华中师范大学

实验报告书

2022 年 11 月 30 日

课程名称:	时间序列分析
专　　业:	统计学
年　　级:	2020 级
学生姓名:	陈启源
学　　号:	2020211946
指导教师:	张晓飞

华中师范大学数学与统计学学院

1 问题 1

1.1 问题重述

根据 (6.1.1) 式写出计算样本自相关函数的 R 程序，并且用该程序完成 6.24 题。

1.2 样本自相关函数 R 程序

对于给定的序列 $\{Y_t\}$，样本的自相关函数计算公式如下所示：

$$r_k = \frac{\sum_{t=k+1}^{n}(Y_t - \overline{Y})(Y_{t-k} - \overline{Y})}{\sum_{t=1}^{n}(Y_t - \overline{Y})^2} \tag{1}$$

```r
acf_func = function(series) {
  # 样本自相关函数计算函数
  n = length(series)
  series_mean = mean(series)
  series = series - series_mean
  acf_result = rep(0, n - 1)
  sum = series %*% series
  for (k in 1:(n - 1)) {
    acf_result[k] = (series[(k + 1):n] %*% series[1:(n - k)]) / sum
  }
  return(acf_result)
}
```

2 习题 6.24

2.1 问题重述

模拟 $\theta = 0.7$ 的 MA(1) 时间序列

(a) n=24，估计 ρ_1 和 r_1.

(b) n=60，估计 ρ_1 和 r_1.

(c) n=120，估计 ρ_1 和 r_1.

(d) 对 (a)(b)(c) 中的每个序列，比较 ρ_1 的计值和论值用图表 6-2 量化这些比较，简要描述估计的精度如何随着样本容量的变化而变化.

2.2 代码

理论自相关函数

```
MA_thm_acf = function(MA_params) {
  q=length(MA_params)
  rho0=1+(MA_params%*%MA_params)
  rho = rep(0,(q+1))
  if (q > 1){
    for (k in 1:(q-1)){
      rho[k]=(-MA_params[k]+(MA_params[1:(q-k)]%*%
      MA_params[(k+1):q]))/rho0
    }
  }
  rho[q]= (-MA_params[q])/rho0
  return(rho)
}
```

6.24 题代码（序列模拟部分代码为上节作业）

```
source("./sm_series.R") # 导入模拟生成序列的函数
source("./acf_func.R") # 导入计算理论、样本自相关函数

set.seed(1111) # 设置随机种子以保证结果可复现

# (a)
sm_series1 = ARMA_func(AR_params = NULL,
                       MA_params = c(0.7),
                       seq_length = 24)
```

```r
(rho1 = MA_thm_acf(c(0.7)))
r1 = acf_func(sm_series1)
r1[1]

# (b)
sm_series2 = ARMA_func(AR_params = NULL,
                       MA_params = c(0.7),
                       seq_length = 60)
(rho2 = MA_thm_acf(c(0.7)))
r2 = acf_func(sm_series2)
r2[1]

# (c)
sm_series3 = ARMA_func(AR_params = NULL,
                       MA_params = c(0.7),
                       seq_length = 120)
(rho3 = MA_thm_acf(c(0.7)))
r3 = acf_func(sm_series3)
r3[1]
```

2.3 结果

(a)

$\rho_1 = -0.4697987, r_1 = -0.2739706.$

(b)

$\rho_1 = -0.4697987, r_1 = -0.3845085.$

(c)

$\rho_1 = -0.4697987, r_1 = -0.4618916.$

(d)

通过观察结果可以发现，对于越大的 n，ρ_1 的估计值距离理论值越接近。量化表格如下表所示：

n	$\sqrt{\mathrm{Var}(\mathrm{r}_1)}$	$\sqrt{\mathrm{Var}(\mathrm{r}_\mathrm{k})}, k > 1$	$\mathrm{Corr}(\mathrm{r}_1, \mathrm{r}_2)$
24	0.1490	0.2490	-0.84
60	0.0942	0.1575	-0.84
120	0.0666	0.1114	-0.84

表 1: 参数为 0.7 的 MA(1) 模型选定的 r_k 的大样本结果

通过表格中的数据可以知道，随着 n 的增大，估计的精度逐渐提高。

3 问题 2

3.1 问题重述

根据 (6.2.8) 式写出计算样本偏自相关函数的 R 程序，并用该程序完成 6.25，6.27，6.28，6.33 题。

3.2 样本偏自相关函数 R 程序

为了验证结果的准确性，本题使用两种方法来计算样本的偏自相关函数。分别按照书本中的公式 6.2.8 和 6.2.9 来计算。

3.2.1 方法一：参考 6.2.8

```r
pacf_func = function(series, max_lag = 20) {
  # 6.2.8的实现
  rho = acf_func(series) # 得到样本自相关函数
  M = matrix(1, nrow = max_lag, ncol = max_lag) #phi是一个函数
  for (i in 1:(max_lag - 1)) {
    for (j in (i + 1):max_lag) {
      M[i, j] = rho[j - i]
      M[j, i] = rho[j - i] #对称矩阵
    }
  }
  # 得到完整的矩阵
  phi_kk = rep(rho[1], max_lag)
  for (k in 2:max_lag) {
    result = qr.solve(M[1:k, 1:k], rho[1:k])
    phi_kk[k] = result[k]
  }
  return(phi_kk)
}
```

3.2.2 方法二：参考 6.2.9

```r
pacf_func_1 = function(series, max_lag = 20) {
  # 偏自相关的另外一种实现 6.2.9
  rho = acf_func(series) # 得到样本自相关函数
  phi = matrix(0, nrow = max_lag, ncol = max_lag) #phi是一个函数
  phi[1, 1] = rho[1]
```

```
  for (k in 2:max_lag) {
    a = rho[k] - phi[(k - 1), 1:(k - 1)] %*% rho[(k - 1):1]
    b = 1 - phi[(k - 1), 1:(k - 1)] %*% rho[1:(k - 1)]
    phi[k, k] = a / b
    for (j in 1:(k - 1)) {
      phi[k, j] = phi[k - 1, j] - phi[k, k] * phi[k - 1, k - j]
    }
  }
  return(diag(phi))
}
```

3.2.3 代码验证

```
> source("./acf_func.R")
> library(TSA)
> data(ar2.s)
>
> pacf_func(ar2.s)
 [1]  0.832735369 -0.764603375 -0.036765964 -0.105071822 -0.037105344
 [6]  0.005517211 -0.022027182  0.051601004 -0.237199486 -0.041937692
[11] -0.007581464  0.005142360 -0.023581125  0.040143306  0.085986034
[16] -0.064500144  0.126806184  0.043526077  0.040602444  0.053742143
> pacf_func_1(ar2.s)
 [1]  0.832735369 -0.764603375 -0.036765964 -0.105071822 -0.037105344
 [6]  0.005517211 -0.022027182  0.051601004 -0.237199486 -0.041937692
[11] -0.007581464  0.005142360 -0.023581125  0.040143306  0.085986034
[16] -0.064500144  0.126806184  0.043526077  0.040602444  0.053742143
> pacf(ar2.s)$acf #书本结果
, , 1

             [,1]
 [1,]  0.832735369
 [2,] -0.764603375
 [3,] -0.036765964
 [4,] -0.105071822
 [5,] -0.037105344
 [6,]  0.005517211
```

```
[7,] -0.022027182
[8,]  0.051601004
[9,] -0.237199486
[10,] -0.041937692
[11,] -0.007581464
[12,]  0.005142360
[13,] -0.023581125
[14,]  0.040143306
[15,]  0.085986034
[16,] -0.064500144
[17,]  0.126806184
[18,]  0.043526077
[19,]  0.040602444
[20,]  0.053742143
```

通过对比书本内置包计算的结果和使用自编函数求解的结果可以发现，参考公式 6.2.8 和参考公式 6.2.9 的算法均可以得到和书本内置库相同的结果。这一结果验证了我们自编算法的正确性。**接下来问题的求解，均使用参考公式 6.2.8 的函数 pacf_func()**。

4 问题 6.25

4.1 问题重述

模拟一个长度 $n = 36, \phi = 0.7$ 的 AR(1) 时间序列.

(a) 计算并画出该模型的理论自相关函数，画出足够多的滞后项直到相关性可以忽略不计.

(b) 计算并画出模序列的样本 ACF，样本值和模式与 (a) 部分的理论 ACF 匹配度如何?

(c) 该模型的理论偏自相关函数是什么?

(d) 计算并画出模拟序列的样本 ACF 本值和模式与 (a) 部分的理论 ACF 匹配程度如何? 用图表 6-1 列出的大样本标准误差量化你的答案.

(e) 计算并画出模序列的样本 PACF，样本值和模式与 (c) 部分的理论 PACF 匹配程度如何? 用列出的大样本标准误差量化你的答案.

4.2 问题求解

4.2.1 (a)

对于 AR(1) 模型而言，理论自相关函数 $\rho_k = \phi^k$，此处 $\phi = 0.7$，所以 $\rho_k = (0.7)^k$。

代码:

```r
source("acf_func.R")
source("sm_series.R")

set.seed(1111)
series = ARMA_func(c(0.7), NULL, 36)
# (a)
thm_acf = AR1_thm_acf(c(0.7))
draw_acf(thm_acf) # 绘图
```

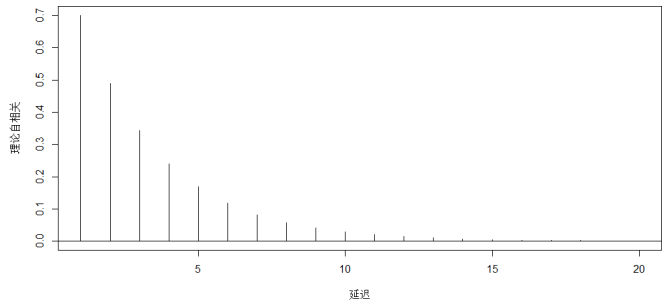

图 1: 理论自相关函数图

4.2.2 (b)

```
# (b)
acf = acf_func(series)
draw_acf(acf,type="样本自相关")
cor(acf[1:20],thm_acf)
```

输出：

```
 [1]  0.7557180868  0.4731693162  0.3187728292  0.1878935292  0.1201211674
 [6]  0.0892679493  0.0701123057  0.0042315156 -0.0453104431 -0.1099890752
[11] -0.1569086282 -0.1679295386 -0.2264620243 -0.3211389510 -0.3449408214
[16] -0.3011113455 -0.2516825792 -0.1832272340 -0.1005578593 -0.0147108574
[21]  0.0058412527 -0.0291037700 -0.0749063076 -0.1051141070 -0.1304395095
[26] -0.0871653467  0.0139529197  0.0463146124  0.0356216403  0.0201457306
[31]  0.0043846093  0.0026592505  0.0017974764  0.0004968046  0.0001974024

[1] 0.9432865
```

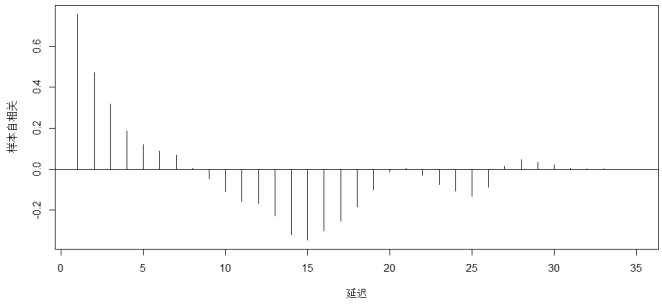

图 2: 样本自相关函数图

通过观察理论自相关函数图和样本自相关函数图，可以发现两者虽然相关系数较高，但是趋势差异较大，匹配程度差。

4.2.3 (c)

由偏自相关函数的定义可以知道，$\phi_{11} = 0.7; \phi_{kk} = 0, k > 1$。

4.2.4 (d)

具体结果参考表 2。

n	$\sqrt{\mathrm{Var}(r_1)}$	$\sqrt{\mathrm{Var}(r_2)}$	$\mathrm{Corr}(r_1, r_2)$	$\sqrt{\mathrm{Var}(r_{10})}$
36	0.1183	0.1866	0.89	0.2833

表 2: 参数为 0.7 的 AR(1) 模型选定的 r_k 的大样本结果

4.2.5 (e)

```r
draw_pacf = function(acf_series, type = "样本偏自相关", n=36) {
  plot(acf_series, type = 'h', xlab = '延迟' ,ylab = type)
  abline(h = 0)
  abline(h = c(2*sqrt(1/n),-2*sqrt(1/n)),lty=2)
}

# (e)
pacf=pacf_func(series)
draw_pacf(pacf,n=36)
```

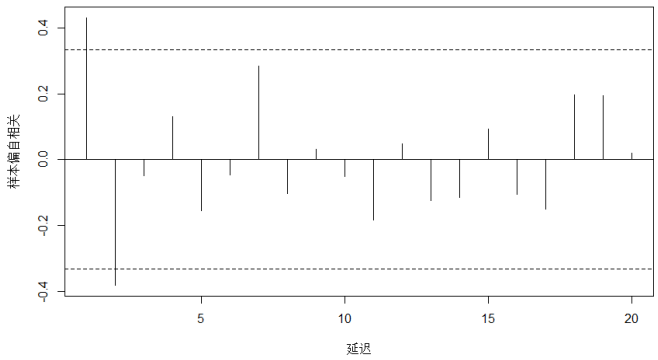

图 3: 样本偏自相关函数图

通过观察样本自相关函数图和理论自相关函数的计算结果，可以发现二者很不匹配。但是在一阶滞后的情况下，二者的取值相对比较接近。同时由于 ϕ_{kk} 在大样本总体下服从一个均值为 0，方差为 $\frac{1}{n}$ 的正态分布，所以结合图中绘制的 $x = \pm\frac{2}{\sqrt{n}}$ 线，发现一阶后所有的结果都落在其中，所以可以认为此处偏自相关函数对于模型识别依旧是有效的。

5 问题 6.27

5.1 问题重述

模拟一个长度 $n = 72, \phi_1 = 0.7, \phi_2 = -0.4$ 的 AR(2) 时间序列.
(a) 计算并画出该模型的理论自相关函数，画出足够多的滞后项，直到相关性可以忽略不计.
(b) 计算并画出模拟序列的样本 ACF. 样本值和模式与 (a) 分的理论 ACF 匹配程度如何?
(c) 该模型的理论偏自相关函数是什么?
(d) 计算并画出模序列的样本 ACF. 本值和式与 (a) 部分的理论 ACF 匹配度如何?
(e) 计算并画出模拟序列的样本 PACF. 样本值和模式与 (c) 部分的理论 PACF 匹配程度如何?

5.2 问题求解

5.2.1 (a)

对于 AR(2) 模型而言，理论自相关函数可以用
代码:

```r
AR2_thm_acf = function(AR_params,max_lag=20) {
  rho = rep(NA, max_lag)
  phi1 = AR_params[1]
  phi2 = AR_params[2]
  rho[1] = phi1 / (1 - phi2)
  rho[2] = (phi2 * (1 - phi2) + phi1 ^ 2) / (1 - phi2)
  for (k in 3:max_lag) {
    rho[k] = phi1 * rho[k - 1] + phi2 * rho[k - 2]
  }
  return(rho)
}

source("acf_func.R")
source("sm_series.R")
set.seed(666)
series = ARMA_func(c(0.7,-0.4),NULL, 72)
# (a)
thm_acf = AR2_thm_acf(c(0.7,-0.4))
draw_acf(thm_acf)
```

5.2.2 (b)

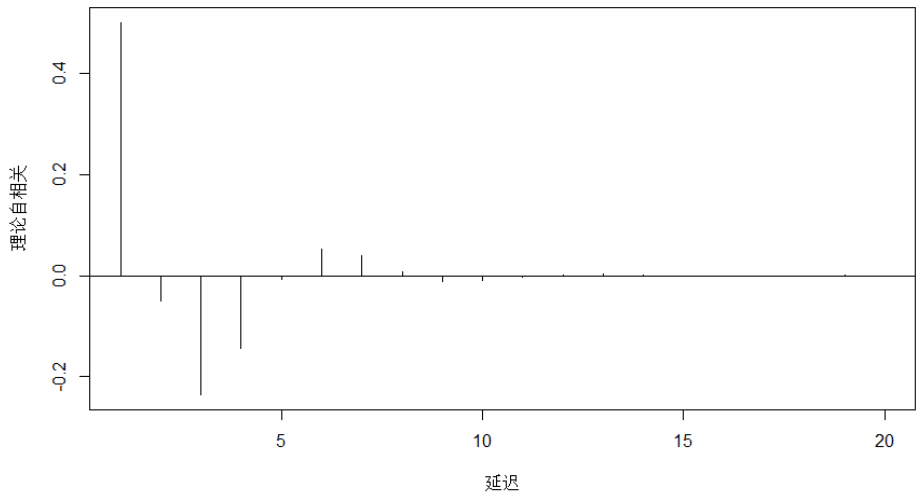

图 4: 理论自相关函数图

```r
# (b)
(acf = acf_func(series))
draw_acf(acf[1:20],type="样本自相关")
cor(acf[1:20],thm_acf)
```

输出:

```
[1]  0.430456760 -0.125766001 -0.273816785 -0.044915986  0.023016913
 [6] -0.050221797  0.098822398  0.153496540  0.087004777 -0.120615535
[11] -0.231538550 -0.122266588 -0.063067094 -0.048787152 -0.021561483
[16]  0.027598090 -0.041725500 -0.067410467  0.131851547  0.253161840
[21]  0.052726461 -0.162678892 -0.112289450  0.068664098  0.064724870
[26] -0.053335460  0.025404707  0.143887031  0.200409043 -0.046618763
[31] -0.154838513 -0.052668070  0.118984048  0.096747540 -0.030772673
[36] -0.044670519 -0.072414601 -0.128530971 -0.094386324  0.008680600
[41]  0.057323489 -0.048380641 -0.134154702 -0.106137976 -0.074378584
[46] -0.006886401  0.060202405  0.139166841  0.118517980 -0.047113780
[51] -0.114929504 -0.073108793  0.062894344  0.066238840 -0.015964390
[56] -0.003995475 -0.014937171 -0.003529220 -0.068862557 -0.062016845
[61] -0.028105237 -0.007457668  0.004189600 -0.020416763  0.009515941
[66]  0.005746024 -0.010914345 -0.016575908  0.002647947  0.012549359
[71] -0.001866896
```

```
16
17 [1] 0.723542
```

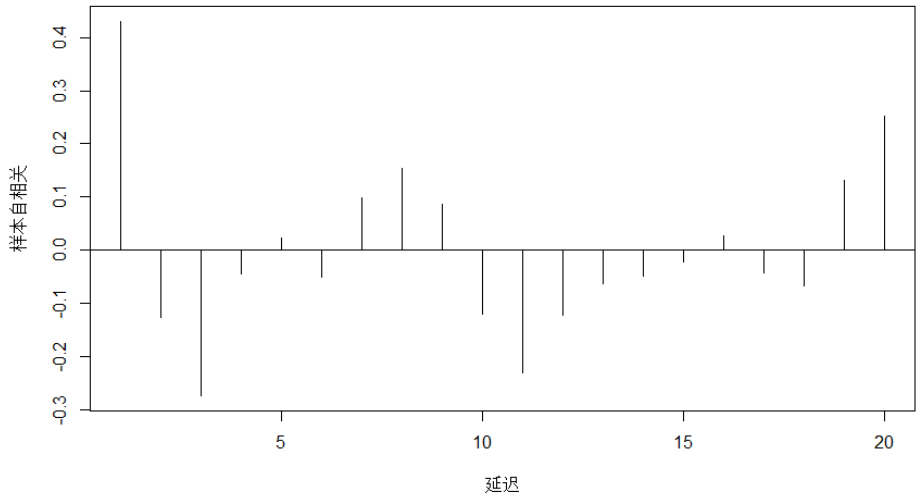

图 5: 样本自相关函数图

通过观察理论自相关函数图和样本自相关函数图，可以发现两者相关系数较低，而且趋势差异较大，匹配程度差。

5.2.3 (c)

由偏自相关函数的定义可以知道，$\phi_{11} = \rho_1 = \frac{\phi_1}{1-\phi_2} = \frac{0.7}{1+0.4} = 0.5$;
$\rho_2 = \frac{\phi_2(1-\phi_2)+\phi_1^2}{1-\phi_2} = \frac{-0.4\times(1+0.4)+0.49}{1+0.4} = -0.05; \phi_{22} = \frac{\rho_2-\rho_1^2}{1-\rho_1^2} = \frac{-0.3}{1-0.25} = -0.4; \phi_{kk} = 0, k > 2$。

5.2.4 (d)

与题 (a) 表述一致。

5.2.5 (e)

```
1 draw_pacf = function(acf_series, type = "样本偏自相关", n=36) {
2   plot(acf_series, type = 'h', xlab = '延迟' ,ylab = type)
3   abline(h = 0)
4   abline(h = c(2*sqrt(1/n),-2*sqrt(1/n)),lty=2)
5 }
6
```

```r
# (e)
pacf=pacf_func(series)
draw_pacf(pacf,n=72)
```

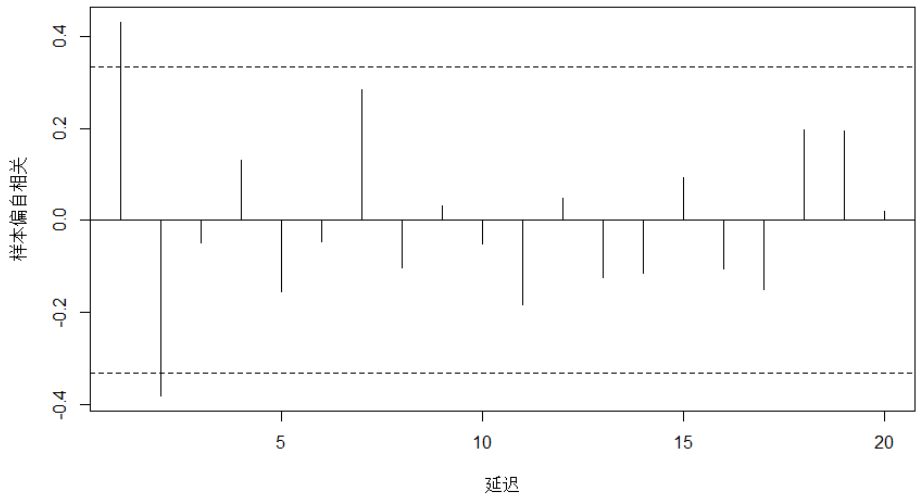

图 6: 样本偏自相关函数图

通过观察样本自相关函数图和理论自相关函数的计算结果，可以发现二者很不匹配。但是在一阶和二阶滞后的情况下，二者的取值相对比较接近。同时由于 ϕ_{kk} 在大样本总体下服从一个均值为 0，方差为 $\frac{1}{n}$ 的正态分布，所以结合图中绘制的 $x = \pm\frac{2}{\sqrt{n}}$ 线，发现一阶后所有的结果都落在其中，所以可以认为此处偏自相关函数对于模型识别依旧是有效的。

6 问题 6.28

6.1 问题重述

模拟一个长度 $n = 36, \theta_1 = 0.7, \theta_2 = -0.4$ 的 MA(2) 时间序列.

(a) 该模型的理论自相关函数是什么?

(b) 计算并画出模拟序列的样本 ACF 样本值和模式与 (a) 部分的理论 ACF 匹配度如何?

(c) 画出模型的理论偏自相关函数，画出足够多的滞后项，直到相关性可以忽略不计.(我们没有计算这个模型的 PACF 的公式相反，我们对这个模型进行一个非常大的样本模拟，例如 n=1000，计算并画出这个模拟的样本 PACF)

(d) 计算并画出 (a) 部分模序列的样本 PACF 本值和模式与 (c) 部分的 “理论” PACF 匹配程度如何?

6.2 问题求解

6.2.1 (a)

对于 MA(2) 模型而言，理论自相关函数可以用

$$\rho_1 = \frac{-\theta_1 + \theta_1\theta_2}{1 + \theta_1^2 + \theta_2^2} \tag{2}$$

$$\rho_2 = \frac{-\theta_2}{1 + \theta_1^2 + \theta_2^2} \tag{3}$$

$$\rho_k = 0 \quad k = 3, 4, \ldots \tag{4}$$

计算得到 $\rho_1 = -0.59; \rho_2 = 0.24; \rho_k = 0, k > 2;$

代码:

```
MA_thm_acf = function(MA_params) {
  q = length(MA_params)
  rho0 = 1 + (MA_params %*% MA_params)
  rho = rep(0, (q + 1))
  if (q > 1) {
    for (k in 1:(q - 1)) {
      rho[k] = (-MA_params[k] + (MA_params[1:(q - k)] %*%
                                   MA_params[(k + 1):q])) / rho0
    }
  }
  rho[q] = (-MA_params[q]) / rho0
  return(rho)
}
```

```r

source("acf_func.R")
source("sm_series.R")
set.seed(666)
series = ARMA_func(NULL,c(0.7,-0.4), 72)
# (a)
(thm_acf = MA_thm_acf(c(0.7,-0.4)))
```

6.2.2 (b)

```r
# (b)
(acf = acf_func(series))
draw_acf(acf[1:20],type="样本自相关")
cor(acf[1:20],thm_acf)
```

输出:

```
 [1] -0.705366214  0.434511679 -0.256900988  0.134973416  0.095066611
 [6] -0.247339518  0.260658916 -0.195795221  0.128579296 -0.034412918
[11] -0.091872562  0.085215048 -0.062662372  0.029236941 -0.082725388
[16]  0.073777788  0.006954973 -0.066477269  0.021474883  0.069287533
[21] -0.020921898 -0.025565517 -0.008672089  0.011799944  0.097006862
[26] -0.179807038  0.233894684 -0.265201299  0.321185980 -0.247260740
[31]  0.147685192 -0.106774794  0.066805942  0.023976241 -0.084033086
[36]  0.059097405 -0.022009153 -0.030302538  0.012300572 -0.036801103
[41]  0.028836522 -0.012472074 -0.073697894  0.093505187 -0.100719703
[46]  0.073744736 -0.041773355 -0.008755313  0.085791096 -0.108547762
[51]  0.107547212 -0.107948143  0.062418135  0.039482172 -0.114481742
[56]  0.147911134 -0.174479865  0.166437547 -0.136358563  0.076225171
[61] -0.034880029 -0.026292098  0.063719498 -0.079527127  0.067292181
[66] -0.057282158  0.036718882 -0.026463508  0.021811025 -0.013215599
[71]  0.002866234
```

通过观察理论自相关函数图，可以发现两者差距较大。在一阶的情况下，数值相对接近，二阶数值相差较大。在二阶后，相差较大，理论自相关函数应该为 0，然而样本自相关函数不为 0，甚至不符合假设检验为 0 的要求（例如三阶、六阶滞后的情况）。

6.2.3 (c)

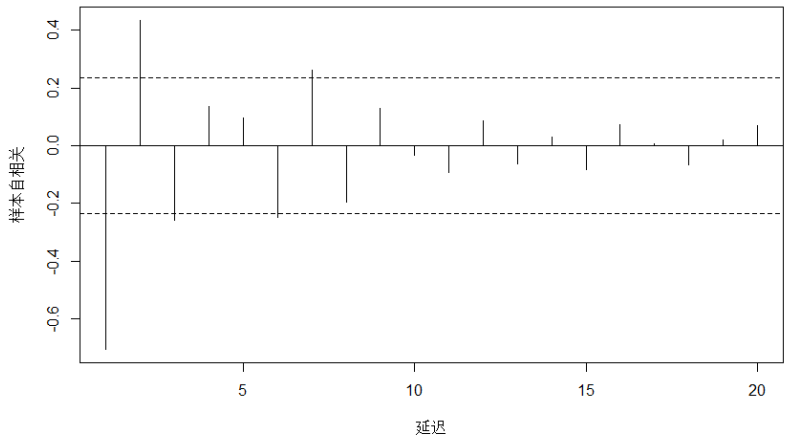

图 7: 样本自相关函数图

```
series = ARMA_func(NULL,c(0.7,-0.4), 1000)
pacf=pacf_func(series)
draw_pacf(pacf,n=1000)
```

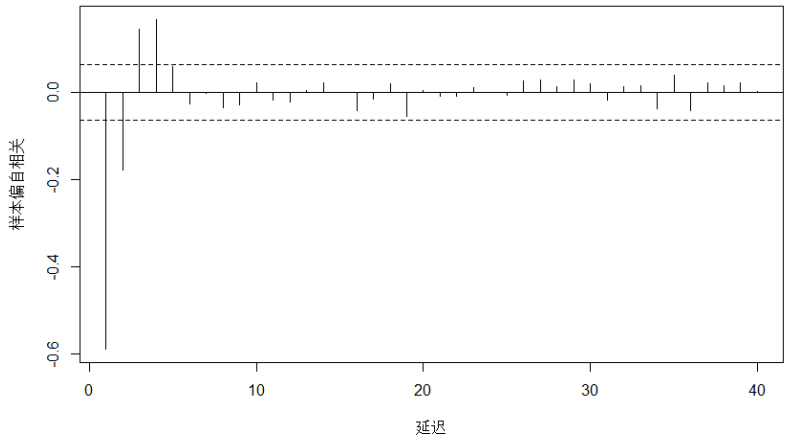

图 8: ” 理论 “偏自相关函数图

以上的结果是利用样本数量为 1000 时的结果，此时将样本的偏自相关函数视作” 理论 “自相关函数。

6.2.4 (d)

```
# (d)
pacf=pacf_func(series)
draw_pacf(pacf,n=72)
```

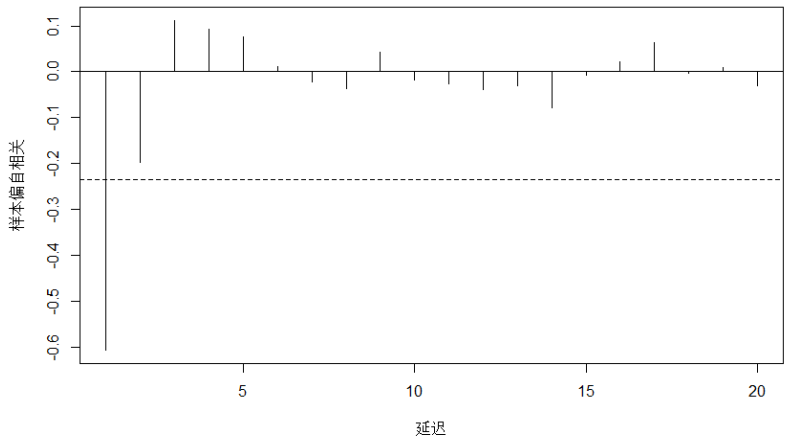

图 9: 样本偏自相关函数图

通过观察样本偏自相关函数图和“理论”偏自相关函数的计算结果，可以发现二者在自相关滞后五阶前都相对比较接近。在五阶后，虽然取值不太匹配，但是偏自相关函数图和“理论”偏自相关函数都可以近似视作 0，所以可以认为二者的匹配程度较好。

7 问题 6.33

7.1 问题重述

名为 deerel 的数据文件包含了 82 个连续的值 (以 0.000 025 英寸为单位)，这些值是在 33 某些指定操作条件下 Deere 公司的机床产生的相对某个设定目标的偏离程度

(a) 展示该序列的时间序列图，并对任何异常点进行评论.

(b) 计算序列的样本 ACF，并对结果进行评论.

(c) 现在用更典型的值代替异常值，然后重新计算样本 ACF. 对与 (b) 部分不同的结果进行评论.

(d) 基于 (c) 部分使用过的修改后的序列计算样本 PACF. 对修正的序列，你将设定什么样的模型? (稍后我们将研究在时间序列建模中处理异常值的其他方法)

7.2 问题求解

7.2.1 (a)

```r
library(TSA)

data("deere1")
# (a)
plot.ts(deere1,type='o')
```

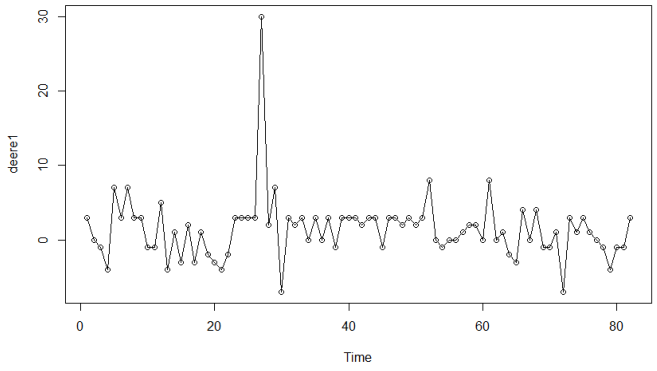

图 10: 时间序列图

从图中可以明显看出，有一个点的取值相比其他点明显更大，因此认为这个是一个异常点，应该予以删除或者替换。

7.2.2 (b)

```r
# (b)
source("acf_func.rR")
(acf=acf_func(deere1))
draw_acf(acf[1:20],type="样本自相关",n=length(acf))
```

结果:

```
 [1]  3.717143e-02  2.441691e-01 -1.670165e-01  4.393057e-02 -4.772897e-02
 [6] -7.542327e-02 -3.959465e-02 -3.085925e-02  3.822371e-02 -6.940739e-02
[11] -3.722807e-02 -9.779756e-02 -1.947289e-02 -1.101530e-01  1.226425e-01
[16] -3.421454e-02  7.098216e-02 -4.614341e-02  8.102801e-02  8.437637e-02
[21]  2.824606e-02  7.325319e-02 -9.240319e-02  1.127692e-02  4.111818e-02
[26]  2.081725e-02 -6.957847e-02 -4.702000e-02 -9.916927e-02 -4.210843e-02
[31] -9.291488e-02  1.789003e-02 -6.173159e-02  1.767826e-01 -1.294956e-02
[36] -2.186029e-02 -8.825916e-02 -1.056375e-01  1.774246e-02 -3.505452e-02
[41]  2.911532e-02  2.605864e-03 -3.140754e-02 -2.311871e-06 -1.216619e-01
[46]  4.226872e-04  3.170732e-03 -2.657727e-02  1.697646e-02  6.659731e-03
[51] -6.895542e-03 -6.417909e-02 -4.662852e-02 -1.004161e-02  2.741417e-02
[56]  2.186838e-02  8.044542e-03 -9.767657e-04  6.513621e-02  5.704158e-03
[61]  6.625631e-02  1.077371e-02  3.111008e-03 -9.622780e-03 -2.376257e-02
[66]  2.961739e-02 -2.101530e-02  5.373907e-02  1.273841e-03  9.645898e-03
[71] -3.343544e-02 -3.364158e-02 -2.015104e-02 -1.648673e-02  1.815705e-02
[76]  1.058914e-02  2.046199e-02 -5.131199e-03 -2.746503e-03 -3.932108e-03
[81]  1.627943e-03
```

绘制延迟为 20 阶时的样本自相关函数结果如下:

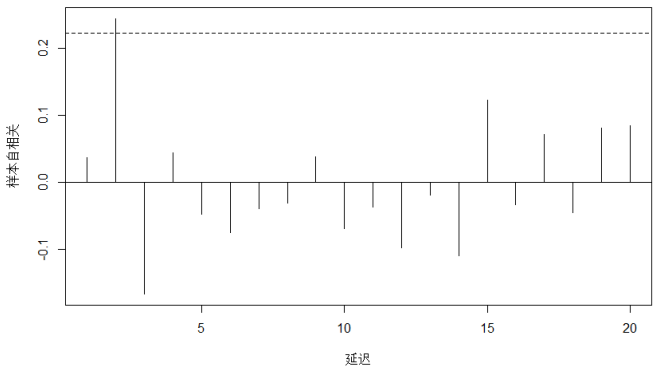

图 11: 样本自相关函数

通过观察样本自相关函数图，可以发现模型在二阶滞后的情况下，展现出一定的相关性。其余阶数的延迟下，自相关性均不明显。

7.2.3 (c)

使用异常值点两侧两个点的均值进行插值替换，并绘制替换后的时间序列图。

```
# (c)
ts_replace=deere1
ts_replace[27]=(ts_replace[26]+ts_replace[28])/2
plot.ts(ts_replace,type='o')
(acf=acf_func(deere1))
draw_acf(acf[1:20],type="样本自相关",n=length(acf))
```

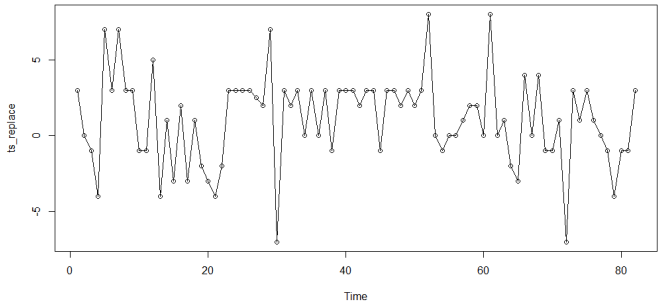

图 12: 替换后时间序列图

可以发现，替换后的时间序列图，相比替换前，可以有效去除异常值点。然后绘制自相关函数图。

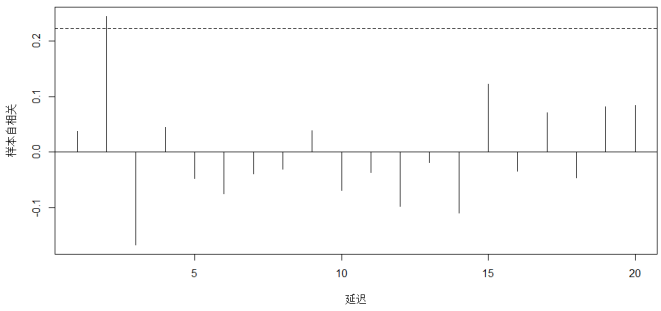

图 13: 替换后样本自相关函数

对比替换前后的样本自相关函数，发现虽然去除了异常值点，但是对于自相关函数的结果影响

较小。自相关函数仍然展现出在二阶滞后的情况下，有一定的相关性。

7.2.4 (d)

```r
# (d)
pacf=pacf_func(ts_replace)
draw_pacf(pacf,n=length(ts_replace))
```

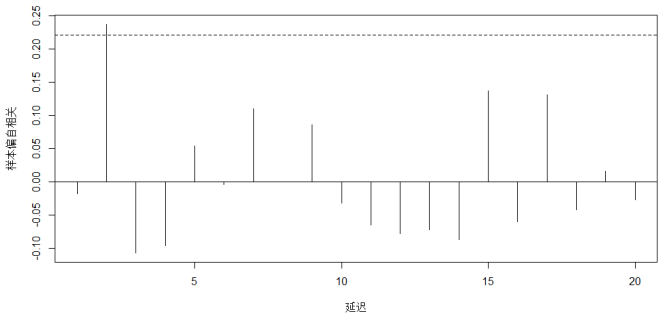

图 14: 替换后样本偏自相关函数

观察替换后样本的偏自相关函数可以发现，偏自相关函数在二阶滞后的情况下展现出一定的显著性。在大于 2 的情况下，则不显著（假设检验没法拒绝偏自相关为 0 的假设）。所以可以考虑使用 AR(2) 模型来建模。