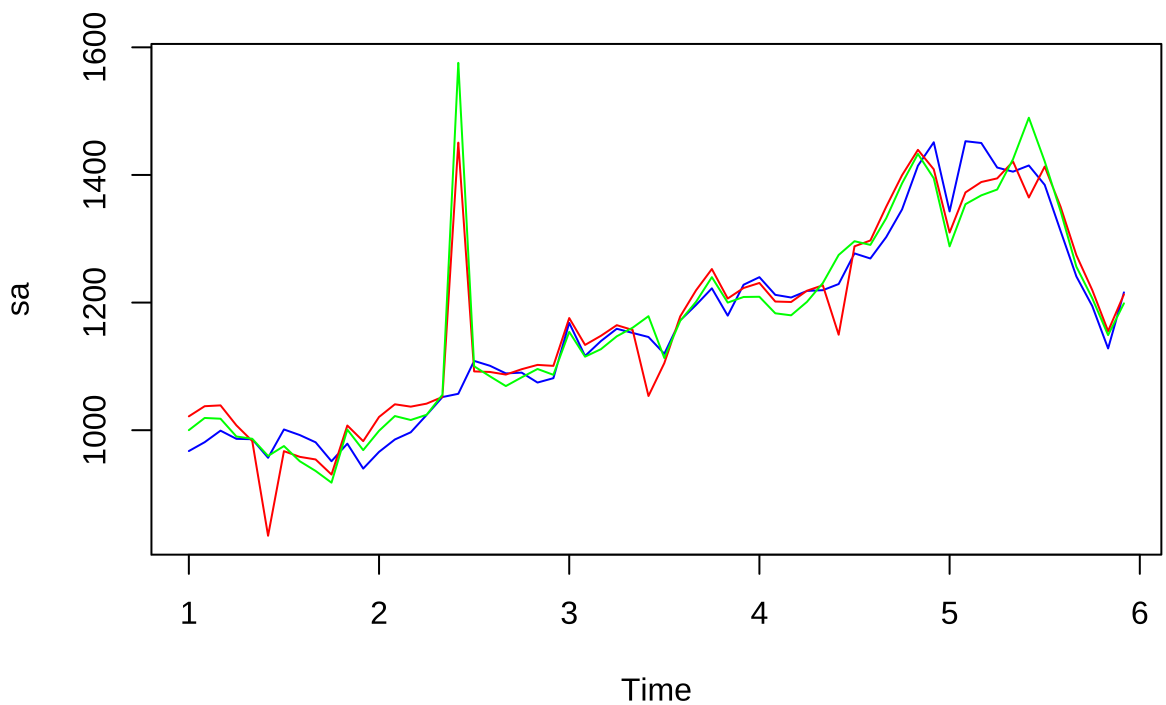
d) Compute and plot the seasonally adjusted data.

```
sa <- seasadj(decomp)
plot(sa)
```



e) 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?

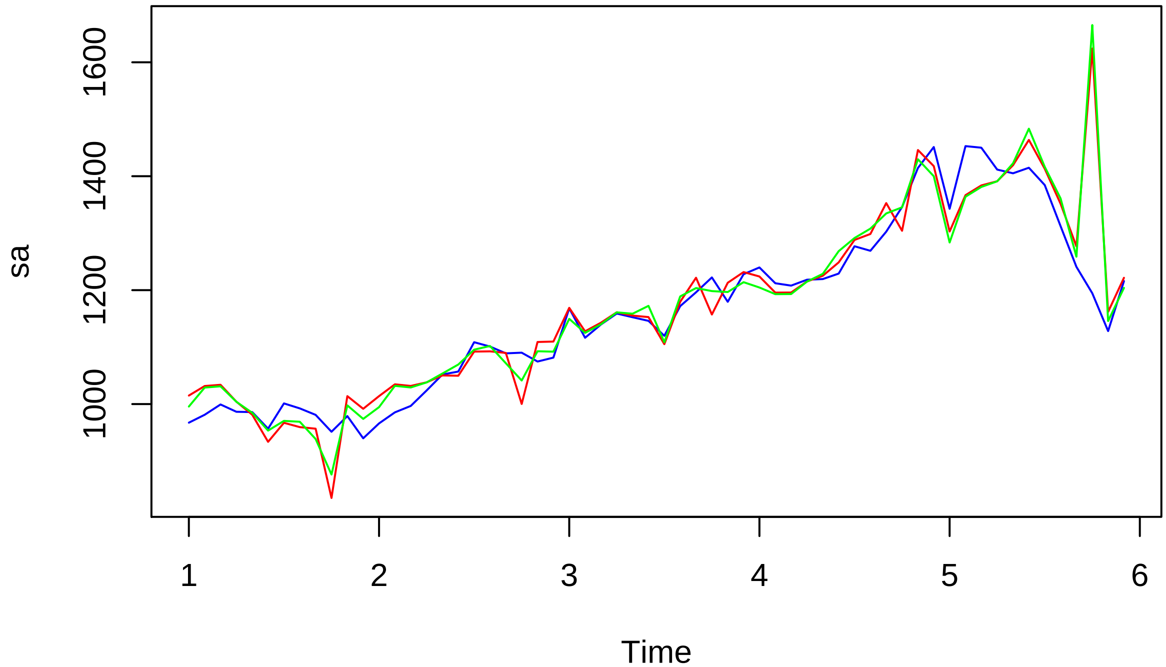
```
x <- plastics
x[18] <- x[18] + 500
sa2 <- seasadj(stl(x, s.window="periodic"))
sa3 <- seasadj(stl(x, s.window="periodic", robust=TRUE))
plot(sa, col="blue", ylim=range(sa,sa2,sa3))
lines(sa2,col="red")
lines(sa3, col="green")
```



If robust=FALSE, the seasonal adjusted series changes across the whole data set. Notice that there are a lot more differences in the red line compared to the blue line. If robust=TRUE, only the outlying point changes noticeably. There are not that many differences from the blue and green lines compared to the blue line.

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

```
#Repeat above function

x <- plastics
x[58] <- x[58] + 500
sa2 <- seasadj(stl(x, s.window="periodic"))
sa3 <- seasadj(stl(x, s.window="periodic", robust=TRUE))
plot(sa, col="blue", ylim=range(sa,sa2,sa3))
lines(sa2,col="red")
lines(sa3, col="green")
```

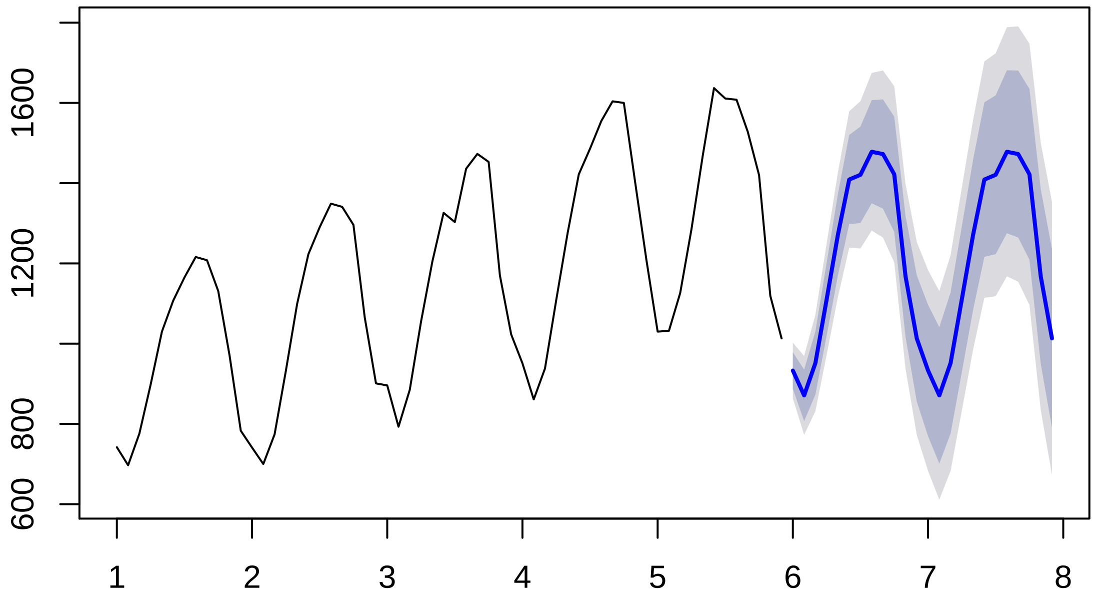


Having an outlier near the end gives it more impact.

g) Use a random walk with drift to produce forecasts of the seasonally adjusted data.

```
fc <- stlf(plastics, method="naive")
plot(fc)
```

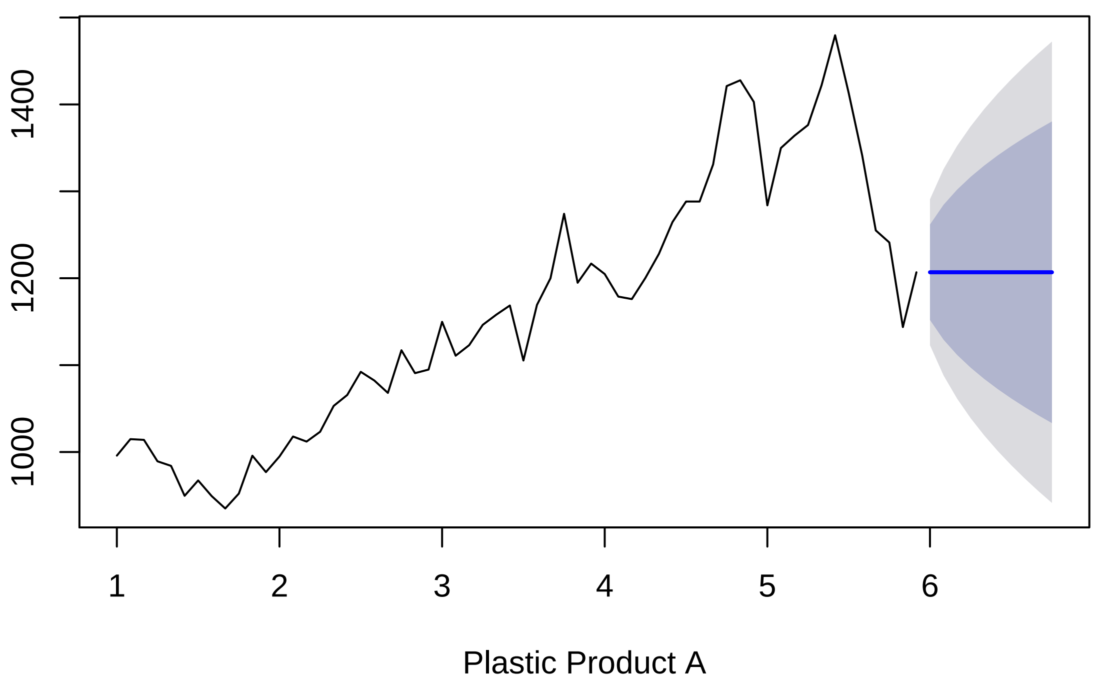
Forecasts from STL + Random walk



h) Reseasonalize the results to give forecasts on the original scale.

```
fit <- stl(plastics, s.window="periodic", robust=TRUE)
fitadj <- seasadj(fit)
plot(naive(fitadj), xlab="Plastic Product A",
  main="Naive forecasts of seasonally adjusted data")
```

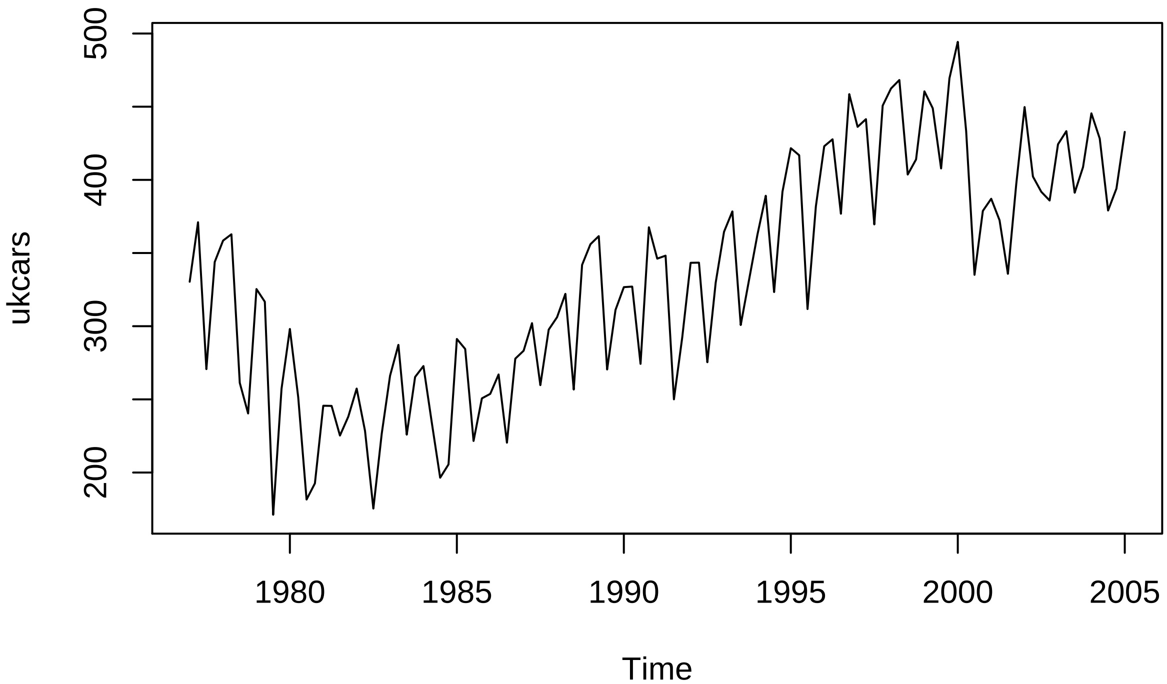
Naive forecasts of seasonally adjusted data



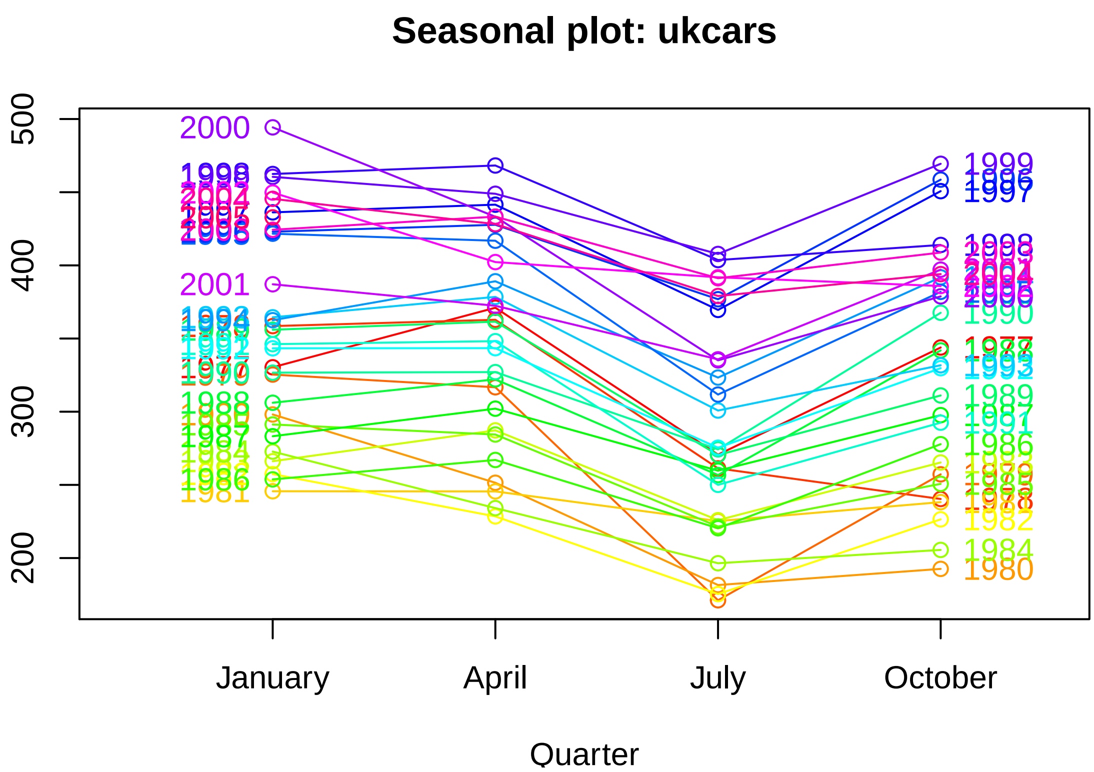
HA 7.3 For this exercise, use the quarterly UK passenger vehicle production data from 1977:1–2005:1 (data set ukcars).

a) Plot the data and describe the main features of the series.

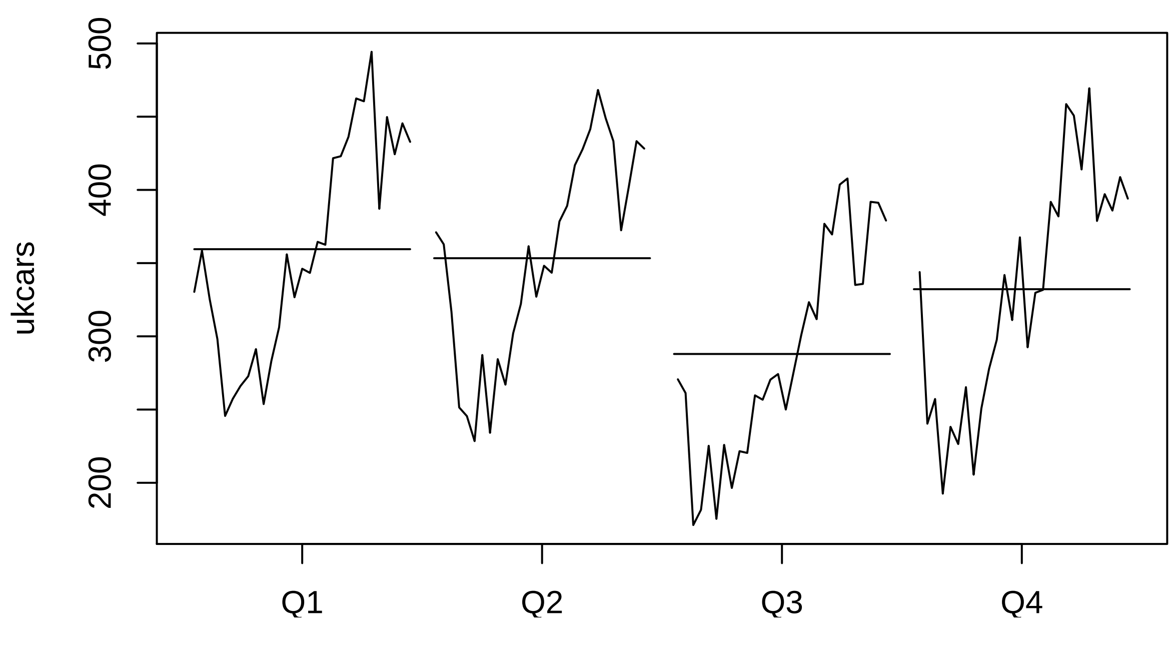
```
plot(ukcars)
```




```
seasonplot(ukcars,year.labels=TRUE, year.labels.left=TRUE, col=rainbow(30))
```

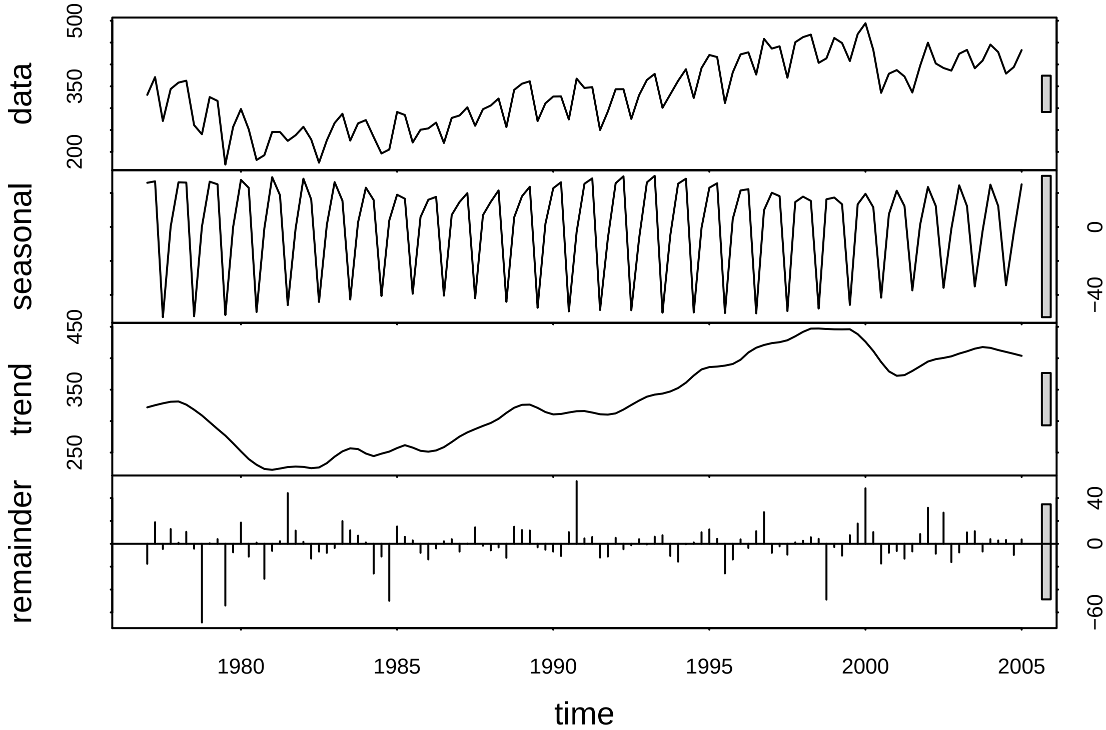



```
monthplot(ukcars)
```



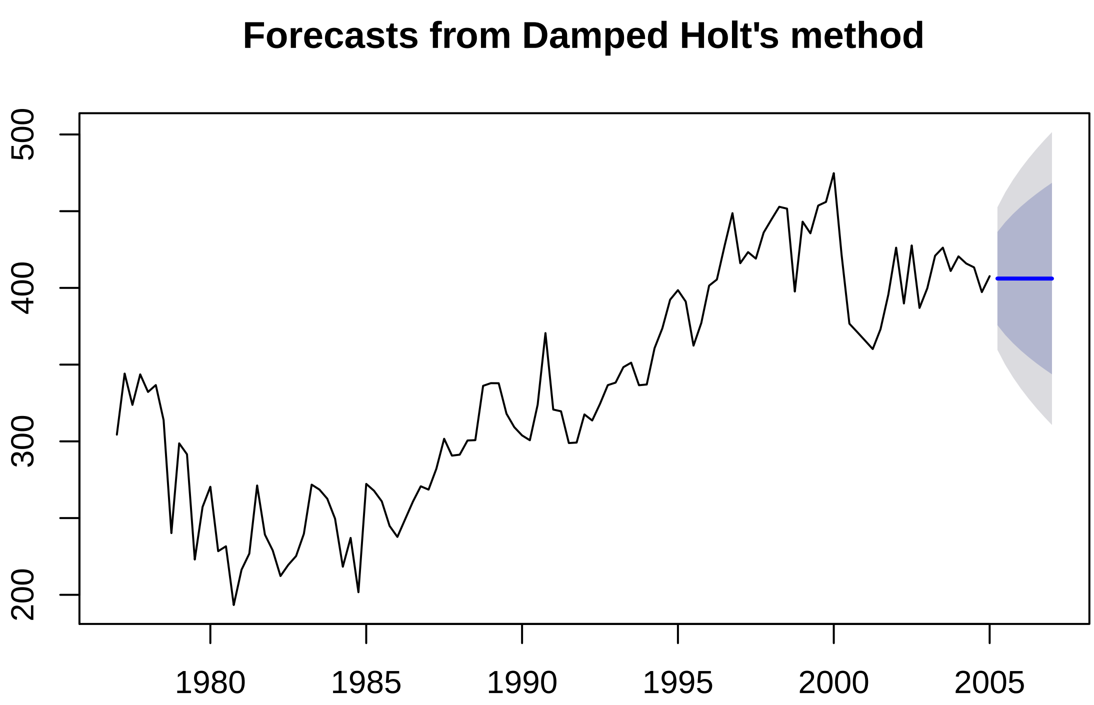
b) Decompose the series using STL and obtain the seasonally adjusted data.

```
decomp <- stl(ukcars, s.window=9, robust=TRUE)
plot(decomp)
```



c) Forecast the next two years of the series using an additive damped trend method applied to the seasonally adjusted data. Then reseasonalize the forecasts. Record the parameters of the method and report the RMSE of the one-step forecasts from your method.

```
sadamped <- seasadj(decomp)
fit <- holt(sadamped, h=8, damped=TRUE)
plot(fit)
```



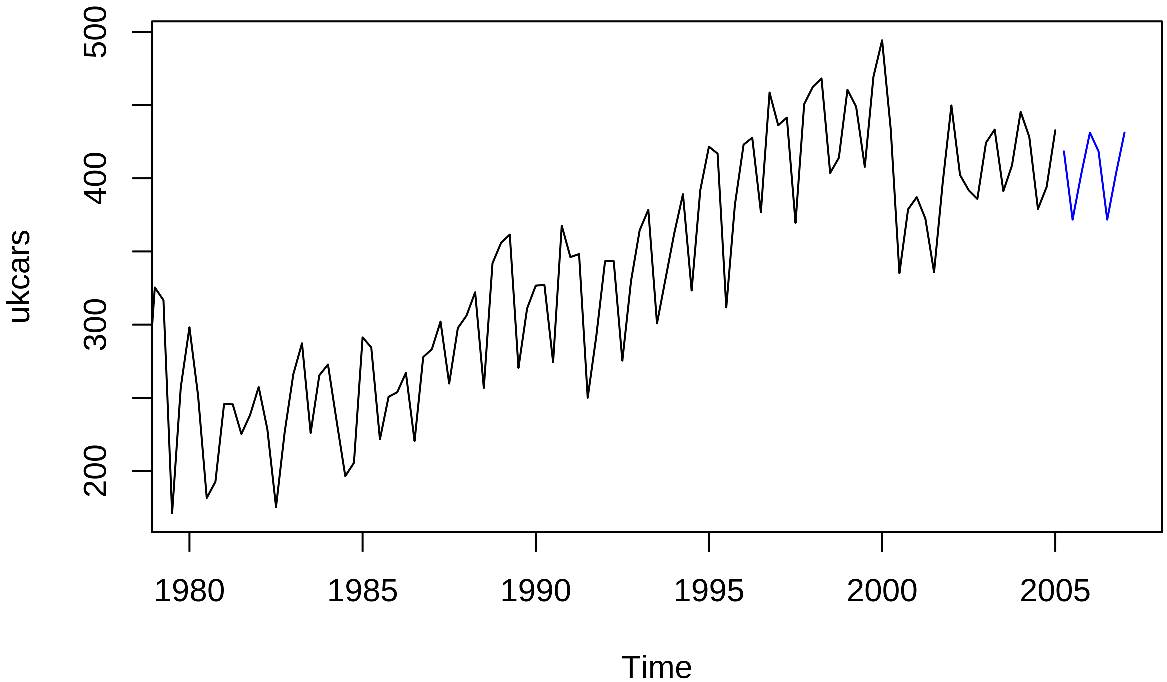
```r
accuracy(fit)
```

```
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 1.535971 23.70867 18.30215 0.05889913 5.888996 0.5931631
##                   ACF1
## Training set 0.0261438
```

```r
fit$model
```

```
## ETS(A,Ad,N)
##
## Call:
##  holt(x = sadamped, h = 8, damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.6776
##     beta  = 1e-04
##     phi   = 0.9324
##
##   Initial states:
##     l = 343.9825
##     b = -4.0396
##
##   sigma:  23.7087
##
##      AIC     AICc      BIC
## 1259.675 1260.236 1273.312
```

```r
twoyear <- rep(decomp$time.series[110:113,"seasonal"],2)
fc <- fit$mean + twoyear
plot(ukcars,xlim=c(1980,2007))
lines(fc, col="blue")
```

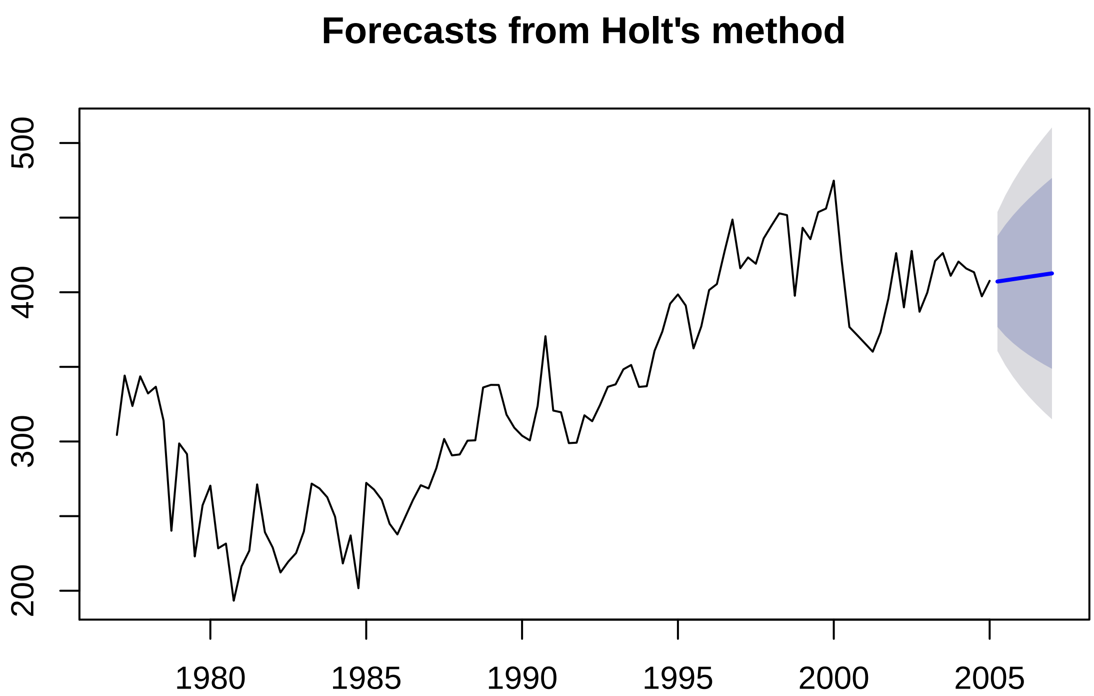


The parameters show a small beta which means the slope isn't changing much over time and a large alpha meaning the intercept is changing quickly.

RMSE of the model is 23.70867

d) Forecast the next two years of the series using Holt's linear method applied to the seasonally adjusted data. Then reseasonalize the forecasts. Record the parameters of the method and report the RMSE of of the one-step forecasts from your method.

```
sa <- seasadj(decomp)
fit <- holt(sa, h=8)
plot(fit)
```



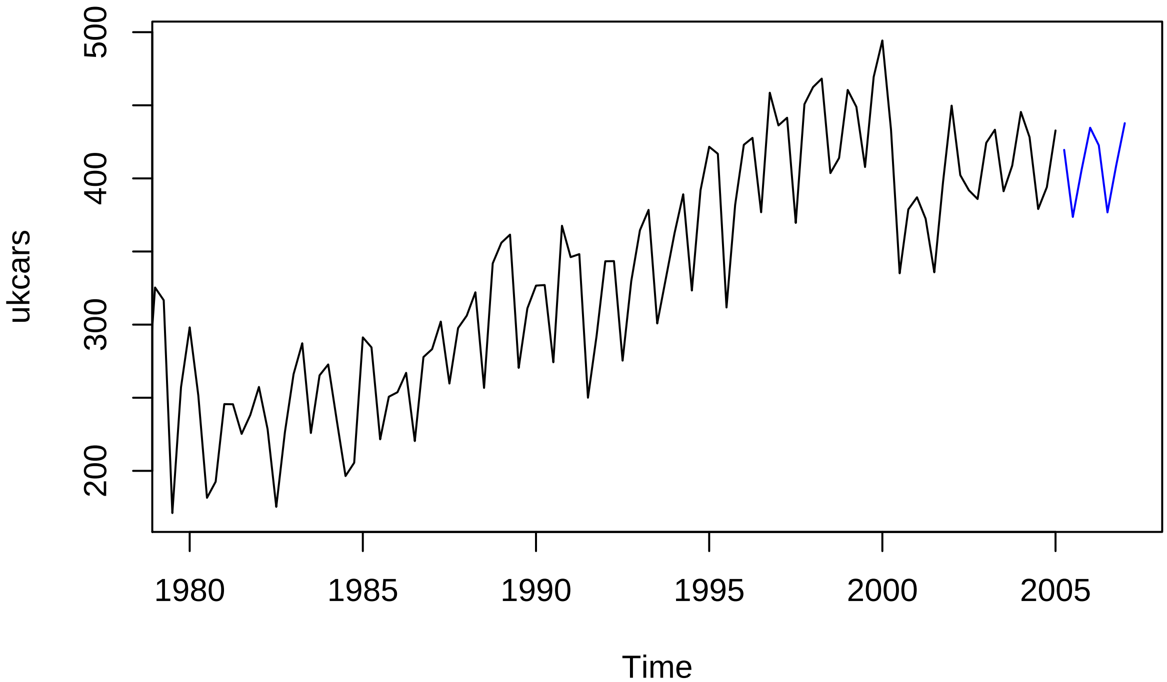

```
accuracy(fit)
```

```
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.2444925 23.75353 18.08048 -0.5325916 5.848881 0.5859791
##                    ACF1
## Training set 0.02156726
```

```
fit$model
```

```
## ETS(A,A,N)
##
## Call:
##  holt(x = sa, h = 8)
##
##   Smoothing parameters:
##     alpha = 0.6976
##     beta  = 1e-04
##
##   Initial states:
##     l = 338.4418
##     b = 0.7792
##
##   sigma:  23.7535
##
##      AIC     AICc      BIC
## 1258.102 1258.472 1269.012
```

```
twoyear <- rep(decomp$time.series[110:113,"seasonal"],2)
fc <- fit$mean + twoyear
plot(ukcars,xlim=c(1980,2007))
lines(fc, col="blue")
```



The parameters are pretty similar compared to the damped time series. The alpha value increased by around .02 and the RMSE of the model increased by .05 (RMSE = 23.75353)

e) Now use ets() to choose a seasonal model for the data.

```
fit <- ets(ukcars)
fit
```

```
## ETS(A,N,A)
##
## Call:
##  ets(y = ukcars)
##
##   Smoothing parameters:
##     alpha = 0.5987
##     gamma = 1e-04
##
##   Initial states:
##     l = 320.7749
##     s=-0.7213 -46.0315 21.052 25.7009
##
##   sigma:  25.249
##
##      AIC     AICc      BIC
## 1275.901 1276.693 1292.265
```

```
accuracy(fit)
```

```
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 1.261938 25.24902 20.28138 -0.1708837 6.667517 0.6609603
##                    ACF1
## Training set 0.04113908
```

ETS Choose A,N,A an additive seasonal component

f) Compare the RMSE of the fitted model with the RMSE of the model you obtained using an STL decomposition with Holt's method. Which gives the better insample fits?

```
accuracy(ets(ukcars))
```

```
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 1.261938 25.24902 20.28138 -0.1708837 6.667517 0.6609603
##                    ACF1
## Training set 0.04113908
```

```
sa <- seasadj(decomp)
fit <- holt(sa, h=8)

accuracy(fit)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.2444925 23.75353 18.08048 -0.5325916 5.848881 0.5859791
##                    ACF1
## Training set 0.02156726
```

The fitted model using ETS has a higher RMSE value, but not by much. ETS model had 25.24902 and STL Decomp with Holts method had 23.75353

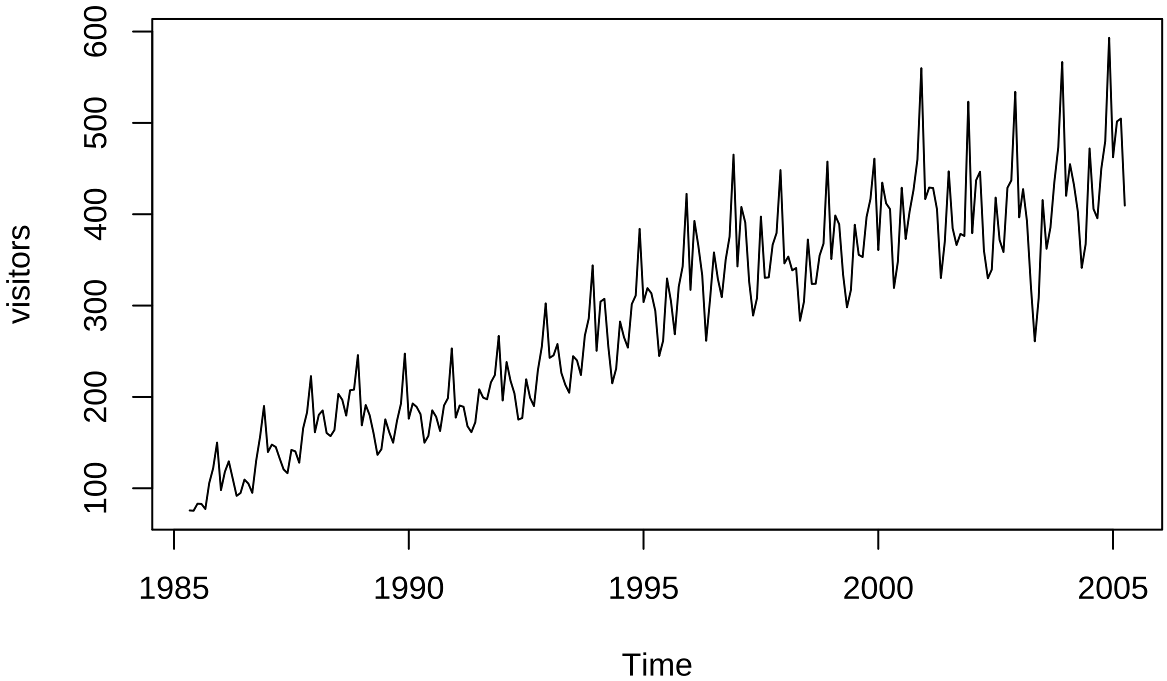
g) Compare the forecasts from the two approaches? Which seems most reasonable?

They are both comparable and don't have that much of a difference.

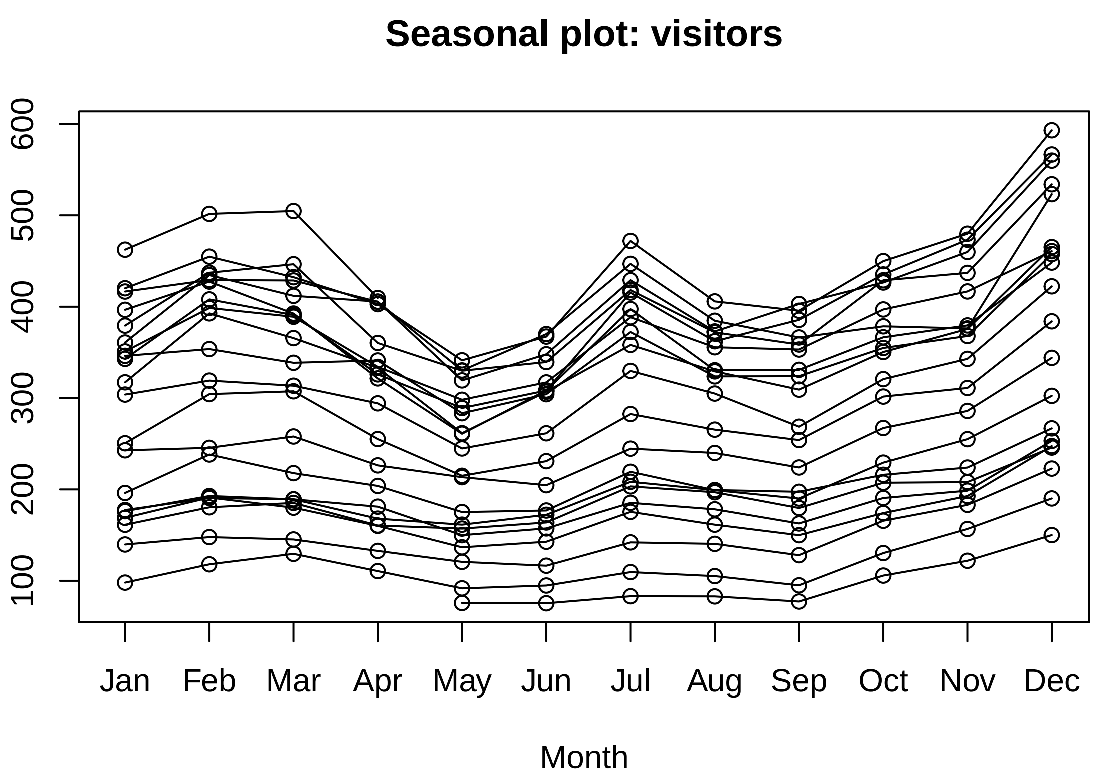
HA 7.4 For this exercise, use the monthly Australian short-term overseas visitors data, May 1985–April 2005. (Data set: visitors.)

a) Make a time plot of your data and describe the main features of the series.

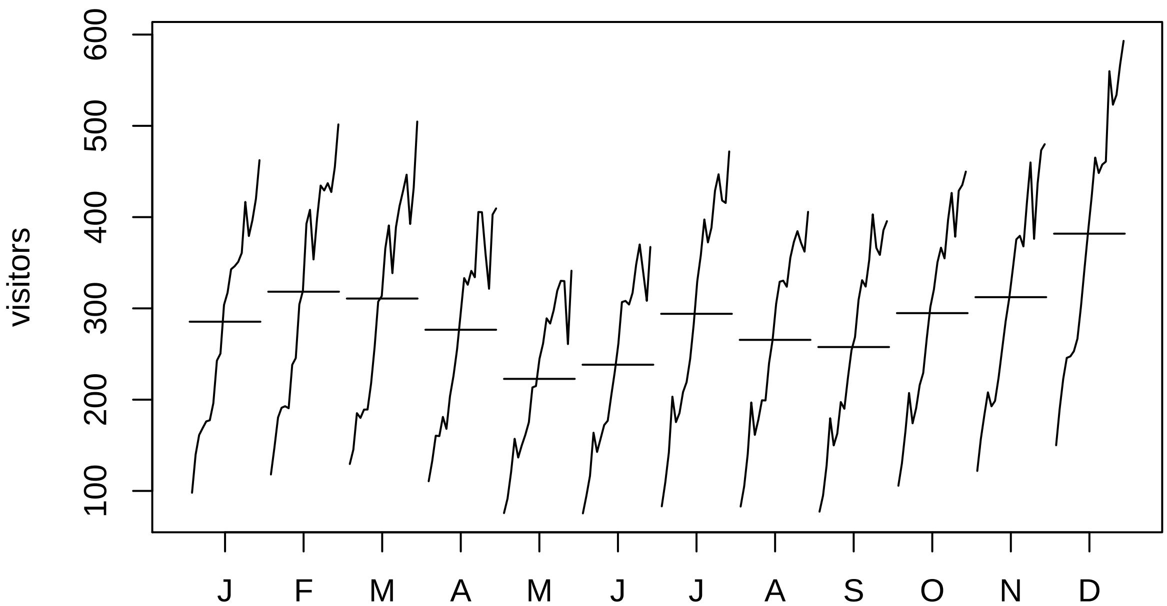
```
plot(visitors)
```



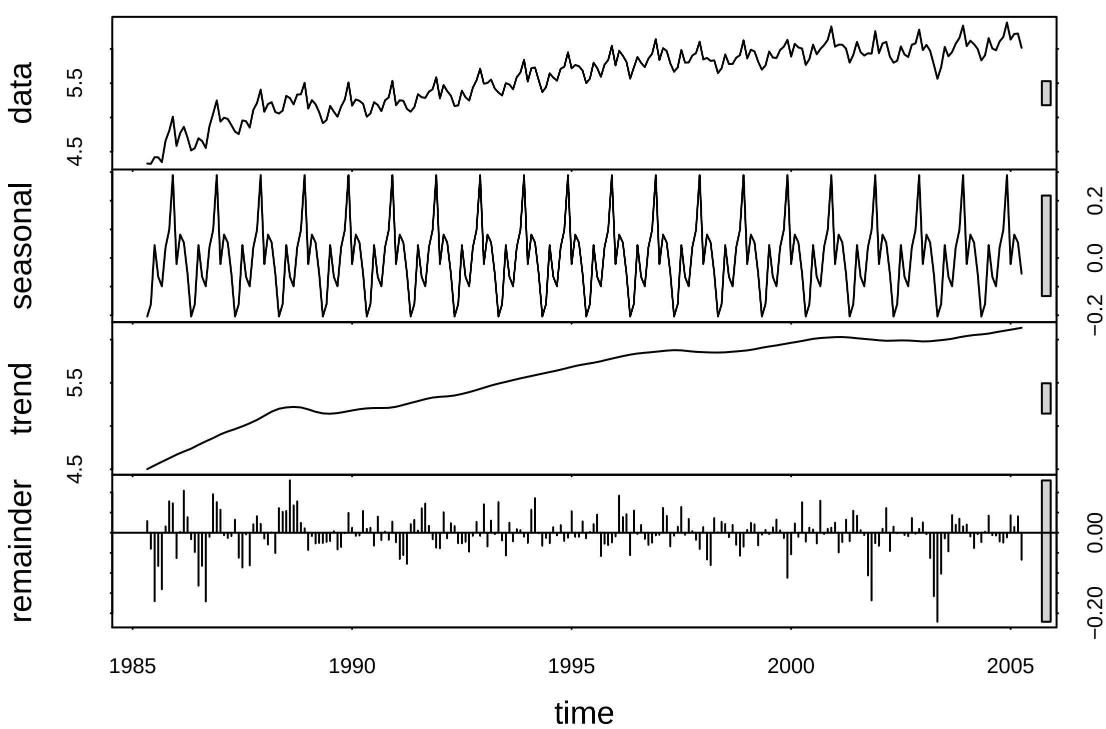

```r
seasonplot(visitors)
```



```r
monthplot(visitors)
```



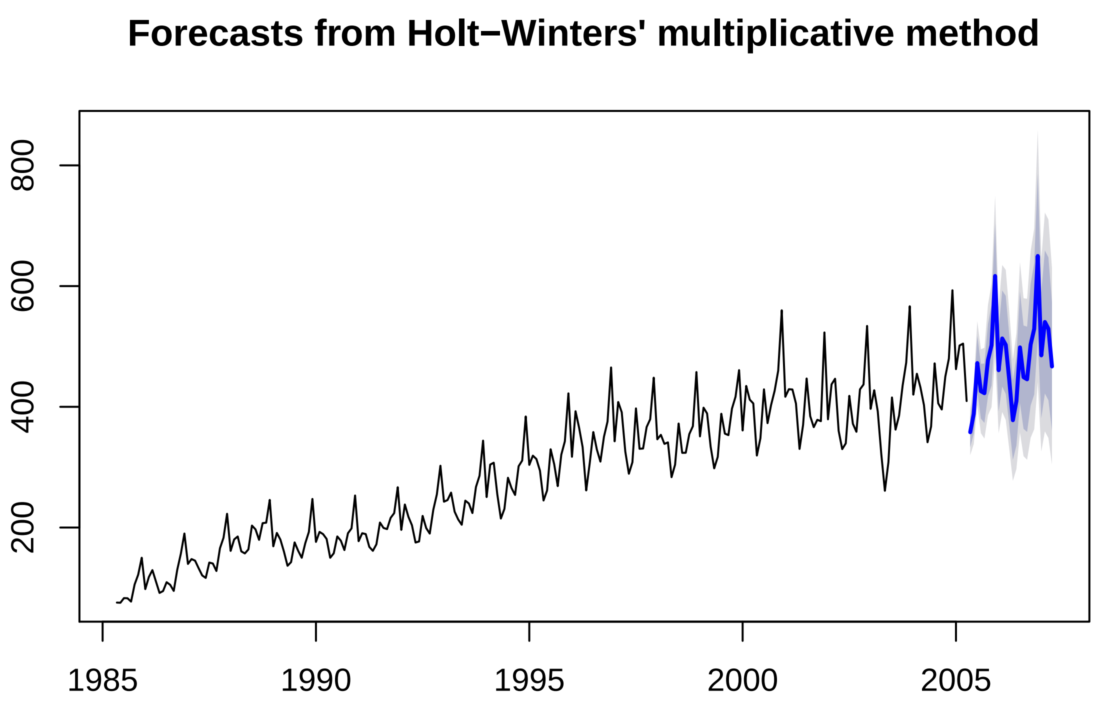

```r
plot(stl(log(visitors),s.window="periodic",robust=TRUE))
```



The dataset visitors is seasonal data with an increasing trend and increasing seasonal fluctuations. Peak seems to be in December, with smaller peak in July, andowest numbers of visitors in May. Each month has had similar increases over time. Seasonality looks stable. Trend relatively flat in recent years. Big negative outliers in 2003 (fourth chart).

b) Forecast the next two years using Holt-Winters' multiplicative method.

```r
fc <- hw(visitors,seasonal="mult")
plot(fc)
```

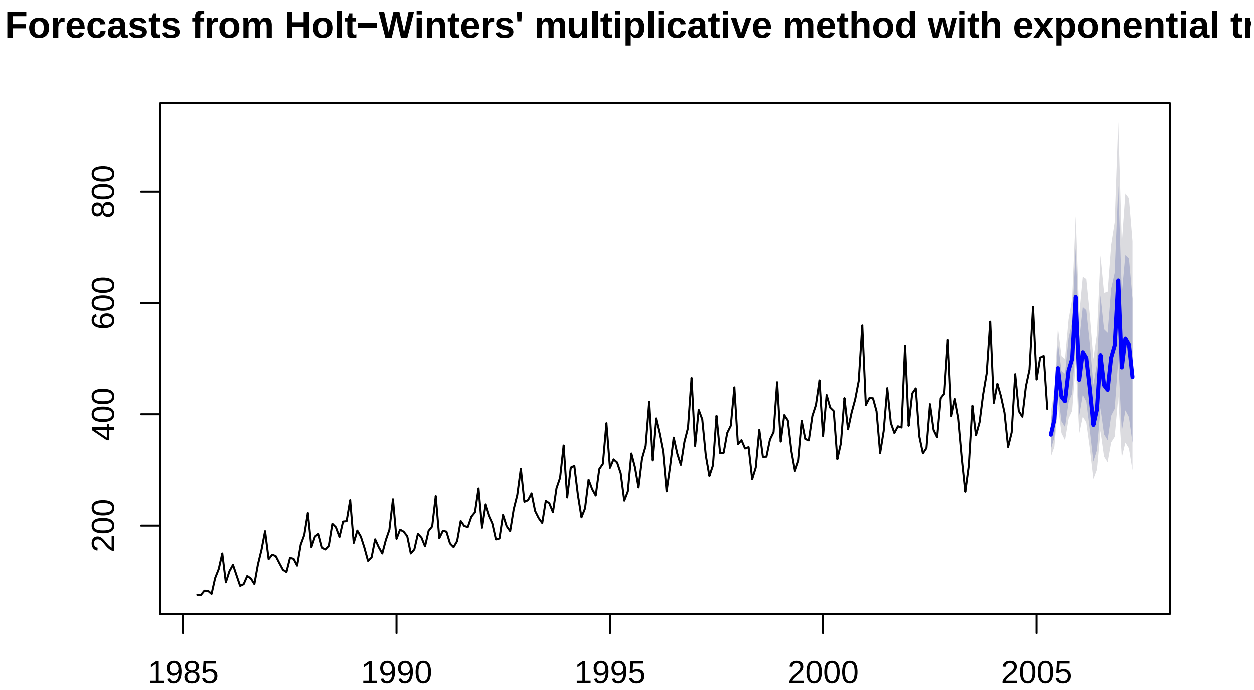


c) Why is multiplicative seasonality necessary here?

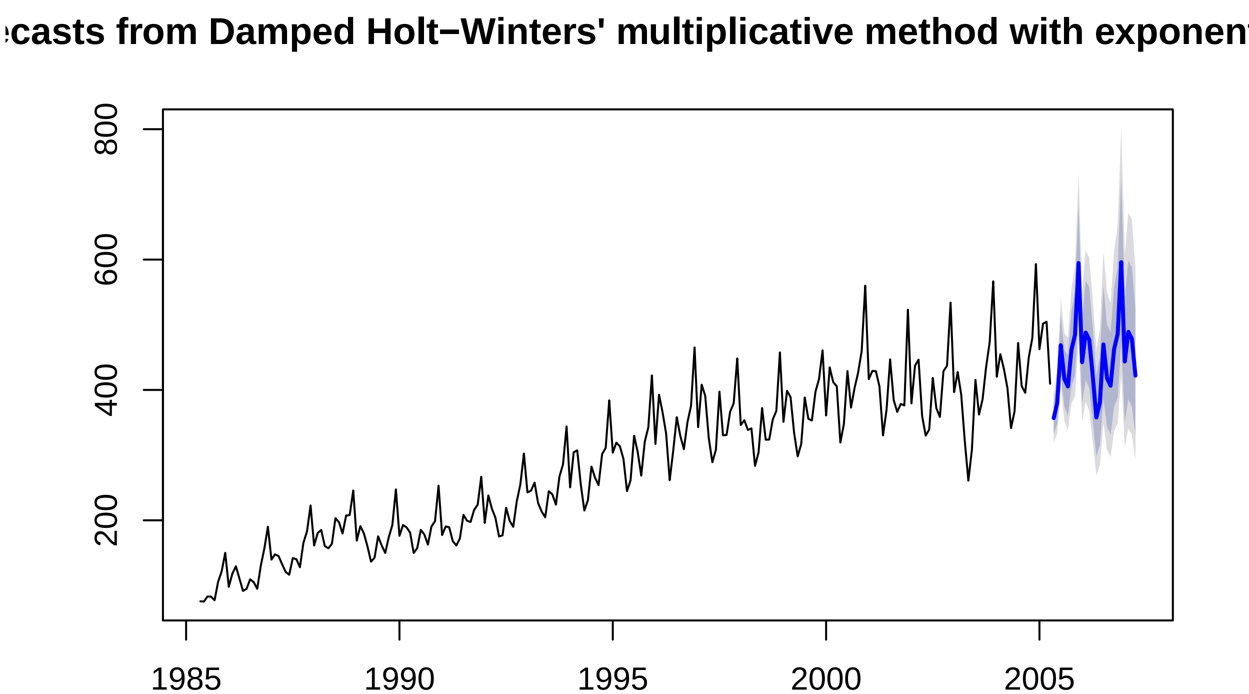
Multiplicative seasonality is necessary because of the increasing size of the fluctuations and increasing variation with the trend. These two observations were noted above in part (a).

d) Experiment with making the trend exponential and/or damped.

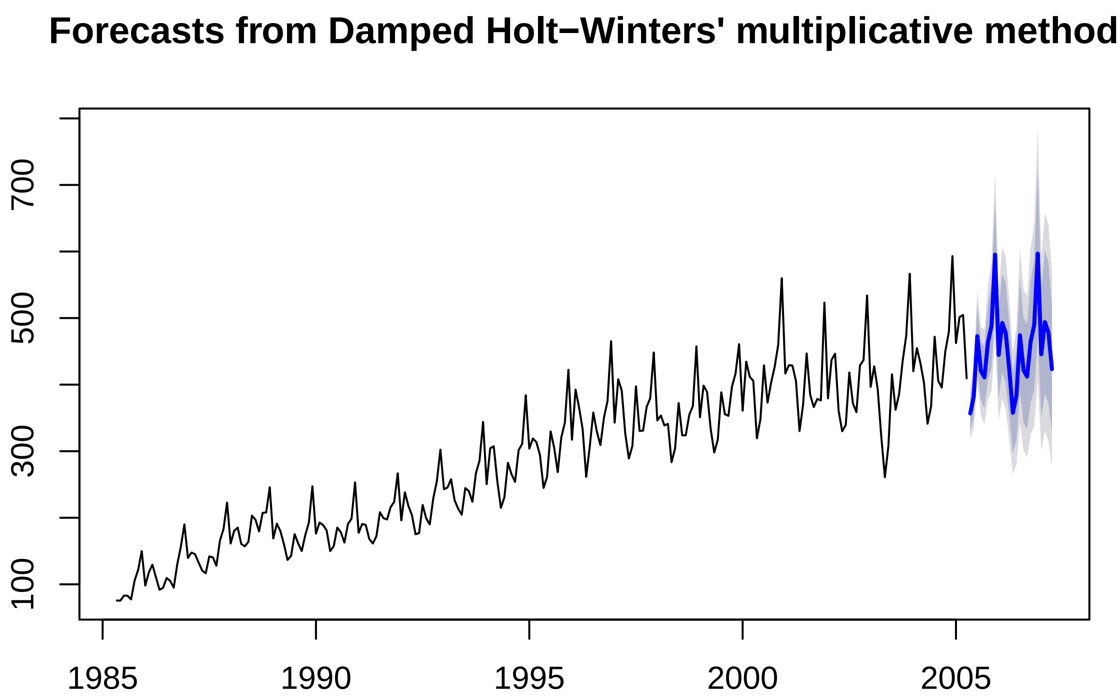
```r
fc1 <- hw(visitors,seasonal="mult",exponential=TRUE)
fc2 <- hw(visitors,seasonal="mult",exponential=TRUE, damped=TRUE)
fc3 <- hw(visitors,seasonal="mult",damped=TRUE)

plot(fc1)
```




```
plot(fc2)
```




```
plot(fc3)
```



e) Compare the RMSE of the one-step forecasts from the various methods. Which do you prefer?

```r
accuracy(fc)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.8614726 14.52211 10.86884 -0.4799156 4.168399 0.4013761
##                     ACF1
## Training set -0.03448764
```

```r
accuracy(fc1)
```

```
##                      ME    RMSE      MAE        MPE     MAPE     MASE
## Training set -0.6175624 14.6899 11.00618 -0.3558085 4.230296 0.406448
##                    ACF1
## Training set 0.08654357
```

```r
accuracy(fc2)
```

```
##                     ME     RMSE      MAE         MPE     MAPE      MASE
## Training set 0.5595893 14.46091 10.66091 -0.07611252 4.075176 0.3936972
##                    ACF1
## Training set -0.0268311
```

```r
accuracy(fc3)
```

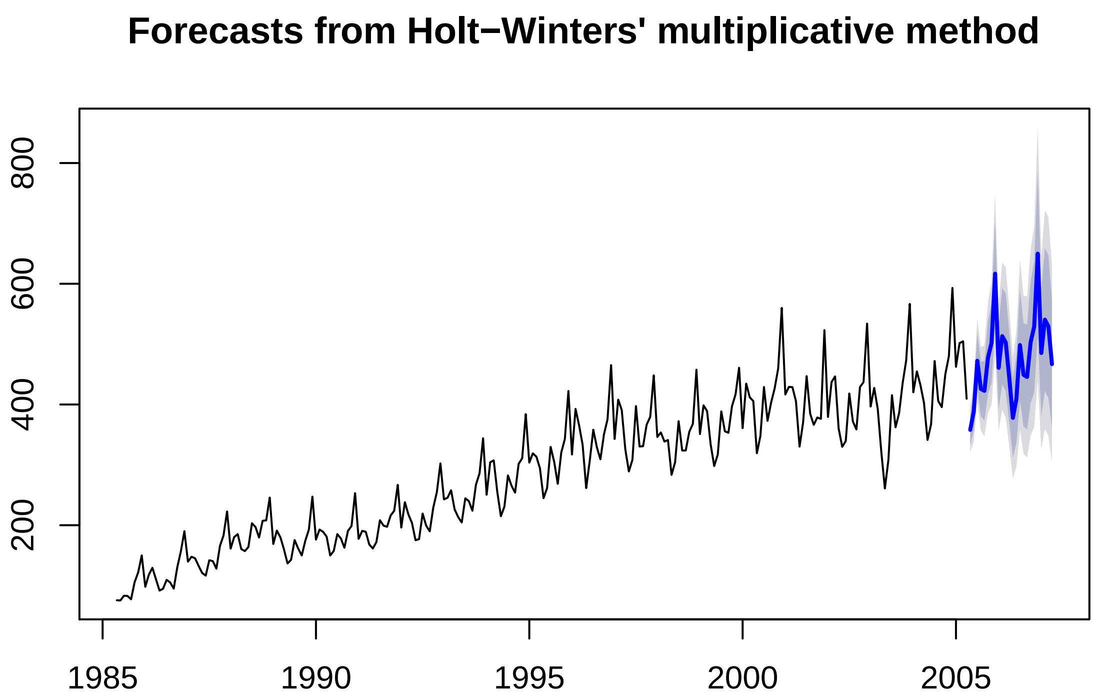
```
##                   ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 1.523643 14.40219 10.64283 0.3591333 4.057262 0.3930297
##                    ACF1
## Training set 0.01526565
```

Additive damped trend (fc3) seems to do best (lowest RMSE) amongst these models. However, it has one more parameter (the damping parameter) than the non-damped version. The damped explonential does better than the non-damped model.

g) Now fit each of the following models to the same data:

multiplicative Holt-Winters' method

```
fc <- hw(visitors,seasonal="mult")
plot(fc)
```



ETS model

```
etsfit <- ets(visitors)
etsfit
```

```
## ETS(M,A,M)
##
## Call:
##  ets(y = visitors)
##
##   Smoothing parameters:
##     alpha = 0.6244
##     beta  = 1e-04
##     gamma = 0.1832
##
##   Initial states:
##     l = 86.3534
##     b = 2.0306
##     s=0.942 1.076 1.0515 0.9568 1.3621 1.1157
##            1.011 0.8294 0.9336 1.0017 0.8649 0.8554
```

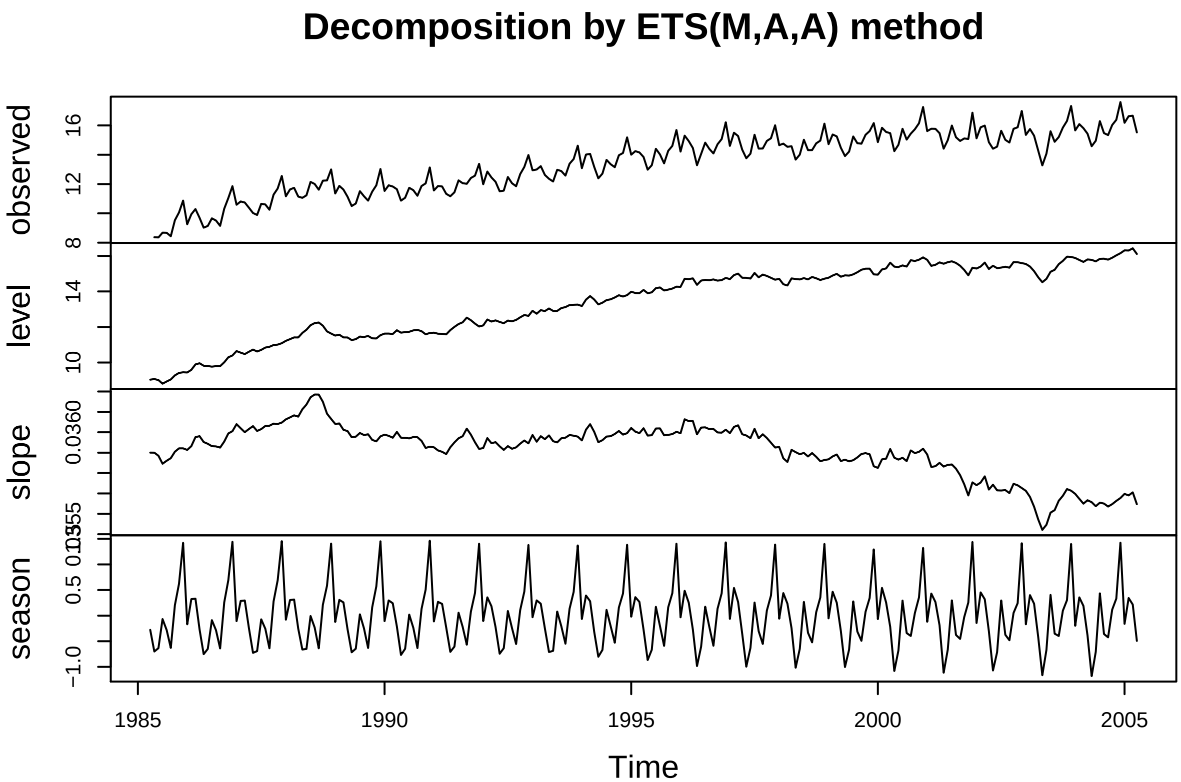
```
##
##   sigma:  0.0515
##
##      AIC     AICc      BIC
## 2598.193 2600.632 2653.883
```

additive ETS model applied to a Box-Cox transformed series

```r
lambda <- BoxCox.lambda(visitors)
boxcox <- (BoxCox(visitors,lambda))

etsboxcox <- ets(boxcox)

etsboxcox
```

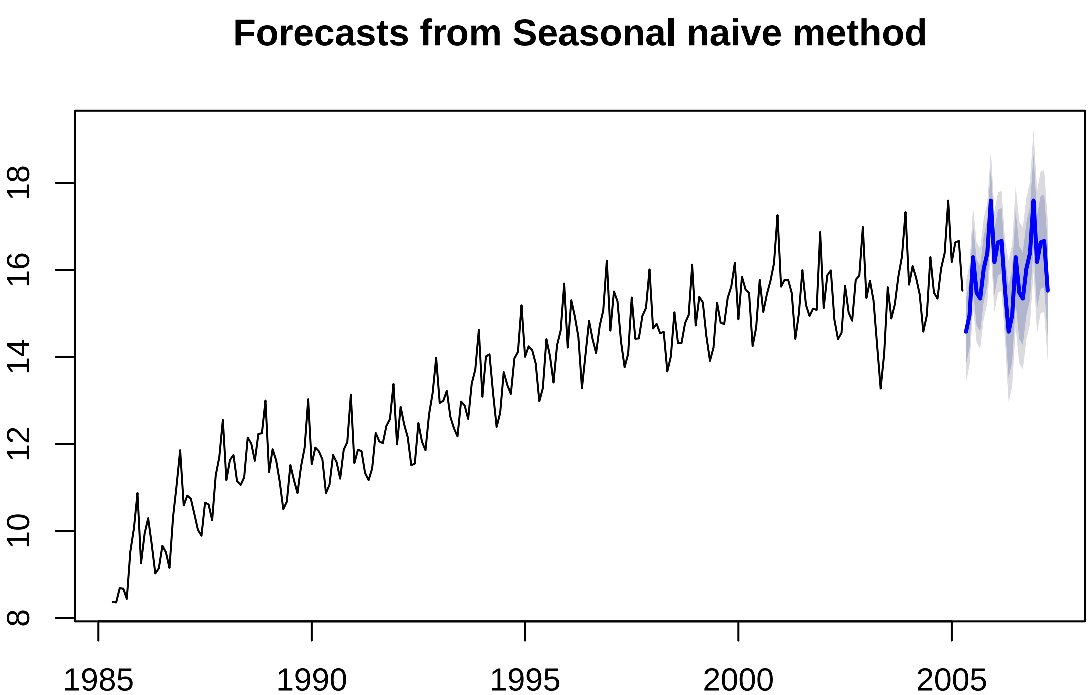
```
## ETS(M,A,A)
##
## Call:
##  ets(y = boxcox)
##
##   Smoothing parameters:
##     alpha = 0.6467
##     beta  = 1e-04
##     gamma = 0.1909
##
##   Initial states:
##     l = 9.0365
##     b = 0.0359
##     s=-0.2782 0.2511 0.2887 -0.1556 1.4204 0.5913
##            0.1475 -0.6527 -0.3085 0.0034 -0.6075 -0.7
##
##   sigma:  0.0179
##
##      AIC     AICc      BIC
## 657.3674 659.8068 713.0576
```

```r
plot(etsboxcox)
```



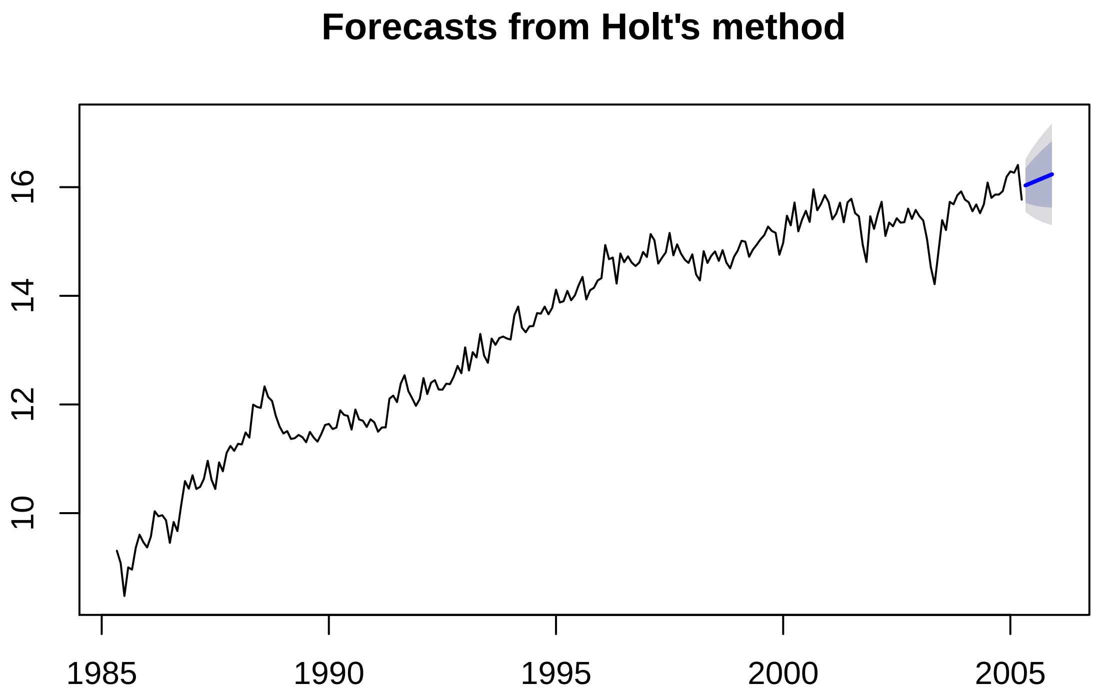
seasonal naive method applied to the Box-Cox transformed series

```
snaive <- snaive(boxcox, h=24)

plot(snaive)
```



STL decomposition applied to the Box-Cox transformed data

```r
stlbc <- stl(boxcox, s.window="periodic", robust=TRUE)

boxcox <- seasadj(stlbc)
decompboxcox <- holt(boxcox, h=8)

plot(decompboxcox)
```

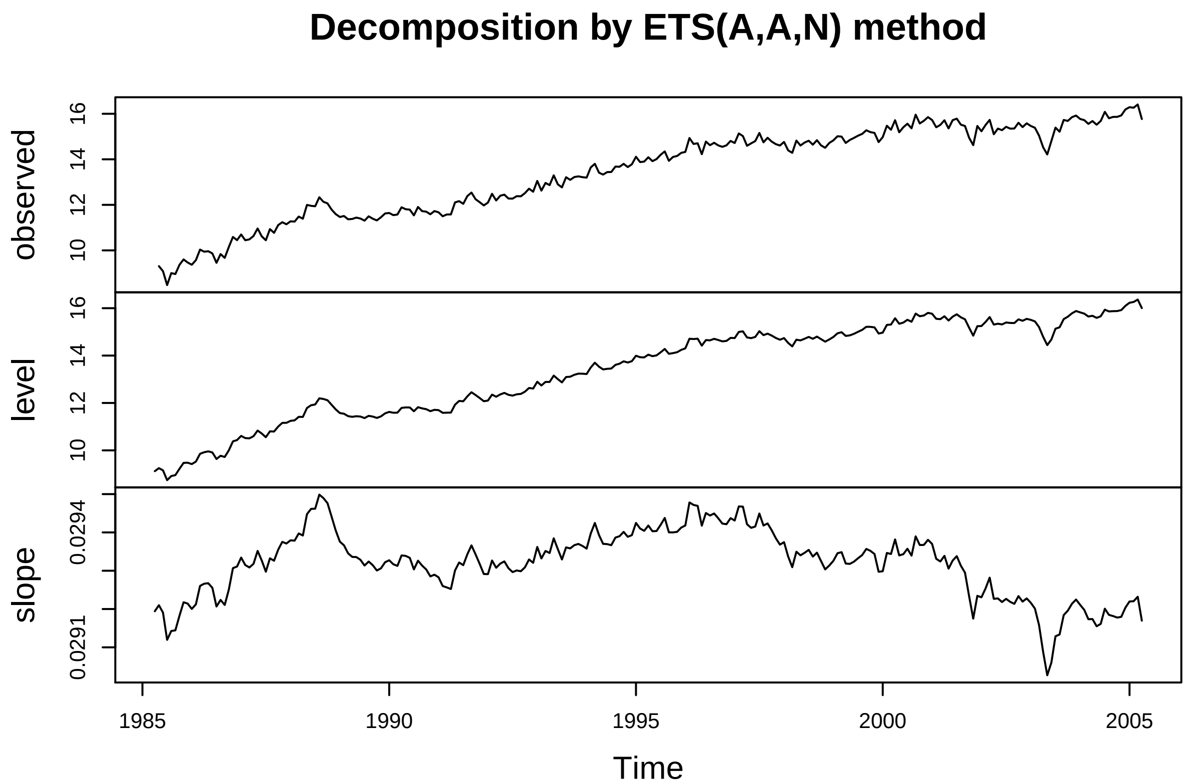


ETS model applied to the seasonally adjusted (transformed) data.

```r
etsboxcox2 <- ets(boxcox)
etsboxcox2
```

```
## ETS(A,A,N)
##
## Call:
##  ets(y = boxcox)
##
##   Smoothing parameters:
##     alpha = 0.6252
##     beta  = 1e-04
##
##   Initial states:
##     l = 9.1203
##     b = 0.0292
##
##   sigma:  0.247
##
##      AIC     AICc      BIC
## 652.2022 652.3724 666.1247
```

```r
plot(etsboxcox2)
```



g) For each model, look at the residual diagnostics and compare the forecasts for the next two years. Which do you prefer?

```r
accuracy(fc)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.8614726 14.52211 10.86884 -0.4799156 4.168399 0.4013761
##                     ACF1
## Training set -0.03448764
```

```r
accuracy(etsfit)
```

```
##                      ME   RMSE     MAE        MPE     MAPE      MASE
## Training set -0.9564743 15.847 11.5215 -0.4307078 4.075378 0.4254781
##                    ACF1
## Training set 0.02434609
```

```r
accuracy(etsboxcox)
```

```
##                        ME      RMSE       MAE         MPE     MAPE
## Training set -0.009975438 0.2413335 0.1886967 -0.07603778 1.415937
##                  MASE      ACF1
## Training set 0.385689 0.0143794
```

```r
accuracy(snaive)
```

```
##                     ME      RMSE       MAE      MPE     MAPE MASE
## Training set 0.3486245 0.5883754 0.4892459 2.717695 3.768233    1
##                   ACF1
## Training set 0.7406243
```

```r
accuracy(decompboxcox)
```

```
##                        ME      RMSE       MAE          MPE     MAPE
## Training set -0.001023284 0.2470334 0.1881132 -0.008466536 1.436249
##                   MASE       ACF1
## Training set 0.3844962 0.02626461
```

```r
accuracy(etsboxcox2)
```

```
##                        ME      RMSE       MAE          MPE     MAPE
## Training set -0.001026554 0.2470334 0.1881128 -0.008493268 1.436247
##                   MASE       ACF1
## Training set 0.3844955 0.02628152
```

The RMSE for the ETS fits are very low, as well as the STL decomp and snaive. This might be as a result of the Box-Cox Transformation. I personally prefer the ETS model because it chooses for you, but the graphics are not as easy to read as compared to the STL decomp and snaive plots.