实验课程名称：<u>实用回归分析</u>

| 实验项目名称 | 自变量选择与逐步回归 | | | 实验成绩 | |
|---|---|---|---|---|---|
| 实 验 者 | | 专业班级 | | 组 别 | |
| 同 组 者 | | | | 实验日期 | 2019 年 5 月 28 日 |

## 第一部分：实验数据及要求

表 5-6

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $y$ |
|---|---|---|---|---|---|---|
| 1968 | 1 051.8 | 1 503.6 | 3.6 | 5.8 | 5.9 | 5 873 |
| 1969 | 1 078.8 | 1 486.7 | 3.5 | 6.7 | 4.5 | 7 852 |
| 1970 | 1 075.3 | 1 434.8 | 5.0 | 8.4 | 4.2 | 8 189 |
| 1971 | 1 107.5 | 2 035.6 | 6.0 | 6.2 | 4.2 | 7 494 |
| 1972 | 1 171.1 | 2 360.8 | 5.6 | 5.4 | 4.9 | 8 534 |
| 1973 | 1 235.0 | 2 043.9 | 4.9 | 5.9 | 5.0 | 8 688 |
| 1974 | 1 217.8 | 1 331.9 | 5.6 | 9.4 | 4.1 | 7 270 |
| 1975 | 1 202.3 | 1 160.0 | 8.5 | 9.4 | 3.4 | 5 020 |
| 1976 | 1 271.0 | 1 535.0 | 7.7 | 7.2 | 4.2 | 6 035 |
| 1977 | 1 332.7 | 1 961.8 | 7.0 | 6.6 | 4.5 | 7 425 |
| 1978 | 1 399.2 | 2 009.3 | 6.0 | 7.6 | 3.9 | 9 400 |
| 1979 | 1 431.6 | 1 721.9 | 6.0 | 10.6 | 4.4 | 9 350 |
| 1980 | 1 480.7 | 1 290.8 | 7.2 | 14.9 | 3.9 | 6 540 |
| 1981 | 1 510.3 | 1 100.0 | 7.6 | 16.6 | 3.1 | 7 675 |
| 1982 | 1 492.2 | 1 039.0 | 9.2 | 17.5 | 0.6 | 7 419 |
| 1983 | 1 535.4 | 1 200.0 | 8.8 | 16.0 | 1.5 | 7 923 |

1、 用 R 软件完成下列的计算分析：对表 5-6 中的自变量分别用前进法、后退法，逐步回归法选择自变量，并根据上述结果分析前进法、后退法与逐步回归法的差异。

## 第二部分：实验过程记录

过程记录（包括操作的步骤或者代码，输出的结果或者图形）：

# 一、前进法

## 1.1 前进法原理

前进法的思想是变量由少到多，每次增加一个，直至没有可引入的变量为止。首先分别对因变量 $y$ 建立 $m$ 个一元线性回归方程，并分别计算这 $m$ 个一元线性回归方程的 $m$ 个回归系数的 $F$ 检验值，记为 $\{F_1^1, F_2^1, \cdots F_m^1\}$，选其最大者记为：

$$F_j^1 = \max\{F_1^1, F_2^1, \cdots F_m^1\}$$

给定显著性水平 $\alpha$，若 $F_j^1 \geq F_\alpha(1, n-2)$，则首先将 $x_j$ 引入回归方程。

接下来因变量 $y$ 分别和 $(x_j, x_1), (x_j, x_2), \cdots (x_j, x_m)$ 建立二元线性回归方程，对这 $m-1$ 回归方程中的回归系数除 $x_j$ 外进行 $F$ 检验，计算 $F$ 值，记为 $\{F_1^2, F_2^2, \cdots F_m^2\}$，选其最大者记为：

$$F_i^2 = \max\{F_1^2, F_2^2, \cdots F_m^2\}$$

给定显著性水平 $\alpha$，若 $F_j^2 \geq F_\alpha(1, n-3)$ ，则将 $x_i$ 引入回归方程。

根据上述方法一直做下去，直至所有未被引入方程的自变量的 $F$ 值均小于 $F_\alpha(1, n-p-1)$ 为止。这时得到的回归方程就是最终确定的方程。

## 1.2 编程实现

在本实验中我们取显著性水平 $\alpha = 0.05$。我们想到使用显著性 P 值做检验，每一步中引入显著性 P 值最小的变量（其 P 值小于 0.05），直到所有变量的显著性 P 值都小于 0.05。

第一步：引进第一个变量

代码如下：

```r

setwd("D:/program file/qq/QQ下载的文档/数学/回归分析/实验二")
data=read.csv("实验二数据.csv",head=T);
X=data[,1:6]
y=data[,7]

#第一次前进
p1=c()
for(i in 1:6){
      fit=lm(y~X[,i])
      S=summary(fit)
      p_value=S$coefficients[2,4]

      p1=rbind(p1,p_value)
   }
rownames(p1)=c("p_c1","p_c2","p_c3","p_c4","p_c5","p_c6")
p1
which(p1== min(p1), arr.ind = TRUE)
p1[which.min(p1)]
```

输出结果如下

```
> p1
          [,1]
p_c1 0.6461131
p_c2 0.4339956
p_c3 0.0498480
p_c4 0.3242039
p_c5 0.8534714
p_c6 0.9684487
> which(p1== min(p1), arr.ind = TRUE)
     row col
p_c3   3   1
> p1[which.min(p1)]
[1] 0.049848
```

由上上述结果可知，$x_3$ 的 P 值最小，且小于 0.05，故我们引入第一个个变量 $x_3$

第二步：引入第二个变量

代码如下：

```
X=X[,-3]
for(i in 1:5){
  fit=lm(y~X[,i]+data$x3)
  S=summary(fit)
  p_value=S$coefficients[2,4]

  p2=rbind(p2,p_value)
}
rownames(p2)=c("p_c1","p_c2","p_c4","p_c5","p_c6")
p2
which(p2== min(p2), arr.ind = TRUE)
p2[which.min(p2)]
```

输出结果如下：

```
           [,1]
p_c1 0.12863902
p_c2 0.07433673
p_c4 0.87993747
p_c5 0.02918142
p_c6 0.11804583
> which(p2== min(p2), arr.ind = TRUE)
     row col
p_c5   4   1
> p2[which.min(p2)]
[1] 0.02918142
```

由上上述结果可知，$x_5$ 的 P 值最小，且小于 0.05，故我们引入第二个变量 $x_5$，进行下一步运算。

第三步：引入第三个变量

代码如下：

```
#引入第三个变量
p3=c()
x=x[,-4]
for(i in 1:4){
  fit=lm(y~x[,i]+data$x3+data$x5)
  s=summary(fit)
  p_value=s$coefficients[2,4]

  p3=rbind(p3,p_value)
}
rownames(p3)=c("p_c1","p_c2","p_c4","p_c6")
p3
which(p3== min(p3), arr.ind = TRUE)
p3[which.min(p3)]
```

结果如下：

```
            [,1]
p_c1 0.56127832
p_c2 0.84670443
p_c4 0.03036694
p_c6 0.91156068
> which(p3== min(p3), arr.ind = TRUE)
     row col
p_c4   3   1
> p3[which.min(p3)]
[1] 0.03036694
```

由上上述结果可知，$x_4$的 P 值最小，且小于 0.05，故我们引入第三个个变量$x_4$，然后进行下一步运算。

第四步：引入第四个变量

代码如下：

```
#引入第四个变量
p4=c()
x=x[,-3]
for(i in 1:3){
  fit=lm(y~x[,i]+data$x3+data$x5+data$x4)
  s=summary(fit)
  p_value=s$coefficients[2,4]

  p4=rbind(p4,p_value)
}
rownames(p4)=c("p_c1","p_c2","p_c6")
p4
which(p4== min(p4), arr.ind = TRUE)
p4[which.min(p4)]
```

结果如下

```
            [,1]
p_c1 0.23833714
p_c2 0.45455353
p_c6 0.05658723
> which(p4== min(p4), arr.ind = TRUE)
     row col
p_c6   3   1
> p4[which.min(p4)]
[1] 0.05658723
```

由上上述结果可知，变量$x_6$的P值最小，但是大于0.05，所以引入变量结束。

使用变量$x_3,x_4,x_5$做回归，得到如下结果：

```
Call:
lm(formula = y ~ data$x3 + data$x4 + data$x5)

Residuals:
     Min       1Q   Median       3Q      Max 
-1519.99  -256.06    88.26   445.29   987.12 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1412.8070  1865.9124   0.757 0.463558    
data$x3        3.4398     0.7821   4.398 0.000868 ***
data$x4     -415.1365   169.1633  -2.454 0.030367 *  
data$x5      348.7293    92.2197   3.782 0.002616 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 796.6 on 12 degrees of freedom
Multiple R-squared:  0.6573,    Adjusted R-squared:  0.5717 
F-statistic: 7.673 on 3 and 12 DF,  p-value: 0.003989
```

最终使用前进法得到的方程为：

$$y = 1412.8070 + 3.4398x_3 - 415.1365x_4 + 348.7293x_5$$

虽然常数项的P值是大于0.05的，但是可以看出变量都是显著不为零的，而且方程也是显著的；一般有意义的都是变量的系数，所以可以认为我们得到的方程是合理的，并且具有实际意义的。

## 二、后退法

### 2.1 后退法原理

后退法与前进法相反，用全部$m$个自变量对因变量$y$建立一元线性回归方程，并分别计算这$m$个回归系数的$F$检验值，记为$\{F_1^m, F_2^m, \cdots F_m^m\}$，选其最大者记为：

$$F_j^m = \min\left\{F_1^m, F_2^m, \cdots F_m^m\right\}$$

给定显著性水平$\alpha$，若$F_j^m \le F_\alpha(1, n-m-1)$ ，则将$x_j$从回归方程中剔除。

接着对剩下的$m-1$个自变量重新建立回归方程，对这$m-1$个回归系数除$x_j$外进行$F$检验，计算$F$值，记为$\{F_1^{m-1}, F_2^{m-1}, \cdots F_m^{m-1}\}$，选其最大者记为：

$$F_i^{m-1} = \min\left\{F_1^{m-1}, F_2^{m-1}, \cdots F_m^{m-1}\right\}$$

给定显著性水平$\alpha$，若$F_j^{m-1} \le F_\alpha(1, n-(m-1)-1)$，则将$x_i$从回归方程中剔除。根据上述方法一直做下去，直至所有未被引入方程的自变量的$F$值均大于$F_\alpha$，没有可剔除的变量为止。

### 2.2 编程实现

同样的，我们取显著性水平$\alpha = 0.05$。依然使用显著性P值做检验，每一步中剔除显著性P值最大且大于0.05的变量（相当于我们剔除了最不显著的变量，是对取F值得改进），直到所有变量的显著性P值都小于0.05。

第一步：剔除第一个变量

代码如下：

```
###后退法
#剔除第一个变量

fit1=lm(y~data[,1]+data[,2]+data[,3]+data[,4]+data[,5]+data[,6])
fit1
summary(fit1)
```

所得结果如下：

```
Call:
lm(formula = y ~ data[, 1] + data[, 2] + data[, 3] + data[, 4] + 
    data[, 5] + data[, 6])

Residuals:
    Min      1Q  Median      3Q     Max 
-527.48 -269.20  -52.06  179.04  796.48 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)   
(Intercept) -2.531e+06  1.054e+06  -2.401  0.03984 * 
data[, 1]    1.305e+03  5.422e+02   2.407  0.03947 * 
data[, 2]   -2.746e+01  1.359e+01  -2.021  0.07404 . 
data[, 3]    3.321e+00  7.967e-01   4.169  0.00241 **
data[, 4]   -1.506e+03  3.248e+02  -4.637  0.00122 **
data[, 5]    2.125e+02  1.463e+02   1.453  0.18022   
data[, 6]   -4.779e+02  2.846e+02  -1.679  0.12741   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 514.6 on 9 degrees of freedom
Multiple R-squared:  0.8927,	Adjusted R-squared:  0.8212 
F-statistic: 12.48 on 6 and 9 DF,  p-value: 0.000645
```

由以上结果我们可以看出，变量$x_5$的P值是最大的且大于0.05，所以我们认为变量$x_5$是不显著的所以我们对变量$x_5$进行剔除处理。

第二步：剔除第二个变量

代码如下：

```
#剔除第二个变量
fit2=lm(y~data[,1]+data[,2]+data[,3]+data[,4]+data[,6])
fit2
summary(fit2)
```

结果如下：

```
> summary(fit2)

Call:
lm(formula = y ~ data[, 1] + data[, 2] + data[, 3] + data[, 4] + 
    data[, 6])

Residuals:
   Min     1Q Median     3Q    Max 
-632.8 -290.5 -163.2  236.3 1036.1 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.668e+06  9.182e+05  -1.817 0.099248 .  
data[, 1]    8.618e+02  4.726e+02   1.824 0.098192 .  
data[, 2]   -1.445e+01  1.077e+01  -1.341 0.209550    
data[, 3]    2.337e+00  4.417e-01   5.291 0.000352 ***
data[, 4]   -1.336e+03  3.194e+02  -4.183 0.001878 ** 
data[, 6]   -7.614e+02  2.184e+02  -3.486 0.005861 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 542.5 on 10 degrees of freedom
Multiple R-squared:  0.8676,	Adjusted R-squared:  0.8014 
F-statistic:  13.1 on 5 and 10 DF,  p-value: 0.0004008
```

由这一步结果我们发现变量 $x_2$ 的 P 值是最大的且大于 0.05,，所以我们认为变量 $x_2$ 是不显著的所以我们对变量 $x_2$ 进行剔除处理。

第三步：剔除第三个变量

代码如下：

```
#剔除第三个变量
fit3=lm(y~data[,1]+data[,3]+data[,4]+data[,6])
fit3
summary(fit3)
```

结果如下:

```
Call:
lm(formula = y ~ data[, 1] + data[, 3] + data[, 4] + data[, 6])

Residuals:
    Min      1Q  Median      3Q     Max 
-809.25 -265.20  -48.12  135.72 1154.31 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -4.454e+05  1.104e+05  -4.033 0.001974 ** 
data[, 1]    2.322e+02  5.614e+01   4.136 0.001655 ** 
data[, 3]    2.310e+00  4.570e-01   5.055 0.000369 ***
data[, 4]   -9.719e+02  1.741e+02  -5.582 0.000165 ***
data[, 6]   -8.280e+02  2.203e+02  -3.759 0.003161 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 561.8 on 11 degrees of freedom
Multiple R-squared:  0.8438,	Adjusted R-squared:  0.787 
F-statistic: 14.85 on 4 and 11 DF,  p-value: 0.0002075
```

由这一步结果我们可以看出，所有变量的 P 值均是小于 0.05 的，即所有变量均是显著不为 0 的，所以后退法到这一步终止。

最终我们得到的回归方程为：

$$y = -445400 + 232.2x_1 + 2.31x_3 - 971.9x_4 - 828x_6$$

对比前进法，我们发现二者差异较大，可以认为是两种方法的缺陷造成的，所以我们接下来使用逐步回归法来进行改进。

## 三、逐步回归法

### 3.1 逐步回归法原理

逐步回归的基本思想是有进有出。具体做法是将变量一个一个引入，每引入一个自变量后，对已选入的变量要进行逐个检查，当原引入的变量由于后面变量的引入而变得不再显著时，要将其剔除。引入一个变量或者从回归方程汇总剔除一个变量，为逐步回归的一步，每一步都要进行 F 检验，以确保每次引入新的变量之前回归方程中只包含显著的变量。这个过程反复进行，知道既无显著的自变量选入回归方程，也无不显著的自变量从回归方程中剔除为止。这样就避免了前进法和后退法各自的缺陷，保证了最后所得的回归子集是最优回归子集。

在逐步回归法中要求引入自变量与剔除自变量的显著性水平 $\alpha$ 值是不同的，要求引入自变量的显著性水平 $\alpha_{entry}$ 小于剔除自变量的显著性水平 $\alpha_{removal}$，以保证自变量的选择与剔除不会陷入“死循环”。

### 3.2 编程实现

本文首先使用了 R 语言自带的逐步回归函数 *step*。其是利用 AIC 赤池信息准则，它考虑了模型的统计拟合度以及用来拟合的参数数目。AIC 值越小，模型越优，它说明模型用较少的参数获得了足够的拟合度。下面我们先用这种方法进行编程求解。

代码如下：

```
#逐步回归法
tdata=data.frame(
x1=data[,1],
x2=data[,2],
x3=data[,3],
x4=data[,4],
x5=data[,5],
x6=data[,6],
y
)
tlm=lm(y~x1+x2+x3+x4+x5+x6,data=tdata)
summary(tlm)
tstep=step(tlm)
summary(tstep)
```

结果如下：

```
> tstep=step(tlm)
Start:  AIC=204.58
y ~ x1 + x2 + x3 + x4 + x5 + x6

       Df Sum of Sq     RSS    AIC
<none>              2383521 204.58
- x5    1    559022 2942543 205.96
- x6    1    746807 3130328 206.94
- x2    1   1081402 3464923 208.57
- x1    1   1533778 3917299 210.53
- x3    1   4603147 6986668 219.79
- x4    1   5694063 8077584 222.11
> summary(tstep)

Call:
lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6, data = tdata)

Residuals:
    Min      1Q  Median      3Q     Max 
-527.48 -269.20  -52.06  179.04  796.48 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)   
(Intercept) -2.531e+06  1.054e+06  -2.401  0.03984 * 
x1           1.305e+03  5.422e+02   2.407  0.03947 * 
x2          -2.746e+01  1.359e+01  -2.021  0.07404 . 
x3           3.321e+00  7.967e-01   4.169  0.00241 **
x4          -1.506e+03  3.248e+02  -4.637  0.00122 **
x5           2.125e+02  1.463e+02   1.453  0.18022   
x6          -4.779e+02  2.846e+02  -1.679  0.12741   
---
Signif. codes:  
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 514.6 on 9 degrees of freedom
Multiple R-squared:  0.8927,	Adjusted R-squared:  0.8212 
F-statistic: 12.48 on 6 and 9 DF,  p-value: 0.000645
```

我们可以很清楚地看到，在一个变量都不剔除的时候使得 AIC 达到了最小值。但是从下面我们又可以看到变量$x_3,x_5,x_6$的 P 值均是大于 0.05 的，也就是说他们是不显著的。经过分析，我们觉得是 R 语言自带的函数为了使 AIC 达到最小而出现了过拟合的现象，导致所建立的模型对原始数据拟合误差最小，但是这种模型在实际中用处是不大的。

所以，我们决定使用课上所讲的方法来重新进行逐步回归。

首先设置$\alpha_{removal}=0.06$，$\alpha_{entry}=0.05$，即若一个变量的 P 值小于 0.05 且是所有变量 P 值最小的我们才将其进行引入处理。但是在剔除变量时，如果变量的 P 值大于 0.06 且是最大值时我们就将其进行剔除处理。符合严进宽出准则。

第一步：引入第一个变量

代码如下：

```
#逐步回归——引入第一个变量
p1=c()
for(i in 1:6){
  fit=lm(y~X[,i])
  S=summary(fit)
  p_value=S$coefficients[2,4]

  p1=rbind(p1,p_value)
}
rownames(p1)=c("p_c1","p_c2","p_c3","p_c4","p_c5","p_c6")
p1
which(p1== min(p1), arr.ind = TRUE)
p1[which.min(p1)]
```

结果如下：

```
          [,1]
p_c1 0.6461131
p_c2 0.4339956
p_c3 0.0498480
p_c4 0.3242039
p_c5 0.8534714
p_c6 0.9684487
> which(p1== min(p1), arr.ind = TRUE)
     row col
p_c3   3   1
> p1[which.min(p1)]
[1] 0.049848
```

由上述结果可知，$x_3$的 P 值最小，且小于 0.05，故引入变量$x_3$。

第二步：引入第二个变量

代码如下：

```
#逐步回归——引入第二个变量
p2=c()
X=X[,-3]
for(i in 1:5){
  fit=lm(y~X[,i]+data$x3)
  S=summary(fit)
  p_value=S$coefficients[2,4]

  p2=rbind(p2,p_value)
}
rownames(p2)=c("p_c1","p_c2","p_c4","p_c5","p_c6")
p2
which(p2== min(p2), arr.ind = TRUE)
p2[which.min(p2)]
```

结果如下：

```
          [,1]
p_c1 0.12863902
p_c2 0.07433673
p_c4 0.87993747
p_c5 0.02918142
p_c6 0.11804583
> which(p2== min(p2), arr.ind = TRUE)
     row col
p_c5   4   1
> p2[which.min(p2)]
[1] 0.02918142
```

由上述结果可知，$x_5$ 的 P 值最小，且小于 0.05，故我们引入个变量 $x_5$，进行下一步运算。

第三步：检查已引入变量的 P 值是否符合规则

代码如下：

```
#第三步：检查已引入变量的P值是否符合规则
summary(lm(y~data[,3]+data[,5]))
```

结果如下：

```
Call:
lm(formula = y ~ data[, 3] + data[, 5])

Residuals:
     Min       1Q   Median       3Q      Max 
-1211.21  -723.51    44.83   643.67  1358.80 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept)  472.2978  2150.1379   0.220  0.82955   
data[, 3]      3.1882     0.9129   3.492  0.00397 **
data[, 5]    212.3245    86.6427   2.451  0.02918 * 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 938 on 13 degrees of freedom
Multiple R-squared:  0.4853,	Adjusted R-squared:  0.4062 
F-statistic:  6.13 on 2 and 13 DF,  p-value: 0.01333
```

可以看出变量 $x_3$ 的 P 值小于 0.06，所以模型保留变量 $x_3$。

第四步：引入第三个变量

代码如下：

```
#第四步：引入第三个变量
p3=c()
X=X[,-4]
for(i in 1:4){
  fit=lm(y~X[,i]+data$x3+data$x5)
  S=summary(fit)
  p_value=S$coefficients[2,4]

  p3=rbind(p3,p_value)
}
rownames(p3)=c("p_c1","p_c2","p_c4","p_c6")
p3
which(p3== min(p3), arr.ind = TRUE)
p3[which.min(p3)]
```

结果如下：

```
           [,1]
p_c1 0.56127832
p_c2 0.84670443
p_c4 0.03036694
p_c6 0.91156068
> which(p3== min(p3), arr.ind = TRUE)
     row col
p_c4   3   1
> p3[which.min(p3)]
[1] 0.03036694
```

由上上述结果可知，$x_4$ 的 P 值最小，且小于 0.05，故我们引入个变量 $x_4$，进行下一步运算。

第五步：对之前引入的变量进行显著性检验：

代码如下：

```
#第五步：对之前引入的变量进行显著性检验：
summary(lm(y~data[,3]+data[,4]+data[,5]))
```

结果如下：

```
Call:
lm(formula = y ~ data[, 3] + data[, 4] + data[, 5])

Residuals:
     Min       1Q   Median       3Q      Max
-1519.99  -256.06    88.26   445.29   987.12

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1412.8070  1865.9124   0.757 0.463558
data[, 3]      3.4398     0.7821   4.398 0.000868 ***
data[, 4]   -415.1365   169.1633  -2.454 0.030367 *
data[, 5]    348.7293    92.2197   3.782 0.002616 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 796.6 on 12 degrees of freedom
Multiple R-squared:  0.6573,	Adjusted R-squared:  0.5717
F-statistic: 7.673 on 3 and 12 DF,  p-value: 0.003989
```

可以看出前几步引入的变量 $x_3,x_5$ 的 P 值均是小于 0.06 的，所以我们不用剔除变量。

第六步：引入第四个变量

代码如下：

```
#逐步回归——引入第四个变量
p4=c()
X=X[,-3]
for(i in 1:3){
  fit=lm(y~X[,i]+data$x3+data$x5+data$x4)
  S=summary(fit)
  p_value=S$coefficients[2,4]

  p4=rbind(p4,p_value)
}
rownames(p4)=c("p_c1","p_c2","p_c6")
p4
which(p4== min(p4), arr.ind = TRUE)
p4[which.min(p4)]
```

结果如下：

```
         [,1]
p_c1 0.23833714
p_c2 0.45455353
p_c6 0.05658723
> which(p4== min(p4), arr.ind = TRUE)
     row col
p_c6   3   1
> p4[which.min(p4)]
[1] 0.05658723
```

我们可以看出变量$x_6$的 P 值是最小的，但是大于 0.05，我们不引入$x_6$。

所以逐步回归法终止。

最终得到的回归结果为：

```
Call:
lm(formula = y ~ data[, 3] + data[, 4] + data[, 5])

Residuals:
     Min       1Q   Median       3Q      Max 
-1519.99  -256.06    88.26   445.29   987.12 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1412.8070  1865.9124   0.757 0.463558    
data[, 3]      3.4398     0.7821   4.398 0.000868 ***
data[, 4]   -415.1365   169.1633  -2.454 0.030367 *  
data[, 5]    348.7293    92.2197   3.782 0.002616 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 796.6 on 12 degrees of freedom
Multiple R-squared:  0.6573,	Adjusted R-squared:  0.5717 
F-statistic: 7.673 on 3 and 12 DF,  p-value: 0.003989
```

我们可以看到所有变量除常数项外系数均是显著不为 0 的，而且回归方程的 P 值是 0.003989，是极其显著的。

所以最终得到的回归方程为：

$$y = 1412.8070 + 3.4398x_3 - 415.1365x_4 + 348.7293x_5$$

## 四、全子集回归法

全子集回归的思想很简单，就是把所有的特征组合都常识建模一遍，然后选择最优的模型。本题中指标 P 的个数较少，故采用全子集回归不失为一种较好的方法。与此同时，全子集回归结果也能与前面逐步回归结果进行对比，在模型选取上具有较好的参考价值。

以下采用了全子集自回归方法，以调整决定系数最大为目标进行求解，代码如下：

```r
#全子集回归
tdata=data.frame(
  x1=data[,1],
  x2=data[,2],
  x3=data[,3],
  x4=data[,4],
  x5=data[,5],
  x6=data[,6],
  y
)

library(leaps)
exp=regsubsets(y~x1+x2+x3+x4+x5+x6,data=tdata,nbest=1,really.big=T)
express=summary(exp)
res=data.frame(express$outmat,调整R方=express$adjr2)
res
```

结果如下：

```
         x1 x2 x3 x4 x5 x6   调整R方
1  ( 1 )        *             0.1938573
2  ( 1 )        *     *       0.4061664
3  ( 1 )        *  *  *       0.5716532
4  ( 1 )  *     *  *     *    0.7869583
5  ( 1 )  *  *  *  *     *    0.8013778
6  ( 1 )  *  *  *  *  *  *    0.8212354
```

上述结果中，第几行表示选几个变量使得调整决定系数最大。例如第二行表示在选两个变量的条件下，选 $x_3$ 和 $x_5$ 使得调整系数最大。同时可知，选六个变量时调整决定系数最大。

以为 $C_p$ 准则选取最优子集代码为：

```r
tdata=data.frame(
  x1=data[,1],
  x2=data[,2],
  x3=data[,3],
  x4=data[,4],
  x5=data[,5],
  x6=data[,6],
  y
)

library(leaps)
exp=regsubsets(y~x1+x2+x3+x4+x5+x6,data=tdata,nbest=1,really.big=T)
express=summary(exp)
res=data.frame(express$outmat,cp=express$cp)
res
```

输出结果为：

```
         x1 x2 x3 x4 x5 x6        cp
1  ( 1 )        *               51.133295
2  ( 1 )        *     *         33.184375
3  ( 1 )        *  *  *         20.753805
4  ( 1 )  *     *  *     *       7.109187
5  ( 1 )  *  *  *  *     *       7.110827
6  ( 1 )  *  *  *  *  *  *       7.000000
> 
```

我们可以看到，选取六个变量时使得 $C_p$ 准则达到最优。

教师签字__________

## 第三部分　结果与讨论（可加页）

在本实验过程中，我们使用了前进法、后退法、逐步回归法、全子集回归法对回归方程进行了变量选取的工作，使得我们将课上所学的内容进行了实践。

实验结果讨论

（1）使用前进法得到的回归方程为：

$$y = 1412.8070 + 3.4398x_3 - 415.1365x_4 + 348.7293x_5$$

使用后退法得到的方程为：

$$y = -445400 + 232.2x_1 + 2.31x_3 - 971.9x_4 - 828x_6$$

使用逐步回归法得到的结果为：

$$y = 1412.8070 + 3.4398x_3 - 415.1365x_4 + 348.7293x_5$$

（2）可以看出前进法和后退法得出的结果不一样。但是逐步回归法得到的结果和前进法得到的结果是一样的，这可能是由于$\alpha_{removal}$，$\alpha_{entry}$的选取具有一定的随机性而导致的，但是我们还是认为逐步回归法得到的结果具有较好的优越性。

（3）我们还使用了 AIC 准则进行逐步回归，可是得到的结果却不尽人意，相信在后续的课程中进行了更深入的学习后我们会有较好的解决方案。

（4）对于全子集回归方法，综合考虑了调整决定系数以及$C_p$准则，均是选择全部变量使得模型达到最优。