蘇州大學
SOOCHOW UNIVERSITY

Soochow University

School of Mathematics and Science

应用回归分析大作业

杜冰

1807403036

2020 年 12 月 21 日

目录

摘要

本文以北卡罗来纳州立大学的健身课程中参与体检的男性身体数据为例，运用 R 软件建立多元回归模型, 预测氧气摄入量。为解决回归模型中的多重共线性问题，利用方差扩大因子进行诊断, 并分别运用了岭回归模型和 Lasso 回归模型对其进行修正，建立了基于多元线性回归的氧气摄入量相对水平预测模型，最后对这两个修正模型的拟合结果进行了对比分析, 得出岭回归模型相对较优的结论。

关键词：R 软件；多重共线性；岭回归；Lasso

1 问题背景

物质代谢和能量代谢是机体内各组织器官技能活动的基础，而运动能力是身体各种机能活动的集中表现。根据能量方式的不同，运动能力可以划分为有氧运动和无氧运动。而有氧供能的能力是基础，对于它已经有大量学者做过研究，其中最大摄氧量是评价有氧能力最常用和最有效的方法。但最大摄氧量很难直接测量，因此有必要通过通过其他身体数据对最大摄氧量进行预测研究，进而得出最大摄氧量的主要影响因素。

2 符号说明

表 1: 符号说明

符号	说明	单位
Age	年龄	*year*
Weight	体重	*kg*
Oxygen	氧气摄入量	$ml/(kg \cdot minute)$
RunTime	跑步 1.5*km* 所需要的时间	*minutes*
RestPulse	安静心率	*bpm*
RunPulse	运动心率	*bpm*
MaxPulse	运动时最大心率	*bpm*

3 多元线性回归模型的建立

3.1 数据预处理

3.1.1 缺失值的判断与处理

首先对数据进行缺失值的检查，代码如下：

```
rm(list=ls())
fitness <- read.csv("D:/fitness.csv", header = T)
sum(is.na(fitness) == TRUE)
## [1] 0
```

由运行结果知，该数据中不含有缺失值。

3.1.2 数据标准化

数据的概要信息输出如下：

```
summary(fitness)
##       Age           Weight         Oxygen         RunTime
##  Min.   :38.00  Min.   :59.08  Min.   :37.39  Min.   : 8.17
##  1st Qu.:44.00  1st Qu.:73.20  1st Qu.:44.96  1st Qu.: 9.78
##  Median :48.00  Median :77.45  Median :46.77  Median :10.47
##  Mean   :47.68  Mean   :77.44  Mean   :47.38  Mean   :10.59
##  3rd Qu.:51.00  3rd Qu.:82.33  3rd Qu.:50.13  3rd Qu.:11.27
##  Max.   :57.00  Max.   :91.63  Max.   :60.05  Max.   :14.03
##    RestPulse       RunPulse       MaxPulse
##  Min.   :40.00  Min.   :146.0  Min.   :155.0
##  1st Qu.:48.00  1st Qu.:163.0  1st Qu.:168.0
##  Median :52.00  Median :170.0  Median :172.0
##  Mean   :53.45  Mean   :169.6  Mean   :173.8
##  3rd Qu.:58.50  3rd Qu.:176.0  3rd Qu.:180.0
##  Max.   :70.00  Max.   :186.0  Max.   :192.0
```

由以上输出结果可得，需要对数据进行标准化，下面对数据进行标准化，并

输出标准化后数据的箱线图。

```
fitness <- data.frame(scale(fitness))#标准化
na <- c(colnames(fitness))
boxplot(fitness$Age, fitness$Weight, fitness$Oxygen, fitness$
   RunTime, fitness$RestPulse, fitness$RunPulse, fitness$
   MaxPulse, main = "Box plot", names = na)
```

画出的箱线图如下图所示：

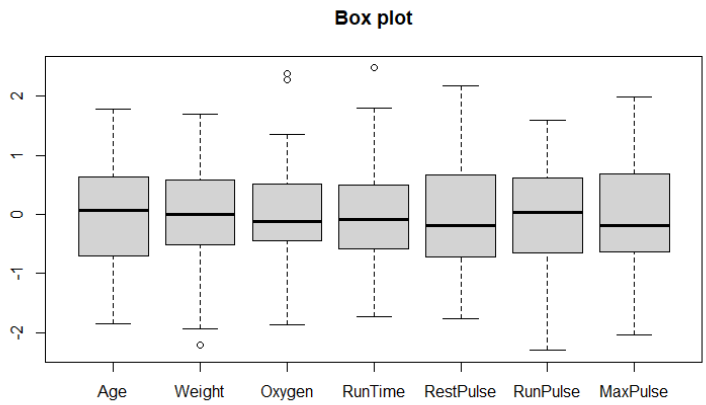


图 1: 箱线图

3.1.3 数据清洗

由上面的分析了解到数据中有离群点，下面对离群点数据进行剔除。

```
outlier_location <- sapply(fitness, function(X){which(X%in%
   boxplot.stats(X)$out)})#找出异常值的位置
todel <- (sort(unique(unlist(outlier_location))))#求并
fitness <- fitness[-todel, ]#剔除离群点
boxplot(fitness$Age, fitness$Weight, fitness$Oxygen, fitness$
   RunTime, fitness$RestPulse, fitness$RunPulse, fitness$
   MaxPulse, main = "Box plot", names = na)
```

剔除异常值后画出的箱线图如下图所示：

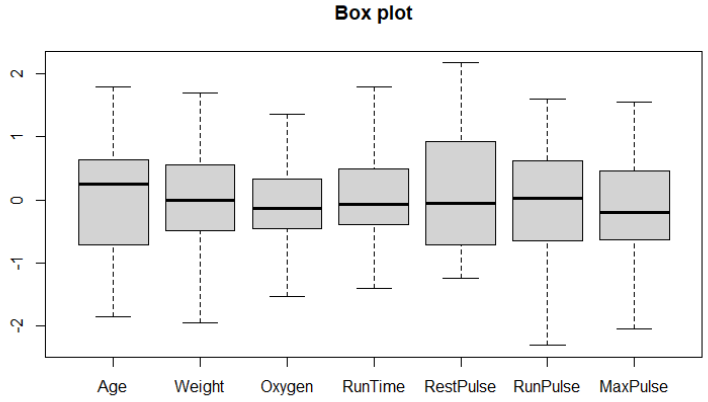


图 2: 剔除异常值后数据的箱线图

从剔除异常值后的箱线图可以看出此时数据中不再含有异常值，下面就此数据进行分析。

3.2 建立多元线性回归模型

建立多元线性模型如下：

$$Oxygen = X\beta + \varepsilon$$

其中，

$$Oxygen = \begin{bmatrix} Oxygen_1 \\ Oxygen_2 \\ \vdots \\ Oxygen_{27} \end{bmatrix}, X = \begin{bmatrix} 1 & Age_1 & Weight_1 & \cdots & MaxPulse_1 \\ 1 & Age_2 & Weight_2 & \cdots & MaxPulse_2 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & Age_{27} & Weight_{27} & \cdots & MaxPulse_{27} \end{bmatrix} \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_6 \end{bmatrix}$$

3.3 异方差性的检验

对上述模型的参数进行估计 (运行结果见附录 B.a)，并作出散点图：

```
fit1<-lm(Oxygen~.,fitness)
#summary(fit1)
e<-resid(fit1)
plot(e,main = '残差散点图',ylab = '残差')
abline(h=0)
```

散点图如下如所示：

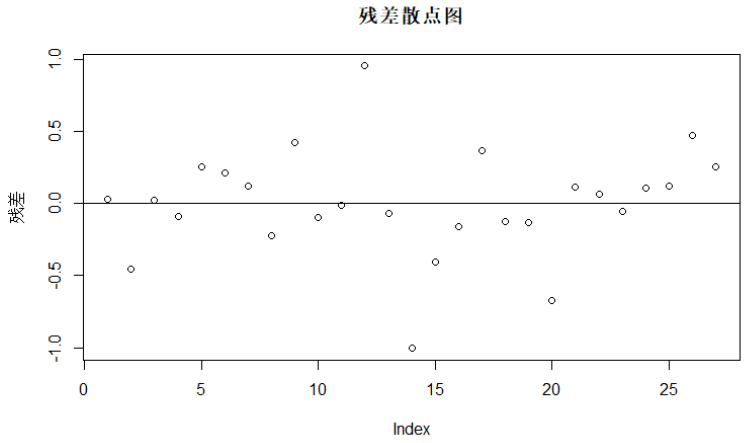


图 3: 散点图

下面对回归方程的异方差性进行检验：

```
library(car)
## Warning: package 'car' was built under R version 4.0.3
## Loading required package: carData
## Warning: package 'carData' was built under R version 4.0.3
ncvTest(fit1)
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.05316445, Df = 1, p = 0.81765
```

由检验结果知，Breusch-Pagan 检验对应的 *P_value* 为 0.81765>0.1，因此在显著性水平 $\alpha = 0.1$ 的情况下，认为不存在异方差性。

3.4 自回归性的检验

下面对回归方程的自回归性进行检验：

```
library(lmtest)
dwtest(fit1)
##
##  Durbin-Watson test
##
## data: fit1
## DW = 1.7343, p-value = 0.1684
## alternative hypothesis: true autocorrelation is greater
   than 0
```

由检验结果知，Durbin-Watson(DW) 检验对应的 *P_value* 为 0.1684>0.1，因此在显著性水平 $\alpha = 0.1$ 的情况下，认为不存在自回归性。

3.5 小结

由参数估计的结果，得到各参数估计 t 检验对应的 *P_value* 分别为：0.196, 0.516, 0.644, 0.182, 0.556。其中 *RunPulse* 对应的 *P_value* 为 0.182，但在实际意义中，运动心率是预测氧气摄入量的重要变量。以上结论说明模型可能失真，考虑的主要原因是自变量间可能存在共线性的问题，接下来进行共线性的诊断。

4 多重共线性诊断及其处理

4.1 多重共线性诊断

进行多重共线性诊断，代码如下：

```
vif(fit1)
##      Age    Weight   RunTime RestPulse RunPulse MaxPulse
##  1.315814 1.078667 1.374538 1.512121 11.432239 11.441176
```

由运行结果知，变量 *RunPulse* 及 *MaxPulse* 的方差扩大因子 (variance inflation factor, VIF) 分别为 11.432239, 11.441176，均大于 10，故认为 *RunPulse* 及 *MaxPulse* 与其余自变量间有严重的多重共线性。

4.2 通过岭回归分析建立线性模型

A.E.Hoerl 在 1962 年首次提出岭回归方法，用以控制与最小二乘估计相关的方差膨胀性和产生的不稳定性，A.E.Hoerl 和 R.W.Kennard [1] 对岭回归给出了具体的分析与证明。G.C.Mcdonald [2] 提供了岭回归方法的简要概述，证明了岭回归的相关性质。岭回归 [3] 是在普通最小二乘的参数估计 $\hat{\beta} = (X'X)^{-1}X'Y$ 中引入矩阵 $kI(k > 0, I$为单位矩阵$)$，得到 $\hat{\beta}(k) = (X'X + kI)^{-1}X'Y$，这里 $\hat{\beta}(k)$ 为岭回归估计，k 为岭参数，由于岭参数不唯一，故得到的 $\hat{\beta}(k)$ 是参数 β 的估计族。本文结合回归系数的岭迹图、岭参数 k 值的基本原则 [3] 以及 R 软件中的 linerRidge() 函数确定最恰当的岭参数 k。

```
library(ridge)
## Warning: package 'ridge' was built under R version 4.0.3
plot(linearRidge(Oxygen~., fitness, lambda = seq(0,1,0.01)))
abline(h = 0)
```

由以上代码得到的岭迹图如下图所示：

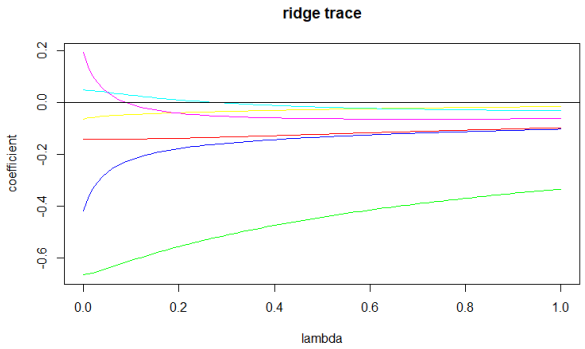


图 4: 岭迹图

由岭迹图可知，有两个变量由正变负，其他稳定，所以需要对两个变量进行剔除，由选取不同岭回归系数的变化情况 (见附录 B.b)，可知这两个变量分别为 *RestPulse* 和 *MaxPulse*，将这两个变量剔除后再进行岭回归。

通过 linearRidge() 函数给出的结果 (见附录 B.c) 来看，岭回归参数为 0.06716248，此时变量 *Weight* 在 $\alpha = 0.1$ 的显著性水平下不显著 ($P\_value = 0.52991$)，故剔除该变量。

将变量 *Weight* 剔除后再次进行岭回归 (见附录 B.d)，最终得到岭回归参数为 0.05033456，此时各变量在显著性水平 $\alpha = 0.1$ 的情况下均显著，此时的岭回归

各变量的参数估计如表 2 所示，故得到的岭回归模型为：

$$Oxygen = -0.08322 - 0.14706 \times Age - 0.62220 \times RunTime - 0.22226 \times RunPulse$$

表 2: 岭回归参数估计

参数	Intercrpt	*Age*	*RunTime*	*RunPulse*
估计值	-0.08322	-0.14706	-0.62220	-0.22226

4.3 通过 Lasso 回归分析建立线性模型

Lasso 最早由 R.Tibshirain [4] 提出，此后广泛应用在变量选择和参数估计中，基本思想是对参数进行压缩，进而选择最重要的变量，定义为：

$$\hat{\beta}^{lasso} = \arg\min_{\beta} RSS + \lambda \sum_{j=1}^{p} |\beta_j|$$

其中，$RRS = \sum_{i=1}^{n} (y_i - x_i\beta)^2$，$\lambda$ 是截断参数 (Tuning Parameter)，$\lambda \sum_{j=1}^{p} |\beta_j|$ 为惩罚项。

表 3: *CP* 统计量的变化值

个数	0	1	2	3	4	5	6
CP 值	56.0792	10.6840	5.6502	2.8553	4.6166	5.4168	7.0000

CP 统计量是选择最优子集的一种方法，其值越小表示所选子集个数最优。表 3(代码及结果见附录 B.e) 为 *CP* 统计量的变化值，可以看出，当变量个数为 3 时，*CP* 值达到最小，故选择 3 个变量进入模型。

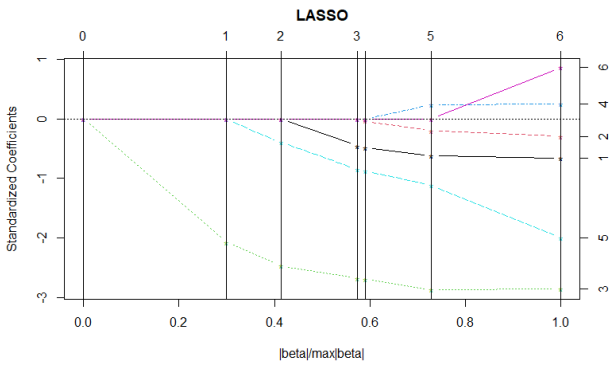


图 5: Lasso 回归效果

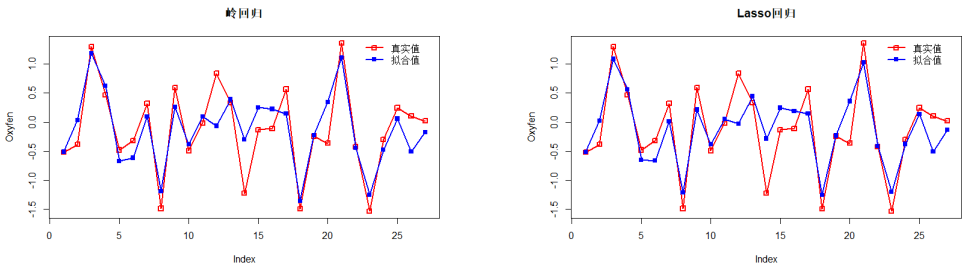
由图 5 可得，这三个变量分别为第 1，3，5 变量，即 *Age*,*RunTime* 和 *RunPulse*，对应的回归参数估计值如表 4(代码及结果见附录 B.f) 所示，故得到的 Lasso 回归模型为：

$$Oxygen = -0.08773 - 0.09677 \times Age - 0.62281 \times RunTime - 0.17625 \times RunPulse$$

表 4: Lasso 回归参数估计

参数	Intercrpt	*Age*	*RunTime*	*RunPulse*
估计值	-0.08773	-0.09677	-0.62281	-0.17625

4.4 岭回归与 Lasso 回归效果对比



(a) 岭回归拟合图　　(b) Lasso 回归拟合图

图 6: 拟合图

通过岭回归与 Lasso 回归得到的拟合值与实际值的拟合图 (代码见附录 B.g) 如图 6 所示，回归拟合值和实际值的走势大致相同，拟合程度较好。观察图像可得，岭回归模型的拟合效果更高，说明岭回归模型相对较优。

5 总结

由岭回归与 Lasso 回归效果对比可得岭回归拟合效果更好，模型相对较优，其模型如下所示：

$$Oxygen = -0.08322 - 0.14706 \times Age - 0.62220 \times RunTime - 0.22226 \times RunPulse$$

从模型结果来看，三个变量中 *RunTime* 对氧气摄入量的影响较大，跑步 1.5*km* 需要的时间越短，说明氧气摄入量越大；*RunPulse* 对氧气摄入量的影响较小，但

也能反应氧气摄入量，即运动时心跳相对越慢，氧气摄入量越大；*Age* 对氧气摄入量的影响最小，随着年龄的增大，氧气摄入量逐渐减小，但该减小速度相对较慢，可以通过锻炼身体进行改善。

参考文献

[1] HOERL A E, KENNARD R W.Ridge regression: biased estimation for nonorthogonal problems[J]. Technometrics, 1970, 12(1): 55-67.

[2] MCDONALD G C. Ridge regression[J].Wires computational statistics, 2009, 1(1): 93-100.

[3] 何小群. 应用回归分析 [M]. 第 5 版. 背景: 中国人民大学出版社, 2019.

[4] TIBSHIRANI R. Regression shrinkage and selection via the Lasso[J]. J R Statist Soc B,1996, 58(1): 267-288.

附录

A. 原始数据

Age	Weight	Oxygen	RunTime	RestPulse	RunPulse	MaxPulse
44	89.47	44.609	11.37	62	178	182
40	75.07	45.313	10.07	62	185	185
44	85.84	54.297	8.65	45	156	168
42	68.15	59.571	8.17	40	166	172
38	89.02	49.874	9.22	55	178	180
47	77.45	44.811	11.63	58	176	176
40	75.98	45.681	11.95	70	176	180
43	81.19	49.091	10.85	64	162	170
44	81.42	39.442	13.08	63	174	176
38	81.87	60.055	8.63	48	170	186
44	73.03	50.541	10.13	45	168	168
45	87.66	37.388	14.03	56	186	192
45	66.45	44.754	11.12	51	176	176
47	79.15	47.273	10.6	47	162	164
54	83.12	51.855	10.33	50	166	170
49	81.42	49.156	8.95	44	180	185
51	69.63	40.836	10.95	57	168	172
51	77.91	46.672	10	48	162	168
48	91.63	46.774	10.25	48	162	164
49	73.37	50.388	10.08	67	168	168
57	73.37	39.407	12.63	58	174	176
54	79.38	46.08	11.17	62	156	165
52	76.32	45.441	9.63	48	164	166
50	70.87	54.625	8.92	48	146	155
51	67.25	45.118	11.08	48	172	172
54	91.63	39.203	12.88	44	168	172
51	73.71	45.79	10.47	59	186	188
57	59.08	50.545	9.93	49	148	155
49	76.32	48.673	9.4	56	186	188
48	61.24	47.92	11.5	52	170	176
52	82.78	47.467	10.5	53	170	172

B. 部分代码及运行结果

a. 多元线性回归模型参数估计

```
fit1 <- lm(Oxygen~.,fitness)
summary(fit1)
##
## Call:
## lm(formula = Oxygen ~ ., data = fitness)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00552 -0.13019 0.02393 0.16569 0.95344
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.06931  0.08686 -0.798   0.434
## Age         -0.14046   0.10497 -1.338   0.196
## Weight      -0.06261   0.09457 -0.662   0.516
## RunTime     -0.66481   0.11631 -5.716 1.36e-05 ***
## RestPulse    0.04871   0.10375  0.470   0.644
## RunPulse    -0.41828   0.30225 -1.384   0.182
## MaxPulse     0.19335   0.32325  0.598   0.556
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
    1
##
## Residual standard error: 0.4272 on 20 degrees of freedom
## Multiple R-squared: 0.7533, Adjusted R-squared: 0.6793
## F-statistic: 10.18 on 6 and 20 DF, p-value: 3.315e-05
```

b. 选取不同岭回归系数的变化

```
linearRidge(Oxygen~., fitness, lambda = seq(0,0.3,0.01))
##
## Call:
## linearRidge(formula = Oxygen ~ ., data = fitness, lambda =
   seq(0,
##     0.3, 0.01))
##
##             (Intercept)        Age     Weight    RunTime
   RestPulse
## lambda=0   -0.06931367 -0.14045539 -0.06260693 -0.6648059
   0.0487123630
## lambda=0.01 -0.07334066 -0.14008123 -0.05890998 -0.6612237
   0.0479764918
## lambda=0.02 -0.07612594 -0.13994759 -0.05620736 -0.6563947
   0.0464555566
## lambda=0.03 -0.07817360 -0.13992186 -0.05409724 -0.6509125
   0.0445311678
## lambda=0.04 -0.07974886 -0.13993786 -0.05236851 -0.6450844
   0.0423993738
## lambda=0.05 -0.08100437 -0.13996030 -0.05089985 -0.6390827
   0.0401692366
## lambda=0.06 -0.08203408 -0.13996973 -0.04961676 -0.6330102
   0.0379047996
## lambda=0.07 -0.08289890 -0.13995537 -0.04847101 -0.6269305
   0.0356450382
## lambda=0.08 -0.08363996 -0.13991142 -0.04743013 -0.6208842
   0.0334141676
## lambda=0.09 -0.08428599 -0.13983509 -0.04647147 -0.6148976
   0.0312273295
## lambda=0.1 -0.08485766 -0.13972540 -0.04557877 -0.6089877
   0.0290938964
## lambda=0.11 -0.08537014 -0.13958247 -0.04474010 -0.6031656
   0.0270194766
## lambda=0.12 -0.08583484 -0.13940714 -0.04394652 -0.5974382
   0.0250071734
```

```
## lambda=0.13 -0.08626046 -0.13920067 -0.04319118 -0.5918096
   0.0230584001
## lambda=0.14 -0.08665374 -0.13896454 -0.04246877 -0.5862819
   0.0211734198
## lambda=0.15 -0.08702000 -0.13870040 -0.04177513 -0.5808559
   0.0193517113
## lambda=0.16 -0.08736345 -0.13840992 -0.04110690 -0.5755316
   0.0175922204
## lambda=0.17 -0.08768747 -0.13809478 -0.04046140 -0.5703078
   0.0158935351
## lambda=0.18 -0.08799482 -0.13775665 -0.03983641 -0.5651834
   0.0142540097
## lambda=0.19 -0.08828774 -0.13739714 -0.03923011 -0.5601565
   0.0126718522
## lambda=0.2 -0.08856808 -0.13701779 -0.03864097 -0.5552253
   0.0111451877
## lambda=0.21 -0.08883740 -0.13662008 -0.03806772 -0.5503876
   0.0096721031
## lambda=0.22 -0.08909697 -0.13620542 -0.03750926 -0.5456412
   0.0082506797
## lambda=0.23 -0.08934789 -0.13577512 -0.03696466 -0.5409840
   0.0068790158
## lambda=0.24 -0.08959106 -0.13533045 -0.03643309 -0.5364136
   0.0055552431
## lambda=0.25 -0.08982727 -0.13487258 -0.03591385 -0.5319277
   0.0042775384
## lambda=0.26 -0.09005718 -0.13440263 -0.03540631 -0.5275242
   0.0030441304
## lambda=0.27 -0.09028136 -0.13392163 -0.03490992 -0.5232009
   0.0018533054
## lambda=0.28 -0.09050032 -0.13343056 -0.03442419 -0.5189555
   0.0007034098
## lambda=0.29 -0.09071447 -0.13293033 -0.03394866 -0.5147859
   -0.0004071484
## lambda=0.3 -0.09092419 -0.13242180 -0.03348294 -0.5106902
   -0.0014798992
```

```
##              RunPulse     MaxPulse
## lambda=0   -0.4182803 0.1933541051
## lambda=0.01 -0.3672492 0.1401165225
## lambda=0.02 -0.3317092 0.1034287932
## lambda=0.03 -0.3054389 0.0766376458
## lambda=0.04 -0.2851583 0.0562357298
## lambda=0.05 -0.2689742 0.0401986273
## lambda=0.06 -0.2557171 0.0272759220
## lambda=0.07 -0.2446255 0.0166534842
## lambda=0.08 -0.2351822 0.0077783972
## lambda=0.09 -0.2270232 0.0002619324
## lambda=0.1 -0.2198855 -0.0061772147
## lambda=0.11 -0.2135735 -0.0117476574
## lambda=0.12 -0.2079394 -0.0166073904
## lambda=0.13 -0.2028689 -0.0208782777
## lambda=0.14 -0.1982725 -0.0246558327
## lambda=0.15 -0.1940790 -0.0280159823
## lambda=0.16 -0.1902308 -0.0310198492
## lambda=0.17 -0.1866813 -0.0337171967
## lambda=0.18 -0.1833920 -0.0361489539
## lambda=0.19 -0.1803308 -0.0383490957
## lambda=0.2 -0.1774709 -0.0403460616
## lambda=0.21 -0.1747898 -0.0421638408
## lambda=0.22 -0.1722680 -0.0438228107
## lambda=0.23 -0.1698892 -0.0453403933
## lambda=0.24 -0.1676391 -0.0467315732
## lambda=0.25 -0.1655053 -0.0480093093
## lambda=0.26 -0.1634772 -0.0491848670
## lambda=0.27 -0.1615454 -0.0502680860
## lambda=0.28 -0.1597015 -0.0512675993
## lambda=0.29 -0.1579384 -0.0521910127
## lambda=0.3 -0.1562495 -0.0530450533
```

c. 剔除变量 *RestPulse* 和 *MaxPulse* 的后岭回归参数估计

```
summary(linearRidge(Oxygen~Age+Weight+RunTime+RunPulse,
   fitness, lambda = "automatic"))
##
## Call:
## linearRidge(formula = Oxygen ~ Age + Weight + RunTime +
   RunPulse,
##     data = fitness, lambda = "automatic")
##
##
## Coefficients:
##             Estimate Scaled estimate Std. Error (scaled) t
   value (scaled)
## (Intercept) -0.08016           NA                 NA
                NA
## Age          -0.15270       -0.71294           0.40192
             1.774
## Weight       -0.05161       -0.24215           0.38550
             0.628
## RunTime      -0.61368       -2.64280           0.39336
             6.719
## RunPulse     -0.22492       -1.07502           0.40186
             2.675
##              Pr(>|t|)
## (Intercept)      NA
## Age           0.07609 .
## Weight        0.52991
## RunTime      1.83e-11 ***
## RunPulse      0.00747 **
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
    1
##
## Ridge parameter: 0.06716248, chosen automatically, computed
    using 4 PCs
##
## Degrees of freedom: model 3.725 , variance 3.472 , residual
    3.979
```

d. 再将变量 *Weight* 剔除后的岭回归参数估计

```
summary(linearRidge(Oxygen~Age+RunTime+RunPulse, fitness,
   lambda = "automatic"))
##
## Call:
## linearRidge(formula = Oxygen ~ Age + RunTime + RunPulse,
   data = fitness,
##     lambda = "automatic")
##
##
## Coefficients:
##             Estimate Scaled estimate Std. Error (scaled) t
   value (scaled)
## (Intercept) -0.08322           NA                 NA
                NA
## Age          -0.14706       -0.68658           0.40135
             1.711
## RunTime      -0.62220       -2.67950           0.39537
             6.777
## RunPulse     -0.22226       -1.06230           0.40296
             2.636
##              Pr(>|t|)
## (Intercept)      NA
## Age           0.08714 .
## RunTime      1.22e-11 ***
## RunPulse      0.00838 **
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
    1
##
## Ridge parameter: 0.05033456, chosen automatically, computed
    using 3 PCs
##
## Degrees of freedom: model 2.841 , variance 2.691 , residual
    2.991
```

e.Lasso 回归计算 *CP* 统计量

```
library(lars)
## Warning: package 'lars' was built under R version 4.0.3
## Loaded lars 1.2
x <- fitness[ ,c(1,2,4,5,6,7)]
x <- as.matrix(x)
y <- fitness[ ,3]
y <- as.matrix(y)
la <- lars(x, y, type = "lasso")
plot(la)
summary(la)
## LARS/LASSO
## Call: lars(x = x, y = y, type = "lasso")
##   Df     Rss     Cp
## 0  1 14.8003 56.0792
## 1  2  6.1487 10.6840
## 2  3  4.8647 5.6502
## 3  4  3.9895 2.8553
## 4  5  3.9459 4.6166
## 5  6  3.7269 5.4168
## 6  7  3.6508 7.0000
```

f.Lasso 回归参数估计

```
coef <- coef.lars(la,mode="step",s=4)
coef
##         Age      Weight     RunTime   RestPulse    RunPulse
   MaxPulse
## -0.09676593 0.00000000 -0.62281176 0.00000000 -0.17624958
   0.00000000
predict(la,data.frame(Age=0,Weight=0,Runtime=0,RestPulse=0,
   RunPulse=0,MaxPulse=0),s=4)
## $s
## [1] 4
##
## $fraction
## [1] 0.5
##
## $mode
## [1] "step"
##
## $fit
## [1] -0.08773361
```

g. 岭回归及 Lasso 回归的拟合图代码

```
#岭回归拟合
Oxygen_ridge<--0.08322-0.14706*fitness$Age-0.62220*fitness$
   RunTime-0.22226*fitness$RunPulse
plot(fitness$Oxygen, type="o",pch=0, col="red", main="岭回归",
    ylab="Oxyfen",lwd=2)
lines(Oxygen_ridge,type="o",pch=15,col="blue",lwd=2)
legend("topright", legend=c("真实值","拟合值"),pch=c(0,15),
   col=c("red","blue"), bty="n", lty=1,lwd=2)

#Lasso回归拟合
Oxygen_lasso <- predict(la,x,s=4)$fit
plot(fitness$Oxygen, type="o",pch=0, col="red", main="Lasso回
   归", ylab="Oxyfen",lwd=2)
lines(Oxygen_lasso,type="o",pch=15,col="blue",lwd=2)
legend("topright", legend=c("真实值","拟合值"),pch=c(0,15),
   col=c("red","blue"), bty="n", lty=1,lwd=2)
```