R Lab - Day 3

Principal Component Analysis

MED3007 V24

2024.01.17

Chi Zhang
Oslo Center for Biostatistics and Epidemiology
chi.zhang@medisin.uio.no

Overview

Topics for this morning

Data exploration and visualization

Review theory from yesterday: Principal component analysis

One example in detail: food data

One example in less detail: NCI60

Exercises, group work, free reading time

Data exploration and visualization

Get to know your data

First thing to do when you have a new dataset, is to get to know it

Not only for genomics data!

Ask questions: e.g.

How many subjects (patients) - number of **rows**

How many different measurements (height, weight, smoker or not) - number of **columns**

Is my data complete? Do they satisfy **assumptions** to do tests? … …

Visualization (plot the data) can be quite useful.

Data exploration and visualization

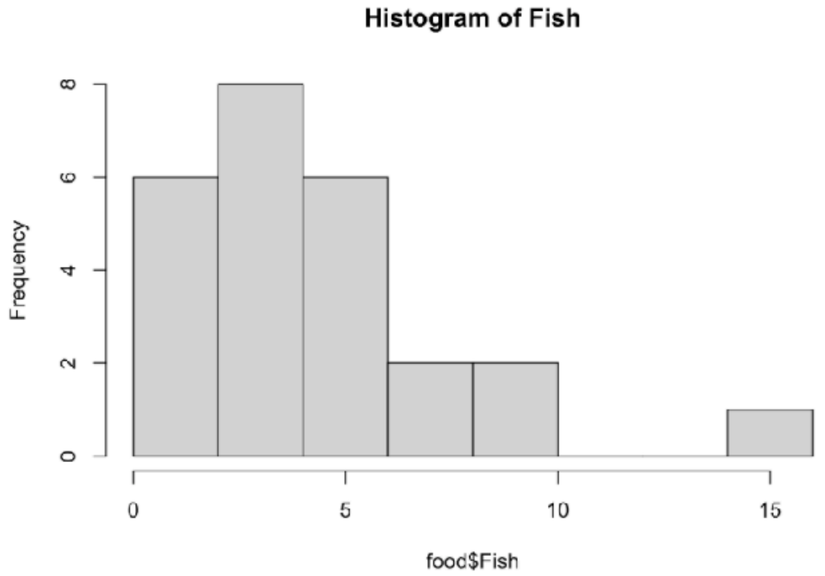
Low dimensional data

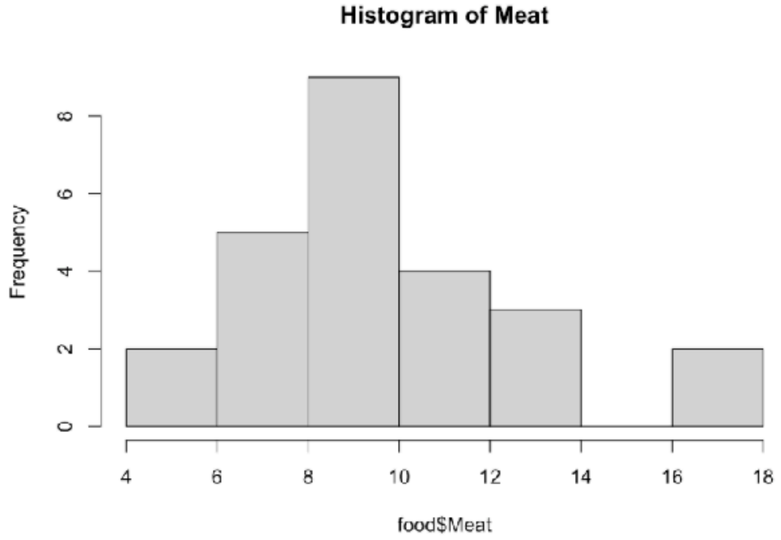
Assuming the data is **numeric** (numbers, not categories)

One variable (measured feature of a subject):

Histogram, **box plot** to visualize the data distribution

Can also compare **two variables** by making two box plots next to each other





Data exploration and visualization

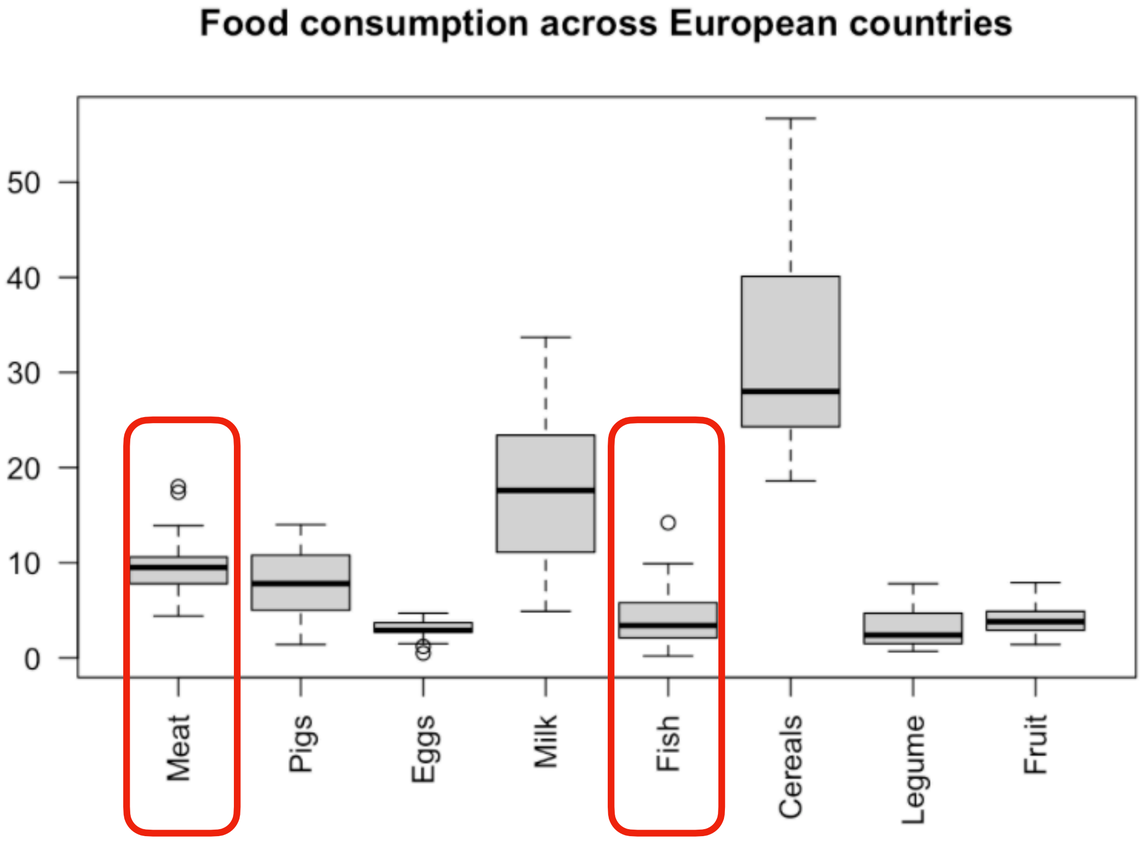
Low dimensional data

Assuming the data is **numeric** (numbers, not categories)

One variable (measured feature of a subject):

Histogram, **box plot** to visualize the data distribution

Can also compare **two variables** by making two box plots next to each other



Data exploration and visualization

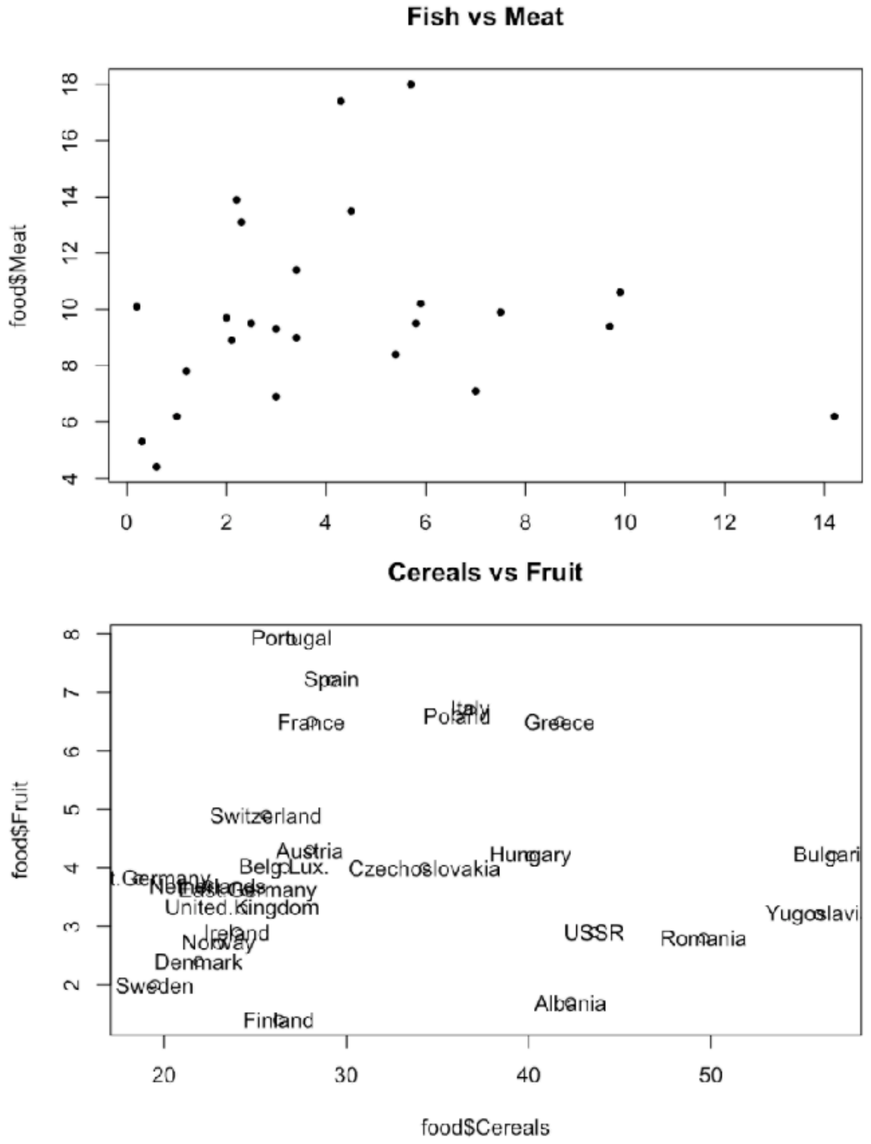
Low dimensional data

Histogram and boxplots show **aggregated** information of the data: frequency, min/max values, median

You can also look at each data point - can even add labels on each point

It is common to plot 2 numeric variables against each other - **scatter plot**

Then you can see if there is any relationship between the two



Data exploration and visualization

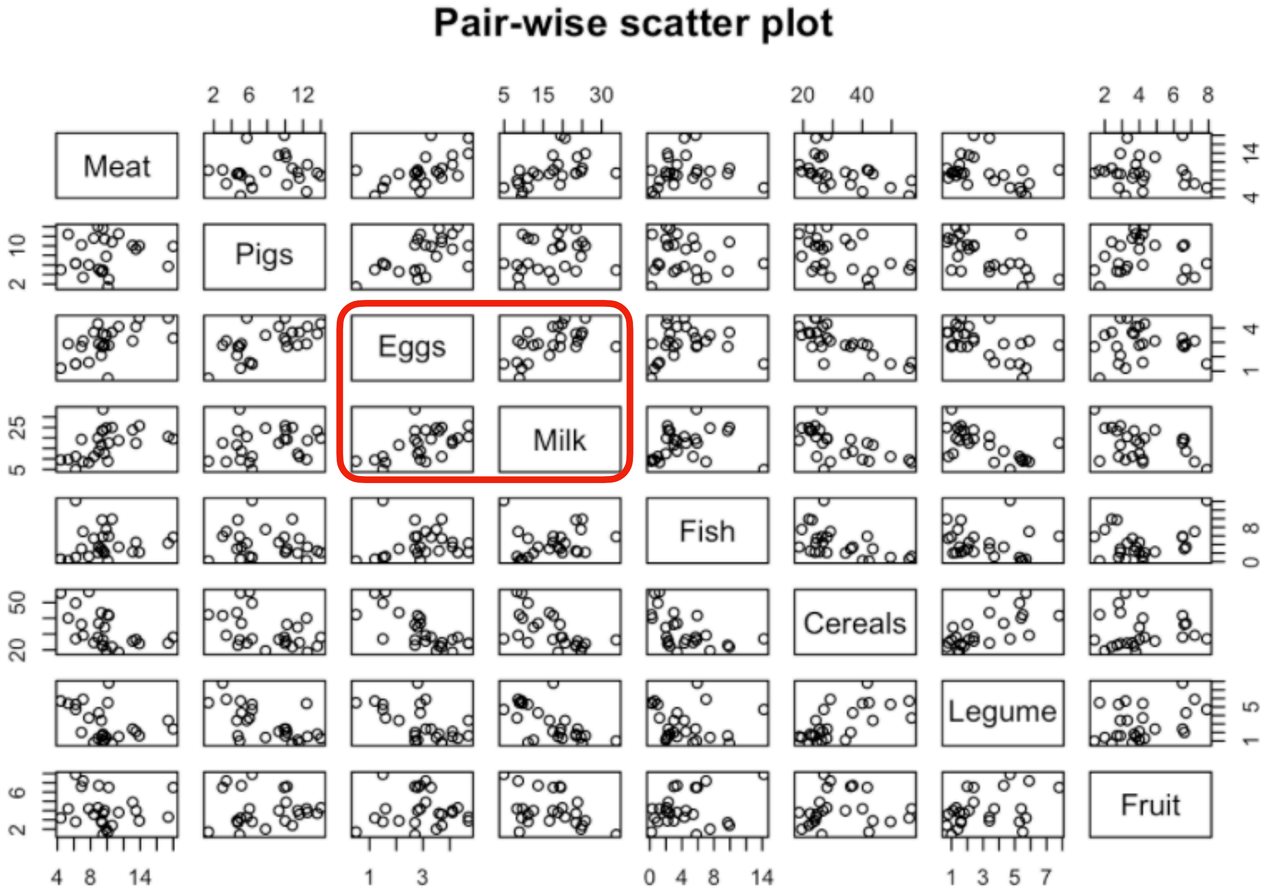
Low dimensional data

Scatter plot can be made for each **pair** of the (numeric) measurements in the same dataset

e.g. positive correlation between **egg and milk**?

However, it is hard to see the pattern when you have many variables

In this example of 8 variables, it is already a bit overwhelming



Data exploration and visualization

High dimensional data

Scatter plots are 2-dimensional plots: only 2 variables visualized at the same time

You can add a 3rd dimension, however it is the best you can do in a static figure

Genomics data has **thousands of variables** - inpractical with scatter plot to identify pair-wise patterns.

How do you explore and present data that is complex?

Dimensionality reduction: work on 2 or 3 *new variables*, rather than thousands of the *original variables*.

Principal component analysis

Dimensional reduction

Example:
v_new = v1 + v2 + 0.5 * v3

3 dimensional data become 1: dimensional reduction.

v1	v2	v3
1	2	3
4	5	6
7	8	9

v_new
1+2+ 0.5*3 = **4.5**
4+5+ 0.5*6 = **12**
7+8+ 0.5*9 = **19.5**

Principal component analysis PCA creates *new variables* based on the original ones in a similar manner

The **coefficients** (loading) to multiply the original data will be computed from the data

Principal components PC: v_new from PCA.

You can visualize the first and second PC in a scatter plot

Food example
PCA

We use a **low-dimensional dataset**, Food.txt

This is not a genomics dataset, but it’s useful to illustrate some key concepts.

We have seen some plots from the same dataset before

We try PCA to reduce 8 dimensional data to 3 dimensional, and see what information they tell us

(You try to replicate the analysis during the exercise session. Note and code provided)

Food example

PCA

Command for PCA:

`prcomp(data, scale=TRUE)`

Data needs to have all numeric entries (country names here are names of the rows, not data entries)

`scale=T` does an operation on the data columns so that each one has variance 1

By creating a variable (pc_food) to save the PCA outcome, you can examine the results, and even plot them

8 principal components computed, and the variance explained by each PC is reported

```
food <- read.table('data/Food.txt', header=T)
# we change the name from pulses to a more common name, legume
colnames(food)[7] <- 'Legume'
head(food) # print first 6 lines
```

```
             Meat Pigs Eggs Milk Fish Cereals Legume Fruit
Albania      10.1  1.4  0.5  8.9  0.2    42.3    5.5   1.7
Austria       8.9 14.0  4.3 19.9  2.1    28.0    1.3   4.3
Belg.Lux.    13.5  9.3  4.1 17.5  4.5    26.6    2.1   4.0
Bulgaria      7.8  6.0  1.6  8.3  1.2    56.7    3.7   4.2
Czechoslovakia 9.7 11.4  2.8 12.5  2.0    34.3    1.1   4.0
Denmark      10.6 10.8  3.7 25.0  9.9    21.9    0.7   2.4
```

```
# need to scale the data
pc_food <- prcomp(food, scale=TRUE)
# pc_food
summary(pc_food)
```

```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7
Standard deviation     1.9251 1.2073 1.0595 0.9315 0.57322 0.52889 0.35617
Proportion of Variance 0.4632 0.1822 0.1403 0.1085 0.04107 0.03497 0.01586
Cumulative Proportion  0.4632 0.6454 0.7857 0.8942 0.93527 0.97024 0.98609
                           PC8
Standard deviation     0.33354
Proportion of Variance 0.01391
Cumulative Proportion  1.00000
```

Food example

PCA

Loadings

Extracted by `pc_food$rotation`

```
loading_food <- pc_food$rotation
# print out the result, but only keep 2 digits
round(loading_food, digits = 2)
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Meat	-0.33	-0.05	-0.20	-0.72	-0.48	-0.20	-0.12	-0.21
Pigs	-0.31	0.15	0.68	0.20	-0.06	-0.06	-0.37	-0.48
Eggs	-0.44	-0.07	0.23	-0.26	0.27	0.53	0.57	-0.06
Milk	-0.41	0.15	-0.36	-0.03	0.70	-0.38	-0.15	-0.13
Fish	-0.13	-0.67	-0.32	0.40	-0.13	0.02	0.11	-0.49
Cereals	0.45	0.29	0.05	-0.13	0.06	-0.32	0.52	-0.56
Legume	0.43	-0.17	-0.05	-0.36	0.34	0.46	-0.46	-0.32
Fruit	0.13	-0.62	0.45	-0.25	0.24	-0.46	0.06	0.23

v1	v2	v3
1	2	3
4	5	6
7	8	9

v_new
1+2+ 0.5*3 = **4.5**
4+5+ 0.5*6 = **12**
7+8+ 0.5*9 = **19.5**

v_new = 1*v1 + 1*v2 + 0.5 * v3

PC1: -0.33 meat - 0.31 pigs - 0.44 eggs - …

PC2: -0.05 meat + 0.15 pigs - 0.07 eggs + …

Food example

PCA

Scores (new variables) extracted by `pc_food$x`

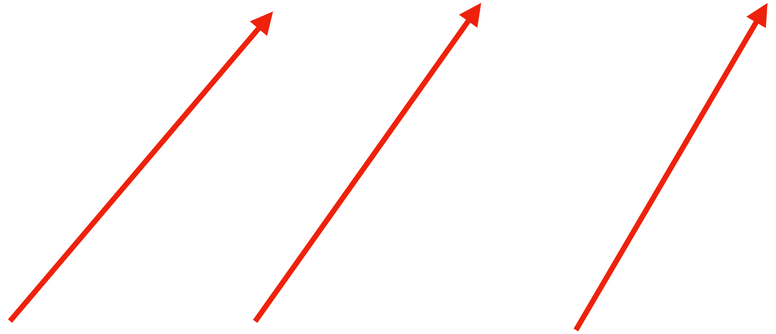
```
scores_food <- pc_food$x
round(scores_food, digits = 2)
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Albania	2.94	1.40	-1.60	-0.52	-1.02	0.22	-0.73	0.52
Austria	-1.61	0.65	1.60	0.30	0.41	0.11	0.19	-0.03
Belg.Lux.	-1.43	-0.13	0.18	-0.70	-0.46	0.27	0.28	-0.08
Bulgaria	2.70	1.02	0.32	-0.10	-0.50	-0.63	0.69	-0.26
Czechoslovakia	-0.23	0.78	1.09	0.36	-0.80	-0.38	0.15	0.17
Denmark	-2.36	-0.34	-0.72	1.25	-0.11	0.07	0.08	-0.73
East.Germany	-1.03	-0.04	1.01	1.07	-0.82	0.55	0.38	0.20
Finland	-1.49	0.92	-2.26	0.91	0.87	-0.55	0.00	0.21
France	-1.46	-1.17	0.27	-1.73	-0.78	-1.10	-0.28	-0.40
Greece	1.97	-1.50	-0.65	-1.44	1.16	0.21	-0.10	-0.52
Hungary	1.50	0.86	1.84	0.26	0.44	0.83	-0.45	-0.36
Ireland	-2.46	0.84	-0.06	-0.92	0.28	0.30	0.20	0.03
Italy	1.22	-0.88	0.39	-0.70	0.43	-0.24	0.37	0.53

v1	v2	v3
1	2	3
4	5	6
7	8	9

v_new
1+2+ 0.5*3 = **4.5**
4+5+ 0.5*6 = **12**
7+8+ 0.5*9 = **19.5**

v_new = 1*v1 + 1*v2 + 0.5 * v3



PC1: -0.33 meat - 0.31 pigs - 0.44 eggs - …

PC1 already contains information from all 8 foods

Food example

PCA

PC1: -0.33 meat - 0.31 pigs - 0.44 eggs - …
Need scale the data before multiply

```
scores_food <- pc_food$x
round(scores_food, digits = 2)
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Albania	2.94	1.40	-1.60	-0.52	-1.02	0.22	-0.73	0.52
Austria	-1.61	0.65	1.60	0.30	0.41	0.11	0.19	-0.03
Belg.Lux.	-1.43	-0.13	0.18	-0.70	-0.46	0.27	0.28	-0.08
Bulgaria	2.70	1.02	0.32	-0.10	-0.50	-0.63	0.69	-0.26
Czechoslovakia	-0.23	0.78	1.09	0.36	-0.80	-0.38	0.15	0.17
Denmark	-2.36	-0.34	-0.72	1.25	-0.11	0.07	0.08	-0.73
East.Germany	-1.03	-0.04	1.01	1.07	-0.82	0.55	0.38	0.20
Finland	-1.49	0.92	-2.26	0.91	0.87	-0.55	0.00	0.21
France	-1.46	-1.17	0.27	-1.73	-0.78	-1.10	-0.28	-0.40
Greece	1.97	-1.50	-0.65	-1.44	1.16	0.21	-0.10	-0.52
Hungary	1.50	0.86	1.84	0.26	0.44	0.83	-0.45	-0.36
Ireland	-2.46	0.84	-0.06	-0.92	0.28	0.30	0.20	0.03
Italy	1.22	-0.88	0.39	-0.70	0.43	-0.24	0.37	0.53

```
food[1,]
```

```
        Meat Pigs Eggs Milk Fish Cereals Legume Fruit
Albania 10.1  1.4  0.5  8.9  0.2    42.3    5.5   1.7
```

```
# scale the data
food_s <- scale(food)
round(food_s[1,], digits = 2) # after centering and scaling
```

Meat	Pigs	Eggs	Milk	Fish	Cereals	Legume	Fruit
0.08	-1.83	-2.24	-1.16	-1.20	0.92	1.22	-1.35

```
# loading for PC1 is the first column
round(loading_food[,1], digits = 2)
```

Meat	Pigs	Eggs	Milk	Fish	Cereals	Legume	Fruit
-0.33	-0.31	-0.44	-0.41	-0.13	0.45	0.43	0.13

```
# let us muliple element by element, then sum together
prod_by_element <- as.numeric(food_s[1,]) * loading_food[, 1]
round(prod_by_element, digits = 2)
```

Meat	Pigs	Eggs	Milk	Fish	Cereals	Legume	Fruit
-0.03	0.58	1.00	0.47	0.15	0.42	0.53	-0.17

```
# sum all elements together
sum(prod_by_element)
```

```
[1] 2.943309
```

Food example
PCA

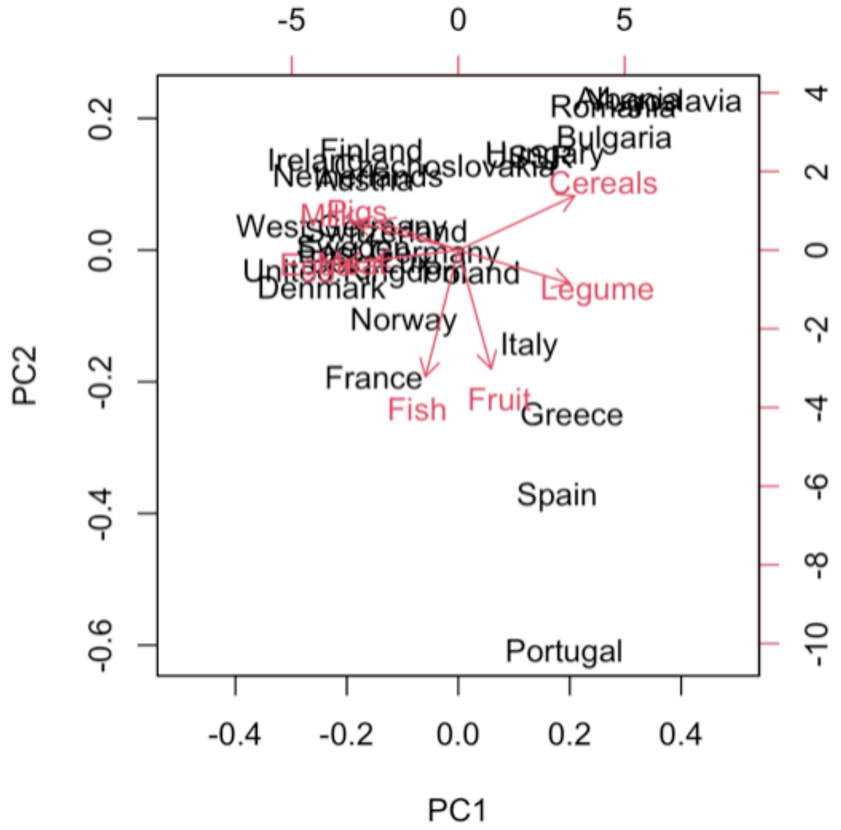
```
# by default, pc1 pc2
# set x-axis limit
# same as biplot(pc_food, choices = c(1,2))
biplot(pc_food, xlim = c(-0.5, 0.5))
```

Scatter plot can be made for each **pair** of the (numeric) measurements in the same dataset

Instead of the original data, we can **visualize the new variables (PCs)**

Can add label (country)

Certain countries cluster together



Food example
PCA

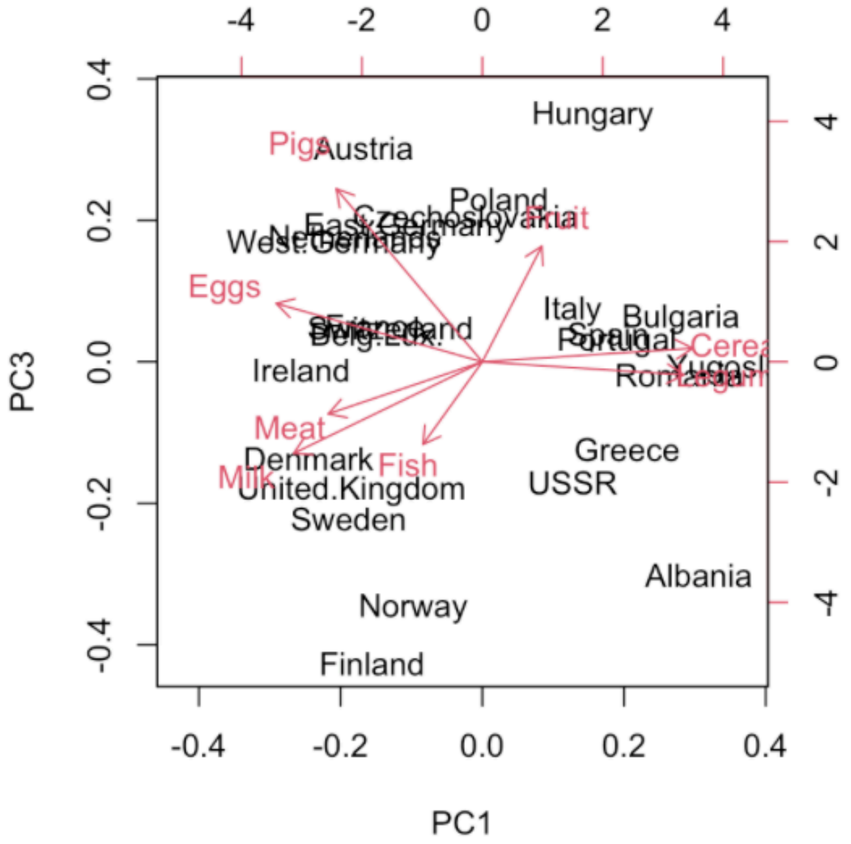
```
# choose pc1 and pc3
biplot(pc_food, choices = c(1,3))
```

Can select PC1 and PC3, PC4 if you wish

Note: these plots are not confirmatory

Unsupervised learning: no outcome label

Should also combine variance explained by PCs

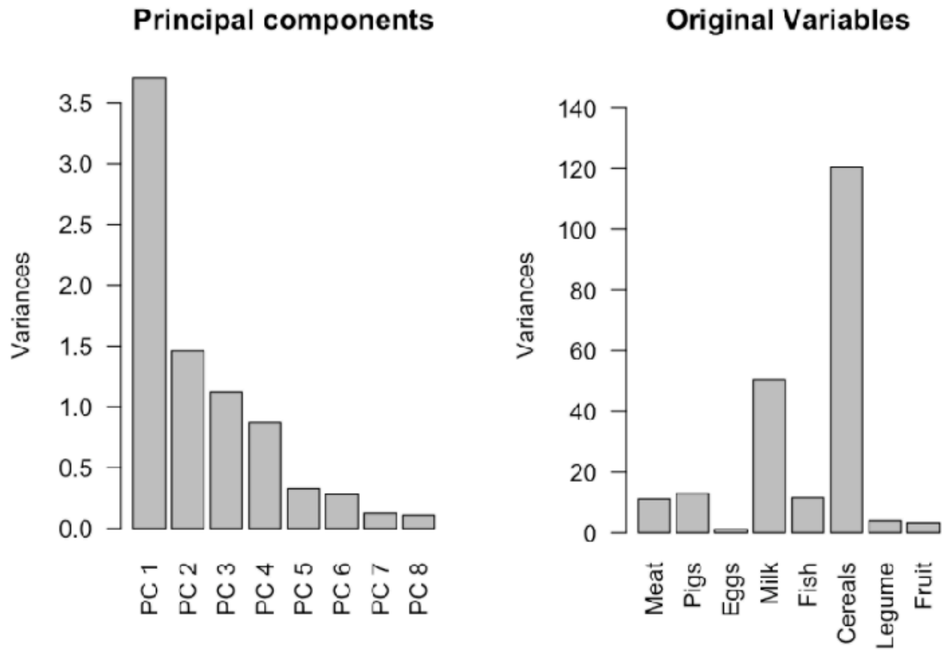


Food example

PCA

PCs are ordered by the amount of variance explained from the data

You might choose to only keep 3 PCs (from 8 to 3 dimensional) that explain 80% variance

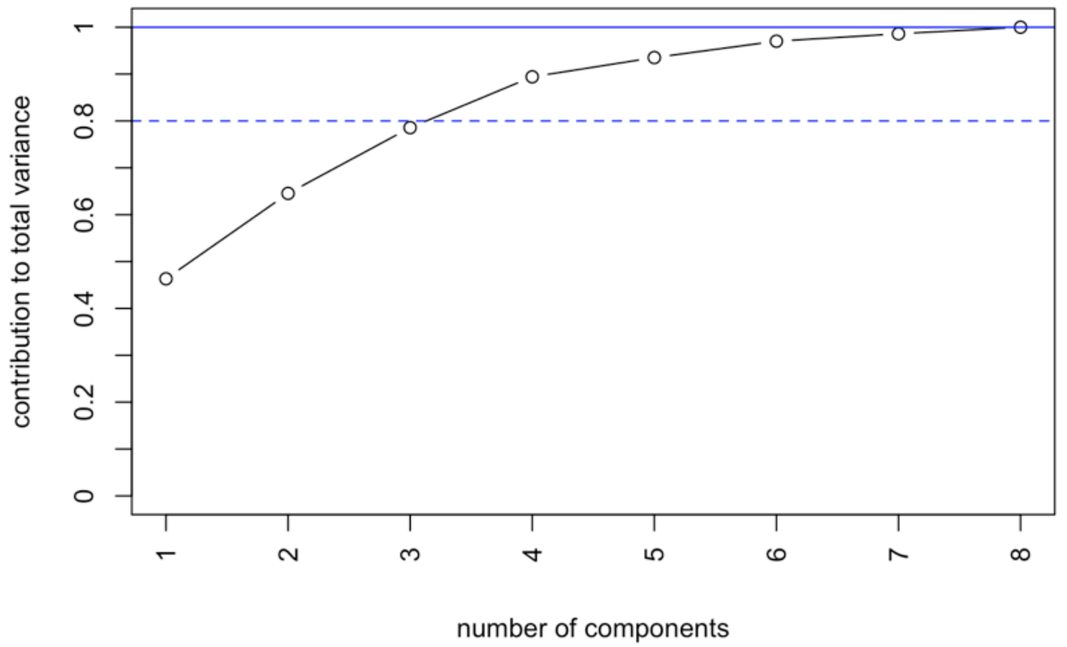



```r
# variance explained by each PC
pc_food_var <- pc_food$sdev^2
# proportion
pc_food_pve <- pc_food_var/sum(pc_food_var)
# print out, keep 3 digits
round(pc_food_pve, digits = 3)
```

```
[1] 0.463 0.182 0.140 0.108 0.041 0.035 0.016 0.014
```

```r
# cumulative of 1st, 2nd, ... 8th PC
cumsum(pc_food_pve)
```

```
[1] 0.4632302 0.6454168 0.7857372 0.8941976 0.9352708 0.9702365 0.9860940
[8] 1.0000000
```



NCI 60 example

PCA

64 cancer cell lines, 6830 gene expression measurements

Ignore the cancer types, as PCA is unsupervised - but we can check how well the clustering corresponds to the true label.

Goal: reduce the dimensionality from 6830 to a more manageable size

(Code available in Exercise 2)

NCI 60 example

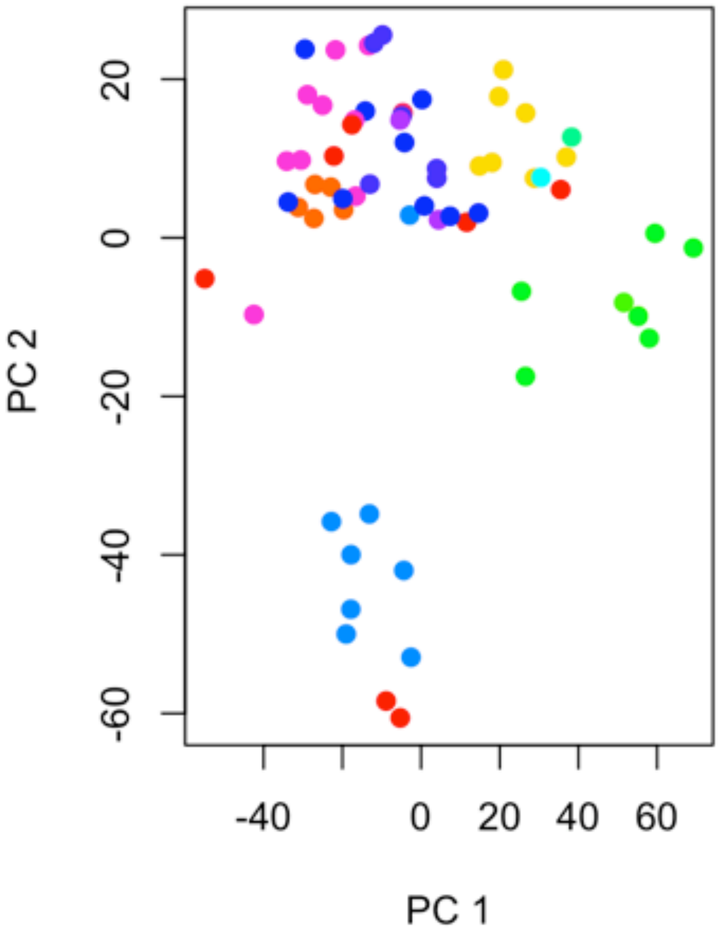
PCA: first 3 PCs

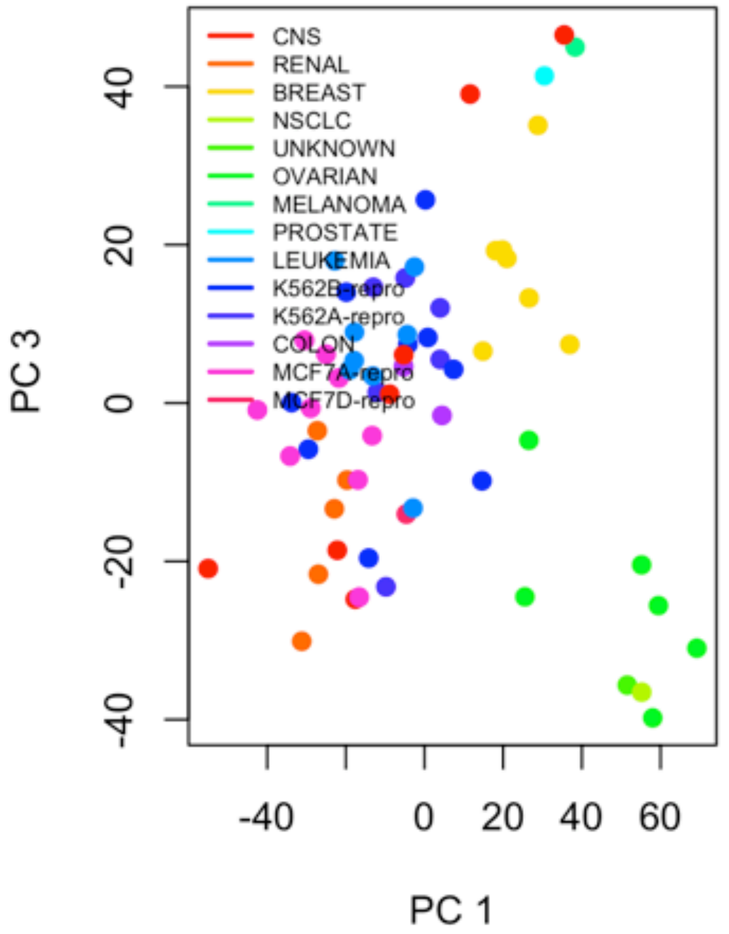
We visualize

PC1 vs PC2

PC1 vs PC3

… along with the cancer types, to see if there are any patterns





NCI 60 example

PCA: proportion of variance explained

In total 64 PCs are produced (from 6830 genes - features)

Still a bit too many to analyse

We can keep the first x PC that explain 80% variance - 32 PCs

It is common to do modeling tasks such as prediction on PCs rather than the original data.