Remove Confounder

18.04.2022

Phuc Loi Luu, PhD

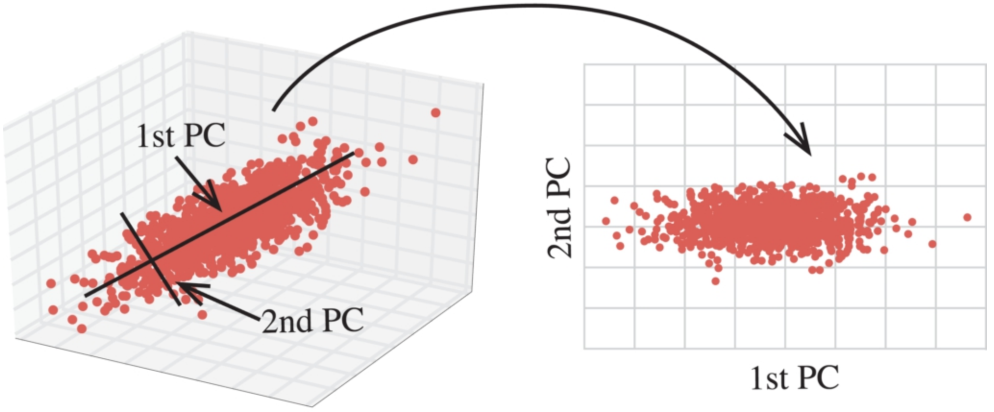
luu.p.loi@googlemail.com

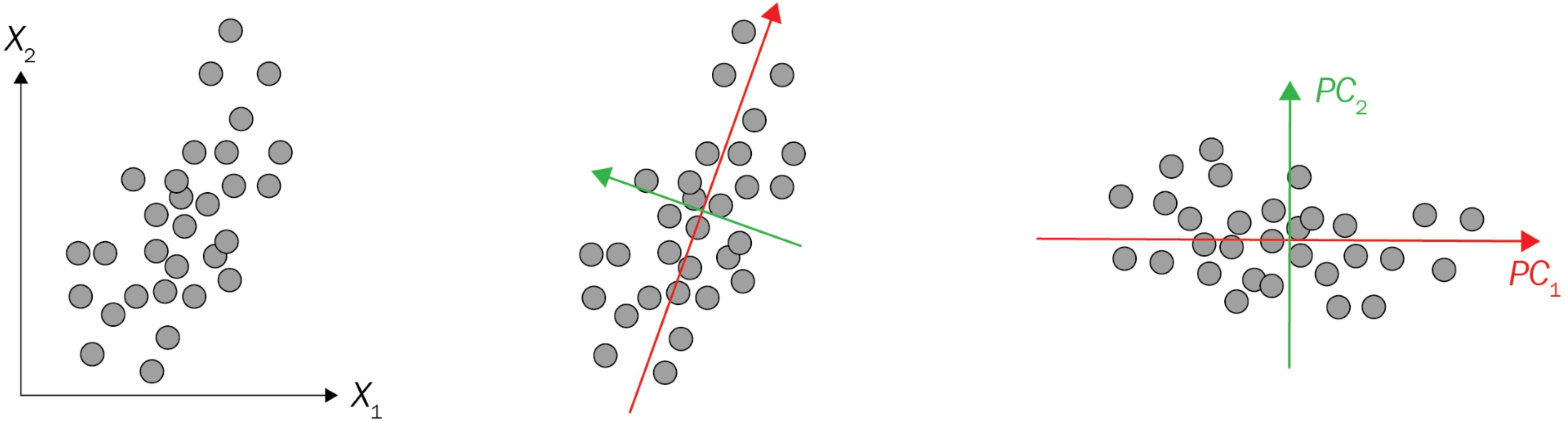
p.luu@garvan.org.au

Content

- Review PCA
- How to calculate PCA step by step in R?
- How to fully recover original data points from PCs?
- How to partially recover original data points from PCs?
- How to remove the known confounder using PCA?

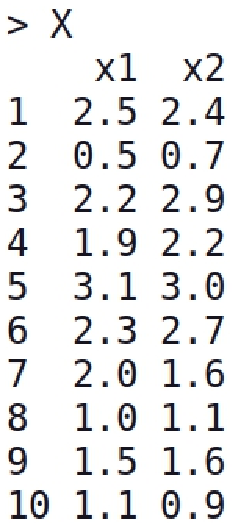
PCA: dimension reduction method



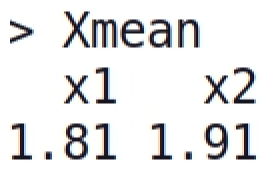


Data in feature space → Find principal components → Data in **p**rincipal **c**omponents space

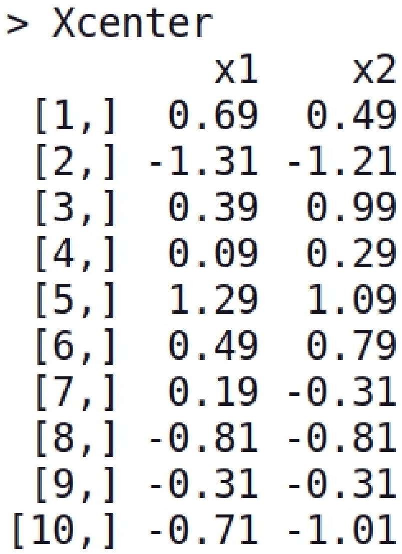
Calculate PCA step by step in R



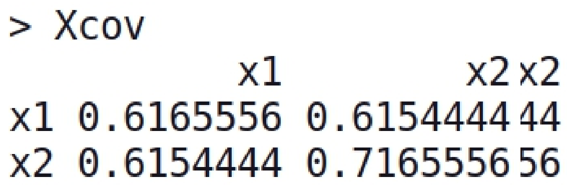
```
> X
    x1  x2
1  2.5 2.4
2  0.5 0.7
3  2.2 2.9
4  1.9 2.2
5  3.1 3.0
6  2.3 2.7
7  2.0 1.6
8  1.0 1.1
9  1.5 1.6
10 1.1 0.9
```



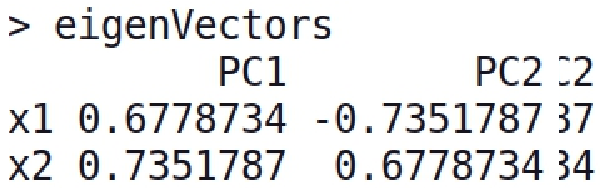
```
> Xmean
  x1   x2
1.81 1.91
```



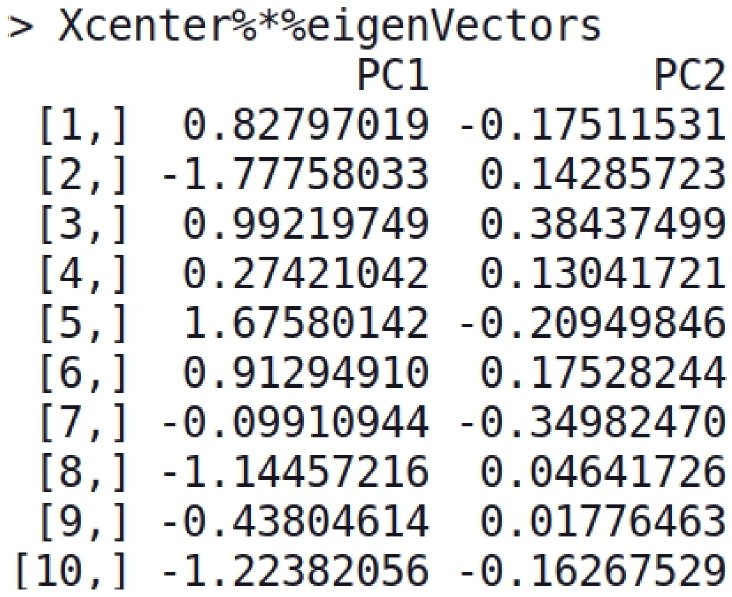
```
> Xcenter
         x1    x2
 [1,]  0.69  0.49
 [2,] -1.31 -1.21
 [3,]  0.39  0.99
 [4,]  0.09  0.29
 [5,]  1.29  1.09
 [6,]  0.49  0.79
 [7,]  0.19 -0.31
 [8,] -0.81 -0.81
 [9,] -0.31 -0.31
[10,] -0.71 -1.01
```



```
> Xcov
          x1        x2x2
x1 0.6165556 0.615444444
x2 0.6154444 0.716555656
```



```
> eigenVectors
         PC1        PC2 :2
x1 0.6778734 -0.7351787 37
x2 0.7351787  0.6778734 34
```



```
> Xcenter%*%eigenVectors
              PC1         PC2
 [1,]  0.82797019 -0.17511531
 [2,] -1.77758033  0.14285723
 [3,]  0.99219749  0.38437499
 [4,]  0.27421042  0.13041721
 [5,]  1.67580142 -0.20949846
 [6,]  0.91294910  0.17528244
 [7,] -0.09910944 -0.34982470
 [8,] -1.14457216  0.04641726
 [9,] -0.43804614  0.01776463
[10,] -1.22382056 -0.16267529
```

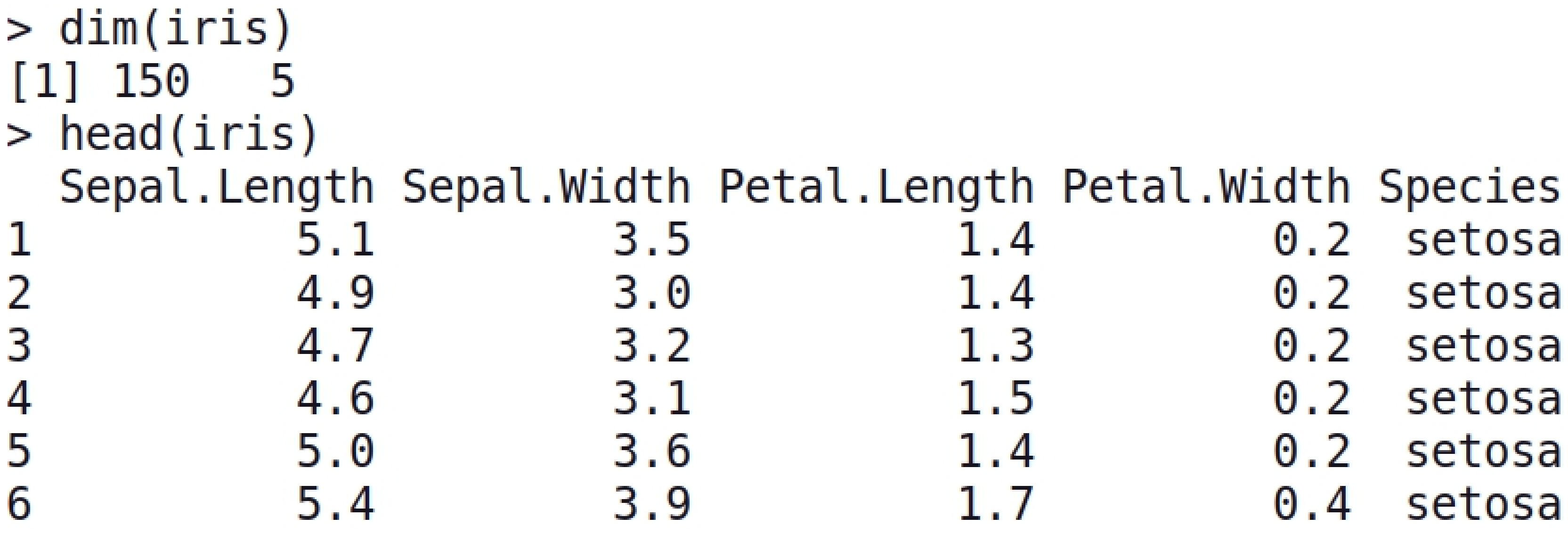
Calculate PCA step by step in R

```
# example
p1 <- "/home/phuluu/Desktop/PCA_confounder/data/data.csv"
X <- read.table(p1, sep=",", header=T)
# head(X)
#    x1  x2
# 1 2.5 2.4
# 2 0.5 0.7
# 3 2.2 2.9
# calculate mean
Xmean <- colMeans(X)
# x1   x2
# 1.81 1.91
# Centering X
Xcenter <- sapply(names(Xmean), function(x){X[,x]-Xmean[x]})
#         x1    x2
#  [1,]  0.69  0.49
#  [2,] -1.31 -1.21
#  [3,]  0.39  0.99
#  [4,]  0.09  0.29
#  [5,]  1.29  1.09
#  [6,]  0.49  0.79
#  [7,]  0.19 -0.31
#  [8,] -0.81 -0.81
#  [9,] -0.31 -0.31
# [10,] -0.71 -1.01
```

```
# Calculate covariance
Xcov <- cov(Xcenter)
#           x1        x2
# x1 0.6165556 0.6154444
# x2 0.6154444 0.7165556
# Calculate eigenvectors and eigenvalues for this
covariance matrix
eigenValues <- eigen(Xcov)$values
# eigenValues
# [1] 1.2840277 0.0490834
eigenVectors <- eigen(Xcov)$vectors
#           [,1]       [,2]
# [1,] 0.6778734 -0.7351787
# [2,] 0.7351787  0.6778734
colnames(eigenVectors) <- paste0("PC", 1:dim(X)[2])
rownames(eigenVectors) <- colnames(X)

# New coordiantes x
Xcenter%*%eigenVectors
```

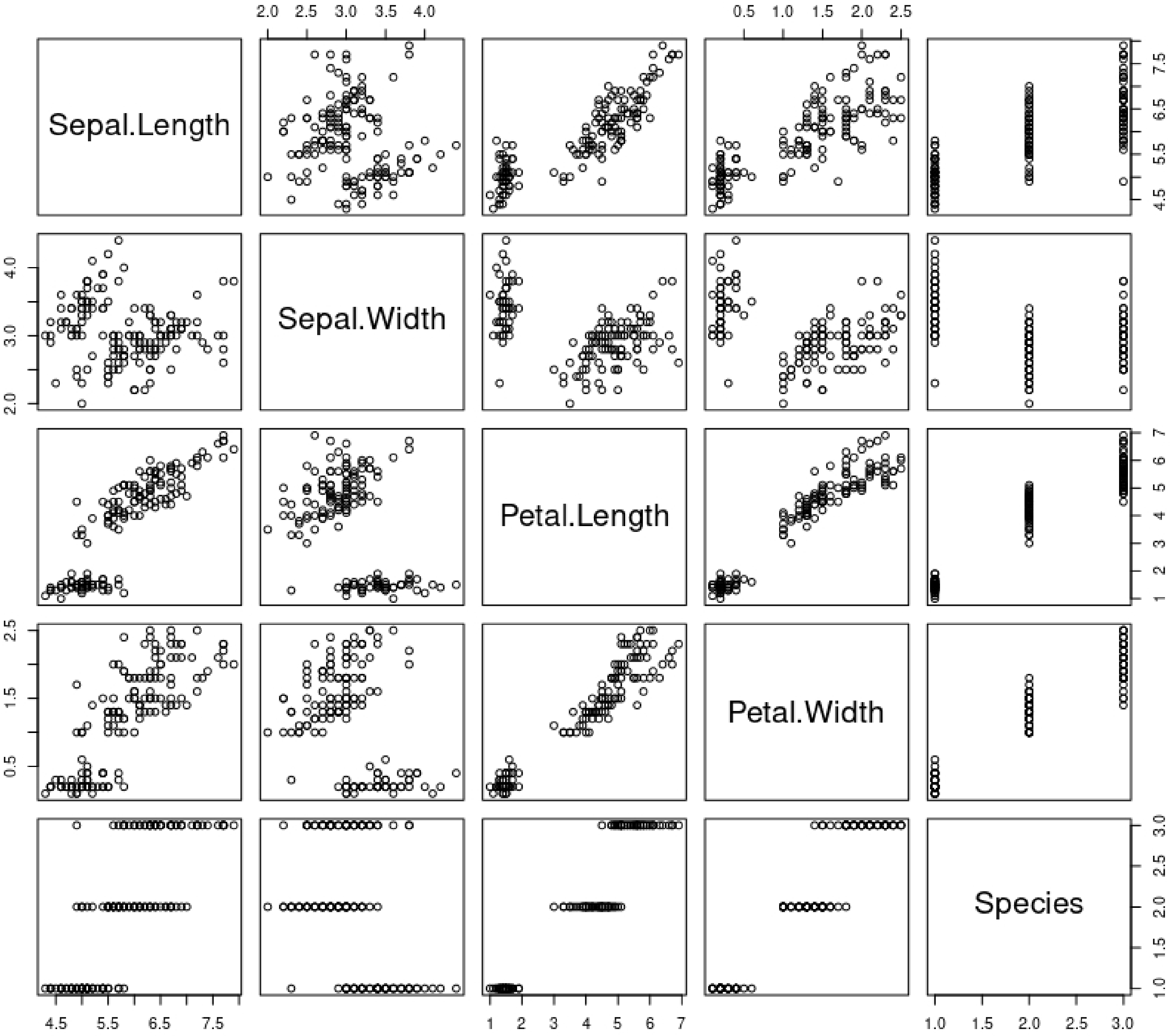
Iris Data



```
> dim(iris)
[1] 150   5
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

Iris Data

```
> pairs(iris)
```

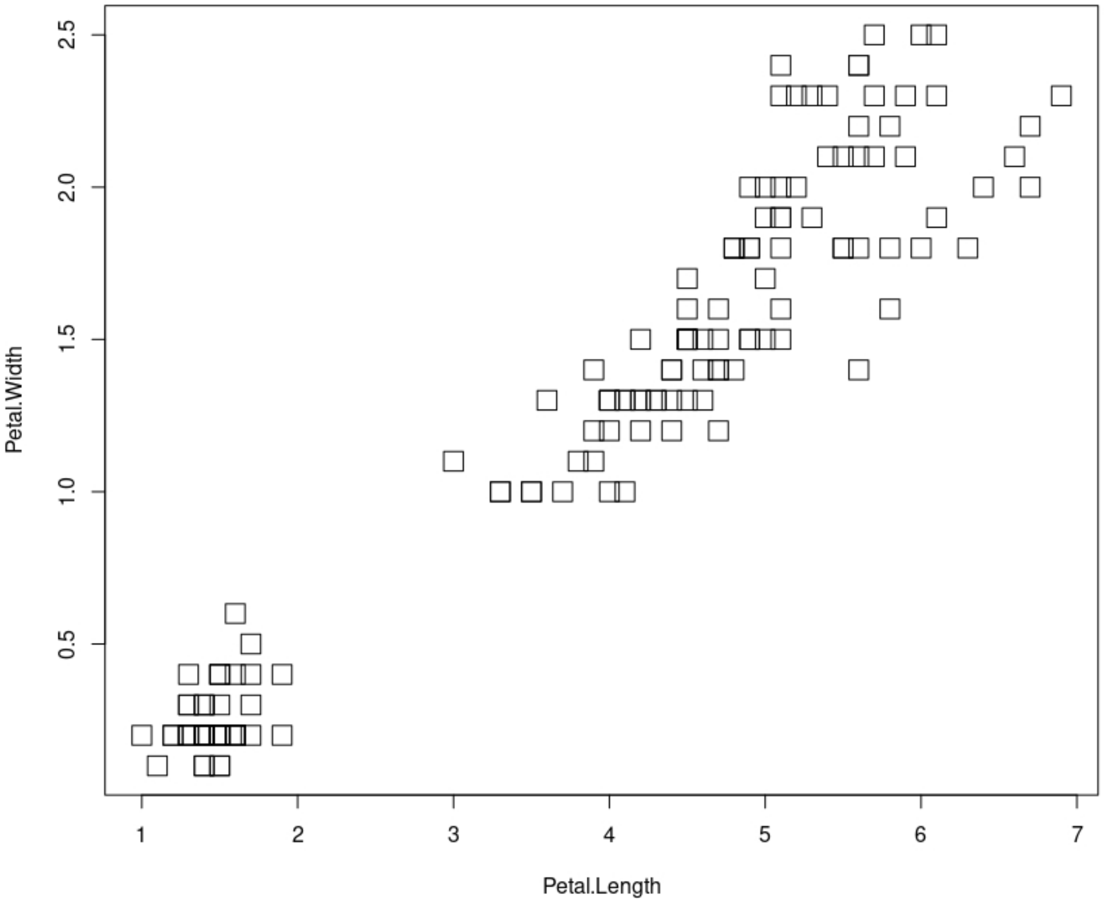


PCA on variables (Petal.Length and Petal.Width)

```
# Select 2 variables
col <- c("Petal.Length", "Petal.Width")
X <- iris[, col]
# Calculate mean
mu <- colMeans(X)
# Calculate PCA
Xpca <- prcomp(X)

# plot original data
plot(X, pch=0, col="black", cex=2)
```

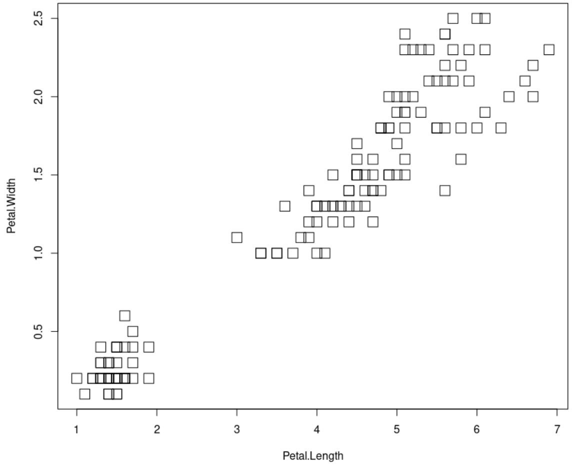
Original data



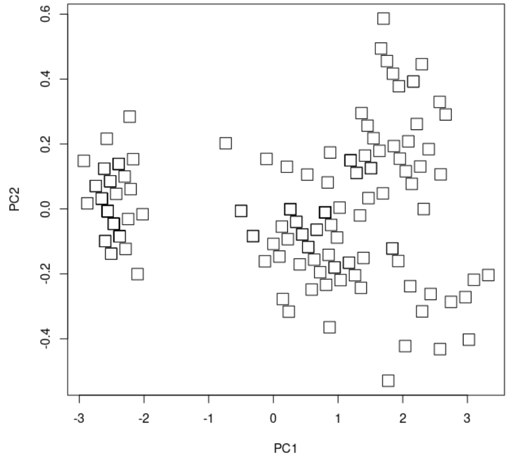
PCA plots

```
# plot original data
plot(X, pch=0, col="black", cex=2)
# PCA plot: PC1 vs PC2
plot(Xpca$x, pch=0, col="black", cex=2)
# plot PC1
plot(Xpca$x[, "PC1"], rep(0, times=length(Xpca$x[,
"PC1"])), pch=0, col="black", cex=2, xlab="PC1",
ylab="", ylim=c(-1,1))
# plot PC2
plot(rep(0, times=length(Xpca$x[, "PC2"])), Xpca$x[,
"PC2"], pch=0, col="black", cex=2, ylab="PC2",
xlab="", xlim=c(-1,1))
```

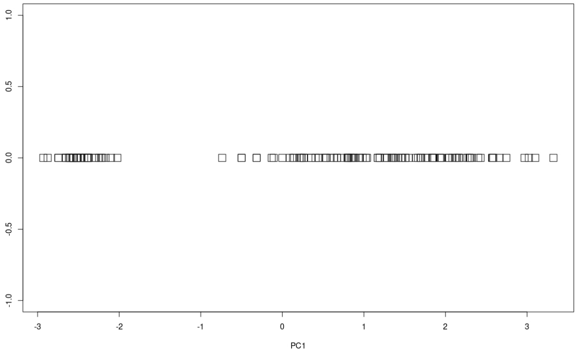
Original data



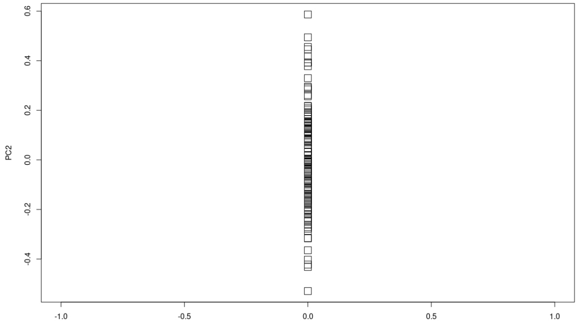
PCA plot



PC1

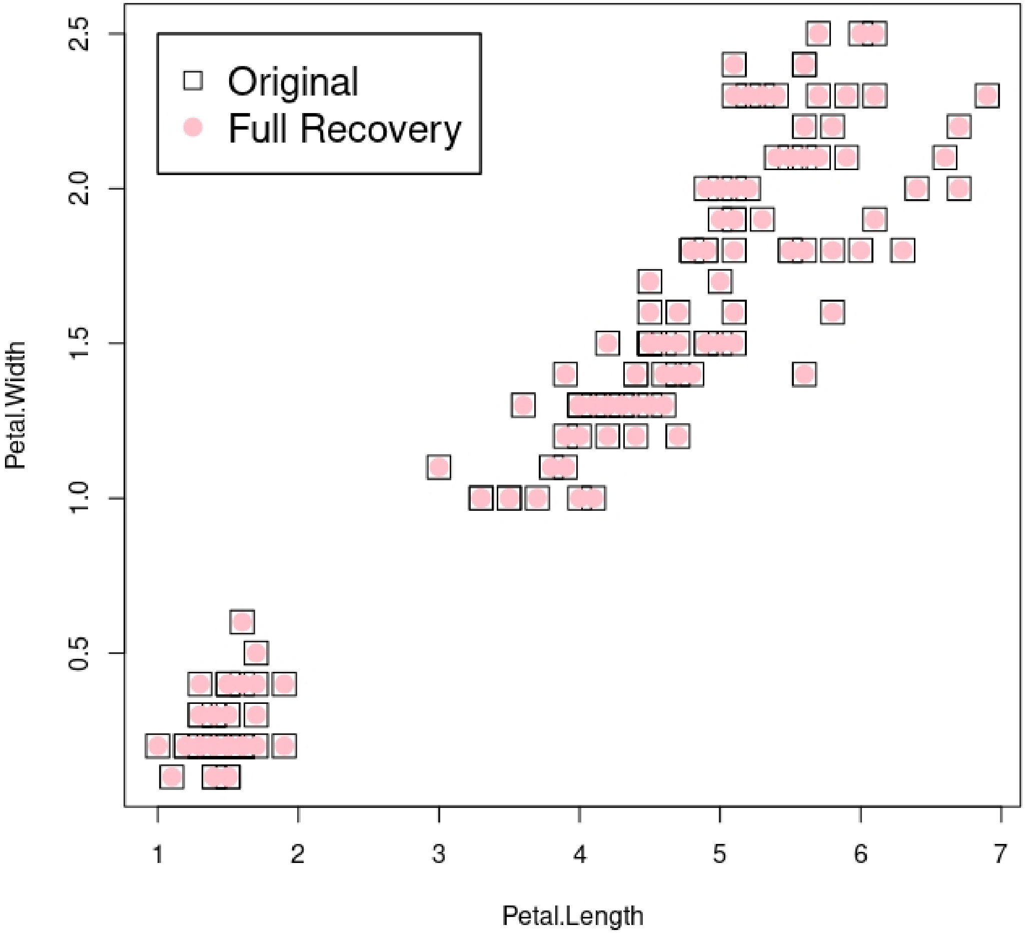


PC2



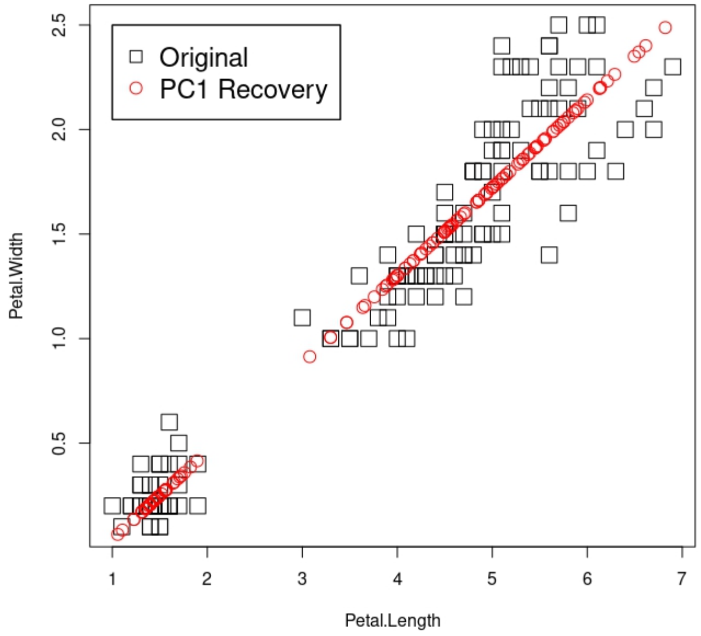
Full recovery from all PCs (PC1+PC2)

```
# Full recovery: Get two PCs
nComp <- 2
Xhat <- Xpca$x[,1:nComp] %*% t(Xpca$rotation[,1:nComp])
Xhat <- scale(Xhat, center = -mu, scale = FALSE)
plot(X, pch=0, col="black", cex=2)
points(Xhat, pch=19, col="pink", cex=1.5)
legend(1, 2.5, legend=c("Original", "Full Recovery"),
       col=c("black", "pink"), pch=c(0, 19), cex=c(1.5, 1.5))
```

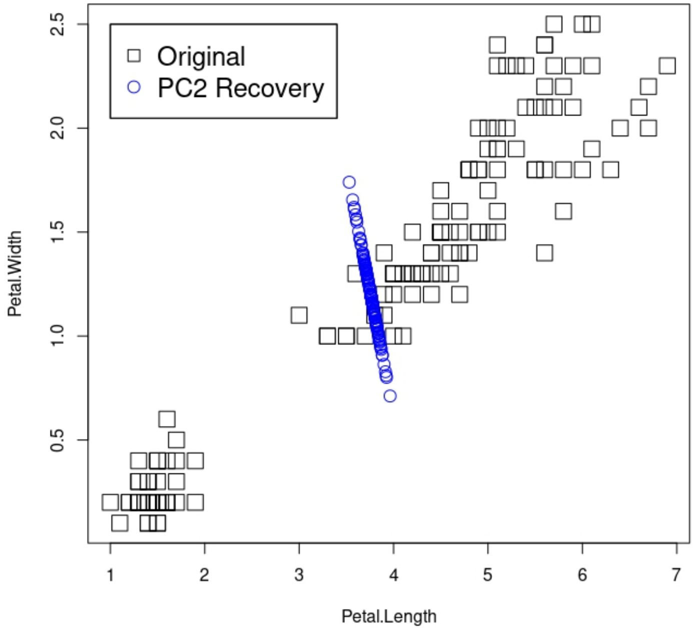


Partial recovery after remove PC1 or PC2

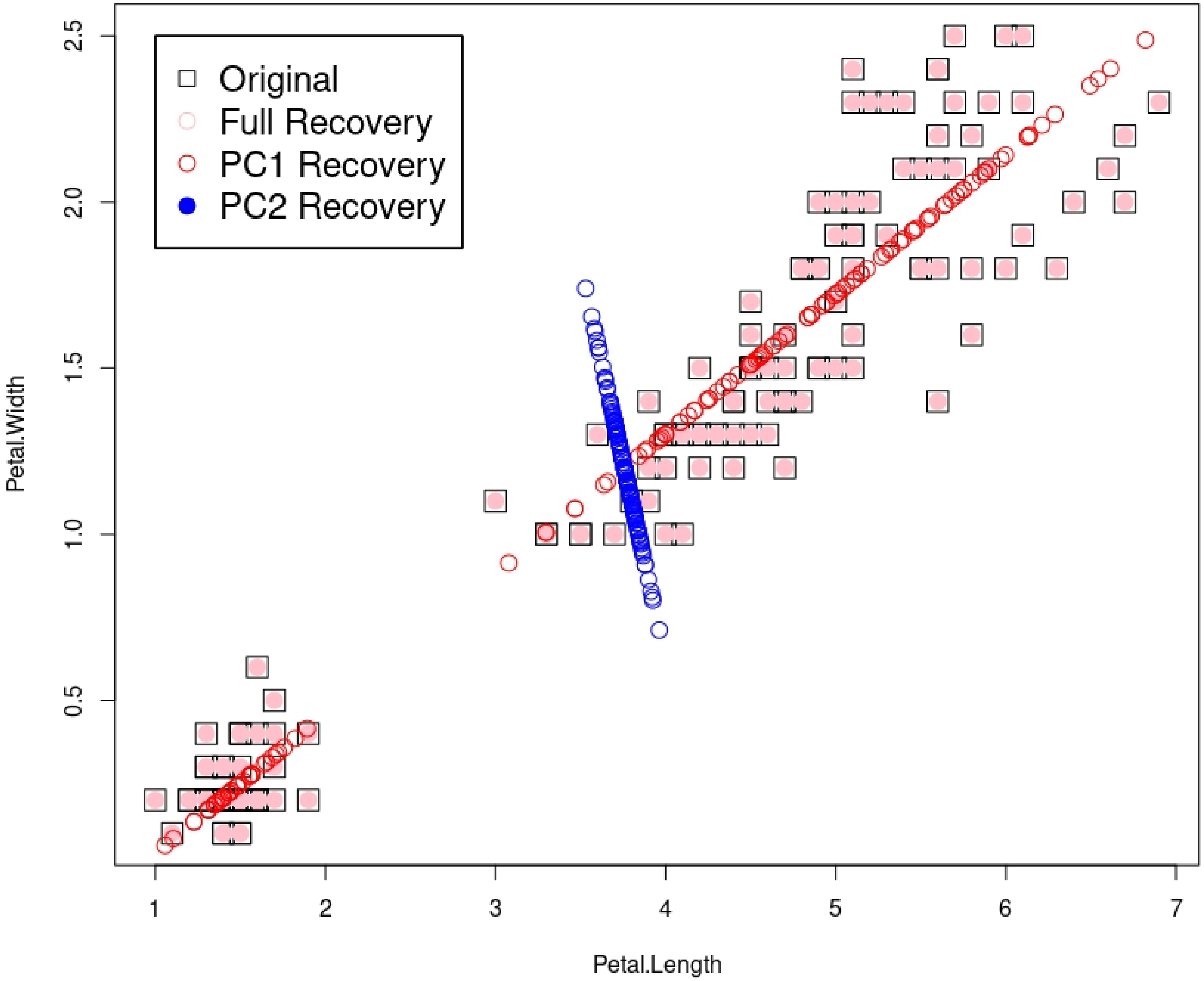
```
# Partial recovery after remove PC2: Get only first PC
nComp <- 1
Xhat <- Xpca$x[,nComp] %*% t(Xpca$rotation[,nComp])
Xhat <- scale(Xhat, center = -mu, scale = FALSE)
plot(X, pch=0, col="black", cex=2)
points(Xhat, pch=1, col="red", cex=1.5)
legend(1, 2.5, legend=c("Original", "PC1 Recovery"),
       col=c("black", "red"), pch=c(0, 1), cex=c(1.5, 1.5))
```



```
# Partial recovery after remove PC1: Get only second PC
nComp <- 2
Xhat <- Xpca$x[,nComp] %*% t(Xpca$rotation[,nComp])
Xhat <- scale(Xhat, center = -mu, scale = FALSE)
plot(X, pch=0, col="black", cex=2)
points(Xhat, pch=1, col="blue", cex=1.5)
legend(1, 2.5, legend=c("Original", "PC2 Recovery"),
       col=c("black", "blue"), pch=c(0, 1), cex=c(1.5, 1.5))
```



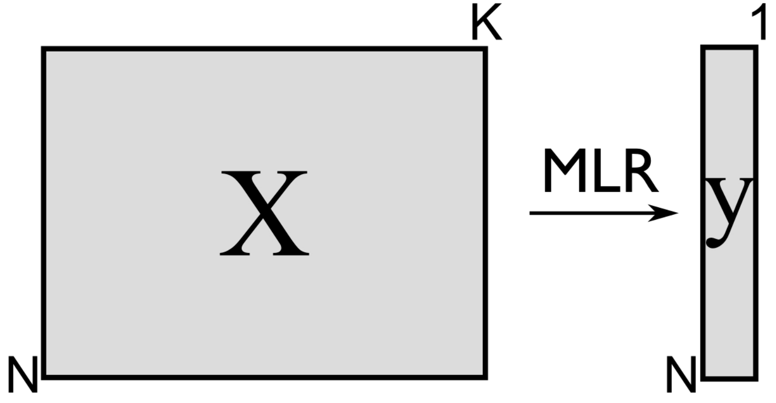
Recovery after remove PC1 or PC2



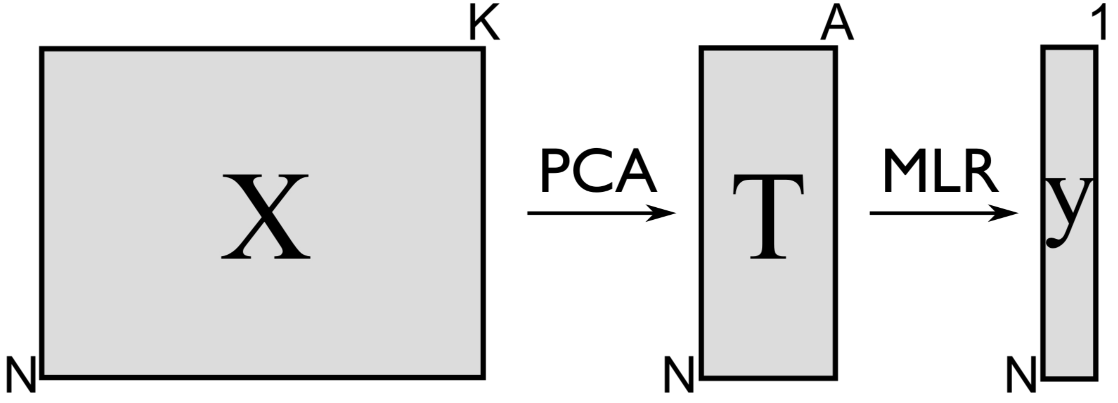
```
### Put all the plots into one figure
plot(X, pch=0, col="black", cex=2)
nComp <- 2
Xhat <- Xpca$x[,1:nComp] %*% t(Xpca$rotation[,1:nComp])
Xhat <- scale(Xhat, center = -mu, scale = FALSE)
points(Xhat, pch=19, col="pink", cex=1.5)
nComp <- 1
Xhat <- Xpca$x[,nComp] %*% t(Xpca$rotation[,nComp])
Xhat <- scale(Xhat, center = -mu, scale = FALSE)
points(Xhat, pch=1, col="red", cex=1.5)
nComp <- 2
Xhat <- Xpca$x[,nComp] %*% t(Xpca$rotation[,nComp])
Xhat <- scale(Xhat, center = -mu, scale = FALSE)
points(Xhat, pch=1, col="blue", cex=1.5)
# Add a legend
legend(1, 2.5, legend=c("Original", "Full Recovery", "PC1 Recovery",
"PC2 Recovery"), col=c("black", "pink", "red", "blue"), pch=c(0, 19, 1,
1), cex=c(1.5, 1.5, 1.5, 1.5))
```

Principal Component Regression (PCR)

Multiple linear regression



Principal component regression



https://learnche.org/pid/latent-variable-modelling/principal-components-regression

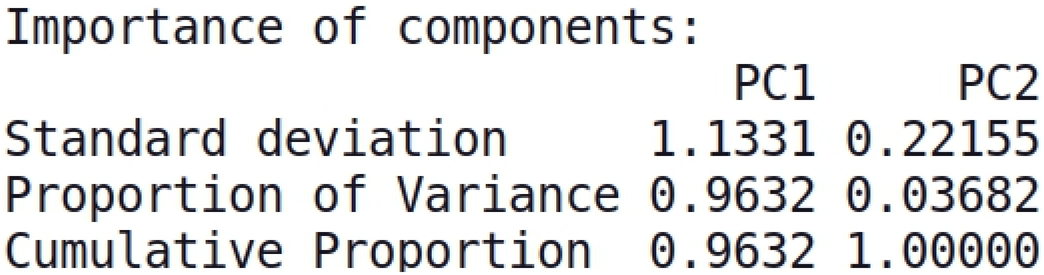
Exercise

1. Is Species a confounder in iris data with threshold P value < 0.05 and R = 0.5?
2. If yes, please remove it from data.

Homework

- How to calculate sdev (standard deviation) in output of prcomp in R?

- Xpca <- prcomp(X)
- Xpca$sdev
- [1] 1.1331495 0.2215477
- summary(Xpca)



```
Importance of components:
                          PC1     PC2
Standard deviation     1.1331 0.22155
Proportion of Variance 0.9632 0.03682
Cumulative Proportion  0.9632 1.00000
```