Huiwen Chen ID: 02156341 uniquename: huiwenc

STATS 500 - Homework 8

Take the fat data, and use the percentage of body fat as the response and the other variables as potential predictors. Remove every tenth observation from the data for use as a test sample. Use the remaining data as a training sample building the following models:

1. Linear regression with all predictors

```
> library(faraway)

> data(fat)

> attach(fat)

> index <- seq(10, 250, by=10)

> ## Extract data and remove ‘‘brozek’’, ‘‘density’’ and ‘‘free’’

> train <- fat[-index, -c(1, 3, 8)]

> test <- fat[index, -c(1, 3, 8)]

> ##Linear regression with all predictors

> model1 <- lm(siri~ ., data = train)

> summary(model1)

Coefficients:

                Estimate Std. Error t value Pr(>|t|)
(Intercept) -19.82090    17.98296  -1.102  0.27162
age           0.06717     0.03409   1.970  0.05013 .
weight       -0.09557     0.05561  -1.718  0.08718 .
height       -0.04456     0.11226  -0.397  0.69183
adipos       -0.04914     0.31640  -0.155  0.87673
neck         -0.43798     0.24846  -1.763  0.07937 .
chest        -0.08242     0.10944  -0.753  0.45219
abdom         1.03016     0.09780  10.533  < 2e-16 ***
hip          -0.20410     0.15574  -1.311  0.19144
thigh         0.25359     0.15187   1.670  0.09644 .
knee          0.02971     0.26088   0.114  0.90944
ankle         0.15723     0.22680   0.693  0.48891
biceps        0.18965     0.18024   1.052  0.29391
forearm       0.46766     0.20384   2.294  0.02275 *
wrist        -1.74316     0.56008  -3.112  0.00211 **

Residual standard error: 4.324 on 212 degrees of freedom  Multiple R-squared:  0.7591,
    Adjusted R-squared:  0.7432  F-statistic: 47.71 on 14 and 212 DF,  p-value: < 2.2e-16

> rmse <- function(x,y) sqrt (mean((x-y)^2))

> rmse(model1$fitted.values,train$siri)

[1] 4.178651

> rmse(predict(model1, test),test$siri)

[1] 4.395559
```

In the full linear regression model, only three predictors show significance. Checking the correlation between each two predictors, we find there exists high colinearity among many predictors. Adjusted R-squared is relatively close to 1, so the fit is not bad in terms of R-squared. Comparing the RMSE of training data and sample data, we see that the performance of the model get worse for the test data. It's necessary to eliminate some of the predictors to reduce noise to the prediction.

2. Linear regression with variables selected using AIC

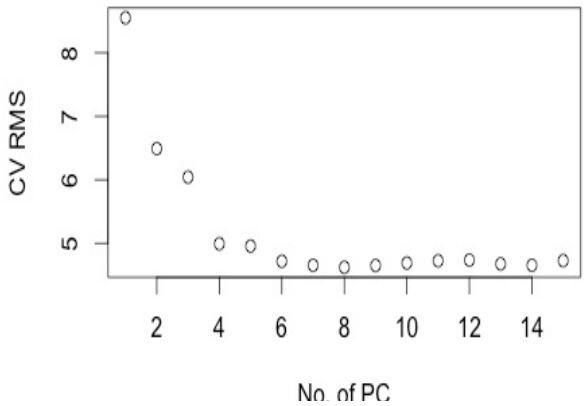
```
> ##variables selected using AIC

> model2 <- step(model1)

Start:  AIC=679.21

……

Step:  AIC=669.44

siri ~ age + weight + neck + abdom + thigh + forearm
+ wrist

            Df Sum of Sq    RSS    AIC

<none>                    4038.1 669.44

- neck       1     54.16 4092.2 670.46

- thigh      1     77.32 4115.4 671.74

- age        1     92.80 4130.9 672.59

- forearm    1    150.29 4188.4 675.73

- wrist      1    173.55 4211.6 676.99

- weight     1    239.75 4277.8 680.53

- abdom      1   3006.10 7044.2 793.75

> summary(model2)

Coefficients:

            Estimate Std. Error t value Pr(>|t|)

(Intercept) -33.79207    9.43053 -3.583 0.000418 ***

age           0.07180    0.03200   2.243 0.025871 *

weight       -0.12792    0.03548  -3.606 0.000385 ***

neck         -0.39624    0.23121  -1.714 0.087978 .

abdom         0.94869    0.07430  12.768  < 2e-16 ***

thigh         0.24222    0.11828   2.048 0.041776 *

forearm       0.53976    0.18906   2.855 0.004718 **

wrist        -1.63732    0.53368  -3.068 0.002427 **

Residual standard error: 4.294 on 219 degrees of
freedom  Multiple R-squared:  0.7546,
    Adjusted R-squared:  0.7467   F-statistic: 96.18
on 7 and 219 DF,  p-value: < 2.2e-16

> rmse(model2$fitted.values, train$siri)

[1] 4.217687

> rmse(predict(model2,test), test$siri)

[1] 4.342456
```

The step-wise elimination process removed 7 variables and adjusted R-squared gets slightly higher, which shows a better fit in the model. Also the performance of the model on test data improved from 4.395 to 4.342, the difference of RMSE between training data and test data decreased.

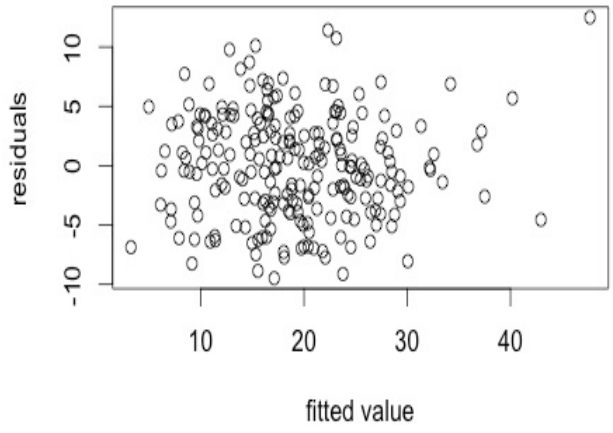
3. Principal component regression

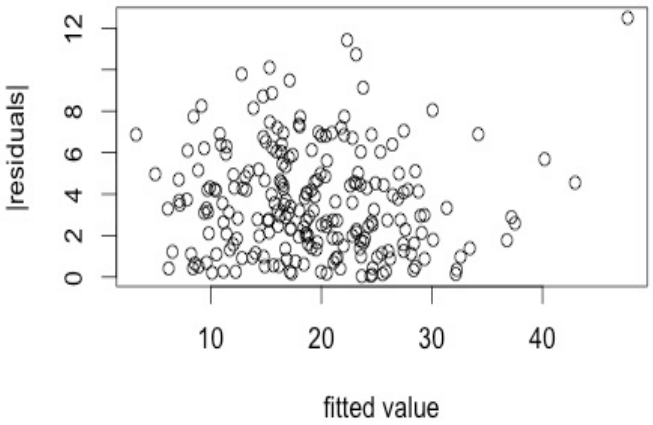
```
> ##PCA

> library(tools)

> library(HSAUR2)

> library(MVA)

> fatpca <- prcomp(train[,-1],scale.=TRUE)

> round(fatpca$sdev, 3)

 [1] 2.974 1.172 1.038 0.828 0.773 0.577 0.548 0.520
0.428 0.367 0.279 0.235 0.215 0.154

> ##Use cross validation for choosing number of PCs

> library(pls)

> set.seed(123)

> modpcr1 <- pcr(siri~ .,data= train, ncomp
=14,validation = "CV",segments = 5)

> rmsCV = RMSEP(modpcr1, estimate = 'CV')
```

```
> which.min(rmsCV$val)

[1] 8

> ##plot the RMSE

> plot(rmsCV$val, xlab = "No. of PC", ylab = "CV RMS")
```




```
> ##compare the RMSE on training data and test data

> rmse(modpcr1$fitted.values[,,7],train$siri)

[1] 4.399395

> yfit <- predict(modpcr1, newdata = test, ncomp =7)

> rmse(yfit,test$siri)

[1] 4.346266

> plot(yfit, yfit-test$siri, xlab = "fitted value", ylab = "residuals")

> plot(yfit, abs(yfit-test$siri), xlab = "fitted value", ylab = "|residuals|")
```



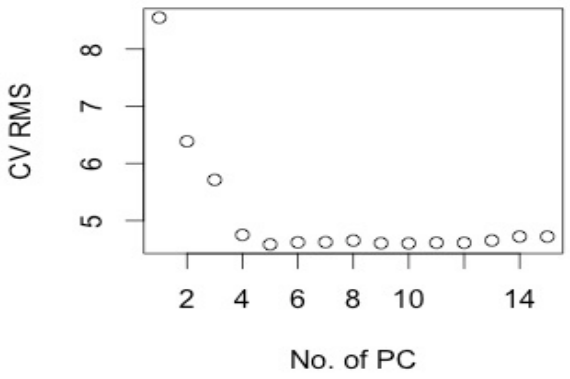


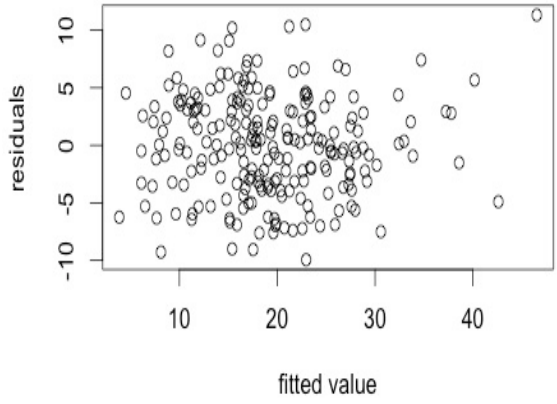
After computing PCA on training data, we find that the standard deviation of the first PC is nearly 2.5 times that of the second PC, and the SD dropped off sharply after the second PC. As the cross validation suggests, we choose the first 7 PCs to fit the model. It shows that the RMSE of test data is even smaller than that of training data, though RMSE of training data in this model is relatively larger than in the two models above.

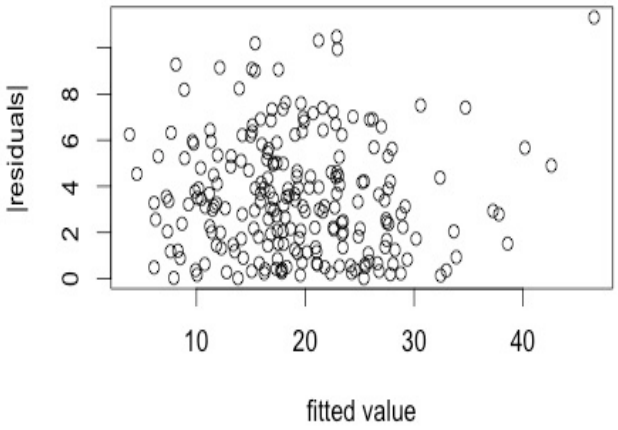
Checking the plots of residuals vs. fitted, we find that residuals are vertically symmetrically scattered around 0, and |residual| is also symmetrically distributed. Then the model shows a constant variance.

4. Partial least squares

```
##Partial least squares

set.seed(123)

modpls <- plsr(siri~., data =train, ncomp
=14,validation= "CV")

pls_rmsCV = RMSEP(modpls, estimate = 'CV')

which.min(pls_rmsCV$val)

[1] 5

plot(pls_rmsCV$val, xlab = "No. of PC", ylab = "CV
RMS")

##compare the RMSE on training data and test data

rmse(modpls$fitted.values[,,4],train$siri)

[1] 4.344006

ypred <- predict(modpls, newdata = test)

rmse(test$siri, ypred[,,4])

[1] 4.392838

plot(ypred[,,4], ypred[,,4]-test$siri, xlab = "fitted
value", ylab = "residuals")

plot(ypred[,,4], abs(ypred[,,4]-test$siri), xlab = "fitted

value", ylab = "|residuals|")
```







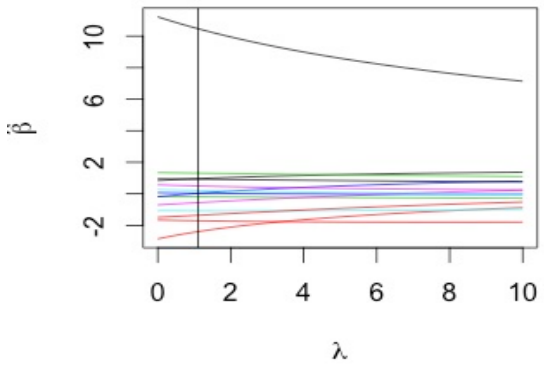
According to cross validation, we choose four components in the PLS model, which is about half the number of components in PCA. The performance of PLS model on training data is better than PCA model, but the performance on test data is not good enough. The scatter of residuals towards fitted values is similar to that in the PCA model, thus the variance is fairly constant.

5. Ridge regression

```
> ##Redge regression

> library(MASS)
```

```
> ##center the training data

> modridge <- lm.ridge(siri~., lambda = seq(0,10,0.1), data =train)

> matplot(modridge$lambda, t(modridge$coef), type = "l", lty = 1, xlab = expression(lambda), ylab = expression(hat(beta)))

> ##select lambda

> which.min(modridge$GCV)

 1.1

  12

> ##compute the fitted the value

> yfit_rid <- modridge$ym + scale(train[,-1], center = modridge$xm, scale = modridge$scales)%*% modridge$coef[,12]

> rmse(yfit_rid ,train$siri)

[1] 4.183926

> ##compare to the sample

> ypred_rid <- modridge$ym + scale(test[,-1], center = modridge$xm, scale = modridge$scales)%*% modridge$coef[,12]

> rmse(ypred_rid ,test$siri)

[1] 4.281613
```

```
> select(modridge)

modified HKB estimator is 1.552127

modified L-W estimator is 4.078233

smallest value of GCV  at 1.1

> abline(v = 1.1)
```



By Redge regression, we achieve a fairly good performance on both training and test data, as the two RMSE are relatively small. And RMSE in test data is most close to that in training data, comparing to the other four models.