Time Series Analysis

Computer Aided Lab 3

This is an ungraded lab sheet. Graded work will appear under Assignments.

---

1 Decomposition

Examine the AirPassengers data that we met in the lecture. This dataset is pre-loaded in `R`. This is monthly totals of international airline passengers, 1949 to 1960. For convenience we will set y as AirPassengers.

```
y=AirPassengers
```

Begin, as always, by plotting the series along with the ACF and PACF using **tsdisplay** from the `forecast` package.

```
library(forecast)
tsdisplay(y)
```

We can run an Augmented Dickey-Fuller test to check for unit roots, using the package `tseries`:

```
library(tseries)
adf.test(y)
```

We obtain a p-value here of less than 0.01 so we reject the null hypothesis that $\omega = 0$ i.e. $\phi_1 = 1$. Thus there is no unit root and the series does not require differencing to get to stationarity.

Clearly visible is a linear trend and a seasonal component. Despite the high autocorrelation across all lags due to the linear trend, we can still read off the seasonal period of 12 months from the ACF plot.

We can estimate the trend component $\hat{TC}_t$ by smoothing out the seasonal component first. **ma()** from the forecast package does this for us:

```
TC=ma(y,12)
```

For forecasting purposes we'd like to fit a linear model to this trend. You can do this as follows.

```
linear_tc=lm(TC~time(y))
```

We can get an even clearer picture of the seasonal period by plotting the de-trended time series.

```
tsdisplay(y-TC)
```

Interestingly this works even if we use the wrong order moving average smooth in the trend estimation.

Assuming a multiplicative model (as per the lecture)

$$\text{AirPassengers}_t = TC_t S_t R_t,$$

we can now de-trend the time series by dividing the time series by $\hat{TC}_t$. Taking the average of the resulting series for corresponding months gives us an estimate of the seasonal component. There are 144 timepoints in the series meaning 12 years to average over:

$$\hat{S}_i = \frac{1}{12}\sum_{j=0}^{11}\frac{\text{AirPassengers}_{12j+i}}{\hat{TC}_{12j+i}}.$$

So for example,

$$\hat{S}_{jan} = \frac{1}{12}\left(\frac{\text{AirPassengers}_1}{\hat{TC}_1} + \frac{\text{AirPassengers}_{13}}{\hat{TC}_{13}} \ldots + \frac{\text{AirPassengers}_{25}}{\hat{TC}_{25}}\right)$$

In R we can do this quickly for all 12 seasonal parts (months) by creating a matrix of $\frac{\text{AirPassengers}}{\hat{TC}}$ and then taking the row means.

```
pseudo_s = y/TC
matrix_s = matrix(pseudo_s,nrow=12)
S = rowMeans(matrix_s,na.rm=TRUE)
```

This being a multiplicative model we can centre our seasonal adjustments by making them have a mean of 1:

```
S = S/mean(S)
```

Now that we have both the estimated trend and estimated seasonal component, we can estimate the random component:

```
R=y/(TC*rep(S,12))
```

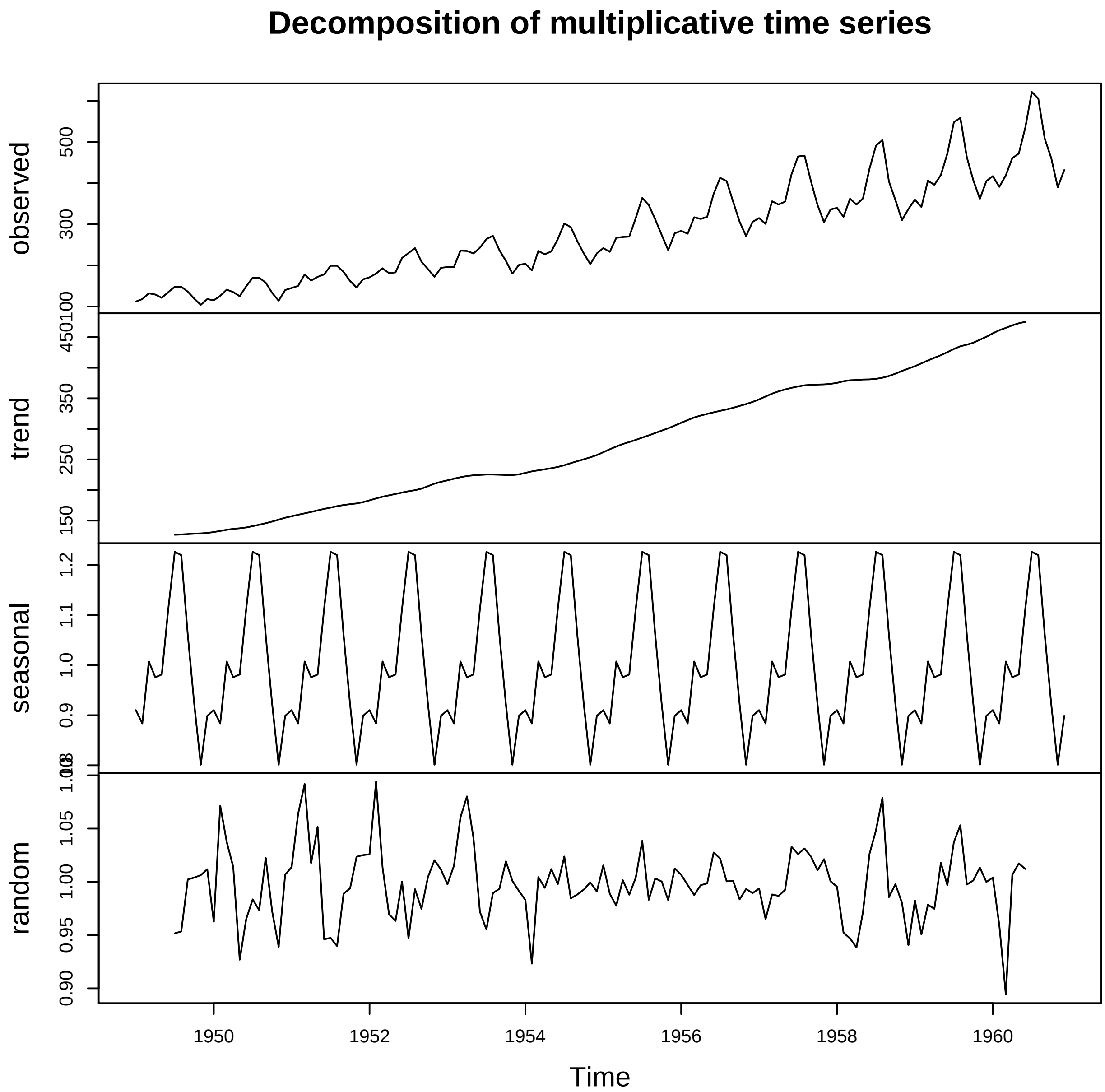
Note that to get all 12 years we simply repeat our estimated seasonal components 12 times.

- **Now explore the use of the decompose() function.** You need to set **type='mult'**:

  ```
  m_decomp=decompose(y,type="mult")
  ```

  This should give you the same estimated trend, seasonal, and random components, which you can also plot:

  ```
  plot(m_decomp)
  ```



2 Forecasting

Forecasting using a fitted ARIMA time series in R is straightforward. You can simply use **forecast()** function from the `forecast` package. The input should be a fitted Arima object, which is the output of using the **Arima()** function in the `forecast` package. You can also set **h** as the number of future times to forecast ahead.

To predict a decomposed time series, you need to predict the deterministic trend and use ARIMA forecasting for the random component.

- **Use tsdisplay(), etc to determine the ARIMA specification of the residuals of the above decomposition.** You should find an ARMA(2,1) model, as ARIMA diagnostics fail until (p,d,q) = (2,0,1) and all parameters are significant for this model.

```
fit=arima(R,order=c(2,0,1))
library(lmtest) # so we can test coefficients for statistical significance
coeftest(fit)
res=residuals(fit)
tsdisplay(res)
Box.test(res,lag=10,t='Ljung-Box')
```

where res are the residuals of the fitted

Now try forecasting your decomposed multiplicative model.

- **Extrapolate the expected (trend) X (seasonal components) 2 years ahead.**

```
ftime=seq(1961,length=24,by=deltat(y)) # 24 future months
d_fy=(linear_tc$coef[1]+linear_tc$coef[2]*ftime)*rep(S[1:12],2)
```

Note that **deltat()** simply extracts the time gap between successive times in the series.

- **Use forecast() to predict 2 years ahead for the random component.**

```
fit=Arima(R,order=c(2,0,1), include.mean=T)
fy=forecast(fit,h=30) # forecasts 6 missing months plus 24 months more
```

The reason for the "missing" 6 months above is that the random component cannot be estimated for the final 6 months of observations as the seasonal component there would depend on averaging over 6 months into the future.

- **Multiply the trend prediction, seasonal prediction, and random prediction mean to get your mean forecast.**

```
msef=d_fy*fy$mean[-c(1:6)]
```

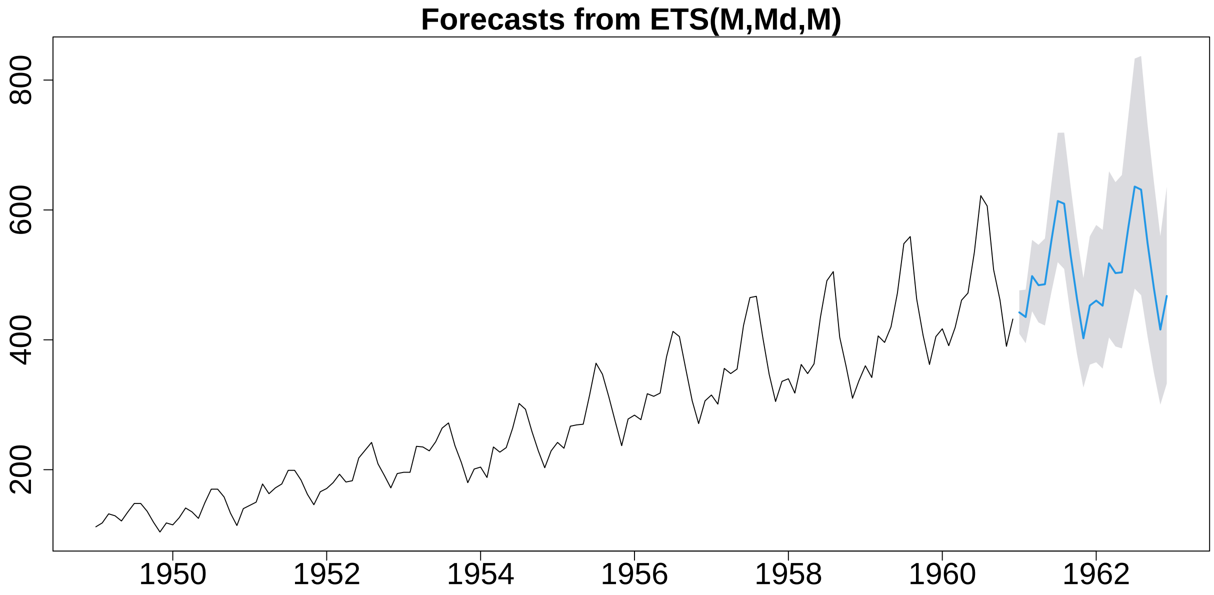
We use $-c(1:6)$ here to skip the final 6 months that are included in the observations but forecast by the **forecast()** function.

- **You can now plot the forecast as follows:**

```
plot(y,xlim=c(time(y)[1],ftime[length(ftime)]),ylim=c(0,700))
lines(ftime, msef, col=4)
```

- **You can obtain a similar result using ets(), that includes confidence intervals and does the decomposition too**:

```
m_ets=ets(y,model="MMM")
m_forecast=forecast(ets(y,model="MMM",damped=T),level=95) # 95% confidenc
plot(m_forecast)
```



Practice Exercises:

Do try these before looking at the solutions on the next page!

Exercise 1. Decompose the air passengers data, this time using an **additive** model. Check that you can get the same results by adapting the code in part 1 above (or by using the decompose function).

Exercise 2. Create a forecast using the additive model for the 6 months following the end of the observation period. Compare this to the multiplicative model results. How do they differ? Which do you prefer?

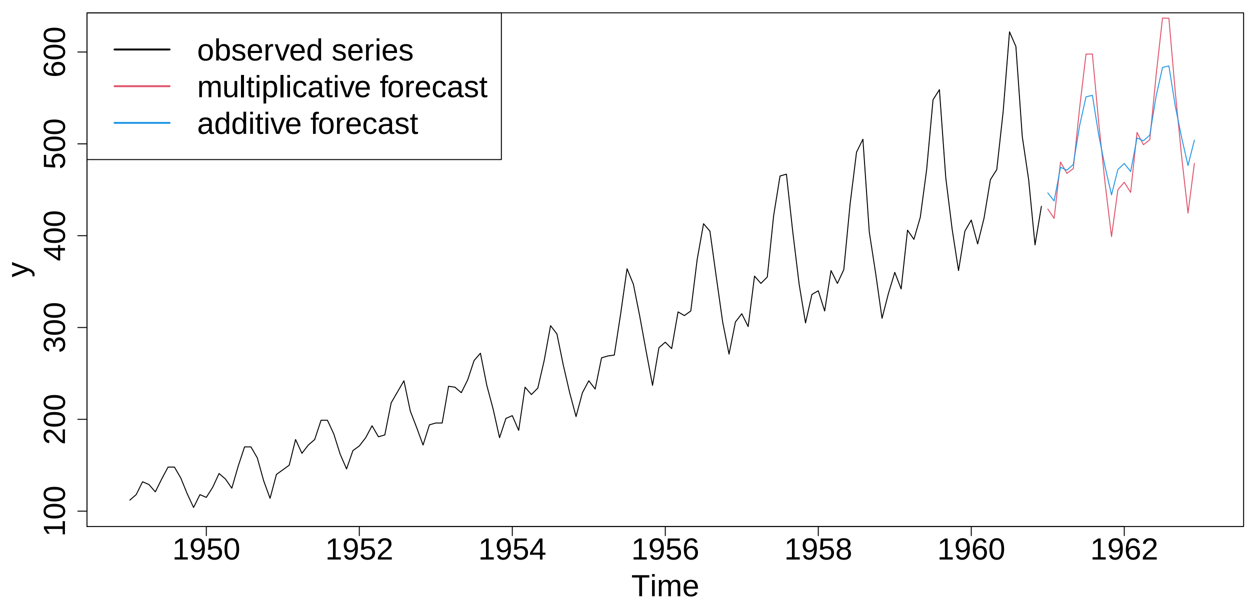
Practice Exercises: Solutions

Exercise 1. Decompose the air passengers data, this time using an **additive** model. Check that you can get the same results by adapting the code in part 1 above (or by using the decompose function).

```
a_pseudo_s = y-TC
a_matrix_s = matrix(a_pseudo_s,nrow=12)
a_S = rowMeans(a_matrix_s,na.rm=TRUE)
a_S = a_S-mean(a_S)
a_R=y-(TC+rep(a_S,length(y)/12))
a_decomp=decompose(y,type="add") # gets the same as above
```

Exercise 2. Create a forecast using the additive model for the 6 months following the end of the observation period. Compare this to the multiplicative model results. How do they differ? Which do you prefer?

```
a_d_fy=(linear_tc$coef[1]+linear_tc$coef[2]*ftime)+rep(a_S[1:12],2)
a_fit=Arima(a_R,order=c(2,0,1), include.mean=T)
a_fy=forecast(fit,h=30) # 24 months plus the 6 missing from the end
asef=a_d_fy+a_fy$mean[-c(1:6)]
plot(y,xlim=c(1949,1963))
lines(ftime, msef, col=2)
lines(ftime, asef, col=4)
legend("topleft", c("observed","multiplicative","additive"),
       col=c(1,2,4), lty=c(1,1,1),lwd=2)
```



Multiplicative looks better, the seasonal variation in the additive model looks too small as they don't get multiplied by the increasing trend component.