INTRODUCCIÓN A SERIES DE TIEMPO

LUIS ORTIZ-CEVALLOS

2021-09-23

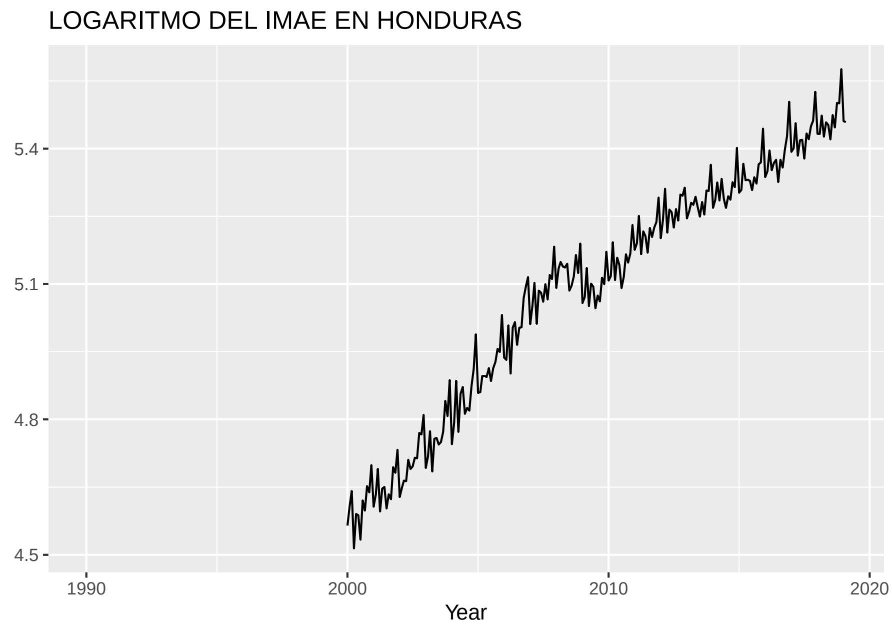
Contents

Chapter 1

¿Qué es una de serie de tiempo?

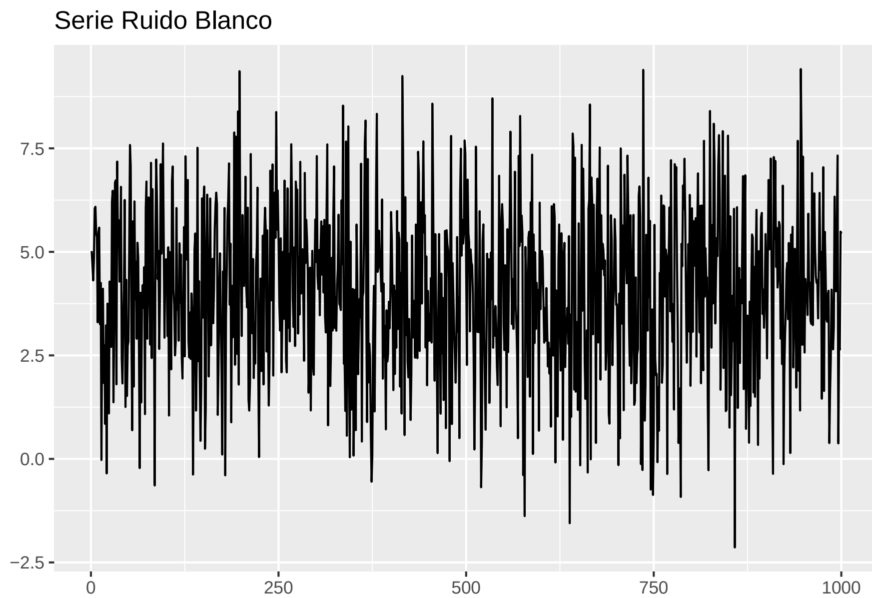
```
library("readr")
library("xts")
library("zoo")
library("astsa")
library("forecast")
library("ggplot2")
library("forecast")
library("ggfortify")
library("stargazer")
library("urca")
library("dynlm")
library("scales")
library("quantmod")
TRIM<-as.xts(read.zoo("FINAL_HN.csv", index.column = 1, sep = ";", header=TRUE, format = "%d/%m/%
MES<-as.xts(read.zoo("MES_HN.csv", index.column = 1, sep = ";", header=TRUE, format = "%d/%m/%Y")
IMAE<-MES$IMAE
P<-ggplot2::autoplot(log(IMAE))+xlab("Year")+
ggtitle("LOGARITMO DEL IMAE EN HONDURAS")
```

```
P
```



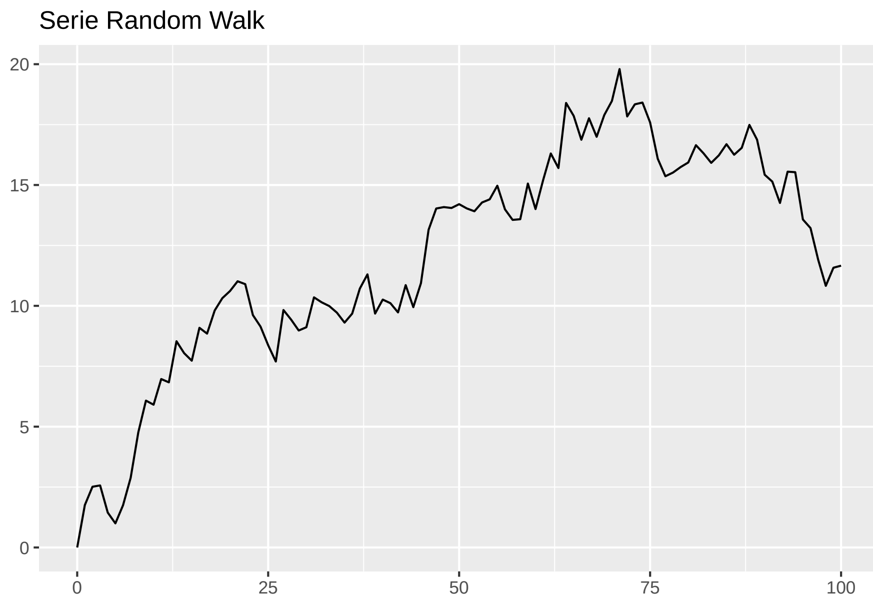

```
X_WN<-arima.sim(list(order=c(0,0,0)), n=1000, mean=4, sd=2)
autoplot(X_WN)+
ggtitle("Serie Ruido Blanco")
```

1.1 Serie Ruido Blanco (WN)

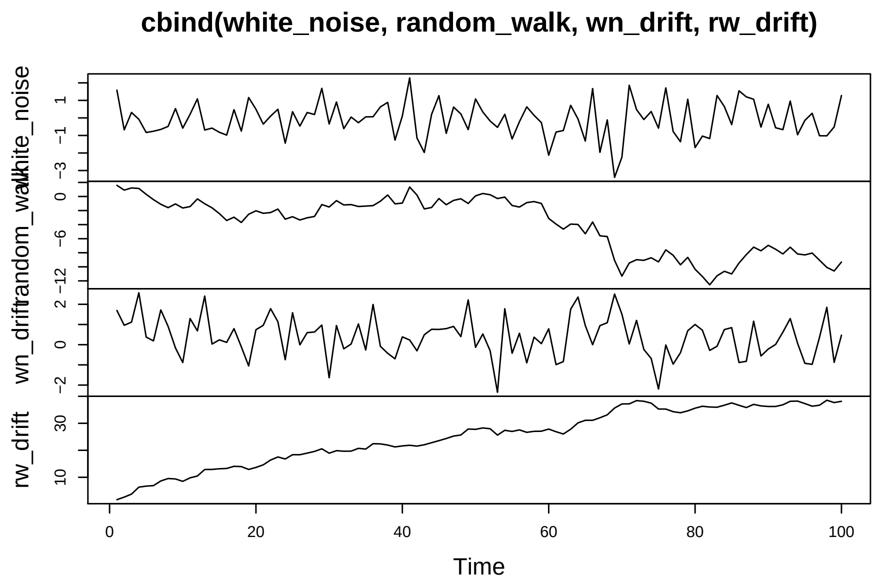



```
X_RW<-arima.sim(list(order=c(0,1,0)), n=100)
autoplot(X_RW)+
ggtitle("Serie Random Walk")
```

1.2 Serie Random Walk (RW)



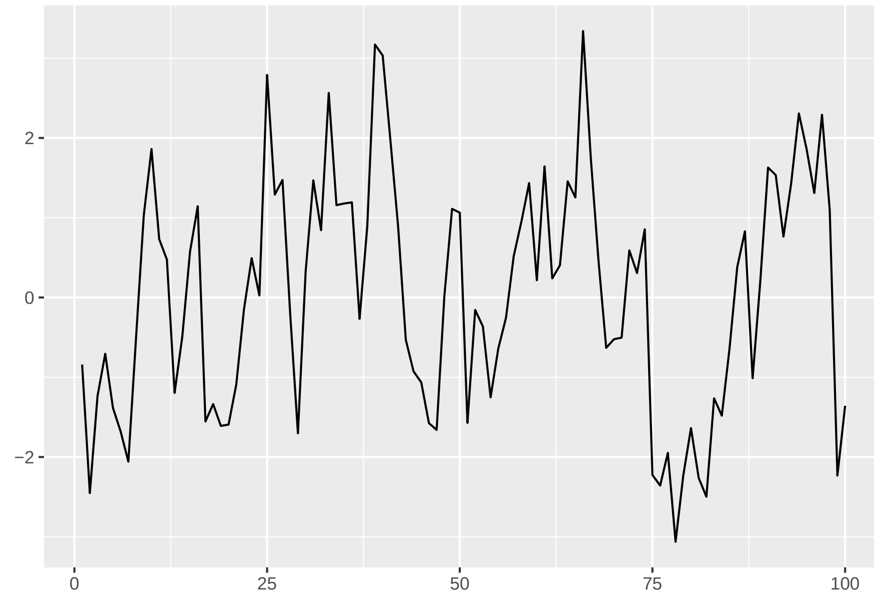

```
white_noise <- arima.sim(list(order = c(0, 0, 0)), n=100)
random_walk <- cumsum(white_noise)
wn_drift <- arima.sim(list(order = c(0, 0, 0)), n=100, mean=0.4)
rw_drift <- cumsum(wn_drift)
plot.ts(cbind(white_noise, random_walk, wn_drift, rw_drift))
```



1.3 Proceso ARMA

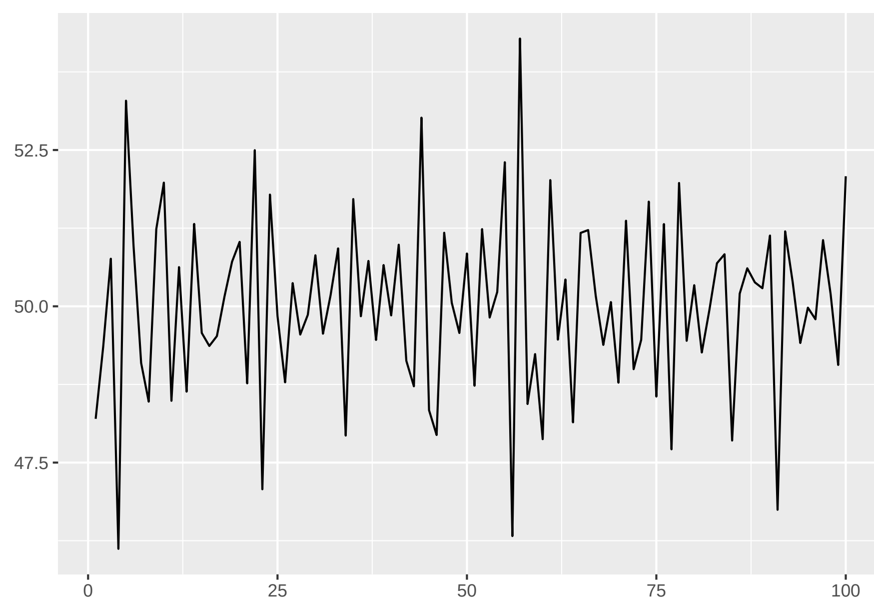
Simulando un proceso AR(1)

```
X_AR1<-arima.sim(list(order=c(1,0,0), ar=c(0.90)), n=100)
autoplot(X_AR1)
```



Simulando un proceso AR(2)

```
X_MA1<-arima.sim(list(order=c(0,0,1), ma=c(-0.98)), n=100)+50
autoplot(X_MA1)
```



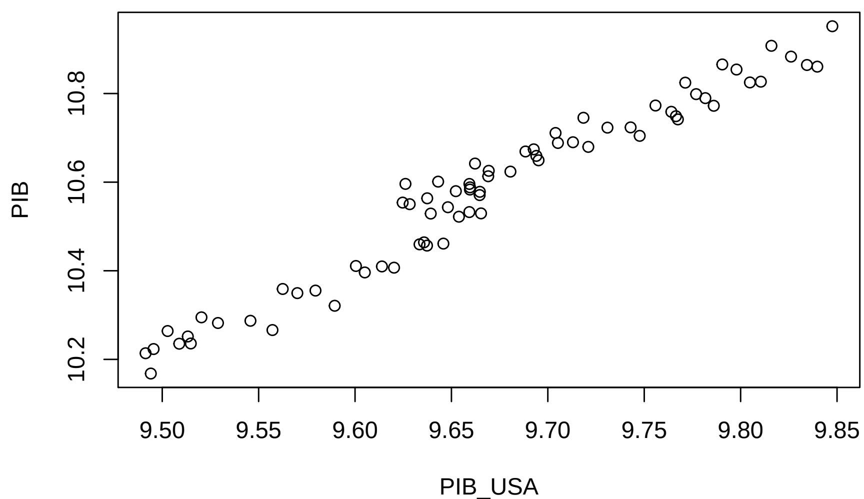
Correlación entre el nivel del PIB de Honduras y el de USA

```
USA<-coredata(log(TRIM$PIB_USA["2001-01-01/"]))
HN<-coredata(log(TRIM$PIB["2001-01-01/"]))
cor(USA,HN)
```

```
##               PIB
## PIB_USA 0.9775886
```

Scatter plot

```
plot(cbind(USA, HN))
```



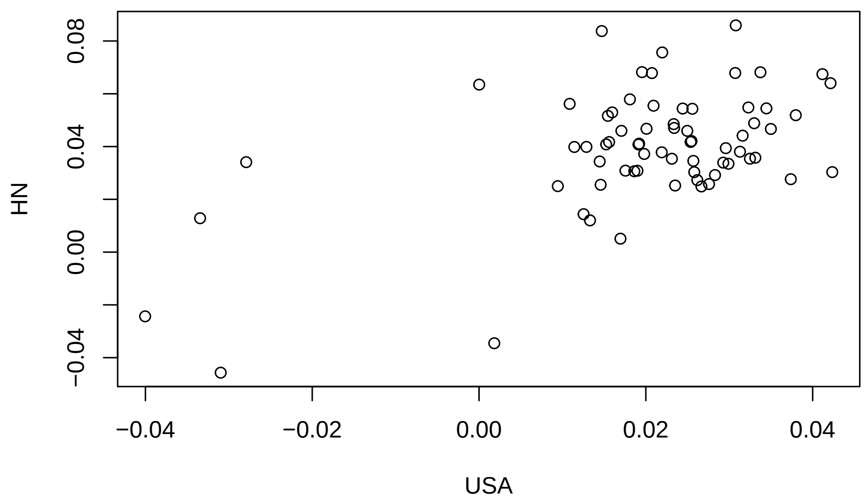
Correlación entre el la tasa de crecimiento del PIB de Honduras y el de USA

```
USA<-coredata(diff(USA, lag=4))
HN<-coredata(diff(HN, lag=4))
cor(USA,HN)
```

```
##               PIB
## PIB_USA 0.5405966
```

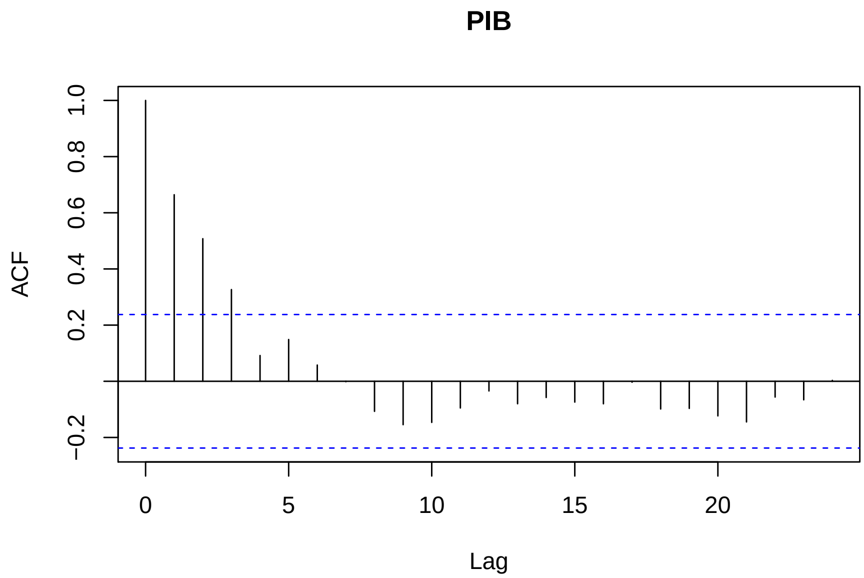
Scatter plot

```
plot(USA, HN)
```



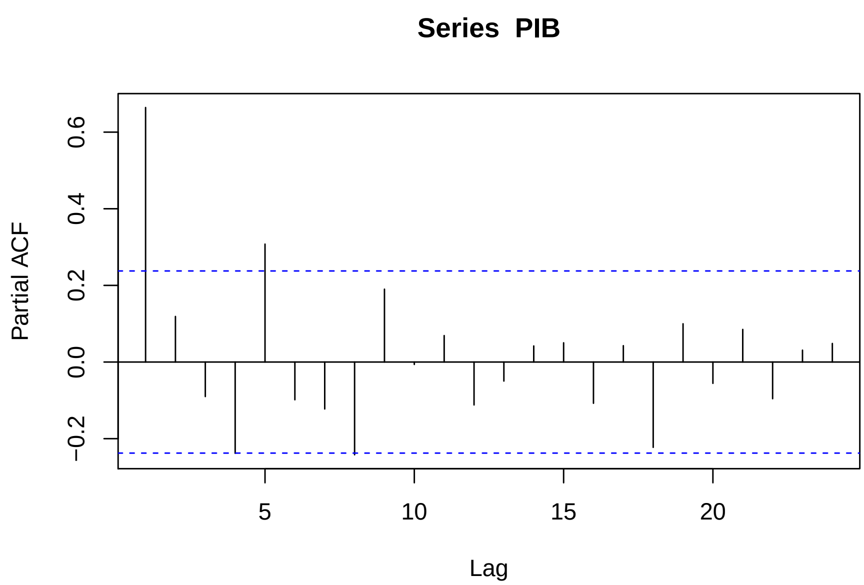
Función de autocorrelación del PIB de Honduras

```
PIB<-as.ts(HN)
acf(PIB, lag.max = 24, plot=TRUE)
```



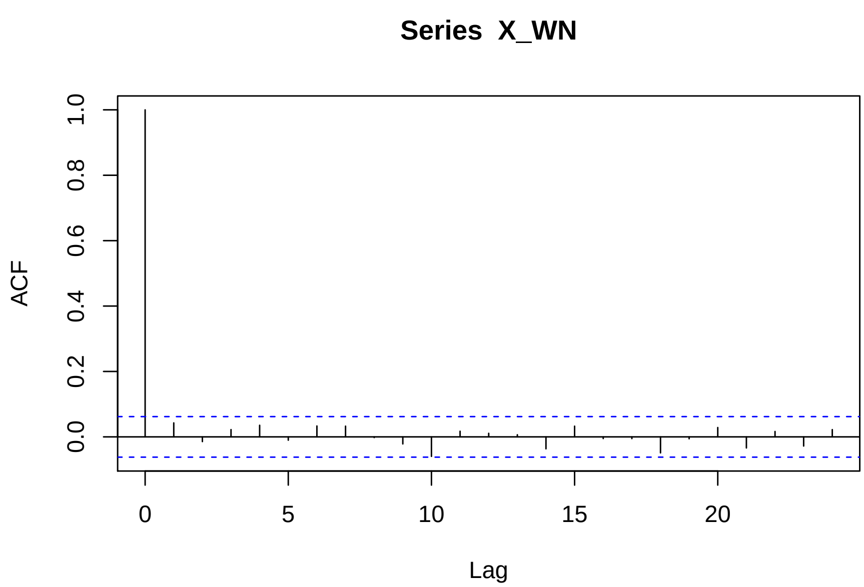
Función de autocorrelación parcial del PIB de Honduras

```
PIB<-as.ts(HN)
pacf(PIB, lag.max = 24, plot=TRUE)
```



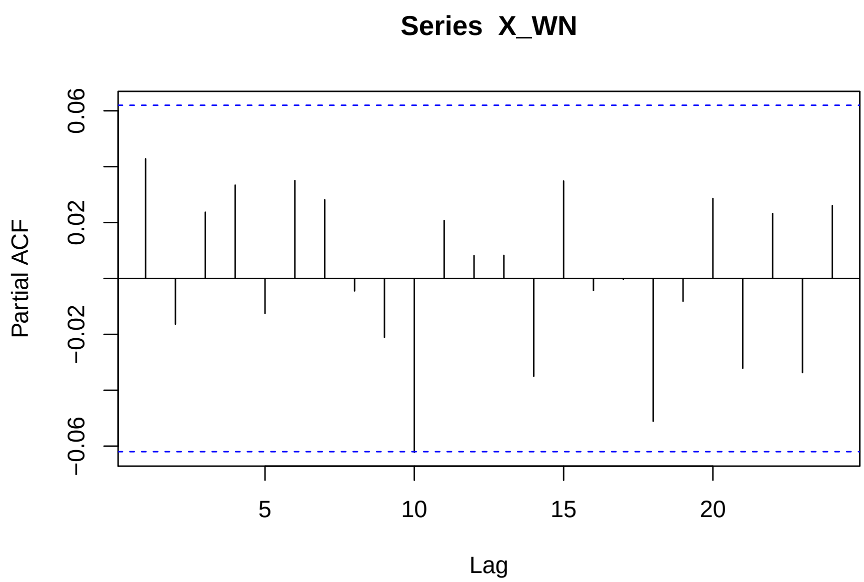
Función de autocorrelación de un proceso ruído blanco

```
acf(X_WN, lag.max = 24, plot=TRUE)
```



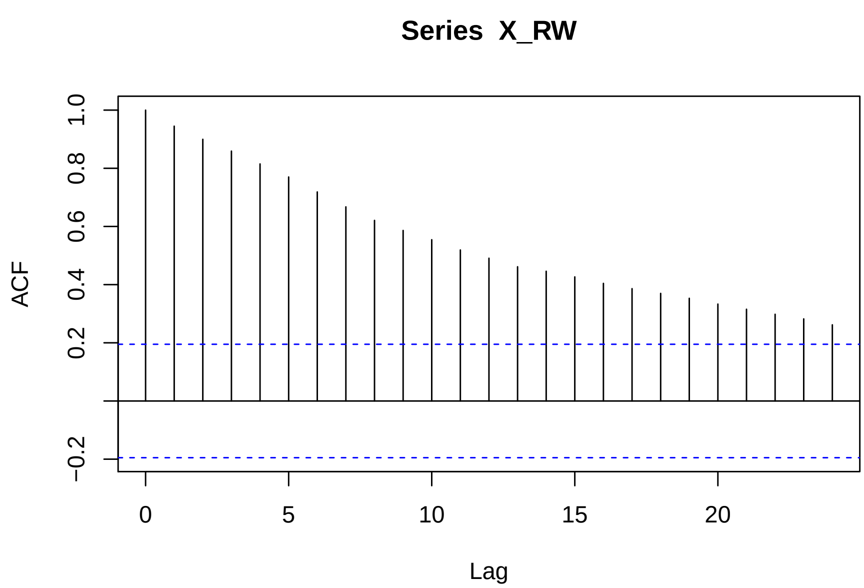
Función de autocorrelación parcial de un proceso ruído blanco

```
pacf(X_WN, lag.max = 24, plot=TRUE)
```



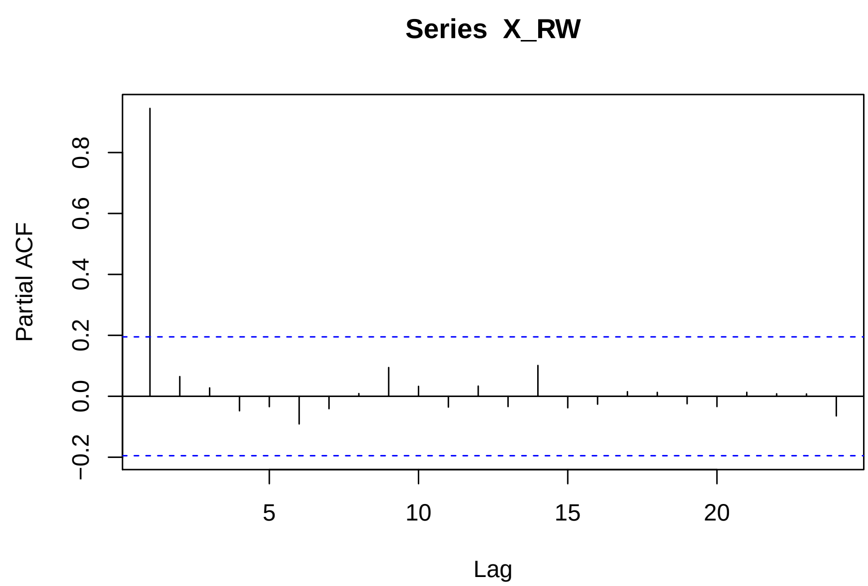
Función de autocorrelación de un proceso RW

```
acf(X_RW, lag.max = 24, plot=TRUE)
```



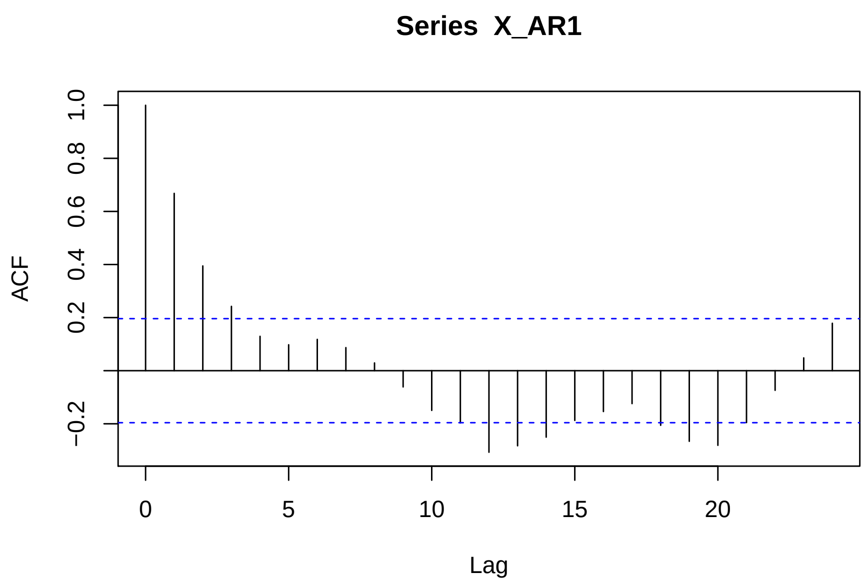
Función de autocorrelación parcial de un proceso RW

```
pacf(X_RW, lag.max = 24, plot=TRUE)
```



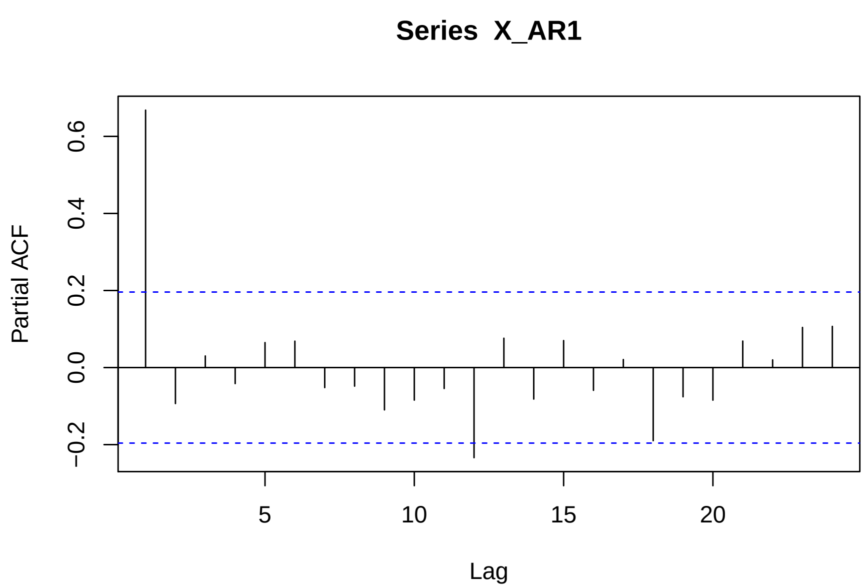
Función de autocorrelación de un proceso AR(1)

```
acf(X_AR1, lag.max = 24, plot=TRUE)
```



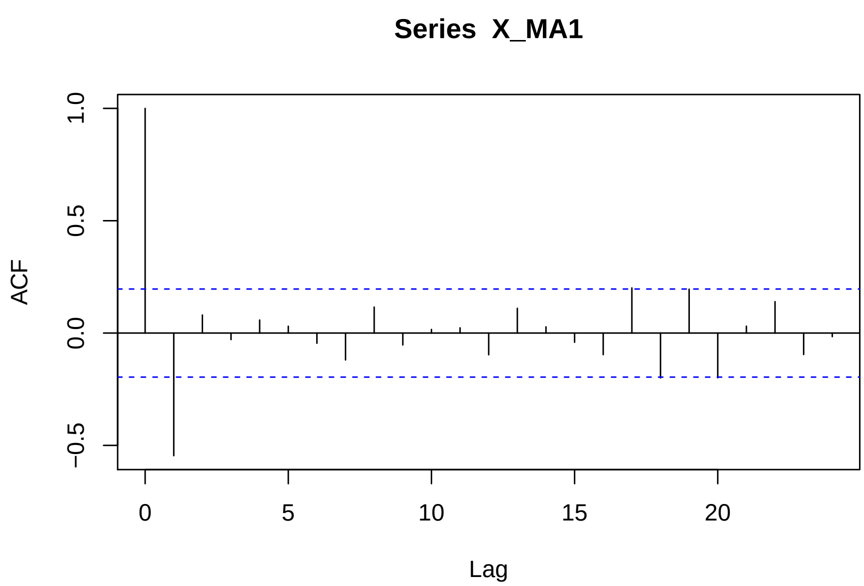
Función de autocorrelación parcial de un proceso AR(1)

```
pacf(X_AR1, lag.max = 24, plot=TRUE)
```



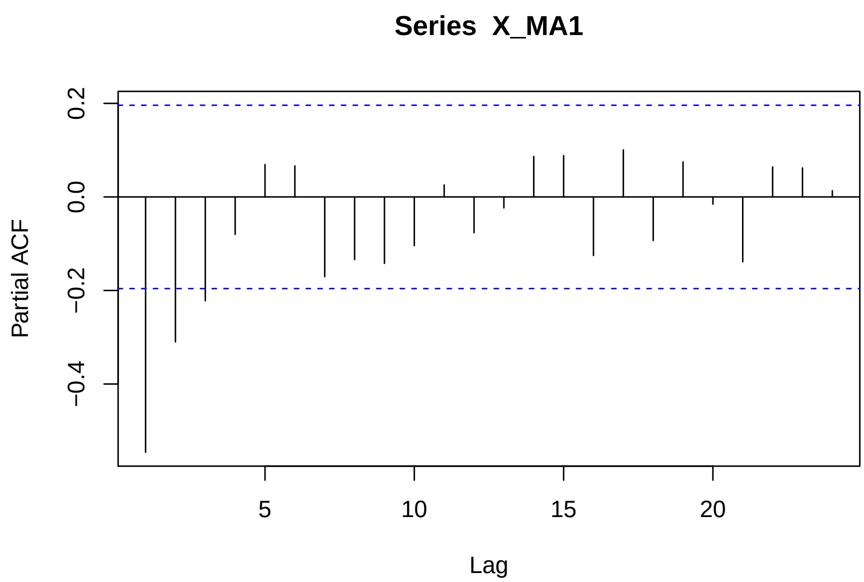
Función de autocorrelación de un proceso MA(1)

```
acf(X_MA1, lag.max = 24, plot=TRUE)
```



Función de autocorrelación parcial de un proceso MA(1)

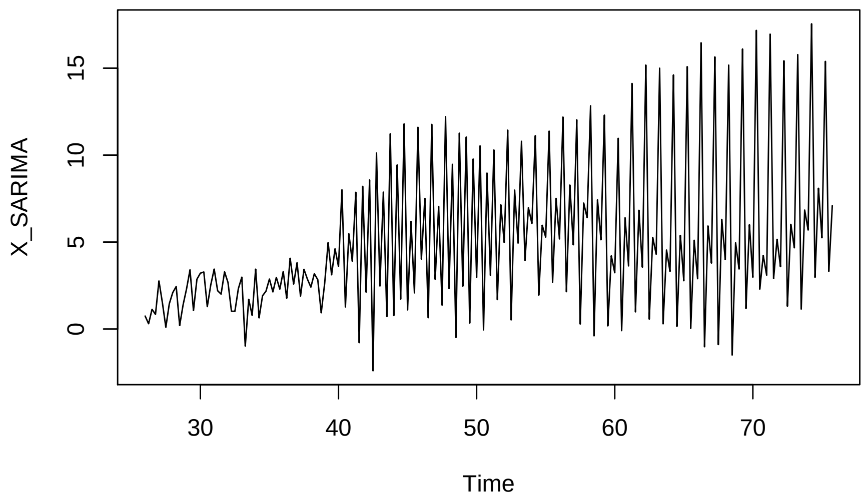
```
pacf(X_MA1, lag.max = 24, plot=TRUE)
```



1.4 Simulación de procesos

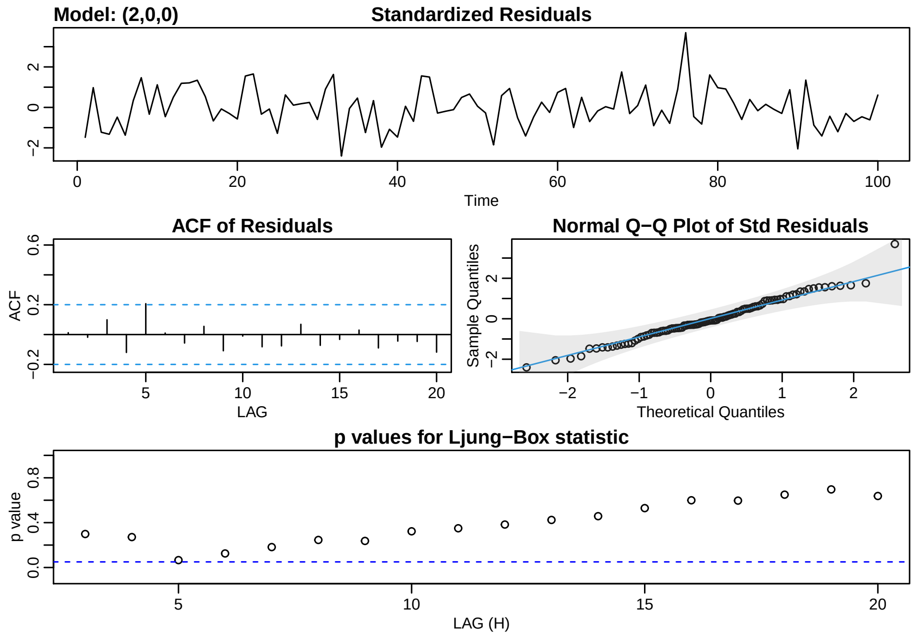
###Estimación de un procesos SARIMA(1,0,1,1,1,1)

```
model    <- Arima(ts(rnorm(100),freq=4), order=c(1,0,1), seasonal=c(1,1,1),
            fixed=c(phi=0.0, theta=-0.0, Phi=0.0, Theta=-0.0))
X_SARIMA<- simulate(model, nsim=200)
plot(X_SARIMA)
```




```
x<-arima.sim(list(order=c(0,0,2), ma=c(1.5,-0.75)), n=100)+50
x_fit<-sarima(x, p=2, d=0, q=0)
```

1.4.1 Estimación de un proceso ARIMA


Model: (2,0,0)
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q−Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung−Box statistic
p value
LAG (H)

```
x_fit$ttable
```

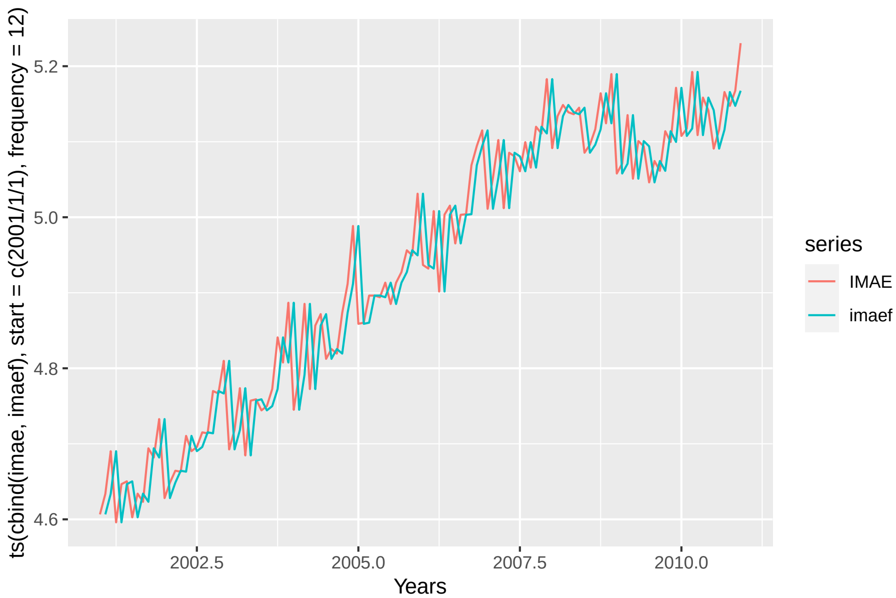
Chapter 2

Pronósticos

2.1 Modelos introductorios

Pronósticos Naive del IMAE de Honduras vs data observada

```
imae<-log(MES$IMAE["2001-01-01/2010-12-01"])
IMAE_NAIVE<-naive(imae)
imaef<-ts(fitted(IMAE_NAIVE), frequency=12, start=c(2001/01/01))
imaef<-as.xts(imaef)
autoplot(ts(cbind(imae, imaef), start = c(2001/01/01), frequency = 12 ),
         facets = FALSE)+xlab("Years")
```

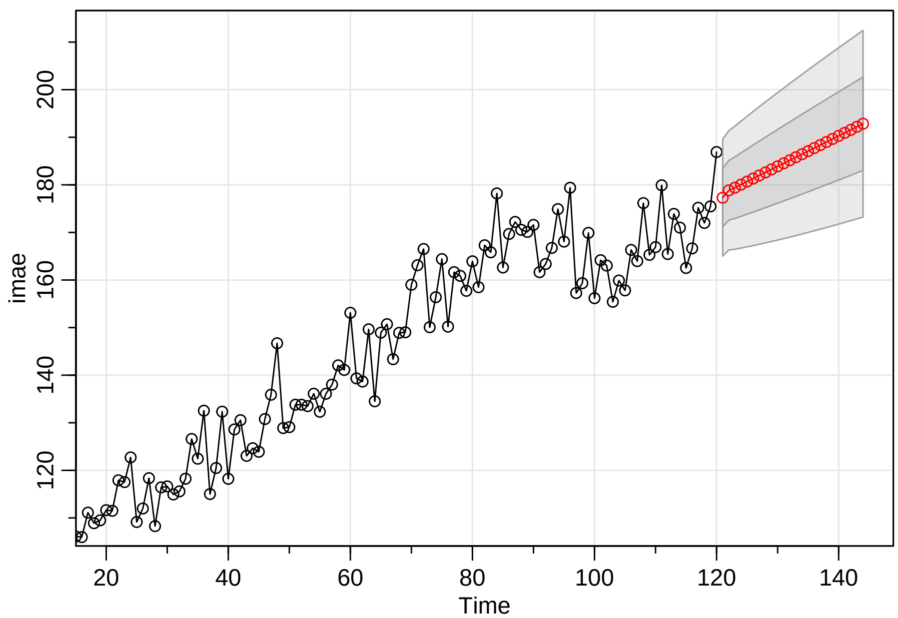


Pronósticos del IMAE de Honduras 24 meses en adelante a partir de un proceso SARIMA(1,1,1)

```
imae<-IMAE["2001-01-01/2010-12-01"]
imaef<-IMAE["/2012-12-01"]
resultado<-sarima.for(imae, n.ahead=24,1,1,1)
```

Table 2.1: Regresión entre el nivel del PIB de Honduras con respecto al de USA

term	estimate	std.error	statistic	p.value
(Intercept)	-9.6056	0.4685	-20.5041	0
log(TRIM$PIB_USA)	2.0873	0.0485	43.0384	0



2.2 Modelos para hacer pronósticos del PIB de Honduras

Modelo de regresión

```
library(knitr)
library(dplyr)
library(broom)
library(AER)
TRIM<-as.xts(read.zoo("FINAL_HN_P.csv", index.column = 1, sep = ";", header=TRUE, format = "%d/%m
M.ols <- lm(log(TRIM$PIB) ~ log(TRIM$PIB_USA))
kable(tidy(M.ols), digits=4, align='c',caption="Regresión entre el nivel del PIB de Honduras con
```

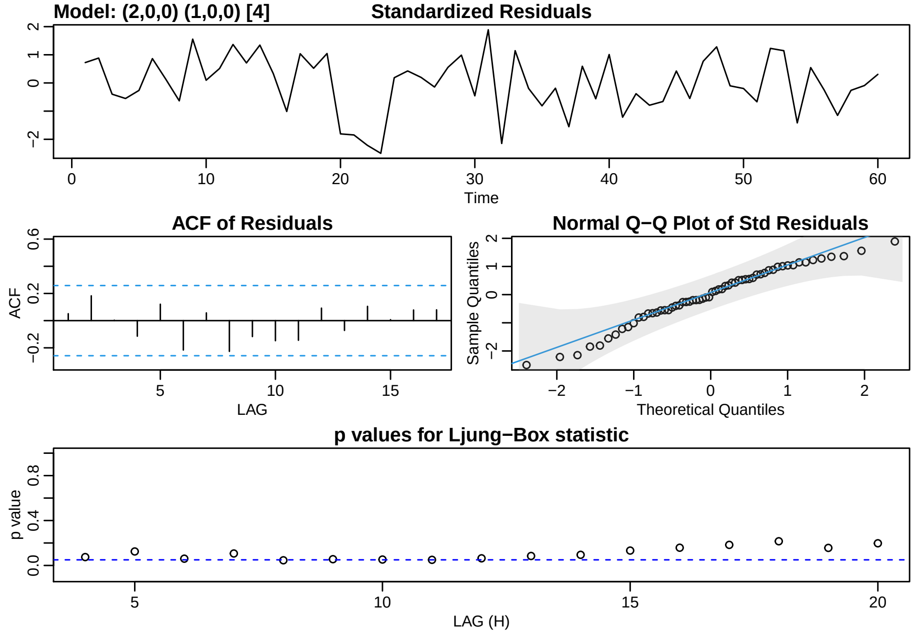
Modelo de regresión para el PIB de Honduras

```
INDEX  <-factor(index(TRIM))
dummies<-model.matrix(~INDEX)
```

```
TRIM   <-merge(TRIM, dummies, join="left")
Y      <-window(diff(log(TRIM$PIB), lag=4)*100, start="2004-03-01", end="2018-12-01")
Y_USA  <-window(diff(log(TRIM$PIB_USA), lag=4)*100, start="2004-03-01", end="2018-12-0
DUM_HN <-window(TRIM[, c("INDEX2005.09.01", "INDEX2006.12.01", "INDEX2008.06.01")], st
i_HN   <-window(diff(TRIM$TASA_P, lag=1)*100, start="2004-03-01", end="2018-12-01")
REG_HN <- merge(DUM_HN, Y_USA, join="left")
REG_HN <- merge(REG_HN, i_HN,  join="left")
PIB_HN <-sarima(Y, 2,0,0,P=1, D=0, Q=0, 4, xreg=REG_HN)
```

```
## initial  value 0.542818
## iter   2 value 0.399627
## iter   3 value 0.369226
## iter   4 value 0.292117
## iter   5 value 0.266433
## iter   6 value 0.252289
## iter   7 value 0.225647
## iter   8 value 0.225239
## iter   9 value 0.217205
## iter  10 value 0.210556
## iter  11 value 0.209208
## iter  12 value 0.204386
## iter  13 value 0.204299
## iter  14 value 0.204282
## iter  15 value 0.204281
## iter  16 value 0.204281
## iter  17 value 0.204281
## iter  18 value 0.204281
## iter  19 value 0.204281
## iter  19 value 0.204281
## iter  19 value 0.204281
## final  value 0.204281
## converged
## initial  value 0.186966
## iter   2 value 0.186057
## iter   3 value 0.185621
## iter   4 value 0.185490
## iter   5 value 0.185264
## iter   6 value 0.185174
## iter   7 value 0.185124
## iter   8 value 0.185091
## iter   9 value 0.185069
## iter  10 value 0.185068
## iter  11 value 0.185068
## iter  12 value 0.185068
## iter  13 value 0.185068
```

```
## iter  13 value 0.185068
## iter  13 value 0.185068
## final  value 0.185068
## converged
```




```r
PIB_HN$ttable
```

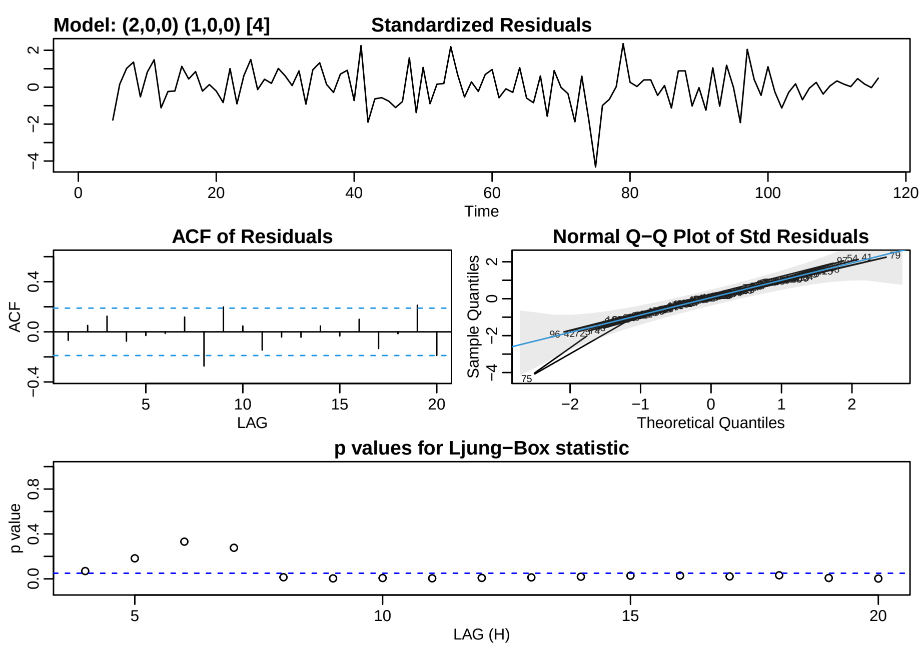
```
##                  Estimate     SE t.value p.value
## ar1                0.5867 0.1333  4.4015  0.0001
## ar2                0.2160 0.1364  1.5839  0.1194
## sar1              -0.3799 0.1277 -2.9750  0.0045
## intercept          2.4738 0.6710  3.6869  0.0006
## INDEX2005.09.01    3.3296 0.9658  3.4473  0.0011
## INDEX2006.12.01    2.0824 1.0148  2.0521  0.0453
## INDEX2008.06.01    2.2639 1.0536  2.1487  0.0364
## PIB_USA            0.7381 0.1920  3.8438  0.0003
## TASA_P             0.0056 0.0028  1.9794  0.0532
```

Modelo de regresión para el PIB de USA

```r
Y_USA     <-window(diff(log(TRIM$PIB_USA), lag=4)*100, start="1990-03-01", end="2018-12-01")
DUM_USA   <-window(TRIM[, c("INDEX2008.12.01", "INDEX2009.12.01")], start="1990-03-01", end="2018
PIB_USA   <-sarima(Y_USA, 2,0,0,P=1, D=0, Q=0, 4, xreg=DUM_USA )
```

```
## initial  value 0.434672
## iter   2 value 0.189957
## iter   3 value 0.021610
```

```
## iter   4 value -0.116248
## iter   5 value -0.251754
## iter   6 value -0.332615
## iter   7 value -0.410324
## iter   8 value -0.433150
## iter   9 value -0.436896
## iter  10 value -0.439417
## iter  11 value -0.440979
## iter  12 value -0.441051
## iter  13 value -0.441096
## iter  14 value -0.441109
## iter  15 value -0.441110
## iter  16 value -0.441110
## iter  17 value -0.441111
## iter  18 value -0.441115
## iter  19 value -0.441117
## iter  20 value -0.441118
## iter  21 value -0.441119
## iter  22 value -0.441119
## iter  22 value -0.441119
## iter  22 value -0.441119
## final  value -0.441119
## converged
```

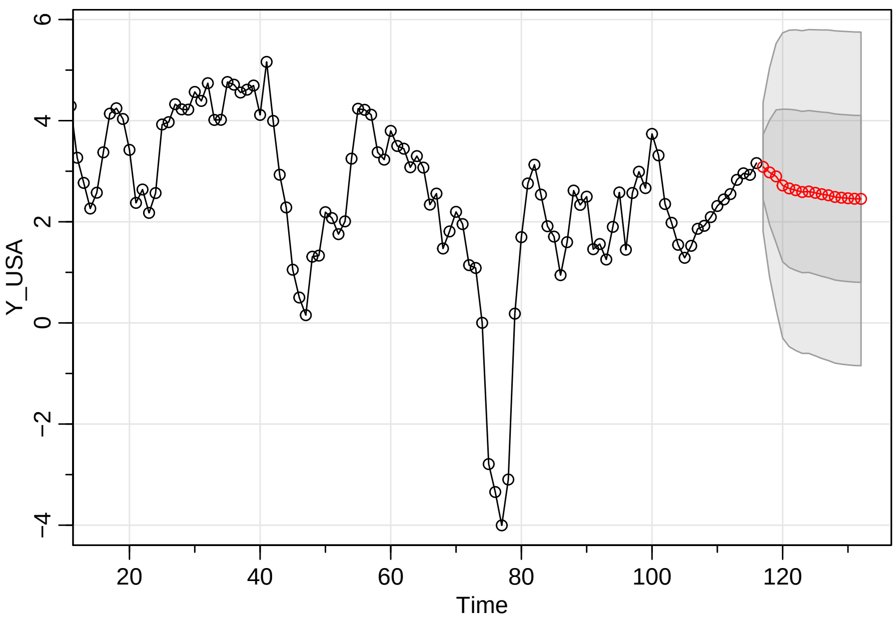


```
PIB_USA$ttable
```

```
##                   Estimate     SE t.value p.value
## ar1                 1.2831 0.0888 14.4513  0.0000
## ar2                -0.3694 0.0899 -4.1111  0.0001
## sar1               -0.3721 0.0925 -4.0228  0.0001
## intercept           2.4027 0.4862  4.9417  0.0000
## INDEX2008.12.01     0.3768 0.3851  0.9784  0.3301
## INDEX2009.12.01    -0.1896 0.3823 -0.4958  0.6210
```

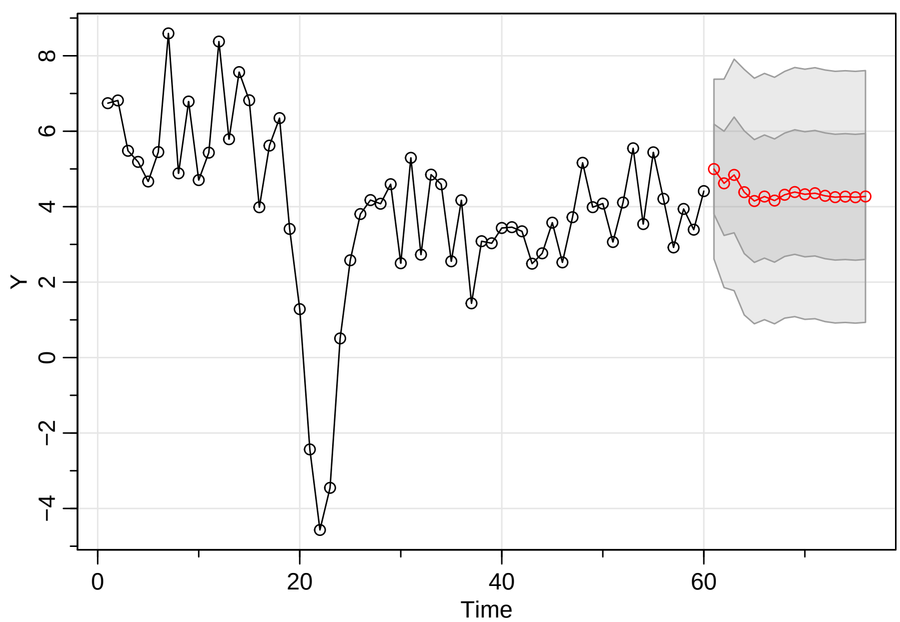
Pronóstico del PIB de USA

```
DUM_USA_N <-window(TRIM[, c("INDEX2008.12.01", "INDEX2009.12.01")], start="2019-03-01", end="2022
Y_USA_N   <-sarima.for(Y_USA,16,2,0,0,1,0,0,4, xreg=DUM_USA, newxreg=DUM_USA_N)
```



Pronóstico del PIB de Honduras

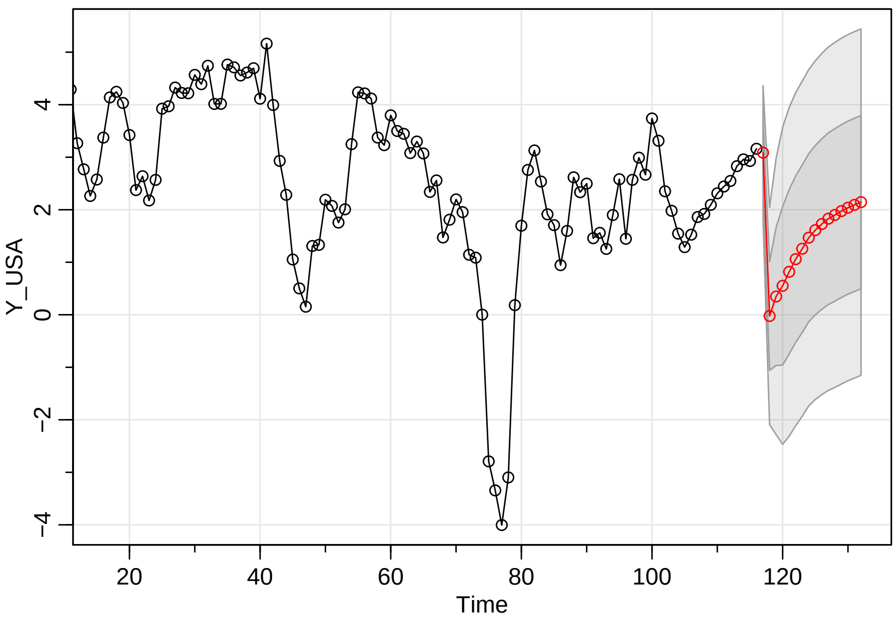
```
dates <- seq(as.Date("2019-03-01"), length = 16, by = "quarter")
DUM_HN_N <-window(TRIM[, c("INDEX2005.09.01", "INDEX2006.12.01", "INDEX2008.06.01")], start="2019
Y_USA_N   <- xts(x=Y_USA_N$pred, order.by = dates)
REG_HN_N<- merge(DUM_HN_N, Y_USA_N, join="left")
data <- rep(1, 16)
i_HN_N = xts(x = data, order.by = dates)
REG_HN_N<- merge(REG_HN_N, i_HN_N, join="left")
Y_N<-sarima.for(Y,16,2,0,0,1,0,0,4, xreg=REG_HN, newxreg=REG_HN_N)
```



2.3 Simulación de shock en el PIB de USA

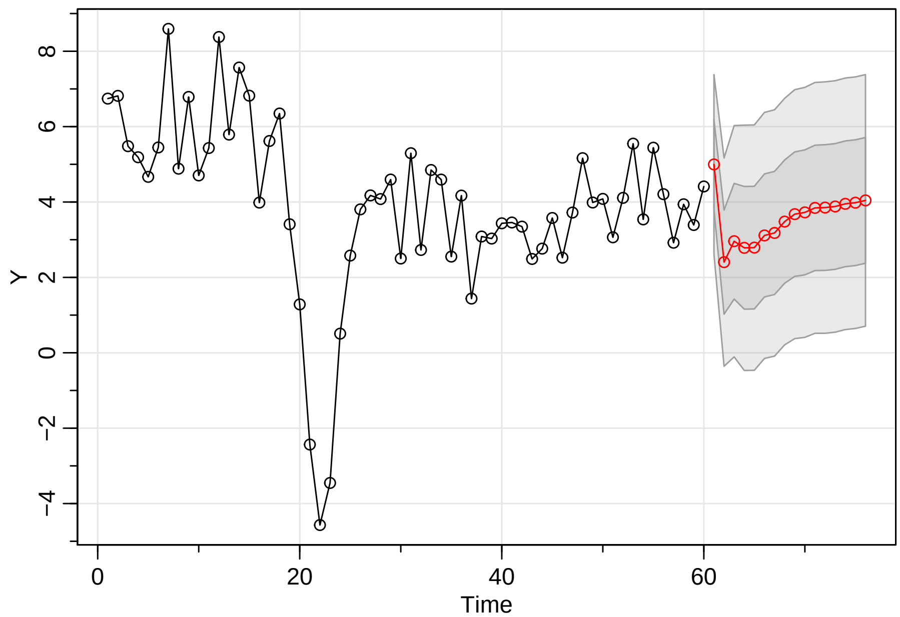
Simulación

```
dates <- seq(as.Date("2019-03-01"), length = 16, by = "quarter")
shock <-c()
shock[1]<- 0
shock[2]<- -3*(1/-0.1896)
for(i in 3:16 ){
  shock[i]<-0.85*shock[i-1]
}
shock_Y_USA= xts(x = shock, order.by = dates)
REG_SHOCK<-window(TRIM[, c("INDEX2008.12.01")], start="2019-03-01", end="2022-12-01")
REG_SHOCK<- merge(REG_SHOCK, shock_Y_USA, join="left")
Y_USA_SHOCK<-sarima.for(Y_USA,16,2,0,0,1,0,0,4, xreg=DUM_USA, newxreg=REG_SHOCK)
```



Transimisión del shock al PIB de Honduras

```
Y_USA_S <- xts(x=Y_USA_SHOCK$pred, order.by = dates)
REG_HN_S<- merge(DUM_HN_N, Y_USA_S, join="left")
REG_HN_S<- merge(REG_HN_S, i_HN_N, join="left")
Y_S<-       sarima.for(Y,16,2,0,0,1,0,0,4, xreg=REG_HN, newxreg=REG_HN_S)
```



Y
8
6
4
2
0
−2
−4
0
20
40
60
Time

Chapter 3

Ejercicio fuera de muestra

3.1 Aplicación sobre la inflación de Honduras

Los pronósticos de inflación a determinado horizonte h, realizados en el momento t a partir de una determinada especificación econométrica j ($\hat{\pi}^{j}_{t+h|t}$) son comparados con los datos efectivos (π_{t+h}), deduciendo los errores de pronósticos (E^{j}_{t+h}) en conformidad a la ecuación la cual aplica para cualquier tipo de ventana h.

$$ECM^{j}_{h} = \frac{\sum_{n=0}^{N=g_h-1} (E^{j}_{t+h+n})^2}{g_h} \tag{3.1}$$

En el siguiente código, se seleccionará la mejor específicación según el criterio de información Akaike (AIC) para la inflación de Honduras tomando como muestra enero 1994 a diciembre 2006.

```
library("xts")
library("zoo")
library("astsa")
library("forecast")
library("ggplot2")
library("forecast")
library("ggfortify")
library("stargazer")
library("urca")
library("dynlm")
library("scales")
library("quantmod")
library("dplyr")
```

```
library("sandwich")
library("knitr")
library("dynlm")
library("stargazer")
MES<-as.xts(read.zoo("MES_HN.csv", index.column = 1, sep = ";", header=TRUE, format = "
INFLA <-(log(MES$IPC)-stats::lag(log(MES$IPC), n=12))/stats::lag(log(MES$IPC), n=12)
INFLA1<-INFLA["1994-01-01/2006-12-01"]
INFLA_fit <-auto.arima(INFLA1, ic = c("aic"))
INFLA_fit
```

```
## Series: INFLA1
## ARIMA(1,1,2) with drift
##
## Coefficients:
##           ar1      ma1      ma2  drift
##       -0.6503  -0.1120  -0.7132  0e+00
## s.e.   0.1504   0.1212   0.1035  1e-04
##
## sigma^2 estimated as 2.125e-06:  log likelihood=793.57
## AIC=-1577.13   AICc=-1576.73   BIC=-1561.91
```

La mejor específicación seleccionada es un a ARIMA(1,1,2). El que formalmente puede escribirse como:

$$\Phi(L)(1-L)\pi_t = \Theta(L)\epsilon_t \tag{3.2}$$

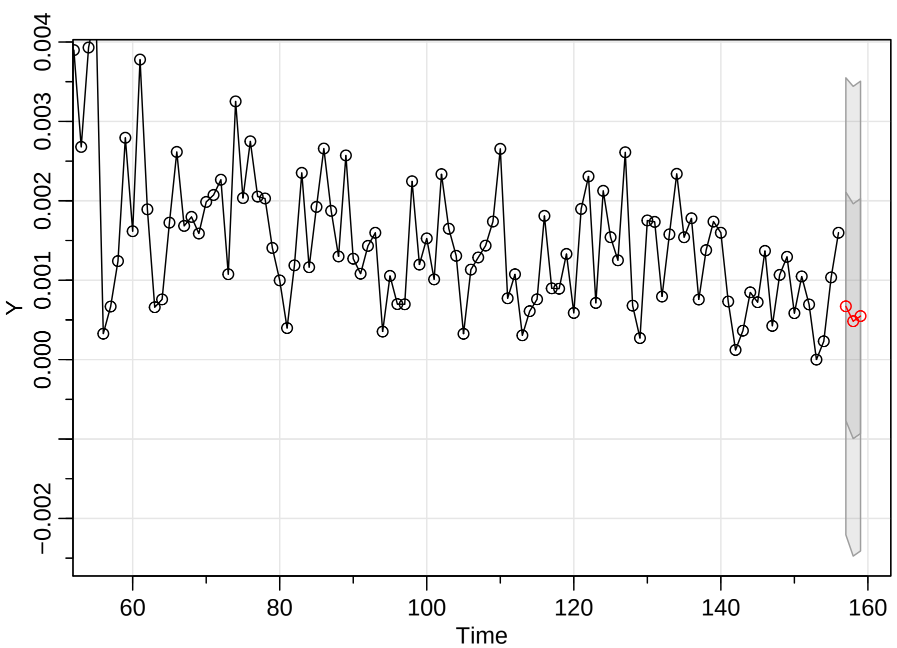
Donde $\Phi(L) = (1+0.65L)$, $\Theta(L) = (1-0.11L-0.71L^2)$.

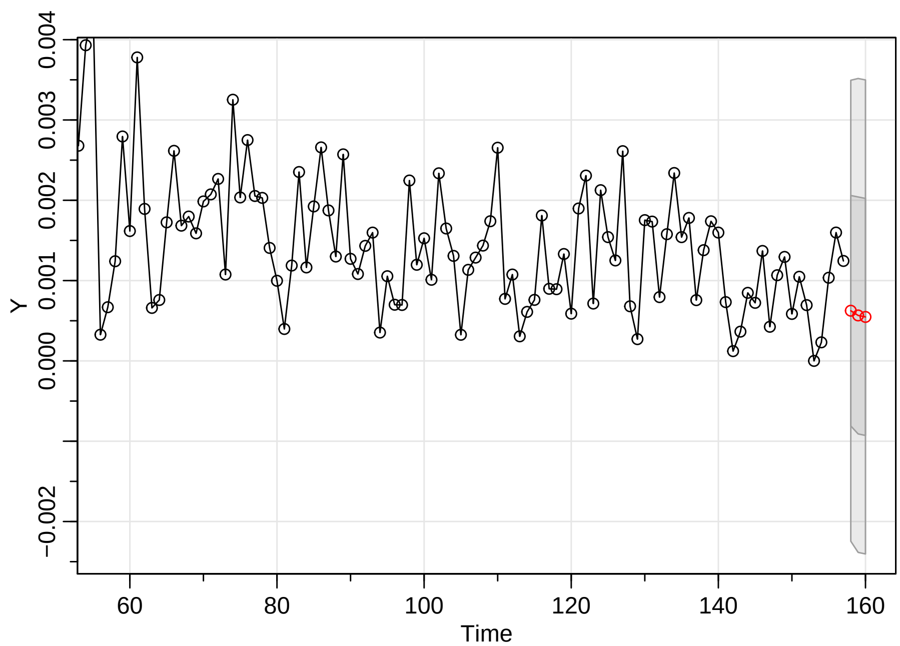
Luego de tener esa específicación podemos comparar su desempeño predictivo con respecto a los datos observados que se encuentran *fuera de la muestra de estimación*.

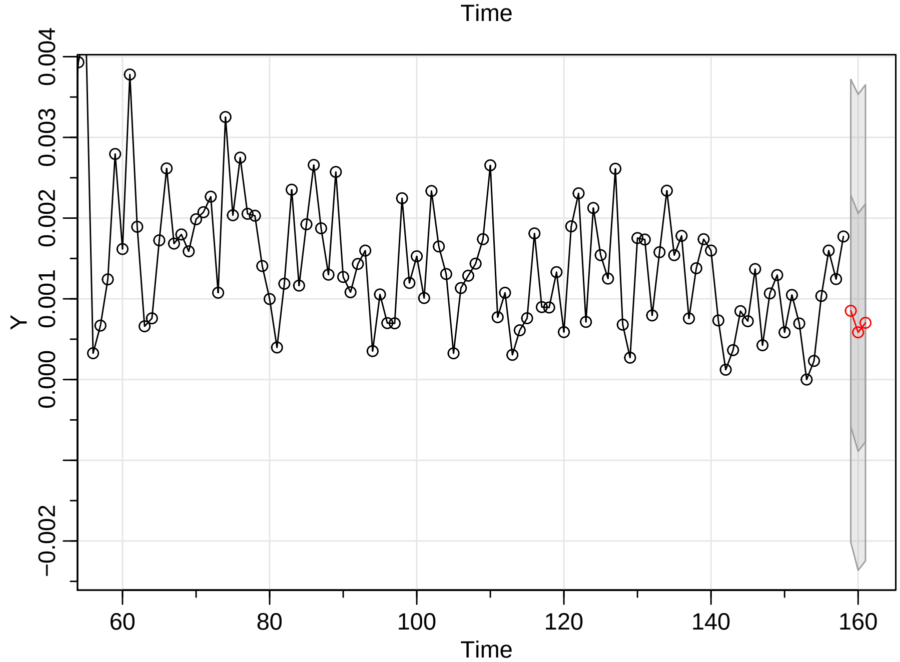
Para ejemplificar se selecciona como horizonte predictivo 3 meses.

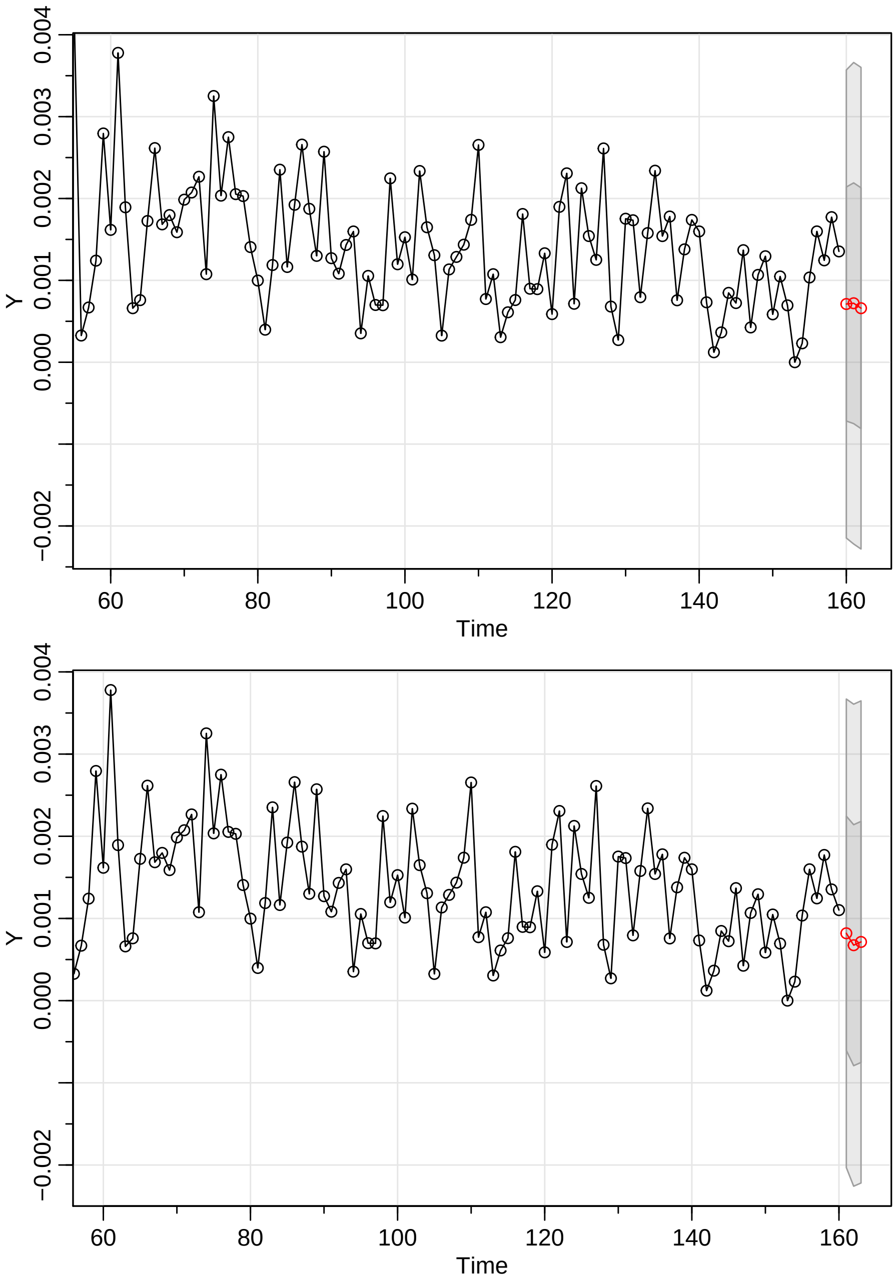
```
FECHA            <-index(MES)
t_inicial        <-first(FECHA,'1 month')
index_final      <-last(index(FECHA))
fecha_contador   <-seq(as.Date(t_inicial), length =index_final, by = "months")
counter          <-c(1:index_final)
contador         <-xts(x=counter, order.by = fecha_contador)
inicio_estimacion<-coredata(contador["1994-01-01"])[1]
final_estimacion <-coredata(contador["2006-12-01"])[1]
final_muestra    <-coredata(contador["2019-02-01"])[1]
H                <-3 #Horizonte predictivo
DENTRO           <- seq(as.Date(FECHA[inicio_estimacion]),
length =final_estimacion+H-inicio_estimacion, by = "months")
FUERA            <- seq(as.Date(FECHA[final_estimacion+1]),
```

```
length =final_muestra-H-final_estimacion, by = "months")
assign(paste('PI_', H, sep=''), xts(x=window(INFLA, start=FECHA[inicio_estimacion], end=FECHA[fir
order.by = DENTRO))
Y <-window(INFLA, start=FECHA[inicio_estimacion], end=FECHA[final_estimacion])
for(i in 1:length(FUERA)){
Y_F<-sarima.for(Y,H,1,1,2, xreg=NULL,
                newxreg=NULL, plot= FALSE)
dates_out<-as.Date(FECHA[final_estimacion+i+H-1])
Y_F_P<-xts(x=Y_F$pred[H], order.by = dates_out)
DENTRO<-seq(as.Date(FECHA[inicio_estimacion]),
length =final_estimacion+1+i-inicio_estimacion, by = "months")
Y <-window(INFLA, start=FECHA[inicio_estimacion], end=FECHA[final_estimacion+i])
assign(paste('PI_', H, sep=''),rbind(get(paste('PI_', H, sep='')),
Y_F_P))
}
```

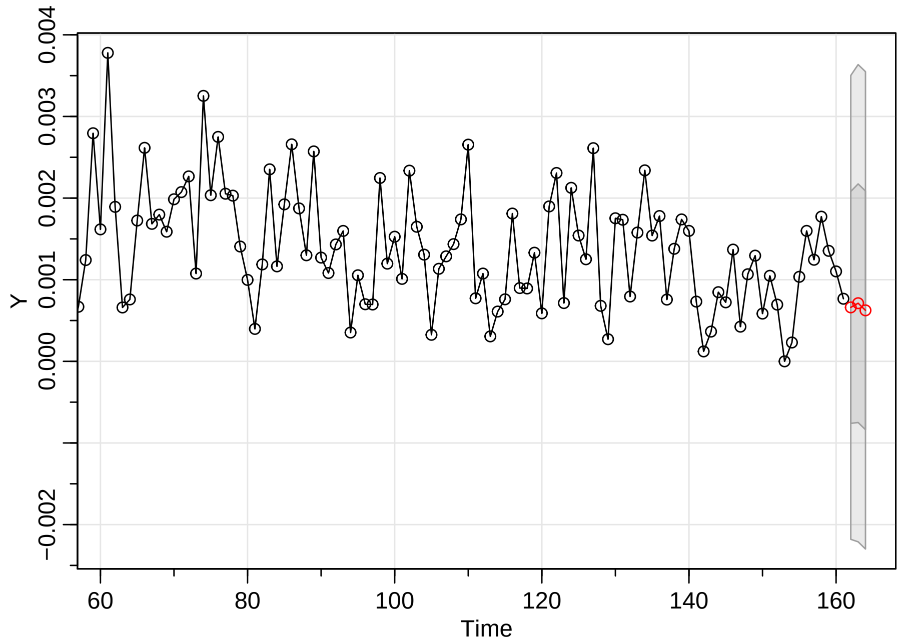



0.004
0.003
0.002
0.001
0.000
−0.002
Y
60
80
100
120
140
160
Time

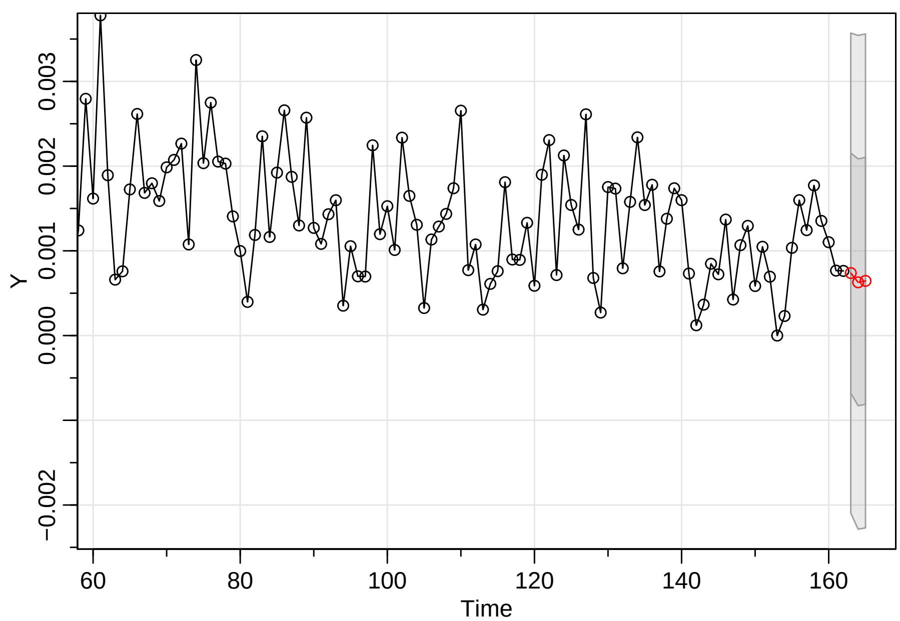

0.004
0.003
0.002
0.001
0.000
−0.002
Y
60
80
100
120
140
160
Time



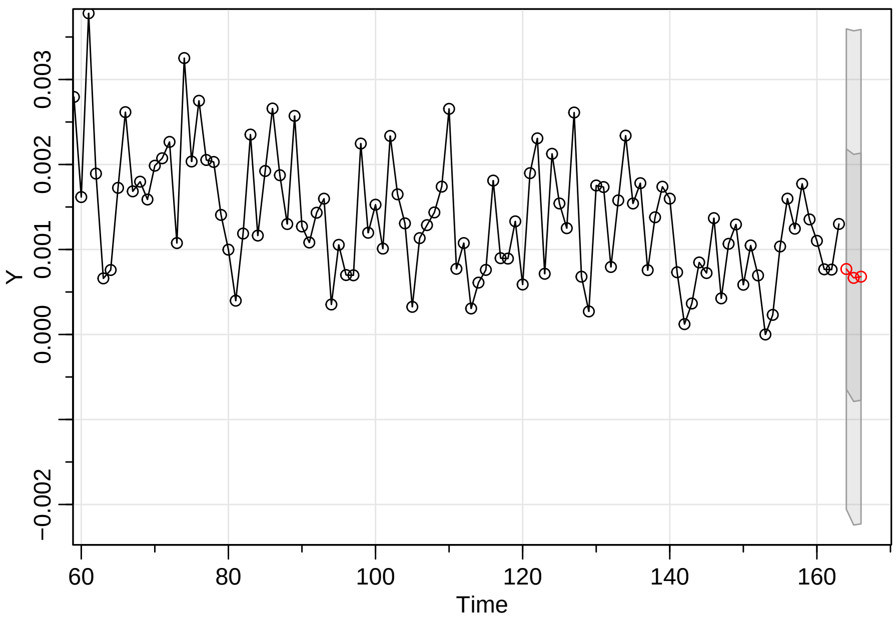
Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
60
80
100
120
140
160

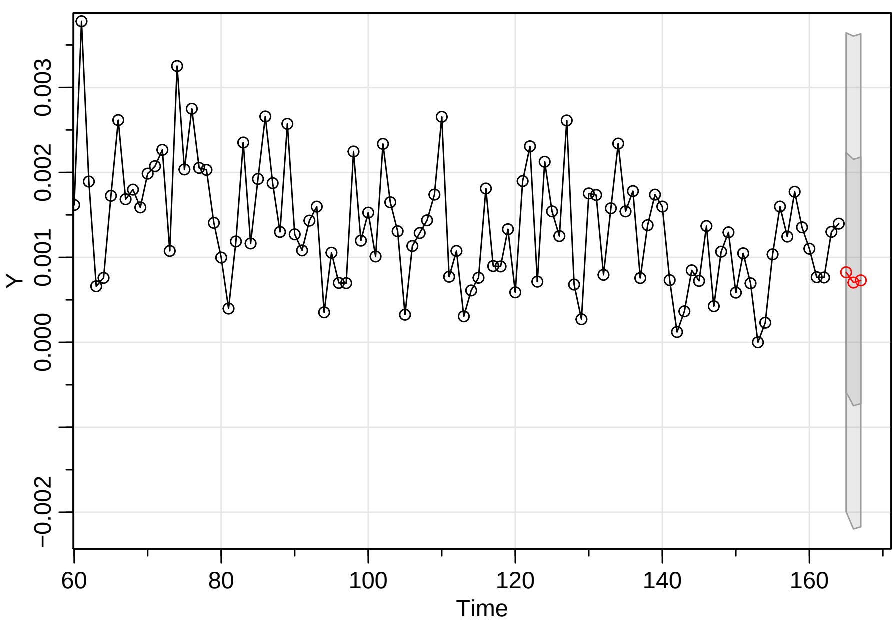


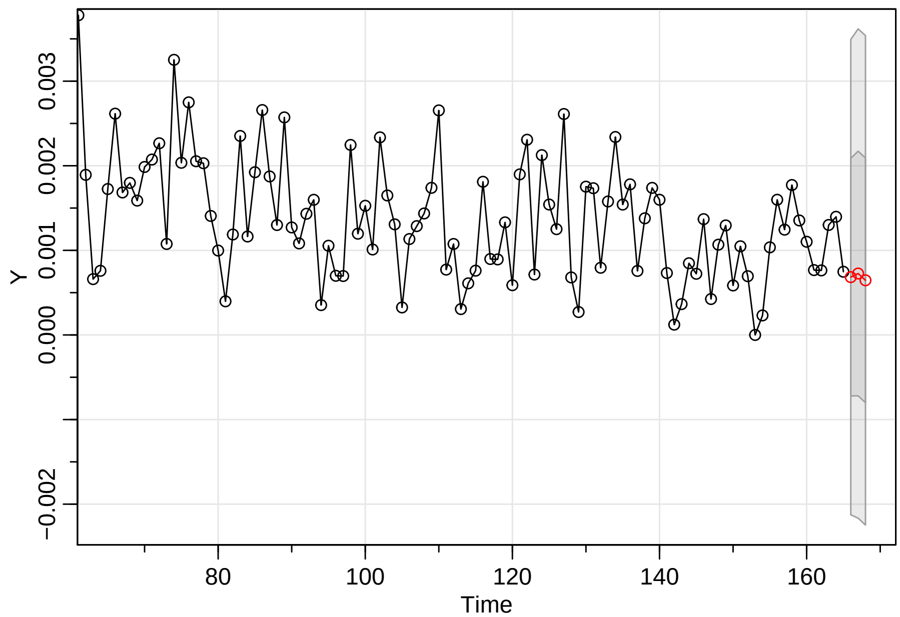
Y
0.004
0.003
0.002
0.001
0.000
−0.002
60
80
100
120
140
160
Time

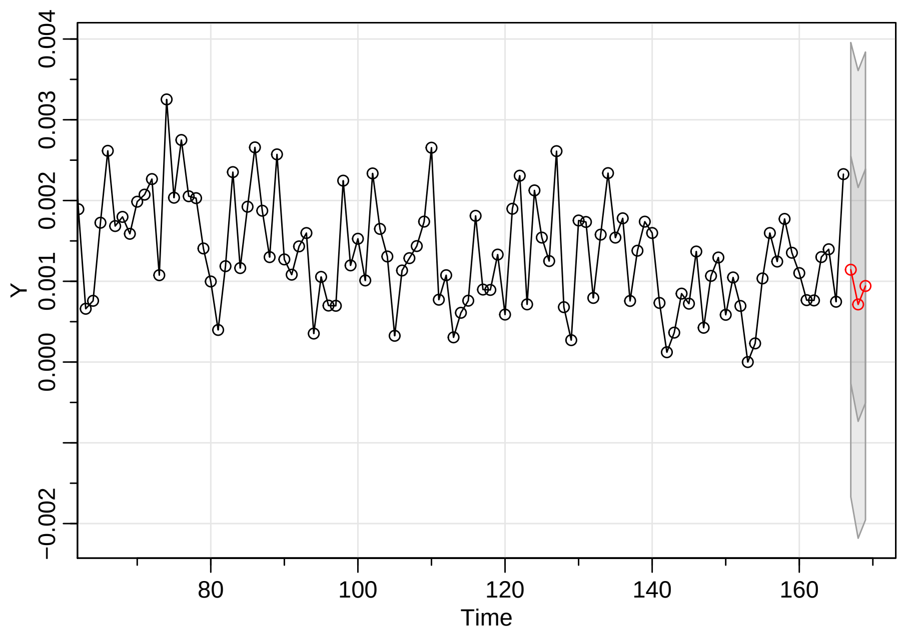


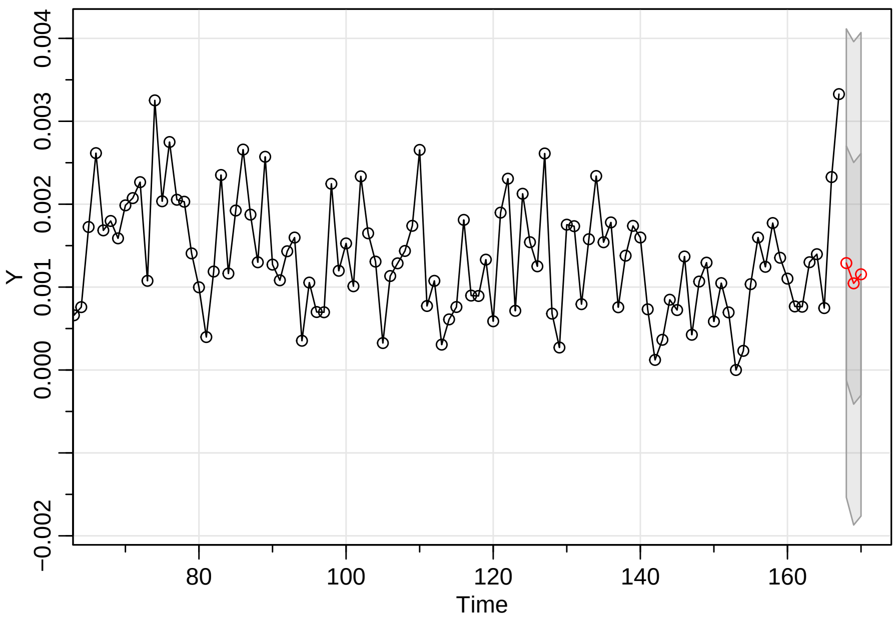
Y
0.003
0.002
0.001
0.000
−0.002
60
80
100
120
140
160
Time


Y
−0.002
0.000
0.001
0.002
0.003
60
80
100
120
140
160
Time

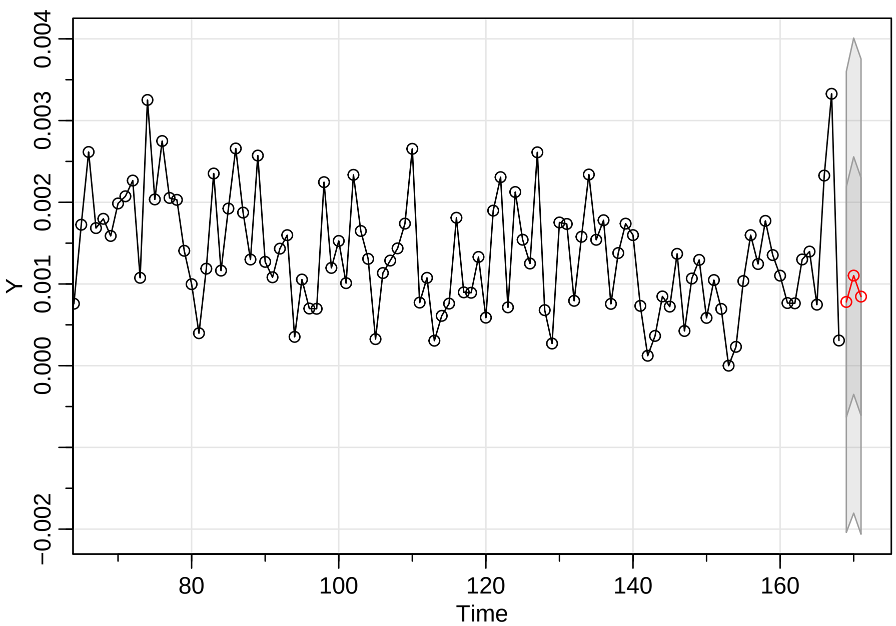

Y
−0.002
0.000
0.001
0.002
0.003
60
80
100
120
140
160
Time


Y
−0.002
0.000
0.001
0.002
0.003
80
100
120
140
160
Time

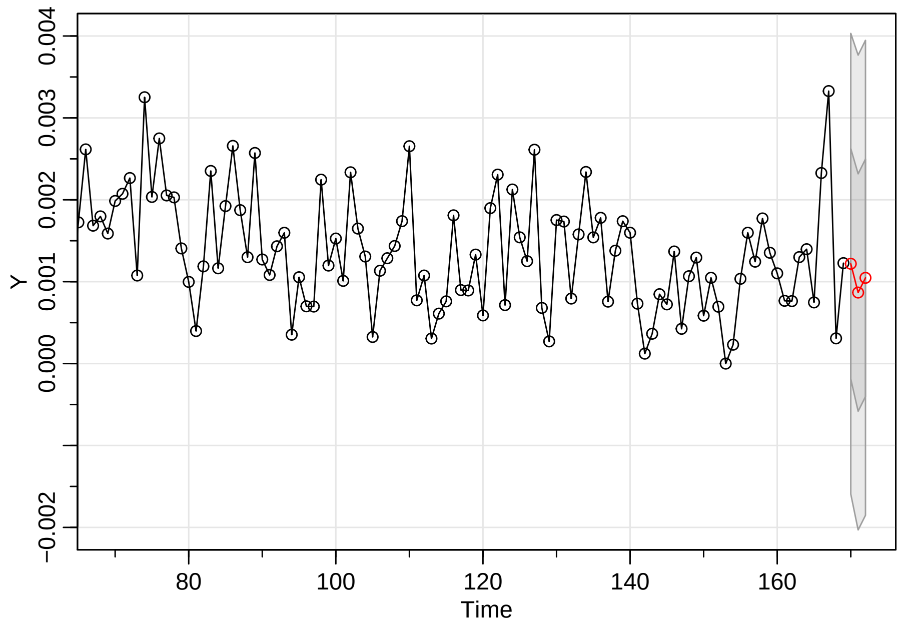

Y
−0.002
0.000
0.001
0.002
0.003
0.004
80
100
120
140
160
Time



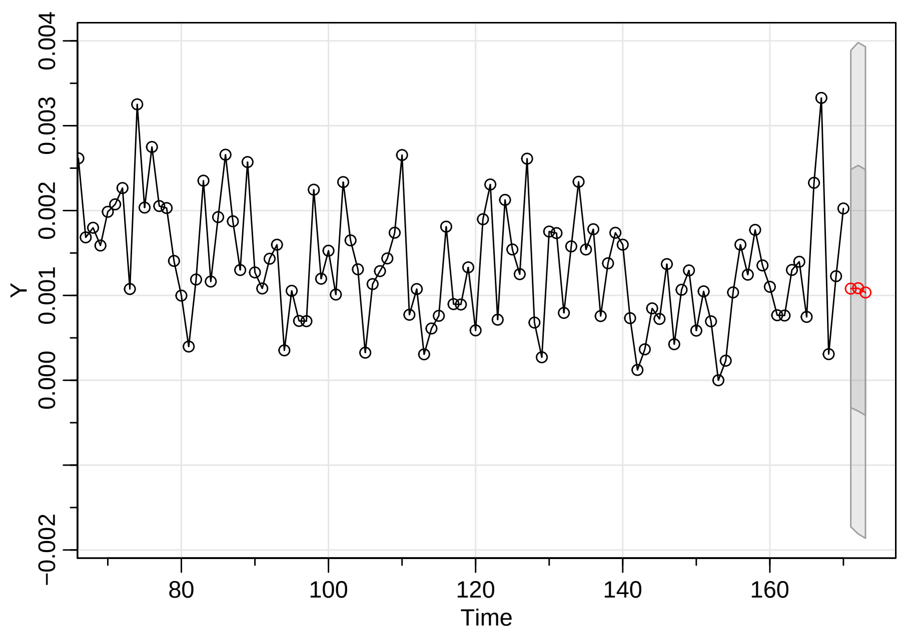
Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
80
100
120
140
160



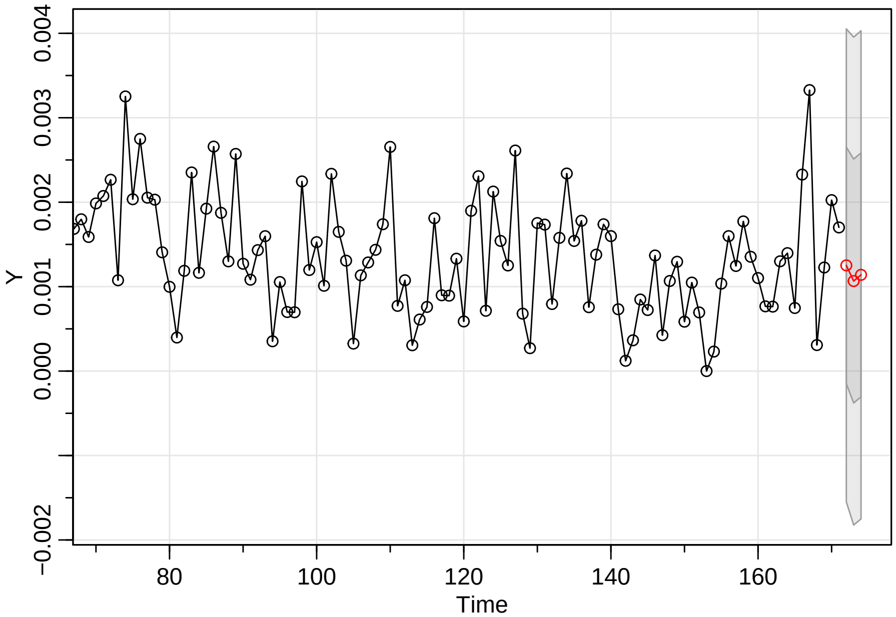
Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
80
100
120
140
160



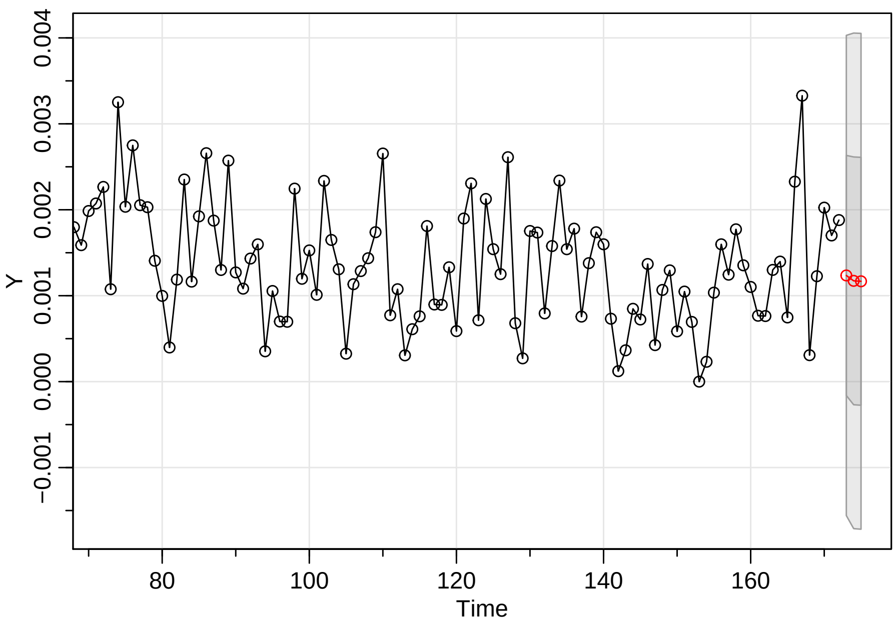
Y
0.004
0.003
0.002
0.001
0.000
−0.002
80
100
120
140
160
Time



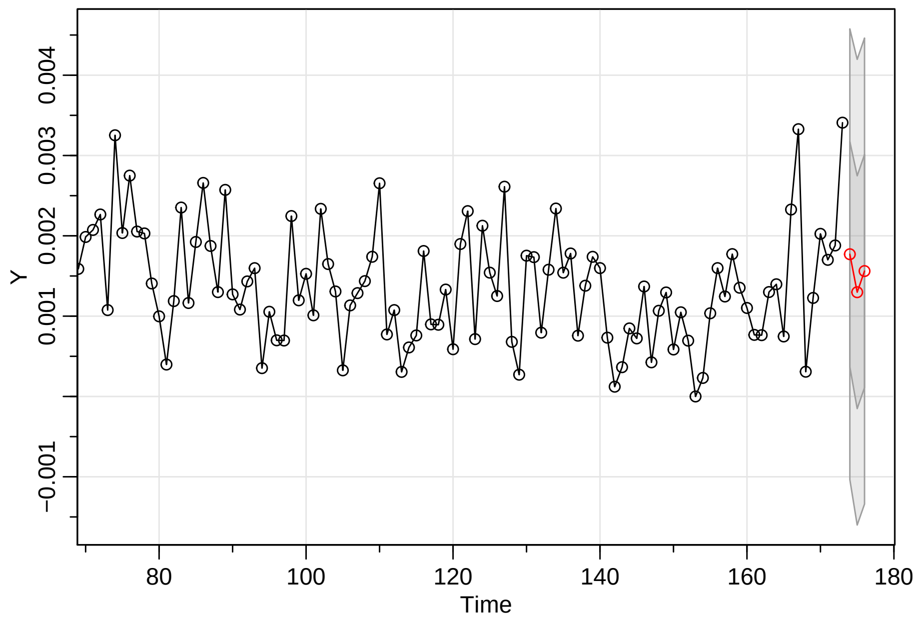
Y
0.004
0.003
0.002
0.001
0.000
−0.002
80
100
120
140
160
Time

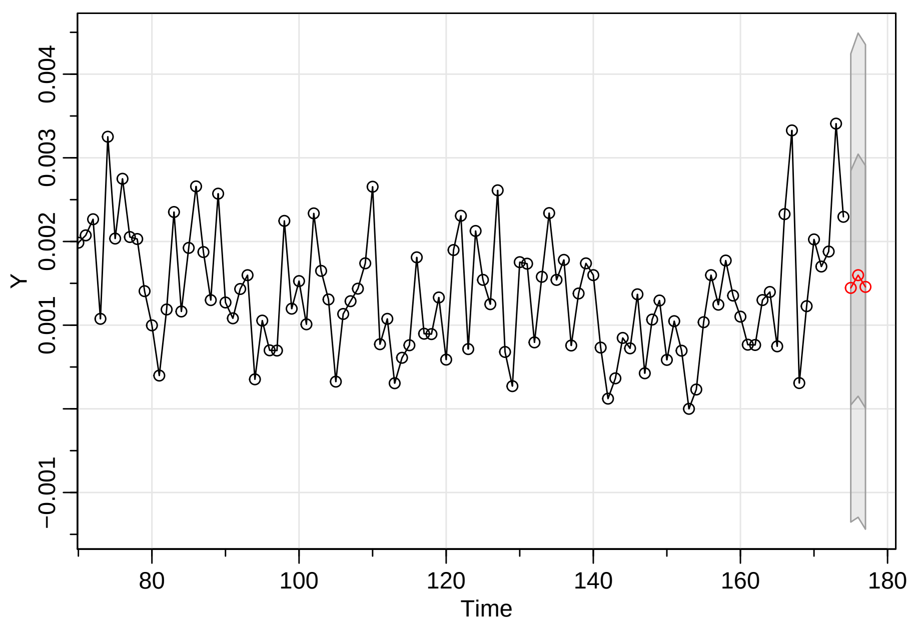


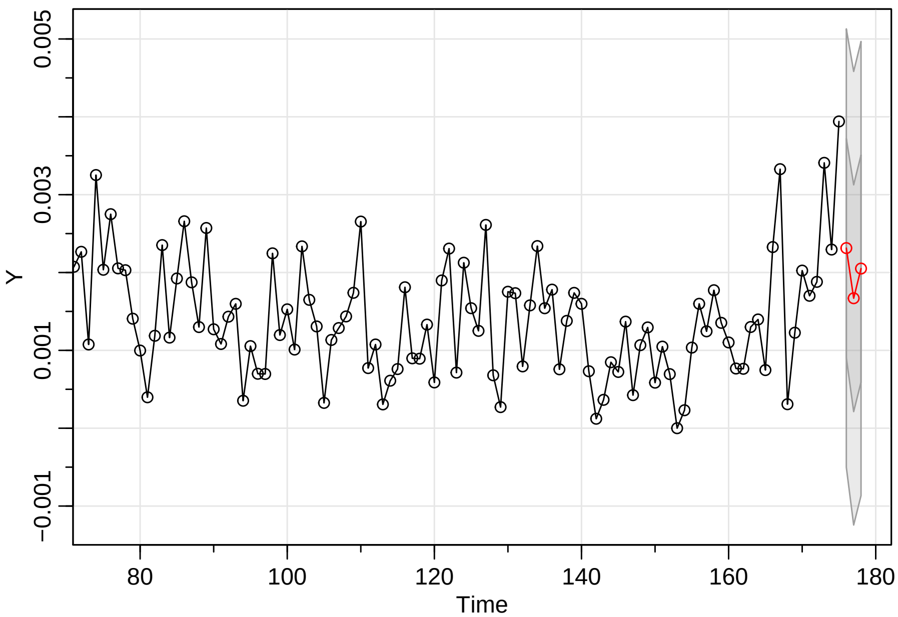
Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
80
100
120
140
160

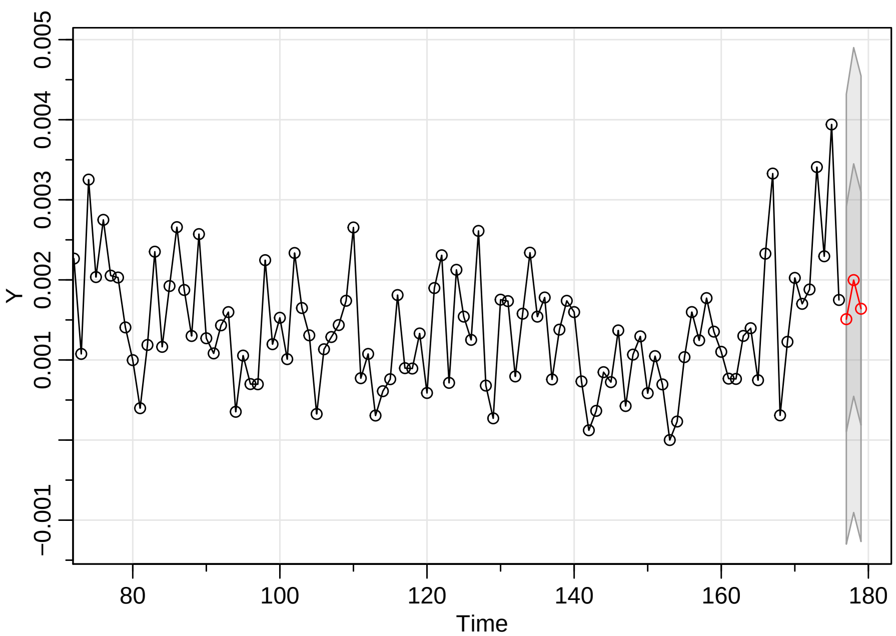


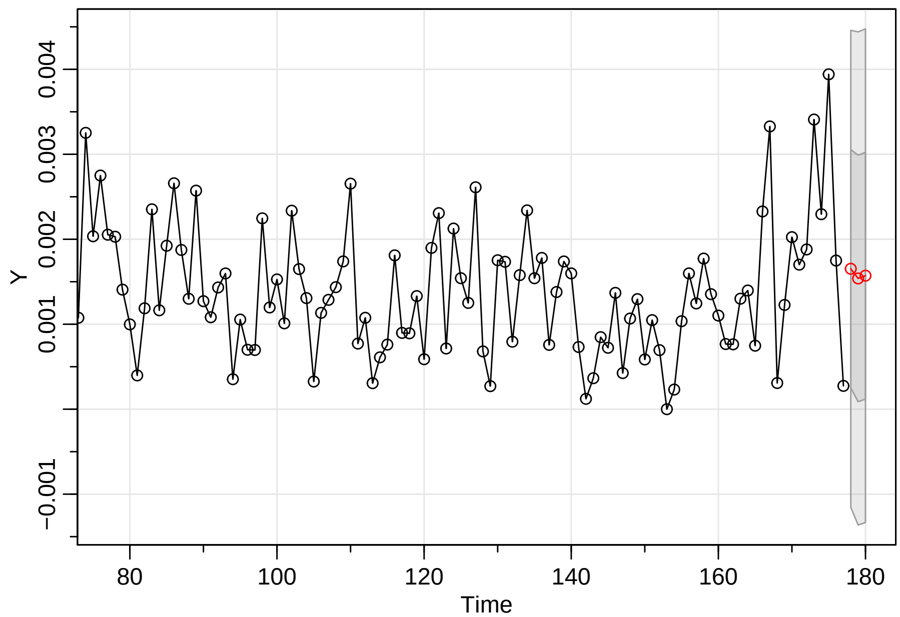
Y
Time
0.004
0.003
0.002
0.001
0.000
−0.001
80
100
120
140
160


Y
0.004
0.003
0.002
0.001
−0.001
80
100
120
140
160
180
Time

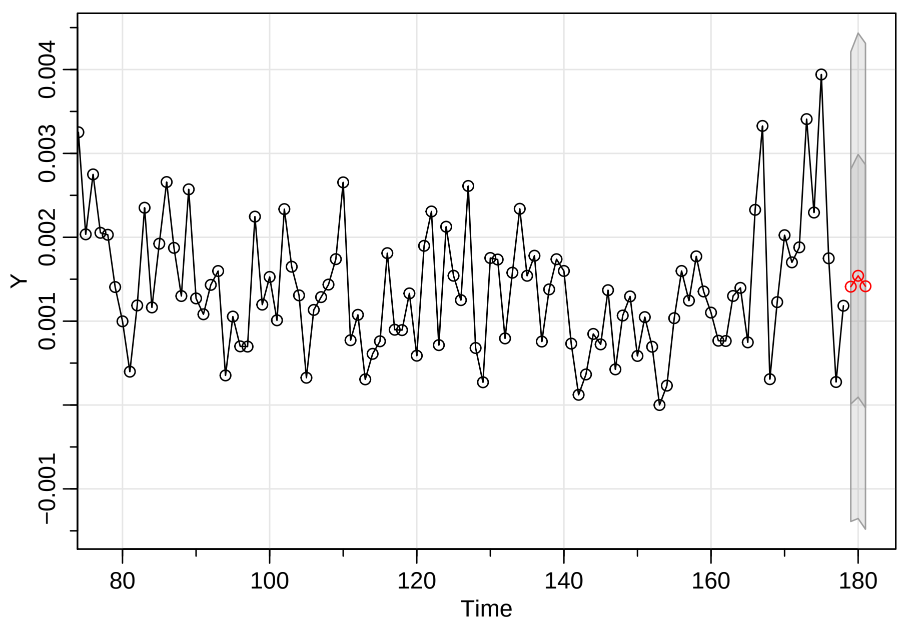

Y
0.004
0.003
0.002
0.001
−0.001
80
100
120
140
160
180
Time


Y
0.005
0.003
0.001
−0.001
80
100
120
140
160
180
Time

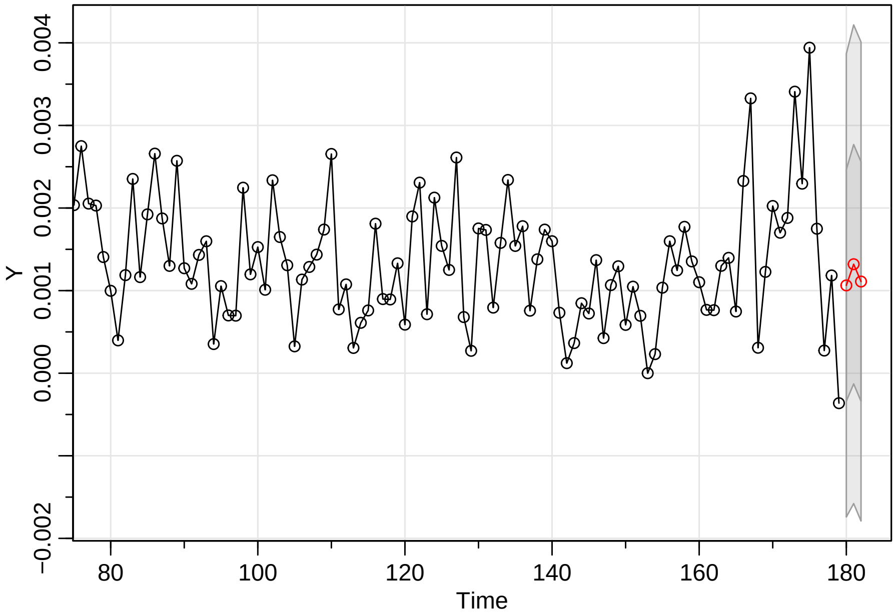

Y
0.005
0.004
0.003
0.002
0.001
−0.001
80
100
120
140
160
180
Time



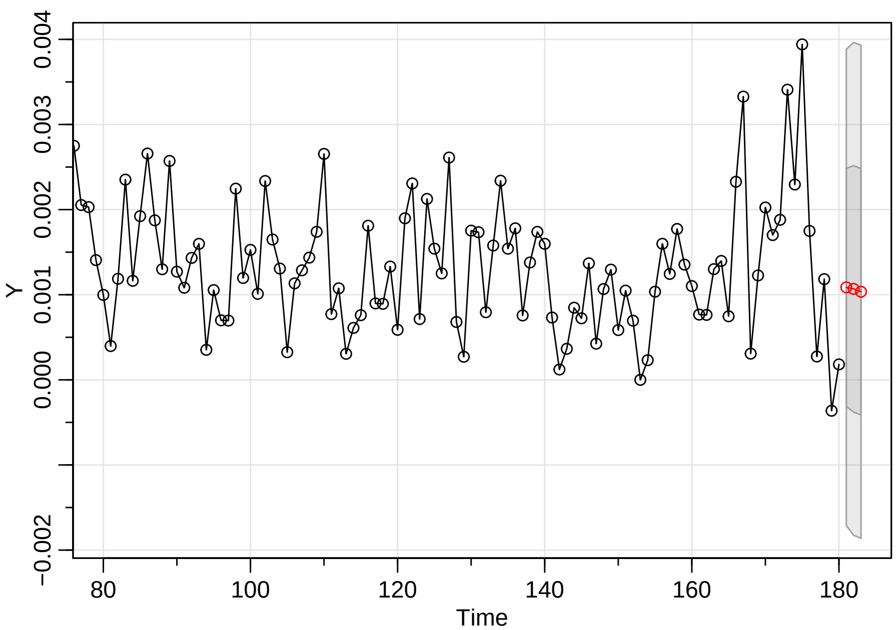
Y
0.004
0.003
0.002
0.001
−0.001
80
100
120
140
160
180
Time



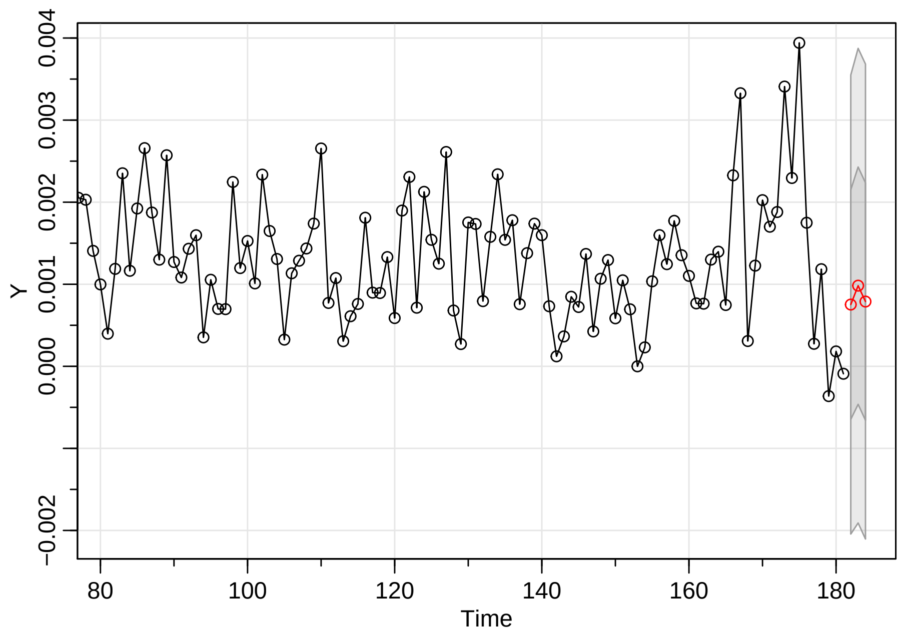
Y
0.004
0.003
0.002
0.001
−0.001
80
100
120
140
160
180
Time

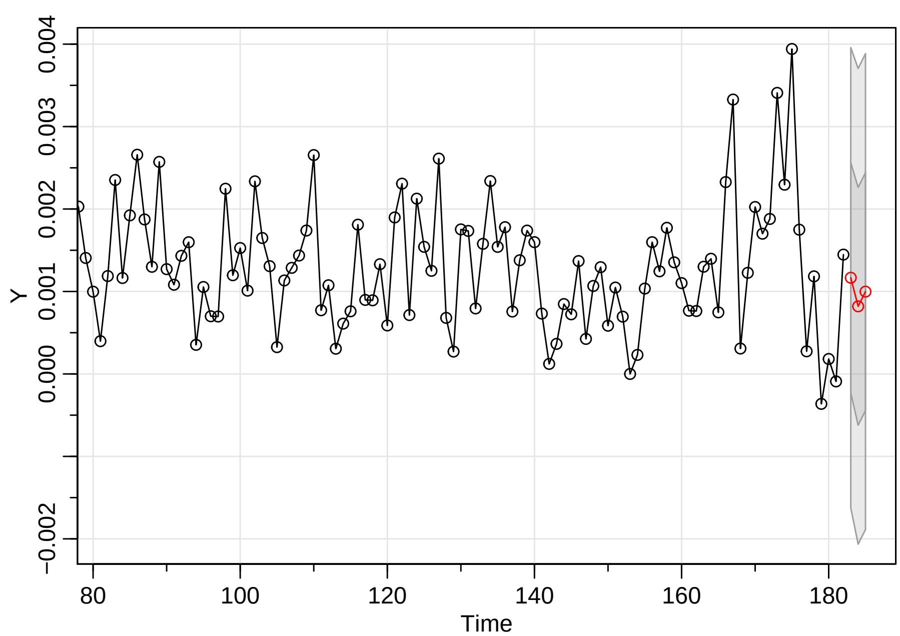


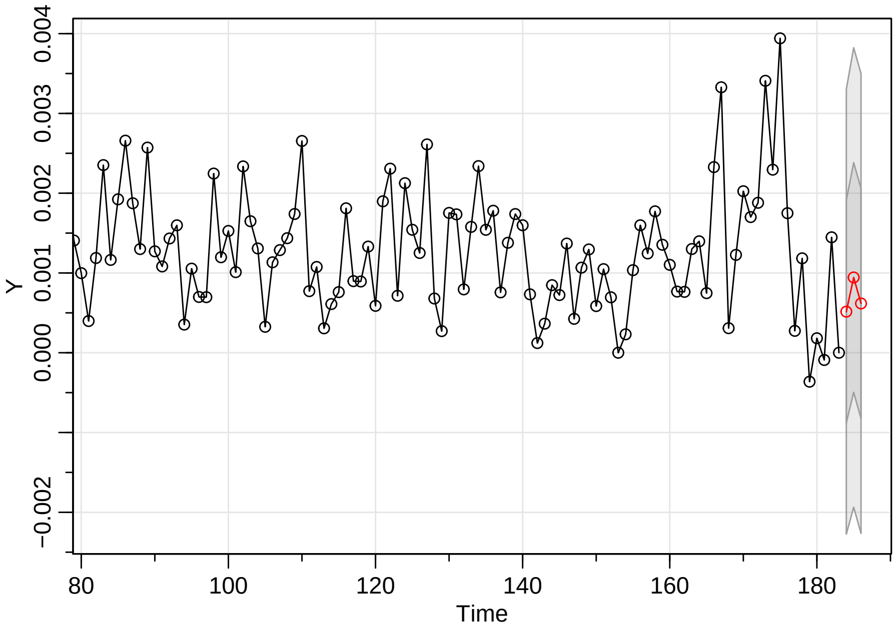
Y
0.004
0.003
0.002
0.001
0.000
−0.002
80
100
120
140
160
180
Time

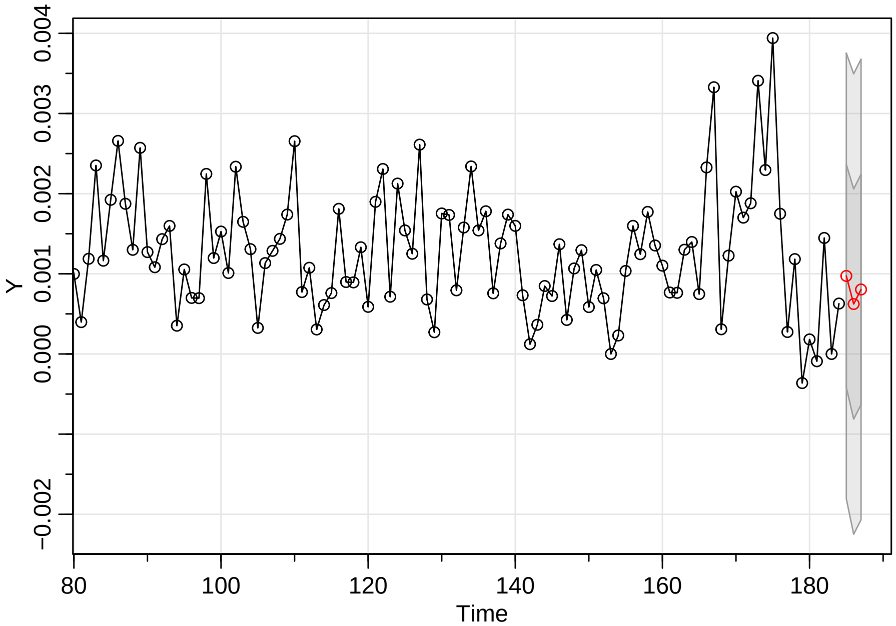


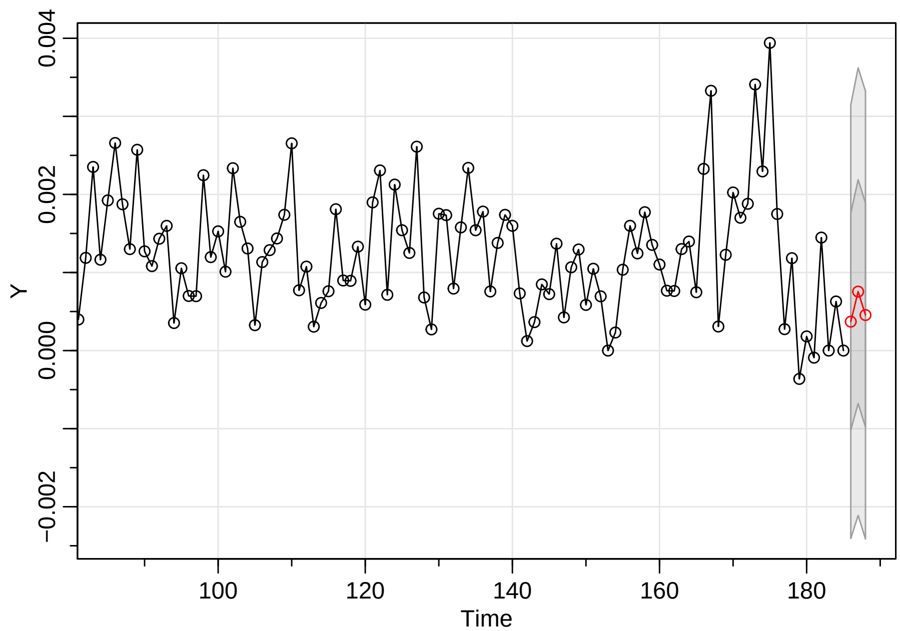
Y
0.004
0.003
0.002
0.001
0.000
−0.002
80
100
120
140
160
180
Time


Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
80
100
120
140
160
180

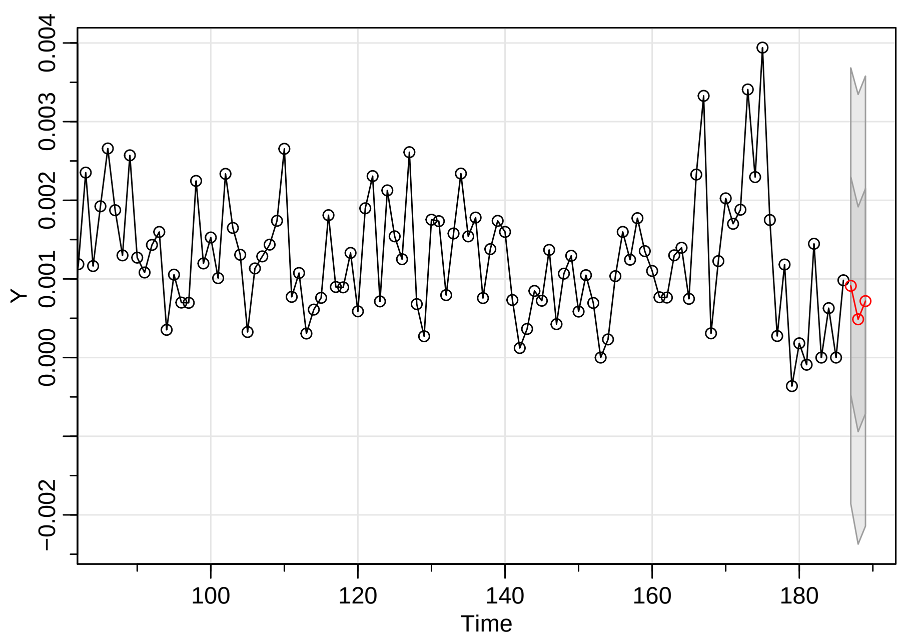

Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
80
100
120
140
160
180


Y
0.004
0.003
0.002
0.001
0.000
−0.002
80
100
120
140
160
180
Time

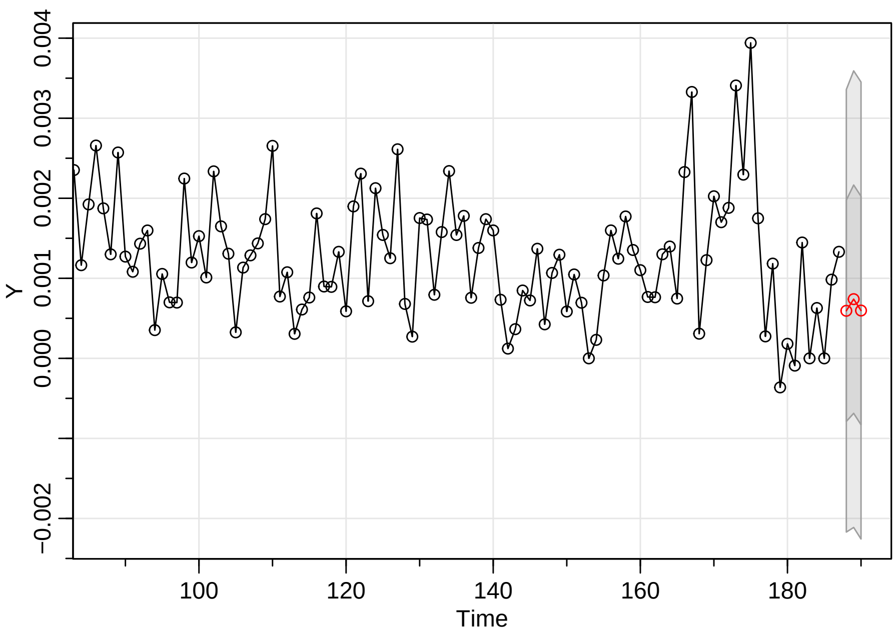

Y
0.004
0.003
0.002
0.001
0.000
−0.002
80
100
120
140
160
180
Time

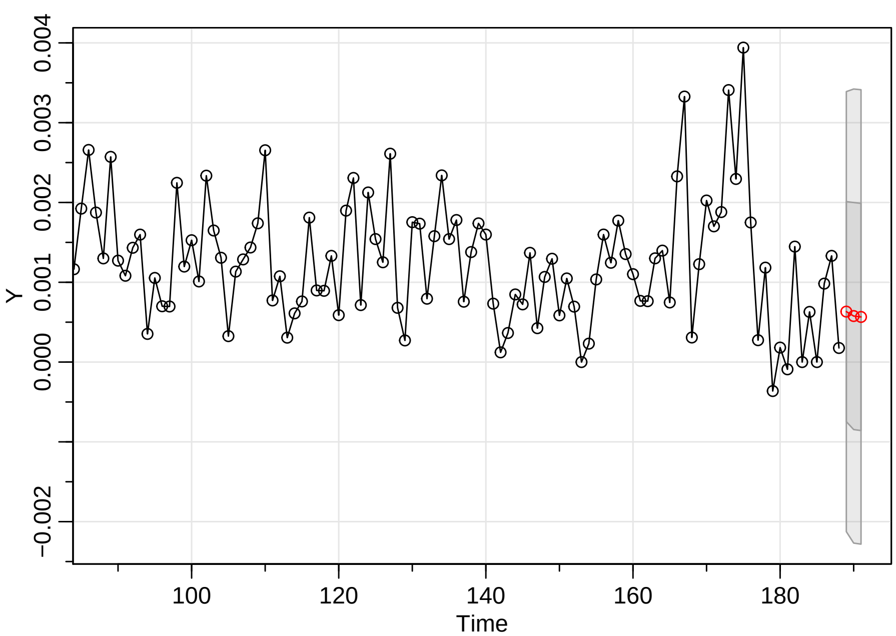


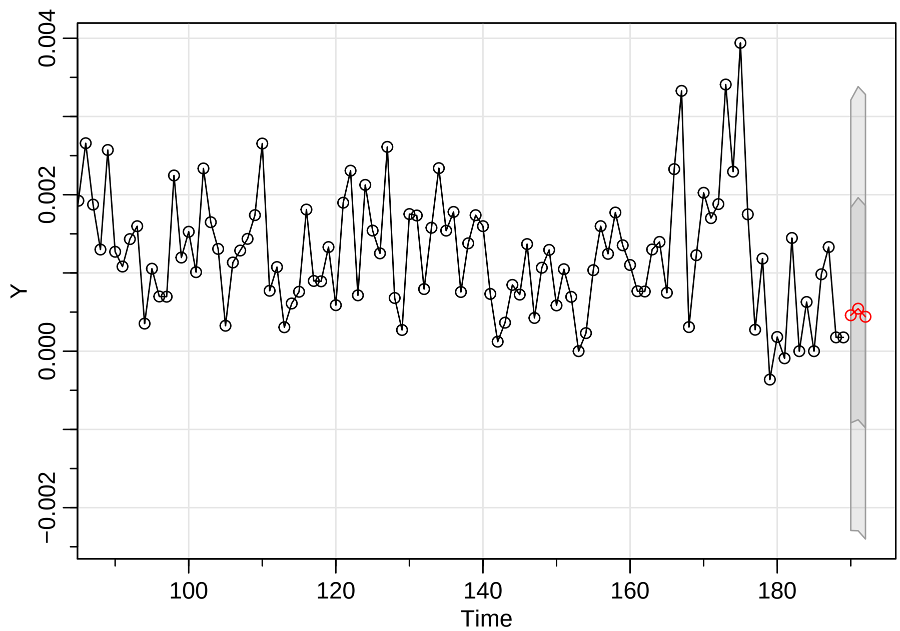
Y
Time
0.004
0.002
0.000
−0.002
100
120
140
160
180

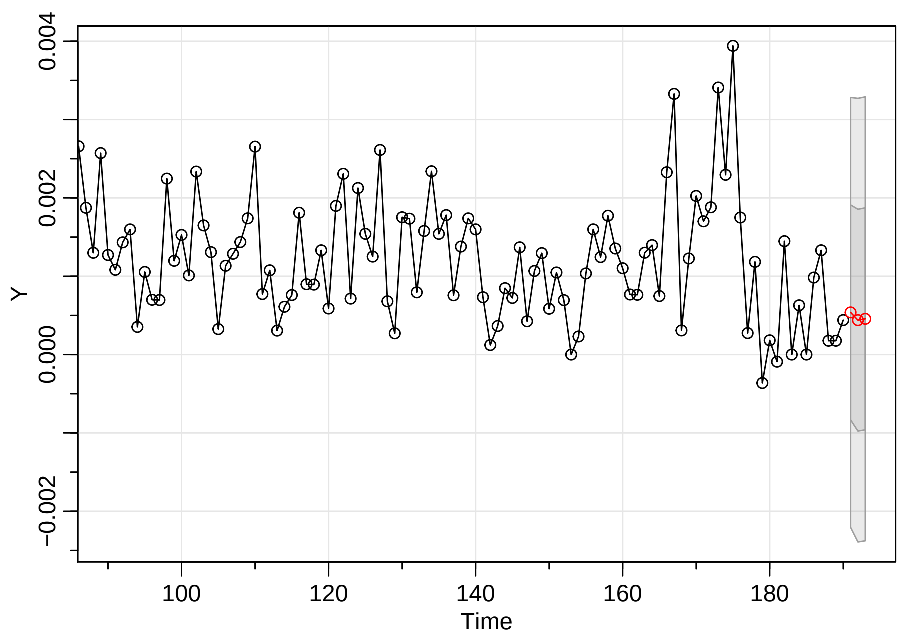


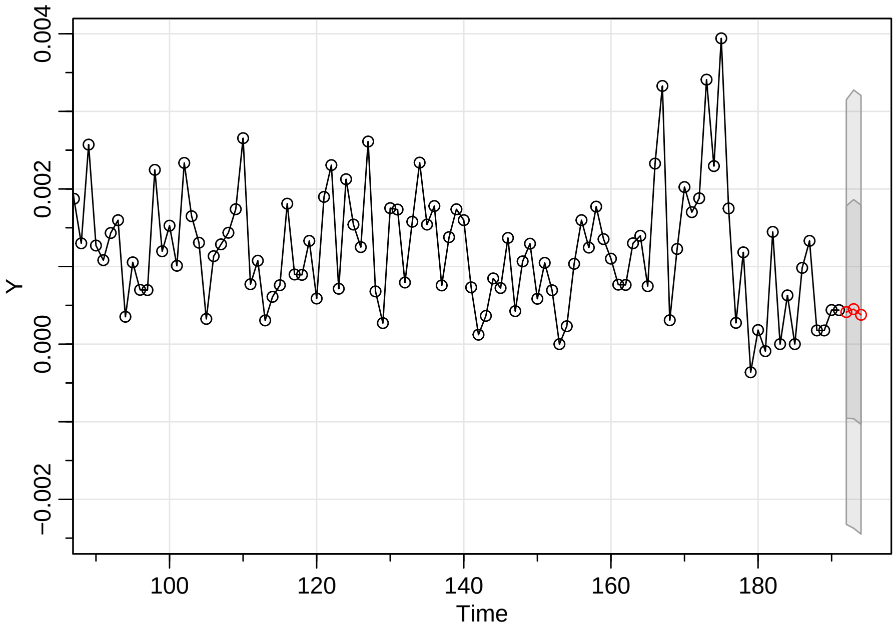
Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180

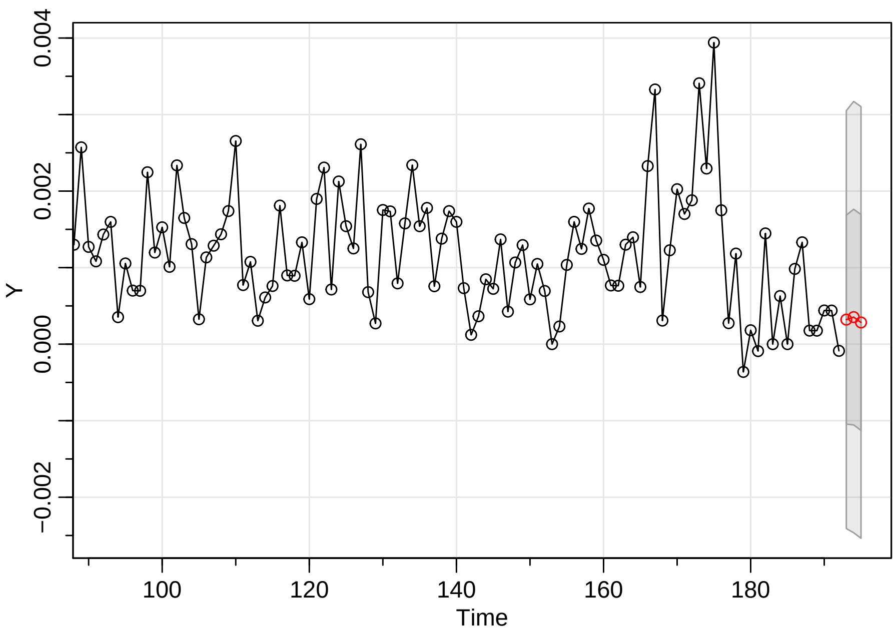

Y
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180
Time

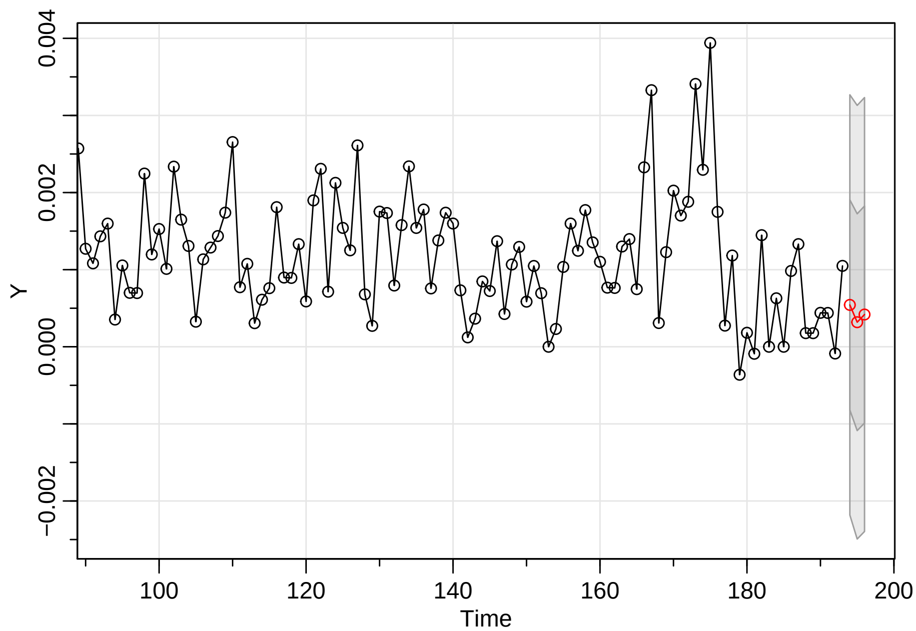

Y
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180
Time


Y
Time
0.004
0.002
0.000
−0.002
100
120
140
160
180

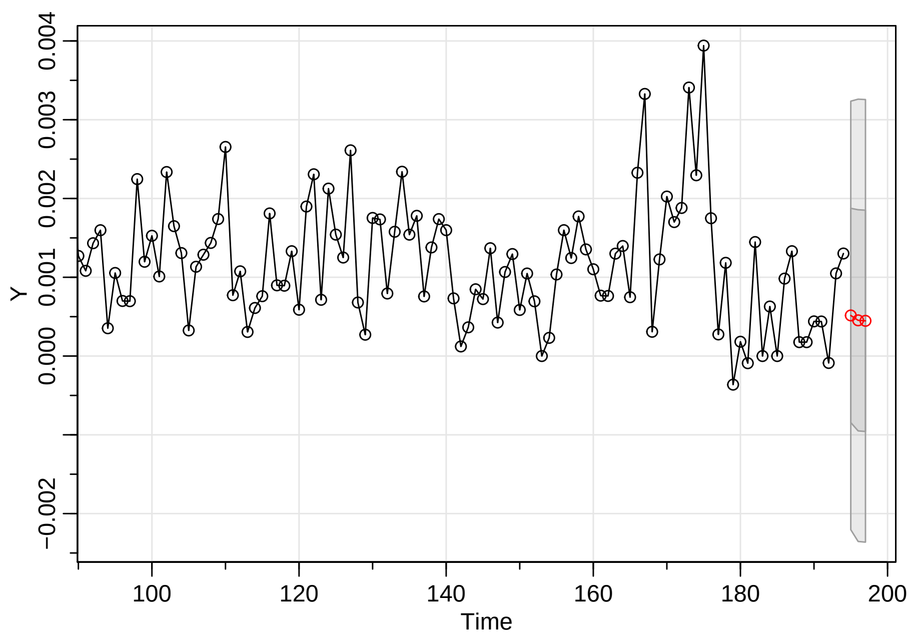

Y
Time
0.004
0.002
0.000
−0.002
100
120
140
160
180


Y
Time
0.004
0.002
0.000
−0.002
100
120
140
160
180

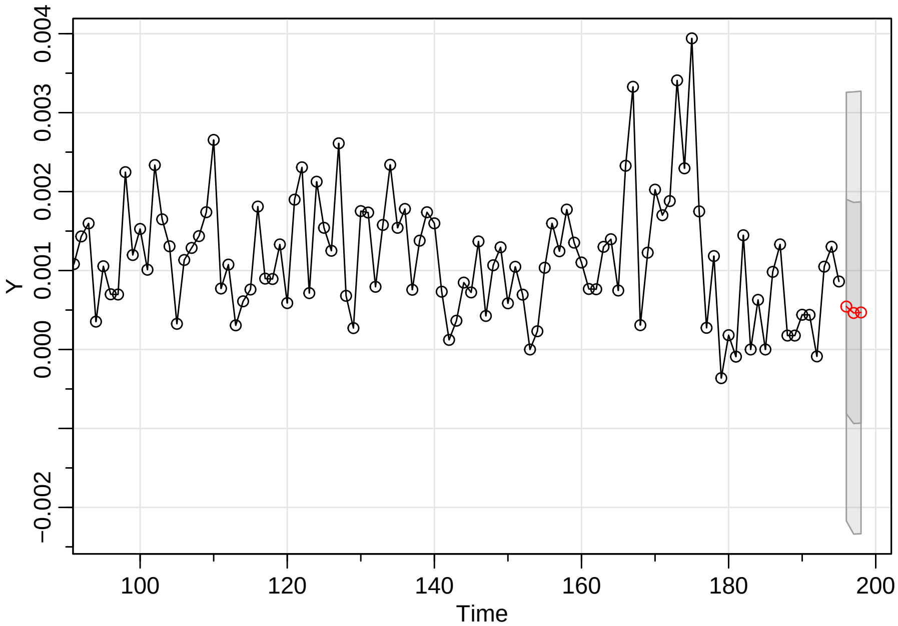

Y
Time
0.004
0.002
0.000
−0.002
100
120
140
160
180

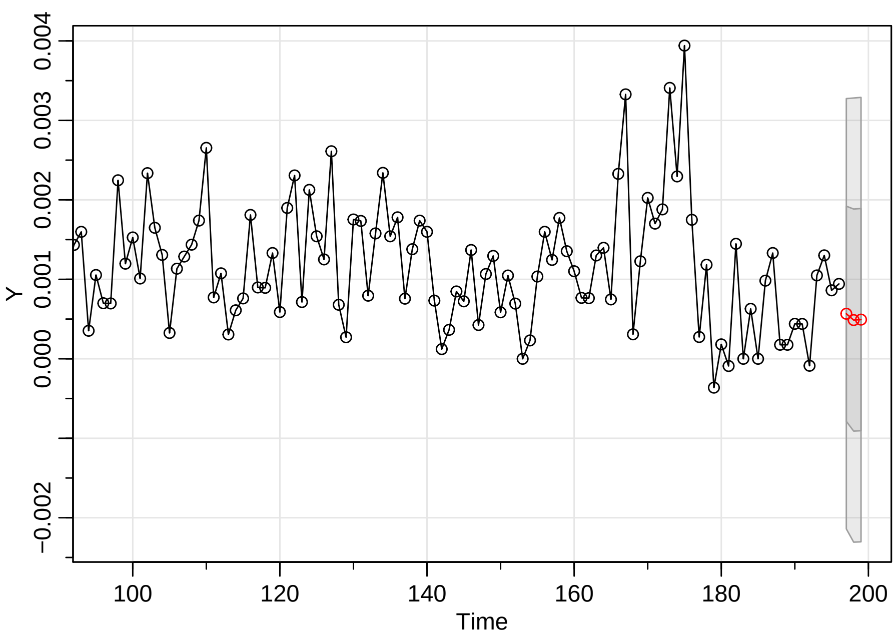


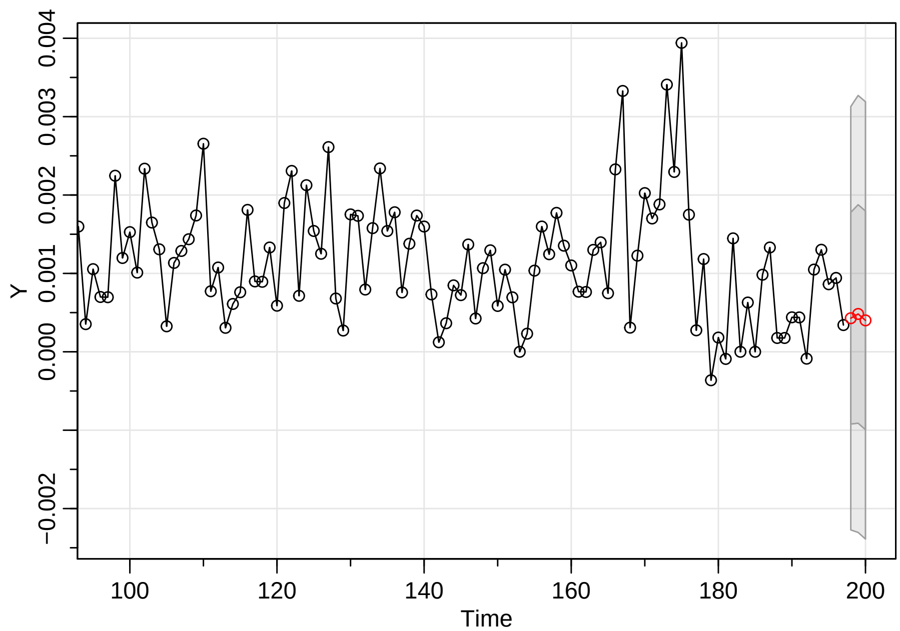
Y
0.004
0.002
0.000
−0.002
100
120
140
160
180
200
Time

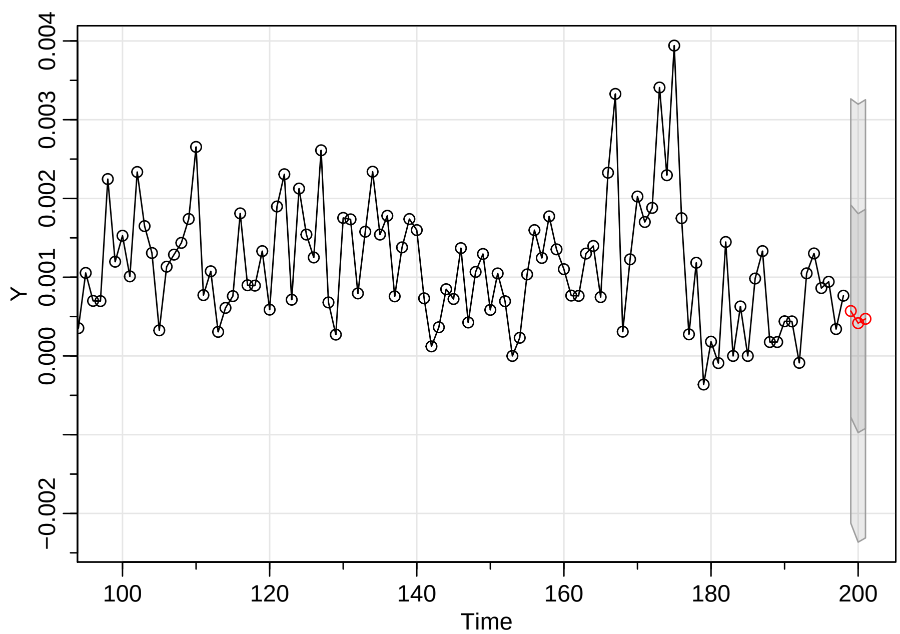


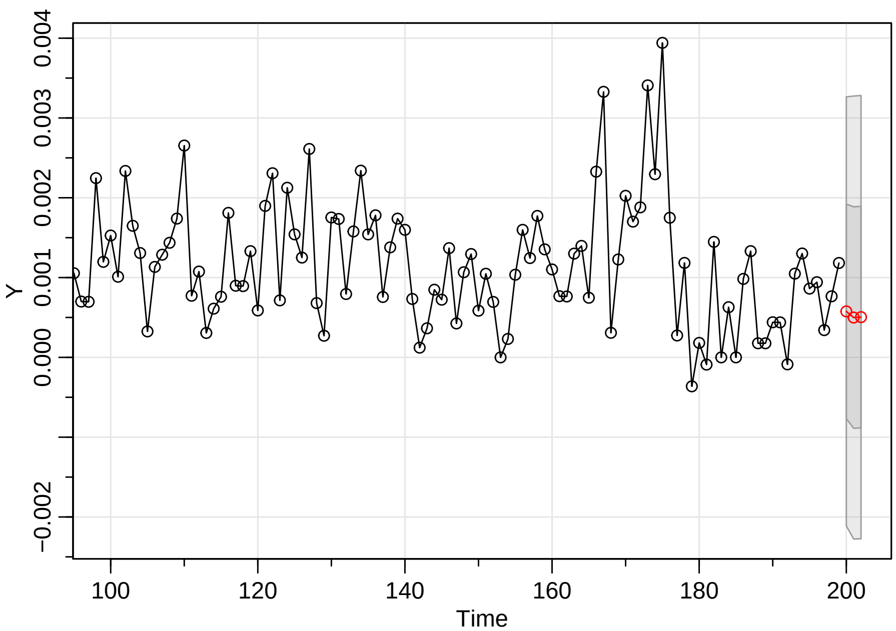
Y
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180
200
Time

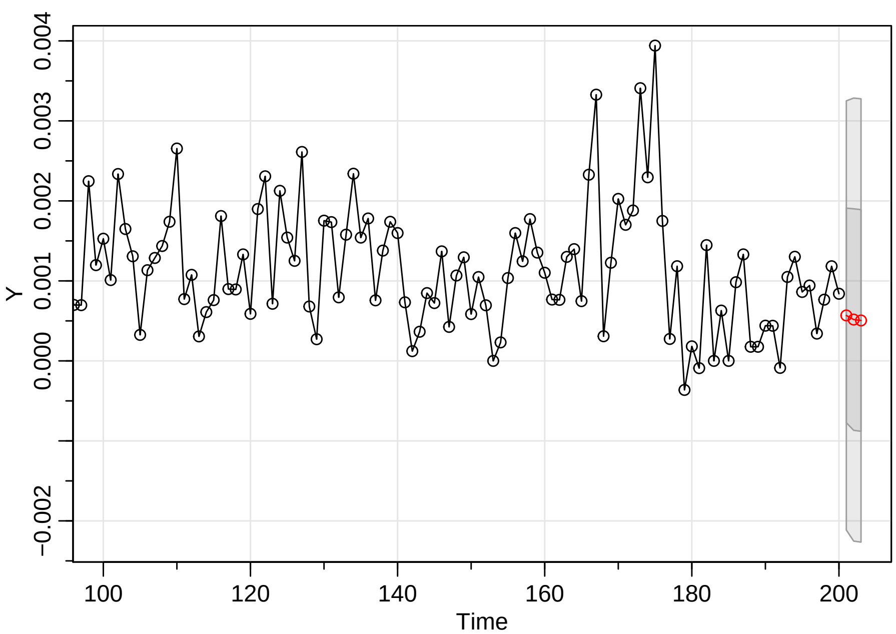

Y
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180
200
Time

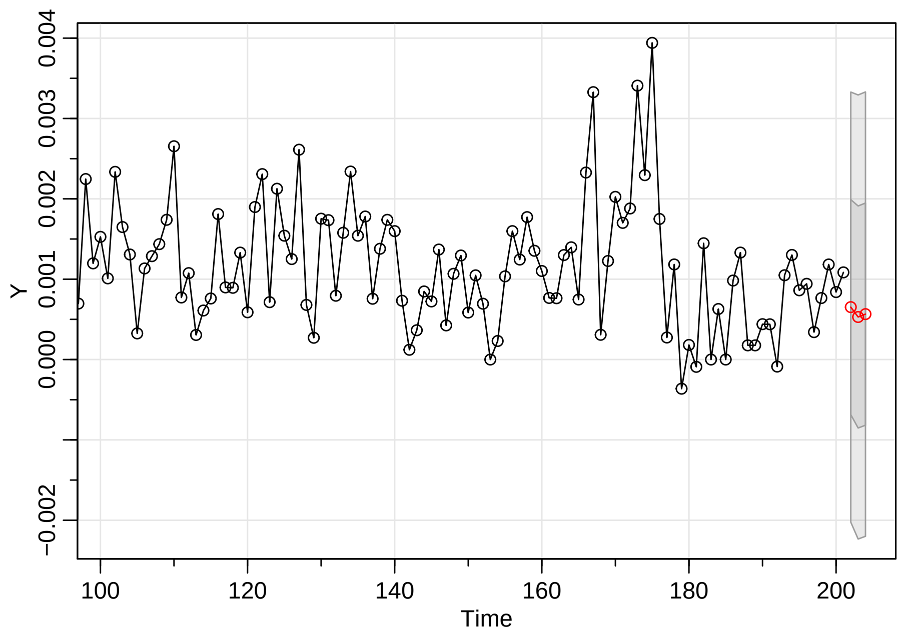

Y
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180
200
Time

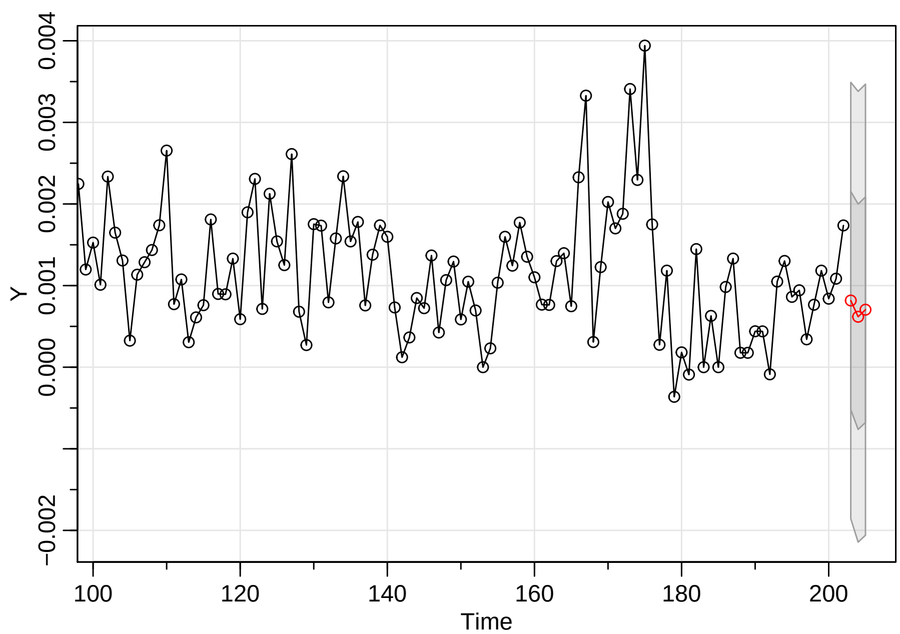

Y
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180
200
Time

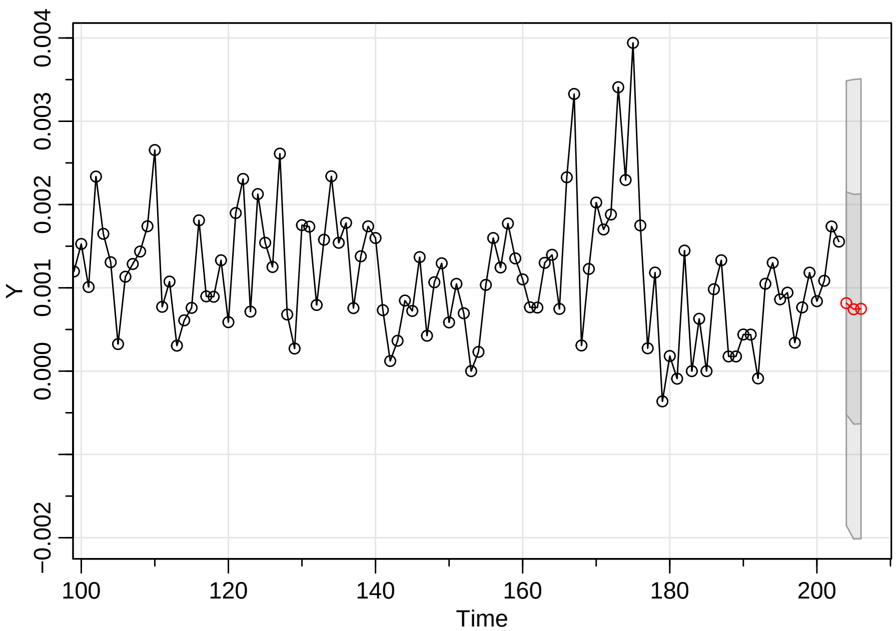

Y
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180
200
Time

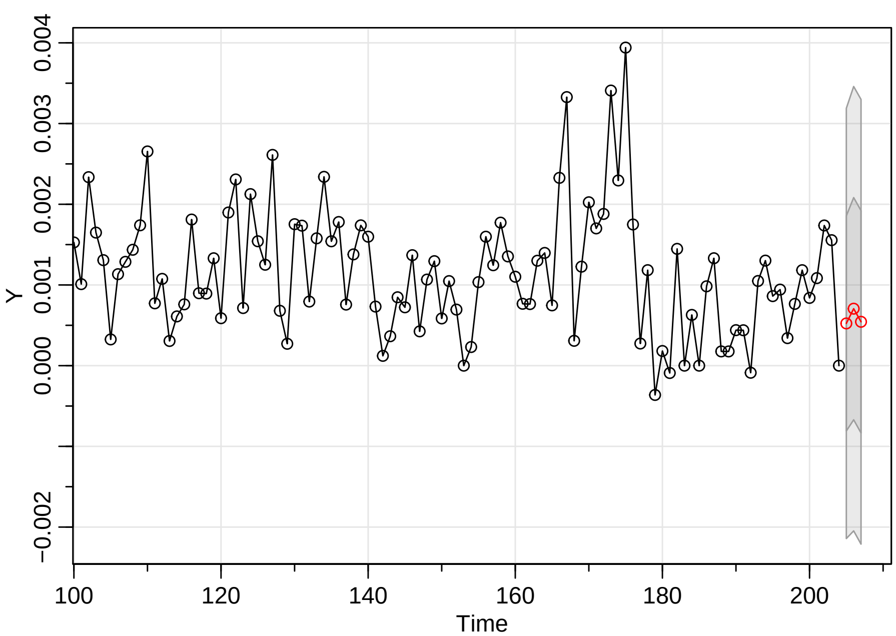

Y
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180
200
Time

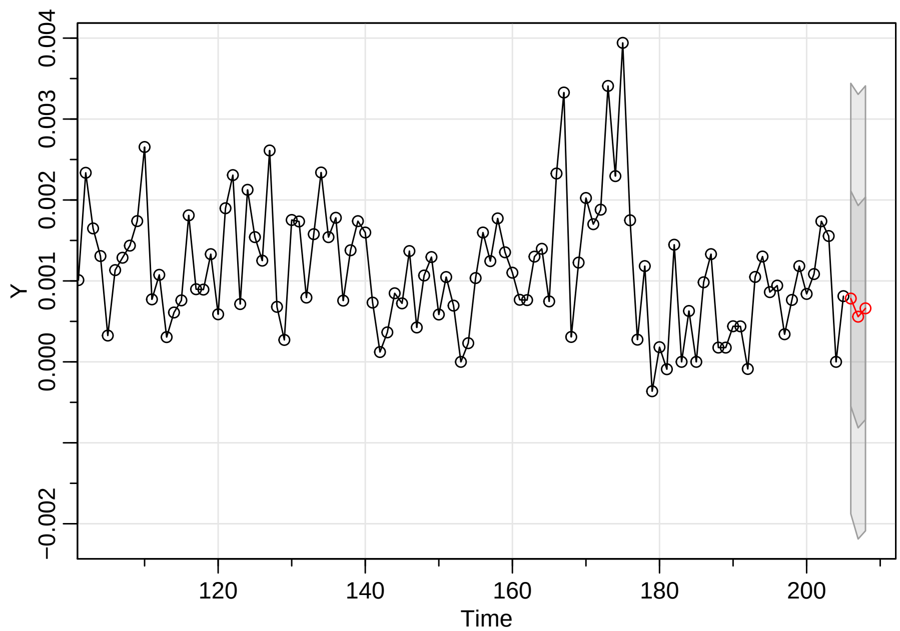

Y
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180
200
Time

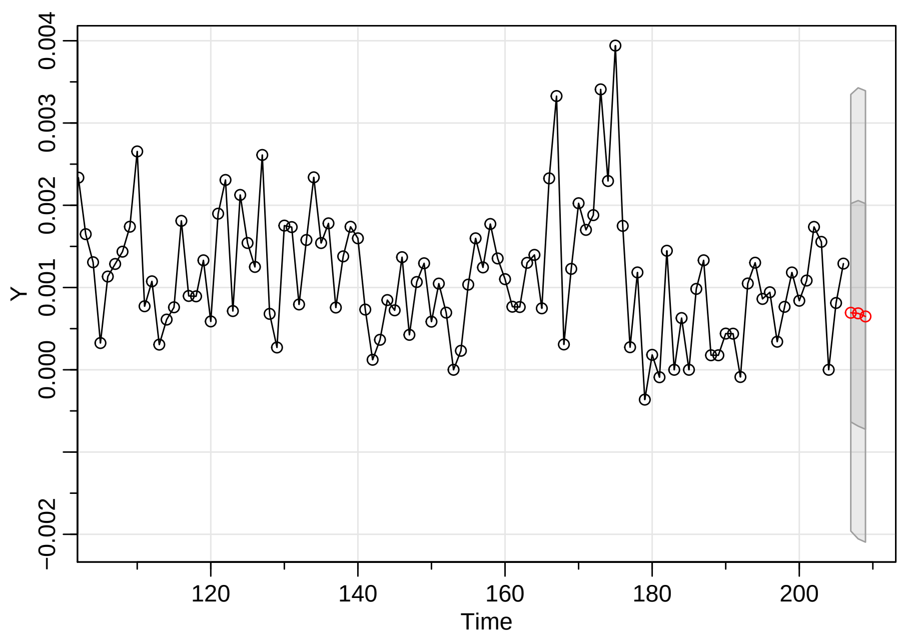

Y
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180
200
Time

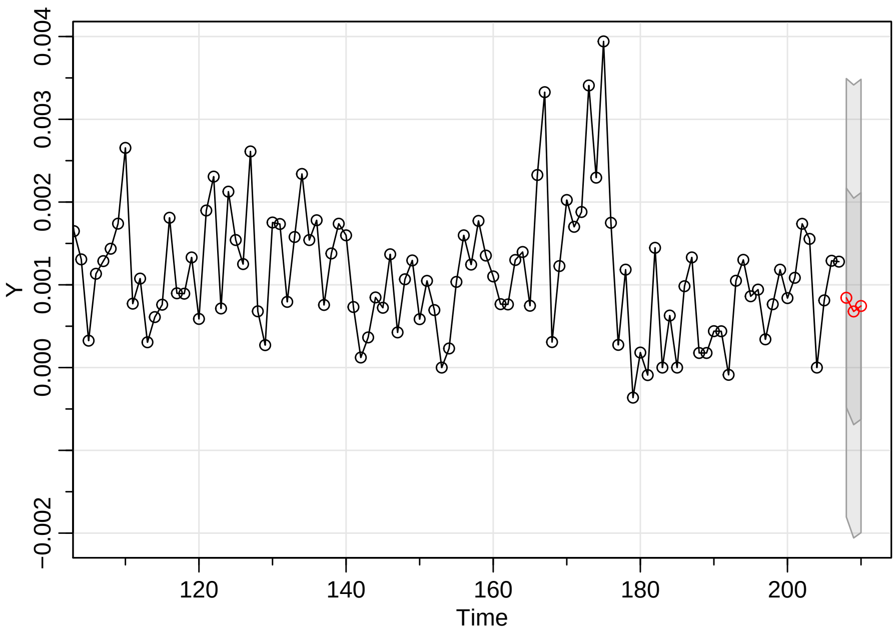

Y
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180
200
Time


Y
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180
200
Time

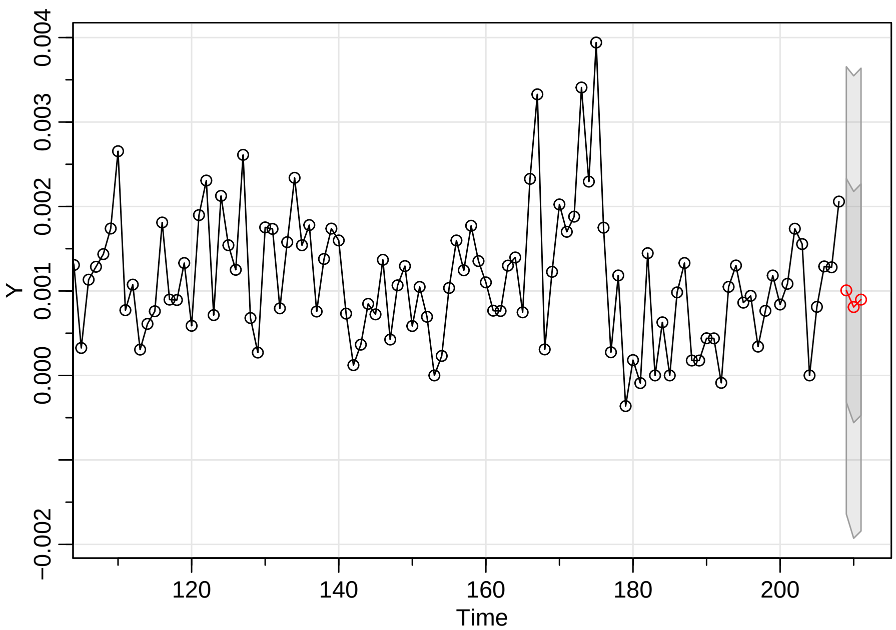

Y
0.004
0.003
0.002
0.001
0.000
−0.002
100
120
140
160
180
200
Time


Y
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
Time

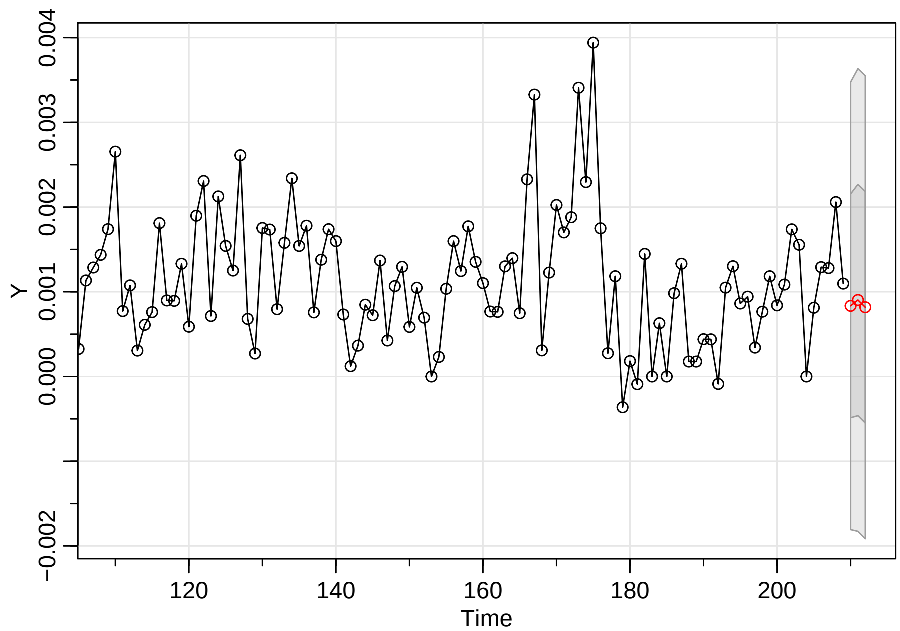

Y
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
Time

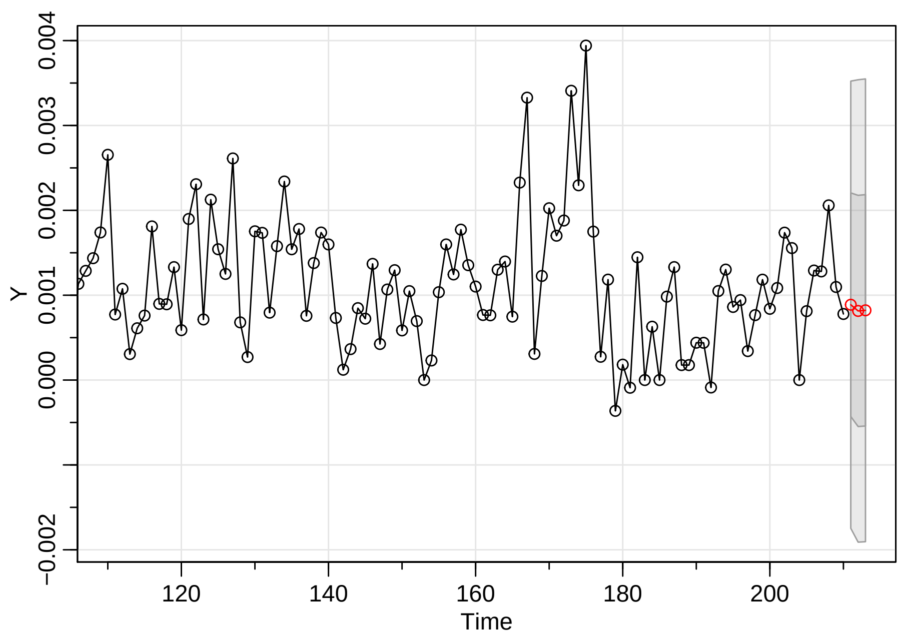


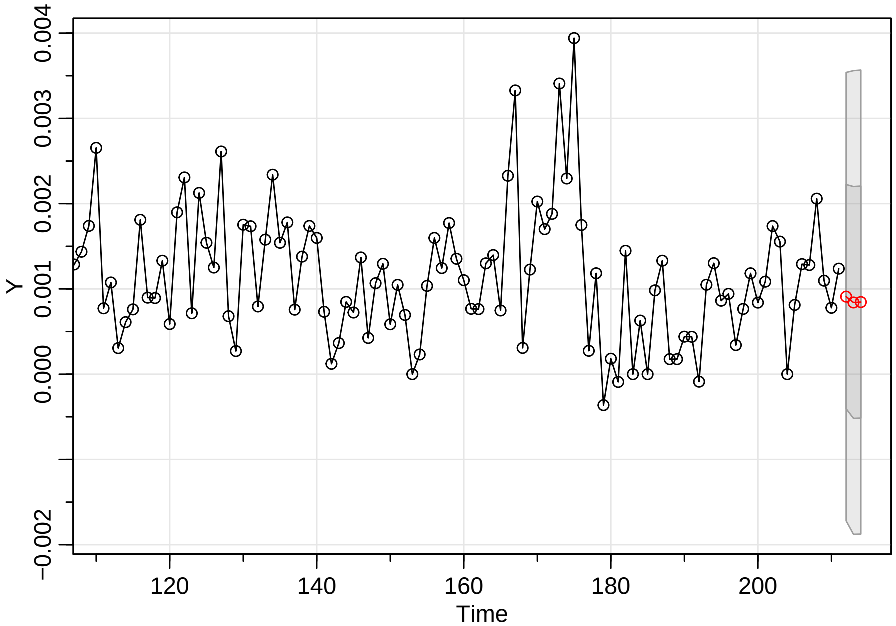
Y
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
Time



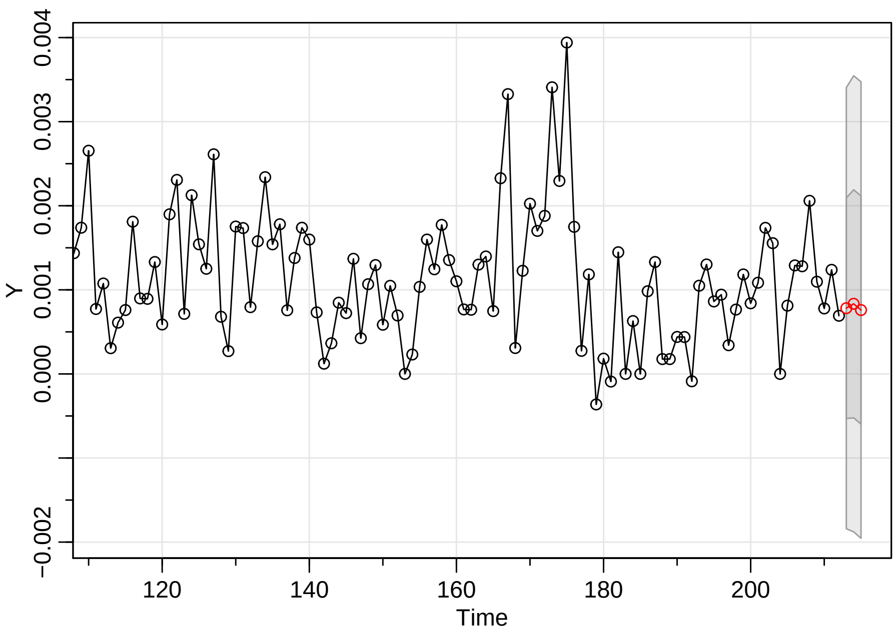
Y
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
Time


Y
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
Time

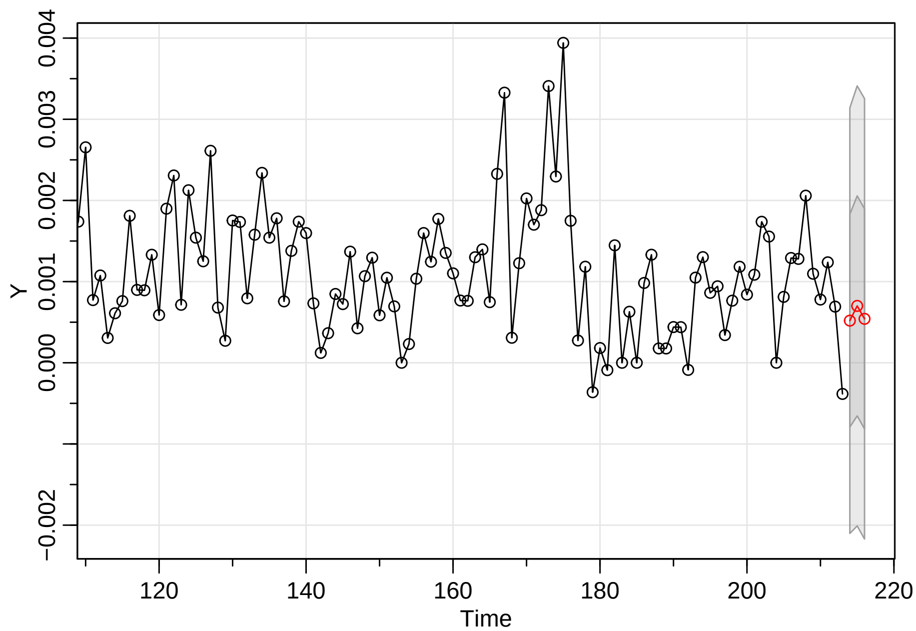

Y
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
Time

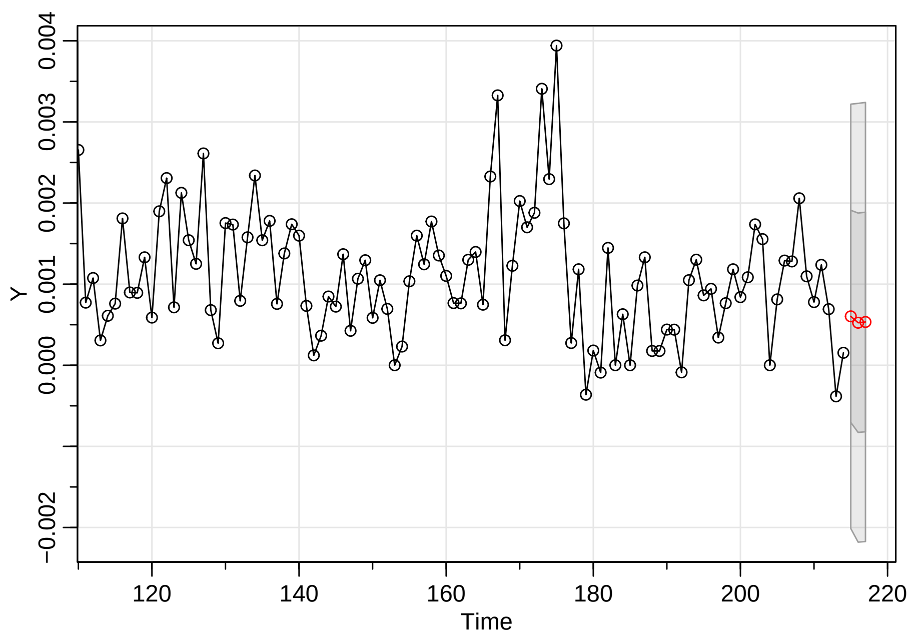


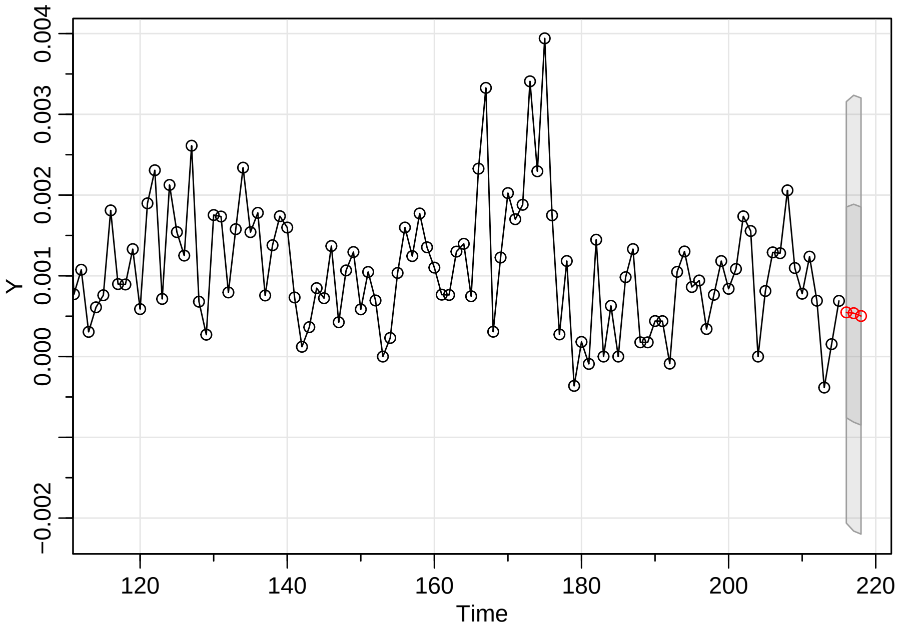
Y
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
Time



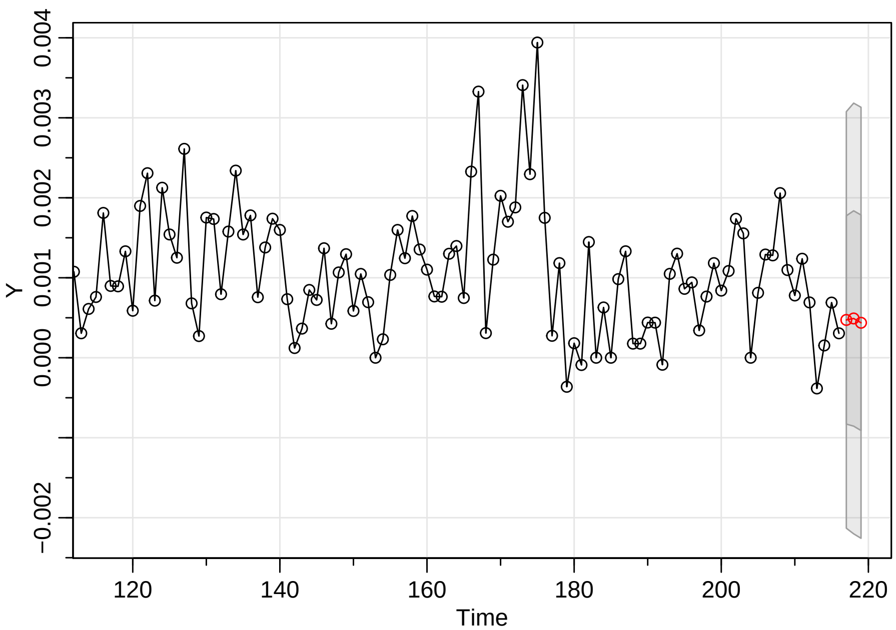
Y
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
Time


Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
220

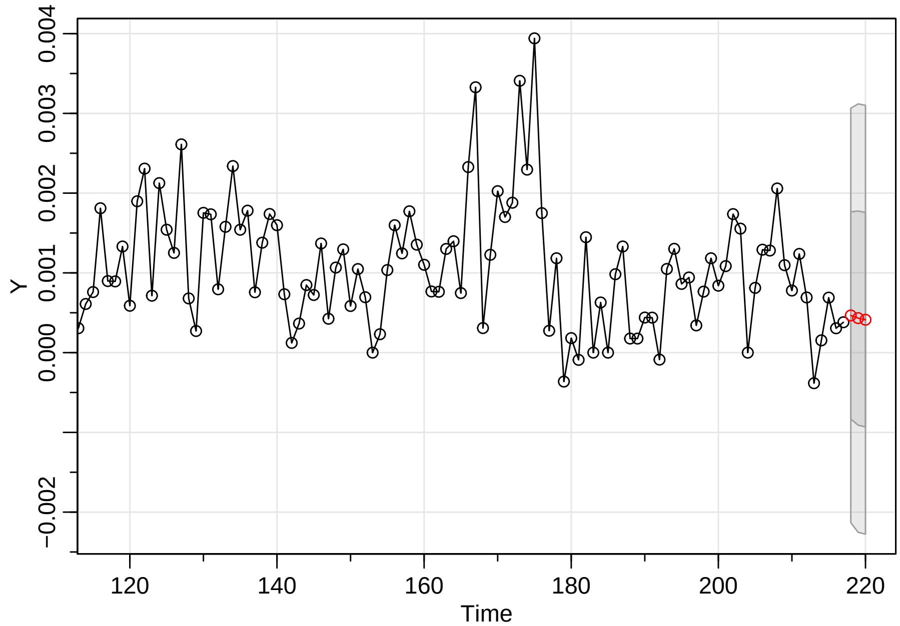

Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
220

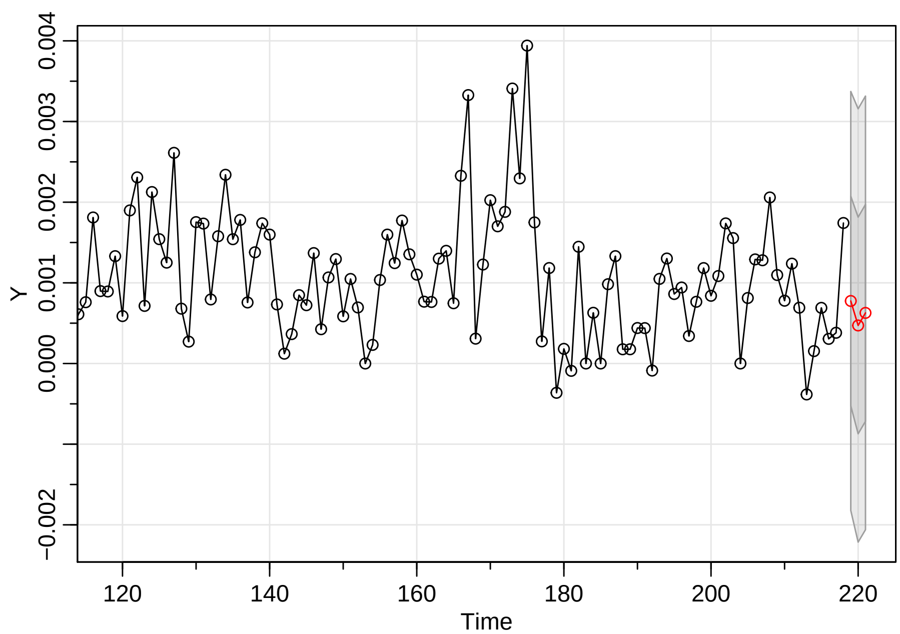


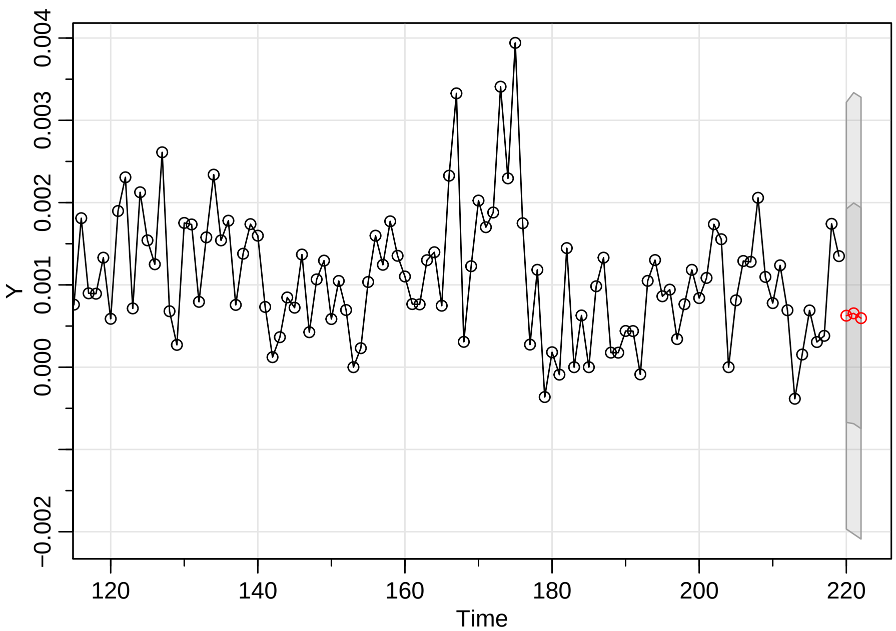
Y
−0.002
0.000
0.001
0.002
0.003
0.004
120
140
160
180
200
220
Time



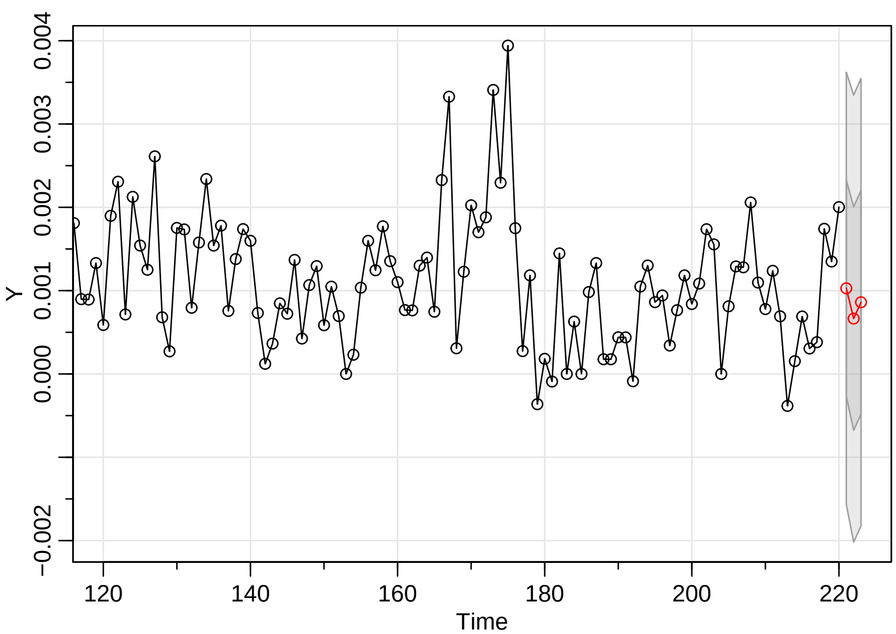
Y
−0.002
0.000
0.001
0.002
0.003
0.004
120
140
160
180
200
220
Time


Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
220

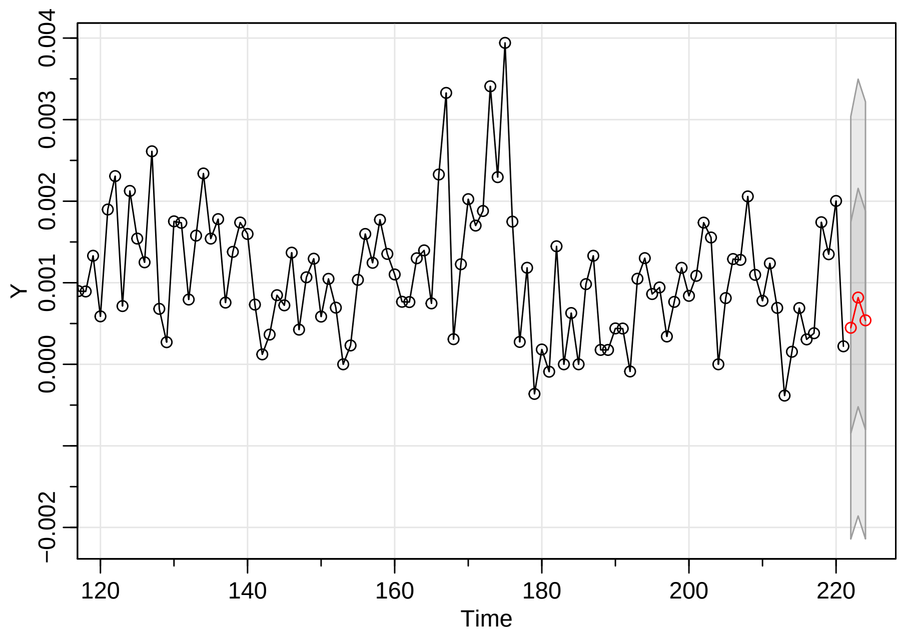

Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
220



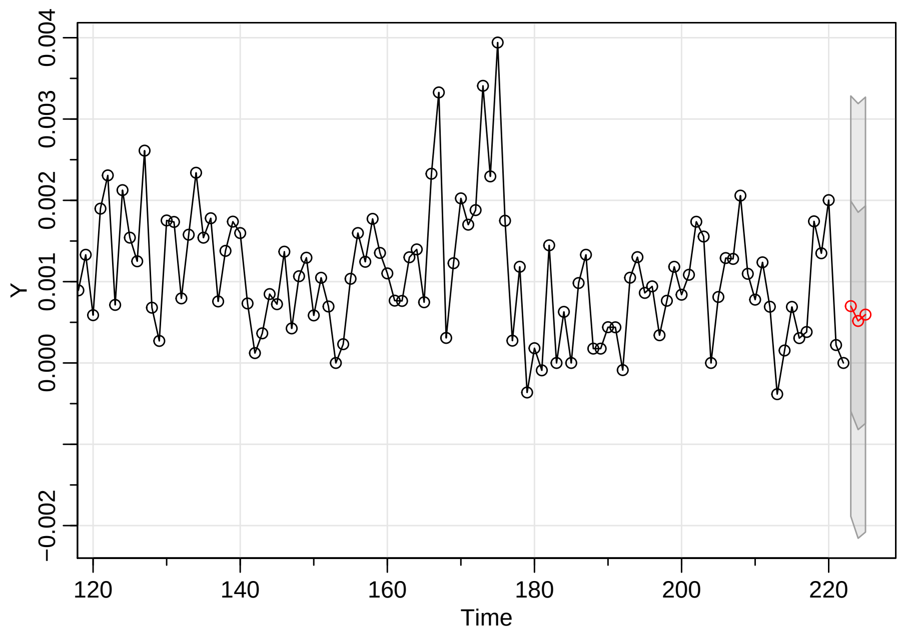
Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
220



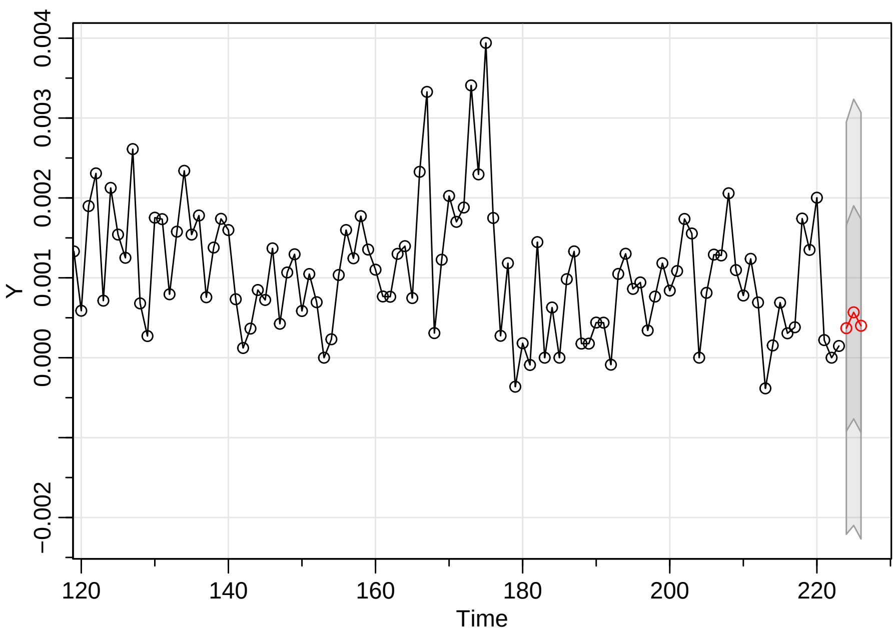
Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
220



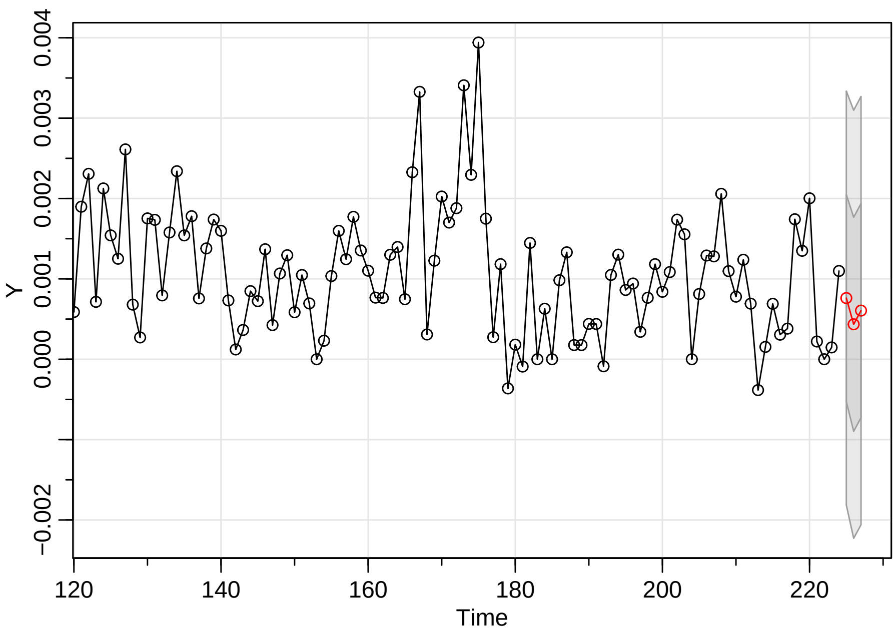
Y
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
220
Time



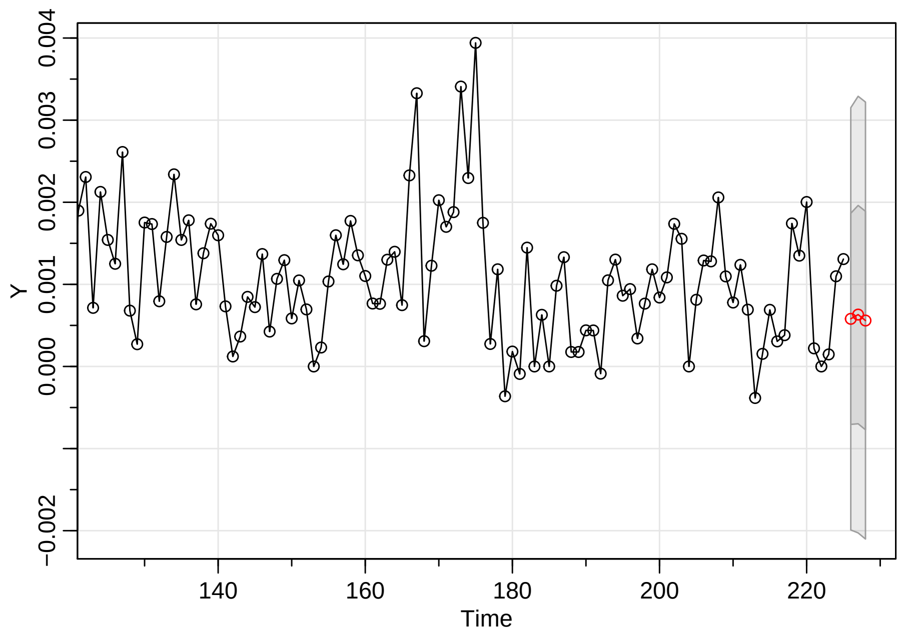
Y
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
220
Time

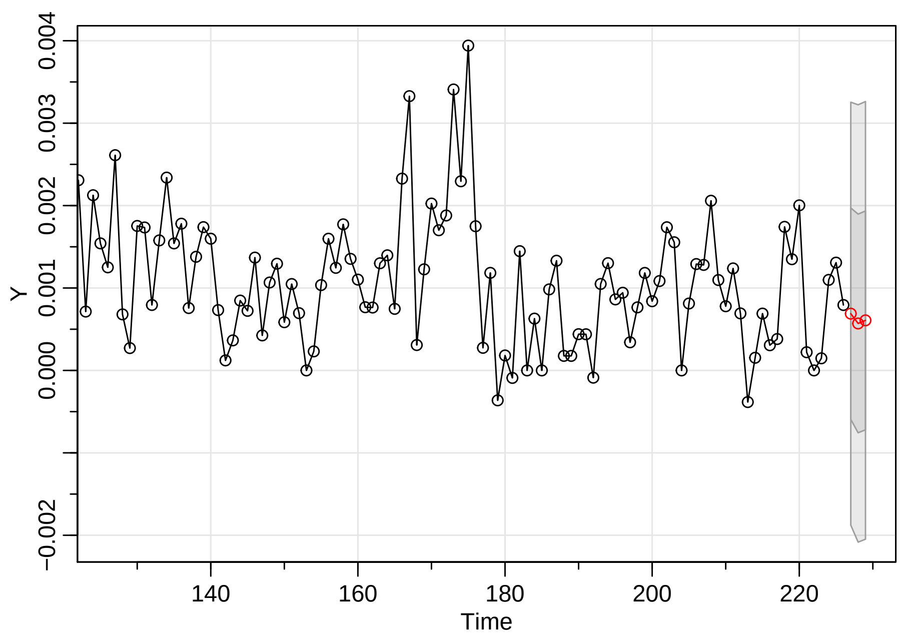


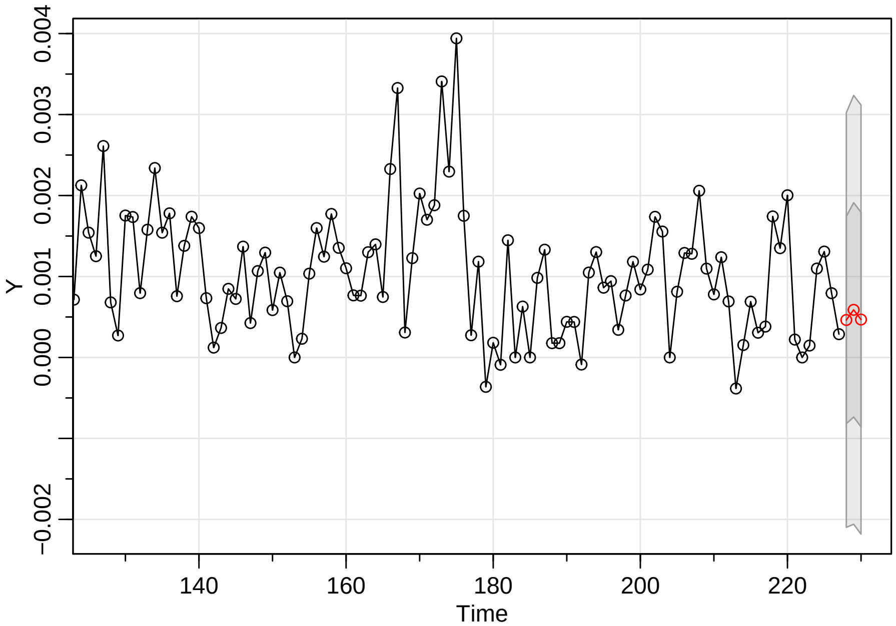
Y
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
220
Time

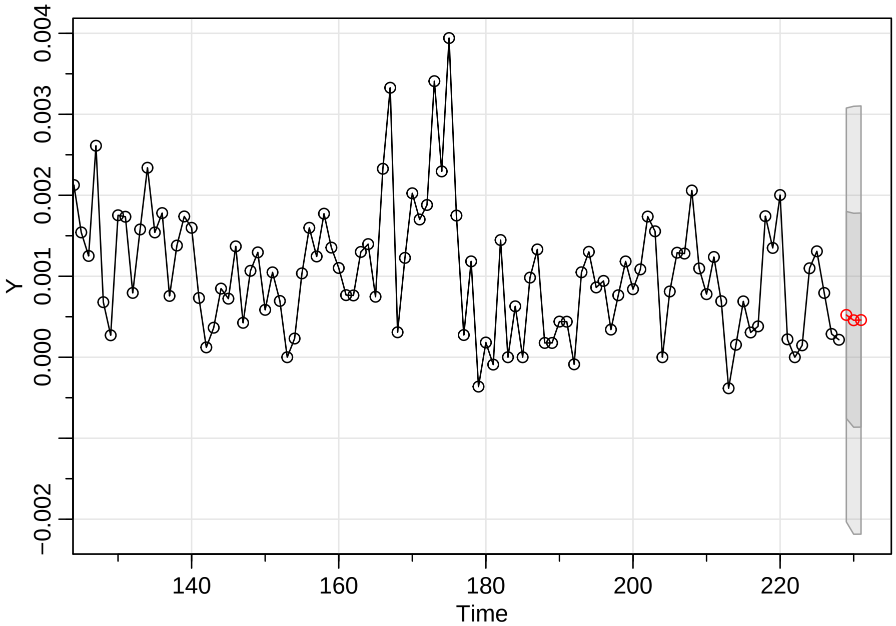


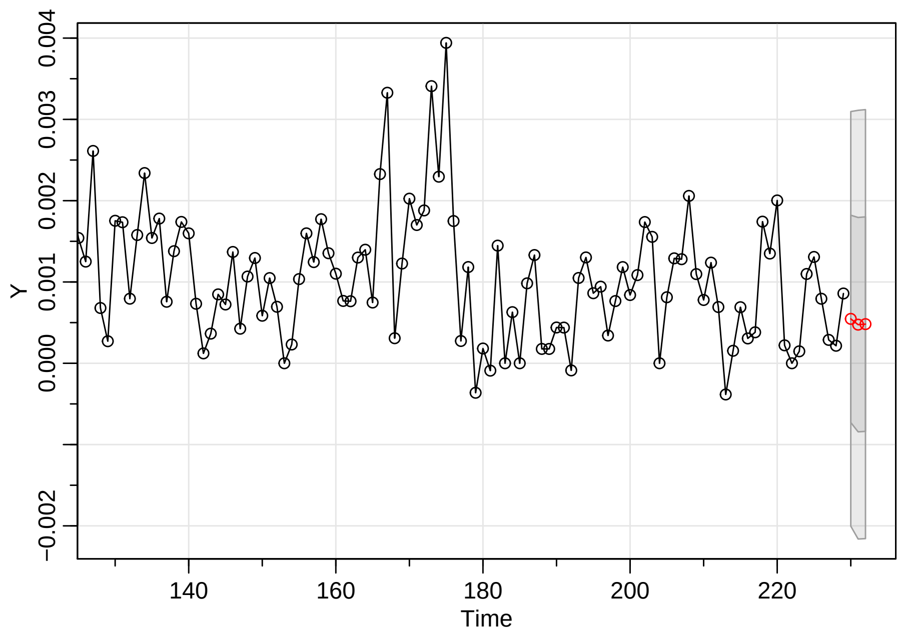
Y
0.004
0.003
0.002
0.001
0.000
−0.002
120
140
160
180
200
220
Time


0.004
0.003
0.002
0.001
0.000
−0.002
Y
140
160
180
200
220
Time

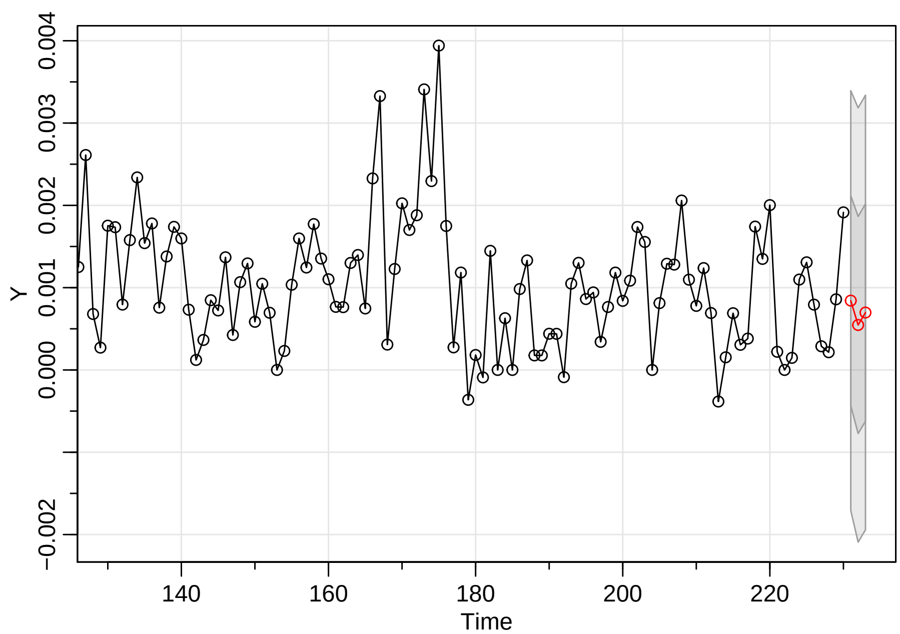

0.004
0.003
0.002
0.001
0.000
−0.002
Y
140
160
180
200
220
Time


Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
Time

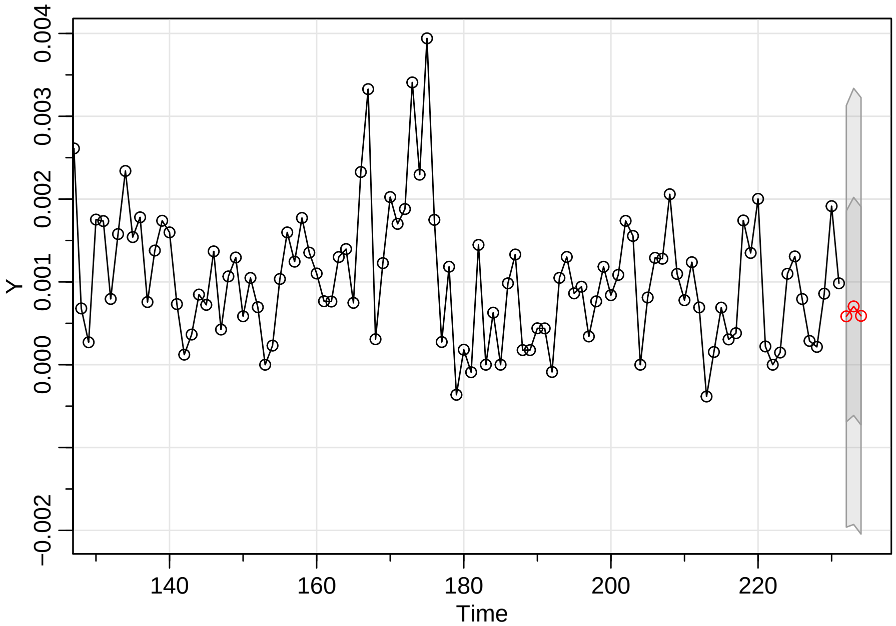

Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
Time



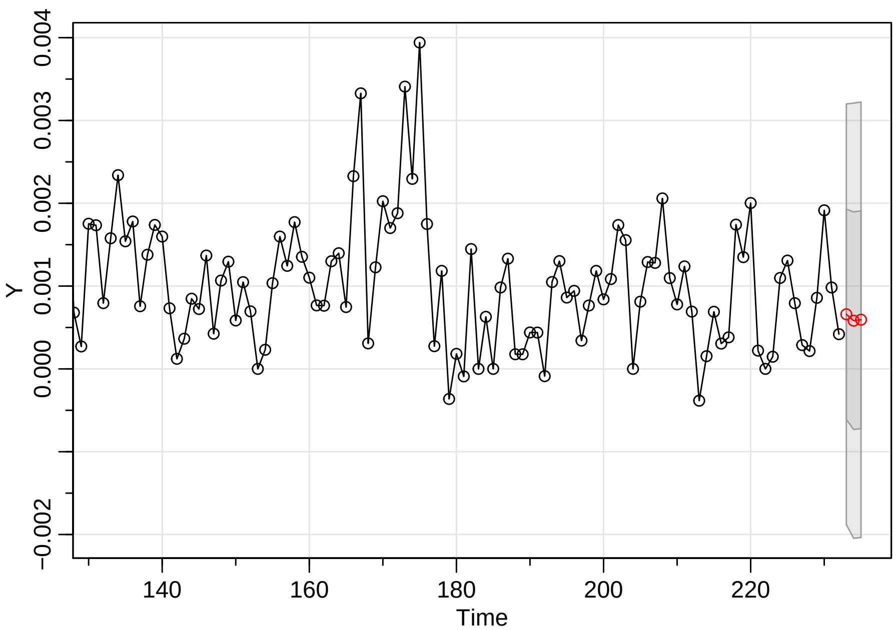
Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
Time



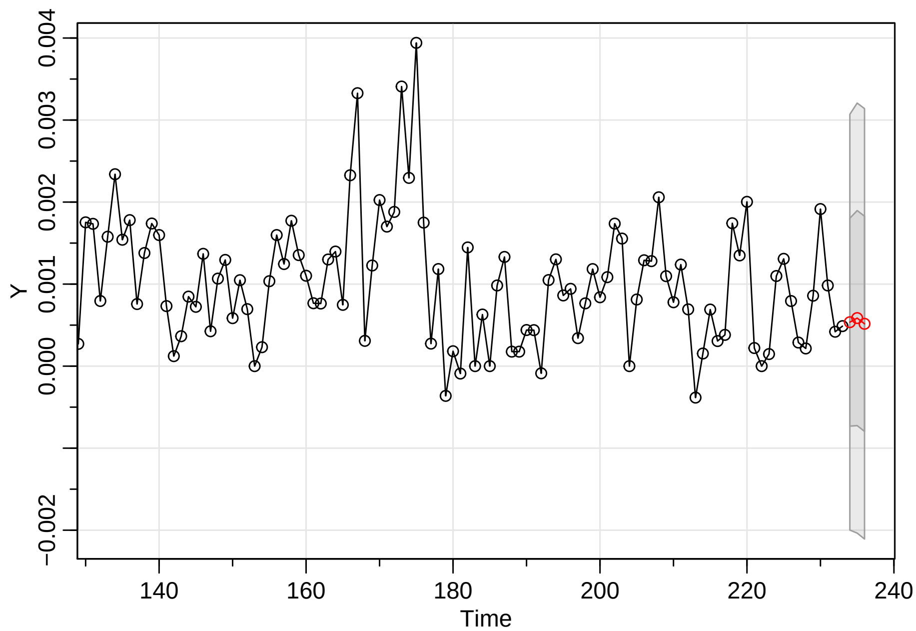
Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
Time

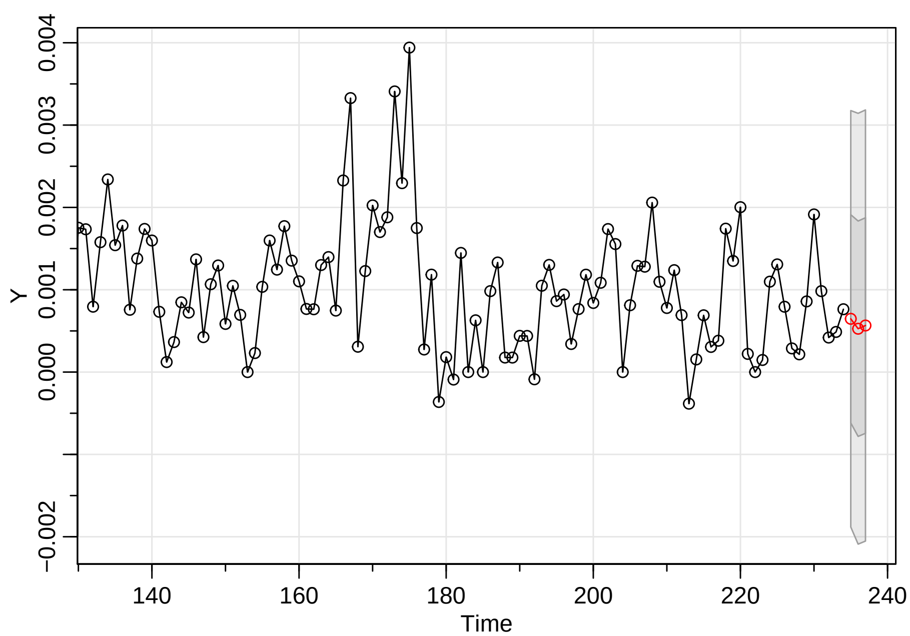


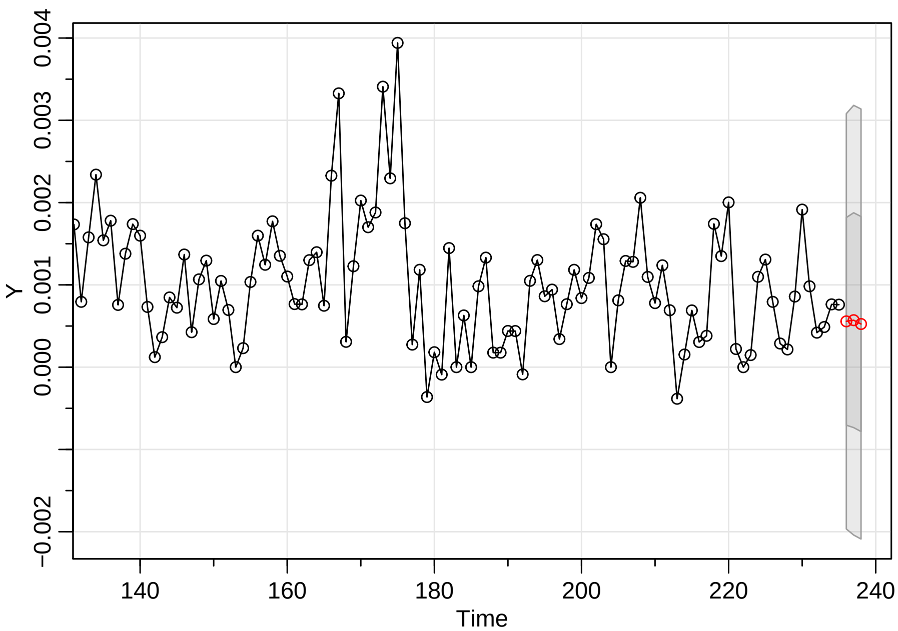
Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
Time



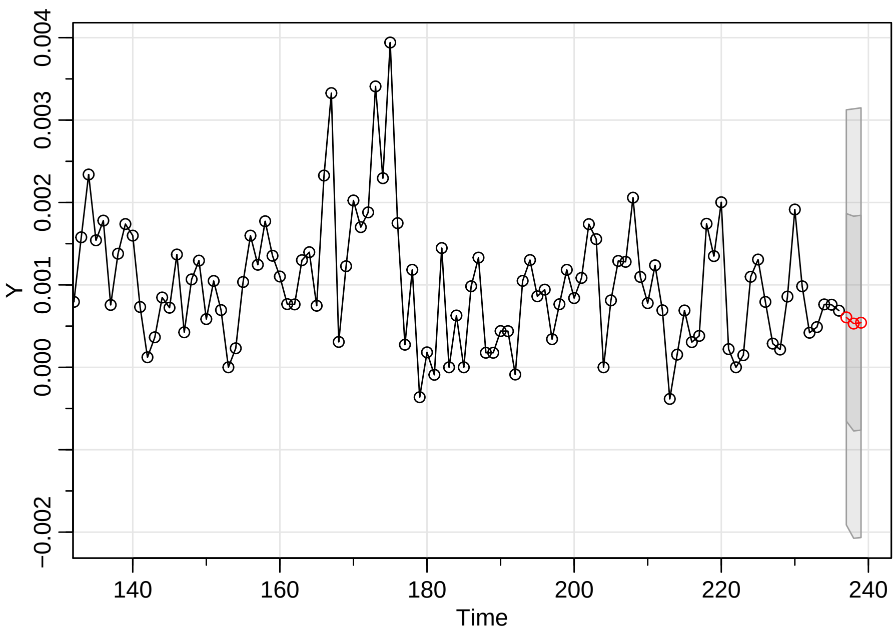
Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
Time


Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
240
Time

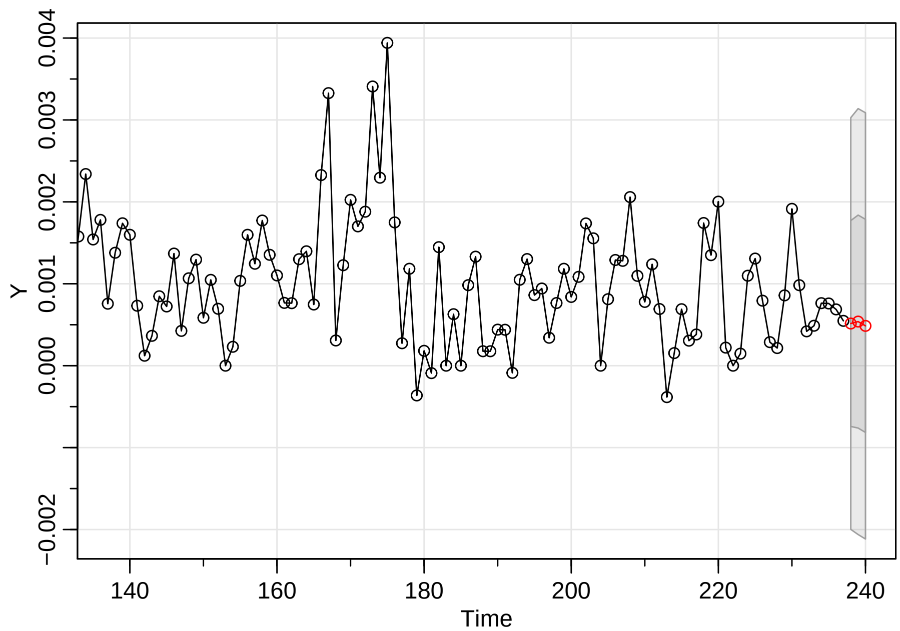

Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
240
Time

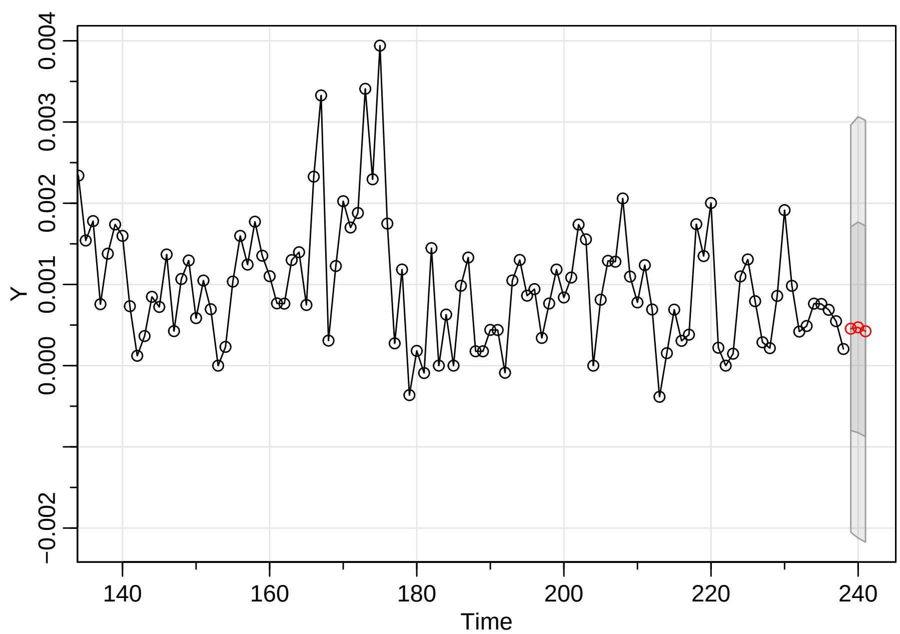


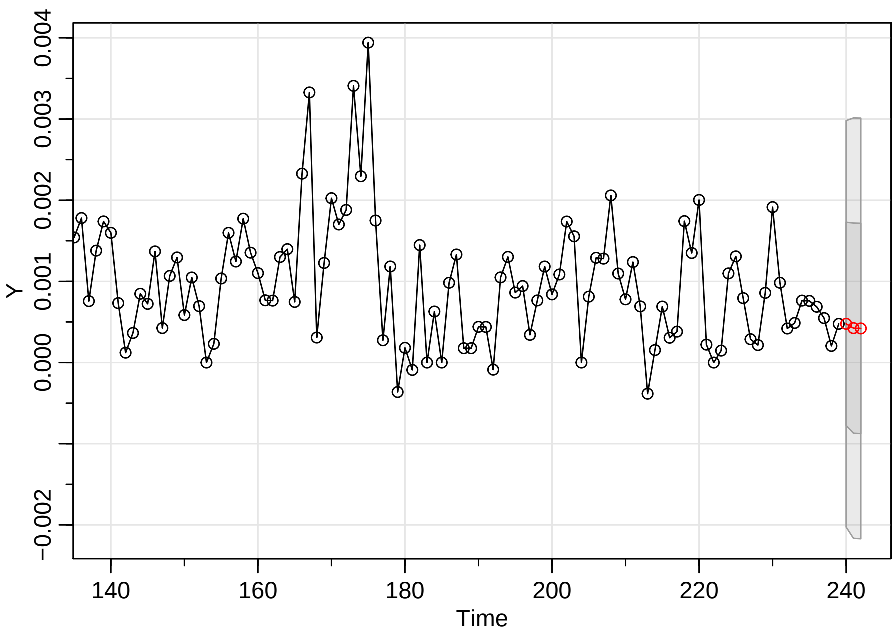
Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
240
Time



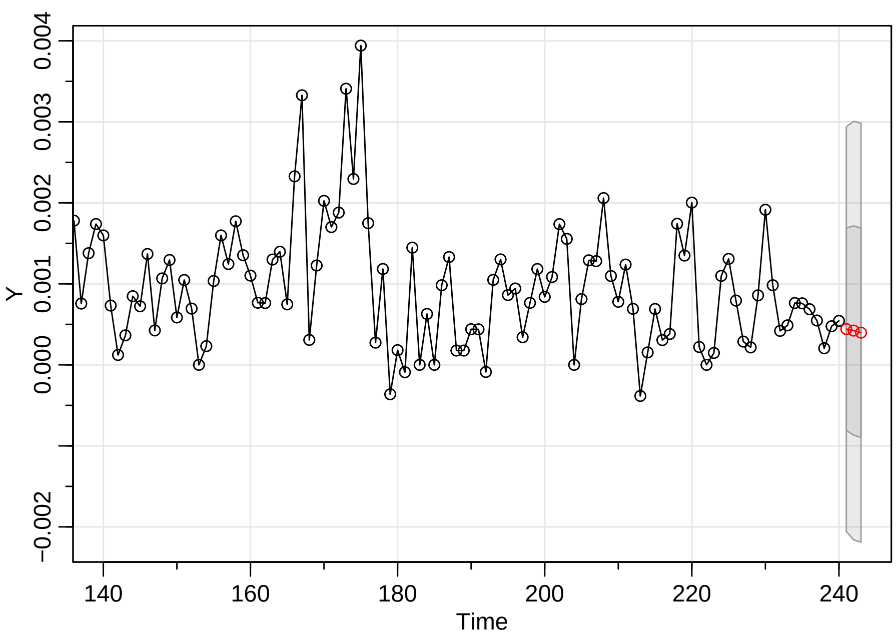
Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
240
Time


Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
240

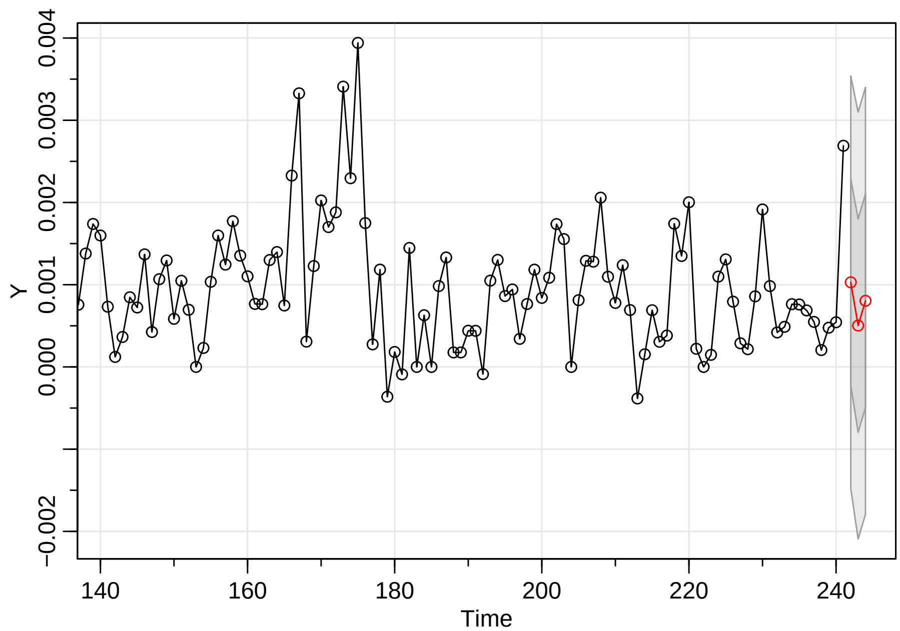

Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
240

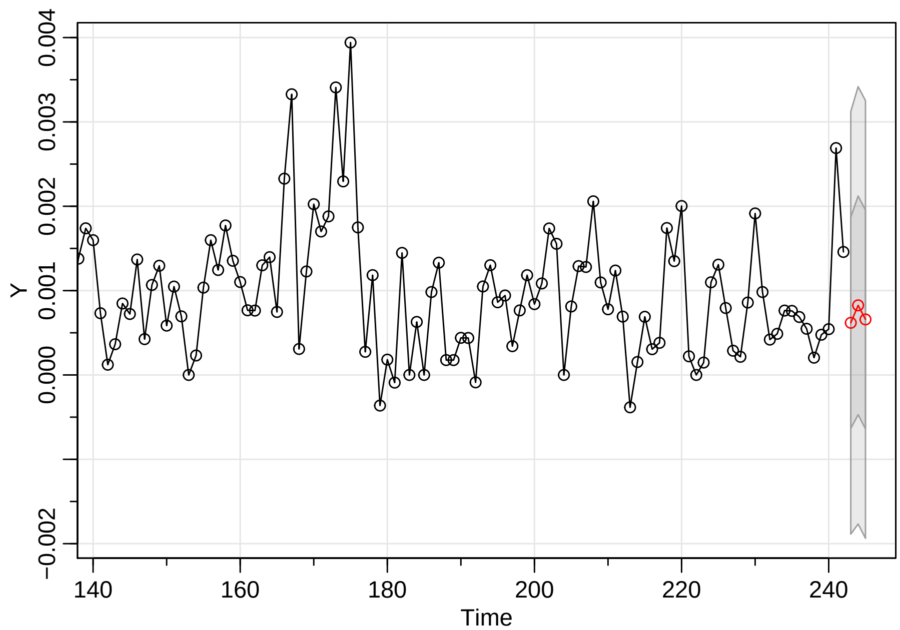


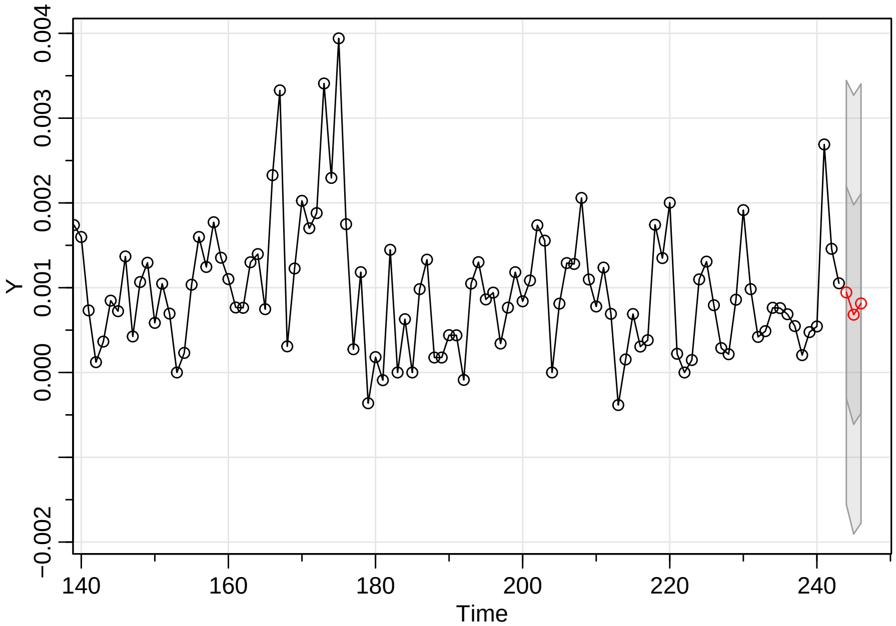
Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
240

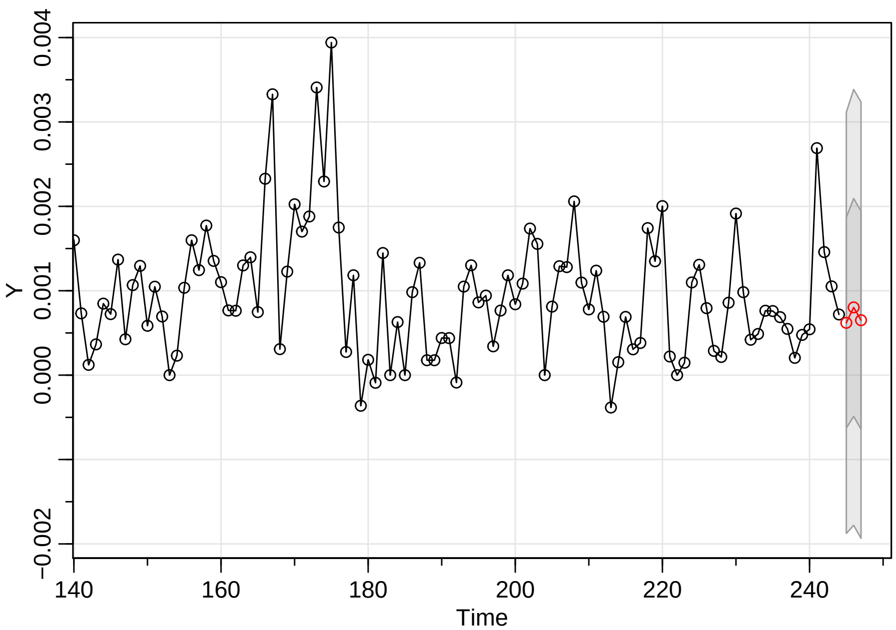


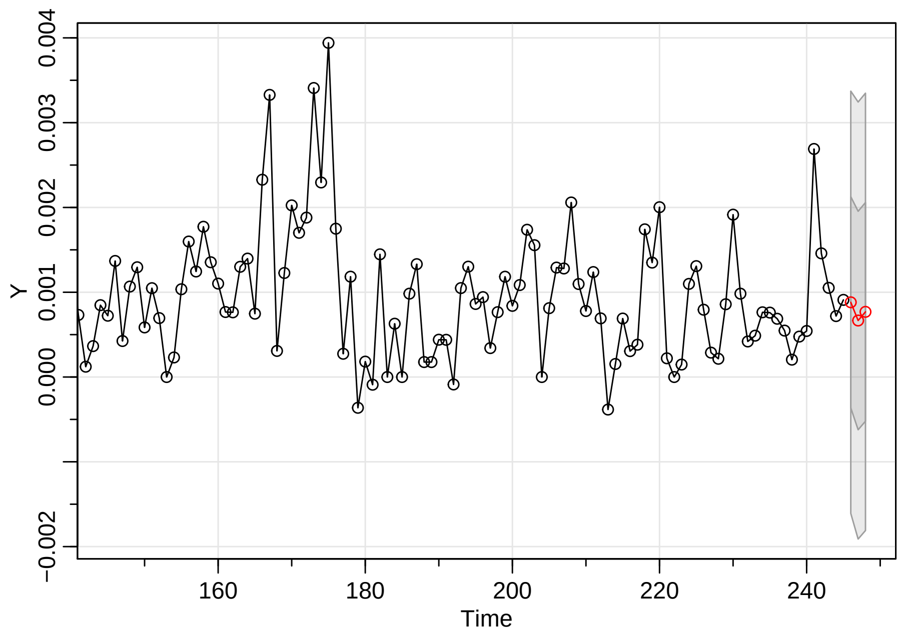
Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
240


Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
240
Time

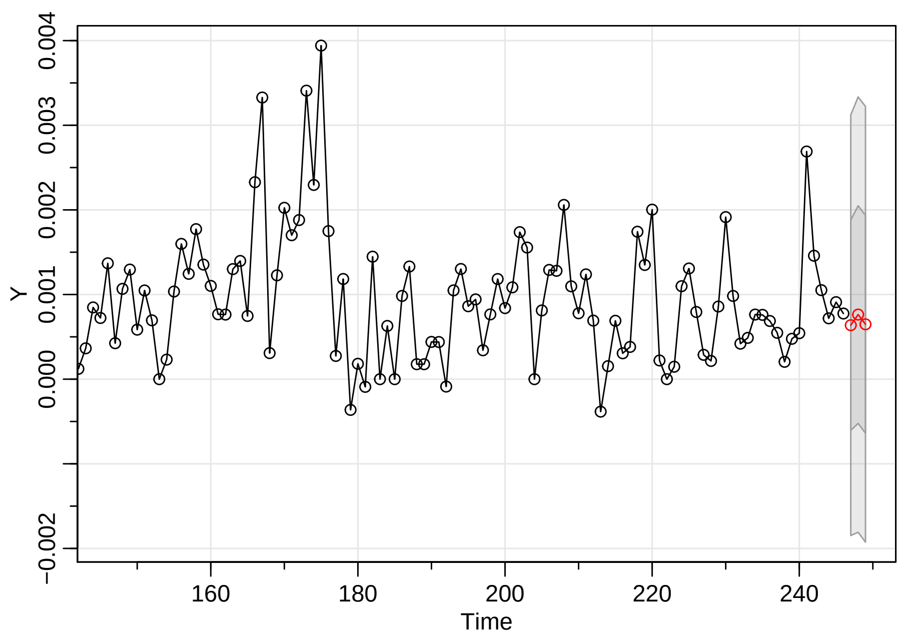

Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
240
Time


Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
240
Time

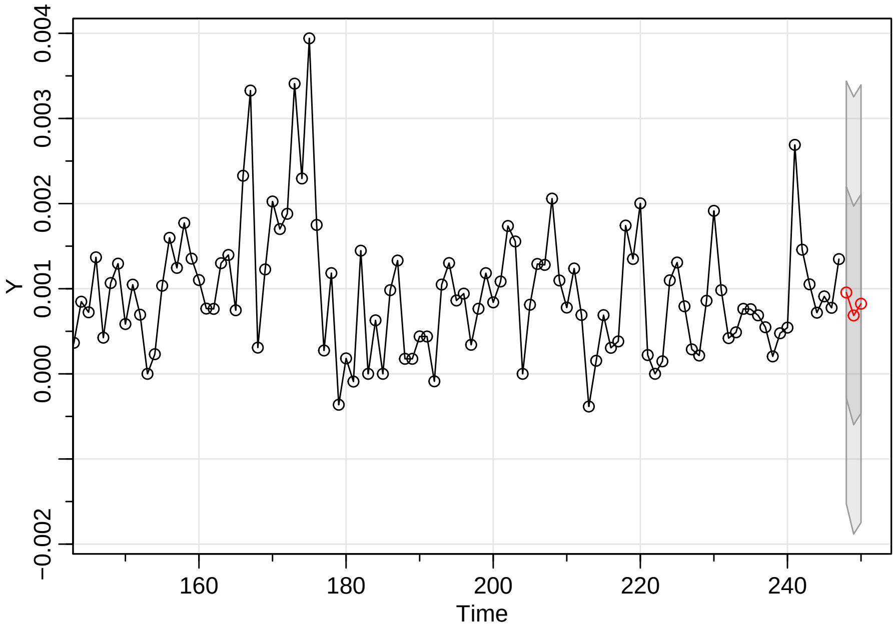

Y
0.004
0.003
0.002
0.001
0.000
−0.002
140
160
180
200
220
240
Time



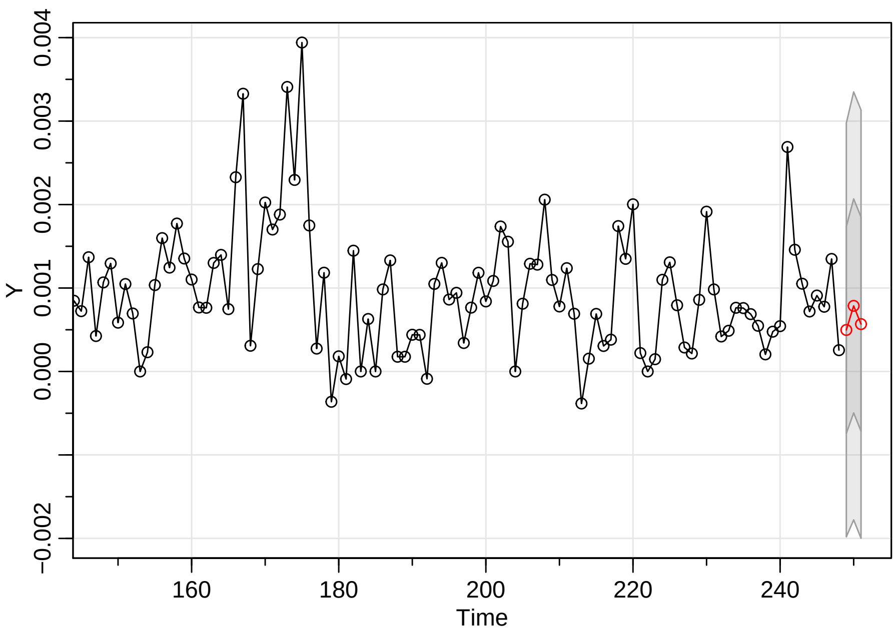
Y
0.004
0.003
0.002
0.001
0.000
−0.002
160
180
200
220
240
Time



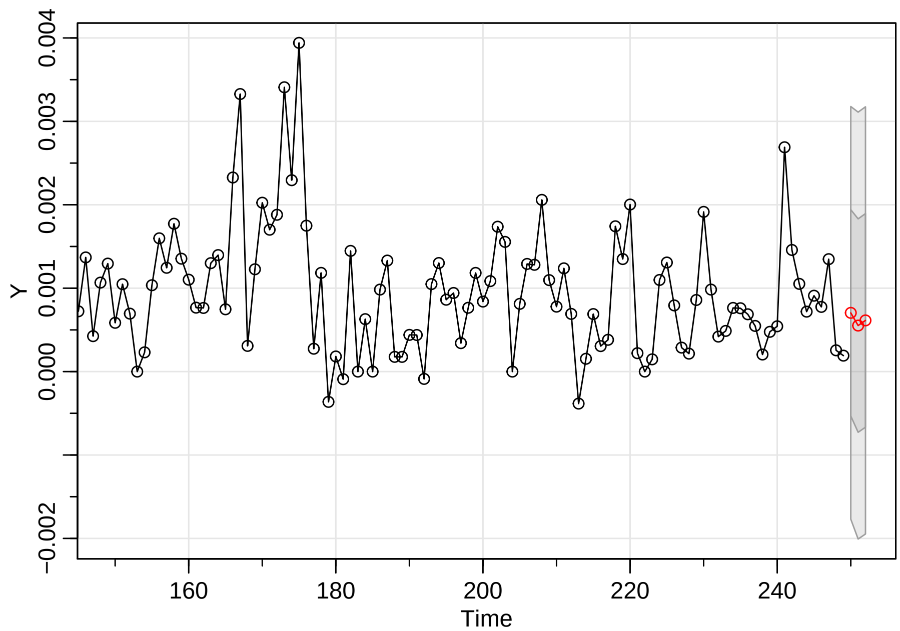
Y
0.004
0.003
0.002
0.001
0.000
−0.002
160
180
200
220
240
Time

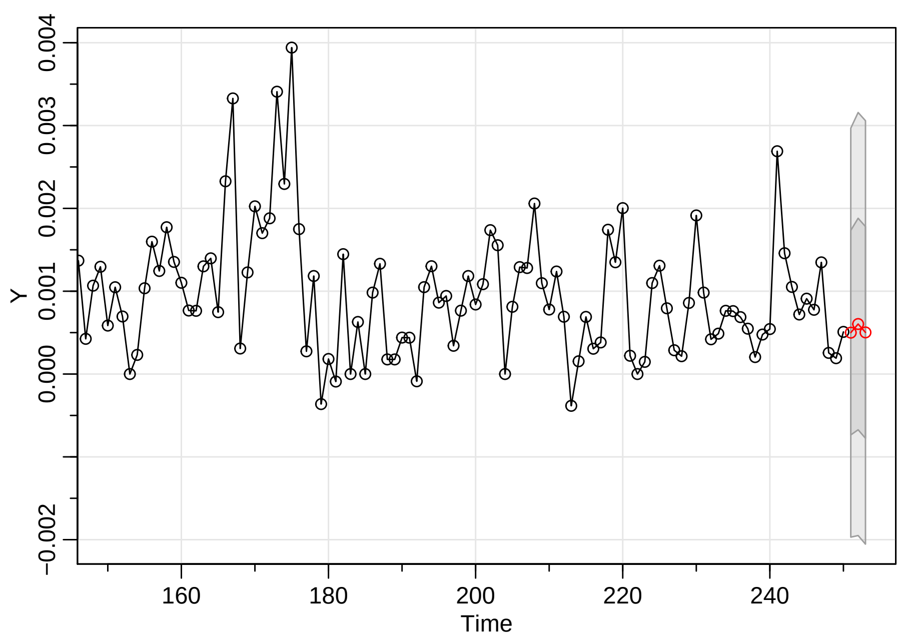


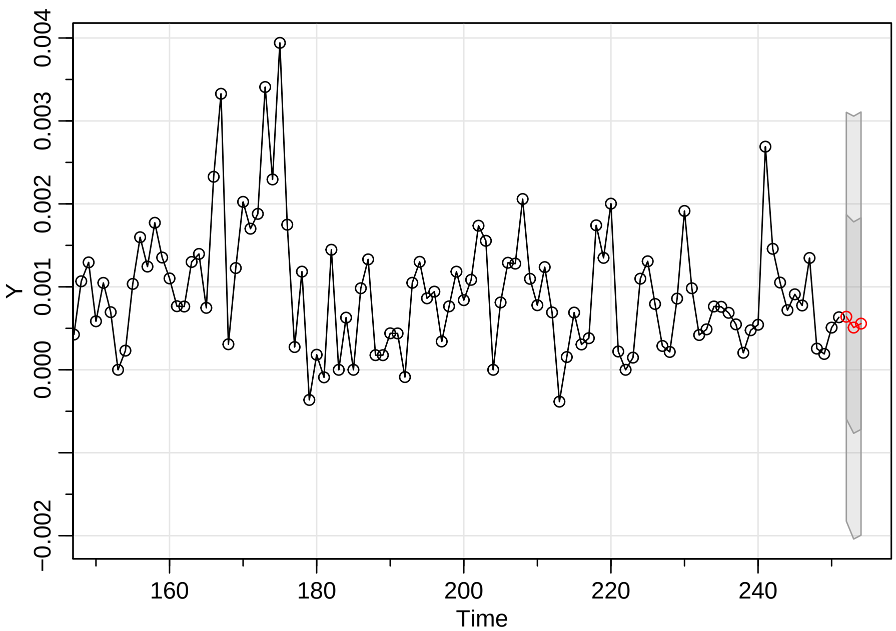
Y
0.004
0.003
0.002
0.001
0.000
−0.002
160
180
200
220
240
Time

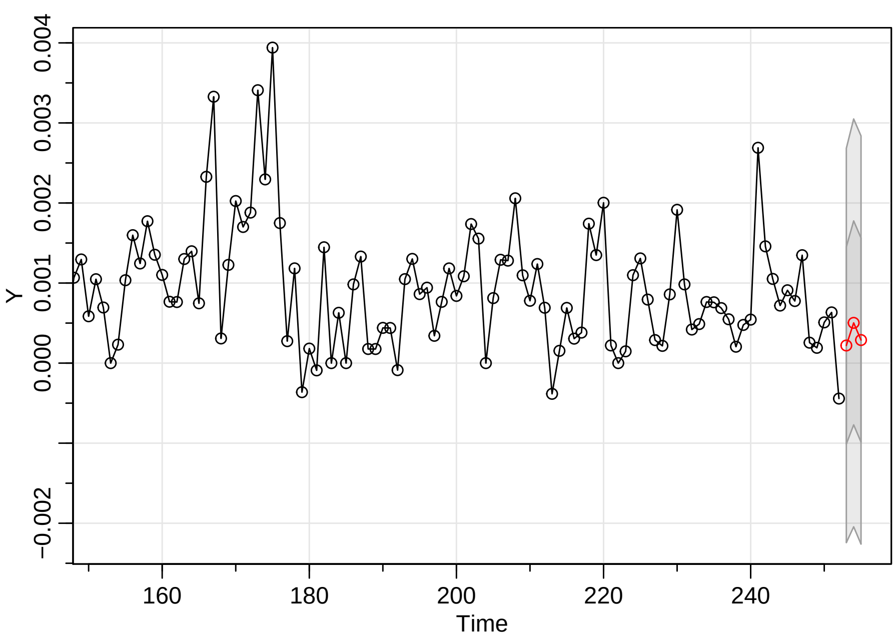


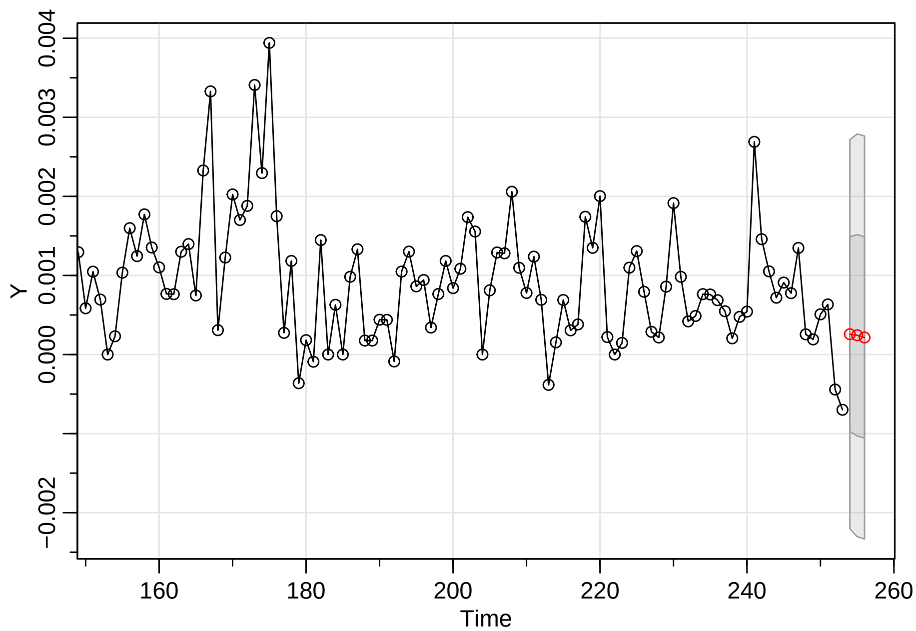
Y
0.004
0.003
0.002
0.001
0.000
−0.002
160
180
200
220
240
Time


Y
Time
−0.002
0.000
0.001
0.002
0.003
0.004
160
180
200
220
240

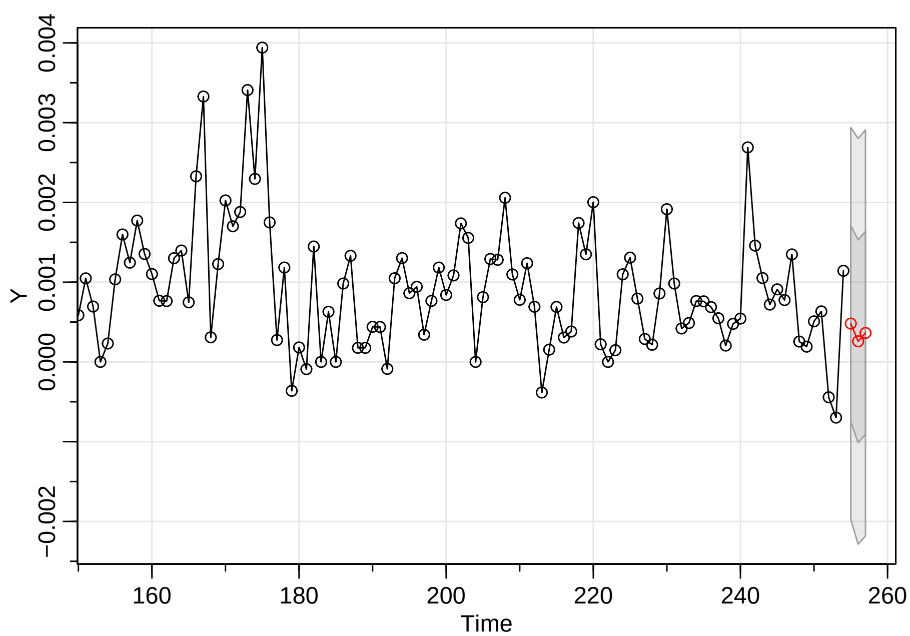

Y
Time
−0.002
0.000
0.001
0.002
0.003
0.004
160
180
200
220
240


Y
−0.002
0.000
0.001
0.002
0.003
0.004
160
180
200
220
240
Time

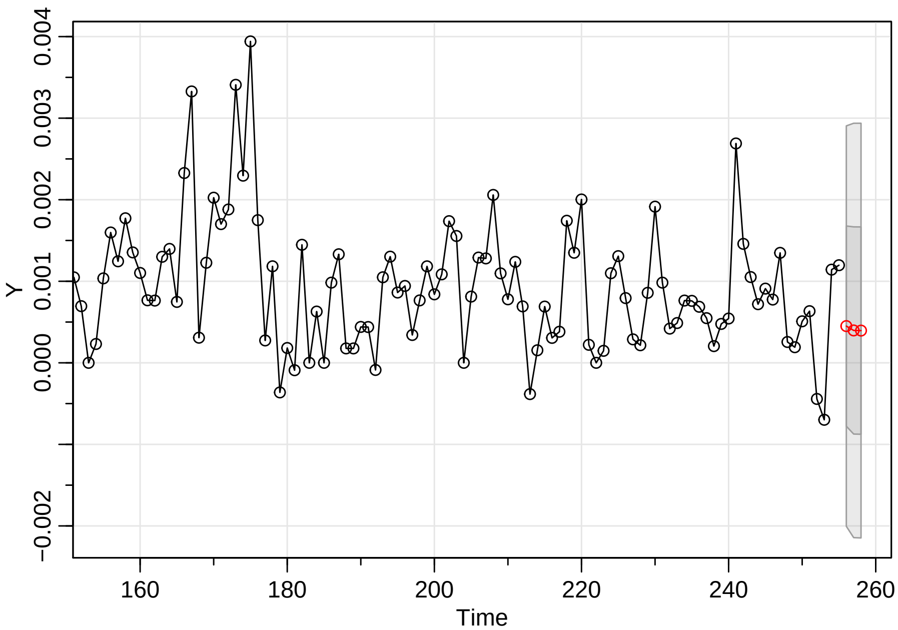

Y
−0.002
0.000
0.001
0.002
0.003
0.004
160
180
200
220
240
Time



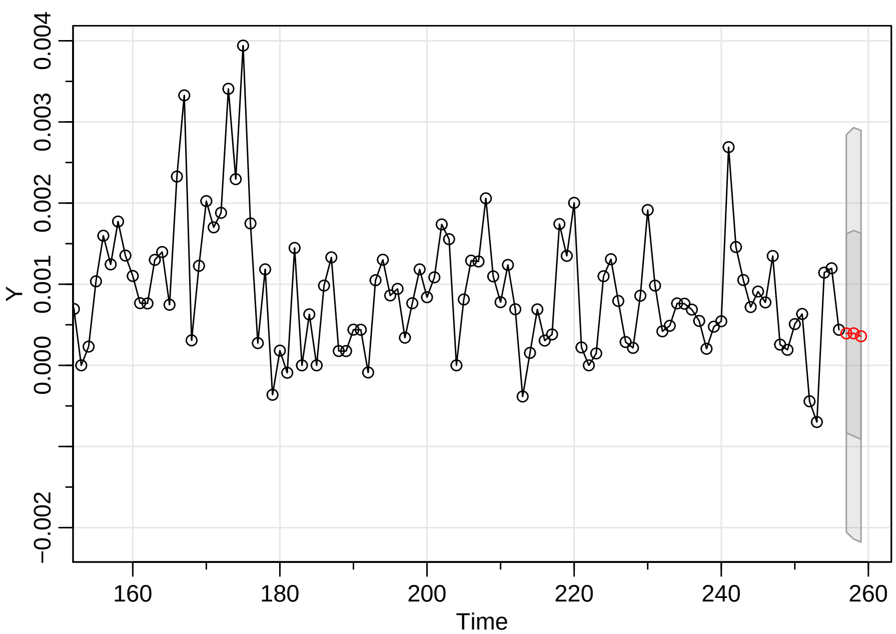
Y
0.004
0.003
0.002
0.001
0.000
−0.002
160
180
200
220
240
260
Time



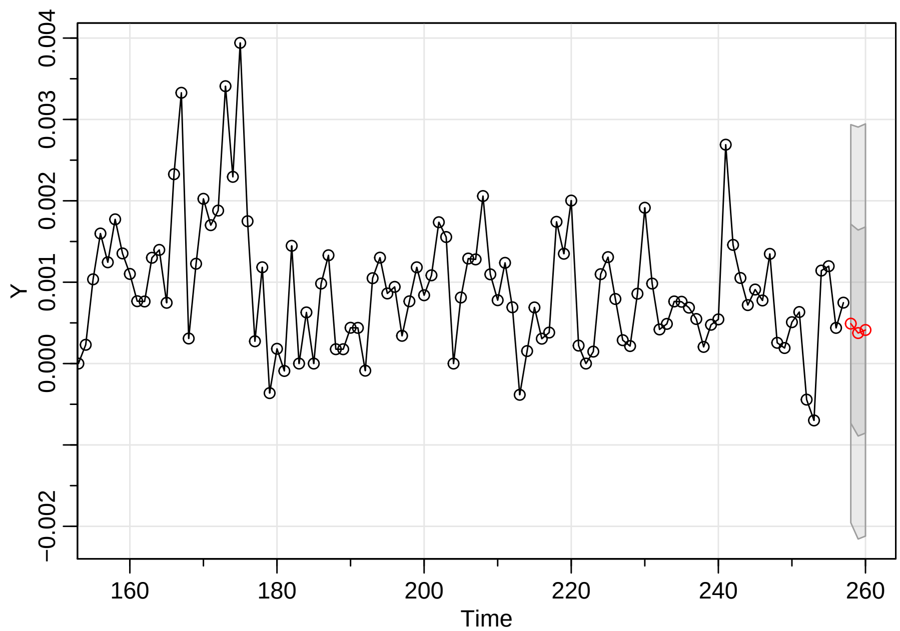
Y
0.004
0.003
0.002
0.001
0.000
−0.002
160
180
200
220
240
260
Time

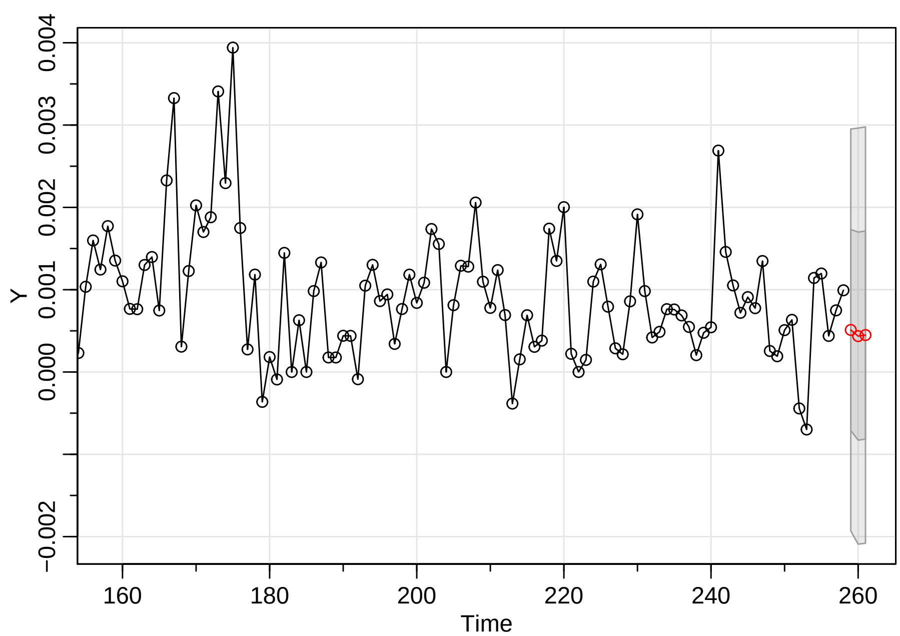


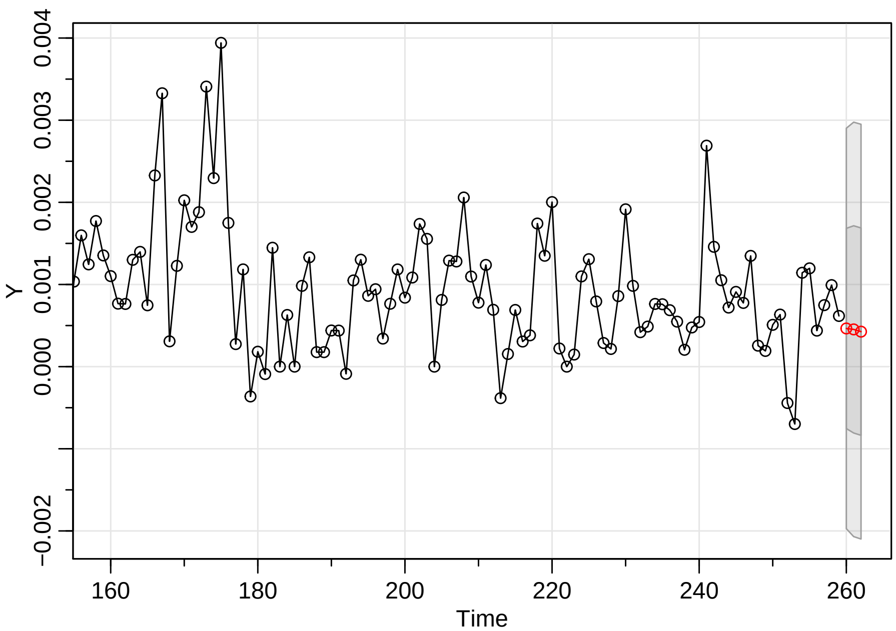
Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
160
180
200
220
240
260



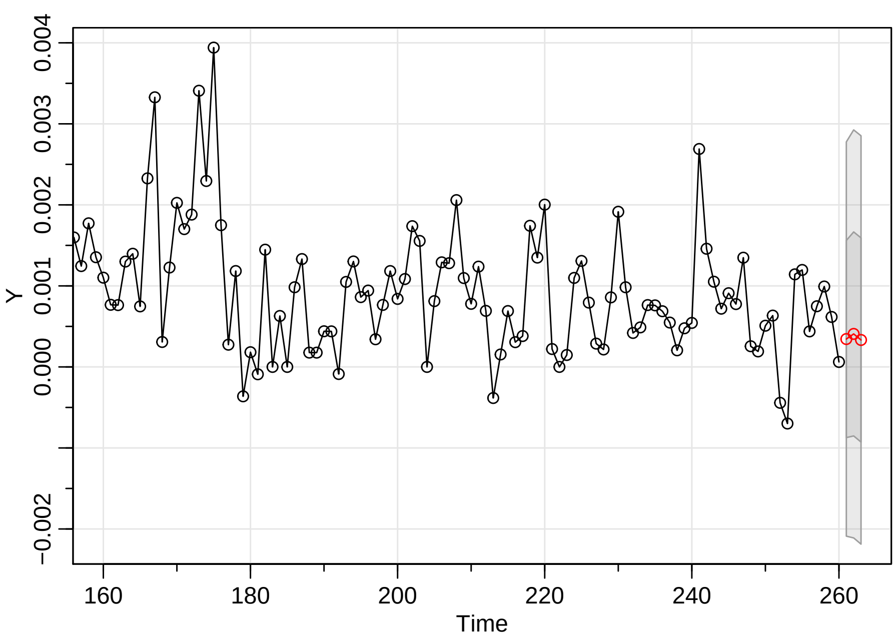
Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
160
180
200
220
240
260


Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
160
180
200
220
240
260

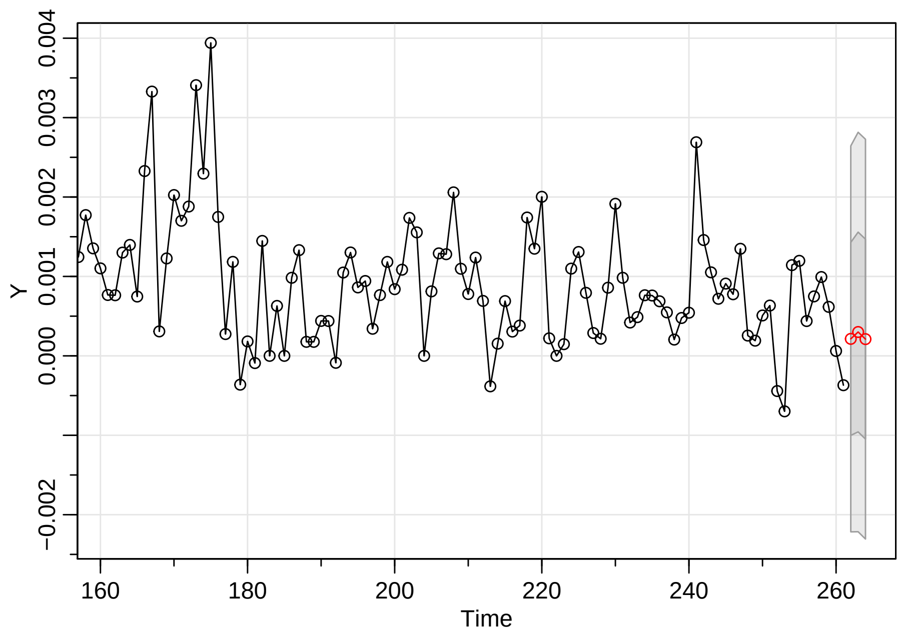

Y
Time
0.004
0.003
0.002
0.001
0.000
−0.002
160
180
200
220
240
260

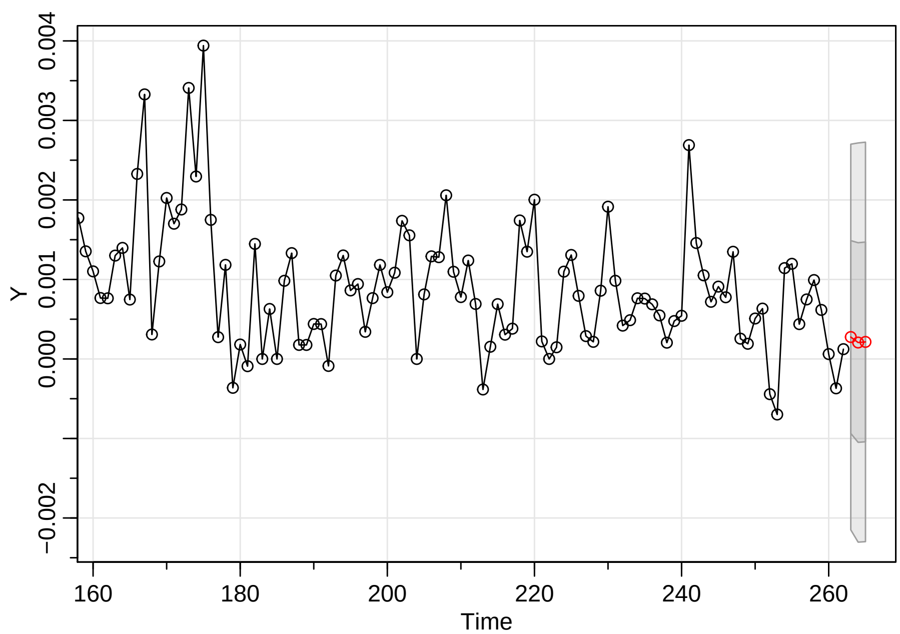


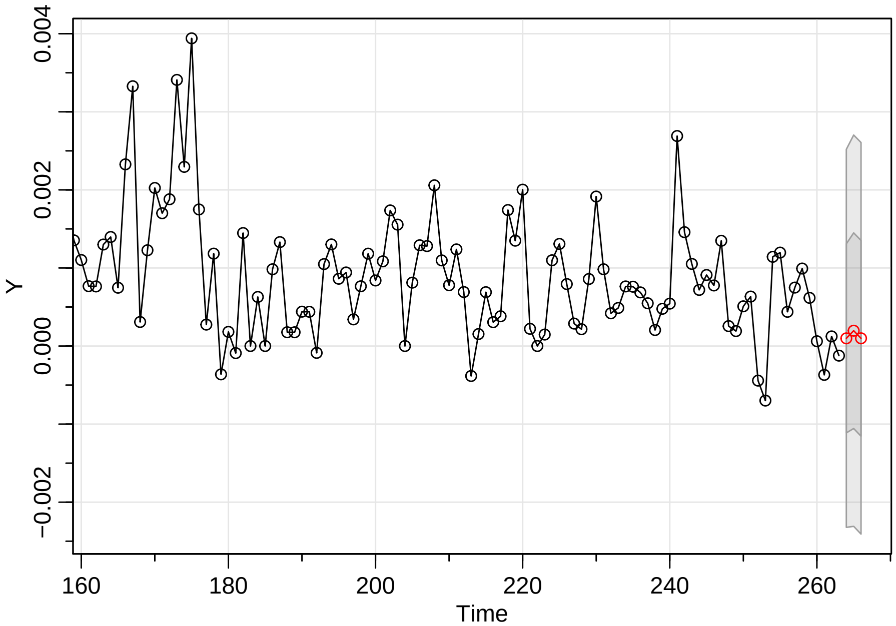
0.004
0.003
0.002
0.001
0.000
−0.002
Y
160
180
200
220
240
260
Time

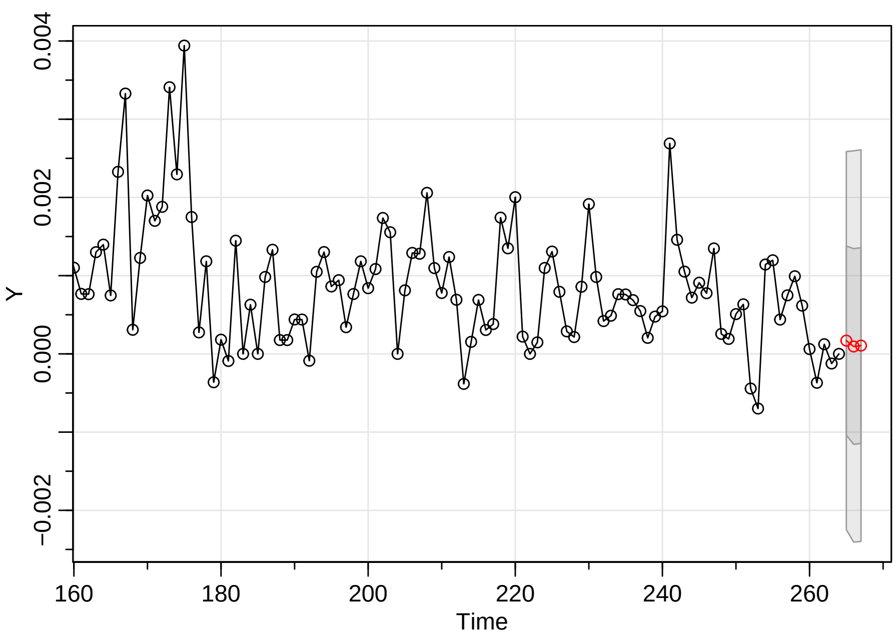


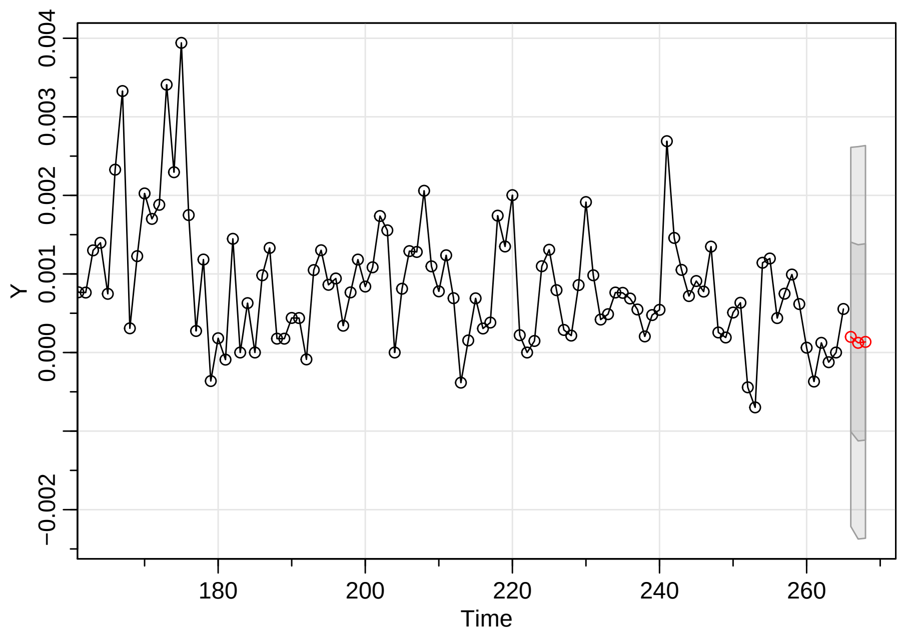
0.004
0.003
0.002
0.001
0.000
−0.002
Y
160
180
200
220
240
260
Time

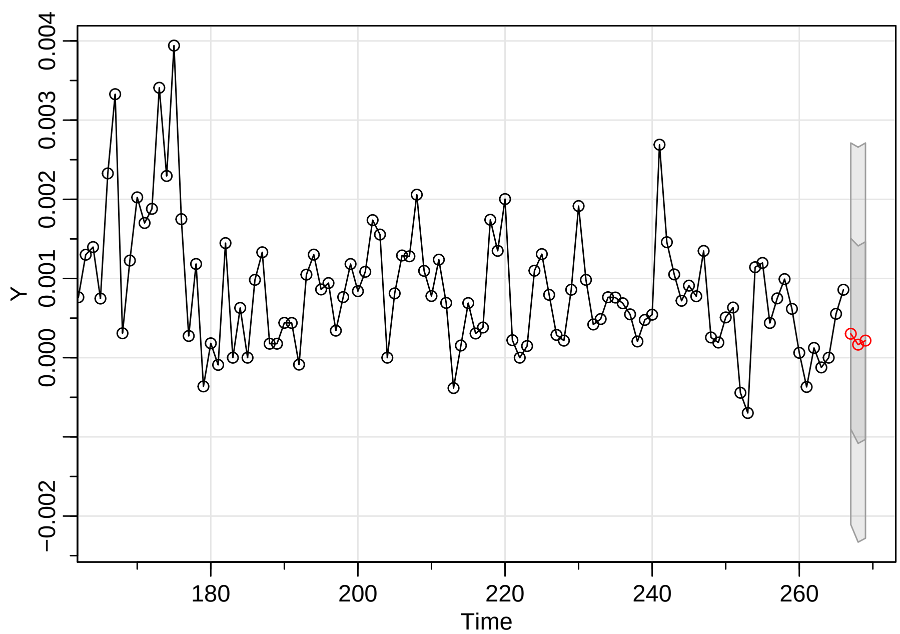

Y
0.004
0.003
0.002
0.001
0.000
−0.002
160
180
200
220
240
260
Time

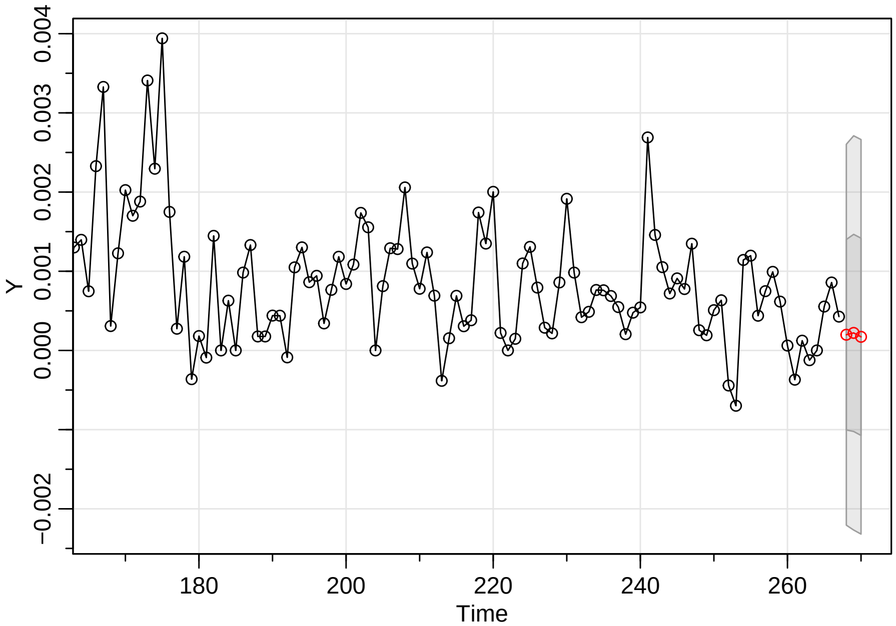

Y
0.004
0.003
0.002
0.001
0.000
−0.002
160
180
200
220
240
260
Time

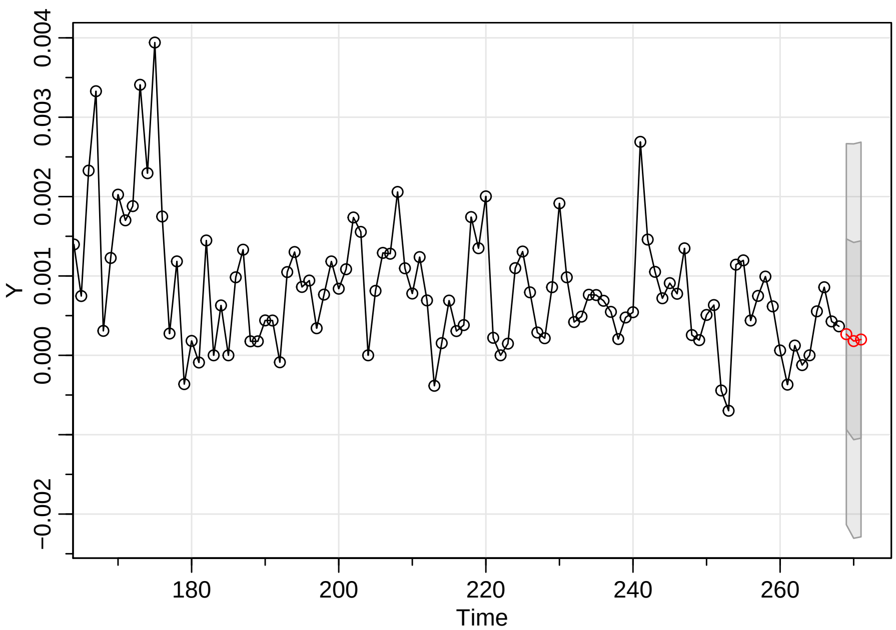

Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
260
Time

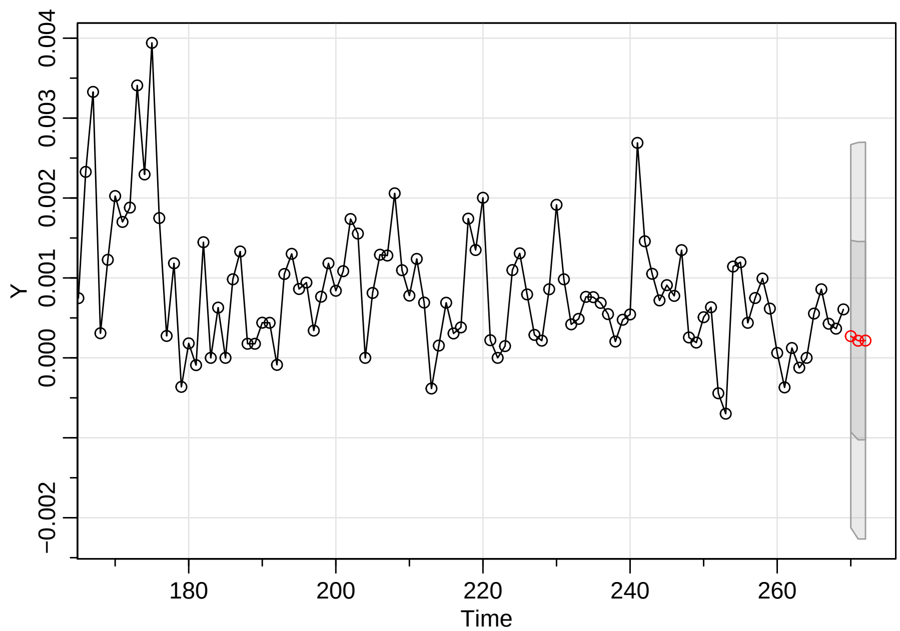

Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
260
Time

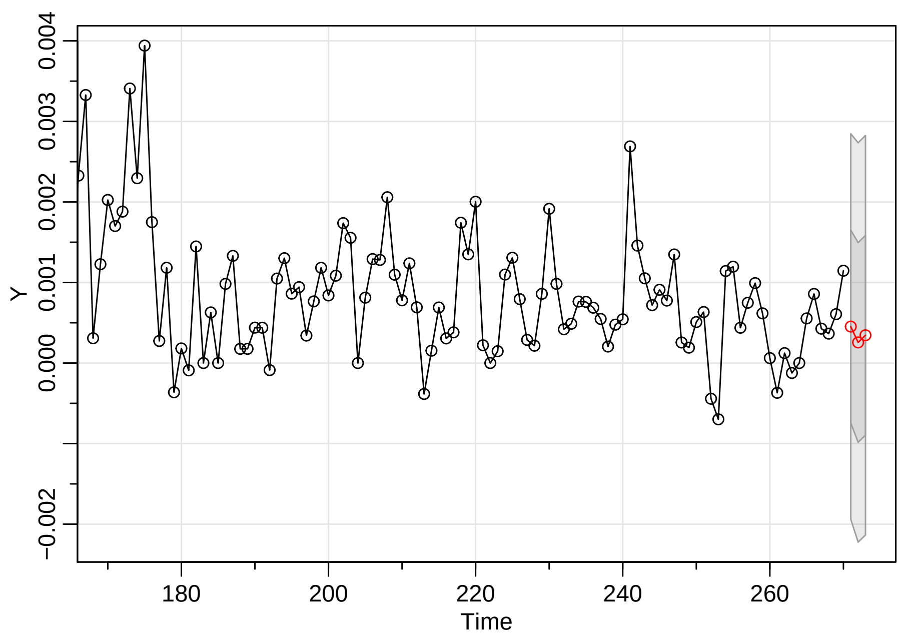

Y
Time

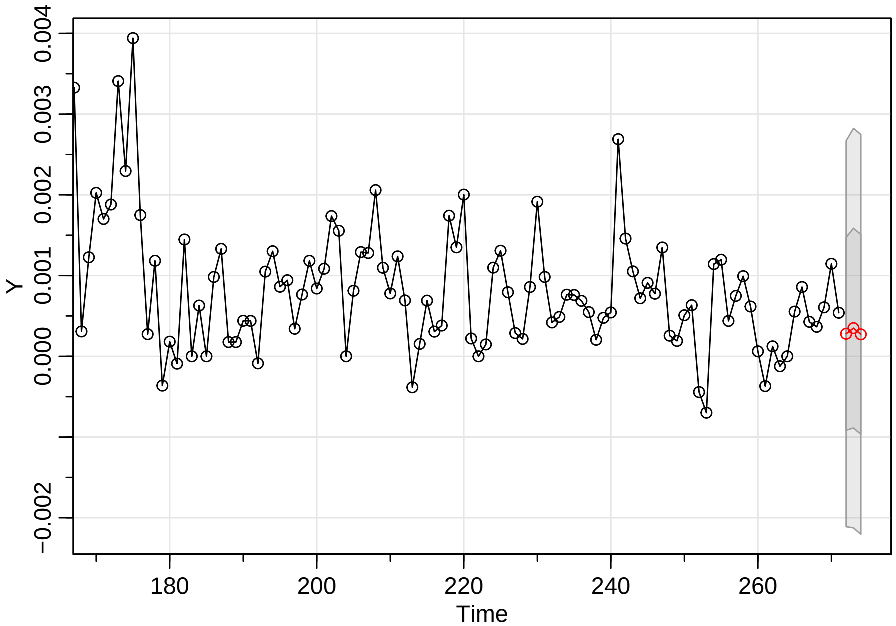

Y
Time

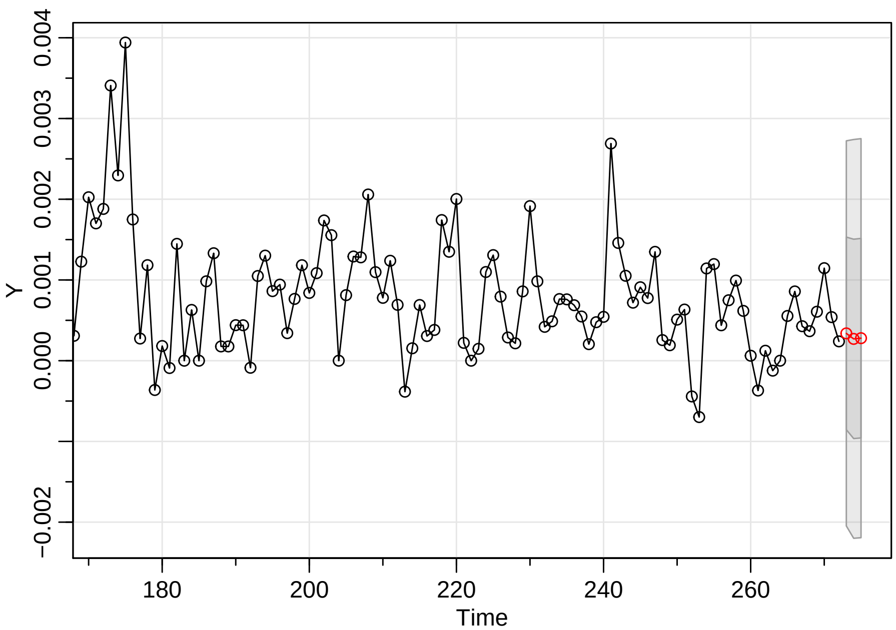

Y
0.004
0.003
0.002
0.001
0.000
−0.002
180
200
220
240
260
Time

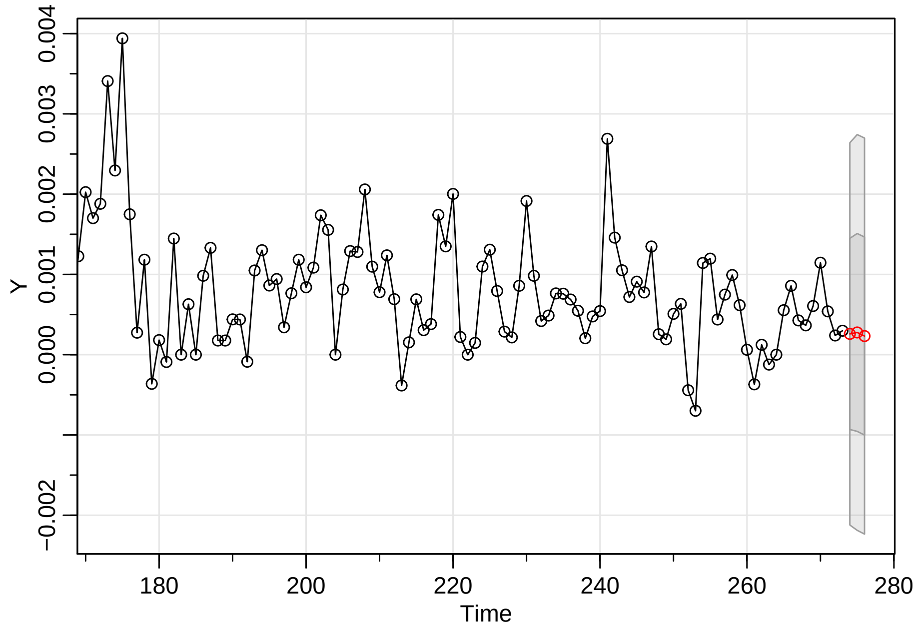

Y
0.004
0.003
0.002
0.001
0.000
−0.002
180
200
220
240
260
Time

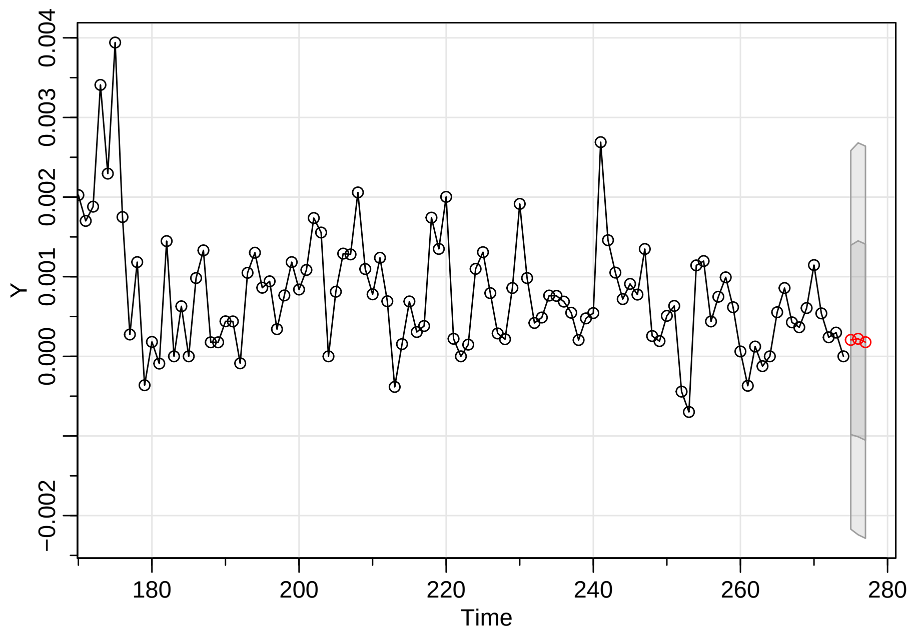

Y
0.004
0.003
0.002
0.001
0.000
−0.002
180
200
220
240
260
Time

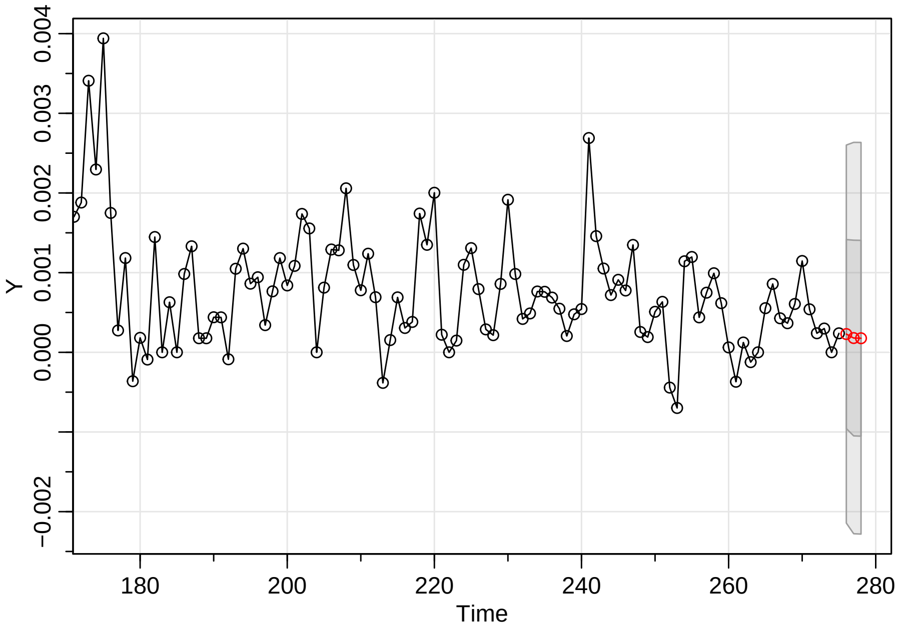

Y
0.004
0.003
0.002
0.001
0.000
−0.002
180
200
220
240
260
Time

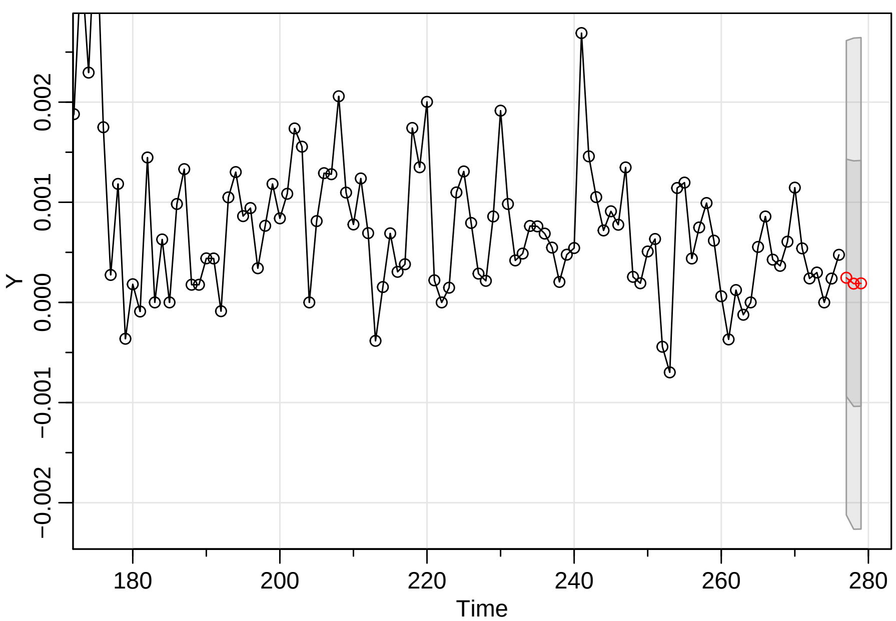

Y
0.004
0.003
0.002
0.001
0.000
−0.002
180
200
220
240
260
Time

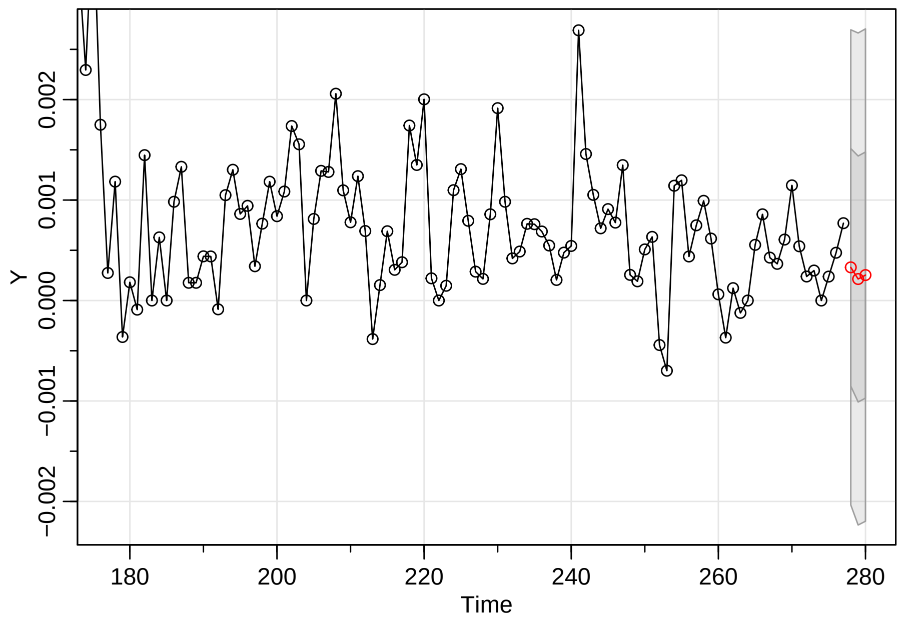

Y
0.004
0.003
0.002
0.001
0.000
−0.002
180
200
220
240
260
Time

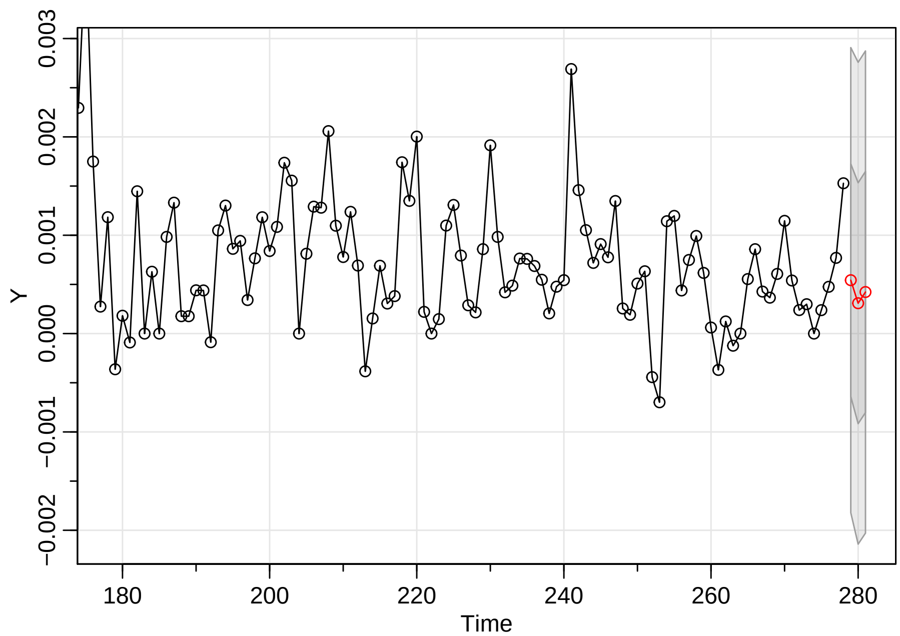

Y
0.004
0.003
0.002
0.001
0.000
−0.002
180
200
220
240
260
280
Time

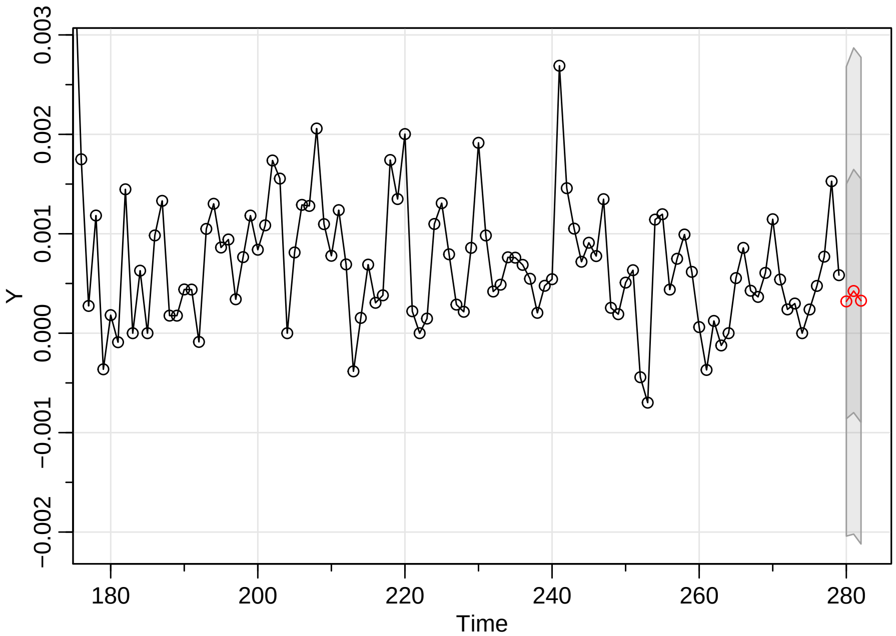

Y
0.004
0.003
0.002
0.001
0.000
−0.002
180
200
220
240
260
280
Time

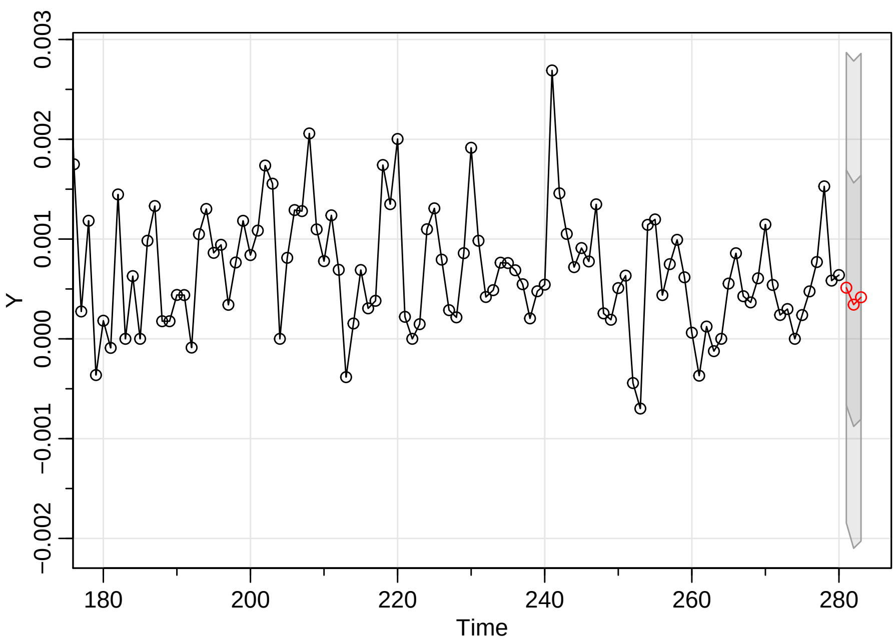

Y
0.004
0.003
0.002
0.001
0.000
−0.002
180
200
220
240
260
280
Time

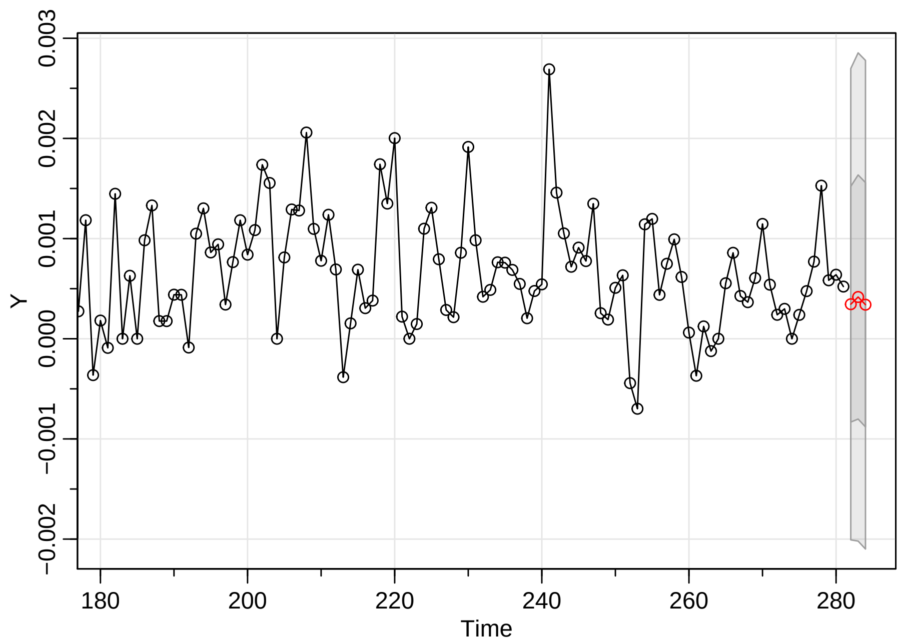

Y
0.002
0.001
0.000
−0.001
−0.002
180
200
220
240
260
280
Time

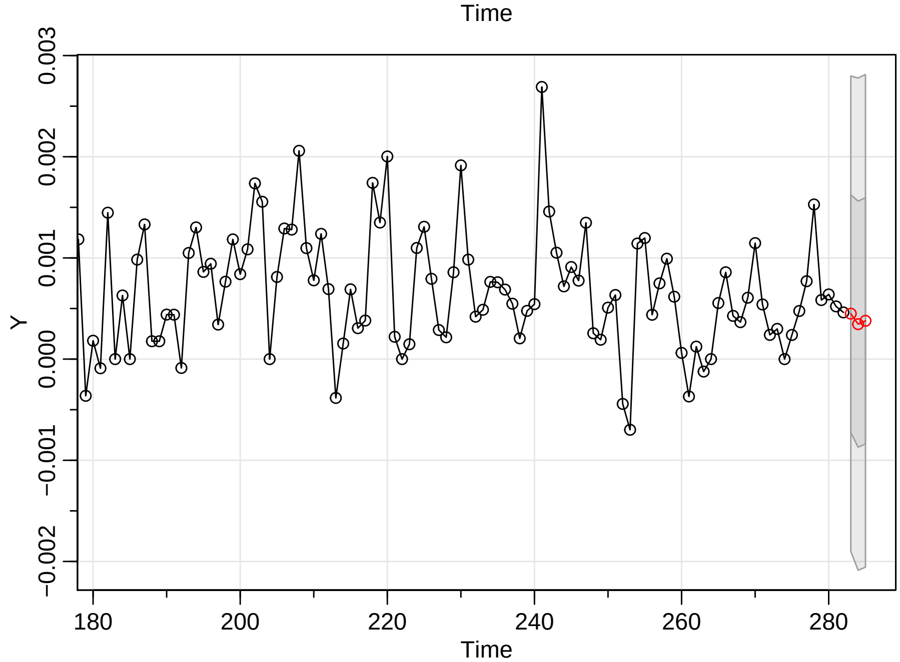

Y
0.002
0.001
0.000
−0.001
−0.002
180
200
220
240
260
280
Time

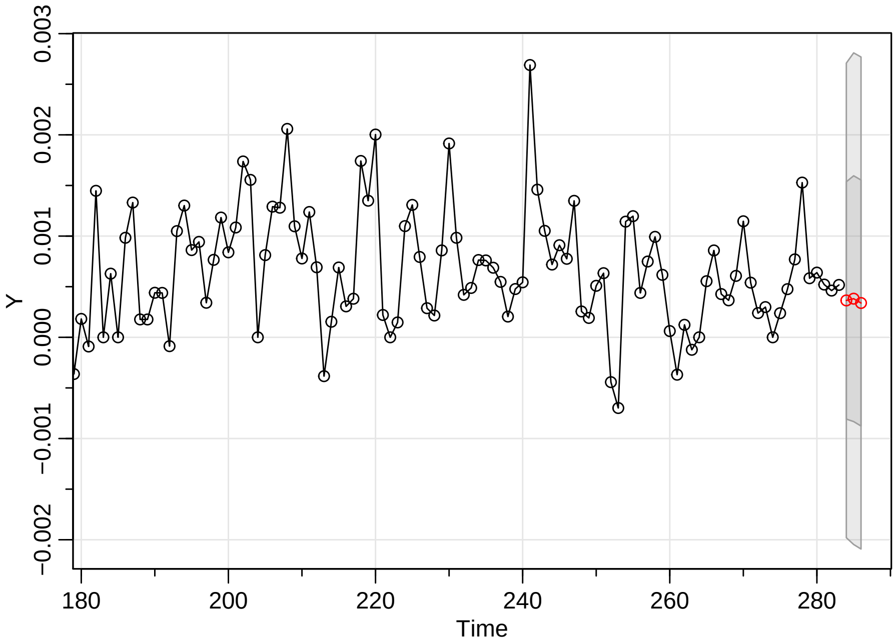

Y
0.003
0.002
0.001
0.000
−0.001
−0.002
180
200
220
240
260
280
Time

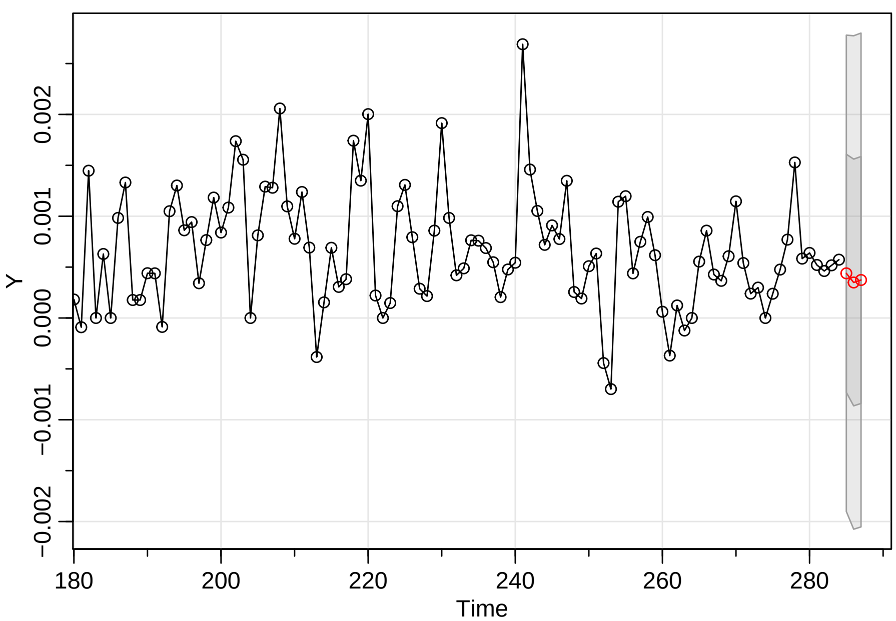

Y
0.003
0.002
0.001
0.000
−0.001
−0.002
180
200
220
240
260
280
Time

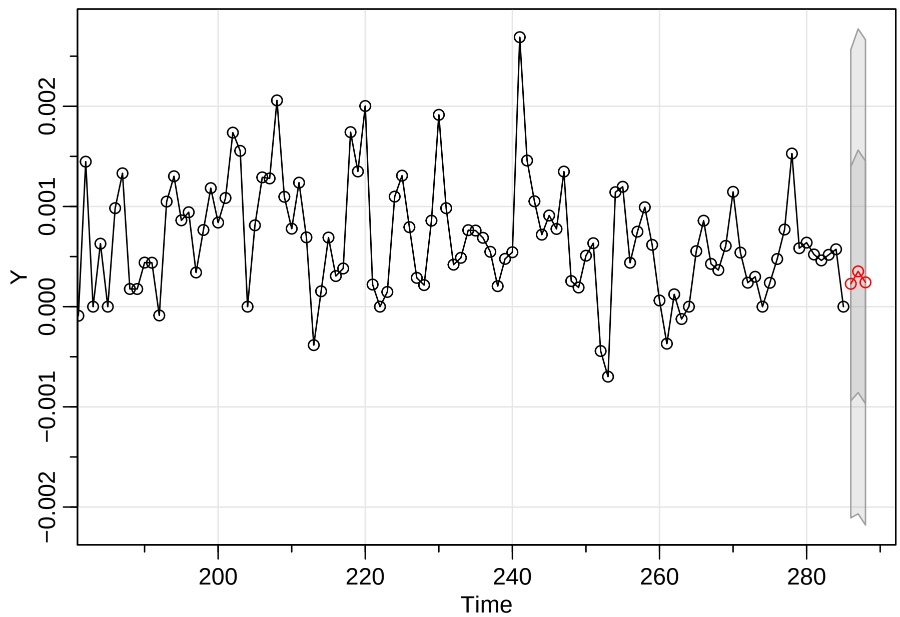

Y
0.003
0.002
0.001
0.000
−0.001
−0.002
180
200
220
240
260
280
Time

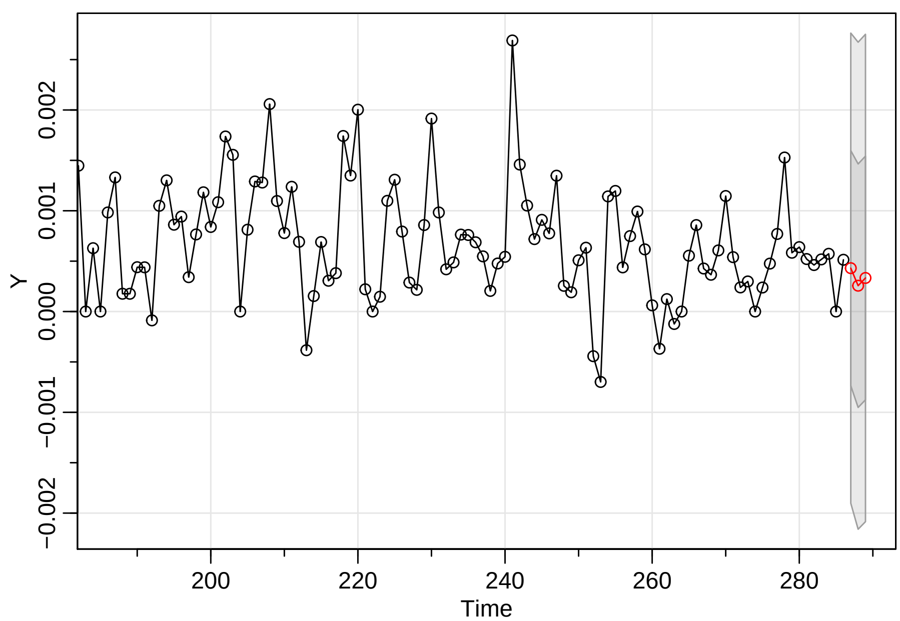

Y
0.003
0.002
0.001
0.000
−0.001
−0.002
180
200
220
240
260
280
Time

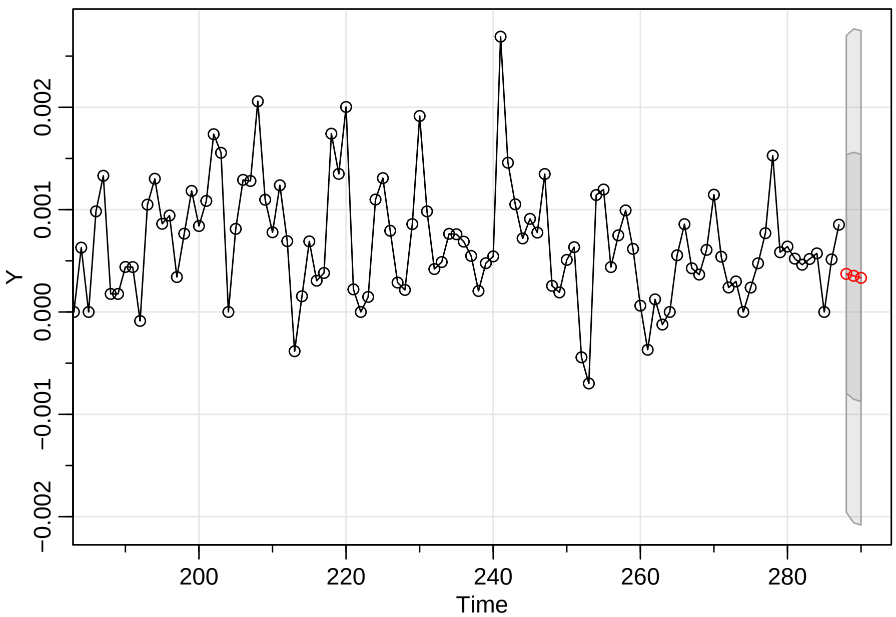

Y
0.003
0.002
0.001
0.000
−0.001
−0.002
180
200
220
240
260
280
Time

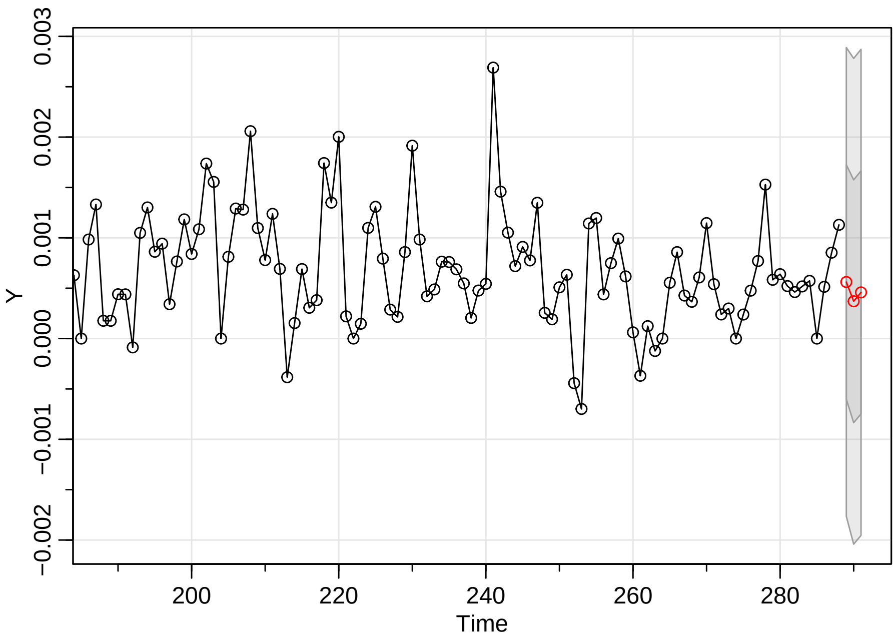

0.003
0.002
0.001
0.000
−0.001
−0.002
Y
180
200
220
240
260
280
Time

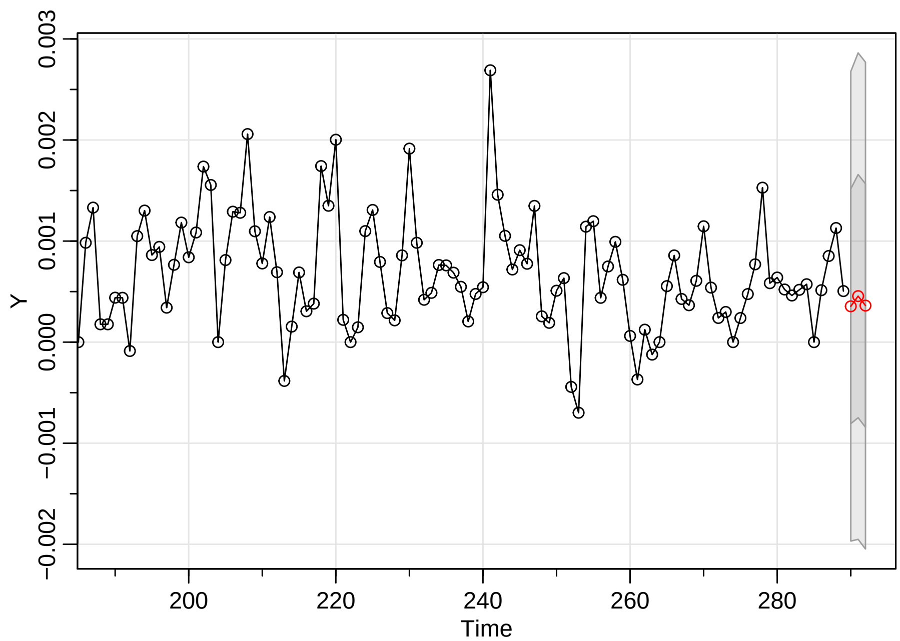

0.002
0.001
0.000
−0.001
−0.002
Y
180
200
220
240
260
280
Time


Y
Time
200
220
240
260
280
−0.002
−0.001
0.000
0.001
0.002

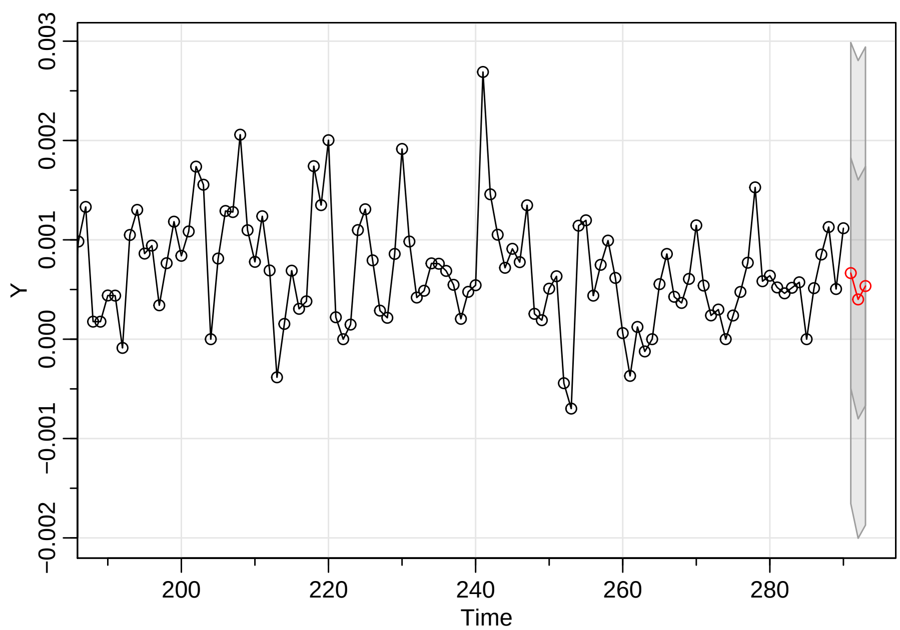

Y
Time
200
220
240
260
280
−0.002
−0.001
0.000
0.001
0.002


Y
Time
0.002
0.001
0.000
−0.001
−0.002
200
220
240
260
280

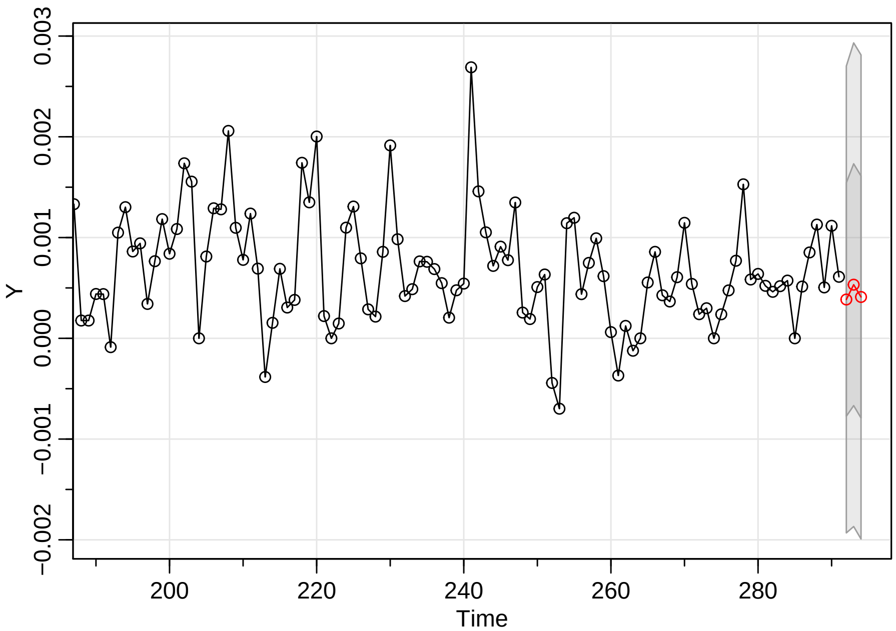

Y
Time
0.003
0.002
0.001
0.000
−0.001
−0.002
200
220
240
260
280

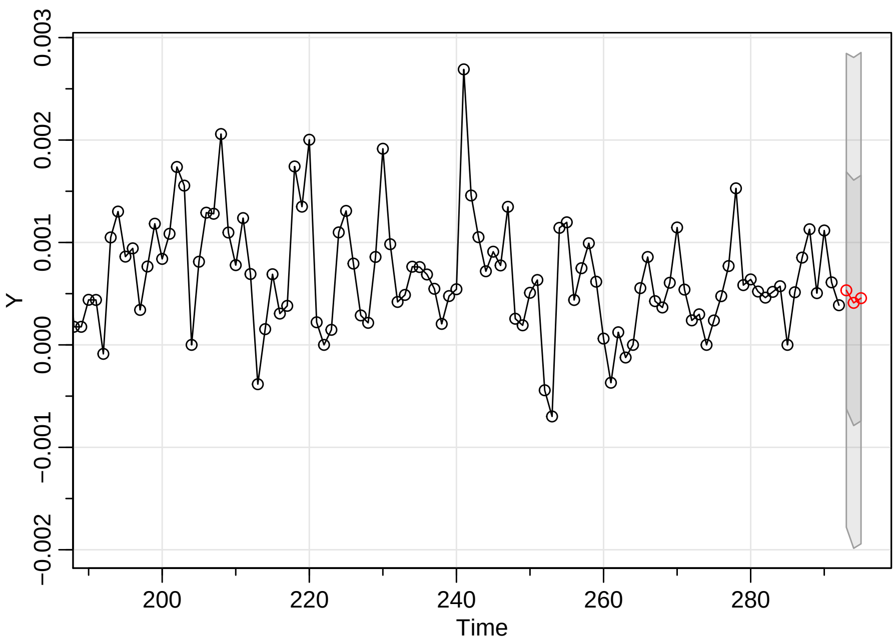


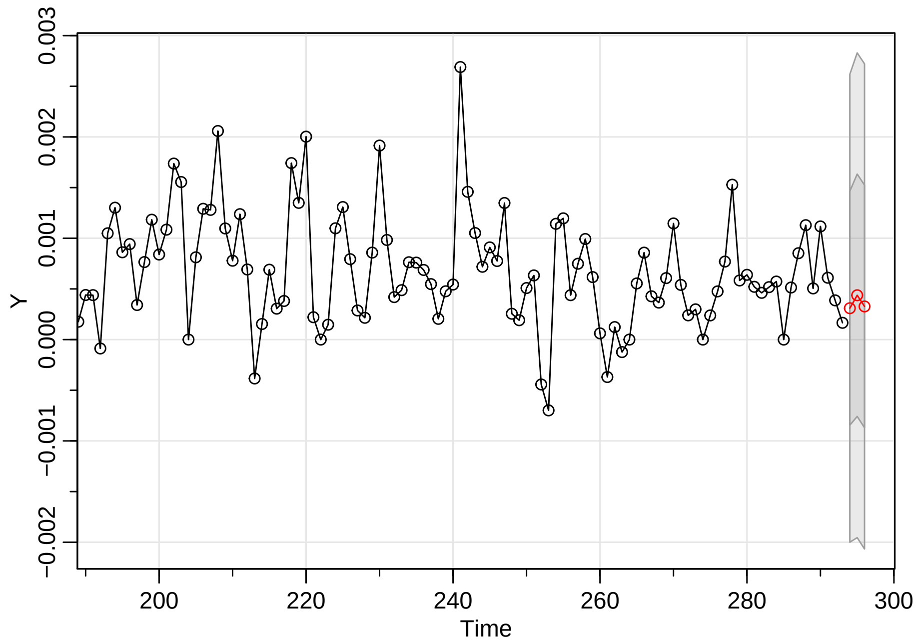
Y
0.003
0.002
0.001
0.000
−0.001
−0.002
200
220
240
260
280
Time

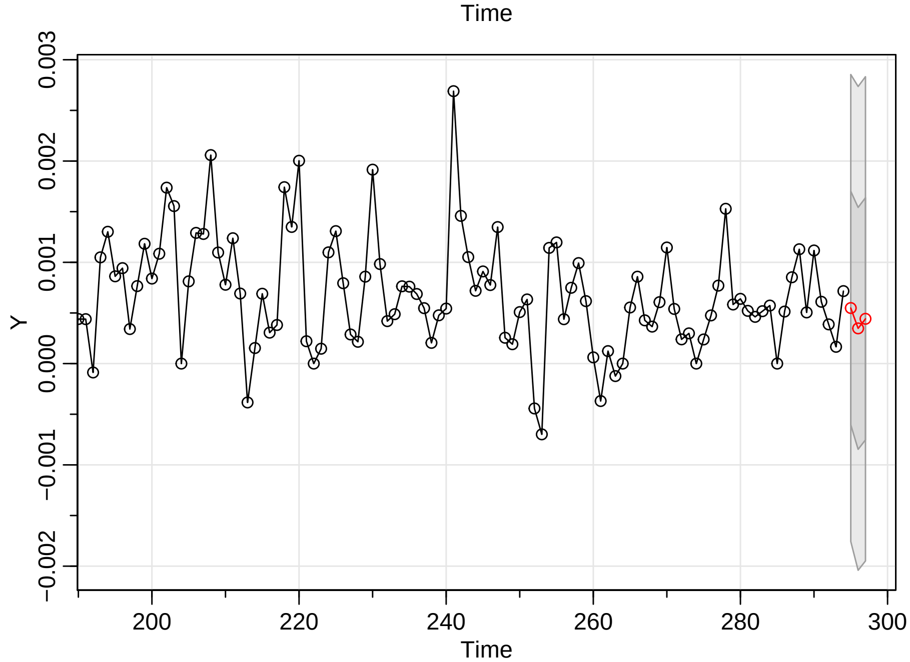


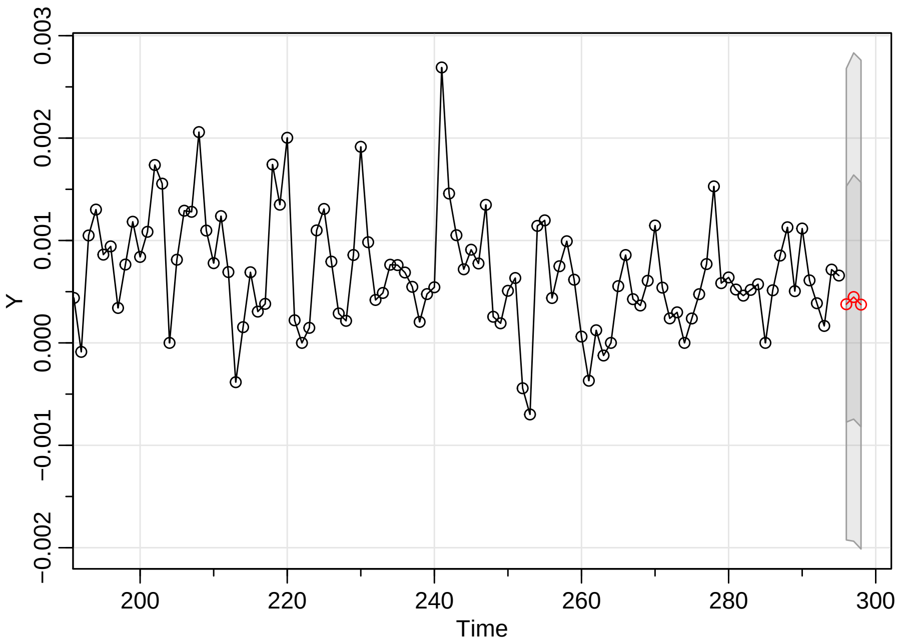
Y
0.003
0.002
0.001
0.000
−0.001
−0.002
200
220
240
260
280
Time


Y
Time
0.003
0.002
0.001
0.000
−0.001
−0.002
200
220
240
260
280

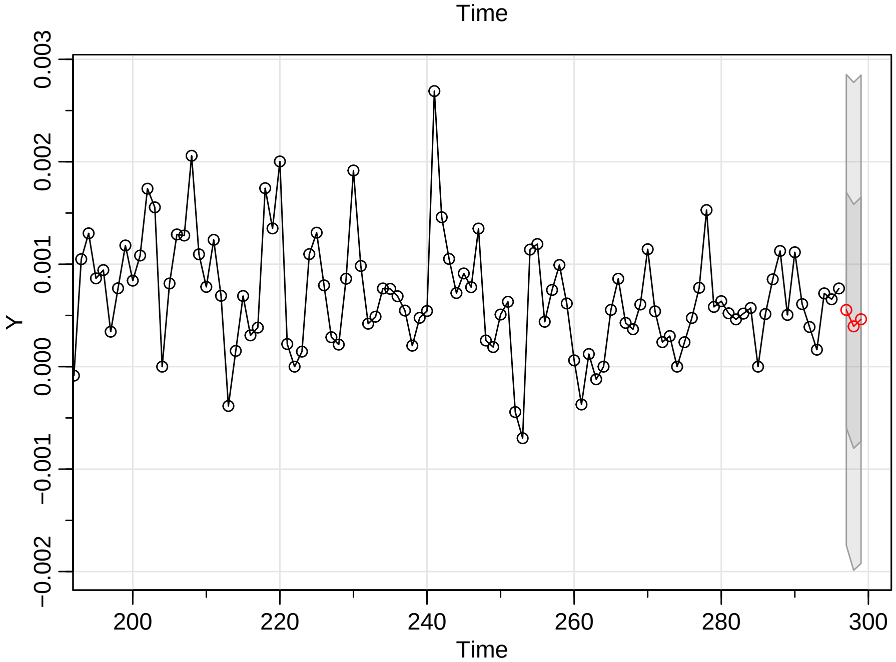

Y
Time
0.003
0.002
0.001
0.000
−0.001
−0.002
200
220
240
260
280


Y
Time
0.003
0.002
0.001
0.000
−0.001
−0.002
200
220
240
260
280
300

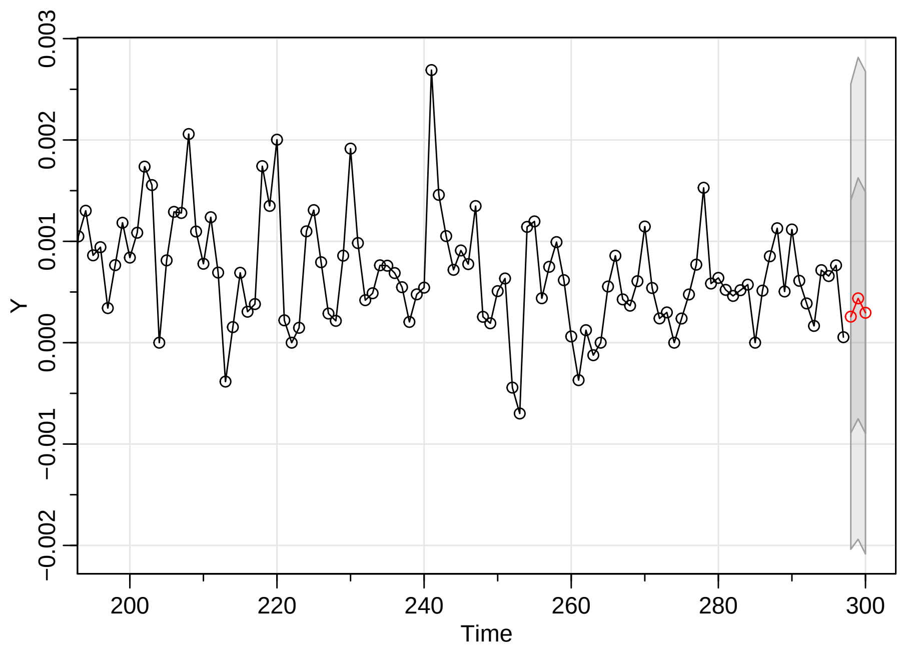

Y
Time
0.003
0.002
0.001
0.000
−0.001
−0.002
200
220
240
260
280
300

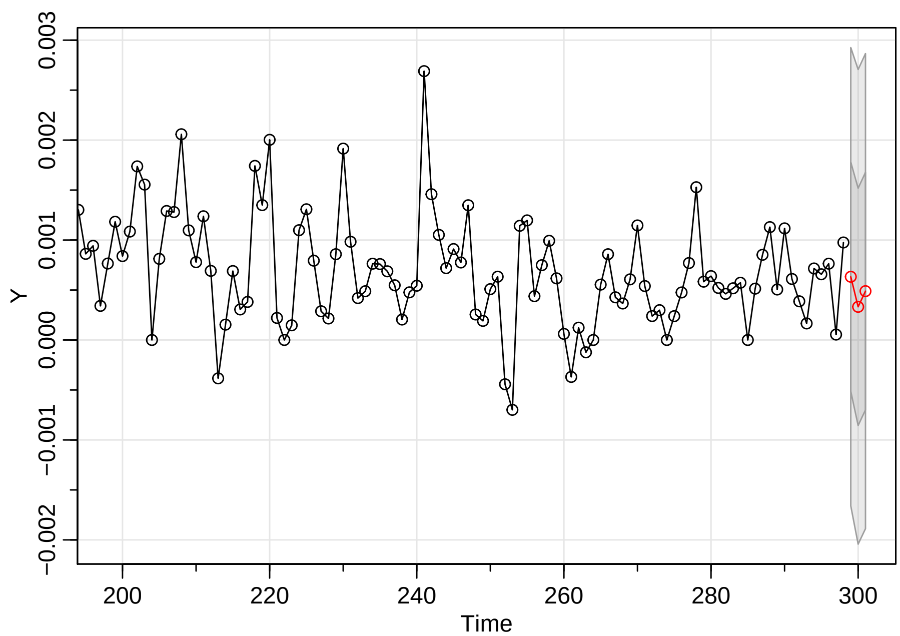


Y
0.003
0.002
0.001
0.000
−0.001
−0.002
200
220
240
260
280
300
Time



Y
0.003
0.002
0.001
0.000
−0.001
−0.002
200
220
240
260
280
300
Time






```
compara<-merge(INFLA,get(paste('PI_', H, sep='')),join='inner')
compara<-data.frame(date=index(compara), coredata(compara))
compara$date<-as.Date(compara$date)
compara<-filter(compara, date>="2007-03-01" & date<="2019-02-01")
compara<-mutate(compara, DIFF =(IPC-IPC.1)^2)
```

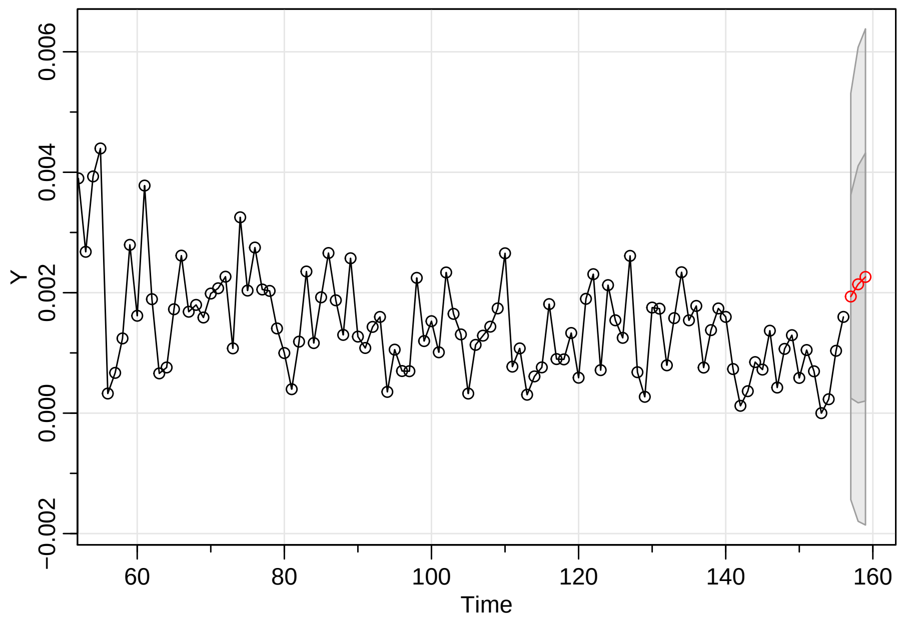
```
ECM_1<-mean(compara$DIFF)
ECM_1
```

```
## [1] 5.891045e-07
```

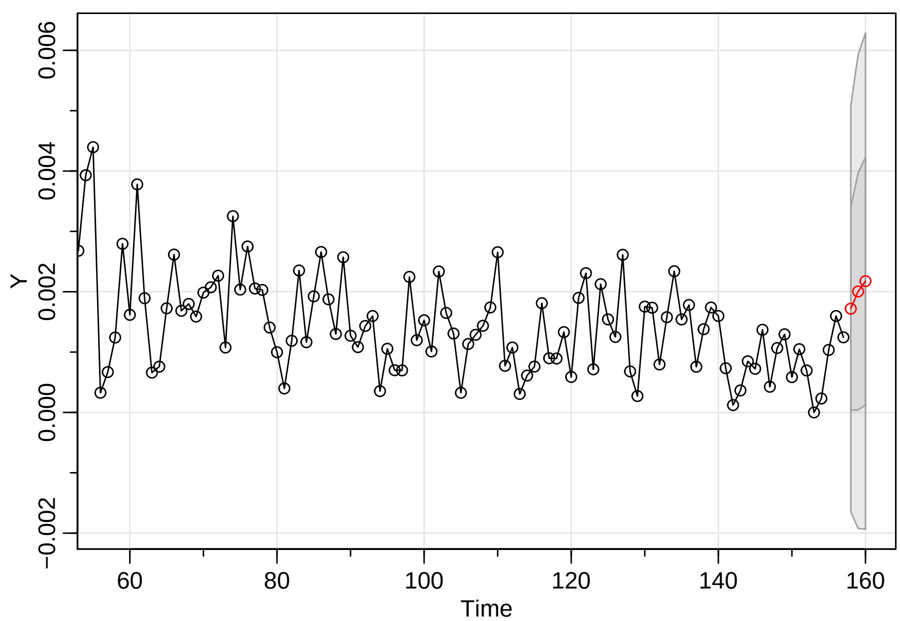
Luego de tener esa específicación podemos comparar su desempeño predictivo con respecto a otra específicación. De acuerdo a la literura un benchmark natural es un simple *random walk* (ver Atkeson and Ohanian (2001) y Meese and Rogoff (1983) entre otros).

En efecto, con un modelo *random walk* podemos calcular para el mismo horizonte de pronóstico (3) su ECM y compararlo con el obtenido por el modelo ARIMA(1,1,2).

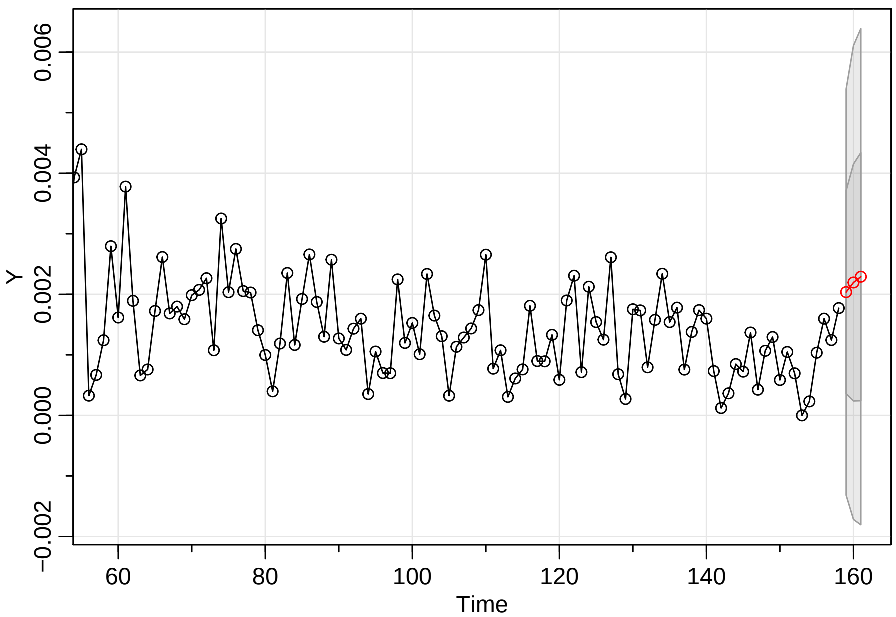
```
DENTRO<- seq(as.Date(FECHA[inicio_estimacion]),
length =final_estimacion+H-inicio_estimacion, by = "months")
assign(paste('PI_', H, sep=''), xts(x=window(INFLA, start=FECHA[inicio_estimacion], end=FECHA[fin
order.by = DENTRO))
Y <-window(INFLA, start=FECHA[inicio_estimacion], end=FECHA[final_estimacion])
for(i in 1:length(FUERA)){
Y_F<-sarima.for(Y,H,1,0,0, xreg=NULL,
                newxreg=NULL, plot= FALSE)
dates_out<-as.Date(FECHA[final_estimacion+i+H-1])
Y_F_P<-xts(x=Y_F$pred[H], order.by = dates_out)
DENTRO<-seq(as.Date(FECHA[inicio_estimacion]),
length =final_estimacion+1+i-inicio_estimacion, by = "months")
Y <-window(INFLA, start=FECHA[inicio_estimacion], end=FECHA[final_estimacion+i])
assign(paste('PI_', H, sep=''),rbind(get(paste('PI_', H, sep='')),
Y_F_P))
}
```



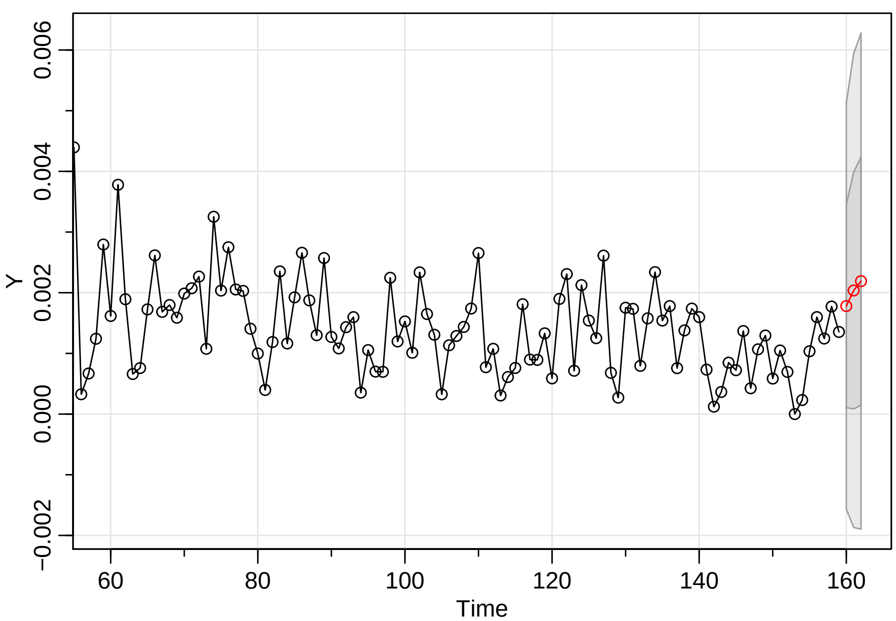
0.006
0.004
0.002
0.000
−0.002
Y
60
80
100
120
140
160
Time



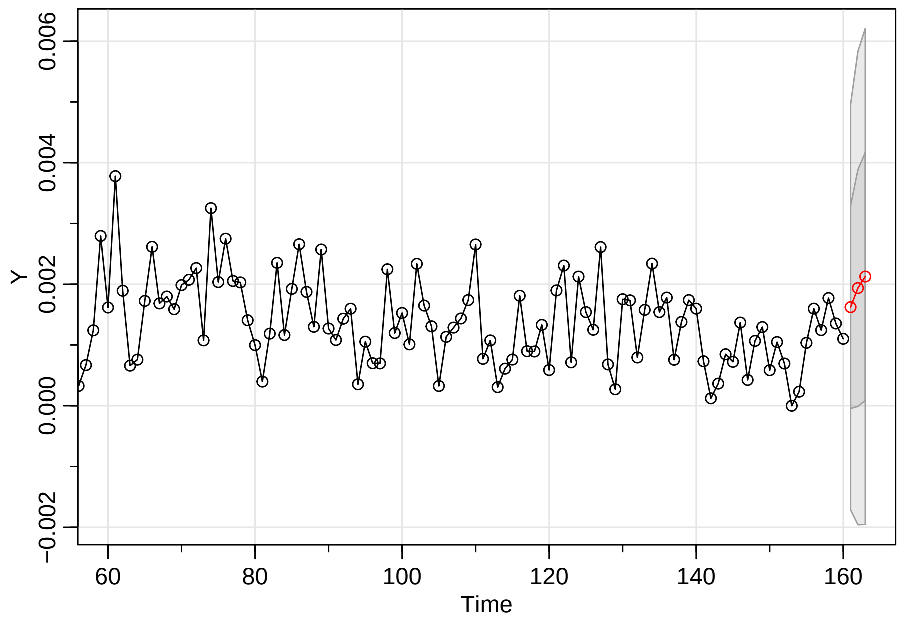
0.006
0.004
0.002
0.000
−0.002
Y
60
80
100
120
140
160
Time

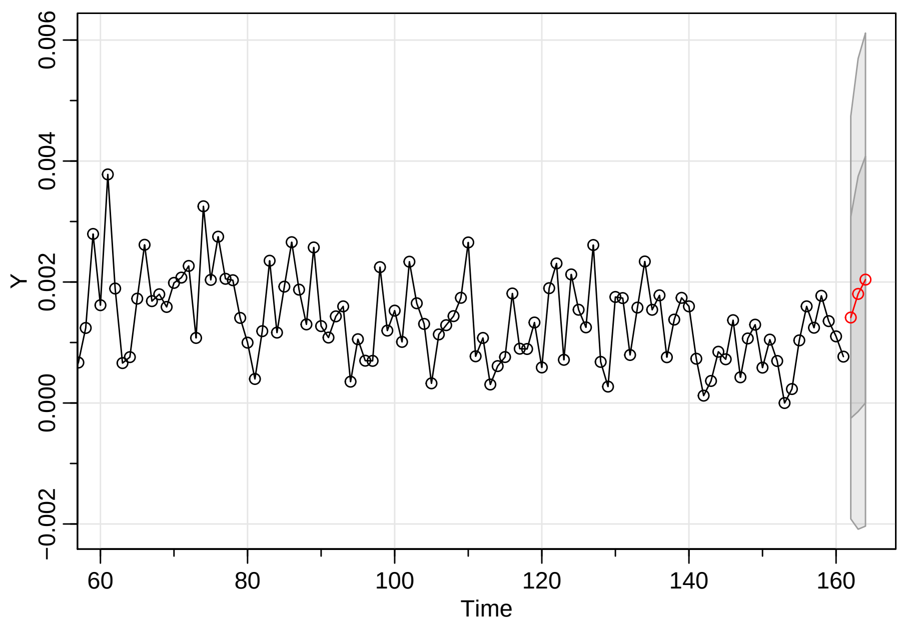


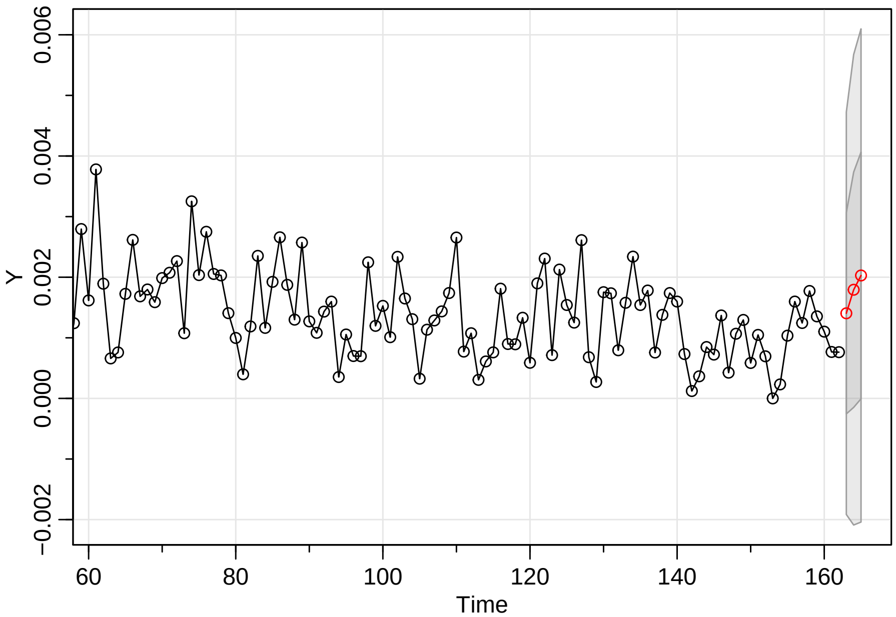
Y
Time
0.006
0.004
0.002
0.000
−0.002
60
80
100
120
140
160

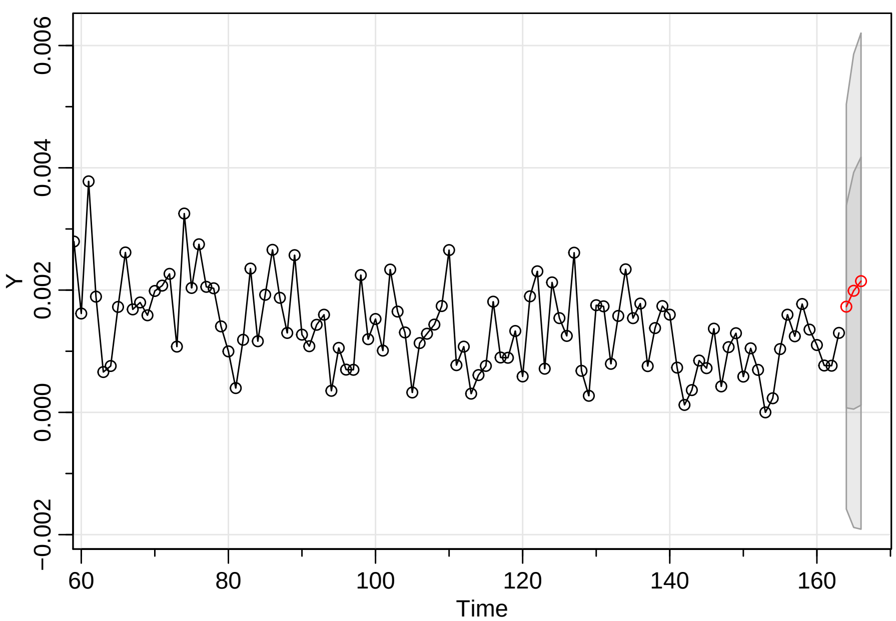


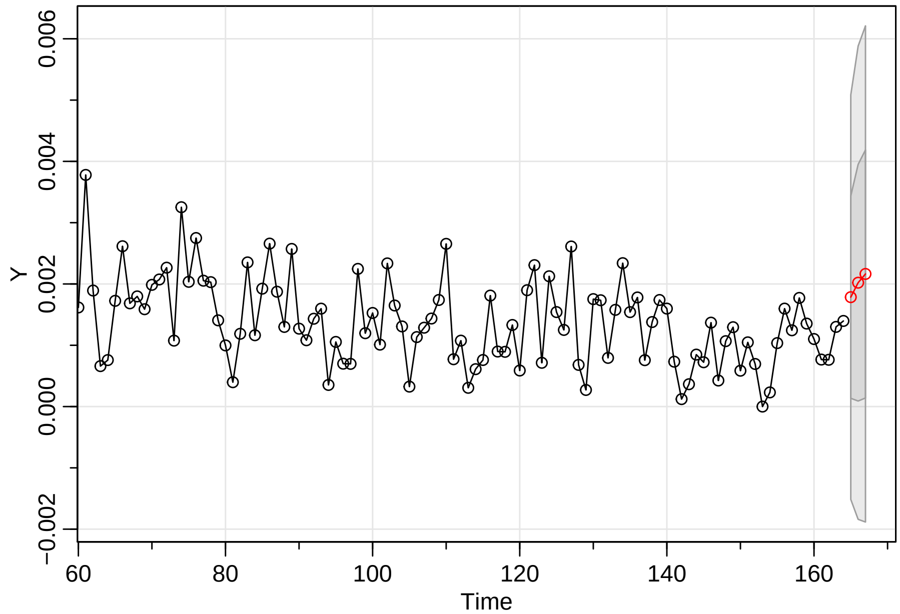
Y
Time
0.006
0.004
0.002
0.000
−0.002
60
80
100
120
140
160

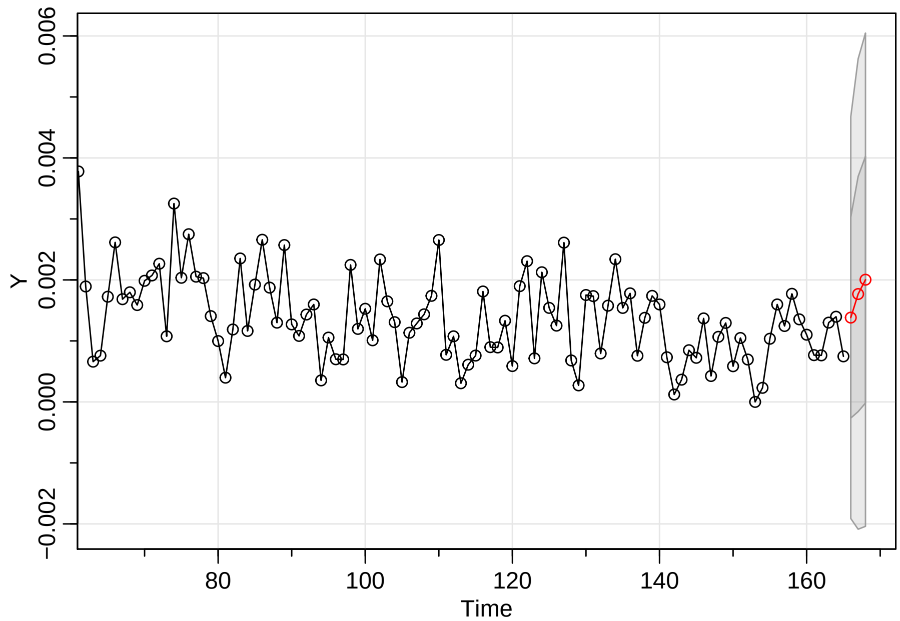

Y
0.006
0.004
0.002
0.000
−0.002
60
80
100
120
140
160
Time

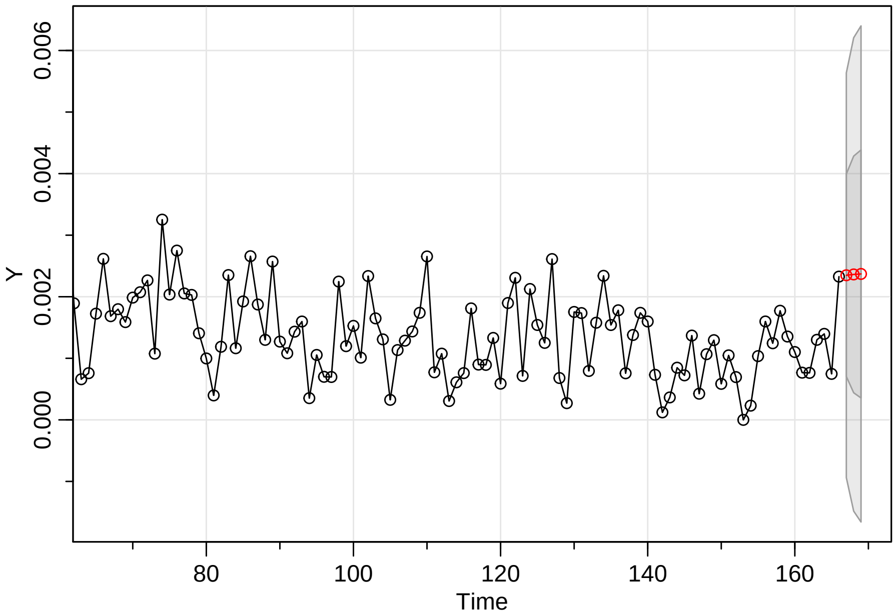

Y
0.006
0.004
0.002
0.000
−0.002
60
80
100
120
140
160
Time


Y
Time
0.006
0.004
0.002
0.000
−0.002
60
80
100
120
140
160

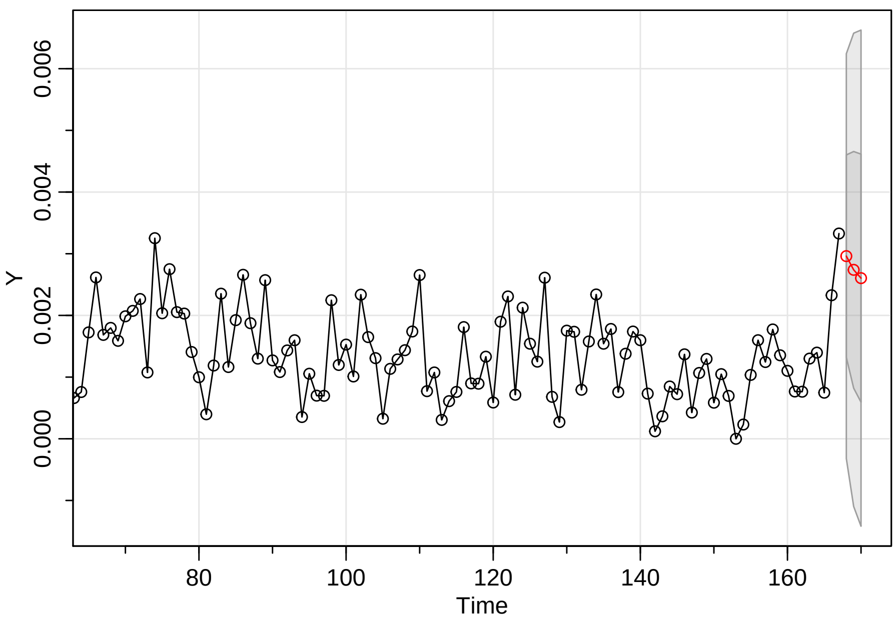

Y
Time
0.006
0.004
0.002
0.000
−0.002
60
80
100
120
140
160


Y
Time
0.006
0.004
0.002
0.000
−0.002
60
80
100
120
140
160

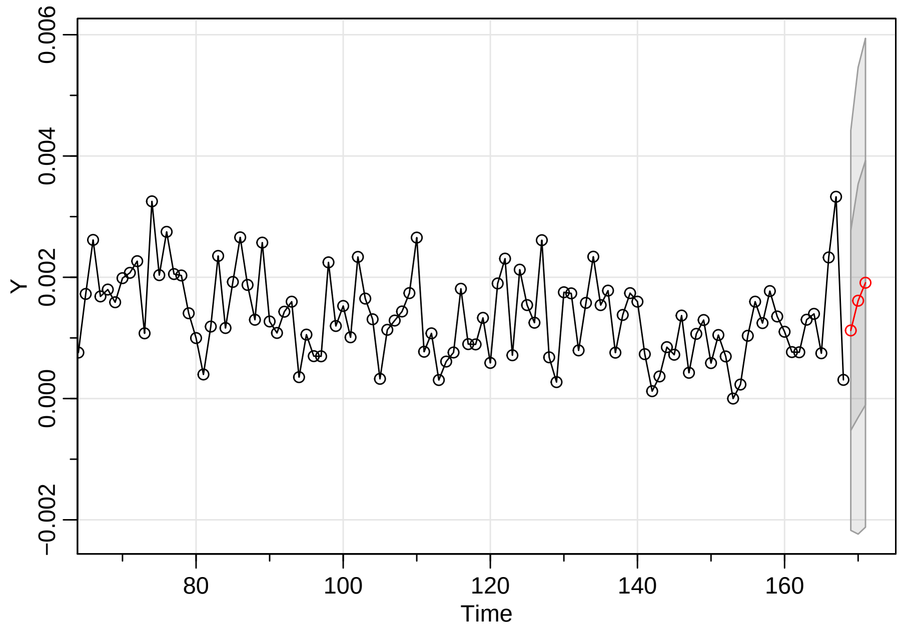

Y
Time
0.006
0.004
0.002
0.000
−0.002
80
100
120
140
160



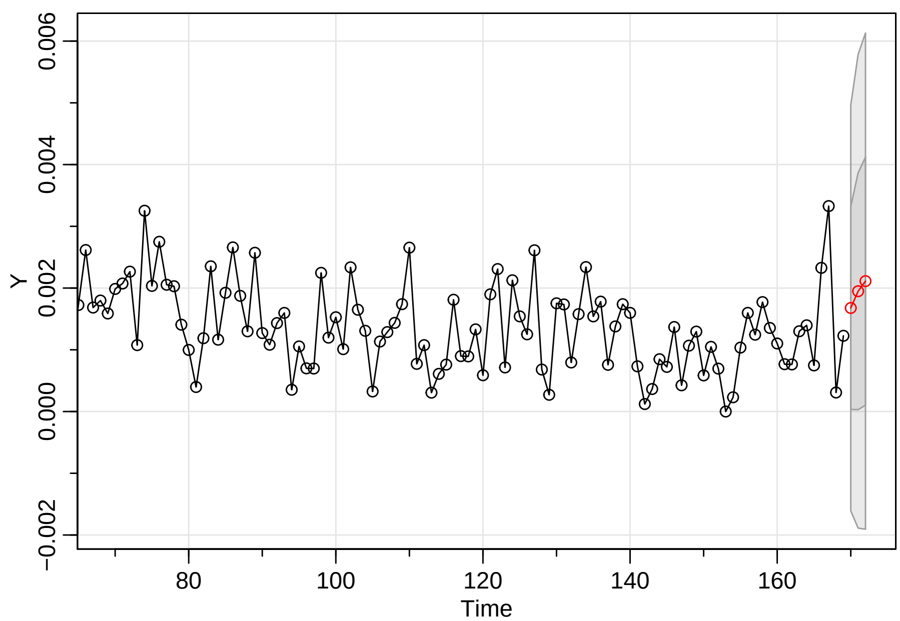
Y
0.006
0.004
0.002
0.000
80
100
120
140
160
Time



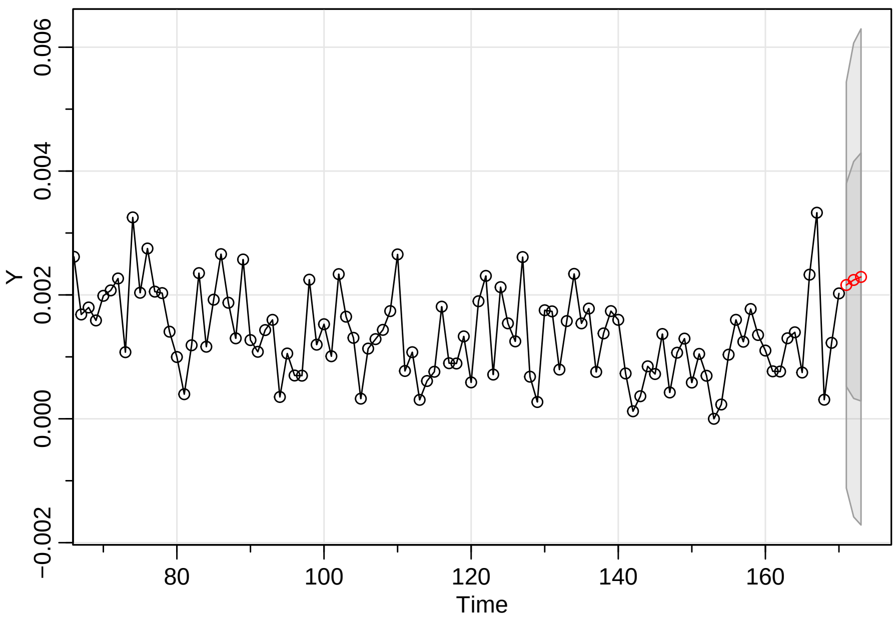
Y
0.006
0.004
0.002
0.000
80
100
120
140
160
Time

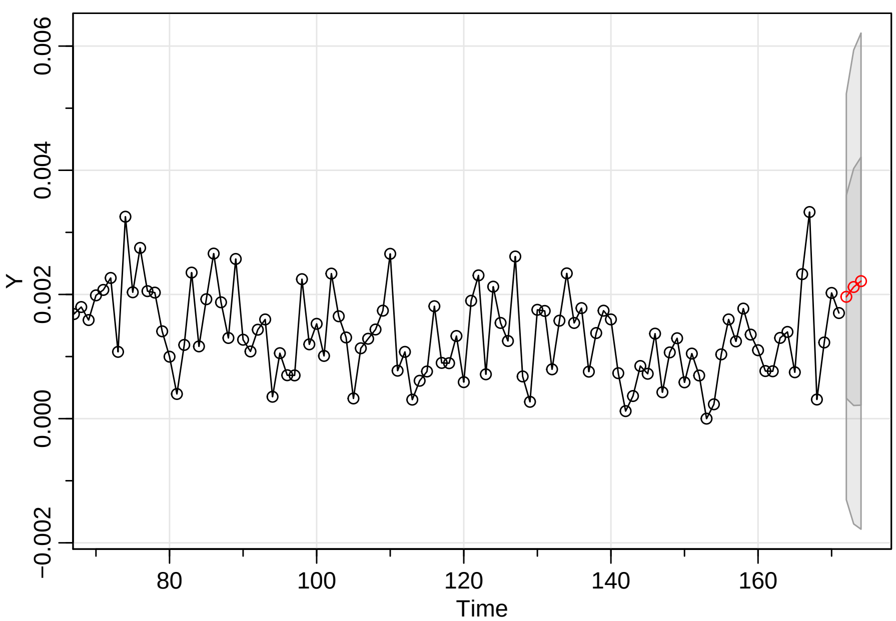


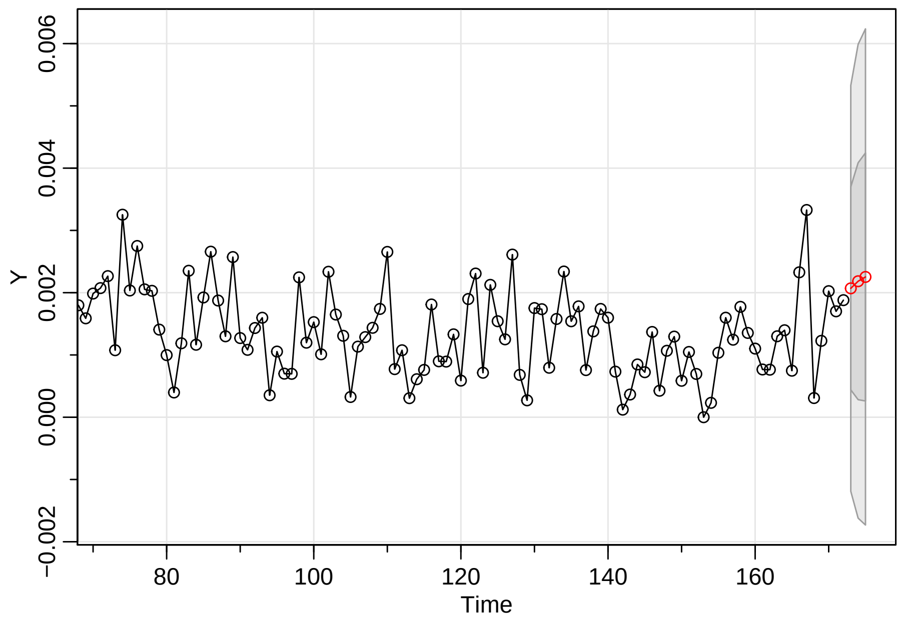
Y
0.006
0.004
0.002
0.000
−0.002
80
100
120
140
160
Time



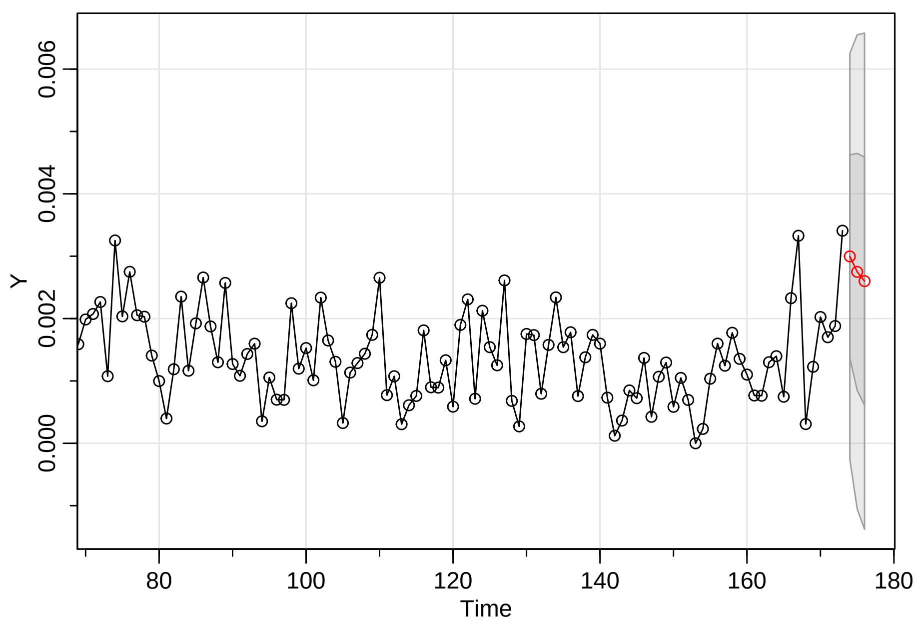
Y
0.006
0.004
0.002
0.000
−0.002
80
100
120
140
160
Time


Y
Time

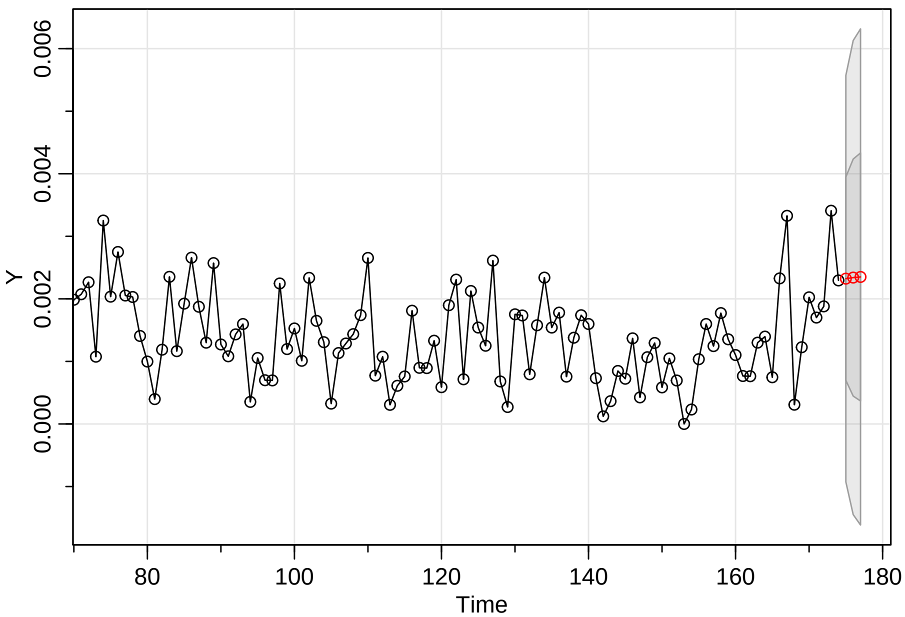

Y
Time

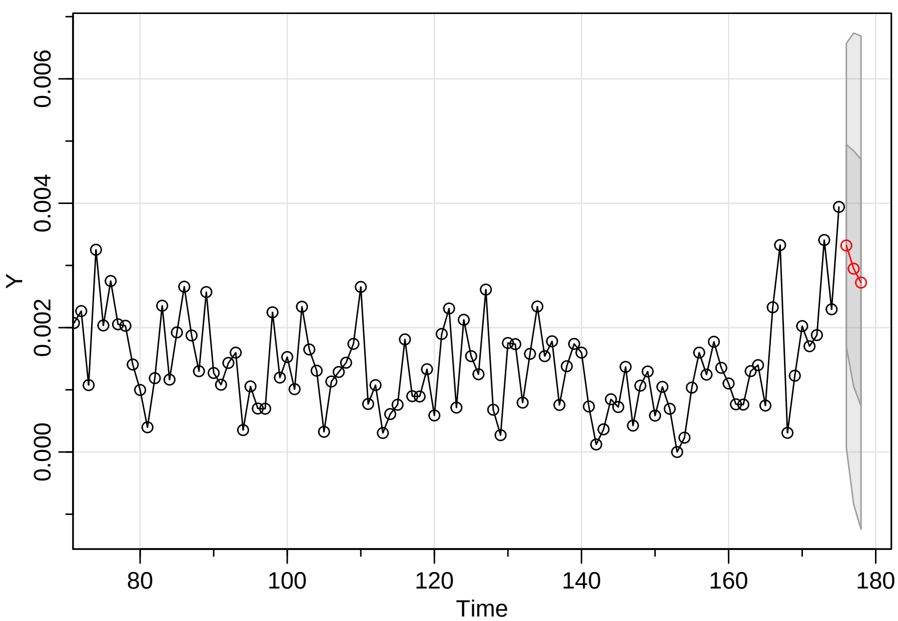


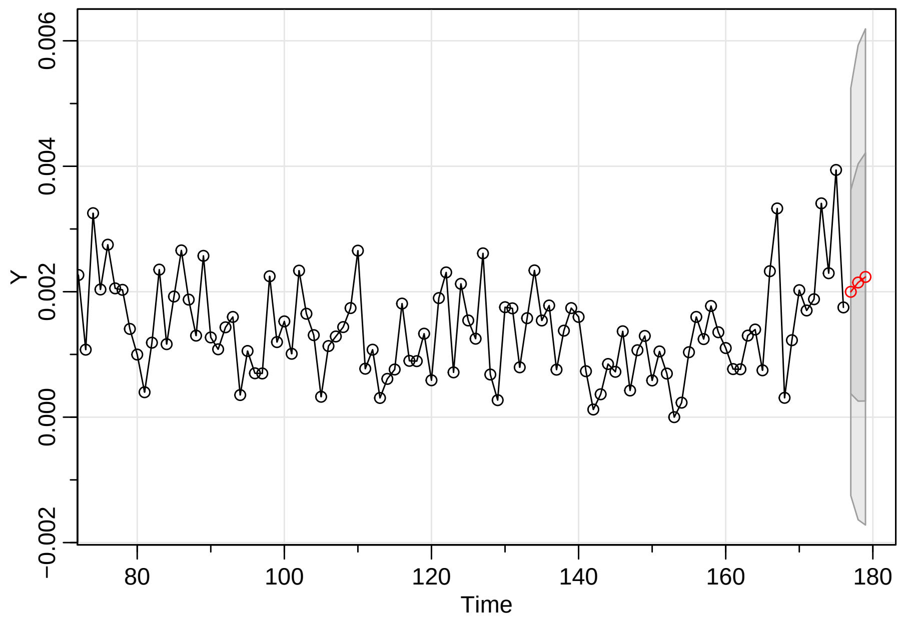
Y
0.006
0.004
0.002
0.000
−0.002
80
100
120
140
160
Time

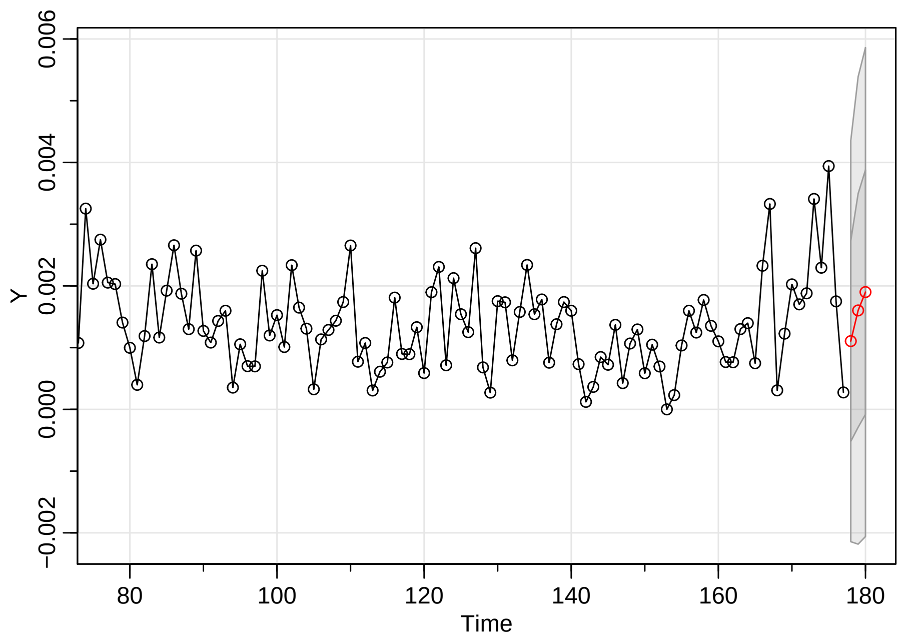


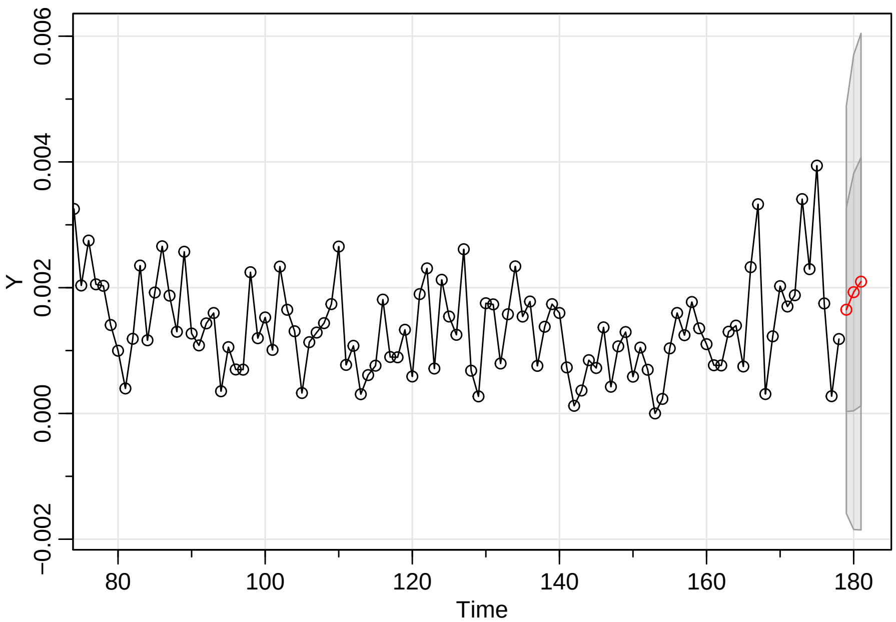
Y
0.006
0.004
0.002
0.000
80
100
120
140
160
180
Time

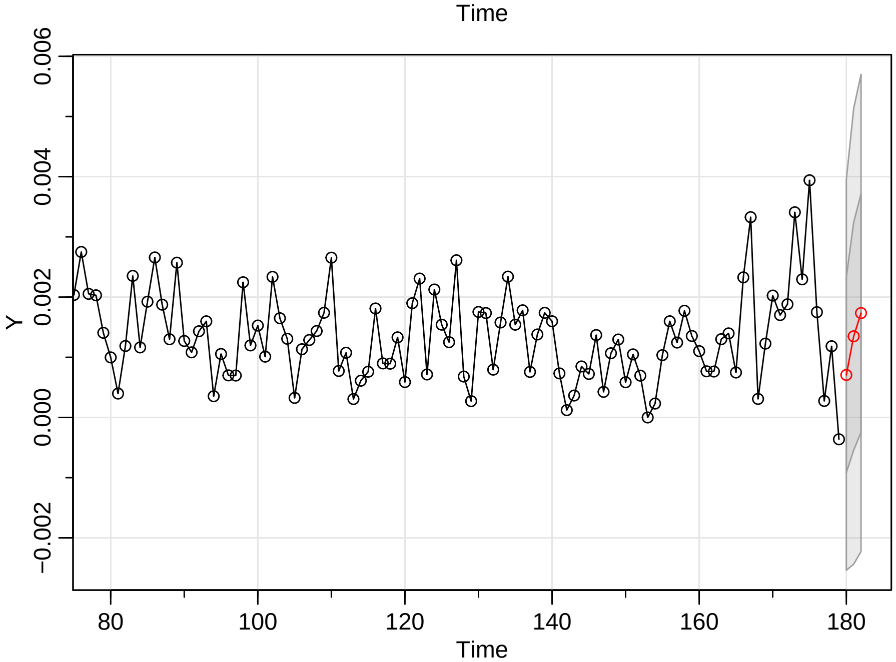

Y
0.000
0.002
0.004
0.006
80
100
120
140
160
180
Time

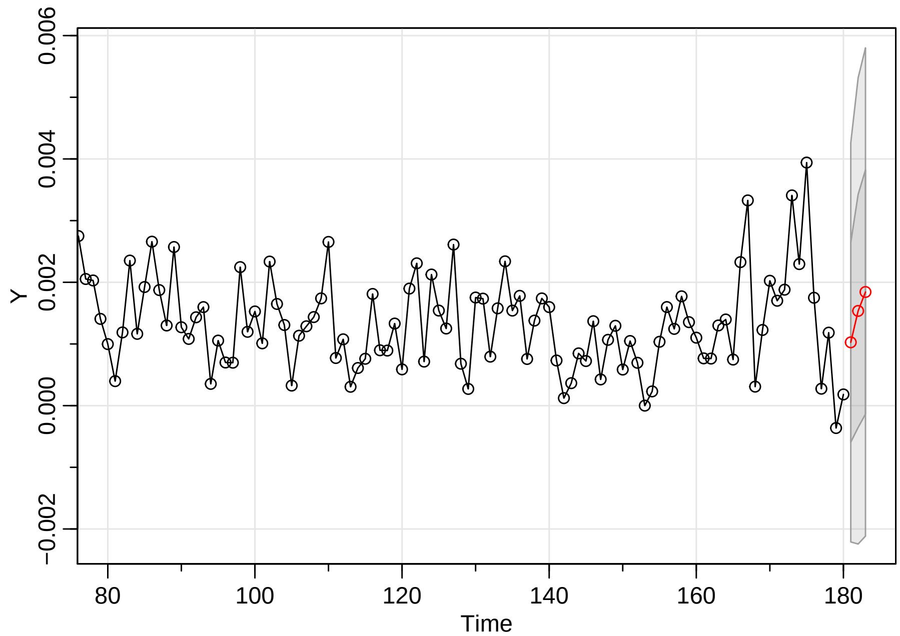

Y
0.000
0.002
0.004
0.006
80
100
120
140
160
180
Time


Y
−0.002
0.000
0.002
0.004
0.006
80
100
120
140
160
180
Time

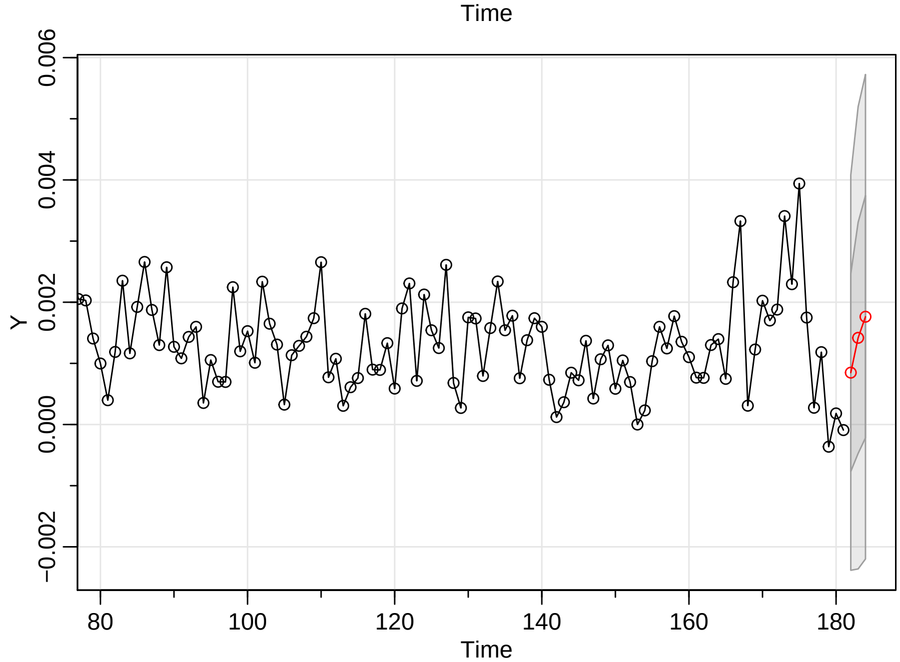

Y
−0.002
0.000
0.002
0.004
0.006
80
100
120
140
160
180
Time


Y
Time
0.006
0.004
0.002
0.000
−0.002
80
100
120
140
160
180

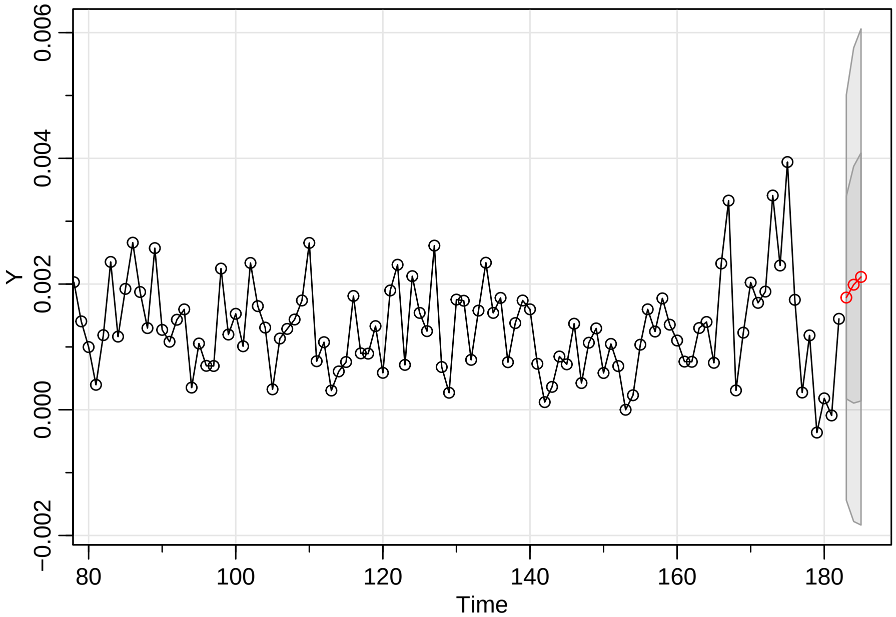

Y
Time
0.006
0.004
0.002
0.000
−0.002
80
100
120
140
160
180



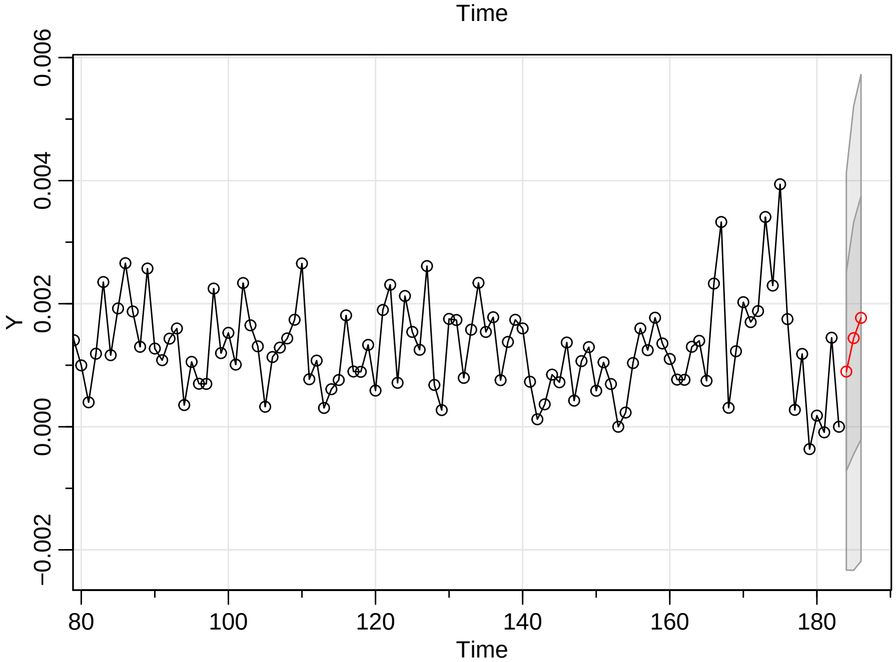
0.006
0.004
0.002
0.000
−0.002
Y
80
100
120
140
160
180
Time



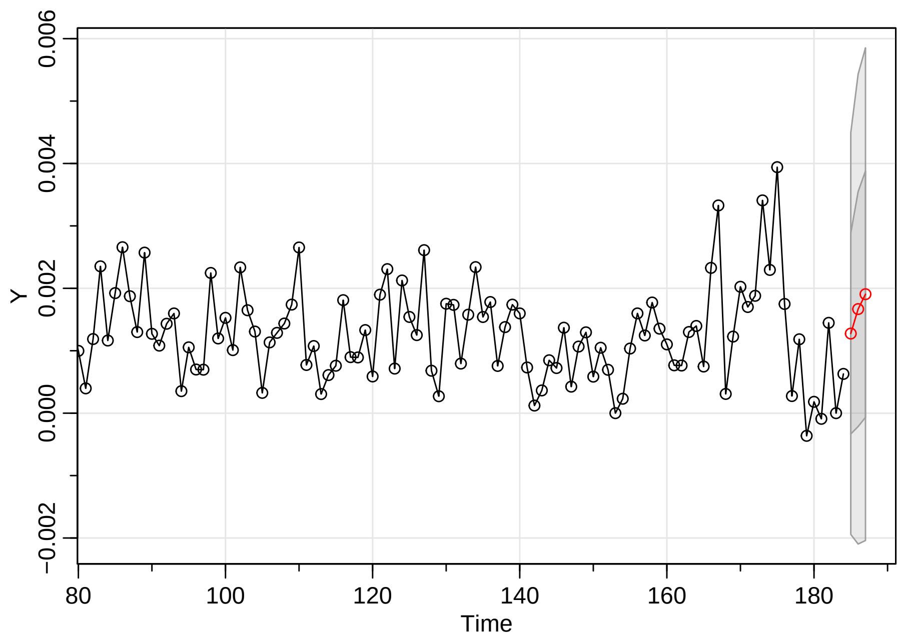
0.006
0.004
0.002
0.000
−0.002
Y
80
100
120
140
160
180
Time



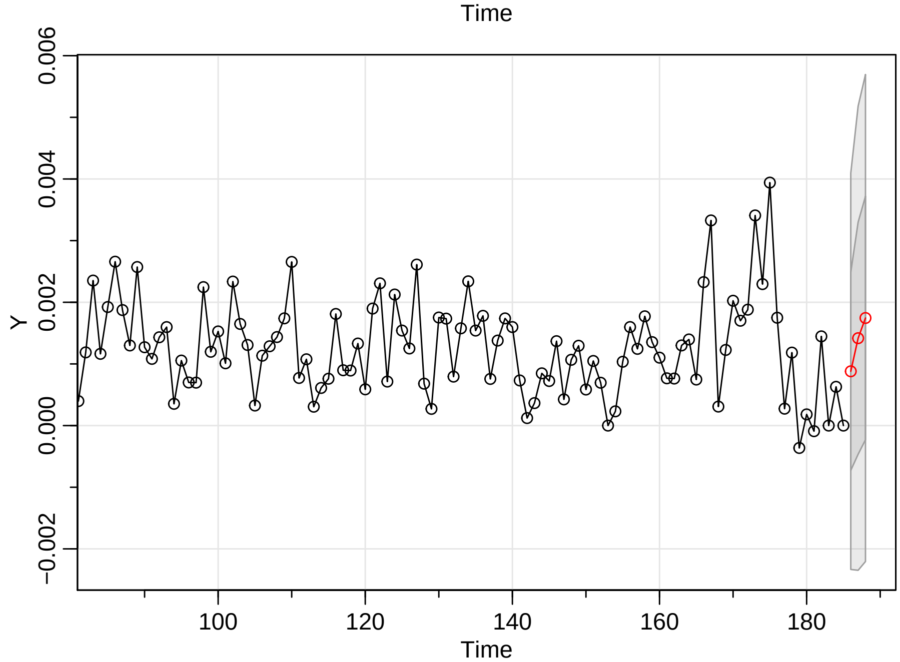
Y
Time
0.006
0.004
0.002
0.000
−0.002
80
100
120
140
160
180



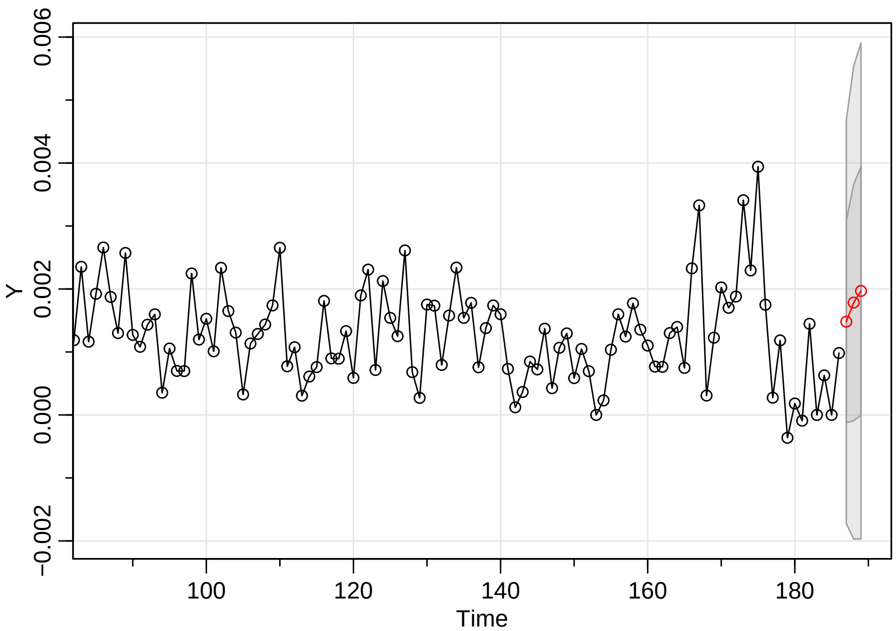
Y
Time
0.006
0.004
0.002
0.000
−0.002
80
100
120
140
160
180

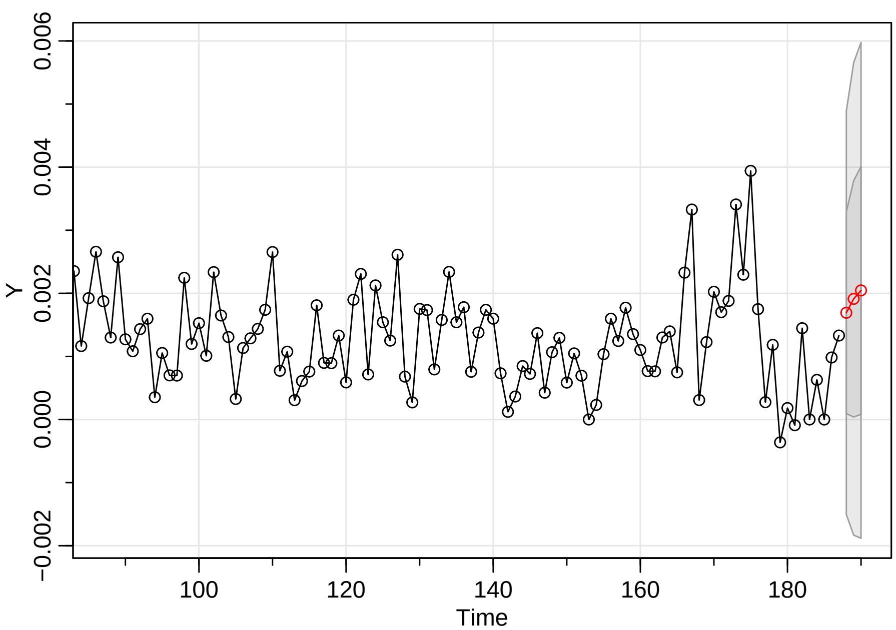


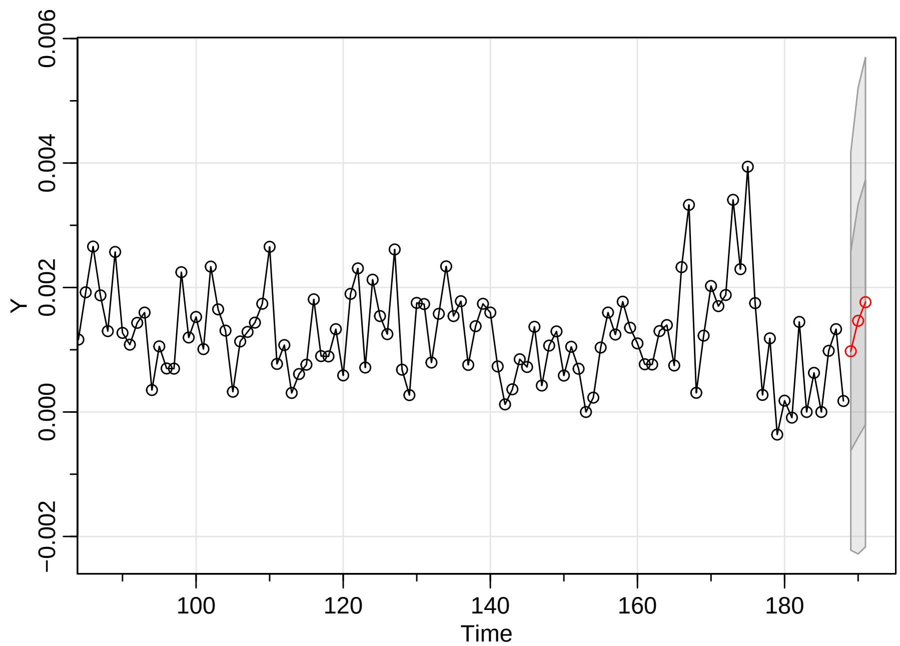
Y
0.006
0.004
0.002
0.000
−0.002
80
100
120
140
160
180
Time

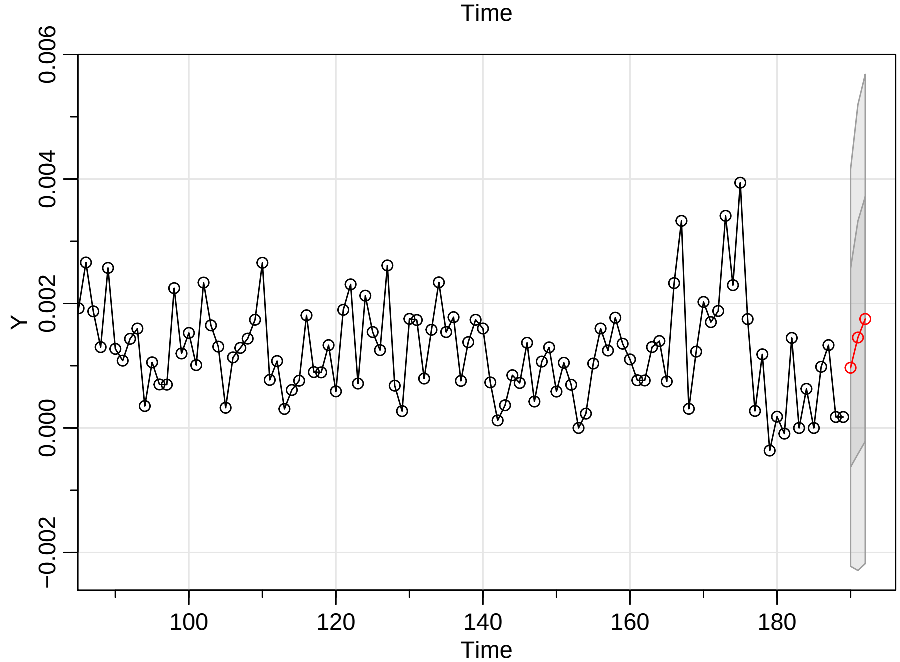


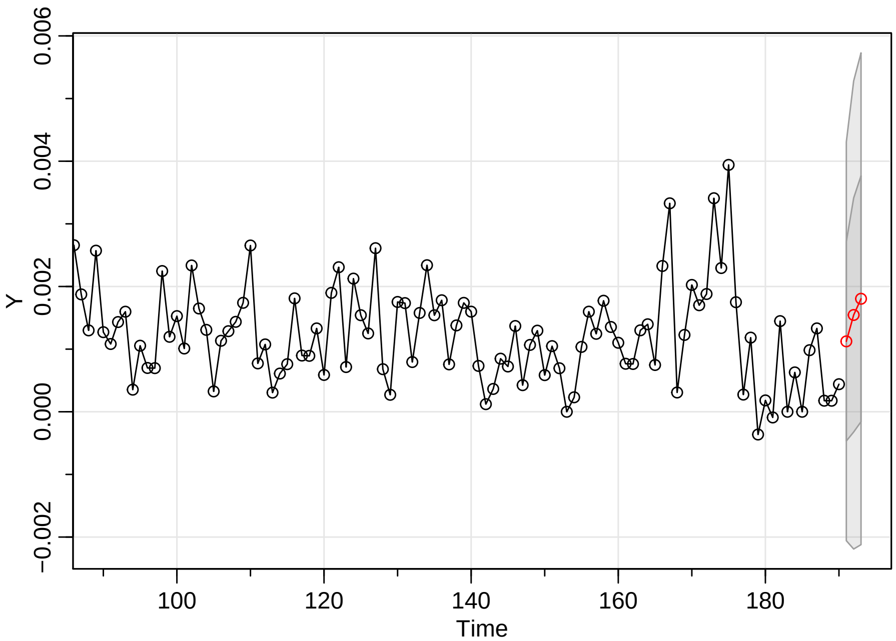
Y
0.006
0.004
0.002
0.000
−0.002
100
120
140
160
180
Time

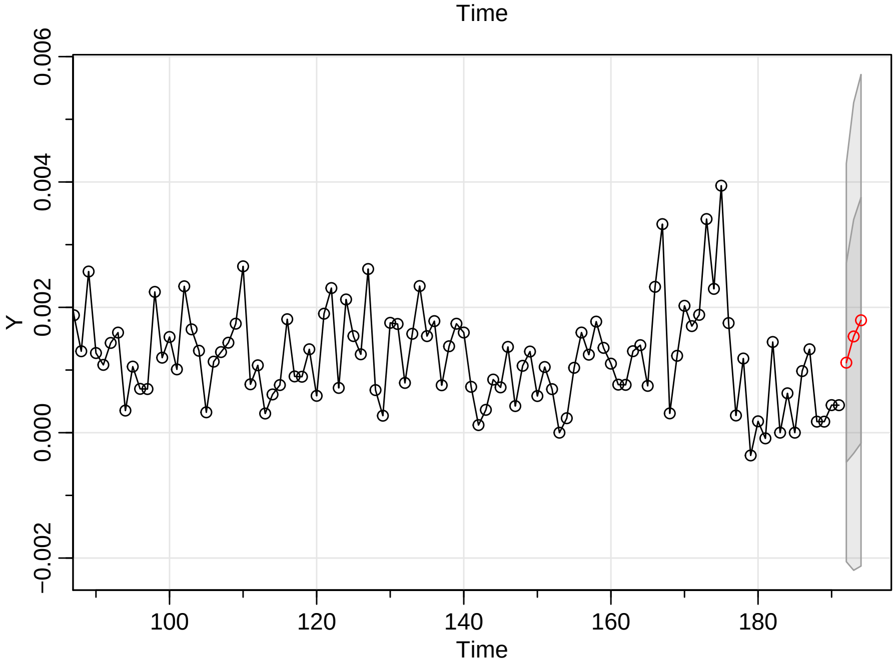

Y
Time
−0.002
0.000
0.002
0.004
0.006
100
120
140
160
180

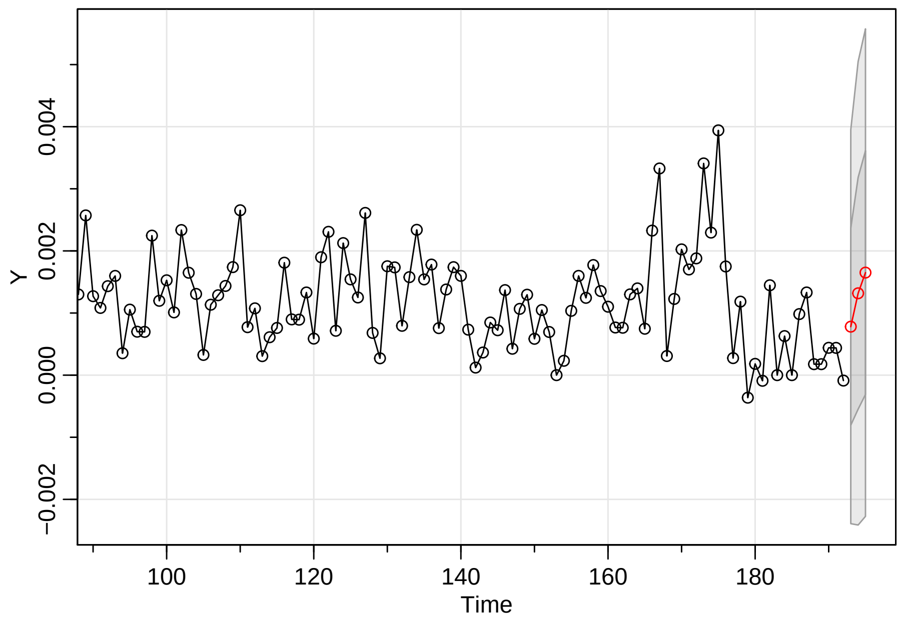

Y
Time
−0.002
0.000
0.002
0.004
0.006
100
120
140
160
180


0.006
0.004
0.002
0.000
−0.002
Y
100
120
140
160
180
Time

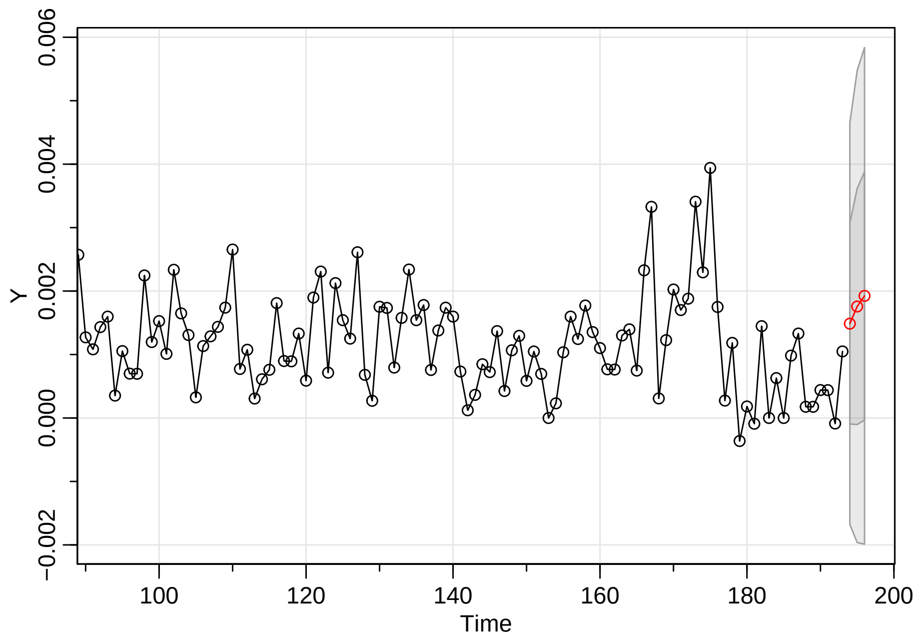

0.006
0.004
0.002
0.000
−0.002
Y
100
120
140
160
180
Time


Y
Time
0.006
0.004
0.002
0.000
−0.002
100
120
140
160
180

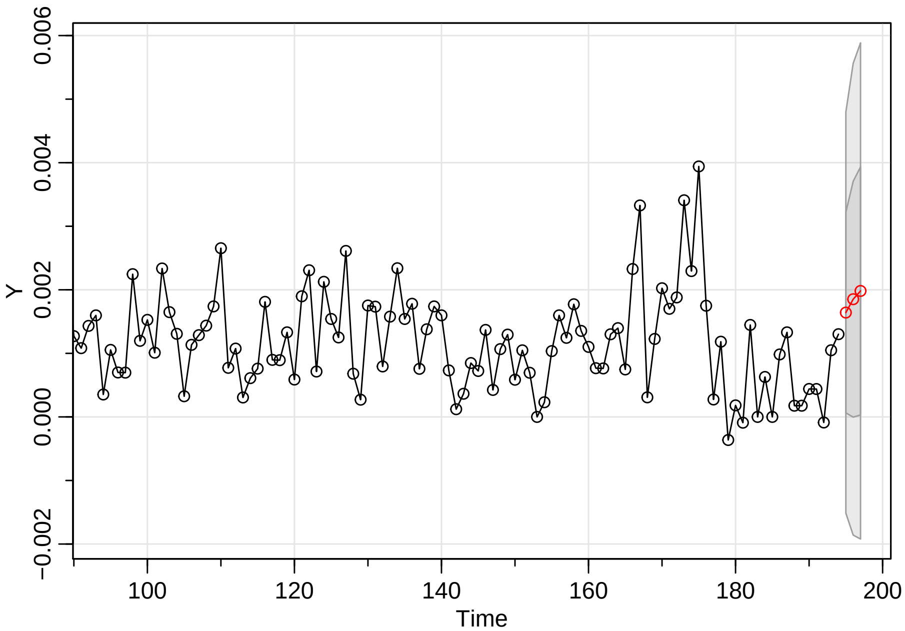

Y
Time
0.006
0.004
0.002
0.000
−0.002
100
120
140
160
180



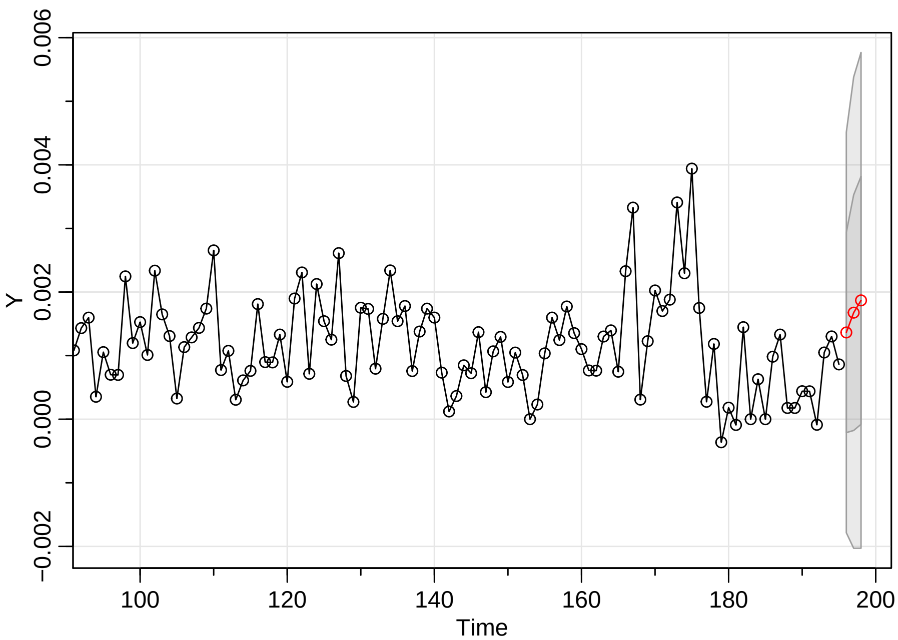
Y
Time
0.004
0.002
0.000
−0.002
100
120
140
160
180



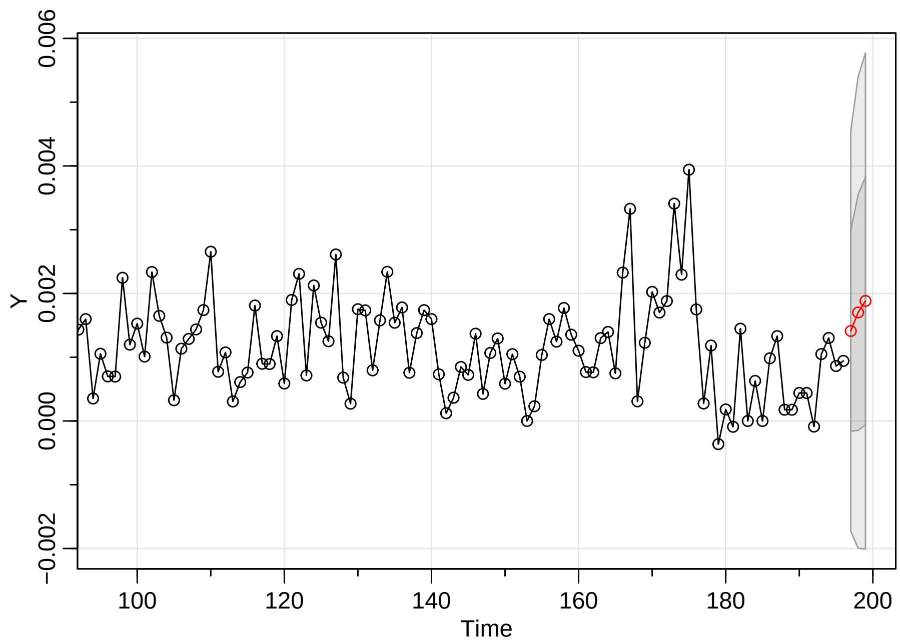
Y
Time
0.006
0.004
0.002
0.000
−0.002
100
120
140
160
180
200



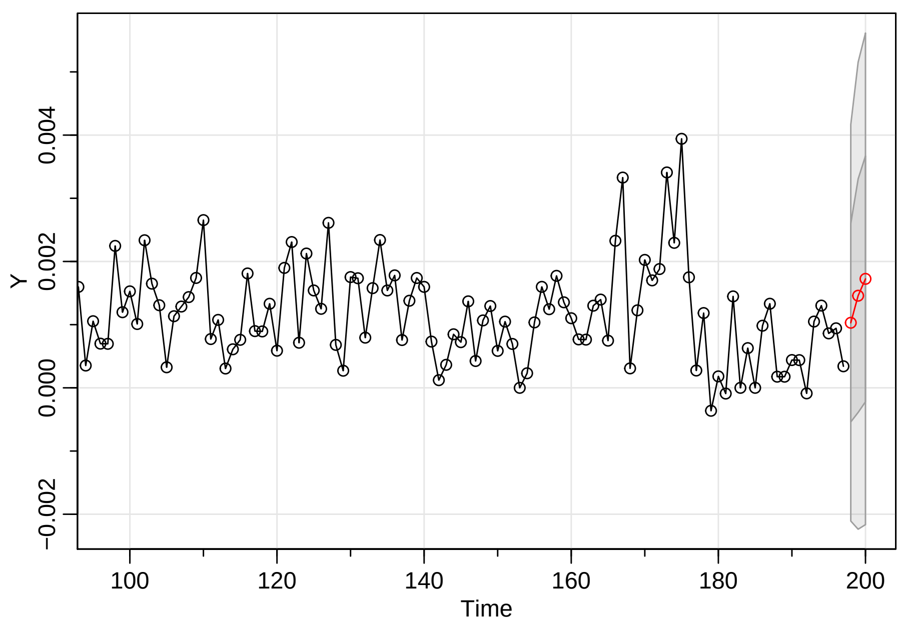
Y
Time
0.006
0.004
0.002
0.000
−0.002
100
120
140
160
180
200



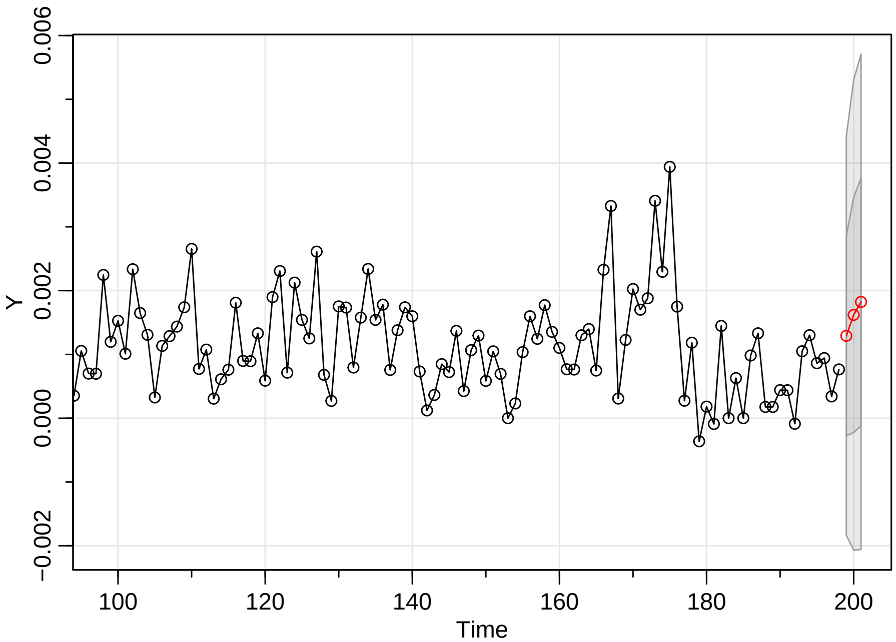
Y
Time
0.006
0.004
0.002
0.000
−0.002
100
120
140
160
180
200

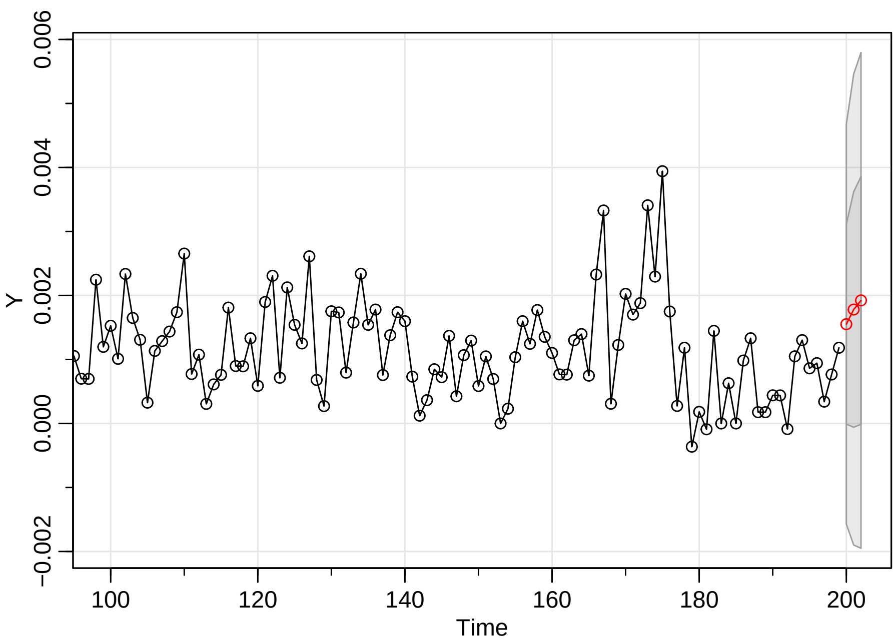


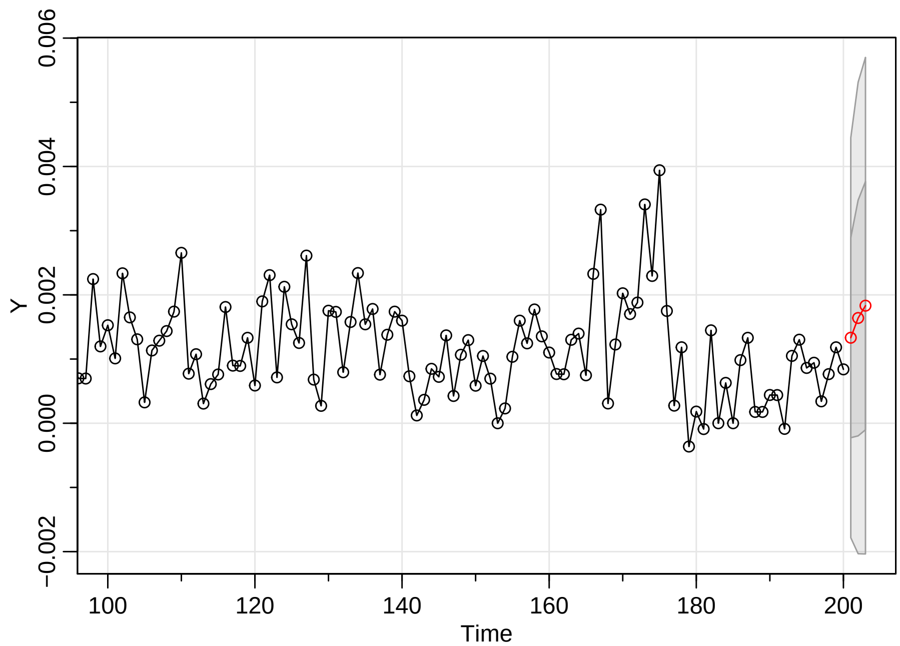
Y
0.006
0.004
0.002
0.000
−0.002
100
120
140
160
180
200
Time

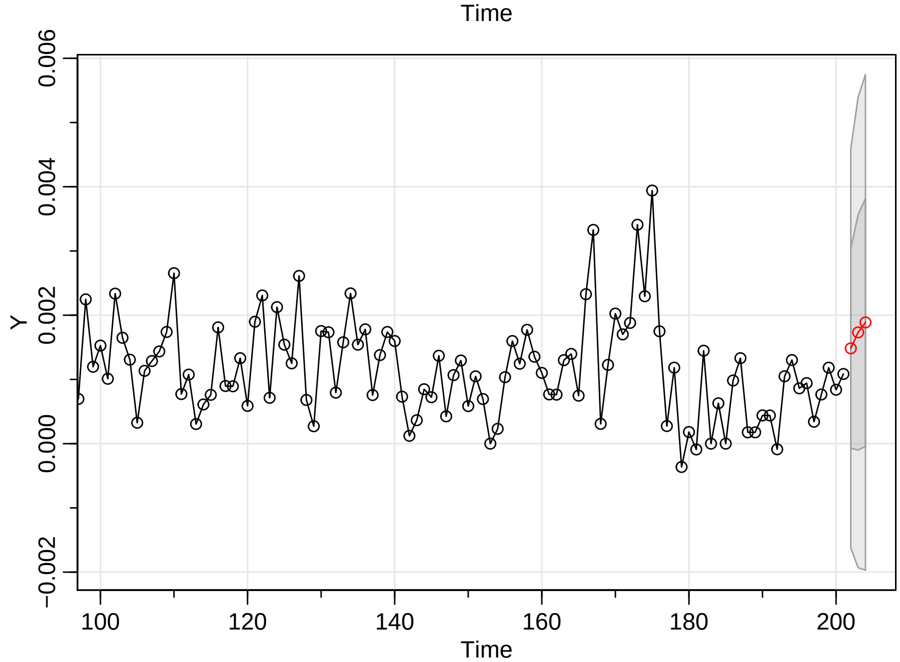


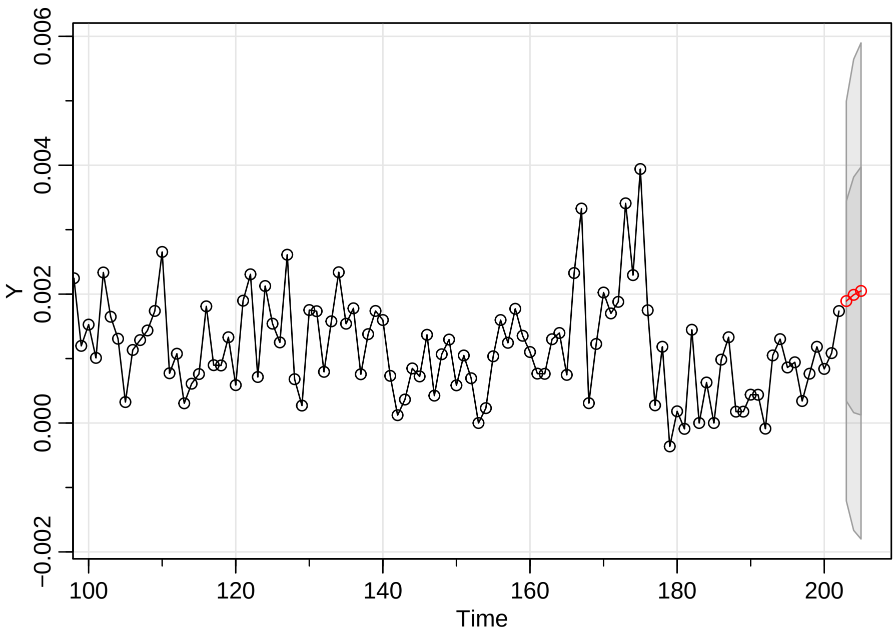
Y
0.004
0.002
0.000
−0.002
100
120
140
160
180
200
Time

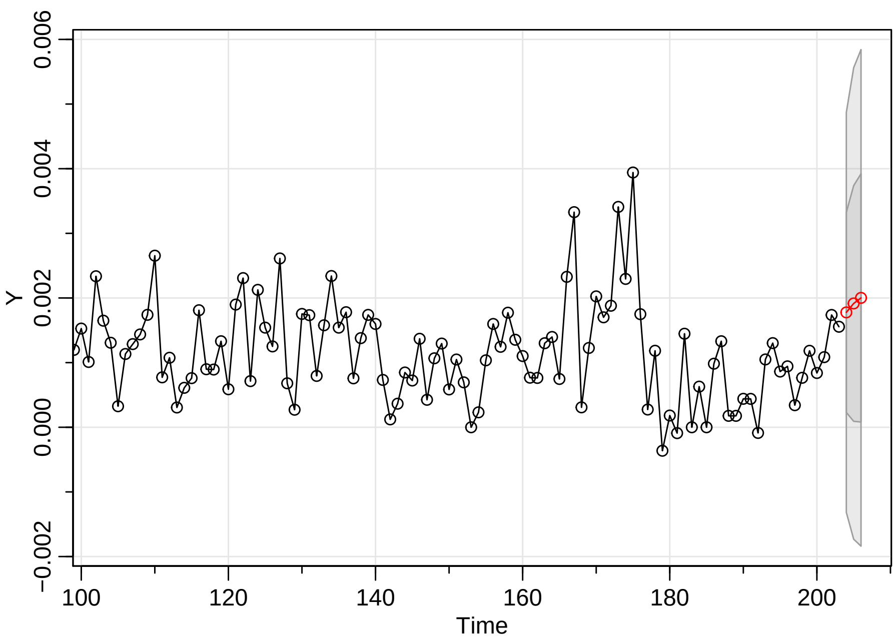

Y
Time
0.006
0.004
0.002
0.000
−0.002
100
120
140
160
180
200

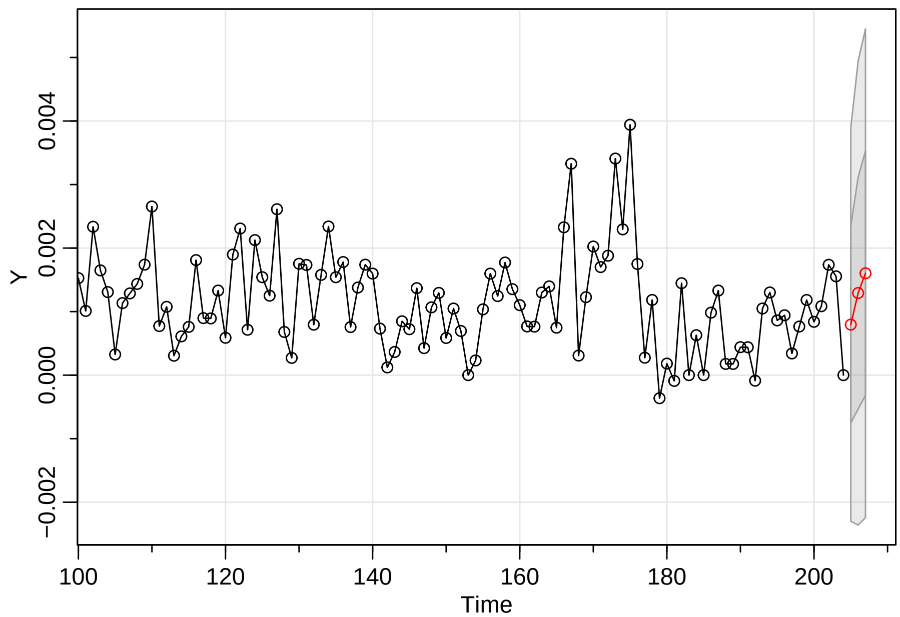

Y
Time
0.006
0.004
0.002
0.000
−0.002
100
120
140
160
180
200


Y
0.006
0.004
0.002
0.000
−0.002
100
120
140
160
180
200
Time

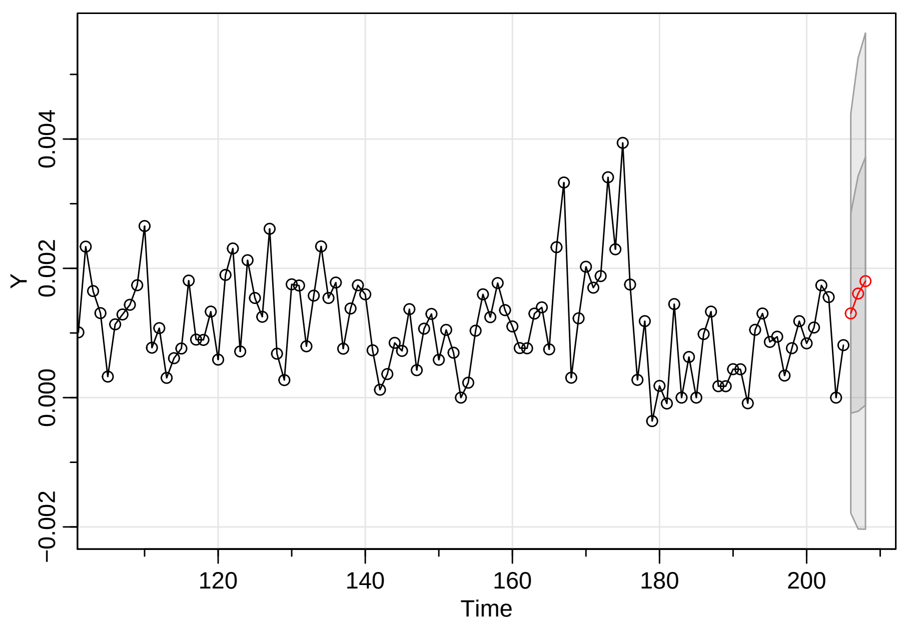

Y
0.006
0.004
0.002
0.000
−0.002
100
120
140
160
180
200
Time


Y
0.006
0.004
0.002
0.000
−0.002
100
120
140
160
180
200
Time

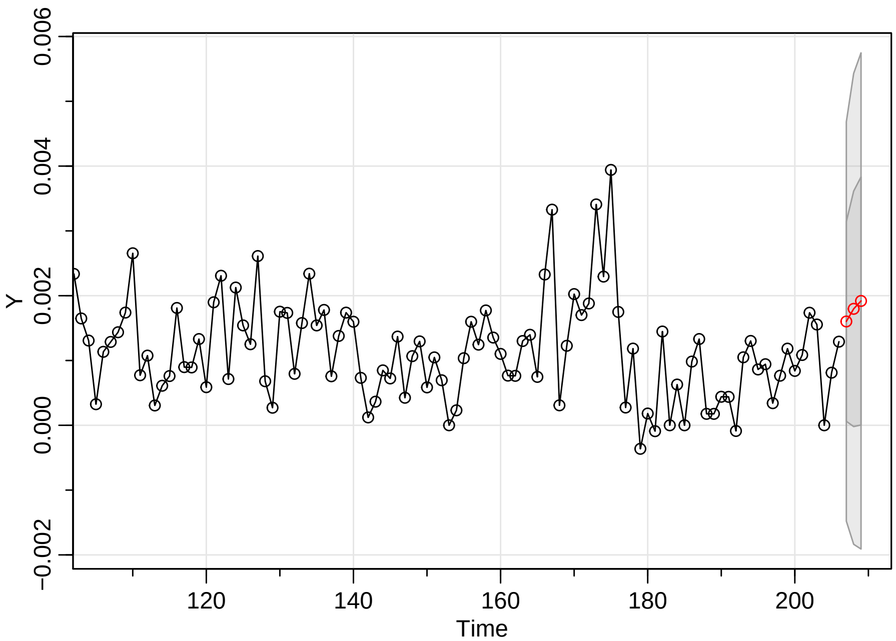

Y
0.006
0.004
0.002
0.000
−0.002
100
120
140
160
180
200
Time

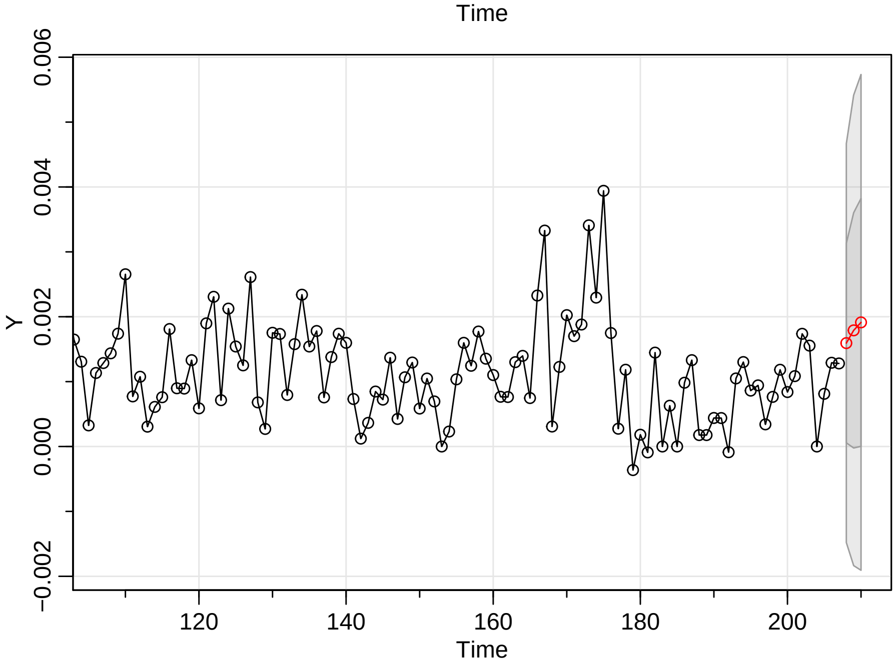


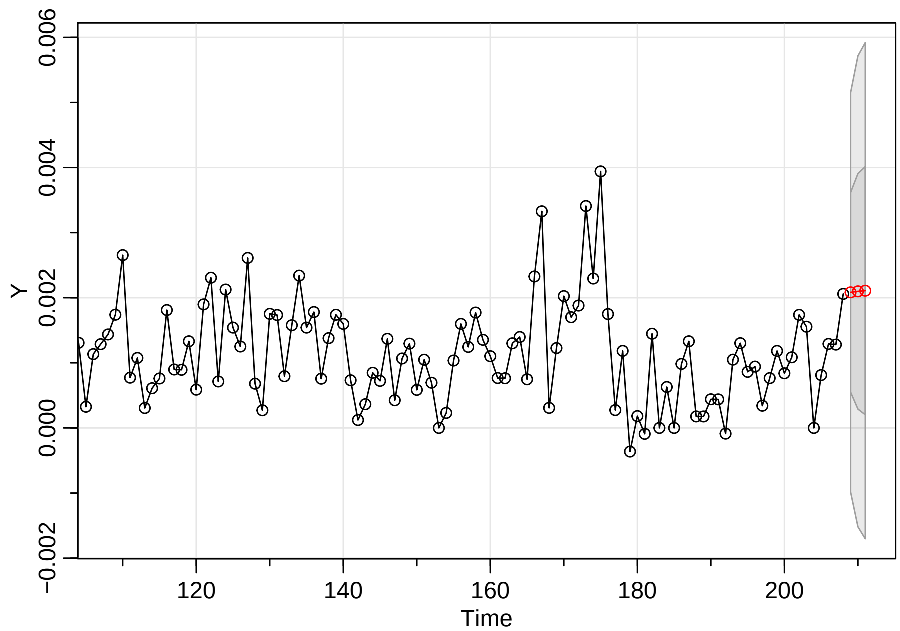
Y
0.004
0.002
0.000
−0.002
100
120
140
160
180
200
Time

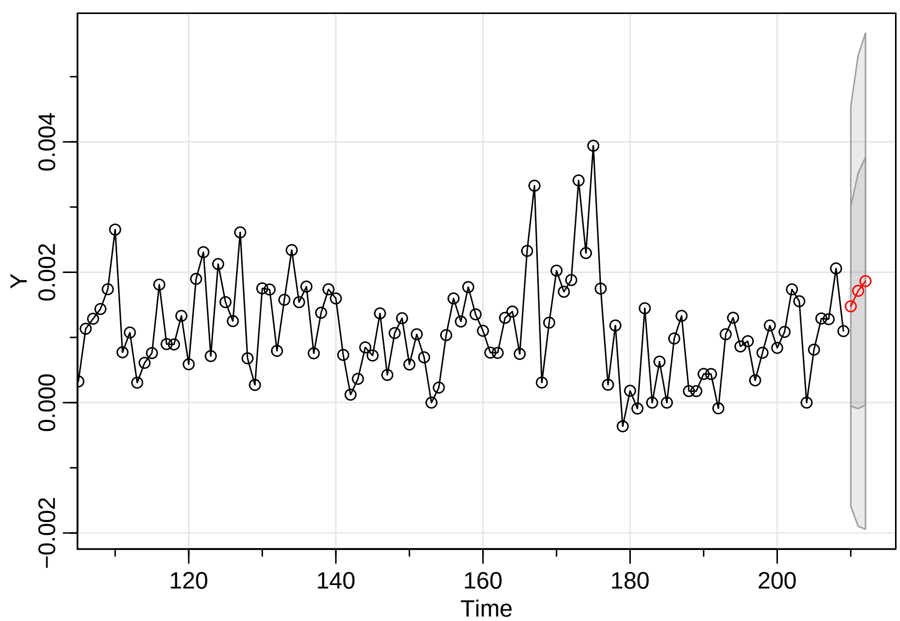


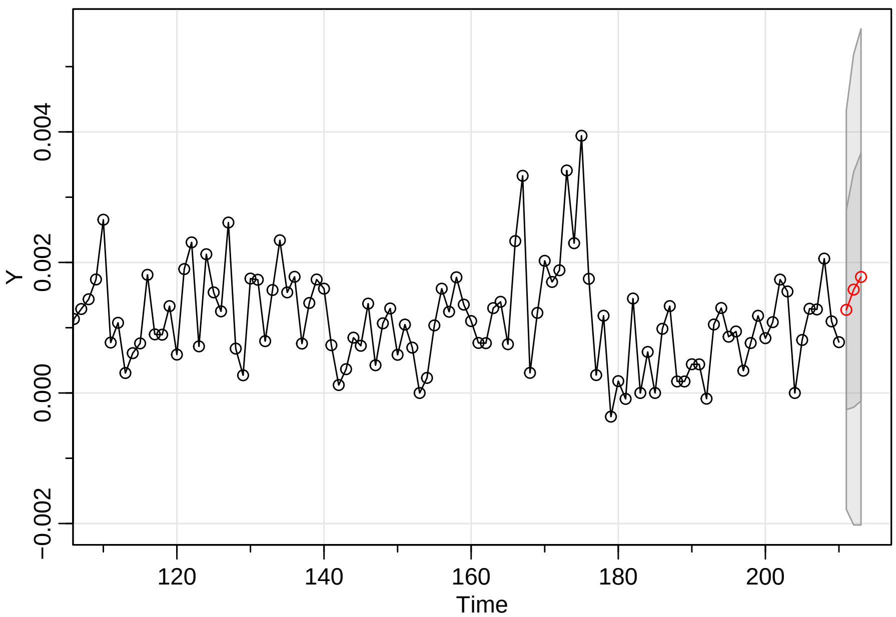
Y
0.004
0.002
0.000
−0.002
120
140
160
180
200
Time


Y
0.006
0.004
0.002
0.000
−0.002
120
140
160
180
200
Time

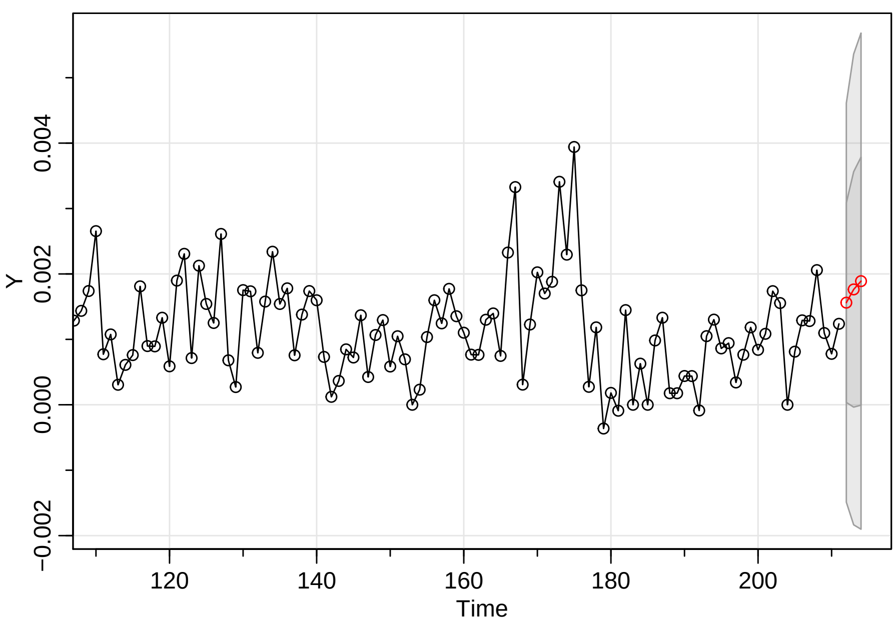

Y
0.006
0.004
0.002
0.000
−0.002
120
140
160
180
200
Time


Y
Time
0.006
0.004
0.002
0.000
−0.002
120
140
160
180
200

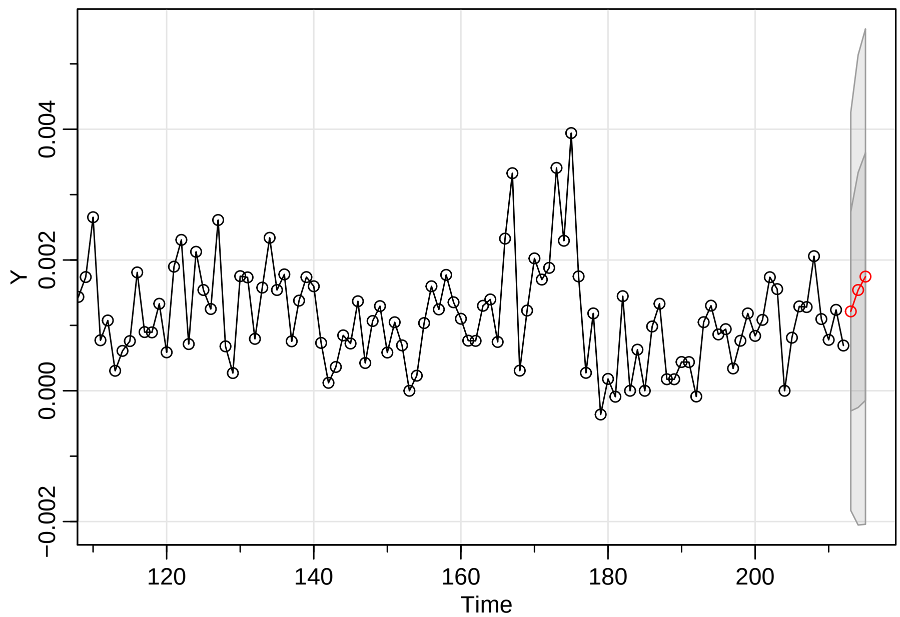

Y
Time
0.004
0.002
0.000
−0.002
120
140
160
180
200

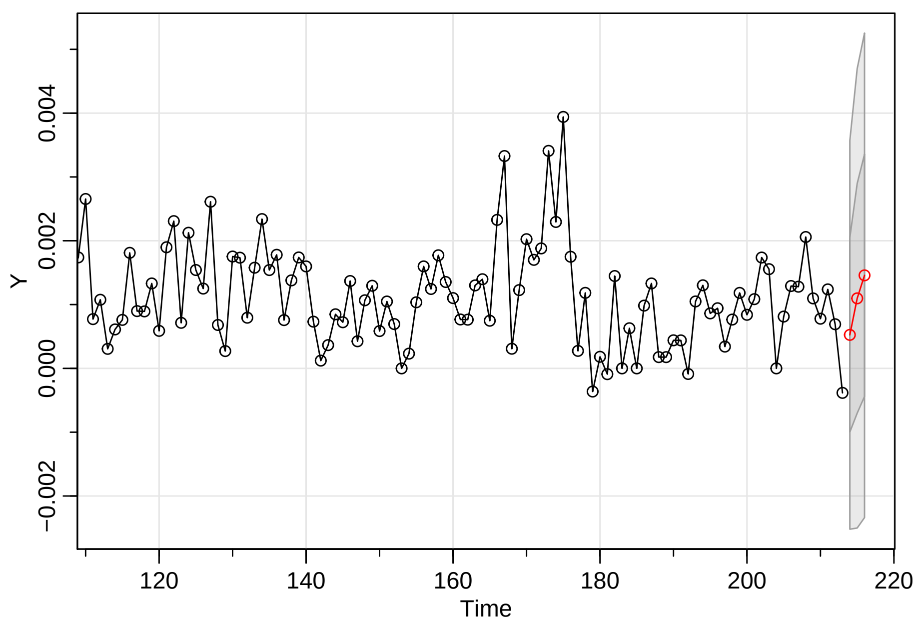


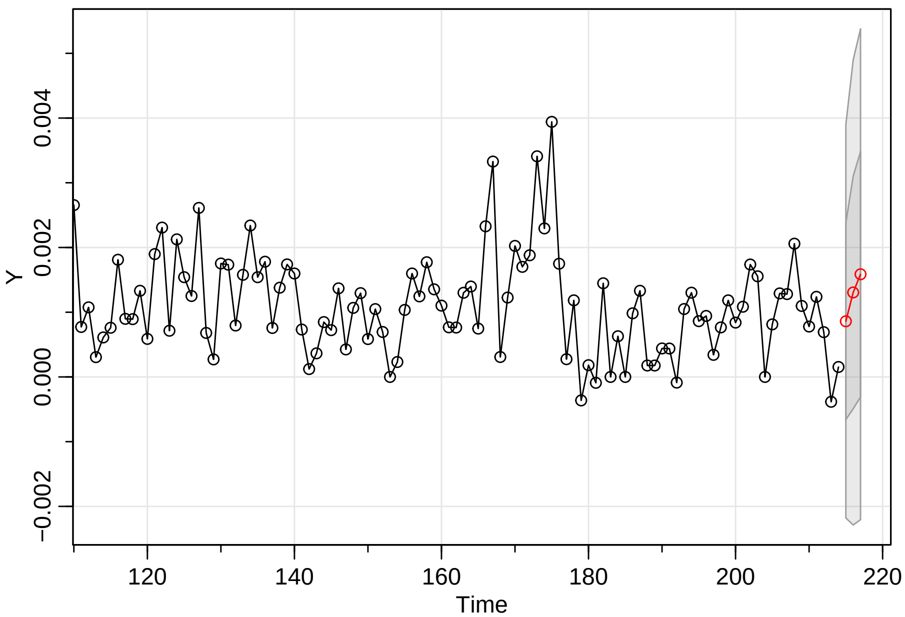
Y
0.004
0.002
0.000
−0.002
120
140
160
180
200
Time

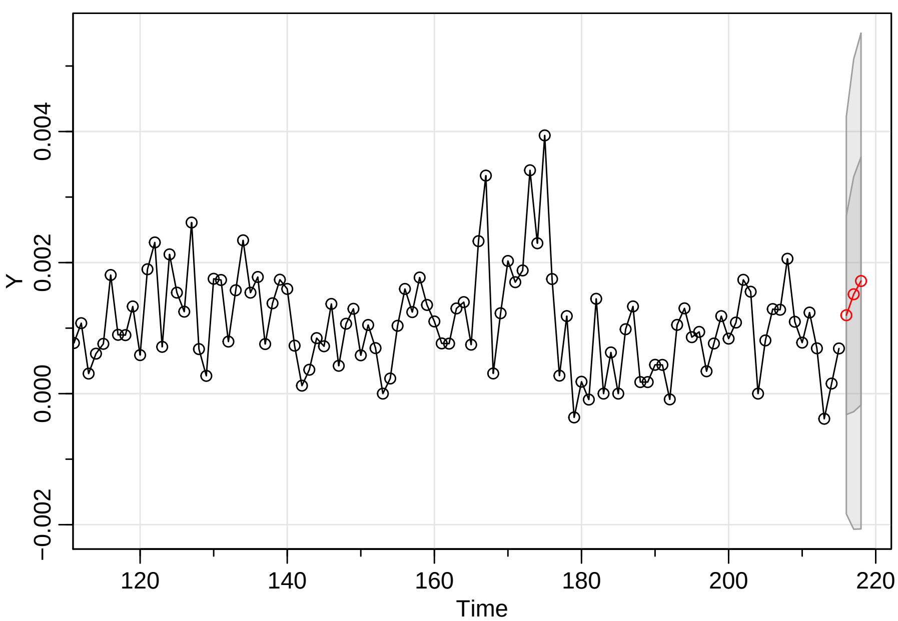


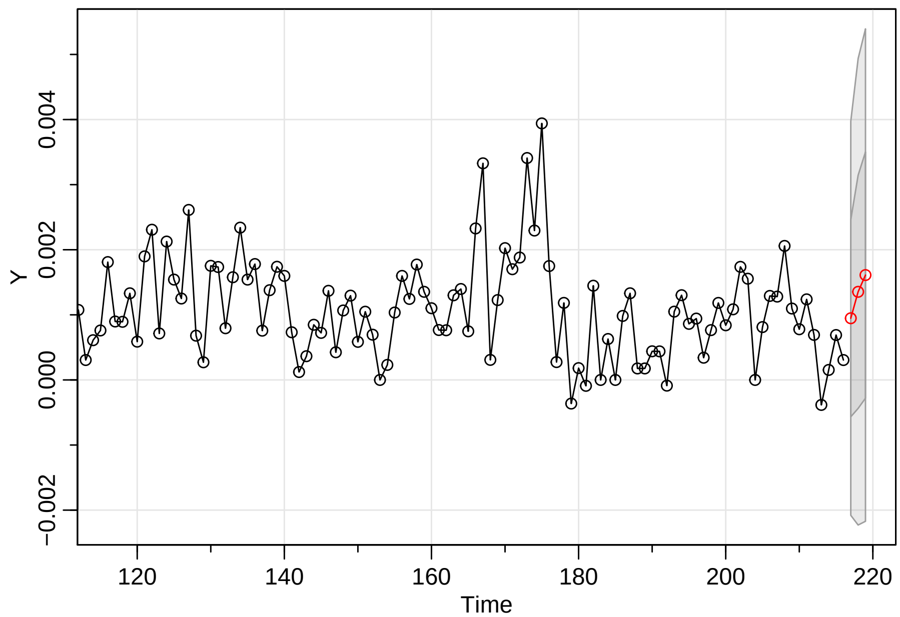
Y
0.004
0.002
0.000
−0.002
120
140
160
180
200
Time

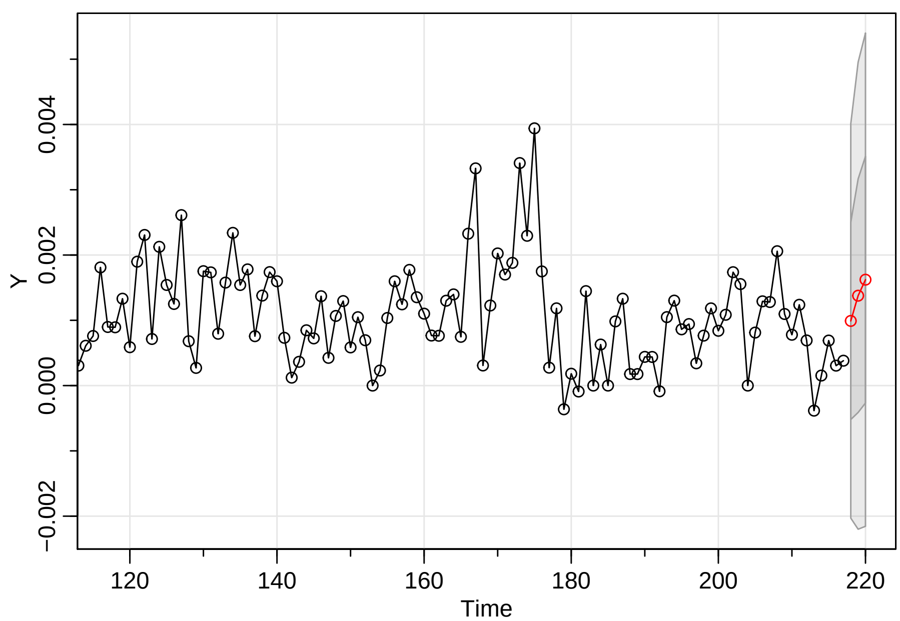

Y
0.004
0.002
0.000
−0.002
120
140
160
180
200
Time

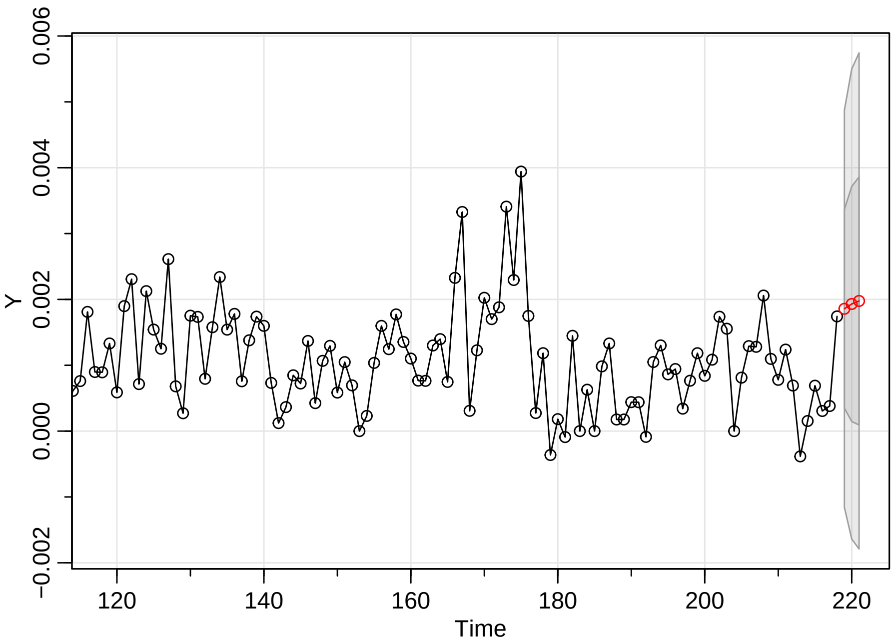

Y
0.004
0.002
0.000
−0.002
120
140
160
180
200
220
Time

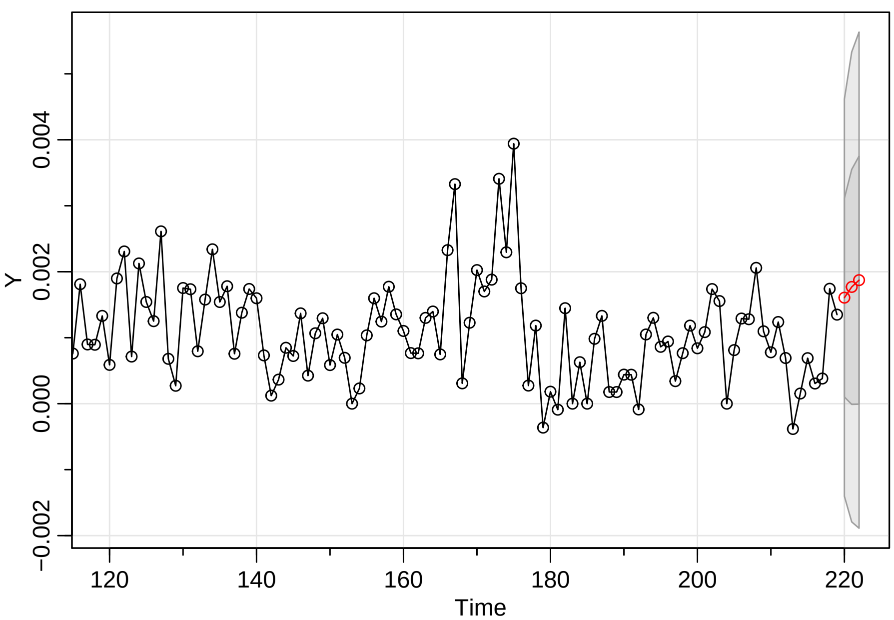

Y
0.004
0.002
0.000
−0.002
120
140
160
180
200
220
Time

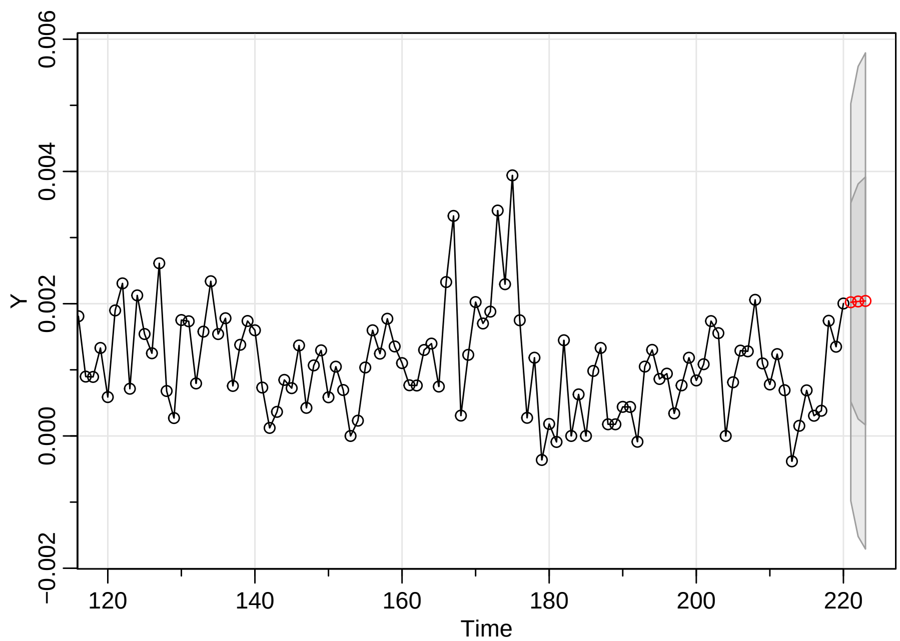

Y
0.004
0.002
0.000
−0.002
120
140
160
180
200
220
Time

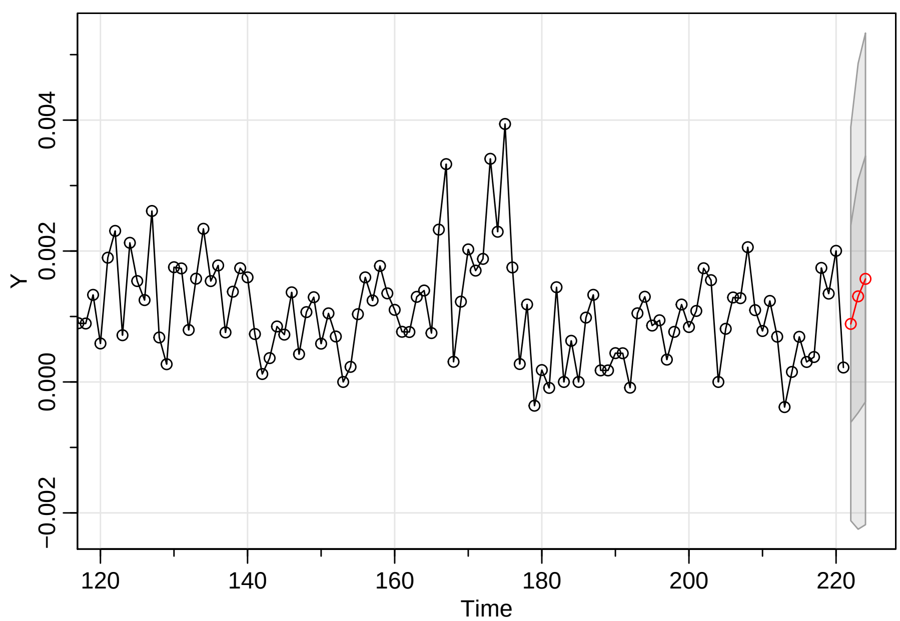

Y
0.004
0.002
0.000
−0.002
120
140
160
180
200
220
Time

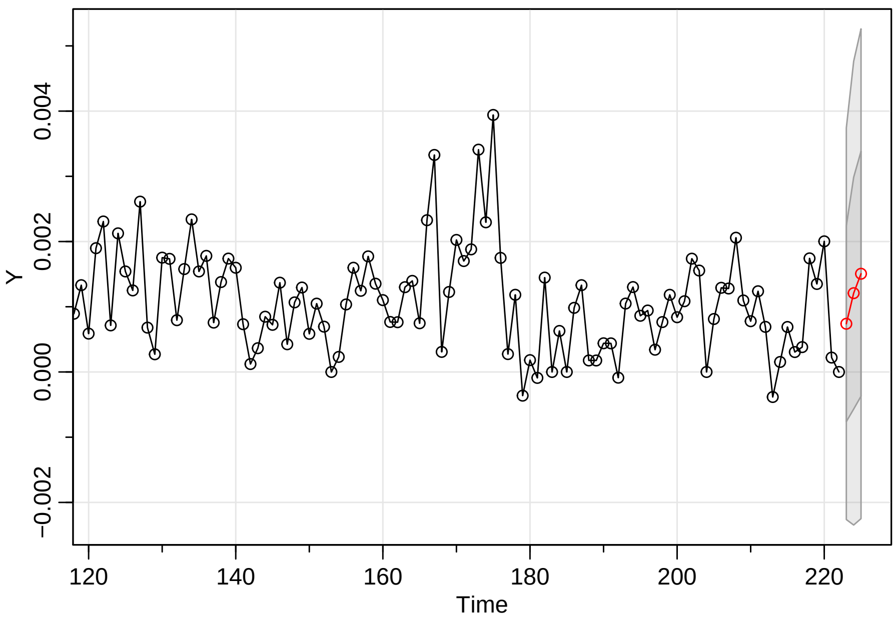

Y
0.004
0.002
0.000
−0.002
120
140
160
180
200
220
Time


Y
Time
0.006
0.004
0.002
0.000
−0.002
120
140
160
180
200
220

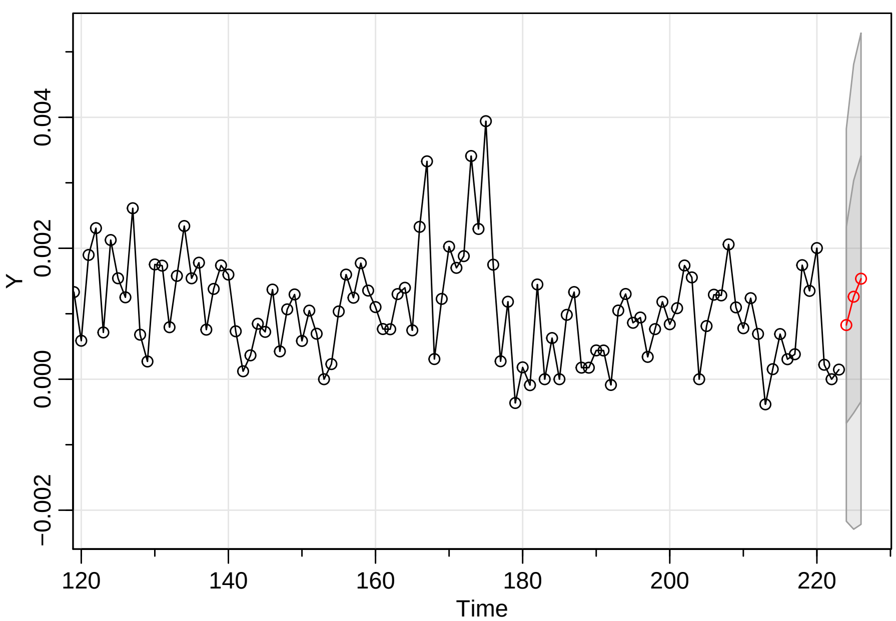

Y
Time
0.004
0.002
0.000
−0.002
120
140
160
180
200
220


Y
Time
0.006
0.004
0.002
0.000
−0.002
120
140
160
180
200
220

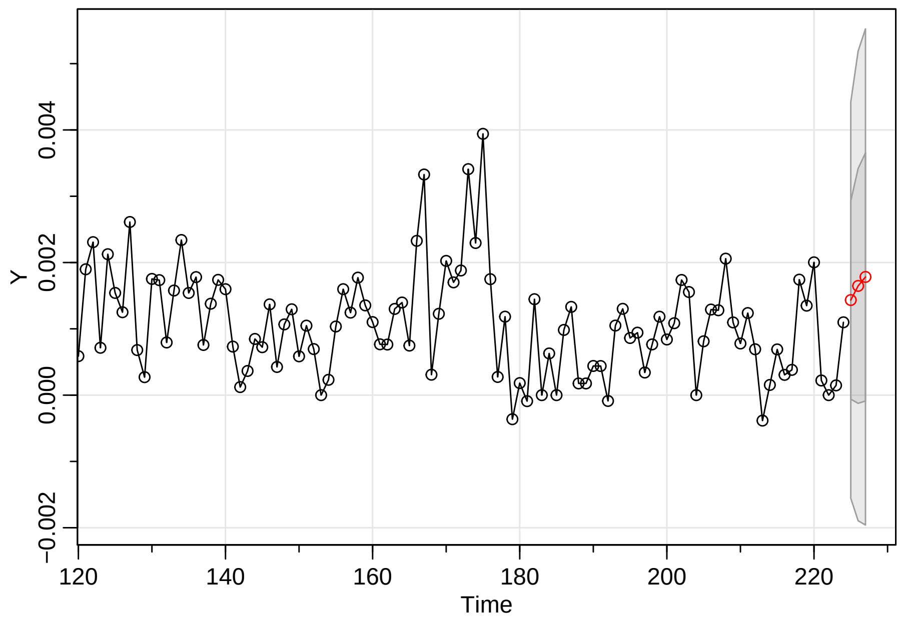

Y
Time
0.004
0.002
0.000
−0.002
120
140
160
180
200
220

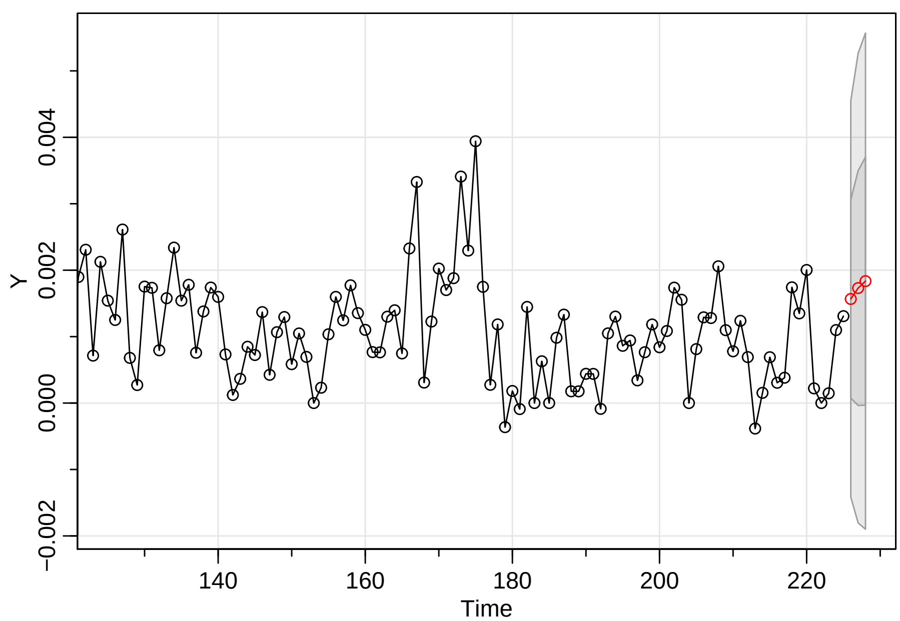


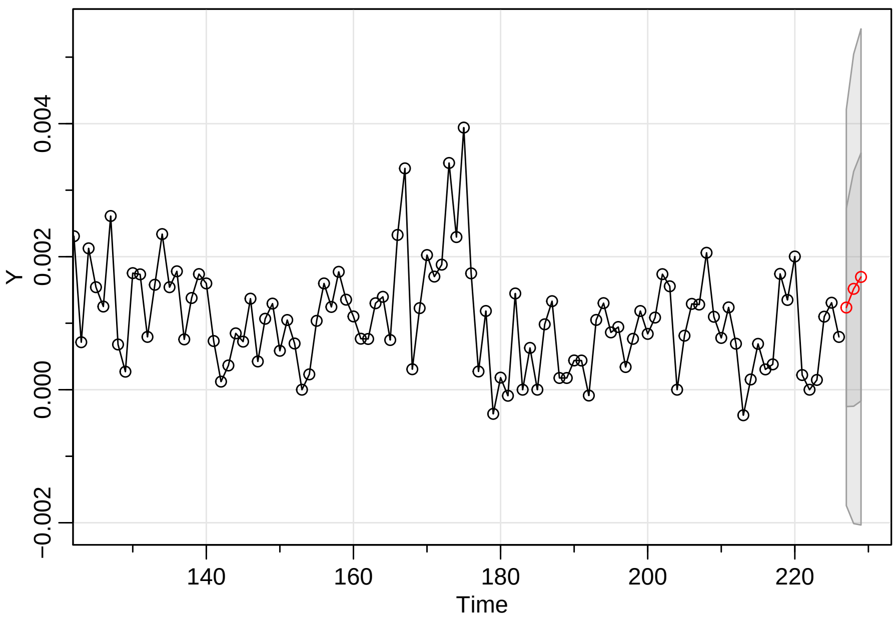
Y
Time

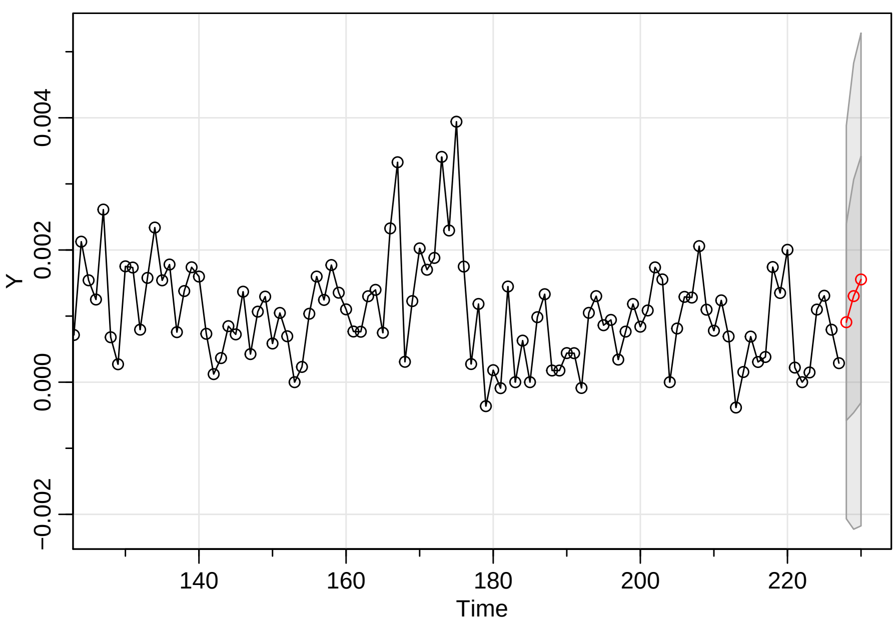


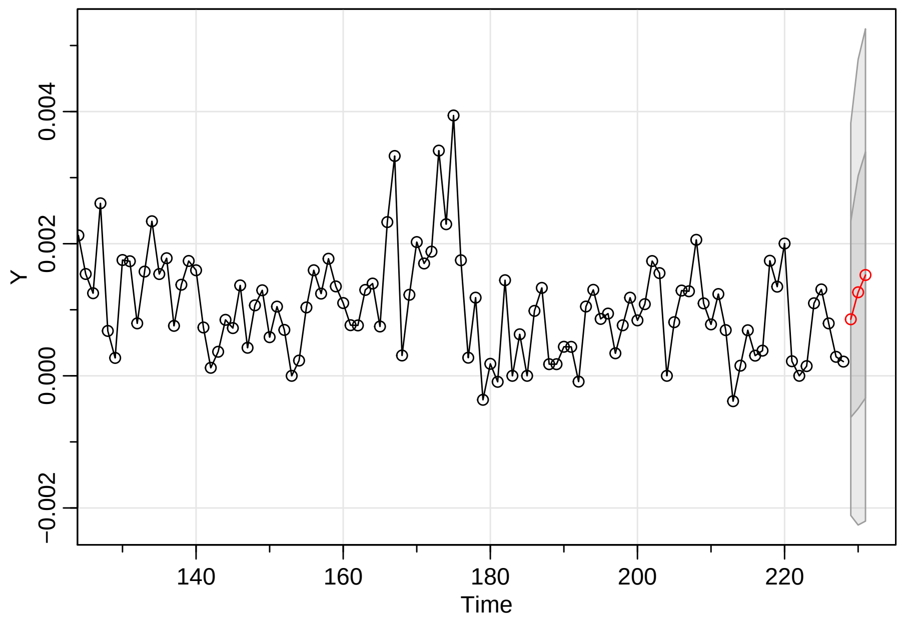
Y
Time

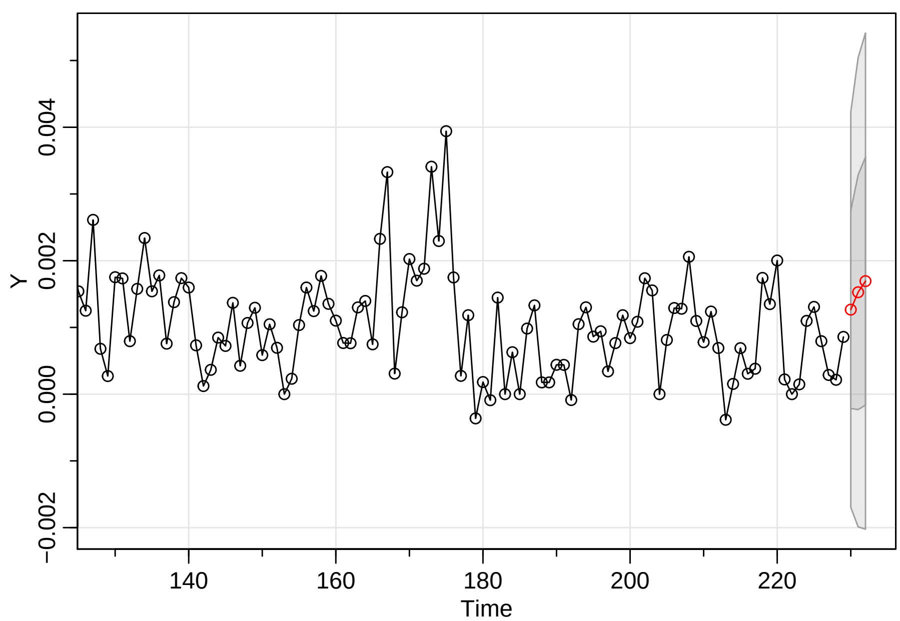

Y
0.004
0.002
0.000
−0.002
120
140
160
180
200
220
Time

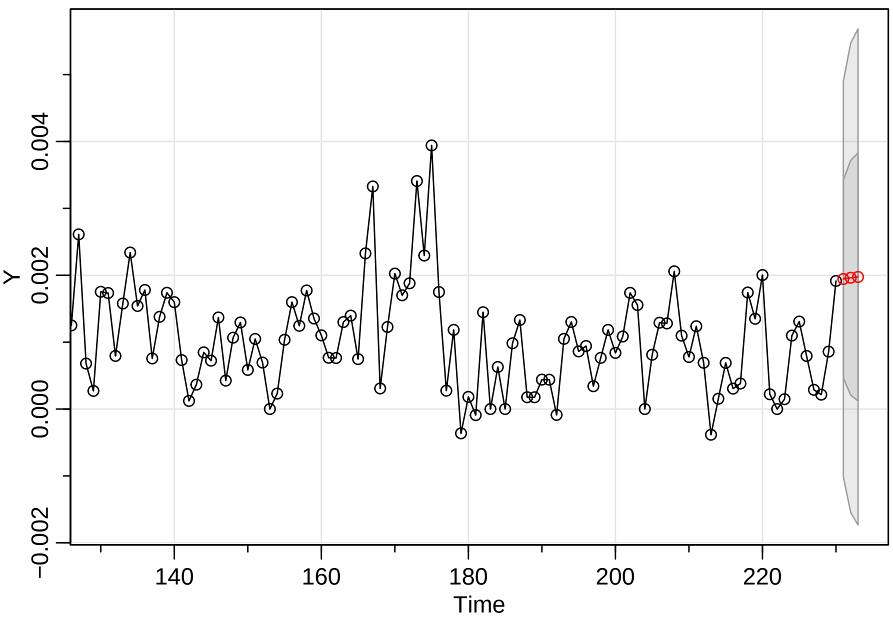

Y
0.004
0.002
0.000
−0.002
140
160
180
200
220
Time

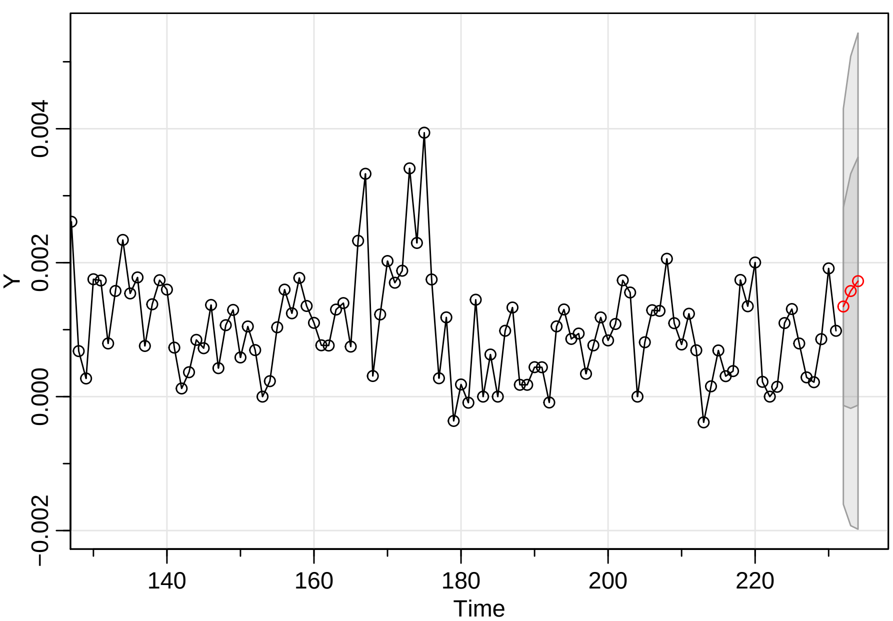

Y
-0.002
0.000
0.002
0.004
140
160
180
200
220
Time

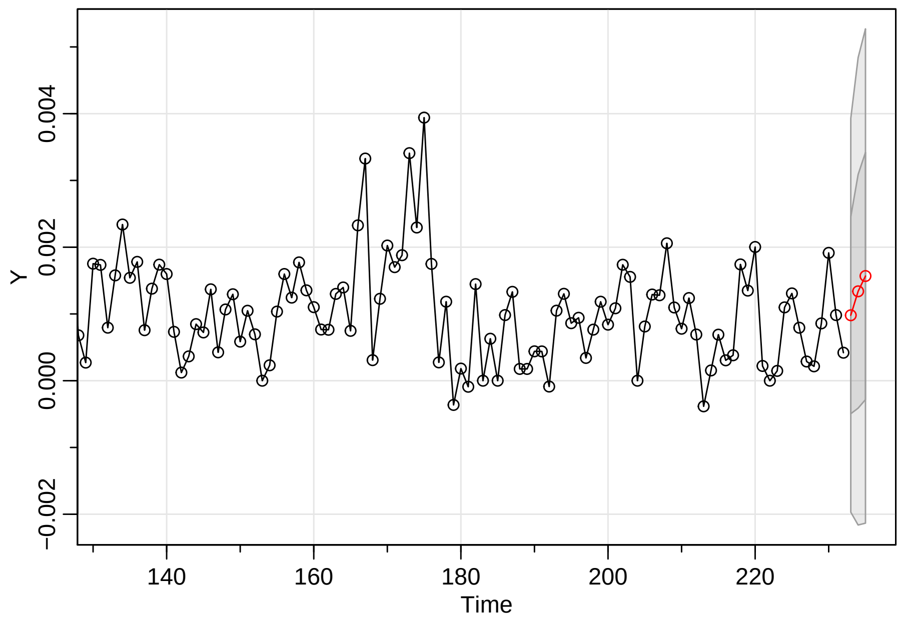

Y
-0.002
0.000
0.002
0.004
140
160
180
200
220
Time

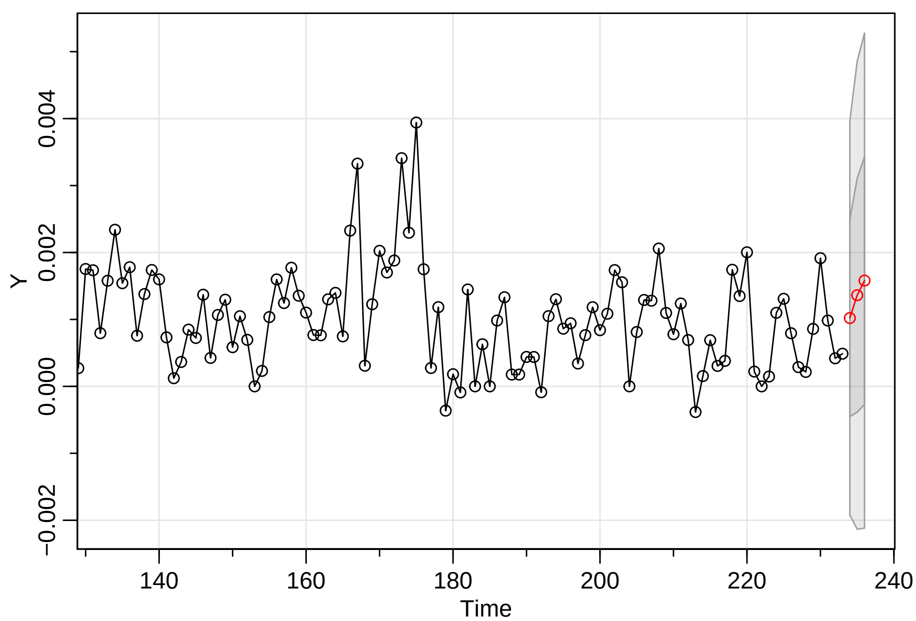

Y
0.004
0.002
0.000
−0.002
140
160
180
200
220
Time

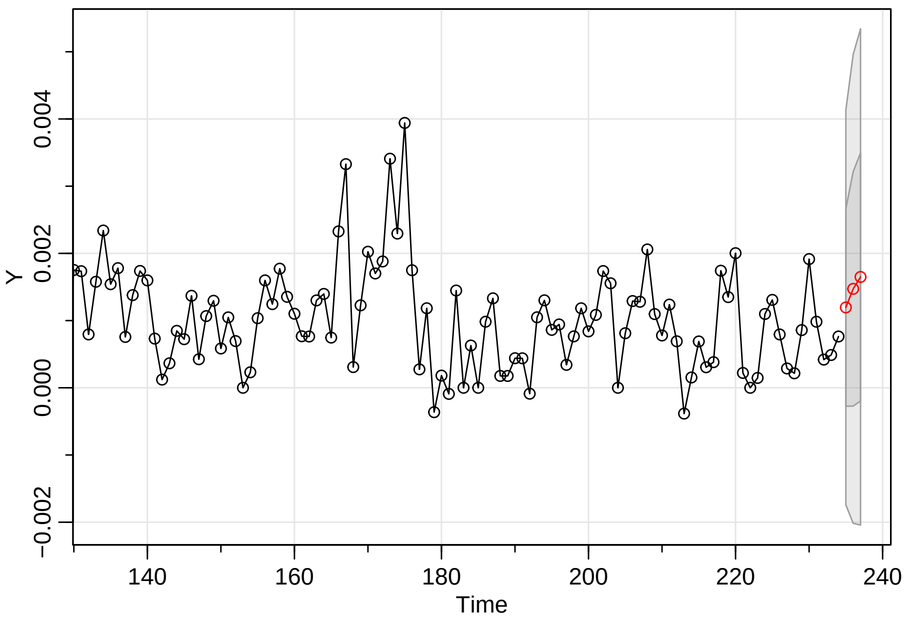

Y
0.004
0.002
0.000
−0.002
140
160
180
200
220
Time

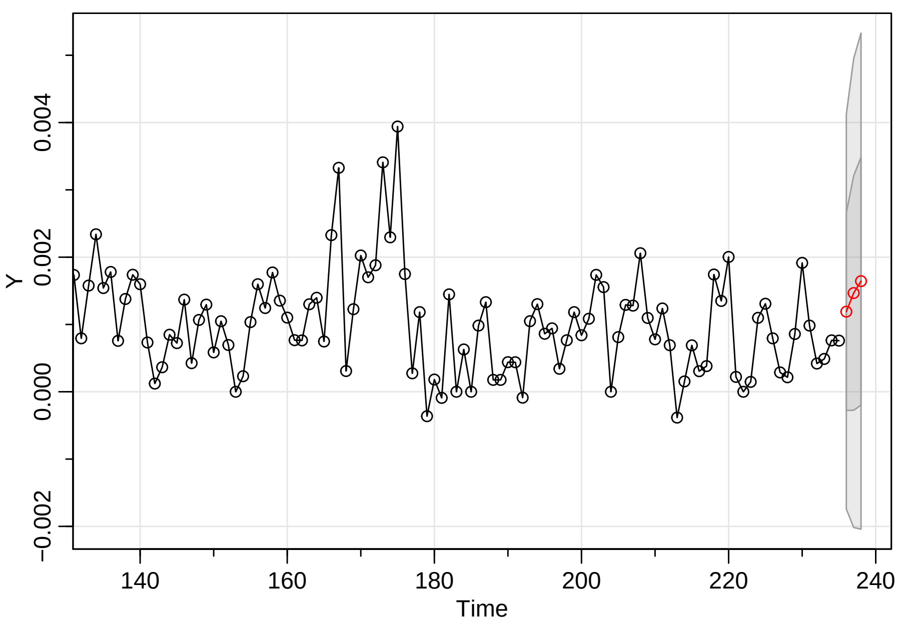

Y
0.004
0.002
0.000
−0.002
140
160
180
200
220
Time

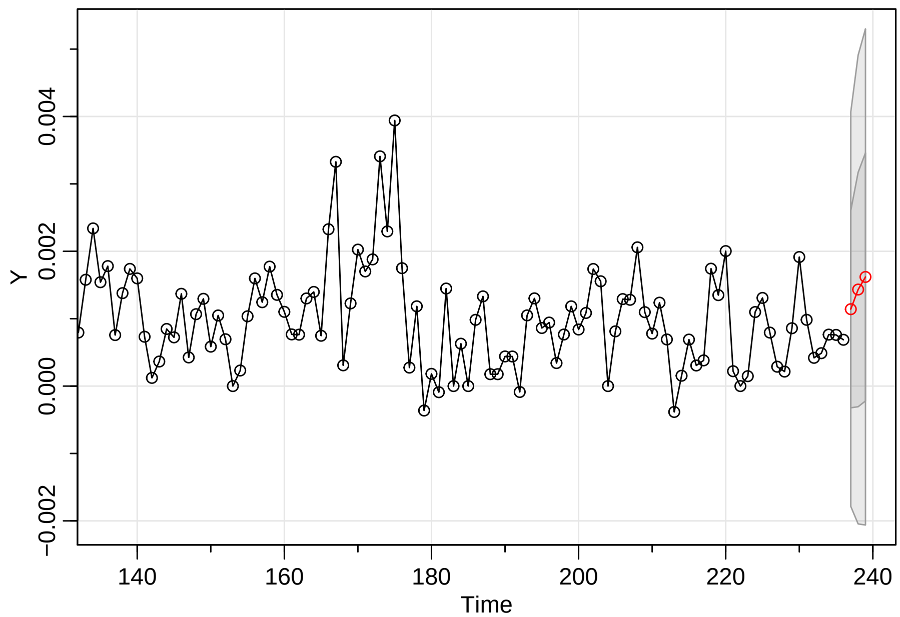

Y
0.004
0.002
0.000
−0.002
140
160
180
200
220
Time

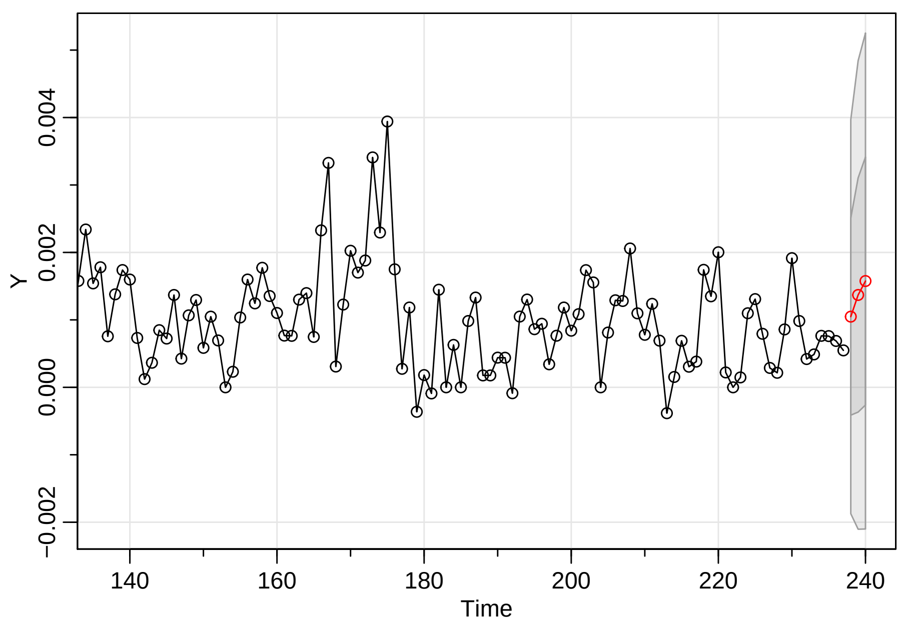

Y
0.004
0.002
0.000
−0.002
140
160
180
200
220
Time

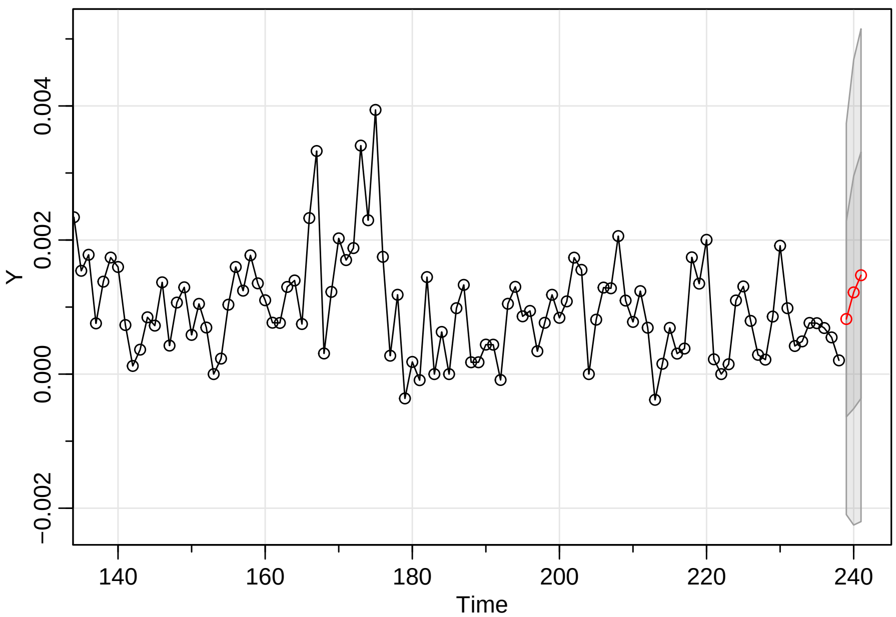

Y
0.004
0.002
0.000
−0.002
140
160
180
200
220
240
Time

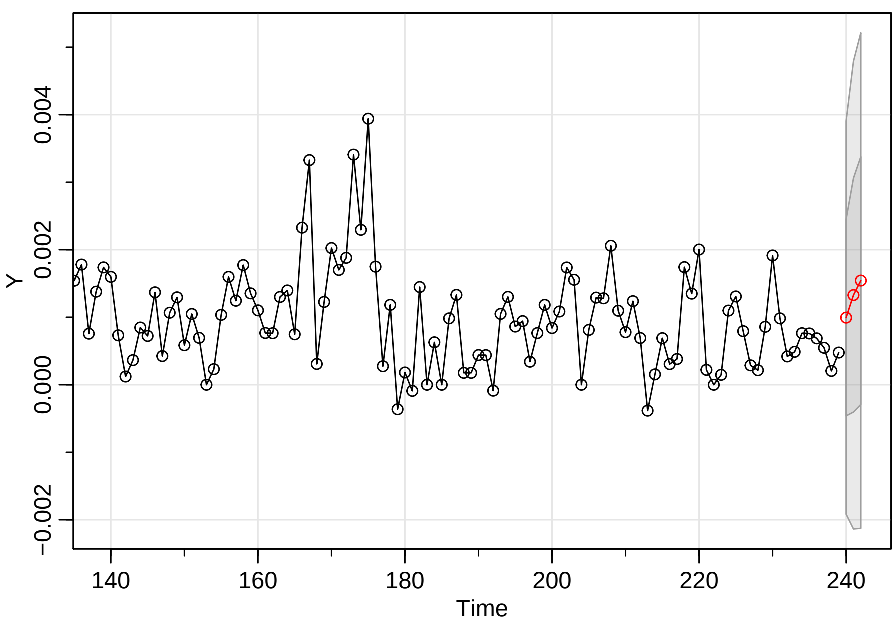

Y
Time
0.004
0.002
0.000
−0.002
140
160
180
200
220
240

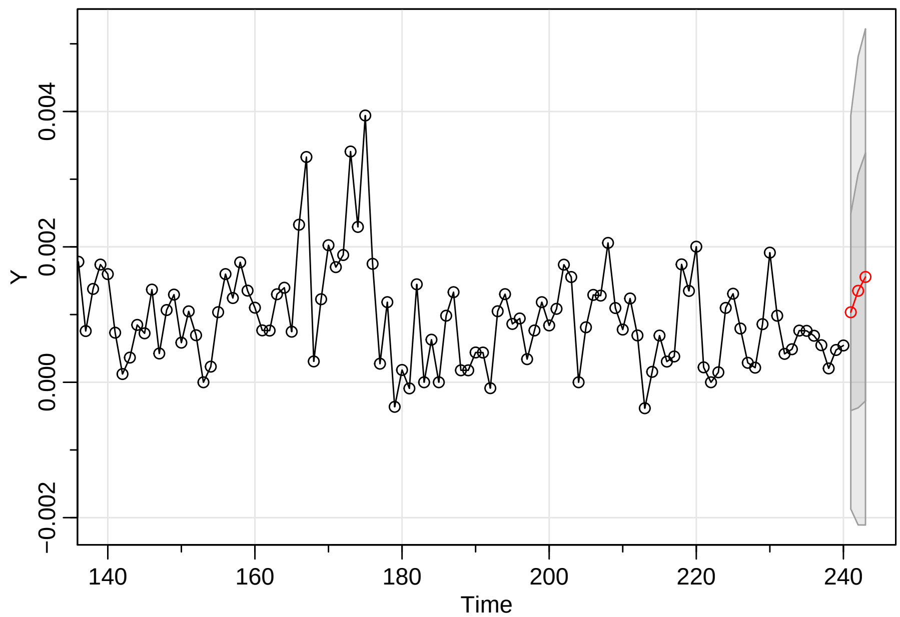

Y
Time
0.004
0.002
0.000
−0.002
140
160
180
200
220
240

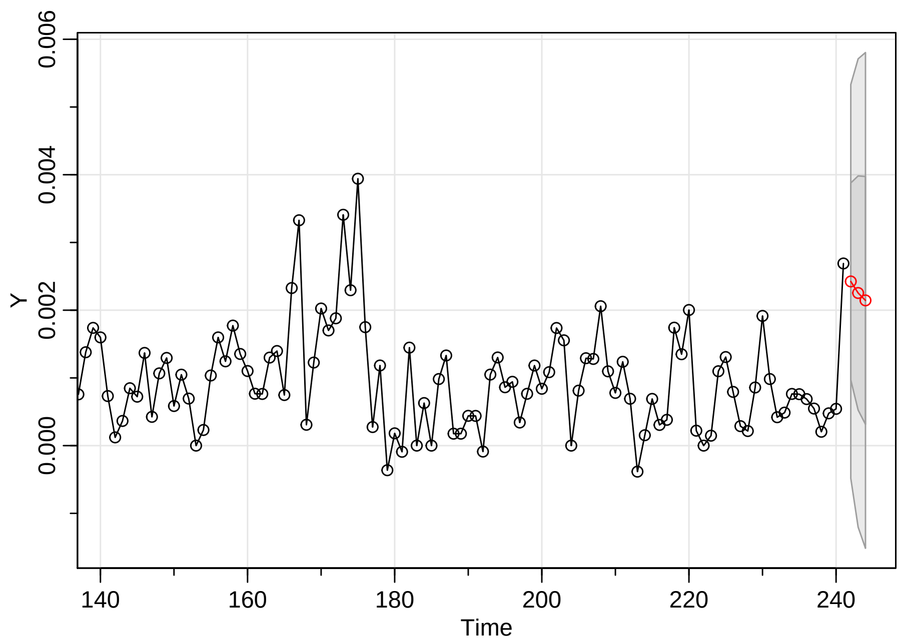

Y
0.004
0.002
0.000
−0.002
140
160
180
200
220
240
Time

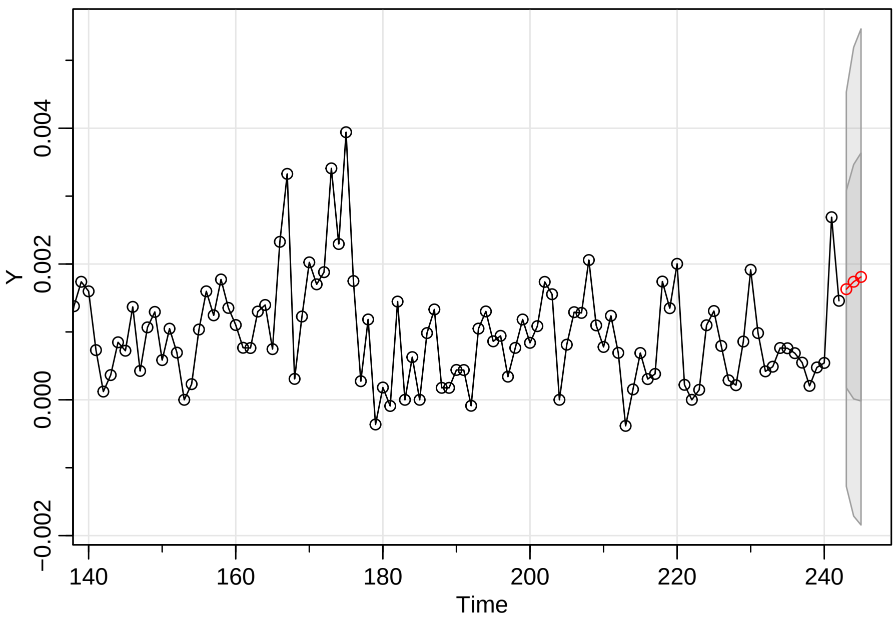

Y
0.004
0.002
0.000
−0.002
140
160
180
200
220
240
Time


Y
Time
0.004
0.002
0.000
−0.002
140
160
180
200
220
240

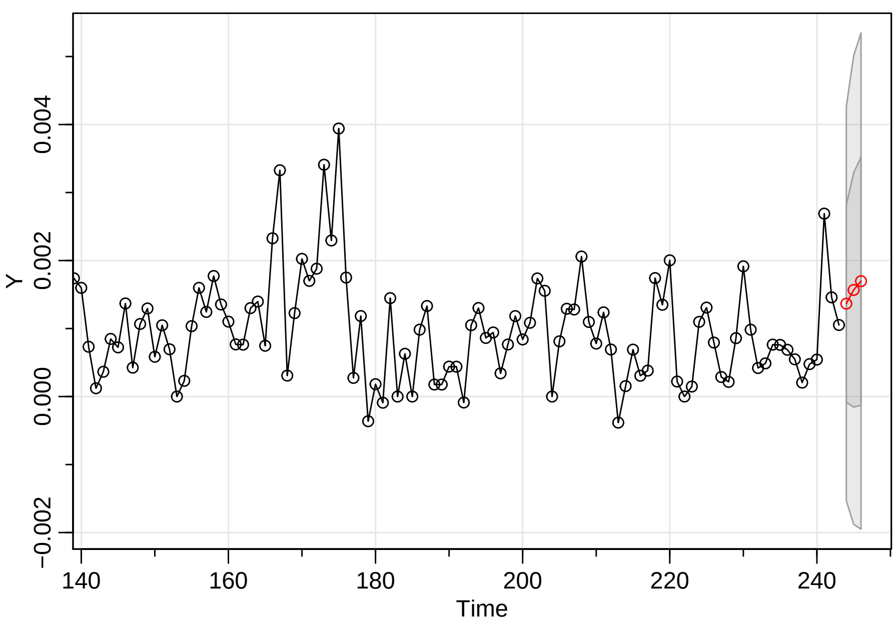

Y
Time
0.004
0.002
0.000
−0.002
140
160
180
200
220
240


Y
Time
0.004
0.002
0.000
−0.002
140
160
180
200
220
240

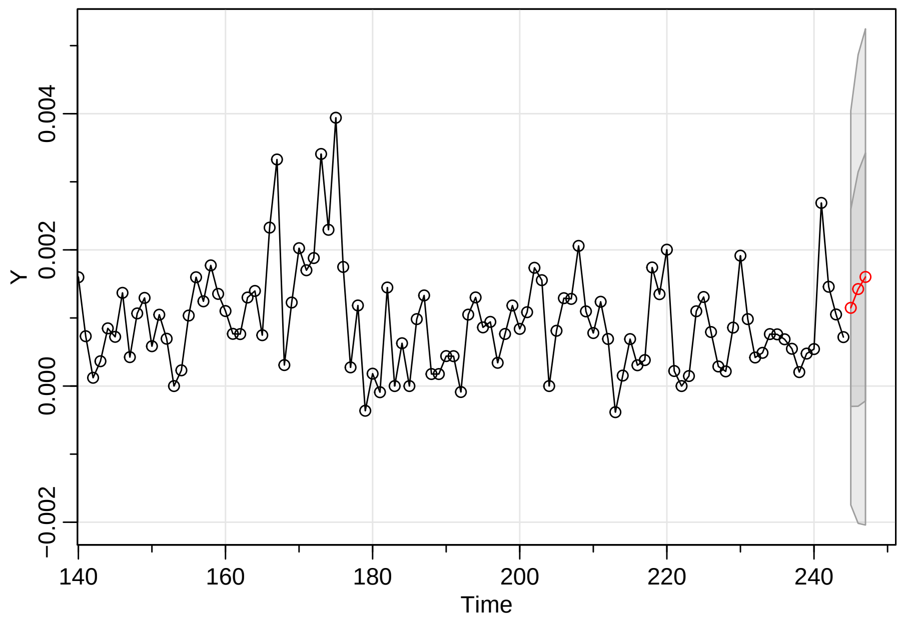

Y
Time
0.006
0.004
0.002
0.000
140
160
180
200
220
240



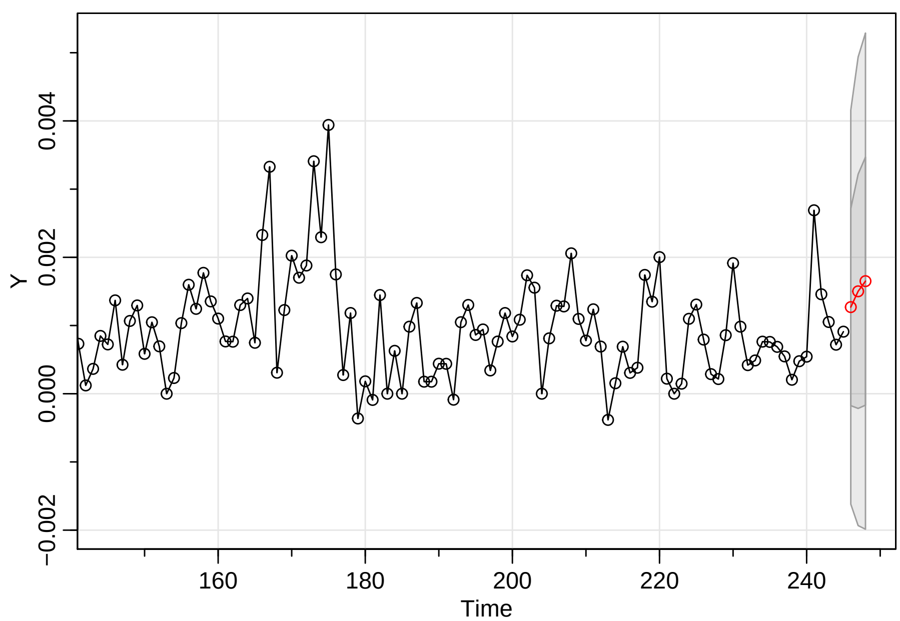
Y
0.004
0.002
0.000
−0.002
140
160
180
200
220
240
Time



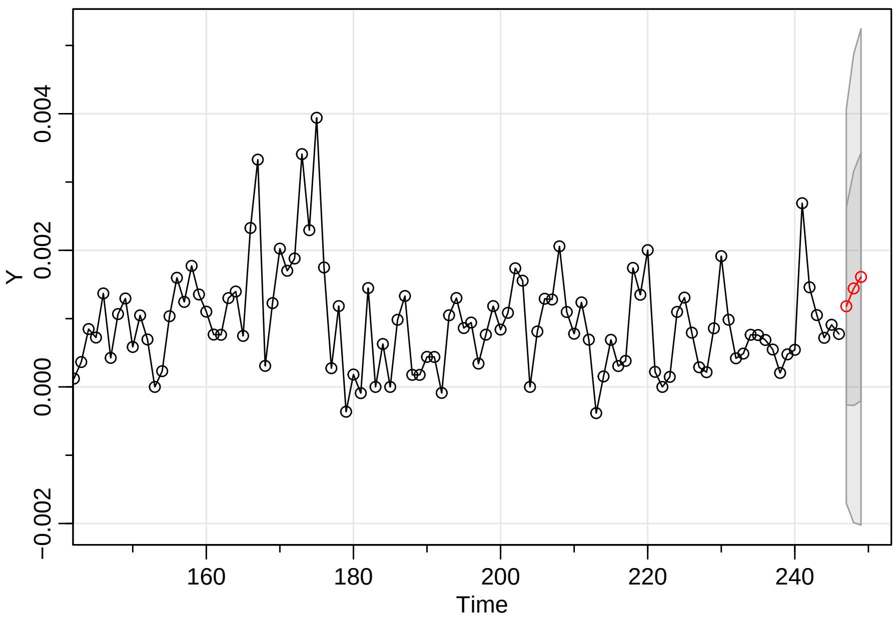
Y
0.004
0.002
0.000
−0.002
140
160
180
200
220
240
Time



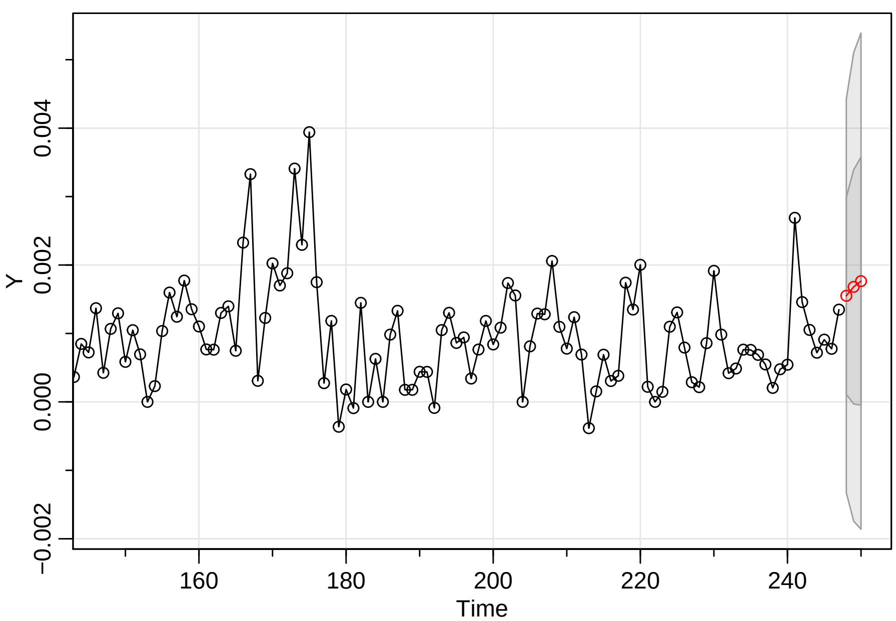
Y
Time



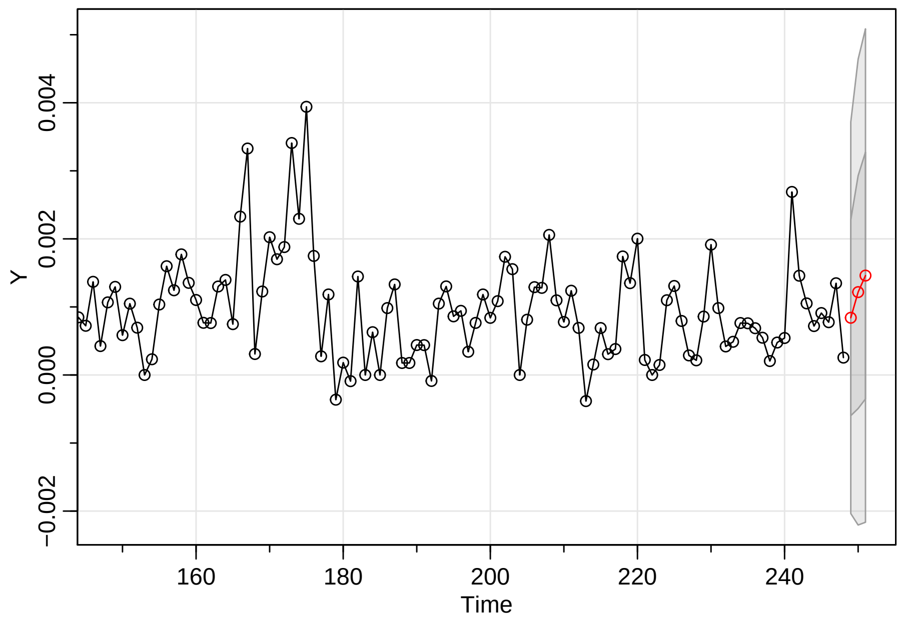
Y
Time



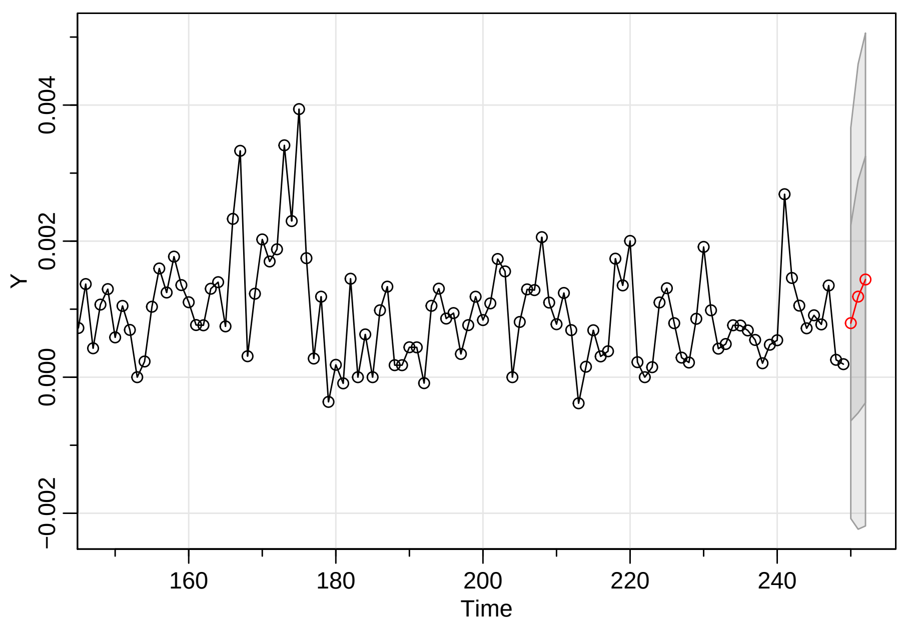
Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
Time



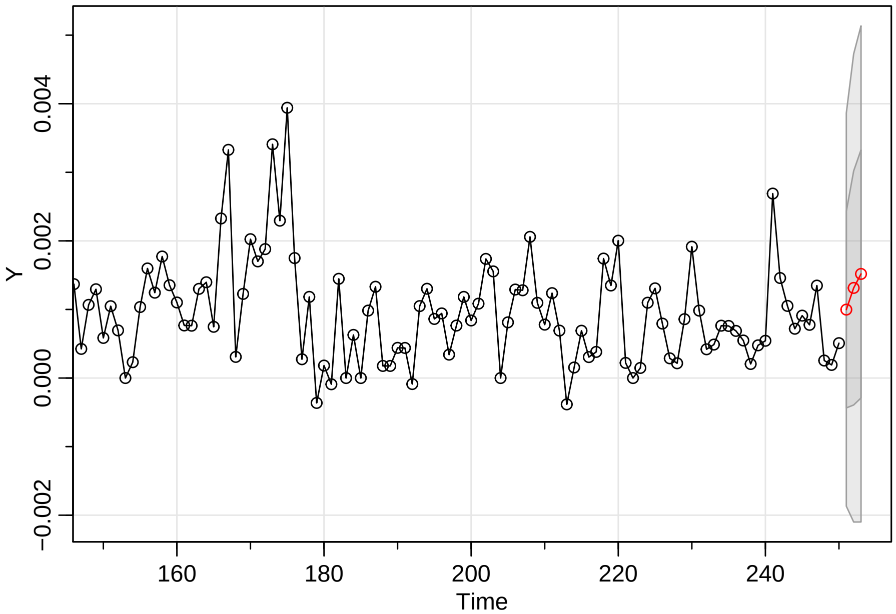
Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
Time



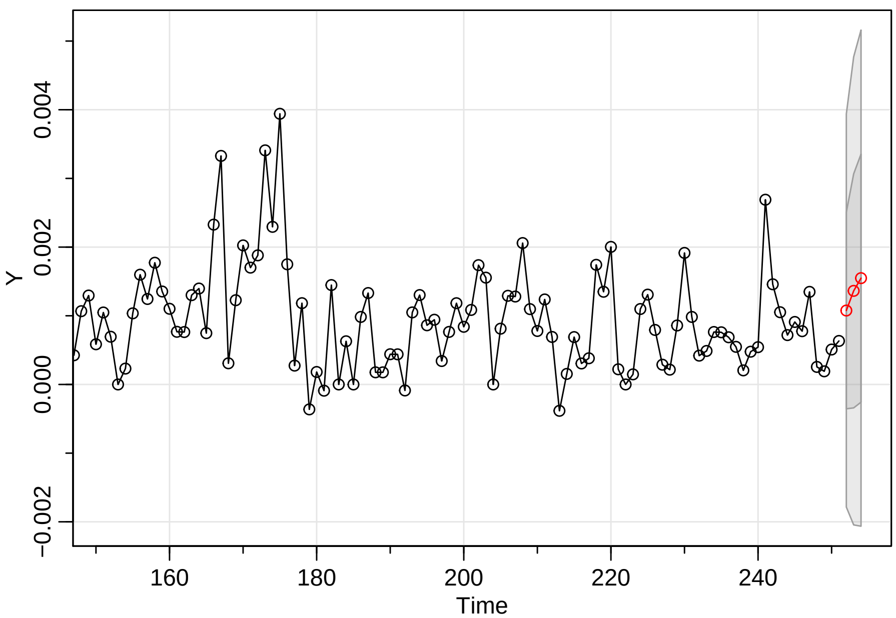
Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
Time



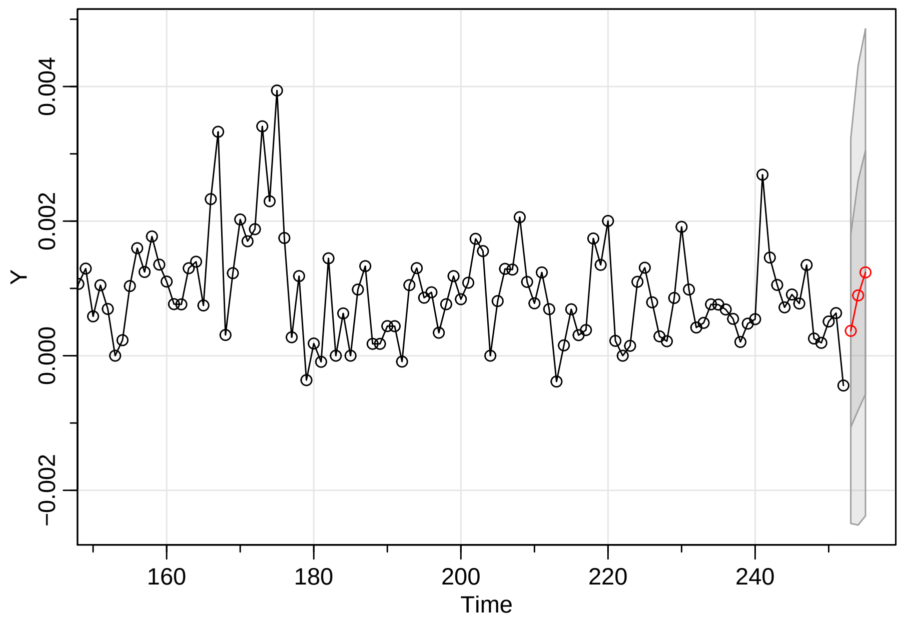
Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
Time



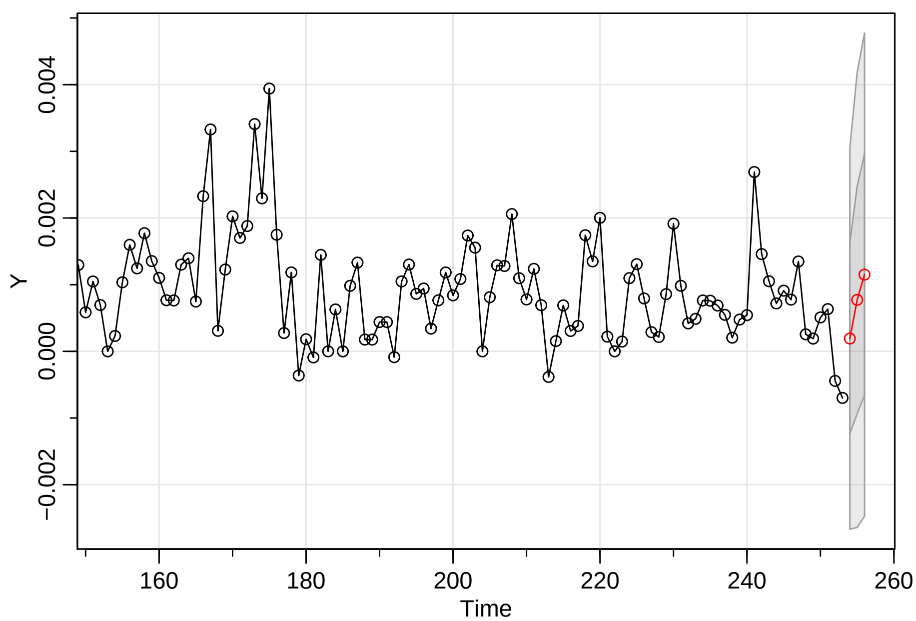
Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
Time



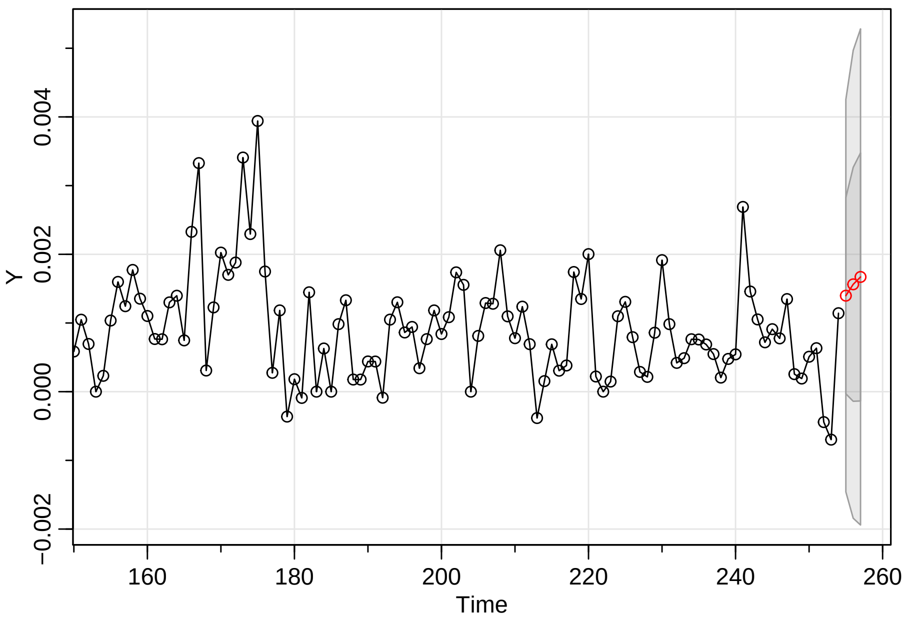
Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
Time

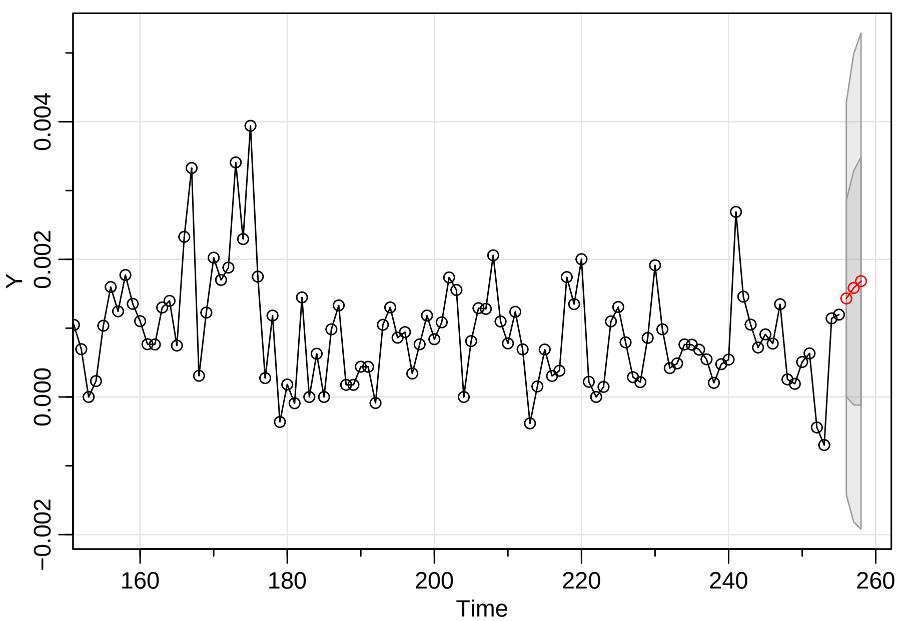


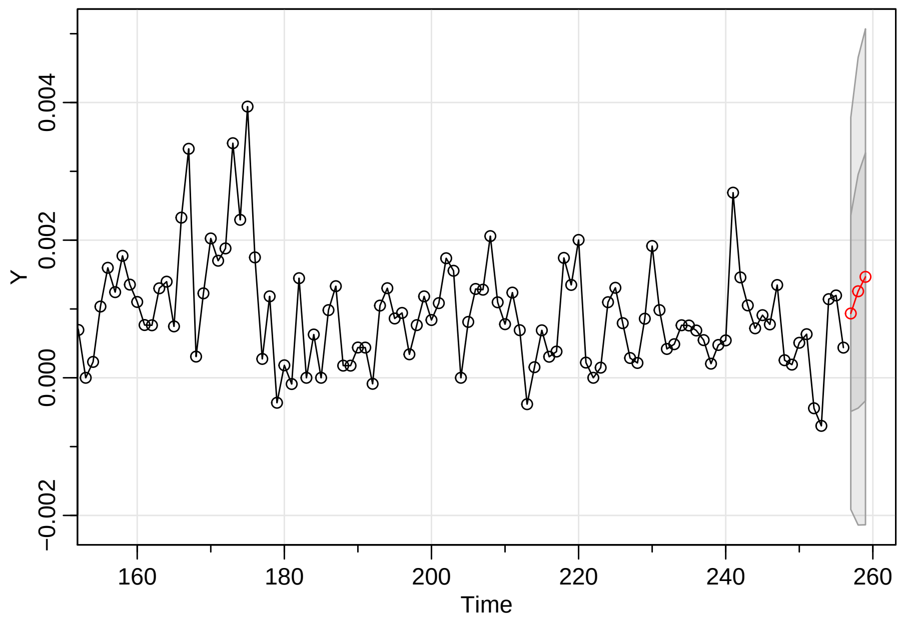
Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
Time

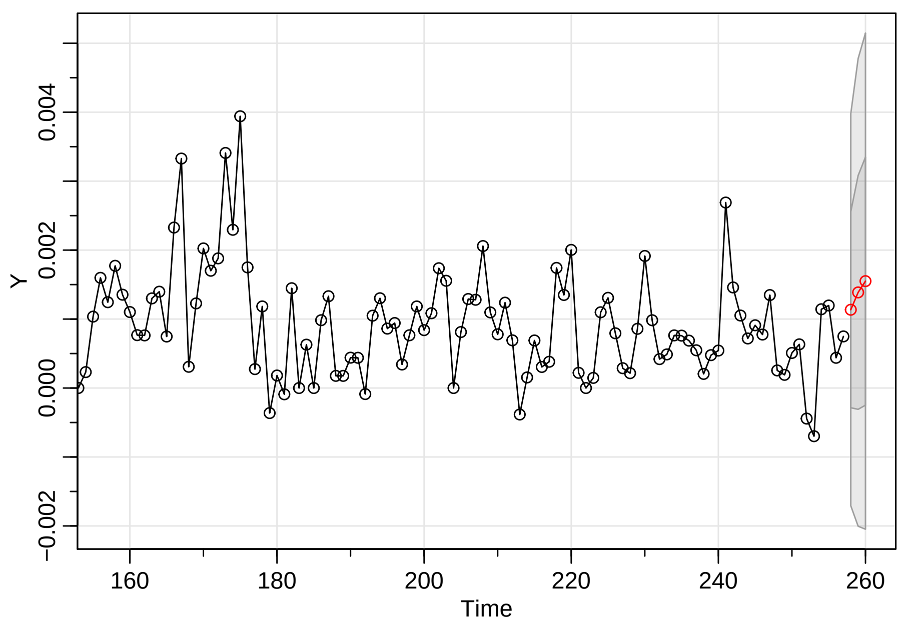


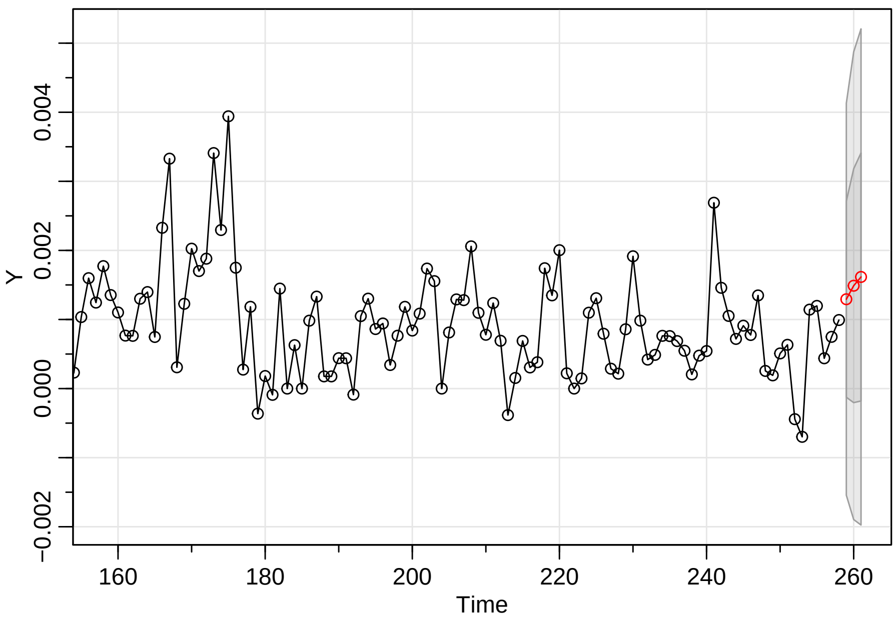
Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
260
Time

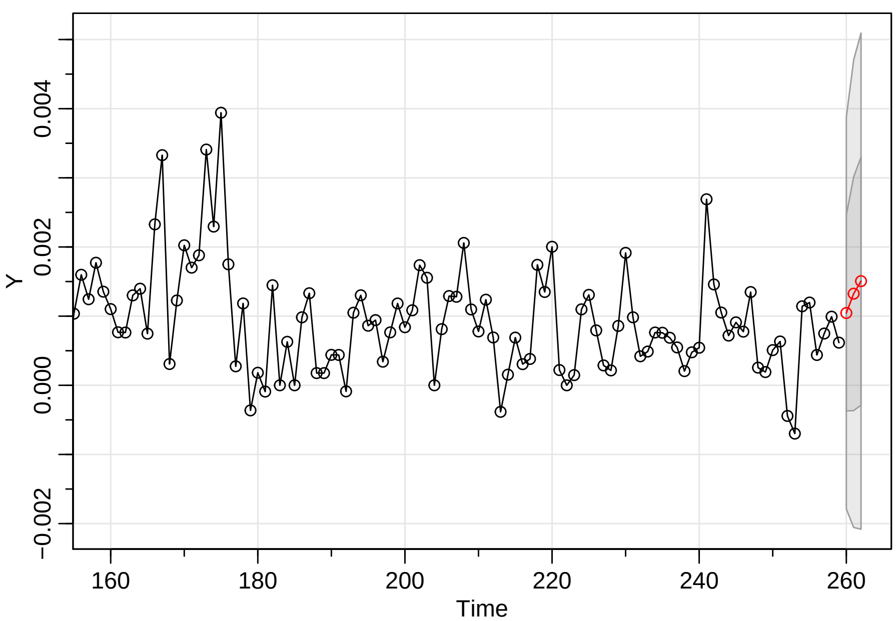

Y
Time
0.004
0.002
0.000
−0.002
160
180
200
220
240
260

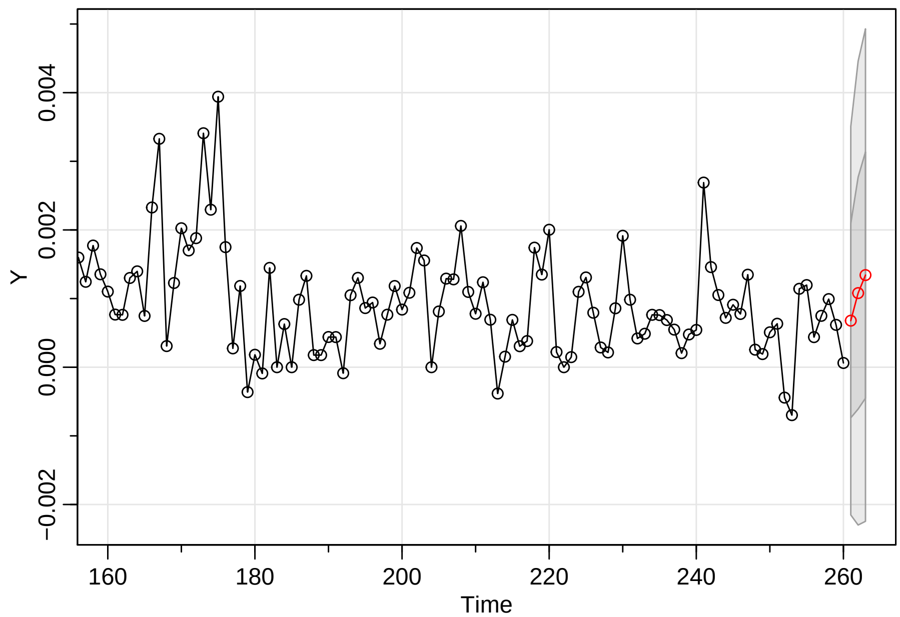

Y
Time
0.004
0.002
0.000
−0.002
160
180
200
220
240
260

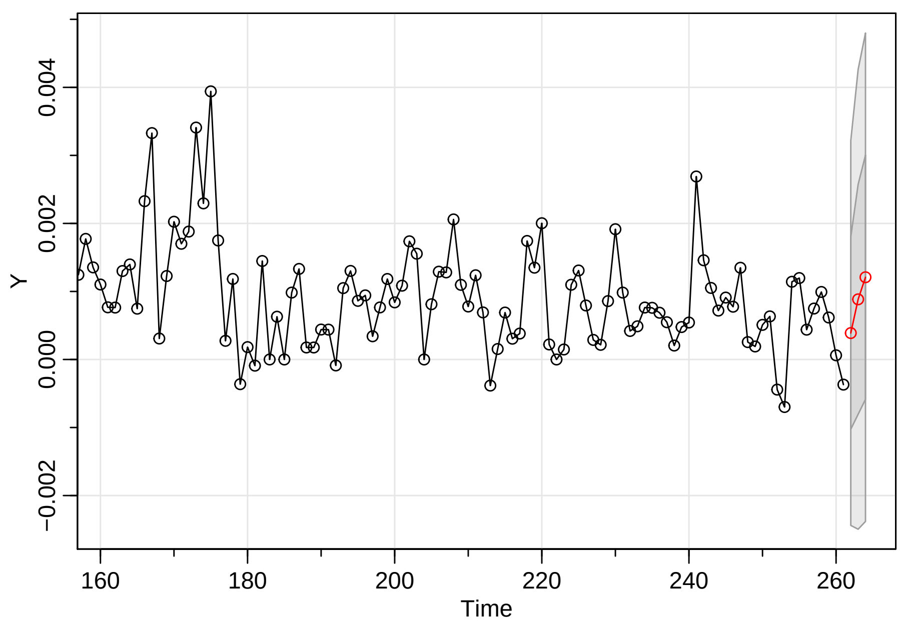

Y
−0.002
0.000
0.002
0.004
160
180
200
220
240
260
Time

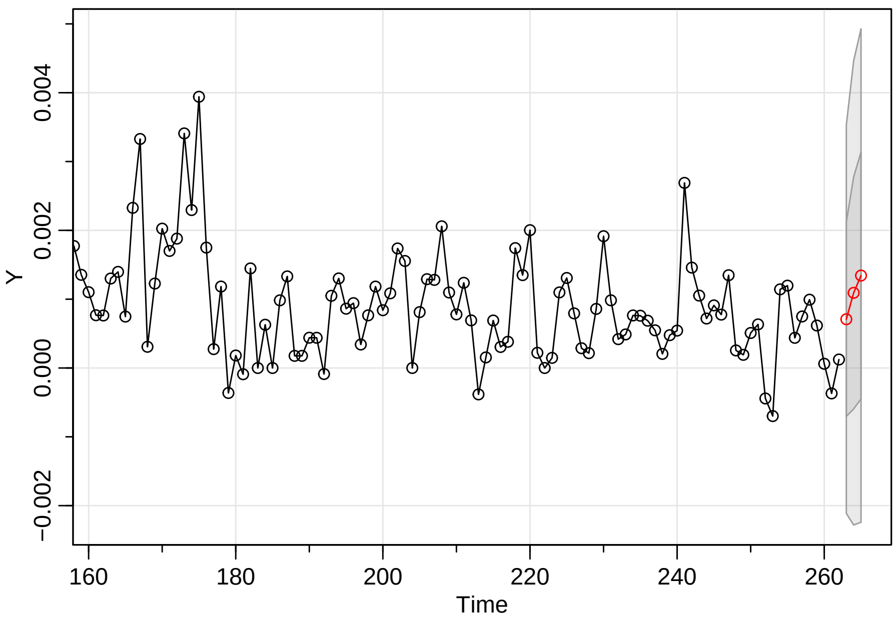

Y
−0.002
0.000
0.002
0.004
160
180
200
220
240
260
Time

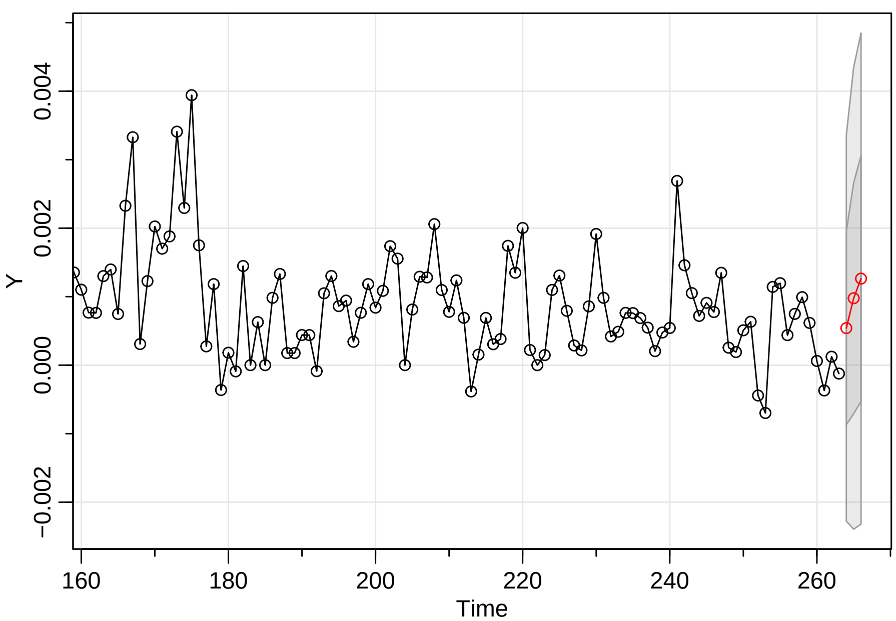

Y
Time
0.004
0.002
0.000
−0.002
160
180
200
220
240
260

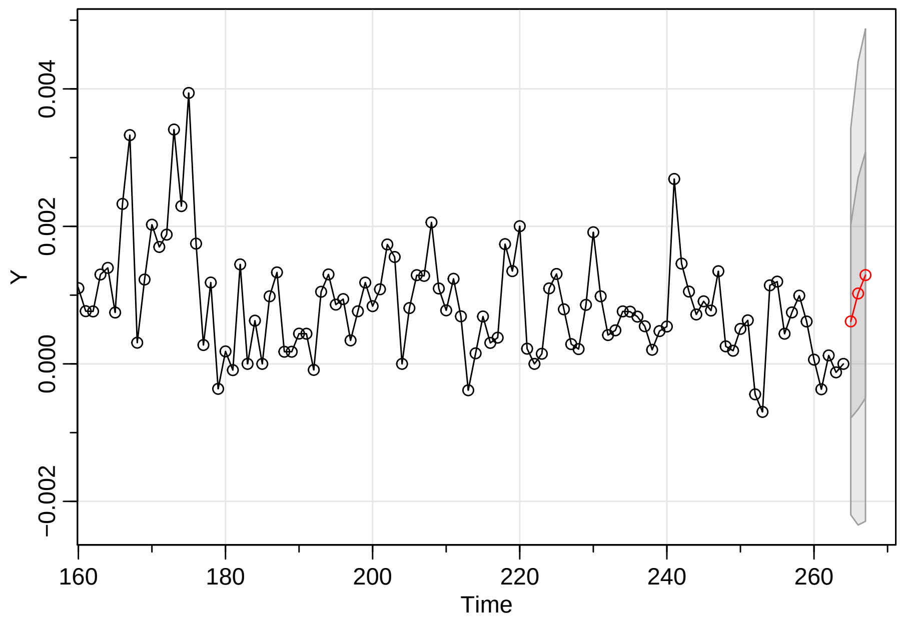

Y
Time
0.004
0.002
0.000
−0.002
160
180
200
220
240
260


Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
260
Time

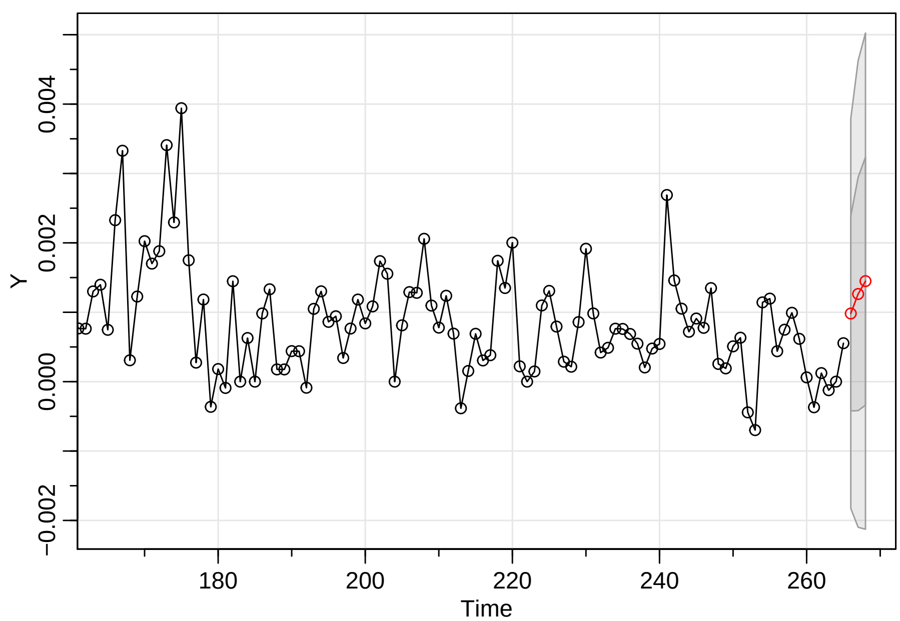

Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
260
Time


Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
260
Time

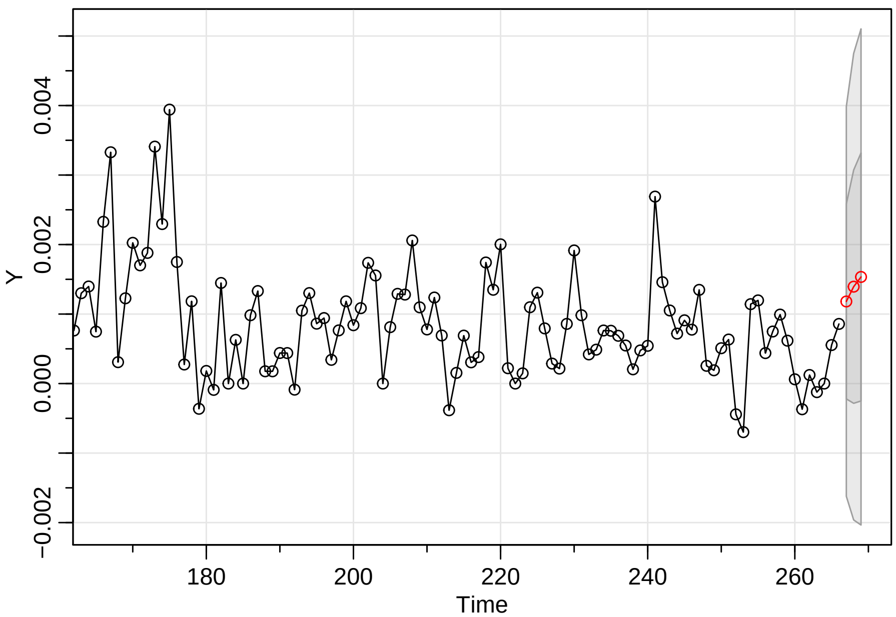

Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
260
Time

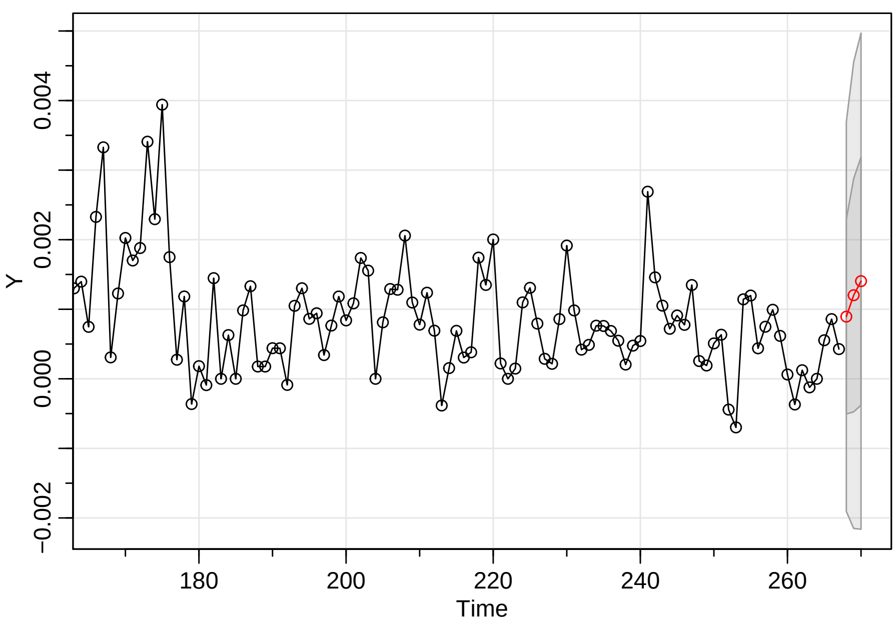


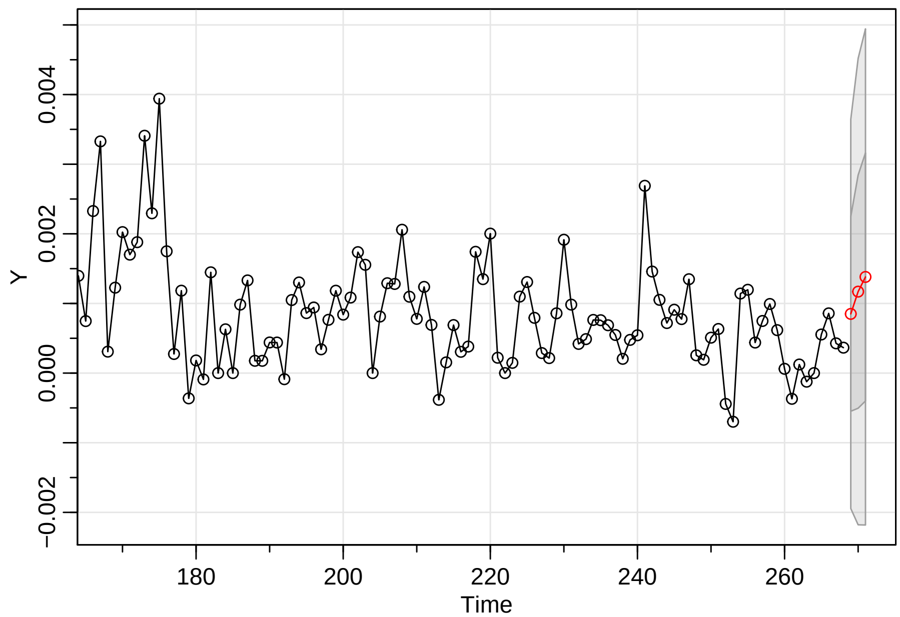
Y
0.004
0.002
0.000
−0.002
160
180
200
220
240
260
Time



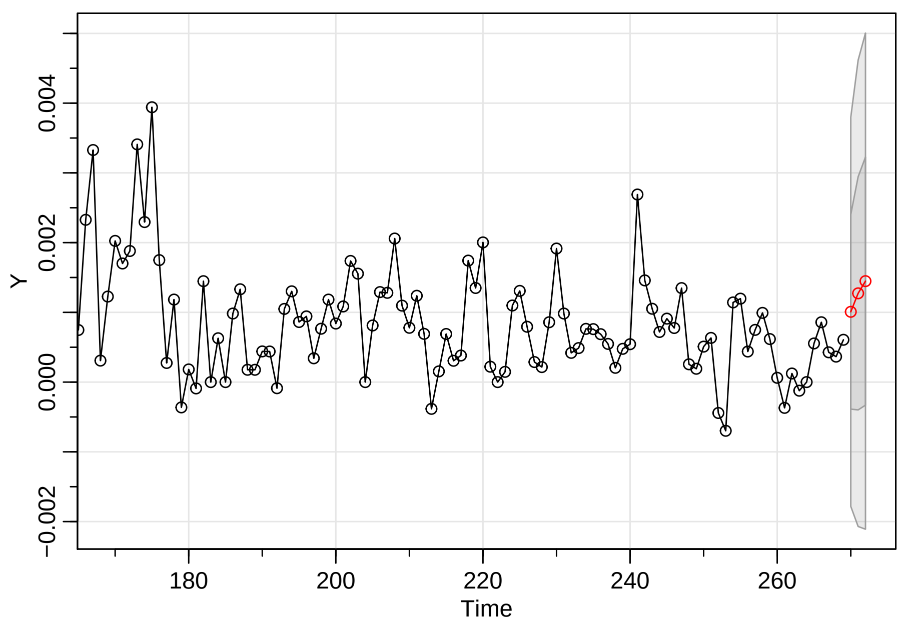
Y
0.004
0.002
0.000
−0.002
180
200
220
240
260
Time


Y
0.004
0.002
0.000
−0.002
180
200
220
240
260
Time

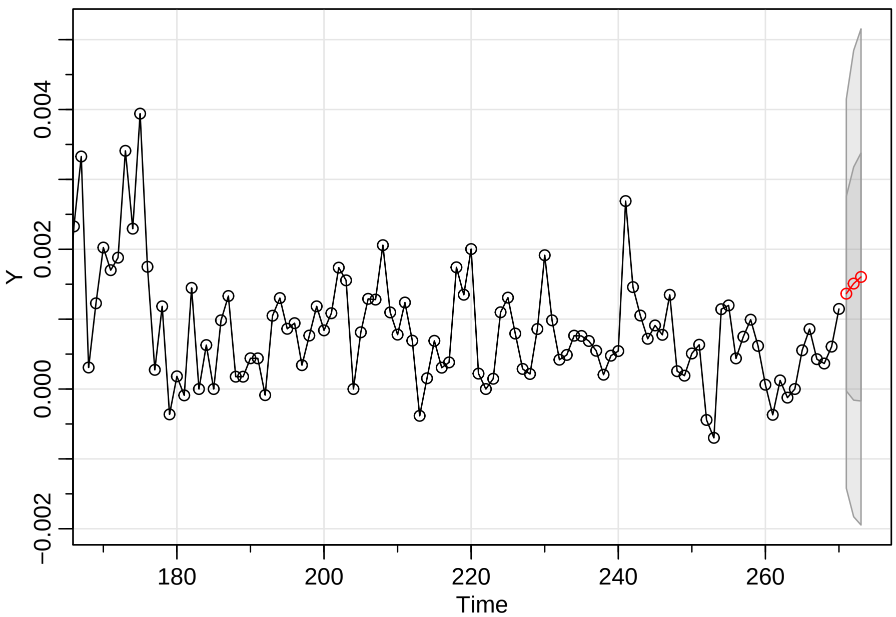

Y
0.004
0.002
0.000
−0.002
180
200
220
240
260
Time

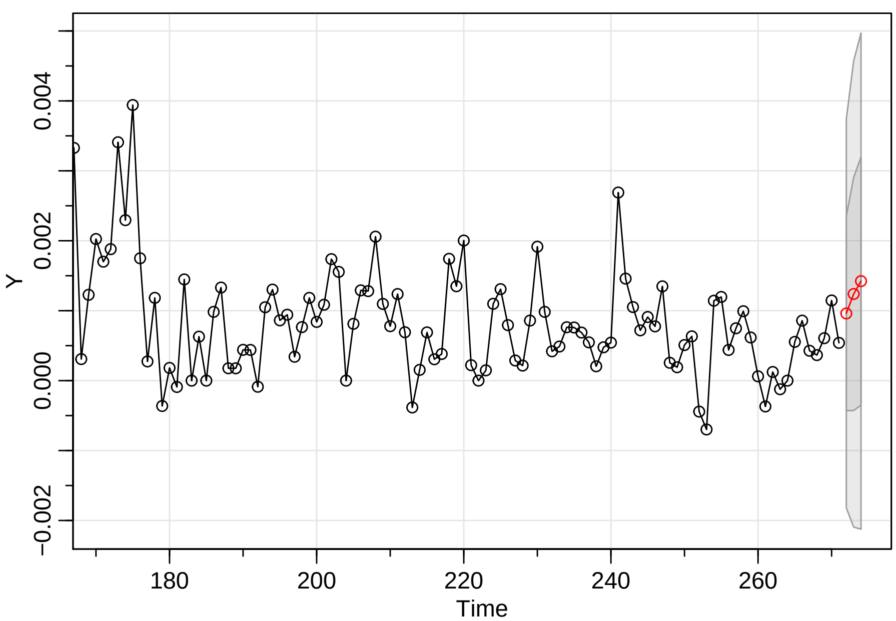


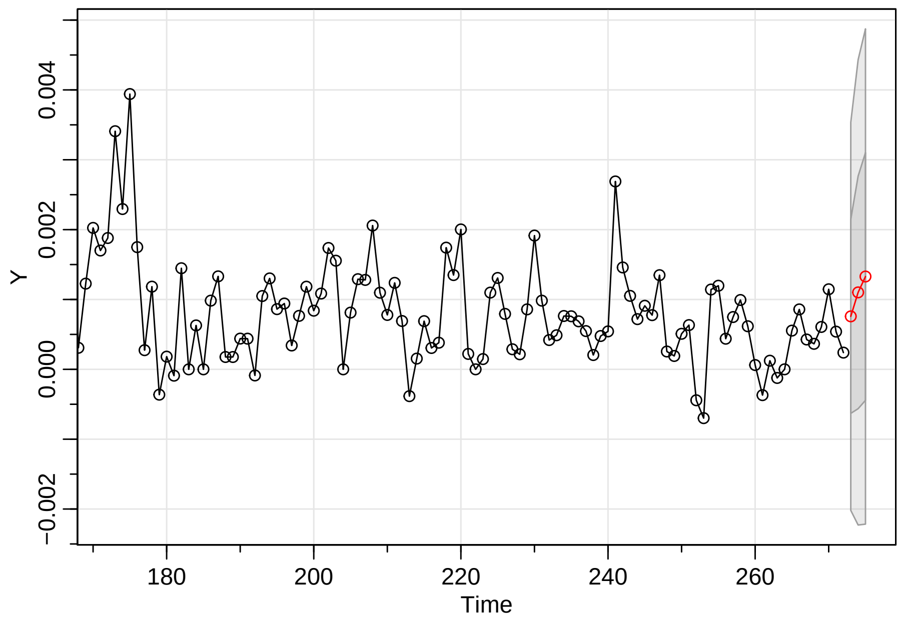
Y
0.004
0.002
0.000
−0.002
180
200
220
240
260
Time

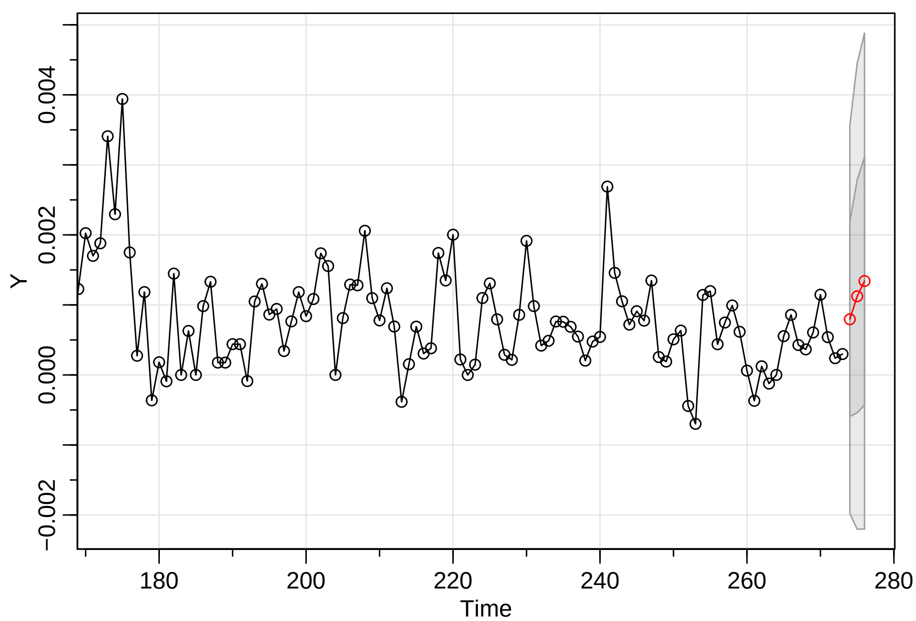


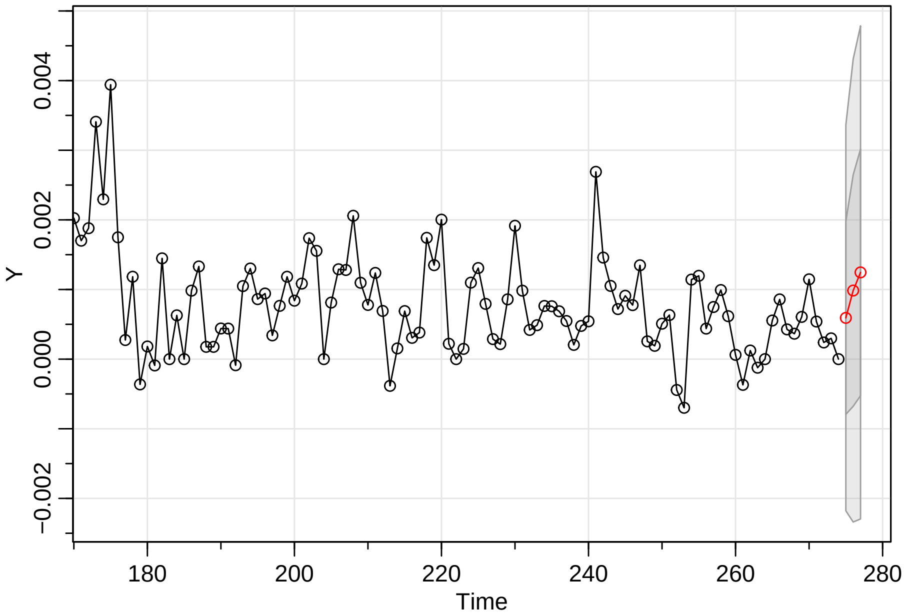
Y
0.004
0.002
0.000
−0.002
180
200
220
240
260
Time

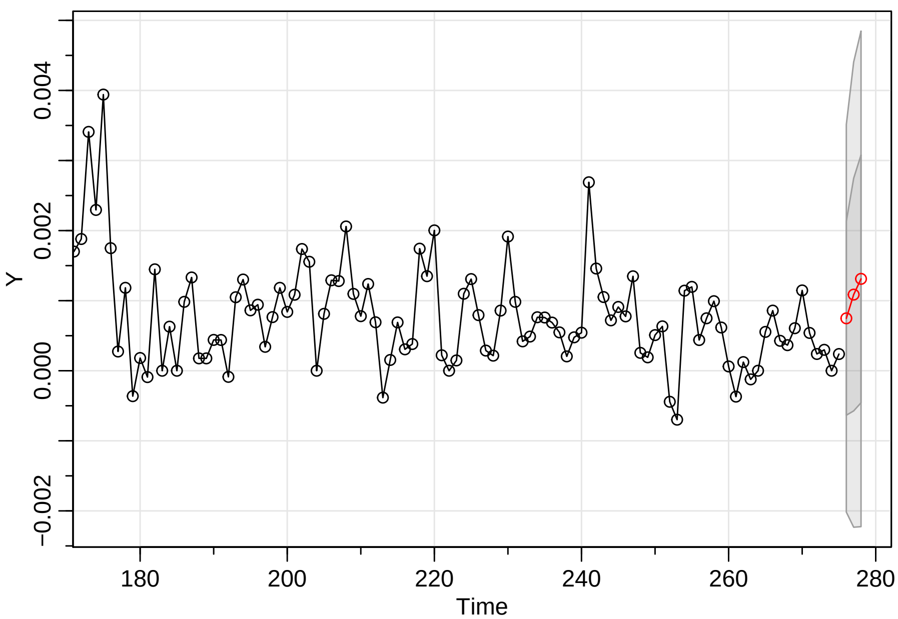

Y
0.004
0.002
0.000
−0.002
180
200
220
240
260
Time

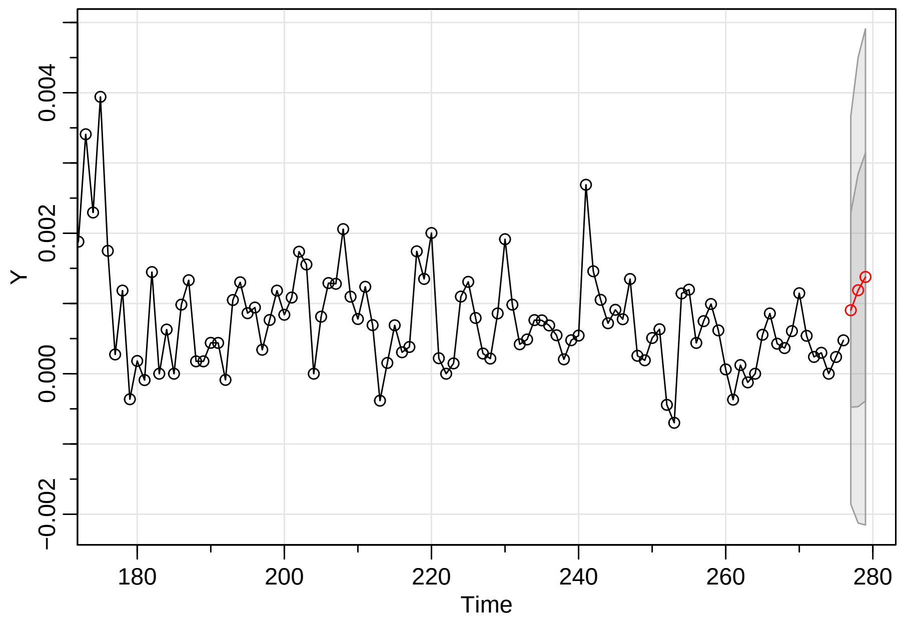

Y
0.004
0.002
0.000
−0.002
180
200
220
240
260
Time

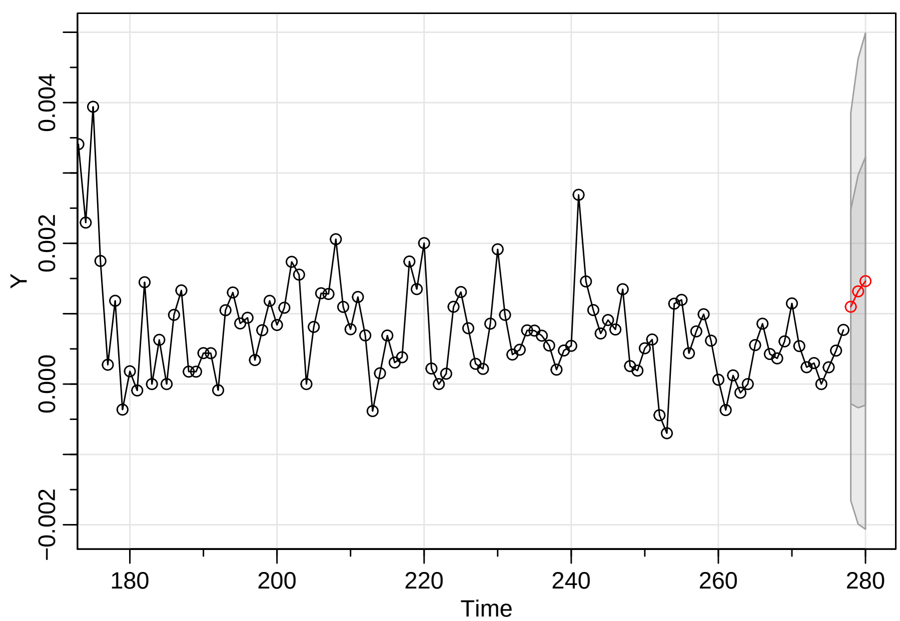

Y
0.004
0.002
0.000
−0.002
180
200
220
240
260
Time

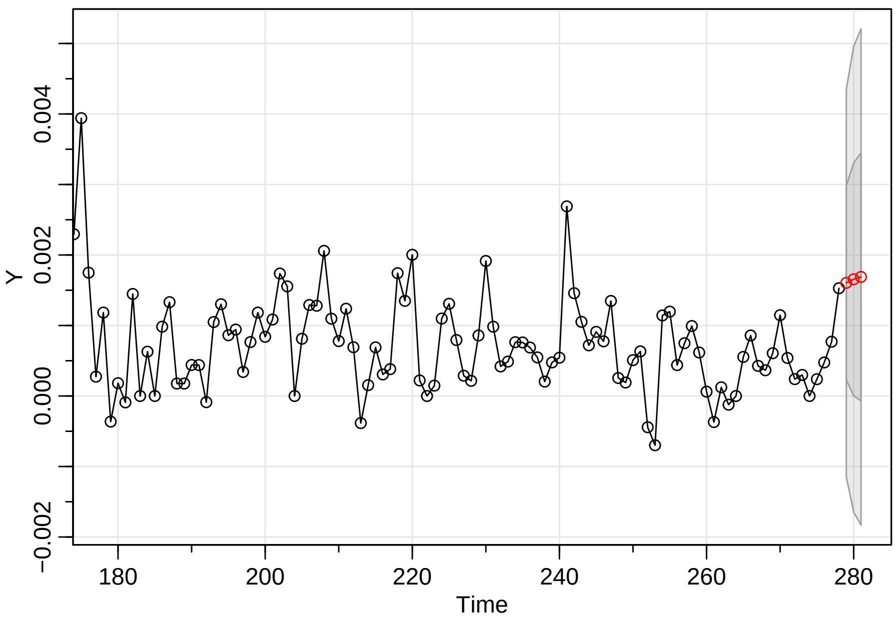

Y
0.004
0.002
0.000
−0.002
180
200
220
240
260
280
Time

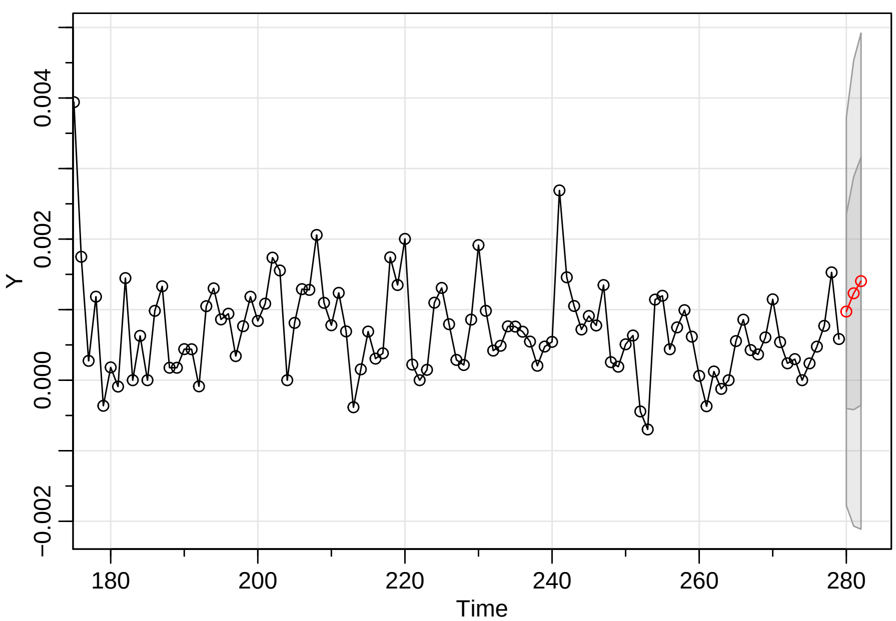

Y
Time
0.004
0.002
0.000
−0.002
180
200
220
240
260
280

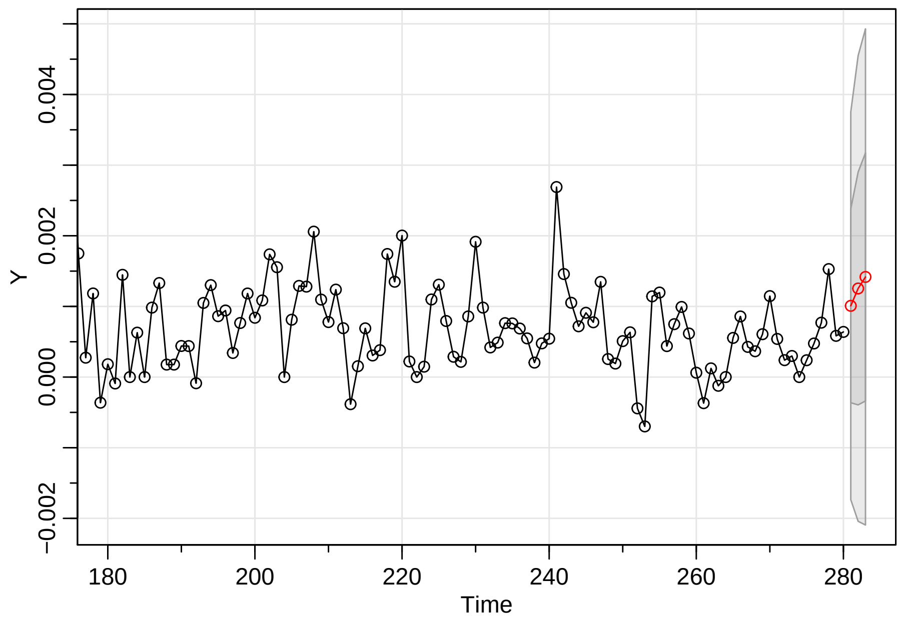

Y
Time
0.004
0.002
0.000
−0.002
180
200
220
240
260
280

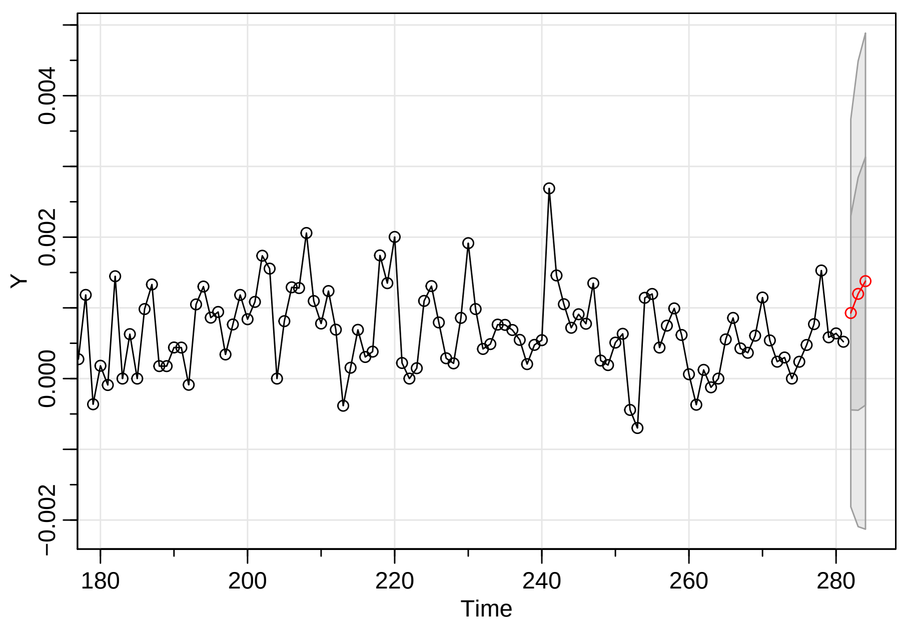

Y
−0.002
0.000
0.002
0.004
180
200
220
240
260
280
Time

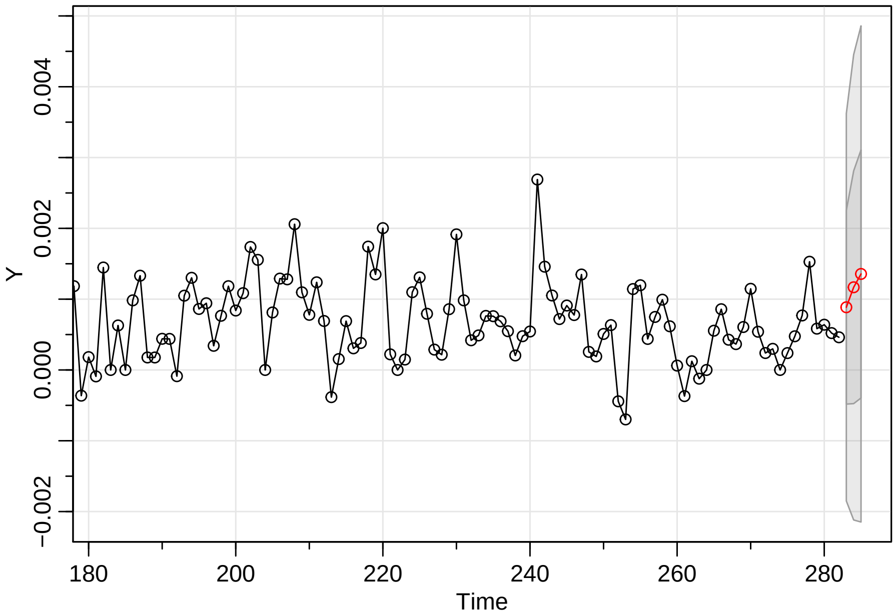

Y
−0.002
0.000
0.002
0.004
180
200
220
240
260
280
Time

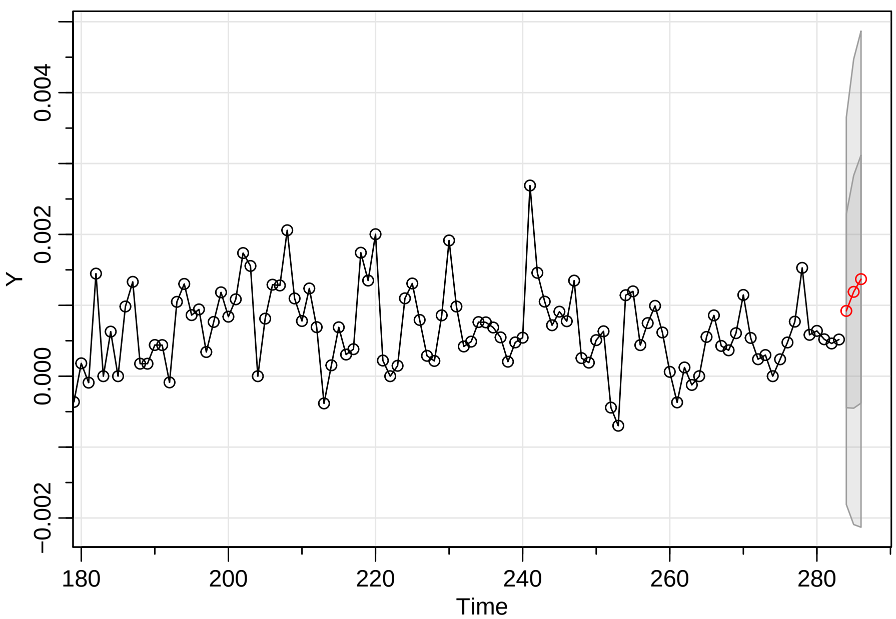

Y
0.004
0.002
0.000
−0.002
180
200
220
240
260
280
Time

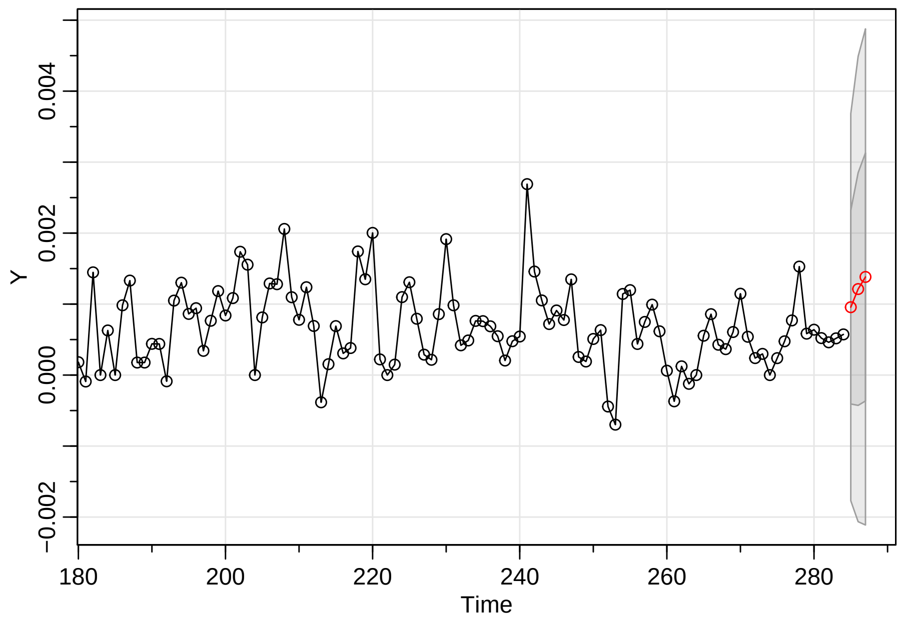

Y
0.004
0.002
0.000
−0.002
180
200
220
240
260
280
Time


Y
Time
−0.002
0.000
0.002
0.004
180
200
220
240
260
280

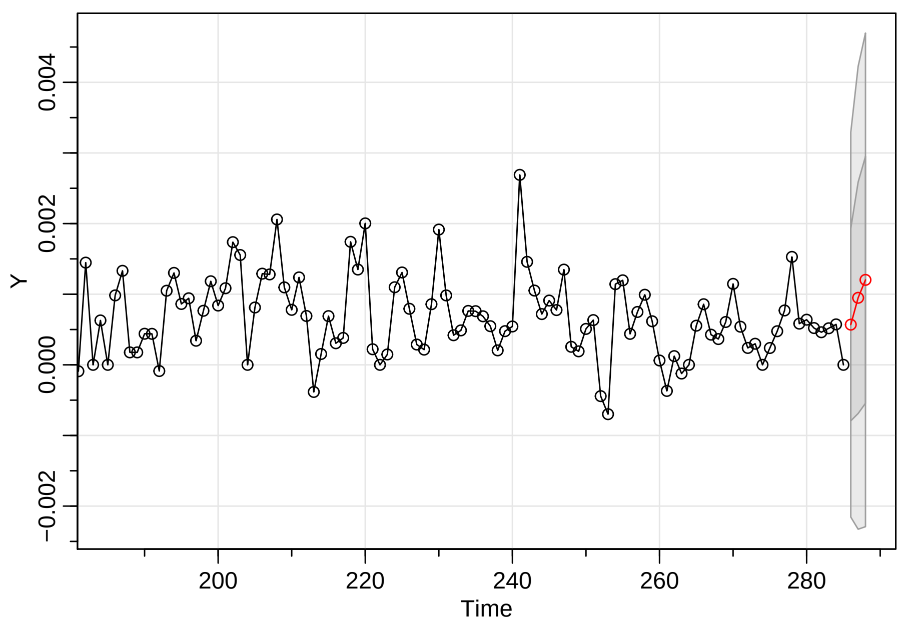

Y
Time
−0.002
0.000
0.002
0.004
180
200
220
240
260
280


Y
Time
180
200
220
240
260
280
−0.002
0.000
0.002
0.004

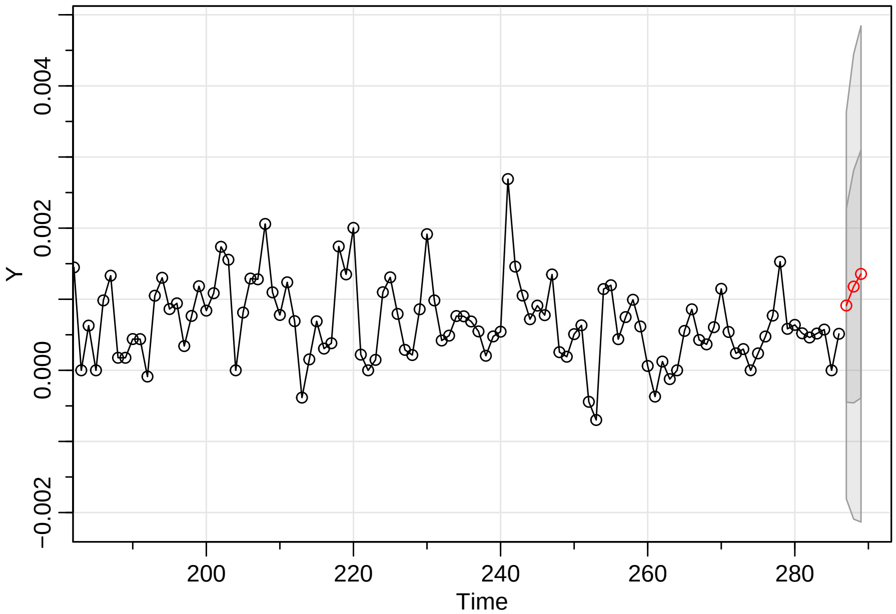

Y
Time
180
200
220
240
260
280
−0.002
0.000
0.002
0.004

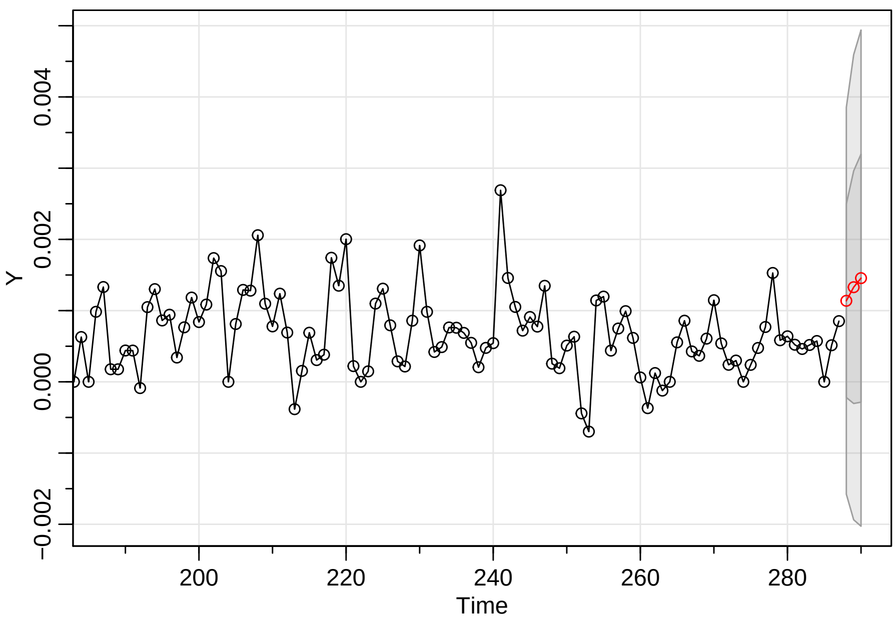


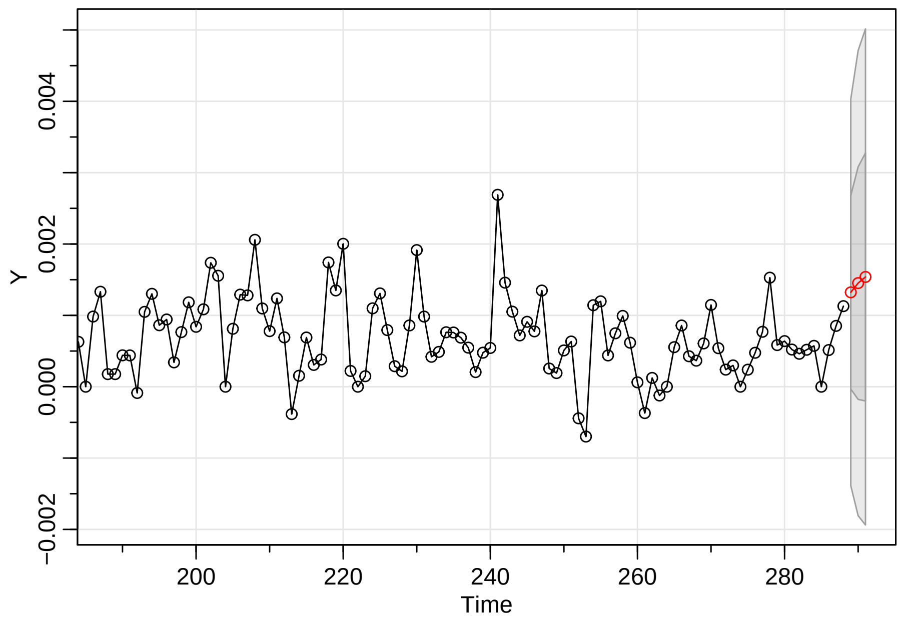
Y
−0.002
0.000
0.002
0.004
180
200
220
240
260
280
Time

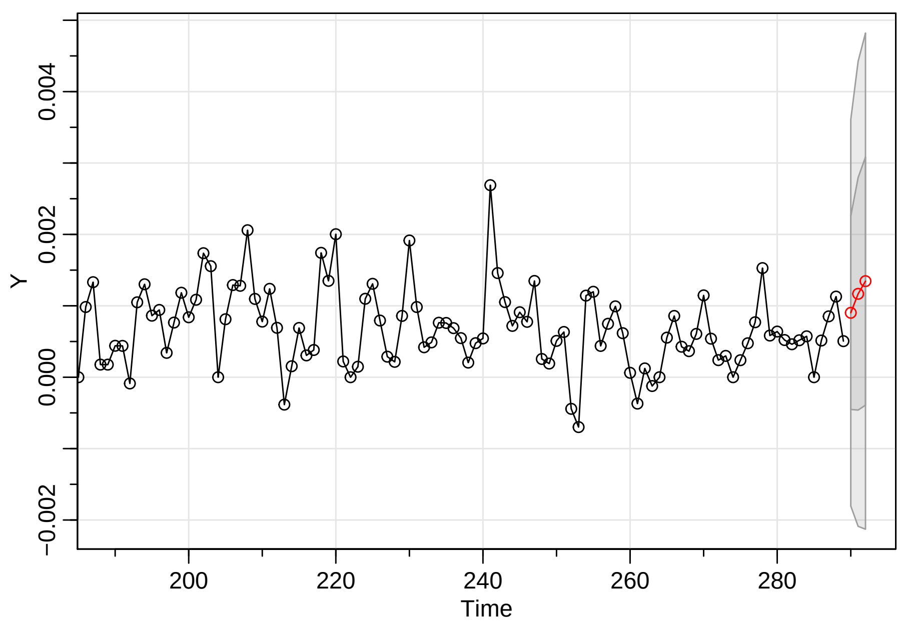


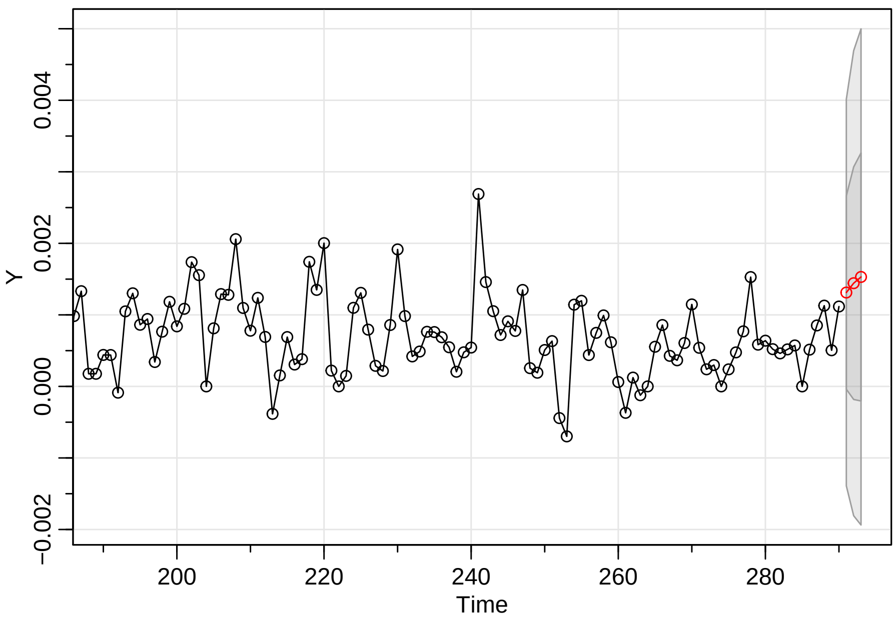
Y
−0.002
0.000
0.002
0.004
200
220
240
260
280
Time


Y
0.004
0.002
0.000
−0.002
200
220
240
260
280
Time

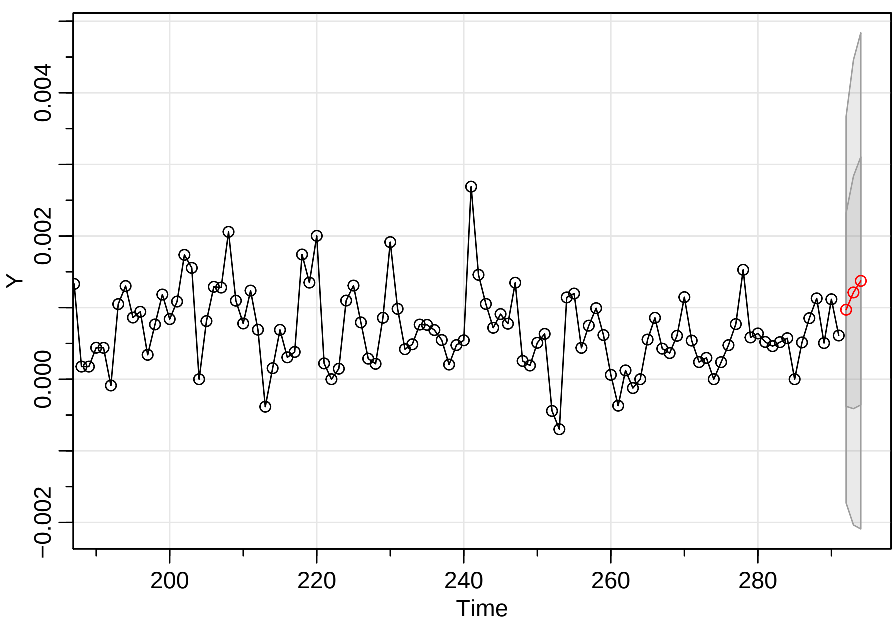

Y
0.004
0.002
0.000
−0.002
200
220
240
260
280
Time


Y
0.004
0.002
0.000
−0.002
200
220
240
260
280
Time

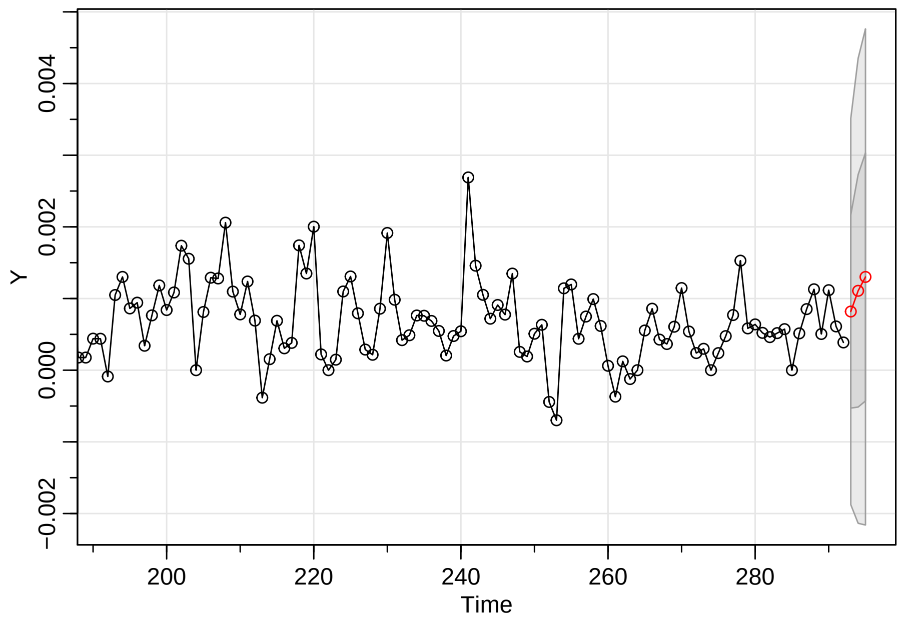

Y
0.004
0.002
0.000
−0.002
200
220
240
260
280
Time



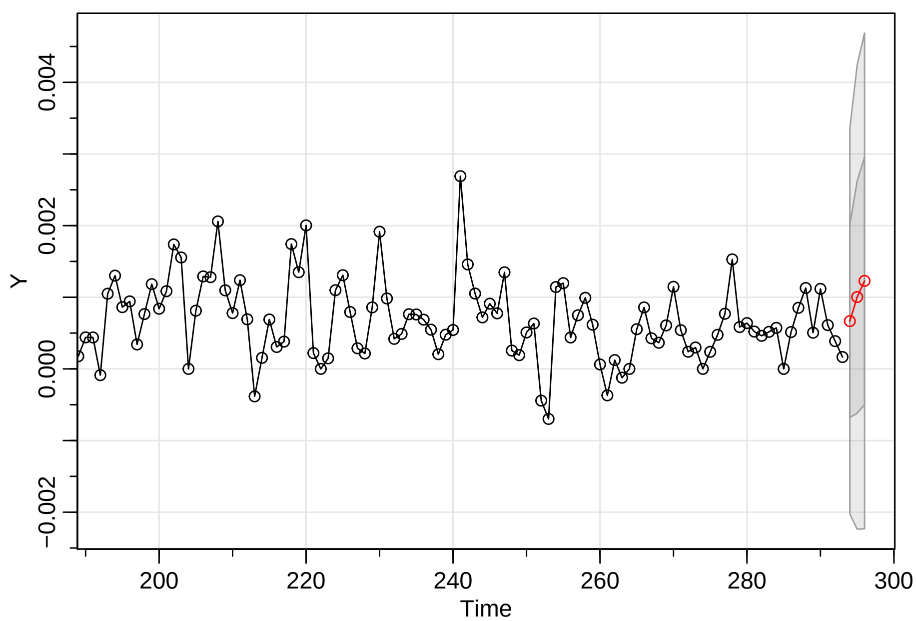
Y
0.004
0.002
0.000
−0.002
200
220
240
260
280
Time



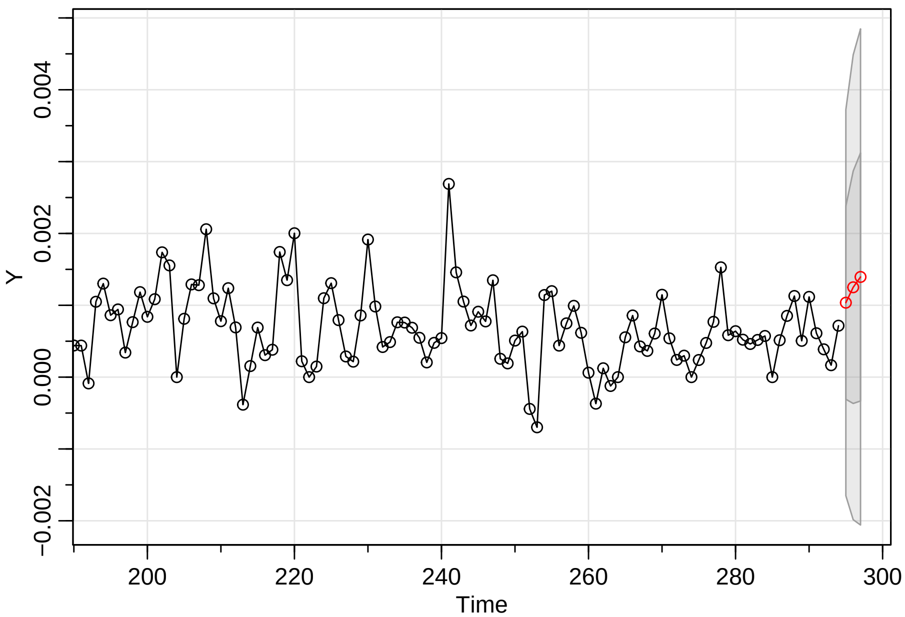
Y
0.004
0.002
0.000
−0.002
200
220
240
260
280
Time

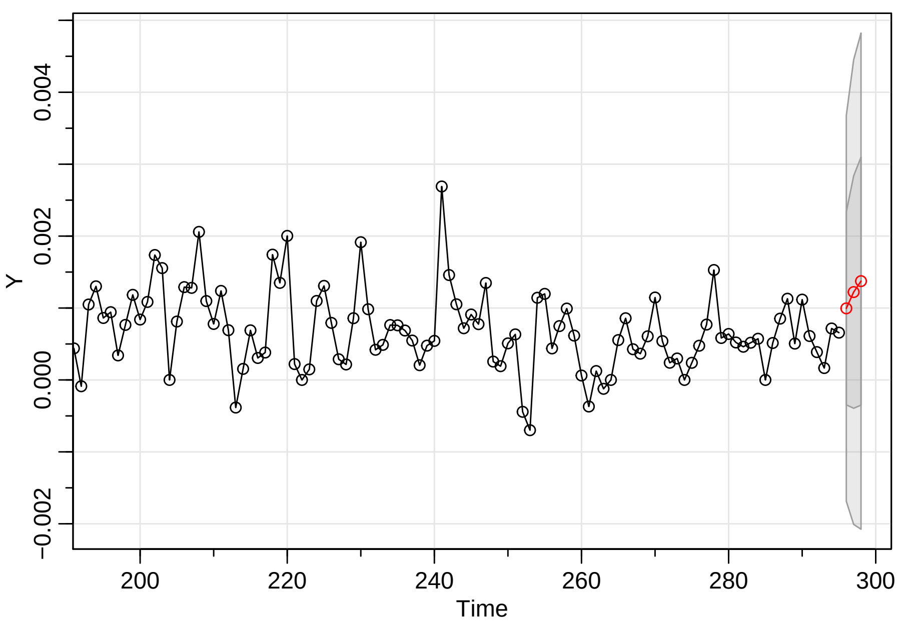


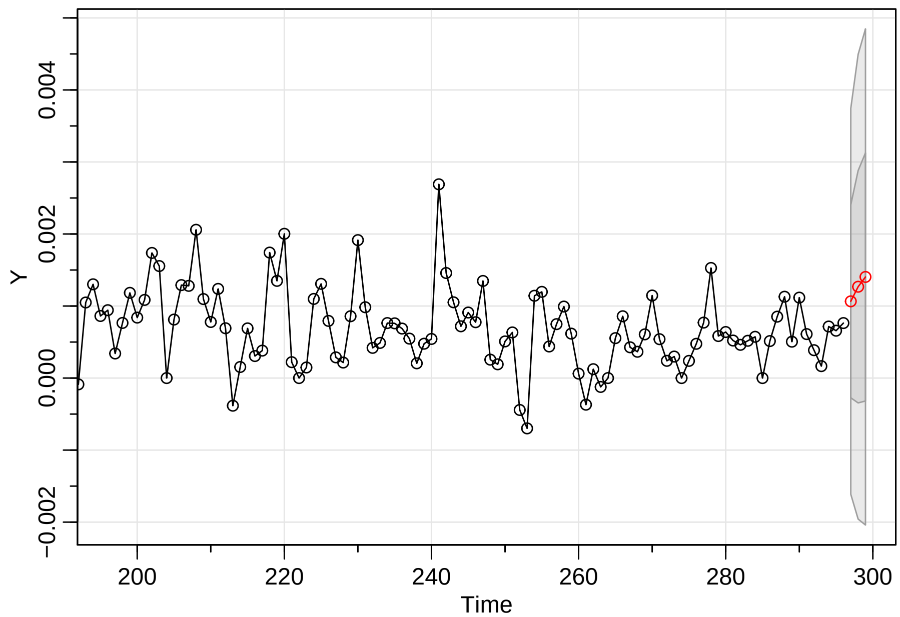
Y
0.004
0.002
0.000
−0.002
200
220
240
260
280
Time

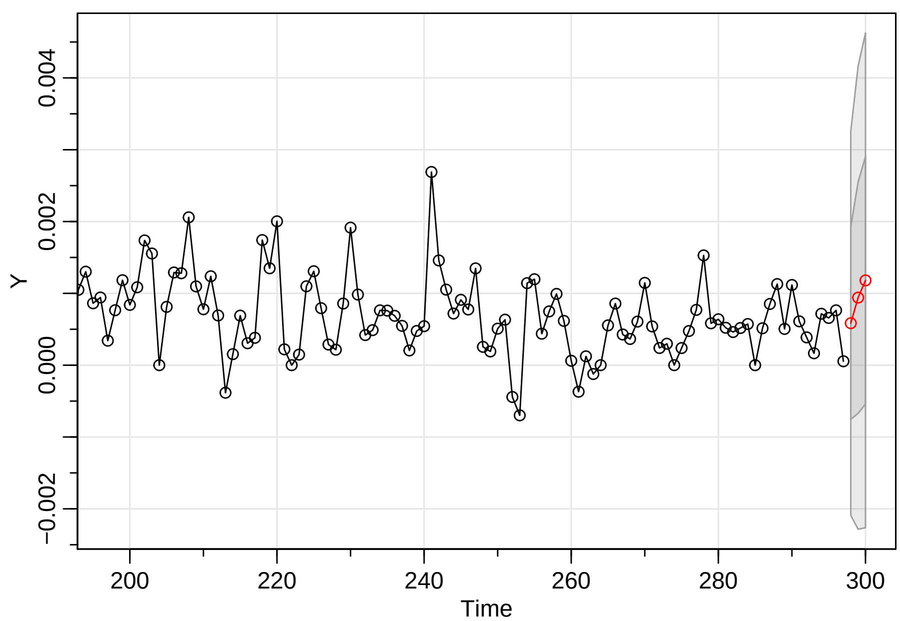


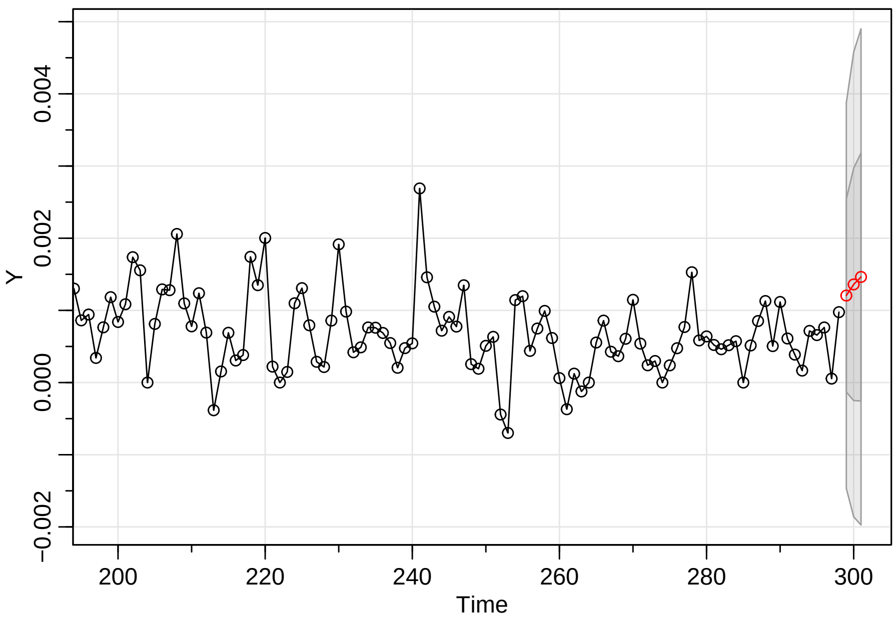
Y
0.004
0.002
0.000
−0.002
200
220
240
260
280
300
Time


Y
Time
0.004
0.002
0.000
−0.002
200
220
240
260
280
300


Y
Time
0.004
0.002
0.000
−0.002
200
220
240
260
280
300


Y
0.004
0.002
0.000
−0.002
200
220
240
260
280
300
Time


Y
0.004
0.002
0.000
−0.002
200
220
240
260
280
300
Time




```
compara2<-merge(INFLA,get(paste('PI_', H, sep='')),join='inner')
compara2<-data.frame(date=index(compara2), coredata(compara2))
compara2$date<-as.Date(compara2$date)
compara2<-filter(compara2, date>="2007-03-01" & date<="2019-02-01")
compara2<-mutate(compara2, DIFF =  (IPC-IPC.1)^2)
ECM_2<-mean(compara2$DIFF)
ECM_2
```

```
## [1] 1.314962e-06
```

Una forma muy usual para comparar dos específicaciones es definir un ratio entre cada uno de los ECM, sí este es mayor a 1 la específicación del que resulta el ECM del denominador sería la mejor.

```
ECM_1/ECM_2
```

```
## [1] 0.4480013
```

Sin embargo, un criterio *formal* para la deducción de referentes predictivos, consiste en conocer sí las diferencias en el ECM entre distintas especificaciones econométricas son significativas desde la perspectiva estadística, ello se realiza a través del test propuesto por Giacomini and White (2006).

La estructura de este test consiste en definir la hipótesis nula que la diferencia en la métrica ECM de una especificación respecto a otra es cero. Luego se construye el estadístico que se denomina Giacomini y White (GW) de acuerdo a la ecuación , el cuál se distribuye asintóticamente normal y se utiliza para contrastar la hipótesis nula.

$$GW_h^{i,j} = \begin{cases} \frac{\bar{\Delta L}_h^{i,j}}{\hat{\sigma}_{g_h}^{i,j}} & \text{si} \quad h = 1 \\ \\ \frac{\bar{\Delta L}_h^{i,j}}{\frac{\hat{\sigma}_{g_h}^{i,j}}{\sqrt{g_h}}} & \text{si} \quad h = \{3, 6, 9, 12\} \end{cases} \quad \forall\ i \neq j \tag{3.3}$$

La aplicación en R sería de la forma siguiente:

```
GW_H<-cbind(compara$date, compara$DIFF, compara2$DIFF)
GW_H<-data.frame(GW_H)
colnames(GW_H)<-c("date", "ARIMA","RW")
GW_H$date<-as.Date(GW_H$date)
GW_H<-mutate(GW_H, delta=ARIMA-RW)
GW_ts<-xts(GW_H[, -1], order.by=as.Date(GW_H$date))
GW_MODEL <- lm(delta ~ 1, data=GW_ts)
list(sqrt(diag(sandwich(GW_MODEL))))
```

```
## [[1]]
##  (Intercept)
## 1.295287e-07
```

Bibliography

Atkeson, A. and Ohanian, L. E. (2001). Are Phillips curves useful for forecasting inflation? *Quarterly Review*, (Win):2–11.

Giacomini, R. and White, H. (2006). Tests of Conditional Predictive Ability. *Econometrica*, 74(6):1545–1578.

Meese, R. A. and Rogoff, K. (1983). Empirical exchange rate models of the seventies : Do they fit out of sample? *Journal of International Economics*, 14(1-2):3–24.