STATISTICAL INFERENCE

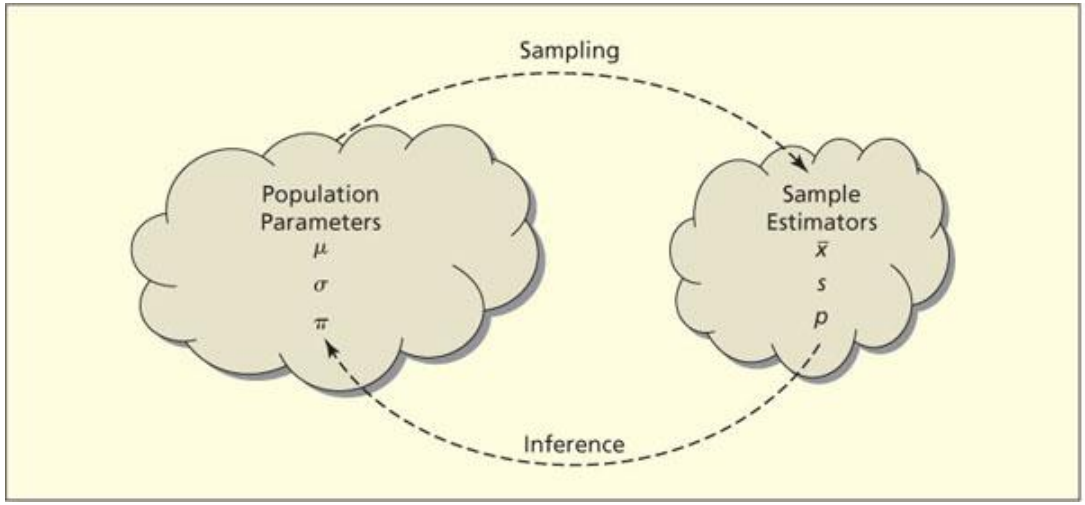


FIGURE 8.3

Sample Estimators of Population Parameters

Main point: How to use the sample to conclude about unknown aspects of the population

Our first topic will be how to summarize the information included in a data collection (what is usually known as descriptive statistics or data exploratory analysis)

Mains points to take into consideration:

- Location
- Variability
- Measures of the possible relationship among the variables in our data collection

Sometimes we have a large number of variables in our data collection and we need to summarize the information underlying a few main points. One of the possible techniques is Principal Components Analysis (PCA).

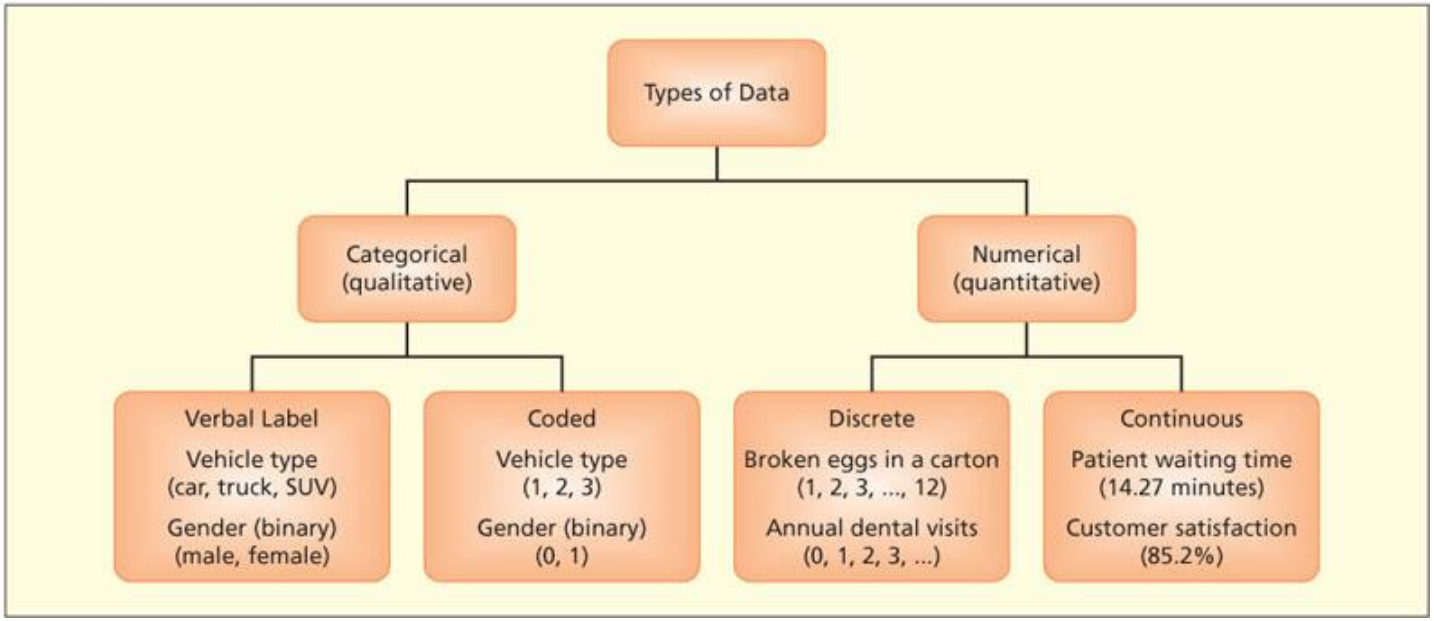
Some points to look at before initializing any analysis:

- Data Types – Categorical *versus* Numerical
- Level of measurement of each variable – Most of the time in actuarial problem we use quantitative variables measured in a ratio scale but … there are exceptions

Data Types – Categorical *versus* Numerical

FIGURE 2.1

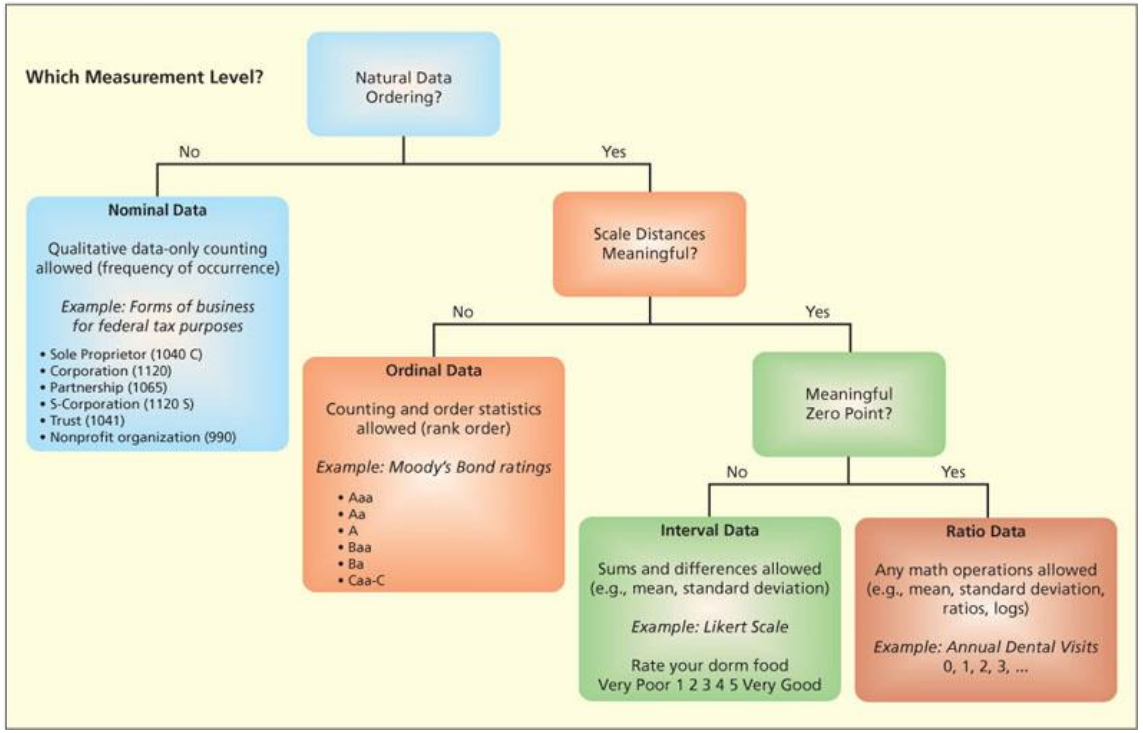
Data Types and Examples



How to determine the level of measurement (Doane and Seward)?

FIGURE 2.3

Determining the Measurement Level



Location and variability measures (counterpart of population measures)

- Mean $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$
- Variance
  - Square of the variations around the mean $s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}$

    Why to divide by $n-1$ instead of n (to be discussed later)?
  - Standard deviation $s = \sqrt{s^2}$
- Covariance $s_{xy} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{n-1}$

  same denominator as the variance
- Correlation coefficient $r_{xy} = \frac{s_{xy}}{s_x \times s_y}$

Association measures (counterpart of population measures)

- Covariance $s_{xy} = \dfrac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$

  same denominator as the variance

- Pearson's correlation coefficient $r_{xy} = \dfrac{s_{xy}}{s_x \times s_y}$

- Spearman´s rank correlation coefficient, r_S

  Replace each value x_i by its rank, $r(x_i)$, and do the same to y_i, obtaining $r(y_i)$. Spearman´s rank correlation coefficient is computed as Pearson´s correlation between $r(x_i)$ and $r(y_i)$.

  When there are no tied ranks, $r_S = 1 - \dfrac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}$

Association and causation

- Association is different from causation
  - **Association** - A relationship between two, or more, variables
  - **Correlation** – Similar to association, depending on how correlation is computed. Pearson´s correlation → linear association
  - **Causation** - Changes in one variable causes changes in the other**.**

Association Measures – Example

Manufacturers of perishable foods often use preservatives to retard spoilage. One concern is that too much preservative will change the flavor of the food. Suppose an experiment is conducted using samples of a food product with varying amounts of preservative added. Both length of time until the food shows signs of spoiling and a taste rating are recorded for each sample. The taste rating is the average rating for three tasters, each of whom rates each sample on a scale from 1 (good) to 5 (bad). Twelve sample measurements are shown in the following table.

	1	2	3	4	5	6	7	8	9	10	11	12
Nº Days	30	47	26	94	67	83	36	77	43	109	56	70
Taste	4.3	3.6	4.5	2.8	3.3	2.7	4.2	3.9	3.6	2.2	3.1	2.9

Compute Pearson's and Spearman's correlation coefficient and comment.

Association Measures – Example (solution)

```
> #### Correlation coefficients example
> x=c(30,47,26,94,67,83,36,77,43,109,56,70)
> y=c(4.3,3.6,4.5,2.8,3.3,2.7,4.2,3.9,3.6,2.2,3.1,2.9)
>
> # Pearson's coefficient
> cor.xy=cor(x,y)    # Pearson's coefficient
>
> avg.x=mean(x); sd.x=sd(x)
> avg.y=mean(y); sd.y=sd(y)
> cbind(avg.x,avg.y,sd.x,sd.y,cor.xy)
     avg.x avg.y     sd.x     sd.y     cor.xy
[1,]  61.5 3.425 26.29034 0.714938 -0.8771227
> # just to check formula
> cov.xy=cov(x,y)
> cov.xy/(sd.x*sd.y) # Pearson's coefficient
[1] -0.8771227
>
```

Association Measures – Example (solution)

```
> # Spearman's coefficient
> cor(rank(x),rank(y)) # Spearman's coefficient
[1] -0.8791607
> rank(x); rank(y)
 [1]  2  5  1 11  7 10  3  9  4 12  6  8
 [1] 11.0  7.5 12.0  3.0  6.0  2.0 10.0  9.0  7.5  1.0  5.0  4.0
> d=rank(x)-rank(y); n=12; 1-6*sum(d^2)/(n*(n^2-1)) # Approx value
since we have one tie
[1] -0.8758741
>
```

PRINCIPAL COMPONENTE ANALYSIS (PCA)

Motivation: A financial analyst is interested in determining the financial health of firms in a given industry. Research studies have identified a number of financial ratios (say about 120) that can be used for such a purpose. Obviously, it would be extremely taxing to interpret the 120 pieces of information for assessing the financial health of firms. However, the analyst's task would be simplified if these 120 ratios could be reduced to a few indices (say about 3), which are linear combinations of the original 120 ratios.

Main purpose of PCA: To capture the main patterns explaining the variability in a data set using a small number of new variables that are uncorrelated linar combinations of the original variables keeping the loss of information under control.

PCA – Introduction

- PCA is one of the multivariate exploratory data analysis techniques. It can be used by itself to reduce the dimension of a data set or as an auxiliary technique for other approaches.
- The new variables to be created are:
  - Linear combinations of the original variables
  - The linear combinations are uncorrelated with each other
  - The maximum number of new variables is equal to the number of original variables (assuming that there is no perfect correlation among the original variables)
- Let us first consider a very simple example: $p = 2$ variables, X_1 and X_2 and $n = 12$ observations for each variable (data are presented in Table 1)

PCA – A simple example

Table1

obs	1	2	3	4	5	6	7	8	9	10	11	12
X_1	16	12	13	11	10	9	8	7	5	3	2	0
X_2	8	10	6	2	8	-1	4	6	-3	-1	-3	0

Compute S, the covariance matrix between X_1 and X_2.

$$S = \begin{bmatrix} s_1^2 = s_{11} & s_{12} \\ s_{21} & s_2^2 = s_{22} \end{bmatrix} \approx \begin{bmatrix} 23.09091 & 16.45455 \\ 16.45455 & 21.09091 \end{bmatrix}$$

using $s_{ij} = \dfrac{\sum_{k=1}^{n}(x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{n-1}$ and $\bar{x}_i = \dfrac{\sum_{k=1}^{n} x_{ik}}{n}$

- The total variance is then

  $s_1^2 + s_2^2 = s_{11} + s_{22} = 23.09091 + 21.09091 = 44.18182$

  $s_1^2 = 23.09091$ (the variance of the first variable) represents 52.26% of this total while $s_2^2 = 21.09091$ represents 47.74%.

PCA – A simple example (cont)

- The first step is to replace the original variables, X_1 and X_2, by 2 linear combinations of them, Y_1 and Y_2, in such a way that the first one will cover must of the varability (most of the total variance).

$$\begin{cases} Y_1 = e_{11} X_1 + e_{12} X_2 \\ Y_2 = e_{21} X_1 + e_{22} X_2 \end{cases} \text{ and } \begin{cases} s_{Y_1}^2 = e_{11}^2 s_{11} + e_{12}^2 s_{22} + 2 e_{11} e_{12} s_{12} \\ s_{Y_2}^2 = e_{21}^2 s_{11} + e_{22}^2 s_{22} + 2 e_{21} e_{22} s_{12} \end{cases}$$

  As it is obvious, if we multiply the coefficient e_{ij} by a constant $k > 1$, $s_{Y_1}^2$ and $s_{Y_2}^2$ will increase and then we need to introduce a constraint before maximizing. The constraint is $e_{i1}^2 + e_{i2}^2 = 1$.

- The second step will be to discuss the data reduction: Is it acceptable to use less linear combinations (Y variables) than the original number of variables?

Let us, first, discuss the first step.

PCA – A simple example (cont)

The problem: $max\, s_{Y_1}^2$ subject to $e_{i1}^2 + e_{i2}^2 = 1.$

We need to maximize the Lagrangean function

$L = e_{11}^2 s_{11} + e_{12}^2 s_{22} + 2 e_{11} e_{12} s_{12} - \lambda(e_{11}^2 + e_{12}^2 - 1)$ in order to e_{11}, e_{12} and λ

$$\begin{cases} \dfrac{\partial L}{\partial e_{11}} = 2 e_{11} s_{11} + 2 e_{12} s_{12} - 2\lambda e_{11} \\ \dfrac{\partial L}{\partial e_{12}} = 2 e_{12} s_{22} + 2 e_{11} s_{12} - 2\lambda e_{12} \\ \dfrac{\partial L}{\partial \lambda} = -(e_{11}^2 + e_{12}^2 - 1) \end{cases}$$

and then we must solve the system $\begin{cases} e_{11} s_{11} + e_{12} s_{12} - \lambda e_{11} = 0 \\ e_{12} s_{22} + e_{11} s_{12} - \lambda e_{12} = 0 \\ -(e_{11}^2 + e_{12}^2 - 1) = 0 \end{cases}$

PCA – A simple example (cont)

Using some matricial notation and defining $\mathbf{e}_1 = \begin{bmatrix} e_{11} \\ e_{12} \end{bmatrix}$, $\mathbf{0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, I as the identity matrix (size 2) and $S = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}$ we can rewrite the problem as

$$L = \mathbf{e}_1^T S \mathbf{e}_1 - \lambda\left(\mathbf{e}_1^T I \mathbf{e}_1 - 1\right) = \mathbf{e}_1^T S \mathbf{e}_1 - \lambda\left(\mathbf{e}_1^T \mathbf{e}_1 - 1\right)$$

$$\frac{\partial L}{\partial \mathbf{e}_1} = 2S\mathbf{e}_1 - 2\lambda I \mathbf{e}_1 = 2(S - \lambda I)\mathbf{e}_1$$

$$\frac{\partial L}{\partial \lambda} = -\lambda\left(\mathbf{e}_1^T \mathbf{e}_1 - 1\right)$$

And then we must solve $(S - \lambda I)\mathbf{e}_1 = \mathbf{0}$ knowing that $\mathbf{e}_1^T \mathbf{e}_1 = 1$ (the same result can be obtained from the previous slide). This is a well known problem in linear algebra: Finding the eigenvalues and the eigen vectors of matrix S

PCA – Eigenvalues and eigenvector of matrix S

Solve $(S-\lambda I)\mathbf{e}_1 = \mathbf{0}$ subject to $\mathbf{e}_1^T\mathbf{e}_1 = 1$

As we have a homogeneous system of equations the trivial solution $\mathbf{e}_1 = \mathbf{0}$ is always possible but irrelevant. So, we must guarantee that the determinant of the system is 0, i.e., $|S-\lambda I| = 0$.

This equation is a polynomial of order k (the number of original variables) and therefore has k roots (the eigenvalues of S), $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k > 0$, as S is definite positive matrix (assuming that none of the variables is a linear combination of the others).

For each root λ_i we get the corresponding eigenvector, $\mathbf{e}_i$, normalized using $\mathbf{e}_i^T\mathbf{e}_i = 1$.

PCA – Eigenvalues and eigenvector of matrix S (cont)

Let us consider the largest eigenvalue, λ_1. As it is a solution of the system we have

$$\begin{cases} (S - \lambda_1 I)\mathbf{e}_1 = \mathbf{0} \\ \mathbf{e}_1^T \mathbf{e}_1 = 1 \end{cases}$$

Pre-multiplying the first equation by $\mathbf{e}_1^T$ originates

$$\mathbf{e}_1^T (S - \lambda_1 I)\mathbf{e}_1 = \mathbf{e}_1^T \mathbf{0} \Leftrightarrow \mathbf{e}_1^T S\,\mathbf{e}_1 - \mathbf{e}_1^T \lambda_1 I \mathbf{e}_1 = 0 \Leftrightarrow \mathbf{e}_1^T S\,\mathbf{e}_1 = \lambda_1 \mathbf{e}_1^T \mathbf{e}_1 = \lambda_1$$

as $\mathbf{e}_1^T \mathbf{e}_1 = 1$.

We get $\mathbf{e}_1^T S\,\mathbf{e}_1 = \lambda_1$ and if we repeat the process for the second largest eigenvalue we get $\mathbf{e}_2^T S\,\mathbf{e}_2 = \lambda_2$ and so on. The variance of the *j-th* linear combination is equal to the *j-th* eigenvalue.

PCA – A simple example (cont)

Back to our simple example we define the polynomial as

$$\begin{vmatrix} 23.09091-\lambda & 16.45455 \\ 16.45455 & 21.09091-\lambda \end{vmatrix} = 0 \Leftrightarrow (23.09091-\lambda)(21.09091-\lambda) - 16.45455^2 = 0$$

$$\Leftrightarrow \lambda^2 - 44.18182\,\lambda + 216.2561 = 0$$

$$\lambda = \frac{44.182 \pm \sqrt{44.182^2 - 4 \times 216.245}}{2}, \text{ i.e., } \lambda_1 = 38.57582 \text{ and } \lambda_2 = 5.606001.$$

The corresponding eigenvectors are given by

$$\begin{cases} (23.09091-\lambda_i)\,e_{i1} + 16.45455\,e_{i2} = 0 \\ e_{i1}^2 + e_{i2}^2 = 1 \end{cases} \Leftrightarrow \begin{cases} e_{i1}^2 = \dfrac{1}{1+\left(\dfrac{23.09091-\lambda_i}{16.45455}\right)^2} \\ e_{i2} = \left(\dfrac{23.09091-\lambda_i}{16.45455}\right) e_{i1} \end{cases}$$

PCA – A simple example (cont)

For each case (see slide we can choose the positive or the negative root to define e_{i1} as the results are equivalent. Let us choose the negative (to get the solution obtained using R)

$$\lambda_1 = 38.57582 \quad (87.31\% \text{ of total variance}) \quad \mathbf{e}_1^T = \begin{bmatrix} -0.722388 & -0.685324 \end{bmatrix}$$

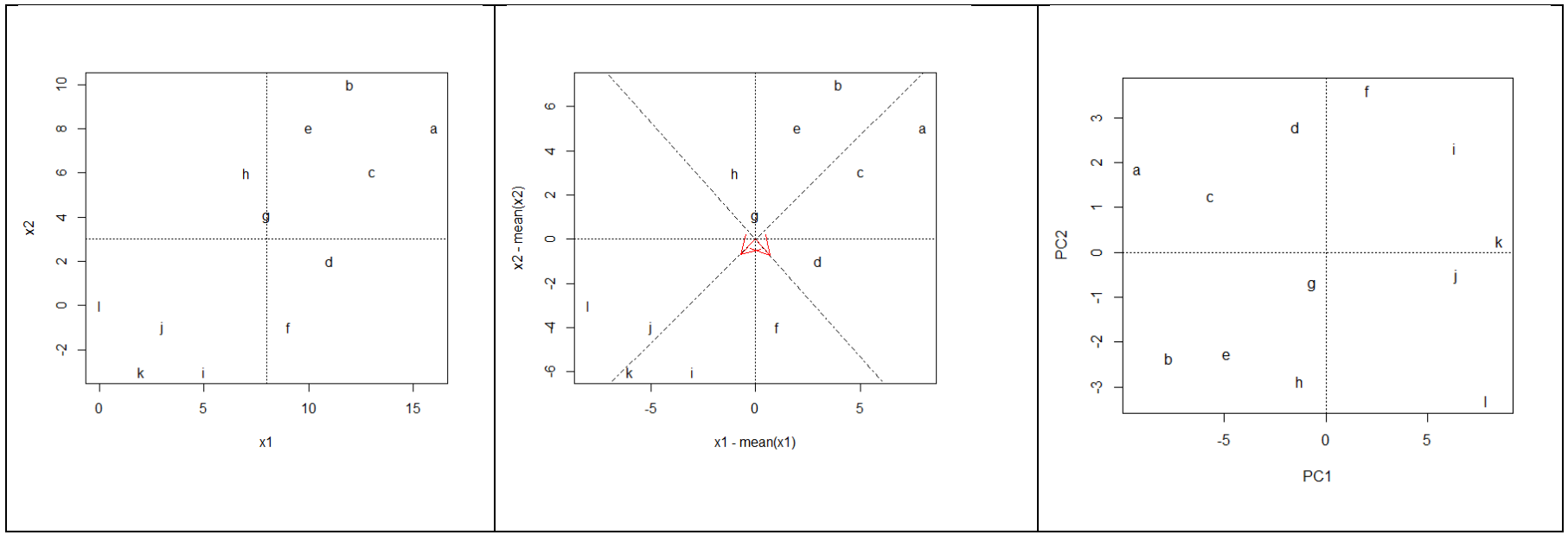
$$\lambda_2 = 5.606001 \quad (12.69\% \text{ of total variance}) \quad \mathbf{e}_2^T = \begin{bmatrix} -0.685324 & 0.728238 \end{bmatrix}$$

At this stage we are using all the information in a different way and we solve step 1.

Instead of representing (plotting) each observation using the original variables (X_1 and X_2) we can use the principal components ($\mathbf{e}_1$ and $\mathbf{e}_2$). For observation i we get

$$\begin{bmatrix} Y_{1i} \\ Y_{2i} \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} X_{1i} \\ X_{2i} \end{bmatrix} = \begin{bmatrix} -0.722388 & 0.685324 \\ -0.685324 & 0.728238 \end{bmatrix} \begin{bmatrix} X_{1i} \\ X_{2i} \end{bmatrix}$$

PCA – A simple example (cont)



- 1st panel: observations (a to l), centering effect (dashed lines) and PC (red)
- 2nd panel: centered observations (a to l), and PC (red) with unit and direction. The direction of each axis is arbitrary (remember that we can choose the negative or the positive root).
- 3rd panel: observations using the new axes system (PC).

PCA

Two more definitions can be useful:

- **Loadings**: different definitions appear in the litterature but the most commom – also called standardized loading – is the correlation coefficient between each PC and each original variable $l_{ij} = \frac{w_{ij}}{s_j}\sqrt{\lambda_i}$ where l_{ij} stand for the loading of the jth variable on PC_i, w_{ij} is the weight , s_j the standard deviation of the *j*-th variable and λ_i the eigenvalue (variance) associated to the *i*-th PC.

- **Scores**: the coordinates of each observation in terms of the principal components (see the right panel on previous slide)

Data reduction: Is the first PC (or how many PC are) enough to represent the data set? The answer depends on:

  - How much variability (% of total variability) is captured by the first PCs.
  - How relevant is the loss of information for the problem under analysis.

Before addressing this point let's see how to use R to analyze the first example:

PCA – Using R – A simple example

```
> x1=c(16,12,13,11,10,9,8,7,5,3,2,0)
> x2=c(8,10,6,2,8,-1,4,6,-3,-1,-3,0)
> x=cbind(x1,x2)
>
> ### Using eigenvalues and eigenvectors - centered only
> x1c=(x1-mean(x1)); x2c=(x2-mean(x2))
> Xc=cbind(x1c,x2c)
> S=(1/(length(x1c)-1))*(t(Xc)%*%Xc); S  # covariance matrix
         x1c      x2c
x1c 23.09091 16.45455
x2c 16.45455 21.09091
> out=eigen(S); out
eigen() decomposition
$`values`
[1] 38.575813  5.606005

$vectors
           [,1]       [,2]
[1,] -0.7282381  0.6853242
[2,] -0.6853242 -0.7282381
```

PCA – Using R – A simple example

```
> cbind(out$values[1]/sum(out$values),out$values[2]/sum(out$values))
          [,1]      [,2]
[1,] 0.8731151 0.1268849
>
> ### Using prcomp function (other sol are available in R)
> out1=prcomp(x,center=T)
> out1 # eigenvalues are the squares of st. dev.
Standard deviations (1, .., p=2):
[1] 6.210943 2.367700

Rotation (n x k) = (2 x 2):
          PC1        PC2
x1 -0.7282381  0.6853242
x2 -0.6853242 -0.7282381
> summary(out1)
Importance of components:
                          PC1    PC2
Standard deviation     6.2109 2.3677
Proportion of Variance 0.8731 0.1269
Cumulative Proportion  0.8731 1.0000
```

PCA – Using R – A simple example

```
> out1$x           # scores
              PC1         PC2
 [1,] -9.2525259  1.8414027
 [2,] -7.7102217 -2.3563703
 [3,] -5.6971632  1.2419065
 [4,] -1.4993902  2.7842106
 [5,] -4.8830971 -2.2705423
 [6,]  2.0130586  3.5982767
 [7,] -0.6853242 -0.7282381
 [8,] -1.3277344 -2.8700386
 [9,]  6.2966594  2.3134563
[10,]  6.3824874 -0.5136683
[11,]  8.4813738  0.2574838
[12,]  7.8818776 -3.2978790
> out1$rotation   # weights, eigen vectors
          PC1         PC2
x1 -0.7282381  0.6853242
x2 -0.6853242 -0.7282381
```

PCA – Using R – A simple example

```
> cor(x,out1$x)    # loadings
           PC1         PC2
x1 -0.9412618  0.3376776
x2 -0.9268425 -0.3754503
```

or, using the formula, $l_{ij} = \frac{w_{ij}}{s_j}\sqrt{\lambda_i}$

```
> # using the formula instead of the correlation
> z.c=rbind(out1$sdev,out1$sdev)
> sd.c=c(sd(x1),sd(x2)); sd.c=cbind(sd.c,sd.c)
> l=out1$rotation*z.c/sd.c; l   # loadings
           PC1         PC2
x1 -0.9412618  0.3376776
x2 -0.9268425 -0.3754503
```

PCA – Some issues

4 issues need to be briefly discussed:

1. In addition to centering (mean correct) should we scale the variables?
2. Number of principal components to extract
3. How to interpret principal components
4. Use of principal component scores

A new example – Food price index (Sharma) – will help to clarify these issues.

The average price (cents per pound – 1973) of five (just to keep things simple) food items are known for 23 US cities. Our main objective is to form a price index (like the Consumer Price Index) using PCA.

The data is presented in the R program: 5 food items and 23 cities

After reading the data set, our next task is to prerform a PCA as we did before

PCA - Example 2 – Food price index

- First step ⟶ reading the data set

```
> dta=read.csv("E:/Risk Models 2018/food price index.csv",header=T,sep=",")
> dta     # Check input
            City Bread Burger Milk Oranges Tomatoes
1        Atlanta  24.5   94.5 73.9    80.1     41.6
2      Baltimore  26.5   91.0 67.5    74.6     53.3
3         Boston  29.7  100.8 61.4   104.0     59.6
4        Buffalo  22.8   86.6 65.3   118.4     51.2
5        Chicago  26.7   86.7 62.7   105.9     51.2
6     Cincinnati  25.3  102.5 63.3    99.3     45.6
7      Cleveland  22.8   88.8 52.4   110.9     46.8
8         Dallas  23.3   85.5 62.5   117.9     41.8
9        Detroit  24.1   93.7 51.5   109.7     52.4
10      Honolulu  29.3  105.9 80.2   133.2     61.7
11       Houston  22.3   83.6 67.8   108.6     42.4
12   Kansas City  26.1   88.9 65.4   100.9     43.2
13   Los Angeles  26.9   89.3 56.2    82.7     38.4
14     Milwaukee  20.3   89.6 53.8   111.8     53.9
15   Minneapolis  24.6   92.2 51.9   106.0     50.7
16      New York  30.8  110.7 66.0   107.3     62.6
17  Philadelphia  24.5   92.3 66.7    98.0     61.7
18    Pittsburgh  26.2   95.4 60.2   117.1     49.3
19     St. Louis  26.5   92.4 60.8   115.1     46.2
20     San Diego  25.5   83.7 57.0    92.8     35.4
21 San Francisco  26.3   87.1 58.3   101.8     41.5
22       Seattle  22.5   77.7 62.0    91.1     44.9
23 Washington DC  24.2   93.8 66.0    81.6     46.2
```

PCA - Example 2 – Food price index

- Second step → 1st PCA using prcomp function (other solutions are available)

```
> attach(dta)
> x=cbind(Bread,Burger,Milk,Oranges,Tomatoes)
> out1=prcomp(x, center=T)
> out1
Standard deviations:
[1] 14.798604  9.577221  6.136994  4.561857  1.740468

Rotation:
                PC1        PC2         PC3         PC4         PC5
Bread    0.02848905  0.1653211 -0.02135748  0.18972574 -0.96716354
Burger   0.20012240  0.6321849 -0.25420475  0.65862454  0.24877074
Milk     0.04167230  0.4421503  0.88874949 -0.10765906  0.03606094
Oranges  0.93885906 -0.3143547  0.12135003  0.06904699 -0.01521357
Tomatoes 0.27558389  0.5279160 -0.36100184 -0.71684022 -0.03429221
 > summary(out1)
Importance of components:
                           PC1    PC2    PC3     PC4     PC5
Standard deviation     14.7986 9.5772 6.1370 4.56186 1.74047
Proportion of Variance  0.5884 0.2464 0.1012 0.05591 0.00814
Cumulative Proportion   0.5884 0.8348 0.9359 0.99186 1.00000
```

PCA - Scaling or not scaling the variables

As we can see PC1 is very much affected by the variable Oranges. This is partialy due to the fact that the variability associated with this variable is much higher than the variability associated with the other variables (see the satndard deviations).

If we do not want that the variability of each variable influences the output we can scale the variables (divide by the standard deviation) – we will analyze the correlation matrix instead of the covariance matrix.

The main point to think about before scaling or not scaling the variables is if we want to give the same (scale the variables) a priori weight to each variable or not.

To use scaled variables we can scale them before performing PCA or jus just replace `out1=prcomp(x,center=T)` by `out1=prcomp(x,center=T,scale=T)`

PCA - Example 2 – Food price index

```
> out2=prcomp(x, center=T,scale=T)
> out2
Standard deviations:
[1] 1.5564279 1.0510352 0.8593489 0.7025748 0.4906784

Rotation:
                PC1         PC2         PC3         PC4          PC5
Bread    0.4961487 -0.30861972  0.38639398  0.50930459 -0.499898868
Burger   0.5757023 -0.04380176  0.26247227 -0.02813712  0.772635014
Milk     0.3395696 -0.43080905 -0.83463952  0.04910000  0.007882237
Oranges  0.2249898  0.79677694 -0.29160659  0.47901574 -0.005966796
Tomatoes 0.5064340  0.28702846  0.01226602 -0.71270629 -0.391201387
> summary(out2)
Importance of components:
                          PC1    PC2    PC3     PC4     PC5
Standard deviation     1.5564 1.0510 0.8593 0.70257 0.49068
Proportion of Variance 0.4845 0.2209 0.1477 0.09872 0.04815
Cumulative Proportion  0.4845 0.7054 0.8531 0.95185 1.00000
```

As we can see we get a different solution. Now, the weights for PC1 are more balanced. For the purpose of the example (CPI) this solution is probably better since we have no reason to give Oranges much more weight than for the other items.

PCA - Number of PC to extract

Remember that the idea is to capture the **main patterns** explaining the variability in a data set using a **small number** of PC. Both topics ("main pattern" and "small number") are linked together and depend on the problem under analysis.

However there are some criteria that can be used when there is no clear answer to this question.

- **Kaiser criterion** – keep PC whose eigenvalues are greater than 1 (scaled data) or greater than the average of all eigenvalues (non-scaled data). Mainly used with scaled data.
- **Scree-plot analysis** – Plot the percent of variance accounted for by each PC and look for an elbow. Choose the value imidiately before the elbow (used with both scaled and non-scaled data) or use the second differences.
- **Parallel analysis** – Based on a simulation procedure (simulation will be discussed latter) that can be simplified using a table of constants (see sharma). More efficient but less used as it is more difficult to compute. In R we can use some packages to get a parallel analysis: paran or psych for instance.

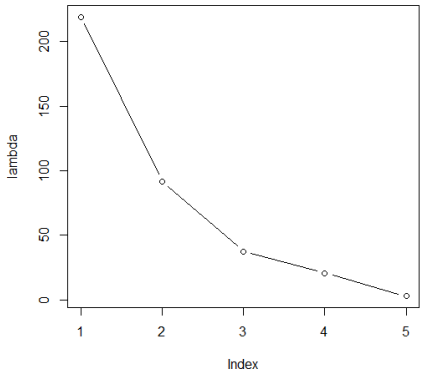
PCA - Example 2 – Food price index

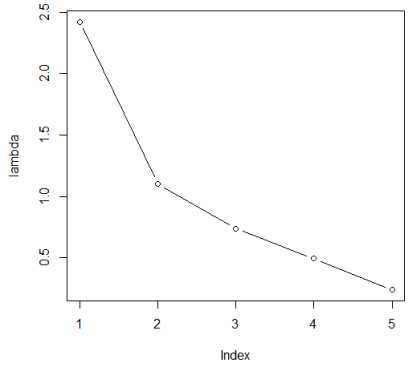
- Kaiser criterion:
  - Scaled data: retain the first 2 PC (remember that the eigenvalues are the square of the standard deviation of the principal components
  - Non-scaled data: retain the first 2 PC

```
> lambda=out1$sdev^2; lambda
[1] 218.998679  91.723169  37.662690  20.810541   3.029229
> mean(lambda)
[1] 74.44486
```

PCA - Example 2 – Food price index

- Scree plot: left panel for centered data and right panel for scaled (and centered) data





For non-scaled dat just replace out2 by out1

```
> # scaled
> lambda=out2$sdev^2
> plot(lambda,type="b")
> diff(lambda,lag=1,differences=2)
[1]  0.951598697  0.121325148 -0.007976802
```

Both scree plots are similar and recommend the use of 2 PC

PCA - Example 2 – Food price index

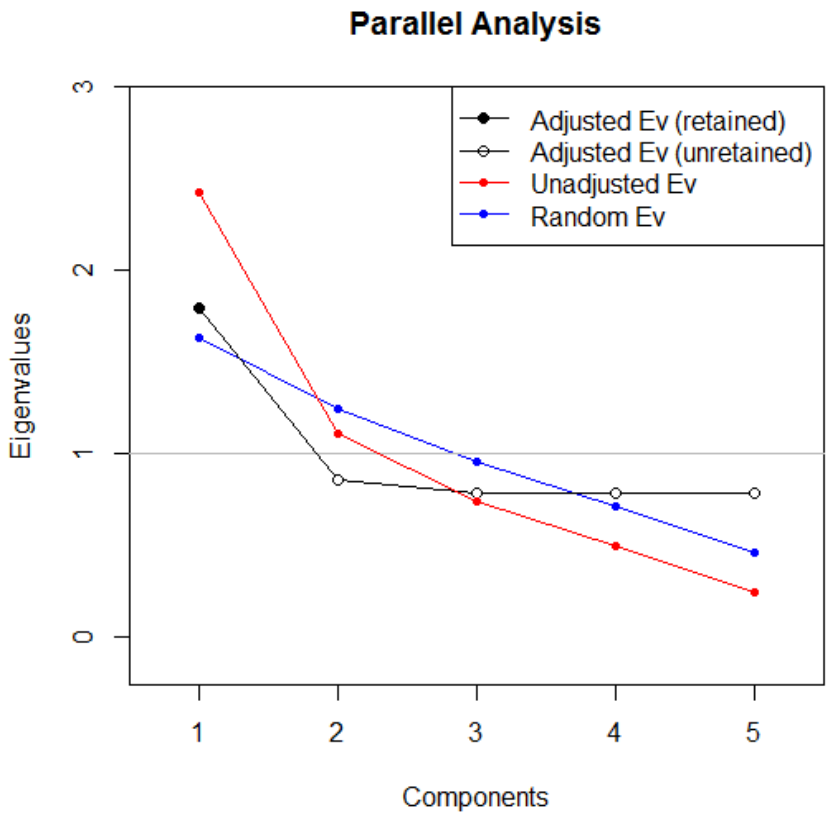
- Parallel analysis – Horn (1965)
  - Using *paran*

```
> require(paran)      # package paran must be installed before
> paran(x,iterations=100,graph=T)
Using eigendecomposition of correlation matrix.
Computing: 10%  20%  30%  40%  50%  60%  70%  80%  90%  100%
Results of Horn's Parallel Analysis for component retention
100 iterations, using the mean estimate

-------------------------------------------------
Component   Adjusted    Unadjusted    Estimated
            Eigenvalue  Eigenvalue    Bias
-------------------------------------------------
1           1.791788    2.422467      0.630679
-------------------------------------------------

Adjusted eigenvalues > 1 indicate dimensions to retain.
(1components retained)
```

PCA - Example 2 – Food price index



PCA - How to interpret principal components?

When possible retained PC can be interpreted using the loadings: The higher the loading (in absolute value) the more influence it had in the formation of the PC. But the main question is how high should the loading be before we can say that a given variable is influential in the formation of he PC. There are no clear answers to this question. In some applied work the value 0.5 or 0.6 for scaled data is used as a cutoff.

PCA - Example 2 – Food price index

Back to the example, compute the loadings for each retained PC - assuming scaled variables and that 2 PC are retained

```
> cor(x,out2$x)     # Loadings

                PC1          PC2
Bread     0.7722197 -0.32437017
Burger    0.8960392 -0.04603719
Milk      0.5285156 -0.45279546
Oranges   0.3501804  0.83744058
Tomatoes  0.7882281  0.30167700
```

Using 0.5 as the cutoff value, PC1 is the “non-fruits” CPI (stricly speaking tomatoe is a fruit but is usually considered as a vegetable) and PC2 is the “fruit” CPI we can interpret PC1.

PCA - Example 2 – Food price index

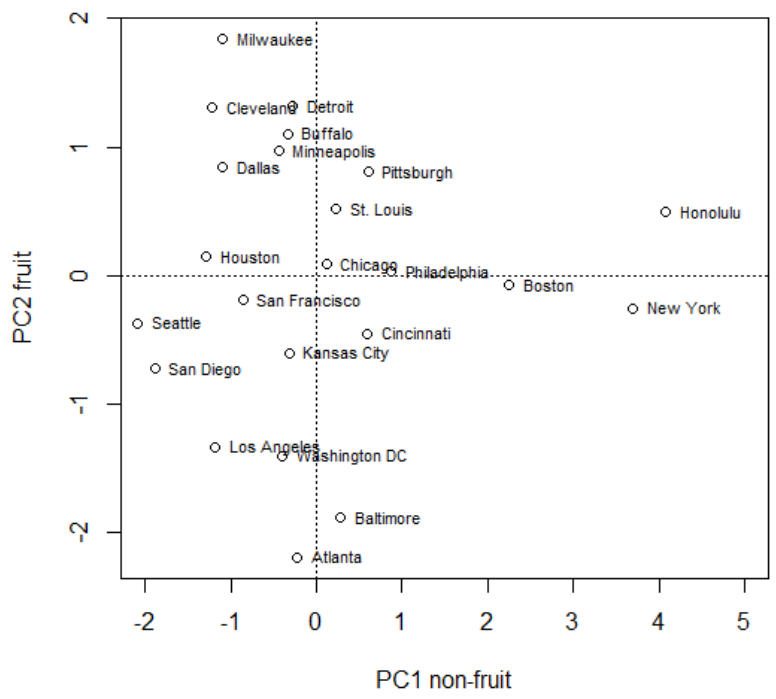
If **non-scaled** data is used, there interprteation is more puzzling

```
> cor(x,out1$x)
                 PC1         PC2
Bread     0.16817616  0.6315875
Burger    0.39199944  0.8014061
Milk      0.08872954  0.6092695
Oranges   0.97573973 -0.2114328
Tomatoes  0.53642447  0.6650256
```

PC1 linked to fruits (and vegetables) and PC2 linked to the remaining items.

PCA – Use of principal component scores

PC scores can be plotted for further interpretting the results (but a clear interpretation is not guaranted).



Broadly speaking we can identify 5 groups:

- High fruit CPI and low non-fruit CPI: Milwaukee, Cleveland, Detroit, Buffalo, Minneapolis,Dallas;
- Avg fruit CPI and low non-fruit CPI : Houston, San Fr, Seattle, San Diego Kansas City;
- Avg fruit CPI and low non-fruit CPI : LA, Wash, Baltimore, Atlanta;
- Avg fruit CPI and avg non-fruit CPI : Pittsburgh, St Louis, Chicago, Philadelphia, Houston, Cincinanti;
- Avg fruit CPI and High non-fruit CPI : New-York, Honolulu,