Principal Components Analysis in R

Ken B. Hanscombe

Learning outcomes

- **When to use PCA**
- **Run a PCA in R**
- **Query the output**
- **Plot the results**

PCA

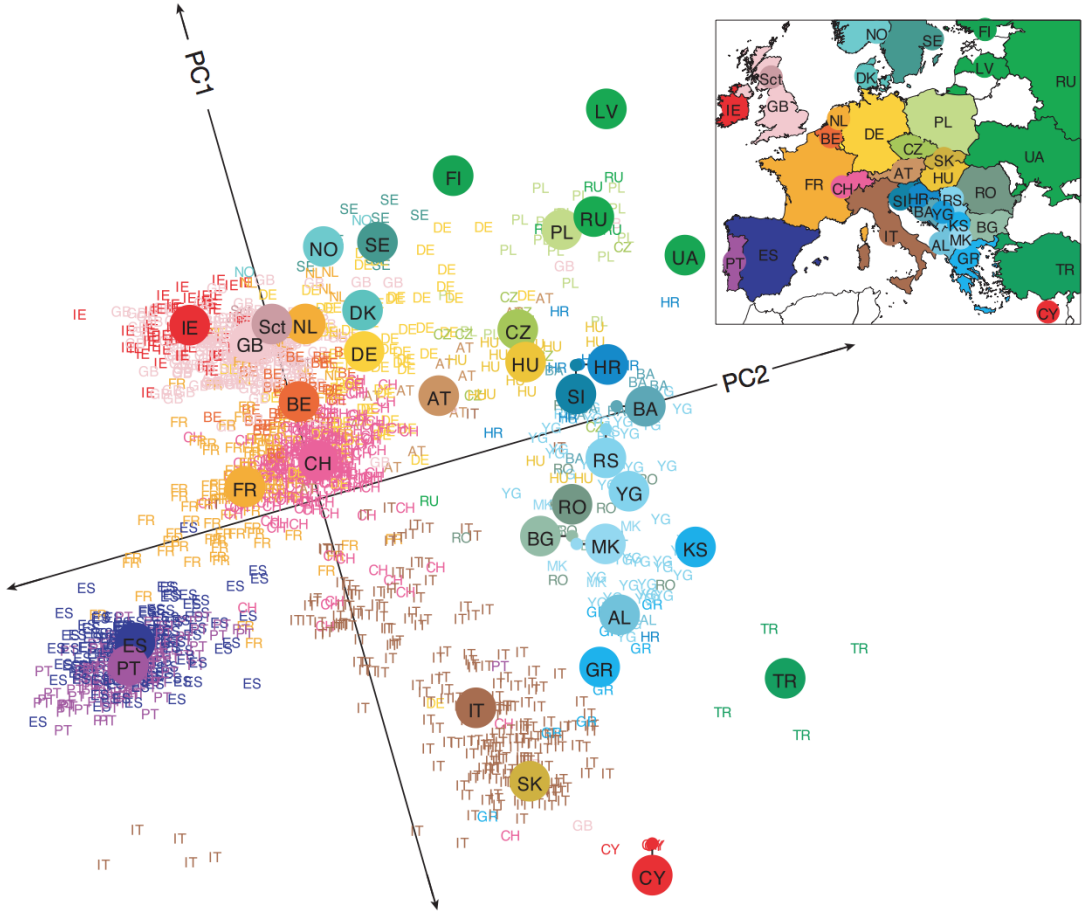
What/ When/ Why do PCA?

- **dissecting and ranking the correlational structure of multivariate data**
  - reveal hidden structure
  - reduce dimensionality / find the best low-dimensional representation
  - filter noise / decrease redundancy
  - preparation for further analysis
- **projection method for finding projections of maximum variability**
  - seeks linear combinations of the columns of X with maximal (or minimal) variance

The results of a PCA are usually discussed in terms of

- **loadings**
  - weight by which each standardized original variable should be multiplied to get the component score
- **component scores**
  - transformed variable values corresponding to a particular data point

An application: population ancestry



Genes mirror geography
Variation at **100,000s genetic markers** reduce to two dimensions: first **2 PCs**

Novembre et al. (2008) Nature

prcomp()

- **Example 1:** Violent crime rates by US state

```
R> USArrests
R> cor(USArrests, use="pairwise.complete.obs")
R> plot(USArrests)
R> ??"principal components analysis"

R> pc1 <- prcomp(USArrests, scale = TRUE)
R> summary(pc1)
R> plot(pc1)
R> biplot(pc1)

R> names(pc1)
R> boxplot(pc1$x,col="grey",frame=F)
```

R> example(prcomp)

prcomp() vs princomp()

- **prcomp()** singular value decomposition of **data matrix**
- **princomp()** eigenanalysis of **covariance** or **correlation matrix**

	prcomp()	**princomp()**
square root of the eigenvalues	$sdev	$sdev
eigenvectors	$rotation	$loadings
means of each variable	$center	$center
scaling used or FALSE	$scale	$scale
principal components	$x	$scores
number of observations of each variable		$n.obs
the call that created the object		$call

* prcomp() numerically stable, preferred

- differences in function **parameters**, **values** return, **technique** used
- summary() of returned object gives variation explained by each component

prcomp()

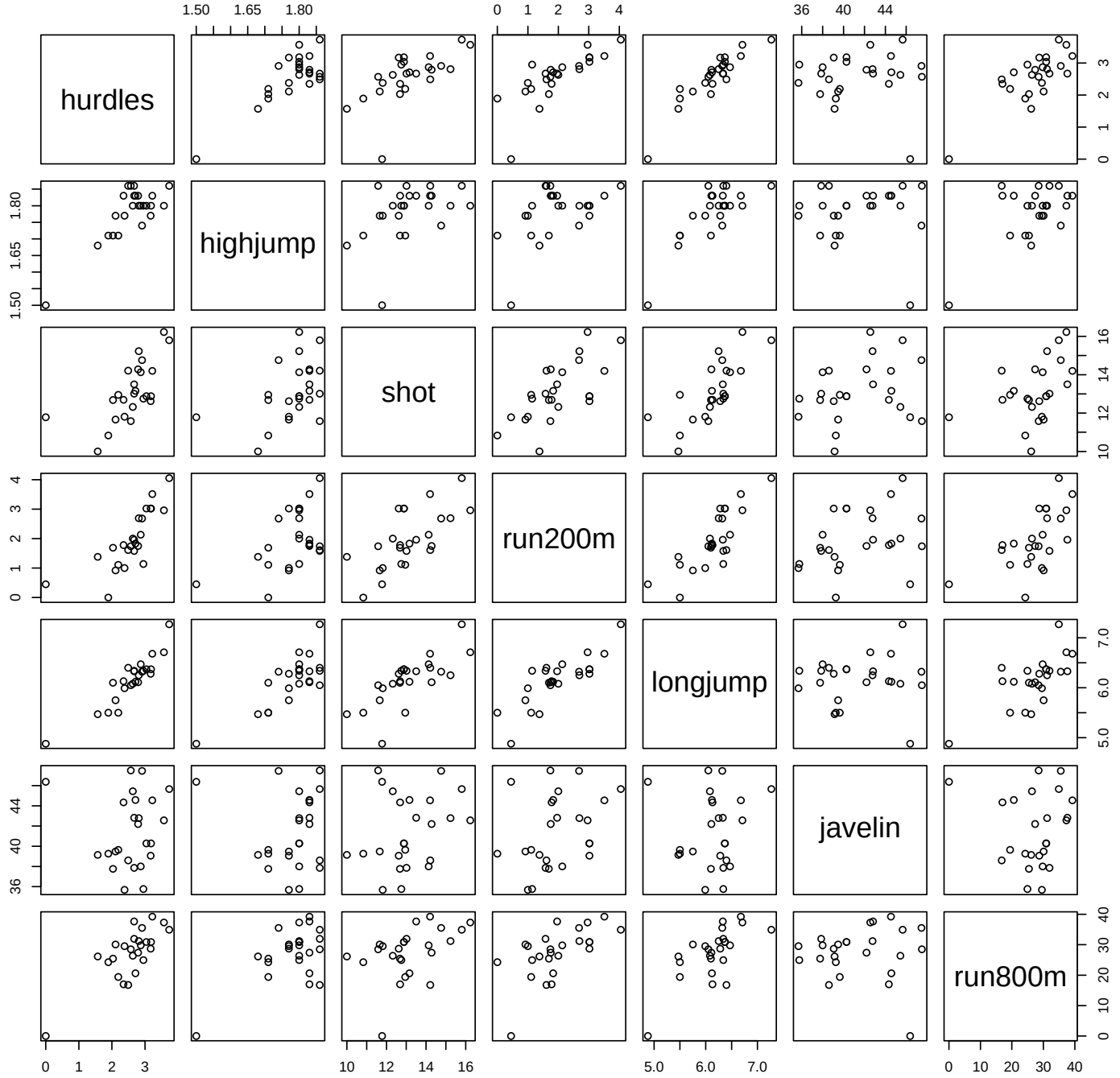
- **Example 2:** The Olympic Heptathlon

```
R> data(heptathlon,package="HSAUR")

# Recode variables
R> heptathlon$hurdles <- max(heptathlon$hurdles)-heptathlon$hurdles
R> heptathlon$run200m <- max(heptathlon$run200m)-heptathlon$run200m
R> heptathlon$run800m <- max(heptathlon$run800m)-heptathlon$run800m

R> pairs(heptathlon[,-which(colnames(heptathlon)=="score")])
```

Everitt & Hothorn (2006) A handbook of statistical analyses using R


hurdles
highjump
shot
run200m
longjump
javelin
run800m

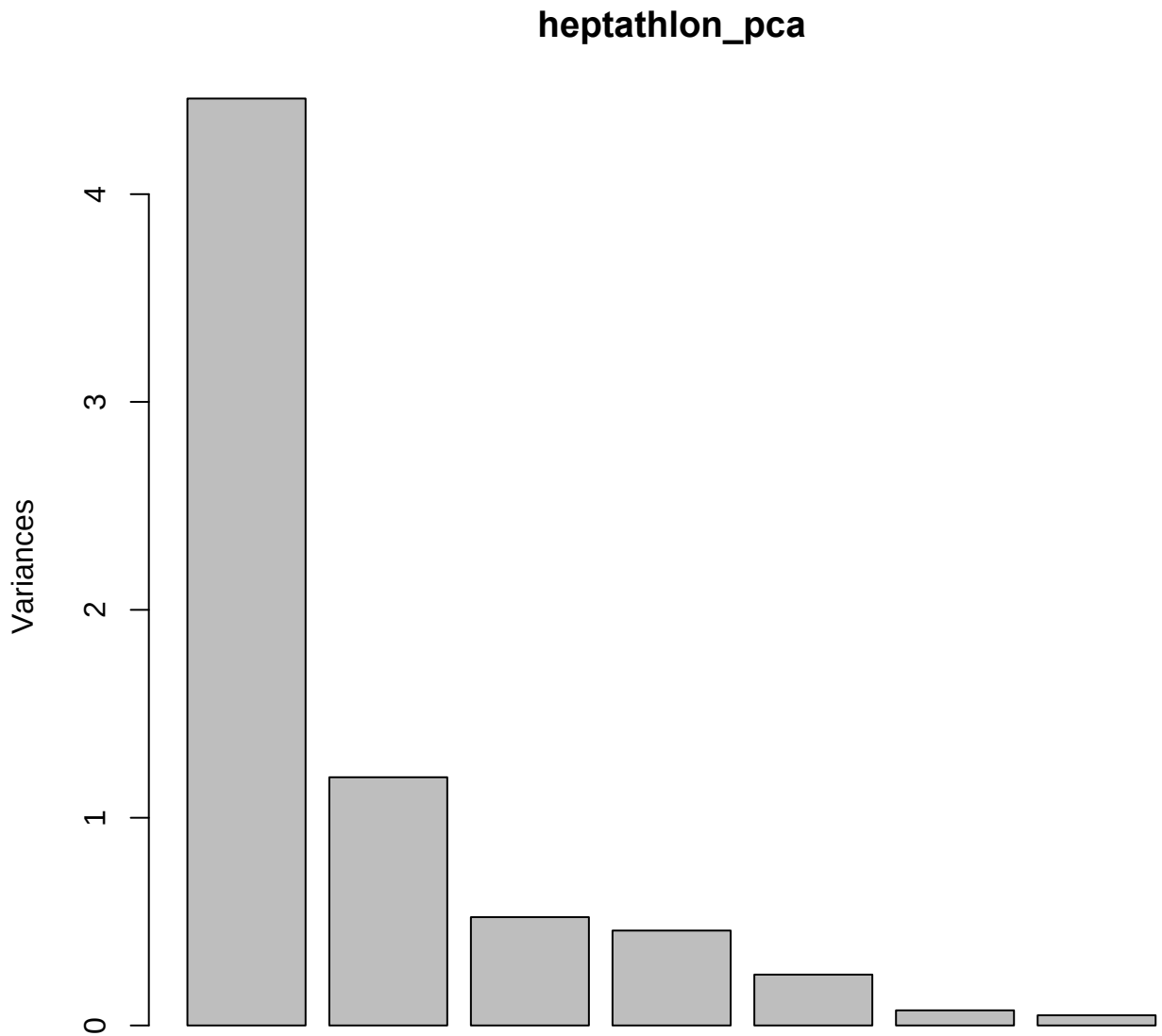
prcomp()

- **Example 2:** cont…

```
R> score <- which(colnames(heptathlon)=="score")
R> heptathlon_pca <- prcomp(heptathlon[,-score],scale=T)
R> summary(heptathlon_pca)

Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6    PC7
Standard deviation     2.1119 1.0928 0.72181 0.67614 0.49524 0.27010 0.2214
Proportion of Variance 0.6372 0.1706 0.07443 0.06531 0.03504 0.01042 0.0070
Cumulative Proportion  0.6372 0.8078 0.88223 0.94754 0.98258 0.99300 1.0000


R> plot(heptathlon_pca)
```


heptathlon_pca
Variances
0
1
2
3
4

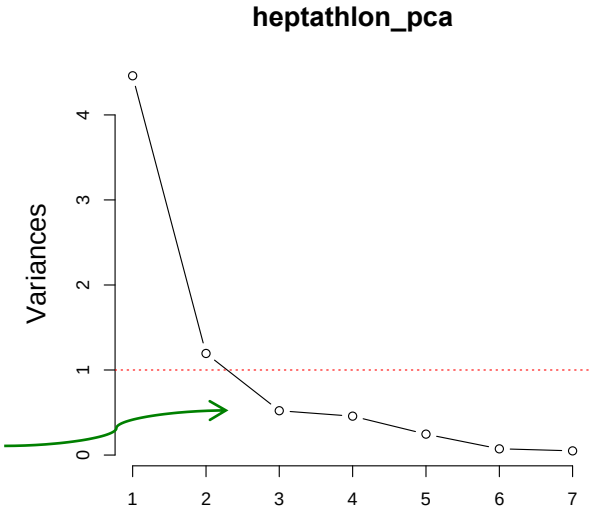
How many components to keep?

- **Kaiser criterion** Kaiser (1960)
  - retain only factors with eigenvalues > 1 (Note: eigenvalues = variances)
  - unless a factor extracts as much as the equivalent of one original variable, drop it

- **Scree test** Cattel (1966)
  - find a place where the smooth decrease of eigenvalues levels off
  - to the right of this point we find only "pc scree"

```
R> plot(heptathlon_pca,type="line",cex.lab=1.5, cex.main=1.5)
R> abline(h=1,lty=3, col="red")
```

How many components to keep?



PC scree

Geological scree

prcomp()

- **Example 2:** cont…

```
R> heptathlon_pca

Standard deviations:
[1] 2.1119364 1.0928497 0.7218131 0.6761411 0.4952441 0.2701029 0.2213617

Rotation:
                PC1         PC2         PC3         PC4         PC5         PC6         PC7
hurdles  -0.4528710  0.15792058 -0.04514996  0.02653873 -0.09494792 -0.78334101  0.38024707
highjump -0.3771992  0.24807386 -0.36777902  0.67999172  0.01879888  0.09939981 -0.43393114
shot     -0.3630725 -0.28940743  0.67618919  0.12431725  0.51165201 -0.05085983 -0.21762491
run200m  -0.4078950 -0.26038545  0.08359211 -0.36106580 -0.64983404  0.02495639 -0.45338483
longjump -0.4562318  0.05587394  0.13931653  0.11129249 -0.18429810  0.59020972  0.61206388
javelin  -0.0754090 -0.84169212 -0.47156016  0.12079924  0.13510669 -0.02724076  0.17294667
run800m  -0.3749594  0.22448984 -0.39585671 -0.60341130  0.50432116  0.15555520 -0.09830963
```

prcomp()

- **Example 2:** cont…

```
# Linear combination for the first PC
R> heptathlon_pca$rotation[,1]

 hurdles   highjump       shot    run200m   longjump    javelin    run800m
-0.4528710 -0.3771992 -0.3630725 -0.4078950 -0.4562318 -0.0754090 -0.3749594


# Calculate the first PC
R> scale(
        as.matrix(heptathlon[,-score]),
        center = heptathlon_pca$center,
        scale = heptathlon_pca$scale
    ) %*% heptathlon_pca$rotation[, 1]
```

prcomp()

- **Example 2:** cont…

```
 [,1]
Joyner-Kersee (USA) -4.121447626
John (GDR)          -2.882185935
Behmer (GDR)        -2.649633766
Sablovskaite (URS)  -1.343351210
Choubenkova (URS)   -1.359025696
Schulz (GDR)        -1.043847471
Fleming (AUS)       -1.100385639
Greiner (USA)       -0.923173639
Lajbnerova (CZE)    -0.530250689
Bouraga (URS)       -0.759819024
Wijnsma (HOL)       -0.556268302
Dimitrova (BUL)     -1.186453832
Scheider (SWI)       0.015461226
Braun (FRG)          0.003774223
Ruotsalainen (FIN)   0.090747709
Yuping (CHN)        -0.137225440
Hagger (GB)          0.171128651
Brown (USA)          0.519252646
Mulliner (GB)        1.125481833
Hautenauve (BEL)     1.085697646
Kytola (FIN)         1.447055499
Geremias (BRA)       2.014029620
Hui-Ing (TAI)        2.880298635
Jeong-Mi (KOR)       2.970118607
Launa (PNG)          6.270021972
```

Compare to:

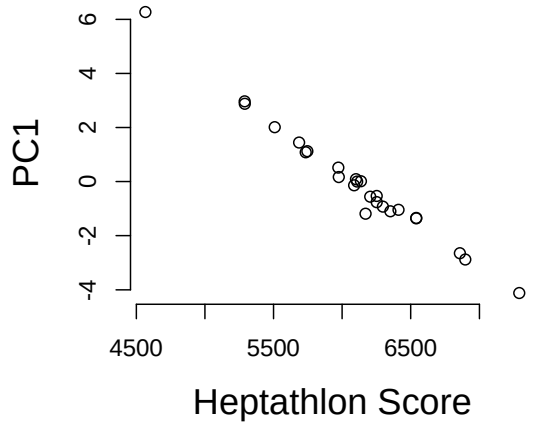
```
R> heptathlon$x
```

prcomp()

- **Example 2:** cont…

```
R> cor(heptathlon$score, heptathlon_pca$x[, 1])
[1] -0.9910978

R> plot(
    heptathlon$score,
    heptathlon_pca$x[, 1],
    xlab="Heptathlon Score",
    ylab="PC1",
    cex.lab=1.5,
    frame=F)
```



Other reading

- Principal Components Analysis: A How-to manual for R. Emily Mankin (Google)

Shows you the matrix algebra and how to do PCA without the built-in functions

- http://stackoverflow.com/questions/14249156/principal-component-analysis-pca-in-r-which-function-to-use

Discusses differences between prcomp() and princomp()

- http://tgmstat.wordpress.com/2013/11/21/introduction-to-principal-component-analysis-pca/
- http://tgmstat.wordpress.com/2013/11/28/computing-and-visualizing-pca-in-r/

Introduction to and visualizing PCA