HIROSHIMA UNIVERSITY

Fundamental Data Science (30104001)

Lecture 8 — Principal Component Analysis
Cluster Analysis in R

Jorge N. Tendeiro

Hiroshima University

Today

We will learn how to perform the following data analysis methods and their general framework, using RStudio:

- Principal component analysis
- Cluster analysis (clustering)

Data files

Download the following files from folder "Lecture 8" in *Moodle* and save them in the `Stat` folder on the desktop:

- `seiseki.csv` (this file was previously used in Lecture 5)
- `iris.csv`

Principal Component Analysis (PCA)

Today's data

Today we will use the data file `seiseki.csv`:

Scores of 166 2nd year junior high students (Sugiyama et al., 2014).

	A	B	C	D	E	F	G	H	I	J
1	ID	kokugo	shakai	sugaku	rika	ongaku	bijutu	taiiku	gika	eigo
2	1	30	43	51	63	60	66	37	44	20
3	2	39	21	49	56	70	72	56	63	16
4	3	29	30	23	57	69	76	33	54	6
5	4	95	87	77	100	77	82	78	96	87
6	5	70	71	78	67	72	82	46	63	44
7	6	67	53	56	61	61	76	70	66	40
8	7	29	26	44	52	37	68	33	43	13
9	8	56	54	37	59	35	64	53	67	7
10	9	45	21	7	44	16	52	34	46	3
11	10	68	41	29	81	55	71	29	72	51
12	11	50	43	80	73	35	50	42	65	10

Variable	Description
`ID`	ID number
`kokugo`	Japanese
`shakai`	Society
`sugaku`	Math

Variable	Description
`rika`	Science
`ongaku`	Music
`bijutu`	Art
`taiiku`	PE

Variable	Description
`gika`	Industrial arts and homemaking
`eigo`	English

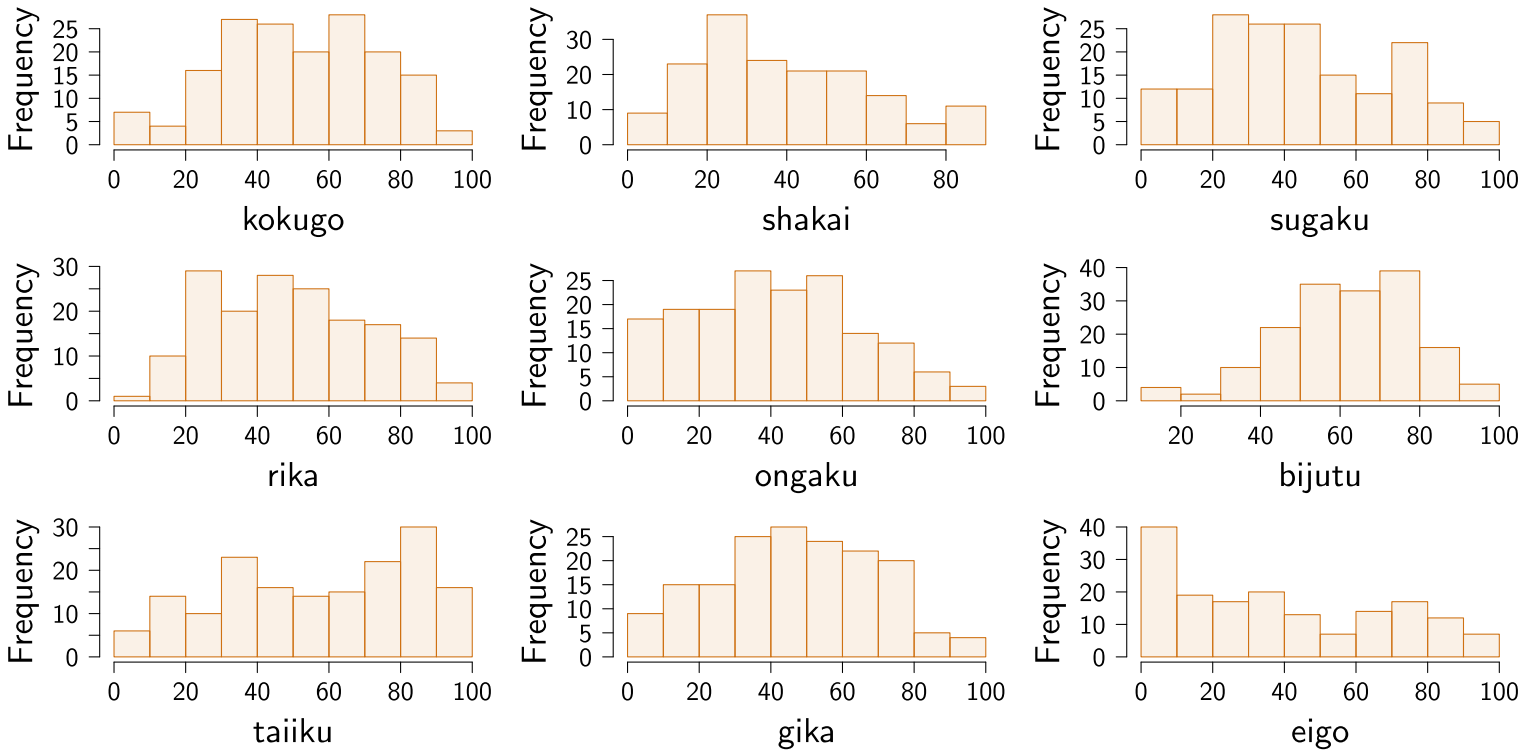
Today's data

	A	B	C	D	E	F	G	H	I	J
1	ID	kokugo	shakai	sugaku	rika	ongaku	bijutu	taiiku	gika	eigo
2	1	30	43	51	63	60	66	37	44	20
3	2	39	21	49	56	70	72	56	63	16
4	3	29	30	23	57	69	76	33	54	6
5	4	95	87	77	100	77	82	78	96	87
6	5	70	71	78	67	72	82	46	63	44
7	6	67	53	56	61	61	76	70	66	40
8	7	29	26	44	52	37	68	33	43	13
9	8	56	54	37	59	35	64	53	67	7
10	9	45	21	7	44	16	52	34	46	3
11	10	68	41	29	81	55	71	29	72	51
12	11	50	43	80	73	35	50	42	65	10
13	12	70	61	61	71	55	56	35	67	22

Goals:

1. Learn about the students' overall characteristics based on the scores on the nine subjects.
2. Visualize the students' characteristics by means of a plot.

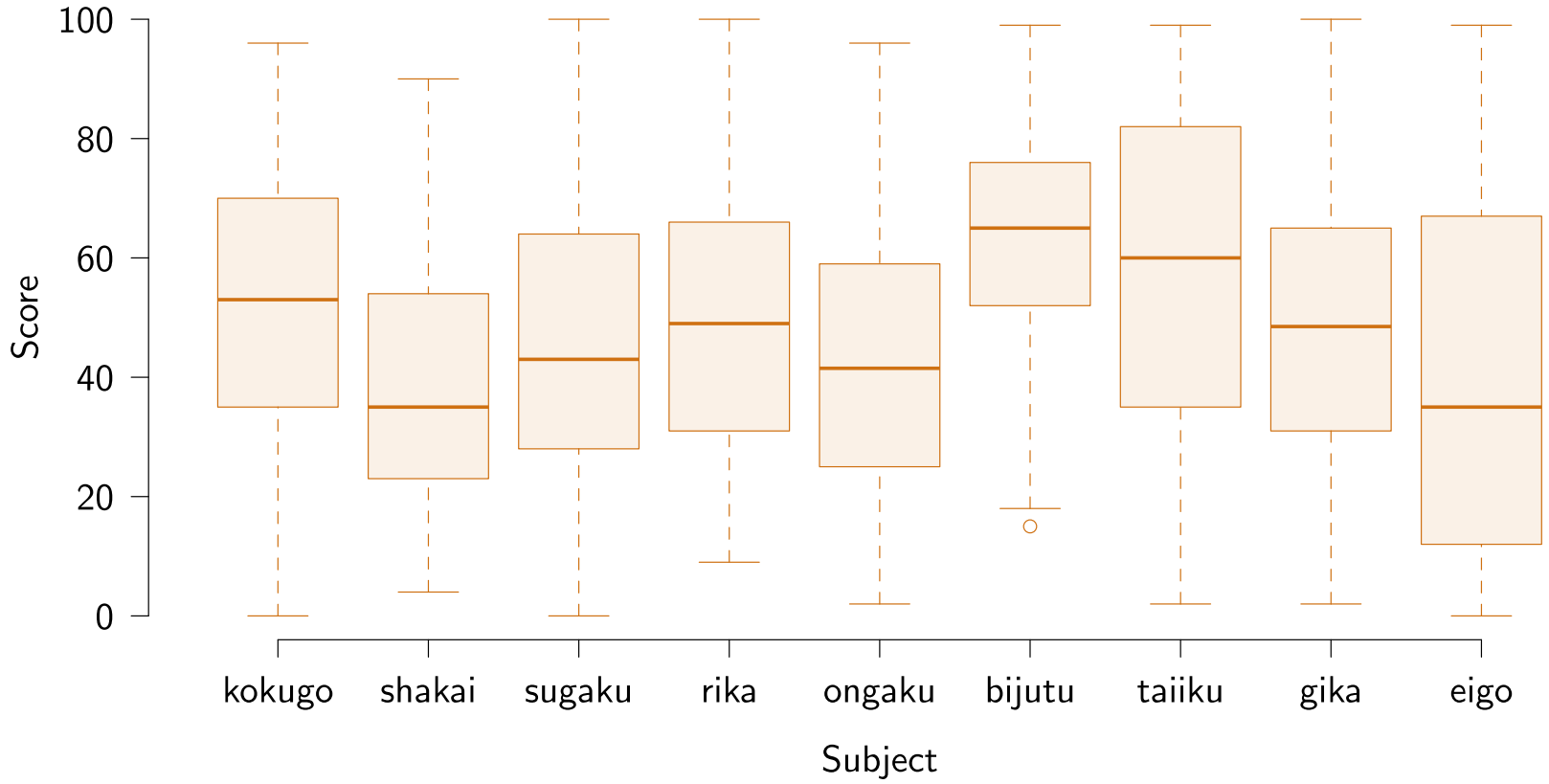
Histograms



There are too many plots!! It's hard to grasp the overall picture.

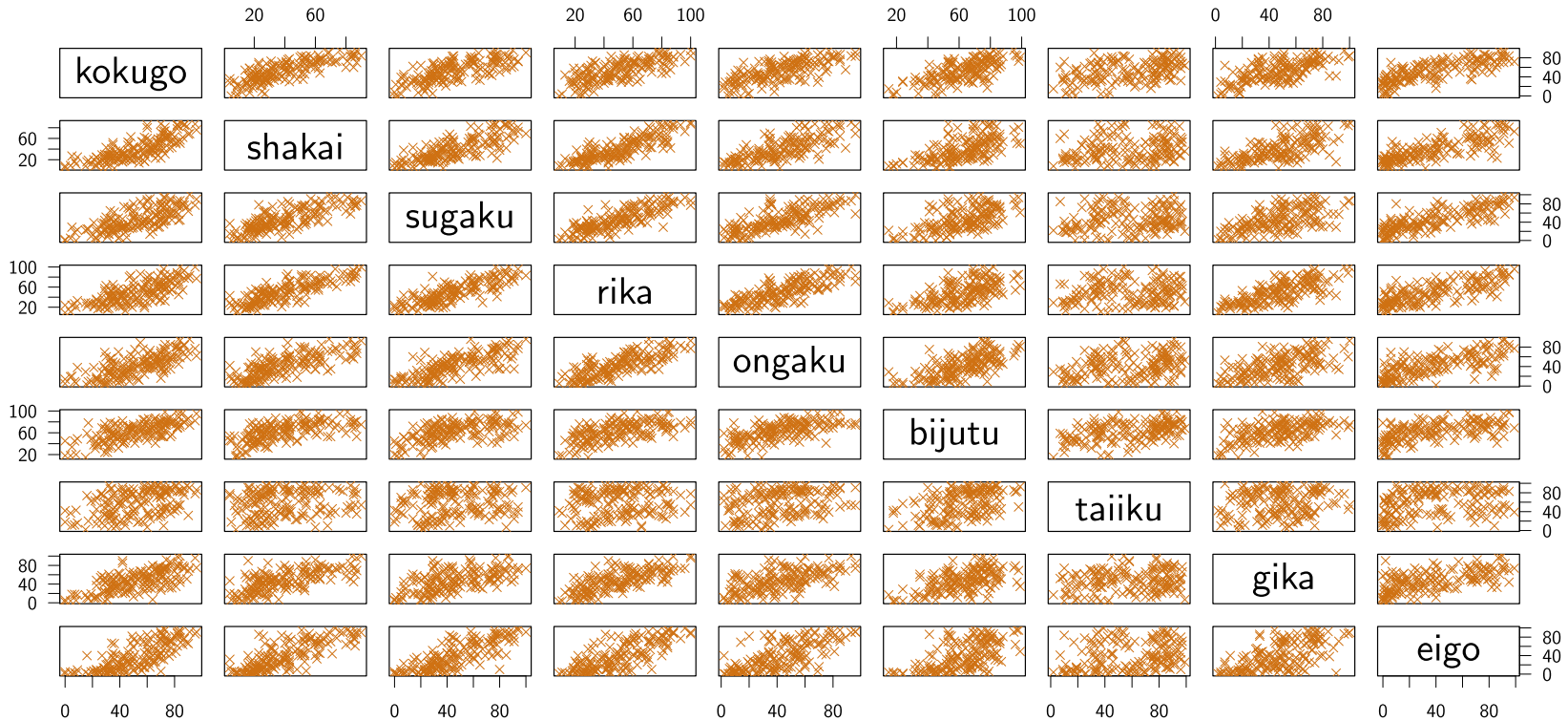
For example, how do the nine different topics relate to each other?

Boxplots



From this boxplot it is also hard to grasp the overall picture.

Scatterplots



Scatterplots only display the relationship between any two topics.

Biplot using principal component analysis

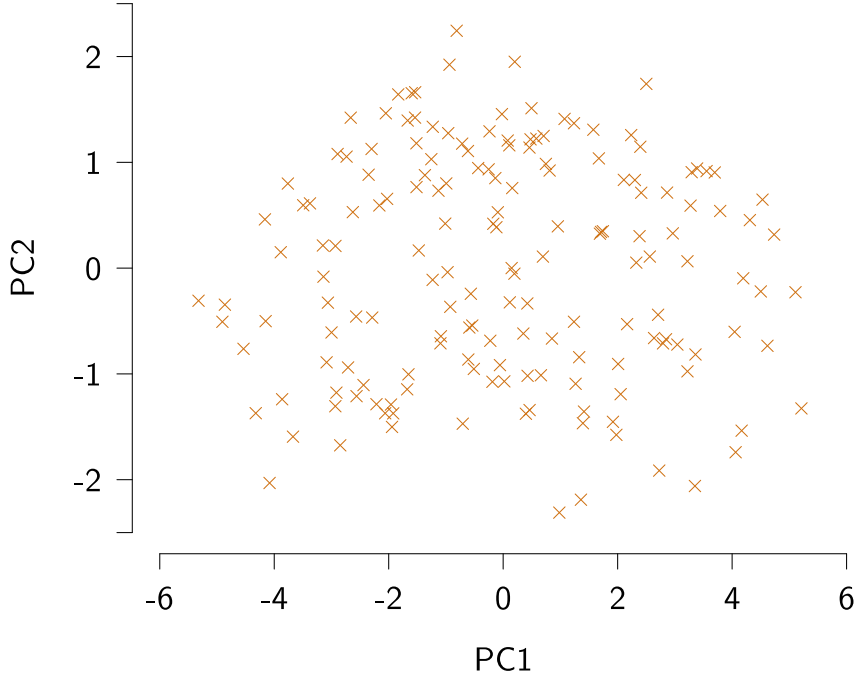
Ideally, we'd like to represent the nine topics using one plot only.

Principal component analysis (PCA) is a method that allows reducing the dimensionality of a data set.

For example, using PCA, we can represent the nine topics on a single two-dimensional plot. $\longrightarrow$

This plot is known as a biplot.

Biplots summarize the associations between all variables in a data set.



Each point on the plot $=$ 2nd year junior high student (166 in total).

Principal component analysis in a nutshell

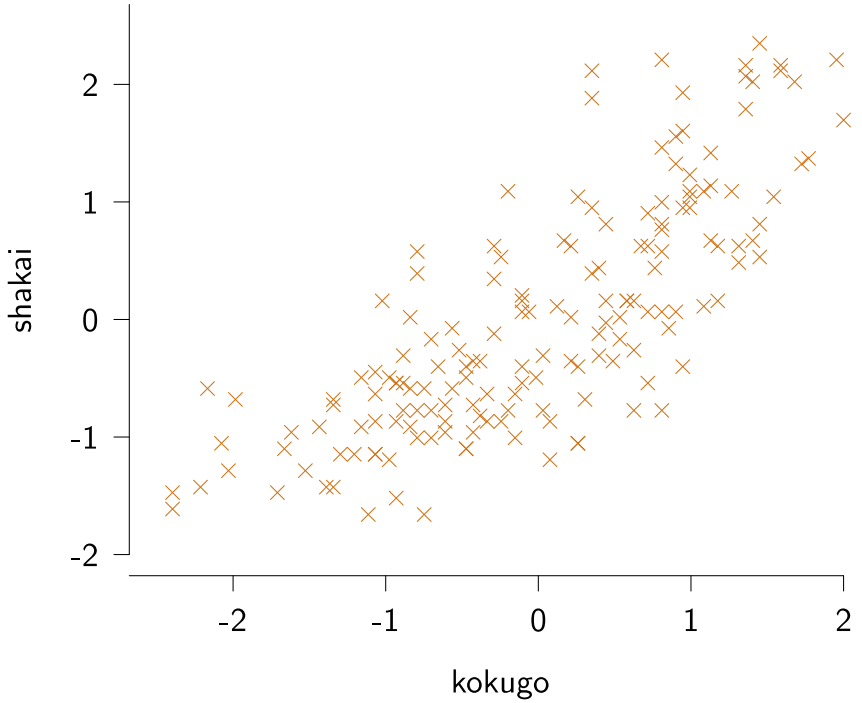
To understand how this 'magic' works, let's focus on the first two `seiseki` variables only (`kokugo`, `shakai`).

This is the scatterplot of both variables, standardized *.

By 'standardized' we mean that each variable, say V, was centered and scaled:

$$V \underset{\text{center}}{\longrightarrow} V - \overline{V} \underset{\text{scale}}{\longrightarrow} \frac{V - \overline{V}}{s_V}$$

(s_V = standard deviation of variable V.)



*This is often done in statistics. The main advantage is that the variables become *unitless* and can be compared directly, even if their original units are different (e.g., years, yen, income, or distance).

Principal component analysis in a nutshell

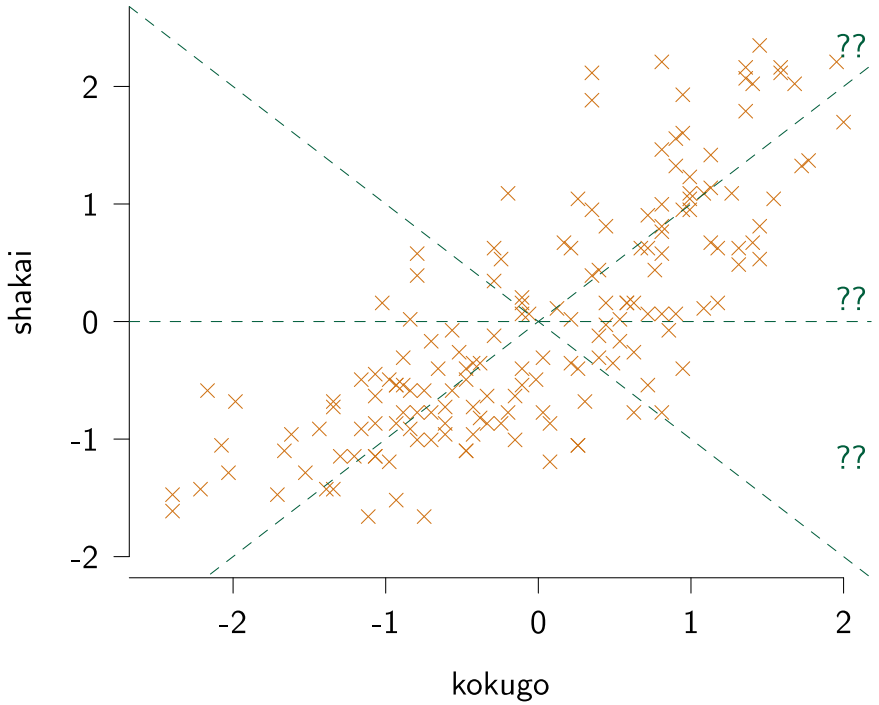
Here's the challenge:

> *These data (the set of `kokugo` and `shakai` variables) are two-dimensional (i.e., they are on a plane). How can we represent these data on a one-dimensional space (i.e., a line), while losing the least amount of information possible?*

The idea is to reduce the dimensionality of the data, while still explaining as much of the data as we can.

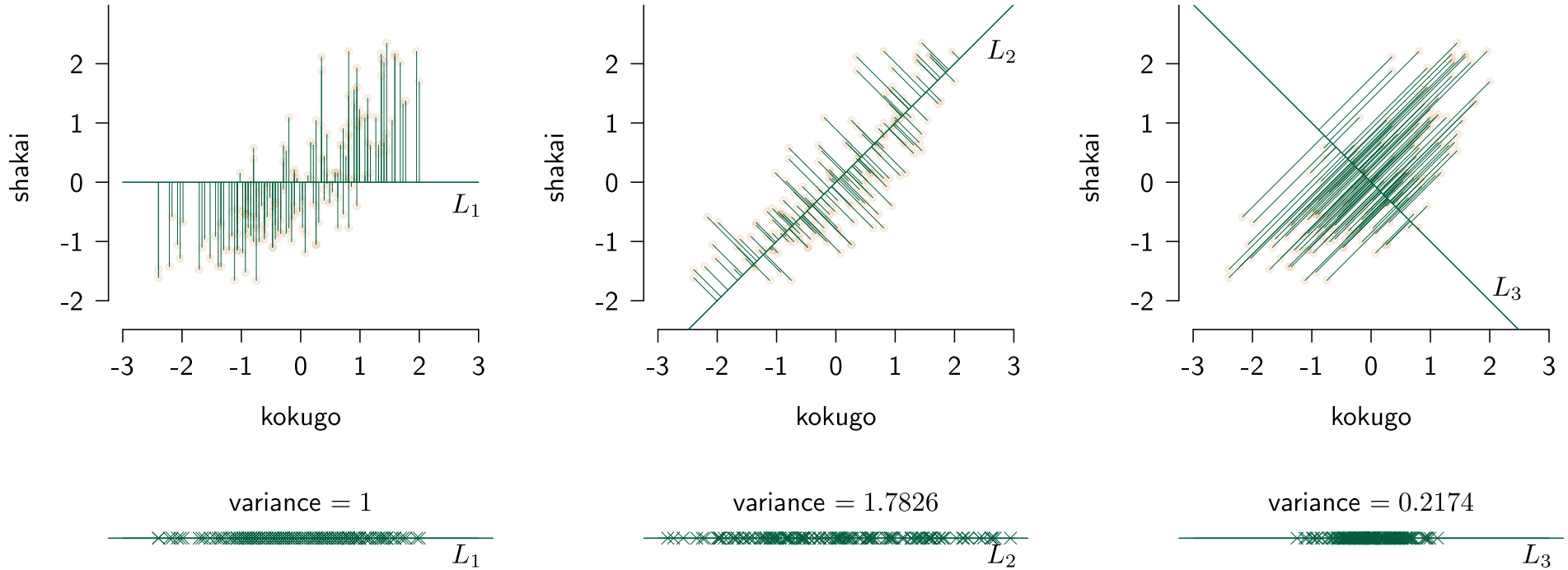
Q: Why?
A: For simplicity (lower dimension spaces are simpler).

Q: But, how to choose the *best* line?...



Principal component analysis in a nutshell

PCA works by finding 'the' line such that *the variance of the points projected on the line is maximal*.



From the three candidate lines above, L_2 is the best.

Principal component analysis in a nutshell

Summary

- PCA finds new axes for the data.
- Actually, the new axes are orthogonal (i.e., they are at right angles).
- The best axis, called the first principal component (1st PC), is such that the data projected onto it have maximum variance.

  > *In statistics, variation = information.*

- The second best axis, called the second principal component (2nd PC), is orthogonal to the 1st PC.
  It is chosen such that the data projected onto it have the second largest variance.

  > *Orthogonal* so that information explained by different PCs does not overlap.

- ...

Principal component analysis in a nutshell

For data sets with k variables, we can find k PCs.

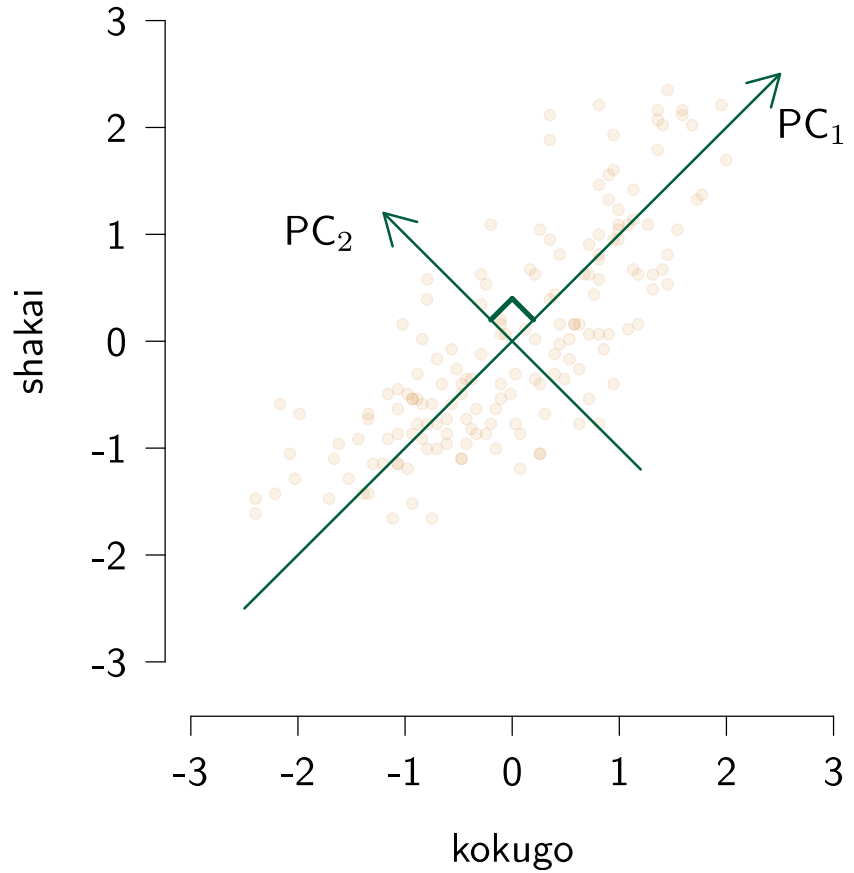
- With k PCs we can explain 100% of the data.

  > No dimension reduction performed. Uninteresting.

- PCA aims at selecting a small number of PCs (say, up to 3), while retaining most of the information in the data.

  > Represent the data in a (often much) lower dimensional space. Super useful!

Principal component analysis in a nutshell



PCA in R

Input

```
# Import seiseki.csv and save the data in object X:
X ← read.csv("seiseki.csv")

# Show the first 3 data rows:
head(X, n = 3)
```

```
# Check variable names:
colnames(X)
```

```
# Drop the first column (IDs) as it is unnecessary;
#   save the data in object Y:
Y ← X[ , -1]

# Show the first 3 data rows:
head(Y, n = 3)
```

```
# Perform PCA and save the results in object
#   'result':
result ← prcomp(Y,
                scale = TRUE # standardized data
                )
```

Output

```
  ID kokugo shakai sugaku rika ongaku bijutu taiiku g
1  1     30     43     51   63     60     66     37
2  2     39     21     49   56     70     72     56
3  3     29     30     23   57     69     76     33
```

```
 [1] "ID"     "kokugo" "shakai" "sugaku" "rika"   "on
 [9] "gika"   "eigo"
```

```
  kokugo shakai sugaku rika ongaku bijutu taiiku gika
1     30     43     51   63     60     66     37   44
2     39     21     49   56     70     72     56   63
3     29     30     23   57     69     76     33   54
```

PCA — Proportion of explained variance

```
summary(result)
```

```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9
Standard deviation     2.4508 1.0479 0.70060 0.63795 0.54796 0.47059 0.42754 0.41376 0.34909
Proportion of Variance 0.6674 0.1220 0.05454 0.04522 0.03336 0.02461 0.02031 0.01902 0.01354
Cumulative Proportion  0.6674 0.7894 0.84394 0.88916 0.92252 0.94713 0.96744 0.98646 1.00000
```

Proportion of variance
This is the proportion of variance of the observed variables explained by each PC.

Note that:

- Variances explained by different PCs do **not** overlap.
- The sum of the proportions of explained variances across all PCs is always equal to 1.

Example:

> PC_1 explains 66.7% of the total variance of the observed variables.
> PC_2 explains 12.2% of the total variance of the observed variables.
> ...

PCA — Proportion of explained variance

```
summary(result)
```

```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9
Standard deviation     2.4508 1.0479 0.70060 0.63795 0.54796 0.47059 0.42754 0.41376 0.34909
Proportion of Variance 0.6674 0.1220 0.05454 0.04522 0.03336 0.02461 0.02031 0.01902 0.01354
Cumulative Proportion  0.6674 0.7894 0.84394 0.88916 0.92252 0.94713 0.96744 0.98646 1.00000
```

Cumulative proportion

This is the accumulated proportion of variance of the observed variables explained by the first PCs.

Example:

> PC_1 explains 66.7% of the total variance of the observed variables.
> PC_1 and PC_2 explain 66.7%+12.2%=78.9% of the total variance of the observed variables.
>
> The first three PCs explain 66.7%+12.2%+5.5%=84.4% of the total variance of the observed variables.
> ...

A low cumulative proportion of explained variance is **not** good. Always check it!

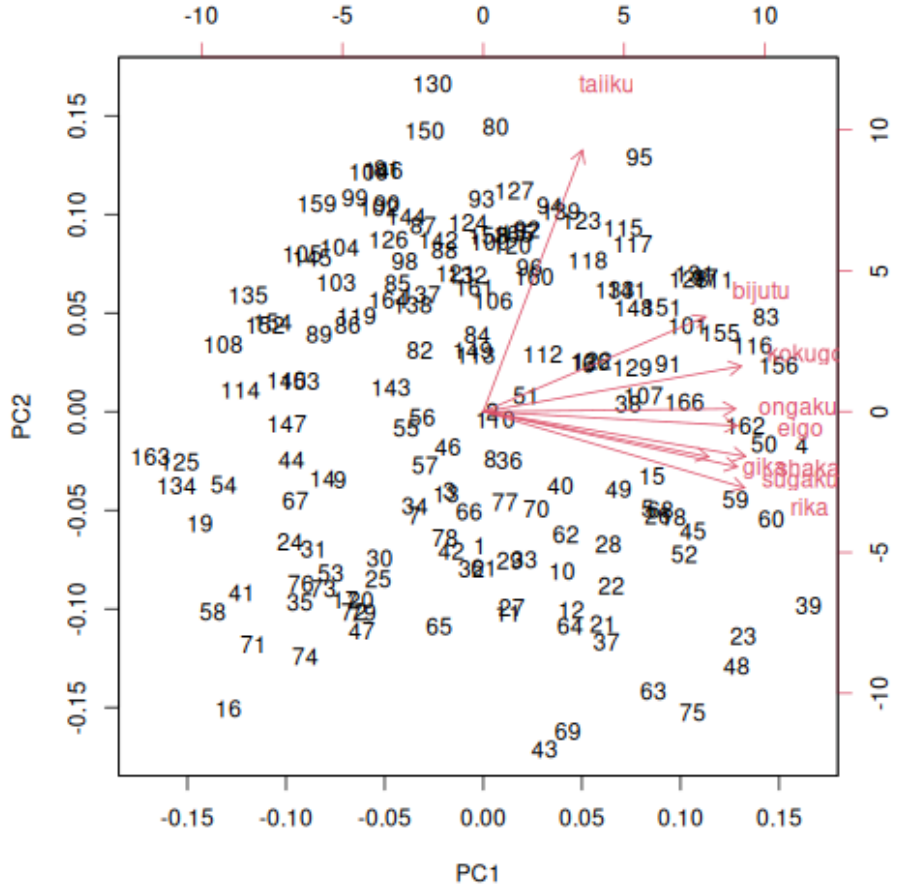
PCA — Biplot

```
biplot(result)
```

The biplot shows the original 9-dimensional data space projected onto the 2-dimensional space spanned by PC_1 and PC_2.

> *Dimension reduction!*

Each numbered label represents a student.



PCA — Biplot

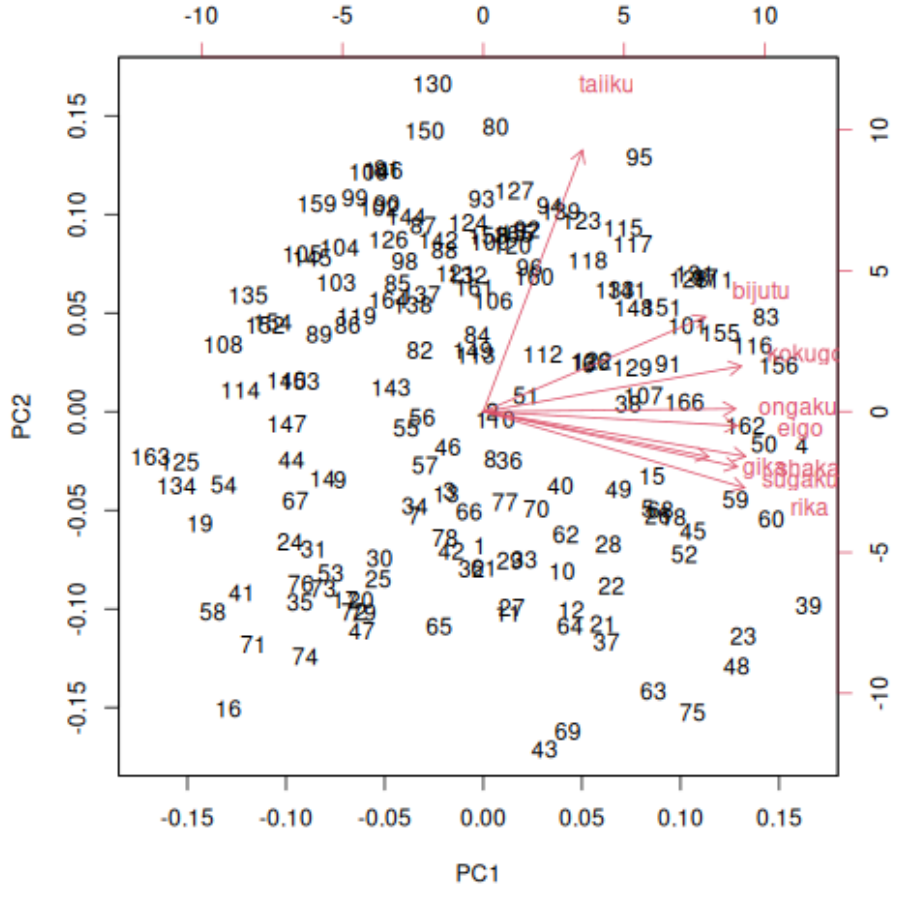
The red arrows display the nine observed variables on the 2-dimensional space spanned by PC_1 and PC_2.

The direction of the red arrows help us to interpret the PCs:

- PC_1: Larger scores on PC_1 indicate large scores on all subjects except for 'taiiku'.
- PC_2: Larger scores on PC_2 indicate a large score on 'taiiku'.

For example, student 75 scored high on all subjects except for 'taiiku'.

We can represent the characteristics of the data in one plot.



The iris data

Download the file `iris.csv` from *Moodle*.

The data set is also available directly in R as data frame `iris`.
See `?iris` for more information.

```
head(iris)  # 150 rows in total
```

```
  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
```

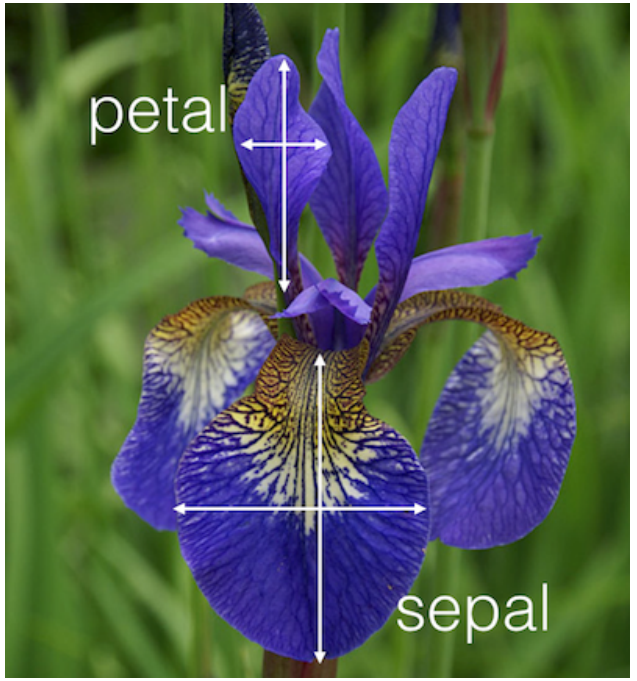


Photo from sebastianraschka.com.

Exercise (1)

1. Fit PCA to the four numerical columns of `iris`. Analyze the standardized data.
   Create a biplot.
   Check the cumulative proportion of explained variance.

2. Check the characteristics of the 61^{st} iris.

Exercise (1) — ANSWER

1. Fit PCA to the four numerical columns of `iris`. Analyze the standardized data.
Create a biplot.
Check the cumulative proportion of explained variance.

Input

```
# Import the data:
iris ← read.csv("iris.csv")
```

```
# Check variable names:
colnames(iris)
```

```
# Drop the first column:
iris ← iris[, -1]
colnames(iris)
```

```
# Fit PCA:
result ← prcomp(iris, scale = TRUE)
```

Output

```
[1] "ID"           "Sepal.Length" "Sepal.Width"  "Pet
```

```
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Pet
```

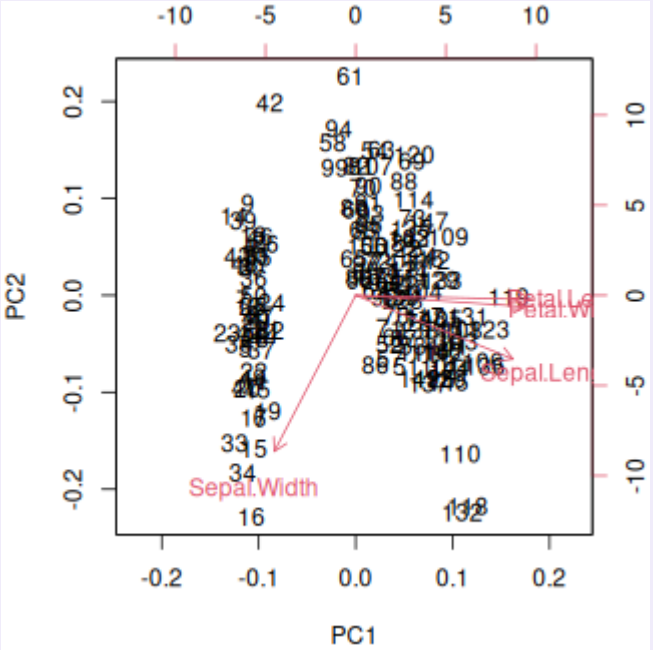
Exercise (1) — ANSWER

1. Fit PCA to the four numerical columns of `iris`. Analyze the standardized data.
Create a biplot.
Check the cumulative proportion of explained variance.

Input

```
# Render the biplot:
biplot(result)
```

Output



Exercise (1) — ANSWER

1. Fit PCA to the four numerical columns of `iris`. Analyze the standardized data.
Create a biplot.
Check the cumulative proportion of explained variance.

Input

```
# Render the biplot:
summary(result)
```

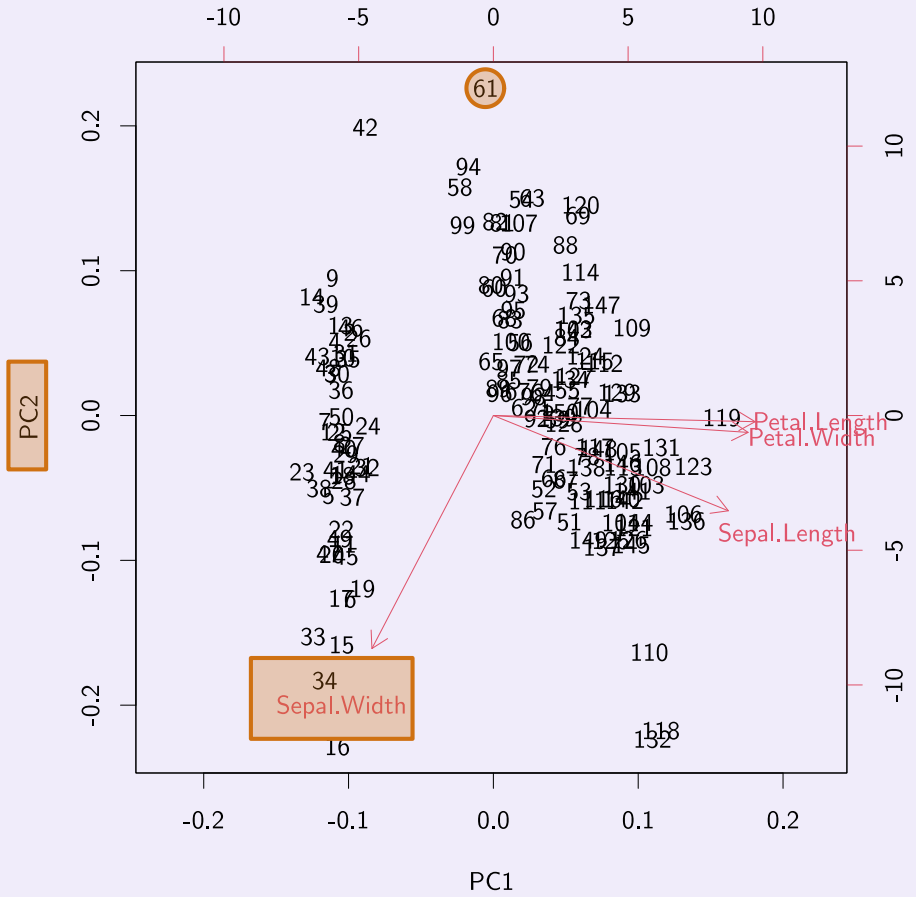
Output

```
Importance of components:
                          PC1    PC2     PC3     PC4
Standard deviation     1.7084 0.9560 0.38309 0.14393
Proportion of Variance 0.7296 0.2285 0.03669 0.00518
Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

The first two PCs explain 95.81% of the total variance of the observed variables.

Exercise (1) — ANSWER

2. Check the characteristics of the 61^{st} iris.



The 1st PC correlates positively with `Petal.Length`, `Petal.Width`, and `Sepal.Length`.

Since the 61^{st} iris has a PC1 score of about 0, we conclude that this iris has 'normal' values `Petal.Length`, `Petal.Width`, and `Sepal.Length`.

The 2nd PC correlates negatively with `Sepal.Width`.

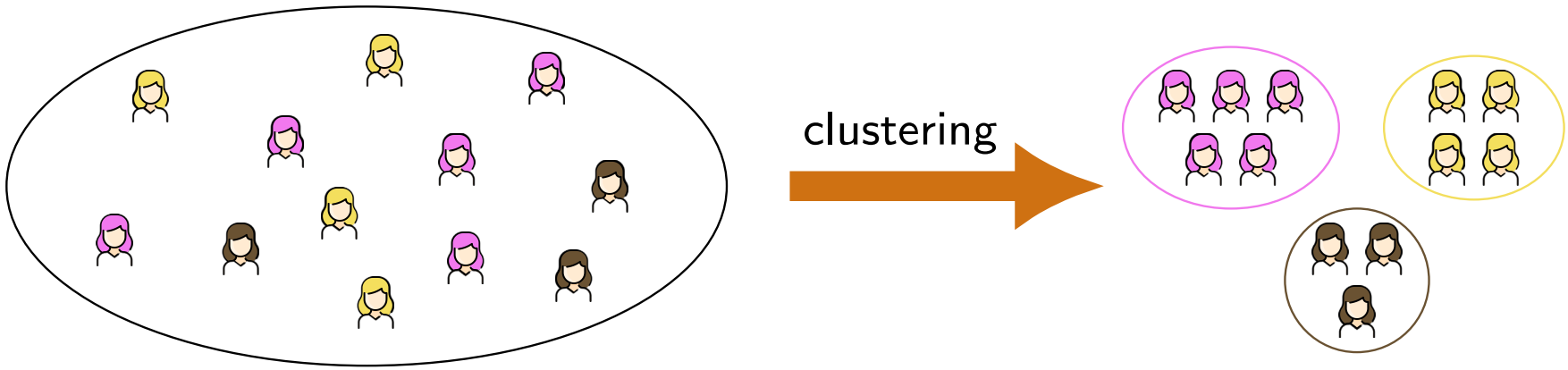
Since the 61^{st} iris has a very large PC2 score, we conclude that this iris has a very short `Sepal.Width`.

Cluster Analysis

Cluster Analysis

Clustering:

> *Automatic method of classifying data into groups such that within-group data are very similar and between-group data are dissimilar, based on a certain classification rule.*

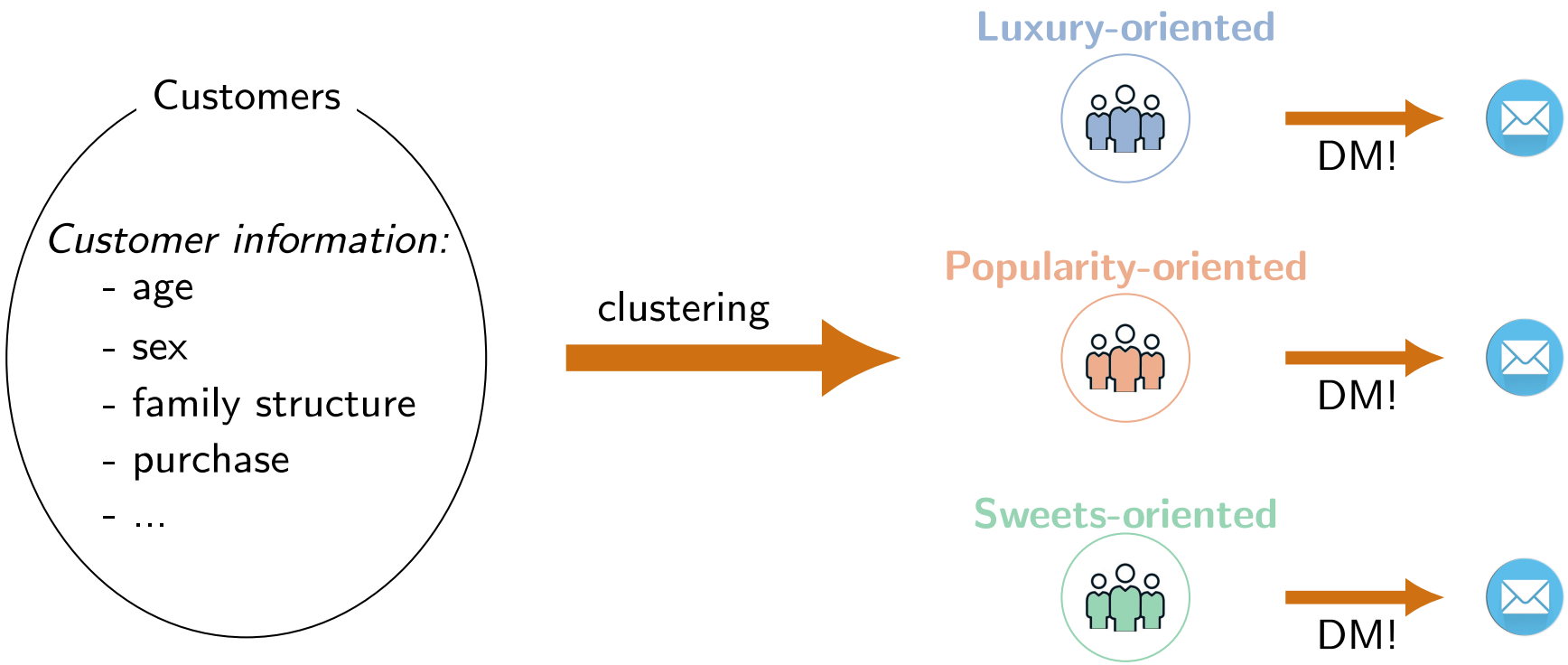


Example:

- Text data $\longrightarrow$ classify tweets.
- Lyrics data $\longrightarrow$ classify songs.

Cluster Analysis — Example

Cluster customers to decide what to direct message (DM) to each customer.



k-means method

The k-means method is a popular algorithm to perform clustering.

The general lines of the k-means method are as follows:

1. Set the number of clusters, k.
   Set k representative points, one per cluster.

2. Repeat until the representative points do not change:

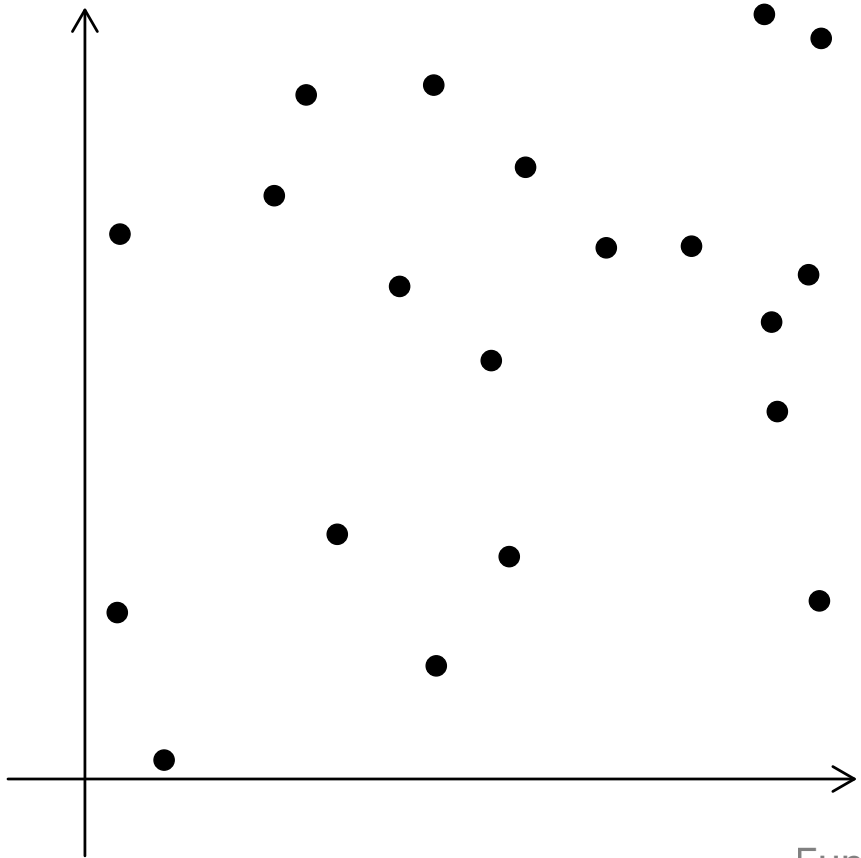
   2a. Assign each data point to the same cluster as the nearest representative point.

   2b. Update each cluster's representative point, to be the mean value of the data points assigned to that cluster.

Let's visualize this algorithm!

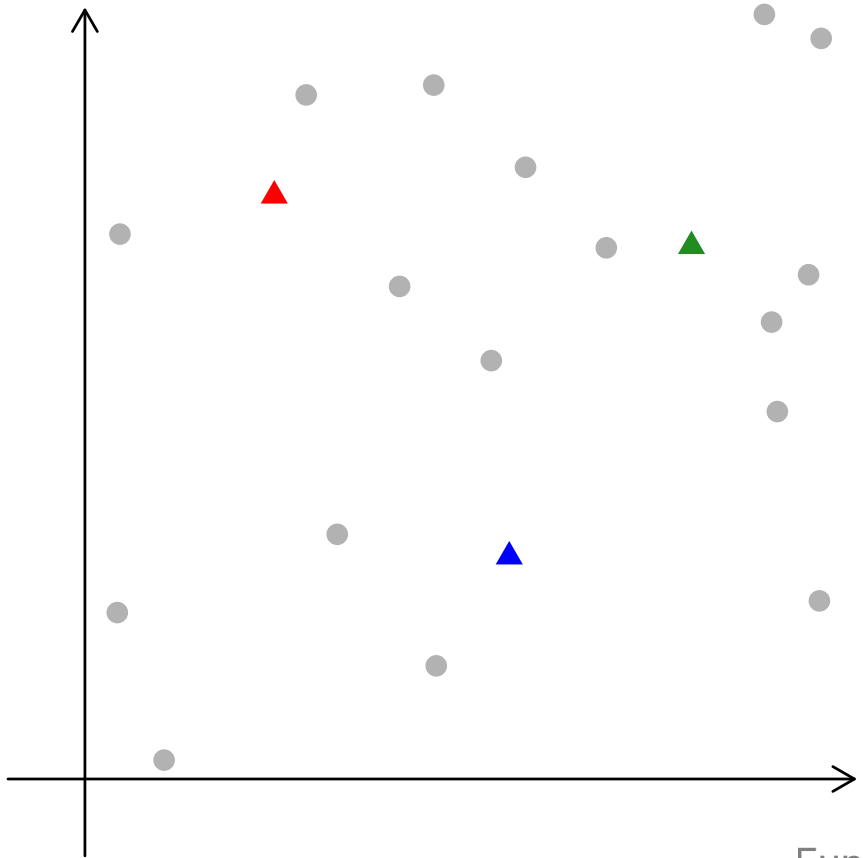
The k-means method illustrated

The data points that we want to cluster:



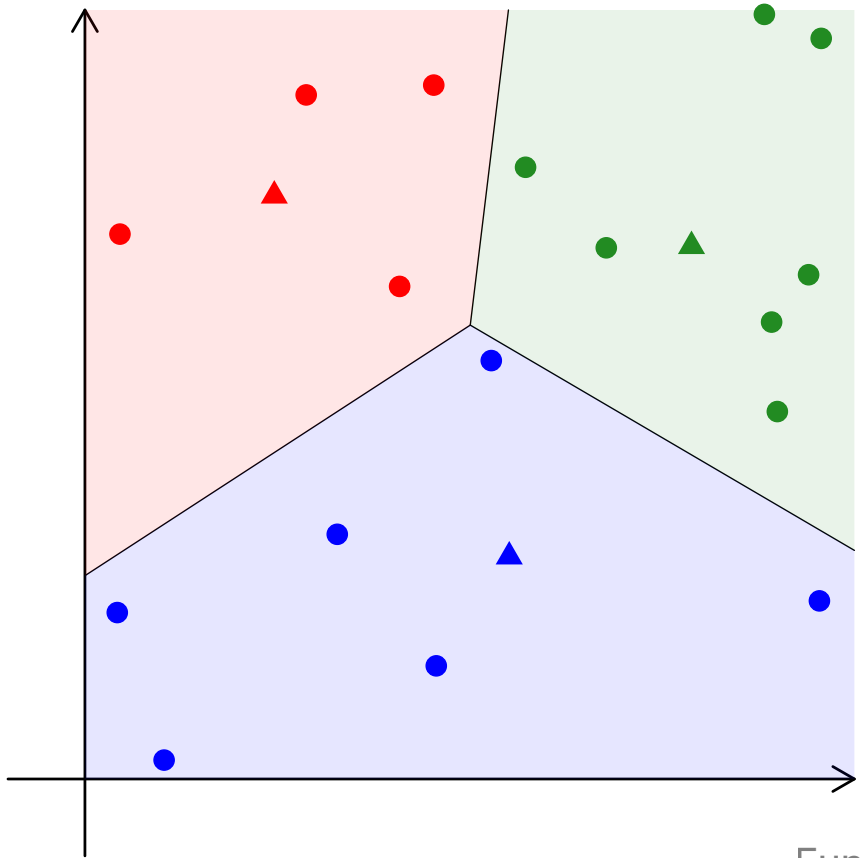
The k-means method illustrated

1. Set the number of clusters, k. $\longrightarrow$ Here we set $k = 3$.
   Set k representative points, one per cluster. $\longrightarrow$ Coloured triangles.



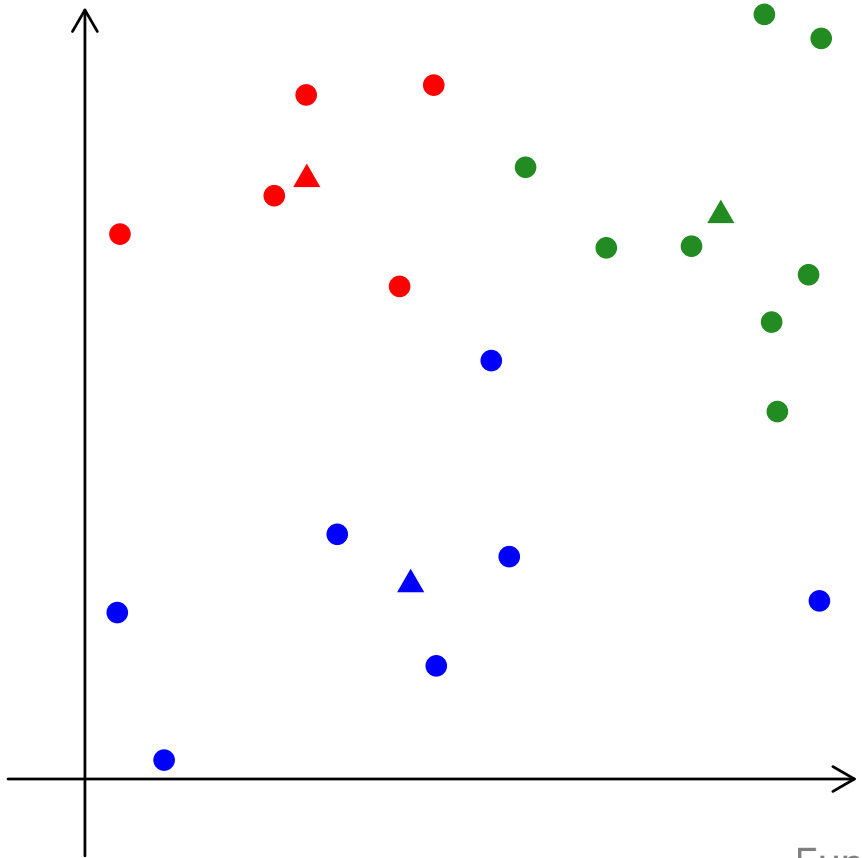
The k-means method illustrated

2. Repeat until the representative points do not change:
   2a. Assign each data point to the same cluster as the nearest representative point.



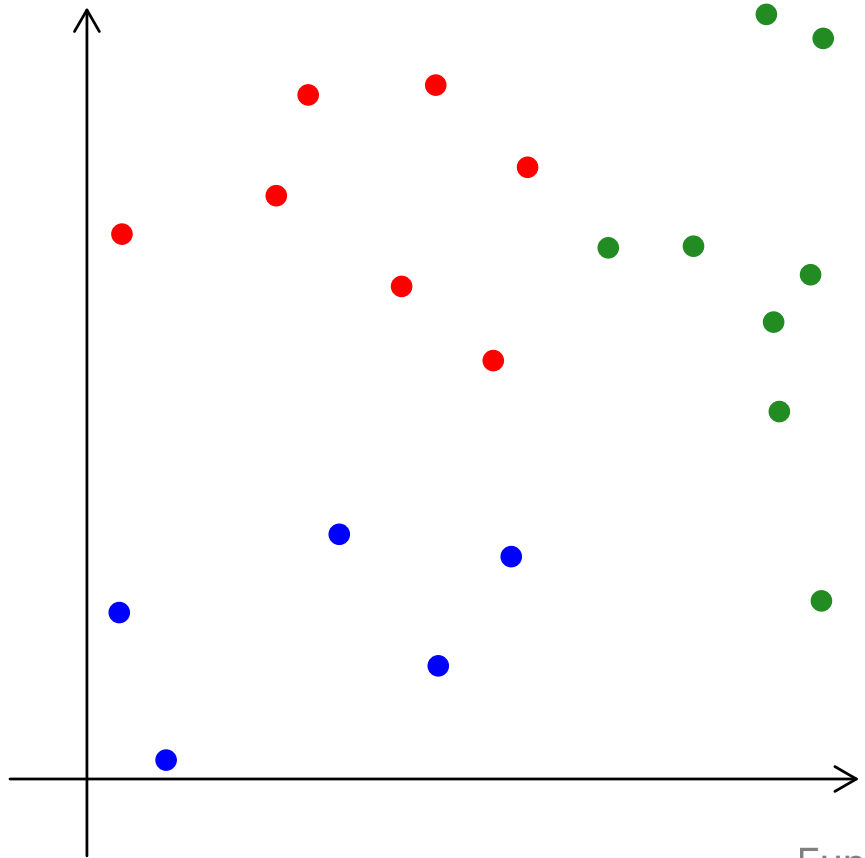
The k-means method illustrated

2. Repeat until the representative points do not change:
   2b. Update each cluster's representative point $\longrightarrow$ New coloured triangles.



The k-means method illustrated

Repeat 2a and 2b until the representative points do not change:



The k-means method in R

Let's use the `iris` data set once more.

```
head(iris)  # 150 rows in total
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1         3.5          1.4         0.2
2          4.9         3.0          1.4         0.2
3          4.7         3.2          1.3         0.2
4          4.6         3.1          1.5         0.2
5          5.0         3.6          1.4         0.2
6          5.4         3.9          1.7         0.4
```

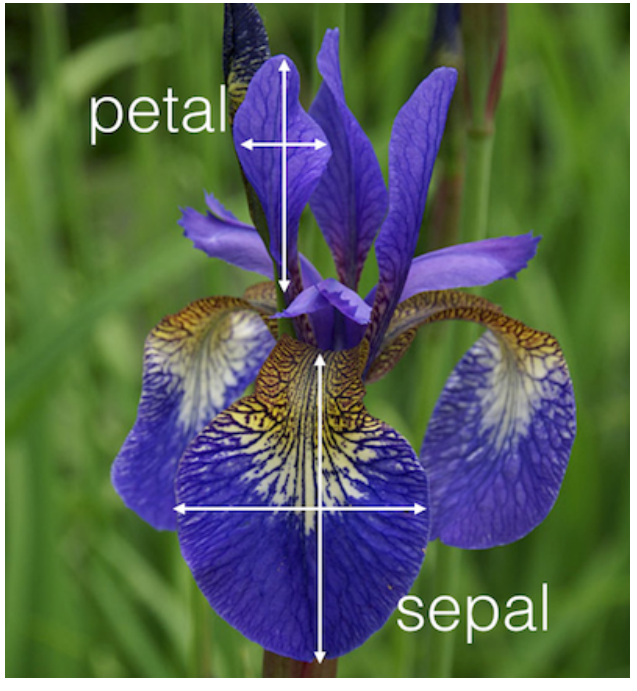
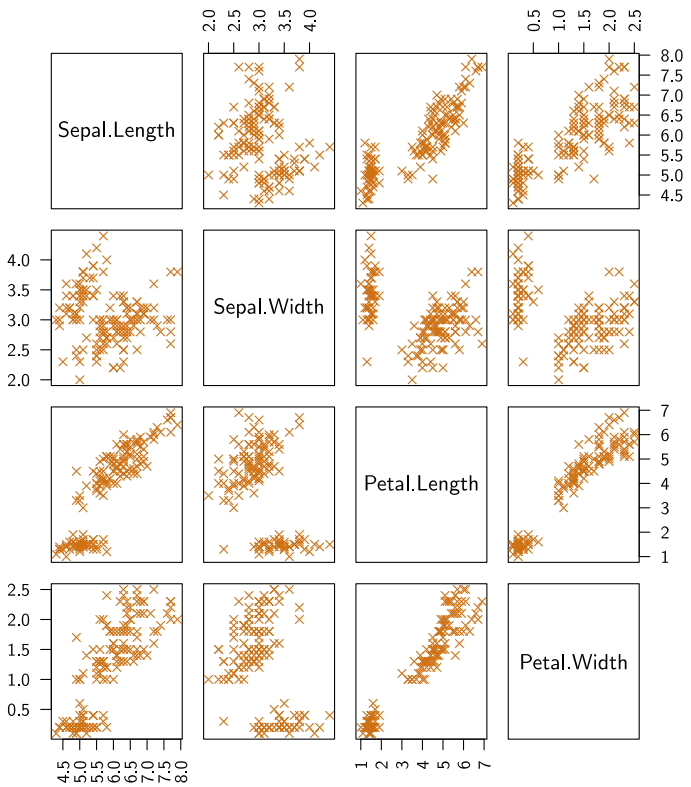


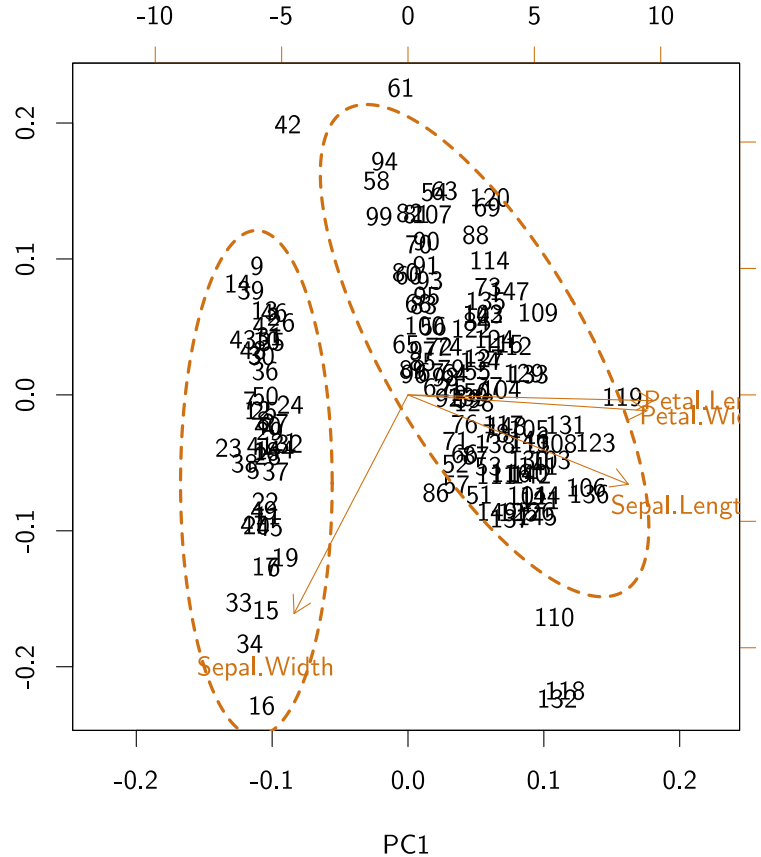
Photo from sebastianraschka.com.

The k-means method in R

```
pairs(iris[, 1:4])
```




```
iris.pca ← prcomp(iris[, 1:4] , scale = TRUE)
biplot(iris.pca)
```



The k-means method in R

The previous plots suggest that setting $k = 2$ may be a good idea.

Input

```
# Show the first 3 rows:
head(iris, n = 3)
```

Output

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1         3.5          1.4         0.2
2          4.9         3.0          1.4         0.2
3          4.7         3.2          1.3         0.2
```

```
# Run the k-means method, and
#   save the outcome in object 'result':
result ← kmeans(iris, centers = 2)
```

```
# See the group assigned to each data point
#   (1 = Group 1, 2 = Group 2):
result$cluster
```

```
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[148] 1 1 1
```

```
# To which group was the 5th flower assigned to?
result$cluster[5]
```

```
[1] 2
```

The k-means method in R

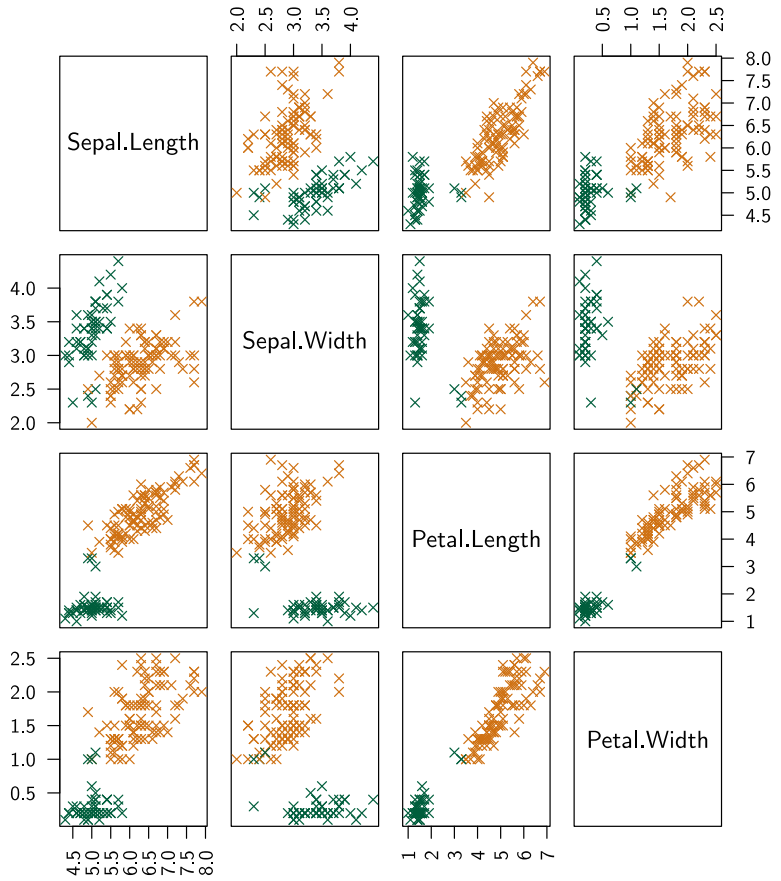
Let's redo the scatterplot pairs plot, with flowers colored by cluster:

```
pairs(iris,
      col = result$cluster
      )

# To use your favorite colors, try:
# pairs(iris,
#       col = c("green", "blue")[result$cluster]
#      )
```

The clustering looks quite OK:

> *There is quite a clear separation between the coloured clusters.*



The k-means method in R

However, the iris data set actually contains three sorts of iris:

Input

```
table(iris$Species)
```

Output

```

    setosa versicolor  virginica 
        50         50         50 
```

Let's redo the cluster analysis, this time using $k = 3$ clusters:

```
# Run the k-means method with 3 clusters,
#   and save the outcome in object 'result.k3':
result.k3 ← kmeans(iris, centers = 3)
```

```
# See the group assigned to each data point
#   (1 = Group 1, 2 = Group 2, 3 = Group 3):
result.k3$cluster
```

```
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[148] 2 2 1
```

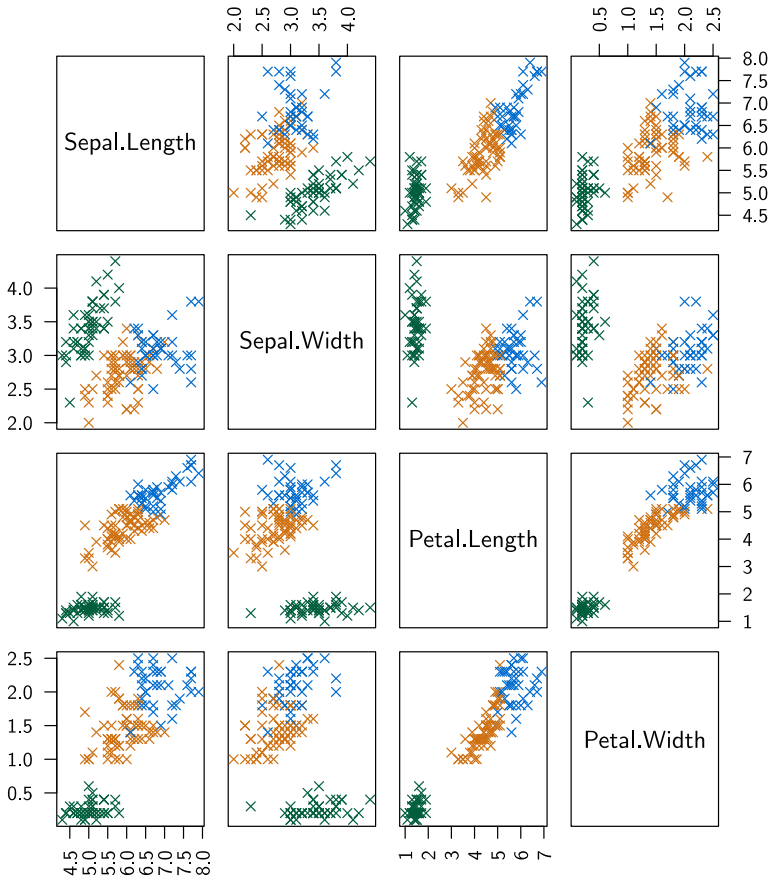
```
# How many flowers were assigned to each group?
table(result.k3$cluster)
```

```

 1  2  3 
62 38 50 
```

The k-means method in R

```
pairs(iris, col = result.k3$cluster)
```



Keep in mind that, when deciding on the number of clusters to use:

1. Looking at scatterplots is important,

but

2. Basing your decision on the purpose of your analysis and available information (e.g., how many flowers species are there) is **more important**.

Exercise (2)

Use the `seiseki` data and divide students into 4 groups by using the k-means method.

Exercise (2) — ANSWER

Use the `seiseki` data and divide students into 4 groups by using the k-means method.

```r
# Import the data:
seiseki ← read.csv("seiseki.csv")
```

```r
colnames(seiseki)
```

```
 [1] "ID"     "kokugo" "shakai" "sugaku" "rika"   "on
```

```r
# Drop unneeded column:
seiseki ← seiseki[, -1]
colnames(seiseki)
```

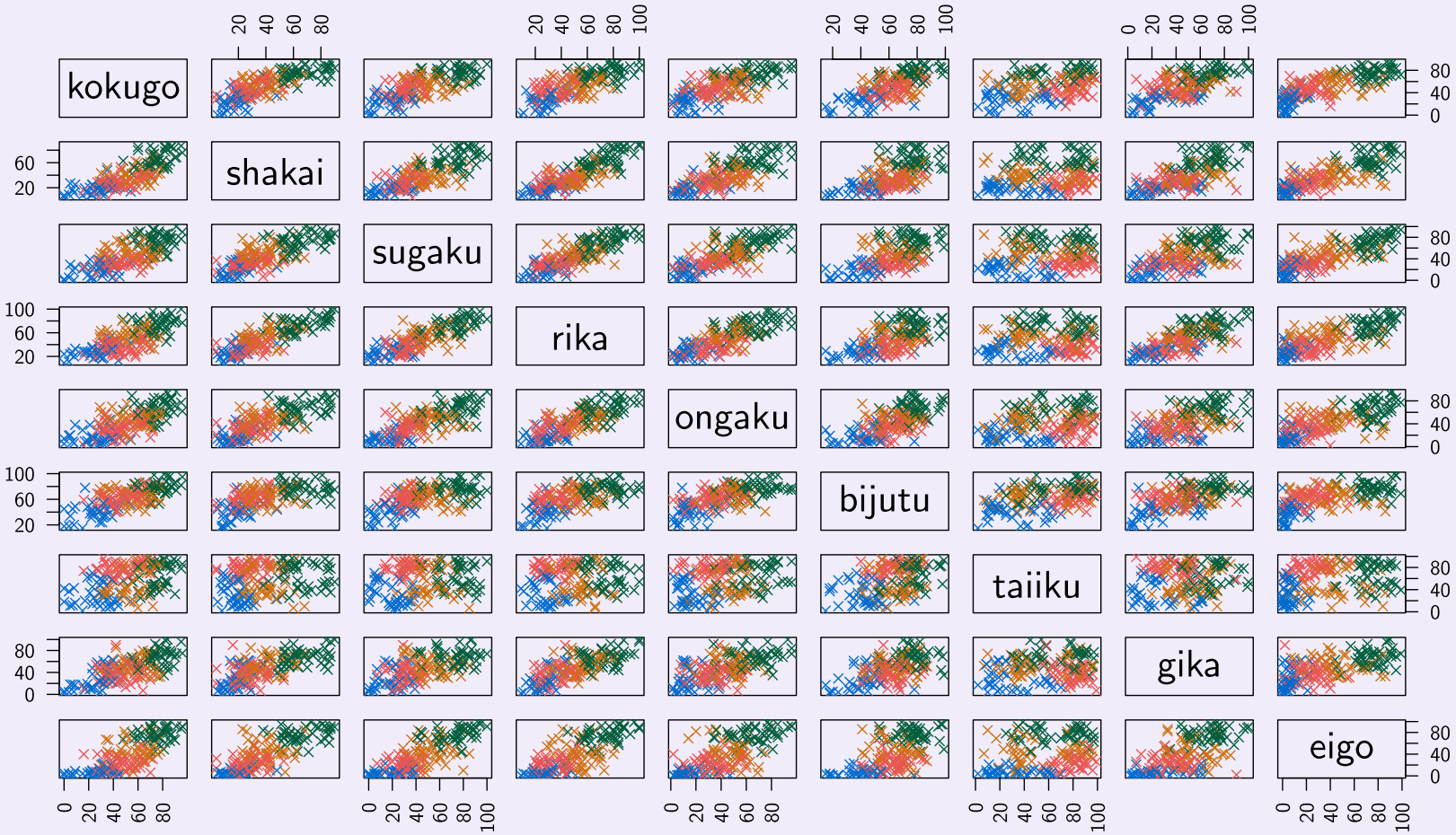
```
[1] "kokugo" "shakai" "sugaku" "rika"   "ongaku" "bij
```

```r
# Run the k-means method with 4 clusters:
result ← kmeans(seiseki, centers = 4)
```

```r
# Create the scatterplot pairs plot:
pairs(seiseki, col = result$cluster)
```

```r
# See next page.
```

Exercise (2) — ANSWER



Summary

Principal component analysis (PCA):

- Summarize multidimensional data into low dimensional data.
- Make a biplot using the 1st and 2nd principal component scores.

PCA allows displaying characteristics of the whole data in one scatter plot!!

Clustering:

- k-means method:
  Define the number of clusters beforehand, and automatically classify data.