



iscte

INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA



emprego  
digital



# Periodic functions

- Amplitude
- Period
- Phase shift

$$y = A \sin[B(x - C)] + D$$

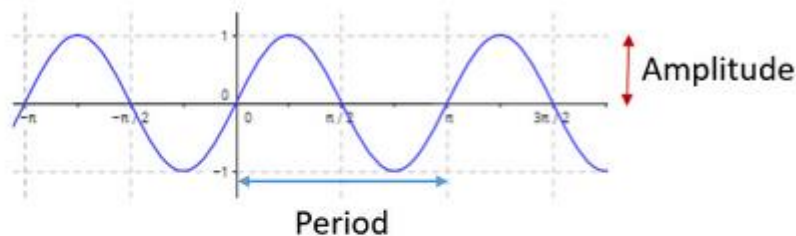
$|A|$  is the amplitude

The period is  $\frac{2\pi}{B}$

Phase (horizontal) shift is  $C$

Vertical shift is  $D$

*The same applies for the Cosine Function.*



$$y = 4 + \frac{1}{2} \sin(x)$$

$$y = -4 \cos(3x - \pi)$$

## Periodic functions – exercises spyder

$$y = 4 + \frac{1}{2} \sin(x)$$

Radians

degrees

samples

$$y = -4 \cos(3x - \pi)$$

What is the period of these functions?

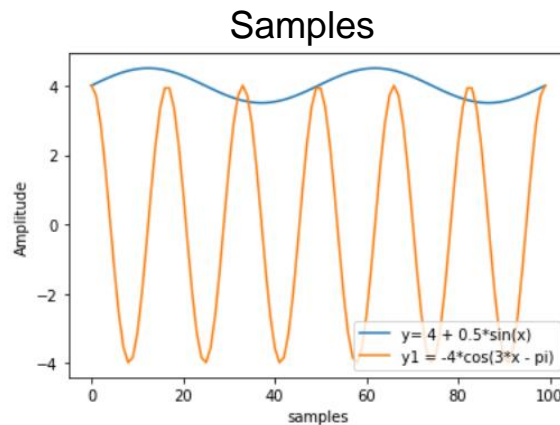
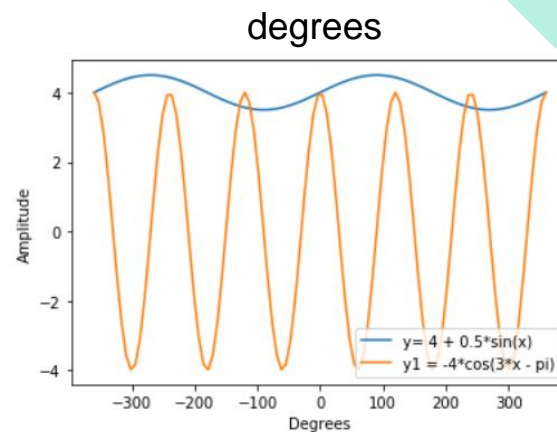
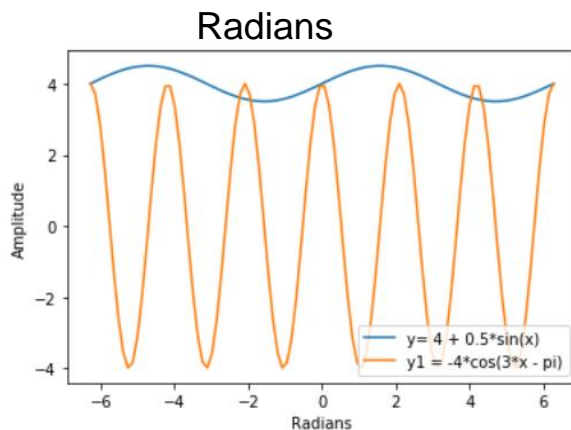
What is the amplitude?

How many samples (minimum amount ) is needed to see the smoothing charts ?

# Periodic functions – exercises spyder

$$y = 4 + \frac{1}{2} \sin(x)$$

$$y = -4 \cos(3x - \pi)$$



```
class BigFile:
```

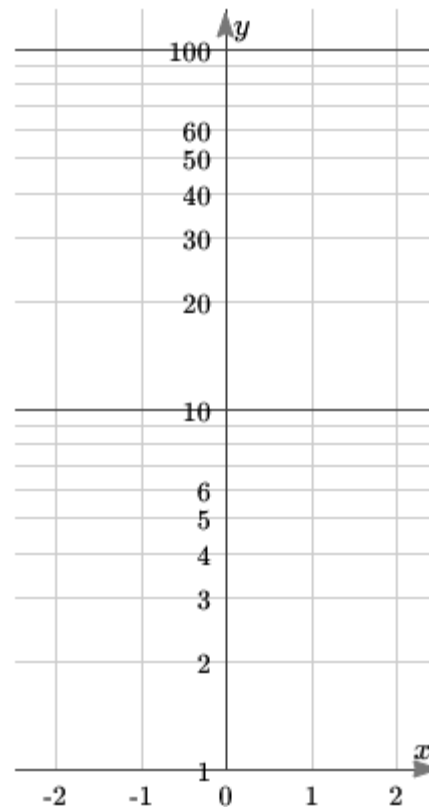
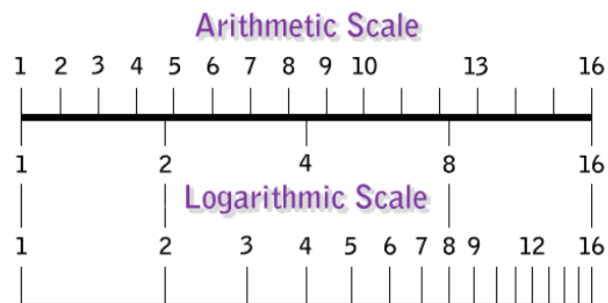
```
    def __init__(self, datadir, ndims):
        idfile = os.path.join(datadir, "id.txt")
        self.names = [x.strip() for x in str.split(open(idfile).read()) if x.strip()]
        self.name2index = dict(zip(self.names, range(len(self.names))))
        self.ndims = ndims
        self.featurefile = os.path.join(datadir, "feature.bin")
        print "[BigFile] %d features, %d dimensions" % (len(self.names), self.ndims)
        print "        binary: %s" % self.featurefile
        print "        txt: %s" % idfile
```

```
    def __getitem__(self, requested, isname=True):
        if isname:
            index_name_array = [self.names[self.name2index[x], x] for x in requested if x in self.names]
        else:
            assert(min(requested) >= 0)
            assert(max(requested) < len(self.names))
            index_name_array = [(x, self.names[x]) for x in requested]
            index_name_array.sort()
            vecs = seq_read(self.featurefile, self.ndims, [x[0] for x in index_name_array])
            return [x[1] for x in index_name_array], vecs

    def shape(self):
        return [len(self.names), self.ndims]
```

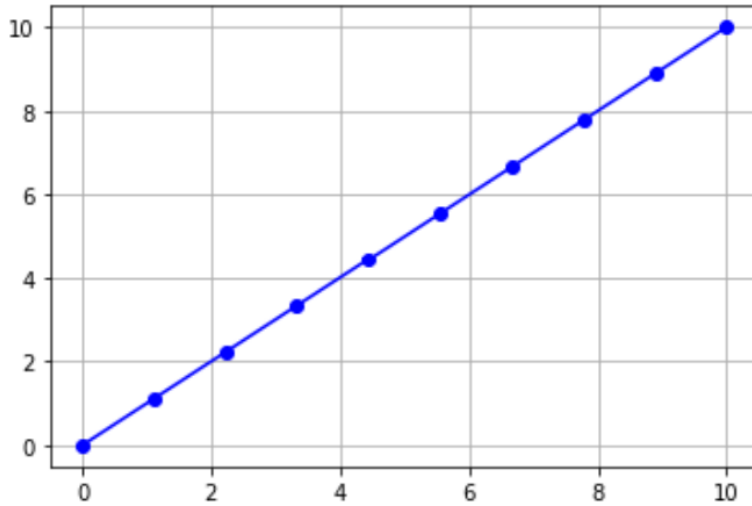
## <Logarithms>

# Log scale



Semilogarithmic axes.

$y = x$

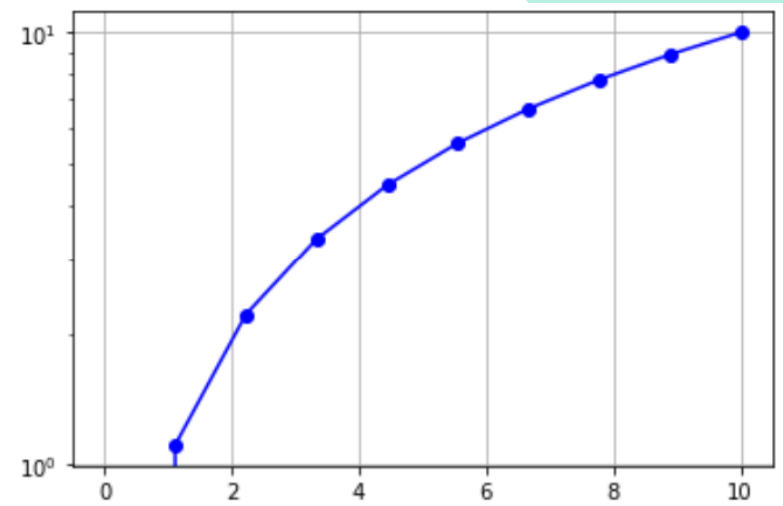


$y = x$

```
import numpy as np
x = np.linspace(0,10,10)
y = x
plt.plot(x, y)
plt.grid()
```

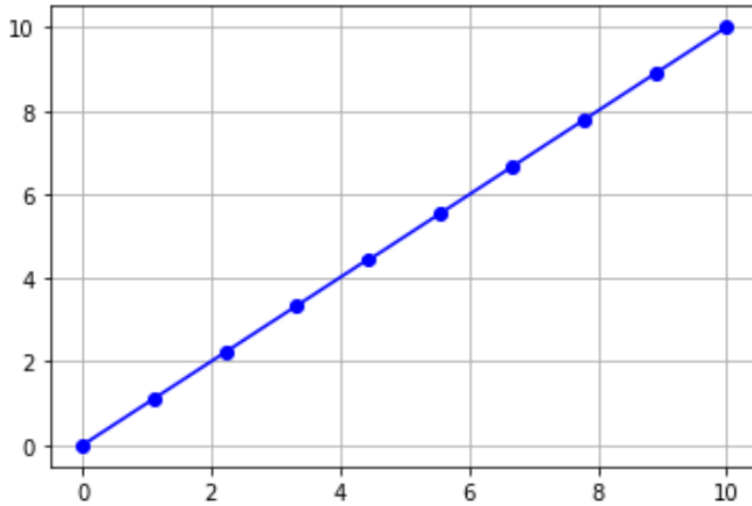
Same function

$y = x$



```
import numpy as np
x = np.linspace(0,10,10)
y = x
plt.plot(x, y)
plt.semilogy()
plt.grid()
```

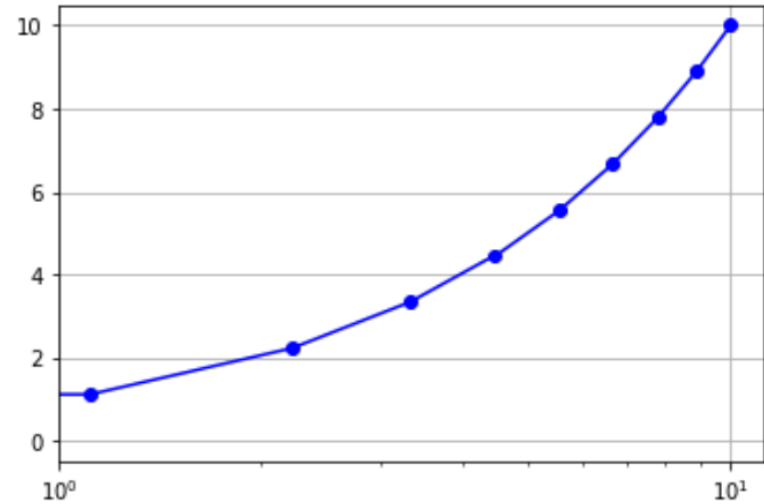
$y = x$



```
import numpy as np
x = np.linspace(0,10,10)
y = x
plt.plot(x, y)
plt.grid()
```

Same function

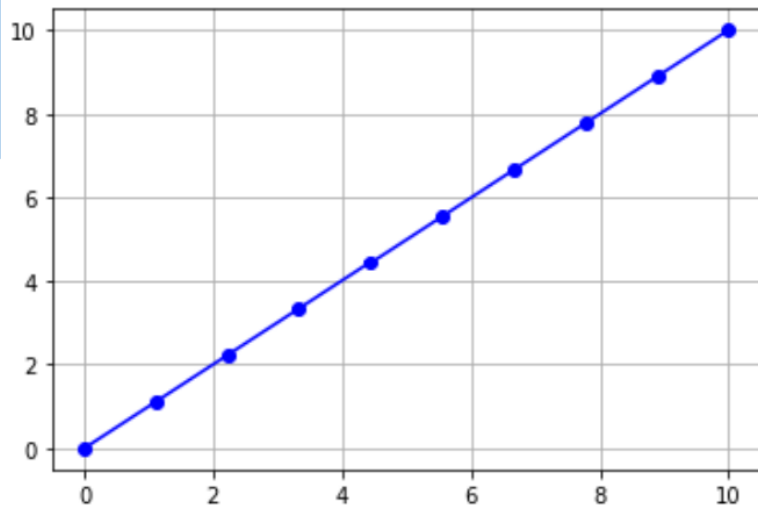
$y = x$



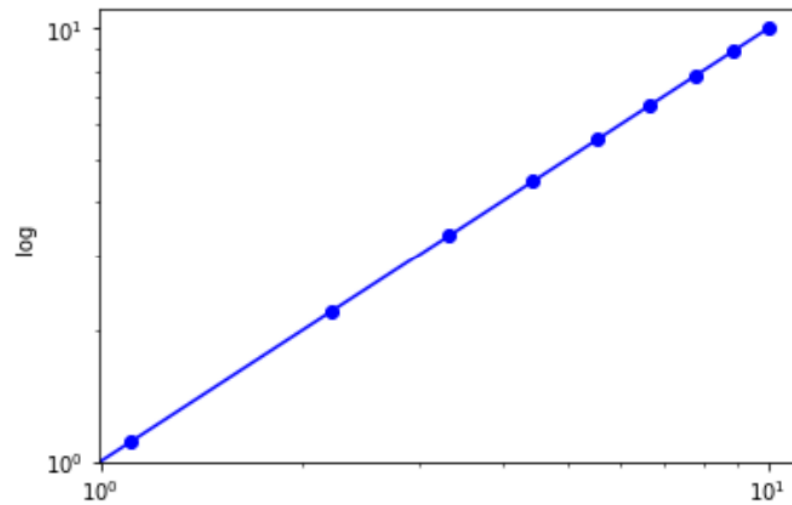
```
import numpy as np
x = np.linspace(0,10,10)
y = x
plt.plot(x, y)
plt.semilogx()
plt.grid()
```



$y = x$



$y = x$



```
import numpy as np
x = np.linspace(0,10,10)
y = x
plt.plot(x, y)
plt.grid()
```

Same function

```
import numpy as np
x = np.linspace(0,10,10)
y = x
plt.yscale('log')
plt.xscale('log')
plt.grid()
```

## Exercises

- (a) The streptococci bacteria population  $N$  at time  $t$  (in months) is given by  $N = N_0 e^{2t}$  where  $N_0$  is the initial population. If the initial population was 100, how long does it take for the population to reach one million?
- (b) The formula for the amount of energy  $E$  (in joules) released by an earthquake is

$$E = 1.74 \times 10^{19} \times 10^{1.44M}$$

where  $M$  is the magnitude of the earthquake on the Richter scale.

- The Newcastle earthquake in 1989 had a magnitude of 5 on the Richter scale. How many joules were released?
- In an earthquake in San Francisco in the 1900's the amount of energy released was double that of the Newcastle earthquake. What was its Richter magnitude?

**Plot the function using linear scale and log scale and tell what is the best way to understand the what is happening with the data**

(a) 4.6054 months

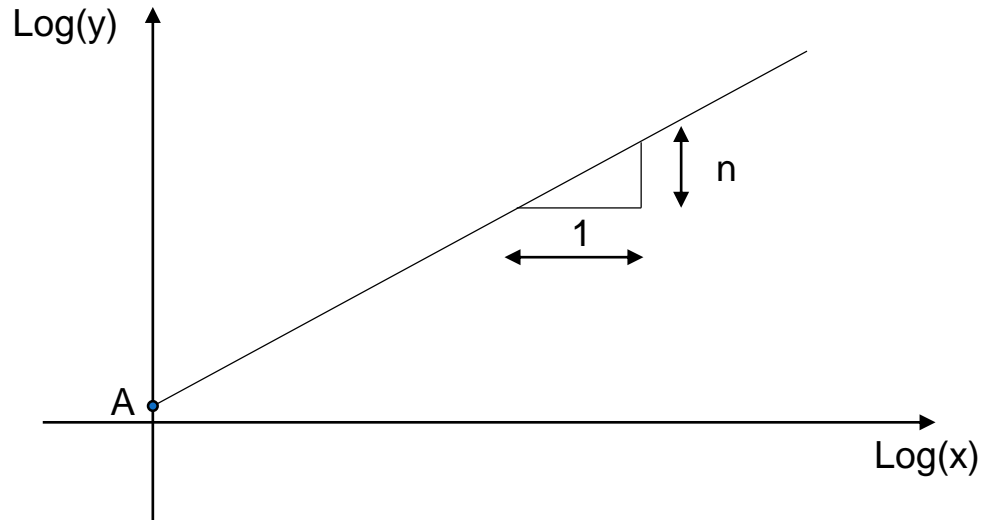
(b) i.  $2.76 \times 10^{26}$  Joules

ii. 5.2 on the Richter scale.

How to find the A and n in the function

$$y = Ax^n$$

We change y to loglog scale  
so we could find A and n easily



Consider the data

$x$	2	30	70	100	150
$y$	4.24	16.4	25.1	30.0	36.7

$$n = \frac{\log(y_2) - \log(y_1)}{\log(x_2) - \log(x_1)} = \frac{1.56 - 0.63}{2.18 - 0.3} = 0.5$$

$$\log_{10}(A) = 0.48.$$

$$A = 10^{0.48} = 3.0$$

$$y = 3x^{\frac{1}{2}}$$

```
import matplotlib.pyplot as plt
import numpy as np
x = [2, 30, 70, 100, 150]
y = [4.24, 16.4, 25.1, 30.0, 36.7]
```

```
logx = np.log10(x)
logy = np.log10(y)
print(np.poly1d(np.polyfit(logx, logy, 1)))
plt.plot(logx, logy, 'o')
```

## Exercises

The data below obeys a power law,  $y = Ax^n$ . Obtain the equation and select the correct statement.

$x$	5	15	30	50	95
$y$	10	90	360	1000	3610

- (a)  $n = 3$     (b)  $A = \frac{3}{2}$     (c)  $n = 4$     (d)  $A = \frac{1}{2}$

Answer:

$$n = 3$$

```
class BigFile:
```

```
    def __init__(self, datadir, ndims):
        idfile = os.path.join(datadir, "id.txt")
        self.names = [x.strip() for x in str.split(open(idfile).read()) if x.strip()]
        self.name2index = dict(zip(self.names, range(len(self.names))))
        self.ndims = ndims
        self.featurefile = os.path.join(datadir, "feature.bin")
        print "[BigFile] %d features, %d dimensions" % (len(self.names), self.ndims)
        print "        binary: %s" % self.featurefile
        print "        txt: %s" % idfile
```

```
    def __getitem__(self, requested, isname=True):
        if isname:
            index_name_array = [self.names[self.name2index[x], x] for x in requested if x in self.names]
        else:
            assert(min(requested) >= 0)
            assert(max(requested) < len(self.names))
            index_name_array = [(x, self.names[x]) for x in requested]
            index_name_array.sort()
            vecs = seq_read(self.featurefile, self.ndims, [x[0] for x in index_name_array])
            return [x[1] for x in index_name_array], vecs

    def shape(self):
        return [len(self.names), self.ndims]
```

<Statistics>

# Statistics

Consider the following two sets of data:

A: 10, 30, 50

B: 20, 30, 40

The mean of both two data sets is 30. But, the distance of the observations from the mean in data set A is larger than in the data set B. Thus, the mean of data set B is a better representation of the data set than is the case for set A.

Commonly used methods to interpreted data are **mean**, **median**, **mode**, **geometric mean**, etc.

**Mean:** Summing up all the observation and dividing by number of observations. Mean of 20, 30, 40 is

$$\bar{x} = \frac{x_1 + x_2 + x_3}{3} = \frac{\sum_{i=1}^3 x_i}{3} \quad \frac{20+30+40}{3} = 30.$$

# Statistics

**Median:** The middle value in an ordered sequence of observations. For example:

$\{9, 3, 6, 7, 5\} \rightarrow \{3, 5, 6, 7, 9\} \rightarrow$  middle value 6.

$\{9, 3, 6, 7, 5, 2\} \rightarrow \{2, 3, 5, 6, 7, 9\} \rightarrow$  average of the two middle values from the sorted sequence,  
 $\frac{5+6}{2} = 5.5$ .

**Mode:** The value that is observed most frequently. The mode is undefined for sequences in which no observation is repeated. For example:

$\{1, 2, 2, 3, 4, 7, 9\} \rightarrow$  mode is 2.



# Statistics

**Geometric mean:** Product of all the observation and finding the root to the power of the number of observations. Geometric mean of 2, 4, 8 is  $\sqrt[3]{2 * 4 * 8} = 4$

$$\sqrt[3]{x_1 \cdot x_2 \cdot x_3} = \left( \prod_{i=1}^3 x_i \right)^{\frac{1}{3}}$$

**Harmonic mean:** Dividing the number of observations by the reciprocal of all the observation. Harmonic mean of 1, 4, 4 is  $\frac{3}{\frac{1}{1} + \frac{1}{4} + \frac{1}{4}} = \frac{3}{1.5} = 2$ .

$$H = \frac{3}{\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3}} = \frac{3}{\sum_{i=1}^3 x_i^{-1}}$$

Variability (or dispersion) measures how much a data is spread around a central value.

Commonly used methods: *variance*, *standard deviation*, *interquartile range*, *etc.*

**Variance**: average of the squares of the deviations of the observations from their mean. For example:

Variance of 5, 7, 3  $\rightarrow$  Mean is  $\frac{5+7+3}{3} = 5$  and the variance is  $\frac{(5-5)^2 + (7-5)^2 + (3-5)^2}{3-1} = 4$

$$s^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2}{3-1}$$

**Standard Deviation**: Square root of the variance. The standard deviation of the above example is  $\sqrt{4} = 2$ .

**Quartile:** Division of the data in four regions that cover the total range of observed values. In other words, the first quartile (Q1) is the first 25% of the data. The second quartile (Q2) is from 25% to 50% of the data. The third quartile (Q3) is from 50% to 75% of the data. For example:

Consider the data: {3, 6, 7, 11, 13, 22, 30, 40, 44, 50, 52, 61, 68, 80, 94}

Divide the data in 4 equal groups. In this case, the group consists of 4 elements:

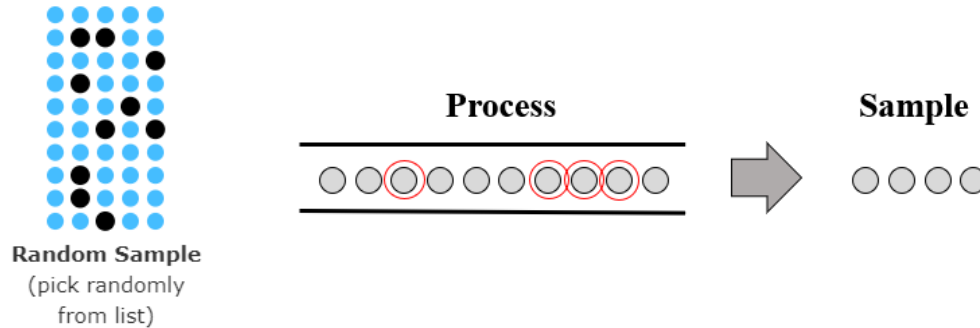
Q1	Q2	Q3
{3, 6, 7, 11, 13, 22, 30, 40, 44, 50, 52, 61, 68, 80, 94}		

The first quartile is  $Q1 = 11$ . The second quartile is  $Q2 = 40$  (this is also the median). The third quartile is  $Q3 = 61$ .

**Inter-quartile Range:** Difference between Q3 and Q1. Considering the last case,  $61 - 40 = 21$ .

# Statistics

**Sampling:** selection of a subset of individuals from a statistical population to estimate characteristics of the whole population.

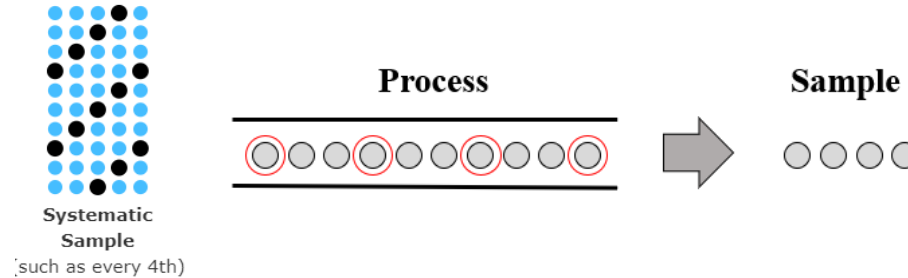


**Random sample:** pick the individuals randomly from the population. It is the best way to avoid bias. The results are better when you pick more individuals. For example, nationwide opinion polls survey around 2000 people, and the results are nearly as good (within about 1%) as asking everyone.

```
Sample = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]  
print("With list:", random.sample(Sample, 5))
```

# Statistics

**Systematic sample:** select individual following a specific rule such as picking an individual every Nth



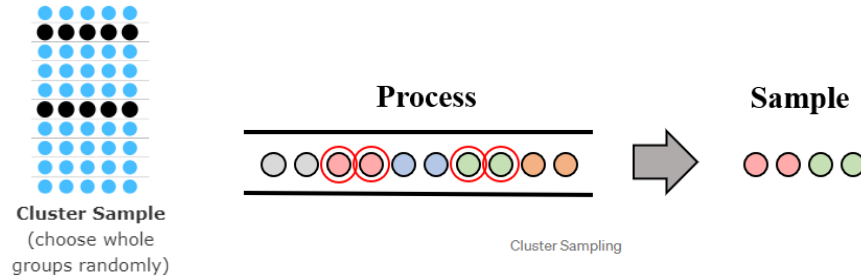
[mathsisfun.com/data/sampling](https://mathsisfun.com/data/sampling)

$N = \text{step} = 2$

```
systematic_sample = []
sample = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
step = 2
print(np.mean(sample))
size = int(len(sample)/step)
for i in range(0,size):
    systematic_sample.append(sample[i*step])
systematic_mean = np.mean(systematic_sample)
print(systematic_mean)
```

# Statistics

**Cluster sample:** divide the individuals in many groups and select randomly whole groups. For example, divide the city in regions and pick randomly 3 regions to survey everyone there.



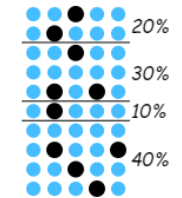
[mathsisfun.com/data/sampling](https://mathsisfun.com/data/sampling)

```
systematic_sample = []
sample = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
step = 3
print(np.mean(sample))
size = int(len(sample)/step)
for i in range(0,size):
    systematic_sample.append(sample[i*step])
    systematic_sample.append(sample[i*step]+1)

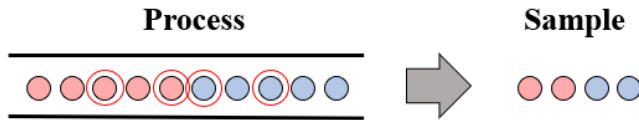
systematic_mean = np.mean(systematic_sample)
print(systematic_mean)
```

# Statistics

**Stratified sample:** divide the individuals in specific categories and pick them according to its proportional in the sample.



**Stratified Sample**  
(randomly, but in  
ratio to group size)



```
import random as rd
def split_list(alist, wanted_parts=1):
    length = len(alist)
    return [ alist[i*length // wanted_parts: (i+1)*length
              // wanted_parts]
            for i in range(wanted_parts) ]

systematic_sample = []
sample = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 ,14,
15]
A, B, C = split_list(sample, 3)
total= split_list(sample, 3)
stratified_list =[]
print('mean all values:', np.mean(sample) )
splited_list = [A, B, C]

for i in range(0,len(splited_list)):
    stratified_list.append(
splited_list[i][random.randint(0,4)])

systematic_mean = np.mean(stratified_list)
print('stratified mean', systematic_mean)
```

## Output:

```
[1, 1, 2, 2, 3, 3, 4, 1, 2]
Mean = 2.111111111111111
Median = 2.0
Mode = 1
Geometric Mean = 1.8761425449123872
Harmonic Mean = 1.6615384615384616
Variance = 0.9876543209876544
Standard deviation = 0.9938079899999066
Q1 = 1.0
Q2 = 2.0
Q3 = 3.0
```

```
import numpy as np
from scipy import stats
from scipy.stats import gmean
from scipy.stats import hmean
import random

numbers = [ 1, 1, 2, 2, 3, 3, 4, 1, 2 ]
print(numbers)
Mean = np.mean(numbers)
Median = np.median(numbers)
Mode = int(stats.mode(numbers)[0]) # index 1 gives frequency
print("Mean = ", Mean, "\nMedian = ", Median, "\nMode = ", Mode)

Geometric_Mean = gmean(numbers)
Harmonic_mean = hmean(numbers)
print("Geometric Mean = ", Geometric_Mean, "\nHarmonic Mean = ",
      Harmonic_mean)

Variance = np.var(numbers)
Standard_Deviation = np.std(numbers)
print("Variance = ", Variance, "\nStandard deviation = ",
      Standard_Deviation)

Q1 = np.percentile(numbers, 25)
Q2 = np.percentile(numbers, 50)
Q3 = np.percentile(numbers, 75)
print("Q1 = ", Q1, "\nQ2 = ", Q2, "\nQ3 = ", Q3) # 1 1 1 2 2 2 3 3
```



# Exercises

1. Ten observations  $x_i$  are given:

4, 7, 2, 9, 12, 2, 20, 10, 5, 9

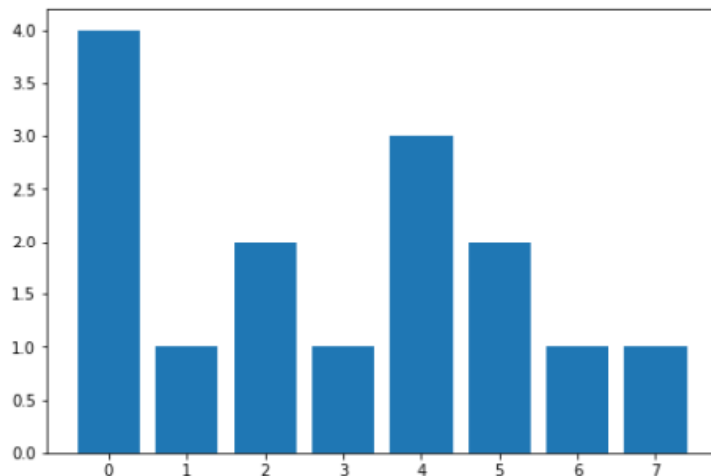
Determine the median, upper, and lower quartile and the inter-quartile range.

2. Four observations  $x_i$  are given:

2, 5, 10, 11

Determine the mean, empirical variance, and empirical standard deviation.

# Exercises



3. Find the median of the data in Figure
4. Find the standard deviation of the data in Figure

# Exercises

Use Pisa data available in <https://data.oecd.org/pisa/mathematics-performance-pisa.htm#indicator-chart>

Calculate Portugal's mode, median, variance and standard deviation for boys and girls over the years for 3 countries

country	AUS		CAN		CZE	
	Boys	girls	Boys	girls	Boys	girls
mode						
Meadian						
Variance						
Std deviation						

## Exercises – Solution

The observations are ordered according to size:

2, 2, 4, 5, 7, 9, 9, 10, 12, 20

The median is the mean of the two “middle” observations, i.e.

$$x(0.5) = \frac{7 + 9}{2} = 8$$

Similarly the lower and upper quartiles are

$$x(0.25) = \frac{2 + 4}{2} = 3, \quad x(0.75) = \frac{10 + 12}{2} = 11$$

The inter-quartile range thus becomes  $11 - 3 = 8$ .

## Exercises – Solution

The observations are ordered according to size:

2, 2, 4, 5, 7, 9, 9, 10, 12, 20

The median is the mean of the two “middle” observations, i.e.

$$x(0.5) = \frac{7 + 9}{2} = 8$$

Similarly the lower and upper quartiles are

$$x(0.25) = \frac{2 + 4}{2} = 3, \quad x(0.75) = \frac{10 + 12}{2} = 11$$

The inter-quartile range thus becomes  $11 - 3 = 8$ .

## Exercises – Solution

Mean is:

$$\bar{x} = \frac{2 + 5 + 10 + 11}{4} = 7$$

The empirical variance is

$$s^2 = \frac{(2 - 7)^2 + (5 - 7)^2 + (10 - 7)^2 + (11 - 7)^2}{4 - 1} = 18$$

The empirical standard deviation thus becomes

$$s = \sqrt{18} \approx 4.24$$


```
class BigFile:
```

```
    def __init__(self, datadir, ndims):
        idfile = os.path.join(datadir, "id.txt")
        self.names = [x.strip() for x in str.split(open(idfile).read()) if x.strip()]
        self.name2index = dict(zip(self.names, range(len(self.names))))
        self.ndims = ndims
        self.featurefile = os.path.join(datadir, "feature.bin")
        print "[BigFile] %d features, %d dimensions" % (len(self.names), self.ndims)
        print "        binary: %s" % self.featurefile
        print "        txt: %s" % idfile
```

```
    def __getitem__(self, requested, isname=True):
        if isname:
            index_name_array = [(self.name2index[x], x) for x in requested if x in self.names]
        else:
            assert(min(requested) >= 0)
            assert(max(requested) < len(self.names))
            index_name_array = [(x, self.names[x]) for x in requested]
            index_name_array.sort()
            vecs = seq_read(self.featurefile, self.ndims, [x[0] for x in index_name_array])
            return [x[1] for x in index_name_array], vecs

    def shape(self):
        return [len(self.names), self.ndims]
```

<Probability>



What are the *chances* that sales will increase if we increase prices?

What is the *likelihood* a new assembly method will increase productivity?

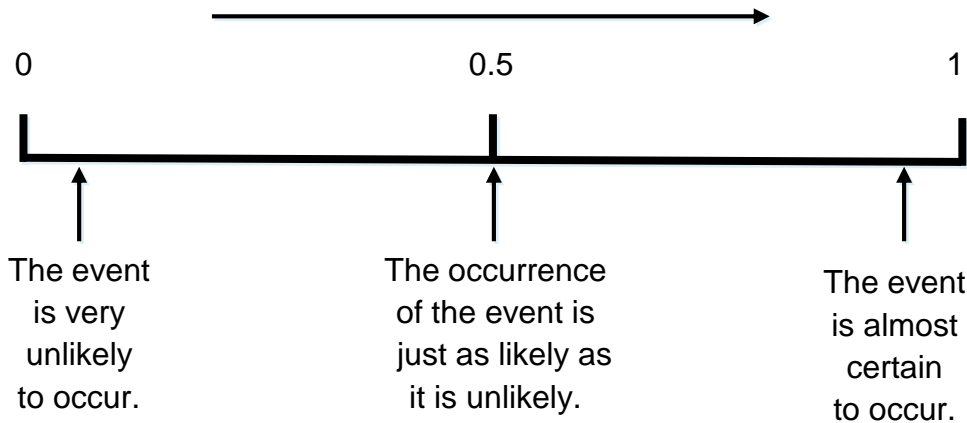
What are the odds that a new investment will be profitable?



- Basic Probability Concepts:  
Sample Spaces and Events, Simple Probability, and Joint Probability,
- Conditional Probability
- Probability Distributions

# Probability

*Total number of outcomes = Sample space*



Probability:

$$\text{Probability event} = \frac{\text{number of ways it can happen}}{\text{Total number of outcomes}}$$

*Total number of outcomes = Sample Space*

- Two Conditions:
- Value is between 0 and 1.
- Sum of the probabilities of all events must be 1.

# Probability

## Sample Space

Deck of 52 cards

- 13 *ranks*: 2, 3, ..., 9, 10, J, Q, K, A
- 4 *suits*: ♥, ♠, ♦, ♣,

What is the probability of picking the **5 of Clubs** from this deck?

$$P = \frac{1}{52}$$

Single probability

```
import numpy as np
import random
import matplotlib.pyplot as plt

simulation_number = 10000
club = 0; spade = 0; diamont = 0; heart = 0; percentages = []
for i in range(0, simulation_number):

    card_points = [1, 12, 11, 10, 2, 3, 4, 5, 6, 7, 8, 9, 10]
    card_signs = ['Heart', 'CLUB', 'DIAMOND', 'SPADE']
    random_point = random.choice(card_points)
    random_sign = random.choice(card_signs)
    random_card = random_point, random_sign
    # print (random_card)
    ##Condition

    if (random_card[0] == 7 and random_card[1] == 'CLUB'):
        club += 1;

X = ['CLUB']
percentages.append(np.round(club/simulation_number*100, 2))
fig = plt.figure()
ax = fig.add_axes([0, 0, 1, 1])
ax.bar(X, percentages)
print(sum(percentages), '%')
```

# Joint probability of A and B

$$P(A \cap B)$$

$$= \frac{\text{Number of Event Outcomes from both A and B}}{\text{Total Number of Possible Outcomes in Sample Space}}$$

e.g.  $P(7 \text{ of Clubs and } 7 \text{ of spade})$

$$= \frac{2}{52 \text{ Total Number of Cards}} = \frac{1}{26}$$



```
##Condition
```

```
if (random_card[0] == 7 and random_card[1] == 'CLUB'):  
    club+=1;  
if (random_card[0] == 7 and random_card[1] == 'SPADE'):  
    spade+=1;
```

## Addition Law

The addition law provides a way to compute the probability of event  $A$ , or  $B$ , or both  $A$  and  $B$  occurring

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

## Multiplication Law for Independent Events

If the probability of event  $A$  is not changed by the existence of event  $B$ , we would say that events  $A$  and  $B$  are independent.

$$P(A \cap B) = P(A) \times P(B)$$

# The Probability of a Compound Event, A or B:

$$P(A \cup B) = \frac{\text{Numer of Event Outcomes from Either A or B}}{\text{Total Outcomes in the Sample Space}}$$

e.g. **P(Red Card or Ace)**

$$= \frac{4 \text{ Aces} + 26 \text{ Red Cards} - 2 \text{ Red Aces}}{52 \text{ Total Number of Cards}} = \frac{28}{52} = \frac{7}{13}$$

```
##Condition
```

```
if ((random_card[1] == 'Heart') or (random_card[1] == 'DIAMOND') or  
    (random_card[0] == 1 and random_card[1] == 'SPADE') or  
    (random_card[0] == 1 and random_card[1] == 'CLUB')):  
    club+=1;
```



## Experiment

Toss a coin

Inspection a part

Conduct a sales call

Roll a die

Play a football game



## Outcomes

Head, tail

Defective, non-defective

Purchase, no purchase

1, 2, 3, 4, 5, 6

Win, lose, tie

## Exercises

1. A fair coin is tossed, and a fair die is thrown. Write down sample spaces for
  - (a) the toss of the coin;
  - (b) the throw of the die;
  - (c) the combination of these experiments.

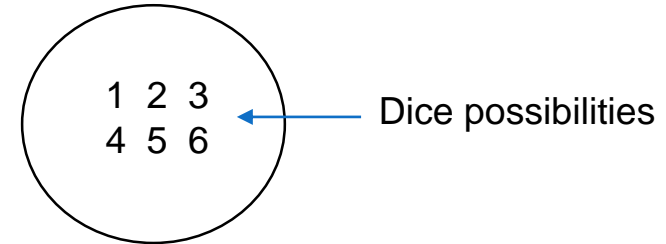


## Exercises

2. What is the chance of rolling a “4” with a dice?

$$\begin{array}{l} \text{Number of ways it can happen} = 1 \\ \text{Total number of outcomes} = 6 \end{array} \Rightarrow P = \frac{1}{6} = 17\%$$

### Sample Space



3 . Find the probability of the **sum** of 2 dices been  $> 7$  or odd ?

$$\begin{array}{l} \text{Number of ways it can happen} = 26 \\ \text{Total number of outcomes} = 36 \end{array}$$

$$P = \frac{26}{36} = 75\%$$

### Dice possibilities

(1,2)(1,4)(1,5)(1,6)  
(2,1)(2,3)(2,5)  
(3,2)(3,4),(3,5) (3,6)  
(4,1)(4,3),(4,4),(4,5),(4,6),  
(5,2),(5,3),(5,4),(5,5),(5,6)  
(6,1),(6,6),(6,2),(6,3),(6,4),(6,5)

Plot the bar chart with the probability of each possible outcome

## Exercises

4 . Find the probability of the **sum** of 2 dices been  $> 7$  or even ?  
Explain why it is different from the exercise 3

5. What is the chance of getting 2 heads from two coins?

$$\begin{array}{l} \text{Number of ways it can happen} = 1 \\ \text{Total number of outcomes} = 4 \end{array} \Rightarrow P = \frac{1}{4} = 25\%$$

(head, head) (head, tail)  
(tail, head) (tail, tail)

Plot the bar chart with the probability of each possible outcome

## Exercises

6. Let  $A$  be the event that a head is tossed, and  $B$  be the event that an odd number is thrown. Directly from the sample space, calculate  $P(A \cap B)$  and  $P(A \cup B)$ .

$P(A) = \frac{1}{2} = P(B)$ . We can assume that the two events are independent, so

$$P(A \cap B) = P(A)P(B) = \frac{1}{4}.$$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) = \frac{3}{4}$$

## Conditional Probability

The Probability of Event A given that Event B has occurred:

$$P(A | B) = \frac{P(A \cap B)}{P(B)}$$

```
card_points =[1]
```

```
if (random_card[0] == 1 and (random_card[1] == 'Heart' or random_card[1] == 'DIAMOND')):  
    club+=1
```

e.g.  $P(\text{Red Card given that it is an Ace}) = \frac{2 \text{ Red Aces}}{4 \text{ Aces}} = \frac{1}{2}$

“

- *Make it work*
- *Make it Right*
- *Make it Fast*

# O futuro profissional começa aqui

iscte  
INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA

 **emprego  
digital**

 **UPskill**