Problem 4

Candidate 10029

03 juni, 2022

4)

```
id <- "1kGOLsnKAOUq2lWKlMjhAF8h71scOWcLO" # google file ID
d.bodyfat <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download", id))[,-c(1)]



set.seed(1234)
training_set_size <- floor(0.80 * nrow(d.bodyfat))

samples <- sample(1: nrow(d.bodyfat),training_set_size , replace=F)
d.body.train <- d.bodyfat[samples,]
d.body.test <- d.bodyfat[-samples,]

str(d.bodyfat)
```

```
## 'data.frame':    252 obs. of  14 variables:
##  $ BodyFat: num  12.3 6.1 25.3 10.4 28.7 20.9 19.2 12.4 4.1 11.7 ...
##  $ Age    : int  23 22 22 26 24 24 26 25 25 23 ...
##  $ Weight : num  154 173 154 185 184 ...
##  $ Height : num  67.8 72.2 66.2 72.2 71.2 ...
##  $ Neck   : num  36.2 38.5 34 37.4 34.4 39 36.4 37.8 38.1 42.1 ...
##  $ Chest  : num  93.1 93.6 95.8 101.8 97.3 ...
##  $ Abdomen: num  85.2 83 87.9 86.4 100 94.4 90.7 88.5 82.5 88.6 ...
##  $ Hip    : num  94.5 98.7 99.2 101.2 101.9 ...
##  $ Thigh  : num  59 58.7 59.6 60.1 63.2 66 58.4 60 62.9 63.1 ...
##  $ Knee   : num  37.3 37.3 38.9 37.3 42.2 42 38.3 39.4 38.3 41.7 ...
##  $ Ankle  : num  21.9 23.4 24 22.8 24 25.6 22.9 23.2 23.8 25 ...
##  $ Biceps : num  32 30.5 28.8 32.4 32.2 35.7 31.9 30.5 35.9 35.6 ...
##  $ Forearm: num  27.4 28.9 25.2 29.4 27.7 30.6 27.8 29 31.1 30 ...
##  $ Wrist  : num  17.1 18.2 16.6 18.2 17.7 18.8 17.7 18.8 18.2 19.2 ...
```

```
head(d.bodyfat)
```

```
##   BodyFat Age Weight Height Neck Chest Abdomen   Hip Thigh Knee Ankle Biceps
## 1    12.3  23 154.25  67.75 36.2  93.1    85.2  94.5  59.0 37.3  21.9   32.0
## 2     6.1  22 173.25  72.25 38.5  93.6    83.0  98.7  58.7 37.3  23.4   30.5
## 3    25.3  22 154.00  66.25 34.0  95.8    87.9  99.2  59.6 38.9  24.0   28.8
## 4    10.4  26 184.75  72.25 37.4 101.8    86.4 101.2  60.1 37.3  22.8   32.4
## 5    28.7  24 184.25  71.25 34.4  97.3   100.0 101.9  63.2 42.2  24.0   32.2
## 6    20.9  24 210.25  74.75 39.0 104.5    94.4 107.8  66.0 42.0  25.6   35.7
```

```
##   Forearm Wrist
## 1    27.4  17.1
## 2    28.9  18.2
## 3    25.2  16.6
## 4    29.4  18.2
## 5    27.7  17.7
## 6    30.6  18.8
```

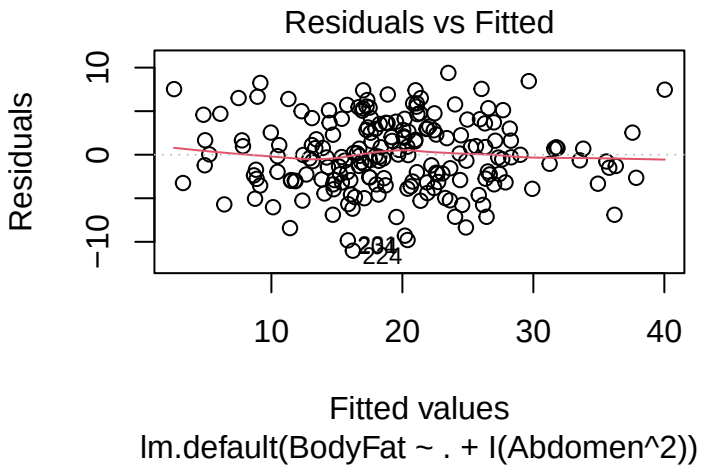
a)

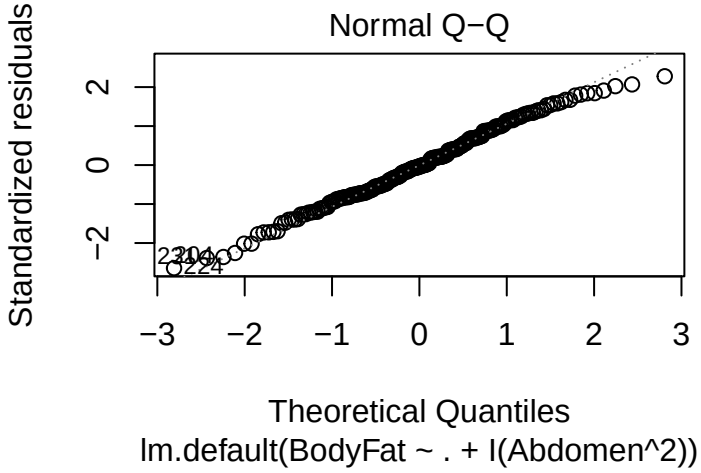
1)

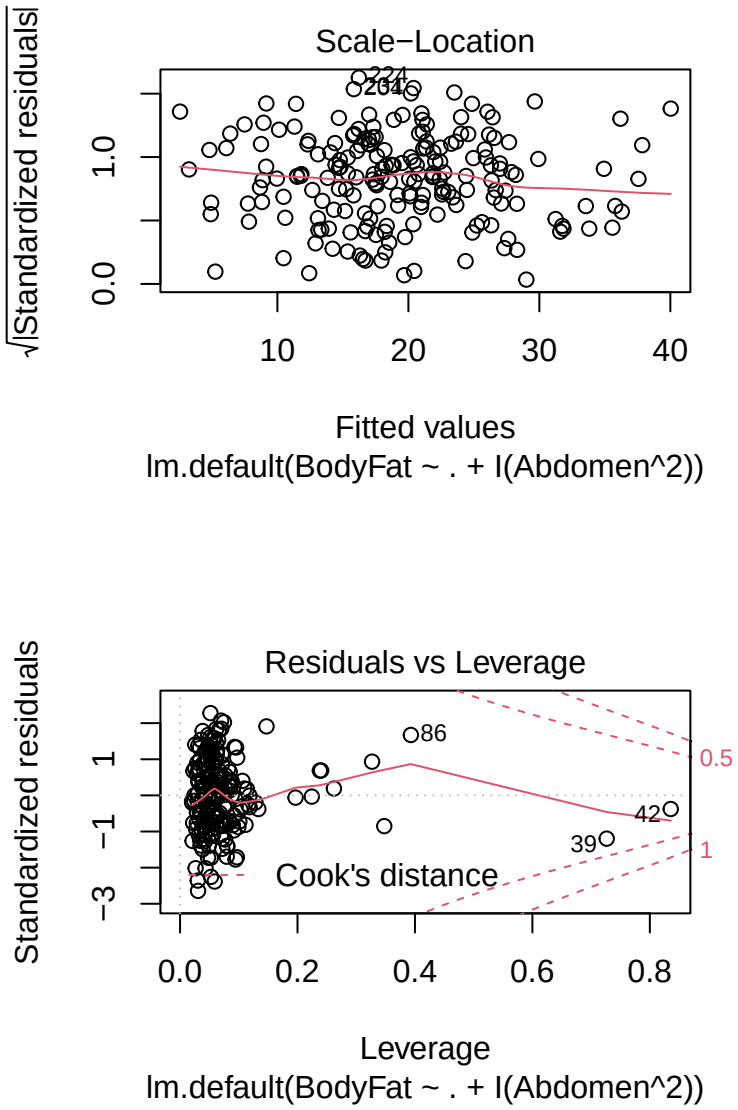
```
lm.fit<-lm(BodyFat~. + I(Abdomen^2), data=d.body.train)

summary(lm.fit)
```

```
##
## Call:
## lm.default(formula = BodyFat ~ . + I(Abdomen^2), data = d.body.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0198  -2.9100  -0.1409   2.9595   9.3920
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -40.455180  22.053225  -1.834 0.068187 .
## Age            0.067543   0.035660   1.894 0.059765 .
## Weight        -0.041957   0.061638  -0.681 0.496910
## Height        -0.066017   0.100857  -0.655 0.513560
## Neck          -0.551787   0.257295  -2.145 0.033285 *
## Chest         -0.084785   0.107160  -0.791 0.429830
## Abdomen        1.675292   0.300530   5.574 8.63e-08 ***
## Hip           -0.093039   0.167284  -0.556 0.578760
## Thigh          0.087891   0.156380   0.562 0.574768
## Knee          -0.106406   0.276498  -0.385 0.700799
## Ankle          0.112232   0.230627   0.487 0.627087
## Biceps         0.350943   0.201611   1.741 0.083391 .
## Forearm        0.332821   0.215280   1.546 0.123807
## Wrist         -2.084782   0.580179  -3.593 0.000418 ***
## I(Abdomen^2)  -0.003919   0.001571  -2.494 0.013490 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.234 on 186 degrees of freedom
## Multiple R-squared:  0.7559, Adjusted R-squared:  0.7375
## F-statistic: 41.15 on 14 and 186 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(1,1))
plot(lm.fit)
```


Residuals vs Fitted
Residuals
10
0
−10
10
20
30
40
Fitted values
lm.default(BodyFat ~ . + I(Abdomen^2))


Normal Q−Q
Standardized residuals
2
0
−2
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm.default(BodyFat ~ . + I(Abdomen^2))



2) $R^2 = 0.7689$ This number tells us how much of the variance our predictors can explain. It will always be between 0 and 1 and we wish for it to be as close to 1 as possible. As we have 0.779 which is a fairly high number, this suggests that our predictors can explain much and that the correlation between predictors and response is significant. The adjusted R^2 is a version of R^2 that penalizes including many predictors. Usually, if we add more and more predictors the R^2 will continue to increase, weh nin reality we are just adding noise. Thus, we use the adjusted R^2 to account for number of predictors. This is in our case 0.7623, which is slightly lower but still close to the R^2. This I interpret to mean that our model, even though it consists of many predictors, is a fairly good fit.

3) The Tukey Anscombe plot is residuals vs fitted.This addresses both the independency of residuals, constant variance of the residuals and the expected value. We assume constant variance and expected value of the residuals to be 0. That means it should form a horisontal band around 0, with equal variation. From the plot we see that this seems to be the case, even though we have a few outliers, and the data at high fitted values is somewhat not sufficient. The QQ-plot addresses the assumptions of the residuals to be normally distributed as it plots the standardized residuals to the theoretical quantiles.

Again we see that they form a decent line of $y = x$ which is what we want, and as mentioned above, the data at the ends seems to be a bit off.

b)

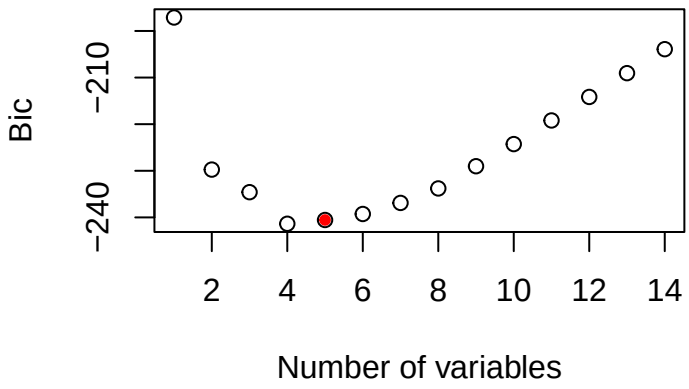
```
fwd.fit =regsubsets(BodyFat ~. + I(Abdomen^2), data=d.body.train, nvmax=15, method="forward") #Fit for
reg.summary = summary(fwd.fit)
plot(reg.summary$bic, xlab="Number of variables", ylab = "Bic" ) #Want low BIC
which.min(reg.summary$bic)
```

```
## [1] 4
```

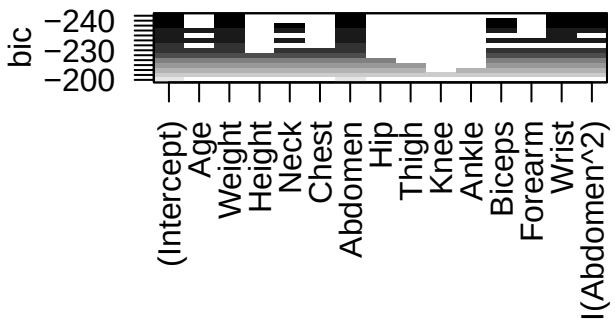
```
min(reg.summary$bic)
```

```
## [1] -241.3626
```

```
points(5,reg.summary$bic[5], pch=20, col="red" )
```




```
#We should use 5 predictors

plot(fwd.fit, scale="bic")
```




```
coef(fwd.fit, id=5)
```

```
##   (Intercept)        Weight       Abdomen        Biceps         Wrist
## -63.676319769  -0.116740911   1.789657155   0.364083703  -1.968437005
##  I(Abdomen^2)
##  -0.004341575
```

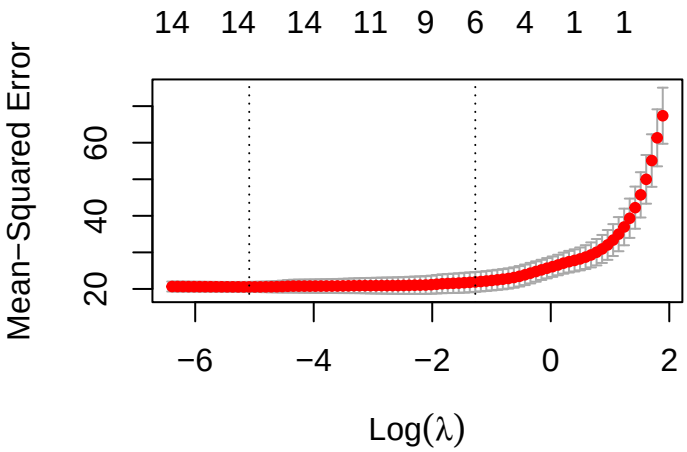
```
#We see that we should use these four coefficients

fwd.lm<-lm(BodyFat~ Weight + Height + Abdomen + Wrist + I(Abdomen^2), data=d.body.train)

preds<-predict(fwd.lm, newdata=d.body.test)

mse=mean((preds-d.body.test$BodyFat)^2)
mse
```

```
## [1] 17.97967
```

The mean squared error is 23.81113

c)

```
xmat<-model.matrix(BodyFat~. + I(Abdomen^2), data=d.body.train)[, -1]
y_train<-d.body.train$BodyFat


xmat_pred<-model.matrix(BodyFat~. + I(Abdomen^2), data=d.body.test)[, -1]
y_test <- d.body.test$BodyFat

set.seed(4268)

cv.lasso<-cv.glmnet(xmat, y_train, nfolds = 5)
plot(cv.lasso)
```




```
lambdalasso<-cv.lasso$lambda.1se

lasso.fit = glmnet(xmat, y_train, lambda = lambdalasso, alpha=1)

lasso.pred=predict(lasso.fit, s=lambdalasso, newx=xmat_pred)

lasso.mse=mean((y_test-lasso.pred)^2)

lasso.mse
```

```
## [1] 20.0992
```

The MSE is 24.54801

3)

```
lasso.fit.2<-glmnet(xmat, y_train, lambda = cv.lasso$lambda.min, alpha=1)
lasso.pred.2 <- predict(lasso.fit.2, s=cv.lasso$lambda.min, newx=xmat_pred)
lasso.mse.2=mean((y_test-lasso.pred.2)^2)

lasso.mse.2
```

```
## [1] 19.35002
```

The MSE for λ_{min} is 25.13923, so the model does not improve and so we should not use this instead of the previous lasso.

d)

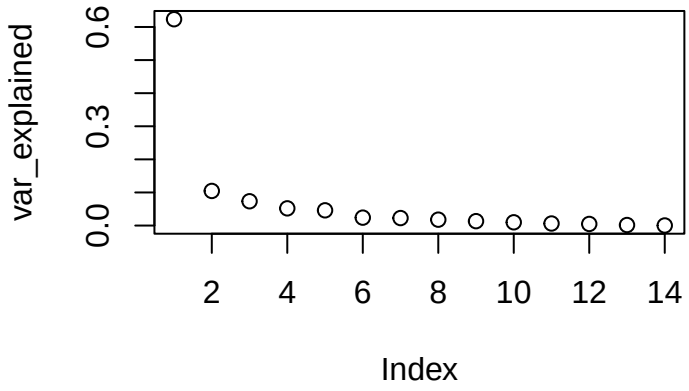
1)

```r
#Data without the response
pca.train<-model.matrix(BodyFat~. + I(Abdomen^2), data=d.body.train)[, -1]


pca.out = prcomp(pca.train, scale=TRUE)
pca.out #std deviations of principle components.
```

```
## Standard deviations (1, .., p=14):
##  [1] 2.9542508 1.2101071 1.0139366 0.8532753 0.8025108 0.5787054 0.5636921
##  [8] 0.4993814 0.4314809 0.3708102 0.2950880 0.2616159 0.1499675 0.0644737
##
## Rotation (n x k) = (14 x 14):
##                       PC1          PC2         PC3         PC4          PC5
## Age           0.01620775  0.651010559 -0.55074760  0.05486054  0.108982037
## Weight        0.33098771 -0.042058621  0.03795053 -0.11440005 -0.094121418
## Height        0.07380289 -0.508744513 -0.58391244 -0.23236152 -0.503583159
## Neck          0.29491000  0.003798268 -0.13085590  0.18576781 -0.104330890
## Chest         0.30755914  0.180707239  0.01316736  0.01476561 -0.104831131
## Abdomen       0.30628284  0.273687686  0.05474262 -0.11211083 -0.156334301
## Hip           0.31491582  0.019369409  0.20739867 -0.18234386 -0.088024190
## Thigh         0.29564787 -0.089787028  0.32506776 -0.08554353 -0.049491889
## Knee          0.29203718 -0.080175579 -0.03122437 -0.17667849  0.095617416
## Ankle         0.20928974 -0.242822927 -0.13042404 -0.35936437  0.758804461
## Biceps        0.28594331 -0.103534764  0.08299022  0.32062346 -0.004515423
## Forearm       0.21784572 -0.217081813 -0.04237931  0.73963489  0.129165119
## Wrist         0.26713008 -0.042478511 -0.39152038  0.10867075  0.205278081
## I(Abdomen^2) 0.30515160  0.273028380  0.07314001 -0.14149566 -0.159328116
##                       PC6          PC7          PC8           PC9         PC10
## Age           -0.08600494 -0.257196859  0.291736523 -0.0006713217  0.23479320
## Weight        -0.01844448  0.018174276 -0.022357725  0.0086800897 -0.05346202
## Height        -0.17448754  0.038792487  0.164425105  0.0595443812  0.11131027
## Neck           0.44704373  0.200251053 -0.058749849 -0.7610981138  0.13355918
## Chest         -0.16626040  0.335830481 -0.001165281  0.0313507207 -0.53616690
## Abdomen       -0.19933864  0.176301228 -0.052730875  0.0607883970  0.06845562
## Hip           -0.05202417 -0.087425717 -0.104972134  0.1379047295  0.23144193
## Thigh          0.14185433 -0.305800850  0.079333500  0.0920723839  0.51302866
## Knee          -0.13335424 -0.719823777 -0.058748898 -0.2721335243 -0.45520147
## Ankle         -0.15156676  0.291677191  0.207914657 -0.0952257071  0.11278618
## Biceps         0.33781825 -0.004544419  0.704246324  0.2976429599 -0.20313713
## Forearm       -0.53729546 -0.038865317 -0.132835185 -0.0320167761  0.17309406
## Wrist          0.44540088 -0.023916571 -0.547562397  0.4570789864 -0.05128385
## I(Abdomen^2) -0.17267313  0.203880991 -0.060789225  0.0572101592  0.09526627
##                      PC11         PC12         PC13         PC14
## Age           0.10091060 -0.17864374  0.04272845  0.002247991
## Weight       -0.08673885 -0.29909712  0.86937011 -0.091655574
## Height        0.05712724  0.02814894 -0.08940503  0.014816063
## Neck         -0.03877122 -0.02663874 -0.08766365 -0.009461630
## Chest         0.58540499 -0.23595244 -0.18118168  0.035890774
## Abdomen      -0.17194000  0.44775306 -0.07872604 -0.687790630
## Hip          -0.33854139 -0.65801178 -0.41566815 -0.046036066
## Thigh         0.60772857  0.16372565  0.01486553  0.033570775
## Knee         -0.09672784  0.16629629 -0.08994082  0.034742680
## Ankle         0.01117963  0.02740232 -0.03671452 -0.002895607
```

```
## Biceps       -0.21989690  0.07559488 -0.06121388  0.005718966
## Forearm      -0.01609032 -0.01539729  0.01192142  0.020309882
## Wrist         0.01048166  0.09089123 -0.02878808  0.008418577
## I(Abdomen^2) -0.25122774  0.34433295  0.02260115  0.715516066
```

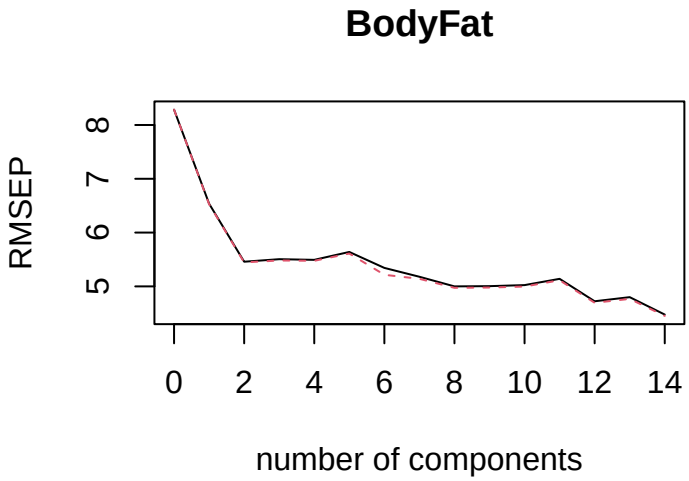
```
var_explained = pca.out$sdev^2 / sum(pca.out$sdev^2)

plot(var_explained)
```



We see that the first two components explain much of the variance, so we would only need these two.

2)

```
pcr.fit<-pcr(BodyFat~. + I(Abdomen^2), data=d.body.train, scale=TRUE, validation="CV", ncomp=14)

validationplot(pcr.fit)
```



```
summary(pcr.fit)
```

```
## Data:    X dimension: 201 14
##  Y dimension: 201 1
## Fit method: svdpc
## Number of components considered: 14
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           8.285    6.538    5.459    5.505    5.493     5.64    5.343
## adjCV        8.285    6.527    5.446    5.481    5.475     5.61    5.215
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       5.179    5.000    5.004     5.023     5.140     4.723     4.799
## adjCV    5.141    4.973    4.978     4.997     5.114     4.695     4.767
##        14 comps
## CV        4.476
## adjCV     4.451
##
## TRAINING: % variance explained
##          1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X          62.34     72.8    80.14    85.34    89.94    92.34    94.61    96.39
## BodyFat    39.68     59.2    60.71    60.74    61.88    67.66    68.03    69.16
##          9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X          97.72     98.70     99.32     99.81     99.97    100.00
## BodyFat    69.43     69.68     69.79     73.41     73.55     75.59
```

```
cverr <- RMSEP(pcr.fit)$val[1,,]
imin <- which.min(cverr) - 1

imin
```

```
## 14 comps
##       14
```

```
preds<-predict(pcr.fit, newdata=d.body.test, ncomp=14)

mse.pcr=mean((preds-d.body.test$BodyFat)^2)
mse.pcr
```

```
## [1] 19.30452
```

From this we see that we need to use all 14 observations, which to me is strange. Usually, I would expect the error to reach a low point around 3 or atleast at 4/5 number of components, but the RMSEP does not drop sufficiently until we have used all 14 components. In the PCA, we saw that the two first PC's were enough to explain most of the variability. When we do a PCR, we do a regression based on PCA. I would suggest that when doing the regression, we found that the variables we thought we could exclude from the PCA analysis, actually made a significant difference in the regression model. Thus, the PCR suggested to include all predictors. Further, we saw that the two lasso models dit not differ much in MSE, and the second lasso model suggests to include 11 predictors, which might give further weight to the argument that in this case, many predictors could be useful. However, I do feel this result is strange, as I would have not expected the PCR to include all 14 components, which seems excessive and might lead to overfitting. The MSE is not much larger than the lasso, considering it is a unsupervised procedure so the results are suprisingly good.