oefeningen hoofdstuk 8 - tijdsreeksen

TijsMartens

12 april 2019

voorbeelden van in de les.

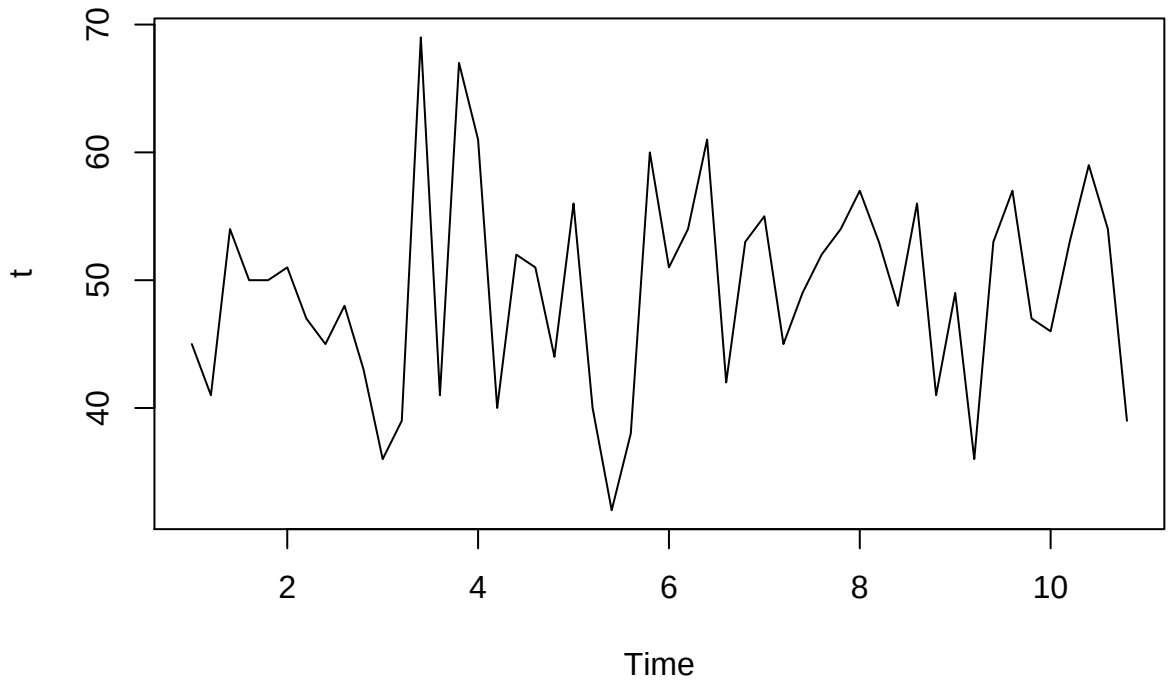
```
x <- round(rnorm(50, 50, 10))
x
```

```
##  [1] 45 41 54 50 50 51 47 45 48 43 36 39 69 41 67 61 40 52 51 44 56 40 32
## [24] 38 60 51 54 61 42 53 55 45 49 52 54 57 53 48 56 41 49 36 53 57 47 46
## [47] 53 59 54 39
```

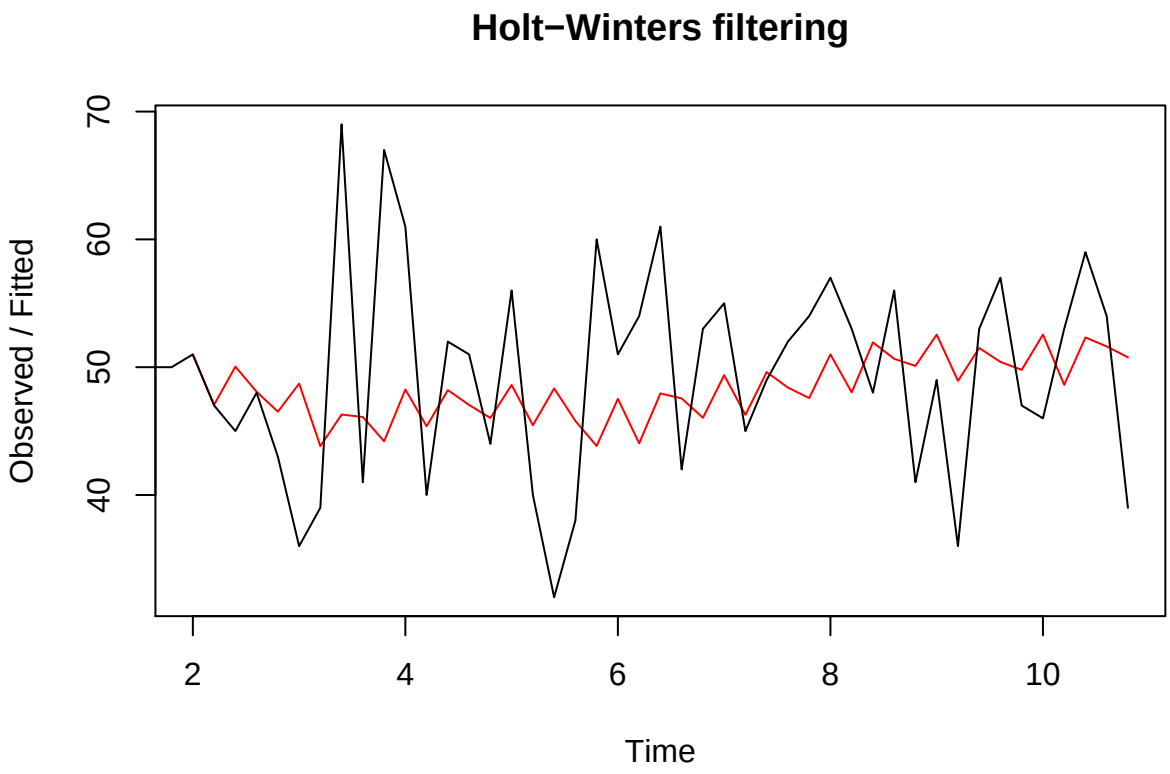
```
t <- ts(x, frequency = 5)
t
```

```
## Time Series:
## Start = c(1, 1)
## End = c(10, 5)
## Frequency = 5
##  [1] 45 41 54 50 50 51 47 45 48 43 36 39 69 41 67 61 40 52 51 44 56 40 32
## [24] 38 60 51 54 61 42 53 55 45 49 52 54 57 53 48 56 41 49 36 53 57 47 46
## [47] 53 59 54 39
```

```
plot(t)
```




```
fit <- HoltWinters(t)

plot(fit)
```

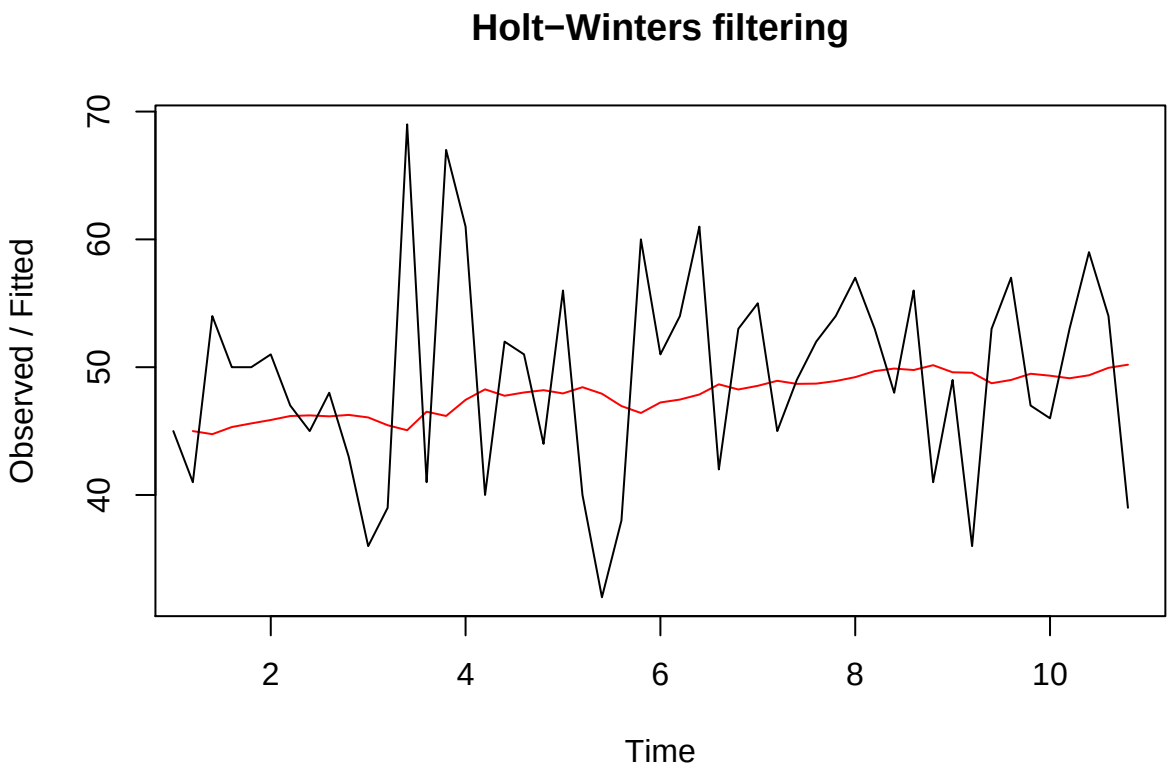



```
fit
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = t)
##
## Smoothing parameters:
##  alpha: 0.06358497
##  beta : 0.1283543
##  gamma: 0
##
## Coefficients:
##          [,1]
## a  51.1635951
## b   0.1002125
## s1  1.6600000
## s2 -1.9400000
## s3  1.3600000
## s4  0.0600000
## s5 -1.1400000
```

merk op dat alpha heel klein is

```
fit <- HoltWinters(t,beta = FALSE, gamma =FALSE)

plot(fit)
```




```
fit
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = t, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.06059083
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 49.51753
```

merk op dat alpha groter is

voorpselling maken

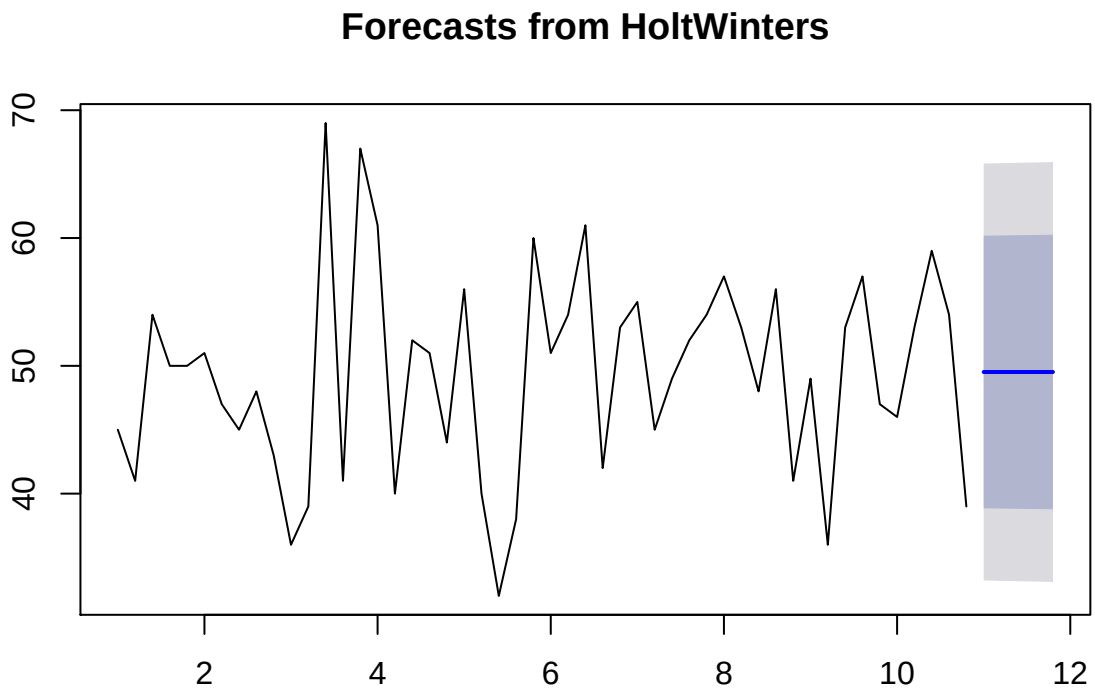
```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.5.3
```

```
f <- forecast(fit, 5)
f
```

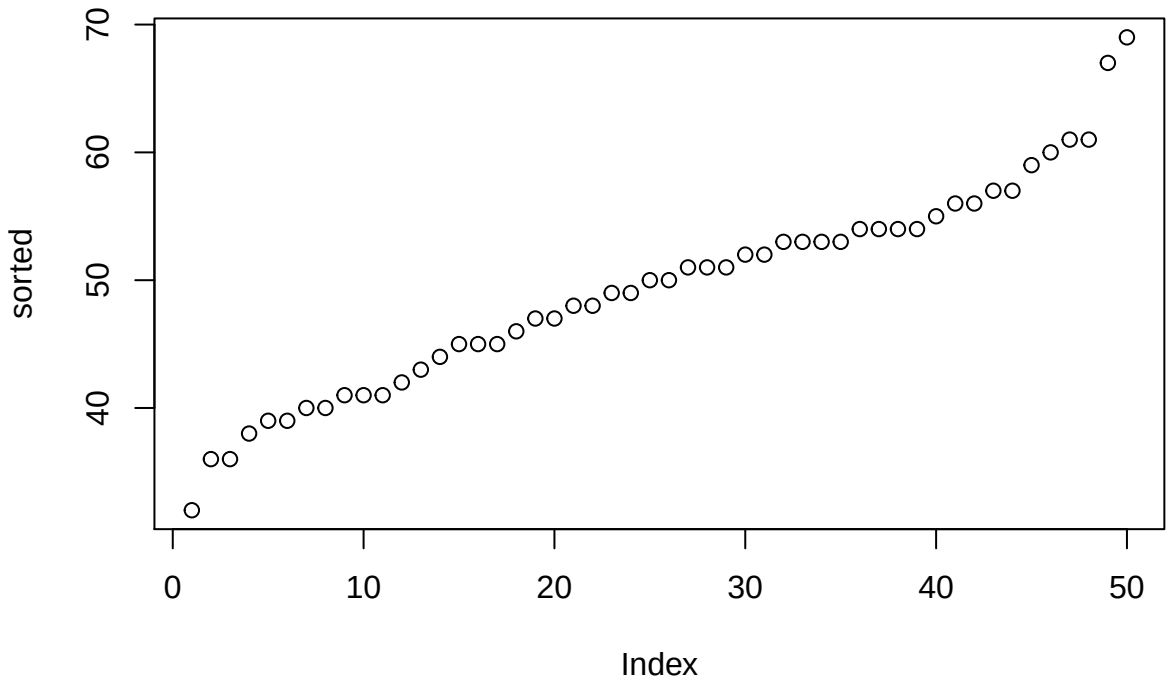
```
##       Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 11.00       49.51753 38.85381 60.18125 33.20877 65.82628
## 11.20       49.51753 38.83425 60.20080 33.17886 65.85619
## 11.40       49.51753 38.81473 60.22032 33.14901 65.88604
## 11.60       49.51753 38.79524 60.23981 33.11921 65.91584
## 11.80       49.51753 38.77579 60.25926 33.08946 65.94559
```

```
plot(f)
```



voorbeelde met stijgende data

```
sorted <- sort(x)
plot(sorted)
```



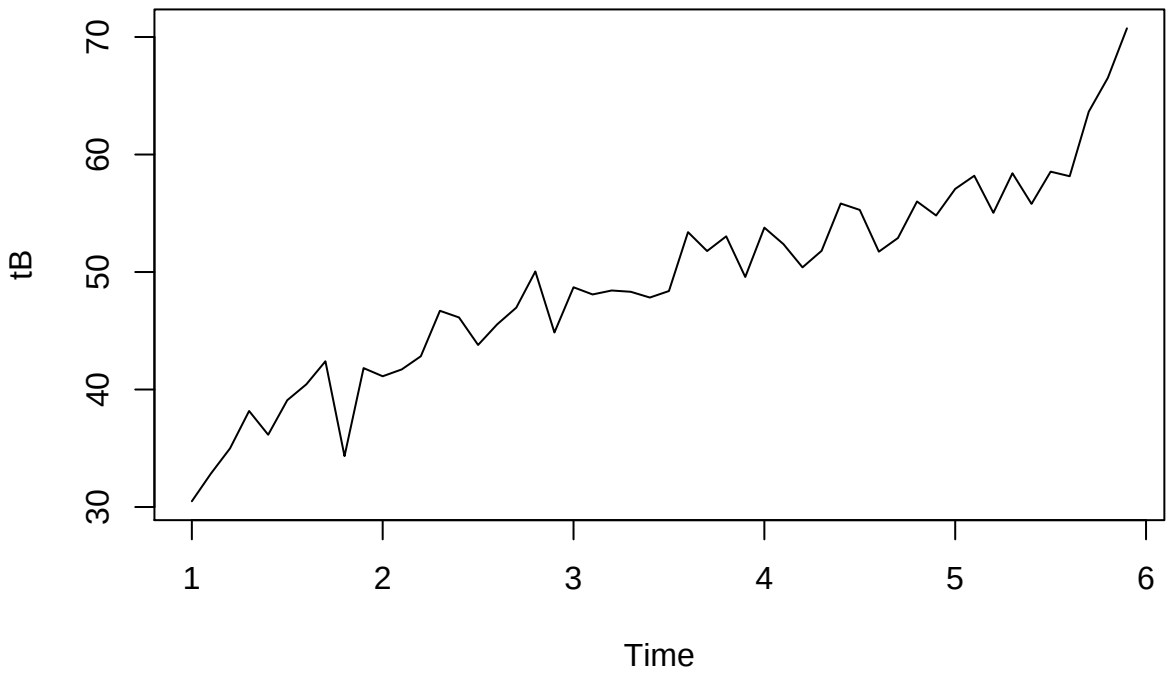

```
# dataset laten veriëren
```

dataset laten veriëren:

```
eta <- rnorm(50,0,2)
betaTS <- sorted + eta
```

tijdsreeks maken

```
tB <- ts(betaTS, frequency = 10)
plot(tB)
```

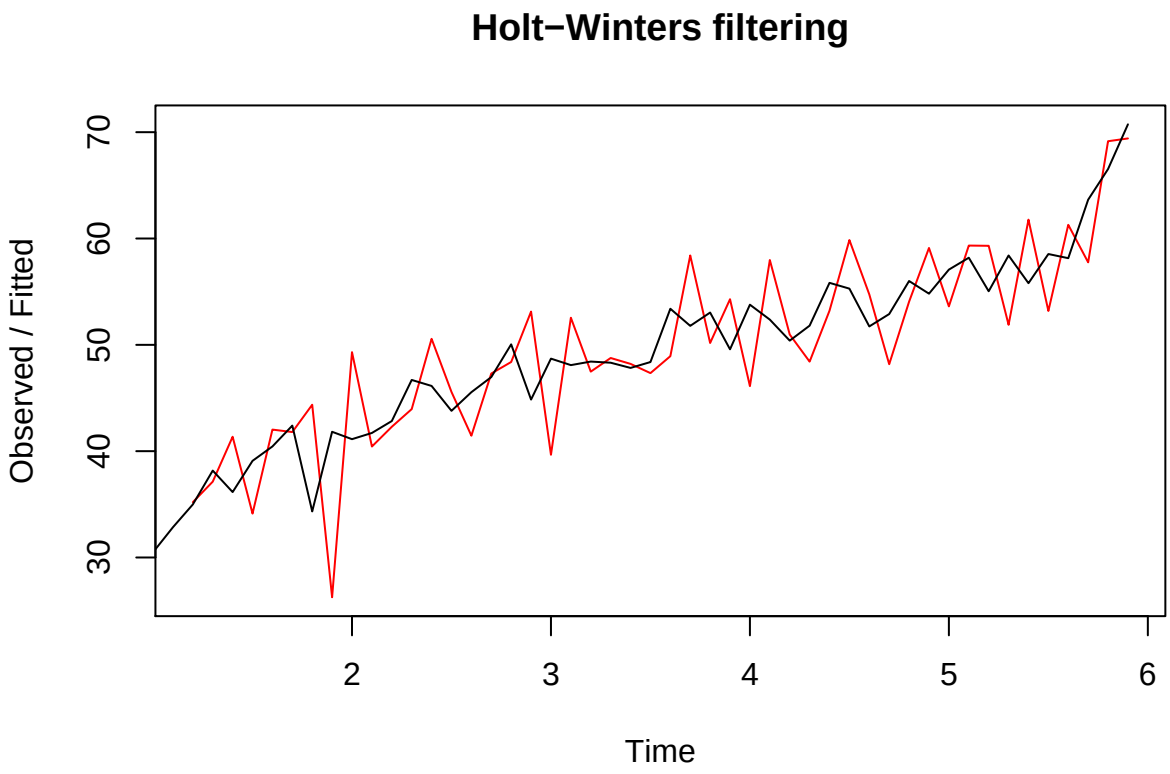


de grafiek stijgt, dus we moeten beta gerbuiken

```
fitB <- HoltWinters(tB, alpha = TRUE, beta = TRUE, gamma=FALSE)
fitB
```

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = tB, alpha = TRUE, beta = TRUE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: TRUE
##  beta : TRUE
##  gamma: FALSE
##
## Coefficients:
##        [,1]
## a 70.737596
## b  4.206424
```

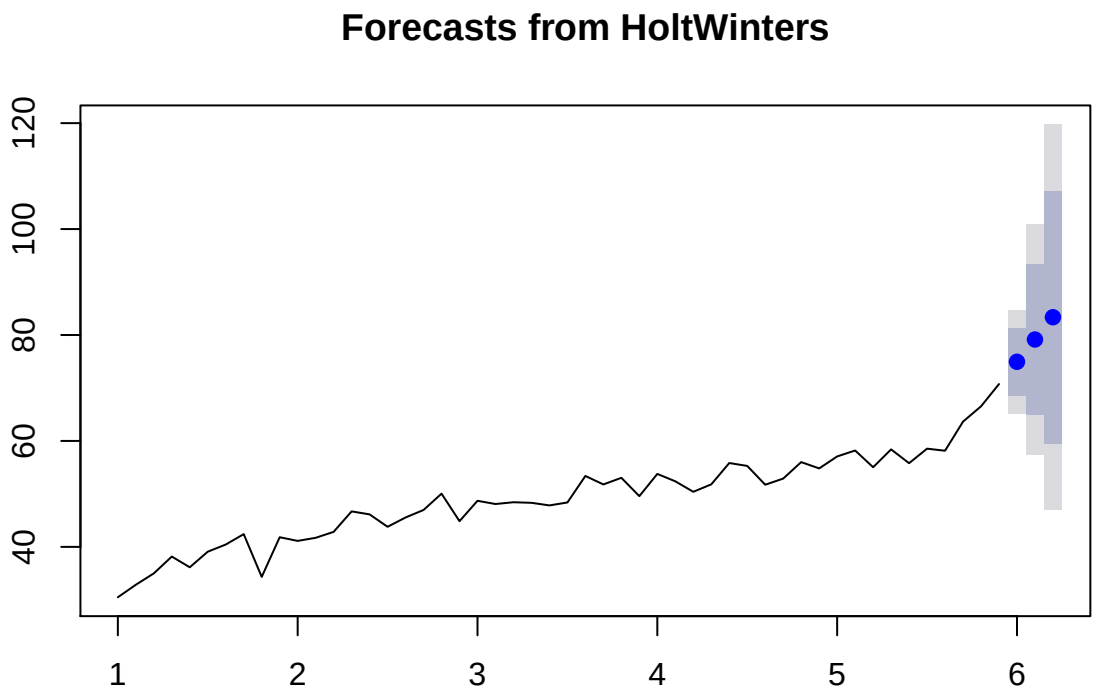
```
plot(fitB)
```



de eerste drie zijn niet ingevuld, want er zijn minstens drie voorgaande observaties nodig om een voorspelling te kunnne maken.

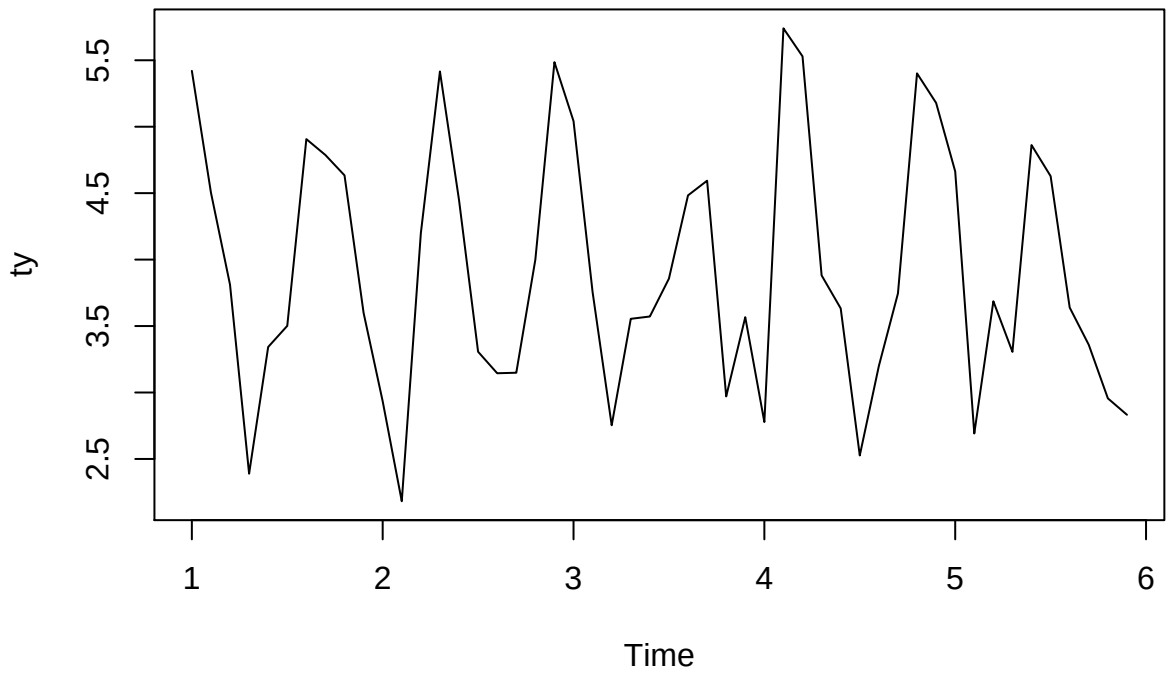
voorspelling maken

```
fB <- forecast(fitB, 3)
plot(fB)
```

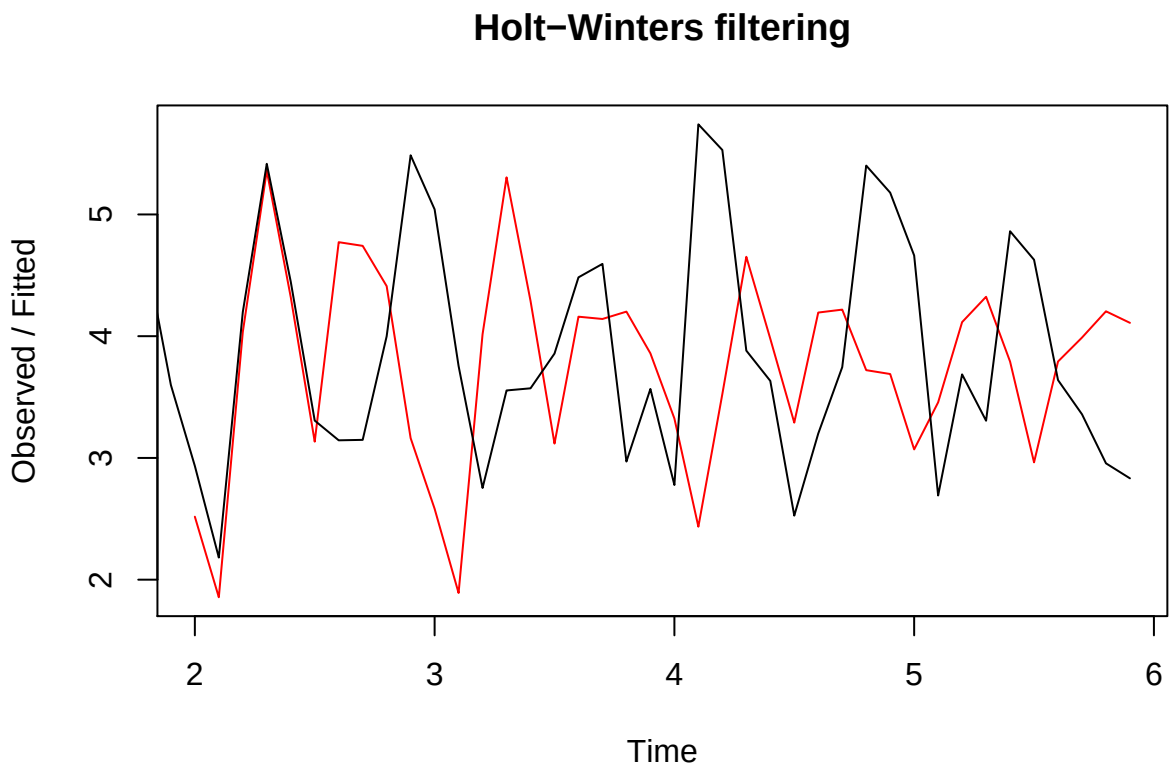


werken met een seizoenscomponent

```
x <- seq(1:50)
y <- sin(x)
eta <- rnorm(50,0, 0.5)
y <- y + 4 + eta
ty <- ts(y, frequency = 10)
plot(ty)
```



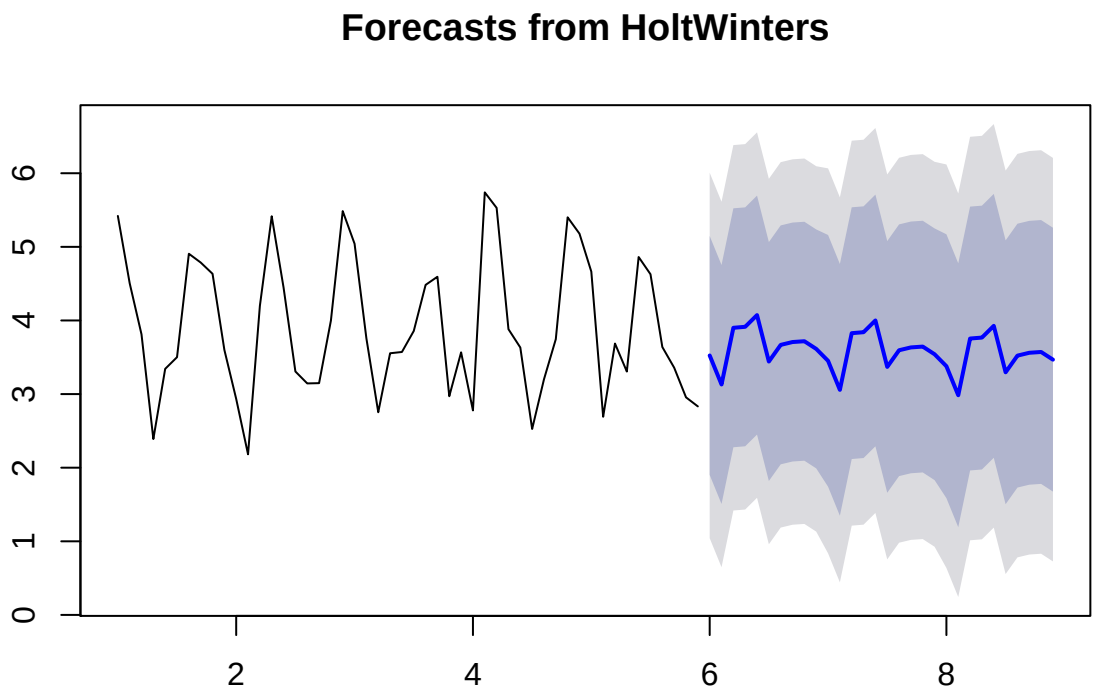

```
my <- HoltWinters(ty)
plot(my)
```




```
my
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = ty)
##
## Smoothing parameters:
##  alpha: 0
##  beta : 0
##  gamma: 0.3311726
##
## Coefficients:
##             [,1]
## a    3.579131953
## b   -0.007301171
## s1  -0.047308631
## s2  -0.434253903
## s3   0.343205317
## s4   0.363764733
## s5   0.530340287
## s6  -0.093358827
## s7   0.140348002
## s8   0.186296910
## s9   0.204546148
## s10  0.107455830
```

```
##voorspelling
fy <- forecast(my, 30)
plot(fy)
```



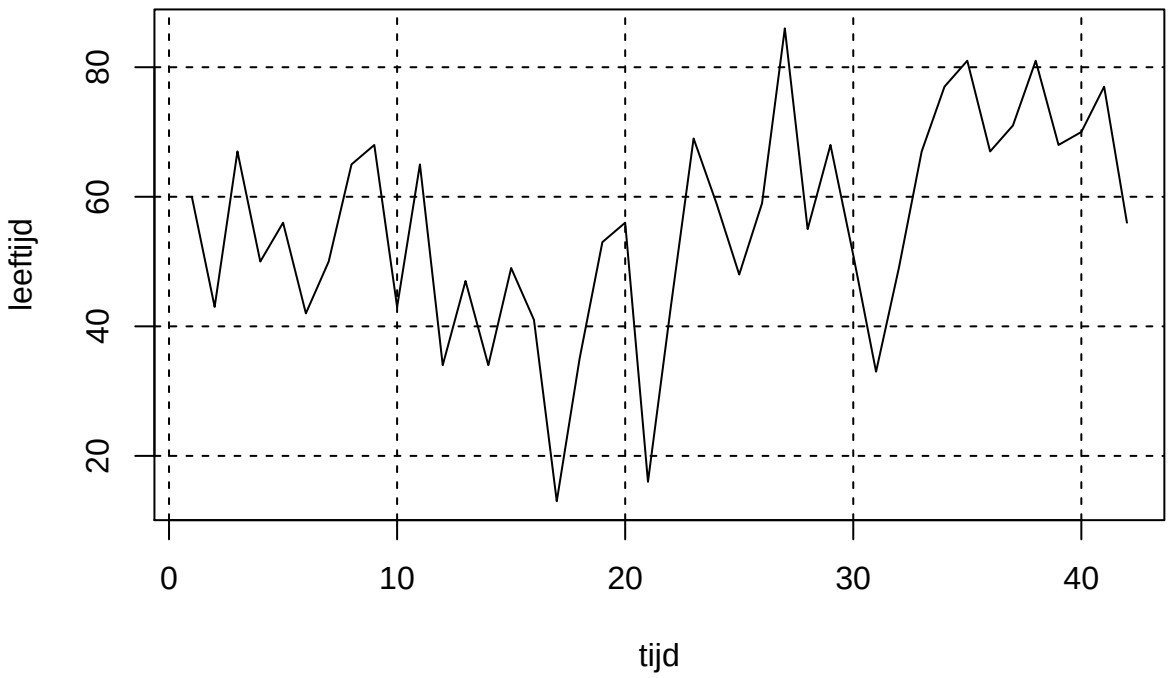
merk op dat de periodes (de dalen en de pieken) worden meegenomen in de voorspelling

voorbeeld blz 120: het voorstellen van een tijdsreeks

```
kings <- scan(file = "C:\\Users\\tijsm\\Google Drive\\HoGent 2018-2019\\2e semester\\Onderzoekstechniek

kings
```

```
##  [1] 60 43 67 50 56 42 50 65 68 43 65 34 47 34 49 41 13 35 53 56 16 43 69
## [24] 59 48 59 86 55 68 51 33 49 67 77 81 67 71 81 68 70 77 56
```

```
kingstimeseries <- ts(kings)

plot.ts(kingstimeseries, ylab = "leeftijd", xlab="tijd")
grid(lty = 2, lwd = 1, col = "black")
```



voorbeeld blz 123 ev. : voorschrijdend gemiddelde

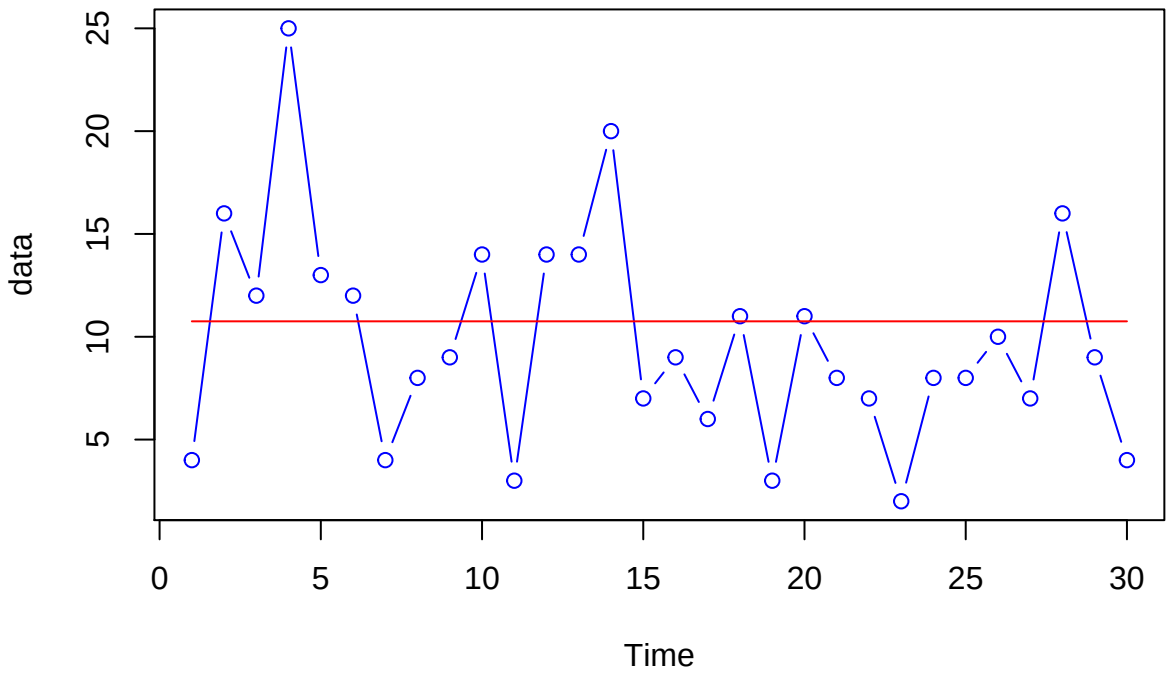
```
data <- c(4 , 16 , 12 , 25 , 13 , 12 , 4 , 8 , 9 , 14, 3 , 14 , 14 , 20 , 7 , 9 , 6 , 11 , 3 , 11, 8 ,

testData <- c(4 , 16 , 12 , 25 , 13 , 12 , 4 , 8 , 9 , 14, 3 , 14 , 14 , 20 , 7 , 9 , 6 , 11 , 3 , 11)

gem <- mean(data[1:20]) # het gemiddelde van de eerste 20 getallen
gem
```

```
## [1] 10.75
```

```
mean(testData)
```

```
## [1] 10.75
```

```
gemy <- rnorm(n = length(data), mean = gem, sd = 0)

plot.ts(data, type = "b", col = "blue")
lines(gemy, type = "l", col = "red")
```



we merken dat x1 = 4 (de eerste waarde) evenveel invloed heeft op het gemiddelde als x20 = 11

het gemiddelde als schatter gebruiken is dus geen goed idee

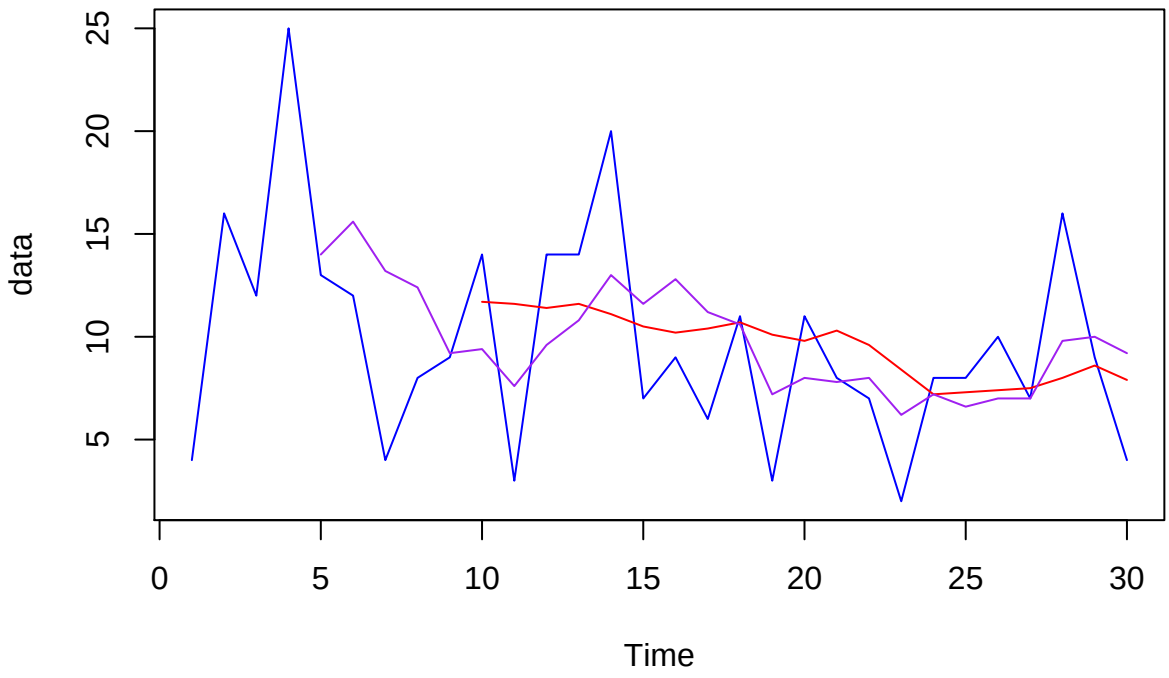
```
library(TTR)
```

```
## Warning: package 'TTR' was built under R version 3.5.3
```

```
library(forecast)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
sma10 <- SMA(x =data,n=10)
sma5 <- SMA(x=data,n=5)
plot.ts(x = data, col = 'blue',type = 'l')
lines(sma10, col='red', type = 'l')
lines(sma5, col='purple', type = 'l')
```



voorbeeld blz 126: enkelvoudige exponentiële afvakking

```
rain <- kings <- scan(file = "C:\\Users\\tijsm\\Google Drive\\HoGent 2018-2019\\2e semester\\Onderzoeks

rain
```

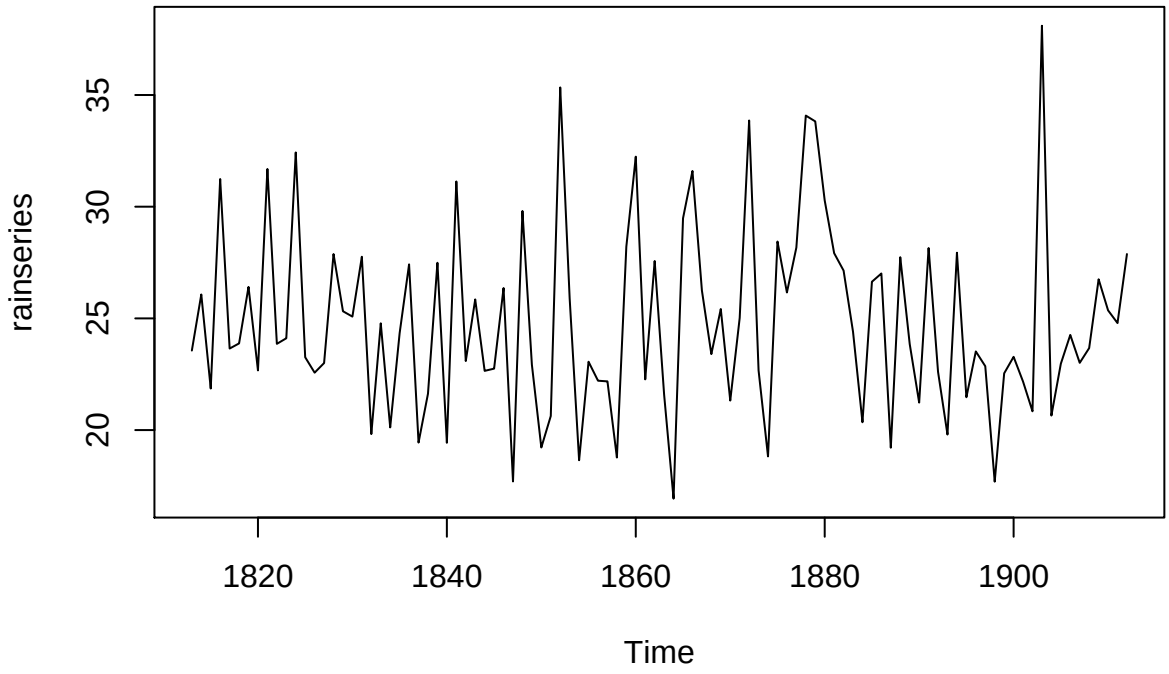
```
##   [1] 23.56 26.07 21.86 31.24 23.65 23.88 26.41 22.67 31.69 23.86 24.11
##  [12] 32.43 23.26 22.57 23.00 27.88 25.32 25.08 27.76 19.82 24.78 20.12
##  [23] 24.34 27.42 19.44 21.63 27.49 19.43 31.13 23.09 25.85 22.65 22.75
##  [34] 26.36 17.70 29.81 22.93 19.22 20.63 35.34 25.89 18.65 23.06 22.21
##  [45] 22.18 18.77 28.21 32.24 22.27 27.57 21.59 16.93 29.48 31.60 26.25
##  [56] 23.40 25.42 21.32 25.02 33.86 22.67 18.82 28.44 26.16 28.17 34.08
##  [67] 33.82 30.28 27.92 27.14 24.40 20.35 26.64 27.01 19.21 27.74 23.85
##  [78] 21.23 28.15 22.61 19.80 27.94 21.47 23.52 22.86 17.69 22.54 23.28
##  [89] 22.17 20.84 38.10 20.65 22.97 24.26 23.01 23.67 26.75 25.36 24.79
## [100] 27.88
```

```
rainseries <- ts(rain, start = c(1813))
rainseries
```

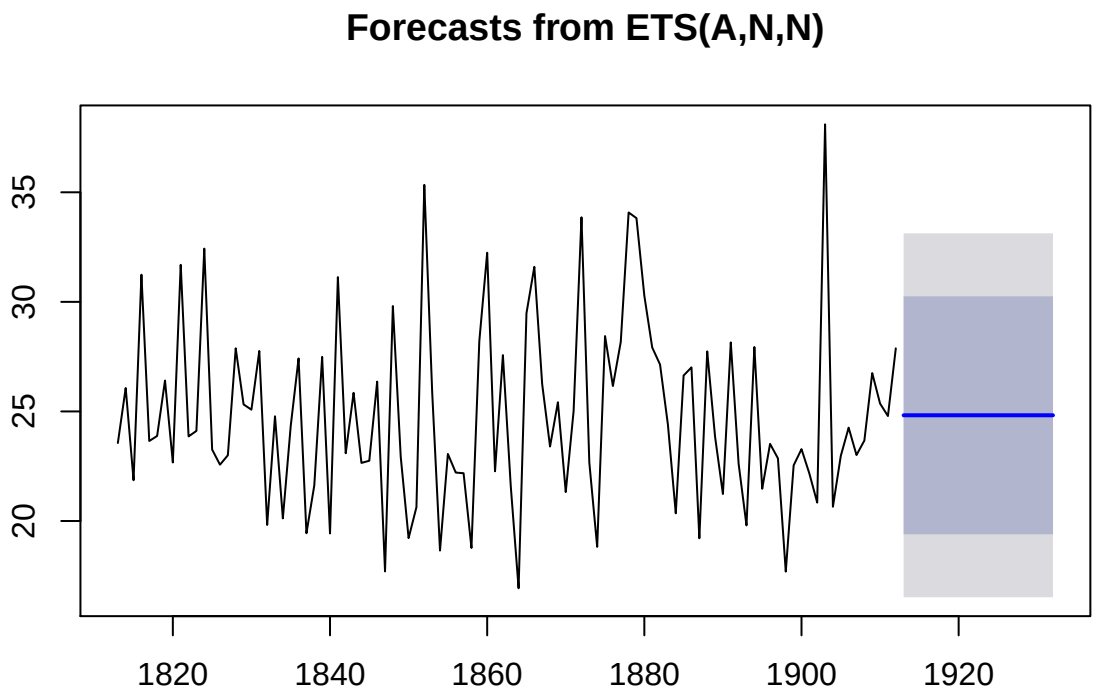
```
## Time Series:
## Start = 1813
## End = 1912
## Frequency = 1
##   [1] 23.56 26.07 21.86 31.24 23.65 23.88 26.41 22.67 31.69 23.86 24.11
##  [12] 32.43 23.26 22.57 23.00 27.88 25.32 25.08 27.76 19.82 24.78 20.12
```

```
##  [23] 24.34 27.42 19.44 21.63 27.49 19.43 31.13 23.09 25.85 22.65 22.75
##  [34] 26.36 17.70 29.81 22.93 19.22 20.63 35.34 25.89 18.65 23.06 22.21
##  [45] 22.18 18.77 28.21 32.24 22.27 27.57 21.59 16.93 29.48 31.60 26.25
##  [56] 23.40 25.42 21.32 25.02 33.86 22.67 18.82 28.44 26.16 28.17 34.08
##  [67] 33.82 30.28 27.92 27.14 24.40 20.35 26.64 27.01 19.21 27.74 23.85
##  [78] 21.23 28.15 22.61 19.80 27.94 21.47 23.52 22.86 17.69 22.54 23.28
##  [89] 22.17 20.84 38.10 20.65 22.97 24.26 23.01 23.67 26.75 25.36 24.79
## [100] 27.88
```

```
plot.ts(rainseries)
```




```
rainseriesforecasts <- forecast(rainseries, h=20)
plot(rainseriesforecasts) ## ?????????????????????? dit staat niet cursus hoe we aan rainseriesforecasts
```



oefening 8.1.

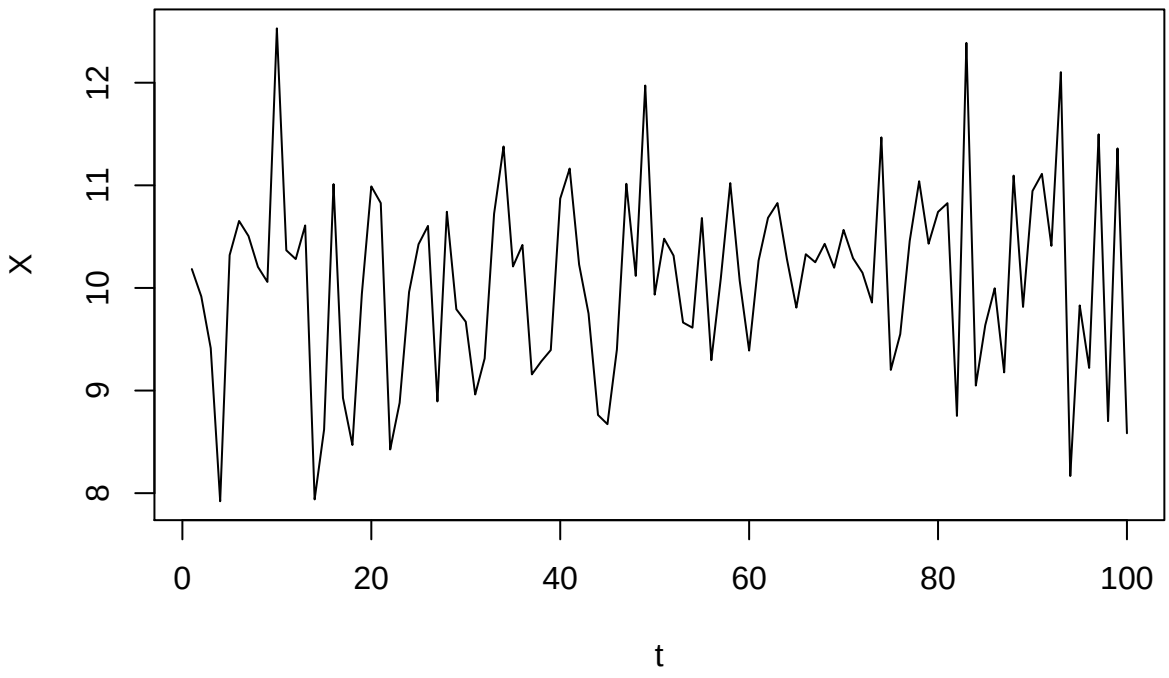
opgave

wat zou volgende tijdsreeks kunnen voorstellen

oplossing

tijdsreeks grafisch voorstellen

```
f <- function (a , b , t ){
    return (a + b * sin ((2 * pi*4) / 4) + b * cos ((2 * pi*4) / 4) + rnorm (1) )
}


t <- seq(from = 1, to = 100, by = 1)


X <- lapply (t , f , a=5,b=5)

plot (x = t , y = X, type = 'l')
```



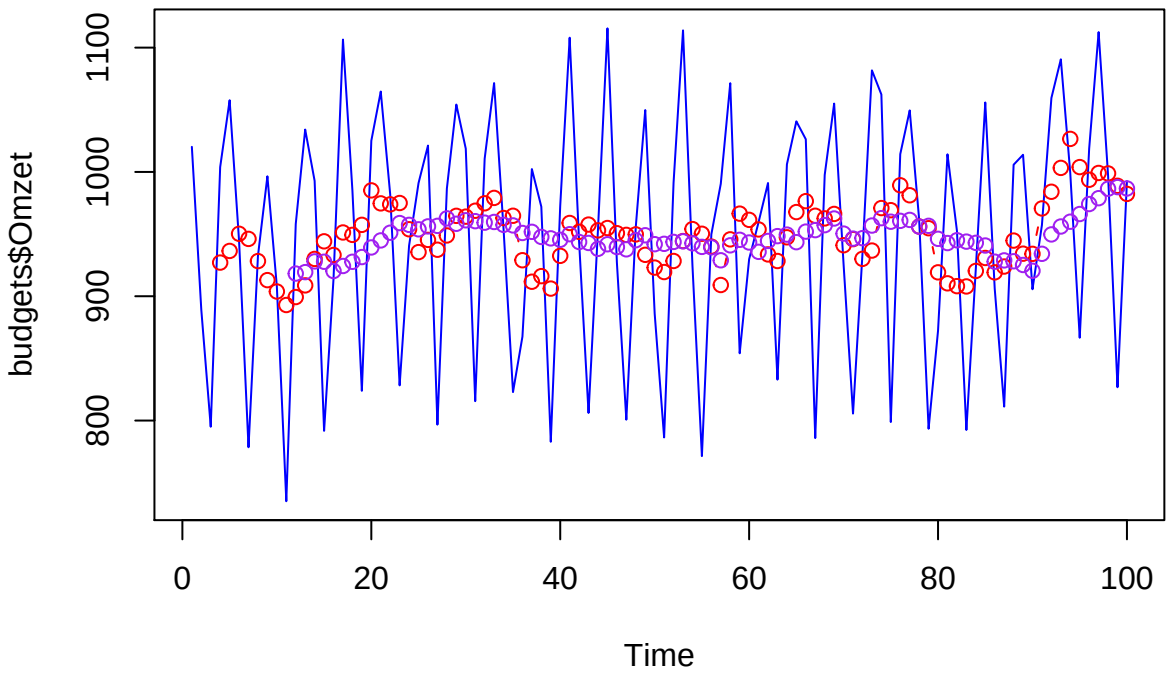
oefening 8.2.

```
budgets <- read.csv("C:\\Users\\tijsm\\Google Drive\\HoGent 2018-2019\\2e semester\\Onderzoekstechnieke
```

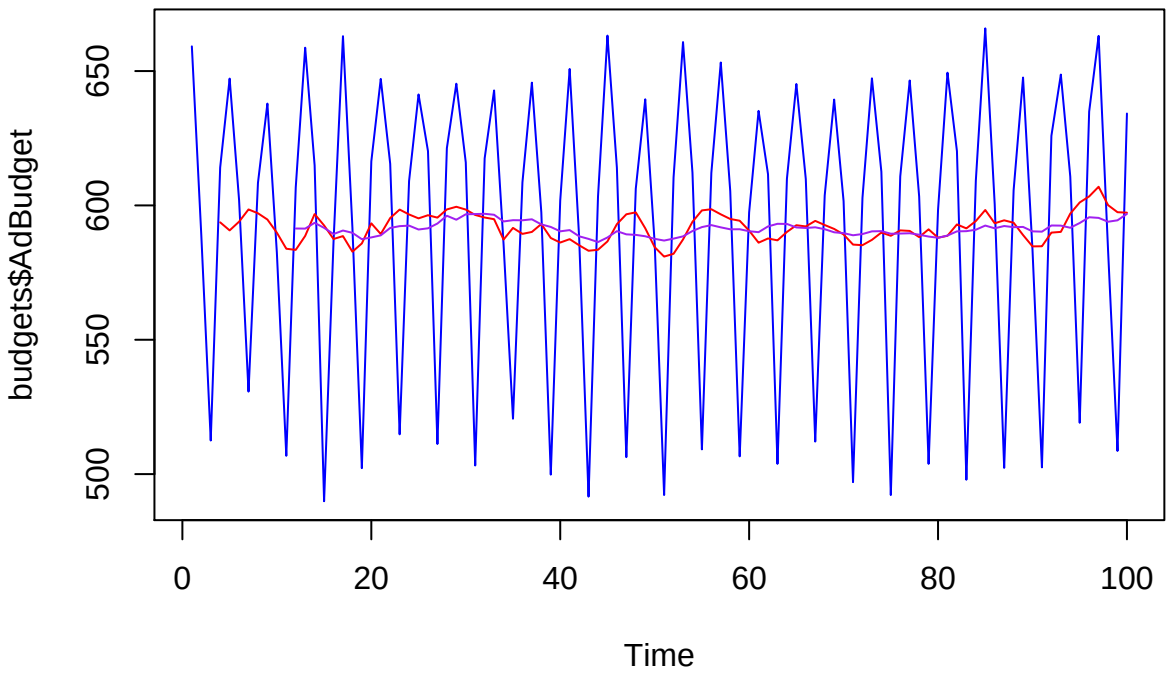
```
library(TTR)
library(forecast)
library(ggplot2)
```

1.

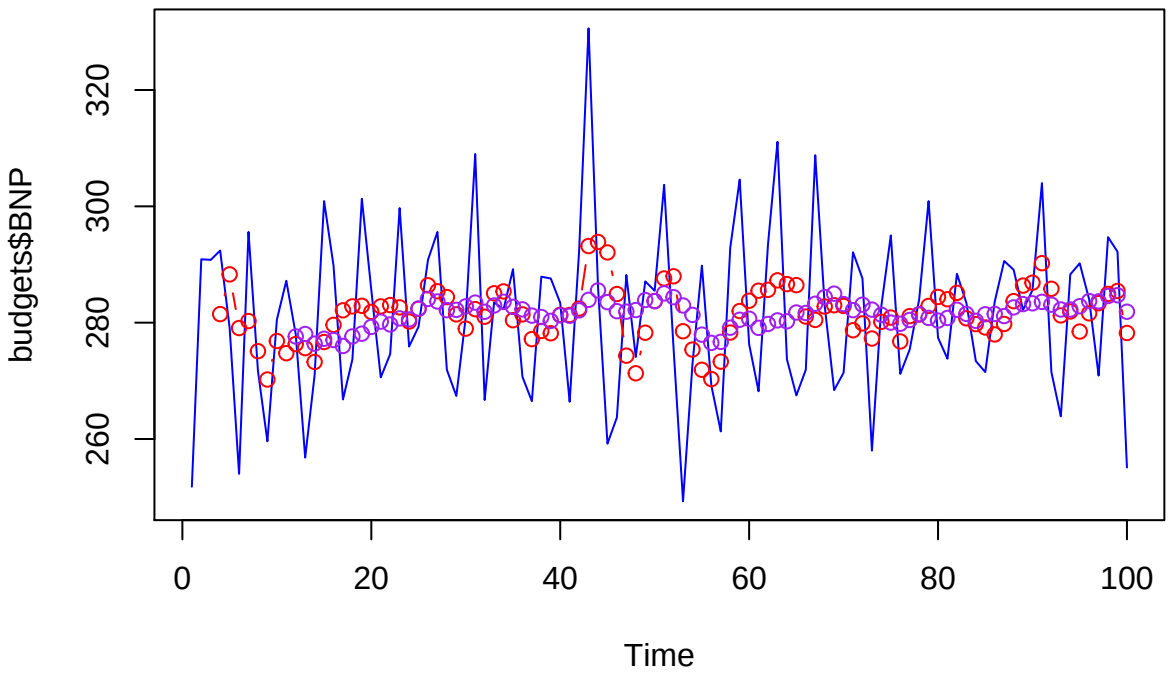
```
omzetsma4 <- SMA(x = budgets$Omzet, n=4)
omzetsma12 <- SMA(x = budgets$Omzet, n=12)
plot.ts(x=budgets$Omzet, col="blue", type = 'l')


lines(omzetsma4, col = "red", type = 'b')
lines(omzetsma12, col = "purple", type = 'b')
```




```
addsBudgetsma4 <- SMA(budgets$AdBudget, n = 4)
addsBudgetsma12 <- SMA(budgets$AdBudget, n = 12)

plot.ts(x = budgets$AdBudget, col='blue', type = 'l')

lines(addsBudgetsma4, col = 'red', type = 'l')
lines(addsBudgetsma12, col = 'purple', type = 'l')
```



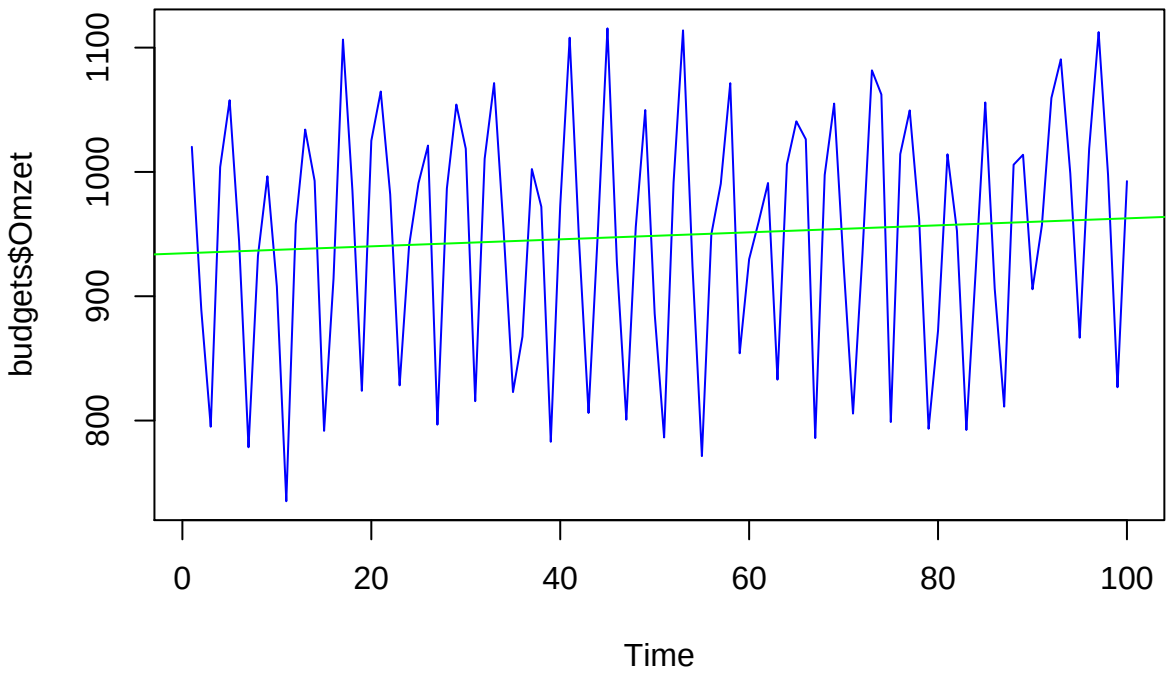

```
bnpsma4 <- SMA(budgets$BNP, n = 4)
bnpsma12 <- SMA(budgets$BNP, n = 12)

plot.ts(budgets$BNP, col = 'blue', type = 'l')

lines(bnpsma4, col = "red", type = 'b')
lines(bnpsma12, col = "purple", type = "b")
```



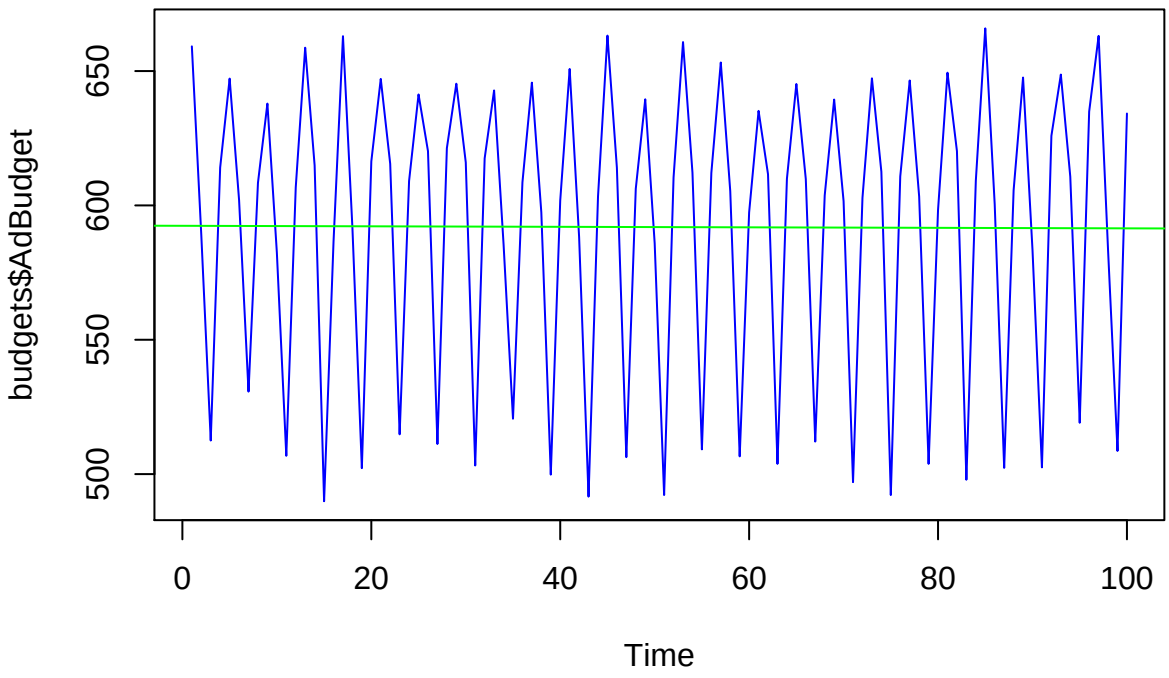
2.

adhv lineaire regressie

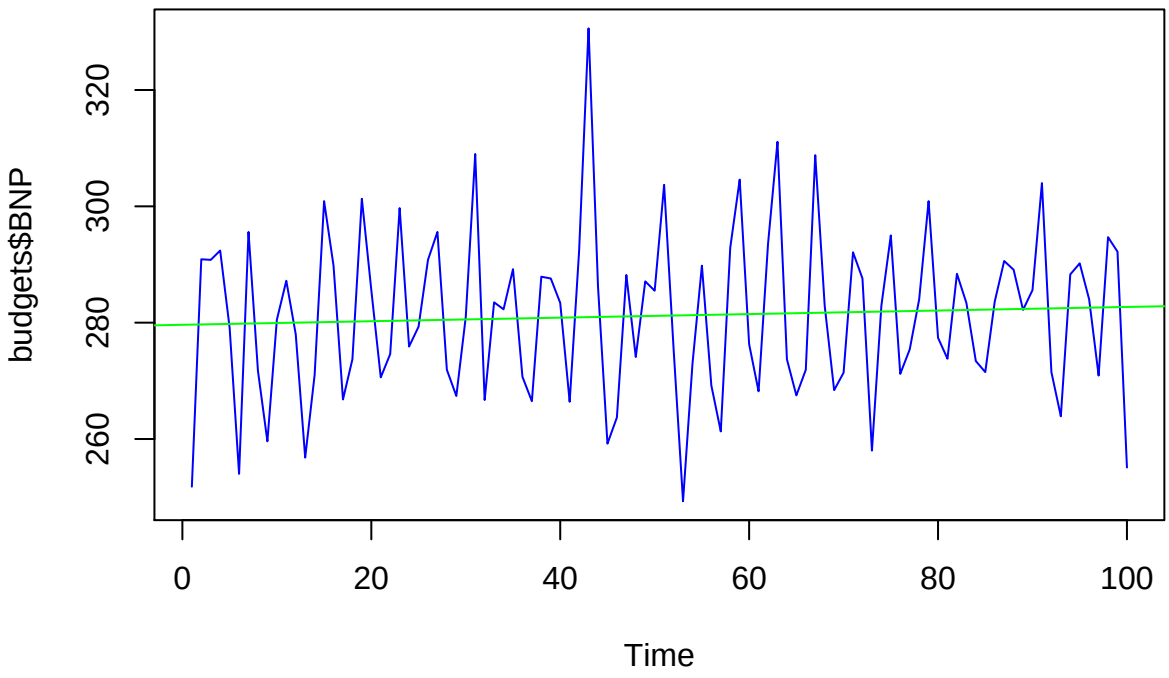
```
linregOmzet <- lm(budgets$Omzet ~ budgets$Kwartaalnummer)
plot.ts(x = budgets$Omzet, col='blue', type = 'l')

abline(linregOmzet, col = 'green')
```




```
linregAddsBudget <- lm(budgets$AdBudget ~ budgets$Kwartaalnummer)
plot.ts(x = budgets$AdBudget, col='blue', type = 'l')

abline(linregAddsBudget, col = 'green')
```




```
linregbnp <- lm(budgets$BNP ~ budgets$Kwartaalnummer)
plot.ts(x = budgets$BNP, col='blue', type = 'l')

abline(linregbnp, col = 'green')
```

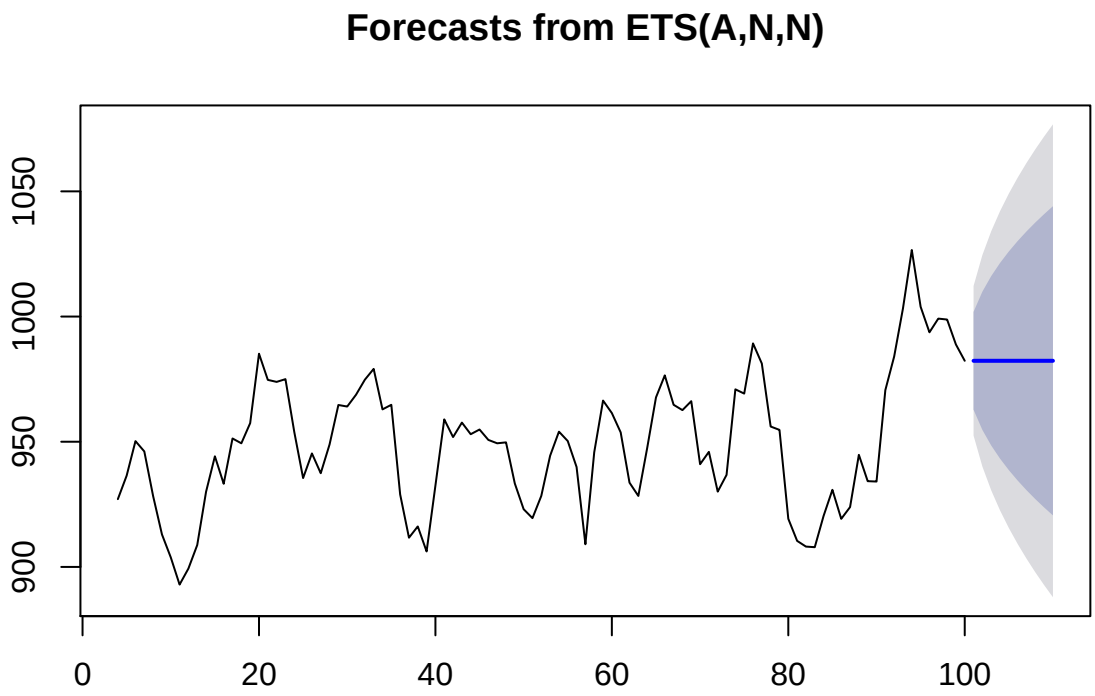


3.

```
voorspellingOmzet4 <- forecast(omzetsma4, h=10)
```

```
## Warning in ets(object, lambda = lambda, biasadj = biasadj,
## allow.multiplicative.trend = allow.multiplicative.trend, : Missing values
## encountered. Using longest contiguous portion of time series
```

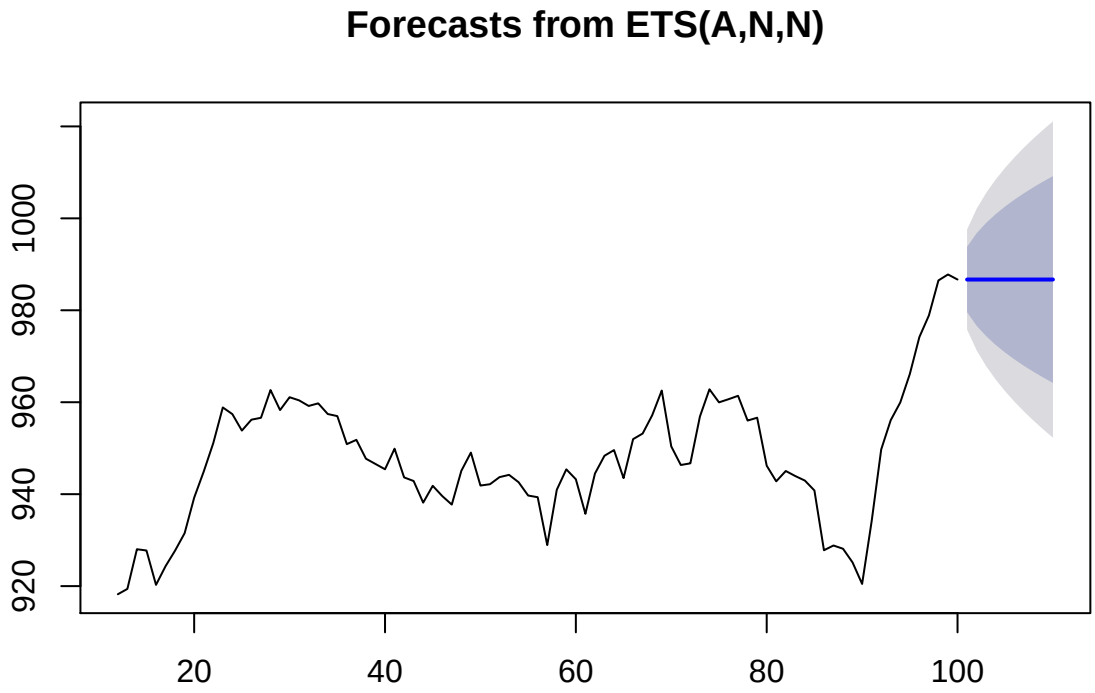
```
plot(voorspellingOmzet4)
```




```
voorspellingOmzet12 <- forecast(omzetsma12, h=10)
```

```
## Warning in ets(object, lambda = lambda, biasadj = biasadj,
## allow.multiplicative.trend = allow.multiplicative.trend, : Missing values
## encountered. Using longest contiguous portion of time series
```

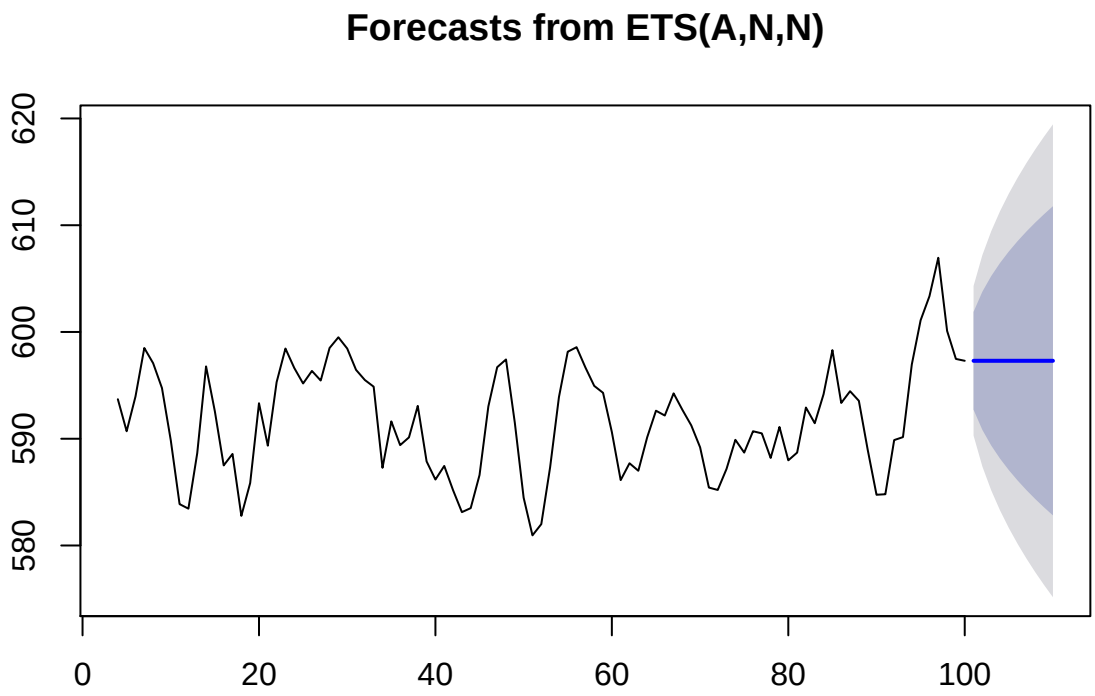
```
plot(voorspellingOmzet12)
```




```
voorspellingAdBudget4 <- forecast(addsBudgetsma4, h=10)
```

```
## Warning in ets(object, lambda = lambda, biasadj = biasadj,
## allow.multiplicative.trend = allow.multiplicative.trend, : Missing values
## encountered. Using longest contiguous portion of time series
```

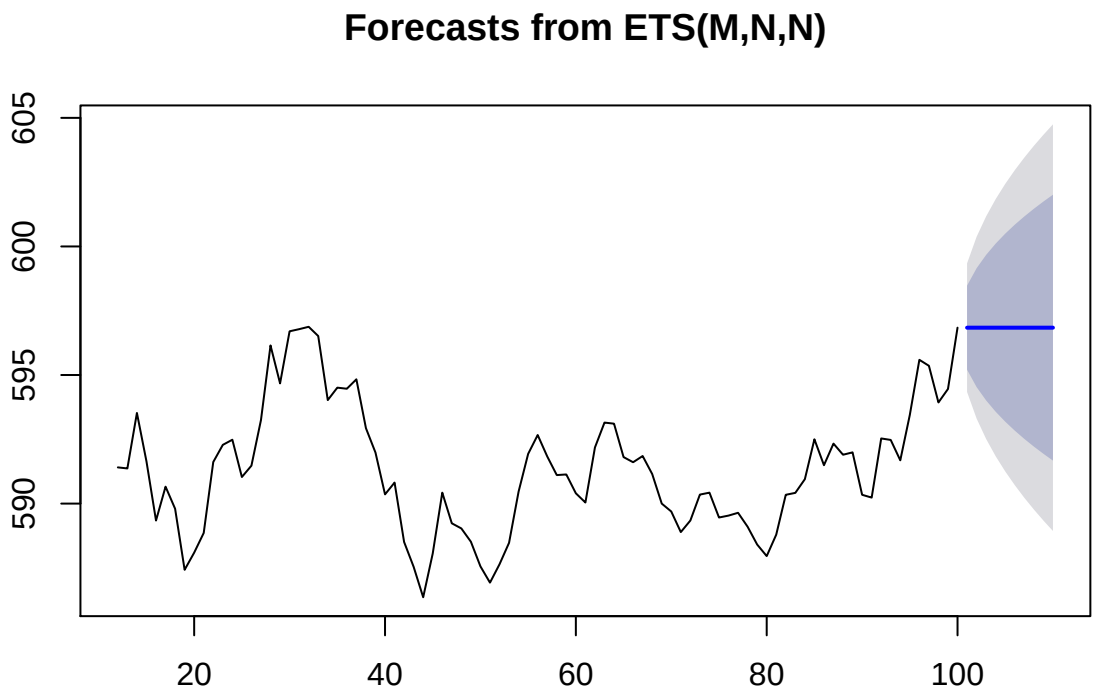
```
plot(voorspellingAdBudget4)
```




```
voorspellingAdBudget12 <- forecast(addsBudgetsma12, h=10)
```

```
## Warning in ets(object, lambda = lambda, biasadj = biasadj,
## allow.multiplicative.trend = allow.multiplicative.trend, : Missing values
## encountered. Using longest contiguous portion of time series
```

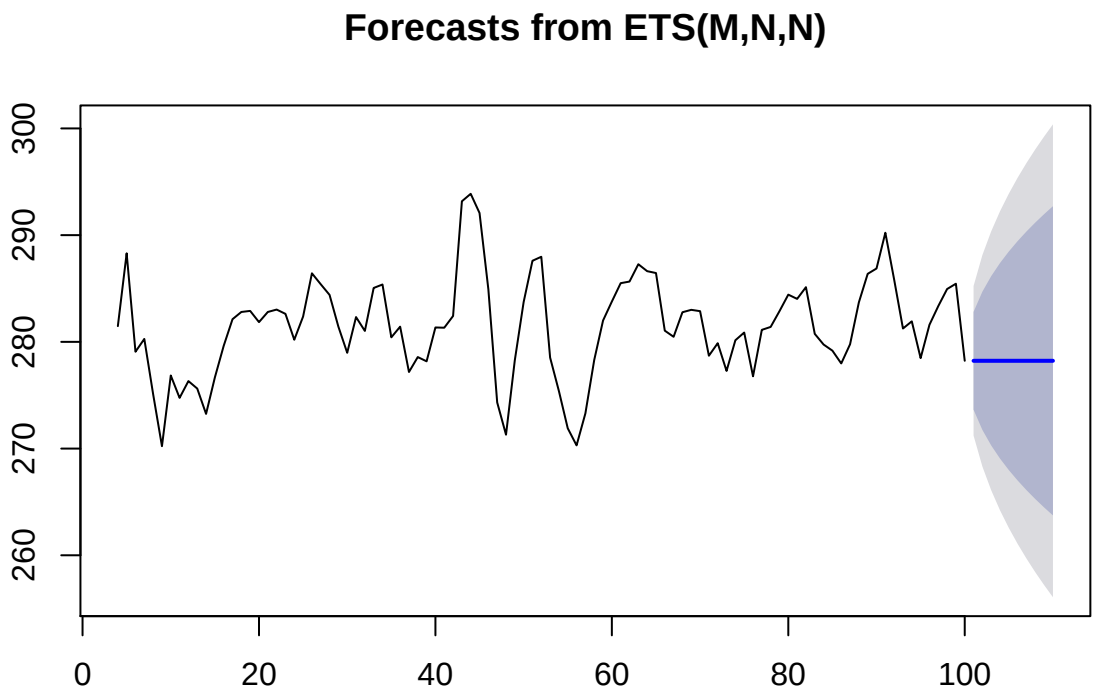
```
plot(voorspellingAdBudget12)
```




```
voorspellingBNP4 <- forecast(bnpsma4, h=10)
```

```
## Warning in ets(object, lambda = lambda, biasadj = biasadj,
## allow.multiplicative.trend = allow.multiplicative.trend, : Missing values
## encountered. Using longest contiguous portion of time series
```

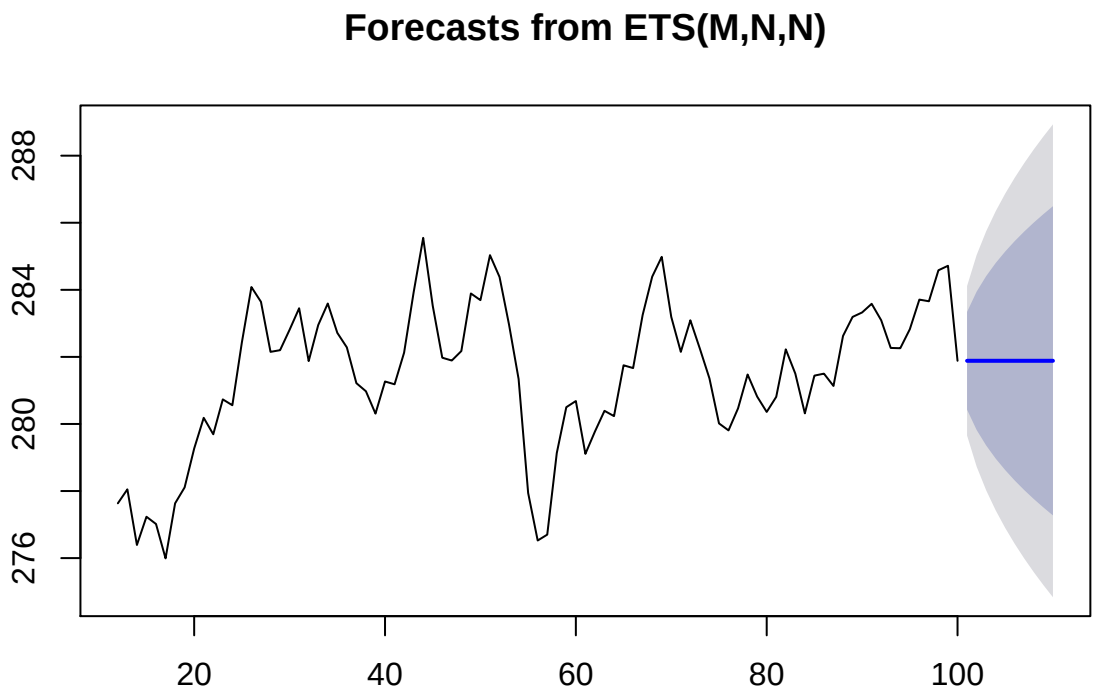
```
plot(voorspellingBNP4)
```




```
voorspellingBNP12 <- forecast(bnpsma12, h=10)
```

```
## Warning in ets(object, lambda = lambda, biasadj = biasadj,
## allow.multiplicative.trend = allow.multiplicative.trend, : Missing values
## encountered. Using longest contiguous portion of time series
```

```
plot(voorspellingBNP12)
```



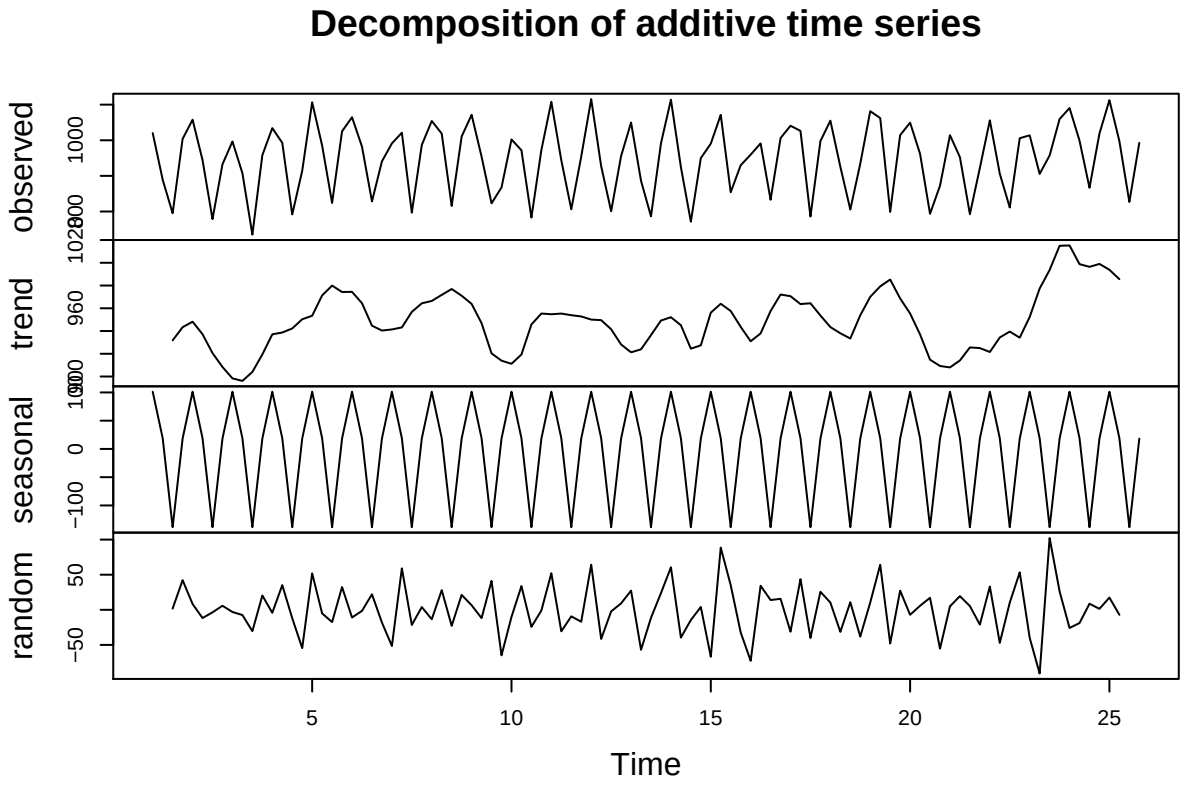
4.

aangezien er 4 kwartalen zijn in een jaar is deze dataset “seasonal”. Voor dit soort data set is driedubbele exponentiële smoothing voorzien.

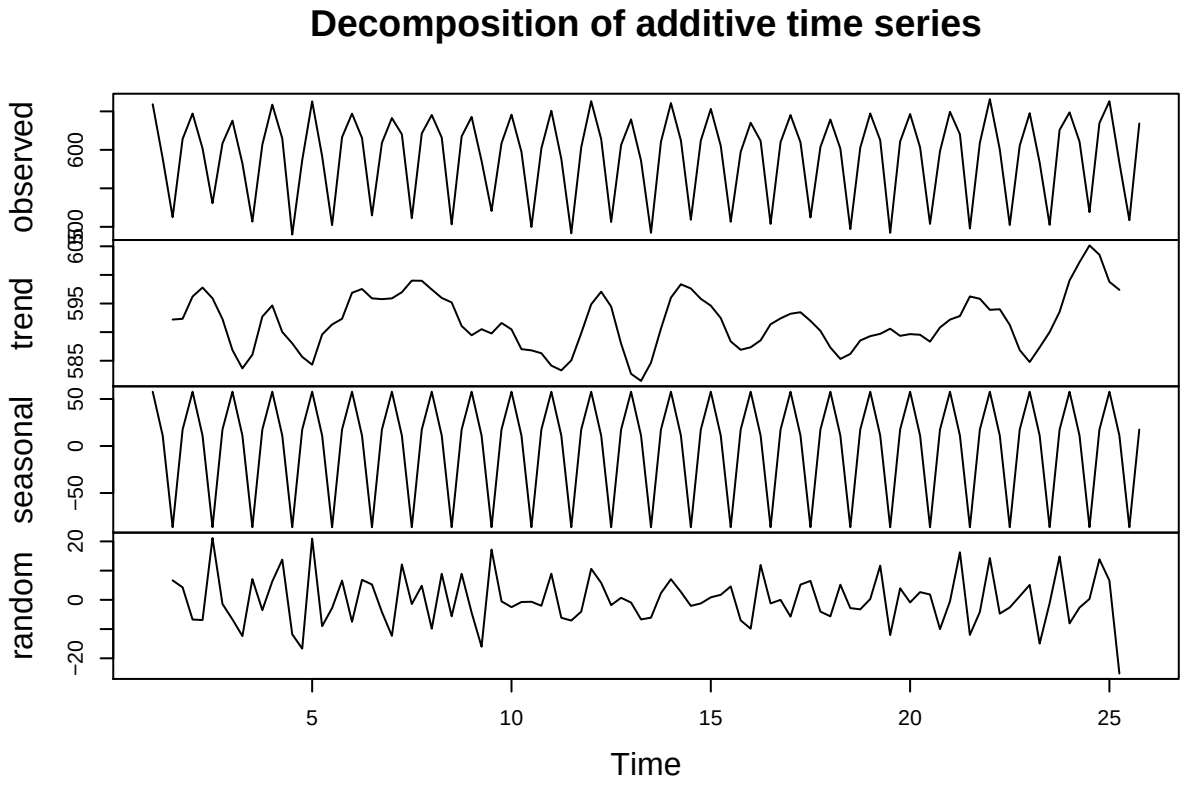
Als er enkel een trend was, zou dubbele exponentiele smoothing volstaan.

5.

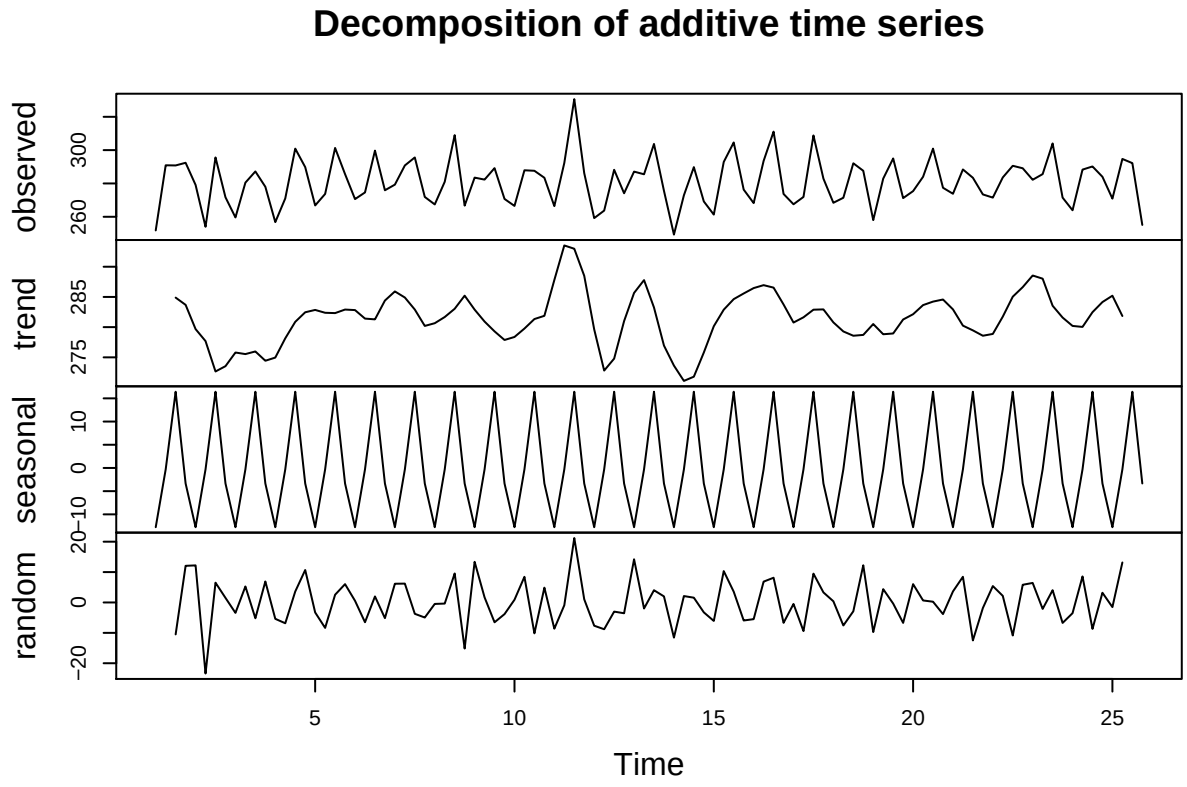
```
omzetTs <- ts(budgets$Omzet, frequency = 4)
decomposedOmzet <- decompose(omzetTs)
plot(decomposedOmzet)
```




```
addsBudgetTs <- ts(budgets$AdBudget, frequency = 4)
decomposedAddsTs <- decompose(addsBudgetTs)
plot(decomposedAddsTs)
```

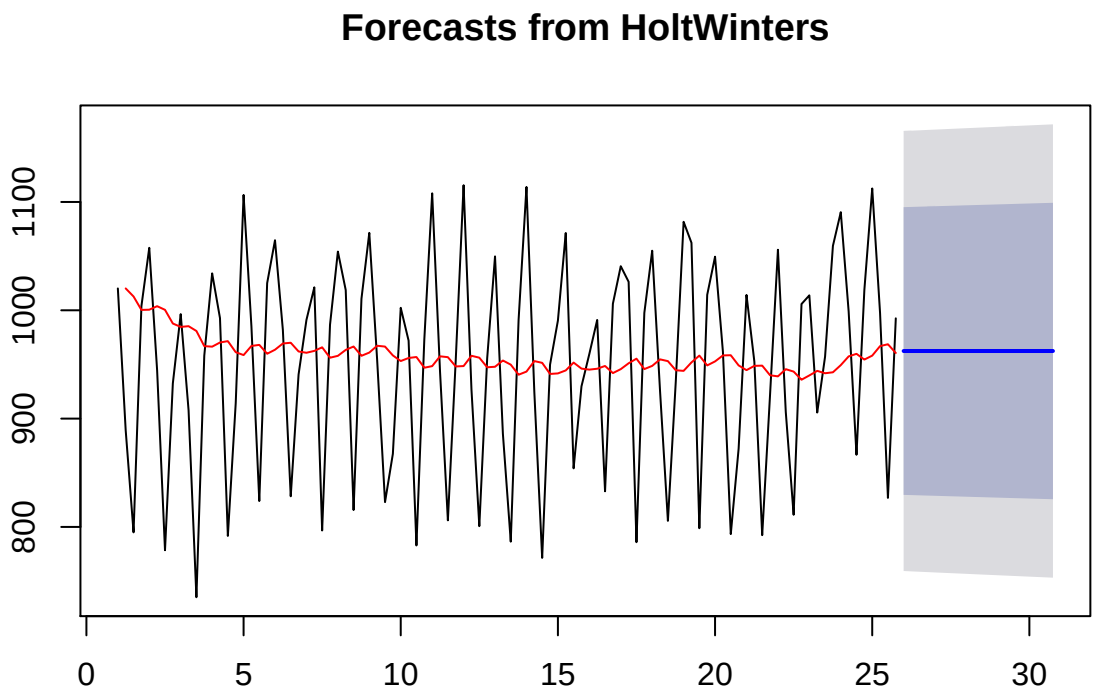



```
bnpTs <- ts(budgets$BNP, frequency = 4)
decomposedBnpTs <- decompose(bnpTs)
plot(decomposedBnpTs)
```

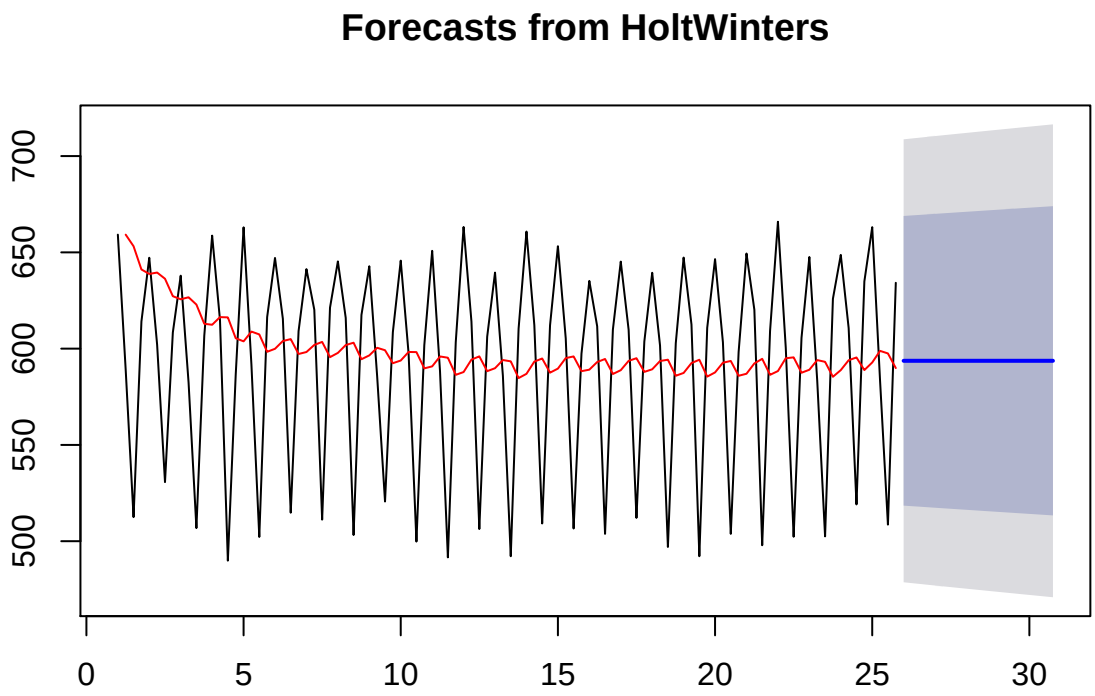


6.

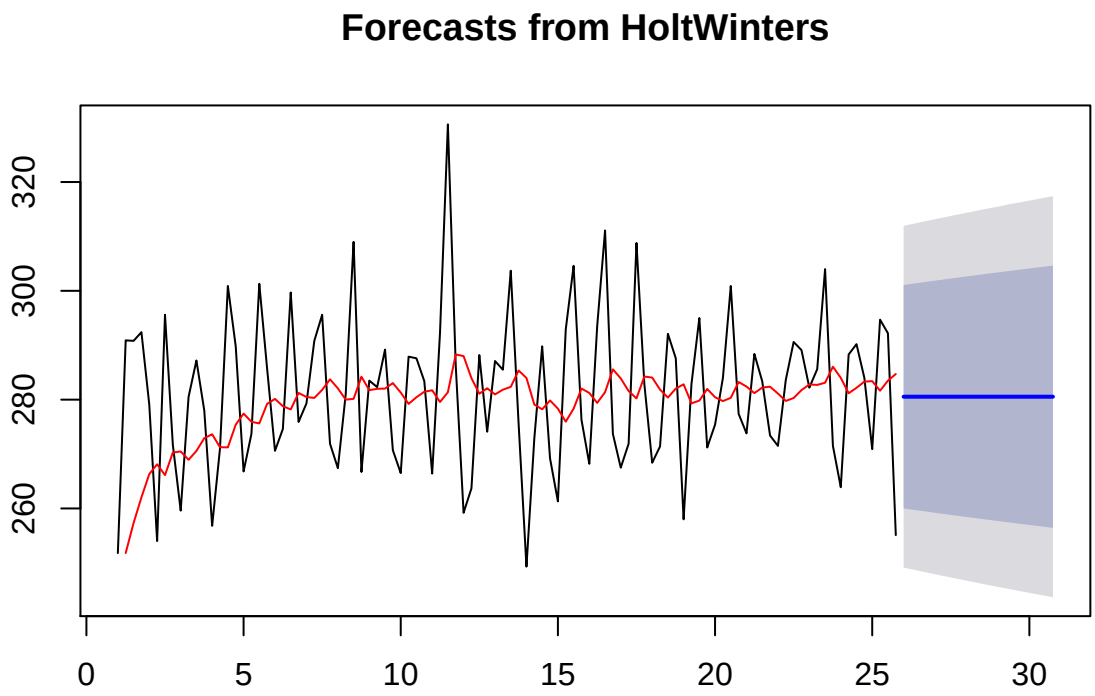
```
s1 <- omzetTs[1]
omzetHoltWinters <- HoltWinters(omzetTs, beta = FALSE, gamma = FALSE, s.start = s1)

omzetVoorspelling <- forecast(omzetHoltWinters, h = 20)

plot(omzetVoorspelling)
lines(omzetHoltWinters$fitted[,1], col= 'red')
```




```
s1Adds <- addsBudgetTs[1] # startwaarde
addsHolstWinters <- HoltWinters(addsBudgetTs, beta = FALSE, gamma =FALSE, s.start = s1Adds)

addsBudgetVoorspelling <- forecast(addsHolstWinters, h = 20)

plot(addsBudgetVoorspelling)
lines(addsHolstWinters$fitted[,1], col = 'red')
```

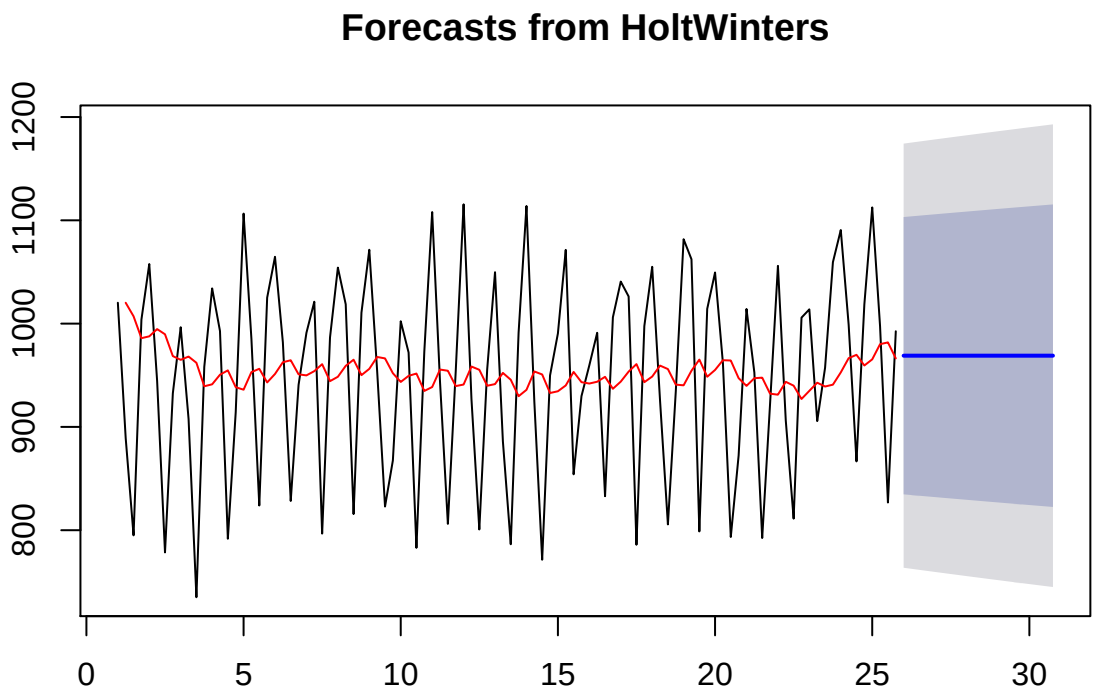



```
s1bnp <- bnpTs[1]

bnpHoltWinters <- HoltWinters(bnpTs, beta = FALSE, gamma = FALSE, s.start = s1bnp)

bnpVoorspelling <- forecast(bnpHoltWinters, h = 20)

plot(bnpVoorspelling)
lines(bnpHoltWinters$fitted[,1], col='red')
```

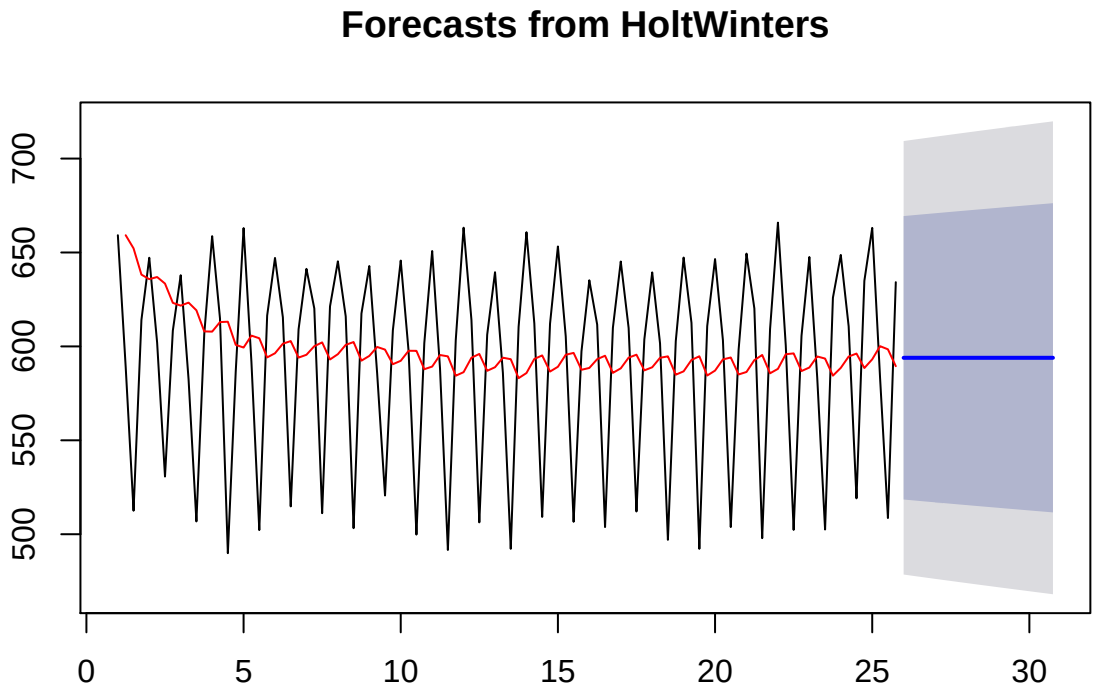


7.

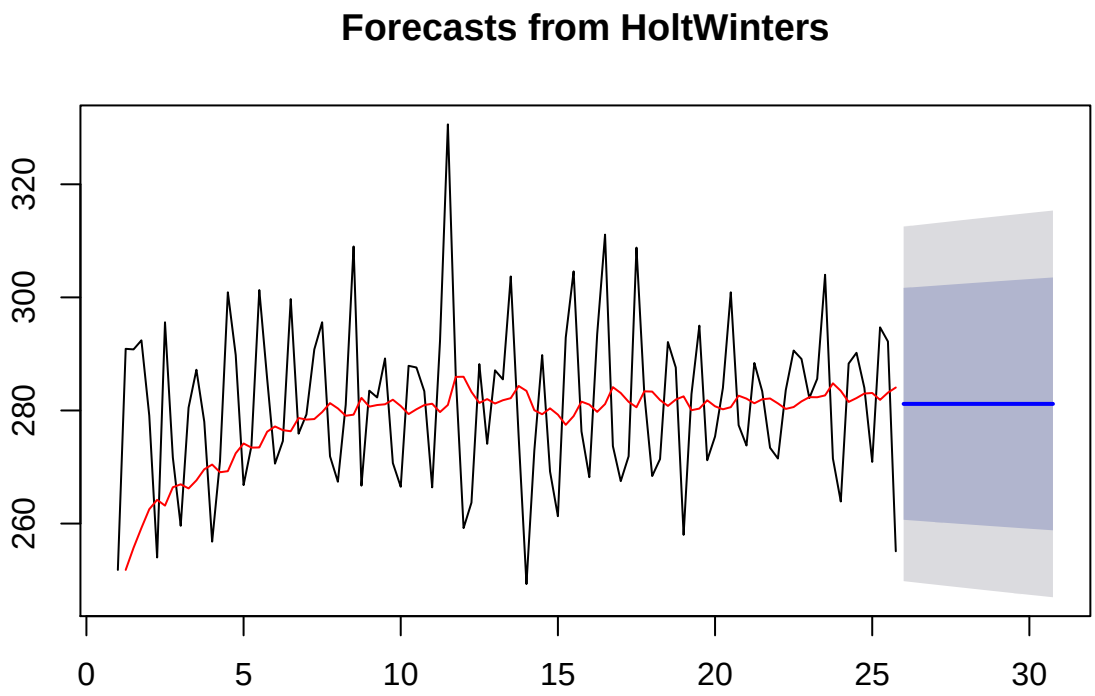
```
s1 <- omzetTs[1]
omzetHoltWinters <- HoltWinters(omzetTs, beta = FALSE, gamma = FALSE, s.start = s1, alpha = 0.1)

omzetVoorspelling <- forecast(omzetHoltWinters, h = 20)

plot(omzetVoorspelling)
lines(omzetHoltWinters$fitted[,1], col= 'red')
```




```
s1Adds <- addsBudgetTs[1] # startwaarde
addsHoltWinters <- HoltWinters(addsBudgetTs,alpha = 0.1 , beta = FALSE, gamma =FALSE, s.start = s1Adds)

addsBudgetVoorspelling <- forecast(addsHoltWinters, h = 20)

plot(addsBudgetVoorspelling)
lines(addsHoltWinters$fitted[,1], col = 'red')
```




```
s1bnp <- bnpTs[1]

bnpHoltWinters <- HoltWinters(bnpTs, beta = FALSE, gamma = FALSE, alpha = 0.1, s.start = s1bnp)

bnpVoorspelling <- forecast(bnpHoltWinters, h = 20)

plot(bnpVoorspelling)
lines(bnpHoltWinters$fitted[,1], col='red')
```

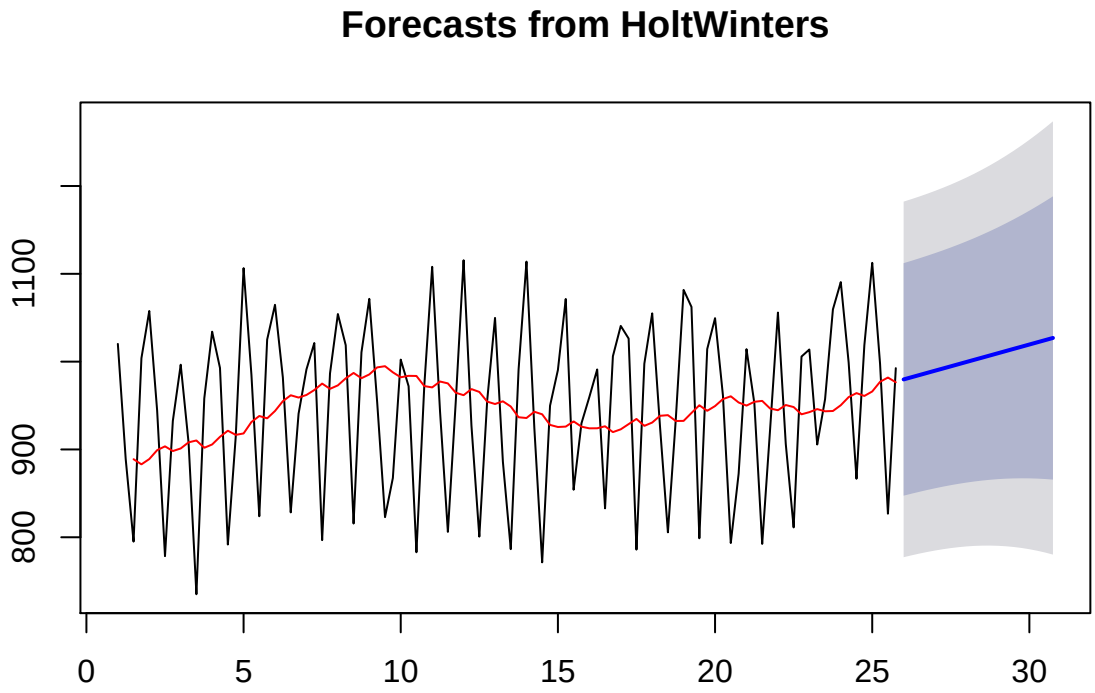


8.

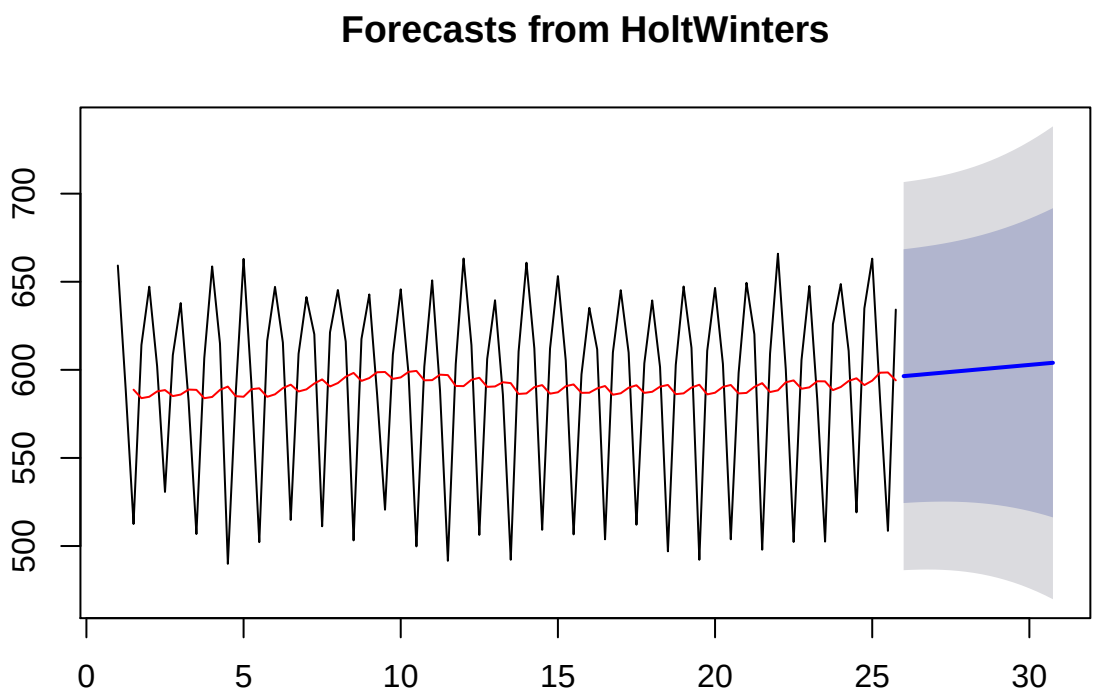
de breedte van de mogelijke voorspelling wordt sneller groter

9.

```
b1 <- (omzetTs[length(omzetTs)] - omzetTs[1]) / (length(omzetTs) - 1)

omzetHoltWinters <- HoltWinters(omzetTs, alpha = 0.05, beta = 0.2, gamma = FALSE, s.start = s1, b.start

omzetVoorspelling <- forecast(omzetHoltWinters, h=20)

plot(omzetVoorspelling)
lines(omzetHoltWinters$fitted[,1], col='red')
```



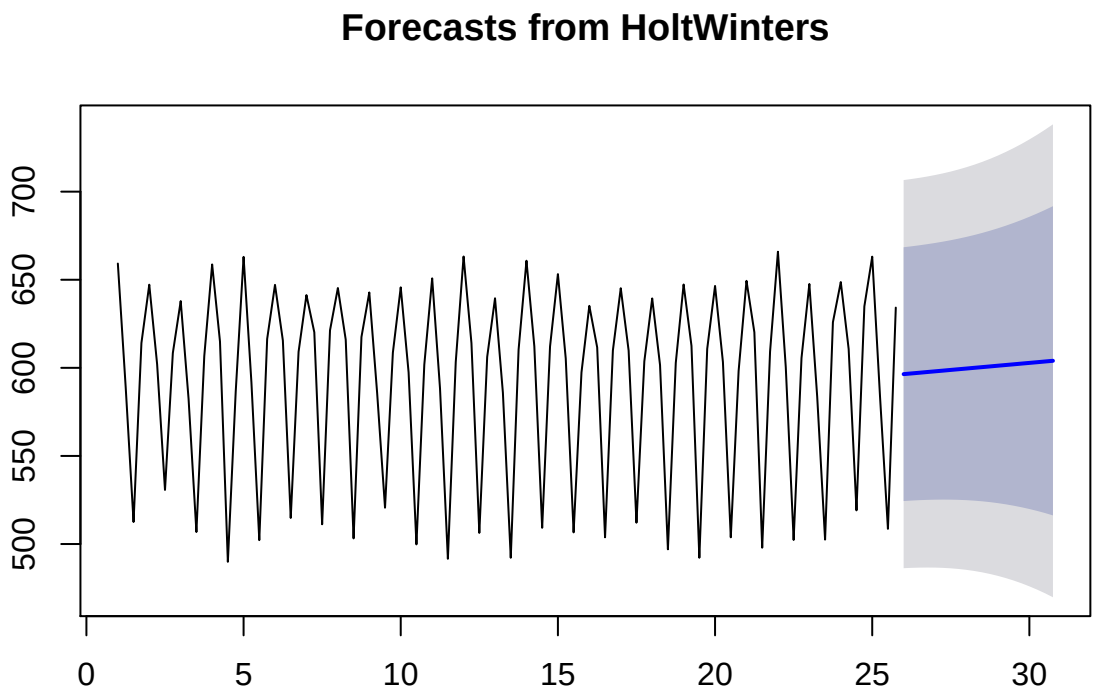

```
b1Adds <- (addsBudgetTs[length(addsBudgetTs)] - addsBudgetTs[1]) / (length(addsBudgetTs) - 1)


addsHoltWinters <- HoltWinters(addsBudgetTs, alpha = 0.05, beta = 0.2, gamma = FALSE, s.start = s1Adds,

addsBudgetVoorspelling <- forecast(addsHoltWinters, h=20)

plot(addsBudgetVoorspelling)
lines(addsHoltWinters$fitted[,1], col='red')
```




```
b1bnp <- - (bnpTs[length(bnpTs)] - bnpTs[1]) / (length(bnpTs) - 1)
b1bnp
```

```
## [1] -0.03333333
```

```
bnpHoltWinters <- HoltWinters(bnpTs, alpha = 0.05, beta = 0.2, gamma = FALSE, s.start = s1bnp, b.start =

bnpVoorspelling <- forecast(addsHoltWinters, h=20)

plot(bnpVoorspelling)
lines(bnpHoltWinters$fitted[,1], col='red')
```



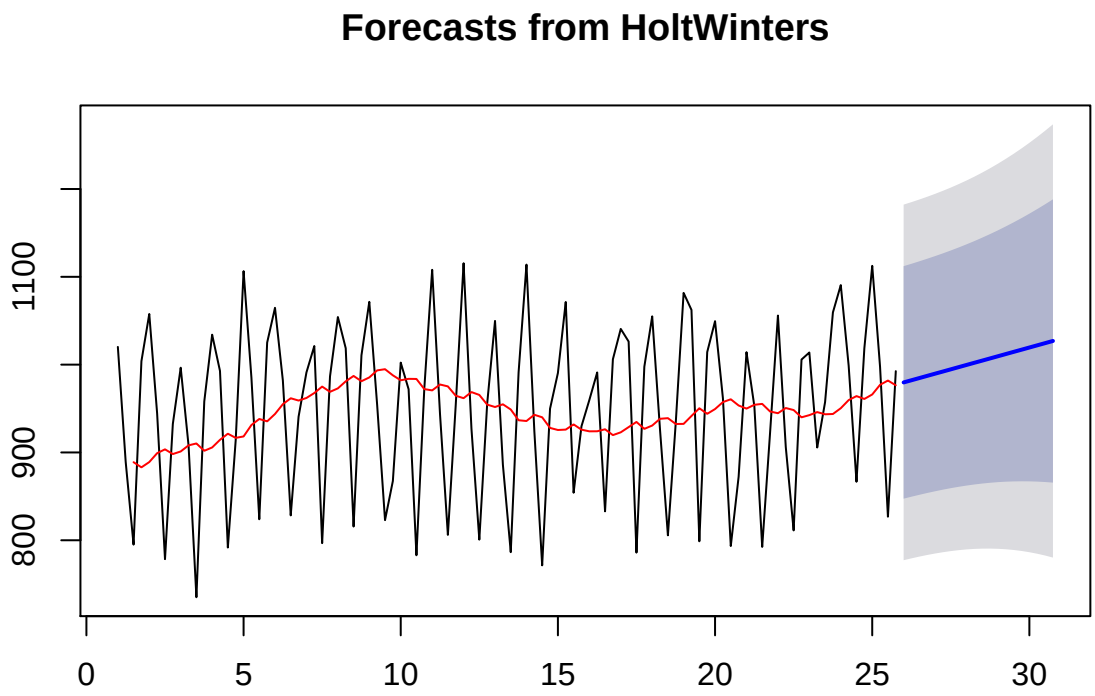
10

reeds gedaan in puntje 9

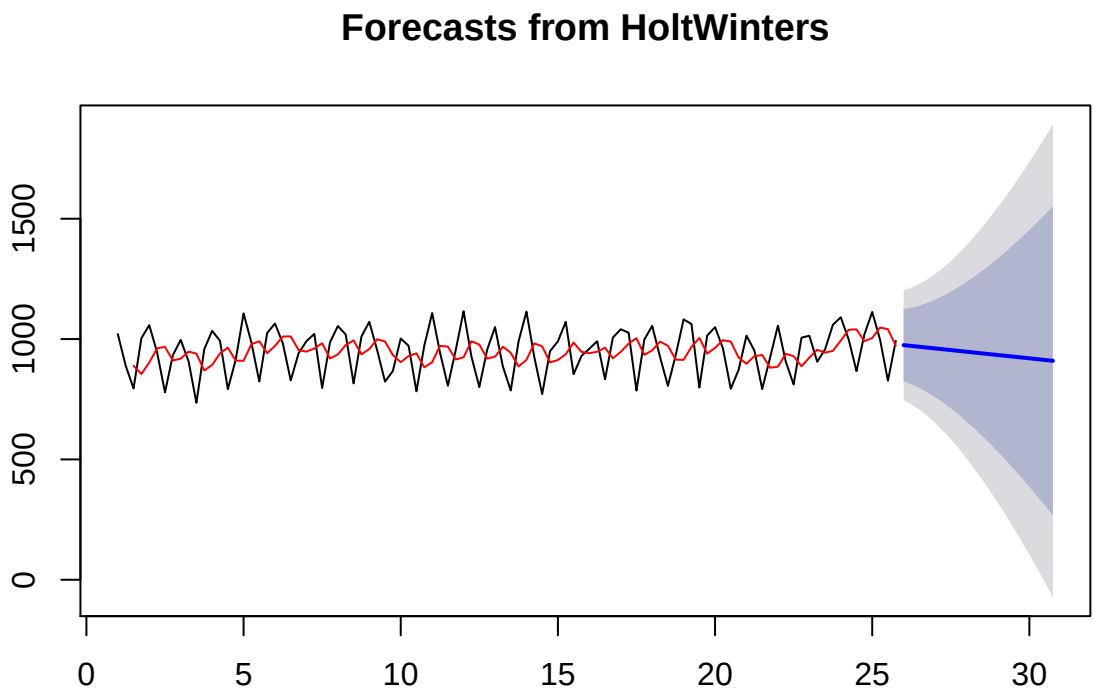
11

enkel voor omzet:

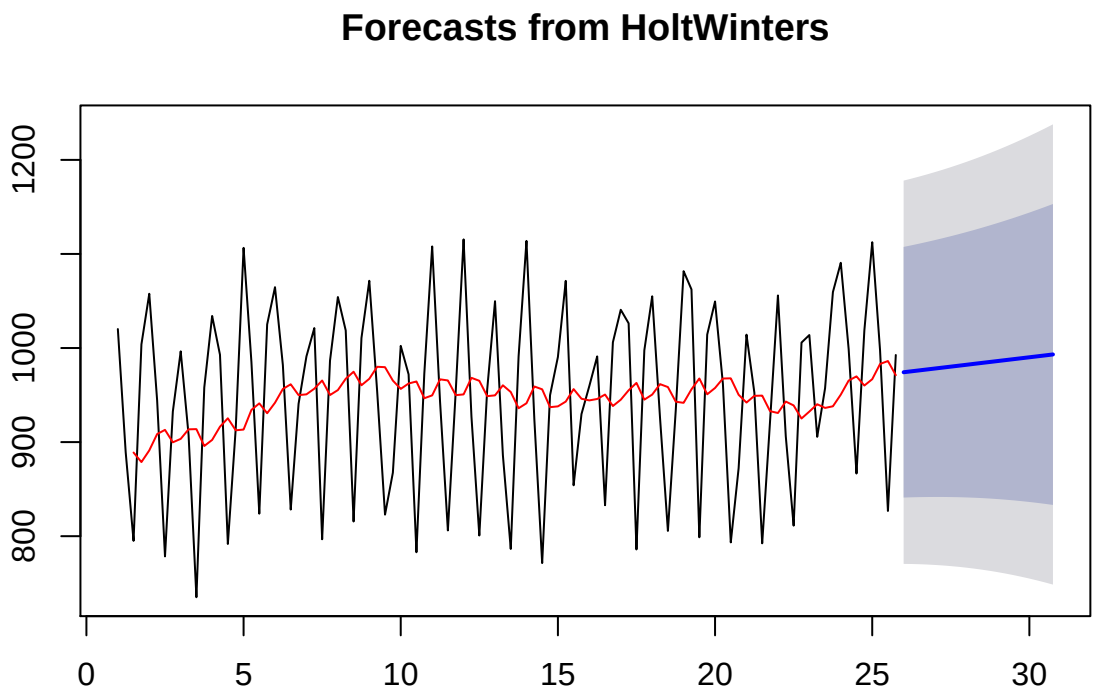
```
b1 <- (omzetTs[length(omzetTs)] - omzetTs[1]) / (length(omzetTs) - 1)

omzetHoltWinters <- HoltWinters(omzetTs, alpha = 0.05, beta = 0.2, gamma = FALSE, s.start = s1, b.start

omzetVoorspelling <- forecast(omzetHoltWinters, h=20)

plot(omzetVoorspelling)
lines(omzetHoltWinters$fitted[,1], col='red')
```



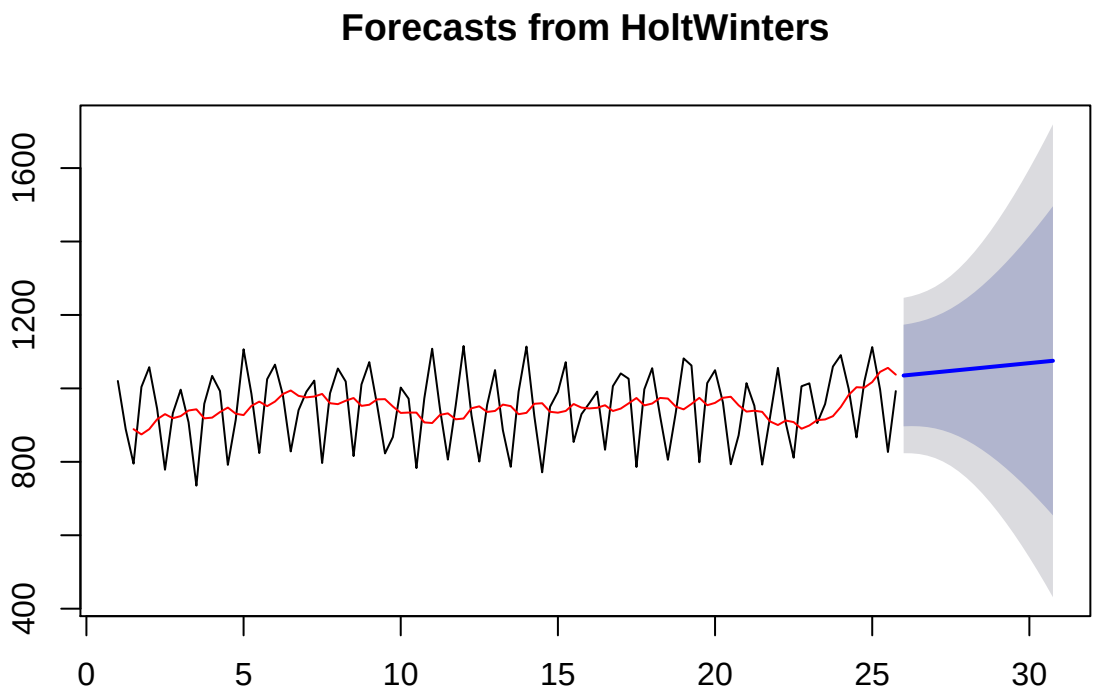

```
## --------------------


b1 <- (omzetTs[length(omzetTs)] - omzetTs[1]) / (length(omzetTs) - 1)

omzetHoltWinters <- HoltWinters(omzetTs, alpha = 0.3, beta = 0.2, gamma = FALSE, s.start = s1, b.start =

omzetVoorspelling <- forecast(omzetHoltWinters, h=20)

plot(omzetVoorspelling)
lines(omzetHoltWinters$fitted[,1], col='red')
```



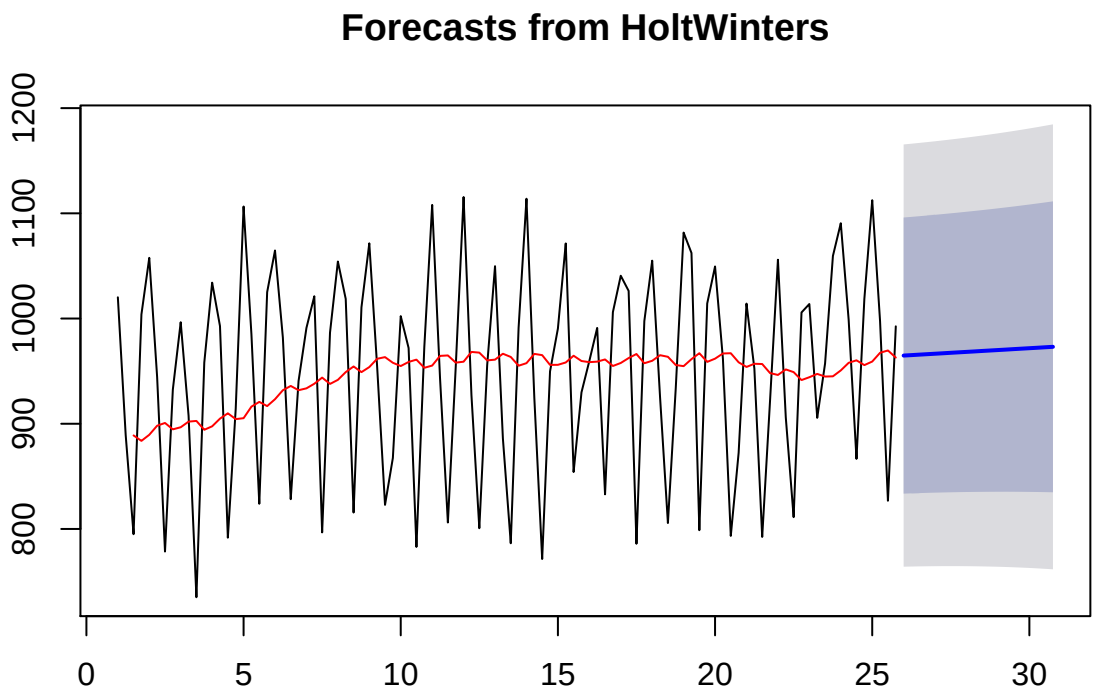

```
## --------------------

b1 <- (omzetTs[length(omzetTs)] - omzetTs[1]) / (length(omzetTs) - 1)

omzetHoltWinters <- HoltWinters(omzetTs, alpha = 0.10, beta = 0.05, gamma = FALSE, s.start = s1, b.star

omzetVoorspelling <- forecast(omzetHoltWinters, h=20)

plot(omzetVoorspelling)
lines(omzetHoltWinters$fitted[,1], col='red')
```



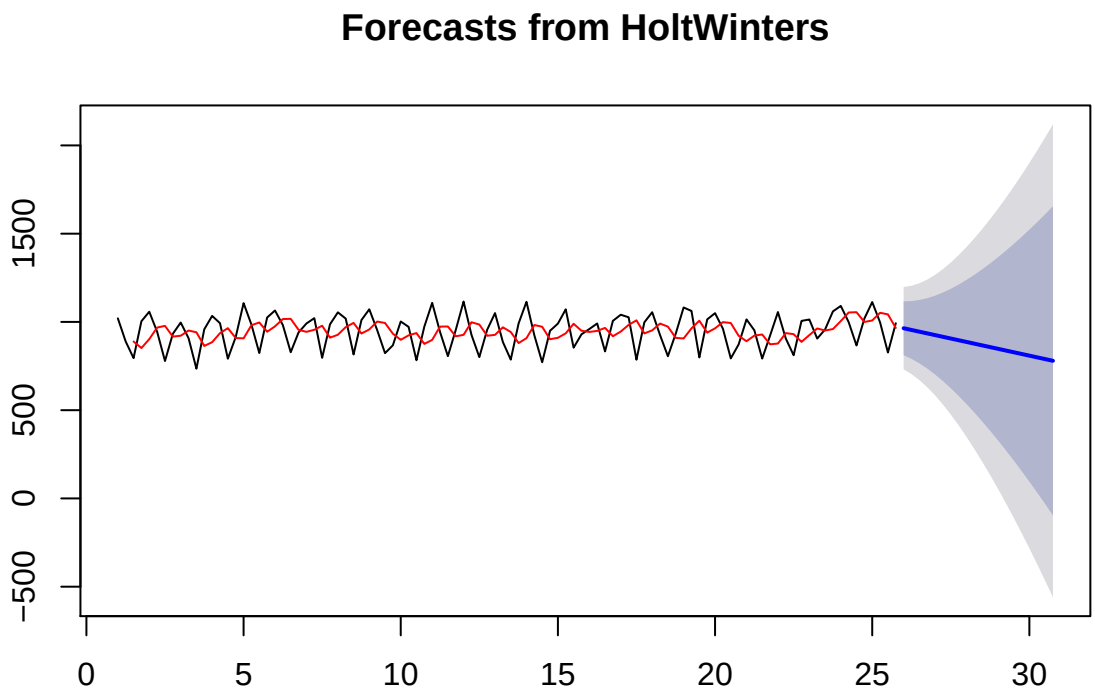

```
## --------------------

b1 <- (omzetTs[length(omzetTs)] - omzetTs[1]) / (length(omzetTs) - 1)

omzetHoltWinters <- HoltWinters(omzetTs, alpha = 0.10, beta = 0.50, gamma = FALSE, s.start = s1, b.star

omzetVoorspelling <- forecast(omzetHoltWinters, h=20)

plot(omzetVoorspelling)
lines(omzetHoltWinters$fitted[,1], col='red')
```




```
## --------------------

b1 <- (omzetTs[length(omzetTs)] - omzetTs[1]) / (length(omzetTs) - 1)

omzetHoltWinters <- HoltWinters(omzetTs, alpha = 0.05, beta = 0.05, gamma = FALSE, s.start = s1, b.star

omzetVoorspelling <- forecast(omzetHoltWinters, h=20)

plot(omzetVoorspelling)
lines(omzetHoltWinters$fitted[,1], col='red')
```

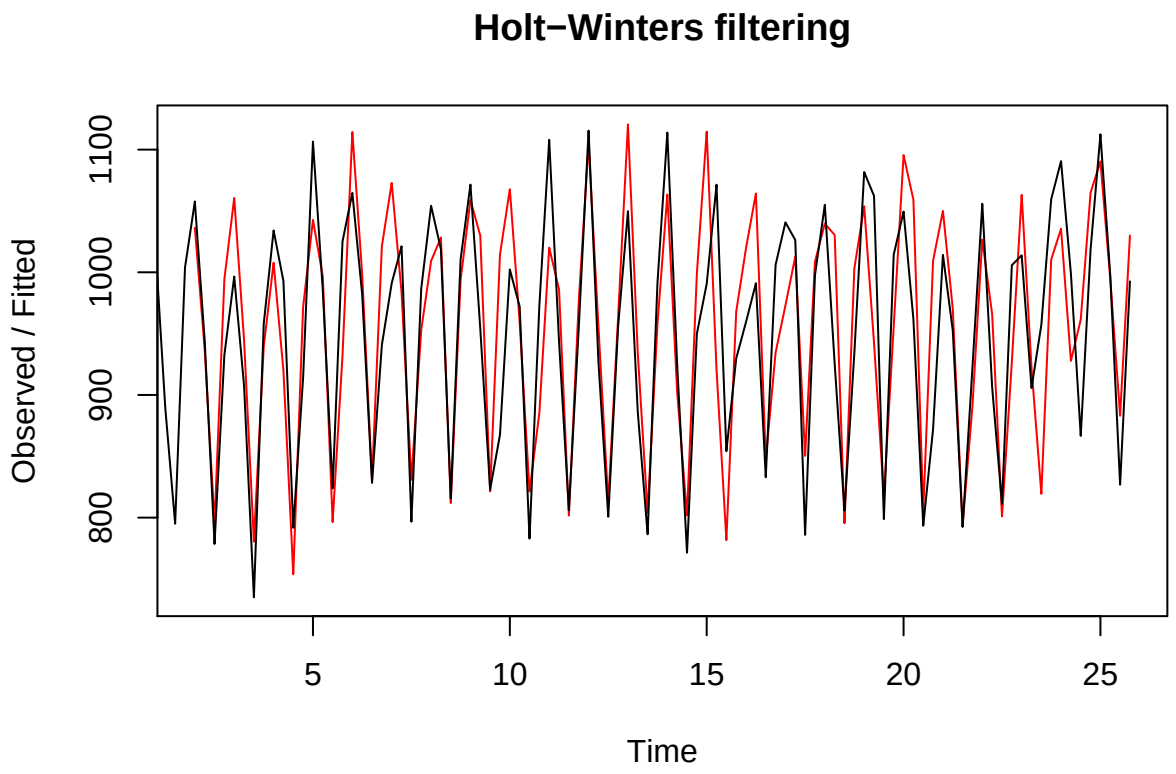



```
## --------------------

b1 <- (omzetTs[length(omzetTs)] - omzetTs[1]) / (length(omzetTs) - 1)

omzetHoltWinters <- HoltWinters(omzetTs, alpha = 0.30, beta = 0.30, gamma = FALSE, s.start = s1, b.star

omzetVoorspelling <- forecast(omzetHoltWinters, h=20)

plot(omzetVoorspelling)
lines(omzetHoltWinters$fitted[,1], col='red')
```

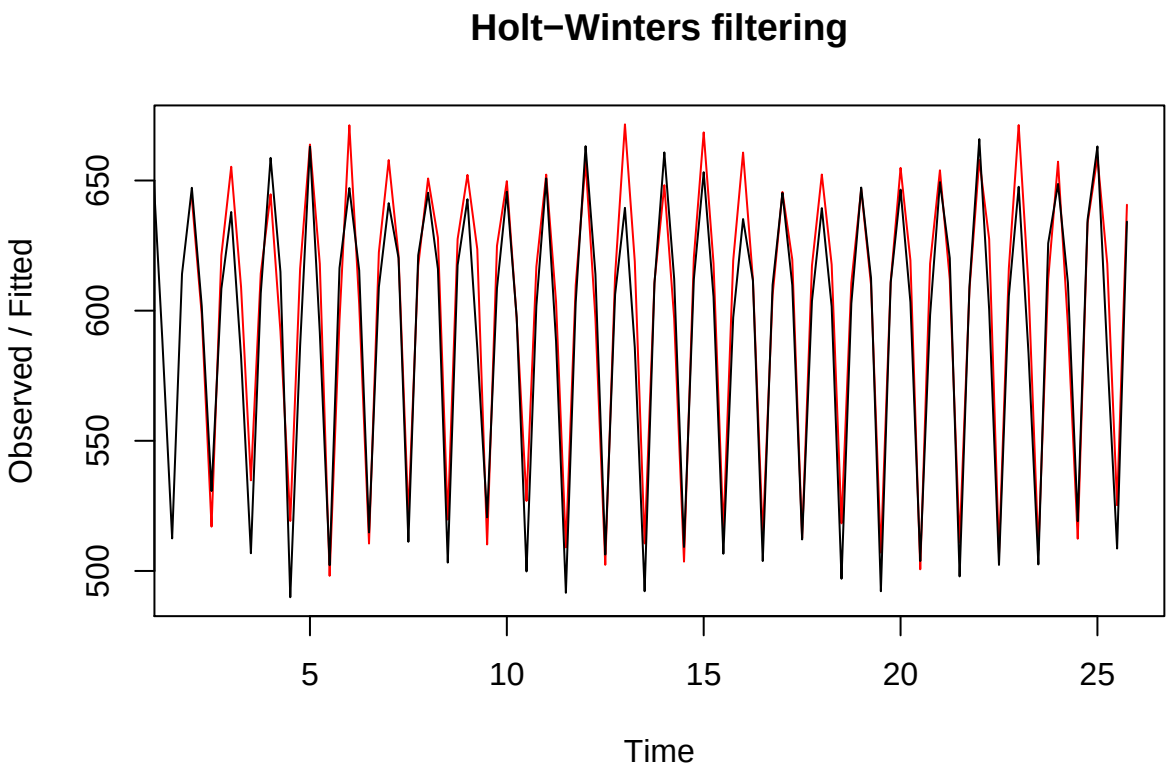


12

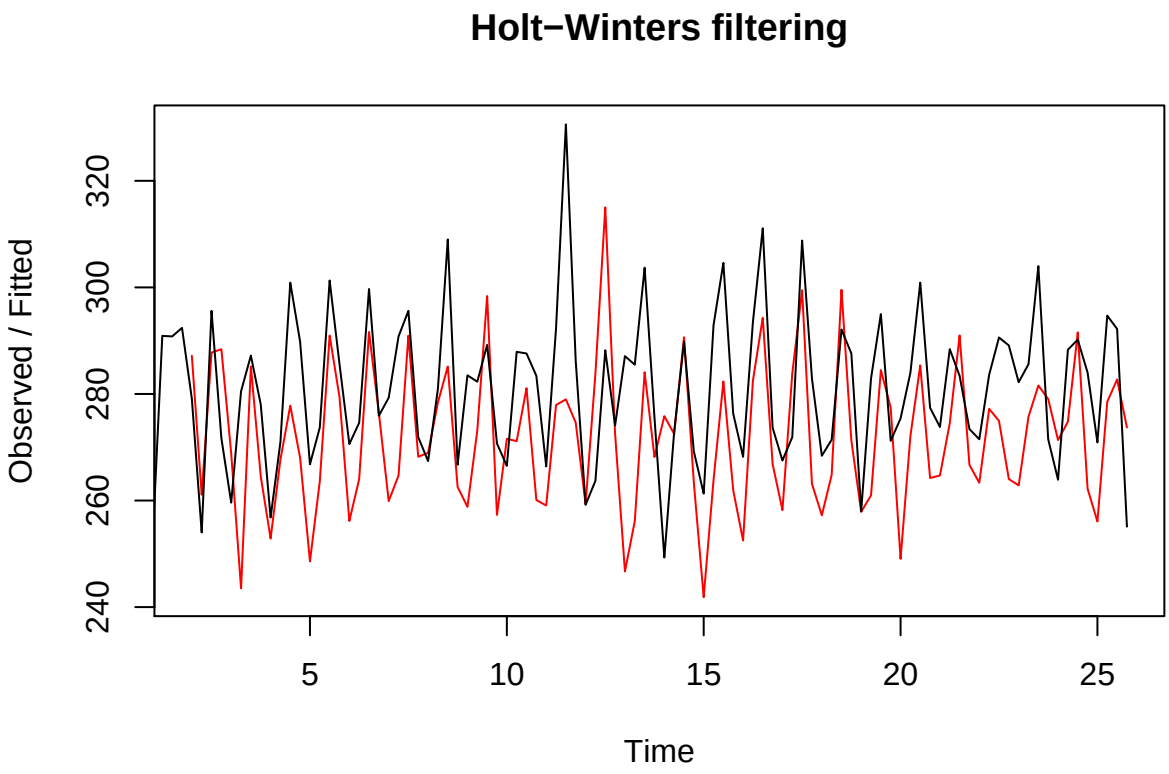
```
omzetHoltWinters <- HoltWinters(omzetTs, alpha = 0.05, beta=0, gamma = 0.9)
plot(omzetHoltWinters)
```




```
addsHoltWinters <- HoltWinters(addsBudgetTs, alpha = 0.05, beta = 0, gamma = 0.9)

plot(addsHoltWinters)
```

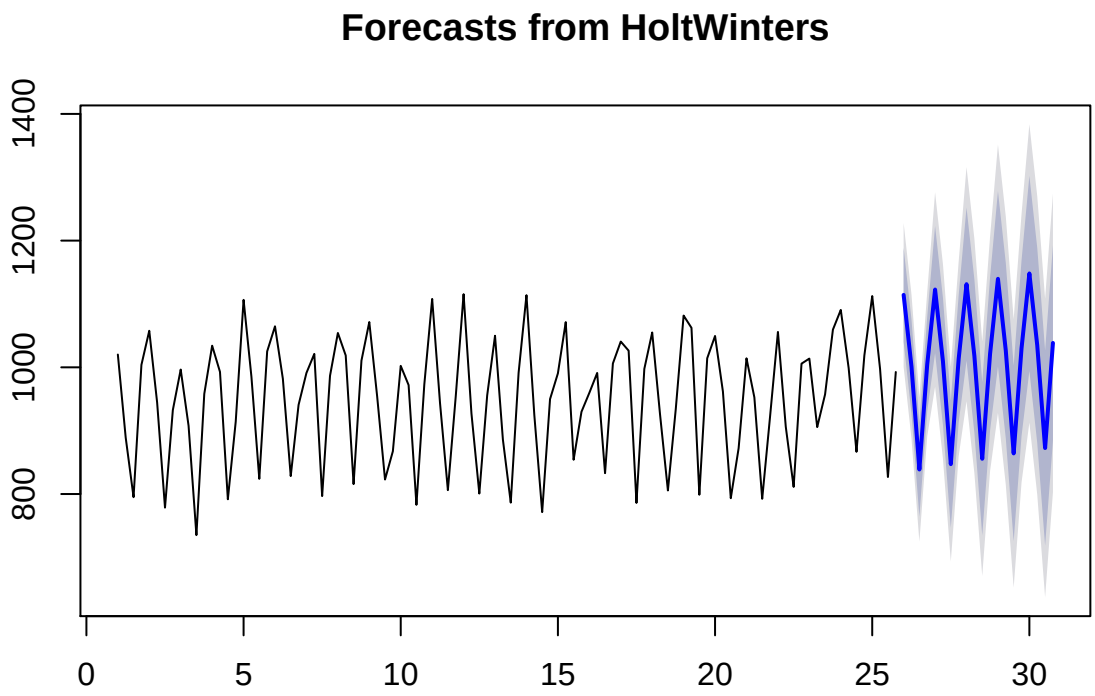



```
bnpHoltWinters <- HoltWinters(bnpTs, alpha = 0.05, beta = 0, gamma = 0.9)

plot(bnpHoltWinters)
```

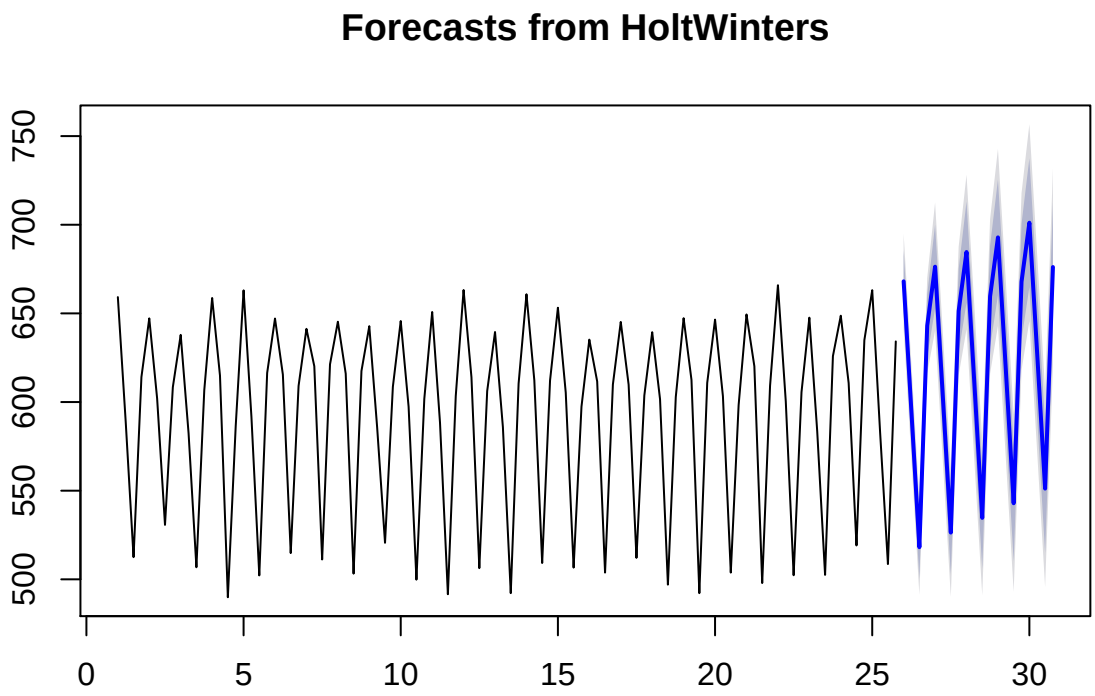


13

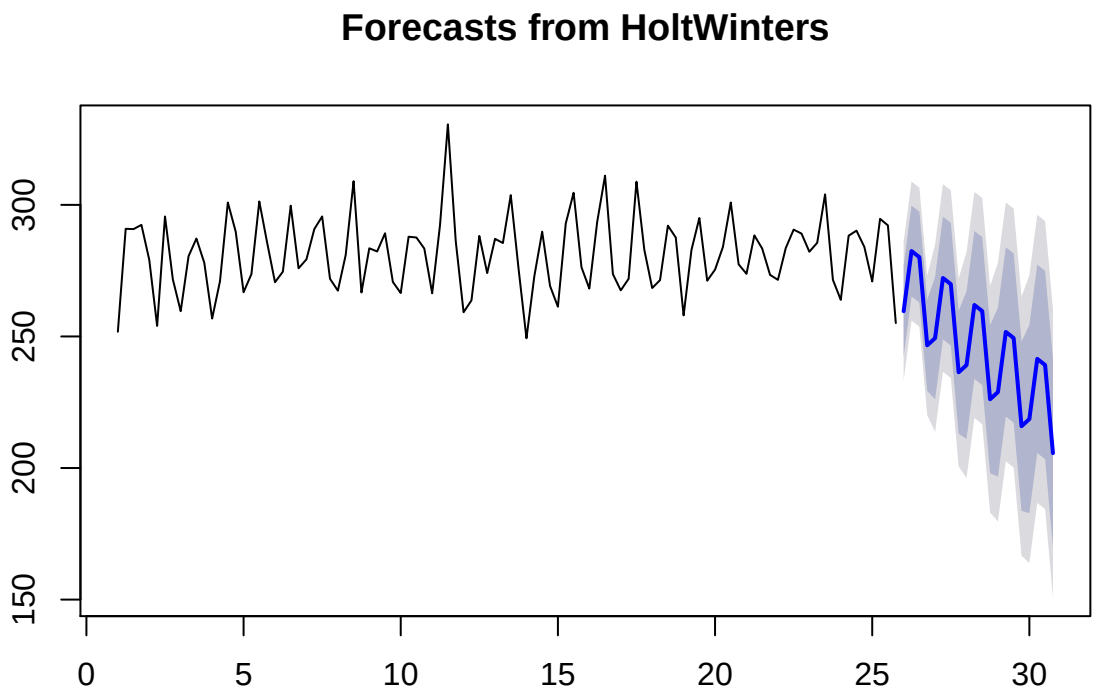
```
omzetVoorspelling <- forecast(omzetHoltWinters, h=20)
plot(omzetVoorspelling)
```




```
addsBudgetVoorspelling <- forecast(addsHoltWinters, h=20)
plot(addsBudgetVoorspelling)
```




```
bnpVoorspelling <- forecast(bnpHoltWinters, h=20)
plot(bnpVoorspelling)
```

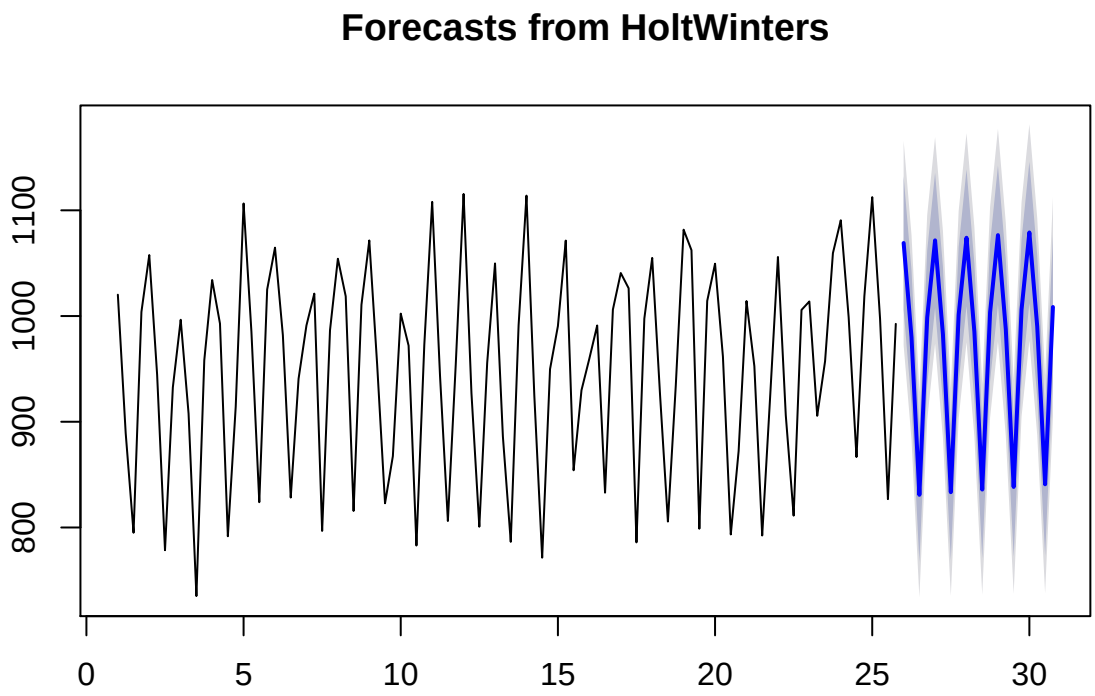


dit is een betere techniek. we hebben te maken met seizoensgebonden elementen. Deze methode houd daar rekening mee

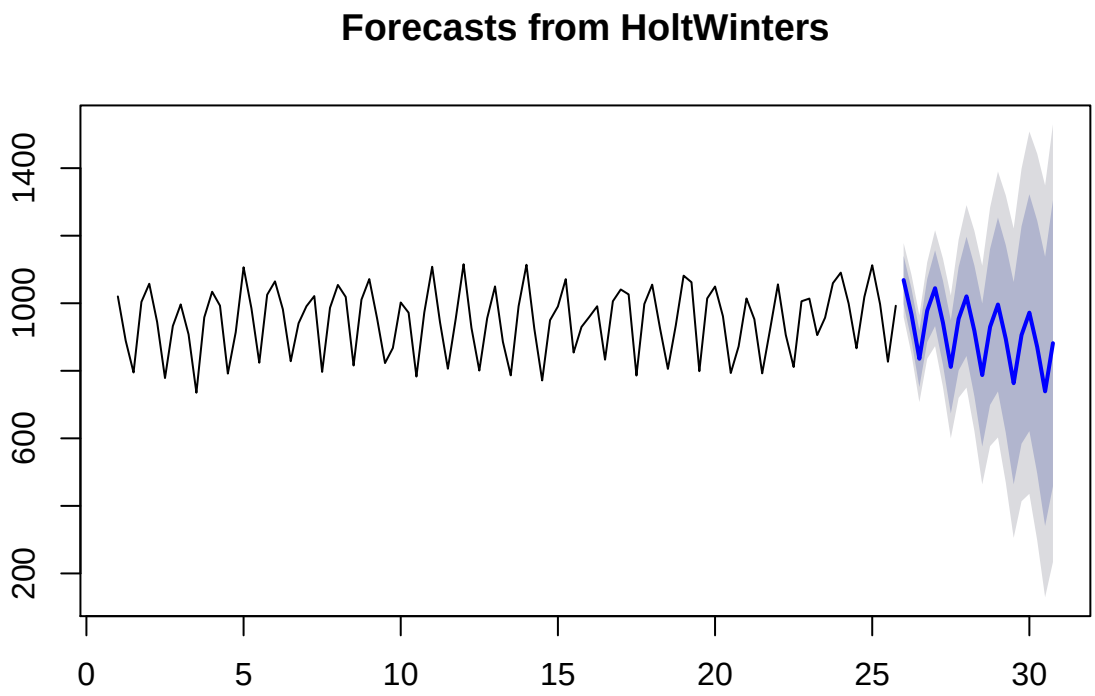
14

enkel gedaan voor omzet

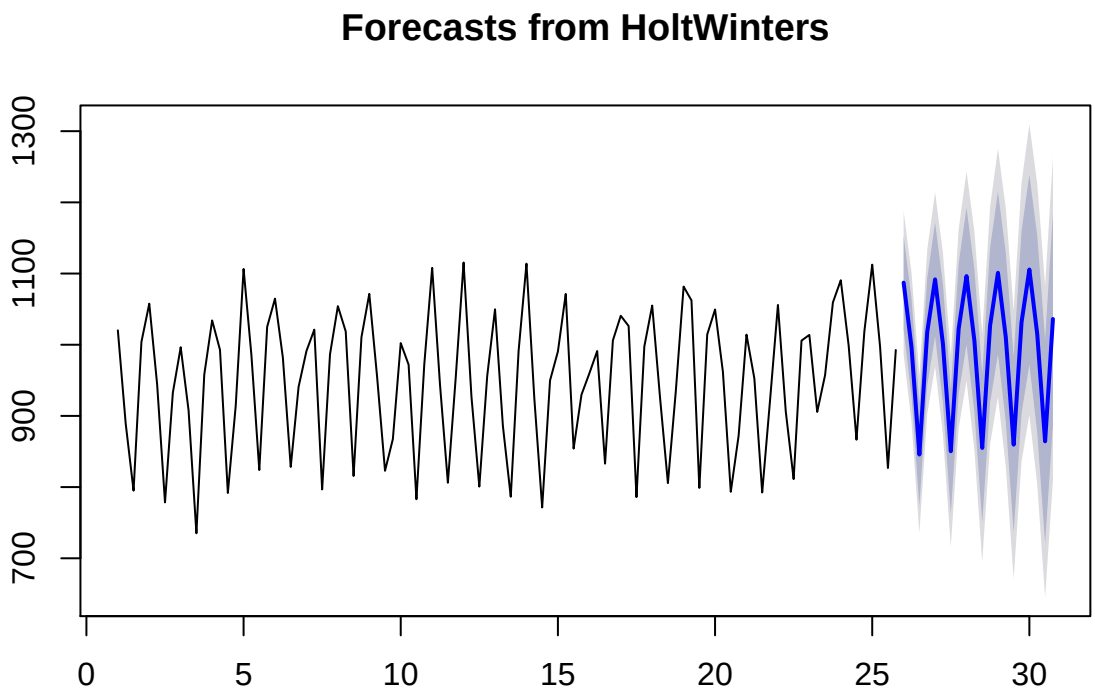
```
omzetHoltWinters <- HoltWinters(omzetTs, alpha = 0.05, beta=0.05, gamma = 0.05)
omzetVoorspelling <- forecast(omzetHoltWinters, h=20)
plot(omzetVoorspelling)
```



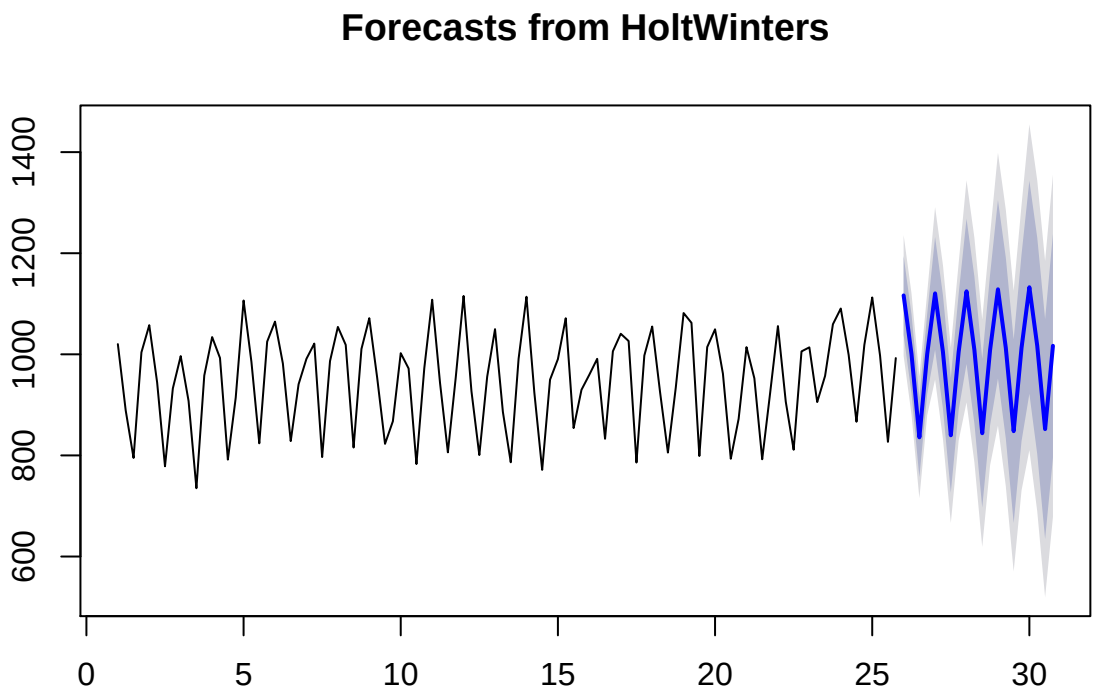

```
## --------------------

omzetHoltWinters <- HoltWinters(omzetTs, alpha = 0.3, beta=0.3, gamma = 0.3)
omzetVoorspelling <- forecast(omzetHoltWinters, h=20)
plot(omzetVoorspelling)
```




```
## --------------------

omzetHoltWinters <- HoltWinters(omzetTs, alpha = 0.3, beta=0.05, gamma = 0.05)
omzetVoorspelling <- forecast(omzetHoltWinters, h=20)
plot(omzetVoorspelling)
```

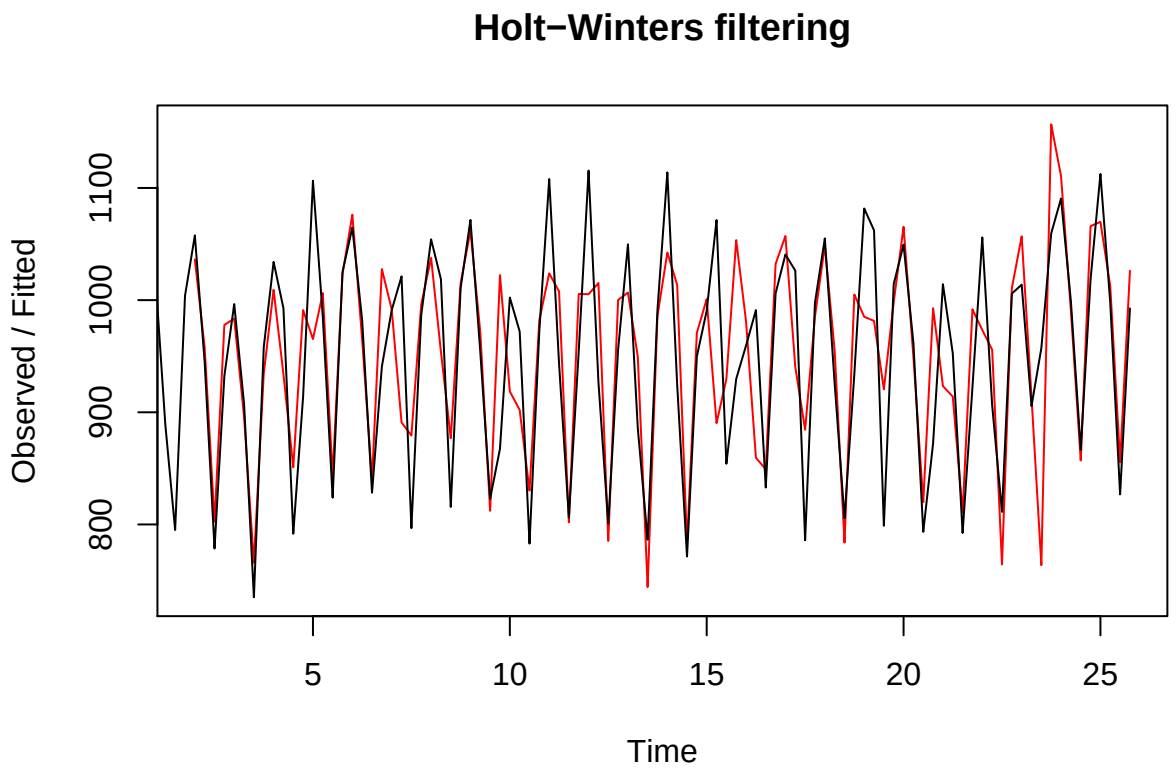



```
## --------------------

omzetHoltWinters <- HoltWinters(omzetTs, alpha = 0.05, beta=0.5, gamma = 0.9)
omzetVoorspelling <- forecast(omzetHoltWinters, h=20)
plot(omzetVoorspelling)
```

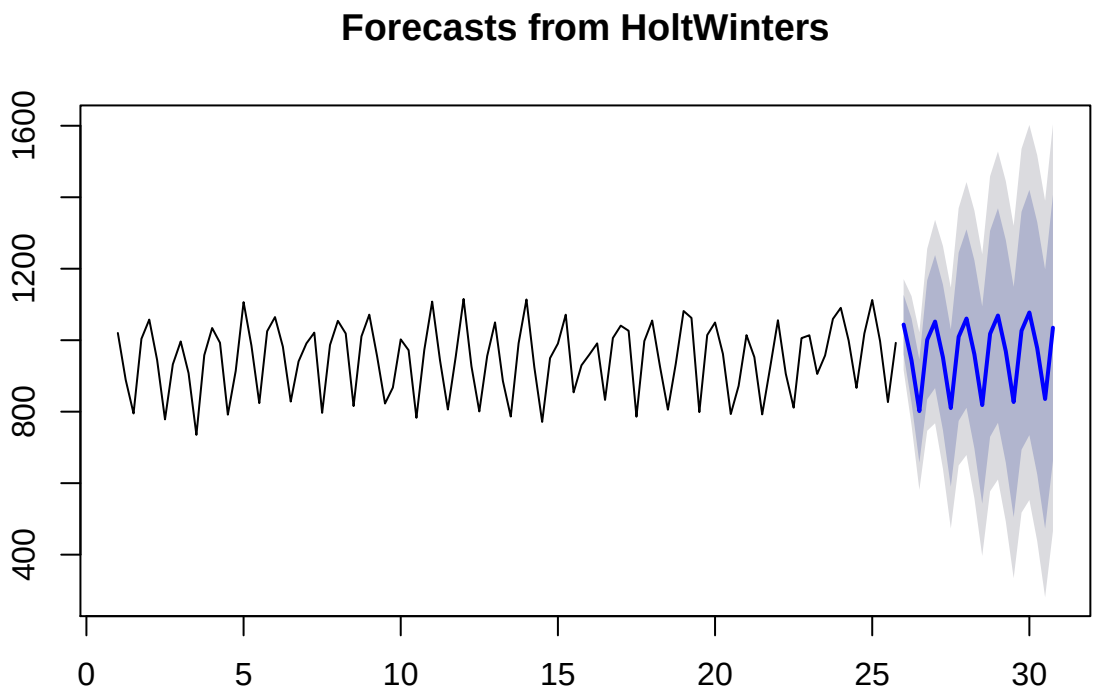


15

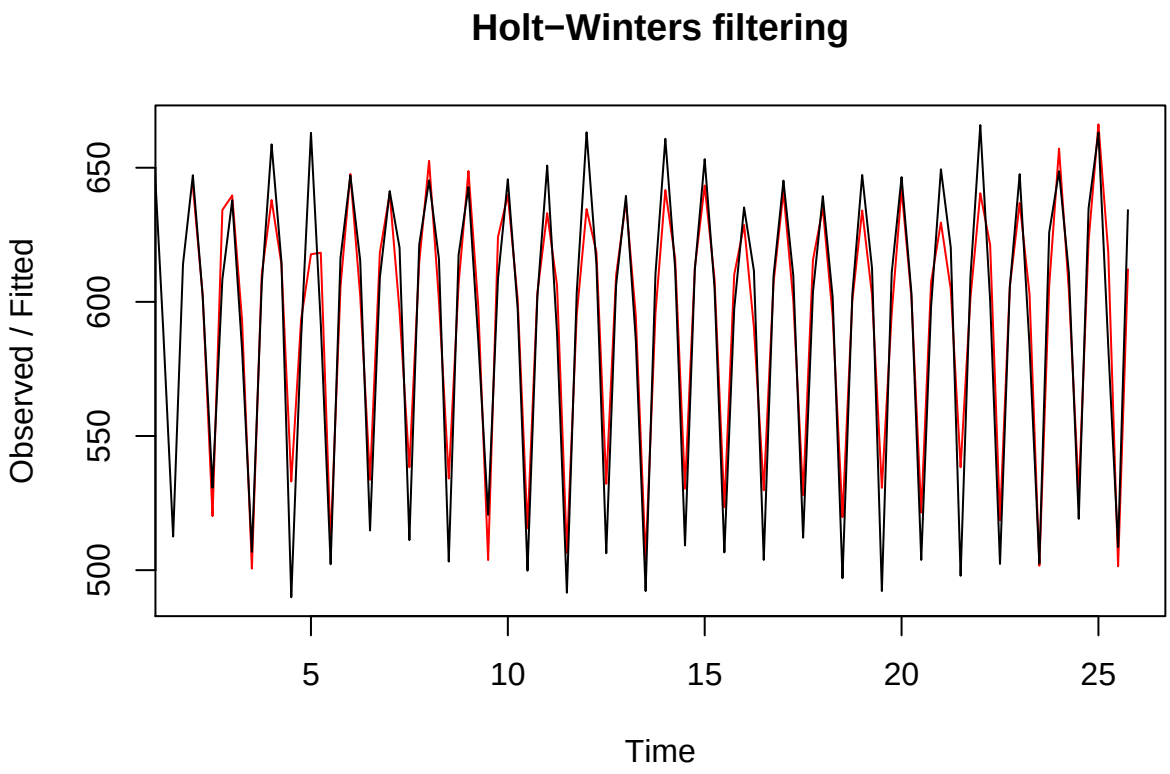
```
omzetHoltWinters <- HoltWinters(omzetTs, alpha = TRUE, beta = 0, gamma = TRUE)
plot(omzetHoltWinters)
```



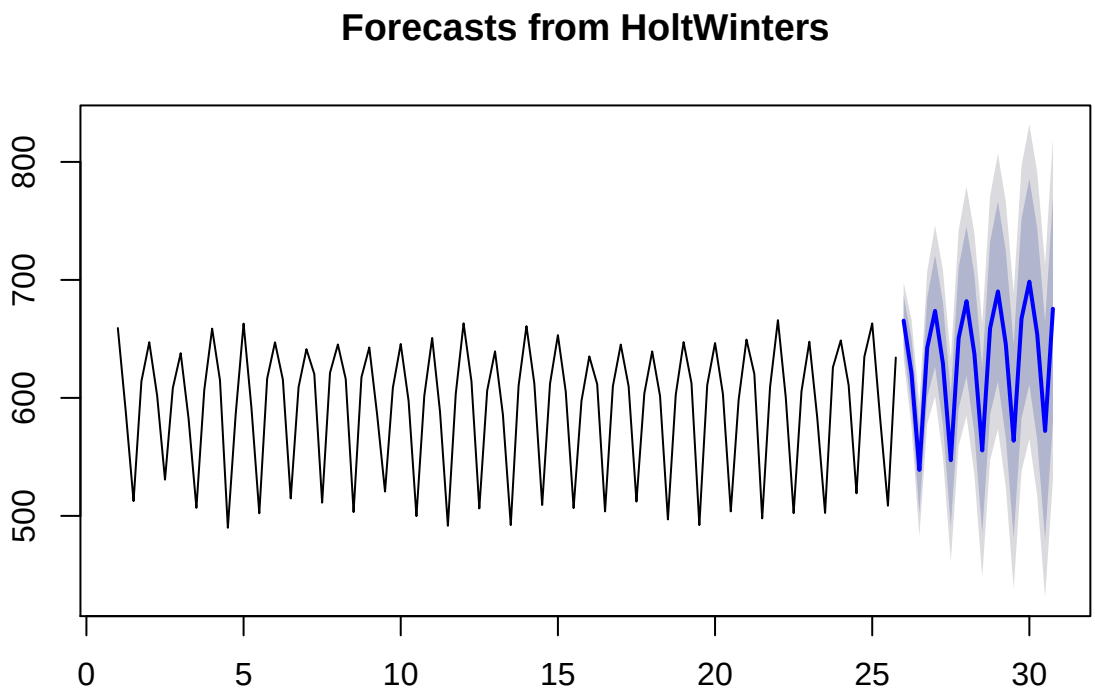

```
omzetVoorspelling <- forecast(omzetHoltWinters, h=20)
plot(omzetVoorspelling)
```



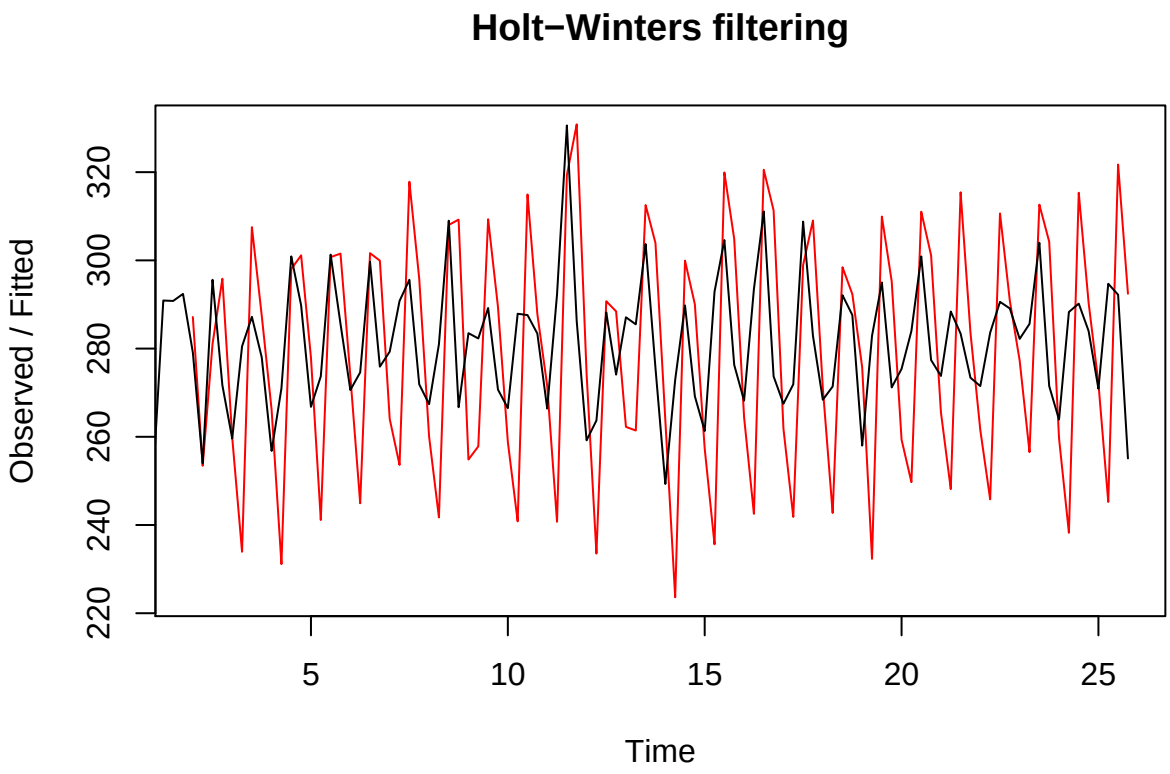

```
addsHoltWinters <- HoltWinters(addsBudgetTs, alpha = TRUE, beta = 0, gamma = TRUE)
plot(addsHoltWinters)
```



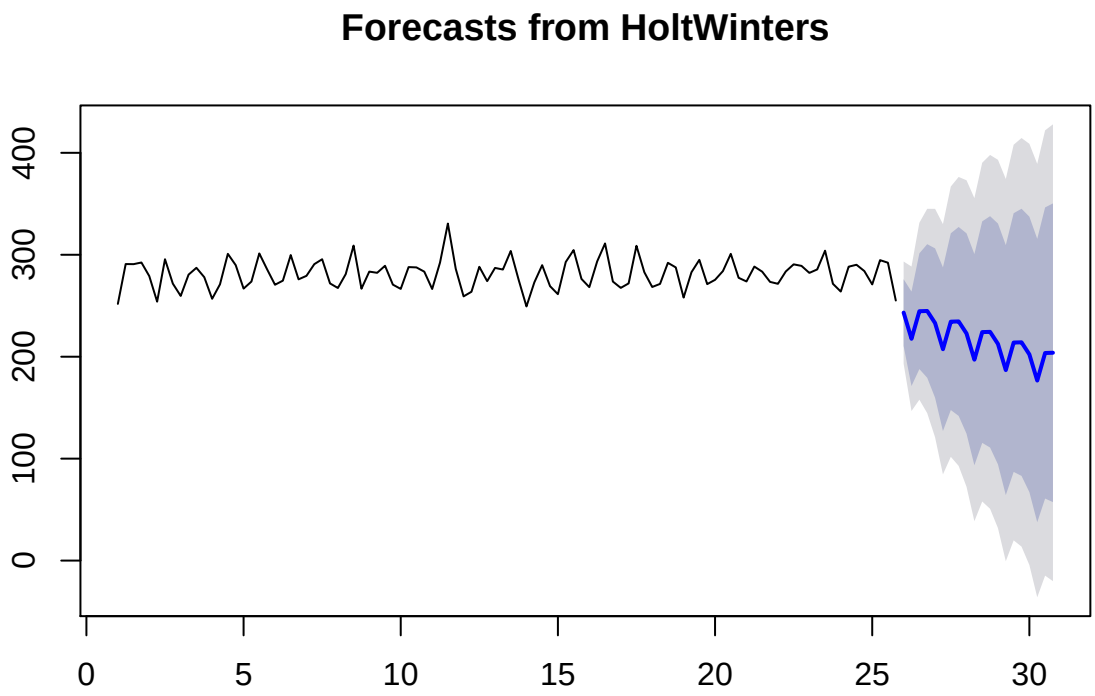

```
addsBudgetVoorspelling <- forecast(addsHoltWinters, h=20)
plot(addsBudgetVoorspelling)
```




```
bnpHoltWinters <- HoltWinters(bnpTs, alpha = TRUE, beta = 0, gamma = TRUE)
plot(bnpHoltWinters)
```

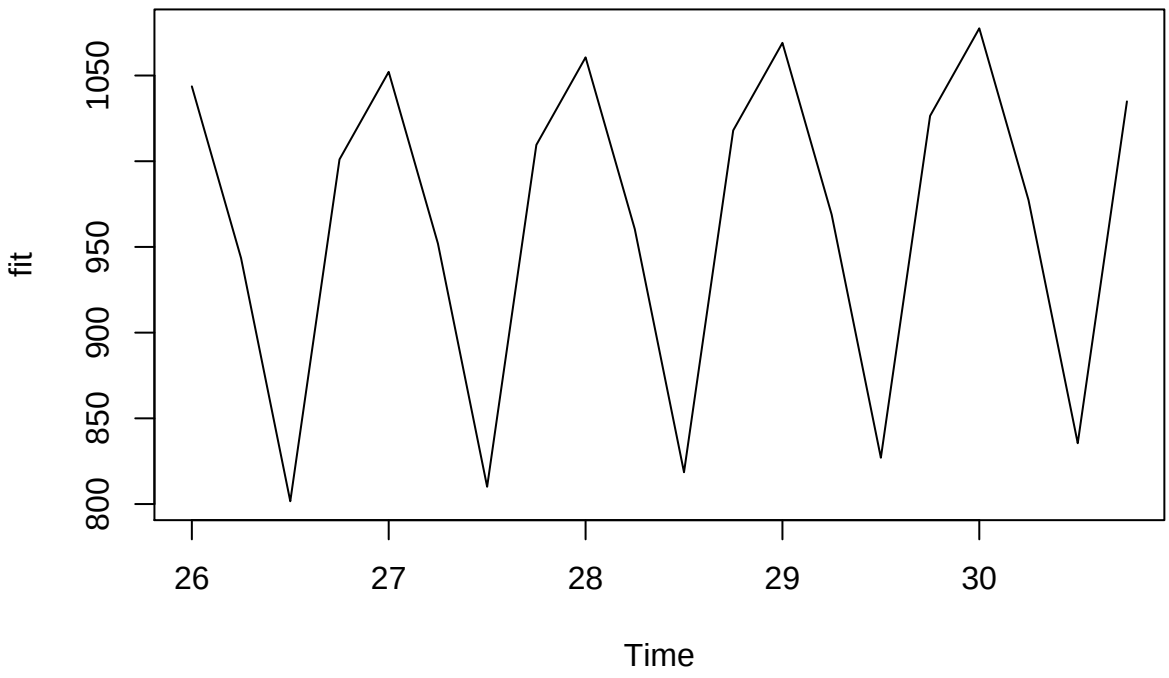



```
bnpVoorspelling <- forecast(bnpHoltWinters, h=20)
plot(bnpVoorspelling)
```

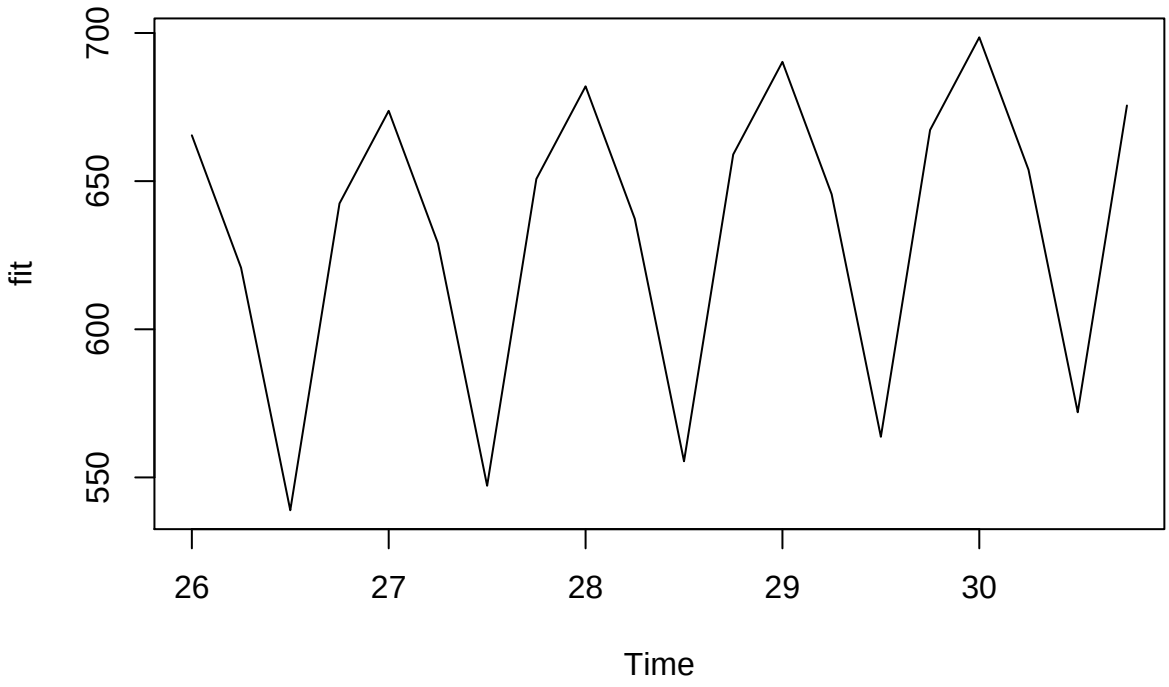


16

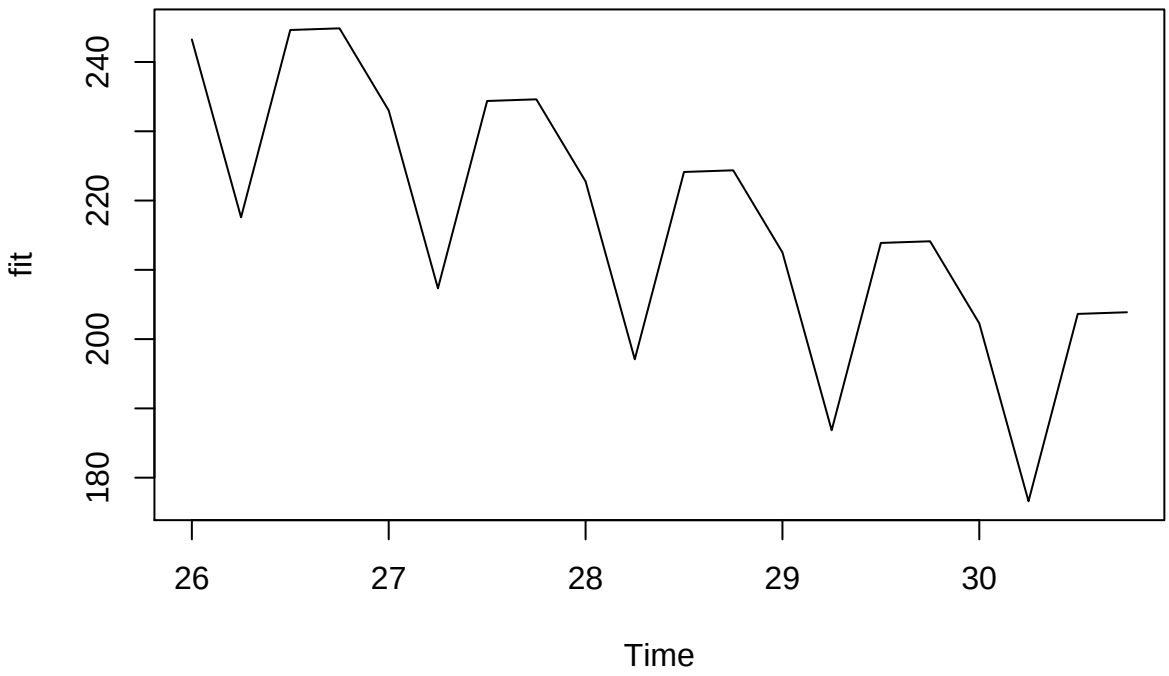
```
omzetPredict <- predict(omzetHoltWinters, n.ahead = 20)

plot(omzetPredict)
```




```
addsBudgetPredict <- predict(addsHoltWinters, n.ahead = 20)

plot(addsBudgetPredict)
```




```
bnpPredict <- predict(bnpHoltWinters, n.ahead = 20)

plot(bnpPredict)
```


fit
180
200
220
240
26
27
28
29
30
Time