2024 秋季本科时间序列

第 4 次作业答案

10 月 27 日

1. (a) 计算交叉二阶矩矩阵：

$$M = \mathbb{E}\left[Z_t Z_t^{\intercal}\right] = \mathbb{E}\begin{bmatrix} 1 \\ X_{t-1} \\ X_{t-2} \end{bmatrix}\begin{bmatrix} 1 & X_{t-1} & X_{t-2} \end{bmatrix} = \begin{bmatrix} 1 & \mathbb{E}[X_{t-1}] & \mathbb{E}[X_{t-2}] \\ \mathbb{E}[X_{t-1}] & \mathbb{E}[X_{t-1}^2] & \mathbb{E}[X_{t-1}X_{t-2}] \\ \mathbb{E}[X_{t-2}] & \mathbb{E}[X_{t-1}X_{t-2}] & \mathbb{E}[X_{t-2}^2] \end{bmatrix}$$

由于 X_t 是平稳 AR(2) 过程，其均值为：

$$\mu_X = \mathbb{E}[X_t] = \frac{\mu}{1-\phi-\psi}$$

因此，

$$\mathbb{E}[X_{t-1}] = \mathbb{E}[X_{t-2}] = \mu_X$$

方差和协方差为：

$$\mathrm{Var}(X_t) = \gamma(0)$$

$$\mathrm{Cov}(X_t, X_{t-1}) = \gamma(1)$$

因此，

$$\mathbb{E}[X_{t-1}^2] = \mathrm{Var}(X_{t-1}) + (\mathbb{E}[X_{t-1}])^2 = \gamma(0) + \mu_X^2$$

$$\mathbb{E}[X_{t-1}X_{t-2}] = \mathrm{Cov}(X_{t-1}, X_{t-2}) + \mathbb{E}[X_{t-1}]\mathbb{E}[X_{t-2}] = \gamma(1) + \mu_X^2$$

因此，矩阵 M 可表示为：

$$M = \begin{bmatrix} 1 & \mu_X & \mu_X \\ \mu_X & \gamma(0) + \mu_X^2 & \gamma(1) + \mu_X^2 \\ \mu_X & \gamma(1) + \mu_X^2 & \gamma(0) + \mu_X^2 \end{bmatrix}$$

计算 M 的行列式：

$$\det(M) = \begin{vmatrix} 1 & \mu_X & \mu_X \\ \mu_X & \gamma(0) + \mu_X^2 & \gamma(1) + \mu_X^2 \\ \mu_X & \gamma(1) + \mu_X^2 & \gamma(0) + \mu_X^2 \end{vmatrix}$$

通过初等变换，从第二、三行中分别减去 μ_X 倍的第一行，得到：

$$M' = \begin{bmatrix} 1 & \mu_X & \mu_X \\ 0 & \gamma(0) & \gamma(1) \\ 0 & \gamma(1) & \gamma(0) \end{bmatrix}$$

因此，行列式为：

$$\det(M) = 1 \times \begin{vmatrix} \gamma(0) & \gamma(1) \\ \gamma(1) & \gamma(0) \end{vmatrix} = \gamma(0)^2 - \gamma(1)^2$$

由于平稳过程的性质，$\gamma(0) > |\gamma(1)|$，且 $\gamma(0) > 0$，因此：

$$\det(M) = \gamma(0)^2 - \gamma(1)^2 > 0$$

因此，矩阵 M 的行列式大于零，说明 M 是正定的，故 M 满秩。

(b) **生成模拟序列并计算谱密度：**

对于每组参数，生成长度为 $T = 1000$ 的 AR(2) 过程 $X_t^{(i)}$，模型为：

$$X_t^{(i)} = \mu + \phi^{(i)} X_{t-1}^{(i)} + \psi^{(i)} X_{t-2}^{(i)} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, 1)$$

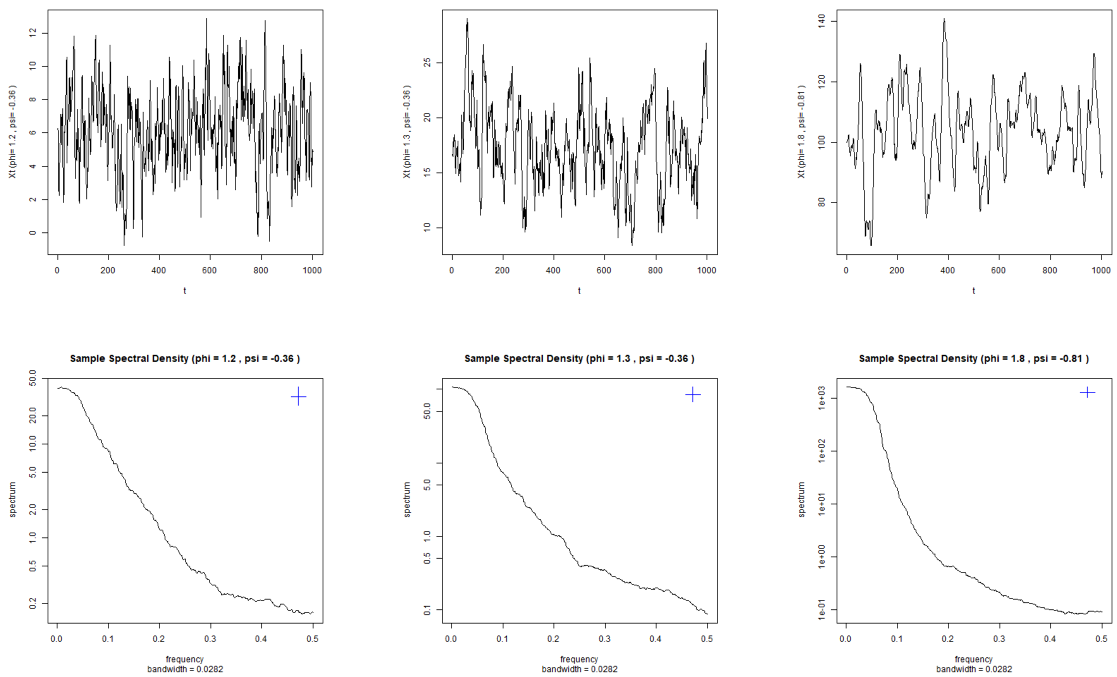
起始值取为 $X_{-1}^{(i)} = X_0^{(i)} = \mathbb{E}[X_t^{(i)}]$。

由于 AR(2) 过程的均值为：

$$\mathbb{E}[X_t^{(i)}] = \frac{\mu}{1 - \phi^{(i)} - \psi^{(i)}}$$

代码如下：

```
mu <- 1
phi <- c(1.2 ,1.3 ,1.8)
psi <- c(-0.36,-0.36,-0.81)
X <- tibble ()
for(i in 1:3){
X[1, i] = mu/(1 - phi[i]-psi[i])
X[2, i] = mu/(1 - phi[i]-psi[i])
for(j in 1:1000){
X[j+2, i] = mu + phi[i]*X[j+1, i] + psi[i]*X[j,i] +
rnorm(1 ,0 ,1)}
plot(X[[i]], type = "l", xlab = "t",
ylab = paste('Xt(phi=', phi[i],',psi=',psi[i],')'))}
spec.pgram(X[[1]], span =100, taper = 0.2)
spec.pgram(X[[2]], span =100, taper = 0.2)
spec.pgram(X[[3]], span =100, taper = 0.2)
```



理论谱密度函数及绘制

对于给定的 AR(2) 过程，其理论谱密度函数为：

$$S_X(\omega) = \frac{1}{|1 - \phi e^{-i2\pi\omega} - \psi e^{-i4\pi\omega}|^2}$$

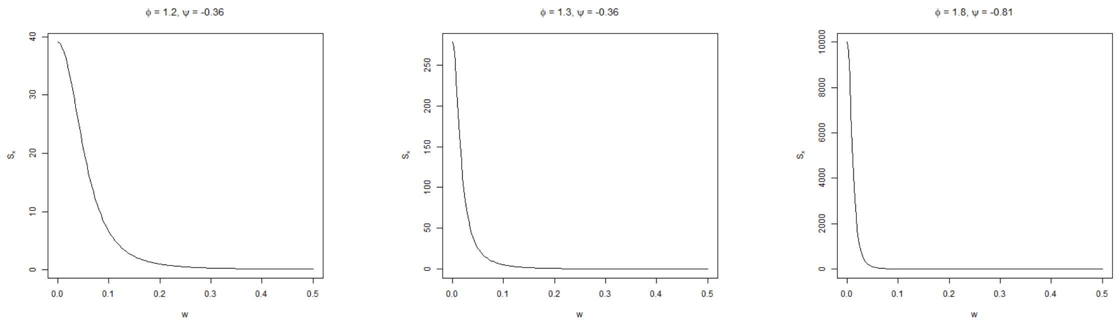
对于我们选定的参数 $(\phi^{(i)}, \psi^{(i)})$，我们可以得到以下谱密度函数：

$$S_X(\omega)\Big|_{(\phi^{(i)},\psi^{(i)})=(-1.2,0.36)} = \frac{1}{|1 + 1.2e^{-i2\pi\omega} - 0.36e^{-i4\pi\omega}|^2}$$
$$S_X(\omega)\Big|_{(\phi^{(i)},\psi^{(i)})=(-1.3,0.36)} = \frac{1}{|1 + 1.3e^{-i2\pi\omega} - 0.36e^{-i4\pi\omega}|^2}$$
$$S_X(\omega)\Big|_{(\phi^{(i)},\psi^{(i)})=(-1.8,0.81)} = \frac{1}{|1 + 1.8e^{-i2\pi\omega} - 0.81e^{-i4\pi\omega}|^2}$$

理论谱密度函数绘制代码如下：

```
x <- seq(0 ,0.5 ,0.0025)
s1 <- vector()
s2 <- vector()
s3 <- vector()
length(x)
for(i in 1:length(x)) {
s1[i] <- 1/(2.5696 -3.264*cos(2*pi*x[i])
+0.72*cos(4*pi*x[i]))
s2[i] <- 1/(2.8196 -3.536*cos(2*pi*x[i])
+0.72*cos(4*pi*x[i]))
s3[i] <- 1/(4.8961 -6.516*cos(2*pi*x[i])
+1.62*cos(4*pi*x[i]))
}
plot(x,s1,type="l",main="phi=1.2, psi=-0.36",
xlab="w",ylab="Sx")
plot(x,s2,type="l",main="phi=1.3,psi=-0.36",
```

```
xlab="w",ylab="Sx")
plot(x,s3,type="l",main="phi=1.8,psi=-0.81",
xlab="w",ylab="Sx")
```



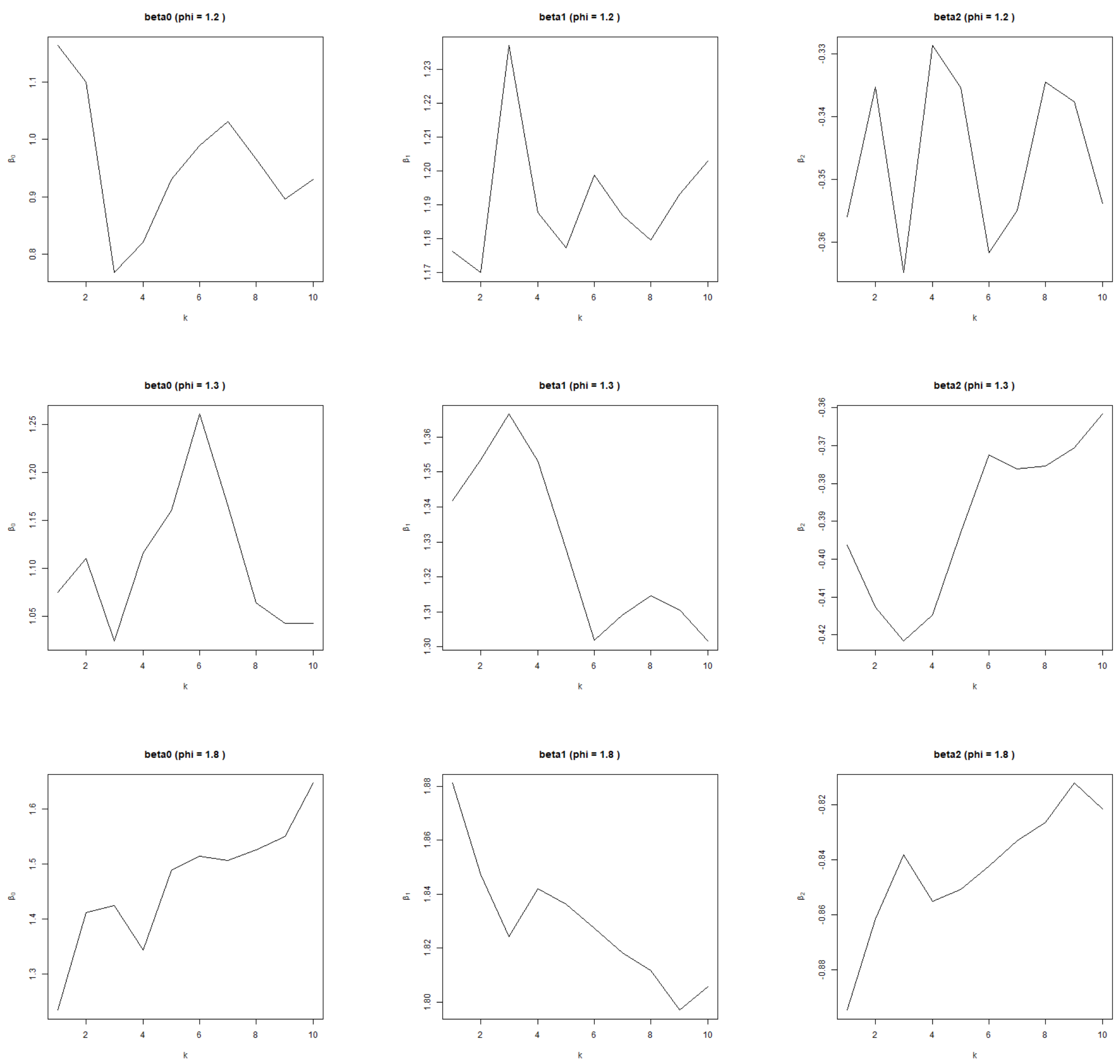
结果对比

通过对比样本谱密度和理论谱密度，发现尽管有限样本导致了样本谱密度的波动性，特别是在高频部分，但总体趋势与理论谱密度一致。随着样本量的增加，样本谱密度逐渐收敛于理论谱密度。

(c) 代码如下：

```
beta0 <- tibble ()
beta1 <- tibble ()
beta2 <- tibble ()
for(i in 1:3){
for(k in 1:10){
y <- unlist(X[3:(100*k+2), i])
x1 <- unlist(X[2:(100*k+1), i])
x2 <- unlist(X[1:(100*k), i])
ols <- coefficients(lm(y ~ x1+x2))
beta0[k, i] = ols[1]
beta1[k, i] = ols[2]
beta2[k, i] = ols[3]
```

```
}
plot(beta0[[i]], type = "l", xlab = "k",
ylab = paste('beta0(phi=', phi[i],')'))
plot(beta1[[i]], type = "l", xlab = "k",
ylab = paste('beta1(phi=', phi[i],')'))
plot(beta2[[i]], type = "l", xlab = "k",
ylab = paste('beta2(phi=', phi[i],')'))
}
```



随着 ϕ 的增大，OLS 估计随样本量在 $100k$ 增加时的收敛性逐渐减弱。

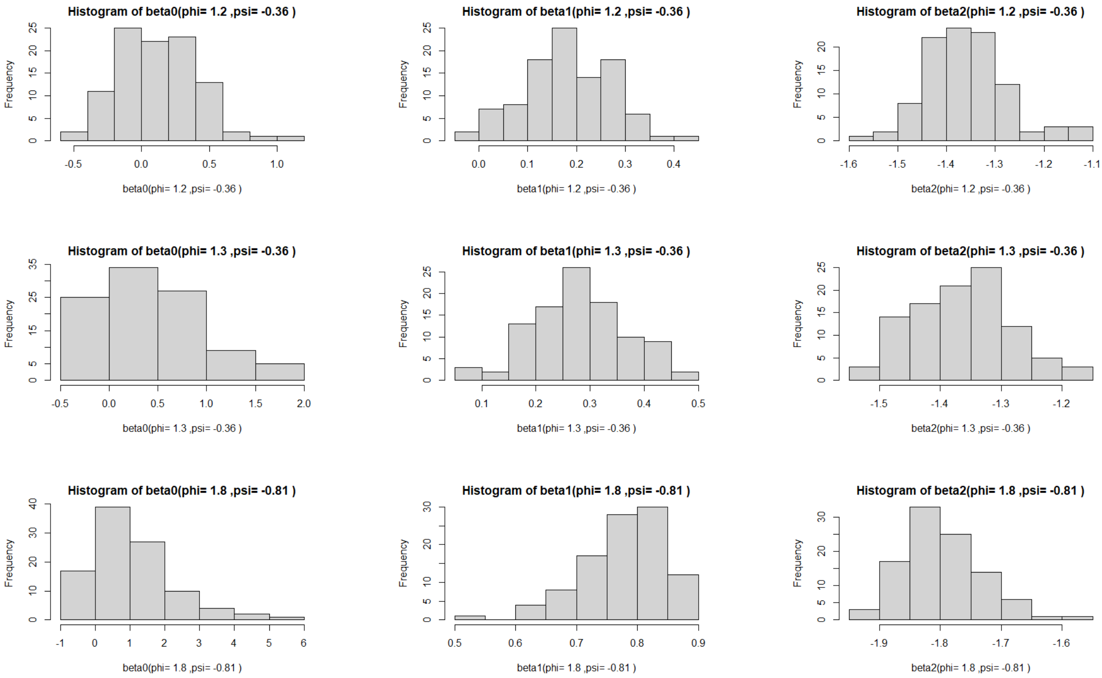
(d) 代码如下：

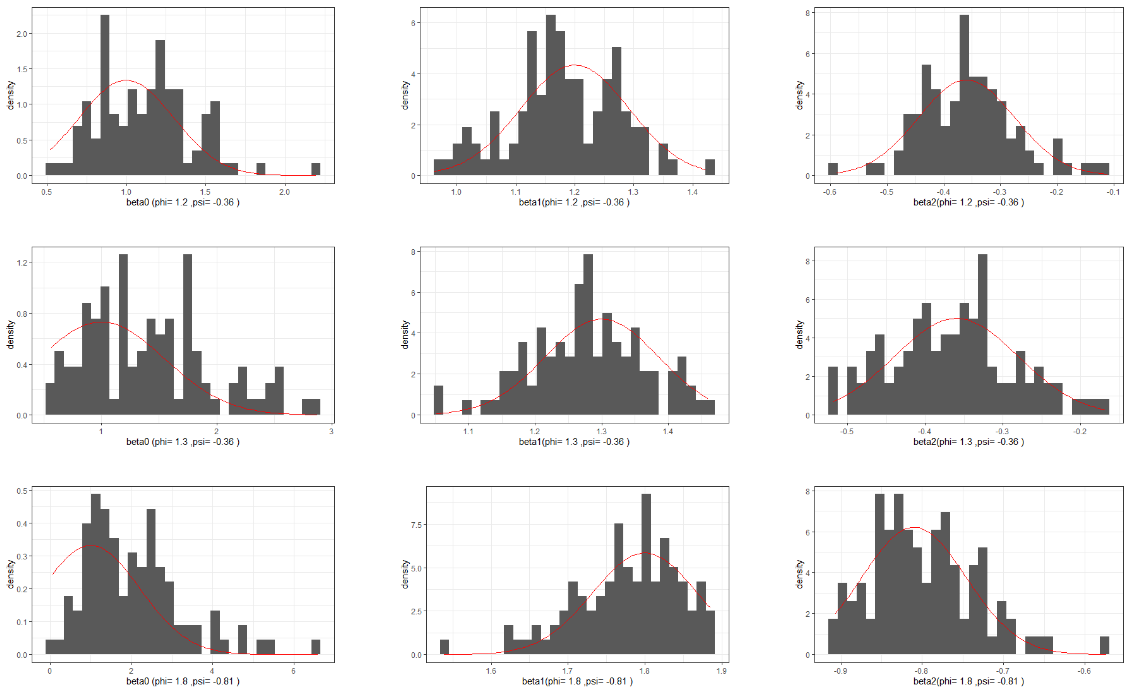
```r
#phi=1.2,psi=-0.36
X1 <- tibble ()
for(i in 1:1000){
X1[1, i] = mu/(1 - phi[1]-psi[1])
X1[2, i] = mu/(1 - phi[1]-psi[1])
for(j in 1:100){
X1[j+2, i] = mu + phi[1]*X1[j+1, i] + psi[1]*X1[j,i]
+ rnorm(1,0,1)}}
# phi = 1.3,psi=-0.36
X2 <- tibble ()
for(i in 1:1000){
X2[1, i] = mu/(1 - phi[2]-psi[2])
X2[2, i] = mu/(1 - phi[2]-psi[2])
for(j in 1:100){
X2[j+2, i] = mu + phi[2]*X2[j+1, i] + psi[2]*X2[j,i]
+ rnorm(1,0,1)}}
# phi = 1.8,psi =-0.81
X3 <- tibble ()
for(i in 1:1000){
X3[1, i] = mu/(1 - phi[3]-psi[3])
X3[2, i] = mu/(1 - phi[3]-psi[3])
for(j in 1:100){
X3[j+2, i] = mu + phi[3]*X3[j+1, i] + psi[3]*X3[j,i]
+ rnorm(1,0,1)}}
X <- list(X1, X2, X3)

```

```
beta0 <- tibble ()
beta1 <- tibble ()
beta2 <- tibble ()
b0 <- tibble ()
b1 <- tibble ()
b2 <- tibble ()
sd_beta0 <- vector("double" ,2)
sd_beta1 <- vector("double" ,2)
sd_beta2 <- vector("double" ,2)
for(i in 1:3){
for(j in 1:1000){
y <- unlist(X[[i]][3:102 , j])
x1 <- unlist(X[[i]][2:101 , j])
x2 <- unlist(X[[i]][1:100 , j])
ols <- coefficients(lm(y ~ x1+x2))
beta0[j, i] <- ols[1]
beta1[j, i] <- ols[2]
beta2[j, i] <- ols[3]
}
hist(beta0[[i]]-mu , xlab =
paste('beta0(phi=', phi[i],',psi=',psi[i],')'),
main = paste('Histogram_of_beta0(phi=', phi[i],',
psi=',psi[i],')'))
hist(beta1[[i]]-mu , xlab =
paste('beta1(phi=', phi[i],',psi=',psi[i],')'),
main = paste('Histogram_of_beta1(phi=', phi[i],',
```

```
psi=',psi[i],')'))
hist(beta2[[i]]-mu , xlab =
paste('beta2(phi=', phi[i],',psi=',psi[i],')'),
main = paste('Histogram_of_beta2(phi=', phi[i],',
psi=',psi[i],')'))

sd_beta0[i] <- sd(beta0[[i]])
sd_beta1[i] <- sd(beta1[[i]])
sd_beta2[i] <- sd(beta2[[i]])
}

sd_beta0
## [1] 0.2981030 0.5450414 1.2014578
sd_beta1
## [1] 0.09186399 0.08519157 0.06825279
sd_beta2
## [1] 0.08491475 0.07978170 0.06417489

for(i in 1:3){
print(ggplot()+ geom_histogram (data=NULL,
mapping=aes(x=beta0[[i]],y=..density..))
+ xlab(paste('beta0(phi=', phi[i],',psi=',psi[i],')'))
+ geom_function(fun = dnorm,color="red",
args=list(mean=mu,sd=sd_beta0[i])))
print(ggplot()+ geom_histogram (data=NULL,
mapping=aes(x=beta1[[i]],y=..density..))
```

```
+ xlab(paste('beta1(phi=', phi[i],',psi=',psi[i],')'))
+ geom_function(fun=dnorm ,color="red",
args=list(mean=phi[i],sd=sd_beta1[i])))
print(ggplot()+ geom_histogram (data=NULL,
mapping=aes(x=beta2[[i]],y=..density..))
+ xlab(paste('beta2(phi=', phi[i],',psi=',psi[i],')'))
+ geom_function(fun=dnorm,color="red",
args=list(mean=psi[i],sd=sd_beta2[i])))
}
```





当特征多项式零点趋近于 1 时，样本分布逐渐偏离正态分布。

(e) 代码如下：

```r
# 理论渐近标准误
mu <- 1
sigma <- 1
```

```
T <- 100
AR2_CM <- function(phi, psi, mu , sigma ){

mu_X <- mu / (1 - phi - psi)

A_matrix <- matrix(c(0, phi, psi,
                     phi, psi, 0,
                     phi, psi, 0),
                     nrow = 3, byrow = TRUE)
I_matrix <- diag(3)
b_vector <- c(1, 0, 0)

gamma_vec <- solve(I_matrix - A_matrix, b_vector)
gamma_0 <- gamma_vec[1]
gamma_1 <- gamma_vec[2]
gamma_2 <- gamma_vec[3]
E_X_t_1 <- mu_X
E_X_t_2 <- mu_X
E_X_t1_sq <- gamma_0 + mu_X^2
E_X_t1_X_t2 <- gamma_1 + mu_X^2
E_X_t2_sq <- gamma_0 + mu_X^2
M <- matrix(c(1, E_X_t_1, E_X_t_2,
                E_X_t_1, E_X_t1_sq, E_X_t1_X_t2,
                E_X_t_2, E_X_t1_X_t2, E_X_t2_sq),
              nrow = 3, byrow = TRUE)
CM <- solve(M)*sigma^2*(1/T)
```

```
return(CM)
}

AR2_CM(phi[1],psi[1],mu=1,sigma=1)
#               [,1]        [,2]        [,3]
#[1,]  0.11742187 -0.00859375 -0.00859375
#[2,] -0.00859375  0.01168750 -0.01031250
#[3,] -0.00859375 -0.01031250  0.01168750
AR2_CM(phi[2],psi[2],mu=1,sigma=1)
#                [,1]         [,2]         [,3]
#[1,]  0.298220551 -0.008646617 -0.008646617
#[2,] -0.008646617  0.011759398 -0.011240602
#[3,] -0.008646617 -0.011240602  0.011759398
AR2_CM(phi[3],psi[3],mu=1,sigma=1)
#                [,1]         [,2]         [,3]
#[1,]  1.561246537 -0.007756233 -0.007756233
#[2,] -0.007756233  0.014038781 -0.013961219
#[3,] -0.007756233 -0.013961219  0.014038781
sd_theory <- vector(mode = "list", length = 3)
for(i in 1:3){
sd_theory [[i]] <- c(sd_beta0 = sqrt(AR2_CM(phi[i],
psi[i],mu=1,sigma=1)[1,1]) ,
sd_beta1 =sqrt(AR2_CM(phi[i],psi[i],mu=1,sigma=1)[2,2]),
sd_beta2 =sqrt(AR2_CM(phi[i],psi[i],mu=1,sigma=1)[3,3]))
}
names(sd_theory) <- c("phi=1.2,psi=-0.36","phi=1.3,
```

```
psi=-0.36","phi=1.8,psi=-0.81")
sd_theory
# $phi =1.2,psi =-0.36
# sd_beta0   sd_beta1   sd_beta2
# 0.3426688  0.1081087  0.1081087
#
# $phi =1.3,psi =-0.36
# sd_beta0   sd_beta1   sd_beta2
# 0.5460957  0.1084408  0.1084408
#
# $phi =1.8,psi =-0.81
# sd_beta0   sd_beta1   sd_beta2
# 1.2494985  0.1184854  0.1184854

# 样本渐近标准误
x <- vector("double" ,101)
for (i in 1:3) {
x[1] <- mu/(1 - phi[i]-psi[i])
x[2] <- mu/(1 - phi[i]-psi[i])
for (j in 1:100) {
x[j+2] = mu + phi[3]*x[j+1] + psi[3]*x[j]
+ rnorm(1 ,0 ,1)}
X <- cbind(rep(1 ,100) , x[2:101], x[1:100])
M <- t(X) %*% X / 100
CM<- 1/100*1*solve(M)
print(CM)
```

```
sd_sample<-c(sd_beta0=sqrt(CM[1 ,1]) ,
sd_beta1=sqrt(CM [2 ,2]),
sd_beta2=sqrt(CM [3 ,3]))
print(sd_sample)
}

#                [,1]          [,2]          [,3]
#[1,]  0.12873521  -0.012615582   0.011248837
#[2,] -0.01261558   0.003611043  -0.003490062
#[3,]  0.01124884  -0.003490062   0.003385072
#sd_beta0    sd_beta1    sd_beta2
#0.35879689  0.06009196  0.05818137
#                 [,1]          [,2]          [,3]
#[1,]  0.099783739  -0.009623794   0.008496649
#[2,] -0.009623794   0.005880669  -0.005827523
#[3,]  0.008496649  -0.005827523   0.005789471
#sd_beta0    sd_beta1    sd_beta2
#0.31588564  0.07668552  0.07608857
#                 [,1]          [,2]          [,3]
#[1,]  0.352987307   0.003876617  -0.007794409
#[2,]  0.003876617   0.003977387  -0.004003679
#[3,] -0.007794409  -0.004003679   0.004074804
#sd_beta0    sd_beta1    sd_beta2
#0.59412735  0.06306653  0.06383420
```

渐进标准误接近样本标准差的值。

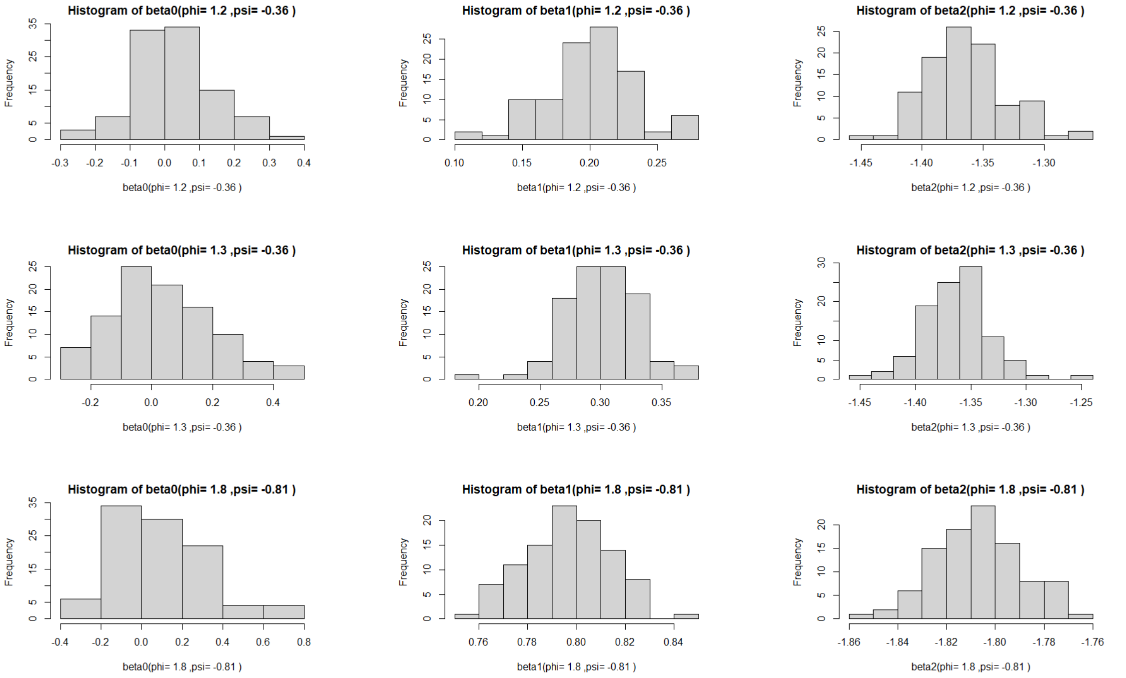
(f) 代码如下：

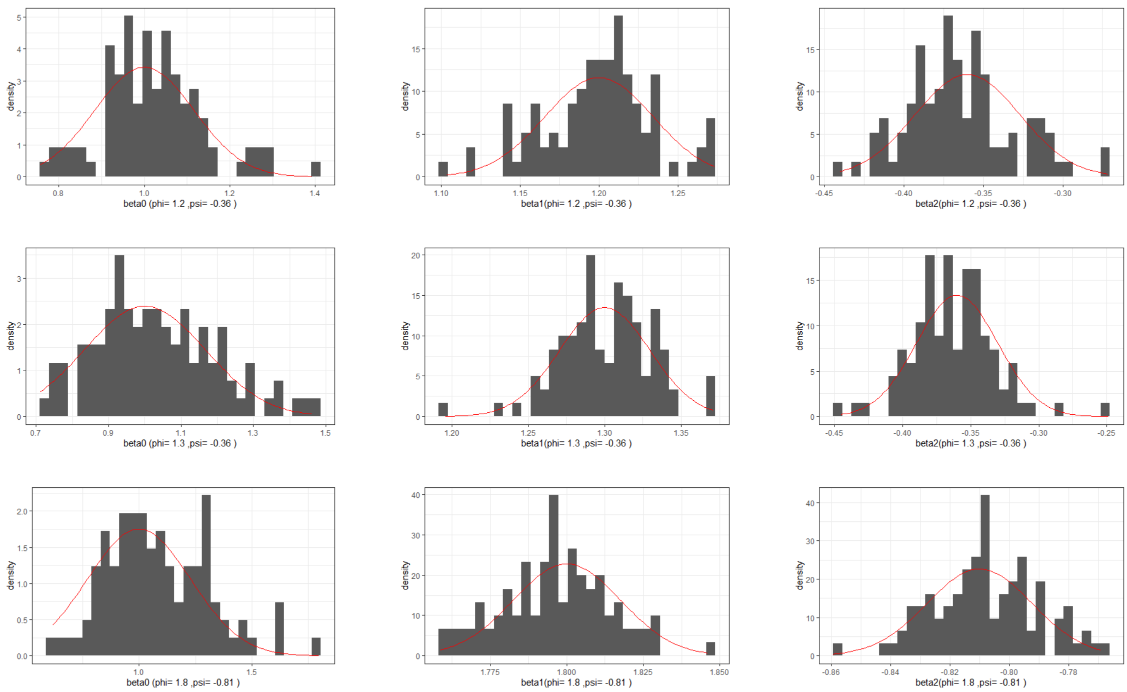
```
X1 <- tibble ()
for(i in 1:1000){
X1[1, i] = mu/(1 - phi[1]-psi[1])
X1[2, i] = mu/(1 - phi[1]-psi[1])
for(j in 1:900){
X1[j+2, i] = mu + phi[1]*X1[j+1, i] + psi[1]*X1[j,i]
+ rnorm(1 ,0 ,1)}}
# phi = 1.3,psi=-0.36
X2 <- tibble ()
for(i in 1:1000){
X2[1, i] = mu/(1 - phi[2]-psi[2])
X2[2, i] = mu/(1 - phi[2]-psi[2])
for(j in 1:900){
X2[j+2, i] = mu + phi[2]*X2[j+1, i] + psi[2]*X2[j,i]
+ rnorm(1 ,0 ,1)}}
# phi = 1.8,psi =-0.81
X3 <- tibble ()
for(i in 1:1000){
X3[1, i] = mu/(1 - phi[3]-psi[3])
X3[2, i] = mu/(1 - phi[3]-psi[3])
for(j in 1:900){
X3[j+2, i] = mu + phi[3]*X3[j+1, i] + psi[3]*X3[j,i]
+ rnorm(1 ,0 ,1)}}
X <- list(X1 , X2 , X3)
beta0 <- tibble ()
beta1 <- tibble ()
```

```
beta2 <- tibble ()
b0 <- tibble ()
b1 <- tibble ()
b2 <- tibble ()
sd_beta0 <- vector("double" ,2)
sd_beta1 <- vector("double" ,2)
sd_beta2 <- vector("double" ,2)
for(i in 1:3){
for(j in 1:100){
y <- unlist(X[[i]][3:902 , j])
x1 <- unlist(X[[i]][2:901 , j])
x2 <- unlist(X[[i]][1:900 , j])
ols <- coefficients(lm(y ~ x1+x2))
beta0[j, i] <- ols[1]
beta1[j, i] <- ols[2]
beta2[j, i] <- ols[3]
}
hist(beta0[[i]]-mu , xlab = paste('beta0(phi=',
phi[i],',psi=',psi[i],')'),
main = paste('Histogram_of_beta0(phi=', phi[i],',
psi=',psi[i],')'))
hist(beta1[[i]]-mu , xlab = paste('beta1(phi=',
phi[i],',psi=',psi[i],')'),
main = paste('Histogram_of_beta1(phi=', phi[i],',
psi=',psi[i],')'))
hist(beta2[[i]]-mu , xlab = paste('beta2(phi=',
```

```
phi[i],',psi=',psi[i],')'),
main = paste('Histogram_of_beta2(phi=', phi[i],',
psi=',psi[i],')'))
sd_beta0[i] <- sd(beta0[[i]])
sd_beta1[i] <- sd(beta1[[i]])
sd_beta2[i] <- sd(beta2[[i]])
}
sd_beta0
sd_beta1
sd_beta2
for(i in 1:3){
print(ggplot()+ geom_histogram (data=NULL ,
mapping=aes(x=beta0[[i]],y=..density..))
+ xlab(paste('beta0(phi=', phi[i],',psi=',psi[i],')'))
+ geom_function(fun = dnorm ,color="red",
args=list(mean=mu ,sd=sd_beta0[i])))
print(ggplot()+ geom_histogram (data=NULL ,
mapping=aes(x=beta1[[i]],y=..density..))
+ xlab(paste('beta1(phi=', phi[i],',psi=',psi[i],')'))
+ geom_function(fun=dnorm ,color="red",
args=list(mean=phi[i],sd=sd_beta1[i])))
print(ggplot()+ geom_histogram (data=NULL ,
mapping=aes(x=beta2[[i]],y=..density..))
+ xlab(paste('beta2(phi=', phi[i],',psi=',psi[i],')'))
+ geom_function(fun=dnorm ,color="red",
args=list(mean=psi[i],sd=sd_beta2[i])))
```

```
}
```






```
# 理论渐近标准误
mu <- 1
sigma <- 1
```

```
T <- 900
AR2_CM <- function(phi, psi, mu , sigma ){

mu_X <- mu / (1 - phi - psi)

A_matrix <- matrix(c(0, phi, psi,
                     phi, psi, 0,
                     phi, psi, 0), nrow = 3, byrow = TRUE)
I_matrix <- diag(3)
b_vector <- c(1, 0, 0)

gamma_vec <- solve(I_matrix - A_matrix, b_vector)
gamma_0 <- gamma_vec[1]
gamma_1 <- gamma_vec[2]
gamma_2 <- gamma_vec[3]
E_X_t_1 <- mu_X
E_X_t_2 <- mu_X
E_X_t1_sq <- gamma_0 + mu_X^2
E_X_t1_X_t2 <- gamma_1 + mu_X^2
E_X_t2_sq <- gamma_0 + mu_X^2
M <- matrix(c(1, E_X_t_1, E_X_t_2,
              E_X_t_1, E_X_t1_sq, E_X_t1_X_t2,
              E_X_t_2, E_X_t1_X_t2, E_X_t2_sq),
            nrow = 3, byrow = TRUE)
CM <- solve(M)*sigma^2*(1/T)
return(CM)
```

```
}

AR2_CM(phi[1],psi[1],mu=1,sigma=1)
#              [,1]          [,2]          [,3]
#[1,]  0.0130468750 -0.0009548611 -0.0009548611
#[2,] -0.0009548611  0.0012986111 -0.0011458333
#[3,] -0.0009548611 -0.0011458333  0.0012986111
AR2_CM(phi[2],psi[2],mu=1,sigma=1)
#              [,1]          [,2]          [,3]
#[1,]  0.0331356168 -0.0009607352 -0.0009607352
#[2,] -0.0009607352  0.0013065998 -0.0012489557
#[3,] -0.0009607352 -0.0012489557  0.0013065998
AR2_CM(phi[3],psi[3],mu=1,sigma=1)
#              [,1]          [,2]          [,3]
#[1,]  0.1734718375 -0.0008618036 -0.0008618036
#[2,] -0.0008618036  0.0015598646 -0.0015512465
#[3,] -0.0008618036 -0.0015512465  0.0015598646

sd_theory <- vector(mode = "list", length = 3)
for(i in 1:3){
sd_theory[[i]] <- c(sd_beta0 = sqrt(AR2_CM(phi[i],
psi[i],mu=1,sigma=1)[1,1]) ,
sd_beta1 =sqrt(AR2_CM(phi[i],psi[i],mu=1,sigma=1)[2,2]),
sd_beta2 =sqrt(AR2_CM(phi[i],psi[i],mu=1,sigma=1)[3,3]))
}
names(sd_theory) <- c("phi=1.2,psi=-0.36","phi=1.3,
```

```
psi=-0.36","phi=1.8,psi=-0.81")
sd_theory
# $phi =1.2,psi =-0.36
#    sd_beta0    sd_beta1    sd_beta2
# 0.11422292  0.03603625  0.03603625
#
# $phi =1.3,psi =-0.36
#    sd_beta0    sd_beta1    sd_beta2
# 0.18203191  0.03614692  0.03614692
#
# $phi =1.8,psi =-0.81
#    sd_beta0    sd_beta1    sd_beta2
# 0.41649950  0.03949512  0.03949512

# 样 本 渐 近 标 准 误
x <- vector("double" ,901)
for (i in 1:3) {
x[1] <- mu/(1 - phi[i]-psi[i])
x[2] <- mu/(1 - phi[i]-psi[i])
for (j in 1:900) {
x[j+2] = mu + phi [3]*x[j+1] + psi[3]*x[j] +
rnorm(1 ,0 ,1)}
X <- cbind(rep(1 ,900) , x[2:901], x[1:900])
M <- t(X) %*% X / 900
CM<- 1/900*1*solve(M)
print(CM)
```

```
sd_sample<-c(sd_beta0=sqrt(CM[1 ,1]) ,
sd_beta1=sqrt(CM [2 ,2]),
sd_beta2=sqrt(CM [3 ,3]))
print(sd_sample)
}
#              [,1]          [,2]          [,3]
#[1,]  0.0304557095 -0.0006450578  0.0003360185
#[2,] -0.0006450578  0.0004005666 -0.0003942706
#[3,]  0.0003360185 -0.0003942706  0.0003912299
#  sd_beta0   sd_beta1   sd_beta2
#0.17451564 0.02001416 0.01977953
#              [,1]          [,2]          [,3]
#[1,]  0.0432229056 -0.0007491774  0.0003430015
#[2,] -0.0007491774  0.0003474072 -0.0003405139
#[3,]  0.0003430015 -0.0003405139  0.0003375386
#  sd_beta0   sd_beta1   sd_beta2
#0.20790119 0.01863886 0.01837222
#              [,1]          [,2]          [,3]
#[1,]  0.0354082586 -0.0001391447 -0.0002124855
#[2,] -0.0001391447  0.0003852756 -0.0003839542
#[3,] -0.0002124855 -0.0003839542  0.0003862379
#  sd_beta0   sd_beta1   sd_beta2
#0.18817082 0.01962844 0.01965294
```

此时估计系数的样本标准差和渐进标准误均近似为先前的 1/3。