

MATEMATIKA ÉS STATISZTIKA DATA SCIENCE-HEZ



**LOGIKAI MŰVELETEK,
HALMAZOK,
FÜGGVÉNYEK,
LOGARITMUS**

MATEMATIKAI ÁLLÍTÁSOK ÉS TAGADÁSUK



Matematikai állításnak (propozíciónak) nevezzük azt a kijelentést, amelyről egyértelműen eldönthető, hogy igaz vagy hamis.

Egy **állítás tagadása** (negációja) az az állítás, amely akkor igaz, ha az eredeti hamis, és fordítva.

Állítás (A)	Igazság érték	Tagadás ($\neg A$ vagy \bar{A})	Igazság érték
$5 > 3$	I	$5 \leq 3$	H
7 osztható 2-vel	H	7 nem osztható 2-vel	I
Minden szám nagyobb 0-nál.	H	Van olyan szám, ami nem nagyobb 0-nál.	I
Van olyan szám, ami páros.	I	Nincs olyan szám, ami páros. Minden szám páratlan.	H

IGAZSÁGTÁBLÁK



Az **igazságtábla** egy olyan táblázat, amely minden lehetséges bemenethez megmutatja a vizsgált állítás (vagy logikai kifejezés) igazságértékét.

Egy állítás tagadásához tartozó igazságtábla tehát így néz ki:

A	$\neg A$ vagy \bar{A}
I	H
H	I

LOGIKAI MŰVELETEK ÁLLÍTÁSOKKAL



Két állítás **konjunkciója** („ÉS” kapcsolatuk) akkor és csak akkor igaz, ha **mindkettő igaz**.

A	B	$A \wedge B$
I	I	I
I	H	H
H	I	H
H	H	H

Két állítás **diszjunkciója** („VAGY” kapcsolatuk) akkor és csak akkor igaz, ha **legalább az egyik igaz**.

A	B	$A \vee B$ *
I	I	I
I	H	I
H	I	I
H	H	H

* Akkor is I lesz az érték, ha mindkét állítás I >>> a matematikában, tehát “MEGENGEDŐ vagy”-ot használunk a hétköznapi szóhasználatban szereplő “KIZÁRÓ vagy” helyett.

DE MORGAN AZONOSSÁGOK



Ezeket az azonosságokat a **logikai műveletekkel** összekapcsolt állítások **tagadásánál** tudjuk alkalmazni:

$$\neg (A \wedge B) \equiv (\neg A) \vee (\neg B)$$

A	B	$A \wedge B$	$\neg (A \wedge B)$
I	I	I	H
I	H	H	I
H	I	H	I
H	H	H	I

$\neg A$	$\neg B$	$(\neg A) \vee (\neg B)$
H	H	H
H	I	I
I	H	I
I	I	I

$$\neg (A \vee B) \equiv (\neg A) \wedge (\neg B)$$

A	B	$A \vee B$	$\neg (A \vee B)$
I	I	I	H
I	H	I	H
H	I	I	H
H	H	H	I

$\neg A$	$\neg B$	$(\neg A) \wedge (\neg B)$
H	H	H
H	I	I
I	H	I
I	I	I

LOGIKAI MŰVELETEK A DS-BEN I.



Logikai művelet	Pythonos operátor
TAGADÁS	'not' vagy '~' vagy '!='
'ÉS'	'and' vagy '&'
'VAGY' (megengedő)	'or' vagy ' '
'VAGY' (kizáró)	'^'

Kiértékelés szabályai:

- ▶ balról jobbra
- ▶ TAGADÁS > ÉS > VAGY
- ▶ () segítségével növelhető a precedencia

Írj egy fizzbuzz(n: int) -> str függvényt, amely:

- "FizzBuzz"-t ad vissza, ha n 3-mal és 5-tel is osztható,
- "Fizz"-t ad vissza, ha n csak 3-mal osztható,
- "Buzz"-t ad vissza, ha n csak 5-tel osztható,
- különben az n-t szöveges formában adja vissza.

```
[1]: def fizzbuzz(n):  
  
    # Order matters: check the strongest (AND) condition first  
    if (n % 3 == 0) & (n % 5 == 0):  
        return "FizzBuzz"  
    elif n % 3 == 0:  
        return "Fizz"  
    elif n % 5 == 0:  
        return "Buzz"  
    else:  
        return str(n)  
  
[2]: # Let's see how our function works  
[fizzbuzz(x) for x in range(16)]  
  
[2]: ['FizzBuzz',  
      '1',  
      '2',  
      'Fizz',  
      '4',  
      'Buzz',  
      'Fizz',  
      '7',  
      '8',  
      'Fizz',  
      'Buzz',  
      '11',  
      'Fizz',  
      '13',  
      '14',  
      'FizzBuzz']
```

```
[2]: df  
[2]:
```

	user_id	age	country	purchased
0	U001	22	HU	True
1	U002	35	DE	False
2	U003	41	HU	True
3	U004	29	FR	False
4	U005	50	HU	True
5	U006	38	PL	True
6	U007	27	HU	False
7	U008	45	DE	False
8	U009	31	FR	True
9	U010	24	HU	False

```
[ ]: # Select Hungarian users over 30 years old  
df[(df['country'] == 'HU') & (df['age'] > 30)]  
  
# Select users from Hungary OR France  
df[(df['country'] == 'HU') | (df['country'] == 'FR')]  
  
# Select users who are NOT from Germany AND are older than 30  
df[~(df['country'] == 'DE') & (df['age'] > 30)]  
  
# Select users who (purchased AND are from Hungary)  
# OR (are older than 40 AND NOT from Germany)  
df[((df['purchased']) & (df['country'] == 'HU'))  
   | ((df['age'] > 40) & ~(df['country'] == 'DE'))]  
  
# Select users who are either from HU OR purchased – but NOT both  
df[(df['country'] == 'HU') ^ (df['purchased'])]
```


HALMAZOK ÉS HALMAZMŰVELETEK



Matematikában a **halmaz** bizonyos tulajdonságokkal rendelkező dolgok összessége. Egy halmazt az **elemei határozzák meg**, és általában az elemeket jellemző közös tulajdonsággal vagy maguknak az elemeknek a felsorolásával adjuk meg.

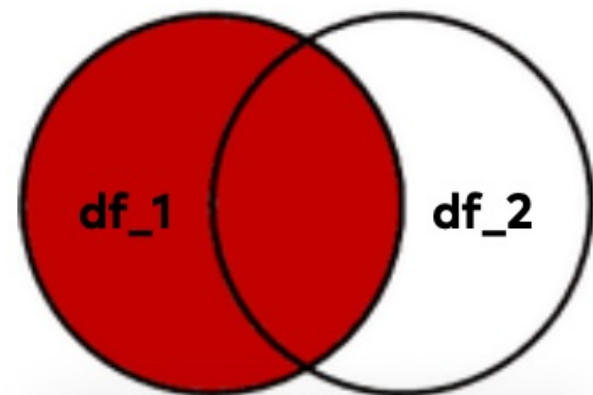
A **halmazműveletek** tulajdonképpen logikai műveletek a halmazelemekre értelmezve.

Halmazművelet	Jelölés	Logikai művelet	Jelölés	Jelentés
KOMPLEMENTER	\overline{A}	TAGADÁS	$\neg A$ vagy \overline{A}	Azok az elemek, amik nincsenek benne a halmazban
METSZET	$A \cap B$	'ÉS'	$A \wedge B$	Azok az elemek, amik mindkét halmazban benne vannak
UNIÓ	$A \cup B$	'VAGY' (megengedő)	$A \vee B$	Azok az elemek, amik legalább az egyik halmazban benne vannak
KÜLÖNBSÉG	$A \setminus B$	-	$A \wedge \neg B$	Azok az elemek, amik csak az A halmazban vannak benne
SZIMMETRIKUS KÜLÖNBSÉG	$A \Delta B$	'VAGY' (kizáró)	$A \oplus B$	Az elemek, amik csak az egyik halmazban vannak benne, de nem mindkettőben.

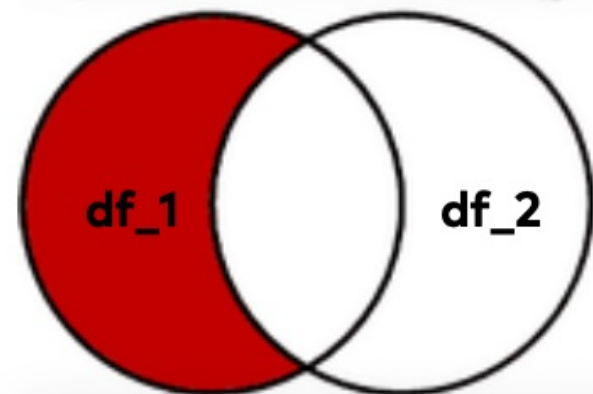
HALMAZMŰVELETEK A DS-BEN



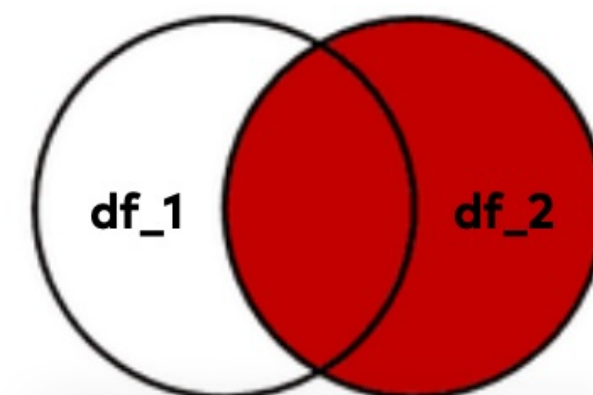
PANDAS MERGES



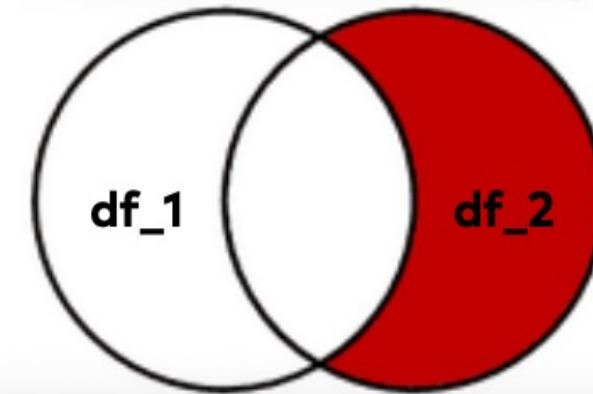
```
merged_df = df_1.merge(df_2,  
                        on='key',  
                        how='left')
```



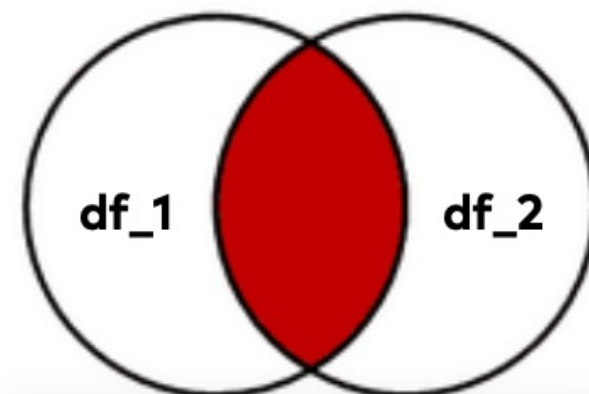
```
merged_df = df_1.merge(df_2, on='key', how='left',  
                       indicator=True).query('_merge == "left_only"')
```



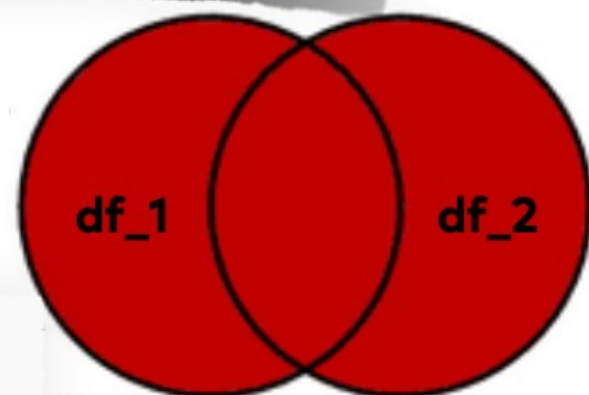
```
merged_df = df_1.merge(df_2,  
                        on='key',  
                        how='right')
```



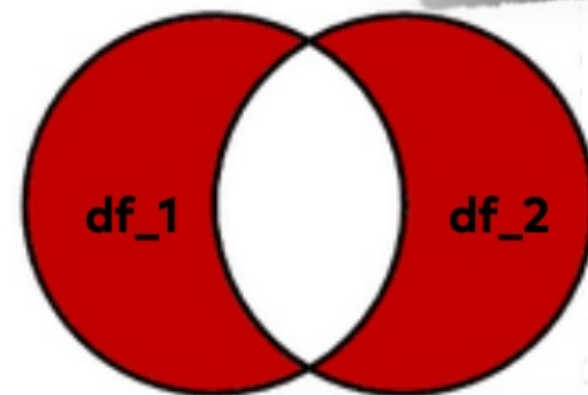
```
merged_df = df_1.merge(df_2, on='key', how='right',  
                       indicator=True).query('_merge == "right_only"')
```



```
merged_df = df_1.merge(df_2,  
                        on='key',  
                        how='inner')
```



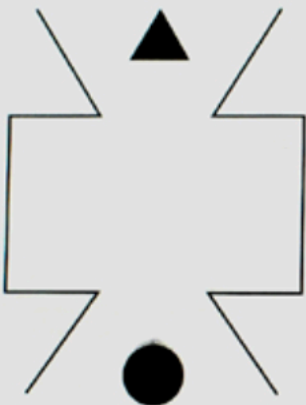
```
merged_df = df_1.merge(df_2,  
                        on='key',  
                        how='outer')
```



```
merged_df = df_1.merge(df_2, on='key', how='outer',  
                       indicator=True).query('_merge != "both"')
```


FÜGGVÉNYEK

3. Mit csinál a gép? Pótold a hiányzó számokat!



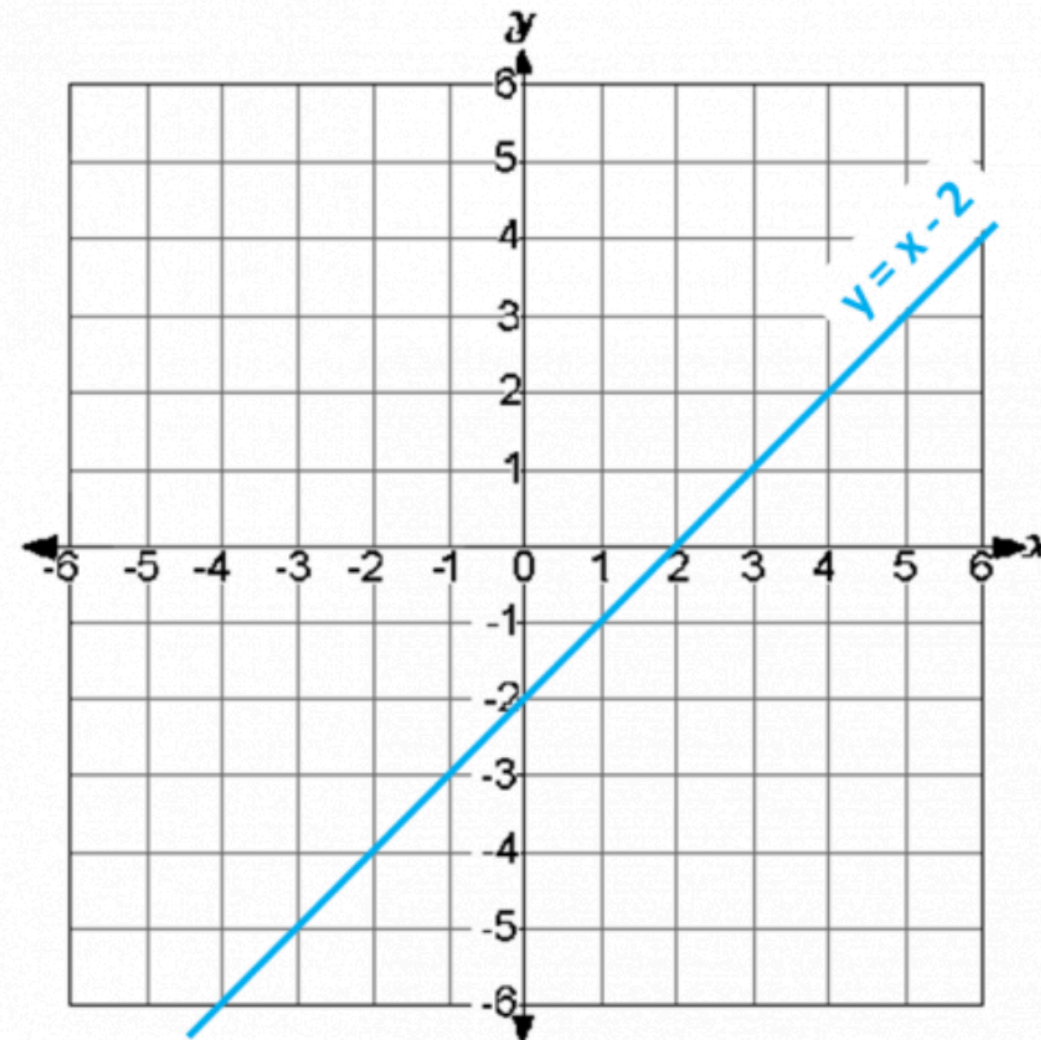
▲	2	5			3	4		
●	0	3	2	0			3	1

$$\blacktriangle - 2 = \bullet$$

$$y = x - 2$$

$$f(x) = x - 2$$

$$f : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto x - 2$$



meredekség = 1

tengelymetszet = - 2

Függvény jellemzés

- 1) értelmezési tartomány
- 2) értékkészlet
- 3) zérushely(ek)
- 4) korlátosság
- 5) monotonitás
- 6) szélsőérték(ek)
- 7) paritás
- 8) periódikusság
- 9) folytonosság
- 10) konvexitás
- 11) inflexiós pont(ok)
- 12) deriválhatóság
- 13) határérték

Függvény definíciója:

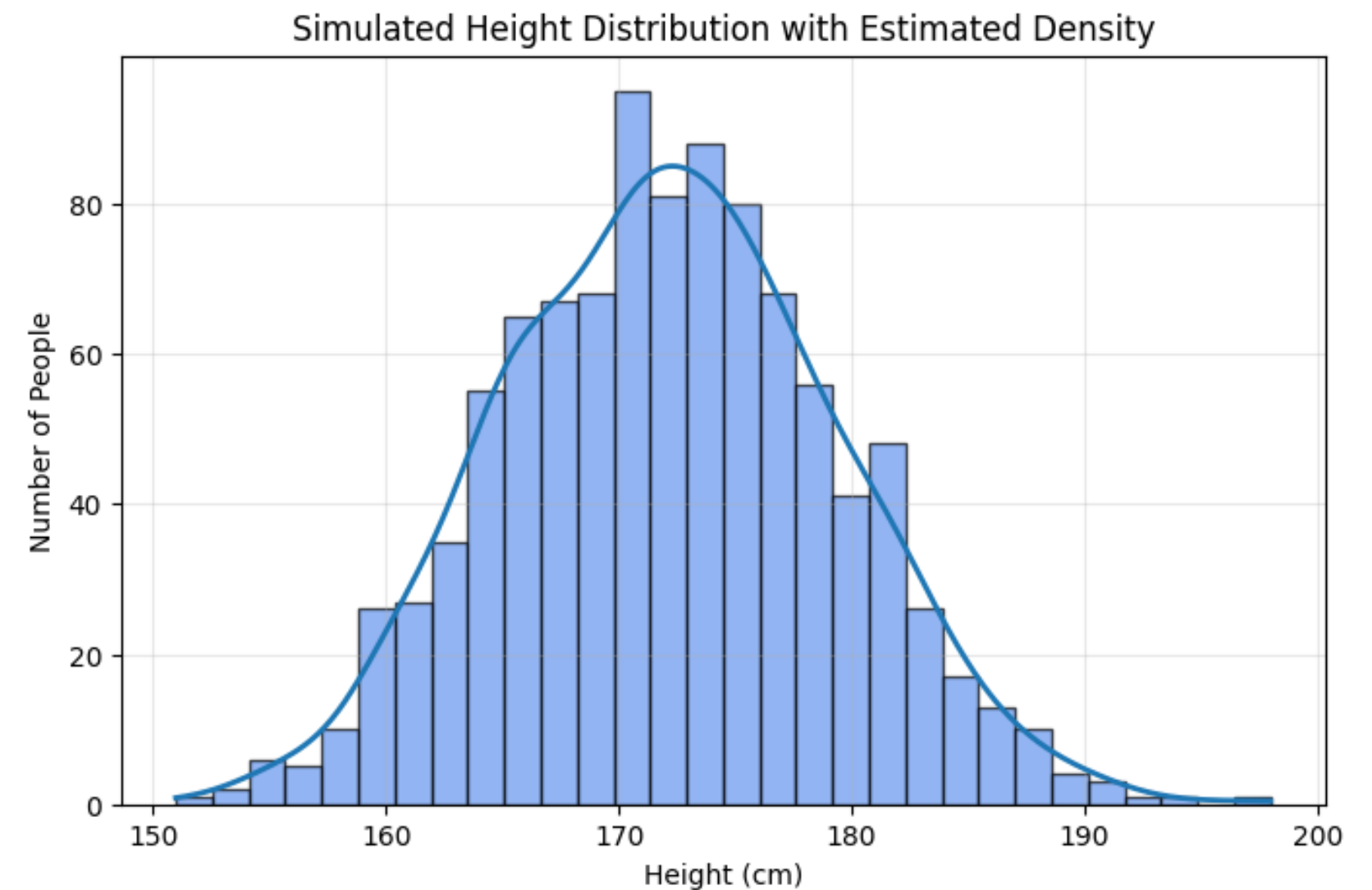
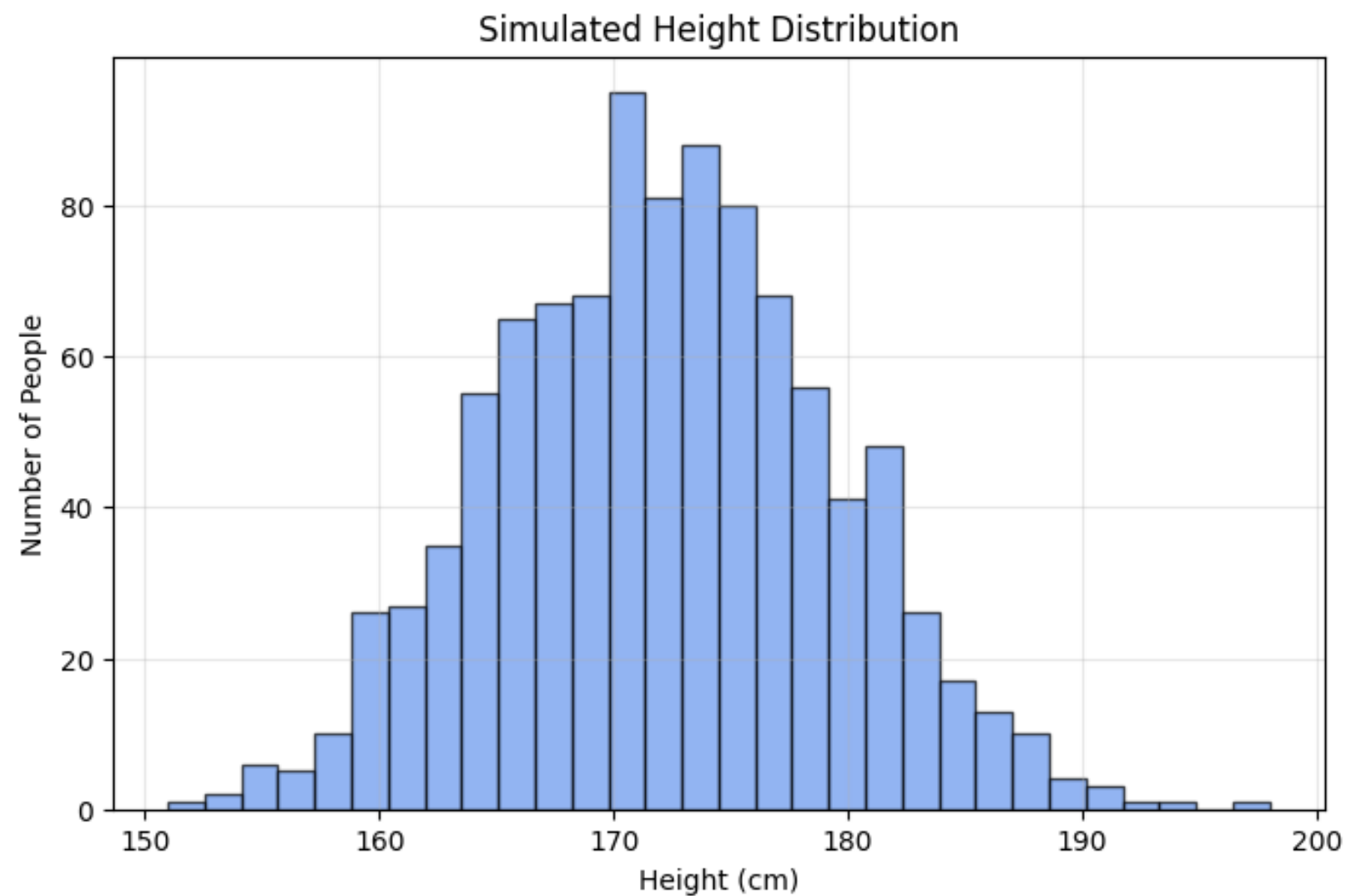
két halmaz közti **egyértelmű hozzárendelés**

FÜGGVÉNYEK A DS-BEN I.

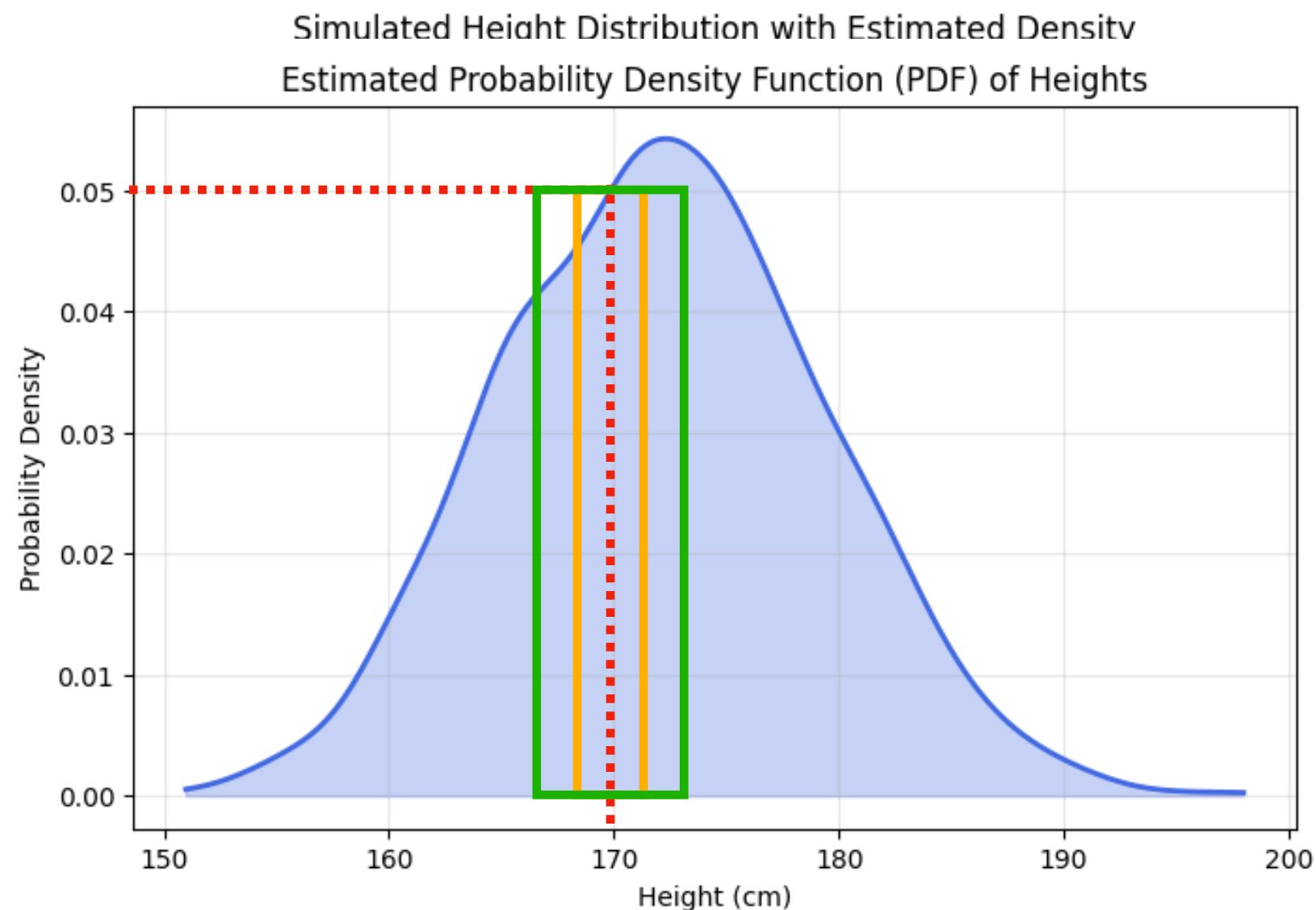


Sűrűség- és eloszlás függvények

1000 fős minta.
Mindenkinek megvan
a magassága cm-ben.



FÜGGVÉNYEK A DS-BEN I.



$x = 170$ cm-nél $y = 0,05$ a valószínűségi sűrűség
a PDF tehát azt mutatja meg, hogy az adott x érték
környezetében milyen intenzíven fordulnak elő értékek

VIGYÁZAT!

Ez NEM azt jelenti, hogy 5% az esélye annak,
hogy valaki 170 cm magas

MIT JELENT AKKOR?

Azt jelenti, hogy a valószínűségi sűrűség ekörül a
pont körül 0.05 — vagyis a 170 cm körüli kis
intervallumok valószínűsége a rájuk rajzolható 0.05
magas téglalapok területével becsülhető:

$$P(169 \leq X \leq 171) \approx 0.05 \times 2 = 0.1$$

Tehát kb. 10% az esélye, hogy valaki 169–171 cm közé esik.

$$P(168 \leq X \leq 172) \approx 0.05 \times 4 = 0.2$$

Tehát kb. 20% az esélye, hogy valaki 168–172 cm közé esik.

Az intervallumokhoz tartozó pontos valószínűséget
a PDF függvény alatti terület adja meg:

$$P(a \leq X \leq b) = \int_a^b f(x) dx$$

PDF

~~(Probability Density Function)
elméleti függvény, gyakorlatban
általában nem ismerjük~~

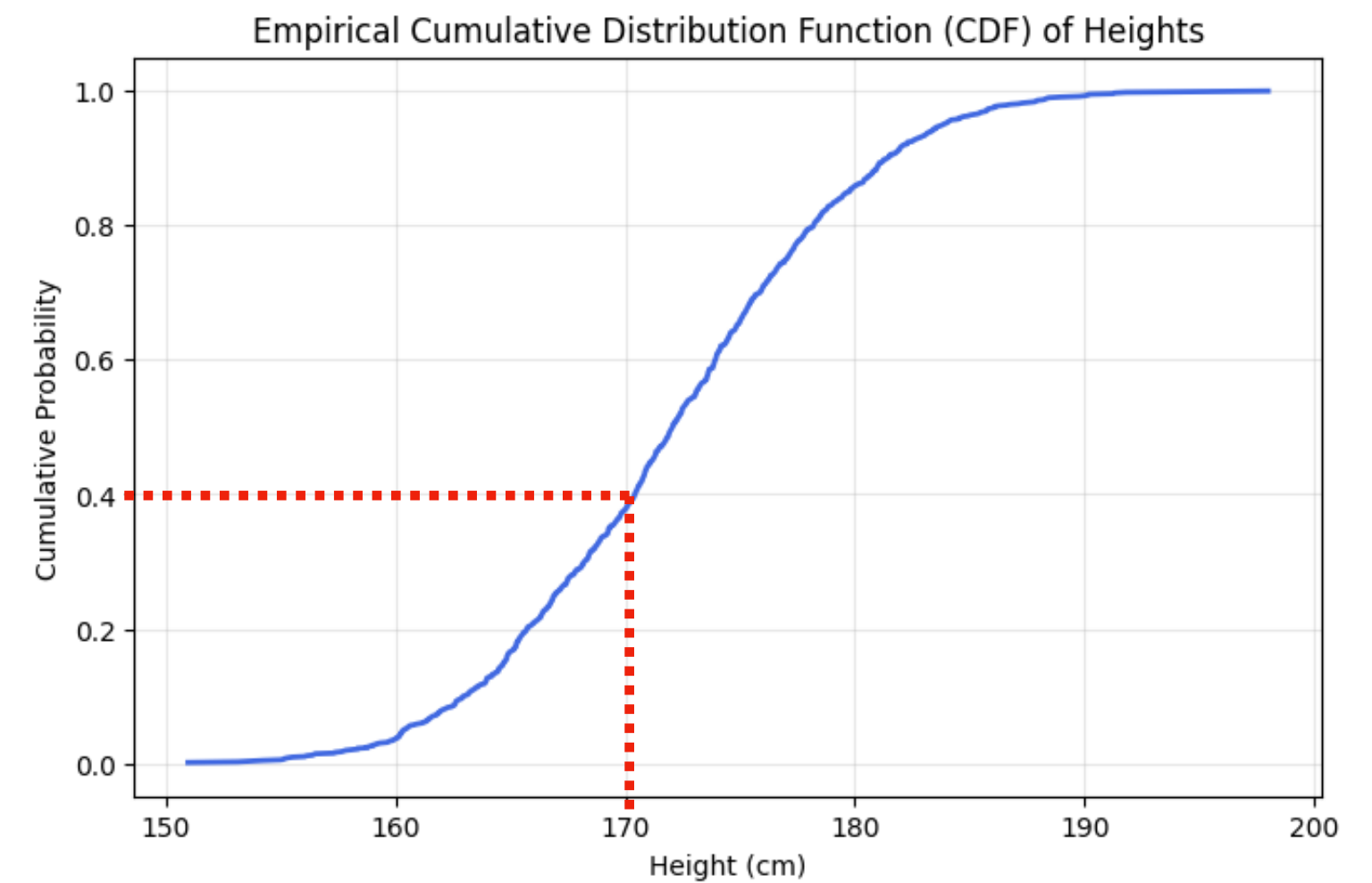
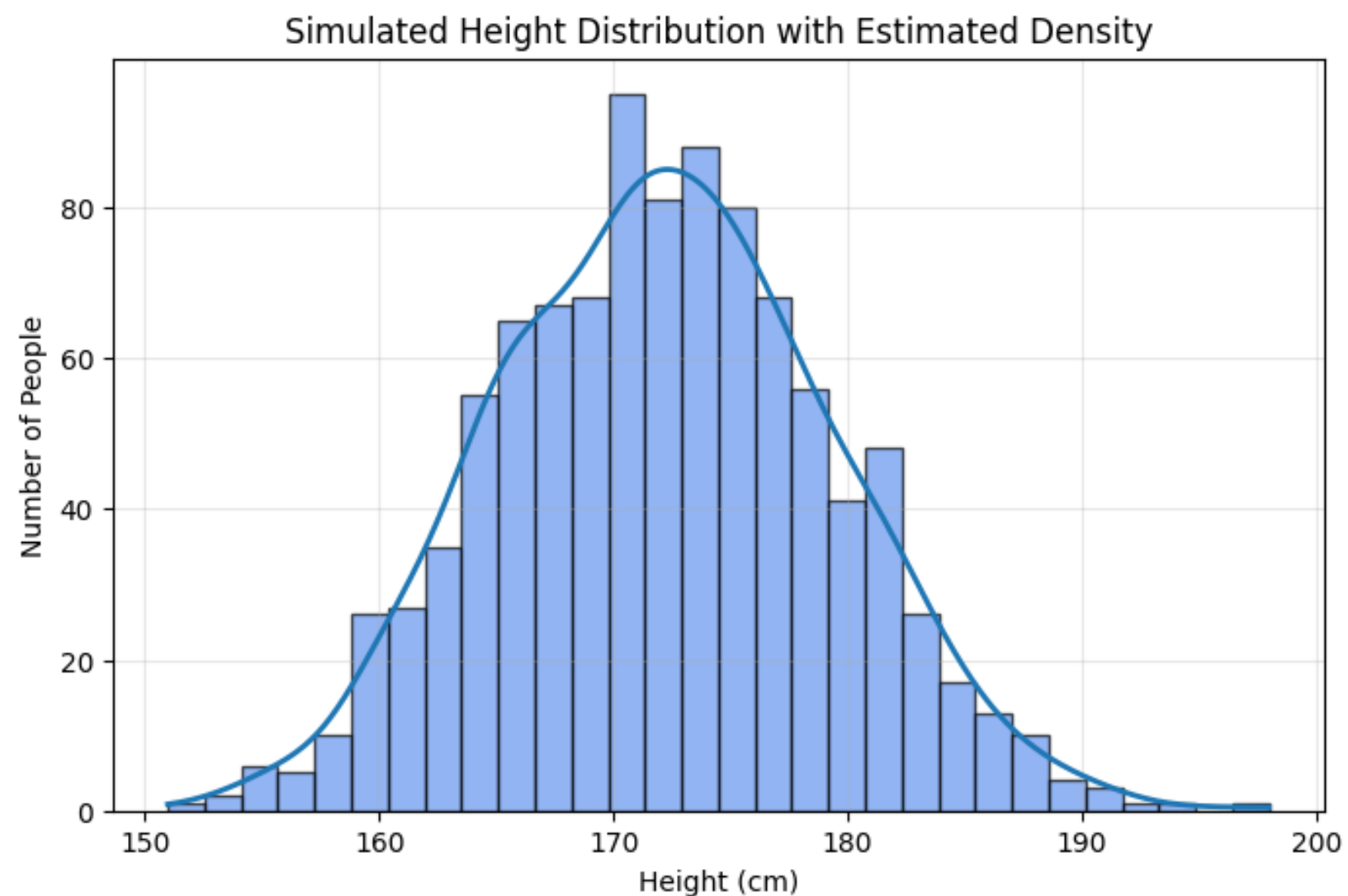
KDE

(Kernel Density Estimation)
adatokból rajzolt görbe, ami a
legjobban becsüli a PDF-et

FÜGGVÉNYEK A DS-BEN II.



1000 fős minta.
Mindenkinek megvan
a magassága cm-ben.



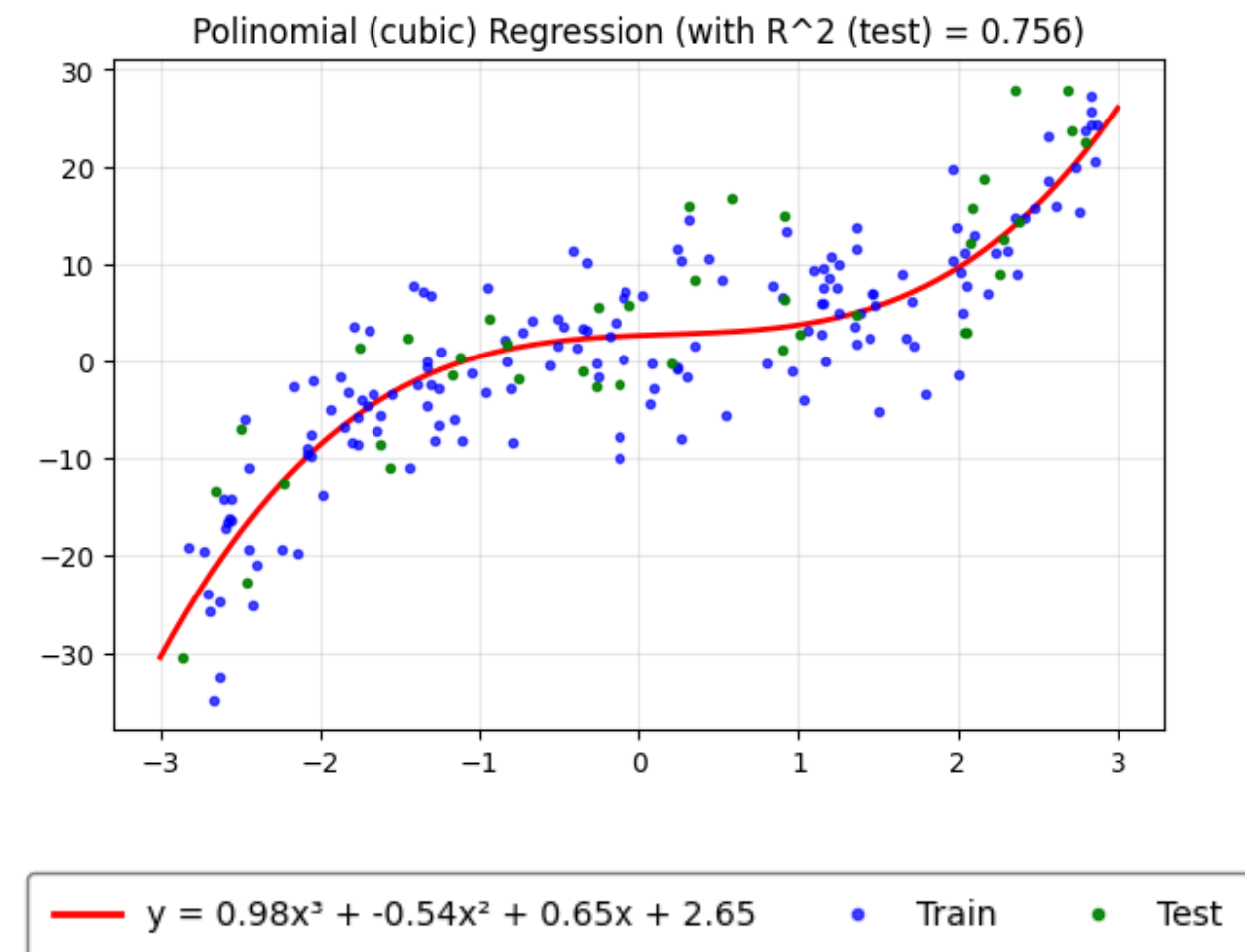
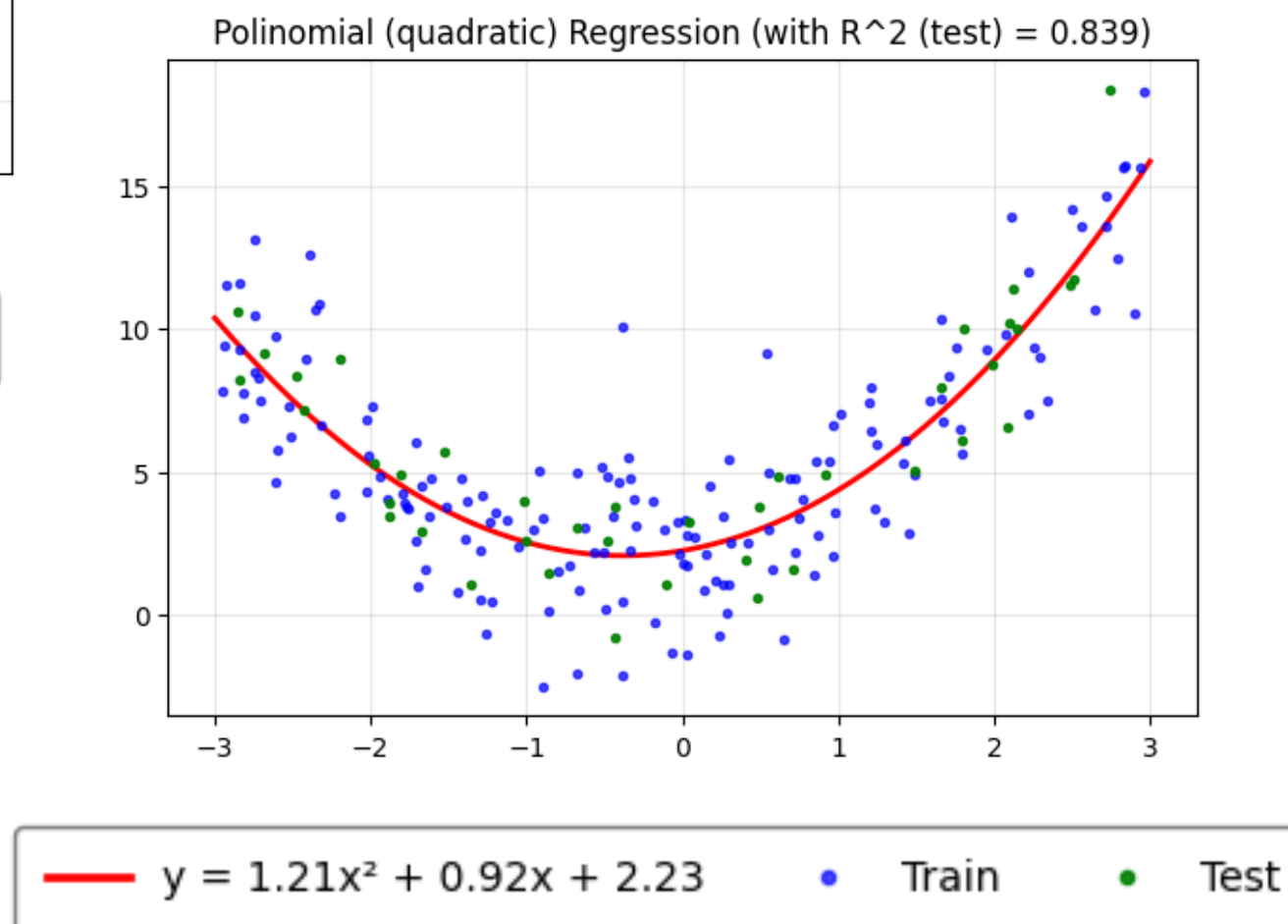
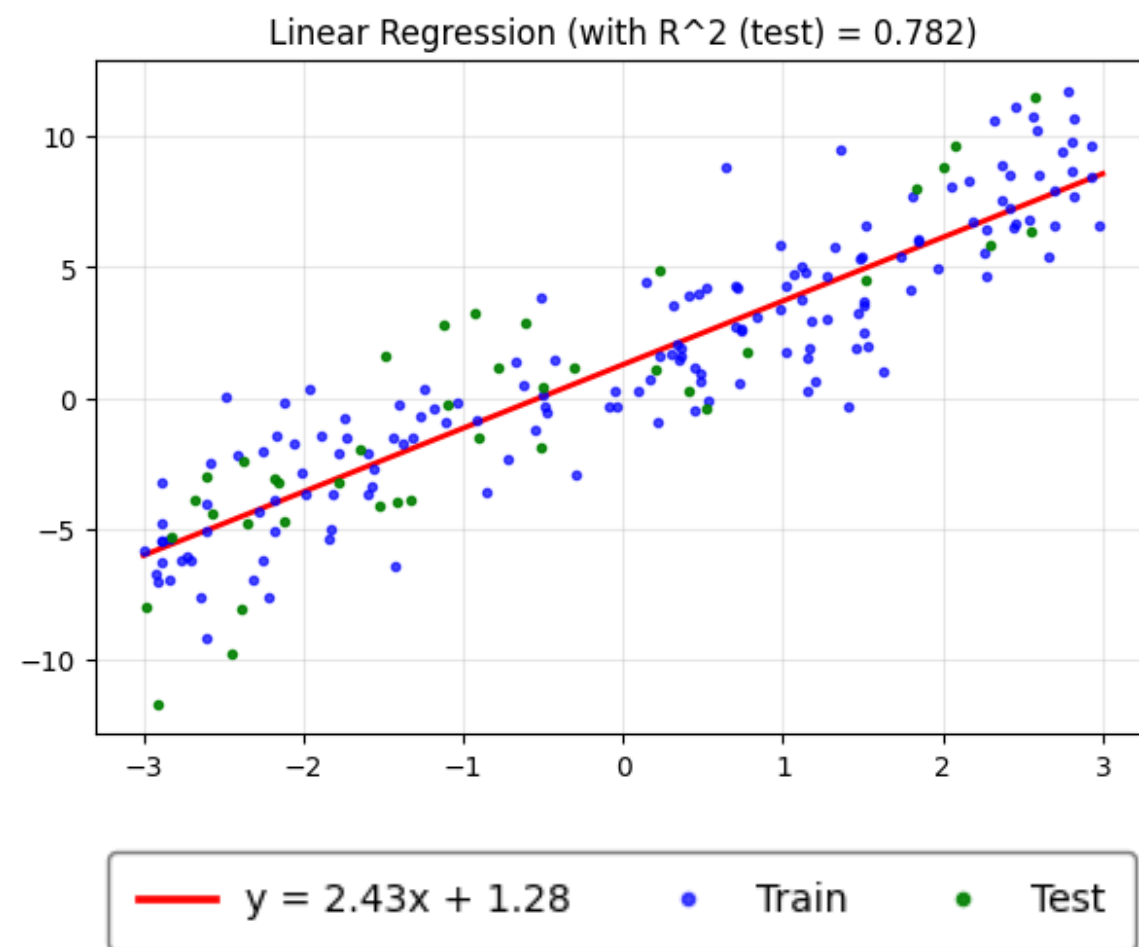
$x = 170$ cm-nél $y = 0.4$ a kumulatív valószínűség
a CDF tehát azt mutatja meg, hogy adott x értékig
bezárólag mekkora valószínűséggel fordul elő a változó

Vagyis jelen esetben a mintánkban az emberek magassága
kb. 40%-ban 170 cm-nél kisebb vagy egyenlő.

FÜGGVÉNYEK A DS-BEN III.



Lineáris- és polinomiális regressziós modellek



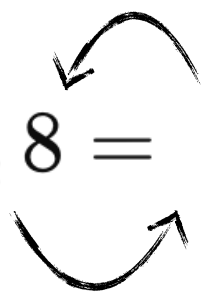
LOGARITMUS DEFINÍCIÓJA ÉS AZONOSSÁGAI

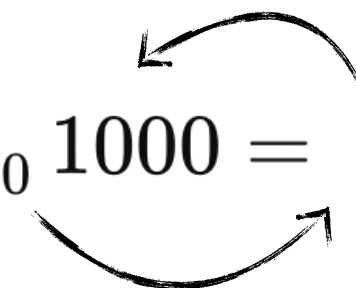


Logaritmus definíciója

Ha $a > 0$, $a \neq 1$, akkor $a^{\log_a b} = b$

$\log_a b$ azt a hatványkitevőt értjük,
amire 'a'-t emelve 'b'-t kapunk

$$\log_2 8 =$$


$$\log_{10} 1000 =$$


Logaritmus azonosságai

$$\log_a (x \cdot y) = \log_a x + \log_a y$$

$$\log_a \left(\frac{x}{y} \right) = \log_a x - \log_a y$$

$$\log_a (x^r) = r \cdot \log_a x$$

Összetett, **nemlineáris** kapcsolatokat
képes **lineárisabbá** tenni.

Sok folyamat **hatványfüggvénnyel** vagy **szorzatos formában** működik, pl.:

$$\text{Revenue} = c \cdot (\text{Traffic})^{a_1} \cdot (\text{ConversionRate})^{a_2}$$

Ha logaritmust veszünk:

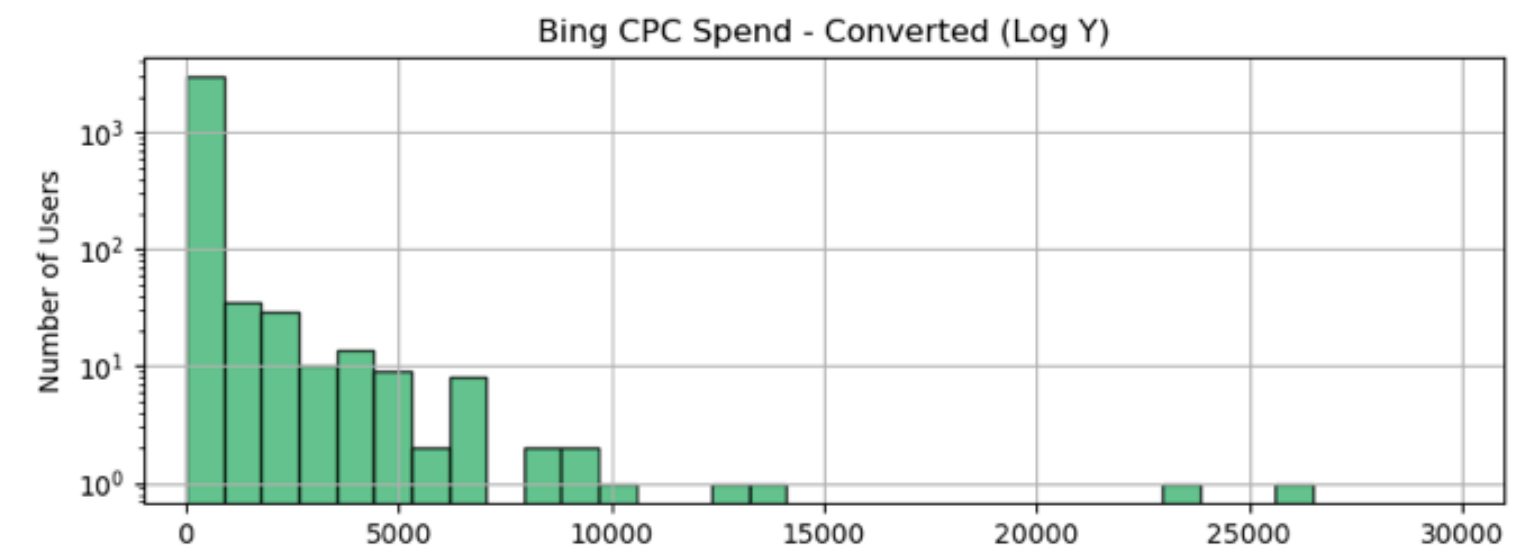
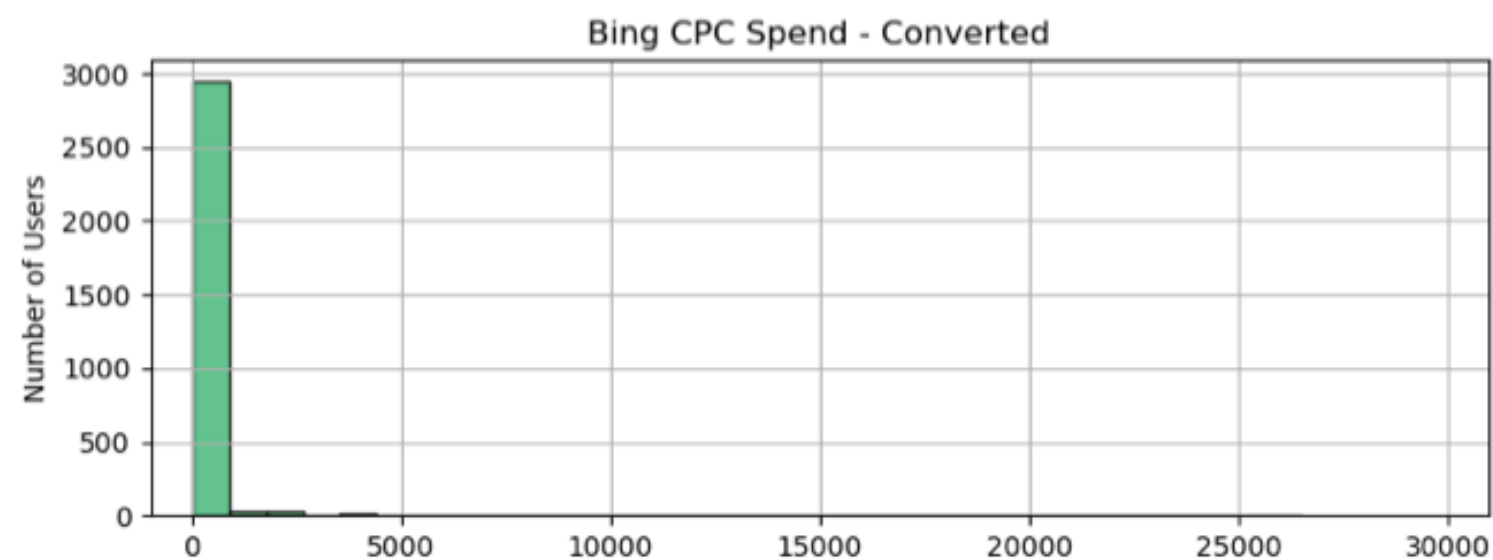
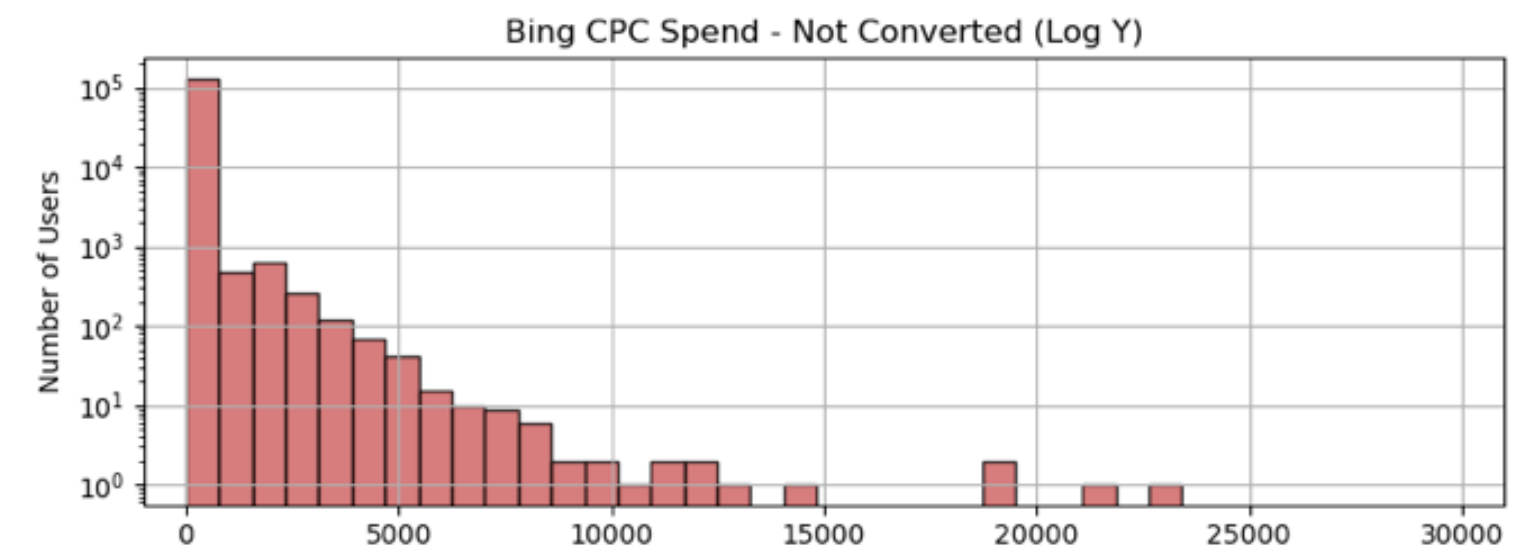
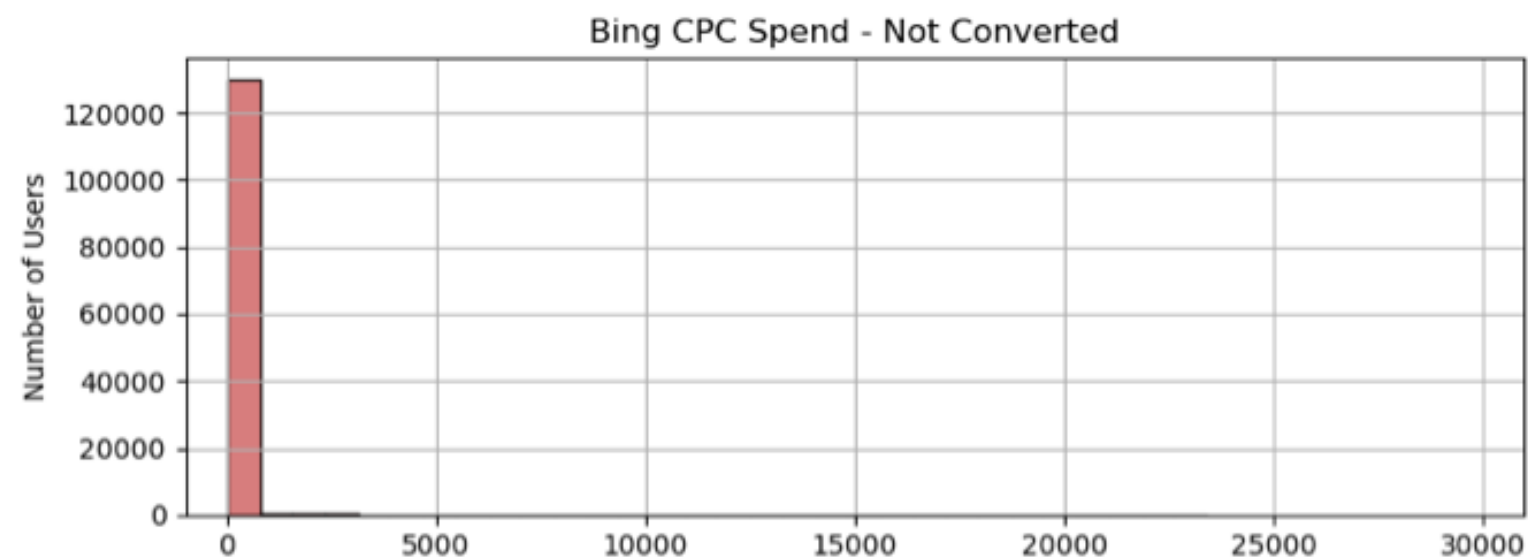
$$\log(\text{Revenue}) = \log(c) + a_1 \log(\text{Traffic}) + a_2 \log(\text{ConversionRate})$$

LOGARITMUS A DS-BEN



"Log-transzformáció" – Skáláz és stabilizál

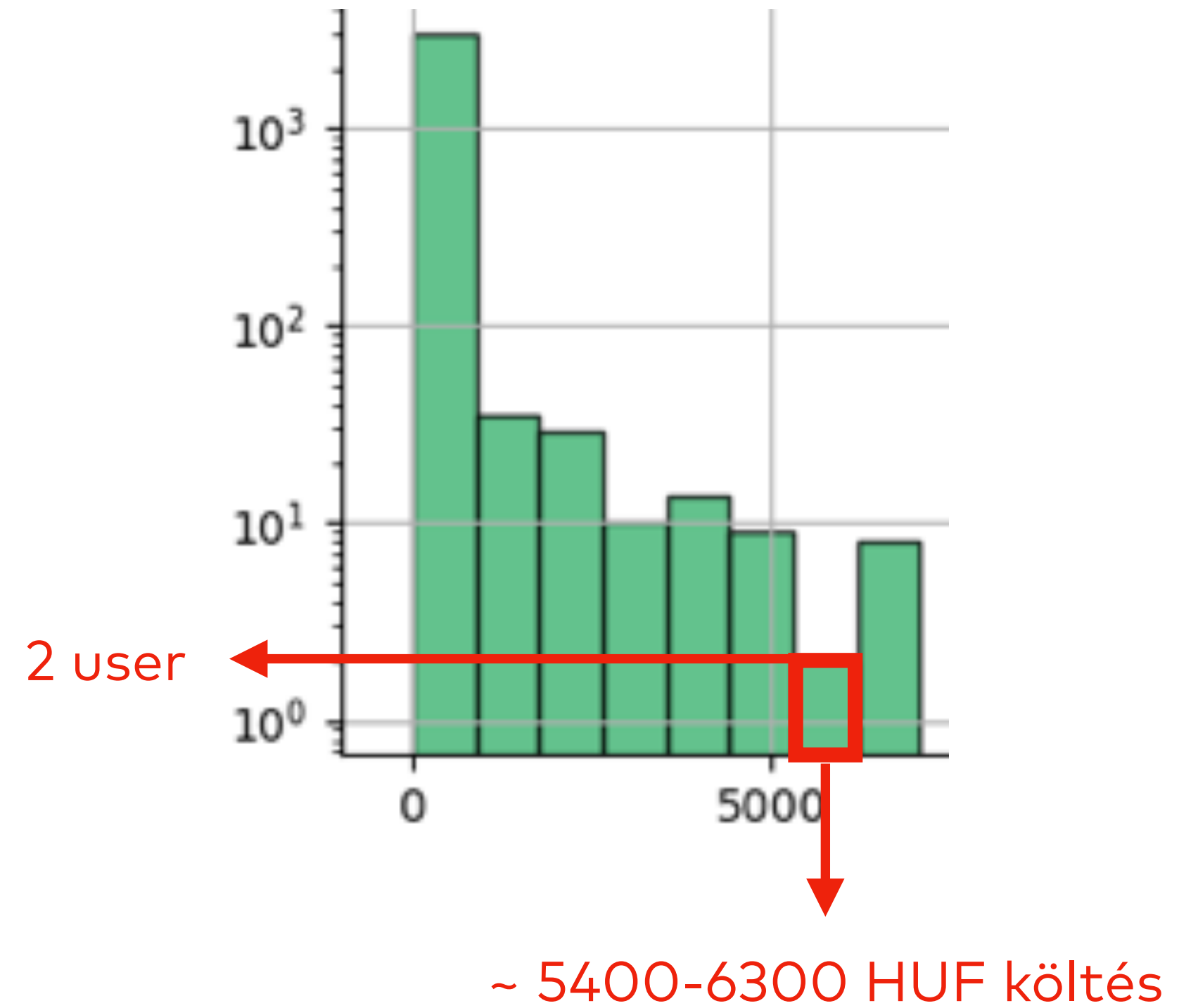
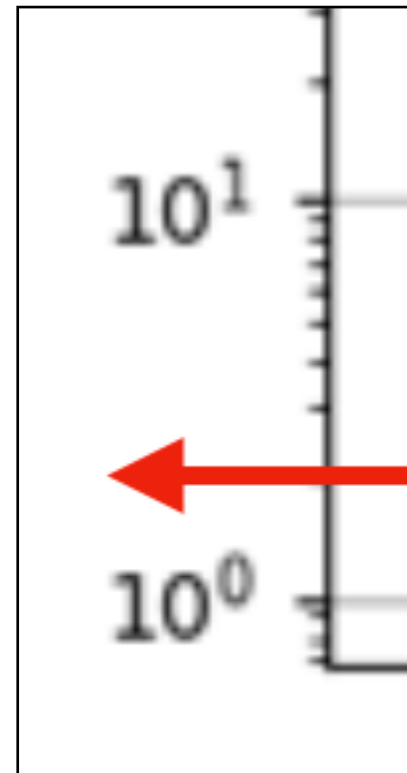
Összenyomja a nagy értékeket – Széthúzza a kis különbségeket – Csökkenti a szórást



LOGARITMUS A DS-BEN



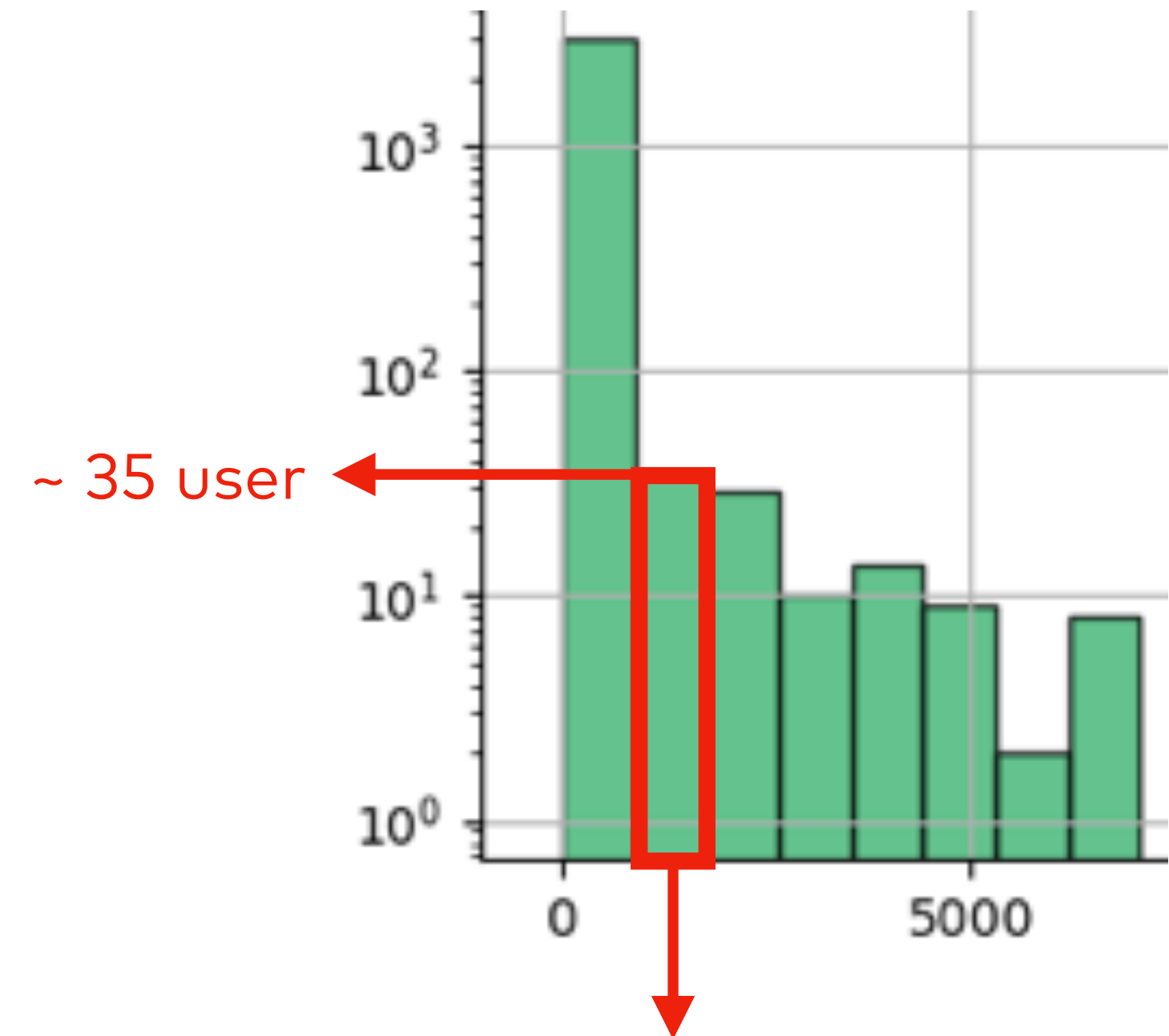
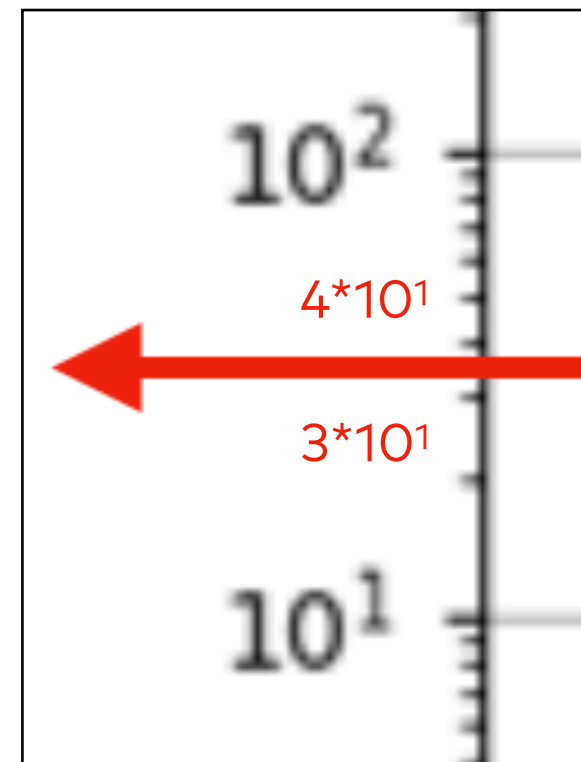
rácsvonal	valós érték
10^0	1
2×10^0	2
3×10^0	3
4×10^0	4
5×10^0	5
6×10^0	6
7×10^0	7
8×10^0	8
9×10^0	9
10^1	10



LOGARITMUS A DS-BEN



rácsvonal	valós érték
10^1	10
2×10^1	20
3×10^1	30
4×10^1	40
5×10^1	50
6×10^1	60
7×10^1	70
8×10^1	80
9×10^1	90
10^2	100



~ 900-1800 HUF költség

GYAKOROLJUNK!



Közösen megoldandó kódolós feladatok az érintett témakörökből:

[1_alkalom_minta_feladatok.ipynb](#)

TOVÁBBI GYAKORLÓ FELADATOK



Önállóan megoldandó kódolós feladatok az érintett témakörökből:

[1_alkalom_gyakorlo_feladatok.ipynb](#)