STAT 500: HW9

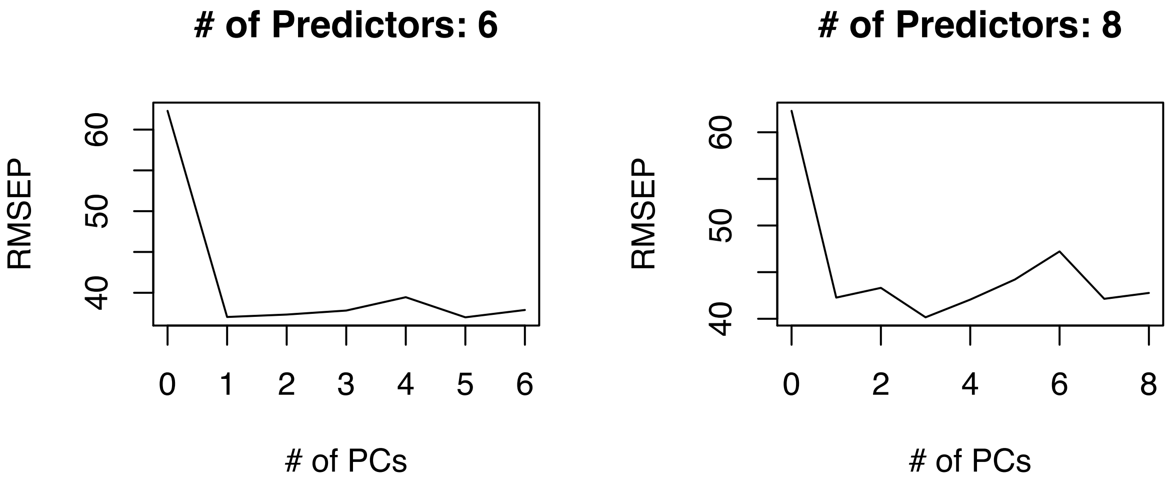
Jasmine Mou

12/05/2017

1. Using the `seatpos` data, perform a PCR analysis with `hipcenter` as the response and `HtShoes`, `Ht`, `Seated`, `Arm`, `Thigh` and `Leg` as predictors. Select an appropriate number of components and give an interpretation to those you choose. Add `Age` and `Weight` as predictors and repeat the analysis. Use both models to predict the response for predictors taking these values:

Age	Weight	HtShoes	Ht	Seated	Arm	Thigh	Leg
64.800	263.700	181.080	178.560	91.440	35.640	40.950	38.790

```r
library(pls)
rmse <- function(x, y) sqrt(mean((x - y)^2))
set.seed(123)

data(seatpos, package = "faraway")
n <- nrow(seatpos)
test_seq <- sample(n, n * 0.25)
grp_predictors_0 = c("HtShoes", "Ht", "Seated", "Arm", "Thigh", "Leg")
grp_all_0 = c("hipcenter", grp_predictors_0)
grp_predictors_1 = c("HtShoes", "Ht", "Seated", "Arm", "Thigh", "Leg", "Age",
    "Weight")
grp_all_1 = c("hipcenter", grp_predictors_1)

df_test = data.frame(matrix(c(181.08, 178.56, 91.44, 35.64, 40.95, 38.79, 64.8,
    263.7), nrow = 1))
colnames(df_test) = c("HtShoes", "Ht", "Seated", "Arm", "Thigh", "Leg", "Age",
    "Weight")

cal_pcr <- function(grp_all, grp_predictors) {
    # assign train and test randomly for cross validation
    sp <- seatpos[, grp_all]
    p <- ncol(sp) - 1
    train_sp <- sp[-test_seq, ]
    test_sp <- sp[test_seq, ]

    ## use pcr + RMSEP plot
    pcr_sp <- pcr(hipcenter ~ ., data = train_sp, validation = "CV", ncomp = p)
    pcrCV_sp <- RMSEP(pcr_sp, estimate = "CV")
    plot(pcrCV_sp, xlab = "# of PCs", ylab = "RMSEP", main = paste("# of Predictors:",
        p))
    pcr_sp$nc_sp <- which.min(pcrCV_sp$val) - 1
    pcr_sp$ypred_sp <- predict(pcr_sp, test_sp[, grp_predictors], ncomp = pcr_sp$nc_sp)
    pcr_sp$rmse <- rmse(pcr_sp$ypred_sp, test_sp$hipcenter)

    pcr_sp$exp <- explvar(pcr_sp)
    # pcr_sp$ld <- loadings(pcr_sp) loadingplot(pcr_sp, comps=1:3,
    # legendpos='topright')
    pcr_sp$acc <- 0
    for (i in 1:pcr_sp$nc_sp) {
        pcr_sp$acc = pcr_sp$acc + pcr_sp$exp[i]
    }

    return(pcr_sp)
}
par(mfrow = c(1, 2))
pcr_sp_0 <- cal_pcr(grp_all_0, grp_predictors_0)  # rmse = 37.34921, seed = 123
pcr_sp_1 <- cal_pcr(grp_all_1, grp_predictors_1)  # rmse = 35.81502, seed = 123
```



```
pcr_sp_fit <- predict(pcr_sp_1, df_test, ncomp = pcr_sp_1$nc_sp)
```

Perform PCR without scaling and use cross-validation over the train data to choose the number of components. Draw a RMSE vs number of components plot for visualization.

With only 6 predictors, we choose the number of components to be 5 according to the plot. The proportion of variability explained by the first 5 components is 99.94%. The RMSE of the model over the test data is 58.85.

With 8 predictors, we choose the number of components to be 3 according to the plot. The proportion of variability explained by the first 3 components is 99.46%. The RMSE of the model over the test data is 35.82.

The predicted response of the sample is -199.5221.

2. Take the `fat` data, and use the percentage of body fat, `siri`, as the response and the other variables, except `brozek` and `density` 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:

```
data(fat, package = "faraway")
n <- nrow(fat)
p <- ncol(fat) - 3
test_seq <- seq(10, n, by = 10)
grp_predictors_fat = colnames(fat)[!colnames(fat) %in% c("brozek", "density")]
train_fat <- fat[-test_seq, grp_predictors_fat]
test_fat <- fat[test_seq, grp_predictors_fat]
```

(a) Linear regression with all predictors.

```
lm_fat <- lm(siri ~ ., data = train_fat)
rmse_lm_fat_train <- rmse(lm_fat$fit, train_fat$siri)
rmse_lm_fat_test <- rmse(predict(lm_fat, test_fat[, -1]), test_fat$siri)
```

(b) Linear regression with variables selected using AIC.

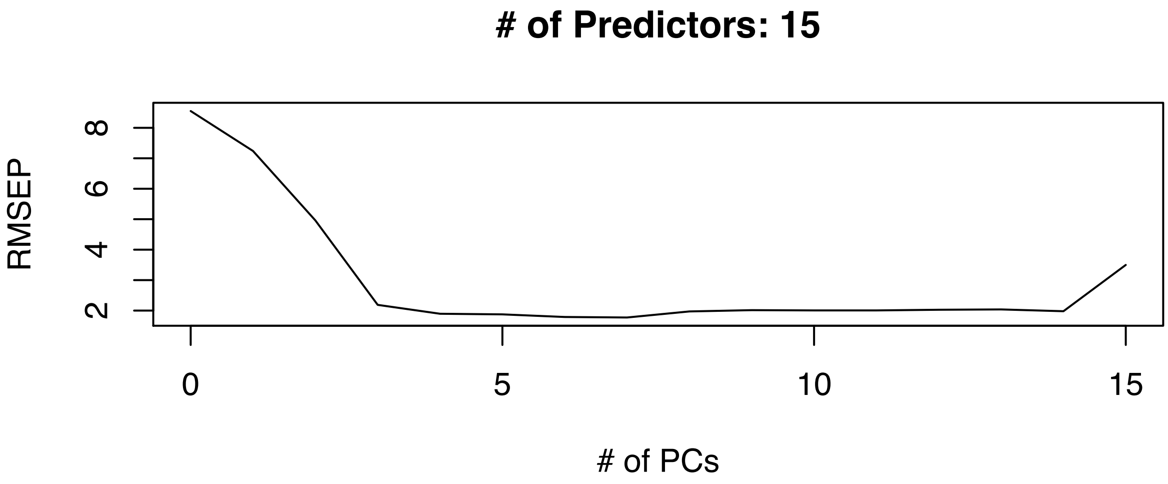
```
lmAIC_fat <- step(lm_fat)
```

```
## Start:  AIC=214.36
## siri ~ age + weight + height + adipos + free + neck + chest +
##     abdom + hip + thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - hip      1       0.0  506.9 212.37
## - neck     1       0.1  507.0 212.39
## - age      1       1.0  507.9 212.81
## - wrist    1       1.1  508.0 212.84
## - knee     1       3.1  510.0 213.75
## - height   1       3.6  510.4 213.94
## <none>                  506.9 214.36
## - biceps   1       5.3  512.2 214.73
## - ankle    1       5.7  512.6 214.89
## - chest    1      22.2  529.0 222.07
## - forearm  1      23.8  530.7 222.77
## - abdom    1      26.5  533.4 223.92
## - thigh    1      30.8  537.7 225.75
## - adipos   1      48.8  555.7 233.21
## - weight   1     582.4 1089.3 386.01
## - free     1    3456.8 3963.7 679.21
##
## Step:  AIC=212.37
## siri ~ age + weight + height + adipos + free + neck + chest +
##     abdom + thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - neck     1       0.1  507.0 210.40
## - age      1       1.0  507.9 210.81
## - wrist    1       1.1  508.0 210.86
## - knee     1       3.2  510.1 211.80
## - height   1       3.5  510.4 211.95
## <none>                  506.9 212.37
## - biceps   1       5.3  512.2 212.73
## - ankle    1       5.7  512.6 212.89
## - chest    1      23.1  530.0 220.50
## - forearm  1      23.8  530.7 220.78
## - abdom    1      27.9  534.9 222.55
## - thigh    1      34.2  541.2 225.21
## - adipos   1      50.3  557.2 231.85
## - weight   1     683.9 1190.8 404.23
## - free     1    3488.9 3995.8 679.05
##
## Step:  AIC=210.4
## siri ~ age + weight + height + adipos + free + chest + abdom +
##     thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - age      1       1.1  508.1 208.88
## - wrist    1       1.3  508.3 208.99
## - knee     1       3.1  510.1 209.80
## - height   1       3.6  510.6 210.02
## <none>                  507.0 210.40
## - biceps   1       5.4  512.4 210.80
## - ankle    1       5.6  512.6 210.89
## - chest    1      23.2  530.2 218.55
## - forearm  1      24.6  531.6 219.15
## - abdom    1      28.0  535.0 220.60
## - thigh    1      34.4  541.4 223.29
## - adipos   1      50.8  557.8 230.07
## - weight   1     689.6 1196.6 403.34
## - free     1    3532.0 4039.0 679.49
##
## Step:  AIC=208.88
## siri ~ weight + height + adipos + free + chest + abdom + thigh +
##     knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - wrist    1       2.9  511.0 208.19
## - height   1       3.3  511.4 208.35
## - knee     1       4.5  512.5 208.87
## <none>                  508.1 208.88
## - ankle    1       5.2  513.2 209.18
## - biceps   1       6.0  514.0 209.53
## - forearm  1      23.6  531.6 217.18
## - chest    1      24.2  532.3 217.46
## - abdom    1      33.7  541.8 221.48
## - thigh    1      35.3  543.3 222.12
## - adipos   1      51.1  559.1 228.63
## - weight   1     699.1 1207.2 403.34
## - free     1    3598.0 4106.0 681.23
##
## Step:  AIC=208.19
## siri ~ weight + height + adipos + free + chest + abdom + thigh +
##     knee + ankle + biceps + forearm
##
##           Df Sum of Sq    RSS    AIC
## - height   1       3.8  514.8 207.89
## <none>                  511.0 208.19
## - knee     1       5.7  516.7 208.72
## - ankle    1       6.9  517.9 209.24
## - biceps   1       7.0  518.0 209.30
## - chest    1      23.8  534.8 216.53
## - forearm  1      27.7  538.7 218.16
## - thigh    1      32.4  543.4 220.13
## - abdom    1      37.3  548.3 222.19
## - adipos   1      49.3  560.3 227.11
## - weight   1     696.5 1207.5 401.40
## - free     1    3798.4 4309.4 690.20
##
## Step:  AIC=207.89
## siri ~ weight + adipos + free + chest + abdom + thigh + knee +
##     ankle + biceps + forearm
##
##           Df Sum of Sq    RSS    AIC
## <none>                  514.8 207.89
## - knee     1       5.1  519.9 208.12
## - ankle    1       7.4  522.2 209.11
## - biceps   1       7.5  522.4 209.18
## - chest    1      24.0  538.9 216.25
## - forearm  1      28.8  543.6 218.23
## - thigh    1      30.0  544.8 218.73
## - abdom    1      39.1  553.9 222.49
## - adipos   1      86.6  601.4 241.18
## - weight   1     819.8 1334.7 422.13
## - free     1    3809.4 4324.2 688.98
```

```
rmse_lmAIC_fat_train <- rmse(lmAIC_fat$fit, train_fat$siri)
rmse_lmAIC_fat_test <- rmse(predict(lmAIC_fat, test_fat[, -1]), test_fat$siri)
```

(c) Principal component regression.

```
pcr_fat <- pcr(siri ~ ., data = train_fat, validation = "CV", ncomp = p)
pcrCV_fat <- RMSEP(pcr_fat, estimate = "CV")
plot(pcrCV_fat, xlab = "# of PCs", ylab = "RMSEP", main = paste("# of Predictors:",
    p))
```



```
pcr_fat$nc <- which.min(pcrCV_fat$val) - 1
rmse_pcr_fat_train <- rmse(predict(pcr_fat, train_fat[, -1], ncomp = pcr_fat$nc),
    train_fat$siri)
rmse_pcr_fat_test <- rmse(predict(pcr_fat, test_fat[, -1], ncomp = pcr_fat$nc),
    test_fat$siri)
```

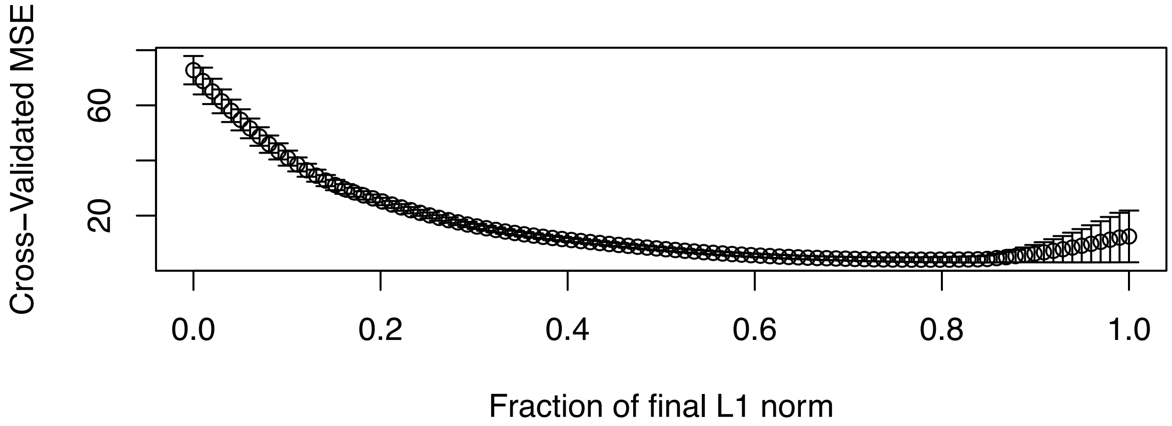
From the plot we see the number of components is chosen at n=7.

(e) Ridge Regression.

```
require(MASS)
rg_fat <- lm.ridge(siri ~ ., data = train_fat, lambda = seq(0, 5, len = 2000))
# matplot(rg_fat$lambda, coef(rg_fat), type='l', xlab=expression(lambda)
# ,ylab=expression(hat(beta)),col=1)
rg_fat$f_lambda <- which.min(rg_fat$GCV)  # 0.01127654
rmse_rg_fat_train <- rmse(cbind(1, as.matrix(train_fat[, -1])) %*% coef(rg_fat)[rg_fat$f_lambda,
    ], train_fat$siri)
rmse_rg_fat_test <- rmse(cbind(1, as.matrix(test_fat[, -1])) %*% coef(rg_fat)[rg_fat$f_lambda,
    ], test_fat$siri)
```

(f) Lasso.

```
require(lars)
train_fat_lsx <- as.matrix(train_fat[, -1])
train_fat_lsy <- train_fat[, 1]
test_fat_lsx <- as.matrix(test_fat[, -1])
ls_fat <- lars(x = train_fat_lsx, y = train_fat_lsy)
lsCV_fat <- cv.lars(x = train_fat_lsx, y = train_fat_lsy)
```



```
ls_fat$t <- lsCV_fat$index[which.min(lsCV_fat$cv)]
# ls_fat$coef <- predict(ls_fat, type='coef', s=ls_fat$t,
# mode='fraction')$coef # to get the coefficients plot(ls_fat)
rmse_ls_fat_train <- rmse(predict(ls_fat, train_fat_lsx, s = ls_fat$t, mode = "fraction")$fit,
    train_fat$siri)
rmse_ls_fat_test <- rmse(predict(ls_fat, test_fat_lsx, s = ls_fat$t, mode = "fraction")$fit,
    test_fat$siri)
```

Use the models you find to predict the response in the test sample. Make a report on the performances of the models.
Model performance: in terms of the test data, the performance rank is PCR > LASSO > AIC > Ridge > LR.

```
df_report = data.frame(LR = c(rmse_lm_fat_train, rmse_lm_fat_test), AIC = c(rmse_lmAIC_fat_train,
    rmse_lmAIC_fat_test), PCR = c(rmse_pcr_fat_train, rmse_pcr_fat_test), Ridge = c(rmse_rg_fat_train,
    rmse_rg_fat_test), LASSO = c(rmse_ls_fat_train, rmse_ls_fat_test))
rownames(df_report) = c("train", "test")
kable(df_report, digits = 4)
```

	LR	AIC	PCR	Ridge	LASSO
train	1.4943	1.506	1.6107	1.4944	1.6213
test	1.1315	1.122	1.0489	1.1280	1.0935

~~~
---
title: 'STAT 500: HW9'
author: "Jasmine Mou"
date: "12/05/2017"
output: pdf_document
---

1. Using the `seatpos` data, perform a PCR analysis with `hipcenter` as
the response and `HtShoes`, `Ht`, `Seated`, `Arm`, `Thigh` and `Leg` as
predictors. Select an appropriate number of components and give an
interpretation to those you choose. Add `Age` and `Weight` as predictors
and repeat the analysis. Use both models to predict the response for
predictors taking these values:
```{r echo=FALSE, results='asis', warning=FALSE}
library(knitr)
df = data.frame(matrix(c('64.800 ', '263.700 ', '181.080 ', '178.560 ',
'91.440 ', '35.640 ', '40.950 ', '38.790'), nrow=1))
colnames(df)=c("Age", "Weight", "HtShoes", "Ht", "Seated", "Arm",
"Thigh", "Leg")
kable(df)
```

```{r, tidy=TRUE, message=FALSE, warning=FALSE, fig.height=3}
library(pls)
rmse <- function(x,y) sqrt(mean((x-y)^2))
set.seed(123)

data(seatpos, package="faraway")
n <- nrow(seatpos)
test_seq <- sample(n, n*0.25)
grp_predictors_0 =  c("HtShoes", "Ht", "Seated", "Arm", "Thigh", "Leg")
grp_all_0 = c("hipcenter", grp_predictors_0)
grp_predictors_1 =  c("HtShoes", "Ht", "Seated", "Arm", "Thigh", "Leg",
"Age", "Weight")
grp_all_1 = c("hipcenter", grp_predictors_1)

df_test = data.frame(matrix(c(181.080, 178.560, 91.440, 35.640, 40.950,
38.790, 64.800, 263.700), nrow=1))
colnames(df_test)=c("HtShoes", "Ht", "Seated", "Arm", "Thigh", "Leg",
"Age", "Weight")

cal_pcr<-function(grp_all, grp_predictors){
  # assign train and test randomly for cross validation
        sp <- seatpos[, grp_all]
        p <- ncol(sp) - 1
        train_sp <- sp[-test_seq,]
        test_sp <- sp[test_seq,]

        ## use pcr + RMSEP plot
        pcr_sp <- pcr(hipcenter ~., data=train_sp, validation="CV",
~~~

```
ncomp=p)
        pcrCV_sp <- RMSEP(pcr_sp, estimate="CV")
        plot(pcrCV_sp, xlab="# of PCs", ylab="RMSEP", main=paste("# of
Predictors:", p))
        pcr_sp$nc_sp <- which.min(pcrCV_sp$val)-1
        pcr_sp$ypred_sp <- predict(pcr_sp, test_sp[,grp_predictors],
ncomp=pcr_sp$nc_sp)
        pcr_sp$rmse <- rmse(pcr_sp$ypred_sp, test_sp$hipcenter)

        pcr_sp$exp <- explvar(pcr_sp)
        # pcr_sp$ld <- loadings(pcr_sp)
        # loadingplot(pcr_sp, comps=1:3, legendpos="topright")
        pcr_sp$acc <- 0
        for(i in 1:pcr_sp$nc_sp){
          pcr_sp$acc = pcr_sp$acc + pcr_sp$exp[i]
        }

        return(pcr_sp)
}
par(mfrow=c(1,2))
pcr_sp_0 <- cal_pcr(grp_all_0, grp_predictors_0) # rmse = 37.34921, seed
= 123
pcr_sp_1 <- cal_pcr(grp_all_1, grp_predictors_1) # rmse = 35.81502, seed
= 123
pcr_sp_fit <- predict(pcr_sp_1, df_test, ncomp=pcr_sp_1$nc_sp)
```

*Perform PCR without scaling and use cross-validation over the train data to choose the number of components. Draw a RMSE vs number of components plot for visualization. *

*With only 6 predictors, we choose the number of components to be `r pcr_sp_0$nc_sp` according to the plot. The proportion of variability explained by the first `r pcr_sp_0$nc_sp` components is `r round(pcr_sp_0$acc,2)`%. The RMSE of the model over the test data is `r round(pcr_sp_0$rmse,2)`. *

*With 8 predictors, we choose the number of components to be `r pcr_sp_1$nc_sp` according to the plot. The proportion of variability explained by the first `r pcr_sp_1$nc_sp` components is `r round(pcr_sp_1$acc,2)`%. The RMSE of the model over the test data is `r round(pcr_sp_1$rmse,2)`. *

The predicted response of the sample is `r round(pcr_sp_fit,4)`.

2. Take the `fat` data, and use the percentage of body fat, `siri`, as the response and the other variables, except `brozek` and `density` 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:
```{r, tidy=TRUE}
data(fat, package="faraway")
n <- nrow(fat)
p <- ncol(fat) - 3
test_seq <- seq(10, n, by=10)
grp_predictors_fat = colnames(fat)[!colnames(fat) %in% c("brozek",
"density")]
train_fat <- fat[-test_seq, grp_predictors_fat]
test_fat <- fat[test_seq, grp_predictors_fat]
```
(a) Linear regression with all predictors.
```{r, tidy=TRUE}
lm_fat <- lm(siri~., data=train_fat)
rmse_lm_fat_train <- rmse(lm_fat$fit, train_fat$siri)
rmse_lm_fat_test <- rmse(predict(lm_fat, test_fat[,-1]), test_fat$siri)
```
(b) Linear regression with variables selected using AIC.
```{r, tidy=TRUE}
lmAIC_fat <- step(lm_fat)
rmse_lmAIC_fat_train <- rmse(lmAIC_fat$fit, train_fat$siri)
rmse_lmAIC_fat_test <- rmse(predict(lmAIC_fat, test_fat[,-1]),
test_fat$siri)
```
(c) Principal component regression.
```{r, tidy=TRUE, fig.height=3}
pcr_fat <- pcr(siri ~., data=train_fat, validation="CV", ncomp=p)
pcrCV_fat <- RMSEP(pcr_fat, estimate="CV")
plot(pcrCV_fat, xlab="# of PCs", ylab="RMSEP", main=paste("# of
Predictors:", p))
pcr_fat$nc <- which.min(pcrCV_fat$val)-1
rmse_pcr_fat_train <- rmse(predict(pcr_fat, train_fat[,-1],
ncomp=pcr_fat$nc), train_fat$siri)
rmse_pcr_fat_test <- rmse(predict(pcr_fat, test_fat[,-1],
ncomp=pcr_fat$nc), test_fat$siri)
```
*From the plot we see the number of components is chosen at n=`r
pcr_fat$nc`.*

(e) Ridge Regression.
```{r, tidy=TRUE, message=FALSE}
require(MASS)
rg_fat <- lm.ridge(siri~. , data=train_fat, lambda = seq(0, 5,
len=2000))
# matplot(rg_fat$lambda, coef(rg_fat), type="l", xlab=expression(lambda)
,ylab=expression(hat(beta)),col=1)
rg_fat$f_lambda <- which.min(rg_fat$GCV) # 0.01127654
rmse_rg_fat_train <- rmse(cbind(1,as.matrix(train_fat[,-1])) %*%
coef(rg_fat)[rg_fat$f_lambda,], train_fat$siri)
rmse_rg_fat_test <- rmse(cbind(1,as.matrix(test_fat[,-1])) %*%
coef(rg_fat)[rg_fat$f_lambda,], test_fat$siri)
~~~

~~~
```

(f) Lasso.
```{r, tidy=TRUE, message=FALSE, fig.height=3}
require(lars)
train_fat_lsx <- as.matrix(train_fat[,-1])
train_fat_lsy <- train_fat[,1]
test_fat_lsx <- as.matrix(test_fat[,-1])
ls_fat <- lars(x=train_fat_lsx, y=train_fat_lsy)
lsCV_fat <- cv.lars(x=train_fat_lsx, y=train_fat_lsy)
ls_fat$t <- lsCV_fat$index[which.min(lsCV_fat$cv)]
# ls_fat$coef <- predict(ls_fat, type="coef", s=ls_fat$t,
mode="fraction")$coef # to get the coefficients
# plot(ls_fat)
rmse_ls_fat_train <- rmse(predict(ls_fat, train_fat_lsx, s=ls_fat$t,
mode="fraction")$fit, train_fat$siri)
rmse_ls_fat_test <- rmse(predict(ls_fat, test_fat_lsx, s=ls_fat$t,
mode="fraction")$fit, test_fat$siri)
```

Use the models you find to predict the response in the test sample. Make
a report on the performances of the models.
*Model performance: in terms of the test data, the performance rank is
PCR > LASSO > AIC > Ridge > LR.*
```{r, tidy=TRUE}
df_report = data.frame(LR=c(rmse_lm_fat_train, rmse_lm_fat_test),
AIC=c(rmse_lmAIC_fat_train, rmse_lmAIC_fat_test),
PCR=c(rmse_pcr_fat_train, rmse_pcr_fat_test), Ridge=c(rmse_rg_fat_train,
rmse_rg_fat_test), LASSO=c(rmse_ls_fat_train, rmse_ls_fat_test))
rownames(df_report) = c("train", "test")
kable(df_report, digits=4)
```
~~~