第三次作业

第 5 章　非平稳时间序列模型

5.7 考虑两个模型:

A: $Y_t = 0.9Y_{t-1} + 0.09Y_{t-2} + e_t$

B: $Y_t = Y_{t-1} + e_t - 0.1e_{t-1}$

(a) 识别每个模型具体的 ARIMA 形式，即 p, d, q 分别是多少，参数值 ϕ 和 θ 是什么?

解：

A: $(1 - 0.9B - 0.09B^2)Y_t = e_t$，即 ARIMA(2, 0, 0)，$\phi_1 = 0.9, \phi_2 = 0.09$, θ 没有。

B: $(1 - B)Y_t = (1 - 0.1B)e_t$，即 ARIMA(0, 1, 1)，$\phi$ 没有，$\theta_1 = 0.1$。 □

(b) 什么情况下两个模型会不同?

在任何情况下两个模型都不同。

(c) 什么情况下两个模型会相似?（比较 ψ 权重和 π 权重）

首先转成一般线性过程计算 ψ 权重，这里使用大模型给出的代码来计算:

```
# 扩展的 ARIMA 转 Wold 系数函数
arima.to.Wold <- function(n, phi, theta, d) {
  # 计算差分算子的逆多项式系数
  diff_coef <- (-1)^(0:d) * choose(d, 0:d)

  # 计算 ARMA 部分的 Wold 系数
  p <- length(phi)
  q <- length(theta)
  psi <- numeric(n)
  psi[1] <- 1

  for(j in 1:(n-1)) {
    ar_part <- if(j <= p) sum(phi[1:j] * psi[j:1]) else sum(phi * psi[j:(j-p+1)])
    ma_part <- if(j <= q) theta[j] else 0
```

```r
    psi[j+1] <- ar_part + ma_part
  }

  # 卷积运算整合差分逆运算
  final_coef <- convolve(psi, rev(diff_coef), type = "open")[1:n]
  return(final_coef)
}

# 示例: ARIMA 模型
phi <- c(0.9, 0.09)
theta <- c(0)
d <- 0
wold_coef <- arima.to.Wold(n=10, phi, theta, d)
cat("A Wold coefficients:\n")
print(round(wold_coef, 4))

# 示例: ARIMA 模型
phi <- c(0)
theta <- c(0.1)
d <- 1
wold_coef <- arima.to.Wold(n=10, phi, theta, d)
cat("B Wold coefficients:\n")
print(round(wold_coef, 4))
```

```
A Wold coefficients:
 [1] 1.0000 0.9000 0.9000 0.8910 0.8829 0.8748 0.8668 0.8588 0.8510 0.8432
B Wold coefficients:
 [1]  1.0 -0.9 -0.1  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

可以看到两个模型的一般线性过程的系数差距很大，所以在任何情况下两个模型都不会相似。

5.10 非平稳 ARIMA 序列可以由以下方法模拟: 首先模拟相应的平稳 ARMA 系列，然后对其 “求和” (实际上部分求和)。使用统计软件来模拟具有不同参数值的各种 IMA 的 (1, 1) 和 IMA(2, 2) 序列。注意在这些模拟序列中出现的任何随机 “趋势”。

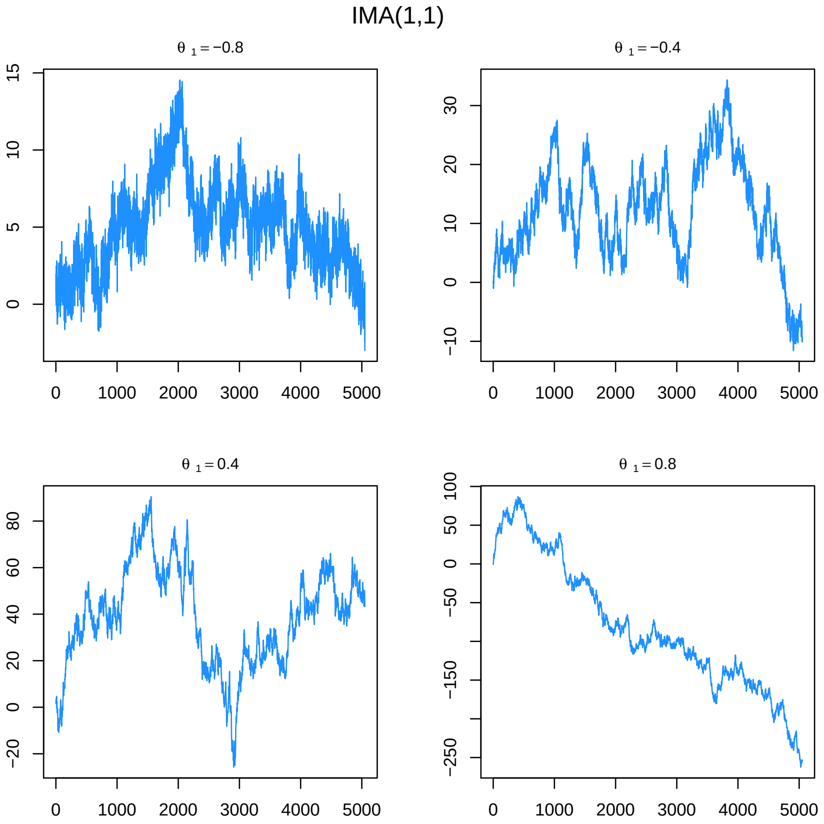
这里使用大模型给出的代码。

```r
# ======================
# IMA(1,1) 多参数模拟 ( 1 不同取值)
# ======================
pdf("1.pdf")
par(mfrow = c(2, 2), mar = c(4, 4, 2, 1), oma = c(0, 0, 2, 0))  # 2x2 子图布局
theta1_values <- c(-0.8, -0.4, 0.4, 0.8)  # 定义四个 MA(1) 参数

set.seed(123)
for(theta in theta1_values) {
  # 生成 MA(1) 序列并一阶求和
```

```
  ma_series <- arima.sim(model = list(ma = theta), n = 5050)
  ima_series <- cumsum(ma_series)

  # 绘制子图
  plot(ima_series, type = "l", col = "#1E90FF",
       main = bquote(theta["  1"] == .(theta)),
       xlab = "", ylab = "", cex.main = 0.9, cex.lab = 0.8)
}
mtext("IMA(1,1)", outer = TRUE, cex = 1.2) # 总标题
dev.off()
```




```

# =====================
# IMA(2,2) 多参数模拟 ( 1 与  2 组合)
# =====================
pdf("2.pdf")
par(mfrow = c(3, 3), mar = c(2, 2, 2, 1))  # 3x3 子图布局
theta_combinations <- list(  # 定义九组 MA(2) 参数
        c(0.8, -0.8), c(0.0, -0.8), c(-0.8, -0.8),
        c(0.8, 0.0), c(0.0, 0.0), c(-0.8, 0.0),
        c(0.8, 0.8), c(0.0, 0.8), c(0.8, 0.8)
)

set.seed(456)
for(i in seq_along(theta_combinations)) {
```

```
  theta <- theta_combinations[[i]]

  # 生成 MA(2) 序列并二阶求和
  ma_series <- arima.sim(model = list(ma = theta), n = 5050)
  ima_series <- cumsum(cumsum(ma_series))  # 双重累积求和

  # 绘制子图
  plot(ima_series, type = "l", col = "#CD5C5C",
       main = bquote(theta["  1"] == .(theta[1]) ~ "," ~ theta["  2"] == .(theta[2])),
       xlab = "", ylab = "", cex.main = 0.8,
       axes = FALSE)
  box()  # 添加边框
  axis(1, cex.axis = 0.7)  # 调整坐标轴字体
  axis(2, cex.axis = 0.7)
}
mtext("IMA(2, 2)", outer = TRUE, cex = 1.2) # 总标题
dev.off()
```