北京大学本科生毕业论文

保洁洗衣粉成份效果的统计分析

北京大学数学科学学院

概率统计系

唐明宇

00601072

2010-6-9

摘要：对案例中提供的数据先进行数据分析，我们选取最近邻方法添补数据，定义距离，以前 50 组数据为基础在其中选取距离缺失数据最近的三组数据的平均数来填补缺失的数据。经过数据标准化后，分别采用三种拟合方法对数据分别进行 18 次回归，分别是线性回归，岭回归，和 Lasso 回归。

关键字：保洁，统计模型，线性回归，岭回归，LASSO 回归

论文基本结构：

- **数据处理：**

  由于题目中给出的数据从观测 51 到观测 86 存在缺失数据，并且缺失数据都位于 PP2,PP3,PP4,和 PP5。所以先考虑填充数据，我们采用最近邻方法填充缺失数据。具体采用 3-neighbor 方法，以前 50 组完整数据观测为基准定义距离为，以缺失数据的观的 PP1,PP6-PP21 等几个变量到前 50 组观测的 PP1,PP6-PP21 的距离的绝对值的和为距离，选取距离最小的三组的相关值的平均值来填充缺失数据。

  将填充好的数据标准化。

- **线性回归**

  1. 先考察自变量 PP1-PP21 的共线性，采用量度$K = \lambda_{max}/\lambda_{min}$，其中$\lambda_{max}$是 X'X 的最大的特征值，$\lambda_{min}$是 X'X 的最小的特征值。我们认为当 K 大于 30 时，存在共线性。同时做各变量的相关系数图来考察变量间的共线性。
  2. 然后再分别对 18 组 O1-O18 做线性回归，考察是否存在异常值，如果有异常值则去掉异常值，重新做回归。
  3. 用 AIC 为标准选取有效变量，去掉存在较强共线性的变量，然后重新做回归。对剩下的变量用 K 方法考察共线性。
  4. 考虑剩下的有效变量的交叉项和二次项，然后重新回归。
  5. 再次以 AIC 为标准选取变量，确定最终变量，并计算此模型的残差平方和。

最好说明依据，如文献或者理由

6. 考察对因变量 Y 做 BOX-COX 变换Y^{λ}，如果有在区间[-2,2]上有使 L 最大的λ，则进行变换，否则，不进行变换。若变换再重新上述的回归分析的过程。

- **岭回归**

岭

1. 领回归内容以应用简介
2. 对项目中的问题，用 chemometrics 包里面的 plotRidge 函数考查岭回归的预测均方误差 MSEP。
3. 选取最佳的岭回归参数 lambda，可以用 MASS 包中的 select 函数反复几步精确计算。
4. 做岭回归的参数拟合。

- **LASSO 回归**

1. LASSO 回归内容及应用简介
2. 对项目中的问题，先用 10 折交叉验证画出 cv 误差曲线图，取出 cv 误差值最小的收缩系数，再画出回归系数收缩图。
3. 用前面取出的最小 cv 误差的收缩系数做 lasso 回归得出回归系数（lasso 回归用 lars 的软件包）。

结论：

我们最终选定线性回归，模型如下：

$$\begin{aligned} O1 = {} & 13.64PP3 + 2.526PP9 - 8.986PP10 + 7.763PP11 + 16.43PP12 + 8.576PP13 \\ & - 0.612PP1 * PP3 - 0.5198PP3 * PP9 - 22.26PP3 * PP12 + 10.28PP3 \\ & * PP17 + 6.525PP10 * PP9 - 8.36PP13 * PP9 + 8.181PP10 * PP10 \\ & - 14.56PP12 * PP11 + 10.26PP12 * PP12 - 30.13PP17 * PP12 \\ & - 7.306PP13 * PP13 + 15.4PP17 * PP17 \end{aligned}$$

三种方法具体分析过程如下：

用最临近法填补缺失数据：

基本思想是用 PP1，PP6-PP21 来定义一个距离，即各分量的绝对值的和。对每个有确实数据的案例，求它与前 50 个案例的距离，选取距离最小的三组，然后将相对应的值求平均填补到空处。

```
###全部数据标准化
x <- read.csv("D:/我的文档/桌面/p&G/data.csv")
z<-x[,2:40]
n<-50
y<-x[1:n, 2:40]
E<-rep(0,39)
std<-rep(0,39)
for(i in 1:39){
    E[i]<-mean(y[,i])
    std[i]<-sqrt(mean((y[,i]-E[i])^2))
}
yy<-x[,2:40]
for(i in 1:39){
    yy[,i]<-(yy[,i]-E[i])/std[i]
}

###对有缺失值的数据，选择距离最近的三组数据，用此三组数据的相关的数据的平均填补缺失数据
##距离定义为第 1，6：21 栏距离差的绝对值的和
for(r in 51:86){
    summ<-rep(0,50)
    ##计算 yy[r,]与前 50 组数据中的第 i 组数据的距离，记为 summ[i]
    for(i in 1:50){
        summ[i]<-0
        for(j in 5:21){
            summ[i]<-abs(yy[i,j]-yy[r,j])+summ[i]
```

```
        }
        summ[i]<-summ[i]+abs(yy[i,1]-yy[r,1])
    }
    ##minorder 存放与 r 行距离最近的三行的行号
    minorder<-order(summ)[1:3]
    #如果第 r 行 2，3 栏是缺失值，则用与第 r 行最近的三行的 2，3 的平均数填补
    if(is.na(yy[r,2])) {
            z[r,2]<-(z[minorder[1],2]+z[minorder[2],2]+z[minorder[3],2])/3
            z[r,3]<-(z[minorder[1],3]+z[minorder[2],3]+z[minorder[3],3])/3
    }
    #如果第 r 行 4，5 栏是缺失值，则用与第 r 行最近的三行的 4，5 的平均数填补
    if(is.na(yy[r,4])) {
            z[r,4]<-(z[minorder[1],4]+z[minorder[2],4]+z[minorder[3],4])/3
            z[r,5]<-(z[minorder[1],5]+z[minorder[2],5]+z[minorder[3],5])/3
    }

}
yy<-z
```

将数据填充好，得到数据 yy；扩充数据，得含有交叉项和二次项的数据 yyy，其中 yy 和 yyy 均已经经过标准化。zz 为 yy 没有标准化的数据，zzz 为 yyy 没有标准化的数据。

```
yyy<-yy
for(i in 1:21){        ##补充交叉项和二次项，V（39+(i-1）*21+j)是 PPi 和 PPj 的乘积的交叉项

    for(j in 1:21){
        yyy[,39+(i-1)*21+j]<-yy[,i]*yy[,j]
    }
```

```
}
zz<-yy
zzz<-yyy
E<-rep(0,480)
std<-rep(0,480)
for(i in 1:480){
    E[i]<-mean(zzz[,i])        ##E 为各列的均值
    std[i]<-sqrt(mean((zzz[,i]-E[i])^2))        ##std 为各列的标准差
}
for(i in 1:480){
    yyy[,i]<-(yyy[,i]-E[i])/std[i]
}
yy<-yyy[,1:39]
```

1.【线性回归】

对于 X 共性的考察，

所谓共线性问题，就是当自变量彼此相关时，估计的效应会由于模型中的其它自变量而改变数值，甚至符号。故在分析时，了解自变量关系的影响是很重要的。这里我们采用条件数 K 来度量样本的共线性程度。

一个基于特征值的常用量称为条件数 K，定义为 K=(最大特征值/最小特征值)$^{0.5}$， K 大于等于 1。大的 K 值表示共线性强。K 是在分析计算机算法的数值特征时自然得产生的一个量，但其作为共线性的一个统计量的含义是不太清楚的。我们这里采用的规则是，当 K>=30 时认为有共线性。

```
XX<-array(0,c(21,21))
for (i in 1:21){  ##计算 t(X)*X 矩阵
    for(j in 1:21){
        XX[i,j]<-yy[,i]%*%yy[,j]
```

}

}

lanmeta<-eigen(XX)$val

##特征向量数组

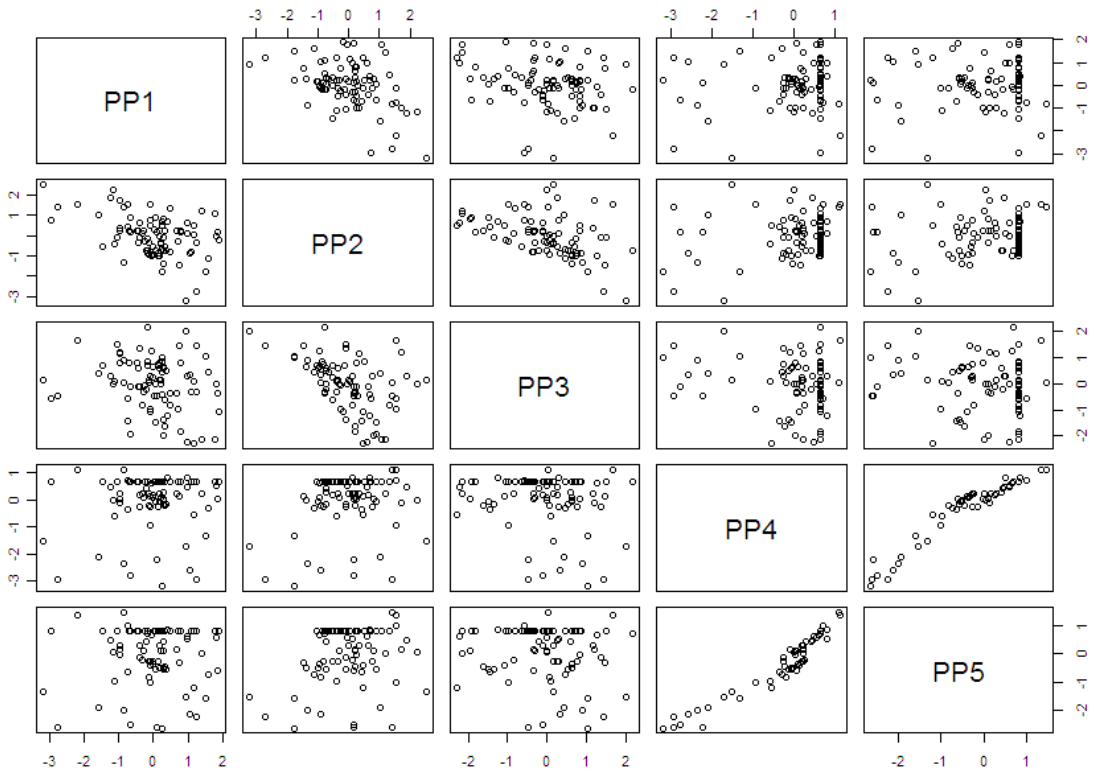
```
> lanmeta
 [1] 660.665138 219.687670 172.458187 135.858749 111.646370 103.386272  75.179720
 [8]  62.791287  54.807361  47.183950  36.607632  25.734716  20.582551  18.691936
[15]  16.676795  15.330995   8.516799   7.448019   5.500896   4.886180   2.358778
```

k<-max(lanmeta)/min(lanmeta)

得到 K= 280.0878>>30，故可以认为 X 有显著的共线性。

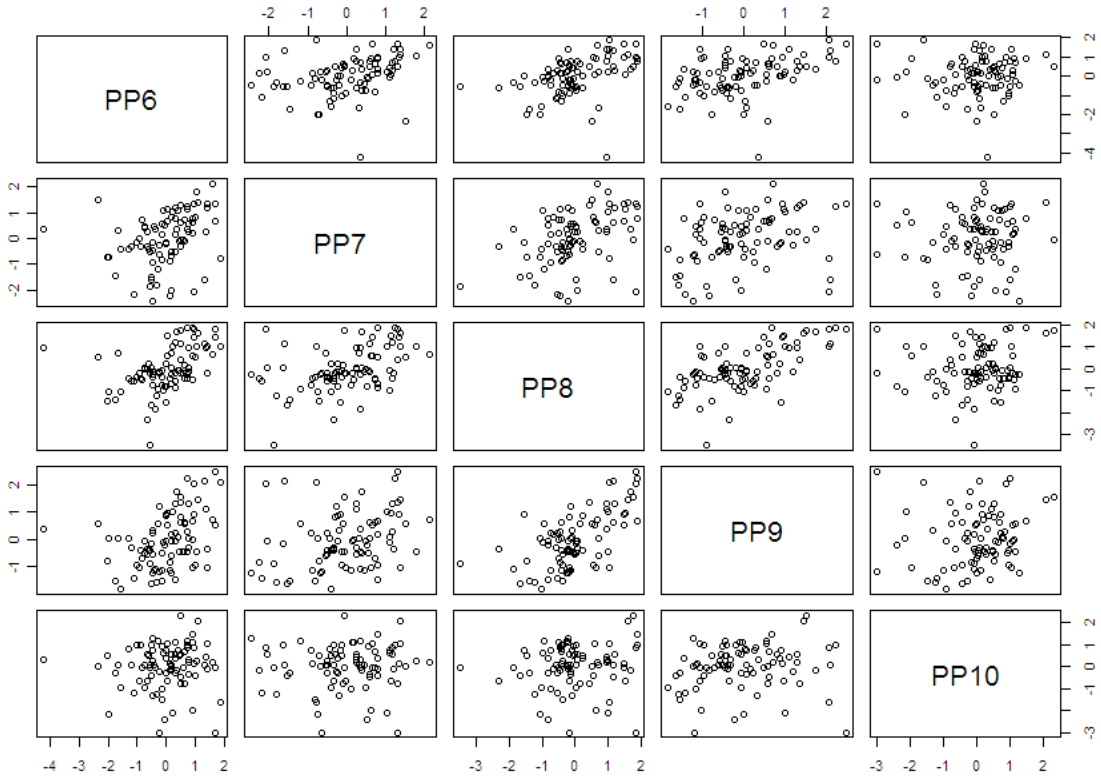
下面考察几幅 X 各自变量之间的散点图：

>pairs(yy[,1:5])

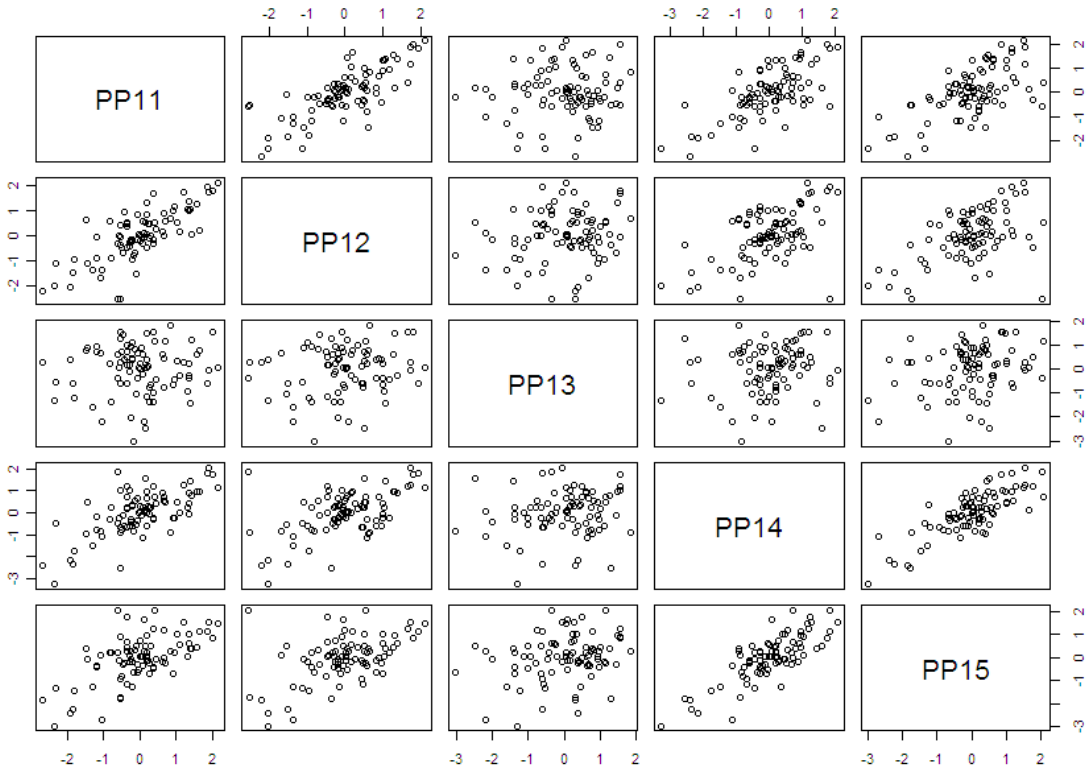


可见，PP4 和 PP5 有很明显的共线性。

> pairs(yy[,6:10])

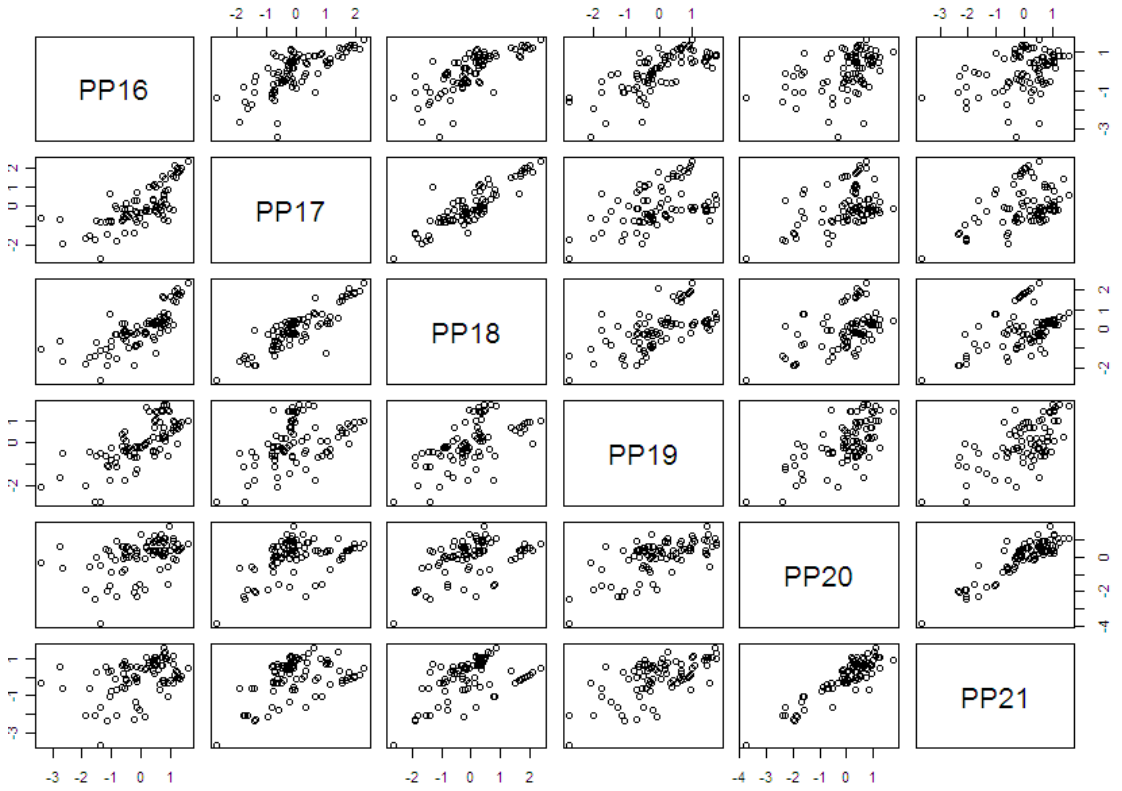


> pairs(yy[,11:15])



由上图，易发现 PP11,PP12; PP14,PP15 有很强的共线性，PP11,PP14;PP11,PP15; PP12,PP14 也略有共线性

> pairs(yy[,16:21])



由上图，PP20,PP21;PP17,PP18 有很强的共线性，PP16,PP17,PP18；PP19,PP20 也略有共线性。

以 O1 为因变量做回归

检验是否有异常值：

所谓异常值就是如果一个案例不遵从某个模型，但其余数据遵从这个模型，则该案例称为异常值。

对于异常值的检验，通常有学生化内残差和学生化外残差，DFFITS 等方法，我们这里采用计算 COOK 距离的办法来寻找异常值，即当 Di>1 时，我们认为第 i 个案例是异常值。COOK 距离的定义式$D_i = \frac{\beta_i - \hat{\beta}^T (X^T X)(\beta_i - \hat{\beta})}{p' \hat{\sigma}^2}$，$D_i$值大的案例被删除后，将引起分析的实质性改变。典型的，具有最大D_i值的案例，或在打的数据集合中，前几个具有最大D_i值的案例是我们感兴趣的。通过与置信域的类比，我们得到测定D_i的一个方法。如果D_i恰好等于自由度为 P'和 n-p'的 F 分布的分位点，则删除第 i 个案例将把β估计移至基于全部数据集合的$(1-\alpha)*100\%$置信域的边缘。大多数 F 分布的 50%分位点接近于 1，所以D_i=1 的一个值将把估计移至约 50%的置信域的边缘。这是一个潜在的重要变化。如果最大的D_i明显小于 1，则删除一个案例不

会太多地改变对β的估计。为更好地研究一个案例的影响，我们必须删除大的D_i的案例，并重新计算分析，看它的哪些方面确实地改变了。

```
##去除异常值，计算 COOK 距离 Di
lm.bb <- lm(O1~
PP1+PP2+PP3+PP4+PP5+PP6+PP7+PP8+PP9+PP10+PP11+PP12+PP13+PP14+PP15+PP16+PP17+P
P18+PP19+PP20+PP21, data=yy)
D<-rep(0,86)
for(i in 1:86){
    lm.bbnew<-lm(O1~
    PP1+PP2+PP3+PP4+PP5+PP6+PP7+PP8+PP9+PP10+PP11+PP12+PP13+PP14+PP15+PP16+PP
    17+PP18+PP19+PP20+PP21,
    data=yy[-i,])
    theta<-21*t(lm.bb$residuals)%*%lm.bb$residuals/(86-21)
    D[i]<-t(coef(lm.bbnew)[-1]-coef(lm.bb)[-1])%*%XX%*%(coef(lm.bbnew)[-1]-coef(lm.bb)[-1])
/theta
}
```

max(D)= 0.1277694<<1

所以，没有异常值。

做线性回归：

先考虑 O1 对于 PP1~PP21 的回归，lm.bb <- lm(O1~
PP1+PP2+PP3+PP4+PP5+PP6+PP7+PP8+PP9+PP10+PP11+PP12+PP13+PP14+PP15+PP16+PP17+P
P18+PP19+PP20+PP21, data=yy)

得到回归系数矩阵如下：

```
Coefficients:
             Estimate Std. Error   t value Pr(>|t|)
(Intercept) -3.583e-16  8.790e-02 -4.08e-15   1.0000
PP1          1.523e-01  1.582e-01     0.963   0.3393
PP2          1.186e-01  1.661e-01     0.714   0.4779
PP3          3.463e-01  1.497e-01     2.314   0.0239 *
PP4         -2.375e-01  3.707e-01    -0.641   0.5240
PP5          4.708e-02  3.718e-01     0.127   0.8996
PP6         -1.582e-01  1.146e-01    -1.380   0.1725
PP7         -1.141e-01  1.113e-01    -1.025   0.3090
PP8         -1.191e-01  1.402e-01    -0.849   0.3989
PP9          4.034e-01  1.529e-01     2.639   0.0104 *
PP10        -3.733e-01  1.657e-01    -2.253   0.0277 *
PP11         3.763e-01  1.896e-01     1.984   0.0515 .
PP12        -2.805e-01  1.886e-01    -1.487   0.1420
PP13         4.449e-01  1.530e-01     2.908   0.0050 **
PP14         1.247e-01  2.062e-01     0.605   0.5476
PP15        -3.970e-02  1.918e-01    -0.207   0.8367
PP16         5.656e-02  2.370e-01     0.239   0.8121
PP17        -2.573e-01  2.558e-01    -1.006   0.3183
PP18        -5.226e-02  2.480e-01    -0.211   0.8338
PP19        -7.692e-02  2.070e-01    -0.372   0.7115
PP20         3.637e-01  2.567e-01     1.417   0.1614
PP21        -1.358e-02  2.423e-01    -0.056   0.9555
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

如图，在 t 检验下，系数由显著性的有 PP3，PP9,PP10,PP13

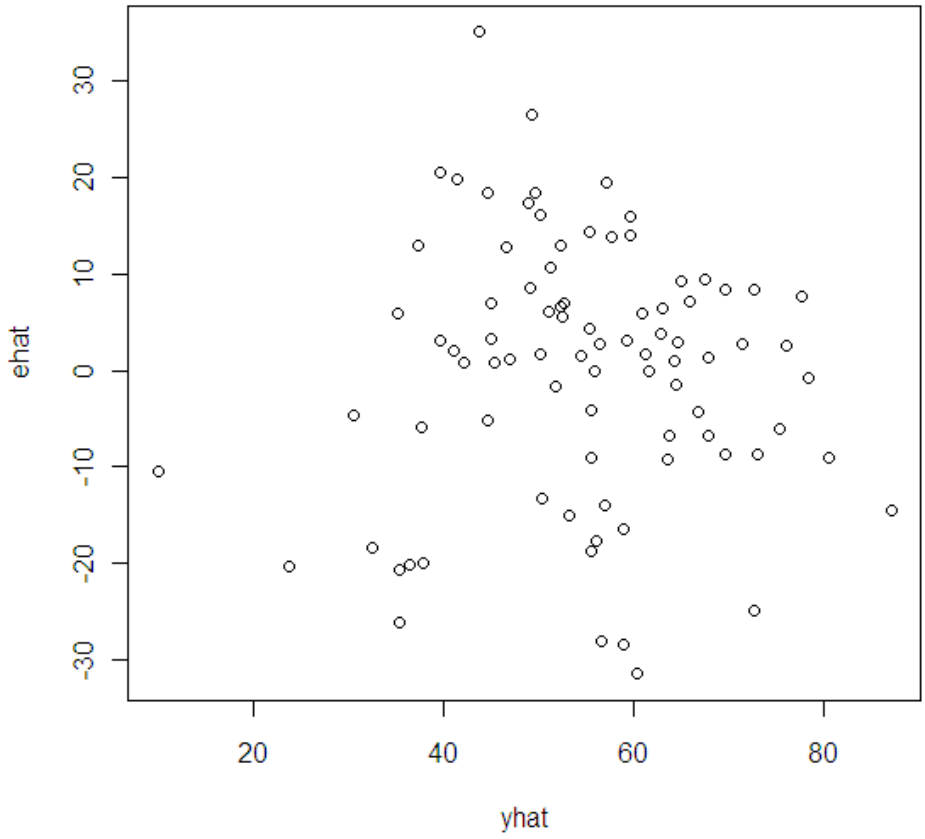
O1 与各自变量的协方差矩阵如下：

```
> cov(yy[,1:21],yy[,22])
              [,1]
PP1  6391.92138167
PP2    -7.14414876
PP3     8.38289281
PP4   -43.14003180
PP5   -96.00543352
PP6    -2.19735294
PP7    -1.42574142
PP8     8.54051694
PP9    55.22495655
PP10   -0.04571248
PP11    1.33403740
PP12   -4.84711828
PP13    6.17214279
PP14    3.53517357
PP15    4.92341795
PP16   -3.00973269
PP17   -6.42973792
PP18   -5.58554036
PP19    0.60763187
PP20    3.83019945
PP21    4.08221601
```

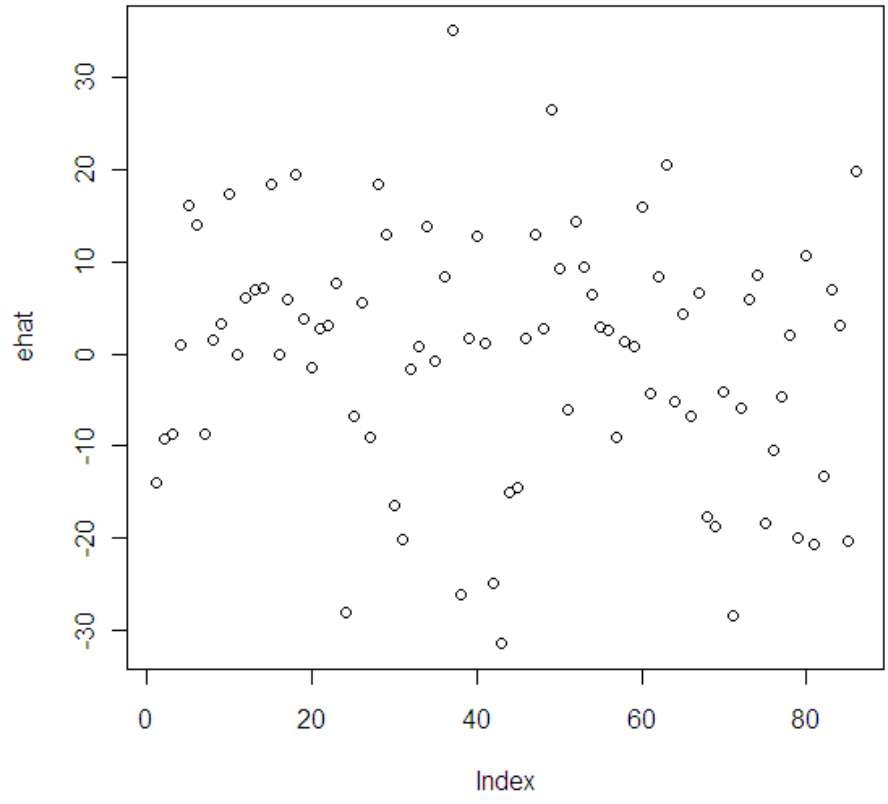
计算回归残差，并做残差图：

```
> lm.bb$residuals
           1            2            3            4            5            6
-13.99541143  -9.22677299  -8.64909720   1.01134221  16.23291922  14.05667378
           7            8            9           10           11           12
 -8.73940177   1.47605445   3.24895986  17.37195931  -0.03979749   6.08515006
          13           14           15           16           17           18
  7.03520750   7.19594782  18.50286152  -0.05500886   5.99271030  19.45709165
          19           20           21           22           23           24
  3.74325382  -1.51106264   2.82537679   3.07210213   7.62328732 -28.09906405
          25           26           27           28           29           30
 -6.70910575   5.65833967  -9.06870322  18.52660266  12.99695936 -16.39631258
          31           32           33           34           35           36
-20.11628281  -1.67064962   0.77447280  13.86375724  -0.74958166   8.37646997
          37           38           39           40           41           42
 35.13124466 -26.18265532   1.65142874  12.79024074   1.22544210 -24.86185912
          43           44           45           46           47           48
-31.50899571 -15.06681910 -14.44808326   1.63632297  13.07254516   2.83516533
          49           50           51           52           53           54
 26.61076239   9.33651299  -5.98983558  14.48194268   9.43593031   6.45059405
          55           56           57           58           59           60
  2.98936059   2.52620720  -9.10908899   1.43663453   0.89075950  15.96196754
          61           62           63           64           65           66
 -4.21006710   8.39925627  20.60849253  -5.09925736   4.35085407  -6.67955752
          67           68           69           70           71           72
  6.64906498 -17.65647722 -18.79287028  -4.14501555 -28.47158715  -5.79103087
          73           74           75           76           77           78
  5.93929674   8.59851059 -18.45510389 -10.46127346  -4.63411181   2.05131659
          79           80           81           82           83           84
-19.91109360  10.70134139 -20.71127330 -13.34605724   7.04119442   3.13639393
          85           86
-20.32118651  19.81326953
```

```
ehat<-rep(0,86)

yhat<-rep(0,86)

for(i in 1:86){

    ehat[i]<-yy[i,22]-t(lm.bb$coef[-1])%*%t(yy[i,1:21])-lm.bb$coef[1]

    yhat[i]<-t(lm.bb$coef[-1])%*%t(yy[i,1:21])+lm.bb$coef[1]

}

plot(yhat,ehat)
```

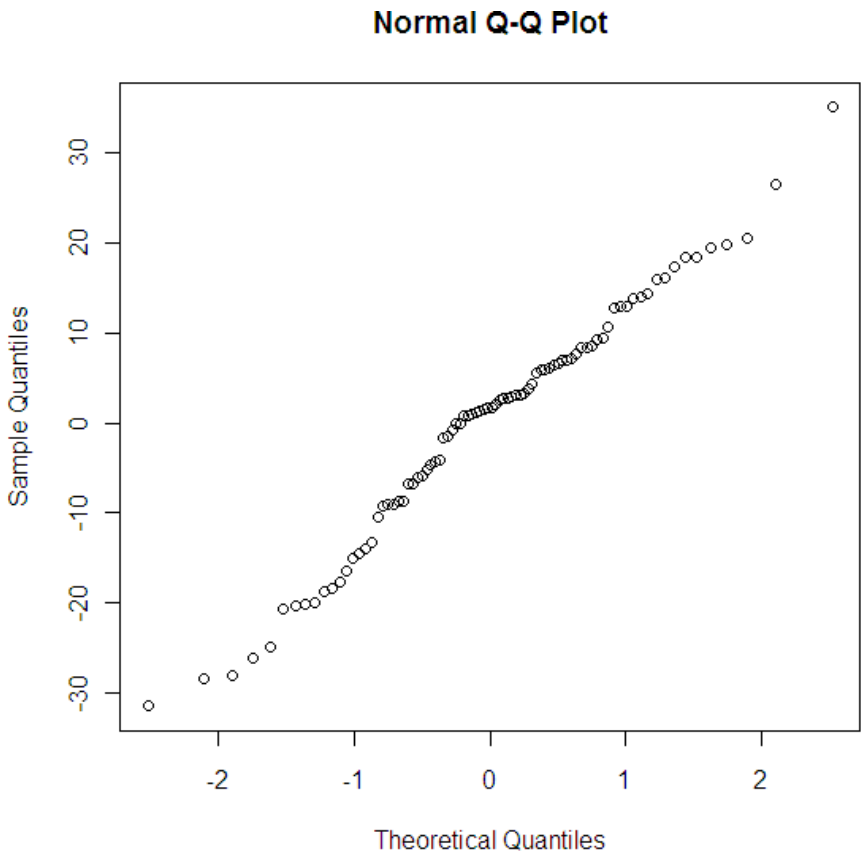

ehat
30
20
10
0
-10
-20
-30
20
40
60
80
yhat

Plot(ehat)


ehat
30
20
10
0
-10
-20
-30
0
20
40
60
80
Index

残差正态性检验：

qqnorm(ehat)




```
> shapiro.test(ehat)

        Shapiro-Wilk normality test

data:  ehat
W = 0.9821, p-value = 0.2812
```

所以，认为残差是正态性的。

变量选取：

用 AIC 的标准选取变量：

(AIC 即 AKAIK INFORMATION CRITERION, $\mathrm{AIC(M)} = -2\mathrm{l(M)} + 2\mathrm{p} = \frac{1}{\sigma^2}\mathrm{RSS(M)} + 2\mathrm{p}$

我们选择测试误差 AIC 最小的模型)

用 Step(lm.bb)得到模型为，

```
Step:  AIC=-31.99
O1 ~ PP1 + PP3 + PP4 + PP6 + PP9 + PP10 + PP11 + PP12 + PP13 +
    PP17 + PP20

       Df Sum of Sq     RSS     AIC
<none>               44.850 -31.988
- PP1   1     1.261  46.112 -31.602
- PP4   1     1.643  46.493 -30.893
- PP6   1     2.108  46.959 -30.037
- PP20  1     2.380  47.230 -29.541
- PP12  1     3.076  47.926 -28.283
- PP10  1     3.759  48.609 -27.066
- PP17  1     3.772  48.622 -27.043
- PP9   1     4.095  48.945 -26.475
- PP3   1     4.394  49.244 -25.950
- PP11  1     5.767  50.617 -23.586
- PP13  1     6.958  51.809 -21.584

Call:
lm(formula = O1 ~ PP1 + PP3 + PP4 + PP6 + PP9 + PP10 + PP11 +     PP12 + PP13 + PP17 + PP20, data = yy)

Coefficients:
(Intercept)          PP1          PP3          PP4          PP6          PP9
 -4.055e-17    1.607e-01    2.466e-01   -1.550e-01   -1.783e-01    3.040e-01
       PP10         PP11         PP12         PP13         PP17         PP20
 -3.518e-01    4.371e-01   -3.477e-01    4.504e-01   -3.003e-01    2.385e-01
```

验证选取变量 PP1,PP3,PP4,PP6,PP9,PP10,PP11,PP12,PP13,PP17,PP20 的共线性。

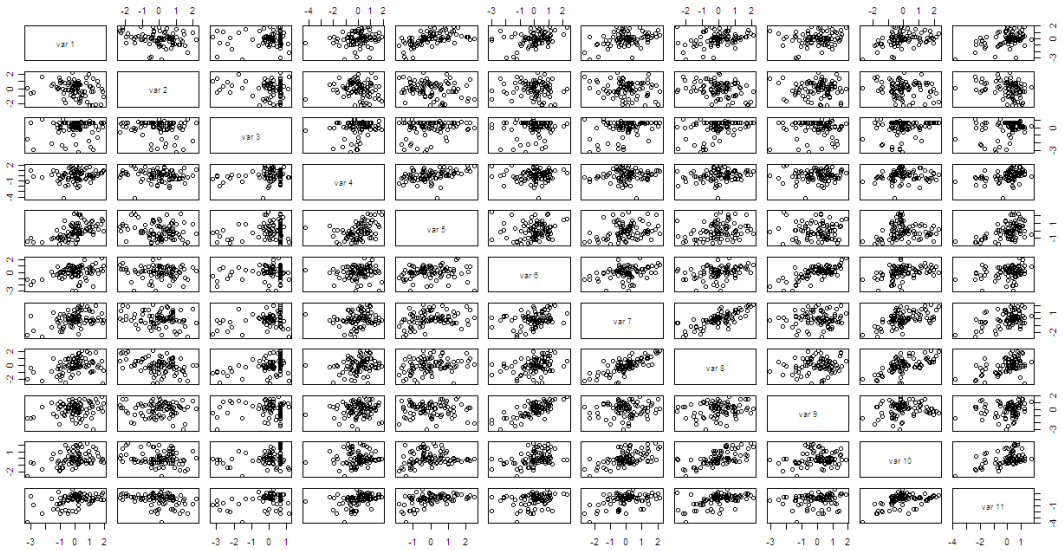
```
newmatrix<-cbind(yyy$PP1,yyy$PP3, yyy$PP4,yyy$PP6 ,yyy$PP9, yyy$PP10, yyy$PP11, yyy$PP12,
yyy$PP13,yyy$PP17,yyy$PP20)


XXnew<-array(0,c(11,11))
for (i in 1:11){  ##计算 t(X)*X 矩阵
      for(j in 1:11){
            XXnew[i,j]<-newmatrix[,i]%*%newmatrix[,j]
      }
}
lanmetanew<-eigen(XXnew)$val
knew<-max(lanmetanew)/min(lanmetanew)
```

得到：

```
> knew
[1] 24.93828<30
```

所以选取的变量已经没有明显的共线性。

画 11 个变量的相关系数图：



可以看出已经没有明显的共线性。

加入交叉项和二次向后，得到线性模型为：

```
lm.bb11<-lm(O1~
PP3+PP9+PP10+PP11+PP12+PP13+PP17+yyy[,39+2*21+3]+yyy[,39+9+2*21]+yyy[,39+10+2*21]+
yyy[,39+11+2*21]+yyy[,39+12+2*21]+yyy[,39+13+2*21]+yyy[,39+17+2*21]+yyy[,39+9+8*21]+yy
y[,39+9+9*21]+yyy[,39+9+10*21]+yyy[,39+9+11*21]+yyy[,39+9+12*21]+yyy[,39+9+16*21]+yyy[,
39+10+9*21]+yyy[,39+10+10*21]+yyy[,39+10+11*21]+yyy[,39+10+12*21]+yyy[,39+10+16*21]+y
yy[,39+11+10*21]+yyy[,39+11+11*21]+yyy[,39+11+12*21]+yyy[,39+11+16*21]+yyy[,39+12+11*
21]+yyy[,39+12+12*21]+yyy[,39+12+16*21]+yyy[,39+13+12*21]+yyy[,39+13+16*21]+yyy[,39+17
+16*21], data=yyy)
```

```
Coefficients:
                              Estimate Std. Error   t value Pr(>|t|)
(Intercept)                 -1.761e-14  8.062e-02 -2.18e-13  1.00000
PP3                          1.296e+01  5.684e+00     2.280  0.02692 *
PP9                          5.973e+00  7.133e+00     0.837  0.40639
PP10                        -1.219e+01  7.078e+00    -1.722  0.09127 .
PP11                         7.968e+00  7.857e+00     1.014  0.31536
PP12                         1.235e+01  1.134e+01     1.089  0.28118
PP13                         1.138e+01  6.480e+00     1.756  0.08522 .
PP17                         1.223e-01  8.570e+00     0.014  0.98867
yyy[, 39 + 2 * 21 + 3]      -4.579e-01  4.327e-01    -1.058  0.29499
yyy[, 39 + 9 + 2 * 21]      -7.043e-01  5.092e-01    -1.383  0.17277
yyy[, 39 + 10 + 2 * 21]      6.596e+00  6.550e+00     1.007  0.31875
yyy[, 39 + 11 + 2 * 21]     -2.502e+00  6.763e+00    -0.370  0.71299
yyy[, 39 + 12 + 2 * 21]     -1.774e+01  6.439e+00    -2.755  0.00817 **
yyy[, 39 + 13 + 2 * 21]     -5.172e+00  4.873e+00    -1.061  0.29369
yyy[, 39 + 17 + 2 * 21]      7.403e+00  5.702e+00     1.298  0.20015
yyy[, 39 + 9 + 8 * 21]      -5.821e-01  7.666e-01    -0.759  0.45123
yyy[, 39 + 9 + 9 * 21]       5.601e+00  4.515e+00     1.240  0.22062
yyy[, 39 + 9 + 10 * 21]      1.366e+00  6.505e+00     0.210  0.83452
yyy[, 39 + 9 + 11 * 21]     -3.134e+00  5.901e+00    -0.531  0.59772
yyy[, 39 + 9 + 12 * 21]     -8.920e+00  4.342e+00    -2.054  0.04519 *
yyy[, 39 + 9 + 16 * 21]      4.072e-01  6.776e+00     0.060  0.95232
yyy[, 39 + 10 + 9 * 21]      6.526e+00  1.196e+01     0.546  0.58768
yyy[, 39 + 10 + 10 * 21]    -1.498e+01  2.363e+01    -0.634  0.52892
yyy[, 39 + 10 + 11 * 21]     3.176e+01  3.267e+01     0.972  0.33556
yyy[, 39 + 10 + 12 * 21]     1.116e+00  1.604e+01     0.070  0.94480
yyy[, 39 + 10 + 16 * 21]    -1.010e+01  1.661e+01    -0.608  0.54605
yyy[, 39 + 11 + 10 * 21]     1.456e+01  1.526e+01     0.954  0.34472
yyy[, 39 + 11 + 11 * 21]    -3.159e+01  2.626e+01    -1.203  0.23465
yyy[, 39 + 11 + 12 * 21]     3.780e+00  1.639e+01     0.231  0.81857
yyy[, 39 + 11 + 16 * 21]    -9.500e-02  1.630e+01    -0.006  0.99537
yyy[, 39 + 12 + 11 * 21]     1.511e+01  1.109e+01     1.363  0.17893
yyy[, 39 + 12 + 12 * 21]    -8.076e+00  1.957e+01    -0.413  0.68162
yyy[, 39 + 12 + 16 * 21]    -3.928e+01  2.272e+01    -1.729  0.08992 .
yyy[, 39 + 13 + 12 * 21]    -9.013e+00  7.869e+00    -1.145  0.25747
yyy[, 39 + 13 + 16 * 21]     2.521e+00  9.658e+00     0.261  0.79517
yyy[, 39 + 17 + 16 * 21]     2.473e+01  1.421e+01     1.741  0.08792 .
```

用 AIC 为标准，选取最终变量：

step(lm.bb11)

```
Step:  AIC=-51.41
O1 ~ PP3 + PP9 + PP10 + PP11 + PP12 + PP13 + yyy[, 39 + 2 * 21 +
    3] + yyy[, 39 + 9 + 2 * 21] + yyy[, 39 + 12 + 2 * 21] + yyy[,
    39 + 17 + 2 * 21] + yyy[, 39 + 9 + 9 * 21] + yyy[, 39 + 9 +
    12 * 21] + yyy[, 39 + 10 + 9 * 21] + yyy[, 39 + 11 + 11 *
    21] + yyy[, 39 + 12 + 11 * 21] + yyy[, 39 + 12 + 16 * 21] +
    yyy[, 39 + 13 + 12 * 21] + yyy[, 39 + 17 + 16 * 21]
```

```
Coefficients:
            (Intercept)                      PP3                      PP9                     PP10
              1.241e-14                1.364e+01                2.526e+00                -8.986e+00
                   PP11                     PP12                     PP13    yyy[, 39 + 2 * 21 + 3]
              7.763e+00                1.643e+01                8.576e+00                -6.120e-01
  yyy[, 39 + 9 + 2 * 21]   yyy[, 39 + 12 + 2 * 21]   yyy[, 39 + 17 + 2 * 21]    yyy[, 39 + 9 + 9 * 21]
             -5.198e-01               -2.226e+01                1.028e+01                 6.525e+00
 yyy[, 39 + 9 + 12 * 21]   yyy[, 39 + 10 + 9 * 21]   yyy[, 39 + 11 + 11 * 21]  yyy[, 39 + 12 + 11 * 21]
             -8.360e+00                8.181e+00               -1.456e+01                 1.026e+01
yyy[, 39 + 12 + 16 * 21]   yyy[, 39 + 13 + 12 * 21]   yyy[, 39 + 17 + 16 * 21]
             -3.013e+01               -7.306e+00                1.540e+01
```

lm.bb12<-lm(formula = O1 ~ PP3 + PP9 + PP10 + PP11 + PP12 + PP13 + yyy[, 39 + 2 * 21 + 3] + yyy[, 39 + 9 + 2 * 21] + yyy[, 39 + 12 + 2 * 21] + yyy[, 39 + 17 + 2 * 21] + yyy[, 39 + 9 + 9 * 21] + yyy[, 39 + 9 + 12 * 21] + yyy[, 39 + 10 + 9 * 21] + yyy[, 39 + 11 + 11 * 21] + yyy[, 39 + 12 + 11 * 21] + yyy[, 39 + 12 + 16 * 21] + yyy[, 39 + 13 + 12 * 21] + yyy[, 39 + 17 + 16 * 21], data = yyy)

```
Coefficients:
                            Estimate Std. Error  t value Pr(>|t|)
(Intercept)                1.241e-14  7.264e-02 1.71e-13 1.000000
PP3                        1.364e+01  3.795e+00    3.596 0.000613 ***
PP9                        2.526e+00  1.813e+00    1.394 0.168071
PP10                      -8.986e+00  3.342e+00   -2.688 0.009052 **
PP11                       7.763e+00  3.903e+00    1.989 0.050751 .
PP12                       1.643e+01  5.135e+00    3.200 0.002103 **
PP13                       8.576e+00  3.407e+00    2.517 0.014238 *
yyy[, 39 + 2 * 21 + 3]    -6.120e-01  3.356e-01   -1.824 0.072644 .
yyy[, 39 + 9 + 2 * 21]    -5.198e-01  3.365e-01   -1.545 0.127091
yyy[, 39 + 12 + 2 * 21]   -2.226e+01  4.186e+00   -5.318  1.3e-06 ***
yyy[, 39 + 17 + 2 * 21]    1.028e+01  3.544e+00    2.902 0.005009 **
yyy[, 39 + 9 + 9 * 21]     6.525e+00  3.149e+00    2.072 0.042135 *
yyy[, 39 + 9 + 12 * 21]   -8.360e+00  2.756e+00   -3.034 0.003435 **
yyy[, 39 + 10 + 9 * 21]    8.181e+00  3.315e+00    2.468 0.016134 *
yyy[, 39 + 11 + 11 * 21] -1.456e+01  7.792e+00   -1.868 0.066070 .
yyy[, 39 + 12 + 11 * 21]   1.026e+01  5.974e+00    1.718 0.090384 .
yyy[, 39 + 12 + 16 * 21] -3.013e+01  1.105e+01   -2.727 0.008161 **
yyy[, 39 + 13 + 12 * 21] -7.306e+00  3.358e+00   -2.176 0.033097 *
yyy[, 39 + 17 + 16 * 21]   1.540e+01  6.086e+00    2.530 0.013776 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6737 on 67 degrees of freedom
Multiple R-squared: 0.6464,     Adjusted R-squared: 0.5514
F-statistic: 6.805 on 18 and 67 DF,  p-value: 2.767e-09
```

-372.5895

模型的残差平方和为：

> sum(lm.bb12$residuals^2)

[1] 30.40677

用 BOX-COX 变换来分析数据。

对 yyy$O1 做变换为 yyy$O1^λ (λ =-2, -1.5, -1, -0.5, 0.5, 1, 1.5, 2), LOG(yyy$O1) 当λ =0

选取λ 使得L(λ)最大，其中 $L(\lambda) = n\ln(|\lambda|) - \frac{n}{2}\ln(RSS_\lambda) + n(\lambda-1)\ln(GM(yyy\$O1))$，当λ ！

=0；$L(\lambda) = -\frac{n}{2}\ln(RSS_\lambda) - n\ln\big(GM(yyy\$O1)\big)$， 当λ $= 0.$

```
lan<-c(-2,-1.5,-1,-0.5,0,0.5,1,1.5,2)
rlan<-rep(0,9)
for(i in 1:9){
    lm.bbb<-lm(O1^lan[i]~ PP1+PP3+PP4+PP6+PP9+PP10+PP11+PP12+PP13+PP17+PP20,
data=yyy)
    rlan[i]<-sum(lm.bbb$residuals^2)
}
lm.bbb<-lm(log(abs(O1))~ PP1+PP3+PP4+PP6+PP9+PP10+PP11+PP12+PP13+PP17+PP20,
data=yyy)
rlan[5]<-rlan[i]<-sum(lm.bbb$residuals^2)

i<-0
L<-rep(0,9)
for(i in i:9){
    if(i!=5){
        L[i]<-86*log(abs(lan[i]))-43*log(rlan[i])+86*(lan[i]-1)*log(prod(yyy$O1))
    }
    else{
        L[i]<--43*log(rlan[i])-86*log(prod(yyy$O1))
    }
}
```

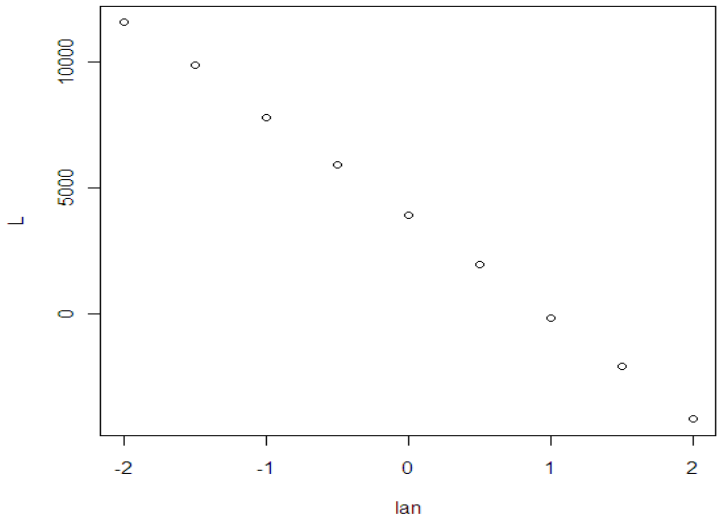
得到结果为：

> L

[1] 11560.1075 9852.0355 7780.0476 5927.4417 3897.8605 1938.0658

[7] -163.5430 -2099.7100 -4193.4034

画 L 关于λ 的图，发现 L 关于λ 下降，所以最大值应该在小于-2 的地方，所以 BOX-COX 变化不适用。



##

2.【LASSO 回归】

LASSO 回归是对传统最小二乘回归的改进，具体来说就是对系数的估计增加一个限制或者惩罚函数，即$\widehat{\beta^{lasso}} = \mathbf{argmin}_{\beta} \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$

我们可以通过将预测子标准化来对常量β_0重新参数化。需要注意的是，惩罚项$\sum_{j=1}^{p} |\beta_j|$式的解在 Yi 上非线性，并需要使用二次规划算法来计算它们。由于这个约束的限制的特性，当λ 很大时，将导致一部分系数收缩到 0.这样 Lasso 实际上做了一个变量子集的选择，这与之前的变量选择的思想是一样的。Lasso 回归是目前处理多重共线性的主要方法之一,相对于其他方法,更容易产生稀疏解,在参数估计的同时实现变量选择,因而可以用来解决检验中的多重共线性问题,以提高检验的效率。

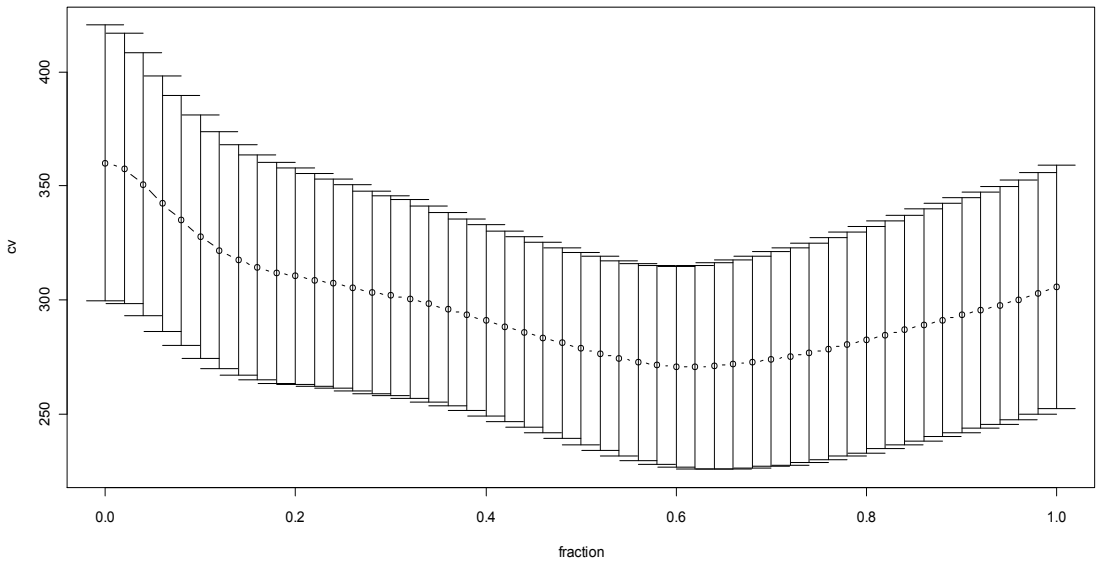
回到项目中，用 Lasso 方法回归，先用 10 折交叉验证画出 CV 误差曲线图，提取 CV 误

差值最小的收缩系数，之后再画出回归系数收缩图。然后，用前面取出的最小 CV 误差的收缩系数做 LASSO 回归得出回归系数。下面是 18 个回归的 CV 误差图，系数收缩图，回归系数，及相应的 R code。（lasso 回归，使用了 lars 的软件包）

K 折交叉验证是用于估计模型的调整参数λ ，思想与 jackknife 相似，将数据分成容量大致相等的 K 份，对每个 k=1，2…K, 取调整参数为λ ，每次留出第 K 份数据，其余 K-1 份数据用于训练，得到参数的估计$\widehat{\boldsymbol{\beta}}^{-\mathbf{k}}(\boldsymbol{\lambda})$，并计算第 K 份数据的预测误差：$\mathbf{E_k}(\boldsymbol{\lambda}) = \sum_{\mathbf{i}\in\mathbf{k}^{\mathbf{th}}}(\mathbf{y_i} - \mathbf{x_i}\widehat{\boldsymbol{\beta}}^{-\mathbf{k}}(\boldsymbol{\lambda}))^2$, 对多个不同的λ ，计算其相对应的误差 CV(λ)，最佳模型为 CV(λ) 最小的模型。在子集选择的例子中，λ 为子集的容量，$\widehat{\boldsymbol{\beta}}^{-\mathbf{k}}(\boldsymbol{\lambda})$为子集容量为λ 的最佳子集的系数，$\mathbf{E_k}(\boldsymbol{\lambda})$为该最佳子集的测试误差的一个估计。K−折交叉验证的测试误差的估计为 CV（λ ）=$\frac{\mathbf{1}}{\mathbf{K}}\sum_{\mathbf{k=1}}^{\mathbf{K}}\mathbf{E_k}(\boldsymbol{\lambda})$.

对 O1 进行 LASSO 回归

```
x<-cbind(PP1,PP2,PP3,PP4,PP5,PP6,PP7,PP8,PP9,PP10,PP11,PP12,PP13,PP14,PP15,PP16,PP17,PP
18,PP19,PP20,PP21,O1)
y<-x[,22]
x<-x[,-22]
library(lars)
cv.lars(x,y,K=10,fraction=seq(0,1,0.02),type="lasso")   ###10 折交叉验证画 CV 误差图
```



如上图，取收缩系数为 0.6。

首先提取出最小二乘方法系数的绝对值之和，再编写一个 lasso 回归的函数，对于小于 1-e03 的系数，取为 0。

```
reg<-function(i,t) {                    #提取普通最小二乘系数
    y<-x[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,i+21)]
    lm.reg<-lm(t~PP1+PP2+PP3+PP4+PP5+PP6+PP7+PP8+PP9+PP10+PP11+PP12+PP13+PP14+P
    P15+PP16+PP17+PP18+PP19+PP20+PP21,data=y)
    summary(lm.reg)
    coef<-lm.reg$coefficients
    sum(abs(coef))
}
coefsum[1]<-reg(1,01)
coefsum[2]<-reg(2,02)
coefsum[3]<-reg(3,03)
coefsum[4]<-reg(4,04)
coefsum[5]<-reg(5,05)
coefsum[6]<-reg(6,06)
coefsum[7]<-reg(7,07)
coefsum[8]<-reg(8,08)
coefsum[9]<-reg(9,09)
coefsum[10]<-reg(10,010)
coefsum[11]<-reg(11,011)
coefsum[12]<-reg(12,012)
coefsum[13]<-reg(13,013)
coefsum[14]<-reg(14,014)
coefsum[15]<-reg(15,015)
coefsum[16]<-reg(16,016)
coefsum[17]<-reg(17,017)
coefsum[18]<-reg(18,018)
```

```
my.lasso<-function(y,x,k)        #lasso 回归函数
{        xm <- apply(x,2,mean)
         ym <- mean(y)
         y <- y - ym
         x <- t( t(x) - xm ) #中心化
         ss <- function (b)
         {
                t( y - x %*% b ) %*% ( y - x %*% b ) + k * sum(abs(b))
         }
         b <- nlm(ss, rep(0,dim(x)[2]))$estimate
         coef<-c(ym,b)
         coef<-matrix(coef,22,1)
         for (i in 1:22)  {
           if (coef[i,]<=1e-03) coef[i]<-0
         }
         coef
}

x<-cbind(PP1,PP2,PP3,PP4,PP5,PP6,PP7,PP8,PP9,PP10,PP11,PP12,PP13,PP14,PP15,PP16
,PP17,PP18,PP19,PP20,PP21,01)
y<-x[,22]
x<-x[,-22]
coef.o1<-my.lasso(y,x,0.6*coefsum[1])
coef.o1
```

```
> coef.o1
               [,1]
 [1,] 54.664186047
 [2,]  0.001552216
 [3,]  0.000000000
 [4,]  2.570786898
 [5,]  0.088943095
 [6,]  0.000000000
 [7,]  0.000000000
 [8,]  0.000000000
 [9,]  0.000000000
[10,]  0.702464841
[11,]  0.000000000
[12,]  2.276442694
[13,]  0.000000000
[14,]  3.161132572
[15,]  1.288524126
[16,]  1.264564015
[17,]  0.000000000
[18,]  0.000000000
[19,]  0.000000000
[20,]  0.000000000
[21,]  2.384268725
[22,]  2.032665650
```

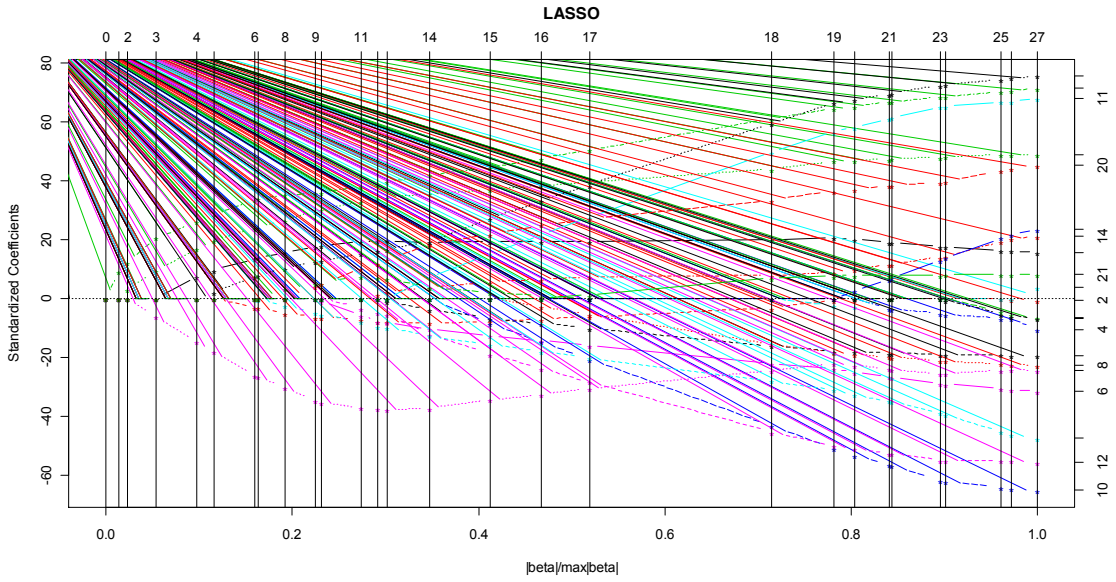
计算残差平方和：

```
x<-cbind(PP1,PP2,PP3,PP4,PP5,PP6,PP7,PP8,PP9,PP10,PP11,PP12,PP13,PP14,PP15,PP16,PP17,PP
18,PP19,PP20,PP21,O1)
sse.o1<-0
for (i in 1:86 ) {
      red<-x[i,22]-t(matrix(c(1,x[i,-22]),22,1))%*%coef.o1[,1]
      sse.o1<-sse.o1+red^2
}
sse.o1
```

```
> sse.o1
          [,1]
[1,] 33048863
>
```

再画出系数收缩图：

```
o1<-lars(x,y)
plot(o1)        #回归系数收缩图
```



3.【岭回归】

岭回归通过对其容量增加惩罚项来收缩回归系数。岭系数极小化罚残差平方和为：

$$\hat{\beta}^{ridge} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} {\beta_j}^2$$

其中，$\lambda \geq 0$是控制收缩量的复杂度参数：λ 值越大，收缩量就越大。系数向 0 收缩。

上式等价于$\hat{\beta}^{ridge} = \operatorname{argmin}_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$， 满足$\sum_{j=1}^{p} {\beta_j}^2 \leq s$

上式，解为$\hat{\beta}^{ridge} = （X^T X + \lambda I）^{-1} X^T Y$，仍然是 y 的线性组合。

这里需要注意的是，当矩阵$A = X^T X$奇异时，最小二乘的结果变的很坏。当矩阵 A 奇异时，一些特征值$\lambda_j = 0$，从而使得$E（\hat{\beta} - \beta）^2$很大，表示$\hat{\beta}$与β之间的偏差很大。同时$V（\hat{\beta} - \beta）^2$也很大，表示结果不稳定。岭回归在矩阵$A = X^T X$求逆之前，将一个正的常数加到 A 的对角线上，使得问题非奇异。

从贝叶斯的观点，正则项可以视为参数的先验。如果假设$y_i \sim N(x_i^T \beta, \sigma^2)$,并且每个$\beta_j$都符合先验分布$N（0, \tau^2）$，岭回归也可以被看做是从后验分布得到的。那么$\beta$的负 log 后验密度就是$RSS_{ridge}(\beta)$,其中$\lambda = \sigma^2 / \tau^2$.

>attach(yy)

>Library（MASS）

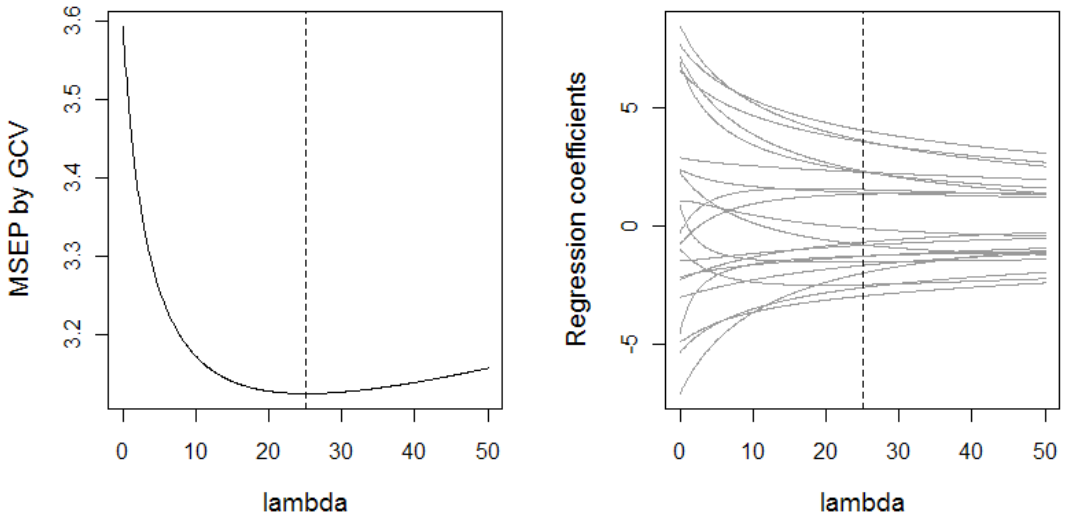
>Library（chemometrics）

【O1】

对于第一个相应变量 O1，先用 chemometrics 包里面的 plotRidge 函数考查岭回归的预测均方误差 MSEP

>res_ridge1<-plotRidge(O1~PP1+PP2+PP3+PP4+PP5+PP6+PP7+PP8+PP9+PP10+PP11+PP12+PP13+PP14+PP15+PP16+PP17+PP18+PP19+PP20+PP21,data=yy,lambda=seq(0,50,0.1))

得到如下结果：



可以看到最佳的岭回归参数 lambda 在 25 左右。可以用 MASS 包中的 select 函数精确计算：

>select(lm.ridge(O1~PP1+PP2+PP3+PP4+PP5+PP6+PP7+PP8+PP9+PP10+PP11+PP12+PP13+PP14+PP15+PP16+PP17+PP18+PP19+PP20+PP21,lambda=seq(0,50,0.5)))

输出结果为：

modified HKB estimator is 10.38313

modified L-W estimator is 24.97606

smallest value of GCV at 25

进一步精确确定 lambda 值：

>select(lm.ridge(O1~PP1+PP2+PP3+PP4+PP5+PP6+PP7+PP8+PP9+PP10+PP11+PP12+PP13+PP14+PP15+PP16+PP17+PP18+PP19+PP20+PP21,lambda=seq(24,26,0.05)))

输出结果为：

modified HKB estimator is 10.38313

modified L-W estimator is 24.97606

smallest value of GCV at 25.1

故岭回归参数取为 25.1：

>fit1<-lm.ridge(O1~PP1+PP2+PP3+PP4+PP5+PP6+PP7+PP8+PP9+PP10+PP11+PP12+PP13+PP14+

PP15+PP16+PP17+PP18+PP19+PP20+PP21,lambda=25.1)

岭回归结果为：

```
> fit1
                    PP1          PP2          PP3          PP4          PP5          PP6
35.754277650  0.001683697 -0.537104109  1.814563735 -0.089528438 -0.060775588 -0.493082555
         PP7          PP8          PP9         PP10         PP11         PP12         PP13
-0.411777037 -0.088413876  0.426086626 -1.219415739  1.660741801 -1.653780406  1.900840387
        PP14         PP15         PP16         PP17         PP18         PP19         PP20
 1.041925931  0.943837802 -0.051486011 -1.981326232 -1.922671321 -0.489088783  1.688881838
        PP21
 1.173600272
```

考察拟合的残差平方和：

> res1<-rep(0,86)

> for(i in 1:86){

+ res1[i]<-(res_ridge1$predicted[i]-O1[i])^2

+ }

>sum(res)

输出为：[1] 17262.72

同样的方法可以用于处理 O2~O18

O2-O18?

上述三种方法均适用于 PP2-PP18 的回归，鉴于方法相似，这里不一一列举。

参考文献：

《应用多元统计分析》

《应用线性回归分析》

《Elements of Statistical Learning》

《统计计算》

致谢

非常感谢沈致远同学和占翔同学对于岭回归部分和 LASSO 回归部分的精心分析和 Rcode 的实证计算。同时，特别感谢姚远老师对于论文选题的总体指导和对数据处理方法的提示，并及时给予我们鼓励和各种统计方法的指点。