```
library(fpp)

##########################################
# Example 1
ukcars
plot(ukcars, ylab = "Production, thousands of cars")

# The data is seasonal
# Some nonlinear trend

#Decomposition of the series using STL
stlFit <- stl(ukcars, s.window = "periodic")
plot(stlFit)
adjusted <- seasadj(stlFit)
plot(adjusted)


#Two years forecast of the series using an additive damped trend method
# applied to the seasonally adjusted data.
# Then the forecasts are reseasonalized.
fcastHoltDamp = holt(adjusted, damped=TRUE, h = 8)
plot(ukcars, xlim = c(1977, 2008))
lines(fcastHoltDamp$mean + stlFit$time.series[2:9,"seasonal"], col = "red", lwd = 2)

#RMSE of one step forecasts
dampHoltRMSE = sqrt(mean(((fcastHoltDamp$fitted + stlFit$time.series[,"seasonal"]) - ukcars)^2))
dampHoltRMSE


#Two years forecast of the series using Holt's linear trend method applied
# to the seasonally adjusted data. Then the forecasts are reseasonalized.
fcastHolt = holt(adjusted, h = 8)
plot(ukcars, xlim = c(1997, 2008))
lines(fcastHolt$mean + stlFit$time.series[2:9,"seasonal"], col = "red", lwd = 2)

#### RMSE of one step forecasts
holtRMSE = sqrt(mean(((fcastHolt$fitted + stlFit$time.series[,"seasonal"]) - ukcars)^2))
holtRMSE

### Two years forecast of the series using ets method
etsFit = ets(ukcars)
fcastEts = forecast(etsFit, h = 8)
plot(fcastEts)

#### RMSE of one step forecasts
etsRMSE = sqrt(mean((fcastEts$fitted - ukcars)^2))
etsRMSE

### RMSE of the Holt's method and ets forecasts
c(HoltRMSE = holtRMSE, EtsRMSE = etsRMSE)

### Comparison of the Holt's linear trend and ets methods

# The forecasts of the Holt's linear trend and ets methods look very similar
# and therefore it is very difficult to judge which one is better.
# ets method has slightly worse fit to the training data than
# the Holt's linear method. But the forecasts are very similar and it is hard to pick between them.




###########################################
# Example 2

# usnetelec series
plot(usnetelec)
Acf(ibmclose, lag.max = 100, main = "")
# ACF does not drop quickly to zero, moreover the value
# $r_1$ is large and positive (almost 1 in this case). All this are signs of a non-stationary time series.

plot(diff(usnetelec))
Acf(diff(usnetelec), lag.max = 100, main = "")

# ACF function show that the differenced
# data behaves almost like white noise (ACF graph value $r_{13}$ is slightly beyond the critical value).

# the data were not needing a Box-Cox transformation.

# usgdp series
plot(usgdp)
```

```
Acf(usgdp, lag.max = 100, main = "")

# ACF does not drop quickly to values inside or almost inside the critical values.
# Moreover the value $r_1$ is large and positive (almost 1 in this case).
# All this are signs of a non-stationary time series.

lambda = BoxCox.lambda(usgdp)
plot(diff(BoxCox(usgdp, lambda)))
Acf(diff(BoxCox(usgdp, lambda)), lag.max = 100, main = "")
```