Time Series Assignment 2

Matthew Dunne

June 30, 2018

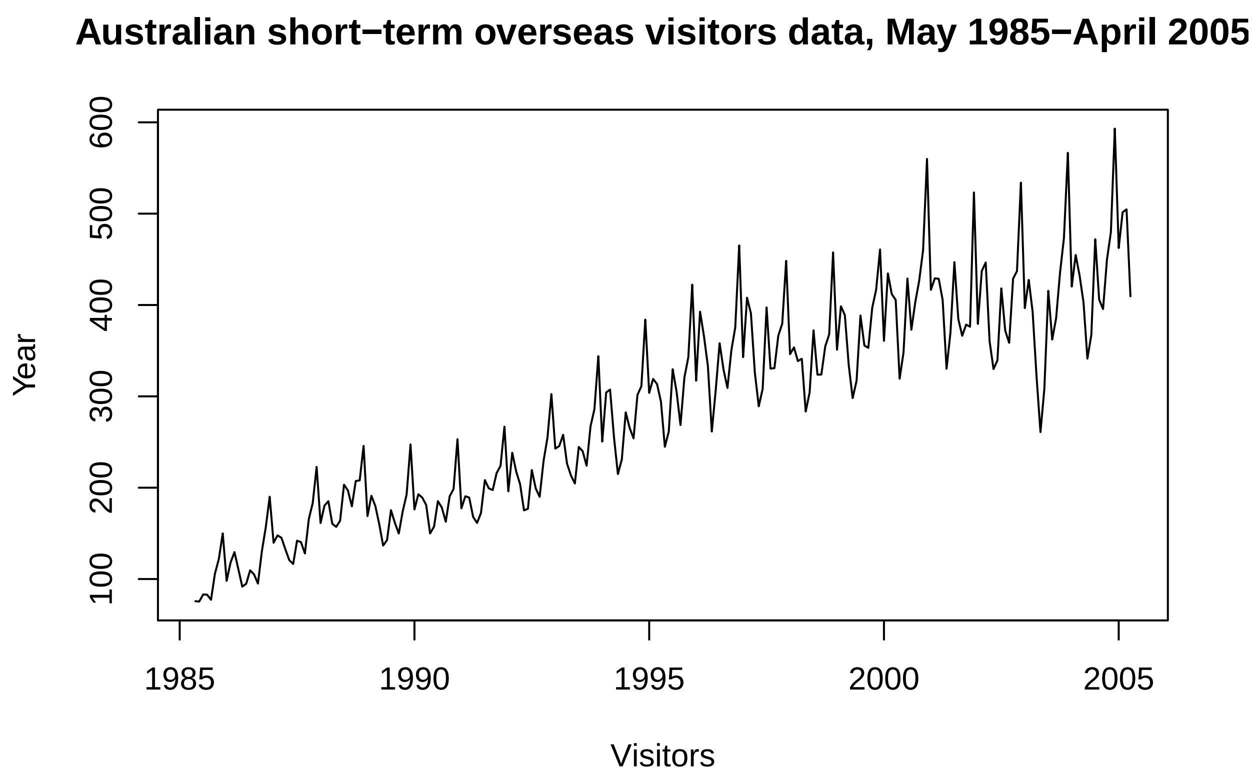
Question 1:

Load the visitors.rda dataset, make a time plot of your data and describe the main features of the series.

```
load("C:/Users/mjdun/Desktop/Time Series/Assignments/visitors.rda")
suppressMessages(library(fpp))
data<-visitors
head(data)
```

```
##        May   Jun   Jul   Aug   Sep   Oct
## 1985  75.7  75.4  83.1  82.9  77.3 105.7
```

```
plot(data, xlab="Visitors", ylab="Year", main="Australian short-term overseas visitors data, May 1985-Ap
```



Main Features:
This data has a pronounced upward trend and seasonality. There is some noise, meaning the pattern of moving from month to month looks the same generall but is not always exactly uniform.

Question 2:

What is the appropriate Holt-Winters method for this data set (multiplicative / additive)? why?

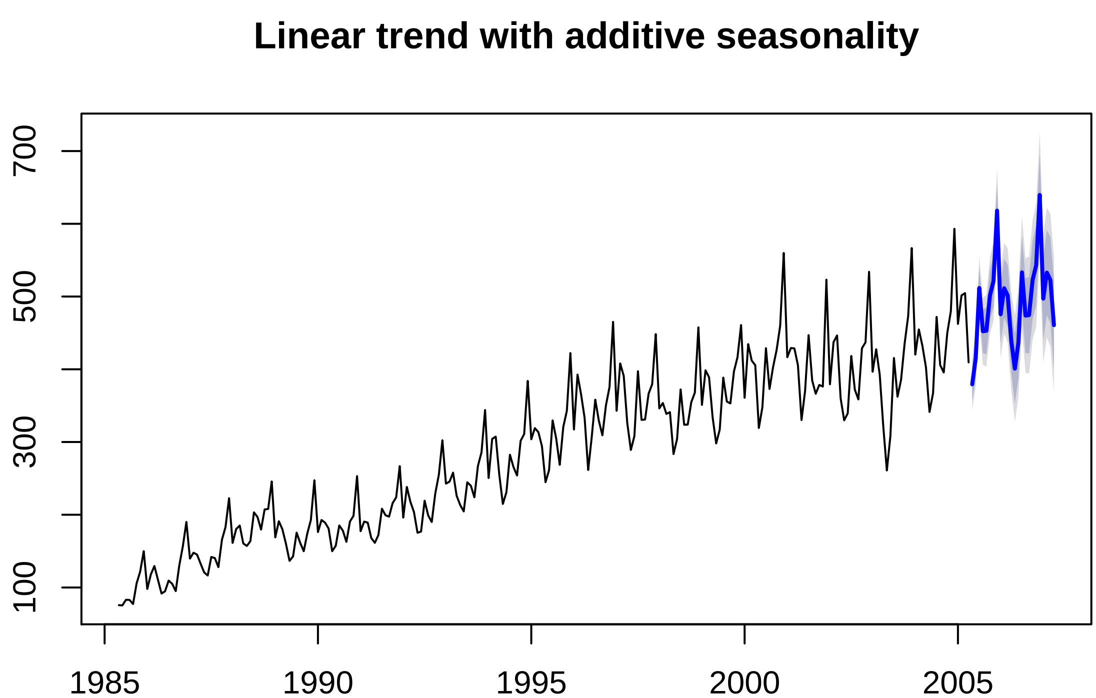
The multiplicative method is appropriate. The amplitude of the seasonal pattern varies (is proportional to) with the average level within the season. In other words the difference between the highest point and the lowest point within a season gets bigger as time goes on.

Question 3 Use the hw() function to forecast the next 15 months using Holt-Winters' methods.

Experiment with the following methods:

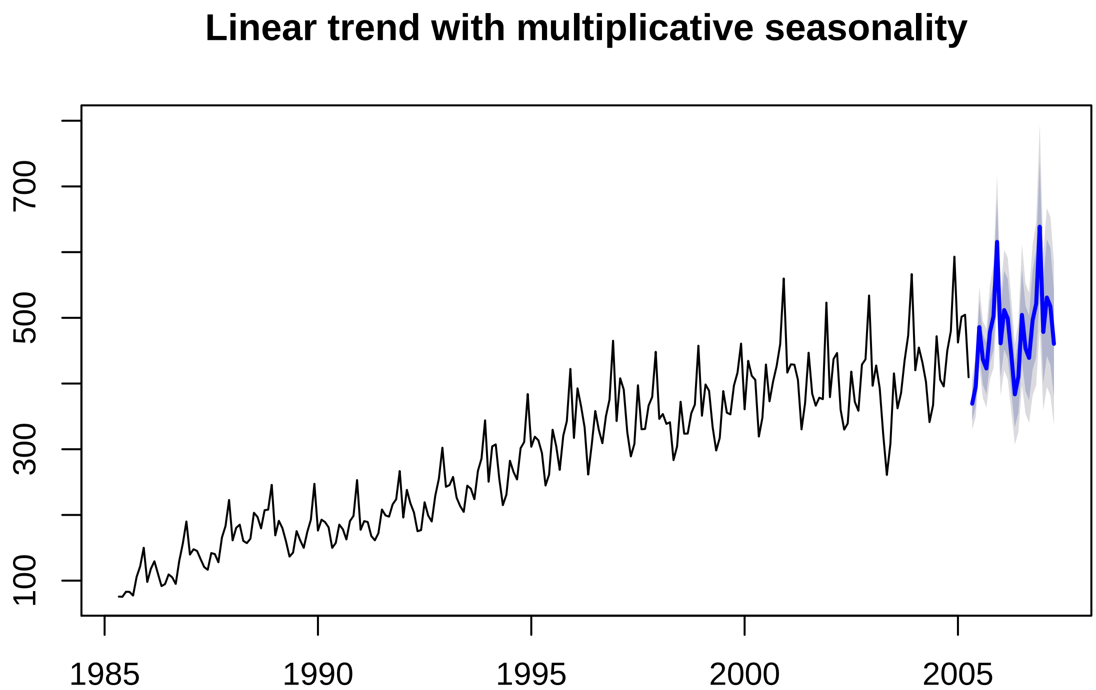
- Linear trend with additive seasonality

```
method1<-hw(data, seasonal = "additive", h=24)
plot(method1, main = "Linear trend with additive seasonality")
```



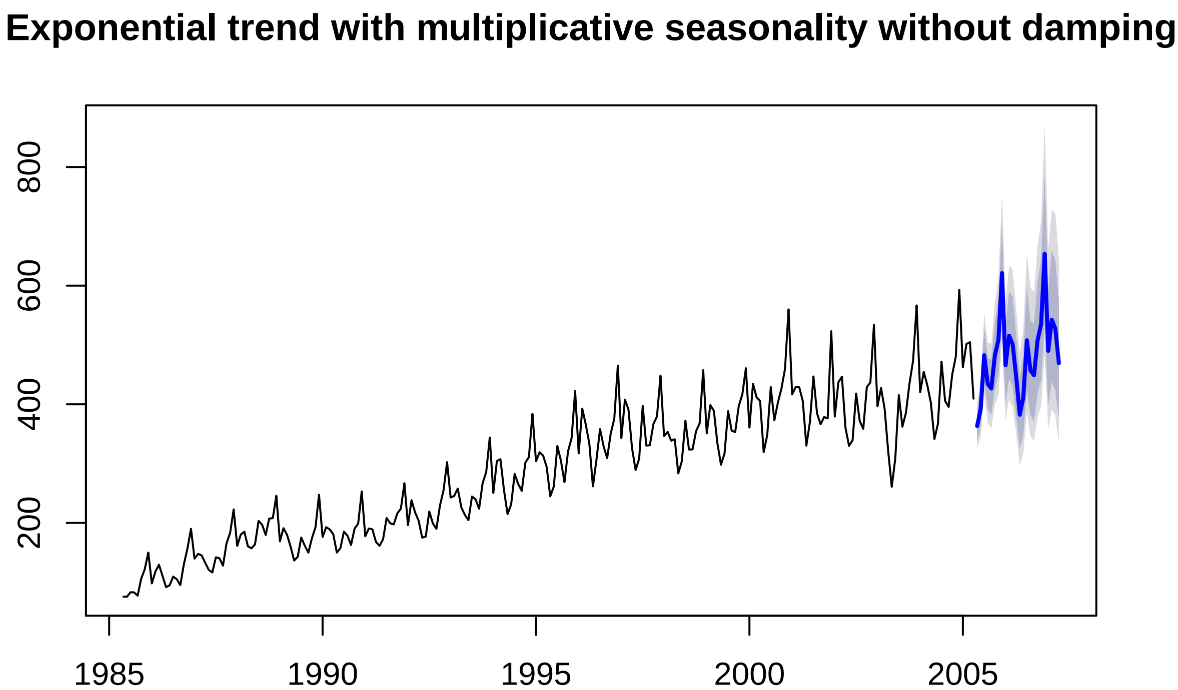
- Linear trend with multiplicative seasonality

```
method2<-hw(data, seasonal = "multiplicative", h=24)
plot(method2, main="Linear trend with multiplicative seasonality")
```



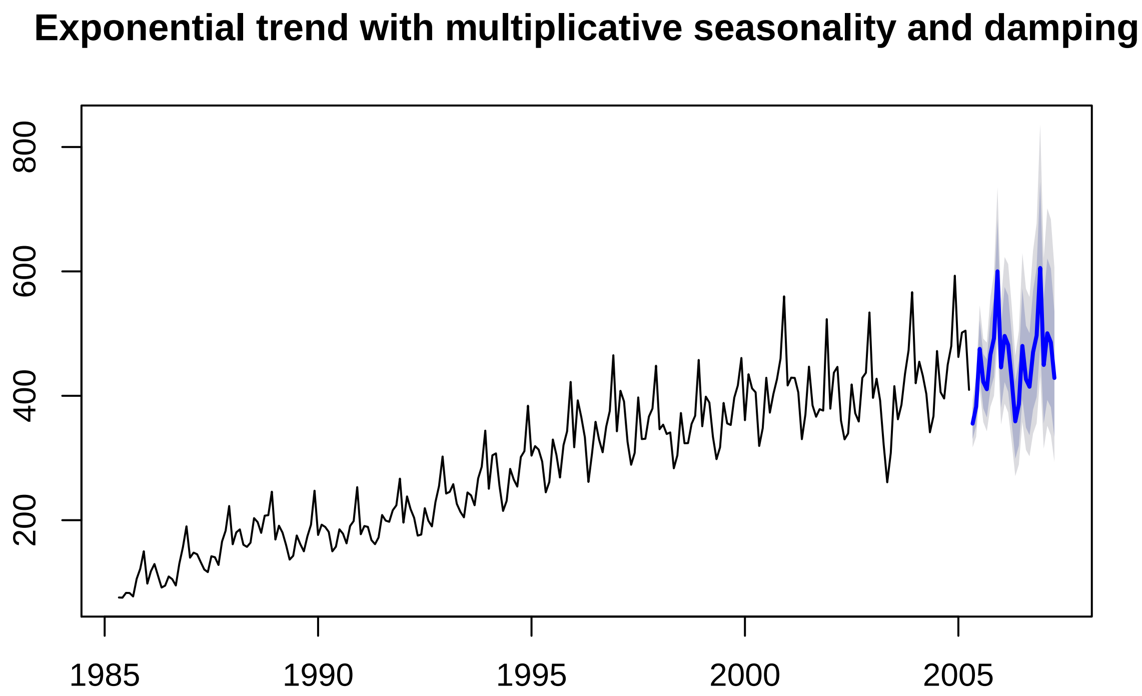
- Exponential trend with multiplicative seasonality without damping

```
method3<-hw(data, seasonal = "multiplicative", exponential = TRUE)
plot(method3, main="Exponential trend with multiplicative seasonality without damping")
```



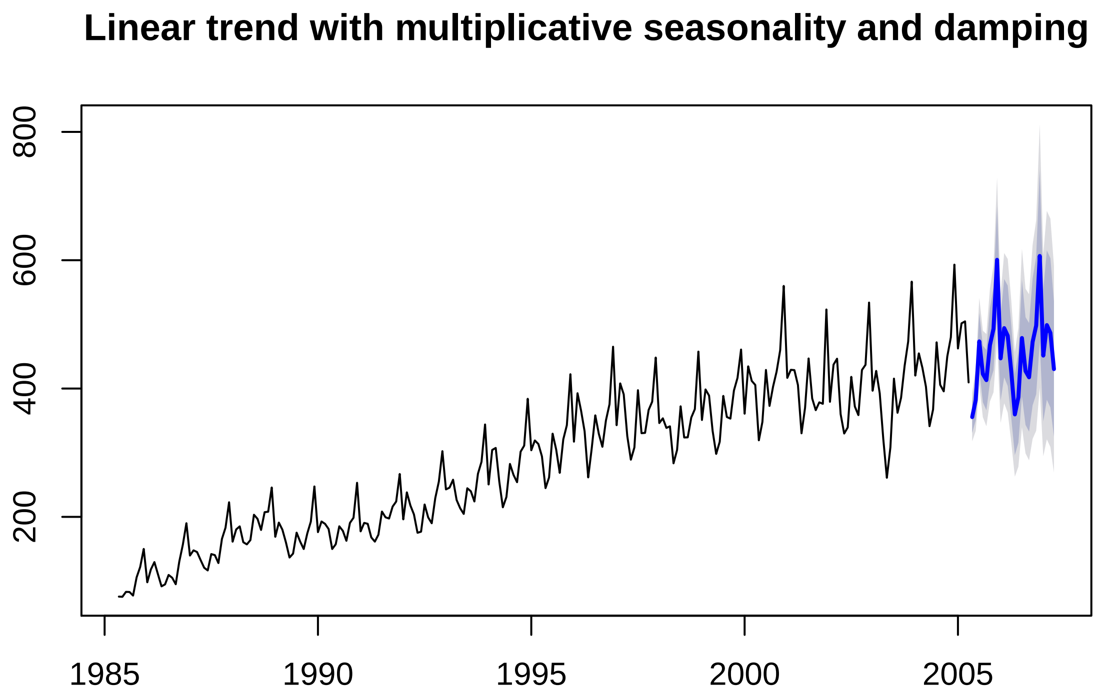
- Exponential trend with multiplicative seasonality and damping

```
method4<-hw(data, seasonal = "multiplicative", exponential = TRUE, damped = TRUE)
plot(method4, main="Exponential trend with multiplicative seasonality and damping")
```



- Linear trend with multiplicative seasonality and damping

```
method5<-hw(data, seasonal = "multiplicative", h=24, damped = TRUE)
plot(method5, main="Linear trend with multiplicative seasonality and damping")
```



Question 4:

Use the accuracy() function to compare the Root-Mean-Square-Error (RMSE) values of the forecasts from the various methods. Which do you prefer and why?

```
suppressMessages(library(fpp))
accuracy(method1)[[2]]
```

```
## [1] 17.98198
```

```
accuracy(method2)[[2]]
```

```
## [1] 14.8295
```

```
accuracy(method3)[[2]]
```

```
## [1] 14.49416
```

```
accuracy(method4)[[2]]
```

```
## [1] 14.45533
```

```
accuracy(method5)[[2]]
```

```
## [1] 14.44801
```

The first two methods (linear additive and linear multiplicative) can be eliminated from consideration based on the RMSE. Of the other three methods, the RMSE is essentially the same so I would be inclined to choose the model that is the simplest and most in line with what we believe about the underlying data.

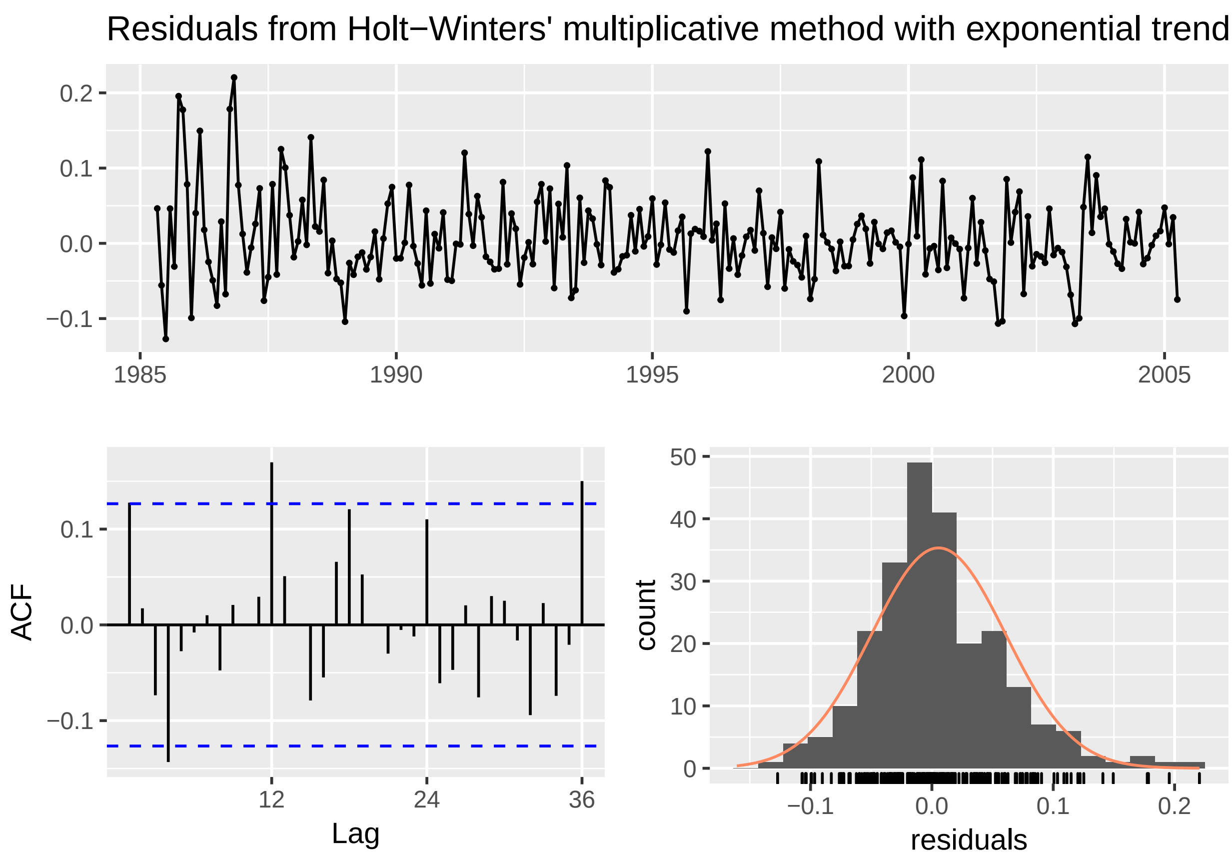
Given that the worst performing model of our 5 is linear (model 1) I would reject model 5 (linear, multiplicative, damping). And given there is no particular reason to expect damping I would reject model 4 (exponential, multiplicative, damping). **So I choose model 3 which assumes an exponential trend, multiplicative seasonality, and no damping.**

Question 5:

Use the checkresiduals() function to check that the residuals from the best model look like white noise and provide a summary of the model's smoothing parameters using the summary() function.

Our "best" model was model 3 (exponential, multiplicative, no damping). As detailed below, **the model's residuals are not white noise.**

```
checkresiduals(method3)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' multiplicative method with exponential trend
## Q* = 31.144, df = 8, p-value = 0.0001324
##
## Model df: 16.   Total lags used: 24
```

```
summary(method3$model)
```

```
## Holt-Winters' multiplicative method with exponential trend
##
```

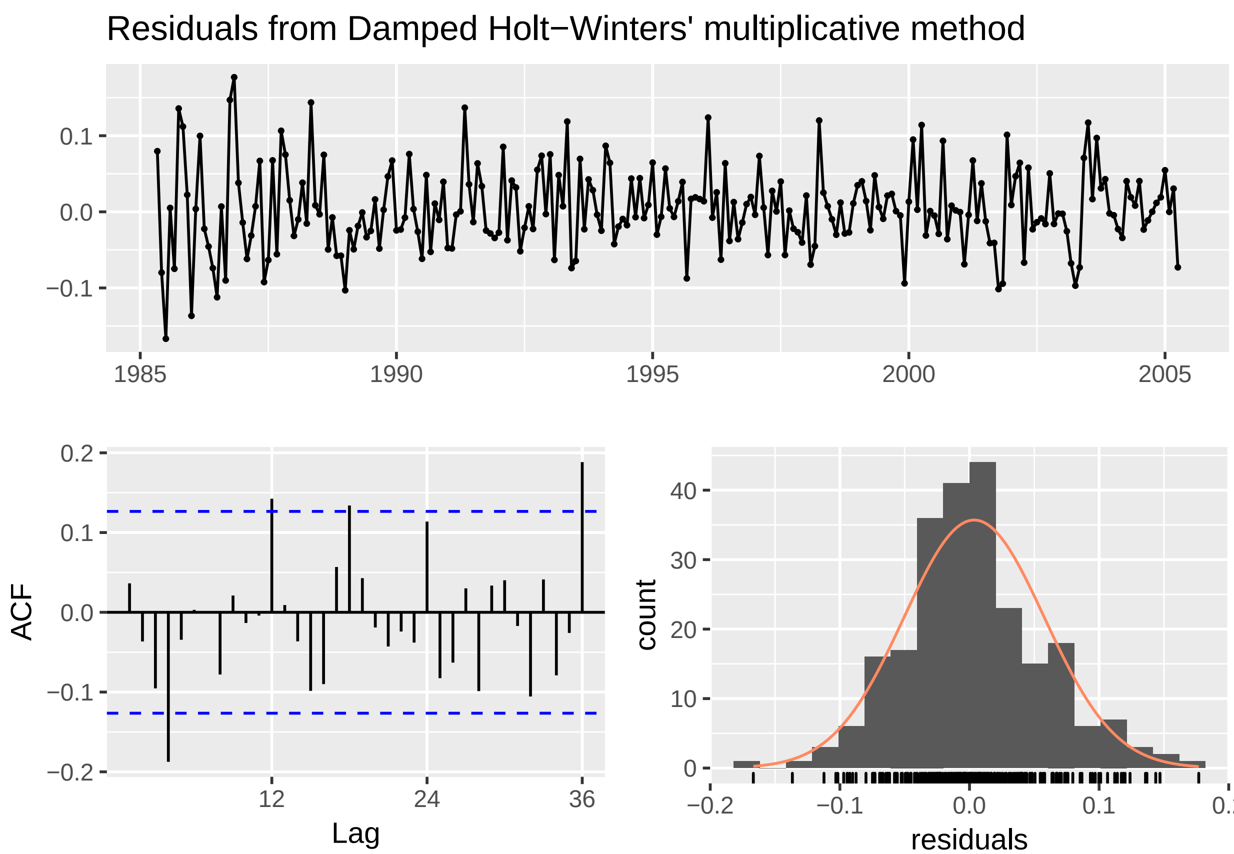
```
## Call:
##  hw(y = data, seasonal = "multiplicative", exponential = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.5722
##     beta  = 0.0013
##     gamma = 1e-04
##
##   Initial states:
##     l = 91.0884
##     b = 1.0025
##     s=0.9278 1.0475 1.0821 0.9815 1.3152 1.0813
##            1.0294 0.9145 0.9348 1.0438 0.8497 0.7923
##
##   sigma:  0.0556
##
##      AIC     AICc      BIC
## 2633.767 2636.524 2692.938
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE   MAPE      MASE
## Training set 0.6442177 14.49416 10.62951 0.2554469 4.0328 0.3925378
##                    ACF1
## Training set 0.07595792
```

The Ljung-Box Test score and p-value lead us to reject the null hypothesis that the errors are independently distributed (white noise). **The residuals are not white noise.** We see this supported in the graphs, where the ACF exceeds the 95% CI in several places.

We see that this holds true even when we check the model that technically had the lowest RMSE (model 5).

Model 5 (linear, multiplicative, damping)

```r
checkresiduals(method5)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Damped Holt-Winters' multiplicative method
## Q* = 34.254, df = 7, p-value = 1.544e-05
##
## Model df: 17.   Total lags used: 24
```

```
summary(method5$model)
```

```
## Damped Holt-Winters' multiplicative method
##
## Call:
##  hw(y = data, h = 24, seasonal = "multiplicative", damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.6306
##     beta  = 0.0071
##     gamma = 1e-04
##     phi   = 0.9797
##
##   Initial states:
##     l = 85.7688
##     b = 3.4912
##     s=0.9328 1.0558 1.0829 0.9805 1.3187 1.0838
##            1.029 0.9097 0.9317 1.0447 0.8442 0.7861
##
```

```
##   sigma:  0.0542
##
##      AIC     AICc      BIC
## 2624.818 2627.913 2687.469
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 0.9123468 14.44801 10.64909 0.07071844 4.064322 0.3932608
##                    ACF1
## Training set 0.01740636
```