MthStat 768

February 21, 2024

Chapter 7: Principal Component Analysis

```r
iris <- read.csv(file = "../Data_csv/Iris.csv")  # two points for knitting, one point for running it in
```

```r
unique(iris$class)
```

```
## [1] "Iris-setosa"     "Iris-versicolor" "Iris-virginica"
```

```r
iris$class <- factor(iris$class)
levels(iris$class)
```

```
## [1] "Iris-setosa"     "Iris-versicolor" "Iris-virginica"
```

```r
levels(iris$class) <- c("Setosa", "Versicolor", "Virginica")
levels(iris$class)
```

```
## [1] "Setosa"     "Versicolor" "Virginica"
```

```r
tapply(X = iris$petal_width, INDEX = iris$class, FUN = mean)
```

```
##     Setosa Versicolor  Virginica
##      0.244      1.326      2.026
```

```r
apply(X = iris[, 1:4], MARGIN = 2, FUN = sd)
```

```
## sepal_length  sepal_width petal_length  petal_width
##    0.8280661    0.4335943    1.7644204    0.7631607
```

```r
iris[, 1:4] <- scale(x = iris[, 1:4])
apply(X = iris[, 1:4], MARGIN = 2, FUN = sd)
```

```
## sepal_length  sepal_width petal_length  petal_width
##            1            1            1            1
```

```r
pca <- princomp(~sepal_length + sepal_width + petal_length +
    petal_width, data = iris)
pca <- princomp(~. - class, data = iris)
summary(pca)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3      Comp.4
## Standard deviation     1.7004154 0.9565979 0.38258453 0.143074535
## Proportion of Variance 0.7277045 0.2303052 0.03683832 0.005151927
## Cumulative Proportion  0.7277045 0.9580098 0.99484807 1.000000000
```

The output of `princomp(..)` is a list with elements. - `sdev`: standard deviation of the components, $\sqrt{\lambda_k}$. - `loadings`: eigenvectors of v. - `scores`: component scores ξ.

```
lmb <- pca$sdev^2
lmb
```

```
##     Comp.1     Comp.2     Comp.3     Comp.4
## 2.89141263 0.91507946 0.14637092 0.02047032
```

```
cumsum(lmb)/sum(lmb)
```

```
##    Comp.1    Comp.2    Comp.3    Comp.4
## 0.7277045 0.9580098 0.9948481 1.0000000
```

Then q=2 gives a good approximation to the data. $\xi_i \in \mathbb{R}^2$

```
xi <- pca$scores[, 1:2]
xi
```

```
##            Comp.1       Comp.2
## 1   -2.256980633  0.504015404
## 2   -2.079459119 -0.653216394
## 3   -2.360044082 -0.317413945
## 4   -2.296503660 -0.573446613
## 5   -2.380801586  0.672514411
## 6   -2.063623476  1.513478267
## 7   -2.437545336  0.074313717
## 8   -2.226383267  0.246787172
## 9   -2.334138096 -1.091489770
## 10  -2.181367969 -0.447131117
## 11  -2.156262875  1.067020956
## 12  -2.319606855  0.158057946
## 13  -2.216656716 -0.706750478
## 14  -2.630902492 -0.935149145
## 15  -2.184971650  1.883668049
## 16  -2.243947781  2.713281331
## 17  -2.195395700  1.508696010
## 18  -2.182866358  0.512587094
## 19  -1.887750154  1.426332361
## 20  -2.332136197  1.154166863
## 21  -1.908163868  0.429027880
## 22  -2.197284291  0.949277150
## 23  -2.764907097  0.487882574
## 24  -1.814333378  0.106394362
## 25  -2.220777687  0.161644638
## 26  -1.950489685 -0.605862870
```

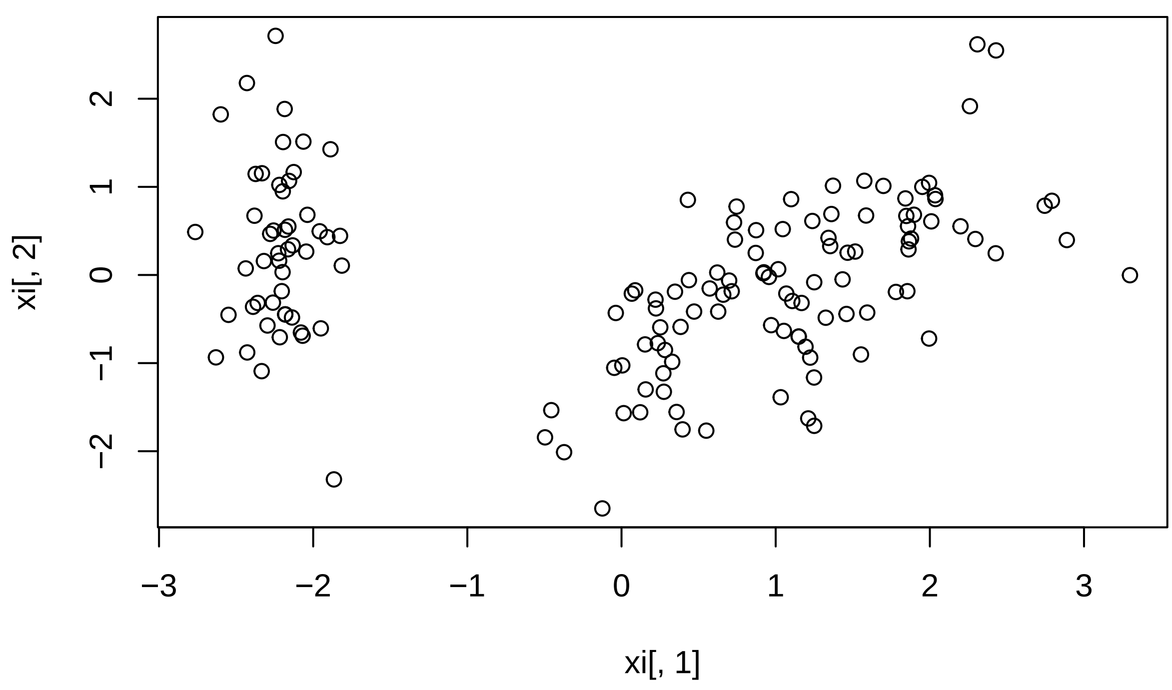
```
## 27  -2.045211662  0.265126115
## 28  -2.160954255  0.550173363
## 29  -2.133159680  0.335516398
## 30  -2.261214914 -0.313827252
## 31  -2.137393960 -0.482326259
## 32  -1.825821430  0.443780131
## 33  -2.599494320  1.822370083
## 34  -2.429810767  2.178094795
## 35  -2.181367969 -0.447131117
## 36  -2.203737172 -0.183722324
## 37  -2.037590402  0.682669420
## 38  -2.181367969 -0.447131117
## 39  -2.427818784 -0.879223933
## 40  -2.163299946  0.291749567
## 41  -2.278892736  0.466429135
## 42  -1.865457766 -2.319919659
## 43  -2.549294047 -0.452301130
## 44  -1.957720744  0.495730895
## 45  -2.126249698  1.167520808
## 46  -2.068428166 -0.689607099
## 47  -2.373307416  1.146790737
## 48  -2.390184347 -0.361180775
## 49  -2.219346197  1.022058561
## 50  -2.198588692  0.032130206
## 51   1.100307520  0.860230593
## 52   0.730035752  0.596636785
## 53   1.237962217  0.612769614
## 54   0.395980711 -1.752298584
## 55   1.069012656 -0.211050863
## 56   0.383174476 -0.589088966
## 57   0.746215186  0.776098609
## 58  -0.496201068 -1.842695569
## 59   0.923129797  0.030229555
## 60   0.004951438 -1.025964037
## 61  -0.124281108 -2.649187653
## 62   0.437265239 -0.058684686
## 63   0.549792127 -1.766663079
## 64   0.714770518 -0.184815166
## 65  -0.037133981 -0.431350036
## 66   0.872966018  0.508295314
## 67   0.346844441 -0.189985179
## 68   0.152880381 -0.788085297
## 69   1.211245424 -1.627902021
## 70   0.156417164 -1.298752329
## 71   0.735791136  0.401126570
## 72   0.470792484 -0.415217206
## 73   1.223888075 -0.937773165
## 74   0.627279600 -0.415419947
## 75   0.698133985 -0.063281927
## 76   0.870620328  0.249871518
## 77   1.250034459 -0.082344239
## 78   1.353704810  0.327722366
## 79   0.659915360 -0.223597000
## 80  -0.047123645 -1.053682478
```

```
## 81   0.121128417 -1.558371690
## 82   0.014071087 -1.568138943
## 83   0.235222819 -0.773333046
## 84   1.053163233 -0.634774729
## 85   0.220677797 -0.279909969
## 86   0.430341477  0.852281697
## 87   1.045909461  0.520453696
## 88   1.032419509 -1.387817168
## 89   0.066843667 -0.211910814
## 90   0.274505447 -1.325375781
## 91   0.271425765 -1.115703812
## 92   0.621089831  0.027450671
## 93   0.328903506 -0.985598884
## 94  -0.372380115 -2.011194576
## 95   0.281999618 -0.851099455
## 96   0.088755770 -0.174324544
## 97   0.223607677 -0.379214256
## 98   0.571967342 -0.153206717
## 99  -0.455486949 -1.534324381
## 100  0.251402252 -0.593871222
## 101  1.841503386  0.868786147
## 102  1.149339414 -0.698984451
## 103  2.198982700  0.552618781
## 104  1.433881765 -0.049843542
## 105  1.861653988  0.290220536
## 106  2.745000701  0.785799704
## 107  0.357177896 -1.554885572
## 108  2.295316375  0.408149357
## 109  1.995051690 -0.721448440
## 110  2.259983444  1.915027471
## 111  1.361348784  0.691631011
## 112  1.593725457 -0.426818953
## 113  1.877960511  0.412949339
## 114  1.248902574 -1.163493524
## 115  1.459173157 -0.442664602
## 116  1.586494399  0.674774813
## 117  1.466367721  0.252347086
## 118  2.429240301  2.548220565
## 119  3.298092266 -0.002353436
## 120  1.249794060 -1.711848991
## 121  2.033683231  0.904369044
## 122  0.970663302 -0.569267278
## 123  2.888388067  0.396463171
## 124  1.324755637 -0.485135293
## 125  1.698550406  1.010762277
## 126  1.951190990  0.999984474
## 127  1.167991627 -0.317831851
## 128  1.016376098  0.065324121
## 129  1.780045543 -0.192627480
## 130  1.858551592  0.553527164
## 131  2.427365491  0.245830912
## 132  2.308349227  2.617415284
## 133  1.854159818 -0.184055790
## 134  1.107561292 -0.294997832
```

```
## 135  1.193470916 -0.814439294
## 136  2.791597293  0.841927658
## 137  1.574879256  1.068893603
## 138  1.342546768  0.420846092
## 139  0.920349720  0.019166162
## 140  1.847363145  0.670177572
## 141  2.009425438  0.608358978
## 142  1.896762527  0.683734258
## 143  1.149339414 -0.698984451
## 144  2.036486021  0.861797778
## 145  1.995007506  1.045049035
## 146  1.864276571  0.381543631
## 147  1.553288230 -0.902290843
## 148  1.515767103  0.265903772
## 149  1.371795548  1.012968390
## 150  0.956095566 -0.022209541
```

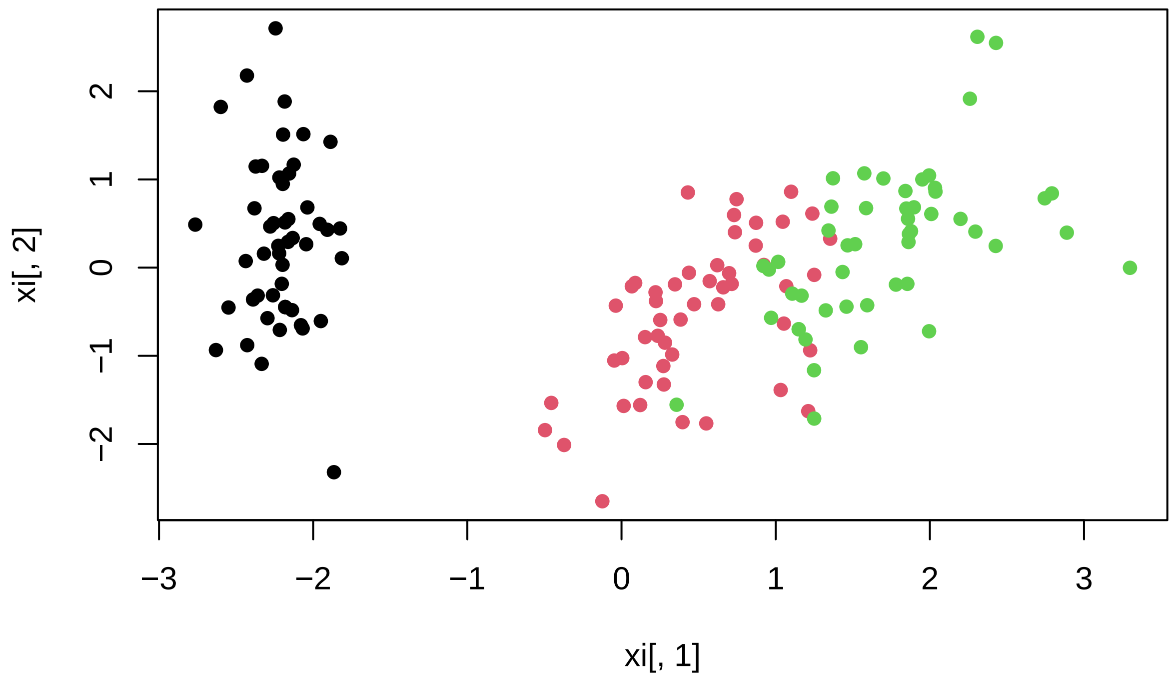
```r
colnames(xi) <- c("PC1", "PC2")
```

```r
plot(xi[, 1], xi[, 2])
```

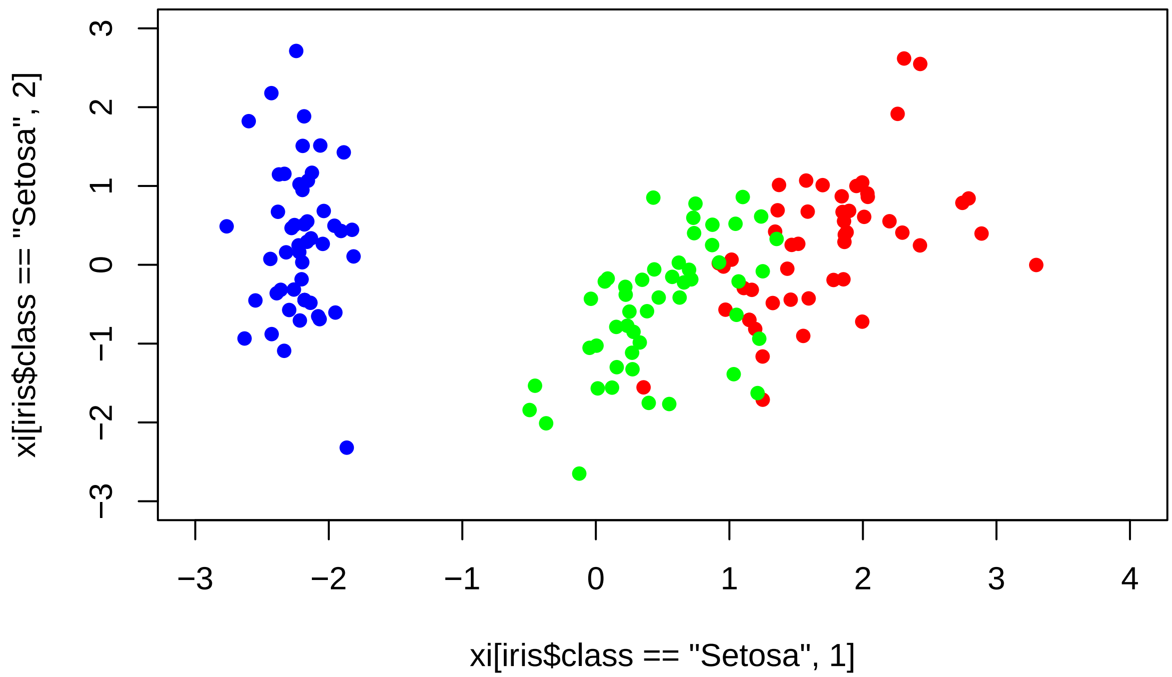


To see if the species of Iris are spatially separated, we can plot ξ_{i1} vs ξ_{i2}.

```r
plot(xi[, 1], xi[, 2], col = iris$class, pch = 16)
```




```r
plot(xi[iris$class == "Setosa", 1], xi[iris$class == "Setosa",
    2], col = "blue", pch = 16, xlim = c(-3, 4), ylim = c(-3,
    3))
points(xi[iris$class == "Virginica", 1], xi[iris$class == "Virginica",
    2], col = "red", pch = 16)
points(xi[iris$class == "Versicolor", 1], xi[iris$class == "Versicolor",
    2], col = "green", pch = 16)
```



We see that Iris Setosa is well separated from the other species. Iris Virginica and Versicolor are also separated to some extent, but not as neatly as they are from Setosa.

```
pca$loadings[, 1]
```

```
## sepal_length  sepal_width petal_length  petal_width
##    0.5223716   -0.2633549    0.5812540    0.5656110
```

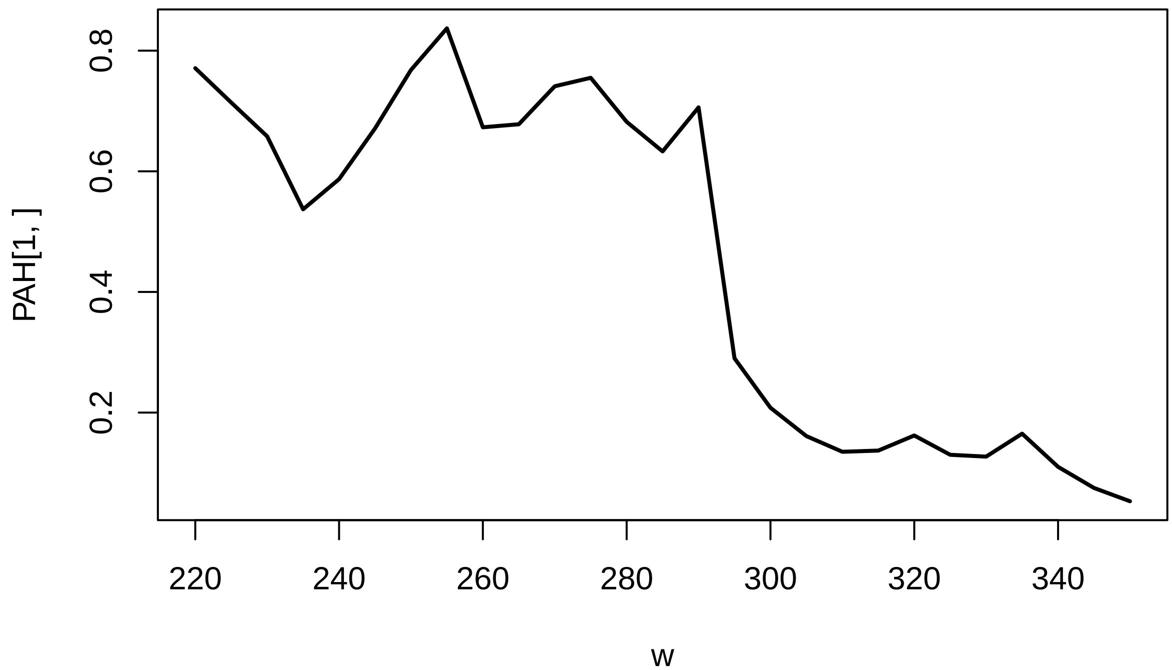
The eigenvectors are the loadings. ξ_{i1} is a contrast between the average of (sepal length, petal length, petal width) and sepal width.

```
PAH <- read.csv(file = "../Data_csv/PAH.csv")
```

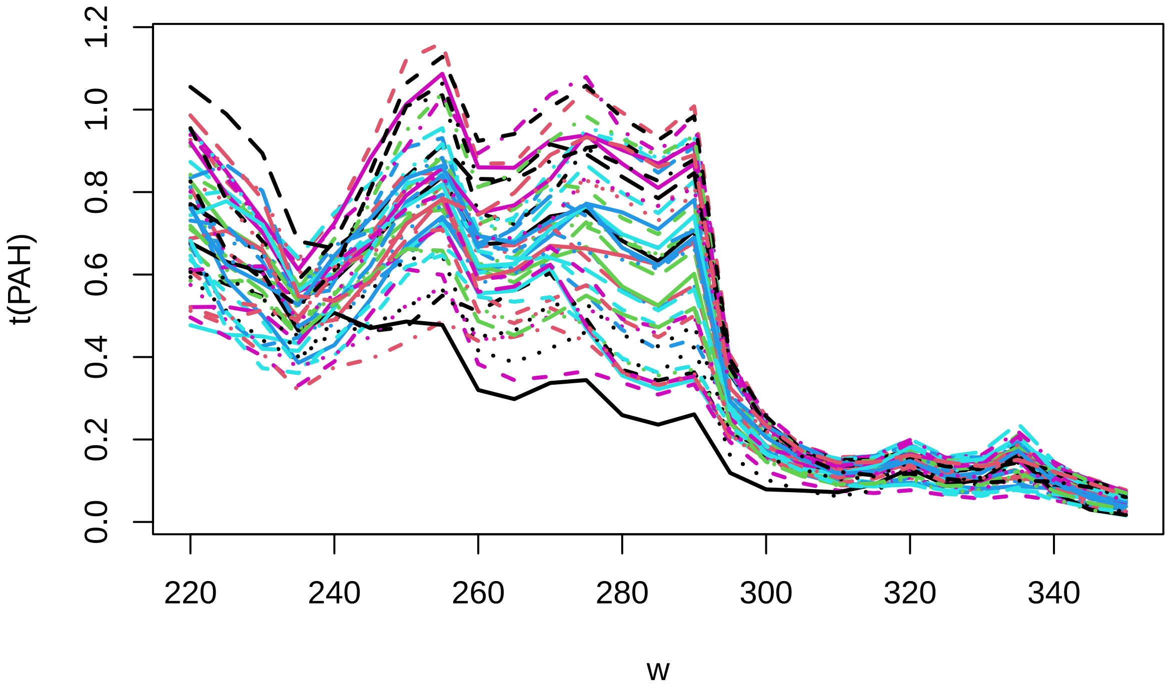
```
PAH <- PAH[, -(1:10)]
```

```
w <- seq(220, 350, by = 5)
```

```
plot(w, PAH[1, ], type = "l", lwd = 2)
```




```
matplot(w, t(PAH), type = "l", lwd = 2)
```


0.0
0.2
0.4
0.6
0.8
1.0
1.2
t(PAH)
220
240
260
280
300
320
340
w