

PCR: Seat position

Gustavo Cepparo and Milica Cudina

Seat Position

We reconsider the **seat position** data set. Recall that it is from the **faraway** library.

```
#install.packages("faraway")  
library(faraway)
```

The data set **seatpos** is used to predict the carseat position (**hipcenter**) based on biometric data of the driver.

```
data(seatpos)
```

This is what we got when we tried multiple linear regression.

```
lm.fit=lm(hipcenter~.,data=seatpos)  
summary(lm.fit)
```

```
##  
## Call:  
## lm(formula = hipcenter ~ ., data = seatpos)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -73.827 -22.833  -3.678   25.017   62.337   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept) 436.43213   166.57162    2.620   0.0138 *      
## Age          0.77572     0.57033    1.360   0.1843      
## Weight       0.02631     0.33097    0.080   0.9372      
## HtShoes     -2.69241     9.75304   -0.276   0.7845      
## Ht           0.60134    10.12987    0.059   0.9531      
## Seated       0.53375     3.76189    0.142   0.8882      
## Arm         -1.32807     3.90020   -0.341   0.7359      
## Thigh       -1.14312     2.66002   -0.430   0.6706      
## Leg         -6.43905     4.71386   -1.366   0.1824      
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 37.72 on 29 degrees of freedom  
## Multiple R-squared:  0.6866, Adjusted R-squared:  0.6001  
## F-statistic:  7.94 on 8 and 29 DF,  p-value: 0.00001306
```

Does PCA help? Let's import the requisite library.

```
library(stats)
```

I will reimagine my predictor variables to be the ones with the biometric data (so, we exclude **Age** and the

response).

```
data=seatpos[,2:8]
attach(data)
```

Let's look at the principal components analysis.

```
pr.out=prcomp(data,scale=TRUE)
pr.out$center
```

```
##      Weight   HtShoes      Ht   Seated      Arm   Thigh      Leg
## 155.63158 171.38947 169.08421 88.95263 32.21579 38.65526 36.26316
```

```
pr.out$scale
```

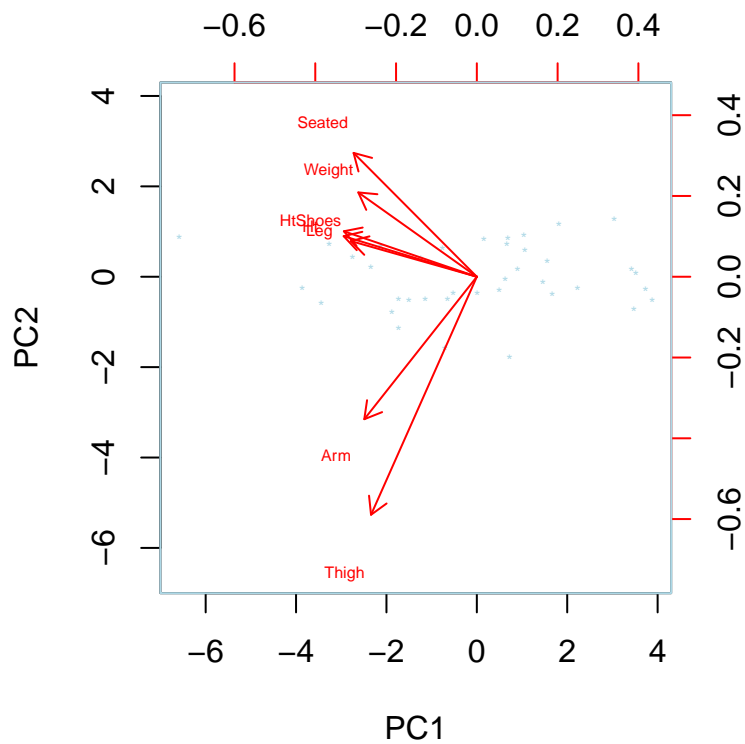
```
##      Weight   HtShoes      Ht   Seated      Arm   Thigh      Leg
## 35.781183 11.148259 11.173316 4.931791 3.371464 3.874985 3.403688
```

```
pr.out$rotation
```

```
##              PC1      PC2      PC3      PC4      PC5      PC6
## Weight -0.3669000 0.2609907 0.3583572 0.8108919 -0.09990080 0.03621216
## HtShoes -0.4115997 0.1407447 -0.1565664 -0.1352201 0.05194816 -0.52851487
## Ht      -0.4122101 0.1256624 -0.1677289 -0.1229614 0.03494283 -0.50618661
## Seated -0.3815355 0.3833142 -0.3163432 -0.1186066 0.51335147 0.57430863
## Arm     -0.3483026 -0.4409166 0.6837027 -0.2740335 0.37316939 0.04934303
## Thigh   -0.3274140 -0.7367171 -0.4882332 0.2871929 -0.08944022 0.14451162
## Leg     -0.3898319 0.1104287 0.1141931 -0.3706867 -0.75849557 0.33164299
##
##              PC7
## Weight -0.003349489
## HtShoes -0.697102953
## Ht      0.716654290
## Seated -0.000440318
## Arm     0.006629621
## Thigh   -0.017704188
## Leg     -0.009235742
```

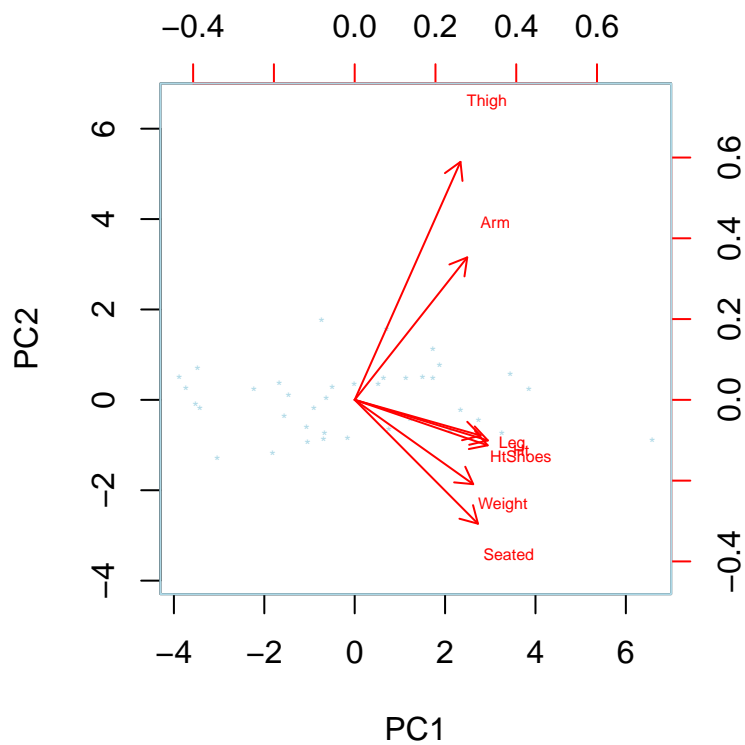
What does the biplot tell us?

```
biplot(pr.out,scale=0, cex=0.5, xlab=rep("*", length(Ht)),
       col=c("lightblue", "red"))
```



I am not happy with all the negative loadings, so let's take the negatives.

```
pr.out$rotation=-pr.out$rotation
#pr.out$rotation
pr.out$x=-pr.out$x
biplot(pr.out,scale=0, cex=0.5, xlabs=rep("*", length(Ht)),
       col=c("lightblue", "red"))
```

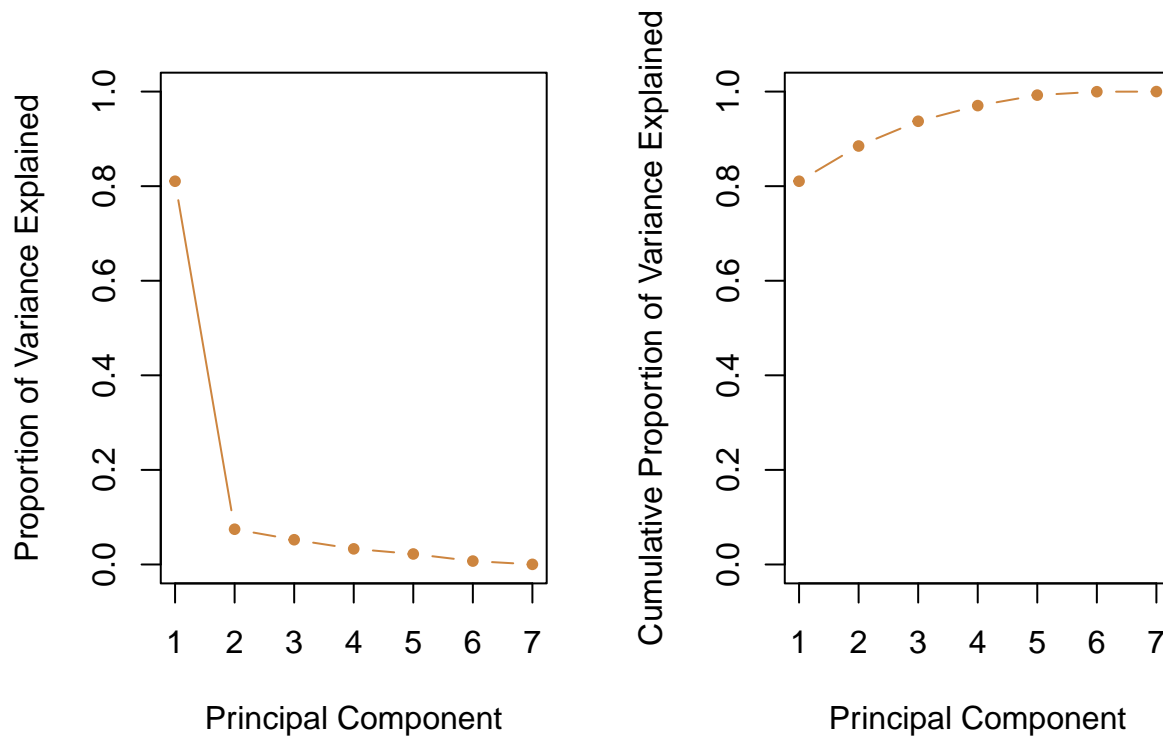


Let's look at the variance explained.

```
pr.var=pr.out$sdev^2
pve=pr.var/sum(pr.var)
pve

## [1] 0.8104205665 0.0745307740 0.0523285870 0.0330533459 0.0221980944
## [6] 0.0072232129 0.0002454193

par(mfrow = c(1, 2))
plot(pve,xlab="Principal Component",
     ylab="Proportion of Variance Explained", col="peru",
     pch=20, ylim=c(0,1),type='b')
plot(cumsum(pve),xlab="Principal Component",
     ylab="Cumulative Proportion of Variance Explained",
     col="peru", pch=20, ylim=c(0,1),type='b')
```



What would principal component regression give us?

```
library(pls)

##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##   loadings

set.seed(1)
data=seatpos[,-1]
attach(data)

## The following objects are masked from data (pos = 4):
##
##   Arm, Ht, HtShoes, Leg, Seated, Thigh, Weight
```

```

pcr.fit=pcr(hipcenter~.,data=data,scale=TRUE,validation="CV")
summary(pcr.fit)

```

```

## Data:      X dimension: 38 7
## Y dimension: 38 1
## Fit method: svdpc
## Number of components considered: 7
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##      (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           60.45   38.63   40.15   40.32   40.44   40.99   43.53
## adjCV         60.45   38.51   39.93   40.08   40.16   40.61   42.97
##      7 comps
## CV           44.07
## adjCV        43.49
##
## TRAINING: % variance explained
##      1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X           81.04   88.50   93.73   97.03   99.25   99.98  100.00
## hipcenter    62.00   62.54   63.50   64.85   66.51   66.61   66.66

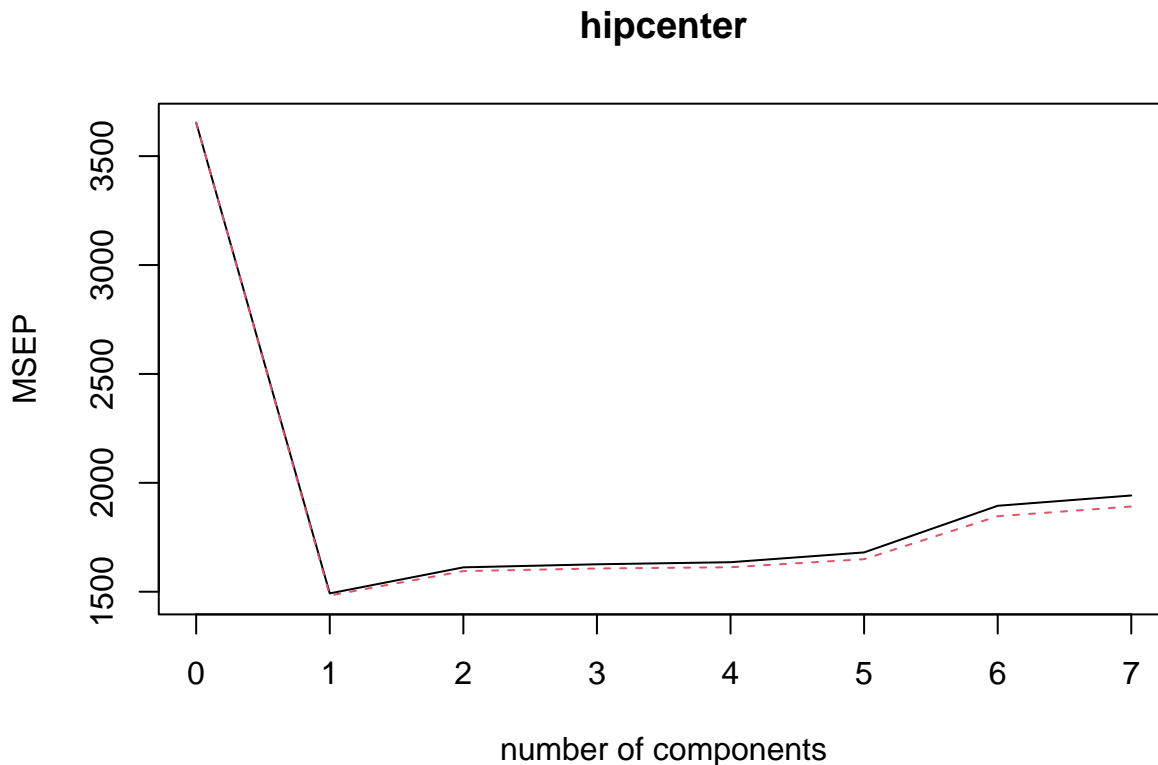
```

Exactly **one** component works wonderfully. Let's confirm with the validation plot.

```

validationplot(pcr.fit,val.type="MSEP")

```



What if we did the training and testing instead?

```

set.seed(1)
train.ind <- sample(nrow(data), floor(nrow(data)*0.6))
training <- data[train.ind,]

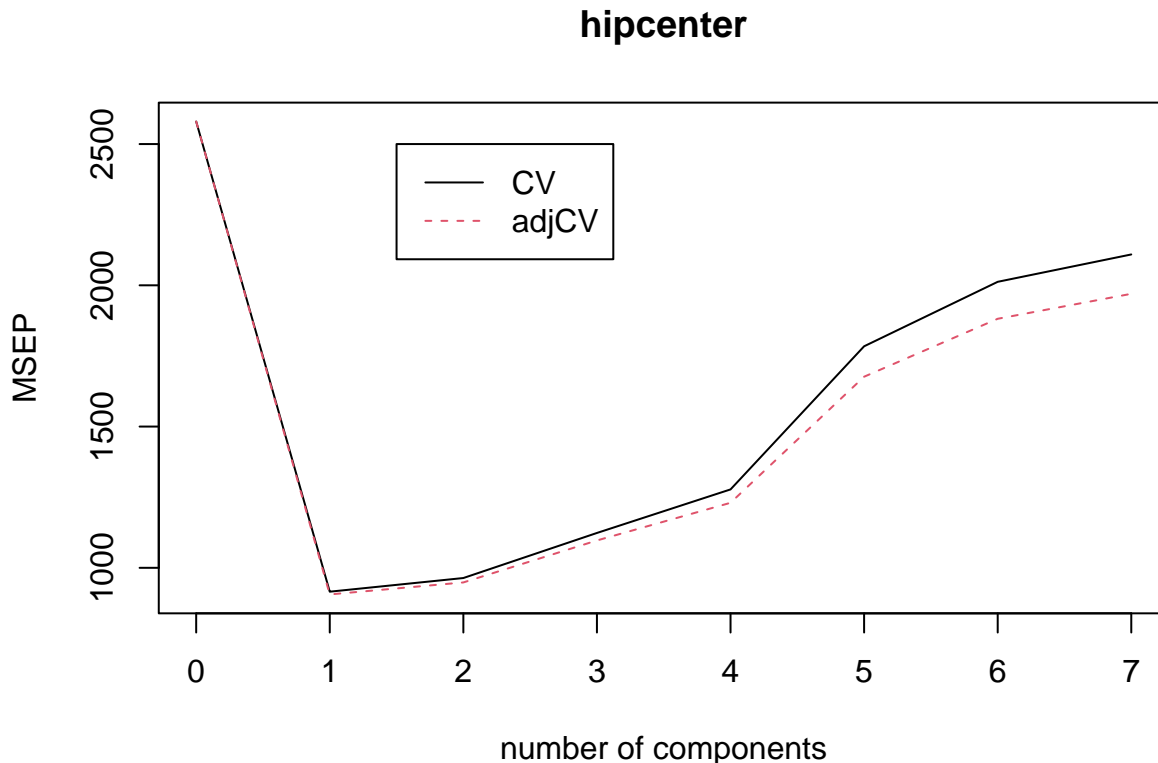
```

```
test <- data[-train.ind,]
```

```
pcr.fit=pcr(hipcenter~.,data=training,scale=TRUE, validation="CV")
summary(pcr.fit)
```

```
## Data:      X dimension: 22 7
## Y dimension: 22 1
## Fit method: svdpc
## Number of components considered: 7
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##      (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           50.79   30.26   31.05   33.52   35.74   42.24   44.86
## adjCV         50.79   30.10   30.79   33.12   35.07   40.94   43.38
##      7 comps
## CV           45.93
## adjCV        44.38
##
## TRAINING: % variance explained
##      1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X           84.77   92.09   96.00   98.58   99.62   99.97  100.00
## hipcenter    69.45   70.44   70.48   72.27   72.31   72.54   73.88
```

```
validationplot(pcr.fit,val.type="MSEP", legendpos = t(c(1.5,2500)))
```



Note

that the RMSE for 1 principal component is 30.26. This implies that the MSE is $30.26^2 = 915.6676$.

Now, we look at the performance on the testing set with 1 principal component.

```

pcr.pred=predict(pcr.fit,newdata=test,ncomp=1)
#pcr.pred
mean((pcr.pred-test$hipcenter)^2)

```

```
## [1] 2229.636
```

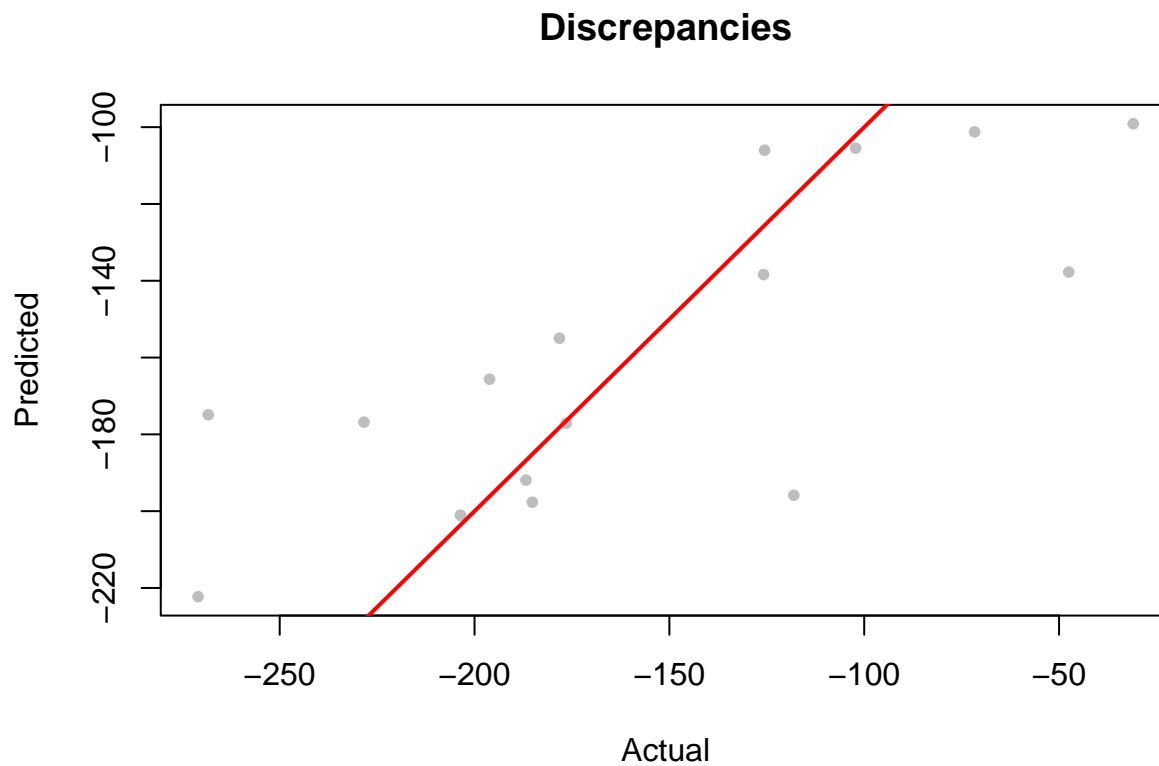
The MSE is much higher than on the training set.

It is interesting to consider the scatterplot of the predicted and actual values in the testing set.

```

plot(pcr.pred~test$hipcenter, pch=20, col="grey",
     main="Discrepancies",
     xlab="Actual", ylab="Predicted")
abline(0,1, col="red", lwd=2)

```



What about the difference?

```

plot(pcr.pred-test$hipcenter, pch=20, col="grey",
     main="Difference",
     xlab="Index", ylab="Difference")
abline(0,0, col="red", lwd=2)

```

