exercise_part02

Forecasting: Linear Trend and ARIMA Models

Original Source

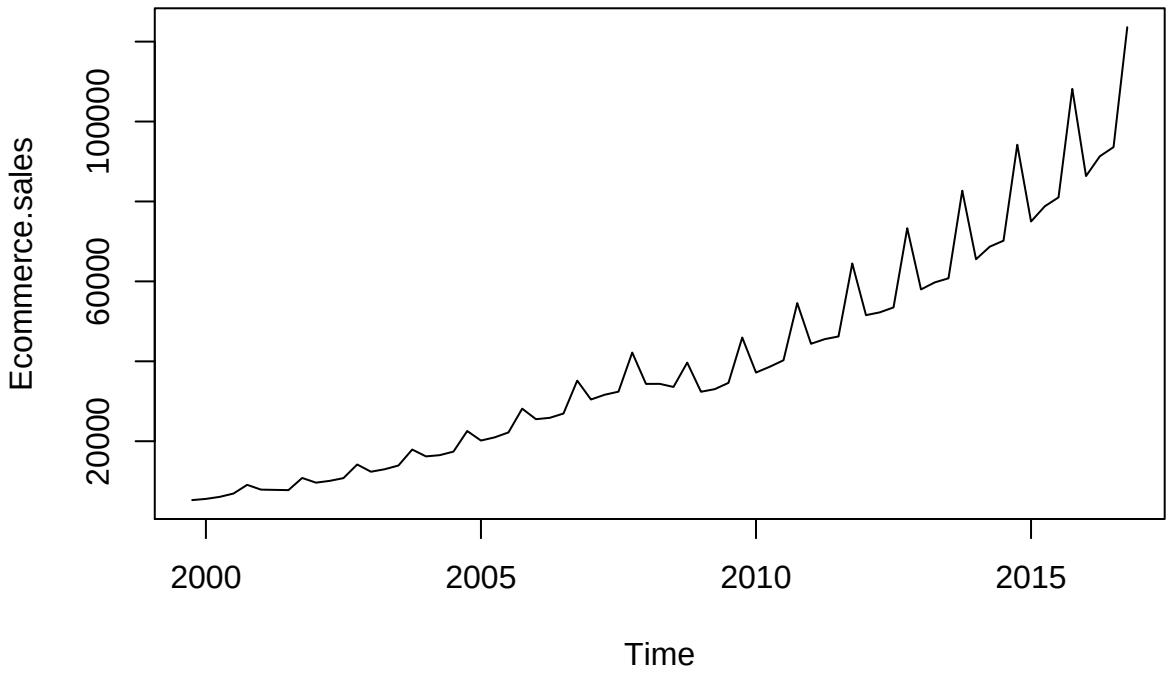
There are two main approaches to time series forecasting. One of them is to find persistent patterns in a time series itself, and extrapolate those patterns. Another is to discover how a series depend on other variables, which serve as predicators.

- naive model
- linear model (based on a simple linear regression)
- ARIMA model (assumes a linear dependence of a time series on its past values and random shocks.)

Exercise 1

Read the data from the filed, and transform it into a time series (ts) object (given that the data is quarterly and the starting period is the fourth quarter of 1999). Plot the obtained series.

```
df <- read.csv("ecommerce.csv")
series <- ts(df, frequency=4, start=c(1999,4))
plot(series)
```



Exercise 2

Make a naive forecast for the next 8 periods using the appropriate function from the **forecast** package (i.e. create an object of the class forecast using the function that implements the naive method of forecasting)(Note thtat this method sets all forecast values equal to the last known time series value).

```
require(forecast)
```

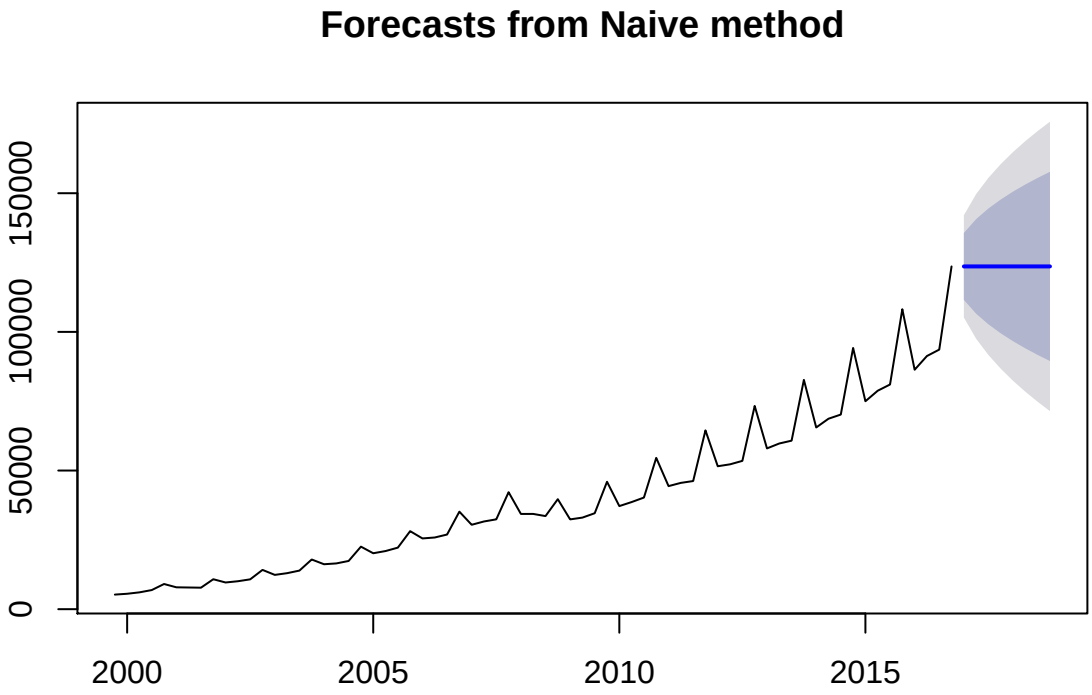
```
## Loading required package: forecast
```

```
fcast_naive <- naive(series, h=8)
```

Exercise 3

Plot the forecast values.

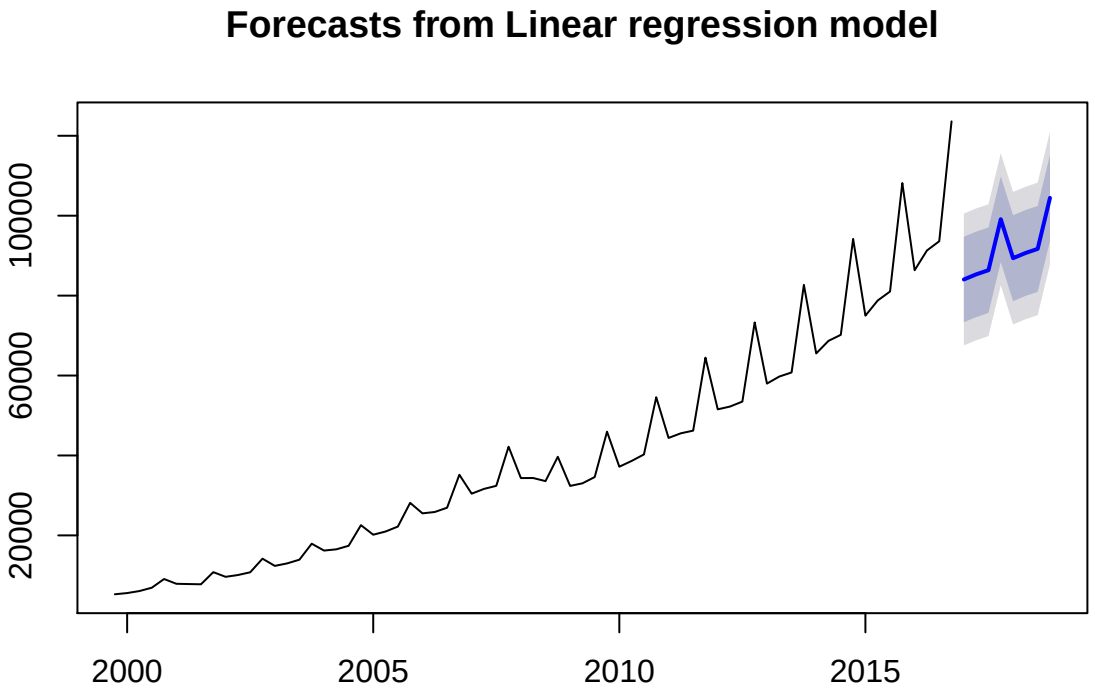
```
plot(fcast_naive)
```



Exercise 4

Make a forecast for the next 8 peridos based on a linear model in two steps: (1) create a linear regression model for the forecast using the tslm function from the forecast package (use the series as the dependent variable, trend and season as independent variables), (2) make a forecast based on the model using the forecast function from the same package. Plot the forecast.

```
model_linear <- tslm(series ~ trend + season)
fcast_linear <- forecast(model_linear, h=8)
plot(fcast_linear)
```



Exercise 5

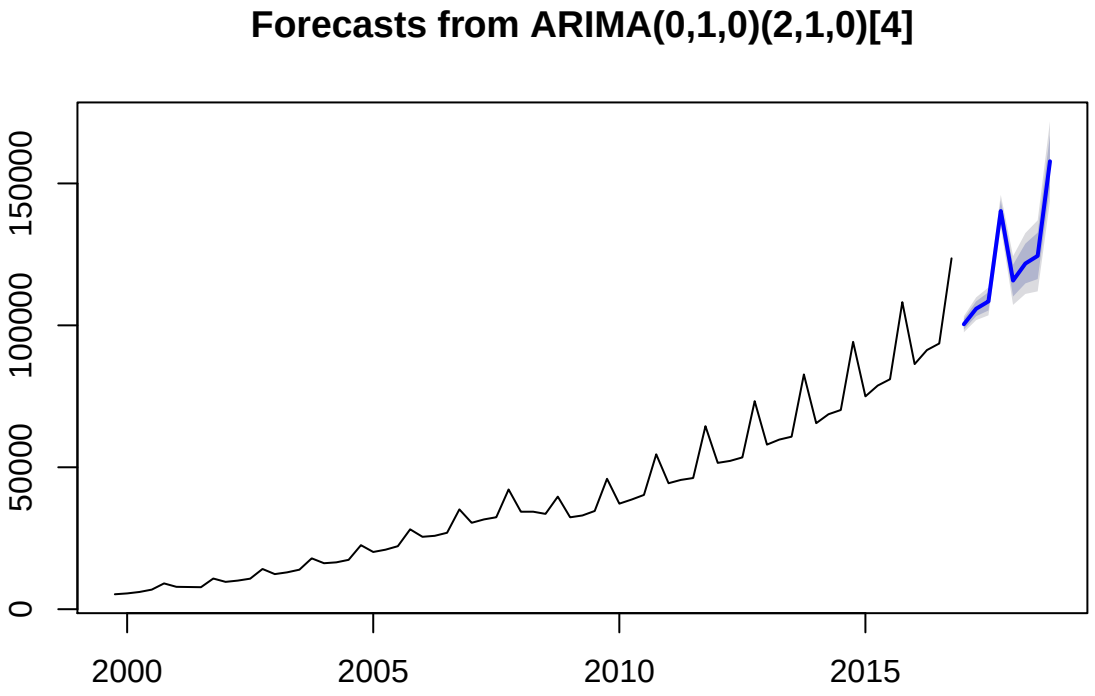
Retrieve forecast errors (residuals) from the linear model based forecast and save them as a separate variable.

```
fcast_linear_residuals <- fcast_linear$residuals
```

Exercise 6

Make a forecast for the next 8 periods based on the ARIMA model in two steps: (1) create a model using the **auto.arima** function from the forecast package, (2) make a forecast based on the model using the forecast function from the same package. Plot the forcast.

```
model_arima <- auto.arima(series)
fcast_arima <- forecast(model_arima, h=8)
plot(fcast_arima)
```



Exercise 7

Print the summary fo the forecast based on the ARIMA model.

```
fcast_arima_summary <- summary(fcast_arima)
```

```
##
## Forecast method: ARIMA(0,1,0)(2,1,0)[4]
##
## Model Information:
## Series: series
## ARIMA(0,1,0)(2,1,0)[4]
##
## Coefficients:
##         sar1    sar2
##       0.2700  0.3421
## s.e.  0.1137  0.1175
##
## sigma^2 estimated as 2092622:  log likelihood=-556.39
## AIC=1118.78   AICc=1119.18   BIC=1125.25
##
## Error measures:
##                     ME    RMSE      MAE        MPE     MAPE      MASE
## Training set 97.00685 1371.25 921.3438 0.07940248 2.968625 0.1564942
##                     ACF1
## Training set 0.01029391
##
## Forecasts:
##         Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## 2017 Q1       100381.6  98527.7 102235.5  97546.32 103216.8
## 2017 Q2       105849.6 103227.8 108471.4 101839.93 109859.3
```

```
## 2017 Q3       108442.8 105231.8 111653.8 103531.96 113353.6
## 2017 Q4       140288.8 136581.0 143996.5 134618.24 145959.3
## 2018 Q1       115787.3 110178.6 121396.0 107209.51 124365.1
## 2018 Q2       121787.1 114775.1 128799.1 111063.21 132511.0
## 2018 Q3       124482.7 116304.8 132660.6 111975.70 136989.7
## 2018 Q4       157800.4 148603.2 166997.6 143734.55 171866.3
```

Exercise 8

Explore the structure the forecast summary. Find the forecast value for the last period, and its 5% confidence interval values.

```
fcast_arima_summary[["Point Forecast"]][8]
```

```
## [1] 157800.4
```

```
fcast_arima_summary[["Lo 95"]][8]
```

```
## [1] 143734.6
```

```
fcast_arima_summary[["Hi 95"]][8]
```

```
## [1] 171866.3
```

Exercise 9

Retrieve forecast errors (residuals) from the ARIMA based forecast.

```
fcast_arima_residuals <- fcast_arima$residuals
```

Exercise 10

Use the errors from the ARIMA based forecast and the errors from the linear model based froecast to compare predictive accuracy of the two models with the Diebold-Mariano test (implemented as a function in the forecast package). Test the hypothesis that the ARIMA based forecast is more accurate than the linear model based forecast.

```
dm.test(fcast_linear_residuals, fcast_arima_residuals, alternative = "greater")
```

```
##
##  Diebold-Mariano Test
##
## data:  fcast_linear_residualsfcast_arima_residuals
## DM = 3.9013, Forecast horizon = 1, Loss function power = 2,
## p-value = 0.000111
## alternative hypothesis: greater
```