(01) Charakteristiky polohy a variability, grafické znázornenie dát

1. Priemerný počet bodov z písomky I.

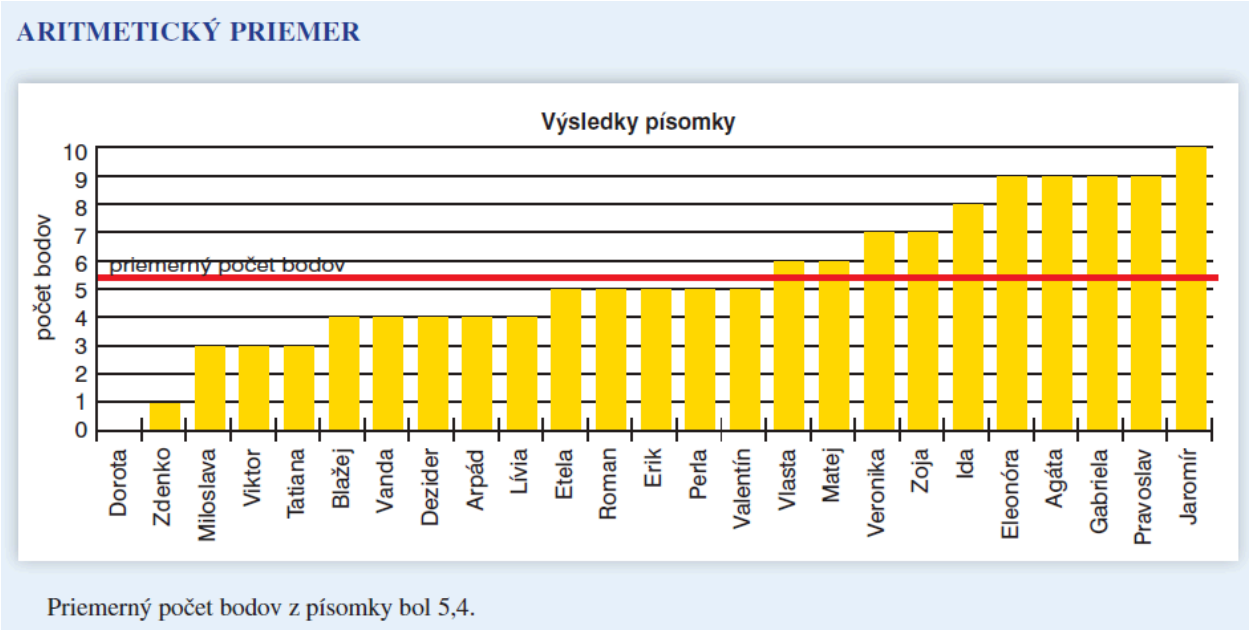
Všeobecne známa situácia – píšeme písomku a zaujíma nás priemerný počet bodov (**aritmetický priemer**).

- Písomku písalo päť študentov, ich bodový zisk bol 90,65, 74, 39, 52 - priemerný počet bodov je (90 + 65 + 74 + 39 + 52)/5 = 62.
- Päťdesiat študentov písalo na za začiatku cvičenia rozcvičku, z ktorej sa dá získať 0, 1 alebo 2 body. Výsledky sú zhrnuté: 12 študentov má 0 bodov, 21 študentov 1 bod, 17 študentov 2 body. Priemerný počet bodov je (12×0 + 21×1 + 17×2)/50 = 1.1. Kvôli možnému zápisu (12/50)×0 + (21/50)×1 + (17/50)×2 = 1.1 môžeme hovoriť **o váženom priemere** čísel 0, 1, 2 s váhami 12/50, 21/50, 17/50.

2. Priemerný počet bodov z písomky II.

Obrázok:
Zbaněk Kubáček: Matematika pre 3. ročník gymnázia a 7. ročník gymnázia s osemročným štúdiom, 1. časť

Príklad zobrazenia výsledkov písomky: zoradíme študentov podľa počtu bodov, spravíme stĺpcový graf. Vyznačíme priemer, vidíme, kto mal podpriemerný a kto nadpriemerný počet bodov.



V Pythone (časť predchádzajúcich dát s rovnakým priemerom):

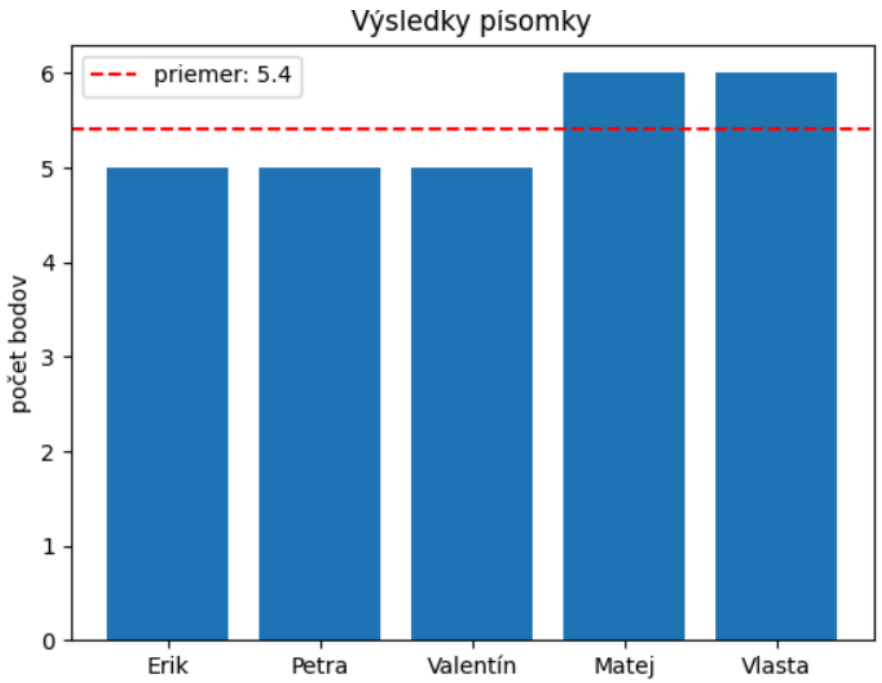
```python
import matplotlib.pyplot as plt
import numpy as np

studenti=["Erik", "Matej", "Petra", "Valentín", "Vlasta"]
body=[5, 6, 5, 5, 6]

priemer=np.mean(body)

usporiadane=sorted(zip(body, studenti))
usporiadane_body, usporiadane_studenti=zip(*usporiadane)

plt.bar(usporiadane_studenti, usporiadane_body)
plt.axhline(priemer, color="red", linestyle="--", label=f'priemer:
{priemer:.1f}')
plt.ylabel("počet bodov")
plt.legend()
plt.title("Výsledky písomky")
plt.show()
```



3. Priemerný plat I. + čo je to medián

Priemer nemusí vždy dobre vystihovať dáta, príklad z knihy *Darrel Huff: How to lie with statistics*

You, I trust, are not a snob, and I certainly am not in the real-estate business. But let's say that you are and I am and that you are looking for property to buy along a road that is not far from the California valley in which I live.

Having sized you up, I take pains to tell you that the average income in this neighborhood is some $15,000 a year. Maybe that clinches your interest in living here; anyway, you buy and that handsome figure sticks in your mind. More than likely, since we have agreed that for the purposes of the moment you are a bit of a snob, you toss it in casually when telling your friends about where you live.

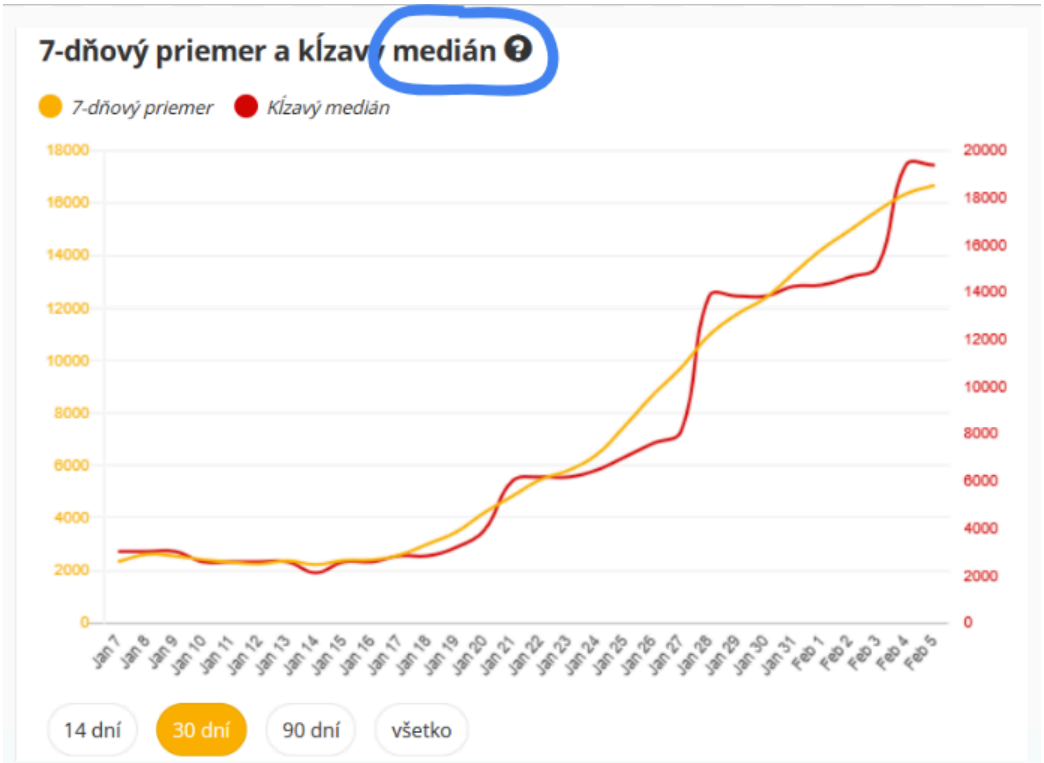
A year or so later we meet again. As a member of some taxpayers' committee I am circulating a petition to keep the tax rate down or assessments down or bus fare down. My plea is that we cannot afford the increase: After all, the average income in this neighborhood is only $3,500 a year. Perhaps you go along with me and my committee in this—you're not only a snob, you're stingy too—but you can't help being surprised to hear about that measly $3,500. Am I lying now, or was I lying last year?

You can't pin it on me either time. That is the essential beauty of doing your lying with statistics. Both those figures are legitimate averages, legally arrived at. Both represent the same data, the same people, the same incomes. All the same it is obvious that at least one of them must be so misleading as to rival an out-and-out lie.

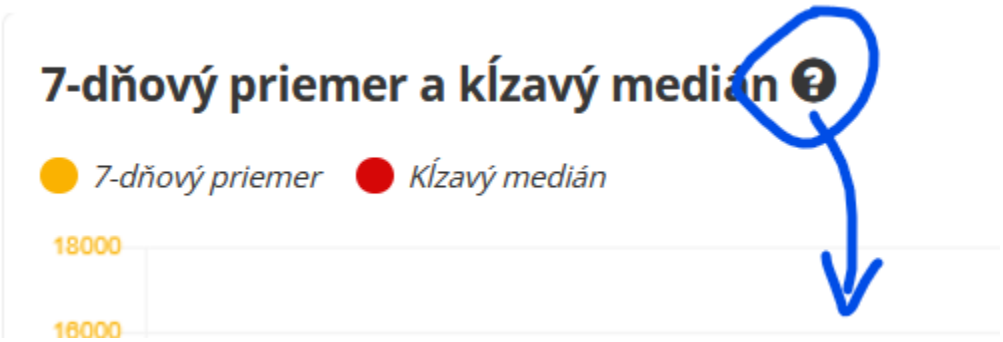
My trick was to use a different kind of average each time, the word "average" having a very loose meaning. It is a trick commonly used, sometimes in innocence but often in guilt, by fellows wishing to influence public opinion or sell advertising space. When you are told that something is an average you still don't know very much about it unless you can find out which of the common kinds of average it is—mean, median, or mode.

The $15,000 figure I used when I wanted a big one is a mean, the arithmetic average of the incomes of all the families in the neighborhood. You get it by adding up all the incomes and dividing by the number there are. The smaller figure is a median, and so it tells you that half the families in question have more than $3,500 a year and half have less. I might also have used the mode, which is the most frequently met-with figure in a series.

4. Medián - príklady a výpočet



Pamätáme si z obdobia kovidu:



Pri uvoľňovaní opatrení prijatých v boji proti pandémii na Slovensku sa počíta kĺzavý medián z denného počtu nových nakazených za uplynulý týždeň očistený o nových nakazených v karanténnych centrách.

Príklad: Ako vypočítať medián?
Ak máme takéto čísla: 10, 15, 13, 81, 9, 21, 2
a usporiadame ich vzostupne : 2, 9, 10, 13, 15, 21, 81

potom medián je číslo 13.

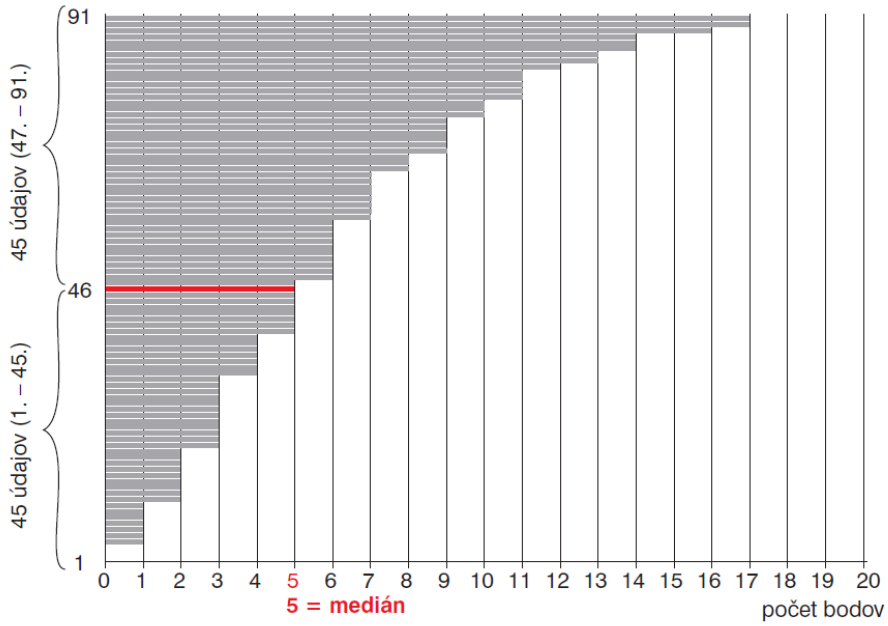
Zavrieť

https://web.archive.org/web/20220207054440/https://spravy.pravda.sk/domace/clanok/615989-online-slovensko-patri-do-prvej-desiatky-v-pocte-nakaz-na-milion-ludi/

- Medián rozdeľuje dáta na “lepšiu polovicu” a “menšiu polovicu”.
- **Aspoň polovica hodnôt je menšia alebo rovná mediánu, aspoň polovica hodnôt je väčšia alebo rovná mediánu.**
- Hodnoty usporiadame:
    - V prípade nepárneho počtu hodnôt je mediánom hodnota v strede
    - V prípade párneho počtu hodnôt vyhovuje definícii mediánu každé číslo medzi dvoma prostrednými – berie sa ich priemer

Príklad s písomkou – kto je v lepšej polovici a kto je v horšej polovici:
Obrázok: Kubáček: Matematika

MEDIÁN



- kto získal viac ako 5 bodov, je iste v prvej (lepšej) polovici,
- kto získal menej ako 5 bodov, je iste v druhej (horšej) polovici
- najviac polovica žiakov má menej ako 5 bodov, najviac polovica má viac ako 5 bodov

Športové výsledky – čas potrebný na umiestnenie v prvej polovici štartového poľa
Obrázok: Kubáček: Matematika

MEDIÁN – hodnota, ktorá súbor rozdelí na *lepšiu a horšiu polovicu*

Výsledková listina

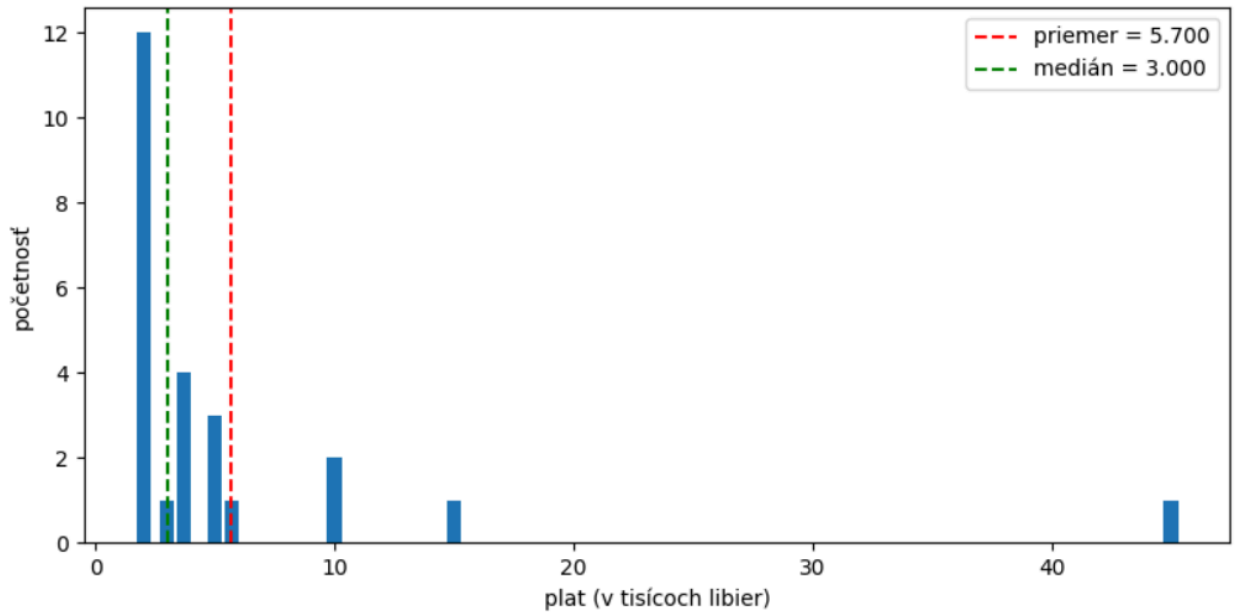
1.	MATÚŠ Ján	00:37:37
2.	KOCIAN Konštantín	00:38:36
3.	ŠVEC Milan	00:38:43
⋮		
30.	ZUŠČÁK František	00:49:21
31.	SCHULZ Eric	00:49:24
⋮		
60.	FERREIRA Framlisco	01:02:44
61.	YILDIZ Deniz Can	01:05:40
62.	KUSTAN Orava	01:08:12

Na umiestenie v prvej polovici štartového poľa behu na 12 km bolo potrebné mať kvalifikačný čas lepší ako 49' 24".

5. Priemerný plat II.

Numerický príklad z knihy *How to lie with statistics* (na nasledujúcej strane obrázok z knihy)

Medián nie je citlivý na extrémne hodnoty, ak by v našich dátach bola namiesto hodnoty 45 hocijaká iná hodnota väčšia ako 3, medián by sa nezmenil (priemer áno).




```
import numpy as np
plat=[45, 15] + [10]*2 + [5.7] + [5]*3 + [3.7]*4 + [3] + [2]*12
plat=[x*1000 for x in plat]
print(np.mean(plat))
print(np.median(plat))
```

```
5700.0
3000.0
```

$45,000

$15,000

$10,000

← ARITHMETICAL AVERAGE

$5,700

$5,000

$3,700

← MEDIAN (the one in the middle 12 above him, 12 below)

$3,000

← MODE (occurs most frequently)

$2,000

6. Modus

Priemer ani medián nemusia vystihovať “typické” pozorovania z dátového súboru.

Výsledky študentov AIN z predmetu Programovanie (1) podľa
https://sluzby.fmph.uniba.sk/infolist/SK/1-AIN-130.html (19. 9. 2025)

Prehľad hodnotení:
A: 29.29% B: 11.45% C: 10.44% D: 7.41% E: 12.12% FX: 29.29%
Celkový počet hodnotení: 1188

Zhoda percent pre A a FX nie je daná zaokrúhľovaním:

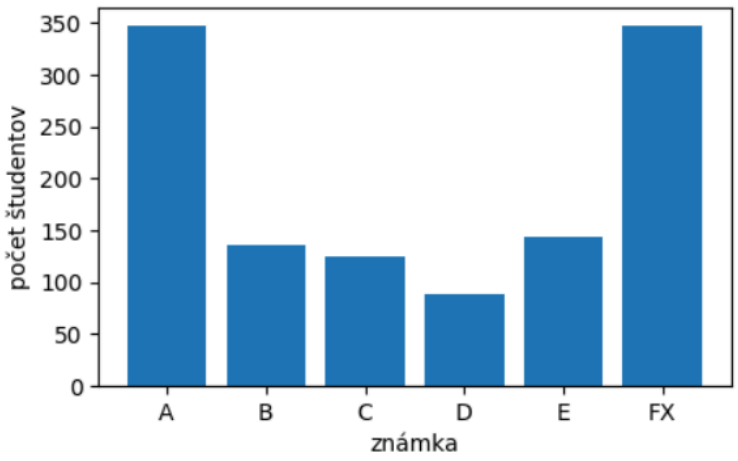
```
1188*0.2929
```

```
347.9652
```

```
[round(n/1188, 4) for n in range(347, 350)]
```

```
[0.2921, 0.2929, 0.2938]
```

```
pocty=[348, 136, 124, 88, 144, 348]
znamky=["A", "B", "C", "D", "E", "FX"]
plt.figure(figsize=(5, 3))
plt.bar(znamky, pocty)
plt.xlabel("známka")
plt.ylabel("počet študentov")
plt.show()
```



Po prevedení na číselné hodnoty je priemerná známka medzi C a D:

```
znamky_numericke=[1, 1.5, 2, 2.5, 3, 4]
print(round(np.average(znamky_numericke, weights=pocty), 3))
```

```
2.394
```

- **Modus** je najčastejšie sa vyskytujúca hodnota v dátach
- Ang. *mode,* vyznačený na obrázku z knihy na predchádzajúcej strane
- V prípade známok z programovania dva modusy: 1 a 4.
- Ak ich je príliš veľa, nie je to užitočná informácia

Príklad z rovnakej knihy:

Altogether too much housing, for instance, has been planned to fit the statistically average family of 3·6 persons. Translated into reality this means three or four persons, which, in turns, means two bedrooms. And this size family, 'average' though it is, actually makes up a minority of all families. 'We build average houses for average families,' say the builders – and neglect the majority that are larger or smaller. Some areas, in consequence of this, have been over-built with two-bedroom houses, underbuilt in respect to smaller and larger units. So here is a statistic whose misleading incompleteness has had costly consequences. Of it a large public-health group has said: 'When we look beyond the arithmetical average to the actual range which it misrepresents, we find that the three-person and four-person families make up only 45 per cent of the total. Thirty-five per cent are one-person and two-person; 20 per cent have more than four persons.'

Common sense has somehow failed in the face of the convincingly precise and authoritative 3·6. It has somehow outweighed what everybody knows from observation: that many families are small and quite a few are large.

V Pythone:

```
import statistics

data=[1, 2, 2, 3]
modus=statistics.mode(data)
print(modus)
```

```
2
```

```
data=[1, 2, 2, 3, 3]
modus=statistics.mode(data)
print(modus)
```

```
2
```

```
data=[1, 2, 2, 3, 3]
modus=statistics.multimode(data)
print(modus)
```

```
[2, 3]
```

7. Tabuľka početností

Videli sme ju v prípade známok z programovania.

- Absolútne početnosti – počet výskytov
- Relatívne početnosti – napr. 0.15 znamená 15 percent

Príklad pre dáta zo stránky

```
from google.colab import files
uploaded = files.upload()
df = pd.read_csv("wo_men.csv")
df.head()
```

Browse... No files selected. Upload widget is only available wl

```
Saving wo_men.csv to wo_men.csv
```

	time	sex	height	shoe_size
0	04.10.2016 17:58:51	woman	160.0	40.0
1	04.10.2016 17:58:59	woman	171.0	39.0
2	04.10.2016 18:00:15	woman	174.0	39.0
3	04.10.2016 18:01:17	woman	176.0	40.0
4	04.10.2016 18:01:22	man	195.0	46.0

```
men_shoe_sizes = df[df['sex'] == 'man']['shoe_size']
count_table = men_shoe_sizes.value_counts().sort_index()
print(count_table)
```

```
shoe_size
41.0    2
42.0    5
44.0    7
46.0    2
48.0    1
50.0    1
Name: count, dtype: int64
```

- `value_counts()` dá absolútne početnosti (vidíme hore)
- `value_counts(normalize=True)` dá relatívne početnosti

8. Použitie relatívnych početností na odhadovanie pravdepodobností

Marilyn vos Savant bola istý čas v Guinessovej knihe rekordov ako človek najvyšším IQ. Vo svojej rubrike odpovedala na rôzne otázky čitateľov, niekedy to boli matematické úlohy. Jedna z nich:

A high school student who hadn't opened his American history book in weeks was dismayed to walk into class and be greeted with a pop quiz. It was in the form of two lists, one naming the 24 presidents in office during the 19th century in alphabetical order and another list noting their terms in office, but scrambled. The object was to match the presidents with their terms. The completely clueless student had to guess every time. On average, how many did he guess correctly?

- *"Guess"* budeme interpretovať tak, že každému prezidentovi priradí práve jedno obdobie, pričom každé takéto priradenie má rovnakú pravdepodobnosť.

- *"On average"* budeme interpretovať tak, že nás zaujíma, aký priemerný počet bodov by študent získal, ak by používal takúto stratégia na veľa písomiek tohto typu.

```
import random

def pisomka(pocet_otazok):
  odpovede=range(1, pocet_otazok + 1)
  tip=random.sample(odpovede, pocet_otazok)
  pocet_spravnych=sum(a == b for a, b in zip(odpovede, tip))
  return pocet_spravnych
```

- Parametre **random.sample –** z čoho a koľko hodnôt vyberáme (bez návratu)
- Typický priebeh simulácií:
  - náhodnú situáciu veľakrát opakujeme:

```
pocet_prezidentov=24
pocet_simulacii=10**5
simulacia=[pisomka(pocet_prezidentov) for _ in range(pocet_simulacii)]
```

  - a potom sa vo výsledkoch pozrieme na to, čo nás zaujíma:

```
import pandas as pd
import numpy as np

# zo zadania
print(np.mean(simulacia))

# odhady pravdepodobnosti
rel_pocetnosti=pd.Series(simulacia).value_counts(normalize=True).sort_inde
x()
print(rel_pocetnosti)
```

```
# zo zadania
print(np.mean(simulacia))
```

```
0.99974
```

```
# odhady pravdepodobnosti
rel_pocetnosti=pd.Series(simulacia).value_counts(normalize=True).sort_index()
print(rel_pocetnosti)
```

```
0    0.36859
1    0.36683
2    0.18398
3    0.06186
4    0.01517
5    0.00293
6    0.00050
7    0.00010
8    0.00002
9    0.00002
Name: proportion, dtype: float64
```

Ak chceme pravdepodobnosti pre všetky hodnoty 0-24:

```
mozne_hodnoty=range(0, pocet_prezidentov + 1)
rel_pocetnosti2 = pd.Series(simulacia).value_counts(normalize=True).reindex(mozne_hodnoty, fill_value=0)
print(rel_pocetnosti2)
```

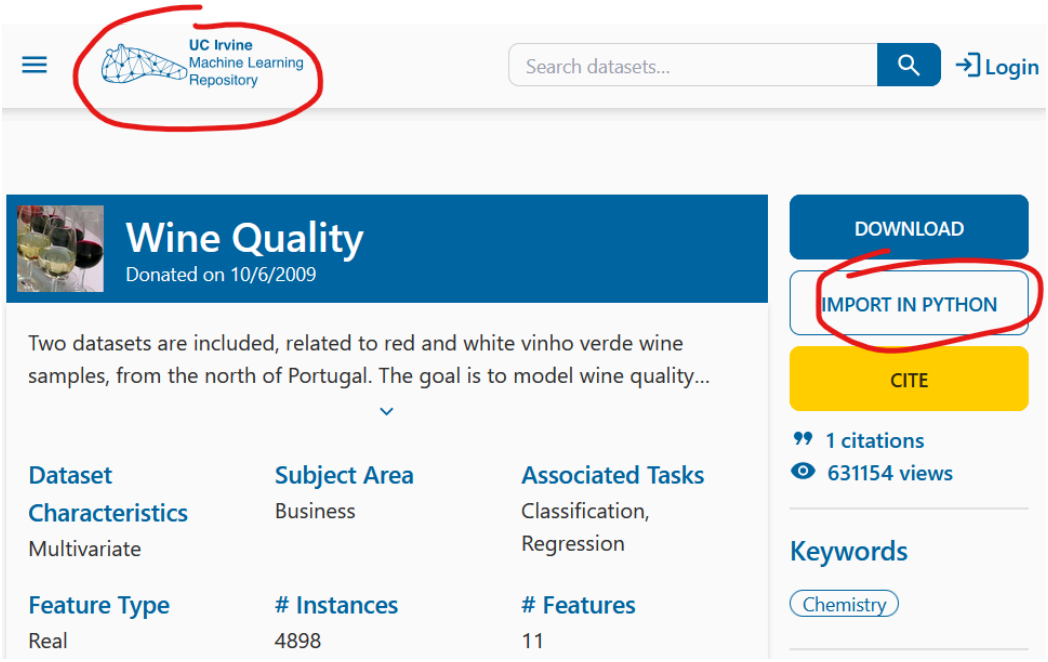
```
0     0.36859
1     0.36683
2     0.18398
3     0.06186
4     0.01517
5     0.00293
6     0.00050
7     0.00010
8     0.00002
9     0.00002
10    0.00000
11    0.00000
12    0.00000
13    0.00000
14    0.00000
15    0.00000
16    0.00000
17    0.00000
18    0.00000
19    0.00000
20    0.00000
21    0.00000
22    0.00000
23    0.00000
24    0.00000
Name: proportion, dtype: float64
```

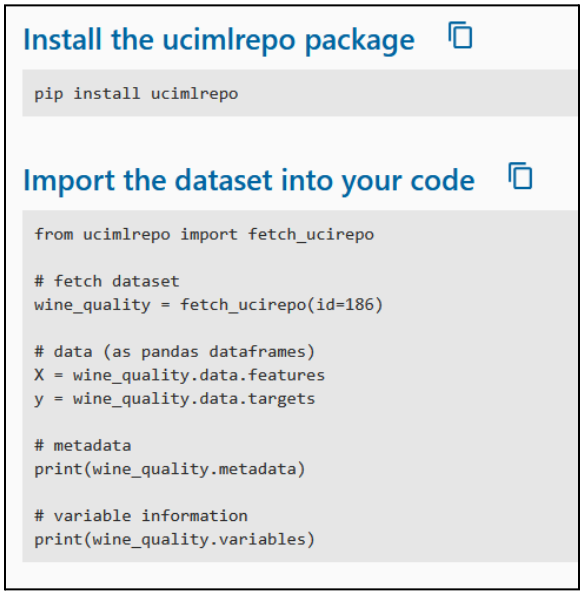
9. Zobrazenie dát s väčším počtom rôznych hodnôt, histogram

Tabuľka ani stĺpcový graf nie je prehľadný - používa sa napríklad **histogram.** Namiesto početností jednotlivých hodnôt sa zobrazujú početnosti hodnôt zo zvolených intervalov.

Príklad s dátami zo stránky https://datalab-12.ics.uci.edu/dataset/186/wine_quality

Pohodlné načítanie do Pythonu:

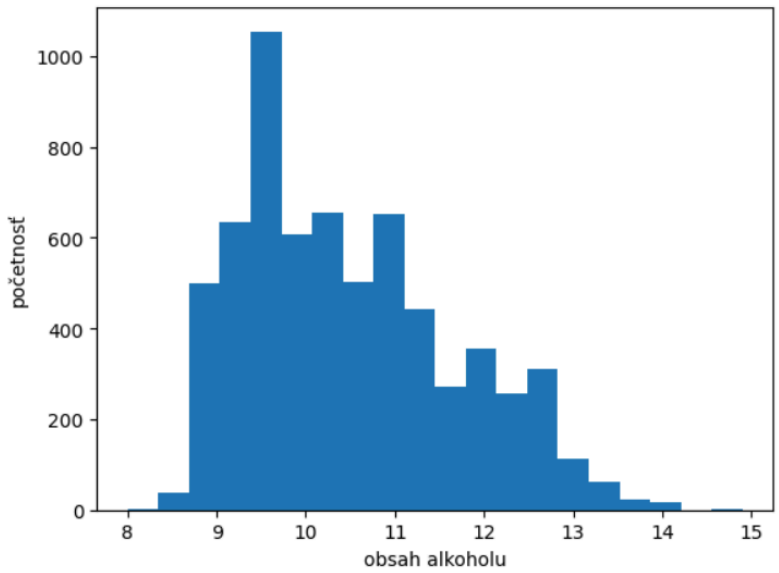




```
alkohol = X["alcohol"]

plt.hist(alkohol, bins=20)

plt.xlabel('obsah alkoholu')
plt.ylabel('početnosť')
plt.show()
```



10. Kvantily

- **p-kvantil** dátového súboru je taká hodnota**,** že
  - p×100% hodnôt je menších alebo rovných tejto hodnote
  - (1-p)×100% hodnôt je väčších alebo rovných tejto hodnote
- Pre p=0,5 dostávame medián
- Analogicky ako v prípade mediánu, ak nie je kvantil určený jednoznačne, berieme priemer krajných bodov

Často sa používajú **kvartily:**

- 0.25-kvantil = 1. kvartil (Q1)
- 0.5-kvantil = medián (Q2)
- 0.75-kvantil = 3. kvartil (Q3)

V Pythone:

```
np.quantile(alkohol, [0.25, 0.5, 0.75])
```

```
import numpy as np

kvantily=np.quantile(alkohol, [0.25, 0.5, 0.75])

print("Q1:", kvantily[0])
print("Q2 (medián):", kvantily[1])
print("Q3:", kvantily[2])
```

```
Q1: 9.5
Q2 (medián): 10.3
Q3: 11.3
```

```
q95=np.quantile(alkohol, 0.95)
print(q95)
```

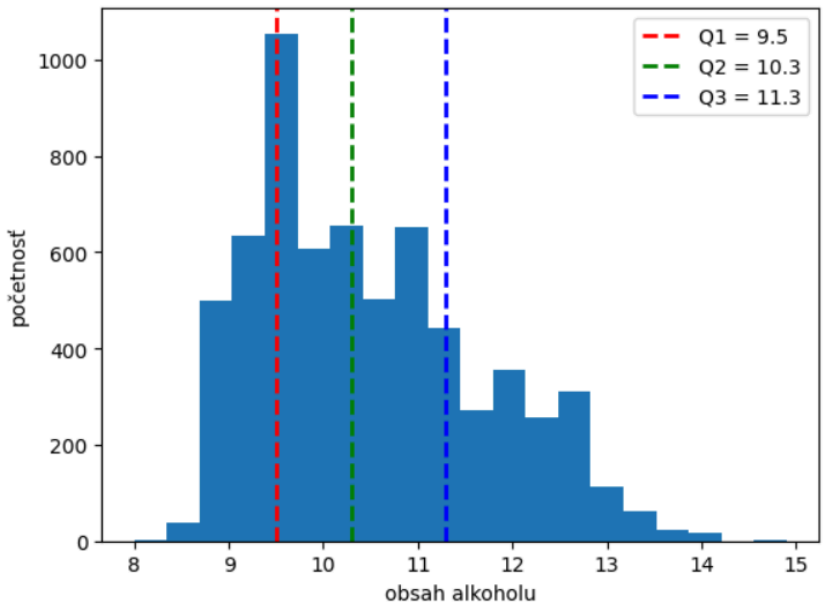
```
12.7
```

“Skúška správnosti”:

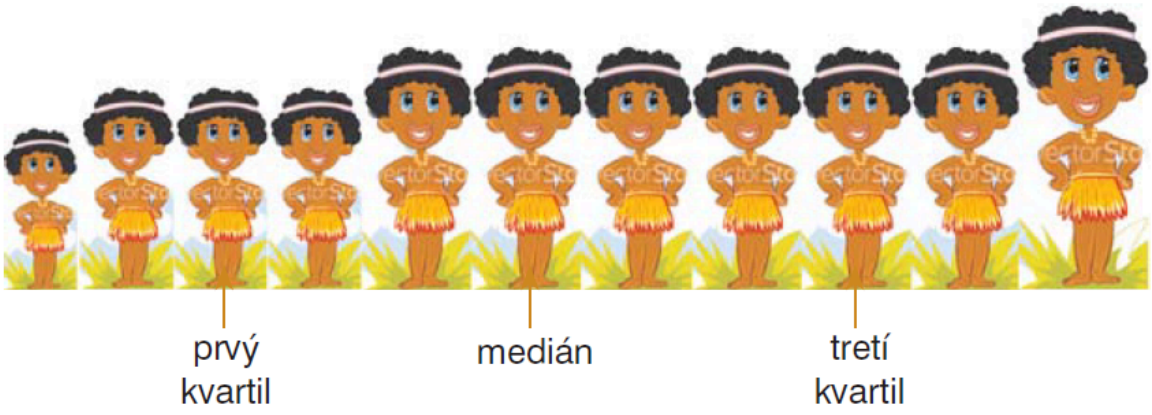
```
print(sum([x <= q95 for x in alkohol])/len(alkohol))
print(sum([x >= q95 for x in alkohol])/len(alkohol))
```

```
0.9552100969678313
0.0547945205479452
```

Grafické znázornenie v histograme:



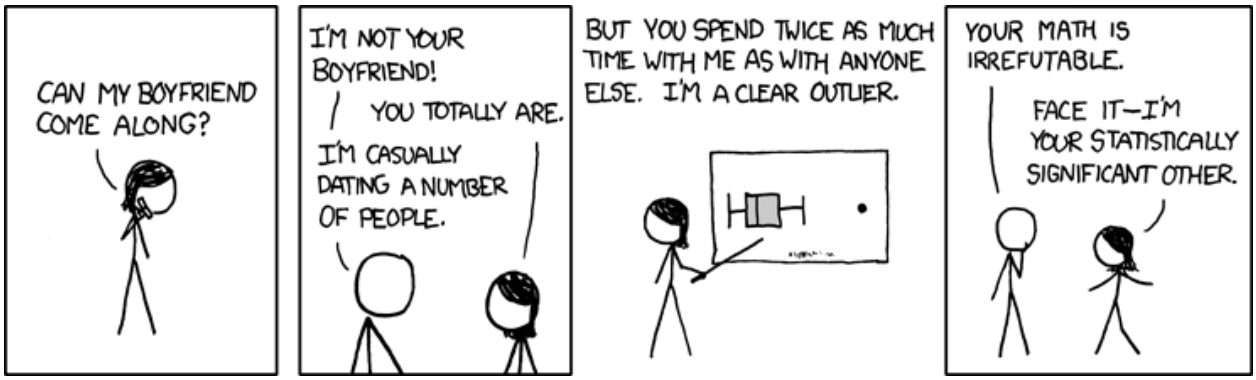
Obrázok + text:
Kubáček: Matematika



POUŽÍVANIE MEDIÁNU ZAVIEDOL V POSLEDNEJ ŠTVRTINE 19. STOROČIA F. GALTON, KTORÉHO MYŠLIENKY VÝZNAMNE OVPLYVNILI ROZVOJ MATEMATICKEJ ŠTATISTIKY. ANTROPOLÓGOM SKÚMAJÚCIM KMENE DOMORODCOV ODPORÚČAL, ABY NECHALI NÁČELNÍKOM ZORADIŤ SVOJICH ĽUDÍ PODĽA VÝŠKY A ZISTILI VÝŠKU V STREDE (MEDIÁN), V PRVEJ A TRETEJ ŠTVRTINE (PRVÝ A TRETÍ KVARTIL).

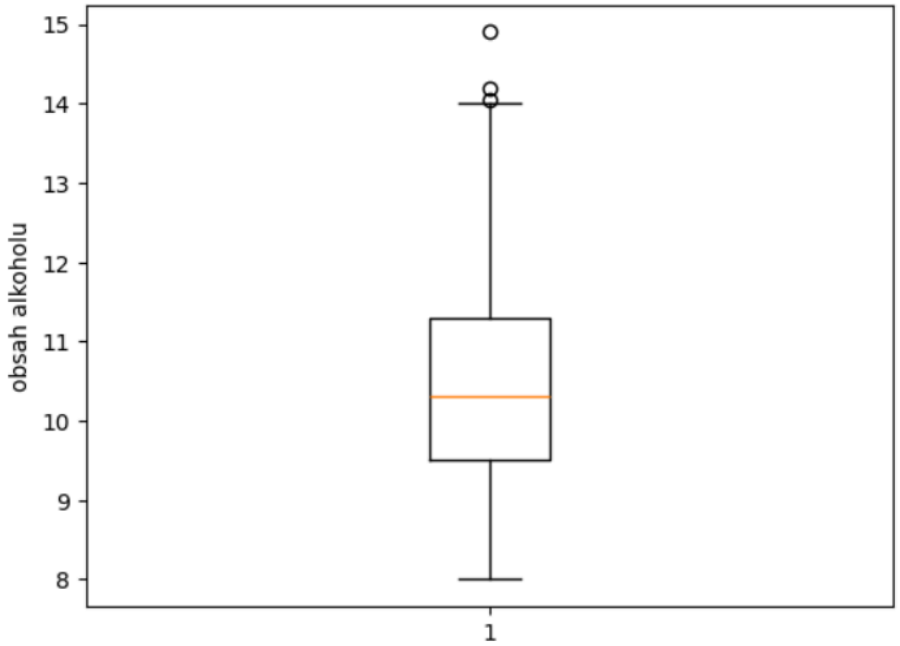
11. Zobrazenie dát s väčším počtom rôznych hodnôt, krabicový graf (boxplot)

https://xkcd.com/539/



Na treťom obrázku je **boxplot:**

```
plt.boxplot(alkohol)
plt.ylabel("obsah alkoholu")
plt.show()
```



- Krabica (box):
  - spodný okraj krabice: 1. kvartil (Q1)
  - čiara vo vnútri: medián (2. kvartil, Q2)
  - horný okraj krabice: 3. kvartil (Q3)
  - z toho sa spočíta medzikvartilové rozpätie (IQR) ako Q3 - Q1

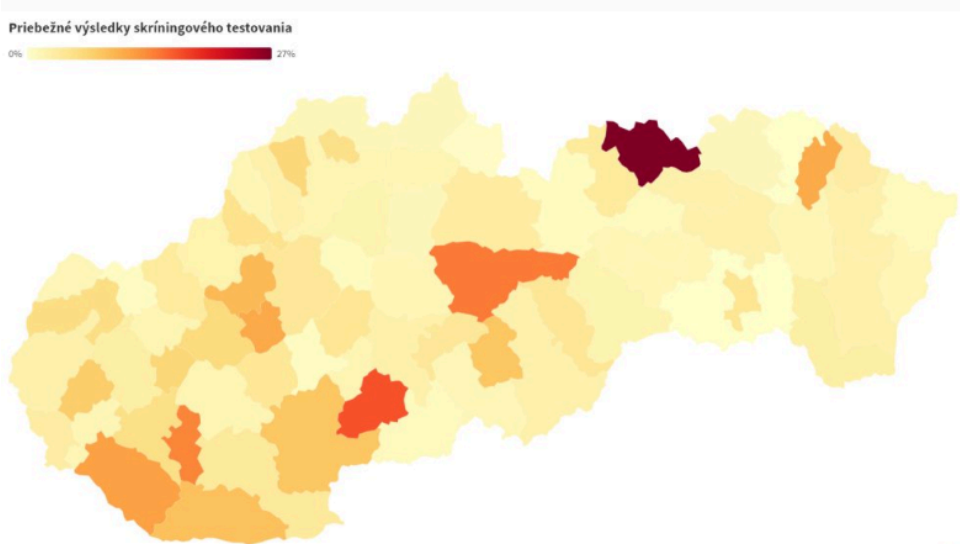
- "Fúzy" (whiskers):
  - siahajú k najnižším a najvyšším hodnotám, ktoré nie sú odľahlé (vysvetlenie v ďalšom bode)

- Odľahlé hodnoty (outliers, “outliery”):
  - pod dolnou alebo nad hornou hranicou
  - štandardná dolná hranica: Q1 − 1.5 × IQR
  - štandardná horná hranica: Q3 + 1.5 × IQR
  - obvykle sa kreslia ako malé krúžky

12. Odľahlé hodnoty (outliery)

- aj všeobecnejšie, nielen na boxplote, rôzne metódy na ich identifikáciu
- hodnoty, ktoré “sú iné” ako ostatné
- rôzny pôvod

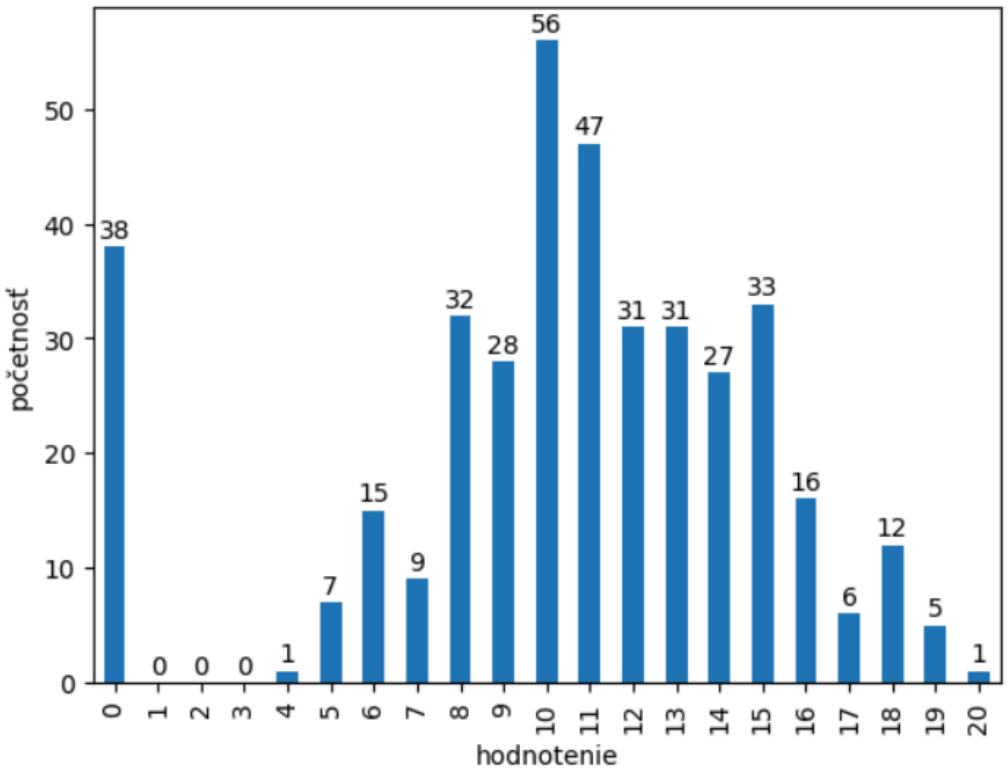
Príklad 1: kovid, testovanie



ÚVZ: Informácie o vysokej pozitivite (26,8 %) v okrese Stará Ľubovňa po štvrtom dni testovania v okrese Stará Ľubovňa teda nie sú správne, možno ide o dôsledok chybne spracovaných dát. Ak by sme vzali do úvahy len pozitivitu osôb z celkového počtu vykonaných laboratórnych vyšetrení antigénovým testom za uvedené obdobie, išlo by o 1,66 %.

Ľubovnianske noviny: Podľa našich informácií, chyba nastala na jednom z odberných miest. Pri zadávaní štatistík boli zamenené údaje počtu testovaných a pozitívnych. V tomto prípade teda zlyhal ľudský faktor.

Príklad 2: známky z matematiky (0-20) portugalských stredoškolských študentov
https://archive.ics.uci.edu/dataset/320/student+performance
student-mat.csv



13. Charakteristiky variability: disperzia a štandardná odchýlka

- Hovorili sme najmä o **charaketeristikách polohy** – kde sa dáta nachádzajú.
- Zaujíma nás však aj to, nakoľko sú dáta rozptýlené – to vyjadrujú **charaketristiky variability.** Túto vlastnosť má napríklad medzikvartilové rozpätie, ktoré sme už spomínali. Najznámejšie charakteristiky variability sú **disperzia (**iné názvy: **rozptyl, variancia**) a **štandardná odchýlka**

Základná myšlienka **disperzie**:

- Budeme si všímať odchýlky dát od priemeru.
- Ak chceme vedieť, nakoľko sú dáta rozptýlené, nemôžeme uvažovať len samotné odchýlky, lebo kladné a záporné sa navzájom zrušia (ľahko sa ukáže, že súčet odchýliek od priemeru je presne nula).
- Preto ich umocníme na druhú, tým sa stratí znamienko a zachová sa len to, či je odchýlka veľká alebo malá.
- Zoberieme priemer z týchto druhých mocnín.

Poznámka:

- Ak ide o disperziu z úplného(tzv. základného) dátového súboru, použije sa presne tento priemer z posledného bodu.
- Ak ide o výber, súčet sa nedelí počtom pozorovaní, ale hodnotou o jednotku menšou (neskôr si vysvetlíme, prečo) - hovorí sa aj o výberovej disperzii.

Rozdiel medzi **základným a výberovým súborom** (angl. population, sample)**:**

- základný súbor: voľby
- výberový súbor: prieskum volebných preferencií

Cieľom je vedieť na základe výberového súboru povedať niečo užitočné o základlnom súbore.

```
import statistics
data=[1, 2, 3, 4, 5]

sample_variance=statistics.variance(data)       # vyberova disperzia
population_variance=statistics.pvariance(data) # disperzia celej populacie

print(sample_variance)
print(population_variance)
```

```
2.5
2
```

Naozaj sa tu deje to, čo sme hovorili:

```
import numpy as np

priemer=np.mean(data)
n=len(data)
kvadraticke_odchylky=[(x - priemer)**2 for x in data]
print(sum(kvadraticke_odchylky)/(n - 1))
print(sum(kvadraticke_odchylky)/n)
```

```
2.5
2.0
```

Štandardná odchýlka je odmocnina z disperzie.
Motivácia:

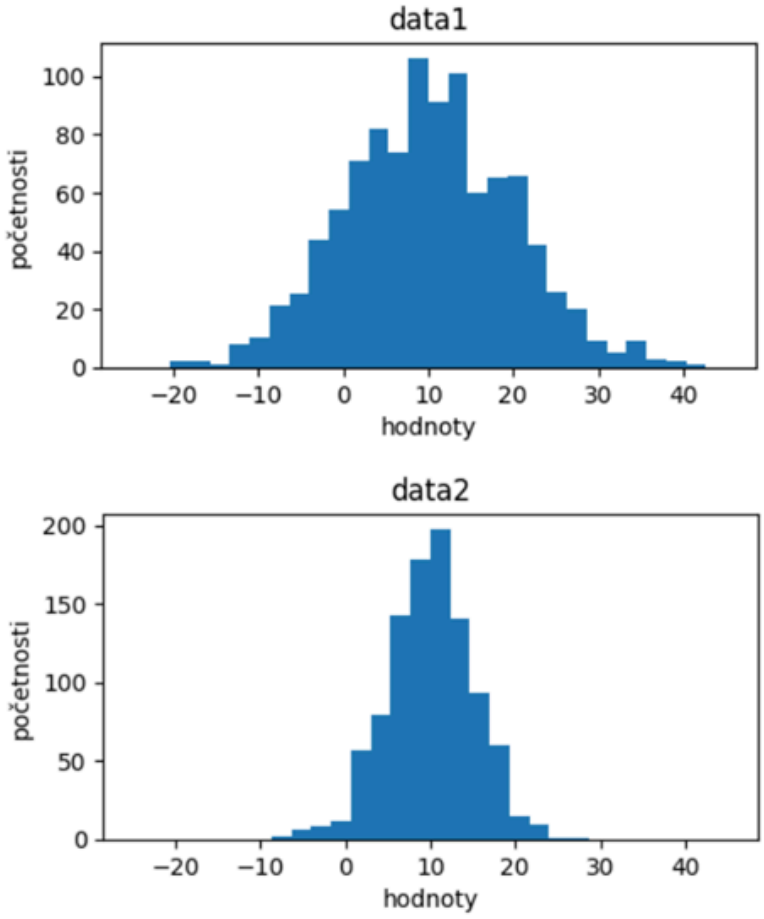
- ak sú dáta napríklad v metroch, disperzia má jednotku meter štvorcový (vznikne umocnením), a teda štandardná odchýlka má rovnaký rozmer ako pôvodné dáta
- má teda zmysel hovoriť napríklad o tom, aký podiel dát je v intervale “priemer plus/mínus dve štandardné odchýlky”

```
sample_std=statistics.stdev(data)
population_std=statistics.pstdev(data)

print(sample_std)
print(population_std)
```

```
1.5811388300841898
1.4142135623730951
```

Vygenerované dáta, na ktorých budeme vidieť rozdiel:




```
var1=statistics.variance(data1)
var2=statistics.variance(data2)

print(f"variancia súboru data1: {var1:.2f}")
print(f"variancia súboru data2: {var2:.2f}")
```

```
variancia súboru data1: 96.01
variancia súboru data2: 24.94
```