R 语言
逐步回归法代码演示

杜冰

苏州大学数学科学学院

2020 年 11 月 27 日

Contents

数据输入结构

变量 1	变量 2	变量 3	变量 4	…	y
⋮	⋮	⋮	⋮	…	⋮

图: 数据输入结构

前进法

输入及输出

forward_ice <- function(data_input, sig)

输入

data: 数据
sig: 显著性水平

输出

使用前进法得到的回归结果

前进法

代码讲解

前进法代码

```
1  #前进法
2  forward_ice <- function(data_input, sig)
3  {
4    #前进法
5    len<-length(data_input)-1
6    observation<-lengths(data_input[,1])
7    index=seq(0,0,length=len)
8    variable=matrix(seq(0,0,length=len*observation),observation,len)
9    y=data.frame(data_input[,len+1],nrow=observation)[,1]
10   flag=0
```

前进法

代码讲解

前进法代码

```
11  while(flag<sig)
12  {
13    i=1
14    min=1
15    while(i<=len)#测试哪一个最显著，并获得相应的索引
16    {
17      if(index[i]==0)
18      {
19        index[i]=1
20        k=1
21        while(k<=len)#记录目前参与回归的变量
22        {
23          a<-data.frame(index[k]*data_input[,k],nrow=observation)
24          for(j in 1:observation) variable[j,k]<-a[j,1]
25          k=k+1
26        }
27        fit<-lm(y~variable[,1:len])#回归
28        temp<-summary(fit)
29        j=0
30        k=0
```

前进法

代码讲解

前进法代码

```
31          while(k<i)
32          {
33            k=k+1
34            if(index[k]!=0)
35            {
36              j=j+1
37            }
38          }
39          p_value<-temp$coefficients[j+1,4]
40          if(p_value<min)
41          {
42            min=p_value
43            location_e=i
44          }
45          index[i]=0
46        }
47        i=i+1
48      }
```

前进法

代码讲解

前进法代码

```
49      #当显著性水平满足条件时，引入一个变量
50      if(min<sig)
51      {
52        index[location_e]=1
53      }
54      flag=min
55    }
56    k=1
57    while(k<=len)#记录目前参与回归的变量
58    {
59      a<-data.frame(index[k]*data_input[,k],nrow=observation)
60      for(j in 1:observation) variable[j,k]<-a[j,1]
61      k=k+1
62    }
63    fit<-lm(data$y~variable[,1:len])
64    temp<-summary(fit)
65    return(temp)
66  }
```

后退法

输入及输出

backward_ice <- function(data_input, sig)

输入

data: 数据
sig: 显著性水平

输出

使用后退法得到的回归结果

后退法

代码讲解

后退法代码

```
1  #后退法
2  backward_ice <- function(data_input, sig)
3  {
4    len<-length(data_input)-1
5    observation<-lengths(data_input[,1])
6    index=seq(1,1,length=len)
7    variable=matrix(seq(0,0,length=len*observation),observation,len)
8    y=data.frame(data_input[,len+1],nrow=observation)[,1]
9    flag=1#初始化flag
```

后退法

代码讲解

后退法代码

```
10  while(flag>sig)
11  {
12    i=1
13    while(i<=len)#记录目前参与回归的变量
14    {
15      a<-data.frame(index[i]*data_input[,i],nrow=observation)
16      for(j in 1:observation) variable[j,i]<-a[j,1]
17      i=i+1
18    }
19    fit<-lm(y~variable[,1:len])#回归
20    temp<-summary(fit)
21    length=length(temp$coefficients[,4])
22    p_value<-temp$coefficients[2:length,4]
23    flag=max(p_value)#取t检验对应最大值
```

后退法

代码讲解

后退法代码

```
24      if(flag>sig)#判断是否大于显著水平
25      {
26        j=0
27        location=max.col(t(p_value))#获得对应变量的相应位置
28        while(location>0)
29        {
30          j=j+1
31          if(index[j]!=0) location=location-1
32        }
33        index[j]=0#删除变量
34      }
35    }
```

后退法

代码讲解

后退法代码

```
36    i=1
37    while(i<=len)#记录目前参与回归的变量
38    {
39      a<-data.frame(index[i]*data_input[,i],nrow=observation)
40      for(j in 1:observation) variable[j,i]<-a[j,1]
41      i=i+1
42    }
43    fit<-lm(data$y~variable[,1:len])
44    temp<-summary(fit)
45    return(temp)
46  }
```

逐步回归法

输入及输出

both_ice <- function(data_input, enter, out)

输入

data: 数据
enter: 进入显著性水平
out: 剔除显著性水平

输出

使用逐步回归法得到的回归结果

逐步回归法

代码讲解

逐步回归法代码

```
1  #逐步回归法
2  both_ice <- function(data_input, enter, out)
3  {
4    len<-length(data_input)-1
5    observation<-lengths(data_input[,1])
6    index=seq(0,0,length=len)
7    variable=matrix(seq(0,0,length=len*observation),observation,len)
8    y=data.frame(data_input[,len+1],nrow=observation)[,1]
9    flag1=0
10   #前进法
```

逐步回归法

代码讲解

逐步回归法代码

```
11  while(flag1<enter)#严进
12  {
13    i=1
14    min=1
15    while(i<=len)#测试哪一个最显著，并获得相应的索引
16    {
17      if(index[i]==0)
18      {
19        index[i]=1
20        k=1
21        while(k<=len)#记录目前参与回归的变量
22        {
23          a<-data.frame(index[k]*data_input[,k],nrow=observation)
24          for(j in 1:observation) variable[j,k]<-a[j,1]
25          k=k+1
26        }
27        fit<-lm(y~variable[,1:len])#回归
28        temp<-summary(fit)
29        j=0
30        k=0
```

逐步回归法

代码讲解

逐步回归法代码

```
31          while(k<i)
32          {
33            k=k+1
34            if(index[k]!=0)
35            {
36              j=j+1
37            }
38          }
39          p_value<-temp$coefficients[j+1,4]
40          if(p_value<min)
41          {
42            min=p_value
43            location_e=i
44          }
45          index[i]=0
46        }
47        i=i+1
48      }
```

逐步回归法

代码讲解

逐步回归法代码

```
49      #当显著性水平满足条件时，引入一个变量
50      if(min<enter)
51      {
52        index[location_e]=1
53      }
54      flag1=min
```

逐步回归法

代码讲解

逐步回归法代码

```
56      #引入变量后进行后退，将显著性差的变量剔除（后退法）
57      flag2=1
58      i=0
59      while(flag2>out)
60      {
61        k=1
62        while(k<=len)#记录目前参与回归的变量
63        {
64          a<-data.frame(index[k]*data_input[,k],nrow=observation)
65          for(j in 1:observation) variable[j,k]<-a[j,1]
66          k=k+1
67        }
68        fit<-lm(y~variable[,1:len])#回归
69        temp<-summary(fit)
70        length=length(temp$coefficients[,4])
71        p_value<-temp$coefficients[2:length,4]
72        flag2=max(p_value)
```

逐步回归法

代码讲解

逐步回归法代码

```
73        if(flag2>out)#宽出
74        {
75          j=0
76          location=max.col(t(p_value))
77          while(location>0)
78          {
79            j=j+1
80            if(index[j]!=0) location=location-1
81          }
82          index[j]=0
83        }
84      }
85    }
```

逐步回归法

代码讲解

逐步回归法代码

```
86    k=1
87    while(k<=len)#记录目前参与回归的变量
88    {
89      a<-data.frame(index[k]*data_input[,k],nrow=observation)
90      for(j in 1:observation) variable[j,k]<-a[j,1]
91      k=k+1
92    }
93    fit<-lm(y~variable[,1:len])#回归
94    temp<-summary(fit)
95    return(temp)
96  }
```

代码测试

数据来源：课本习题

测试代码

```
library(readxl)
setwd('D:/2020秋/应用回归分析/代码汇报')
data<-read_xlsx("ex9.xlsx")
forward_ice(data,0.05)
backward_ice(data,0.1)
both_ice(data,0.05,0.1)
```

代码测试

测试结果

```
> forward_ice(data,0.05)

Call:
lm(formula = data$y ~ variable[, 1:len])

Residuals:
    Min      1Q  Median      3Q     Max
-372.26 -102.79   -7.77  157.98  313.69

Coefficients: (3 not defined because of singularities)
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        874.60021  106.86563   8.184 2.67e-07 ***
variable[, 1:len]1  -0.61119    0.12382  -4.936 0.000125 ***
variable[, 1:len]2  -0.35305    0.08840  -3.994 0.000940 ***
variable[, 1:len]3        NA         NA      NA       NA
variable[, 1:len]4        NA         NA      NA       NA
variable[, 1:len]5   0.63671    0.08914   7.143 1.65e-06 ***
variable[, 1:len]6        NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 183.1 on 17 degrees of freedom
Multiple R-squared:  0.9958,    Adjusted R-squared:  0.9951
F-statistic:  1356 on 3 and 17 DF,  p-value: < 2.2e-16

There were 50 or more warnings (use warnings() to see the first 50)
```

代码测试

测试结果

```
> backward_ice(data,0.1)

Call:
lm(formula = data$y ~ variable[, 1:len])

Residuals:
    Min      1Q  Median      3Q     Max
-372.26 -102.79   -7.77  157.98  313.69

Coefficients: (3 not defined because of singularities)
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        874.60021  106.86563   8.184 2.67e-07 ***
variable[, 1:len]1  -0.61119    0.12382  -4.936 0.000125 ***
variable[, 1:len]2  -0.35305    0.08840  -3.994 0.000940 ***
variable[, 1:len]3        NA         NA      NA       NA
variable[, 1:len]4        NA         NA      NA       NA
variable[, 1:len]5   0.63671    0.08914   7.143 1.65e-06 ***
variable[, 1:len]6        NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 183.1 on 17 degrees of freedom
Multiple R-squared:  0.9958,    Adjusted R-squared:  0.9951
F-statistic:  1356 on 3 and 17 DF,  p-value: < 2.2e-16

There were 31 warnings (use warnings() to see them)
```

代码测试

测试结果

```
> both_ice(data,0.05,0.1)

Call:
lm(formula = y ~ variable[, 1:len])

Residuals:
    Min      1Q  Median      3Q     Max
-372.26 -102.79   -7.77  157.98  313.69

Coefficients: (3 not defined because of singularities)
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        874.60021  106.86563   8.184 2.67e-07 ***
variable[, 1:len]1  -0.61119    0.12382  -4.936 0.000125 ***
variable[, 1:len]2  -0.35305    0.08840  -3.994 0.000940 ***
variable[, 1:len]3        NA         NA      NA       NA
variable[, 1:len]4        NA         NA      NA       NA
variable[, 1:len]5   0.63671    0.08914   7.143 1.65e-06 ***
variable[, 1:len]6        NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 183.1 on 17 degrees of freedom
Multiple R-squared:  0.9958,    Adjusted R-squared:  0.9951
F-statistic:  1356 on 3 and 17 DF,  p-value: < 2.2e-16

There were 50 or more warnings (use warnings() to see the first 50)
```