```r
# Load package
library("gmwm")

## Loading required package:  ggplot2

library("tikzDevice")

# Simulate 100 observation from a Gaussian white noise
set.seed(1)
Xt = gen.gts(WN(sigma2 = 1), N = 100)

# Compute autocorrelation
acf_Xt = ACF(Xt)

# Plot autocorrelation
tikz("acf1.tex", width = 6, height = 4)
plot(acf_Xt, show.ci = FALSE)
dev.off()

## pdf
##   2
```

```r
# Load package
library("gmwm")
library("tikzDevice")

# Plot autocorrelation
tikz("acf2.tex", width = 6, height = 4)
plot(acf_Xt)
dev.off()

## pdf
##   2
```

```r
library(gmwm)
library(tikzDevice)
library(gridExtra)

## @knitr example_hydro
# Load data
hydro = read.csv("/Users/stephane/Documents/Time series book/ITS/data/precipitation.csv", he

# Construct gts object
hydro = gts(na.omit(hydro[,2]), start = 1907, freq = 12, name = 'Precipitation Data',
            unit = "month")
```

```
# Plot data
tikz("HYDRO2.tex", width = 8, height = 4)
a = autoplot(hydro)  +  ylab("Mean Monthly Precipitation (mm)")
b = plot(ACF(hydro))
grid.arrange(a, b, nrow = 1)
dev.off()

## pdf
##   2
```

```
library(gmwm)
library(tikzDevice)
library(gridExtra)

## @knitr example_hydro
# Load data
hydro = read.csv("/Users/stephane/Documents/Time series book/ITS/data/precipitation.csv", he

# Construct gts object
hydro = gts(na.omit(hydro[,2]), start = 1907, freq = 12, name = 'Precipitation Data',
            unit = "month")

# Compute p-values for varying lags
lags = 7:30
pval = rep(NA, length(lags))
for (i in 1:length(lags)){
  pval[i] = Box.test(hydro, lag = lags[i])$p.val
}

# Plot data
tikz("HYDRO3.tex", width = 6, height = 4)
plot(NA, xlim = range(lags), ylim = c(0, max(pval)), xlab = "$h$", ylab = "P-value")
grid()
abline(h=0.05, lwd = 2, col = "red")
points(lags, pval, pch = 16, col = "blue", lty = 2, type = "b")
dev.off()

## pdf
##   2
```

```
library(gmwm)
library(tikzDevice)
library(gridExtra)
```

```r
# Simulate Xt
set.seed(1)
model = AR1(phi = 0.9, sigma2 = 1)
Xt = gen.gts(model)

# Construct Yt
epsilon = 0.01
nb_outlier = rbinom(1,length(Xt),epsilon)
Yt = Xt
Yt[sample(1:length(Xt),nb_outlier)] = rnorm(nb_outlier,0,10)

# Add names
Xt = gts(Xt, name = "$(X_t)$")
Yt = gts(Yt, name = "$(Y_t)$")

# Plot data
tikz("rob1.tex", width = 8, height = 6)
a = autoplot(Xt) + ylim(range(Yt)) + ylab("$(X_t)$")
b = autoplot(Yt) + ylab("$(Y_t)$")
grid.arrange(a, b, nrow = 2)
dev.off()

## pdf
##   2
```

```r
library(gmwm)
library(tikzDevice)
library(gridExtra)

# Plot autocorrelation
tikz("rob2.tex", width = 8, height = 4)
a = plot(ACF(Xt))
b = plot(ACF(Yt))
grid.arrange(a, b, nrow = 1)
dev.off()

## pdf
##   2
```

```r
library(gmwm)
library(tikzDevice)
library(gridExtra)
library(robcor)
```

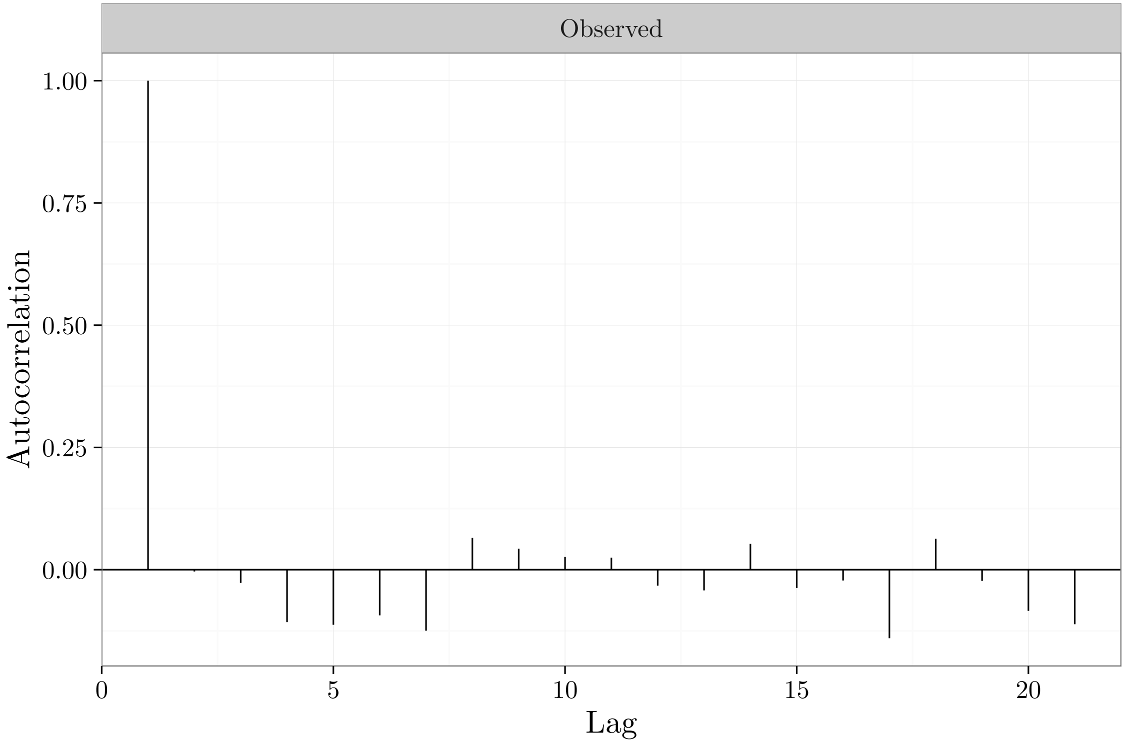
```
ACF.Xt = ACF(Xt)
ACF.Yt = ACF(Yt)

ACF.Xt[,,] = robacf(Xt, plot=FALSE)$acf
ACF.Yt[,,] = robacf(Yt, plot=FALSE)$acf

# Plot autocorrelation
tikz("rob3.tex", width = 8, height = 4)
a = plot(ACF.Xt)
b = plot(ACF.Yt)
grid.arrange(a, b, nrow = 1)
dev.off()

## pdf
##   2
```

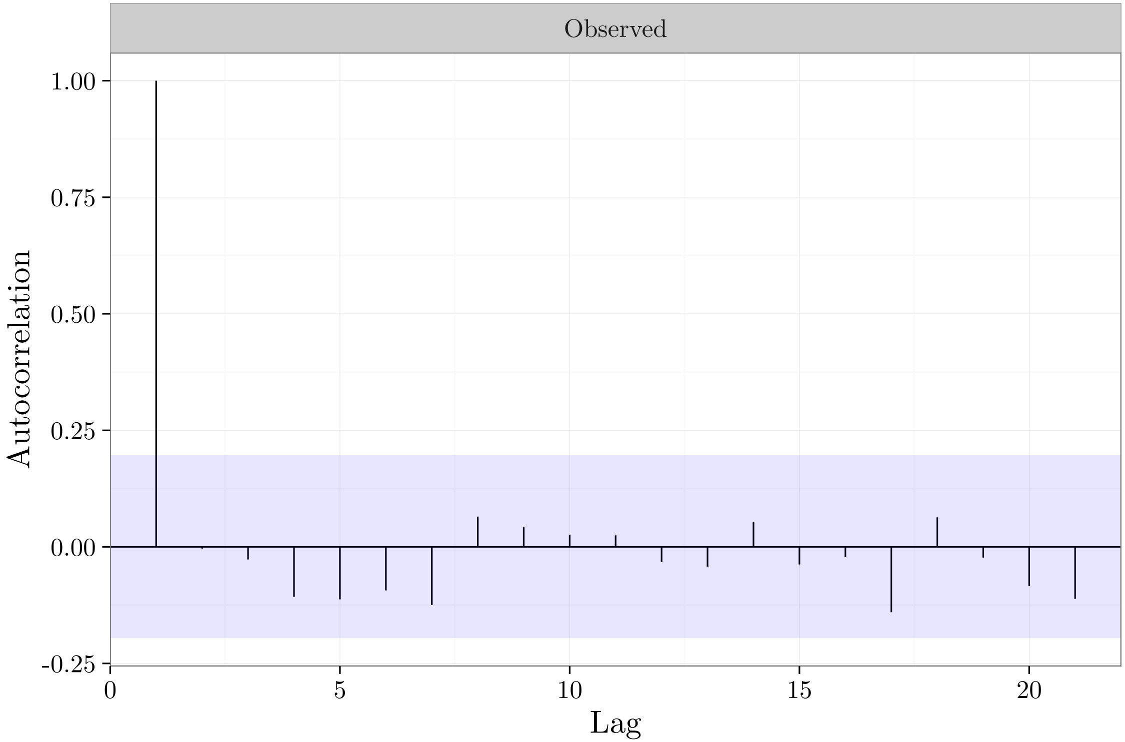
```
library(gmwm)
library(tikzDevice)
library(gridExtra)

# Load packages
library("robcor")

# Define sample size
n = 1000

# Define proportion of "extreme" observation
alpha = 0.01

# Extreme observation are generated from N(0,sigma2.cont)
sigma2.cont = 10

# Number of Monte-Carlo replications
B = 1000

# Define model AR(1)
phi = 0.9
sigma2 = 1
model = AR1(phi = phi, sigma2 = sigma2)

# Initialization of result array
result = array(NA, c(B,2,20))

# Start Monte-Carlo
for (i in 1:B){
```

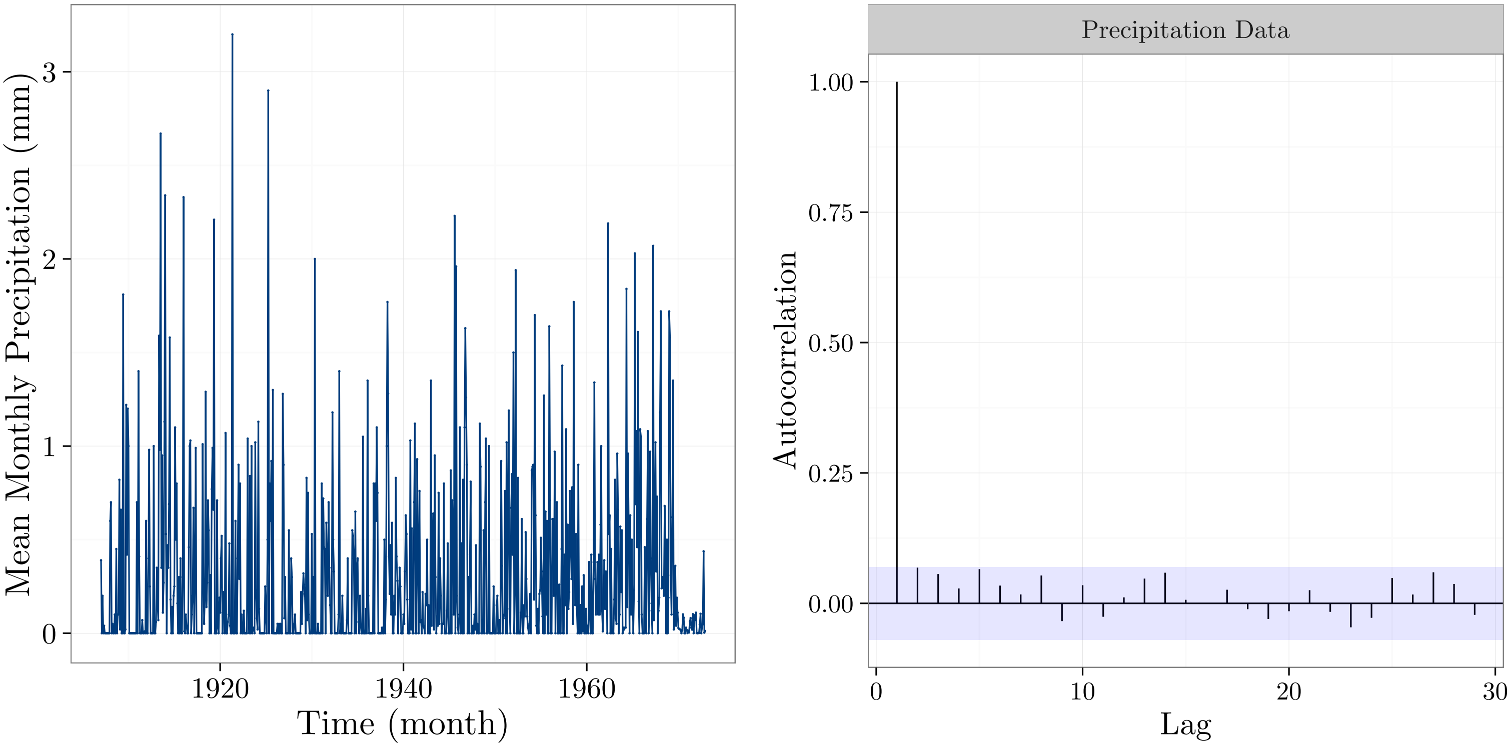
```
  # Simulate AR(1)
  Xt = gen.gts(model, N = n)

  # Compute standard and robust ACF of Xt and Yt
  acf = ACF(Xt)
  rob_acf = robacf(Xt, plot=FALSE)$acf

  # Store ACFs
  result[i,1,] = acf[1:20]
  result[i,2,] = rob_acf[1:20]
}


# Compare empirical distribution of standard and robust ACF based on Xt

# Vector of lags considered (h <= 20)
lags = c(1,2,5,10) + 1

tikz("rob4.tex", width = 8, height = 6)
# Make graph
par(mfrow = c(2,2))

for (i in 1:4){
  boxplot(result[,1,lags[i]], result[,2,lags[i]], col = "lightgrey",
          names = c("Standard","Robust"), main = paste("lag: h = ", lags[i]-1),
          ylab = "Sample autocorrelation")
  abline(h = phi^(lags[i]-1), col = 2, lwd = 2)
}
dev.off()

## pdf
##   2
```

```
library(gmwm)
library(tikzDevice)
library(gridExtra)

# Load packages
library("robcor")

# Define sample size
n = 1000

# Define proportion of "extreme" observation
alpha = 0.01
```

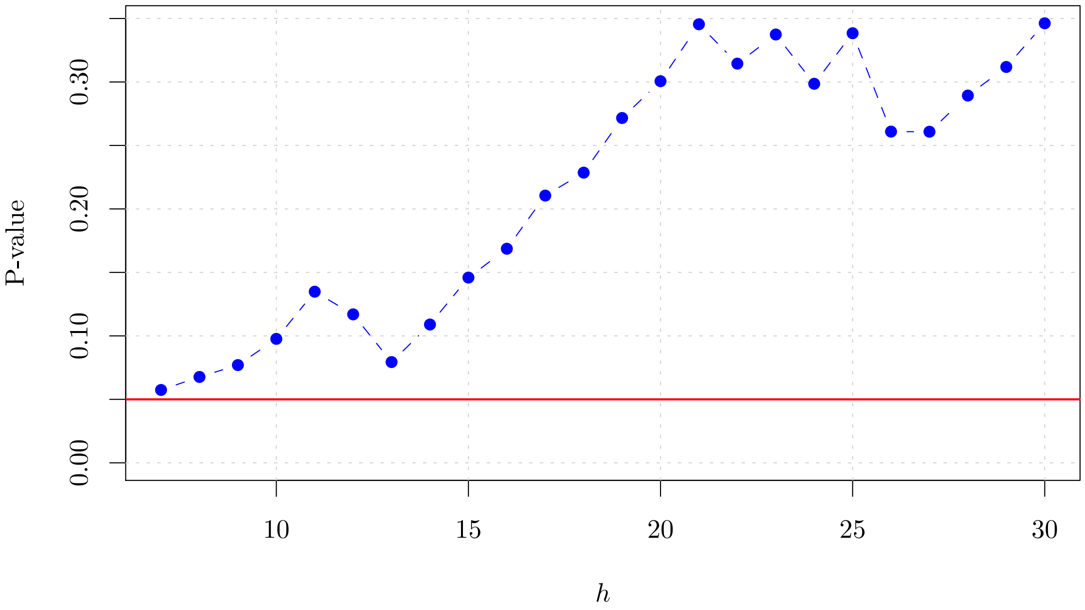
```r
# Extreme observation are generated from N(0,sigma2.cont)
sigma2.cont = 10

# Number of Monte-Carlo replications
B = 1000

# Define model AR(1)
phi = 0.9
sigma2 = 1
model = AR1(phi = phi, sigma2 = sigma2)

# Initialization of result array
result = array(NA, c(B,2,20))

# Start Monte-Carlo
for (i in 1:B){
  # Simulate AR(1)
  Xt = gen.gts(model, N = n)

  # Add a proportion alpha of extreme observations to Yt
  Xt[sample(1:n,round(alpha*n))] = rnorm(round(alpha*n), 0, sigma2.cont)

  # Compute standard and robust ACF of Xt and Yt
  acf = ACF(Xt)
  rob_acf = robacf(Xt, plot=FALSE)$acf

  # Store ACFs
  result[i,1,] = acf[1:20]
  result[i,2,] = rob_acf[1:20]
}


# Compare empirical distribution of standard and robust ACF based on Xt

# Vector of lags considered (h <= 20)
lags = c(1,2,5,10) + 1

tikz("rob5.tex", width = 8, height = 6)
# Make graph
par(mfrow = c(2,2))

for (i in 1:4){
  boxplot(result[,1,lags[i]], result[,2,lags[i]], col = "lightgrey",
          names = c("Standard","Robust"), main = paste("lag: h = ", lags[i]-1),
          ylab = "Sample autocorrelation")
```

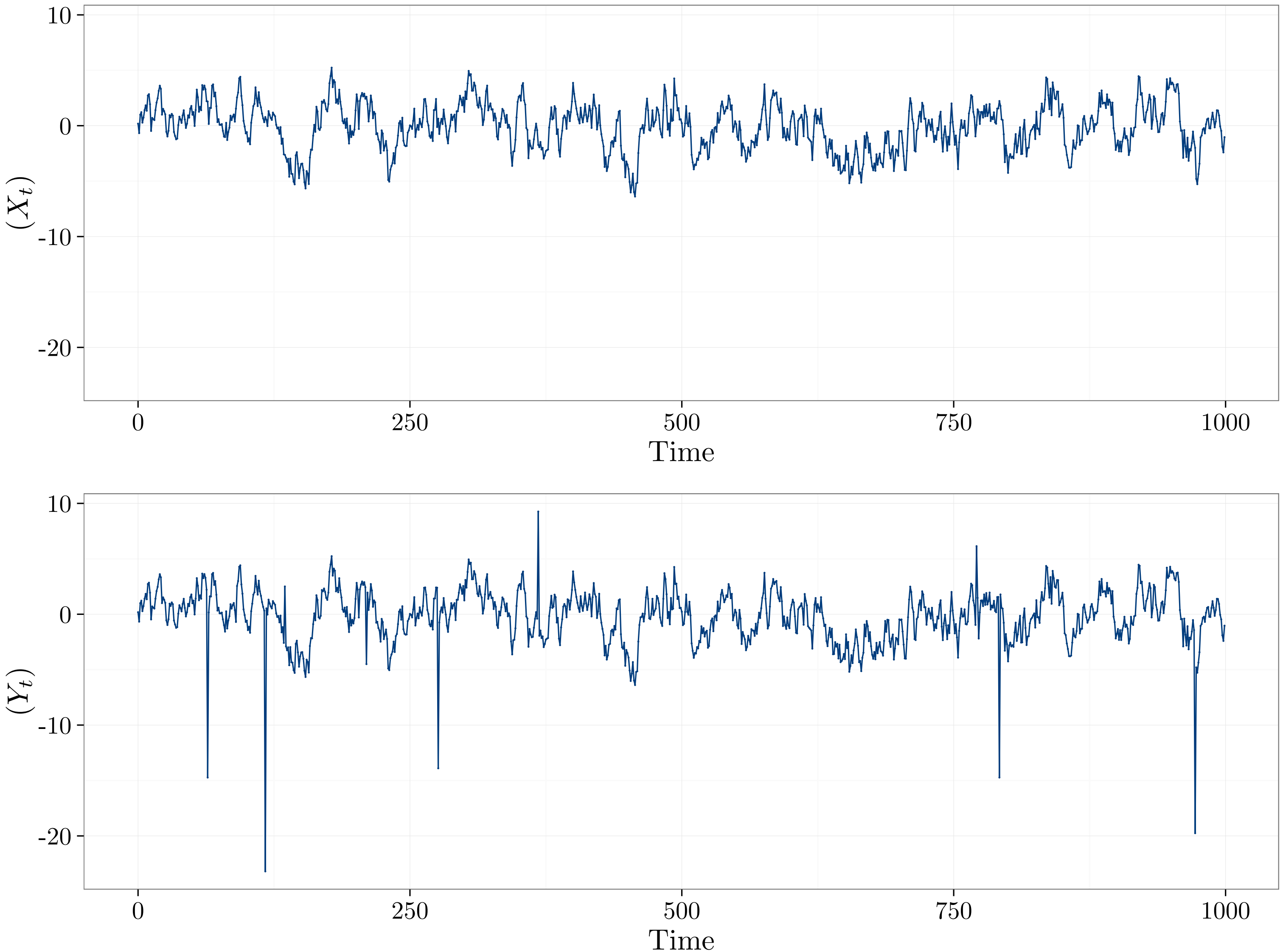

Observed
1.00
0.75
0.50
0.25
0.00
Autocorrelation
0
5
10
15
20
Lag


Observed
1.00
0.75
0.50
0.25
0.00
-0.25
0
5
10
15
20
Lag
Autocorrelation

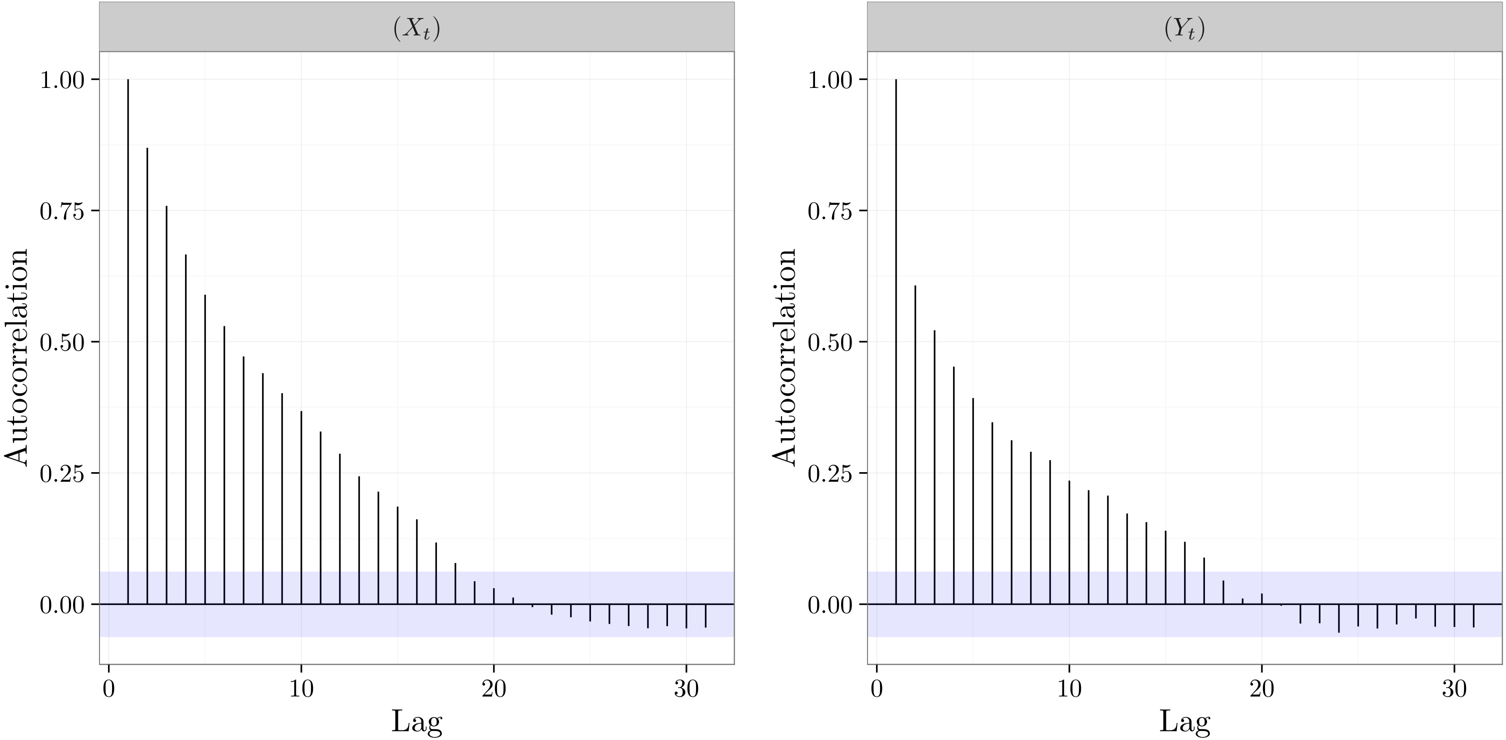


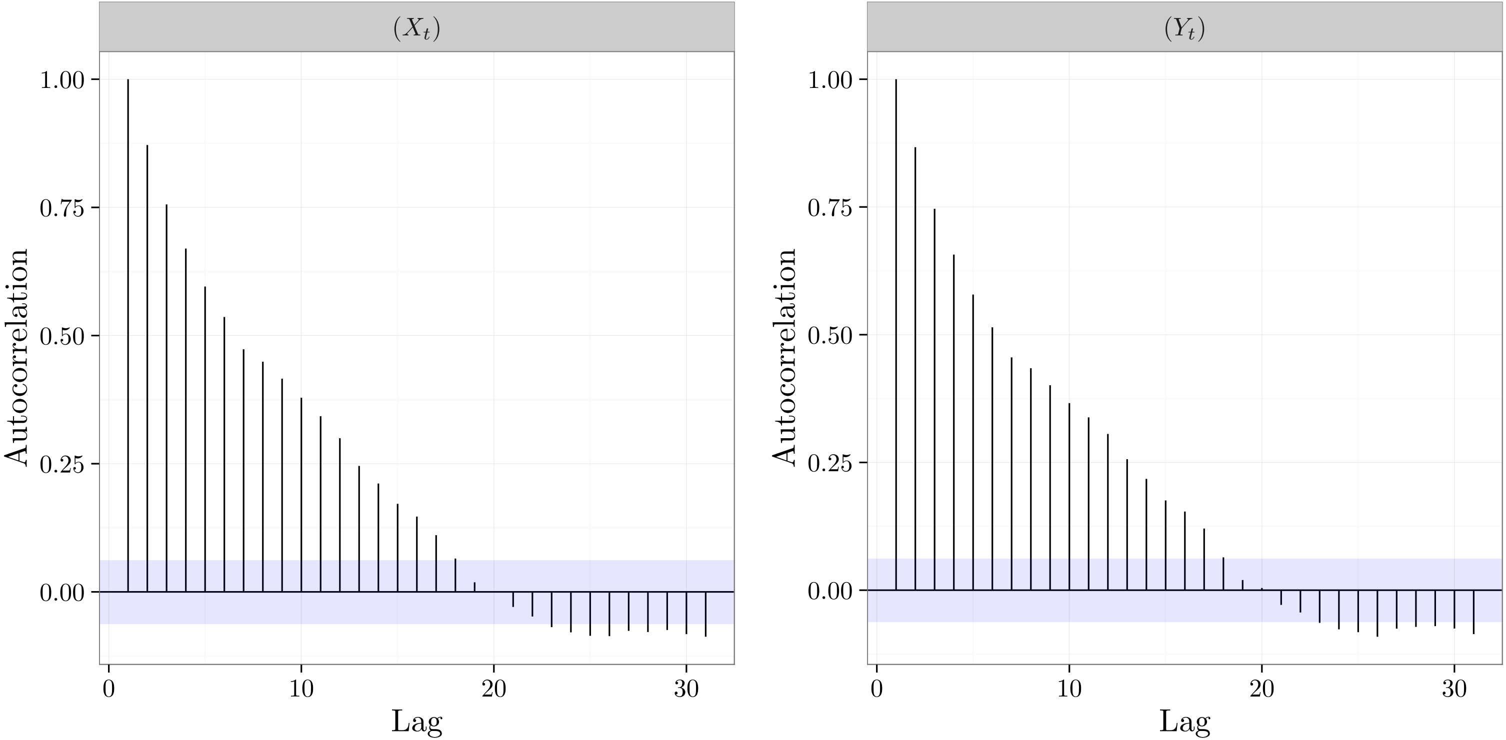
Mean Monthly Precipitation (mm)
0
1
2
3
1920
1940
1960
Time (month)
Precipitation Data
Autocorrelation
0.00
0.25
0.50
0.75
1.00
0
10
20
30
Lag


0.00
0.10
0.20
0.30
P-value
10
15
20
25
30
h

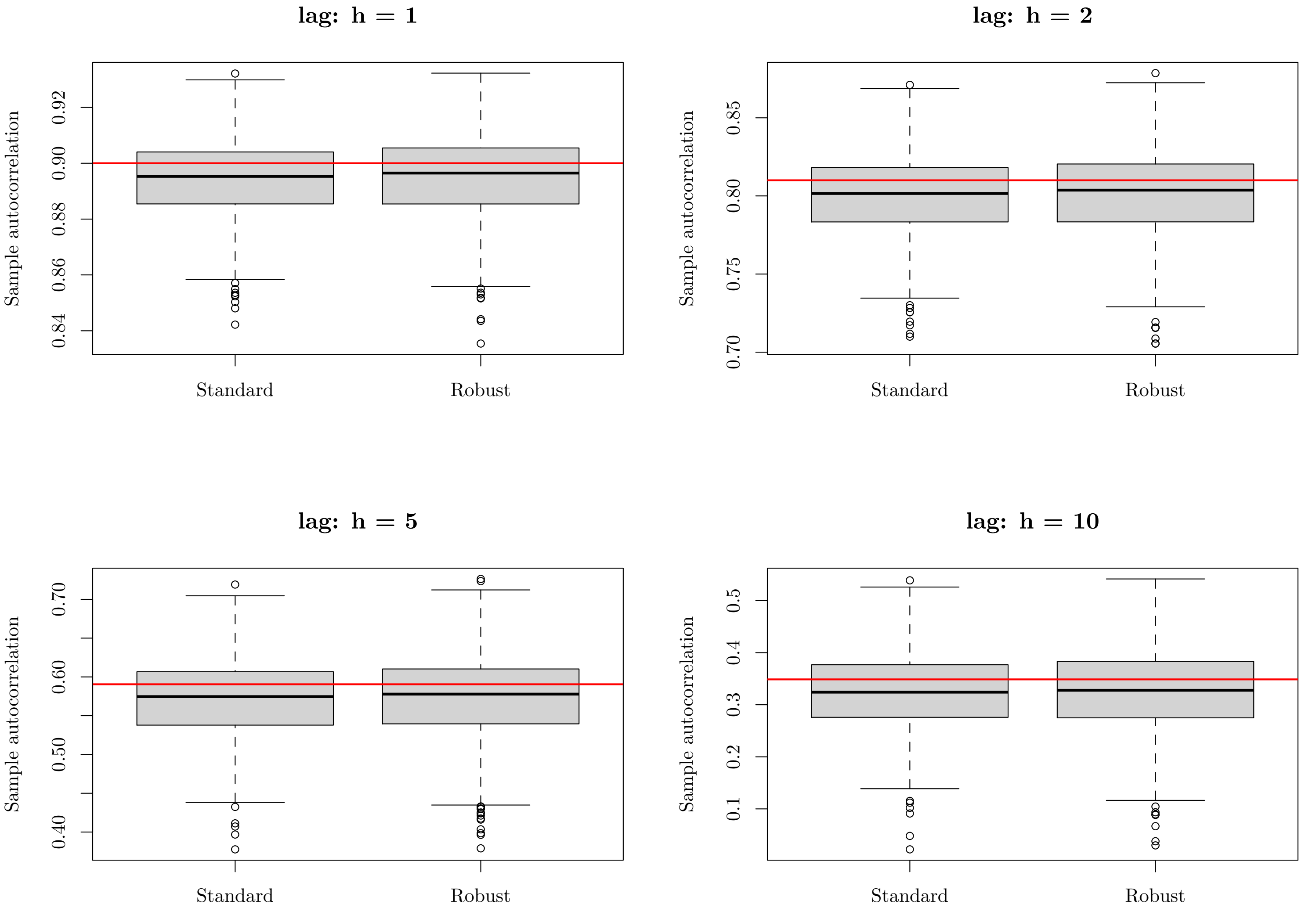


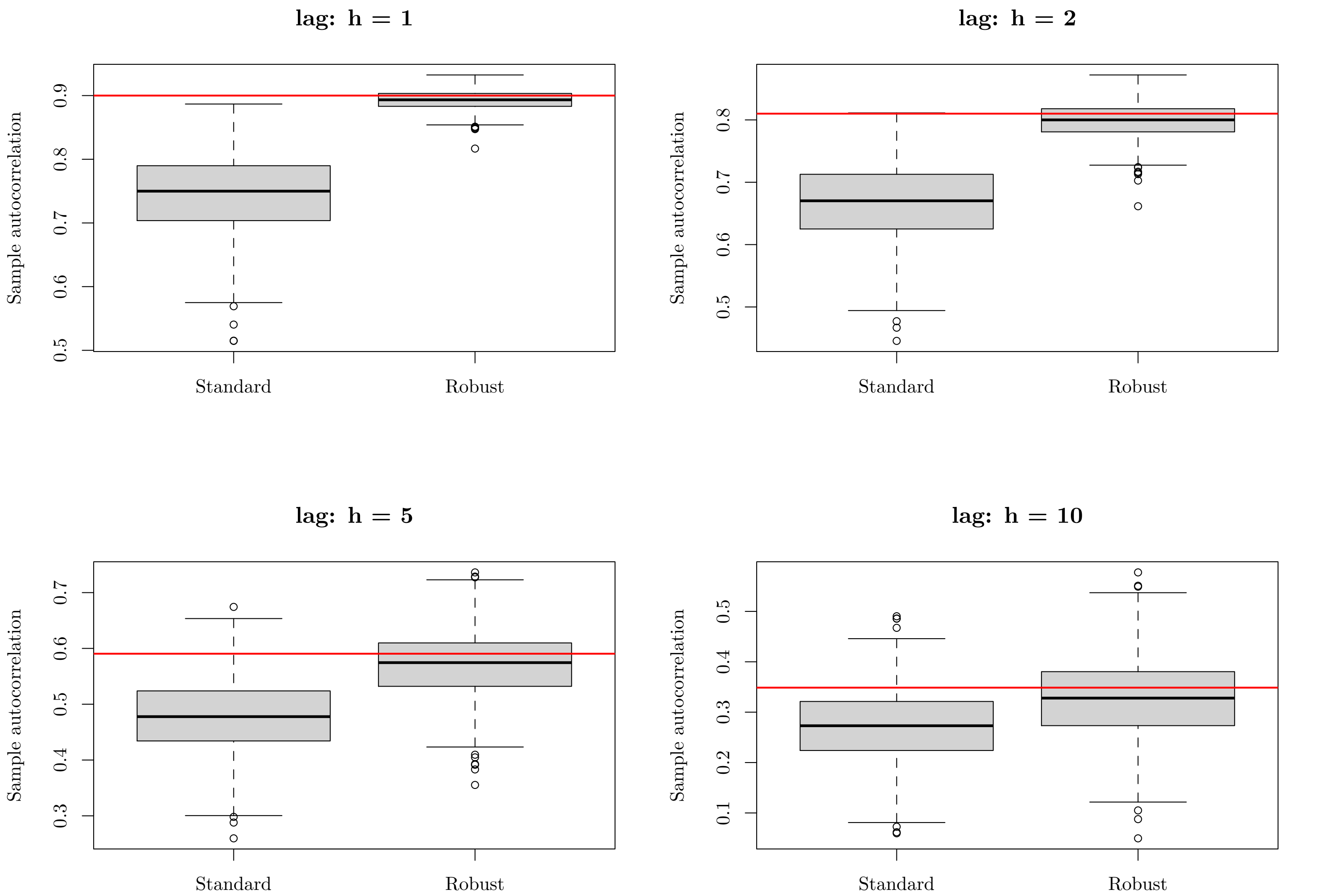
10
0
-10
-20
(X_t)
0
250
500
750
1000
Time
10
0
-10
-20
(Y_t)
0
250
500
750
1000
Time

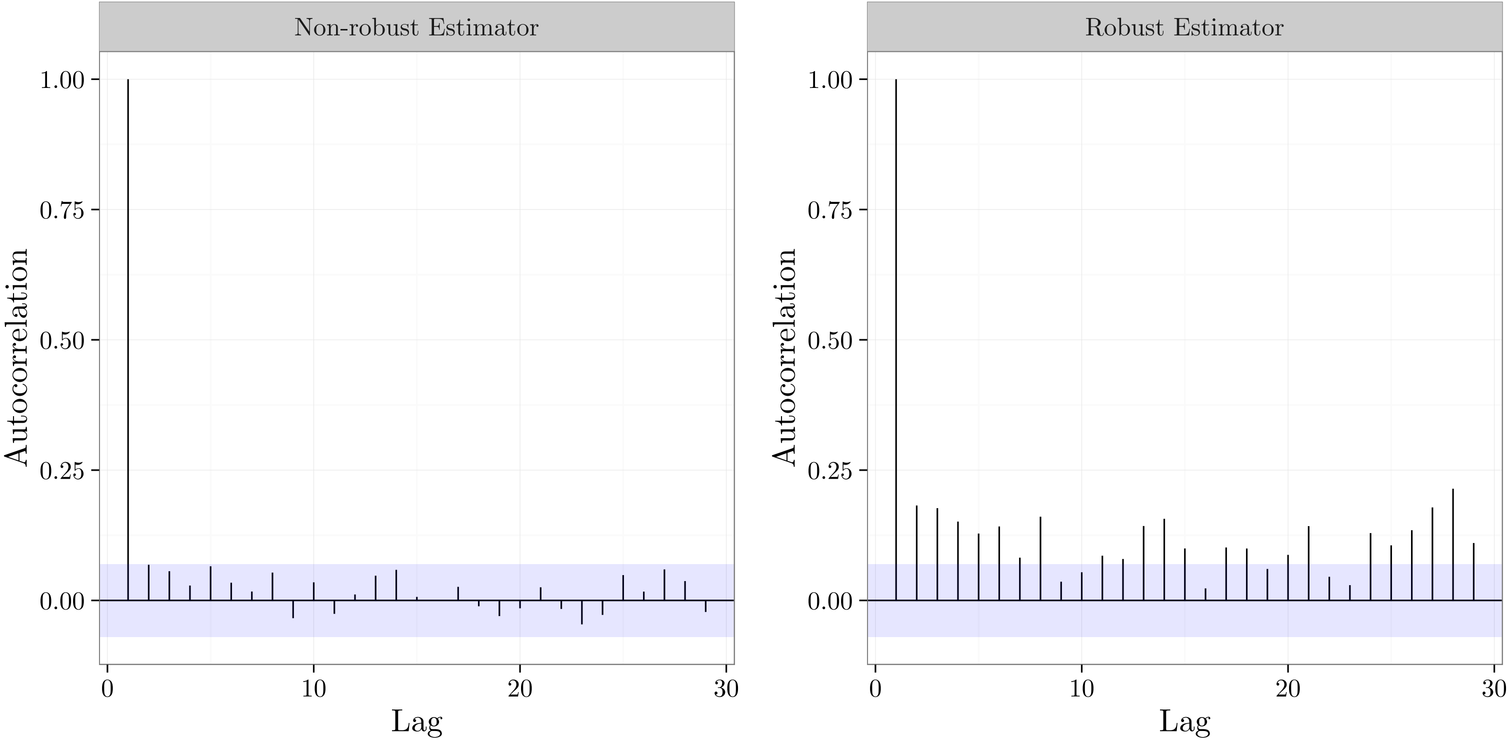

(X_t)
(Y_t)
Autocorrelation
Lag
1.00
0.75
0.50
0.25
0.00
0
10
20
30



(X_t)
(Y_t)
Autocorrelation
1.00
0.75
0.50
0.25
0.00
0
10
20
30
Lag


lag: h = 1
lag: h = 2
lag: h = 5
lag: h = 10
Sample autocorrelation
Standard
Robust
0.84
0.86
0.88
0.90
0.92
0.70
0.75
0.80
0.85
0.40
0.50
0.60
0.70
0.1
0.2
0.3
0.4
0.5


lag: h = 1
lag: h = 2
lag: h = 5
lag: h = 10
Sample autocorrelation
Standard
Robust
0.5
0.6
0.7
0.8
0.9
0.1
0.2
0.3
0.4


Non-robust Estimator
Robust Estimator
Autocorrelation
1.00
0.75
0.50
0.25
0.00
0
10
20
30
Lag