第四次作业

第 6 章　模式识别

6.14 对一个长度为 169 的序列，我们求得 $r_1 = 0.41, r_2 = 0.32, r_3 = 0.26, r_4 = 0.21, r_5 = 0.16$。什么样的 ARIMA 模型适合这种自相关模式?

解：

$\frac{2}{\sqrt{169}} = \frac{2}{13} = 0.153846153846154 < 0.41 = r_1$，拒绝 MA(1)。

$\frac{2}{\sqrt{169}}\sqrt{1+0.41^2} = 0.166274884688297 < 0.32 = r_2$，拒绝 MA(2)。

$\frac{2}{\sqrt{169}}\sqrt{1+0.41^2+0.32^2} = 0.173409936039316 < 0.26 = r_3$，拒绝 MA(3)。

$\frac{2}{\sqrt{169}}\sqrt{1+0.41^2+0.32^2+0.26^2} = 0.177963496024212 < 0.21 = r_4$，拒绝 MA(4)。

$\frac{2}{\sqrt{169}}\sqrt{1+0.41^2+0.32^2+0.26^2+0.21^2} = 0.180872311035737 > 0.16 = r_4$，接受 MA(5)。

$\hat{\phi_{11}} = r_1 = 0.41 > \frac{2}{13}$，拒绝 AR(1)。

$\hat{\phi_{22}} = \frac{r_2 - r_1^2}{1 - r_1^2} = 0.182594061786272 > \frac{2}{13}$，拒绝 AR(2)。

$\hat{\phi_{21}} = \hat{\phi_{11}} - \hat{\phi_{22}}\hat{\phi_{11}} = 0.335136434667628$

$\hat{\phi_{33}} = \frac{r_3 - \hat{\phi_{21}}r_2 - \hat{\phi_{22}}r_1}{1 - \hat{\phi_{21}}r_1 - \hat{\phi_{22}}r_2} = 0.0968618083541609 < \frac{2}{13}$，接受 AR(3)。

所以可以使用 ARIMA(3,0,0) 或者 ARIMA(0,0,5)。 □

6.15 某序列及其一阶差分序列的样本 ACF 列于下表，此处 $n = 100$。

滞后	1	2	3	4	5	6
Y_t 的 ACF	0.97	0.97	0.93	0.85	0.80	0.71
∇Y_t 的 ACF	-0.42	0.18	-0.02	0.07	-0.10	-0.09

只基于这些信息，我们会为该序列考虑什么样的 ARIMA 模型?

解：

未差分时，ACF 不截尾，一阶差分后 ACF 快速趋于 0。

$\frac{2}{\sqrt{100}} = 0.2 < |-0.42| = |r_1|$，拒绝 MA(1)。

$\frac{2}{\sqrt{100}}\sqrt{1+0.42^2} = 0.2169239498073 > 0.18$，接受 MA(2)。

$\left|\hat{\phi_{11}}\right| = 0.42 > 0.2$，拒绝 AR(1)。

$r_1 = -0.42, r_2 = 0.18$

$\hat{\phi_{22}} = \frac{r_2 - r_1^2}{1 - r_1^2} = 0.00437105390966489 < 0.2$，接受 AR(2)。

所以可以考虑 ARIMA(2,1,0) 或者 ARIMA(0,1,2)。 □

6.16 对一个长度为 64 的序列，样本偏自相关函数如下：

滞后	1	2	3	4	5
PACF	0.47	-0.34	0.20	0.02	-0.06

这种情况下，我们应该考虑什么样的模型?

解：

$\frac{2}{\sqrt{64}} = \frac{2}{8} = 0.25$

$\left|\hat{\phi_{11}}\right| = 0.47 > 0.25$，拒绝 AR(1)。

$\left|\hat{\phi_{22}}\right| = 0.34 > 0.25$，拒绝 AR(2)。

$\left|\hat{\phi_{33}}\right| = 0.20 < 0.25$，接受 AR(3)。

所以应该考虑 AR(3)。 □

6.20 模拟 $n = 48, \phi = 0.7$ 的 AR(1) 时间序列。

(a) 计算该模型在一阶和 5 阶滞后处的理论自相关系数。

```
phi <- 0.7          # AR(1) 系数
n <- 48             # 样本量
n_sim_c <- 10       # (c) 项模拟次数
n_sim_d <- 1000     # (d) 项模拟次数

rho_k <- function(k) -theta / (1+theta^2)
cat("(a) 理论自相关系数:\n",
    "rho1 =", rho_k(1), "\n",
    "rho5 =", rho_k(5), "\n\n")
```

```
(a) 理论自相关系数:
rho1 = 0.7
rho5 = 0.16807
```

(b) 计算 1 阶和 5 阶滞后处样本自相关系数，并将其与理论自相关值进行比较。用方程 (6.1.5) 和 (6.1.6) 量化这个比较。

```
run_ar_simulation <- function (n) {
  ar_sim <- arima.sim(model = list(ar = phi), n = n)
  acf_vals <- acf(ar_sim, plot = FALSE, lag.max = 5)$acf
```

```
  results_b <- data.frame(
    Lag = c(1, 5),
    Theoretical = c(rho_k(1), rho_k(5)),
    Sample = c(acf_vals[1], acf_vals[5])
  )
  results_b$Difference <- results_b$Sample - results_b$Theoretical
  results_b$Standard_Deviation <- c(sqrt((1-phi ^ 2) / n), sqrt(1 / n * (1 + phi ^
  ↪  2) / (1 - phi ^ 2)))

  return(results_b)
}

cat("(b) 样本与理论值比较:\n")
add_a_table(run_ar_simulation(n))
```

	Lag	Theoretical	Sample	Difference	Standard_Deviation
1	1	0.7	0.607000115383271	-0.0929998846167286	0.103077640640442
2	5	0.16807	-0.138049906031594	-0.306119906031594	0.246710382983561

(c) 使用新的模拟重复 (b)。描述在相同条件下，估计的精度如何随所选样本的不同而变化。

```
combined <- map_df(1:n_sim_c, ~ {
  run_ar_simulation(n = .x * 100) %>%
          mutate(simulation_times = as.integer(.x * 100))  # .x 表示当前迭代次数
}, .progress = TRUE)    # 显示进度条
add_a_table(combined)
```

	Lag	Theoretical	Sample	Difference	Standard_Deviation	simulation_times
1	1	0.7	0.654391247411627	-0.0456087525883728	0.0714142842854285	100
2	5	0.16807	0.160702948570478	-0.00736705142952221	0.170925967232922	100
3	1	0.7	0.716362198819802	0.0163621988198021	0.0504975246918104	200
4	5	0.16807	0.271219815625015	0.103149815625015	0.120862910511269	200
5	1	0.7	0.761059565720873	0.0610595657208732	0.0412310562561766	300
6	5	0.16807	0.280424703845057	0.112354703845057	0.0986841531934245	300
7	1	0.7	0.659422469660548	-0.0405775303394521	0.0357071421427143	400
8	5	0.16807	0.110825779980553	-0.0572442200194473	0.0854629836164608	400
9	1	0.7	0.660486483669117	-0.039513516330883	0.0319374388453426	500
10	5	0.16807	0.209316225867902	0.0412462258679022	0.0764404163705429	500
11	1	0.7	0.697133206959067	-0.00286679304093329	0.0291547594742265	600
12	5	0.16807	0.22167898231985	0.0536089823198499	0.0697802339187225	600
13	1	0.7	0.681254697451475	-0.0187453025485252	0.0269920623252731	700
14	5	0.16807	0.157081312526352	-0.0109886874736478	0.0646039431287837	700
15	1	0.7	0.678736060212811	-0.0212639397871894	0.0252487623459052	800
16	5	0.16807	0.129038261687166	-0.0390317383128335	0.0604314552556343	800
17	1	0.7	0.705853695625197	0.005853695625197	0.0238047614284762	900
18	5	0.16807	0.11303902708582	-0.05503097291418	0.0569753224109739	900
19	1	0.7	0.715593812102739	0.0155938121027386	0.0225831795812724	1000
20	5	0.16807	0.231794430788759	0.0637244307887589	0.0540515367723341	1000

从表格中可以看出，样本数量越大，估计的精度越高。

(d) 如果软件允许，重复模拟序列并多次计算 r_1 和 r_5，并且构建 r_1 和 r_5 的样本分布。描述估计的精度如何随着相同条件下所选择的样本的不同而变化。方程 (6.1.5) 给出的大样本方差与你的样本分布的方差接近程度如何?

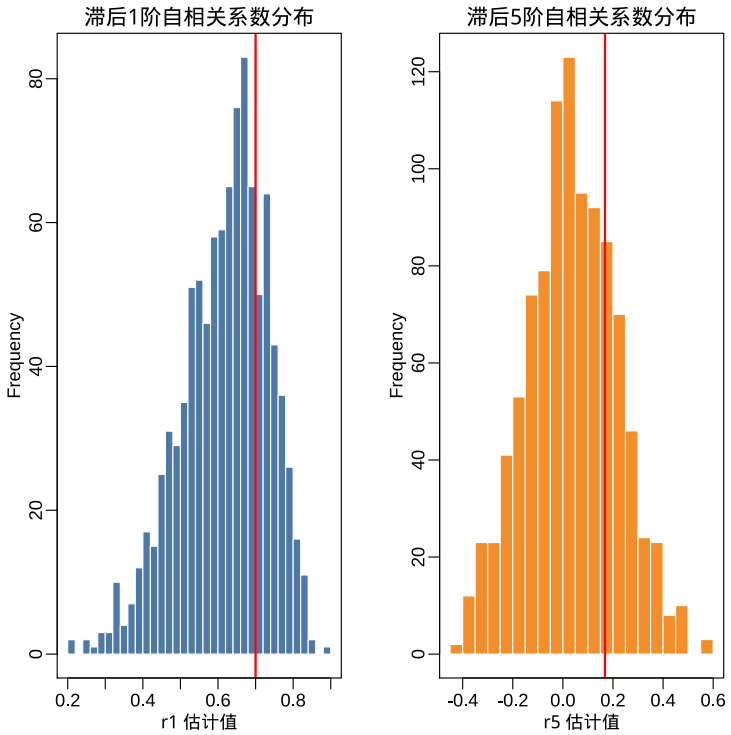
```r
# 高效模拟函数
simulate_ar_acf <- function() {
  sim_data <- arima.sim(model = list(ar = phi), n = n)
  acf_vals <- acf(sim_data, plot = FALSE, lag.max = 5)$acf
  c(r1 = acf_vals[1], r5 = acf_vals[5])  # 直接返回 r1 和 r5
}

# 执行大规模模拟
big_sim <- replicate(n_sim_d, simulate_ar_acf())

# 可视化分布
par(mfrow = c(1, 2), mar = c(4, 3, 1.5, 1.5), mgp=c(1.5, 0.5, 0))

# r1 分布直方图
h1 <- hist(big_sim["r1", ], breaks = 30, col = "#4E79A7",
main = iconv(" 滞后 1 阶自相关系数分布", to="gbk"),
xlab = iconv("r1 估计值", to="gbk"), border = "white")
abline(v = rho_k(1), col = "red", lwd = 2)
box()
```

```r
# r5 分布直方图
h2 <- hist(big_sim["r5", ], breaks = 30, col = "#F28E2B",
main = iconv(" 滞后 5 阶自相关系数分布", to="gbk"),
xlab = iconv("r5 估计值", to="gbk"), border = "white")
abline(v = rho_k(5), col = "red", lwd = 2)
box()
```



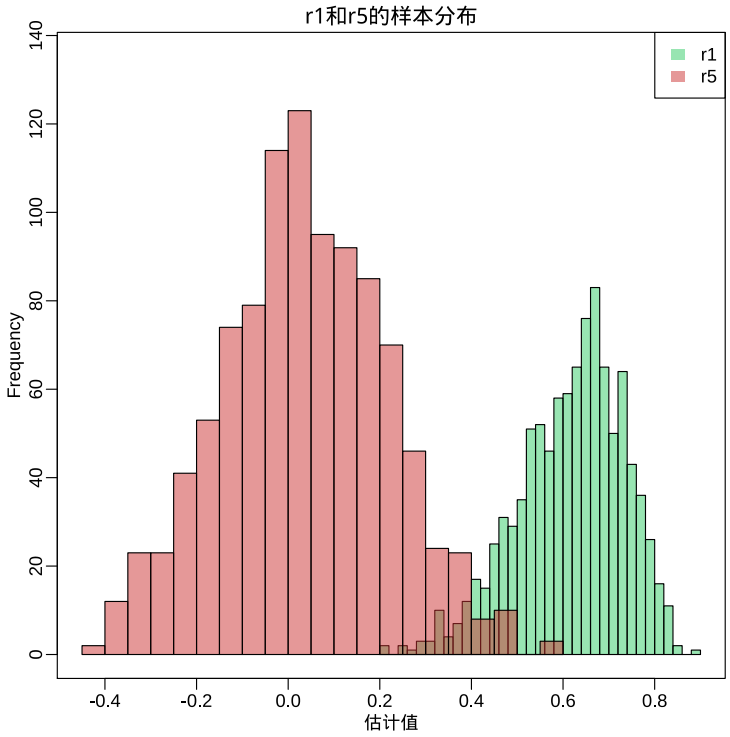
两个直方图叠在一起对比一下：

```r
# 计算纵轴最大值
y_max <- max(h1$counts, h2$counts)

par(mfrow = c(1, 1), mar = c(4, 3, 1.5, 1.5), mgp=c(1.5, 0.5, 0))

# 绘制叠加直方图
plot(h1, col = rgb(0.2, 0.8, 0.4, 0.5),  # 半透明绿色
     xlim = range(h1$breaks, h2$breaks),
     ylim = c(0, y_max * 1.1),
     main = iconv("r1 和 r5 的样本分布", to="gbk"),
     xlab = iconv(" 估计值", to="gbk"))
plot(h2, col = rgb(0.8, 0.2, 0.2, 0.5),  # 半透明红色
     add = TRUE)  # 关键参数：叠加图形

# 添加图例
legend("topright",
       legend = c(iconv("r1", to="gbk"), iconv("r5", to="gbk")),
       fill = c(rgb(0.2,0.8,0.4,0.5), rgb(0.8,0.2,0.2,0.5)),
       border = NA)
box()
```



样本数量越大，估计的精度越高。从 **(c)** 的表格中可以看到，方程 (6.1.5) 给出的大样本方差与模拟的样本分布的方差还是比较接近的。

6.21 模拟 $n = 60, \theta = 0.5$ 的 MA(1) 时间序列。

(a) 计算该模型 1 阶滞后处的理论自相关系数。

```
theta <- 0.5       # MA(1) 系数
n <- 60            # 样本量
n_sim_c <- 10      # (c) 项模拟次数
n_sim_d <- 1000    # (d) 项模拟次数

rho_k <- function(k) -theta / (1+theta^2)
cat("(a) 理论自相关系数:\n",
    "rho1 =", rho_k(1), "\n")
```

```
(a) 理论自相关系数:
rho1 = -0.4
```

(b) 计算 1 阶滞后处样本自相关系数，并将其与理论自相关系数进行比较。用图表 6-2 量化这个比较。

```
run_ma_simulation <- function (n) {
  ar_sim <- arima.sim(model = list(ma = -theta), n = n) # 这里要加个负号，因为它的模
  ↪  型里的参数好像是不带减号的
  acf_vals <- acf(ar_sim, plot = FALSE, lag.max = 1)$acf
```

```
  results_b <- data.frame(
          Lag = c(1),
          Theoretical = c(rho_k(1)),
          Sample = c(acf_vals[1])
  )
  results_b$Difference <- results_b$Sample - results_b$Theoretical
  results_b$Standard_Deviation <- c(sqrt(1 / n * (1 + 2 * rho_k(1) ^ 2)))

  return(results_b)
}

cat("(b) 样本与理论值比较:\n")
add_a_table(run_ma_simulation(n))
```

	Lag	Theoretical	Sample	Difference	Standard_Deviation
1	1	-0.4	-0.397564576710337	0.00243542328966284	0.148323969741913

(c) 使用新的模拟重复 (b)。描述在相同条件下，估计的精度如何随所选样本的不同而变化。

```
combined <- map_df(1:n_sim_c, ~ {
  run_ma_simulation(n = .x * 100) %>%
          mutate(simulation_times = as.integer(.x * 100))  # .x 表示当前迭代次数
}, .progress = TRUE)    # 显示进度条
add_a_table(combined)
```

	Lag	Theoretical	Sample	Difference	Standard_Deviation	simulation_times
1	1	-0.4	-0.131413989248538	0.268586010751462	0.114891252930761	100
2	1	-0.4	-0.494148738730896	-0.0941487387308959	0.0812403840463596	200
3	1	-0.4	-0.416051926120893	-0.016051926120893	0.066332495807108	300
4	1	-0.4	-0.488706457873169	-0.0887064578731688	0.0574456264653803	400
5	1	-0.4	-0.334411518503811	0.0655884814961888	0.0513809303146605	500
6	1	-0.4	-0.393862539692495	0.00613746030750539	0.0469041575982343	600
7	1	-0.4	-0.384708914802163	0.0152910851978375	0.0434248118673448	700
8	1	-0.4	-0.369246694567918	0.0307533054320819	0.0406201920231798	800
9	1	-0.4	-0.42953456464491	-0.0295345646449095	0.0382970843102535	900
10	1	-0.4	-0.414798647897431	-0.0147986478974315	0.0363318042491699	1000

从表格中可以看出，样本数量越大，估计的精度越高。

(d) 如果软件允许，重复模拟序列并多次计算 r_1，并且构建 r_1 的样本分布。描述估计的精度如何随着相同条件下所选择的样本的不同而变化。图表 6-2 给出的大样本方差与你的样本分布的方差接近程度如何?

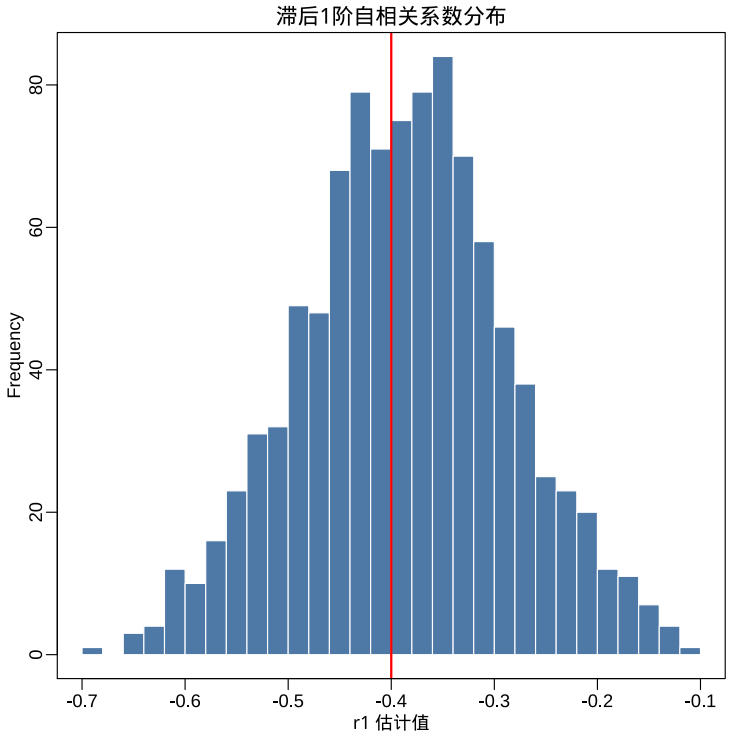
```
simulate_ma_acf <- function () {
  ar_sim <- arima.sim(model = list(ma = -theta), n = n) # 这里要加个负号，因为它的模
  ↪  型里的参数好像是不带减号的
```

```r
  acf_vals <- acf(ar_sim, plot = FALSE, lag.max = 1)$acf
  acf_vals[1]
}

big_sim <- replicate(n_sim_d, simulate_ma_acf())

# 可视化分布
par(mfrow = c(1, 1), mar = c(4, 3, 1.5, 1.5), mgp=c(1.5, 0.5, 0))

# r1 分布直方图
h1 <- hist(big_sim, breaks = 30, col = "#4E79A7",
           main = iconv(" 滞后 1 阶自相关系数分布", to="gbk"),
           xlab = iconv("r1 估计值", to="gbk"), border = "white")
abline(v = rho_k(1), col = "red", lwd = 2)
box()
```



样本数量越大，估计的精度越高。从 (c) 的表格中可以看到，图表 6-2 给出的大样本方差与模拟的样本分布的方差还是比较接近的。