6-7章实验

谭铭瑞

2023-05-26

library(haven)  
xt5\_9 <- read\_sav("C:/Users/Administrator/Desktop/xt5.9.sav")  
View(xt5\_9)  
library(car)

## 载入需要的程辑包：carData

lm5 = lm(y~x1+x2+x3+x4+x5+x6,data = xt5\_9)  
summary(lm5)

##   
## Call:  
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6, data = xt5\_9)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -374.18 -82.44 -3.00 91.05 237.52   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.348e+03 2.211e+03 0.610 0.551859   
## x1 -6.410e-01 1.669e-01 -3.840 0.001804 \*\*   
## x2 -3.170e-01 2.044e-01 -1.551 0.143216   
## x3 -4.127e-01 5.485e-01 -0.752 0.464294   
## x4 -2.110e-03 2.428e-02 -0.087 0.931962   
## x5 6.711e-01 1.280e-01 5.241 0.000125 \*\*\*  
## x6 -7.541e-03 8.128e-03 -0.928 0.369220   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 191.8 on 14 degrees of freedom  
## Multiple R-squared: 0.9962, Adjusted R-squared: 0.9946   
## F-statistic: 618 on 6 and 14 DF, p-value: 3.81e-16

#方差扩大因子法诊断多重共线性  
vif(lm5)#x1,x2,x3,x4,x5vif都大于10，都有较大可能存在多重共线性

## x1 x2 x3 x4 x5 x6   
## 319.484477 2636.564359 479.287849 27.177337 1860.726476 1.742651

cor(xt5\_9$x1,xt5\_9$x2)#比如说x1和x2相关性0.99高度相关

## [1] 0.9943469

#特征根  
XX = cor(xt5\_9[,2:7])  
kappa(XX,exact = TRUE)#k<100则多重共线性程度小，100<k<1000存在较强多重共线性，k>1000存在严重多重共线性

## [1] 21642.62

#先用第五章逐步回归选元  
lm5\_step = step(lm5,direction = "both")

## Start: AIC=226.27  
## y ~ x1 + x2 + x3 + x4 + x5 + x6  
##   
## Df Sum of Sq RSS AIC  
## - x4 1 278 515554 224.28  
## - x3 1 20834 536110 225.10  
## - x6 1 31684 546960 225.52  
## <none> 515276 226.27  
## - x2 1 88536 603812 227.60  
## - x1 1 542591 1057867 239.37  
## - x5 1 1011046 1526322 247.07  
##   
## Step: AIC=224.28  
## y ~ x1 + x2 + x3 + x5 + x6  
##   
## Df Sum of Sq RSS AIC  
## - x3 1 22801 538355 223.19  
## - x6 1 39639 555193 223.83  
## <none> 515554 224.28  
## + x4 1 278 515276 226.27  
## - x2 1 190111 705666 228.87  
## - x1 1 870111 1385665 243.04  
## - x5 1 1713154 2228708 253.02  
##   
## Step: AIC=223.19  
## y ~ x1 + x2 + x5 + x6  
##   
## Df Sum of Sq RSS AIC  
## - x6 1 31792 570147 222.39  
## <none> 538355 223.19  
## + x3 1 22801 515554 224.28  
## + x4 1 2245 536110 225.10  
## - x2 1 566230 1104585 236.28  
## - x1 1 847748 1386103 241.05  
## - x5 1 1704674 2243029 251.16  
##   
## Step: AIC=222.39  
## y ~ x1 + x2 + x5  
##   
## Df Sum of Sq RSS AIC  
## <none> 570147 222.39  
## + x6 1 31792 538355 223.19  
## + x3 1 14954 555193 223.83  
## + x4 1 11386 558761 223.97  
## - x2 1 534873 1105019 234.29  
## - x1 1 817120 1387267 239.06  
## - x5 1 1710882 2281029 249.51

summary(lm5\_step)#R=0.9958

##   
## Call:  
## lm(formula = y ~ x1 + x2 + x5, data = xt5\_9)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -372.27 -102.79 -7.78 157.94 313.69   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 874.58627 106.86620 8.184 2.67e-07 \*\*\*  
## x1 -0.61116 0.12382 -4.936 0.000125 \*\*\*  
## x2 -0.35304 0.08840 -3.994 0.000940 \*\*\*  
## x5 0.63669 0.08914 7.142 1.65e-06 \*\*\*  
## ---  
## 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

#再用多重共线性  
vif(lm5\_step)

## x1 x2 x5   
## 192.8706 541.4595 989.8335

#删除最大的x5  
lm5\_x1x2 = lm(y~x1+x2,data = xt5\_9)  
vif(lm5\_x1x2)#发现还是没法消除多重共线性

## x1 x2   
## 88.69731 88.69731

#换一个方法，先用多重共线性剔除  
lm5\_dropx2 = lm(y~x1+x3+x4+x5+x6,data = xt5\_9)#删除vif最大的x2  
vif(lm5\_dropx2)

## x1 x3 x4 x5 x6   
## 276.968819 306.617361 11.605489 632.895698 1.645146

lm5\_dropx2x5 = lm(y~x1+x3+x4+x6,data = xt5\_9)#删除x5  
vif(lm5\_dropx2x5)

## x1 x3 x4 x6   
## 160.512580 111.949275 11.507017 1.539699

lm5\_dropx2x5x1 = lm(y~+x3+x4+x6,data = xt5\_9)#删除x1  
vif(lm5\_dropx2x5x1)#此时方程已经消除多重共线性

## x3 x4 x6   
## 4.018087 4.508706 1.484981

summary(lm5\_dropx2x5x1)

##   
## Call:  
## lm(formula = y ~ +x3 + x4 + x6, data = xt5\_9)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -628.52 -109.40 -0.69 165.52 913.37   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.296e+03 1.870e+03 -1.228 0.236   
## x3 1.359e+00 9.681e-02 14.036 8.84e-11 \*\*\*  
## x4 3.143e-02 1.906e-02 1.649 0.117   
## x6 3.702e-03 1.446e-02 0.256 0.801   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 369.8 on 17 degrees of freedom  
## Multiple R-squared: 0.983, Adjusted R-squared: 0.98   
## F-statistic: 328.2 on 3 and 17 DF, p-value: 3.055e-15

lm5\_duochongstep = step(lm5\_dropx2x5x1,direction = "both")#在进行自变量选元

## Start: AIC=251.91  
## y ~ +x3 + x4 + x6  
##   
## Df Sum of Sq RSS AIC  
## - x6 1 8961 2333864 249.99  
## <none> 2324902 251.91  
## - x4 1 371939 2696841 253.02  
## - x3 1 26942858 29267761 303.10  
##   
## Step: AIC=249.99  
## y ~ x3 + x4  
##   
## Df Sum of Sq RSS AIC  
## <none> 2333864 249.99  
## - x4 1 461190 2795054 251.78  
## + x6 1 8961 2324902 251.91  
## - x3 1 26940711 29274575 301.10

summary(lm5\_duochongstep)#R = 0.983

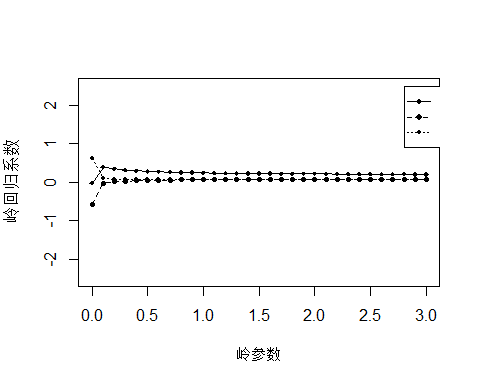
##   
## Call:  
## lm(formula = y ~ x3 + x4, data = xt5\_9)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -643.14 -105.66 -4.29 168.60 908.91   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.307e+03 1.820e+03 -1.267 0.2212   
## x3 1.359e+00 9.426e-02 14.415 2.5e-11 \*\*\*  
## x4 3.304e-02 1.752e-02 1.886 0.0755 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 360.1 on 18 degrees of freedom  
## Multiple R-squared: 0.983, Adjusted R-squared: 0.9811   
## F-statistic: 519.3 on 2 and 18 DF, p-value: < 2.2e-16

#我们可以发现用多重共线性剔除变量的结果与逐步回归选元的结果是完全相反的

#第七章岭回归  
library(MASS)  
  
ridge5 = lm.ridge(y~x1+x2+x5-1,data = xt5\_9,lambda = seq(0,3,0.1))  
beta = coef(ridge5)  
beta

## x1 x2 x5  
## 0.0 -0.02205999 -0.58580470 0.63266229  
## 0.1 0.39832354 -0.02857884 0.08878772  
## 0.2 0.35318213 0.01115158 0.07547837  
## 0.3 0.32270860 0.03051441 0.07173349  
## 0.4 0.30177659 0.04233207 0.07012305  
## 0.5 0.28661398 0.05034513 0.06926066  
## 0.6 0.27513076 0.05613965 0.06872679  
## 0.7 0.26612202 0.06051935 0.06835884  
## 0.8 0.25885267 0.06393893 0.06808318  
## 0.9 0.25285093 0.06667611 0.06786264  
## 1.0 0.24780090 0.06891052 0.06767687  
## 1.1 0.24348333 0.07076363 0.06751403  
## 1.2 0.23974141 0.07232066 0.06736687  
## 1.3 0.23646007 0.07364312 0.06723076  
## 1.4 0.23355299 0.07477660 0.06710266  
## 1.5 0.23095416 0.07575559 0.06698048  
## 1.6 0.22861225 0.07660667 0.06686279  
## 1.7 0.22648675 0.07735066 0.06674856  
## 1.8 0.22454525 0.07800410 0.06663705  
## 1.9 0.22276153 0.07858031 0.06652771  
## 2.0 0.22111414 0.07909011 0.06642013  
## 2.1 0.21958538 0.07954242 0.06631401  
## 2.2 0.21816048 0.07994462 0.06620910  
## 2.3 0.21682706 0.08030290 0.06610522  
## 2.4 0.21557463 0.08062246 0.06600223  
## 2.5 0.21439428 0.08090774 0.06590001  
## 2.6 0.21327835 0.08116251 0.06579847  
## 2.7 0.21222025 0.08139003 0.06569755  
## 2.8 0.21121427 0.08159310 0.06559718  
## 2.9 0.21025544 0.08177417 0.06549732  
## 3.0 0.20933941 0.08193537 0.06539792

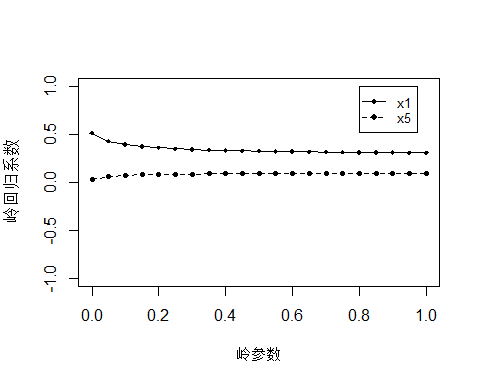
k = ridge5$lambda  
plot(k,k,type = "n",xlab = "岭参数",ylab = "岭回归系数",ylim = c(-2.5,2.5))  
linetype = c(1:5)  
char = c(18:22)  
for (i in 1:3)   
 lines(k,beta[,i],type = "o",lty = linetype[i],pch = char[i],cex = 0.75)  
legend(2.8,2.5,inset = 0.5,legend = c("x1","x2","x5"),cex = 0.8,pch = char,lty = linetype)



#先运行一遍上面的代码以后  
#观察到在k=0.5后趋于平稳，x2和x5之和比较稳定，从岭回归的角度看x2与x5只需保留一个，x1在k小时趋于零，在k稳定后大于0认为其较显著稳定  
#故删除掉x2再进行岭回归并吧参数设置为0-1步长为0.02  
ridge51 = lm.ridge(y~x1+x2+x5-x2-1,data = xt5\_9,lambda = seq(0,1,0.05))  
beta1 = coef(ridge51)  
beta1

## x1 x5  
## 0.00 0.5144136 0.02895148  
## 0.05 0.4304283 0.05758846  
## 0.10 0.3932653 0.07012402  
## 0.15 0.3722034 0.07712303  
## 0.20 0.3585746 0.08156643  
## 0.25 0.3489863 0.08462091  
## 0.30 0.3418380 0.08683683  
## 0.35 0.3362762 0.08850777  
## 0.40 0.3318038 0.08980462  
## 0.45 0.3281119 0.09083361  
## 0.50 0.3249985 0.09166425  
## 0.55 0.3223256 0.09234394  
## 0.60 0.3199962 0.09290611  
## 0.65 0.3179396 0.09337500  
## 0.70 0.3161034 0.09376865  
## 0.75 0.3144479 0.09410072  
## 0.80 0.3129424 0.09438177  
## 0.85 0.3115627 0.09462009  
## 0.90 0.3102897 0.09482226  
## 0.95 0.3091080 0.09499361  
## 1.00 0.3080050 0.09513844

k1 = ridge51$lambda  
plot(k1,k1,type = "n",xlab = "岭参数",ylab = "岭回归系数",ylim = c(-1,1))  
for (i in 1:2)   
 lines(k1,beta1[,i],type = "o",lty = linetype[i],pch = char[i],cex = 0.75)  
legend(0.8,1,inset = 0.5,legend = c("x1","x5"),cex = 0.8,pch = char,lty = linetype)



#可以选择k=0.8趋于稳定得到的岭回归方程

做完这些后，发现，在第六章要求去除多重共线性时，最小二乘法没法做到自变量选元的同时去除多重共线性

而在第七章，用岭回归来代替最小二乘法来去除多重共线性的同时自变量选元，还能保留较多的自变量.