1807403036+ 杜冰 +5

1 问题:

在研究国家财政收入时，我们把财政收入按收入形式分为：各项税收收入、企业收入、债务收入、国家能源交通重点建设基金收入、基本建设贷款归还收入、国家预算调节基金收入、其他收入等。为了建立国家财政收入回归模型，我们以财政收入 y(亿元) 为因变量，自变量如下：x1 为农业增加值 (亿元)；x2 为工业增加值 (亿元)；x3 为建筑业增加值 (亿元)；x4 为人口数 (万人)；x5 为社会消费总额 (亿元)；x6 为受灾面积 (万公顷)。从《中国统计年鉴》获得 1978 − 1998 年共 21 个年份的统计数据，见附录 1。由定性分析知，所选自变量都与因变量 y 有较强的相关性, 分别用后退法和逐步回归法做自变量选元。

解:

1. 后退法:

   运行代码如下所示（运行结果见附录 2）:

```
rm(list=ls())
library(readxl)
setwd('D:/2020秋/应用回归分析/作业/第五次作业')
data<-read_xlsx("ex9.xlsx")
index<-matrix(c(1,1,1,1,1,1),6,1)
variable<-matrix(c(data$x1,data$x2,data$x3,data$x4,data$
    x5,data$x6),21,6)
#后退法
flag=1
i=0
while(flag>0.01)
{
        variable=matrix(c(index[1]*data$x1,index[2]*data$x2,
            index[3]*data$x3,index[4]*data$x4,index[5]*data$
            x5,index[6]*data$x6),21,6)
```

```
        fit <-lm(data$y~variable[,1]+variable[,2]+variable
            [,3]+variable[,4]+variable[,5]+variable[,6])
        temp<-summary(fit)
        length=7-i
        p_value<-temp$coefficients[2:length,4]
        flag=max(p_value)
        flag
        if(flag>0.01)
        {
                j=0
                location=max.col(t(p_value))
                while(location>0)
                {
                        j=j+1
                        if(index[j]!=0) location=location-1
                }
                index[j]=0
                i=i+1
        }
}
summary(fit)
```

通过后退法得到的回归函数为：

$$\hat{y} = 874.60021 - 0.61119x_1 - 0.35305x_2 + 0.63671x_5$$

2. 逐步回归法

运行代码如下所示（运行结果见附录 3）：

```
rm(list=ls())
library(readxl)
setwd('D:/2020秋/应用回归分析/作业/第五次作业')
data<-read_xlsx("ex9.xlsx")
#逐步回归法
enter=0.05#进入置信水平
```

```
out=0.1#剔除置信水平
index<-matrix(c(0,0,0,0,0,0),6,1)#索引
variable<-matrix(c(data$x1,data$x2,data$x3,data$x4,data$
    x5,data$x6),21,6)
flag1=0
flag2=1
min=1
while(flag1<enter||flag2>out)
{
        i=1
        while(i<=6)
        {
                if(index[i]==0)
                {
                        index[i]=1
                        variable=matrix(c(index[1]*data$x1,
                            index[2]*data$x2,index[3]*data$x3,
                            index[4]*data$x4,index[5]*data$x5,
                            index[6]*data$x6),21,6)
                        fit <-lm(data$y~variable[,1]+variable
                            [,2]+variable[,3]+variable[,4]+
                            variable[,5]+variable[,6])
                        temp<-summary(fit)
                        j=0
                        k=0
                        while(k<i)
                        {
                                k=k+1
                                if(index[k]!=0)
                                {
                                        j=j+1
                                }
                        }
```

```
                p_value<-temp$coefficients[j+1,4]
                if(p_value<min)
                {
                        min=p_value
                        location_e=i
                }
                index[i]=0
        }
        i=i+1
}
#满足xx条件时引入一个变量
if(min<enter)
{
        index[location_e]=1
}
#引入变量后进行后退，将显著性差的变量剔除（后退法）
flag2=1
i=0
while(flag2>out)
{
        variable=matrix(c(index[1]*data$x1,index[2]*
            data$x2,index[3]*data$x3,index[4]*data$x4,
            index[5]*data$x5,index[6]*data$x6),21,6)
        fit <-lm(data$y~variable[,1]+variable[,2]+
            variable[,3]+variable[,4]+variable[,5]+
            variable[,6])
        temp<-summary(fit)
        length=length(temp$coefficients[,4])
        p_value<-temp$coefficients[2:length,4]
        flag2=max(p_value)
        flag2
        if(flag2>out)
        {
```

```
63                  j=0
64                  location=max.col(t(p_value))
65                  while(location>0)
66                  {
67                          j=j+1
68                          if(index[j]!=0) location=location
                                -1
69                  }
70                  index[j]=0
71                  i=i+1
72          }
73          #判断是否有还可以再进入的变量
74          flag1=1
75          i=1
76          while(i<=6)
77          {
78                  if(index[i]==0)
79                  {
80                          index[i]=1
81                          variable=matrix(c(index[1]*data
                                $x1,index[2]*data$x2,index
                                [3]*data$x3,index[4]*data$x4
                                ,index[5]*data$x5,index[6]*
                                data$x6),21,6)
82                          fit <-lm(data$y~variable[,1]+
                                variable[,2]+variable[,3]+
                                variable[,4]+variable[,5]+
                                variable[,6])
83                          temp<-summary(fit)
84                          j=0
85                          k=0
86                          while(k<i)
87                          {
```

```
                                        k=k+1
                                        if(index[k]!=0)
                                        {
                                                j=j+1
                                        }
                                }
                                p__value<-temp$coefficients[j
                                    +1,4]
                                if(p__value<flag1)
                                {
                                        flag1=p__value
                                        location__e=i
                                }
                                index[i]=0
                        }
                        i=i+1
                }
        }
}
variable=matrix(c(index[1]*data$x1,index[2]*data$x2,index[3]
    *data$x3,index[4]*data$x4,index[5]*data$x5,index[6]*data
    $x6),21,6)
fit <-lm(data$y~variable[,1]+variable[,2]+variable[,3]+
    variable[,4]+variable[,5]+variable[,6])
summary(fit)
```

所以通过逐步回归法得到的回归函数为：

$$\hat{y} = 874.60021 - 0.61119x_1 - 0.35305x_2 + 0.63671x_5$$

2 附录

2.1 附录 1

年份	农业 x1	工业 x2	建筑业 x3	人口 x4	最终消费 x5	受灾面积 x6	财政收入 y
1978	1018.4	1607.0	138.2	96259	2239.1	50760	1132.3
1979	1258.9	1769.7	143.8	97542	2619.4	39370	1146.4
1980	1359.4	1996.5	195.5	98705	2976.1	44530	1159.9
1981	1545.6	2048.4	207.1	100072	3309.1	39790	1175.8
1982	1761.6	2162.3	220.7	101654	3637.9	33130	1212.3
1983	1960.8	2375.6	270.6	103008	4020.5	34710	1367.0
1984	2295.5	2789.0	316.7	104357	4694.5	31890	1642.9
1985	2541.6	3448.7	417.9	105851	5773.0	44370	2004.8
1986	2763.9	3967.0	525.7	107507	6542.0	47140	2122.0
1987	3204.3	4585.8	665.8	109300	7451.2	42090	2199.4
1988	3831.0	5777.2	810.0	111026	9360.1	50870	2357.2
1989	4228.0	6484.0	794.0	112704	10556.5	46990	2664.9
1990	5017.0	6858.0	859.4	114333	11365.2	38470	2937.1
1991	5288.6	8087.1	1015.1	115823	13145.9	55470	3149.5
1992	5800.0	10284.5	1415.0	117171	15952.1	51330	3483.4
1993	6882.1	14143.8	2284.7	118517	20182.1	48830	4349.0
1994	9457.2	19359.6	3012.6	119850	26796.0	55040	5218.1
1995	11993.0	24718.3	3819.6	121121	33635.0	45821	6242.2
1996	13844.2	29082.6	4530.5	122389	40003.9	46989	7408.0
1997	14211.2	32412.1	4810.6	123626	43579.4	53429	8651.1
1998	14599.6	33429.8	5262.0	124810	46405.9	50145	9876.0

2.2 附录 2

后退法运行结果为：

```
> rm(list=ls())
> library(readxl)
> setwd('D:/2020秋/应用回归分析/作业/第五次作业')
> data<-read_xlsx("ex9.xlsx")
> index<-matrix(c(1,1,1,1,1,1),6,1)
> variable<-matrix(c(data$x1,data$x2,data$x3,data$x4,data$x5,
    data$x6),21,6)
> #后退法
> flag=1
> i=0
> while(flag>0.01)
+ {
      +   variable=matrix(c(index[1]*data$x1,index[2]*data$x2,
          index[3]*data$x3,index[4]*data$x4,index[5]*data$x5,
          index[6]*data$x6),21,6)
      +   fit <-lm(data$y~variable[,1]+variable[,2]+variable[,3]+
          variable[,4]+variable[,5]+variable[,6])
      +   temp<-summary(fit)
      +   length=7-i
      +   p_value<-temp$coefficients[2:length,4]
      +   flag=max(p_value)
      +   flag
      +   if(flag>0.01)
      +   {
              +     j=0
              +     location=max.col(t(p_value))
              +     while(location>0)
              +     {
                      +       j=j+1
                      +       if(index[j]!=0) location=location-1
                      +     }
```

```
28                  +     index[j]=0
29                  +     i=i+1
30                  +   }
31          + }
32  > summary(fit)
33
34  Call:
35  lm(formula = data$y ~ variable[, 1] + variable[, 2] + variable[,
36  3] + variable[, 4] + variable[, 5] + variable[, 6])
37
38  Residuals:
39  Min      1Q  Median      3Q     Max
40  -372.26 -102.79   -7.77  157.98  313.69
41
42  Coefficients: (3 not defined because of singularities)
43  Estimate Std. Error t value Pr(>|t|)
44  (Intercept)    874.60021  106.86563   8.184 2.67e-07 ***
45  variable[, 1]   -0.61119    0.12382  -4.936 0.000125 ***
46  variable[, 2]   -0.35305    0.08840  -3.994 0.000940 ***
47  variable[, 3]         NA         NA      NA       NA
48  variable[, 4]         NA         NA      NA       NA
49  variable[, 5]    0.63671    0.08914   7.143 1.65e-06 ***
50  variable[, 6]         NA         NA      NA       NA
51  ---
52  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
        1
53
54  Residual standard error: 183.1 on 17 degrees of freedom
55  Multiple R-squared: 0.9958,     Adjusted R-squared: 0.9951
56  F-statistic:  1356 on 3 and 17 DF, p-value: < 2.2e-16
```

2.3 附录 3

逐步回归法运行结果为：

```
> rm(list=ls())
> library(readxl)
> setwd('D:/2020秋/应用回归分析/作业/第五次作业')
> data<-read_xlsx("ex9.xlsx")
> #逐步回归法
> enter=0.05#进入置信水平
> out=0.1#剔除置信水平
> index<-matrix(c(0,0,0,0,0,0),6,1)#索引
> variable<-matrix(c(data$x1,data$x2,data$x3,data$x4,data$x5,
    data$x6),21,6)
> flag1=0
> flag2=1
> min=1
> while(flag1<enter||flag2>out)
+ {
      +   i=1
      +   while(i<=6)
      +   {
            +     if(index[i]==0)
            +     {
                  +       index[i]=1
                  +       variable=matrix(c(index[1]*data$x1,
                      index[2]*data$x2,index[3]*data$x3,index
                      [4]*data$x4,index[5]*data$x5,index[6]*
                      data$x6),21,6)
                  +        fit <-lm(data$y~variable[,1]+
                      variable[,2]+variable[,3]+variable[,4]+
                      variable[,5]+variable[,6])
                  +       temp<-summary(fit)
                  +       j=0
                  +       k=0
```

```
26                  +       while(k<i)
27                  +       {
28                  +           k=k+1
29                  +           if(index[k]!=0)
30                  +           {
31                  +               j=j+1
32                  +           }
33                  +       }
34                  +       p_value<-temp$coefficients[j+1,4]
35                  +       if(p_value<min)
36                  +       {
37                  +           min=p_value
38                  +           location_e=i
39                  +       }
40                  +       index[i]=0
41                  +   }
42              +   i=i+1
43              +  }
44          +   #满足xx条件时引入一个变量
45          +   if(min<enter)
46          +   {
47              +   index[location_e]=1
48              +  }
49          +   #引入变量后进行后退，将显著性差的变量剔除（后退法）
50          +   flag2=1
51          +   i=0
52          +   while(flag2>out)
53          +   {
54              +   variable=matrix(c(index[1]*data$x1,index[2]*
                    data$x2,index[3]*data$x3,index[4]*data$x4,
                    index[5]*data$x5,index[6]*data$x6),21,6)
55              +    fit <-lm(data$y~variable[,1]+variable[,2]+
                    variable[,3]+variable[,4]+variable[,5]+variable
```

```
    [,6])
+   temp<-summary(fit)
+   length=length(temp$coefficients[,4])
+   p_value<-temp$coefficients[2:length,4]
+   flag2=max(p_value)
+   flag2
+   if(flag2>out)
+   {
+     j=0
+     location=max.col(t(p_value))
+     while(location>0)
+     {
+       j=j+1
+       if(index[j]!=0) location=
          location-1
+     }
+     index[j]=0
+     i=i+1
+   }
+   #判断是否有还可以再进入的变量
+   flag1=1
+   i=1
+   while(i<=6)
+   {
+     if(index[i]==0)
+     {
+       index[i]=1
+       variable=matrix(c(index[1]*
          data$x1,index[2]*data$x2,index[3]
          *data$x3,index[4]*data$x4,index
          [5]*data$x5,index[6]*data$x6)
          ,21,6)
+       fit <-lm(data$y~variable
```

```
                                    [,1]+variable[,2]+variable[,3]+
                                    variable[,4]+variable[,5]+variable
                                    [,6])
83                          +       temp<-summary(fit)
84                          +       j=0
85                          +       k=0
86                          +       while(k<i)
87                          +       {
88                              +       k=k+1
89                              +       if(index[k]!=0)
90                              +       {
91                                  +       j=j+1
92                                  +       }
93                              +     }
94                          +       p__value<-temp$coefficients
                                    [j+1,4]
95                          +       if(p__value<flag1)
96                          +       {
97                              +       flag1=p__value
98                              +       location__e=i
99                              +     }
100                         +       index[i]=0
101                         +     }
102                 +       i=i+1
103                 +     }
104             +   }
105     + }
106 > variable=matrix(c(index[1]*data$x1,index[2]*data$x2,index[3]*
      data$x3,index[4]*data$x4,index[5]*data$x5,index[6]*data$x6)
      ,21,6)
107 > fit<-lm(data$y~variable[,1]+variable[,2]+variable[,3]+variable
      [,4]+variable[,5]+variable[,6])
108 > summary(fit)
```

```
109
110 Call:
111 lm(formula = data$y ~ variable[, 1] + variable[, 2] + variable[,
112 3] + variable[, 4] + variable[, 5] + variable[, 6])
113
114 Residuals:
115 Min 1Q Median 3Q Max
116 -372.26 -102.79 -7.77 157.98 313.69
117
118 Coefficients: (3 not defined because of singularities)
119 Estimate Std. Error t value Pr(>|t|)
120 (Intercept) 874.60021 106.86563 8.184 2.67e-07 ***
121 variable[, 1] -0.61119 0.12382 -4.936 0.000125 ***
122 variable[, 2] -0.35305 0.08840 -3.994 0.000940 ***
123 variable[, 3] NA NA NA NA
124 variable[, 4] NA NA NA NA
125 variable[, 5] 0.63671 0.08914 7.143 1.65e-06 ***
126 variable[, 6] NA NA NA NA
127 ---
128 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
    1
129
130 Residual standard error: 183.1 on 17 degrees of freedom
131 Multiple R-squared: 0.9958, Adjusted R-squared: 0.9951
132 F-statistic: 1356 on 3 and 17 DF, p-value: < 2.2e-16
```