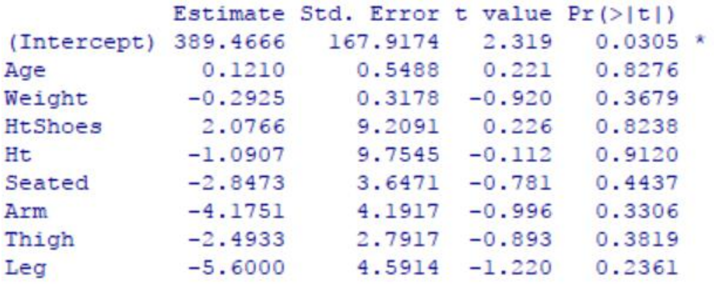
STATS 500 HOMEWORK8 YUAN YIN

At the first beginning, we separate our data in to groups, the first 30 data is for setting up the models, and the last 8 data is for test our model.

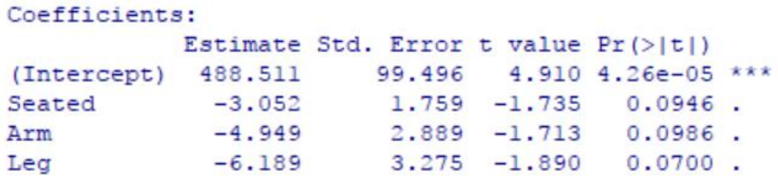
1. Next , we do the linear regression with all predictors as the original model. The result is as follows:



```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 389.4666   167.9174   2.319   0.0305 *
Age           0.1210     0.5488   0.221   0.8276
Weight       -0.2925     0.3178  -0.920   0.3679
HtShoes       2.0766     9.2091   0.226   0.8238
Ht           -1.0907     9.7545  -0.112   0.9120
Seated       -2.8473     3.6471  -0.781   0.4437
Arm          -4.1751     4.1917  -0.996   0.3306
Thigh        -2.4933     2.7917  -0.893   0.3819
Leg          -5.6000     4.5914  -1.220   0.2361
```

We can see that all the predictors are not significant. Also we check MSE for linear regression model, and the result is 697.3854, the MSE for test data is 4071.671 which is much larger than training data. It shows that linear regression model is not good enough for prediction.

2. Then, we us AIC method to select the predictors we need, so that we can make our model smaller. Using step() function and our final model is $y_{hipcenter} = \beta_0 + \beta_{seated}x_{seated} + \beta_{arm}x_{arm} + \beta_{leg}x_{leg}$ (AIC = 207.03 at this moment). And the value of coefficient is as follows:



```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  488.511     99.496   4.910 4.26e-05 ***
Seated        -3.052      1.759  -1.735   0.0946 .
Arm           -4.949      2.889  -1.713   0.0986 .
Leg           -6.189      3.275  -1.890   0.0700 .
```

This model looks more concise than our original model, now we check the MSE for this model. The result is 760.779 which is a little larger than the original one,however, MSE for test data is 3685.052, which is smaller than before. It show that this model is better on prediction.

3. Now we want to use principal component regression method to shrink our model. We use cross validation to pick the best number of PC. The result is as follows:

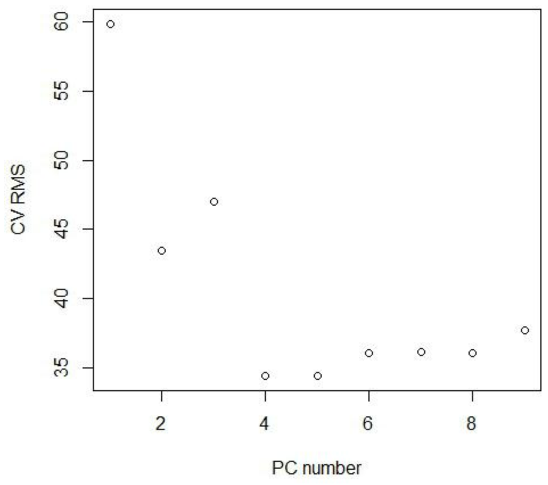


Fig 1 CV RMS vs PC number

We can see that the smallest RMS is around 4 or 5, and the truth is when we use which.min() function to find the PC number getting smallest RMS, the result is either 4 or 5. From our figure above, we pick the smallest one as k=4. so we pick our model with first 4 PCs, here is the first four PCs frequency:

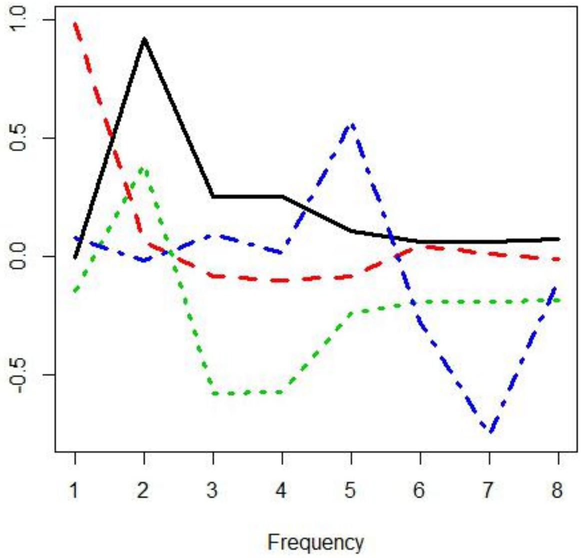


Fig 2 the frequency of first 4 PCs

Also we need to check MSE to see if this model is better or not, for training data, MSE is 765.117, for test data, MSE is 3488.767, again we find that for training data, there is a little lager than original model, and for test data, MSE is much smaller than before, and it's even smaller than AIC method. It shows that PCR method is better in prediction than original model, almost the same with AIC method.

4. Using PLS method to pick the model can take the effect of response into account. We plot the RMS with CV as follows:

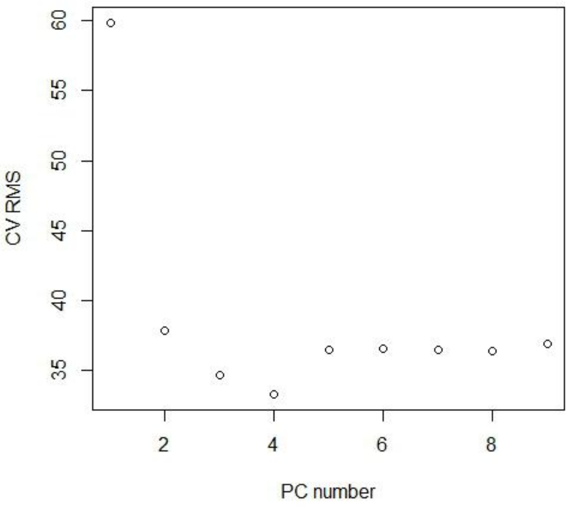


Fig 3 CV RMS vs PC number

We can see that it's quite the same with PCR method, also the smallest RMS occurs at k=4, which means we should pick the model with first four PCs. Compute MSE for training data: 733.5486, which is close to PCR method. Compute MSE for test data: 3582.222, which is almost the same with PCR method. It shows that PLS method is quite the same with PCR method.

5. There is also another method called ridge regression. We use this method to check if we can get better prediction. First we need to compute applicable λ so that we can constrain our model. Using GCV method to find it and the result is 21.52. We plot

the shink process of our model with λ goes up as follows:

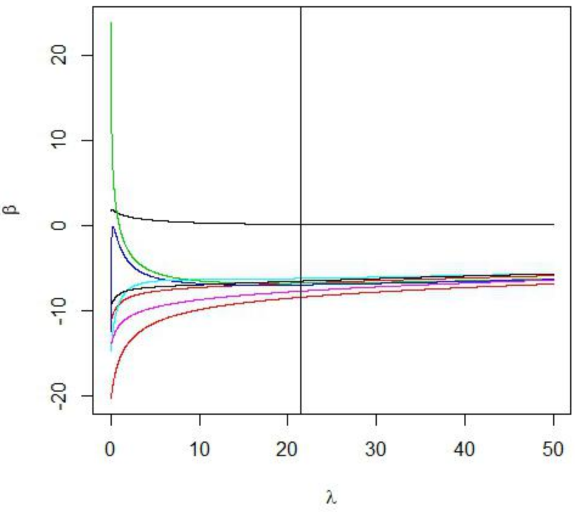
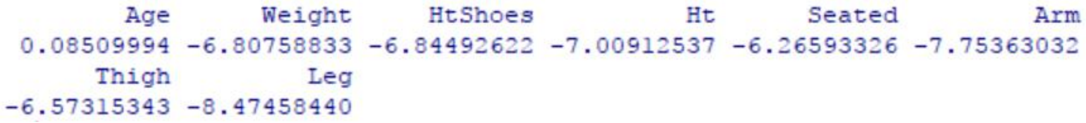


Fig 4 shink of model

The vertical is the appropriate λ we get with GCV method which is 21.52. Also we find the corresponding β with the λ we pick, and the coefficient of every predictors is as follows:



```
           Age      Weight     HtShoes          Ht      Seated         Arm 
 0.08509994 -6.80758833 -6.84492622 -7.00912537 -6.26593326 -7.75363032 
       Thigh         Leg 
-6.57315343 -8.47458440 
```

This is the coefficient of model we pick with ridge regression. Also, we need to check the MSE, the result for training data is 753.5039 and the MSE for test data is 3559.882, quite same with PCR and PLS method.

6. At last, we use Lasso method to shrink our model. Also we need to use CV to pick regularization parameter t. we plot the change of coefficient with t goes up as follows:

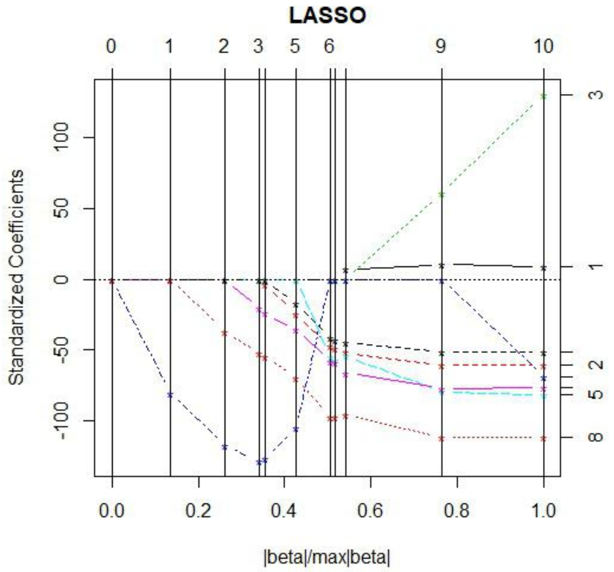


Fig 5 change of coefficients

From the figure above, we can clearly see that the coefficients shrink to 0 when t decreases to 0. We use CV method to compute the appropriate t and the result is 0.2727273, also we can confirm this from the plot below:

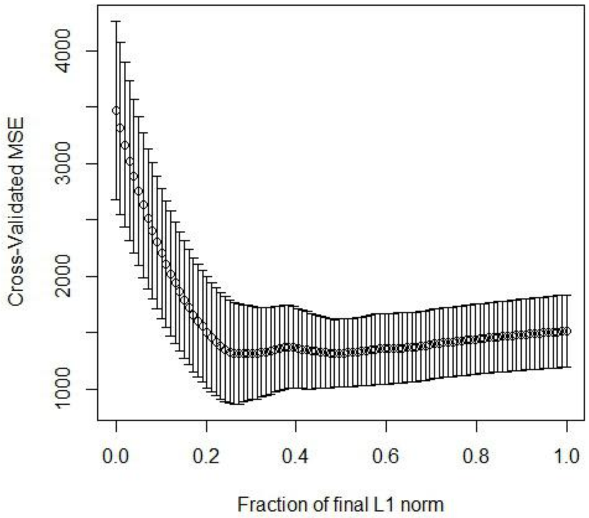
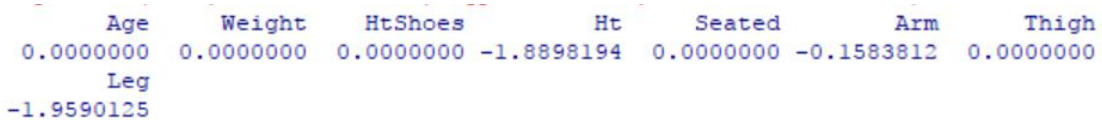


Fig 6 CV MSE vs t

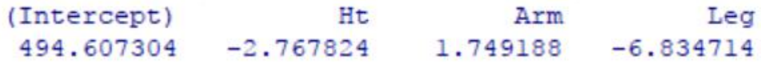
We can see that the smallest t appears around 0.3. Then we check MSE to see if this model is better at prediction. The MSE for test data is 2909.559, which is much smaller than all the model before. It shows that Lasso method gets the best model we need for prediction. To write out our model, we first check the coefficient with t = 0.2727273, the result is as follows:



```
       Age    Weight   HtShoes         Ht    Seated        Arm     Thigh
 0.0000000 0.0000000 0.0000000 -1.8898194 0.0000000 -0.1583812 0.0000000
       Leg
-1.9590125
```

We can see that only Ht, Arm and Leg is not zero, which means when t = 0.2727273, there are only these three predictor exist.

In this way, we can get our model is: $y_{hipcenter} = \beta_0 + \beta_{ht}x_{ht} + \beta_{arm}x_{arm} + \beta_{leg}x_{leg}$, and the coefficient is:



```
(Intercept)          Ht         Arm         Leg
 494.607304   -2.767824    1.749188   -6.834714
```

In conclusion, for our regression, AIC, PCR, PLS and Ridge regression method are quite the same good at prediction, but Lasso regression acts much better than the others

Appendix

```
matplot(1:8, seatpca$rot[,1:4], type = "l", xlab = "Frequency", ylab = "", lwd = 3)
library(pls)
ypcr = pcr(hipcenter ~ . , data = trainseat, ncomp = 8, validation = "CV", segments = 3)
rmsCV = RMSEP(ypcr, estimate = 'CV')
which.min(rmsCV$val)
# plot the RMSE
plot(rmsCV$val, xlab = "PC number", ylab = "CV RMS")
mse(ypcr$fitted.values[,,4], trainseat$hipcenter)
# get test error
yfit = predict(ypcr, newdata = testseat, ncomp = 4)
mse(testseat$hipcenter, yfit)

## PLS with CV
ypls = plsr(hipcenter ~ . , data = trainseat, ncomp = 8, validation = "CV")
pls_rmsCV = RMSEP(ypls, estimate = 'CV')
```

```r
plot(pls_rmsCV$val, xlab = "PC number", ylab = "CV RMS")
which.min(pls_rmsCV$val)
dim(ypls$fitted.values)
# 30  1  8
mse(ypls$fitted.values[,,4], trainseat$hipcenter)
ypred.test = predict(ypls, newdata = testseat)
dim(ypred.test)
# 8 1 8
mse(ypred.test[,,4], testseat$hipcenter)

## ridge regression with GCV
library(MASS)
yridge = lm.ridge(hipcenter ~ . , lambda = seq(0, 50, 0.01), data = trainseat)
matplot(yridge$lambda, t(yridge$coef), type = "l", lty = 1, xlab = expression(lambda), ylab =
expression(hat(beta)))
# select lambda
select(yridge)
abline(v = 21.52)
which.min(yridge$GCV)
# fitted values
yfit = yridge$ym + scale(trainseat[,-9], center = yridge$xm, scale = yridge$scales) %*% yridge$coef[,2153]
yridge$coef[,2153]
mse(yfit, trainseat$hipcenter)
# prediction
ypred = yridge$ym + scale(testseat[,-9], center = yridge$xm, scale = yridge$scales) %*% yridge$coef[,2153]
mse(ypred, testseat$hipcenter)

## Lasso
require(lars)
set.seed(123)
lmod = lars(as.matrix(trainseat[,-9]), trainseat$hipcenter)
plot(lmod)
cvlmod = cv.lars(as.matrix(trainseat[,-9]), trainseat$hipcenter)
cvlmod$index[which.min(cvlmod$cv)]
testx = as.matrix(testseat[,-9])
predlars = predict(lmod,testx, s = 0.2727273, mode = "fraction")
mse(testseat$hipcenter, predlars$fit)
predlars = predict(lmod, s = 0.2727273, type = "coef", mode = "fraction")
plot(predlars$coef, type = "h", ylab = "Coefficient")
predict(lmod, s = 0.2727273, type = "coef", mode = "fraction")$coef
coef(lm(hipcenter ~ Ht + Arm + Leg, data = seatpos))
```