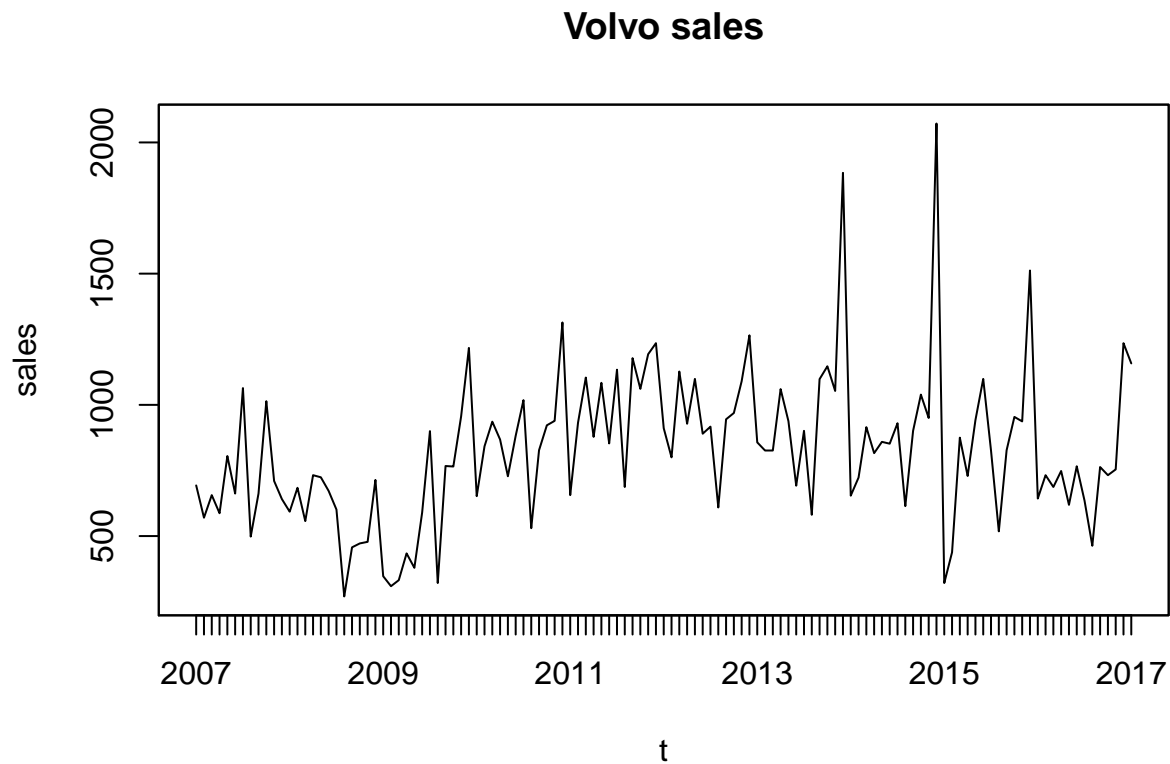


Volvo sales

Let's see the sales of Volvo.

```
suppressMessages(library(zoo))
suppressMessages(library(forecast))
df = read.csv('datasets_830_1554_norway_new_car_sales_by_make.csv')
df = subset(df, Make == 'Volvo')
sales = zooreg(data = df$Quantity, as.yearmon("2007-01-01"), freq = 12)
N = length(sales)
t = c(1:N)
Q = factor(c(rep(c(1:12), N/12), c(1:(N%12)))))
plot(sales, main = "Volvo sales", xlab = "t")
```



Regression

It seems reasonable to have seasonality of 12 months.

Seasonal dummies

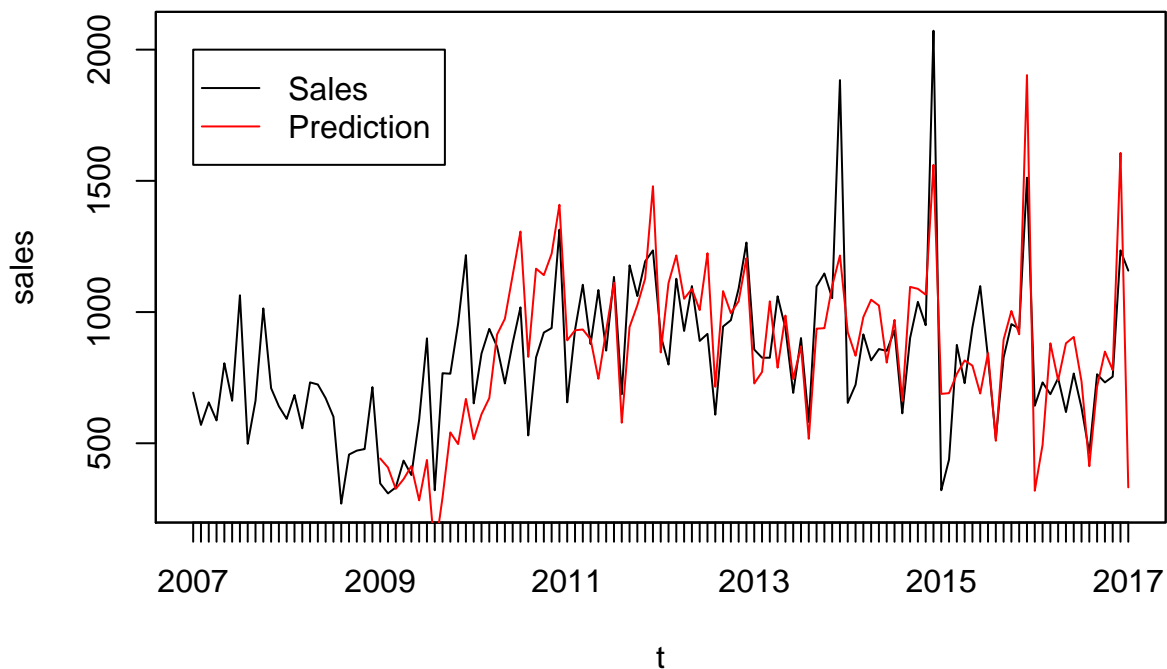
```
next_step = function(t) {
  train = data.frame(
    t = t[1:24],
    sales = sales[t[1:24]],
    Q = Q[t[1:24]]
  )
  mod = lm(sales~t+Q, data = train)
  prediction = predict(mod, data.frame(t=t[25], Q=Q[t[25]]))
}
```

```

    return(prediction)
}
prediction = rollapply(t, 25, next_step)
plot(sales, main = "Seasonal dummies prediction 1 step ahead", xlab = "t")
lines(time(sales)[25:length(t)], prediction, col = "red")
legend(
  x = time(sales)[1],
  y = 2000,
  legend = c('Sales', 'Prediction'),
  col = c('black', 'red'),
  pch = c('', ''),
  lty = c(1, 1)
)

```

Seasonal dummies prediction 1 step ahead



```

mape = mean(abs((prediction - sales[25:length(sales)])/sales[25:length(sales)]))
paste("MAPE =", mape)

```

```
## [1] "MAPE = 0.214958623085887"
```

Polynomial

```

library(zoo)
next_step = function(t) {
  train = data.frame(
    t = t[1:24],
    sales = sales[t[1:24]],
    Q = Q[t[1:24]]
  )
  mod = lm(sales~poly(t, 2)+Q, data = train)
  prediction = predict(mod, data.frame(t=t[25], Q=Q[t[25]]))
}

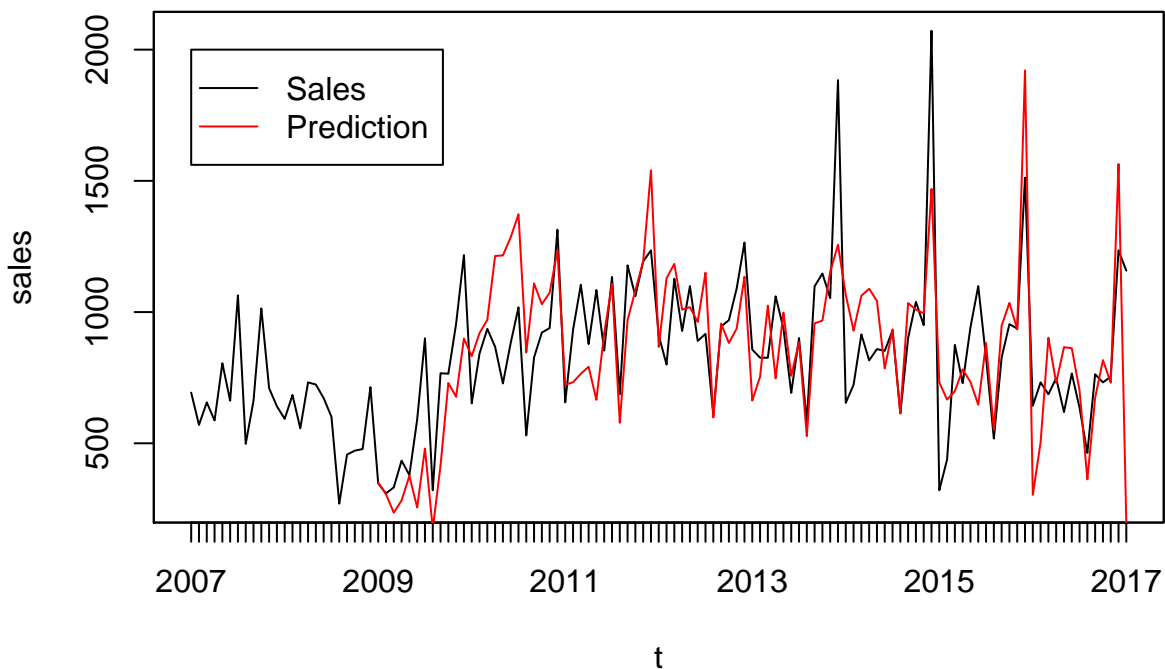
```

```

    return(prediction)
}
prediction = rollapply(t, 25, next_step)
plot(sales, main = "Polynomial 2nd order and seasonal dummies prediction 1 step ahead", xlab = "t")
lines(time(sales)[25:length(t)], prediction, col = "red")
legend(
  x = time(sales)[1],
  y = 2000,
  legend = c('Sales', 'Prediction'),
  col = c('black', 'red'),
  pch = c('', ''),
  lty = c(1, 1)
)

```

Polynomial 2nd order and seasonal dummies prediction 1 step ahead



```

mape = mean(abs((prediction - sales[25:length(sales)])/sales[25:length(sales)]))
paste("MAPE =", mape)

```

```
## [1] "MAPE = 0.213598454013614"
```

Exponential smoothing

Exponential smoothing

```

next_step = function(t) {
  train = data.frame(
    t = t[1:24],
    sales = sales[t[1:24]],
    Q = Q[t[1:24]]
  )
  mod = HoltWinters(train$sales, beta = F, gamma = F)

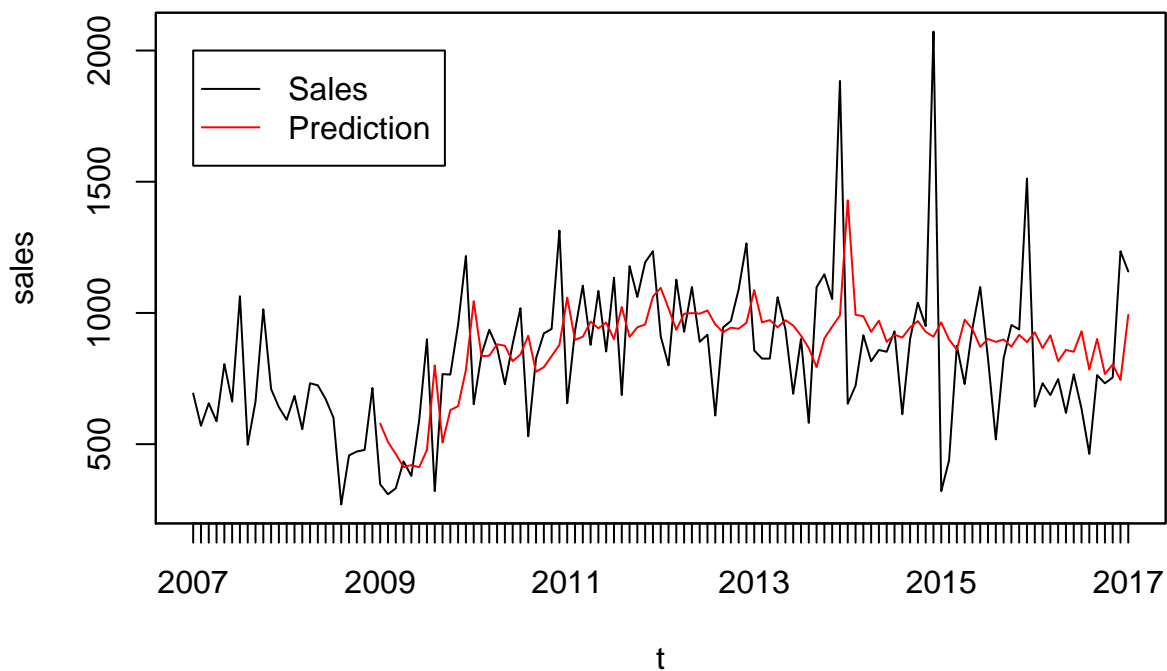
```

```

prediction = forecast(mod, 1)
return(prediction$mean)
}
prediction = rollapply(t, 25, next_step)
plot(sales, main = "Exponential smoothing prediction 1 step ahead", xlab = "t")
lines(time(sales)[25:length(t)], prediction, col = "red")
legend(
  x = time(sales)[1],
  y = 2000,
  legend = c('Sales', 'Prediction'),
  col = c('black', 'red'),
  pch = c('', ''),
  lty = c(1, 1)
)

```

Exponential smoothing prediction 1 step ahead



```

mape = mean(abs((prediction - sales[25:length(sales)])/sales[25:length(sales)]))
paste("MAPE =", mape)

```

```
## [1] "MAPE = 0.277045393426693"
```

Holt

```

next_step = function(t) {
  train = data.frame(
    t = t[1:24],
    sales = sales[t[1:24]],
    Q = Q[t[1:24]]
  )
  mod = HoltWinters(train$sales, beta = T, gamma = F)
  prediction = forecast(mod, 1)
}

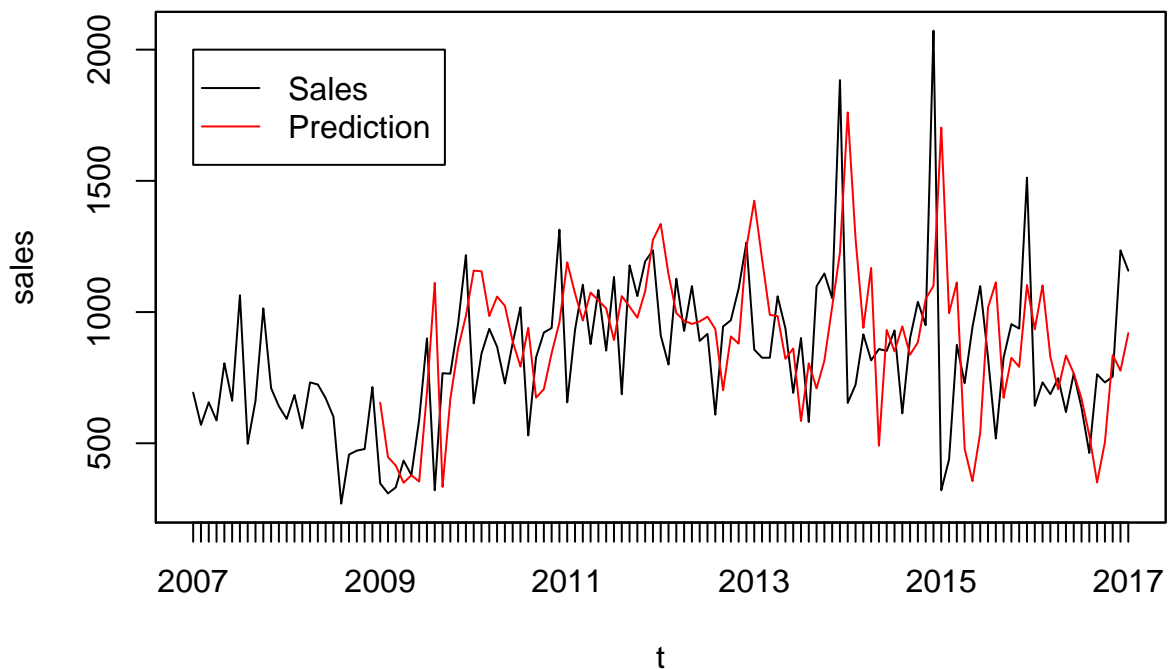
```

```

    return(prediction$mean)
}
prediction = rollapply(t, 25, next_step)
plot(sales, main = "Holt prediction 1 step ahead", xlab = "t")
lines(time(sales)[25:length(t)], prediction, col = "red")
legend(
  x = time(sales)[1],
  y = 2000,
  legend = c('Sales', 'Prediction'),
  col = c('black', 'red'),
  pch = c('', ''),
  lty = c(1, 1)
)

```

Holt prediction 1 step ahead



```

mape = mean(abs((prediction - sales[25:length(sales)])/sales[25:length(sales)]))
paste("MAPE =", mape)

```

```
## [1] "MAPE = 0.367475797650731"
```

Additive Holt-Winters

```

next_step = function(t) {
  train = data.frame(
    t = t[1:24],
    sales = sales[t[1:24]],
    Q = Q[t[1:24]]
  )
  mod = HoltWinters(ts(train$sales, frequency = 12), beta = T, gamma = T, seasonal = "additive")
  prediction = forecast(mod, 1)
  return(prediction$mean)
}

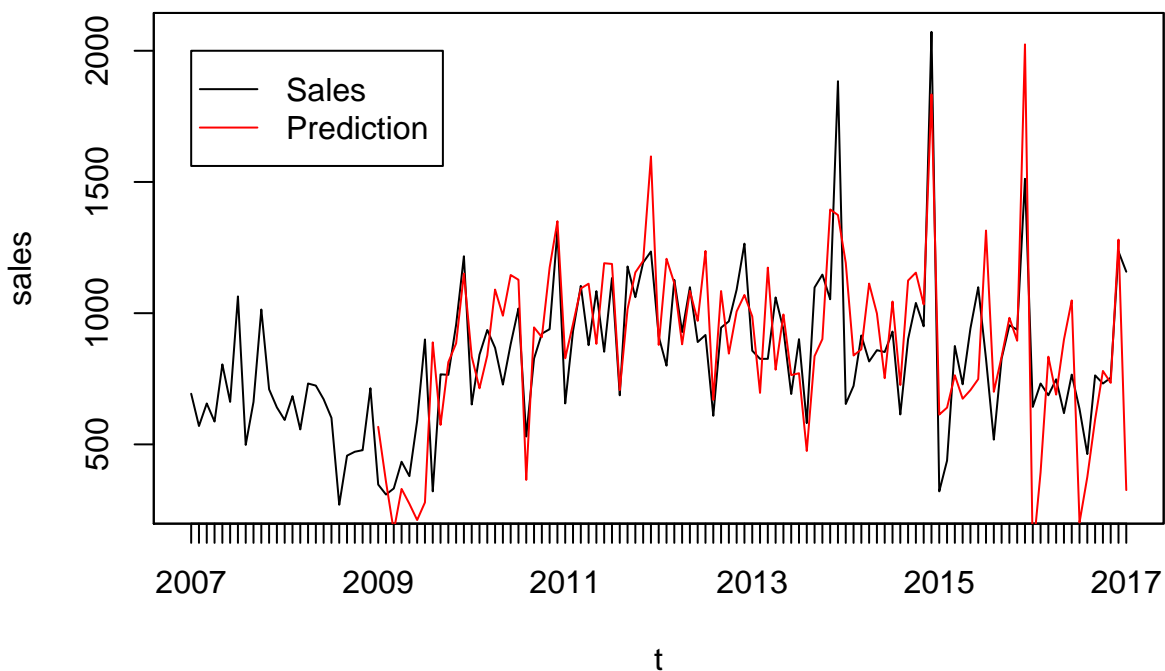
```

```

}
prediction = rollapply(t, 25, next_step)
plot(sales, main = "Additive Holt-Winters prediction 1 step ahead", xlab = "t")
lines(time(sales)[25:length(t)], prediction, col = "red")
legend(
  x = time(sales)[1],
  y = 2000,
  legend = c('Sales', 'Prediction'),
  col = c('black', 'red'),
  pch = c('', ''),
  lty = c(1, 1)
)

```

Additive Holt-Winters prediction 1 step ahead



```

mape = mean(abs((prediction - sales[25:length(sales)])/sales[25:length(sales)]))
paste("MAPE =", mape)

```

```
## [1] "MAPE = 0.247905784030637"
```

Seasonal Holt-Winters

```

next_step = function(t) {
  train = data.frame(
    t = t[1:24],
    sales = sales[t[1:24]],
    Q = Q[t[1:24]]
  )
  mod = HoltWinters(ts(train$sales, frequency = 12), beta = T, gamma = T, seasonal = "multiplicative")
  prediction = forecast(mod, 1)
  return(prediction$mean)
}

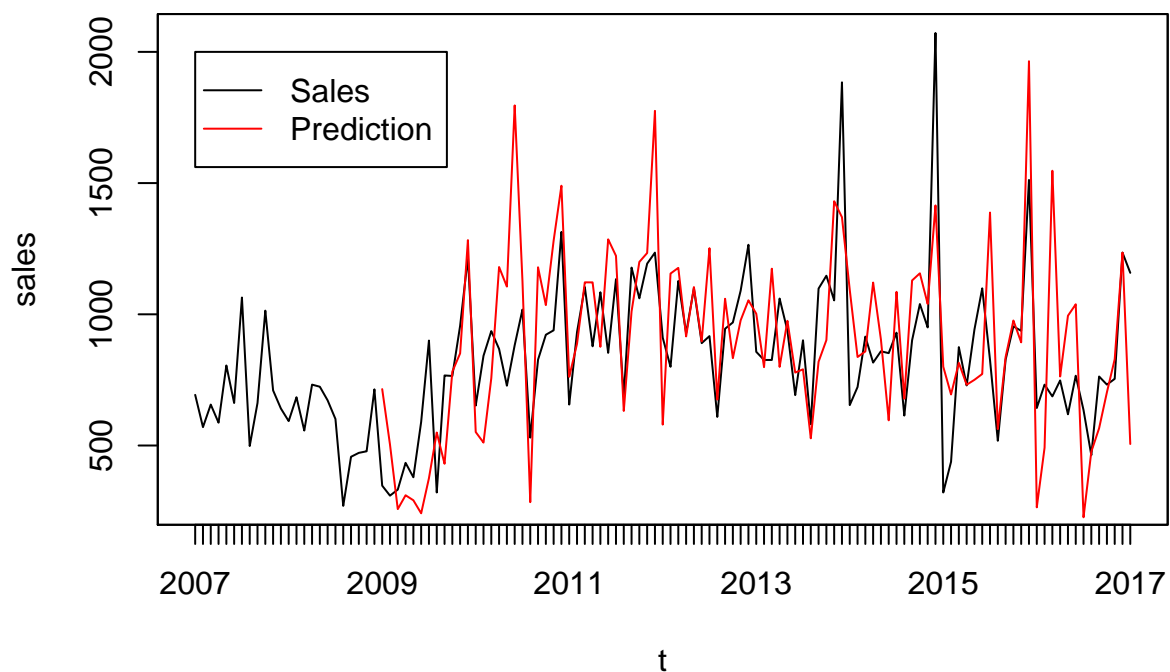
```

```

prediction = rollapply(t, 25, next_step)
plot(sales, main = "Seasonal Holt-Winters prediction 1 step ahead", xlab = "t")
lines(time(sales)[25:length(t)], prediction, col = "red")
legend(
  x = time(sales)[1],
  y = 2000,
  legend = c('Sales', 'Prediction'),
  col = c('black', 'red'),
  pch = c('', ''),
  lty = c(1, 1)
)

```

Seasonal Holt-Winters prediction 1 step ahead



```

mape = mean(abs((prediction - sales[25:length(sales)]) / sales[25:length(sales)]))
paste("MAPE =", mape)

```

```
## [1] "MAPE = 0.276258405722419"
```