test

auzzer

5/17/2020

setwd("/Users/auzzer\_pang")  
data = read.csv("loan\_bank.csv",head = T)  
data<-data[1:25,]

#首先建立y与其余四个变量的回归方程，并详细展示回归结果  
lm1 = lm(y~x1+x2+x3+x4,data = data)  
summary(lm1)

##   
## Call:  
## lm(formula = y ~ x1 + x2 + x3 + x4, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.9198 -0.9507 -0.2880 1.0334 3.1037   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.02164 0.78237 -1.306 0.20643   
## x1 0.04004 0.01043 3.837 0.00103 \*\*  
## x2 0.14803 0.07879 1.879 0.07494 .   
## x3 0.01453 0.08303 0.175 0.86285   
## x4 -0.02919 0.01507 -1.937 0.06703 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.779 on 20 degrees of freedom  
## Multiple R-squared: 0.7976, Adjusted R-squared: 0.7571   
## F-statistic: 19.7 on 4 and 20 DF, p-value: 1.035e-06

#下面进行多重共线性诊断，分别使用方差扩大因子法和condtion index  
library(car)

## Loading required package: carData

vif(lm1)

## x1 x2 x3 x4   
## 5.330807 1.889860 3.834823 2.781220

XX = cor(data[,2:5])  
XX

## x1 x2 x3 x4  
## x1 1.0000000 0.6787718 0.8484164 0.7797022  
## x2 0.6787718 1.0000000 0.5858315 0.4724310  
## x3 0.8484164 0.5858315 1.0000000 0.7466458  
## x4 0.7797022 0.4724310 0.7466458 1.0000000

kappa(XX , exact = TRUE )

## [1] 23.23595

#采用后退法选择变量  
#采用AIC进行后退法  
reg2 = lm(y~.,data = data)  
regbackward2 = step(reg2,direction="backward")#按照AIC原则自动选择模型

## Start: AIC=33.22  
## y ~ x1 + x2 + x3 + x4  
##   
## Df Sum of Sq RSS AIC  
## - x3 1 0.097 63.376 31.255  
## <none> 63.279 33.217  
## - x2 1 11.168 74.447 35.280  
## - x4 1 11.868 75.147 35.514  
## - x1 1 46.594 109.873 45.011  
##   
## Step: AIC=31.26  
## y ~ x1 + x2 + x4  
##   
## Df Sum of Sq RSS AIC  
## <none> 63.376 31.255  
## - x2 1 11.333 74.709 33.368  
## - x4 1 12.147 75.523 33.639  
## - x1 1 69.939 133.315 47.846

summary(regbackward2)

##   
## Call:  
## lm(formula = y ~ x1 + x2 + x4, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.8531 -0.8766 -0.3685 0.9586 3.0772   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.971605 0.711240 -1.366 0.1864   
## x1 0.041039 0.008525 4.814 9.31e-05 \*\*\*  
## x2 0.148858 0.076817 1.938 0.0662 .   
## x4 -0.028502 0.014206 -2.006 0.0579 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.737 on 21 degrees of freedom  
## Multiple R-squared: 0.7973, Adjusted R-squared: 0.7683   
## F-statistic: 27.53 on 3 and 21 DF, p-value: 1.802e-07

#采用AIC原则自动选择模型-逐步回归法  
reg3 = lm(y~.,data = data)  
regboth = step(reg3,direction="both")#按照AIC原则自动选择模型

## Start: AIC=33.22  
## y ~ x1 + x2 + x3 + x4  
##   
## Df Sum of Sq RSS AIC  
## - x3 1 0.097 63.376 31.255  
## <none> 63.279 33.217  
## - x2 1 11.168 74.447 35.280  
## - x4 1 11.868 75.147 35.514  
## - x1 1 46.594 109.873 45.011  
##   
## Step: AIC=31.26  
## y ~ x1 + x2 + x4  
##   
## Df Sum of Sq RSS AIC  
## <none> 63.376 31.255  
## + x3 1 0.097 63.279 33.217  
## - x2 1 11.333 74.709 33.368  
## - x4 1 12.147 75.523 33.639  
## - x1 1 69.939 133.315 47.846

summary(regboth)

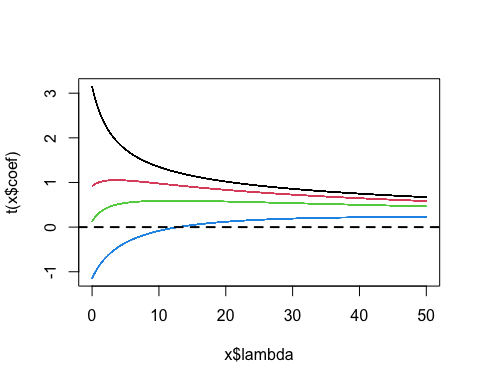
##   
## Call:  
## lm(formula = y ~ x1 + x2 + x4, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.8531 -0.8766 -0.3685 0.9586 3.0772   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.971605 0.711240 -1.366 0.1864   
## x1 0.041039 0.008525 4.814 9.31e-05 \*\*\*  
## x2 0.148858 0.076817 1.938 0.0662 .   
## x4 -0.028502 0.014206 -2.006 0.0579 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.737 on 21 degrees of freedom  
## Multiple R-squared: 0.7973, Adjusted R-squared: 0.7683   
## F-statistic: 27.53 on 3 and 21 DF, p-value: 1.802e-07

library(MASS)  
ridge1 = lm.ridge(y~x1+x2+x3+x4,data = data)  
ridge1

## x1 x2 x3 x4   
## -1.02163976 0.04003935 0.14803389 0.01452935 -0.02919287

#显示模型的参数系数

plot(lm.ridge(y~x1+x2+x3+x4,data = data,lambda= seq(0,50,0.001)))  
abline( h = 0,col = "black",lwd = 2,lty = 2)



select(lm.ridge(y~x1+x2+x3+x4,data,lambda= seq(0,50,0.001)))

## modified HKB estimator is 0.5223352   
## modified L-W estimator is 0.6343875   
## smallest value of GCV at 0.947

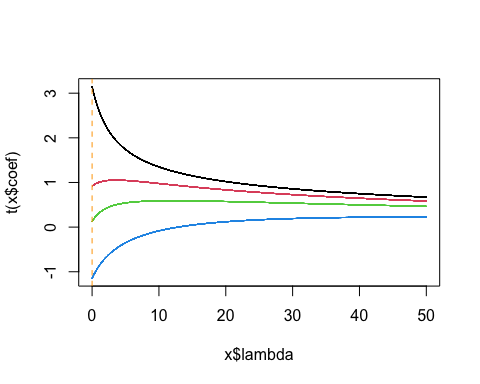
names(ridge1)

## [1] "coef" "scales" "Inter" "lambda" "ym" "xm" "GCV" "kHKB"   
## [9] "kLW"

ridge1$lambda[which.min(ridge1$GCV)]#找到GCV最小时的lambda

## [1] 0

plot(lm.ridge(y~x1+x2+x3+x4,data = data,lambda= seq(0,50,0.001)))  
abline(v = ridge1$lambda[which.min(ridge1$GCV)],  
 col = "orange", lty = 2)

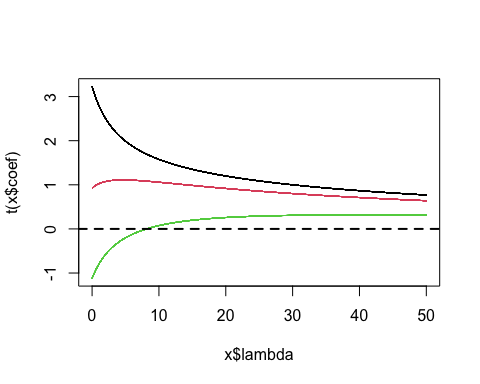


#利用后退法和逐步回归法结果进行岭回归  
library(MASS)  
ridge2 = lm.ridge(y~x1+x2+x4,data = data)  
ridge2

## x1 x2 x4   
## -0.97160495 0.04103950 0.14885813 -0.02850153

#显示模型的参数系数

plot(lm.ridge(y~x1+x2+x4,data = data,lambda= seq(0,50,0.001)))  
abline( h = 0,col = "black",lwd = 2,lty = 2)



select(lm.ridge(y~x1+x2+x4,data,lambda= seq(0,50,0.001)))

## modified HKB estimator is 0.2404709   
## modified L-W estimator is 0.3026694   
## smallest value of GCV at 0.545

names(ridge2)

## [1] "coef" "scales" "Inter" "lambda" "ym" "xm" "GCV" "kHKB"   
## [9] "kLW"

ridge2$lambda[which.min(ridge2$GCV)]#找到GCV最小时的lambda

## [1] 0

plot(lm.ridge(y~x1+x2+x4,data = data,lambda= seq(0,50,0.001)))  
abline(v = ridge2$lambda[which.min(ridge2$GCV)],  
 col = "orange", lty = 2)

