

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:

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

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:

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

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

因此,

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

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

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：

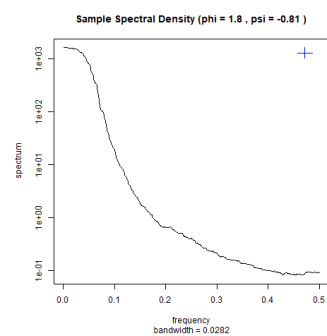
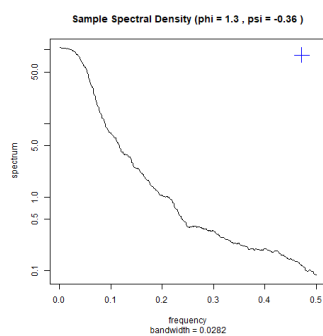
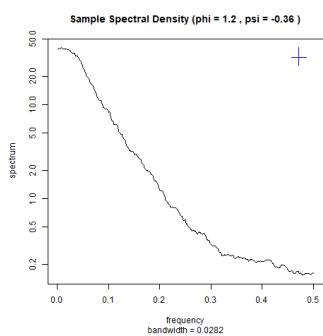
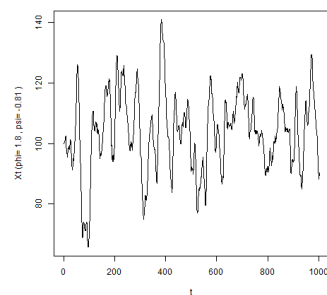
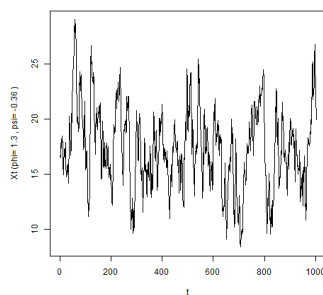
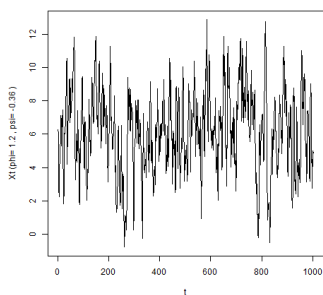
$$\mathbb{E}[X_t^{(i)}] = \frac{\mu}{1 - \phi^{(i)} - \psi^{(i)}}$$

代码如下：

```

1 mu <- 1
2 phi <- c(1.2 ,1.3 ,1.8)
3 psi <- c(-0.36,-0.36,-0.81)
4 X <- tibble ()
5 for(i in 1:3){
6 X[1, i] = mu/(1 - phi[i]-psi[i])
7 X[2, i] = mu/(1 - phi[i]-psi[i])
8 for(j in 1:1000){
9 X[j+2, i] = mu + phi[i]*X[j+1, i] + psi[i]*X[j,i] +
10 rnorm(1 ,0 ,1)}
11 plot(X[[i]], type = "l", xlab = "t",
12 ylab = paste('Xt(phi=', phi[i],',psi=',psi[i],')'))}
13 spec.pgram(X[[1]], span =100, taper = 0.2)
14 spec.pgram(X[[2]], span =100, taper = 0.2)
15 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：

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

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

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

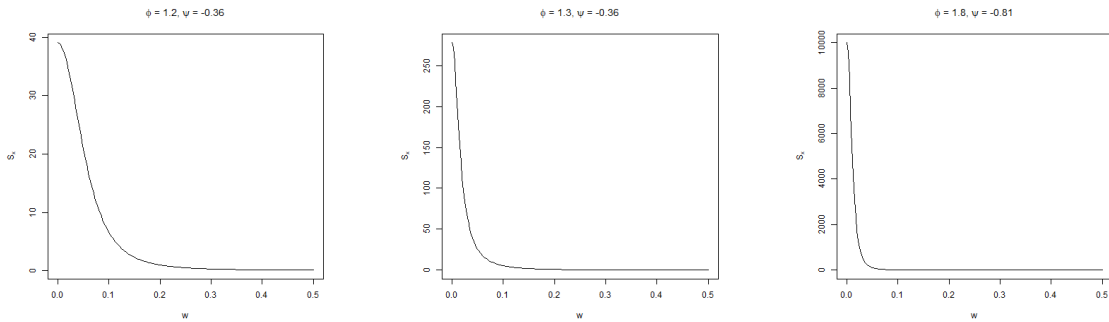
理论谱密度函数绘制代码如下：

```
1 x <- seq(0 ,0.5 ,0.0025)
2 s1 <- vector()
3 s2 <- vector()
4 s3 <- vector()
5 length(x)
6 for(i in 1:length(x)) {
7 s1[i] <- 1/(2.5696 -3.264*cos(2*pi*x[i])
8 +0.72*cos(4*pi*x[i]))
9 s2[i] <- 1/(2.8196 -3.536*cos(2*pi*x[i])
10 +0.72*cos(4*pi*x[i]))
11 s3[i] <- 1/(4.8961 -6.516*cos(2*pi*x[i])
12 +1.62*cos(4*pi*x[i]))
13 }
14 plot(x,s1,type="l",main="phi=1.2, psi=-0.36",
15 xlab="w",ylab="Sx")
16 plot(x,s2,type="l",main="phi=1.3,psi=-0.36",
```

```

17 xlab="w",ylab="Sx")
18 plot(x,s3,type="l",main="phi=1.8,psi=-0.81",
19 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：

```

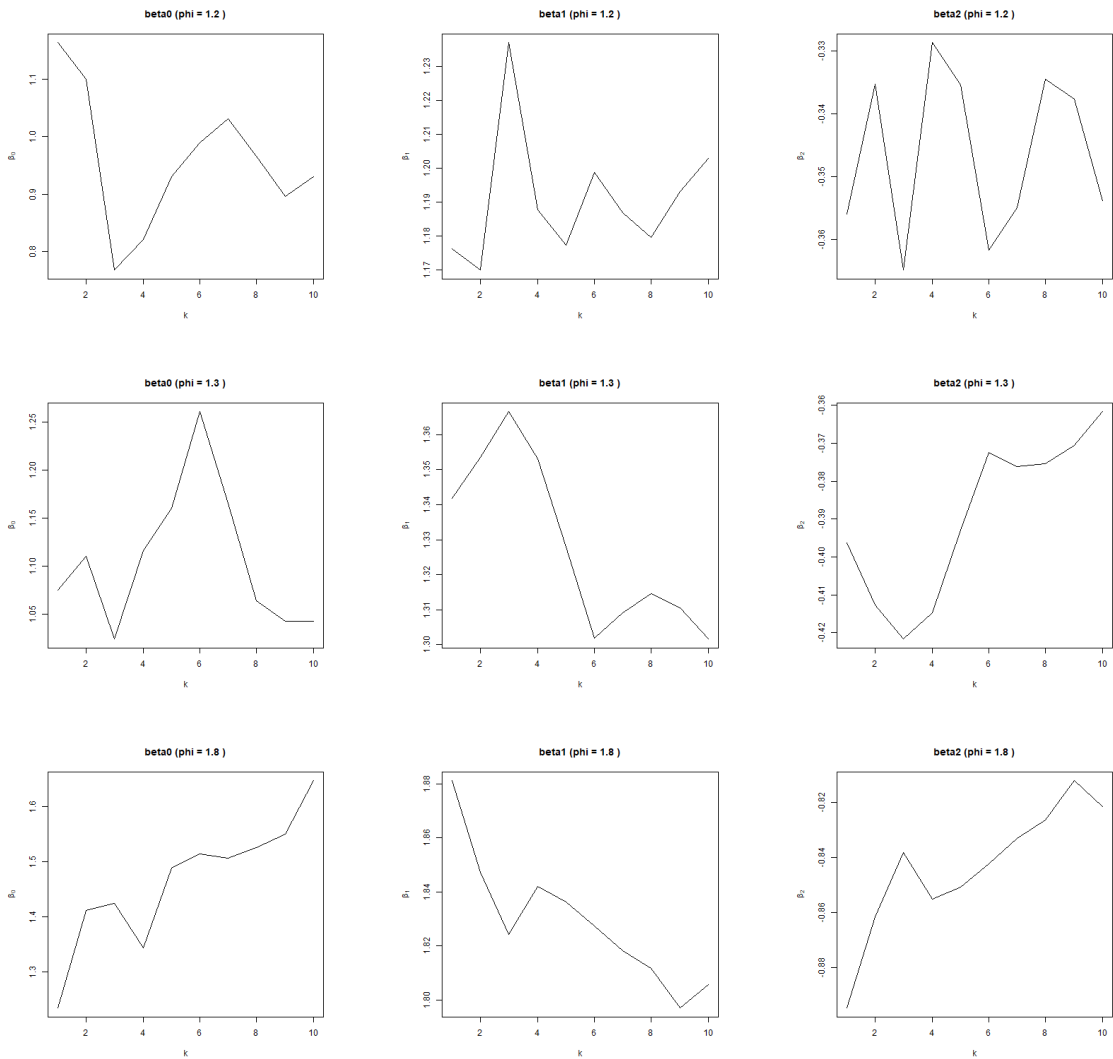
1 beta0 <- tibble ()
2 beta1 <- tibble ()
3 beta2 <- tibble ()
4 for(i in 1:3){
5   for(k in 1:10){
6     y <- unlist(X[3:(100*k+2), i])
7     x1 <- unlist(X[2:(100*k+1), i])
8     x2 <- unlist(X[1:(100*k), i])
9     ols <- coefficients(lm(y ~ x1+x2))
10    beta0[k, i] = ols[1]
11    beta1[k, i] = ols[2]
12    beta2[k, i] = ols[3]

```

```

13 }
14 plot(beta0[[i]], type = "l", xlab = "k",
15 ylab = paste('beta0(phi=', phi[i], ')'))
16 plot(beta1[[i]], type = "l", xlab = "k",
17 ylab = paste('beta1(phi=', phi[i], ')'))
18 plot(beta2[[i]], type = "l", xlab = "k",
19 ylab = paste('beta2(phi=', phi[i], ')'))
20 }

```



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:

```
1 #phi=1.2,psi=-0.36
2 X1 <- tibble ()
3 for(i in 1:1000){
4 X1[1, i] = mu/(1 - phi[1]-psi[1])
5 X1[2, i] = mu/(1 - phi[1]-psi[1])
6 for(j in 1:100){
7 X1[j+2, i] = mu + phi[1]*X1[j+1, i] + psi[1]*X1[j,i]
8 + rnorm(1,0,1)}}
9 # phi = 1.3,psi=-0.36
10 X2 <- tibble ()
11 for(i in 1:1000){
12 X2[1, i] = mu/(1 - phi[2]-psi[2])
13 X2[2, i] = mu/(1 - phi[2]-psi[2])
14 for(j in 1:100){
15 X2[j+2, i] = mu + phi[2]*X2[j+1, i] + psi[2]*X2[j,i]
16 + rnorm(1,0,1)}}
17 # phi = 1.8,psi =-0.81
18 X3 <- tibble ()
19 for(i in 1:1000){
20 X3[1, i] = mu/(1 - phi[3]-psi[3])
21 X3[2, i] = mu/(1 - phi[3]-psi[3])
22 for(j in 1:100){
23 X3[j+2, i] = mu + phi[3]*X3[j+1, i] + psi[3]*X3[j,i]
24 + rnorm(1,0,1)}}
25 X <- list(X1, X2, X3)
26
```

```

27 beta0 <- tibble ()
28 beta1 <- tibble ()
29 beta2 <- tibble ()
30 b0 <- tibble ()
31 b1 <- tibble ()
32 b2 <- tibble ()
33 sd_beta0 <- vector("double" ,2)
34 sd_beta1 <- vector("double" ,2)
35 sd_beta2 <- vector("double" ,2)
36 for(i in 1:3){
37   for(j in 1:1000){
38     y <- unlist(X[[i]][3:102 , j])
39     x1 <- unlist(X[[i]][2:101 , j])
40     x2 <- unlist(X[[i]][1:100 , j])
41     ols <- coefficients(lm(y ~ x1+x2))
42     beta0[j, i] <- ols[1]
43     beta1[j, i] <- ols[2]
44     beta2[j, i] <- ols[3]
45   }
46   hist(beta0[[i]]-mu , xlab =
47     paste('beta0(phi=', phi[i], ',psi=',psi[i],')'),
48     main = paste('Histogram_of_beta0(phi=', phi[i], ',
49     psi=',psi[i],')'))
50   hist(beta1[[i]]-mu , xlab =
51     paste('beta1(phi=', phi[i], ',psi=',psi[i],')'),
52     main = paste('Histogram_of_beta1(phi=', phi[i], ',

```

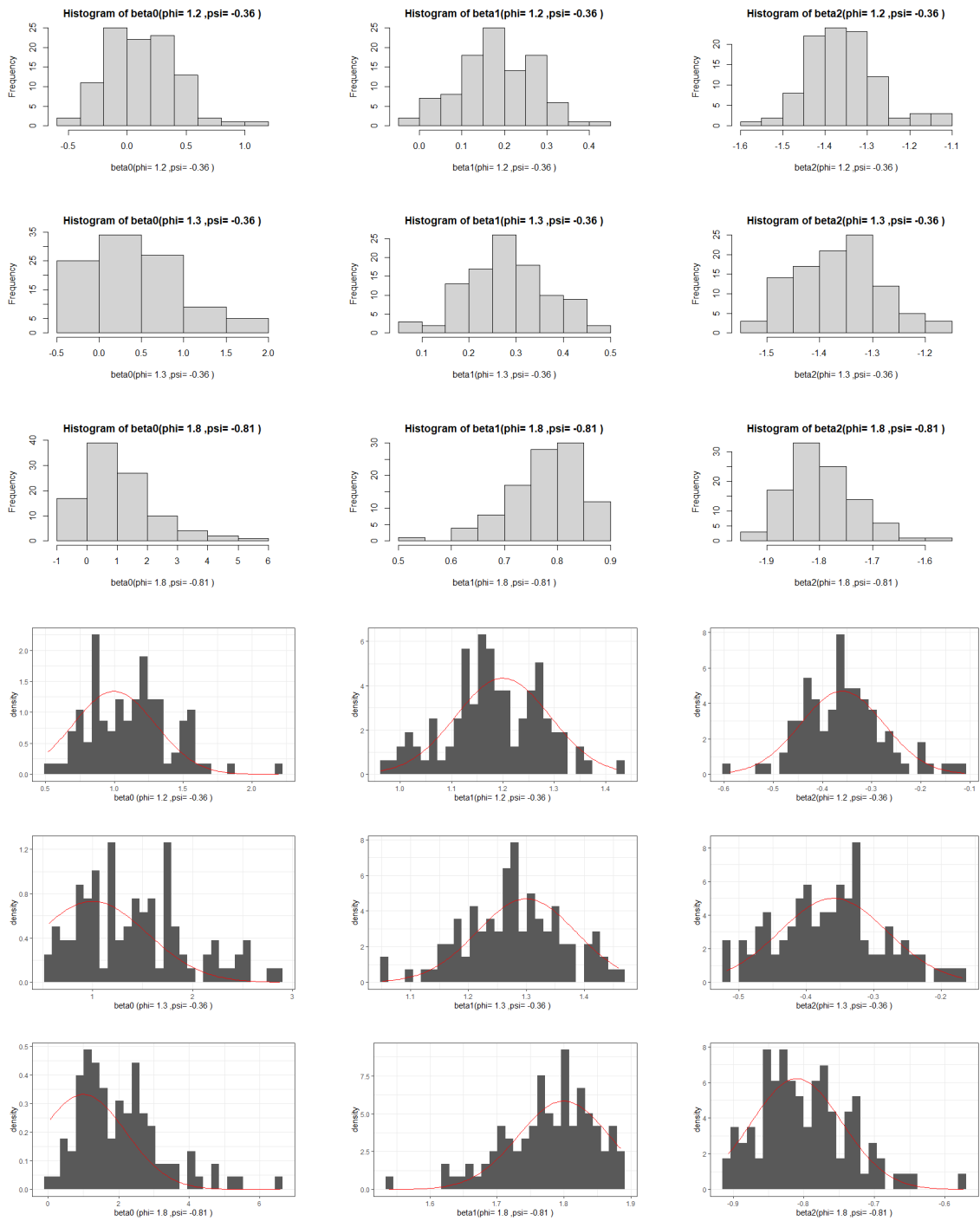


```

53 psi=',psi[i],')'))
54 hist(beta2[[i]]-mu , xlab =
55 paste('beta2(phi=', phi[i],',psi=',psi[i],')'),
56 main = paste('Histogram_of_beta2(phi=', phi[i],',
57 psi=',psi[i],')'))
58
59 sd_beta0[i] <- sd(beta0[[i]])
60 sd_beta1[i] <- sd(beta1[[i]])
61 sd_beta2[i] <- sd(beta2[[i]])
62 }
63
64 sd_beta0
65 ## [1] 0.2981030 0.5450414 1.2014578
66 sd_beta1
67 ## [1] 0.09186399 0.08519157 0.06825279
68 sd_beta2
69 ## [1] 0.08491475 0.07978170 0.06417489
70
71 for(i in 1:3){
72 print(ggplot()+ geom_histogram (data=NULL,
73 mapping=aes(x=beta0[[i]],y=..density..))
74 + xlab(paste('beta0(phi=', phi[i],',psi=',psi[i],')'))
75 + geom_function(fun = dnorm,color="red",
76 args=list(mean=mu,sd=sd_beta0[i])))
77 print(ggplot()+ geom_histogram (data=NULL,
78 mapping=aes(x=beta1[[i]],y=..density..))

```

```
79 + xlab(paste('beta1(phi=', phi[i], ',psi=',psi[i],')'))
80 + geom_function(fun=dnorm ,color="red",
81 args=list(mean=phi[i],sd=sd_beta1[i]))
82 print(ggplot()+ geom_histogram (data=NULL,
83 mapping=aes(x=beta2[[i]],y=..density..))
84 + xlab(paste('beta2(phi=', phi[i], ',psi=',psi[i],')'))
85 + geom_function(fun=dnorm,color="red",
86 args=list(mean=psi[i],sd=sd_beta2[i]))
87 }
```



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：

```

1 # 理论渐近标准误
2 mu <- 1
3 sigma <- 1

```

```

4 T <- 100
5 AR2_CM <- function(phi, psi, mu , sigma ){
6
7 mu_X <- mu / (1 - phi - psi)
8
9 A_matrix <- matrix(c(0, phi, psi,
10                     phi, psi, 0,
11                     phi, psi, 0),
12                    nrow = 3, byrow = TRUE)
13 I_matrix <- diag(3)
14 b_vector <- c(1, 0, 0)
15
16 gamma_vec <- solve(I_matrix - A_matrix, b_vector)
17 gamma_0 <- gamma_vec[1]
18 gamma_1 <- gamma_vec[2]
19 gamma_2 <- gamma_vec[3]
20 E_X_t_1 <- mu_X
21 E_X_t_2 <- mu_X
22 E_X_t1_sq <- gamma_0 + mu_X^2
23 E_X_t1_X_t2 <- gamma_1 + mu_X^2
24 E_X_t2_sq <- gamma_0 + mu_X^2
25 M <- matrix(c(1, E_X_t_1, E_X_t_2,
26              E_X_t_1, E_X_t1_sq, E_X_t1_X_t2,
27              E_X_t_2, E_X_t1_X_t2, E_X_t2_sq),
28            nrow = 3, byrow = TRUE)
29 CM <- solve(M)*sigma^2*(1/T)

```

```

30 return(CM)
31 }
32
33 AR2_CM(phi [1],psi [1],mu=1,sigma=1)
34 #           [,1]           [,2]           [,3]
35 #[1,]  0.11742187 -0.00859375 -0.00859375
36 #[2,] -0.00859375  0.01168750 -0.01031250
37 #[3,] -0.00859375 -0.01031250  0.01168750
38 AR2_CM(phi [2],psi [2],mu=1,sigma=1)
39 #           [,1]           [,2]           [,3]
40 #[1,]  0.298220551 -0.008646617 -0.008646617
41 #[2,] -0.008646617  0.011759398 -0.011240602
42 #[3,] -0.008646617 -0.011240602  0.011759398
43 AR2_CM(phi [3],psi [3],mu=1,sigma=1)
44 #           [,1]           [,2]           [,3]
45 #[1,]  1.561246537 -0.007756233 -0.007756233
46 #[2,] -0.007756233  0.014038781 -0.013961219
47 #[3,] -0.007756233 -0.013961219  0.014038781
48 sd_theory <- vector(mode = "list", length = 3)
49 for(i in 1:3){
50 sd_theory [[i]] <- c(sd_beta0 = sqrt(AR2_CM(phi [i],
51 psi [i],mu=1,sigma=1)[1,1]) ,
52 sd_beta1 =sqrt(AR2_CM(phi [i],psi [i],mu=1,sigma=1)[2,2]) ,
53 sd_beta2 =sqrt(AR2_CM(phi [i],psi [i],mu=1,sigma=1)[3,3]))
54 }
55 names(sd_theory) <- c("phi=1.2,psi=-0.36","phi=1.3,

```

```

56 psi=-0.36","phi=1.8,psi=-0.81")
57 sd_theory
58 # $phi =1.2,psi ==-0.36
59 # sd_beta0 sd_beta1 sd_beta2
60 # 0.3426688 0.1081087 0.1081087
61 #
62 # $phi =1.3,psi ==-0.36
63 # sd_beta0 sd_beta1 sd_beta2
64 # 0.5460957 0.1084408 0.1084408
65 #
66 # $phi =1.8,psi ==-0.81
67 # sd_beta0 sd_beta1 sd_beta2
68 # 1.2494985 0.1184854 0.1184854
69
70 # 样本渐近标准误
71 x <- vector("double" ,101)
72 for (i in 1:3) {
73 x[1] <- mu/(1 - phi[i]-psi[i])
74 x[2] <- mu/(1 - phi[i]-psi[i])
75 for (j in 1:100) {
76 x[j+2] = mu + phi[3]*x[j+1] + psi[3]*x[j]
77 + rnorm(1 ,0 ,1)}
78 X <- cbind(rep(1 ,100) , x[2:101], x[1:100])
79 M <- t(X) %*% X / 100
80 CM<- 1/100*1*solve(M)
81 print(CM)

```

```

82 sd_sample<-c(sd_beta0=sqrt(CM[1 ,1]) ,
83 sd_beta1=sqrt(CM [2 ,2]),
84 sd_beta2=sqrt(CM [3 ,3]))
85 print(sd_sample)
86 }
87
88 #           [,1]           [,2]           [,3]
89 #[1,]  0.12873521 -0.012615582  0.011248837
90 #[2,] -0.01261558  0.003611043 -0.003490062
91 #[3,]  0.01124884 -0.003490062  0.003385072
92 #sd_beta0  sd_beta1  sd_beta2
93 #0.35879689 0.06009196 0.05818137
94 #           [,1]           [,2]           [,3]
95 #[1,]  0.099783739 -0.009623794  0.008496649
96 #[2,] -0.009623794  0.005880669 -0.005827523
97 #[3,]  0.008496649 -0.005827523  0.005789471
98 #sd_beta0  sd_beta1  sd_beta2
99 #0.31588564 0.07668552 0.07608857
100 #           [,1]           [,2]           [,3]
101 #[1,]  0.352987307  0.003876617 -0.007794409
102 #[2,]  0.003876617  0.003977387 -0.004003679
103 #[3,] -0.007794409 -0.004003679  0.004074804
104 #sd_beta0  sd_beta1  sd_beta2
105 #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：

```

1 X1 <- tibble ()
2 for(i in 1:1000){
3 X1[1, i] = mu/(1 - phi[1]-psi[1])
4 X1[2, i] = mu/(1 - phi[1]-psi[1])
5 for(j in 1:900){
6 X1[j+2, i] = mu + phi[1]*X1[j+1, i] + psi[1]*X1[j,i]
7 + rnorm(1 ,0 ,1)}}
8 # phi = 1.3, psi=-0.36
9 X2 <- tibble ()
10 for(i in 1:1000){
11 X2[1, i] = mu/(1 - phi[2]-psi[2])
12 X2[2, i] = mu/(1 - phi[2]-psi[2])
13 for(j in 1:900){
14 X2[j+2, i] = mu + phi[2]*X2[j+1, i] + psi[2]*X2[j,i]
15 + rnorm(1 ,0 ,1)}}
16 # phi = 1.8, psi = -0.81
17 X3 <- tibble ()
18 for(i in 1:1000){
19 X3[1, i] = mu/(1 - phi[3]-psi[3])
20 X3[2, i] = mu/(1 - phi[3]-psi[3])
21 for(j in 1:900){
22 X3[j+2, i] = mu + phi[3]*X3[j+1, i] + psi[3]*X3[j,i]
23 + rnorm(1 ,0 ,1)}}
24 X <- list(X1 , X2 , X3)
25 beta0 <- tibble ()
26 beta1 <- tibble ()

```



```

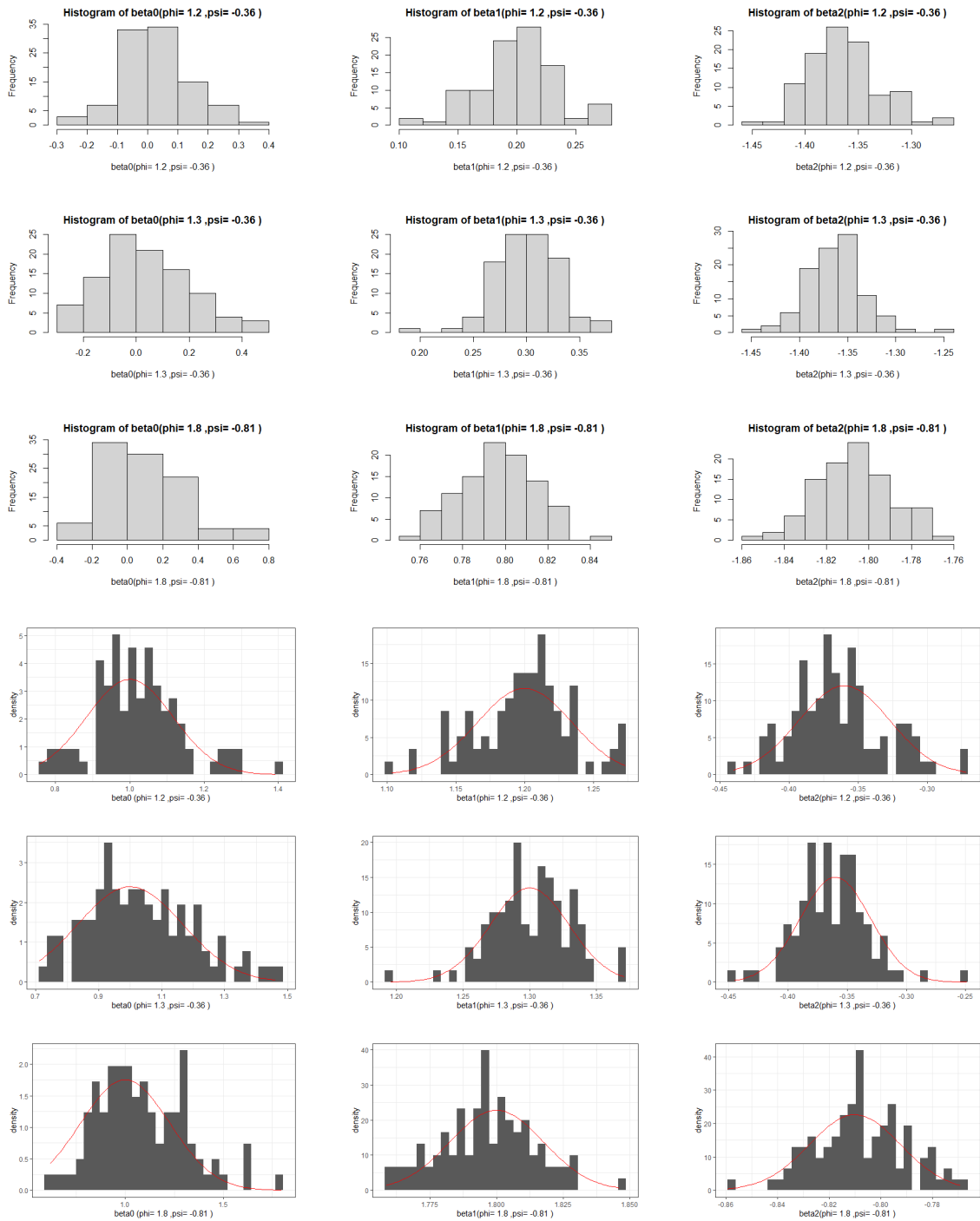
27 beta2 <- tibble ()
28 b0 <- tibble ()
29 b1 <- tibble ()
30 b2 <- tibble ()
31 sd_beta0 <- vector("double" ,2)
32 sd_beta1 <- vector("double" ,2)
33 sd_beta2 <- vector("double" ,2)
34 for(i in 1:3){
35   for(j in 1:100){
36     y <- unlist(X[[i]][3:902 , j])
37     x1 <- unlist(X[[i]][2:901 , j])
38     x2 <- unlist(X[[i]][1:900 , j])
39     ols <- coefficients(lm(y ~ x1+x2))
40     beta0[j, i] <- ols[1]
41     beta1[j, i] <- ols[2]
42     beta2[j, i] <- ols[3]
43   }
44   hist(beta0[[i]]-mu , xlab = paste('beta0(phi=',
45     phi[i], ',psi=',psi[i],')'),
46     main = paste('Histogram_of_beta0(phi=', phi[i], ',
47     psi=',psi[i],')'))
48   hist(beta1[[i]]-mu , xlab = paste('beta1(phi=',
49     phi[i], ',psi=',psi[i],')'),
50     main = paste('Histogram_of_beta1(phi=', phi[i], ',
51     psi=',psi[i],')'))
52   hist(beta2[[i]]-mu , xlab = paste('beta2(phi=',

```

```

53 phi[i],',psi=',psi[i],')'),
54 main = paste('Histogram_of_beta2(phi=', phi[i],',
55 psi=',psi[i],')')')
56 sd_beta0[i] <- sd(beta0[[i]])
57 sd_beta1[i] <- sd(beta1[[i]])
58 sd_beta2[i] <- sd(beta2[[i]])
59 }
60 sd_beta0
61 sd_beta1
62 sd_beta2
63 for(i in 1:3){
64 print(ggplot()+ geom_histogram (data=NULL ,
65 mapping=aes(x=beta0[[i]],y=..density..))
66 + xlab(paste('beta0(phi=', phi[i],',psi=',psi[i],')'))
67 + geom_function(fun = dnorm ,color="red",
68 args=list(mean=mu ,sd=sd_beta0[i])))
69 print(ggplot()+ geom_histogram (data=NULL ,
70 mapping=aes(x=beta1[[i]],y=..density..))
71 + xlab(paste('beta1(phi=', phi[i],',psi=',psi[i],')'))
72 + geom_function(fun=dnorm ,color="red",
73 args=list(mean=phi[i],sd=sd_beta1[i])))
74 print(ggplot()+ geom_histogram (data=NULL ,
75 mapping=aes(x=beta2[[i]],y=..density..))
76 + xlab(paste('beta2(phi=', phi[i],',psi=',psi[i],')'))
77 + geom_function(fun=dnorm ,color="red",
78 args=list(mean=psi[i],sd=sd_beta2[i])))

```



```
1 # 理论渐近标准误
```

```
2 mu <- 1
```

```
3 sigma <- 1
```

```

4 T <- 900
5 AR2_CM <- function(phi, psi, mu , sigma ){
6
7 mu_X <- mu / (1 - phi - psi)
8
9 A_matrix <- matrix(c(0, phi, psi,
10                     phi, psi, 0,
11                     phi, psi, 0), nrow = 3, byrow = TRUE)
12 I_matrix <- diag(3)
13 b_vector <- c(1, 0, 0)
14
15 gamma_vec <- solve(I_matrix - A_matrix, b_vector)
16 gamma_0 <- gamma_vec[1]
17 gamma_1 <- gamma_vec[2]
18 gamma_2 <- gamma_vec[3]
19 E_X_t_1 <- mu_X
20 E_X_t_2 <- mu_X
21 E_X_t1_sq <- gamma_0 + mu_X^2
22 E_X_t1_X_t2 <- gamma_1 + mu_X^2
23 E_X_t2_sq <- gamma_0 + mu_X^2
24 M <- matrix(c(1, E_X_t_1, E_X_t_2,
25               E_X_t_1, E_X_t1_sq, E_X_t1_X_t2,
26               E_X_t_2, E_X_t1_X_t2, E_X_t2_sq),
27             nrow = 3, byrow = TRUE)
28 CM <- solve(M)*sigma^2*(1/T)
29 return(CM)

```

```

30 }
31
32 AR2_CM(phi [1], psi [1], mu=1, sigma=1)
33 #           [,1]           [,2]           [,3]
34 #[1,]  0.0130468750 -0.0009548611 -0.0009548611
35 #[2,] -0.0009548611  0.0012986111 -0.0011458333
36 #[3,] -0.0009548611 -0.0011458333  0.0012986111
37 AR2_CM(phi [2], psi [2], mu=1, sigma=1)
38 #           [,1]           [,2]           [,3]
39 #[1,]  0.0331356168 -0.0009607352 -0.0009607352
40 #[2,] -0.0009607352  0.0013065998 -0.0012489557
41 #[3,] -0.0009607352 -0.0012489557  0.0013065998
42 AR2_CM(phi [3], psi [3], mu=1, sigma=1)
43 #           [,1]           [,2]           [,3]
44 #[1,]  0.1734718375 -0.0008618036 -0.0008618036
45 #[2,] -0.0008618036  0.0015598646 -0.0015512465
46 #[3,] -0.0008618036 -0.0015512465  0.0015598646
47
48 sd_theory <- vector(mode = "list", length = 3)
49 for(i in 1:3){
50 sd_theory[[i]] <- c(sd_beta0 = sqrt(AR2_CM(phi [i],
51 psi [i], mu=1, sigma=1) [1,1]) ,
52 sd_beta1 =sqrt(AR2_CM(phi [i], psi [i], mu=1, sigma=1) [2,2]) ,
53 sd_beta2 =sqrt(AR2_CM(phi [i], psi [i], mu=1, sigma=1) [3,3]))
54 }
55 names(sd_theory) <- c("phi=1.2,psi=-0.36","phi=1.3,

```

```

56 psi=-0.36","phi=1.8,psi=-0.81")
57 sd_theory
58 # $phi =1.2,psi ==-0.36
59 #   sd_beta0   sd_beta1   sd_beta2
60 # 0.11422292 0.03603625 0.03603625
61 #
62 # $phi =1.3,psi ==-0.36
63 #   sd_beta0   sd_beta1   sd_beta2
64 # 0.18203191 0.03614692 0.03614692
65 #
66 # $phi =1.8,psi ==-0.81
67 #   sd_beta0   sd_beta1   sd_beta2
68 # 0.41649950 0.03949512 0.03949512
69
70 # 样本渐近标准误
71 x <- vector("double" ,901)
72 for (i in 1:3) {
73 x[1] <- mu/(1 - phi[i]-psi[i])
74 x[2] <- mu/(1 - phi[i]-psi[i])
75 for (j in 1:900) {
76 x[j+2] = mu + phi [3]*x[j+1] + psi[3]*x[j] +
77 rnorm(1 ,0 ,1)}
78 X <- cbind(rep(1 ,900) , x[2:901], x[1:900])
79 M <- t(X) %*% X / 900
80 CM<- 1/900*1*solve(M)
81 print(CM)

```

```

82 sd_sample<-c(sd_beta0=sqrt(CM[1 ,1]) ,
83 sd_beta1=sqrt(CM [2 ,2]),
84 sd_beta2=sqrt(CM [3 ,3]))
85 print(sd_sample)
86 }
87 #           [,1]           [,2]           [,3]
88 #[1,]  0.0304557095 -0.0006450578  0.0003360185
89 #[2,] -0.0006450578  0.0004005666 -0.0003942706
90 #[3,]  0.0003360185 -0.0003942706  0.0003912299
91 # sd_beta0 sd_beta1 sd_beta2
92 #0.17451564 0.02001416 0.01977953
93 #           [,1]           [,2]           [,3]
94 #[1,]  0.0432229056 -0.0007491774  0.0003430015
95 #[2,] -0.0007491774  0.0003474072 -0.0003405139
96 #[3,]  0.0003430015 -0.0003405139  0.0003375386
97 # sd_beta0 sd_beta1 sd_beta2
98 #0.20790119 0.01863886 0.01837222
99 #           [,1]           [,2]           [,3]
100 #[1,]  0.0354082586 -0.0001391447 -0.0002124855
101 #[2,] -0.0001391447  0.0003852756 -0.0003839542
102 #[3,] -0.0002124855 -0.0003839542  0.0003862379
103 # sd_beta0 sd_beta1 sd_beta2
104 #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。