# 1

## A)

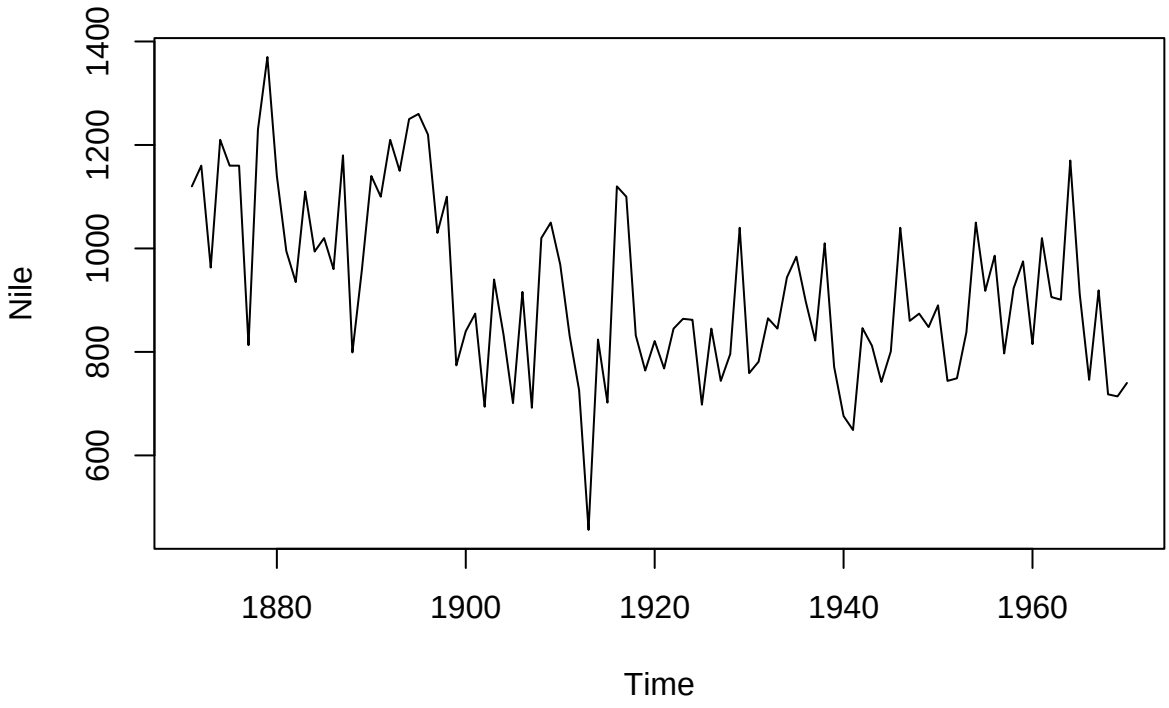
O parâmetro $\alpha$ chamado de constante de alisamento ou parâmetro da suavização, esta contido num intervalo (0,1), serve para refletir a influência dos valores passados:

- Mais próximo de 0 indica previções que dependem mais dos valores mais antigos.
- Mais próximo de 1 indica previções que dependem mais das observações mais recentes.

# 3

## A) Nile

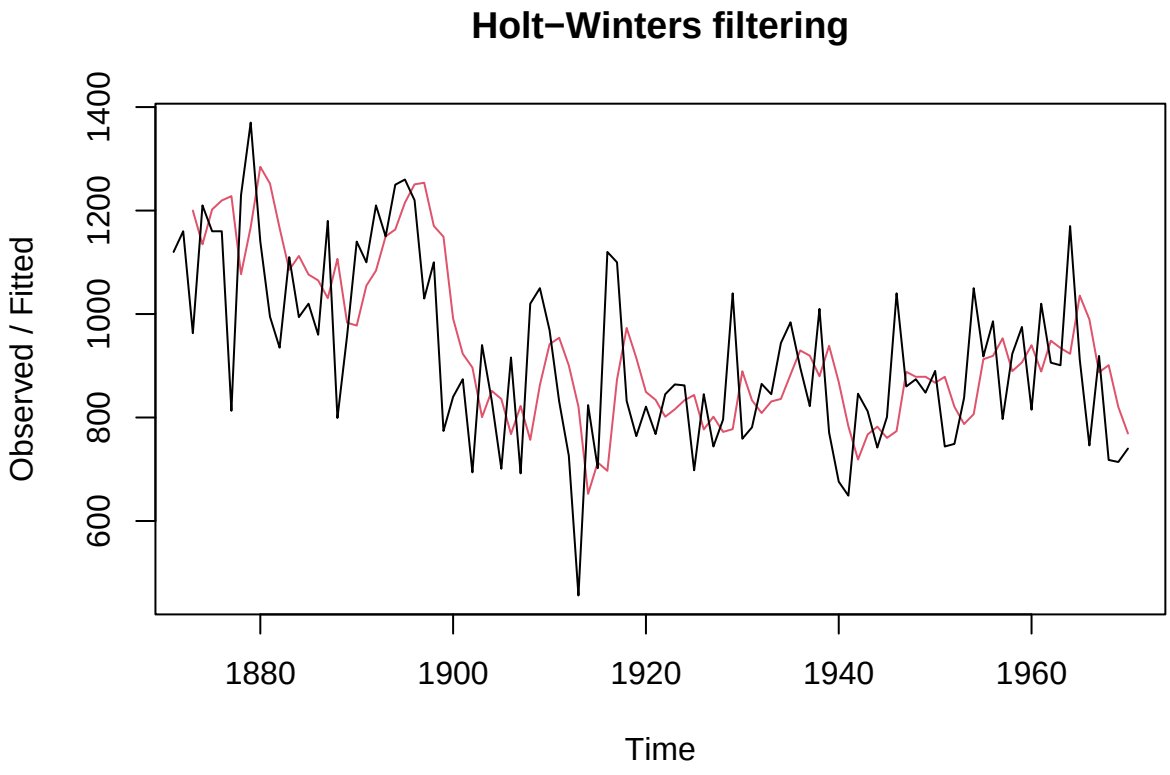
```
plot(Nile)
```



```
(fit.nile <- HoltWinters(x = Nile, gamma=FALSE))
```

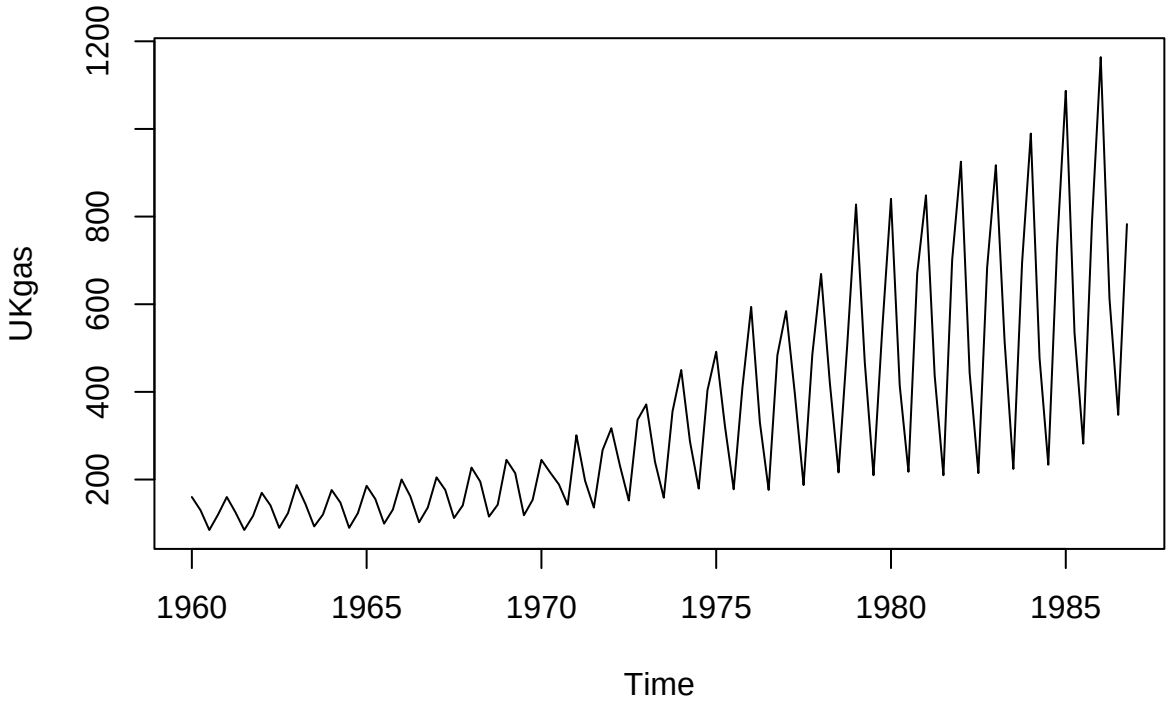
```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = Nile, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.4190643
##  beta : 0.05987705
##  gamma: FALSE
##
## Coefficients:
##         [,1]
## a 756.913740
## b  -7.424597
```

```
plot(fit.nile)
```



### B) UKgas

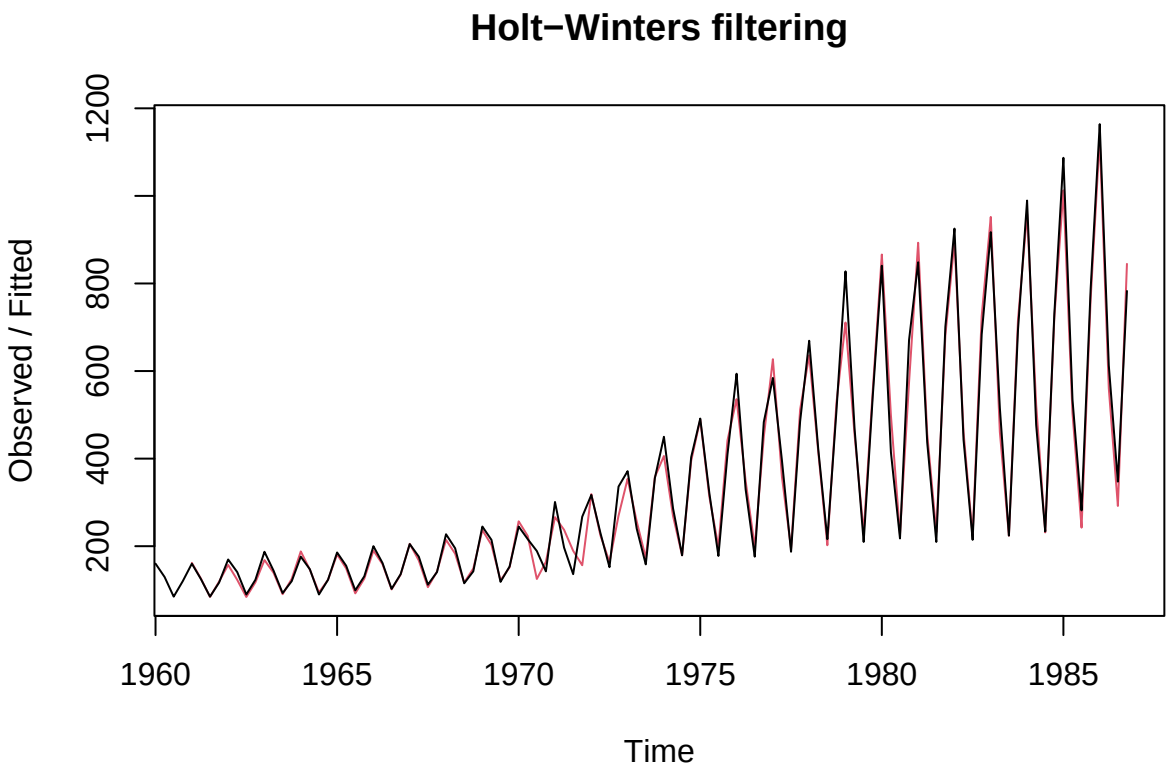
```
plot(UKgas)
```




```
(fit.UKgas <- HoltWinters(x = UKgas, seasonal = "multiplicative"))
```

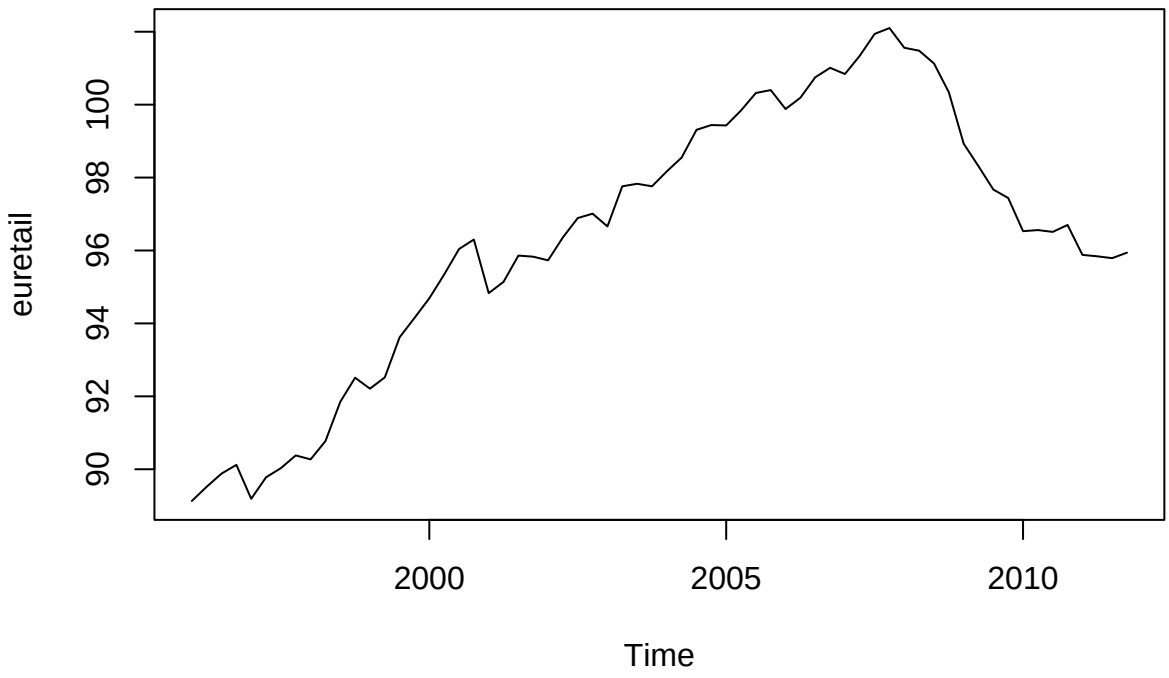
```
## Holt-Winters exponential smoothing with trend and multiplicative seasonal component.
##
## Call:
## HoltWinters(x = UKgas, seasonal = "multiplicative")
##
## Smoothing parameters:
##  alpha: 0.02412858
##  beta : 1
##  gamma: 0.7828624
##
## Coefficients:
##           [,1]
## a  515.0263067
## b    9.9725235
## s1   2.3829116
## s2   1.2182884
## s3   0.6645498
## s4   1.5453760
```

```
plot(fit.UKgas)
```



### C) euretail

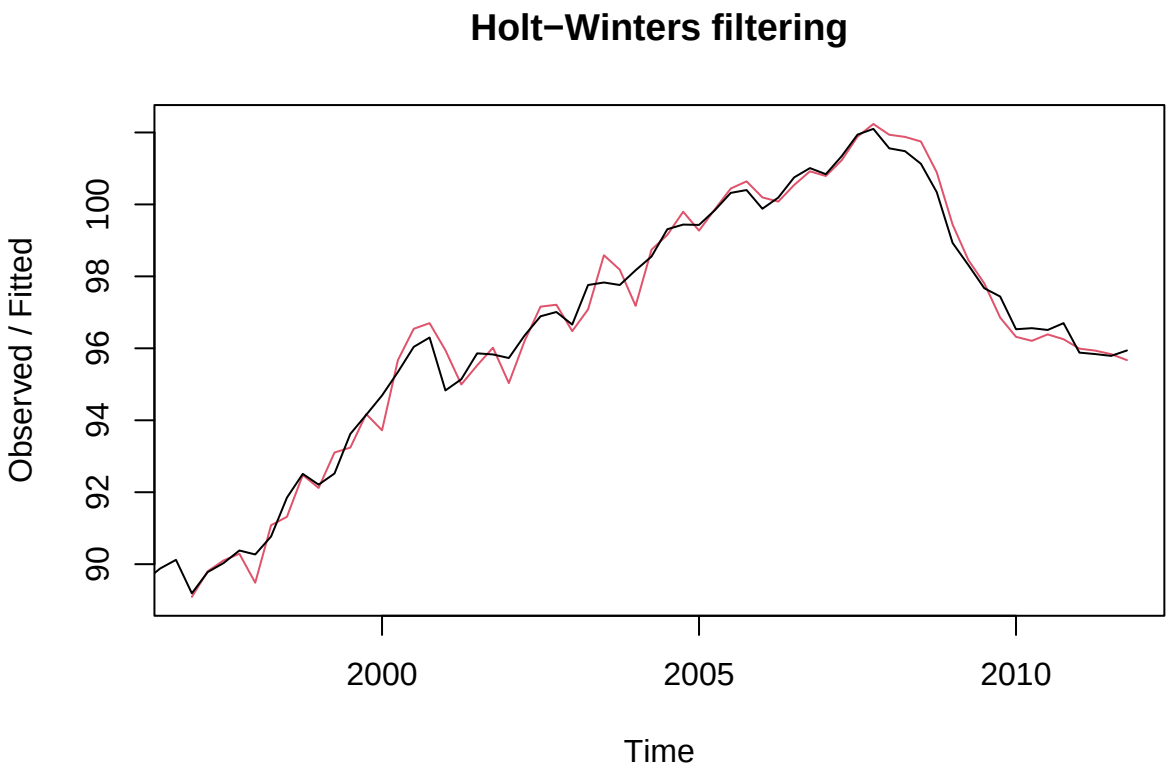
```
plot(euretail)
```




```
(fit.euretail <- HoltWinters(x = euretail, seasonal = "additive"))
```

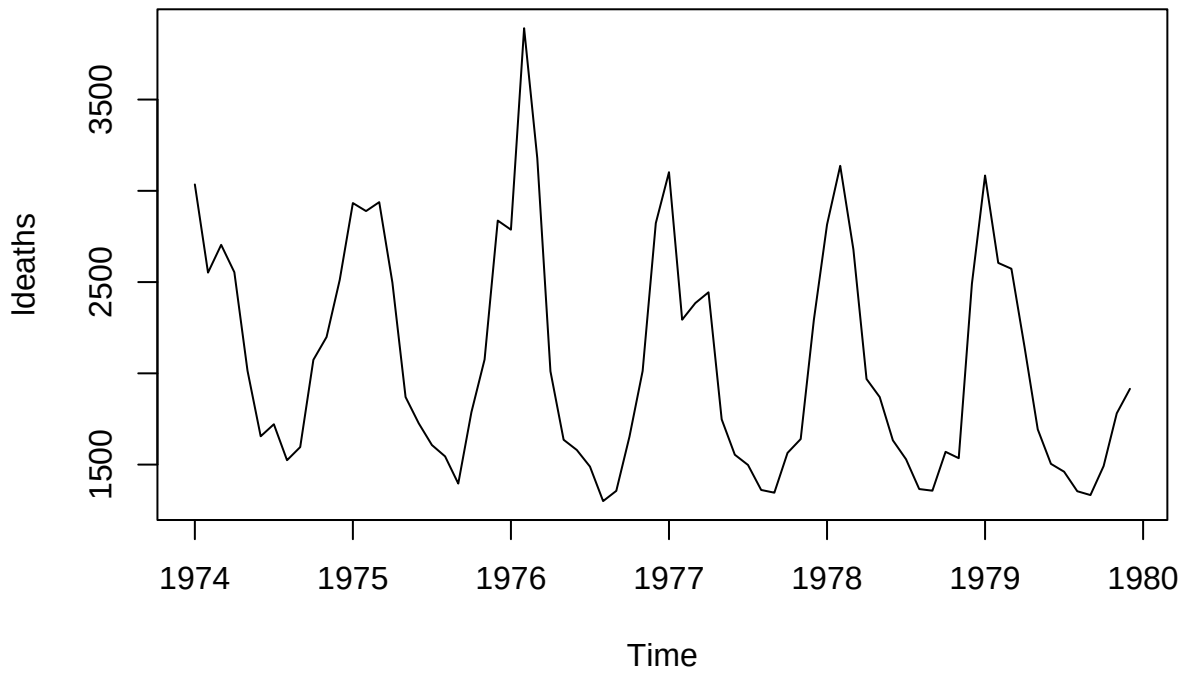
```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = euretail, seasonal = "additive")
##
## Smoothing parameters:
##  alpha: 0.8691965
##  beta : 0.449748
##  gamma: 1
##
## Coefficients:
##          [,1]
## a  95.6666446
## b  -0.1406139
## s1 -0.3394992
## s2 -0.1075213
## s3  0.1105958
## s4  0.2733554
```

```
plot(fit.euretail)
```



### D) ldeaths

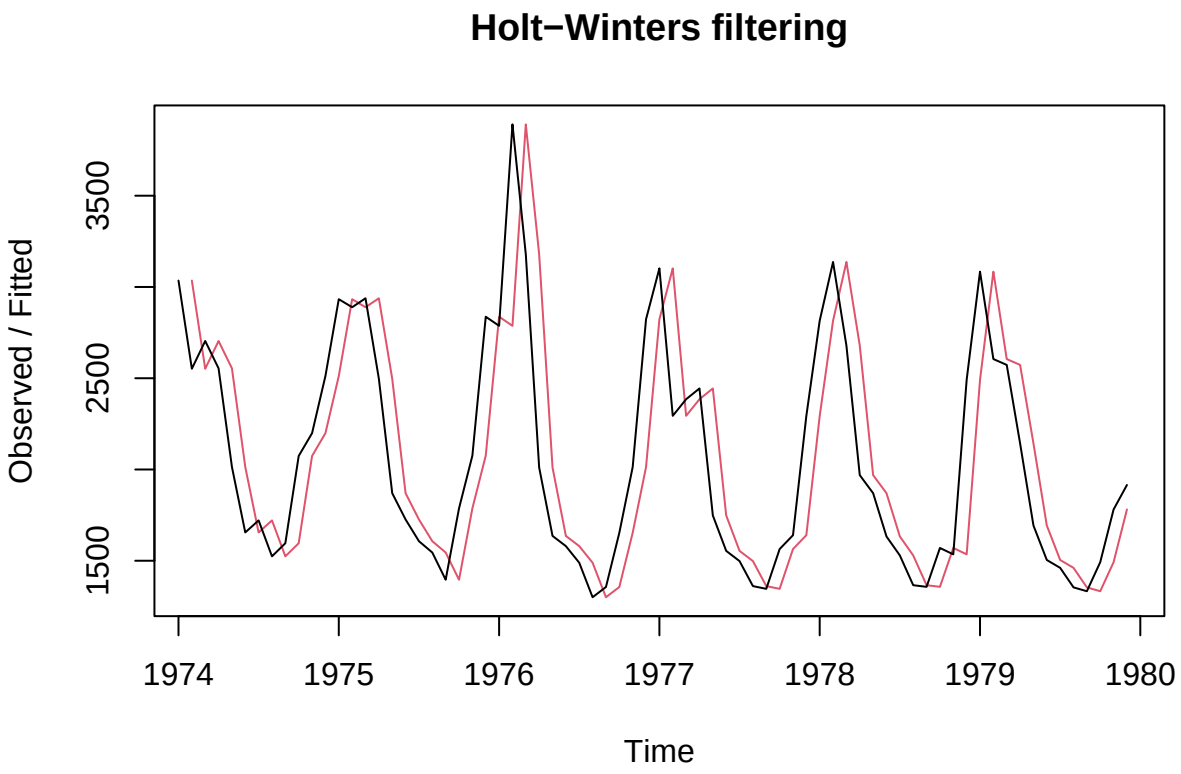
```
plot(ldeaths)
```




```
(fit.ldeaths <- HoltWinters(x = ldeaths, beta=FALSE, gamma=FALSE))
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = ldeaths, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.9999339
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 1914.991
```

```
plot(fit.ldeaths)
```

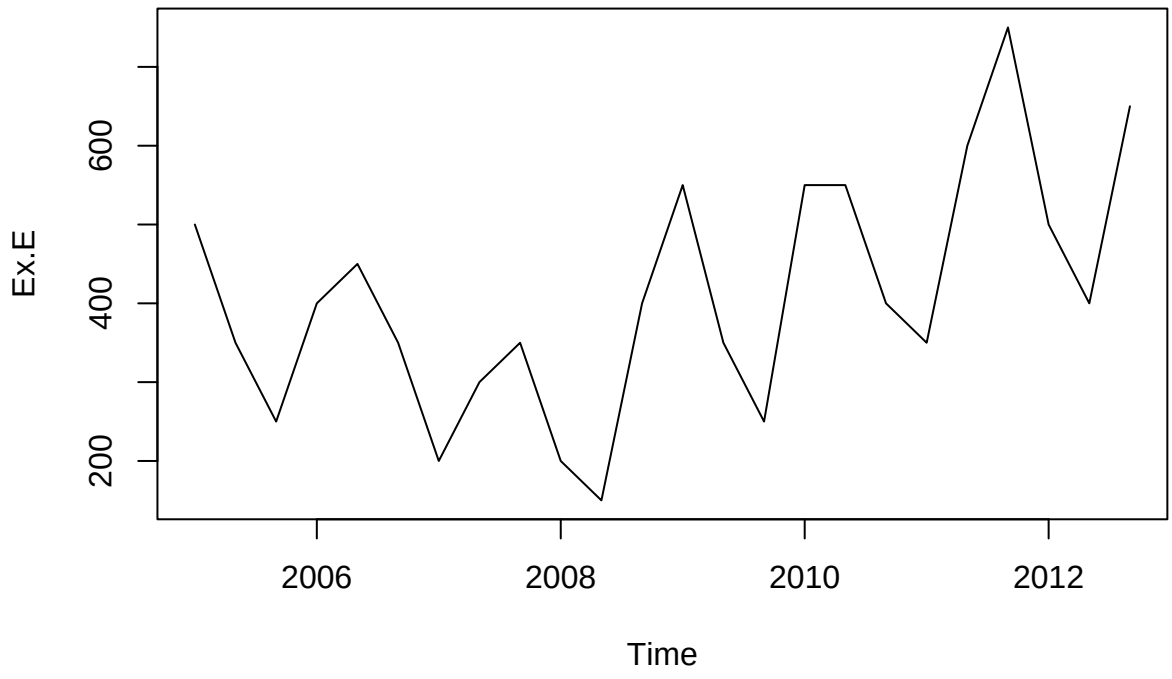

Holt−Winters filtering
Observed / Fitted
1500
2500
3500
1974
1975
1976
1977
1978
1979
1980
Time

### E) Sales

```
vendas <- c(500, 350, 250, 400, 450, 350, 200, 300, 350, 200, 150, 400, 550, 350, 250,
550, 550, 400, 350, 600, 750, 500, 400, 650)

Ex.E <- ts(vendas, start = 2005, frequency = 3)

plot(Ex.E)
```

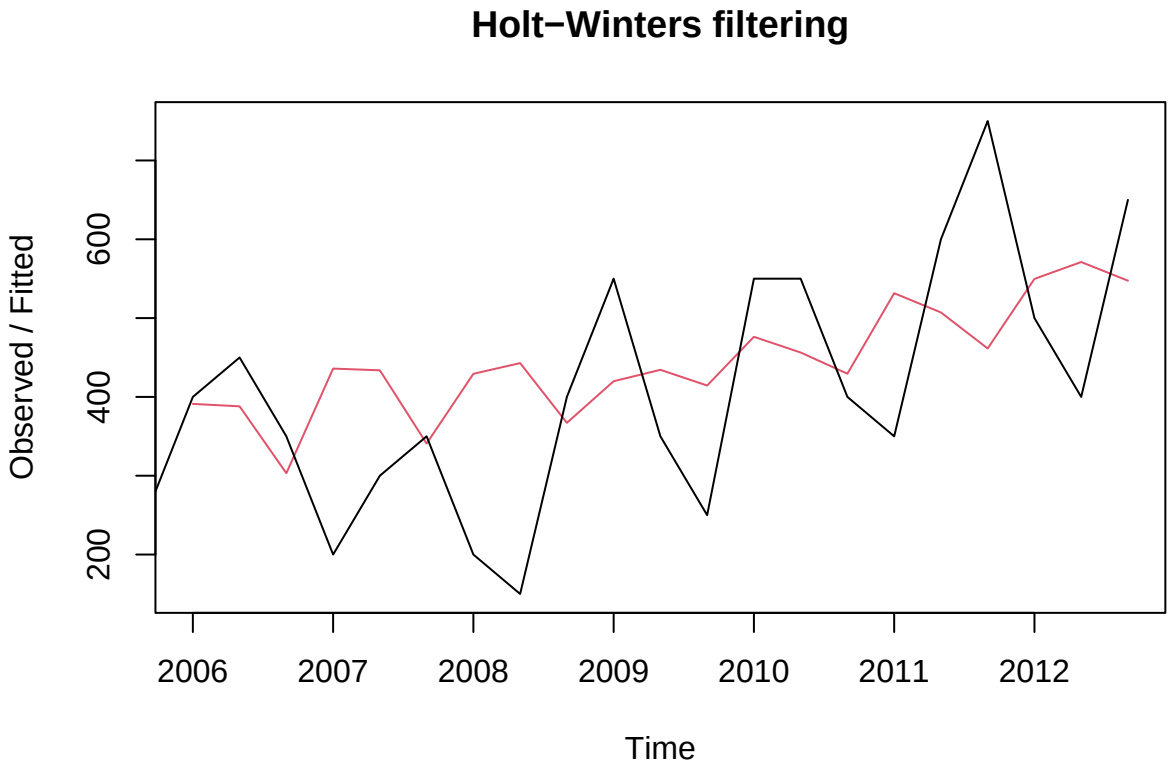



```
(fit.Ex.E <- HoltWinters(x = Ex.E, seasonal = "additive"))
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = Ex.E, seasonal = "additive")
##
## Smoothing parameters:
##  alpha: 0.02946233
##  beta : 0
##  gamma: 0.157255
##
## Coefficients:
##         [,1]
## a  594.72784
```

```
## b   13.33333
## s1 -29.39493
## s2 -38.41639
## s3 -28.56955
```

```
plot(fit.Ex.E)
```



# 4

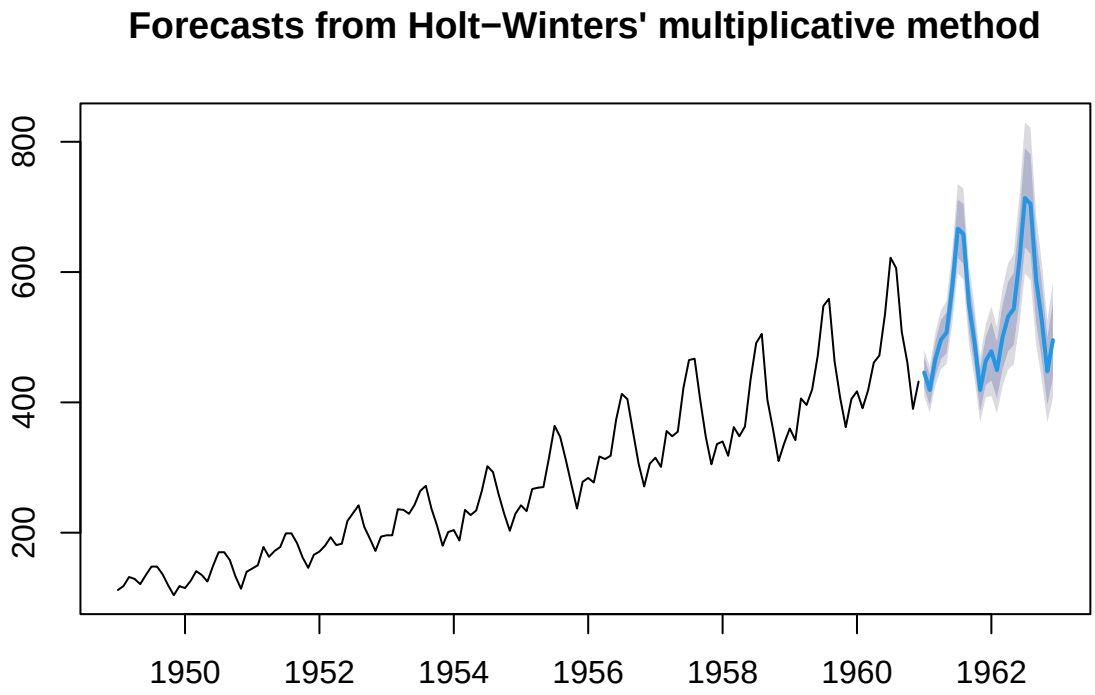
## A) AirPassengers

```
fit.hw.mult <- hw(AirPassengers, h=24, seasonal = "m", initial = "o")
summary(fit.hw.mult)
```

```
##
## Forecast method: Holt-Winters' multiplicative method
##
## Model Information:
## Holt-Winters' multiplicative method
##
## Call:
##  hw(y = AirPassengers, h = 24, seasonal = "m", initial = "o")
##
##   Smoothing parameters:
##     alpha = 0.3146
##     beta  = 0.0071
##     gamma = 0.5977
##
##   Initial states:
##     l = 120.3796
##     b = 1.7757
##     s = 0.9298 0.7946 0.9024 1.0451 1.1338 1.1388
##            1.0529 0.9638 1.0349 1.0807 0.9854 0.9378
##
##   sigma:  0.0407
##
##      AIC     AICc      BIC
## 1405.654 1410.511 1456.141
##
## Error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set 1.256973 10.63256 7.790649 0.2182707 2.914411 0.2432275 0.2135914
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1961       445.8901 422.6577 469.1225 410.3592 481.4210
## Feb 1961       418.9478 396.0288 441.8667 383.8963 453.9993
## Mar 1961       466.4298 439.7182 493.1414 425.5780 507.2816
## Apr 1961       496.1291 466.4627 525.7955 450.7583 541.4999
## May 1961       507.1463 475.5546 538.7381 458.8309 555.4617
## Jun 1961       575.6281 538.3478 612.9083 518.6129 632.6432
## Jul 1961       666.6573 621.8494 711.4652 598.1295 735.1850
## Aug 1961       658.4970 612.6386 704.3554 588.3627 728.6313
## Sep 1961       550.0907 510.4559 589.7255 489.4745 610.7069
## Oct 1961       491.7130 455.1069 528.3190 435.7289 547.6971
## Nov 1961       418.8086 386.6330 450.9842 369.6003 468.0169
## Dec 1961       463.7188 426.9948 500.4428 407.5543 519.8833
## Jan 1962       478.5040 433.5276 523.4805 409.7185 547.2896
## Feb 1962       449.4074 406.2454 492.5694 383.3969 515.4179
```

```
## Mar 1962       500.1396 451.0781 549.2010 425.1065 575.1726
## Apr 1962       531.7730 478.5135 585.0325 450.3196 613.2263
## May 1962       543.3672 487.8249 598.9096 458.4226 628.3119
## Jun 1962       616.4994 552.2059 680.7929 518.1710 714.8279
## Jul 1962       713.7167 637.8042 789.6291 597.6185 829.8148
## Aug 1962       704.7115 628.2918 781.1313 587.8376 821.5855
## Sep 1962       588.4751 523.4340 653.5163 489.0033 687.9469
## Oct 1962       525.8278 466.6127 585.0428 435.2661 616.3894
## Nov 1962       447.7002 396.3464 499.0539 369.1614 526.2389
## Dec 1962       495.5277 437.6488 553.4067 407.0096 584.0459
```

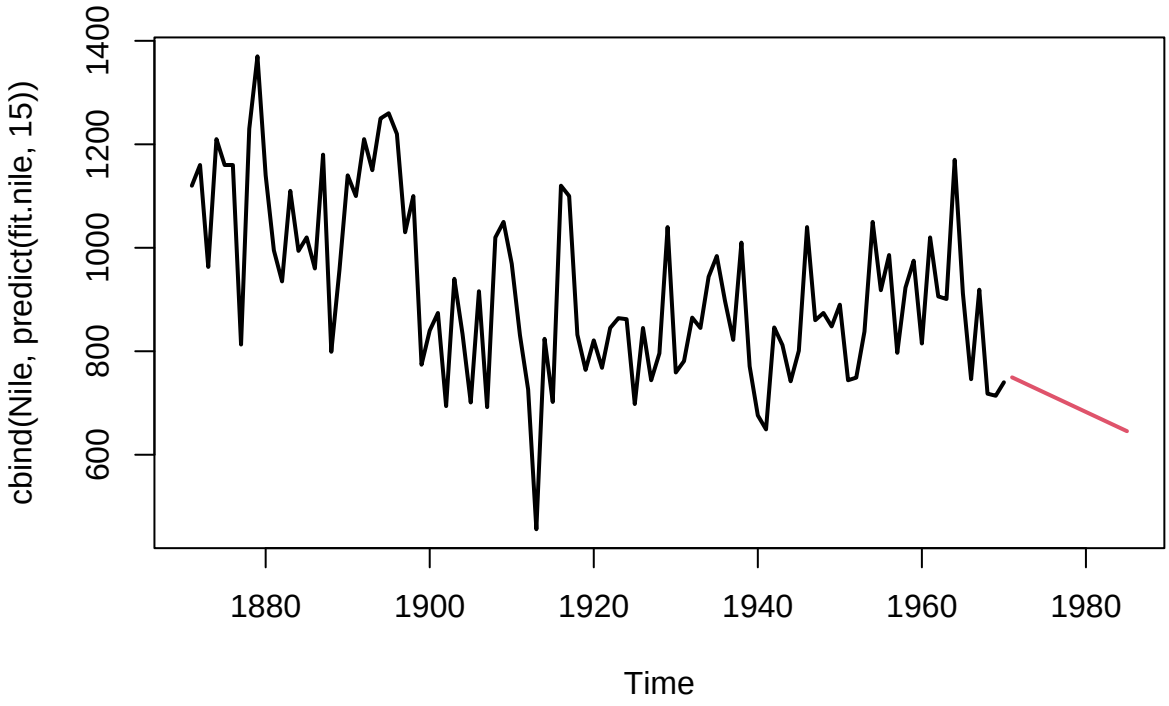
```r
plot(fit.hw.mult)
```




### B) Nile

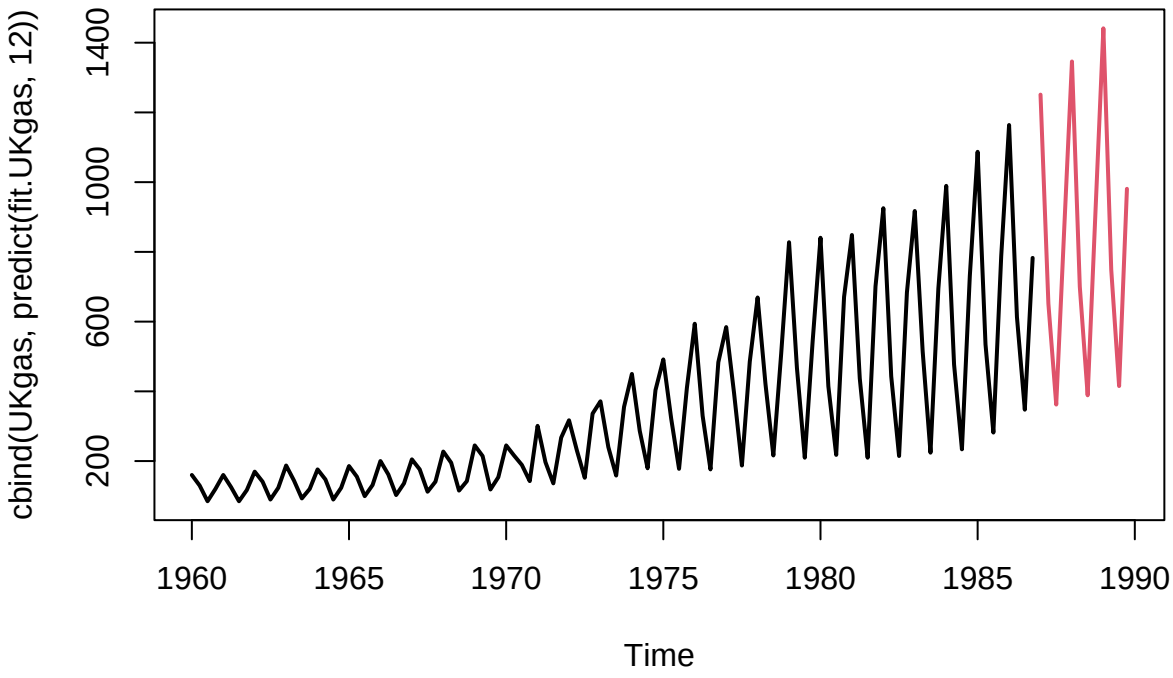
```r
# plot(predict(fit.nile, 20))

plot(cbind(Nile, predict(fit.nile, 15)), plot.type='single', col=c(1,2), lwd=2)
```



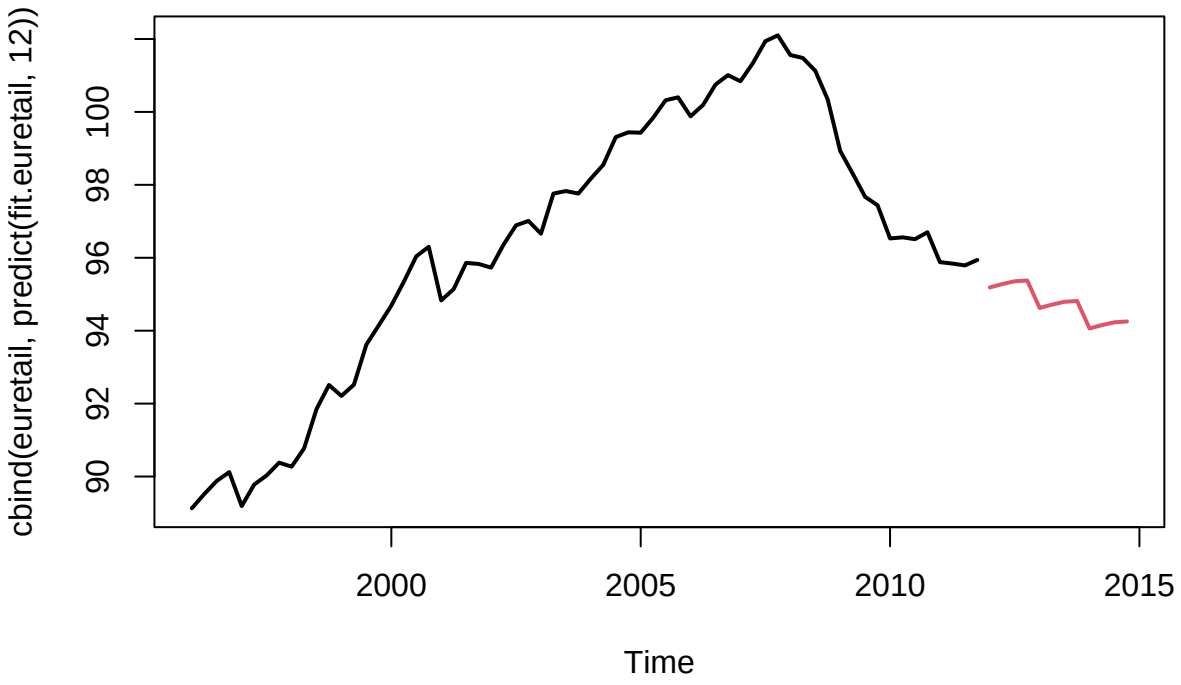

### B) UKgas

```
plot(cbind(UKgas, predict(fit.UKgas, 12)), plot.type='single', col=c(1,2), lwd=2)
```



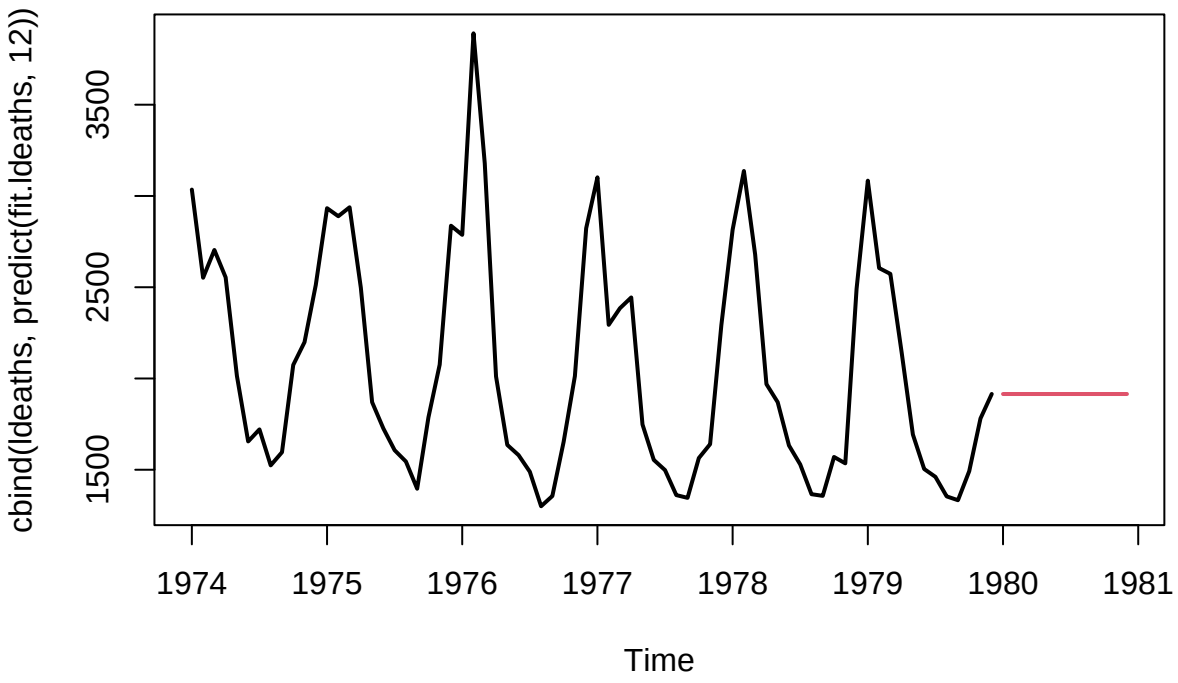

### B) euretail

```
plot(cbind(euretail, predict(fit.euretail, 12)), plot.type='single', col=c(1,2), lwd=2)
```



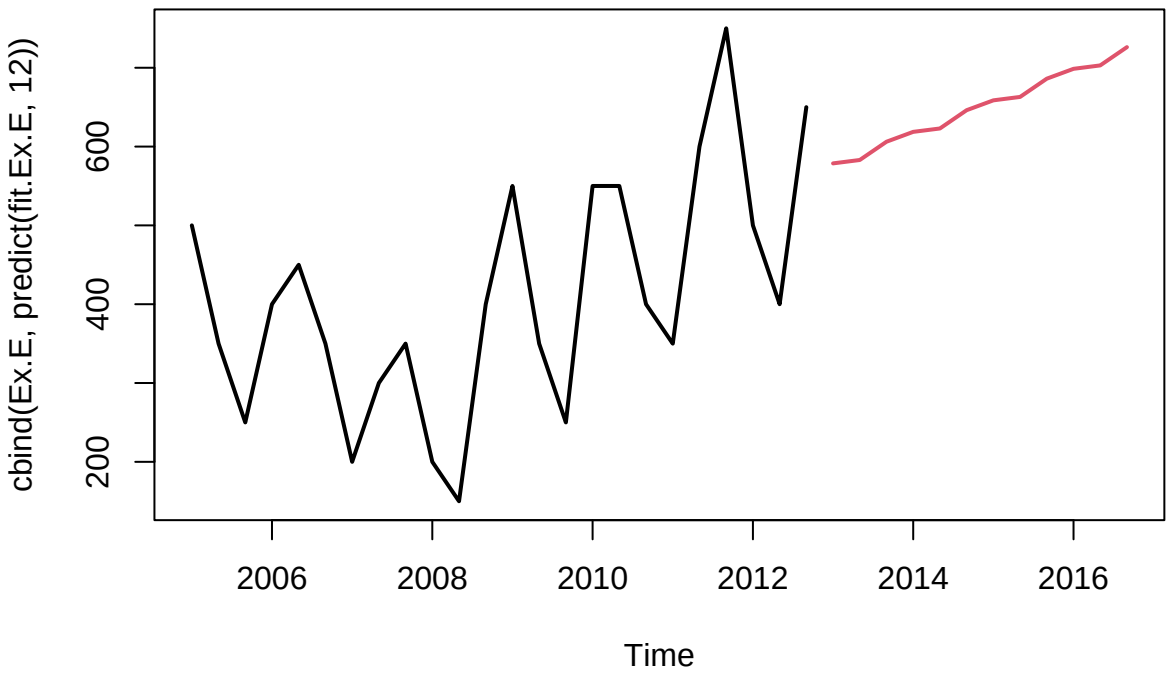

### B) ldeaths

```
plot(cbind(ldeaths, predict(fit.ldeaths, 12)), plot.type='single', col=c(1,2), lwd=2)
```




### B) Sales

```
plot(cbind(Ex.E, predict(fit.Ex.E, 12)), plot.type='single', col=c(1,2), lwd=2)
```


cbind(Ex.E, predict(fit.Ex.E, 12))
200
400
600
2006
2008
2010
2012
2014
2016
Time