Lecture 11

Math statistics

Math statistics

Mathematical statistics is a branch of mathematics that develops methods for obtaining, describing and processing experimental data in order to identify and study the patterns of random mass phenomena for scientific and practical conclusions.

The subject of mathematical statistics: the study of random variables (events, processes) based on the results of observations.

Math statistics

Tasks of mathematical statistics:

- collection, description and ordering of statistical material;
- selection and determination of the type of distribution for the sets of random variables obtained in the experiment;
- estimation, at least approximate, of distribution parameters;
- verification of the plausibility of the put forward hypothesis about the correspondence of the statistical material to theoretical conclusions.

Population and sample

The set of all studied objects or possible results of all conceivable observations of some random variable that can be obtained under given conditions is called the **population**.

The number **N** (finite or infinite) of objects (observations) in the aggregate is called its **size**.

A set of randomly selected n objects (measurements) of a random variable from the general population is called a **sample**. The number **n** is the sample size.

The sample must be **representative**. It is believed that the sample is representative if all objects of the population have the same probability of being included in the sample, i.e., the choice is made randomly.

Statistical distribution of the sample

Discrete case.

Let for the study of some random variable X a sample is extracted from the population, in which the value x_1 observed m_1 times, x_2 - m_2 times, …, x_k - m_k times.

$m_1 + m_2 + \cdots + m_k = n -$ **sample size**;

$x_1, x_2, \dots, x_k -$**values**; $m_1, m_2, \dots, m_k -$ **frequencies;**

$p_i^* = \frac{m_i}{n}, i = 1, \dots, k -$ **relative frequencies,** $\sum_{i=1}^{k} p_i^* = 1$

Statistical distribution of the sample

An ordered list of values and their corresponding frequencies is called the **statistical distribution of the sample.**

x_i	x_1	x_2	…	x_k
m_i	m_1	m_2	…	m_k
$p_i^* = \frac{m_i}{n}$	p_1^*	p_2^*	…	p_k^*

Statistical distribution of the sample

Continuous case.

If the attribute is continuous or the number of different values in the sample is large, an **interval variation series** is composed.

The entire span of measurement of sample values $[x_{min}; x_{max}]$ divided into partial intervals.

The number of intervals can be determined using the Sturgess formula:

$$k \approx 1 + \log_2 n$$

The length of the interval is found by the formula: $h = \frac{x_{\max} - x_{\min}}{k}$

Statistical distribution of the sample

Partial gaps are entered in the first line of the table of the interval series $[x_0; x_1],\ (x_1; x_2],\ \ldots,\ (x_{k-1}; x_k]$ having the same length h.

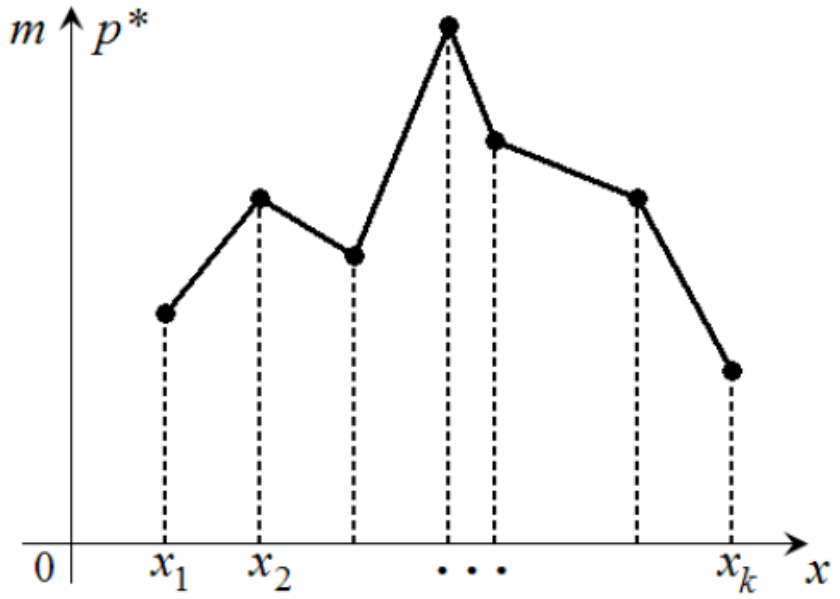
On the second line enter the number of observations $m_i, i = 1, \ldots, k$, included in each interval.

$(x_{i-1}; x_i]$	$[x_0; x_1]$	$(x_1; x_2]$	$\ldots$	$(x_{k-1}; x_k]$
m_i	m_1	m_2	$\ldots$	m_k
$p_i^* = \frac{m_i}{n}$	p_1^*	p_2^*	$\ldots$	p_k^*

Graphical representation of a statistical series

Discrete variation series (frequency polygon).

A polygon of frequencies is a broken line connecting the points of a discrete series $(x_1; m_1)$, $(x_2; m_2)$, …, $(x_k; m_k)$, polygon of relative frequencies is a broken line connecting the points $(x_1; p_1^*)$, $(x_2; p_2^*)$, …, $(x_k; p_k^*)$.



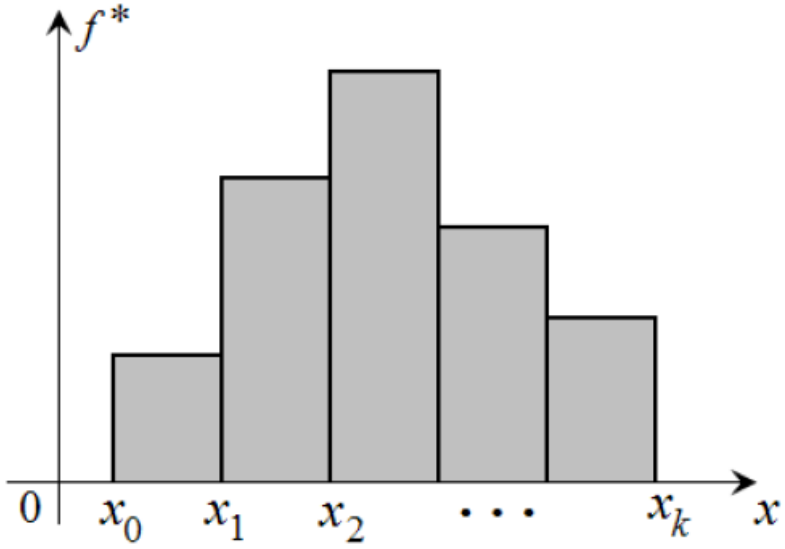
Graphical representation of a statistical series

Interval variation series (histogram).

A histogram of relative frequencies is a stepped figure consisting of rectangles whose bases are partial intervals (length $h_i = x_i - x_{i-1}$) and the height is $f_i^* = \frac{p_i^*}{h}$.

Beginning of the first partial interval:

$$x_0 = x_{\min} - \frac{h}{2}$$



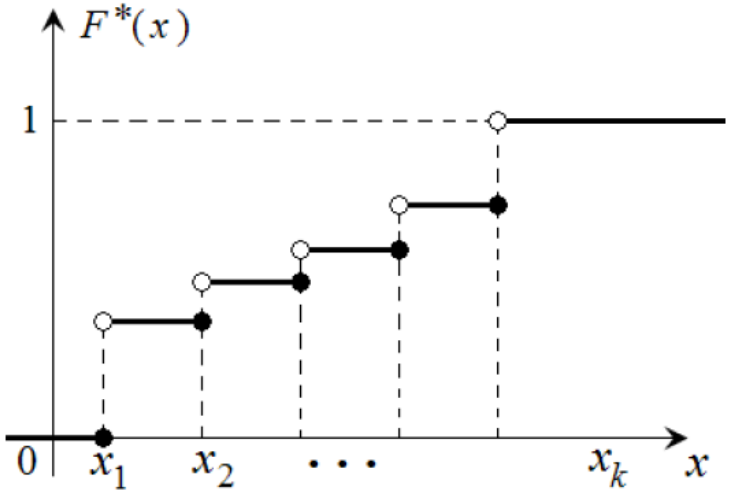
Empirical cumulative distribution function (eCDF)

Function $F^*(x)$, specifying for each value x the relative frequency of the event $\{X < x\}$. That as:

$$F^*(x) = \frac{m_x}{n},$$

where m_x – number of sample values of X, less than x, a n – sample size.

$$F^*(x) = \begin{cases} 0, & x \le x_1, \\ p_1^*, & x_1 < x \le x_2, \\ p_1^* + p_2^*, & x_2 < x \le x_3, \\ p_1^* + p_2^* + p_3^*, & x_3 < x \le x_4, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, & \\ 1, & x > x_k. \end{cases}$$



Numerical characteristics of the sample

Sample mean $\overline{x}$ is called the average of all values of the sample. This is a statistical analogue of the expectation of a random variable.

$$\overline{x} = \frac{x_1 + x_2 + \ldots + x_n}{n} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$$

If statistics are grouped:

$$\overline{x} = \frac{1}{n} \cdot \sum_{i=1}^{k} x_i m_i = \sum_{i=1}^{k} x_i \cdot \frac{m_i}{n} = \sum_{i=1}^{k} x_i p_i^*$$

Numerical characteristics of the sample

Sample variance s^2:

$$s^2 = \frac{1}{n}\left((x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \ldots + (x_n - \bar{x})^2\right) = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - \bar{x})^2$$

If statistics are grouped:

$$s^2 = \frac{1}{n} \cdot \sum_{i=1}^{k} (x_i - \bar{x})^2 \cdot m_i = \sum_{i=1}^{k} (x_i - \bar{x}_в)^2 \cdot \frac{m_i}{n} = \sum_{i=1}^{k} (x_i - \bar{x})^2 \cdot p_i^*.$$

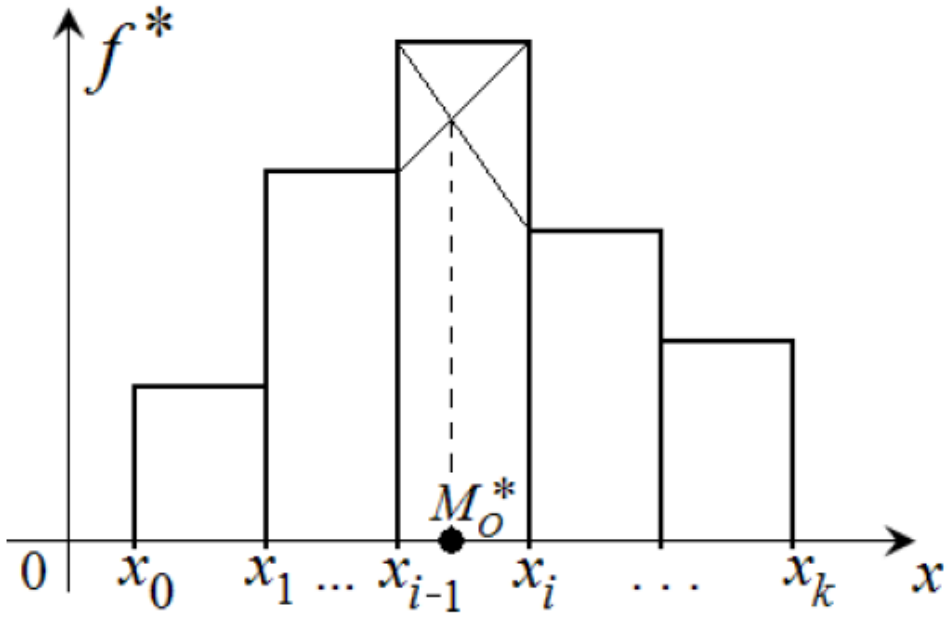
This is the statistical analog of the variance of the theoretical distribution.

Sample standard deviation s: $s = \sqrt{s^2}$

Descriptive characteristics of the sample

Mode M_0^* is called such a value of the options, which corresponds to the highest frequency.

$$M_o^* = x_{i-1} + h_i \cdot \frac{p_i^* - p_{i-1}^*}{(p_i^* - p_{i-1}^*) + (p_i^* - p_{i+1}^*)}$$



Descriptive characteristics of the sample

The median M_e^* is the value of the feature that falls in the middle of the ranged series of observations.

$$M_e^* = x_{i-1} + \frac{h_i}{p_i^*} \cdot \left(0,5 - \sum_{j=1}^{i-1} \frac{h_j}{p_j^*} \right)$$

To compare the scattering values with respect to the sample mean of two variational series, the coefficient of variation is used.

$$V^* = \frac{\sigma_в}{\overline{x}_в} \cdot 100\%$$

Properties of the sample mean and sample variance

If the values increase (decrease) by the same number C, then the sample mean will also increase (decrease) by this number, and the sample variance will remain unchanged.

If the options increase (decrease) by h times, then the sample mean will also increase (decrease) by h times, and the sample variance will increase (decrease) by h^2 times.

Empirical moments

Sample initial moment of the k-th order:

$$\nu_k^* = \sum_{i=1}^{r} x_i^k \, p_i^*$$

Sample central moment of the k-th order:

$$\mu_k^* = \sum_{i=1}^{r} (x_i - \overline{x}\ )^k \cdot p_i^*$$

Asymmetry and kurtosis

Asymmetry and kurtosis are used to estimate the deviation of the empirical distribution from the normal one.

The sample coefficient of asymmetry (“skewness”) is the value

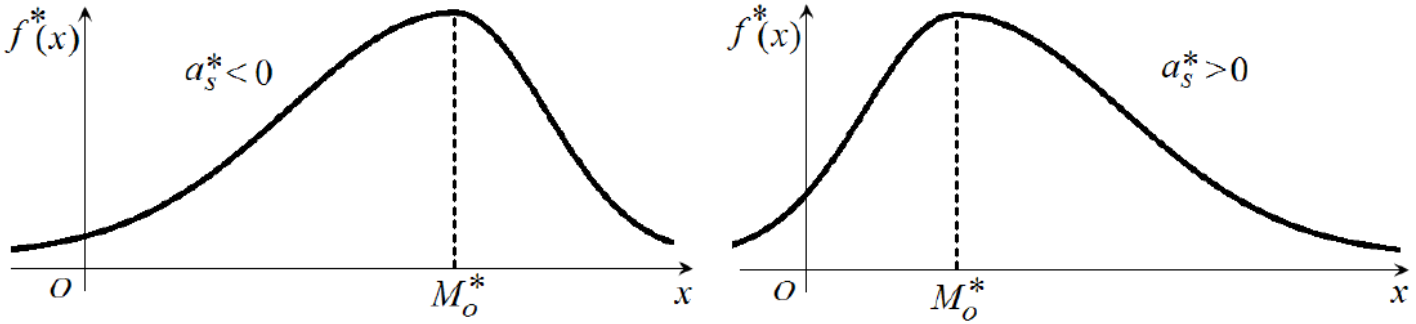
$$a_s^* = \frac{\mu_3^*}{\sigma_в^3}.$$

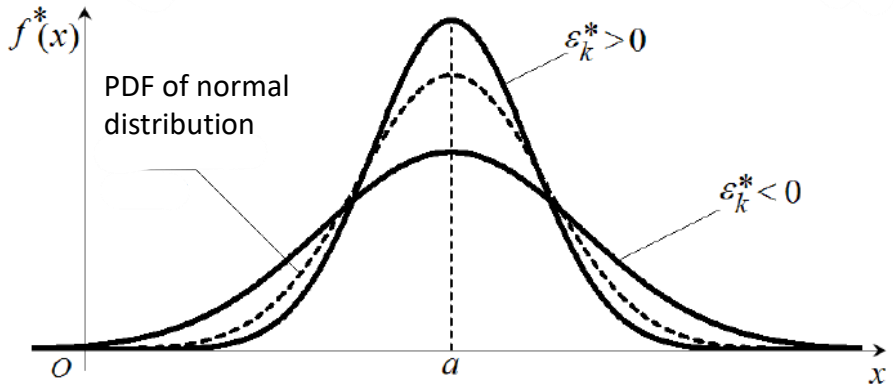
The sample coefficient of kurtosis ("pointedness") is determined by the formula

$$\varepsilon_k^* = \frac{\mu_4^*}{\sigma_в^4} - 3.$$

Kurtosis characterizes the steepness of the rise of the distribution curve compared to the normal curve

Asymmetry and kurtosis





Math statistics in Python

https://colab.research.google.com/

Math statistics in Python

The **statistics** package will be used. This module provides functions for calculating mathematical statistics of numeric (Real-valued) data.

Let's randomly generate some data set.

```
import random
import statistics

rand_list=[]
n=100
for i in range(n):
    rand_list.append(random.randint(1,500))
print(rand_list)
```

Math statistics in Python

To show this list better:

```
for x, i in enumerate(rand_list):
    if x % 5 == 0:
        print()
    print(i, end=' ')
```

```
196 163 68 462 6
9 338 400 208 256
206 382 26 87 105
168 92 205 133 458
364 269 35 402 151
35 423 4 200 219
358 76 33 485 88
101 166 407 6 215
172 34 390 295 28
38 193 150 151 262
101 96 118 8 277
416 92 52 299 331
253 480 417 206 134
68 189 237 155 347
247 248 411 488 46
236 460 412 245 360
167 168 161 209 23
483 304 499 167 96
265 51 464 267 23
233 262 70 336 321
```

Math statistics in Python

Plotting a histogram.

In Python, a basic histogram can be plotted using either **matplotlib** or **seaborn**

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Math statistics in Python

Plotting a histogram.

Let the width of each column be 20. Number of columns:

```
import math
num=math.ceil((max(rand_list)-min(rand_list))/20)
print(num)
```

```
25
```

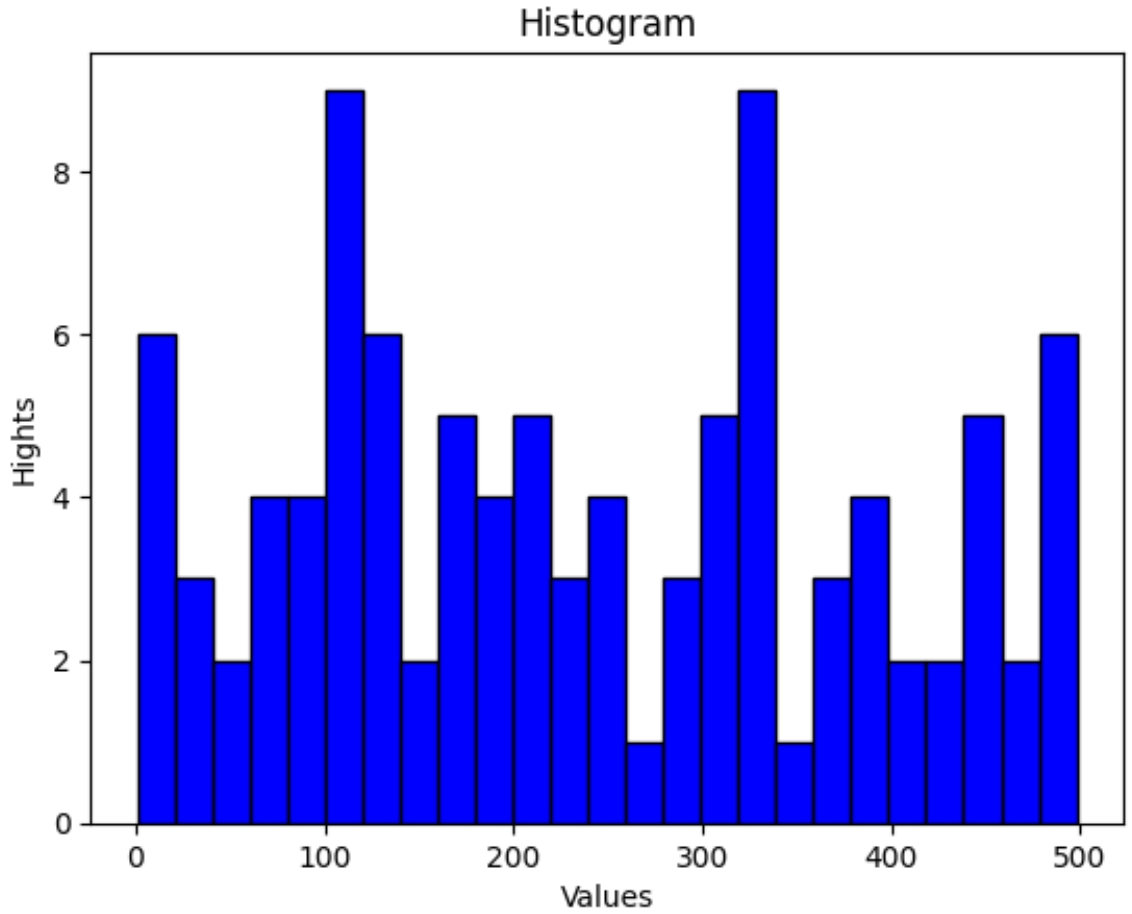
Math statistics in Python

Plotting a histogram.

```
import matplotlib.pyplot as plt
plt.hist(rand_list, color = 'blue', edgecolor = 'black',
         bins = num)
plt.title('Histogram')
plt.xlabel('Values')
plt.ylabel('Hights')
```

Math statistics in Python

Plotting a histogram.



Math statistics in Python

Density plots.

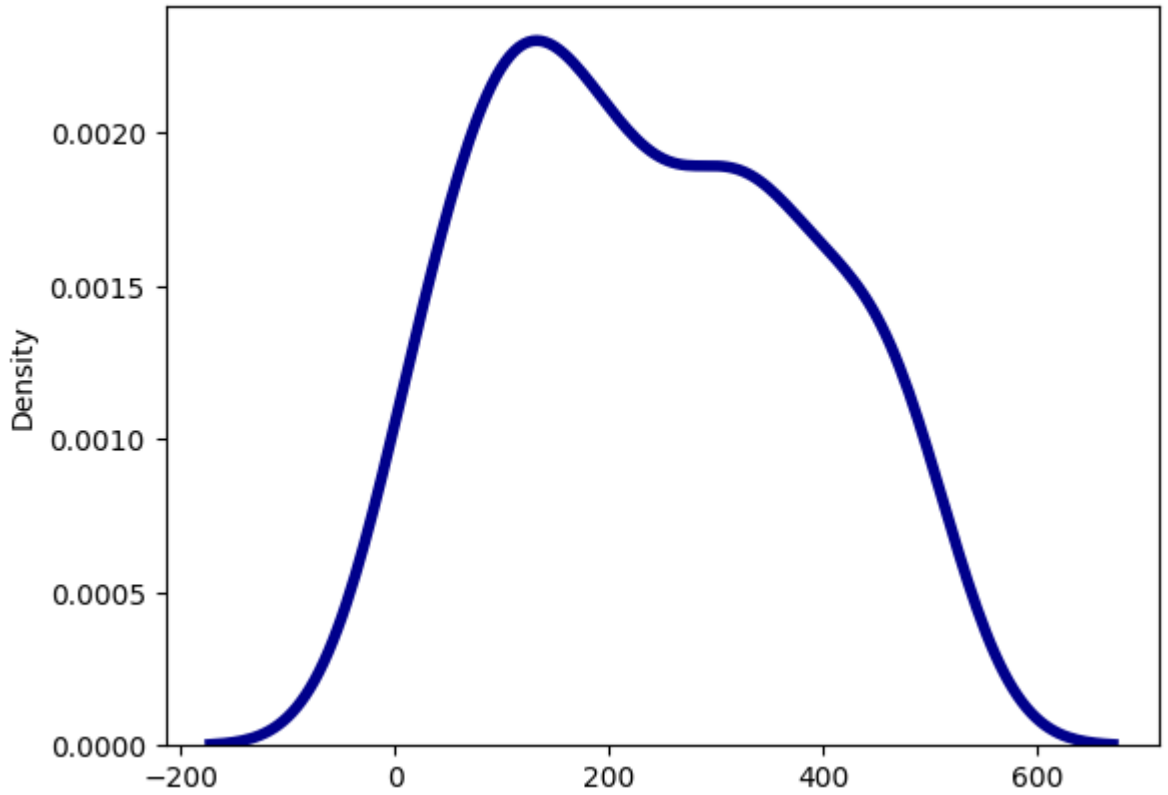
The distribution graph can be called a continuous smoothed analogue of the histogram.

The x-axis here, like the histogram, represents the value of the variable. The y-axis on the density plot is the probability density function, not the probability itself. The difference between the two is that the probability density is the probability per unit along the x-axis.

```
import seaborn as sns
sns.distplot(rand_list, hist=False, kde=True,
             bins=num, color = 'darkblue',
             hist_kws={'edgecolor':'black'},
             kde_kws={'linewidth': 4})
```

Math statistics in Python

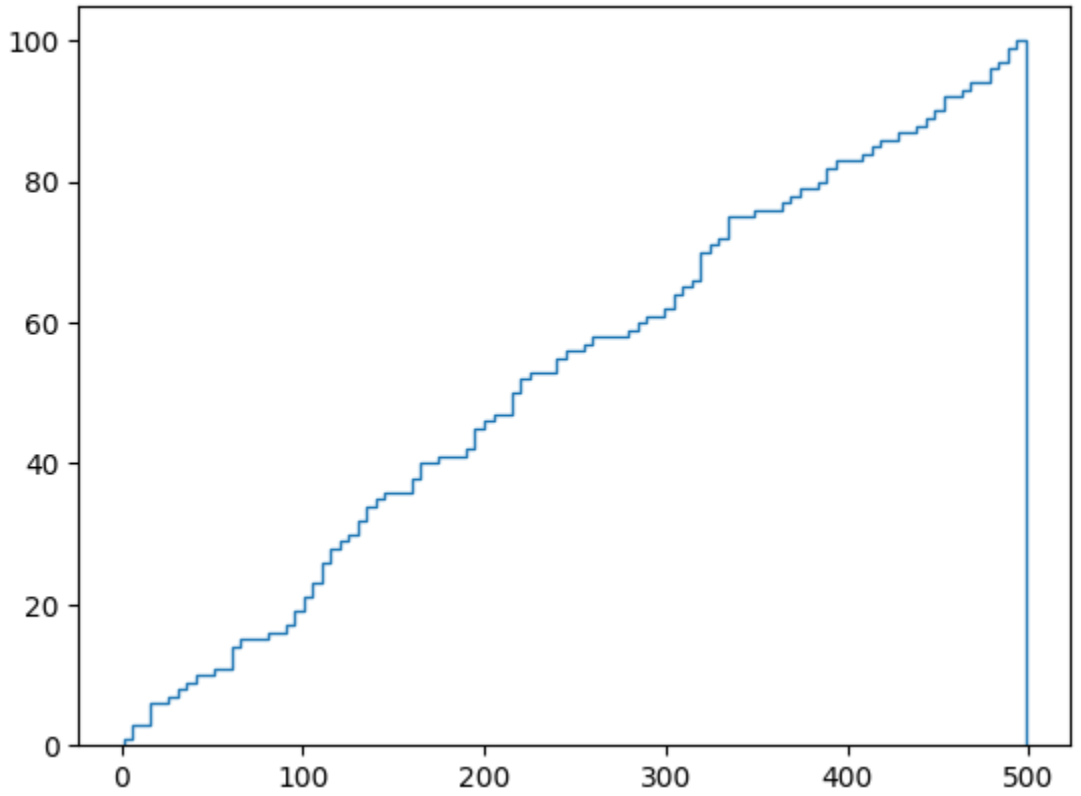
Density plots.



Math statistics in Python

Empirical cumulative distribution function.

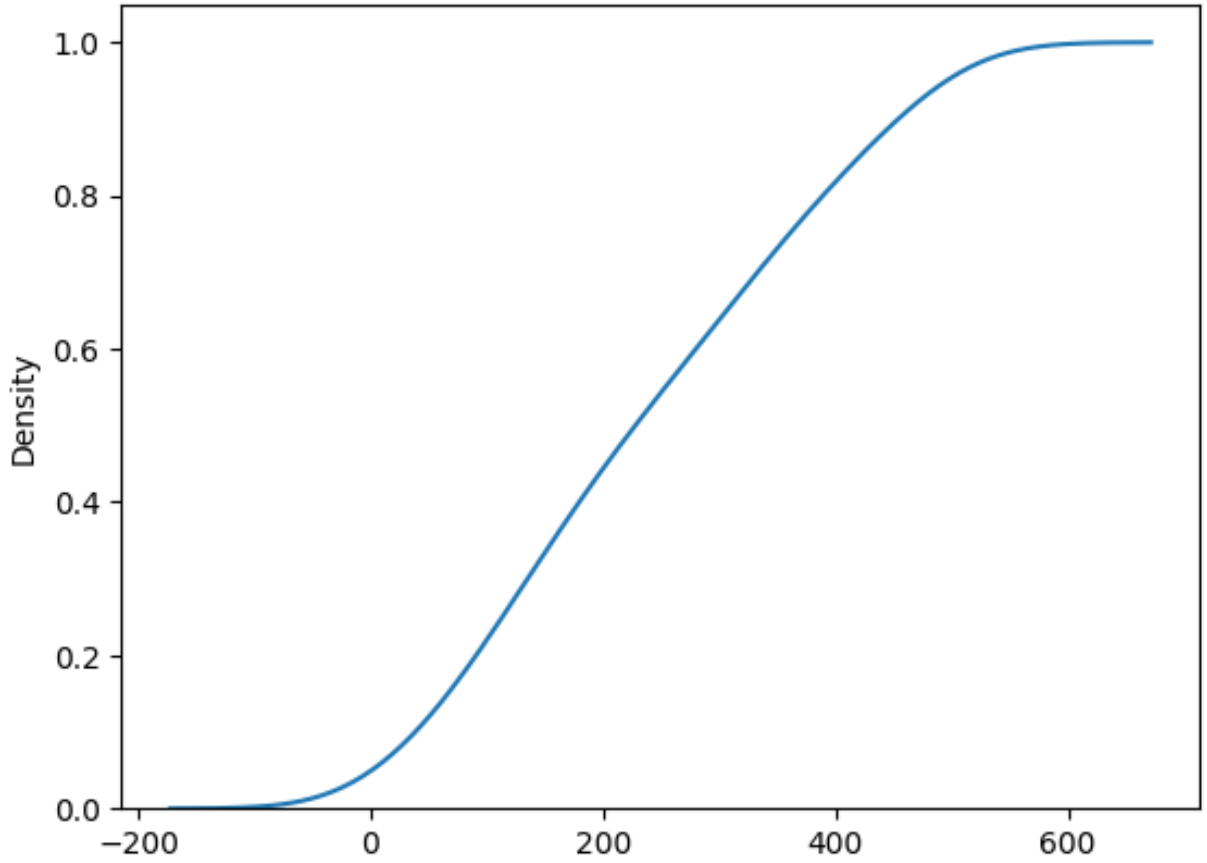
```
import matplotlib.pyplot
matplotlib.pyplot.hist(rand_list, histtype='step', cumulative=True, bins=len(rand_list))
```



Math statistics in Python

Empirical cumulative distribution function.

```
import seaborn as sns
sns.kdeplot(rand_list, cumulative=True)
```



Math statistics in Python

Calculating the Mean $\overline{x}$.

You can use the **mean(data)** function to calculate the mean of some data:

```
mean_value=statistics.mean(rand_list)
print(mean_value)
```

```
237.66
```

The **mean** is a good indicator of the average, but a few extreme values can result in a mean that is far from the actual central location. In some cases, it is more desirable to determine the most frequent data point in a data set. The **mode() (M_0^*)** function returns the most common data point from discrete numeric and non-numeric data. Mode is the most frequently occurring value of a sample of observations (i.e. the value that corresponds to the maximum probability density). A sample may have one mode (unimodal distribution), two (bimodal distribution), etc.

```
mode_value=statistics.mode(rand_list)
print(mode_value)
```

```
169
```

Math statistics in Python

Calculating the median M_e^*.

Relying on mode to calculate the central value can be a little misleading. This will always be the most popular data point, regardless of all other values in the data set. Another way to determine the central location is to use the **median()** function. It returns the median value of the given numeric data by calculating the average of the two middle points, if necessary. If the number of data points is odd, the function returns the middle point. If the number of data points is even, it returns the average of the two median values.

```
median_value=statistics.median(rand_list)
print(median_value)
```

```
219.5
```

Math statistics in Python

Important:

the mean is very sensitive to outliers, i.e. the presence of even one observation in the sample that differs significantly from the others can greatly affect the value of the sample mean. The median, unlike the mean, is less sensitive to outliers, but it does not provide any information about the spread of the data. For a symmetrical distribution (e.g. normal), the median and mean coincide.

Math statistics in Python

Calculating the median M_e^*.

The **median** is a sample value such that half of the observations are no less than it, and the other half are no more. If the sample is ordered from the largest value to the smallest, the median will divide it in half: if the set contains an even number of observations, the median is the average value of the two middle observations.

```
median_value=statistics.median(rand_list)
print(median_value)
```

```
219.5
```

Math statistics in Python

Practice intuitively determining descriptive statistics on small samples without having to calculate them in Python. All samples are sorted for convenience.

Sample:

1 2 3 4 5 6 7

Compare the mean with the median.

Math statistics in Python

Let's calculate descriptive statistics for a data set (pd.DataFrame) about the players of the main squad of the football club.

```python
import pandas as pd

df = pd.DataFrame({'player_number': [10, 8, 12, 22, 36, 7, 1, 20, 9, 4, 14],
 'height': [168, 176, 178, 191, 185, 183, 185, 179, 169, 183, 167],
 'weight': [76, 77, 79, 81, 82, 79, 74, 84, 73, 71, 68],
 'score': [95, 86, 94, 96, 95, 95, 89, 83, 99, 78, 82]})

print(df)
```

	player_number	height	weight	score
0	10	168	76	95
1	8	176	77	86
2	12	178	79	94
3	22	191	81	96
4	36	185	82	95
5	7	183	79	95
6	1	185	74	89
7	20	179	84	83
8	9	169	73	99
9	4	183	71	78
10	14	167	68	82

Math statistics in Python

We can calculate the mean, median, and mode separately for each column of the Pandas dataframe, for example, for the player's height variable:

```
mean_height = df['height'].mean()
median_height = df['height'].median()
mode_height = df['height'].mode()

print(mean_height, median_height, mode_height, sep = "\n")
```

```
178.54545454545453
179.0
0    183
1    185
Name: height, dtype: int64
```

Math statistics in Python

Or we can get descriptive statistics for all columns with numeric data by applying the .mean(), .median(), and .mode() methods to the entire dataframe at once:

```
mean = df.mean()
print(mean)
```

```
player_number     13.000000
height           178.545455
weight            76.727273
score             90.181818
dtype: float64
```

Math statistics in Python

These methods can also be applied to data packed into separate series (pd.Series), for example:

```
weight = pd.Series([76, 77, 79, 81, 82, 79, 74, 84, 73, 71, 68])

mean_weight = weight.mean()
print(mean_weight)
```

```
76.72727272727273
```

Math statistics in Python

We can also calculate the mean and median using the Numpy library and the mode using the Scipy library:

```
import numpy as np
from scipy import stats


mean_height = np.mean(df['height'])

median_height = np.median(df['height'])

mode_height = stats.mode(df['height'])

print(mean_height, median_height, median_height, sep = "\n")
```

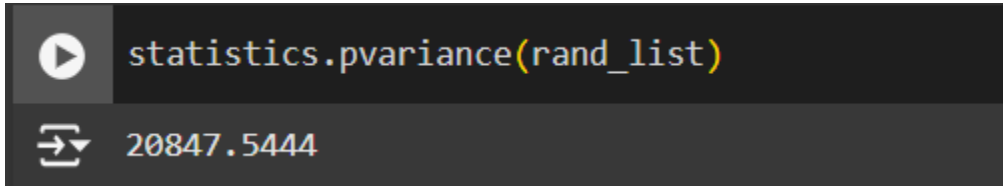
```
178.54545454545453
179.0
179.0
```

Math statistics in Python

Measuring data dissemination

Determining how much the data deviates from the typical or average value of a data set is just as important as calculating the center or average value itself. The **statistics** module has four different functions that help us calculate this spread of data.

You can use the **pvariance(data, mu=None)** function to calculate the variance of a data set (**sample variance s^2**). The second argument is optional in this case. The **mu** value, if provided, must be equal to the mean of the data. The mean is calculated automatically if the value is missing. This function is useful when you want to calculate the variance of the entire population.



Math statistics in Python

To calculate **sample standard deviation** ($s = \sqrt{s^2}$), you can use the function **stdev(data, xBar=None)**.

```
statistics.stdev(rand_list)
```

```
145.1141814453903
```

Covariation

Covariance is a measure of the linear relationship between two random variables, i.e. a measure of their joint variability:

- positive covariance means that as one variable increases, the other also tends to increase (and vice versa)
- negative covariance means that as one variable increases, the other tends to decrease (and vice versa).

The covariance of two variables x and y from samples of equal size n is calculated as:

$$cov(x, y) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - M_x)(y_i - M_y)$$

where M_x, M_y sample means for x and y values.

Covariation

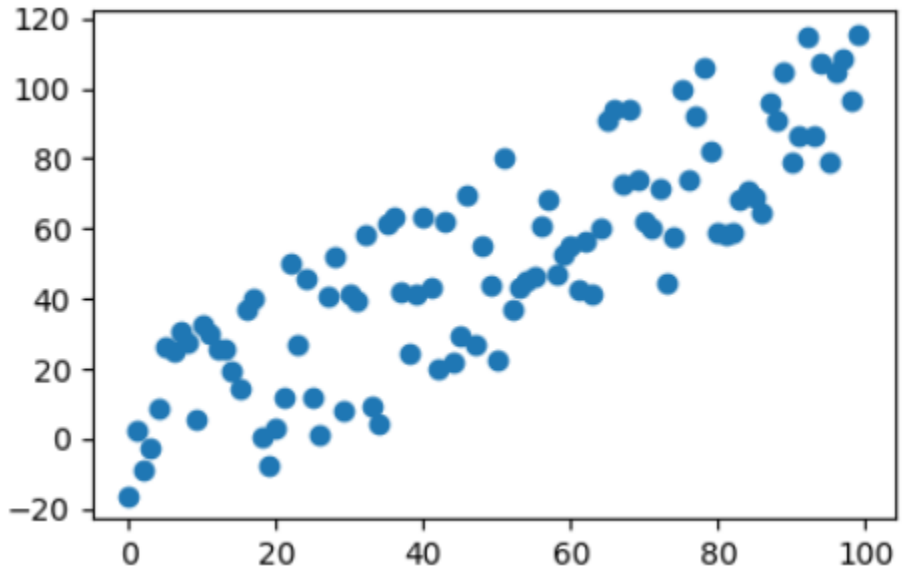
The covariance itself can take any positive or negative values, so to judge the degree of relationship between the quantities, we normalize it by the product of the standard deviations of the random variables. As a result, the resulting correlation value will take a value from -1 (strict negative linear dependence) to 1 (strict positive linear dependence). A correlation value of 0 indicates the absence of a linear dependence between the random variables. By performing additional transformations, we can obtain another version of the Pearson correlation formula:

$$corr(x, y) = \frac{\sum_{i=1}^{n} (x_i - M_x)(y_i - M_y)}{\sqrt{\sum_{i=1}^{n} (x_i - M_x)^2} \sqrt{\sum_{i=1}^{n} (y_i - M_y)^2}}$$

Important: Neither correlation nor covariance reflects a causal relationship between variables. A classic example of incorrect use of correlation analysis is, for example, inferring the existence of a causal relationship based on a high correlation between the damage caused by a fire and the number of firefighters involved in extinguishing it. Also, correlation by itself does not distinguish between possible indirect effects of quantity X on quantity Y, for example, through a third quantity Z.

Covariation

Try to roughly estimate the correlation coefficient from the graph of the dependence of the quantities:



- from 70% to 100% inclusive
- from 40% to 70% inclusive
- from 0% to 40% inclusive
- from 0% to -40% inclusive
- from -40% to -70% inclusive
- from -70% to 100% inclusive

Covariation

The study found that the divorce rate per capita in Maine was highly significantly correlated (99.26%) with the average margarine consumption. What this means:

1. Rising Divorce Rates in Maine Lead to Increased Margarine Consumption
2. No Conclusion of Causality
3. Rising Margarine Consumption in Maine Leads to Increased Divorce Rates

Covariation

The Pearson correlation between two random variables x and y is a normalized version of the covariance and is calculated as:

$$corr(x, y) = \frac{cov(x, y)}{sd_x sd_y}$$

where sd_x, sd_y are sample standard deviations of the values x and y

Covariation

Let's look at how to calculate the Pearson correlation coefficient in Python using the scipy library.Let's estimate the correlation coefficient between the value of the banking sector's net interest margin (NIM) and the value of the average annual interbank market interest rate (rate) over 11 years using .pearsonr:

```
[ ]  from scipy import stats

     # data
     NIM = pd.Series([7.6, 7.9, 8.3, 7.2, 6.9, 7.9, 7.4, 7.8, 5.9, 7.1, 6.8])
     rate = pd.Series([10, 12, 12, 8, 8, 7.5, 7.5, 7.5, 6.5, 7, 7])

     # we calculate the Pearson correlation value between two data series NIM and rate
     corr = stats.pearsonr(NIM,rate).statistic
     print(corr)
```

```
0.6821733938700639
```

Covariation

In pandas, we can calculate pairwise correlations between a large number of different variables (for example, contained in the columns of a dataframe).

We will estimate the correlations between the values of the banking sector net interest margin (NIM), the average annual interbank market rate (rate), and annual GDP growth (GDP_growth) over 11 years, and present the result as a correlation matrix (3 by 3):

```python
import pandas as pd

df = pd.DataFrame({'NIM': [7.6, 7.9, 8.3, 7.2, 6.9, 7.9, 7.4, 7.8, 5.9, 7.1, 6.8],
 'rate': [10, 12, 12, 8, 8, 7.5, 7.5, 7.5, 6.5, 7, 7],
 'GDP_growth': [2.5, 1.8, 3.1, 1.9, 2.4, 2.8, 1.0, 3.2, 2.1, 2.2, 0.5]})

corr_matrix = df.corr().round(3)
print(corr_matrix)
```

```
               NIM   rate  GDP_growth
NIM          1.000  0.682       0.445
rate         0.682  1.000       0.281
GDP_growth   0.445  0.281       1.000
```

Quantiles and quartiles

Quantiles are values that a given random variable does not exceed with a fixed probability. For example, the 0.35 quantile is a distribution value such that 35% of all values are less than or equal to it, and the rest (65%) are greater than it. If the probability is given as a percentage, the quantile is called a percentile, i.e. the 0.35 quantile is the 35th percentile.

There are three quantile values called **quartiles**:

-0.25 quantile (first or lower quartile, also known as the 25th percentile),

-0.5 quantile (second quartile, also known as the 50th percentile or median of the distribution),

-0.75 quantile (third or upper quartile, also known as the 75th percentile).Quartiles divide the sample into 4 equal parts (quarters).

Quantiles and quartiles

What is the interval of values that includes 95% of all observations in the sample? Select one option from the list

1. 0 to 95th percentile
2. 5th percentile to 100
3. 2.5th percentile to 97.5th percentile

Quantiles and quartiles

Using the example of a dataset containing data on the return on equity of corporate banks (ROE), their cost of funding (COF) and credit rating (rating), we will learn how to calculate quantiles and build a Box Plot.

We can calculate all the necessary quantiles at once for the dataset in question, for example, a column with return on equity (ROE) values:

```
import pandas as pd

df = pd.DataFrame({'ROE': [7.4, 7.8, 5.9, 7.1, 6.8, 7.6, 7.9, 8.3, 7.2, 6.9, 7.9],
 'COF': [6.3, 4.2, 6.2, 8.1, 8.4, 7.5, 6.5, 7.8, 6.9, 7.1, 7.2],
 'rating': ['AAA', 'BB+', 'BB+', 'AAA', 'BBB-', 'AA', 'AAA', 'AAA', 'BBB-', 'AAA', 'BB+']})

# list of quartiles
Q = [0.25, 0.5, 0.75]

print(df['ROE'].quantile(Q))
```

```
0.25    7.00
0.50    7.40
0.75    7.85
```