MID-TERM EXAM-1

Yi Xiong

2021/3/8

Contents

1.(20pts) Following the developments in Module 2.2, prove that the standard error in the estimation of the sample mean is $O(\frac{1}{\sqrt{n}})$ and that of the sample variance is $O(\frac{1}{\sqrt{n}})$

Let $x \sim N(\mu, \sigma^2)$

Assume μ is not known, σ^2 is known
Estimator $\hat{x} = \phi(x_1, x_2, ..., x_n)$

$$
\begin{aligned}
Var(\hat{x}(n)) &= E(\hat{x}(n) - \mu)^2 \\
&= E[(\frac{1}{n}\sum_{i=1}^{n} x_i) - \mu]^2 \\
&= E[\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)]^2 \\
&= \frac{1}{n^2}\sum_{i=1}^{n} E(x_i - \mu)^2 \\
&= \frac{\sigma^2}{n}
\end{aligned}
$$

$$
SE\ for\ \hat{x}(n) = \sqrt{\frac{\sigma^2}{n}} = \frac{\sigma}{\sqrt{n}} = O(\frac{1}{\sqrt{n}})
$$

Assume μ is known, σ^2 is not known

$$
Var(\hat{\sigma}^2) = E[\hat{\sigma}^2 - \sigma^2]^2
$$

Let $y_i = x_i - \mu$, $y_i \sim N(\mu, \sigma^2)$, $E(y_i) = 0$, $E(y_i^2) = \sigma^2$

$$
\begin{aligned}
Var(\hat{\sigma}^2) &= E[\frac{1}{n}\sum_{i=1}^{n}(y_i^2 - \sigma^2)]^2 \\
&= \frac{1}{n^2}\{\sum_{i=1}^{n}(y_i - \sigma^2)^2 + 2\sum_{i<j}(y_i^2 - \sigma^2)(y_j^2 - \sigma^2)\} \\
&\because y_i\ are\ IID \\
&= \frac{1}{n^2}\sum_{i=1}^{n} E(y_i^2 - \sigma^2)^2 \\
&= \frac{1}{n^2}\sum_{i=1}^{n}[E(y_i^4) - 2E(y_i^2)\sigma^2 + \sigma^4] \\
&= \frac{1}{n^2}\sum_{i=1}^{n}[3\sigma^4 - 2\sigma^4 + \sigma^4] \\
&= \frac{1}{n^2}\sum_{i=1}^{n} 2\sigma^4 \\
&= \frac{2\sigma^4}{n}
\end{aligned}
$$

$$
SE\ for\ Var(\hat{\sigma}^2) = \sqrt{\frac{2\sigma^4}{n}} = \frac{\sqrt{2\sigma^4}}{\sqrt{n}} = O(\frac{1}{\sqrt{n}})
$$

2.(20pts) Download two time series from the website that exhibit seasonality and trend.

http://www.statsci.org/datasets.html

Zurich Monthly Sunspot Numbers 1749 - 1983 (seasonality)

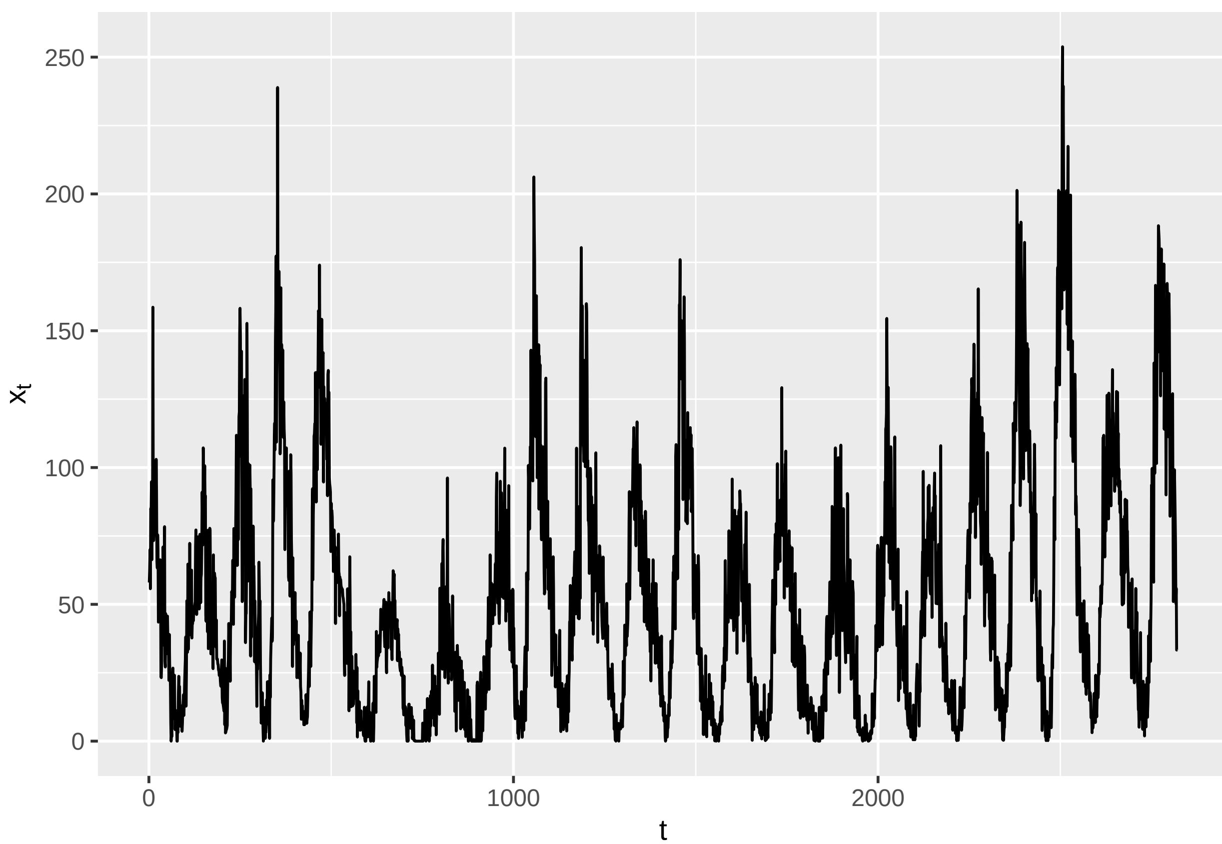
T11.1

Mauna Loa Carbon Dioxide (trend)

MLCO2.DAT

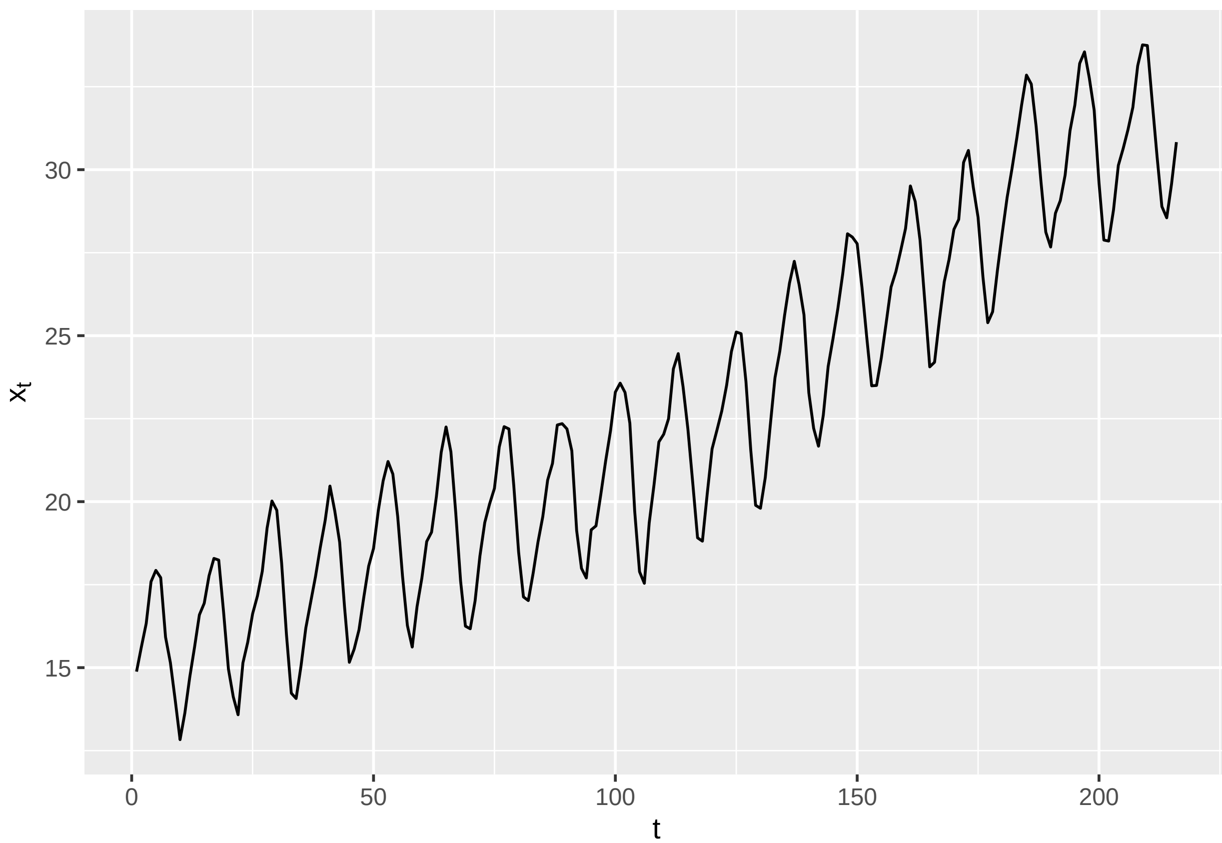
a. (5pts) Plot the original series, x_t

```
# Zurich Monthly Sunspot

library(ggplot2)
df<-read.table("T11.1")
x<-c(t(df[,5:16]))
t<-1:length(x)
df<-data.frame(x=t,y=x)
p<-ggplot(df,aes(x=x,y=y))+
  geom_line()+
  xlab("t")+
  ylab(expression(x[t]))
p
```




```
# MLCO2.DAT
```

```r
co<-scan("MLCO2.DAT",skip =2)
t2<-1:length(co)
df<-data.frame(x=t2,y=co)
p<-ggplot(df,aes(x=x,y=y))+
  geom_line()+
  xlab("t")+
  ylab(expression(x[t]))
  # scale_x_continuous(breaks = scales::pretty_breaks(n = 100))
p
```



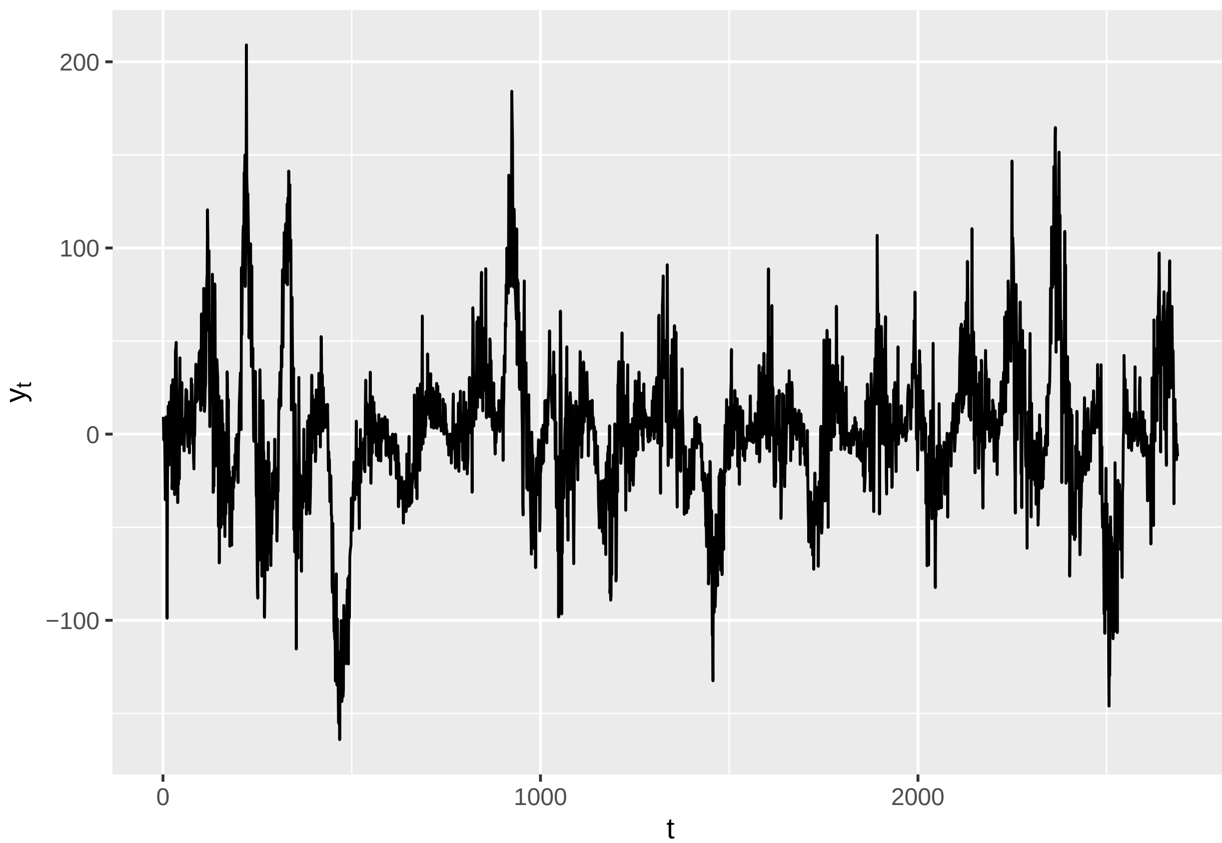
b. (5pts) Plot y_t after removing the seasonality from x_t

The solar cycle or solar magnetic activity cycle is a nearly periodic 11-year change in the Sun's activity measured in terms of variations in the number of observed sunspots on the solar surface.

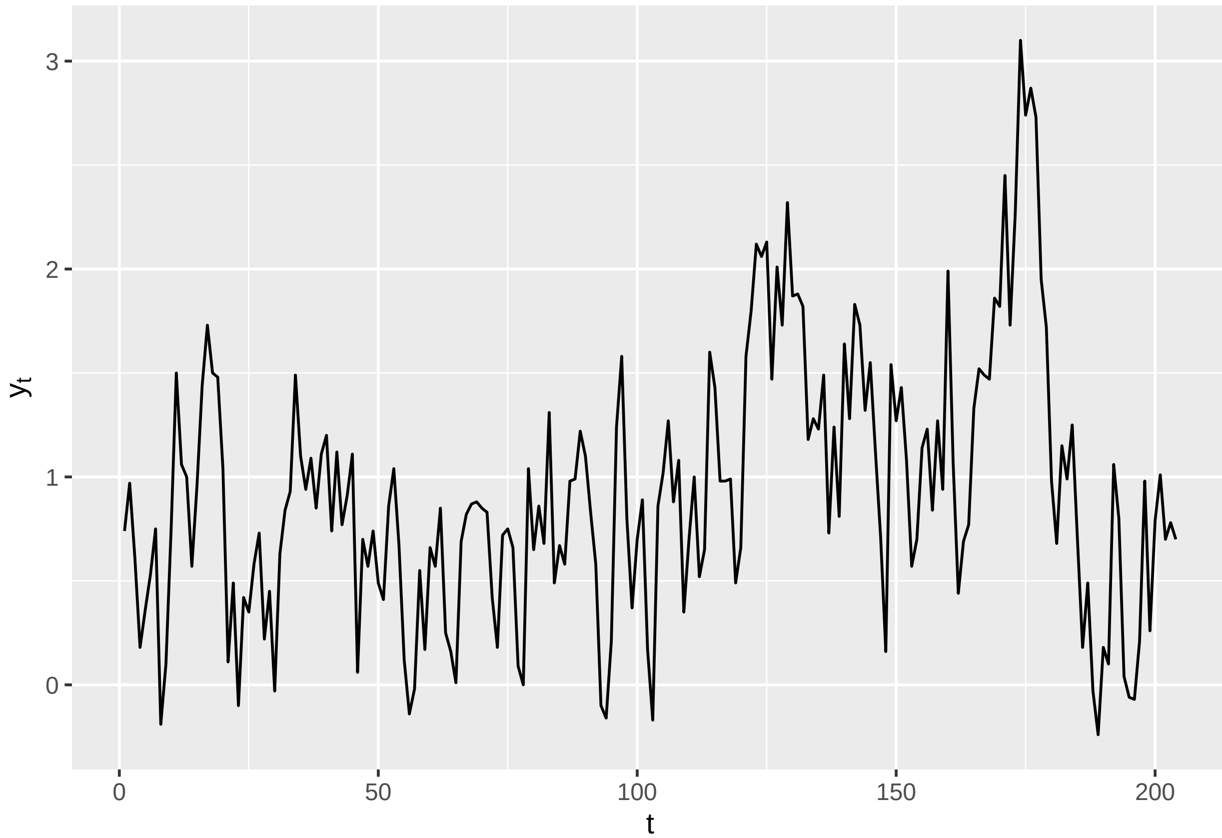
$\bigtriangledown x_t = x_t - x_{t-d}$ does not contain the seasonal part

```r
# to fit the cycle

# f = function(t, a, b, c, d) {
#   a * sin(2 * pi / b * t + c) + d
# }
# fit <-
#   nls(y ~ f(x, a, b, c, d),
#       data = df,
```

```r
#       start = list(
#         a = 75,
#         b = 132,
#         c = 70,
#         d = 75
#       ))
# fitResult <- summary(fit)
# a <- fitResult[["coefficients"]]["a", "Estimate"]
# b <- fitResult[["coefficients"]]["b", "Estimate"]
# c <- fitResult[["coefficients"]]["c", "Estimate"]
# d <- fitResult[["coefficients"]]["d", "Estimate"]
# f = function(t) {
#   a * sin(2 * pi / b * t + c) + d
# }
# p + stat_function(fun = f,
#                   color = "darkred",
#                   size = 1)

# remove the seasonality of sunpot data

n <- length(x)
sunCycle<-11*12
x1<-x[1:(n - sunCycle)]
x2<-x[(1 + sunCycle):(n)]
y<-x2-x1
df<-data.frame(x=1:(n - sunCycle),y=y)
p<-ggplot(df,aes(x=x,y=y))+
  geom_line()+
  xlab("t")+
  ylab(expression(y[t]))
p
```




```r
# remove the seasonality of co2

n <- length(co)
coCycle<-12
x1<-co[1:(n - coCycle)]
x2<-co[(1 + coCycle):n]
y2<-x2-x1
df<-data.frame(x=1:(n - coCycle),y=y2)
p<-ggplot(df,aes(x=x,y=y))+
  geom_line()+
  xlab("t")+
  ylab(expression(y[t]))
p
```

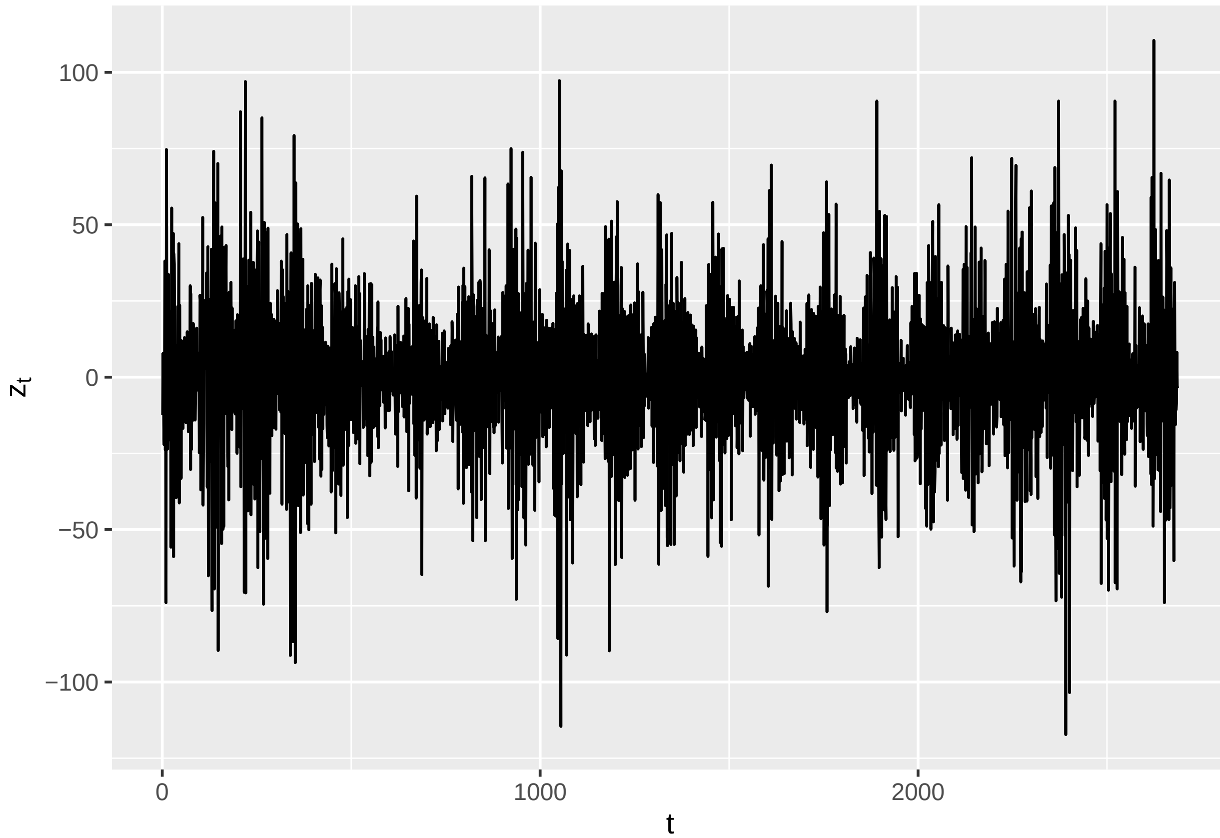


c. (5pts) Plot z_t after removing the trend in y_t

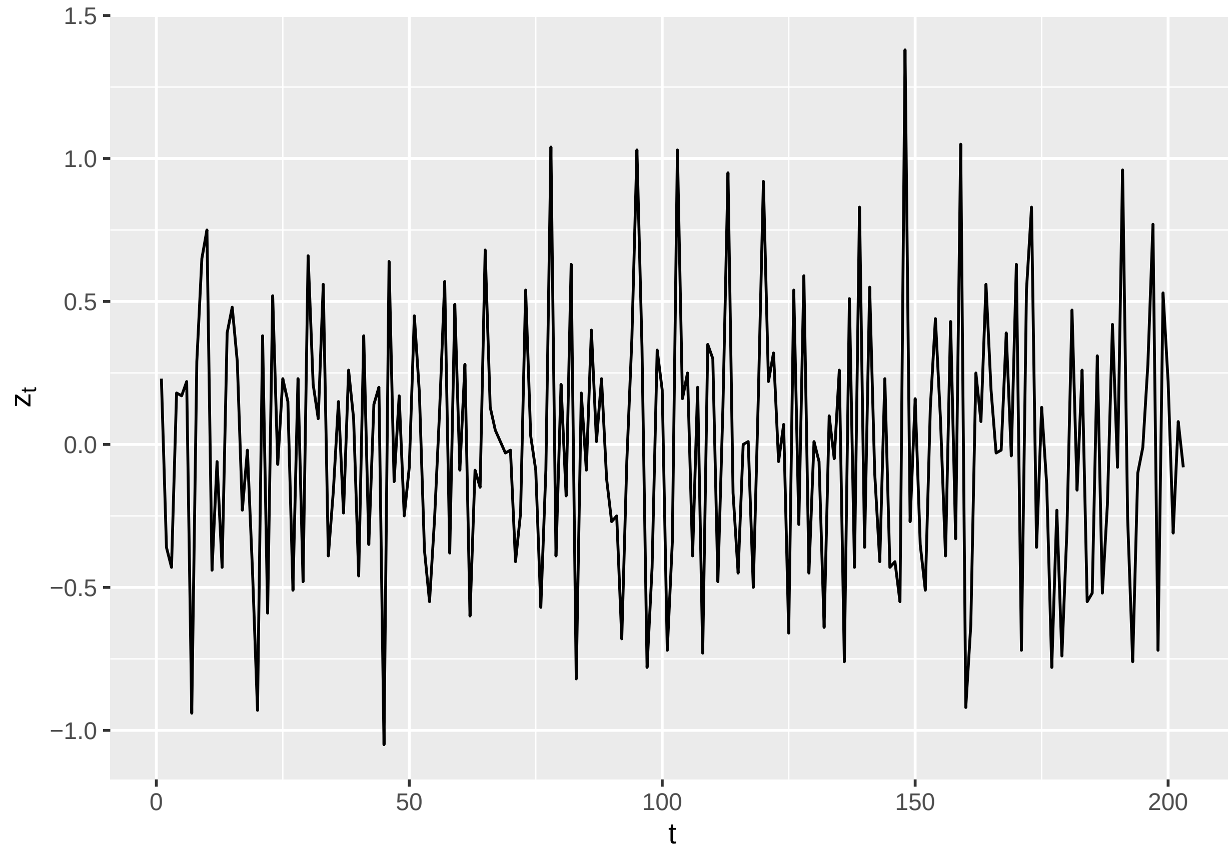
Remove linear trend

$\xi_t = \bigtriangledown \mu_t = \mu_t - \mu_{t-1}$

```r
# Remove linear trend of sunplot data

n <- length(y)
x1<-y[1:(n - 1)]
x2<-y[(1 + 1):(n)]
z<-x2-x1
df<-data.frame(x=1:(n - 1),y=z)
p<-ggplot(df,aes(x=x,y=y))+
  geom_line()+
  xlab("t")+
  ylab(expression(z[t]))
p
```




```r
# Remove linear trend of co2 data

n <- length(y2)
x1<-y2[1:(n - 1)]
x2<-y2[(1 + 1):(n)]
z2<-x2-x1
df<-data.frame(x=1:(n - 1),y=z2)
p<-ggplot(df,aes(x=x,y=y))+
  geom_line()+
  xlab("t")+
  ylab(expression(z[t]))
p
```



d. (5pts) Compute the ACF for z_t

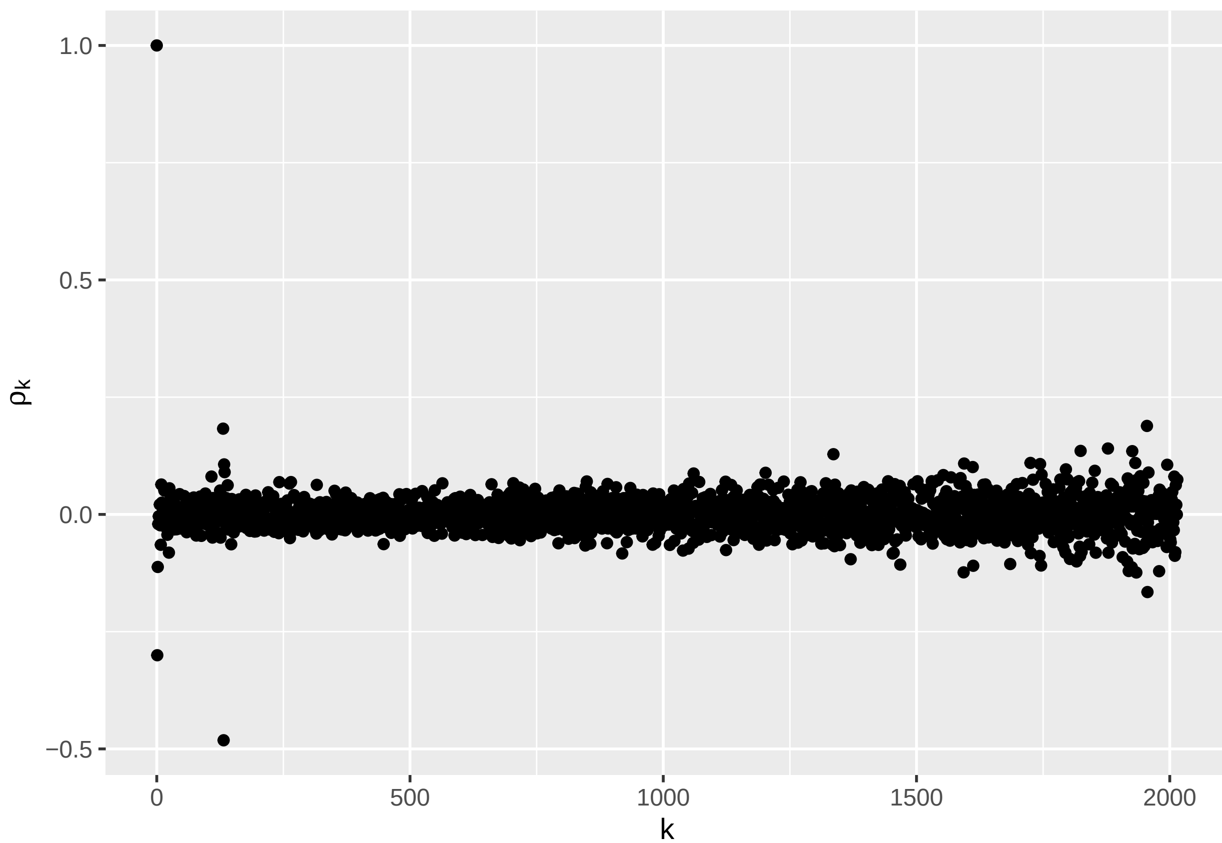
```
# autocorrelation of sunpot data
n <- length(z)
getCov <- function(k) {
  z1 <- z[1:(n - k)]
  z2 <- z[(1 + k):(n)]
  cov(z1, z2)
}
k <- 0:floor(3 / 4 * n)
cov1 <- sapply(k, getCov)
dfCov<-data.frame(k=k,AutoCove=cov1)
head(dfCov)
```

```
##   k    AutoCove
## 1 0  567.410176
## 2 1 -170.388465
## 3 2  -63.597974
## 4 3  -11.631735
## 5 4   -2.246955
## 6 5  -11.161271
```

```
p <- cov1 / cov1[1]
dfP <- data.frame(k = k, p = p)
head(dfP)
```

```
##   k            p
## 1 0  1.000000000
## 2 1 -0.300291521
## 3 2 -0.112084655
## 4 3 -0.020499694
## 5 4 -0.003960019
## 6 5 -0.019670552
```

```
dfP2 <- as.data.frame(spline(dfP$k, dfP$p, n = 200))
p2 <- ggplot(dfP, aes(x = k, y = p)) +
  geom_point()+
  # geom_line(data = dfP2, aes(x = x, y = y)) +
  xlab(expression(k)) +
  ylab(expression(rho[k]))
p2
```




```
# autocorrelation of co2 data

n <- length(z2)
getCov <- function(k) {
```

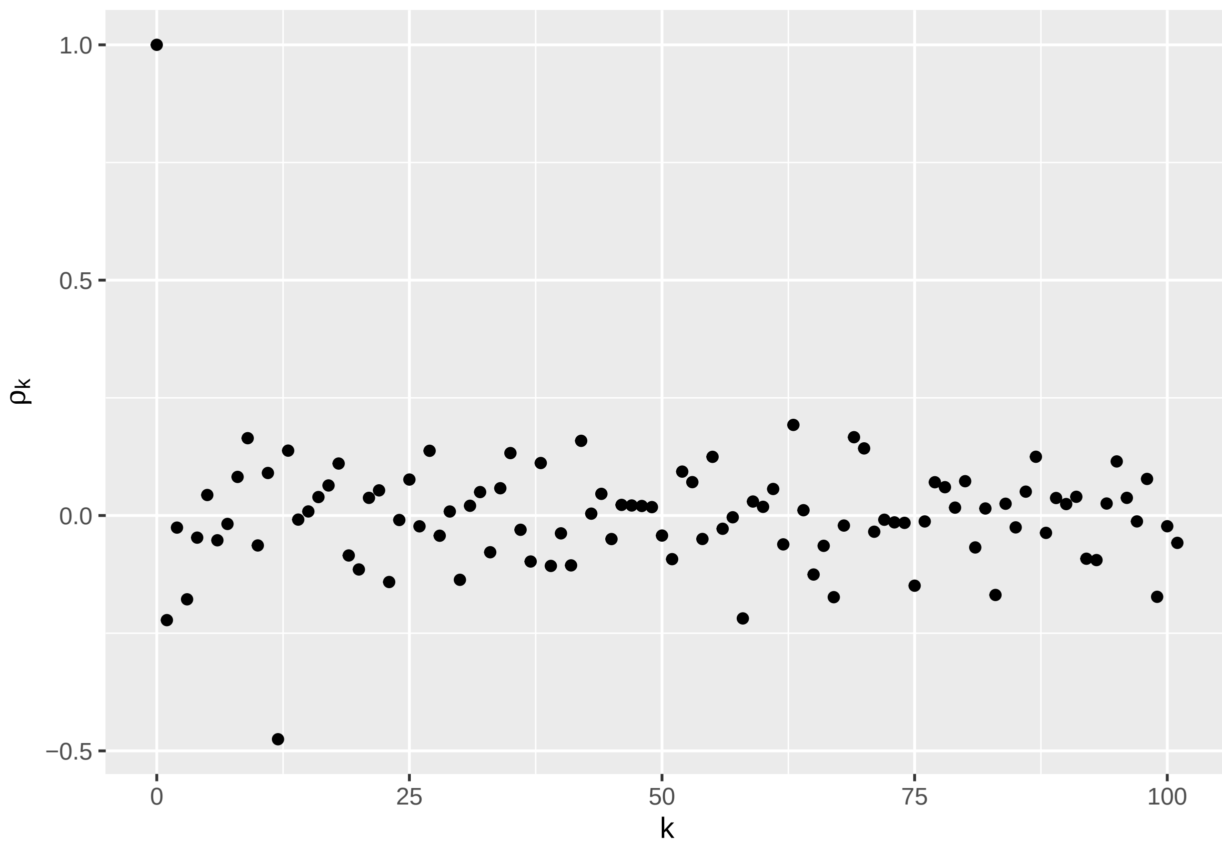
```
  x1 <- z2[1:(n - k)]
  x2 <- z2[(1 + k):(n)]
  cov(x1, x2)
}
k <- 0:floor(1 / 2 * n)
cov1 <- sapply(k, getCov)
dfCov<-data.frame(k=k,AutoCove=cov1)
head(dfCov)
```

```
##   k     AutoCove
## 1 0  0.208633624
## 2 1 -0.046367396
## 3 2 -0.005363910
## 4 3 -0.037142221
## 5 4 -0.009809017
## 6 5  0.009094478
```

```
p <- cov1 / cov1[1]
dfP <- data.frame(k = k, p = p)
head(dfP)
```

```
##   k           p
## 1 0  1.00000000
## 2 1 -0.22224316
## 3 2 -0.02570971
## 4 3 -0.17802606
## 5 4 -0.04701552
## 6 5  0.04359066
```

```
dfP2 <- as.data.frame(spline(dfP$k, dfP$p, n = 200))
p2 <- ggplot(dfP, aes(x = k, y = p)) +
  geom_point()+
  # geom_line(data = dfP2, aes(x = x, y = y)) +
  xlab(expression(k)) +
  ylab(expression(rho[k]))
p2
```



3. (20pts) Follow the developments in Modules in Part 4:

a. (5pts) Derive an expression for the ACF for MA (2) model

MA(q) definition

$$y_t = \mu + \sum_{j=0}^{q} \theta_j \epsilon_{t-j},\ q \geqslant 1\ and\ \theta_0 = 1$$

MA(2) definition

$$y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}$$

Autocovariance of MA(q)

$$\begin{aligned}\gamma_0 &= E(y_t - \mu)^2 \\ &= [\sum_{j=0}^{q} \theta_j^2]\sigma^2\end{aligned}$$

$$\gamma_j = E[(y_t - \mu)(y_{t-j} - \mu)]$$
$$\gamma_j = \begin{cases} (\theta_j + \theta_{j+1}\theta_1 + \theta_{j+2}\theta_2 + ... + \theta_q\theta_{q-j})\sigma^2\ for\ 1 \leqslant j \leqslant q \\ 0\ \ for\ j > q \end{cases}$$

Autocovariance of MA(2)

$$\begin{aligned}\gamma_0 &= (1+\theta_1^2+\theta_2^2)\sigma^2 \\ \gamma_1 &= (\theta_1+\theta_2\theta_1)\sigma^2 \\ \gamma_2 &= \theta_2\sigma^2 \\ \gamma_j &= 0,\ j \geqslant 3\end{aligned}$$

Autocorrelation of MA(2)

$$\begin{aligned}\rho_1 &= \frac{(\theta_1+\theta_2\theta_1)\sigma^2}{(1+\theta_1^2+\theta_2^2)\sigma^2} \\ \rho_2 &= \frac{\theta_2\sigma^2}{(1+\theta_1^2+\theta_2^2)\sigma^2} \\ \rho_j &= 0,\ j \geqslant 3\end{aligned}$$

b. (5pts) Derive an expression for the ACF for AR (2). Apply this to: $y_t = -0.6t_{t-1} + 0.3y_{t-2} + \varepsilon_t$

definition

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \epsilon_t$$

For stationarity:

$$\begin{aligned}&\lambda^2 - \phi_1\lambda - \phi_2 = 0 \\ &\lambda = \frac{\phi_1 \pm \sqrt{\phi_1^2 + 4\phi 2}}{2} \\ &|\lambda| < 1\end{aligned}$$

Autocovariance

$$y_t - \mu = \phi_1(y_{t-1}-\mu) + \phi_2(y_{t-2}-\mu) + \epsilon_t$$

$$\begin{aligned}\gamma_0 &= E[(y_t-\mu)(y_t-\mu)] \\ &= E[\phi_1(y_t-\mu)(y_{t-1}-\mu)] \\ &+ E[\phi_2(y_t-\mu)(y_{t-2}-\mu)] \\ &+ E[(y_t-\mu)\epsilon_t] \\ &= \phi_1\gamma_1 + \phi_2\gamma_2 + \sigma^2\end{aligned}$$

$$for\ j \geqslant 1$$

$$\begin{aligned}\gamma_j &= E[(y_t-\mu)(y_{t-j}-\mu)] \\ &= E[\phi_1(y_{t-1}-\mu)(y_{t-j}-\mu)] \\ &+ E[\phi_2(y_{t-2}-\mu)(y_{t-j}-\mu)] \\ &+ E[(y_{t-j}-\mu)\epsilon_t] \\ &= \phi_1\gamma_1 + \phi_2\gamma_2\end{aligned}$$

Autocorrelation

$$
for\ j \geqslant 1
$$

$$
\begin{aligned}
\rho_j &= \frac{\gamma_j}{\gamma_0} \\
&= \theta_1 \frac{\gamma_{j-1}}{\gamma_0} + \theta_2 \frac{\gamma_{j-2}}{\gamma_0} \\
&= \phi_1 \rho_{j-1} + \phi_2 \rho_{j-2}
\end{aligned}
$$

$$
\begin{aligned}
\rho_1 &= \phi_1 \rho_0 + \phi_2 \rho_{-1} \\
\rho_1 &= \frac{\phi_1}{1-\phi_2} \\
\rho_2 &= \phi_1 \rho_1 + \phi_2 \rho_0 \\
&= \phi_1 \rho_1 + \phi_2
\end{aligned}
$$

$$
\gamma_0 = \frac{(1-\phi_2)\sigma^2}{(1+\phi_2)[(1-\phi_2)^2 - \phi_1^2]}
$$

Summary

$$
\begin{aligned}
\rho_1 &= \frac{\phi_1}{1-\phi_2} \\
\rho_2 &= \phi_1 \rho_1 + \phi_2 \\
\rho_j &= \phi_1 \rho_{j-1} + \phi_2 \rho_{j-2} \quad for\ j \geqslant 3
\end{aligned}
$$

for $y_t = -0.6t_{t-1} + 0.3y_{t-2} + \varepsilon_t$

ρ_1

```
q1 <- -0.6
q2 <- 0.3
p1<-q1/(1-q2)
p1
```

```
## [1] -0.8571429
```

```
p2<-q1*p1+q2
p2
```

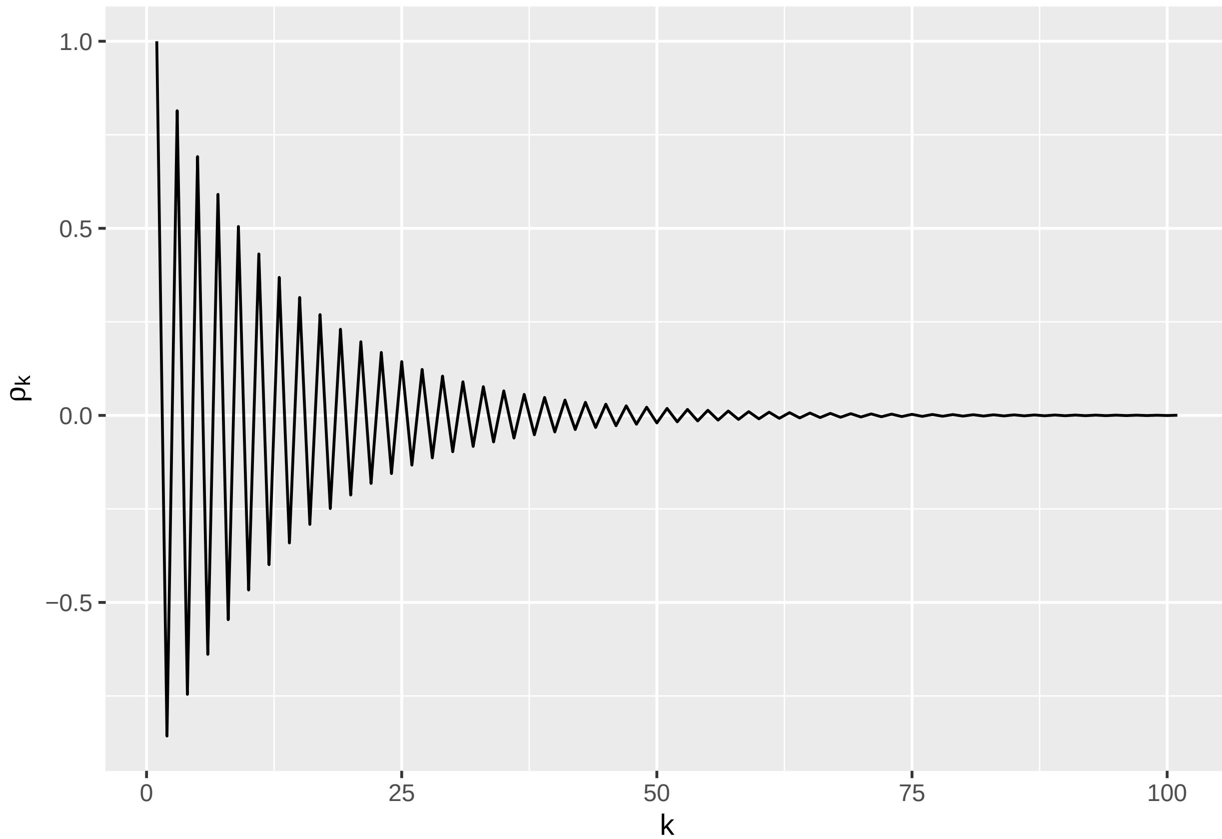
```
## [1] 0.8142857
```

```
pj<-c(p1,p2)
for (j in 3:100){
  pj<-c(pj,q1*pj[j-1]+q2*pj[j-2])
}
pj<-c(1,pj)
head(pj)
```

```
## [1]  1.0000000 -0.8571429  0.8142857 -0.7457143  0.6917143 -0.6387429
```

c. (5pts) Plot the ACF

```
df<-data.frame(x = 1:length(pj), y = pj)
p <- ggplot(df, aes(x = x, y = y)) +
  geom_line() +
  xlab(expression(k)) +
  ylab(expression(rho[k]))
p
```



d. (5pts) Derive an expression for the ACF of ARMA (1,2)

definition

$$y_t = c + \phi_1 y_{t-1} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}$$

Autocovariance

$$y_t - \mu = \phi_1(y_{t-1} - \mu) + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2}$$

$$\begin{aligned}
\gamma_0 &= E[y_t - \mu]^2 \\
&= E[\phi_1(y_{t-1} - \mu) + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2}]^2 \\
&= \phi_1^2\gamma_0 + \sigma^2 + \theta_1^2\sigma^2 + \theta_2^2\sigma^2 + 2\phi_1\theta_1\sigma^2 + 2\phi_1\theta_2(\phi_1 + \theta_1)\sigma^2
\end{aligned}$$

$$\gamma_0 = \frac{\sigma^2[1 + 2\phi_1\theta_1 + 2\phi_1\theta_2(\phi_1 + \theta_1) + \theta_1^2 + \theta_2^2]}{1 - \phi_1^2}$$

$$\begin{aligned}
\gamma_1 &= E[(y_t - \mu)(y_{t-1} - \mu)] \\
&= E\{[\phi_1(y_{t-1} - \mu) + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2}][y_{t-1} - \mu]\} \\
&= \phi_1\gamma_0 + \theta_1\sigma^2 + \theta_2(\phi_1 + \theta_1)\sigma^2
\end{aligned}$$

$$\begin{aligned}
\gamma_2 &= E[(y_t - \mu)(y_{t-2} - \mu)] \\
&= E\{[\phi_1(y_{t-1} - \mu) + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2}][y_{t-2} - \mu]\} \\
&= \phi_1\gamma_1 + \theta_2\sigma^2
\end{aligned}$$

$$for\ j \geqslant 3$$

$$\begin{aligned}
\gamma_j &= E[(y_t - \mu)(y_{t-j} - \mu)] \\
&= E\{[\phi_1(y_{t-1} - \mu) + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2}][y_{t-j} - \mu]\} \\
&= \phi_1 E[(y_{t-1} - \mu)(y_{t-j} - \mu)] \\
&= \phi_1\gamma_{j-1}
\end{aligned}$$

Autocorrelation

$$\begin{aligned}
\gamma_0 &= \frac{\sigma^2[1 + 2\phi_1\theta_1 + 2\phi_1\theta_2(\phi_1 + \theta_1) + \theta_1^2 + \theta_2^2]}{1 - \phi_1^2} \\
\gamma_1 &== \phi_1\gamma_0 + \theta_1\sigma^2 + \theta_2(\phi_1 + \theta_1)\sigma^2 \\
\gamma_2 &== \phi_1\gamma_1 + \theta_2\sigma^2 \\
\gamma_j &= \phi_1\gamma_{j-1}\ \ for\ j \geqslant 3 \\
\rho_1 &= \frac{\gamma_1}{\gamma_0} \\
\rho_2 &= \frac{\gamma_2}{\gamma_0} \\
\rho_j &= \frac{\gamma_j}{\gamma_0}
\end{aligned}$$