

# 时间序列分析作业与第四次实验报告

姓名：康江睿

学号：2018213779

指导老师： 张晓飞

2020 年 10 月 30 日

## 1 AR(2)模型的自相关函数的计算程序与习题4.9

### 1.1 函数AR2简介

函数AR2实现了对给定参数的AR(2)模型的自相关函数的计算与可视化。除此以外，函数提供了计算模型对应特征方程的根的选项。代码如下：

```
1 AR2 <- function(phi1, phi2, lag, plot = FALSE, root = FALSE){
2
3   library("ggplot2");
4
5   k <- lag;
6   rhos <- integer(k);
7   rhos[1] <- phi1/(1-phi2);rhos[2] <- phi1*rhos[1]+phi2;
8   for (i in 3:k){
9     rhos[i] <- phi1*rhos[i-1]+phi2*rhos[i-2];
10  }
11
12  if (plot == TRUE){
13    data <- data.frame(lags = 1:k, acfval = rhos);
14    p <- ggplot(data, aes(x = lags,y = acfval))+
15      geom_segment(aes(x = lags, xend = lags, y = 0, yend = acfval))+geom_point(size = 3, color = "red")+
16      theme(axis.text.x = element_text(angle = 45, hjust = 0.8, vjust = 0.9))+
17      scale_x_continuous("滞后数", breaks = seq(2,k,2))+theme(legend.position="none")+
18      scale_y_continuous("自相关函数值", breaks = seq(-1,1,0.2))
19  }else{
20    p <- NA;
21  }
22
23  if (root == TRUE){
```

```

24 Delta <- sqrt(complex(real = phi1^2+4*phi2));
25 if (Im(Delta) == 0){
26   roots <- c((phi1-Re(Delta))/(-2*phi2),(phi1+Re(Delta))/(-2*phi2));
27   results <- list(ACFval = rhos, Lag = k, ACFplot = p, Roots = roots);
28 } else{
29   roots <- c((phi1-Delta)/(-2*phi2),(phi1+Delta)/(-2*phi2));
30   R <- sqrt(-phi2);
31   Theta <- acos(phi1/2/R);
32   results <- list(ACFval = rhos, Lag = k, ACFplot = p, Roots = roots, Damping_Factor = R, Frequency
33                   = Theta);
34 } else{
35   results <- list(ACFval = rhos, Lag = k, ACFplot = p);
36 }
37 return(results)
38
39 }

```

## 1.2 函数AR2的使用说明

(1)对于AR(2)模型:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + e_t$$

对应到函数AR2, 则函数的输入参数phi1是模型中 $\phi_1$ 的值, phi2是模型中 $\phi_2$ 的值。

(2)输入参数lag是自相关函数的最大滞后阶数; 如: 对于lag=k的情形, 则函数计算1-k阶滞后的自相关函数值。

(3)输入参数plot是选择是否绘制模型的1-k阶自相关函数图的参数; 要求是布尔型变量。

(4)输入参数root是选择是否计算模型对应特征方程的根的参数; 要求是布尔型变量。如果根为复数, 则函数还会计算模型自相关函数的阻尼因子和频率。

(5)输出参数results为一个列表。具体内容如下:

| 列表results包含的成分 | 含义                                  |
|----------------|-------------------------------------|
| ACFval         | 模型的自相关函数值                           |
| Lag            | 自相关函数的最大滞后阶数                        |
| ACFplot        | 模型的自相关函数图（如果plot=FALSE，则ACFplot=NA） |
| Roots          | 模型对应特征方程的根（仅在root=TRUE时输出）          |
| Damping_Factor | 自相关函数的阻尼因子（仅在特征根为复数时输出）             |
| Frequency      | 自相关函数的频率（仅在特征根为复数时输出）               |

表 1: 列表results包含的成分与对应的含义

### 1.3 使用函数AR2解决习题4.9

在本部分，所有最大滞后阶数Lag统一设置为30。首先将函数AR2载入工作环境。

```
1 > source('D:/R Files/TSACourse/AR2.R', encoding = 'UTF-8')
```

(a) 实现本题的代码如下：

```
1 > results=AR2(0.6,0.3,30,TRUE,TRUE)
2 > results$ACFplot
3 > results$Roots
4 [1] 1.081666 -3.081666
```

可以发现，此时的特征根是实数。绘制的自相关函数图如下：

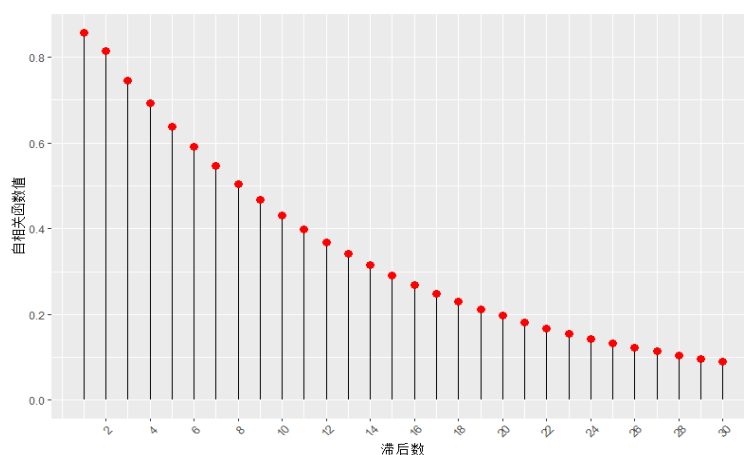


图 1: 习题4.9(a)的1-30阶滞后自相关函数图

(b) 实现本题的代码如下：

```

1 > results=AR2(-0.4,0.5,30,TRUE,TRUE)
2 > results$ACFplot
3 > results$Roots
4 [1] 1.869694 -1.069694

```

可以发现，此时的特征根是实数。绘制的自相关函数图如下：

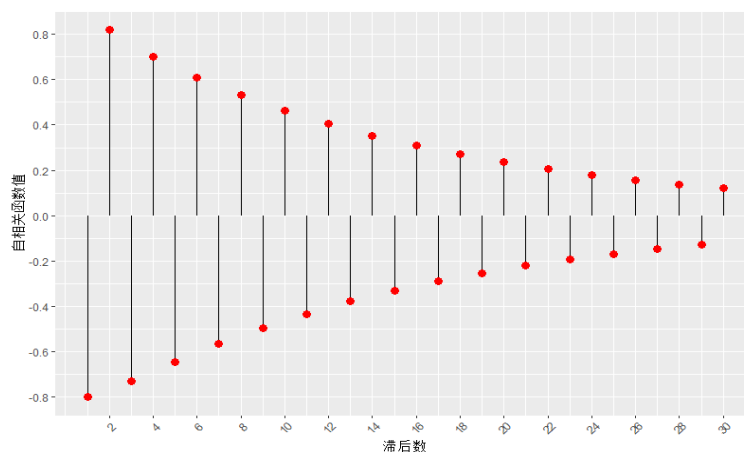


图 2: 习题4.9(b)的1-30阶滞后自相关函数图

(c) 实现本题的代码如下：

```

1 > results=AR2(1.2,-0.7,30,TRUE,TRUE)
2 > results$ACFplot
3 > results$Roots
4 [1] 0.8571429-0.8329931i 0.8571429+0.8329931i

```

可以发现，此时的特征根是复数。

```

1 > results$Damping_Factor
2 [1] 0.83666
3 > results$Frequency
4 [1] 0.7711105

```

如上所示，阻尼因子 $R = 0.8367$ ，频率 $\Theta = 0.7711$ 。

绘制的自相关函数图如下：

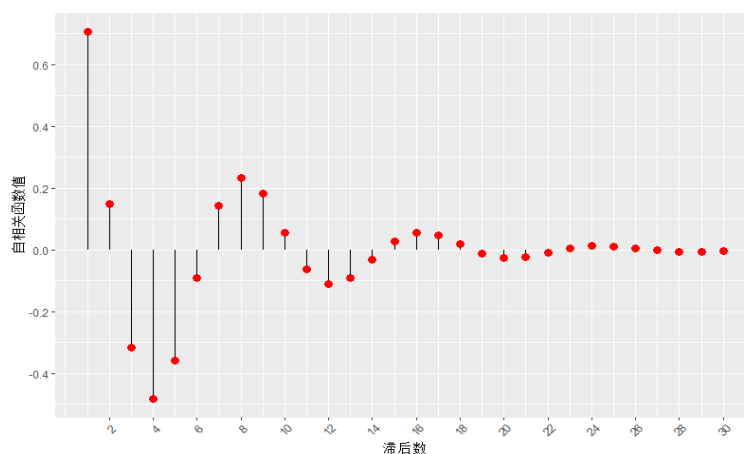


图 3: 习题4.9(c)的1-30阶滞后自相关函数图

(d) 实现本题的代码如下:

```
1 > results=AR2(-1,-0.6,30,TRUE,TRUE)
2 > results$ACFplot
3 > results$Roots
4 [1] -0.8333333-0.9860133i -0.8333333+0.9860133i
```

可以发现，此时的特征根是复数。

```
1 > results$Damping_Factor
2 [1] 0.7745967
3 > results$Frequency
4 [1] 2.27247
```

如上所示，阻尼因子 $R = 0.7746$ ，频率 $\Theta = 2.2725$ 。

绘制的自相关函数图如下：

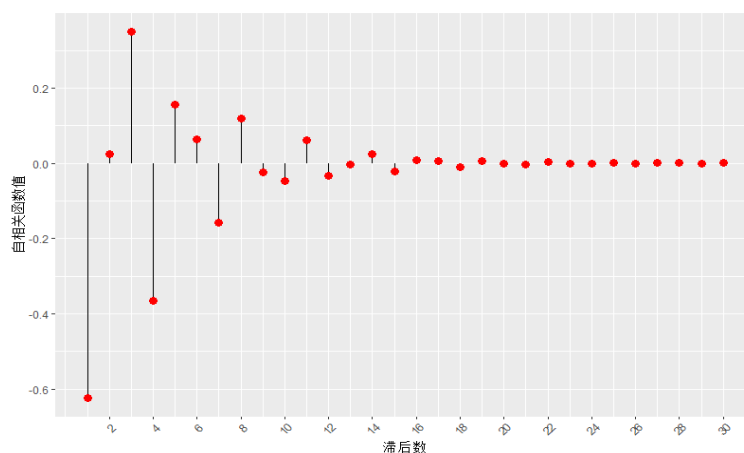


图 4: 习题4.9(d)的1-30阶滞后自相关函数图

(e) 实现本题的代码如下：

```
1 > results=AR2(0.5,-0.9,30,TRUE,TRUE)
2 > results$ACFplot
3 > results$Roots
4 [1] 0.277778-1.016834i 0.277778+1.016834i
```

可以发现，此时的特征根是复数。

```
1 > results$Damping_Factor
2 [1] 0.9486833
3 > results$Frequency
4 [1] 1.304124
```

如上所示，阻尼因子 $R = 0.9487$ ，频率 $\Theta = 1.3041$ 。

绘制的自相关函数图如下：

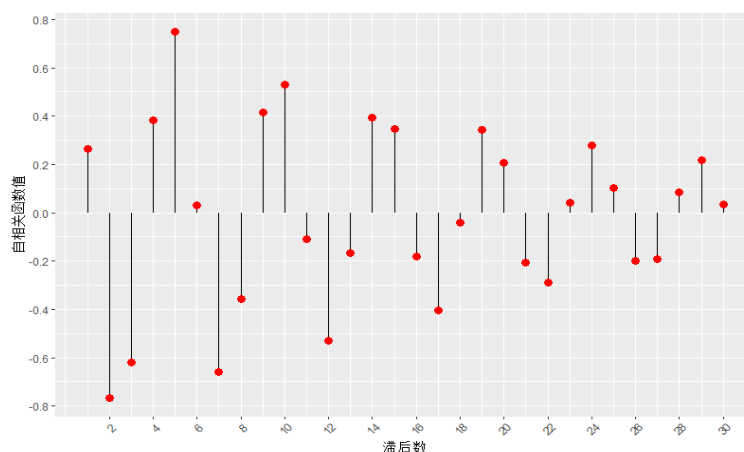


图 5: 习题4.9(e)的1-30阶滞后自相关函数图

(f) 实现本题的代码如下：

```
1 > results=AR2(-0.5,-0.6,30,TRUE,TRUE)
2 > results$ACFplot
3 > results$Roots
4 [1] -0.416667-1.221907i -0.416667+1.221907i
```

可以发现，此时的特征根是复数。

```
1 > results$Damping_Factor
2 [1] 0.7745967
3 > results$Frequency
4 [1] 1.899428
```

如上所示，阻尼因子 $R = 0.7746$ ，频率 $\Theta = 1.8994$ 。

绘制的自相关函数图如下：

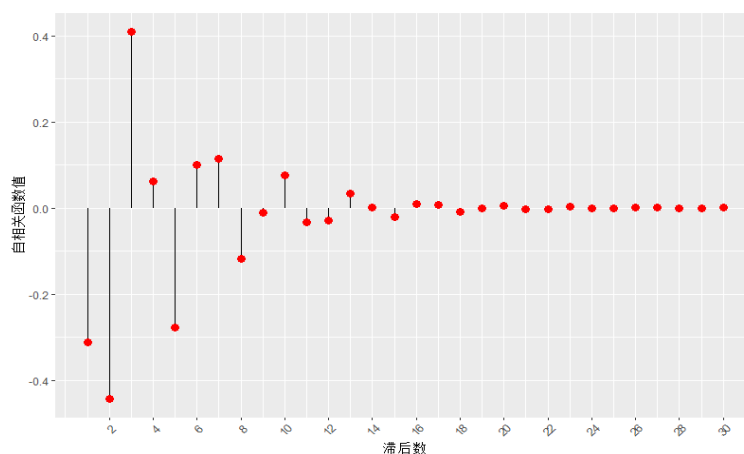


图 6: 习题4.9(f)的1-30阶滞后自相关函数图

## 2 ARMA(1,1)模型的自相关函数的计算程序与习题4.10

### 2.1 函数ARMA11简介

函数ARMA11实现了对给定参数的ARMA(1,1)模型的自相关函数的计算与可视化。代码如下：

```

1 ARMA11 <- function(phi,theta,lag,plot = FALSE){
2   k <- lag;
3   x <- 0:(k-1);
4   rhos <- (1-theta*phi)*(phi-theta)*(phi^x)/(1-2*theta*phi+theta^2);
5
6   if (plot == TRUE){
7     data <- data.frame(lags = 1:k, acfval = rhos);
8     p <- ggplot(data, aes(x = lags,y = acfval))+
9       geom_segment(aes(x = lags, xend = lags, y = 0, yend = acfval))+geom_point(size = 3, color = "red")+
10      theme(axis.text.x = element_text(angle = 45, hjust = 0.8, vjust = 0.9))+
11      scale_x_continuous("滞后数", breaks = seq(2,k,2))+theme(legend.position="none")+
12      scale_y_continuous("自相关函数值", breaks = seq(-1,1,0.2))
13   } else {
14     p = NA;
15   }
16
17   results <- list(ACFval = rhos, Lag = k, ACFplot = p);
18   return(results);
19
20 }
```

## 2.2 函数ARMA11的使用说明

(1)对于ARMA(1,1)模型:

$$Y_t = \phi_1 Y_{t-1} + e_t - \theta$$

对应到函数AR2, 则函数的输入参数phi是模型中 $\phi$ 的值, theta是模型中 $\theta$ 的值。

(2)输入参数lag是自相关函数的最大滞后阶数; 如: 对于lag=k的情形, 则函数计算1-k阶滞后的自相关函数值。

(3)输入参数plot是选择是否绘制模型的1-k阶自相关函数图的参数; 要求是布尔型变量。

(4)输出参数results为一个列表。具体内容如下:

| 列表results包含的成分 | 含义                                    |
|----------------|---------------------------------------|
| ACFval         | 模型的自相关函数值                             |
| Lag            | 自相关函数的最大滞后阶数                          |
| ACFplot        | 模型的自相关函数图 (如果plot=FALSE, 则ACFplot=NA) |

表 2: 列表results包含的成分与对应的含义

## 2.3 使用函数ARMA11解决习题4.10

在本部分, 所有最大滞后阶数Lag统一设置为30。首先将函数ARMA11载入工作环境

```
1 > source('D:/R Files/TSACourse/ARMA11.R', encoding = 'UTF-8')
```

(a) 实现本题的代码如下:

```
1 > results=ARMA11(0.7,0.4,30,TRUE)
2 > results$ACFplot
```

绘制的自相关函数图如下:



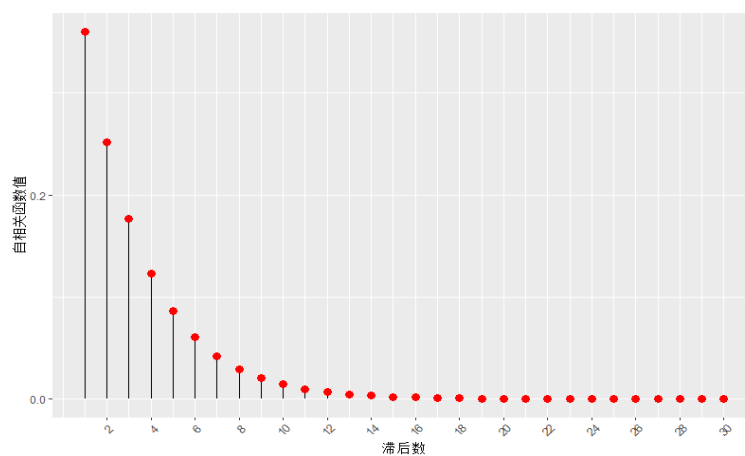


图 7: 习题4.10(a)的1-30阶滞后自相关函数图

(b) 实现本题的代码如下:

```
1 > results=ARMA11(0.7,-0.4,30,TRUE)
2 > results$ACFplot
```

绘制的自相关函数图如下:

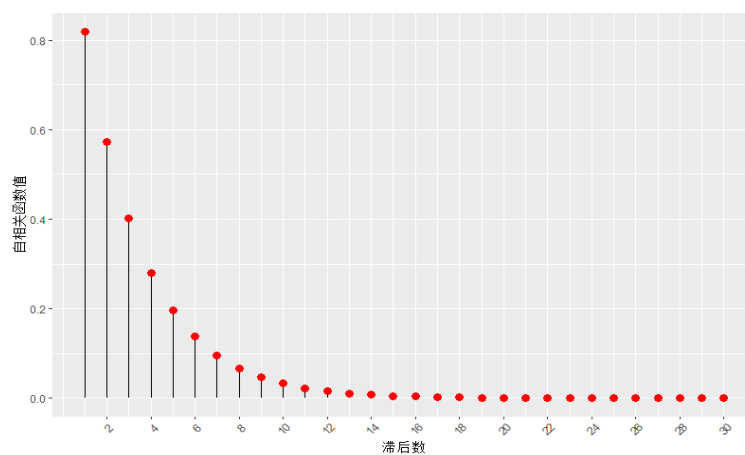


图 8: 习题4.10(b)的1-30阶滞后自相关函数图