

Adatbányászat a Gyakorlatban

5. Gyakorlat: Gyakorisági adatok kezelése

Kuknyó Dániel
Budapesti Gazdasági Egyetem

2024/25
1.félév

1 Bevezetés

2 Adattáblák

3 Gépi tanulás

1 Bevezetés

2 Adattáblák

3 Gépi tanulás

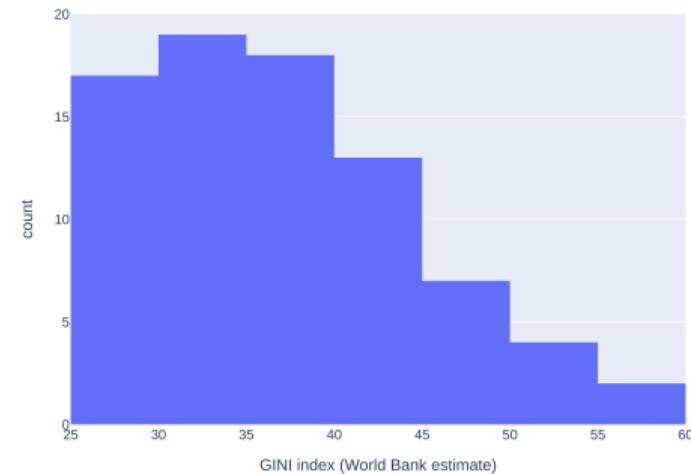
Hisztogramok létrehozása

Hisztogram

A hisztogram egy statisztikai grafikon, amely az adatok eloszlását mutatja be. Oszlopdiagram formájában ábrázolja, hogy az adatok milyen gyakorisággal fordulnak elő különböző intervallumokban.

Hisztogram létrehozása plotly segítségével:

```
1 px.histogram(data_frame=df, x=gini)
```



Hisztorogramok felbontása

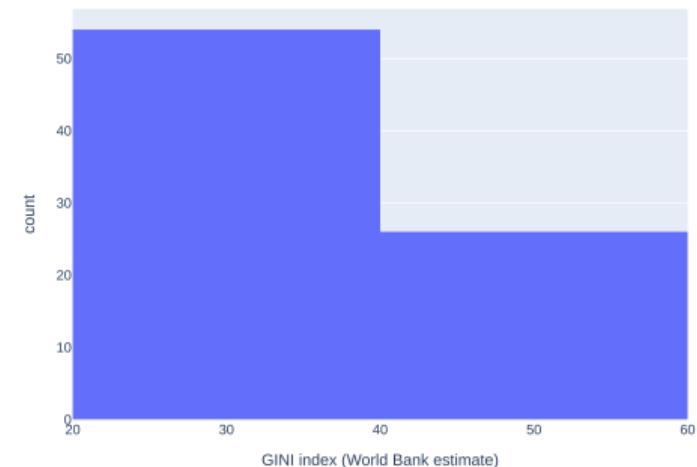
Osztályköz

Az osztályközök határozzák meg, hogy az adatok milyen tartományokba kerülnek, és ezek az intervallumok határozzák meg a hisztogram oszlopainak szélességét.

Az osztályközök száma az nbins paraméter segítségével állítható.

```
1 for n in [2, 45, 500]:  
2     px.histogram(data_frame=df, x=gini,  
                     nbins=n)
```

nbins = 2



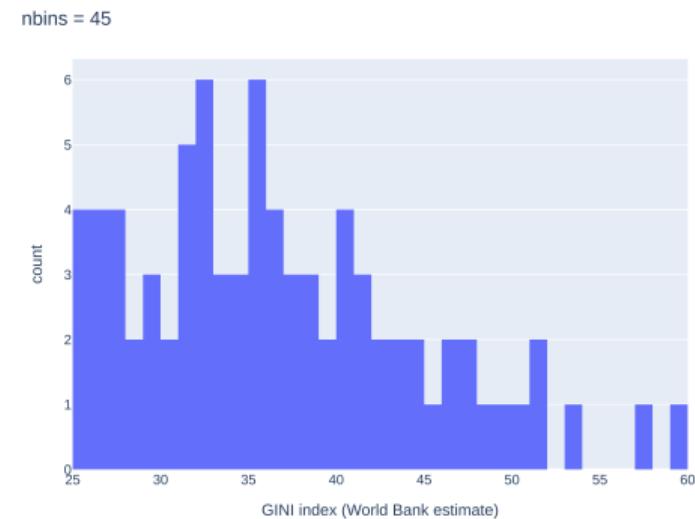
Hisztorogramok felbontása

Osztályköz

Az osztályközök határozzák meg, hogy az adatok milyen tartományokba kerülnek, és ezek az intervallumok határozzák meg a hisztogram oszlopainak szélességét.

Az osztályközök száma az nbins paraméter segítségével állítható.

```
1 for n in [2, 45, 500]:  
2     px.histogram(data_frame=df, x=gini,  
                     nbins=n)
```



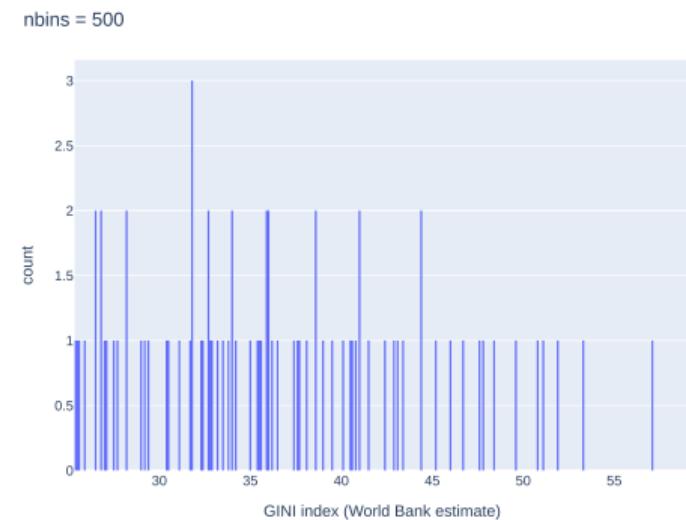
Hisztorogramok felbontása

Osztályköz

Az osztályközök határozzák meg, hogy az adatok milyen tartományokba kerülnek, és ezek az intervallumok határozzák meg a hisztorogram oszlopainak szélességét.

Az osztályközök száma az nbins paraméter segítségével állítható.

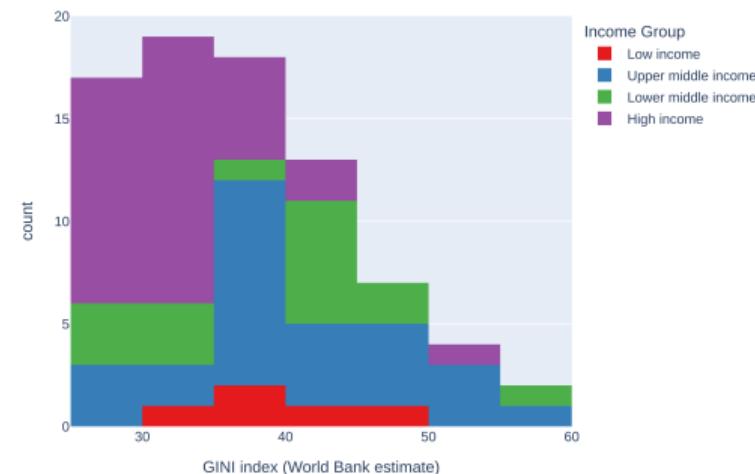
```
1 for n in [2, 45, 500]:  
2     px.histogram(data_frame=df, x=gini,  
                     nbins=n)
```



Hisztogram hasítása színekkel

Plotly express diagramokat lehetséges változón belüli csoportonként meghasítani. Ennek eléréséhez a color paramétert kell a megfelelő változóra állítani.

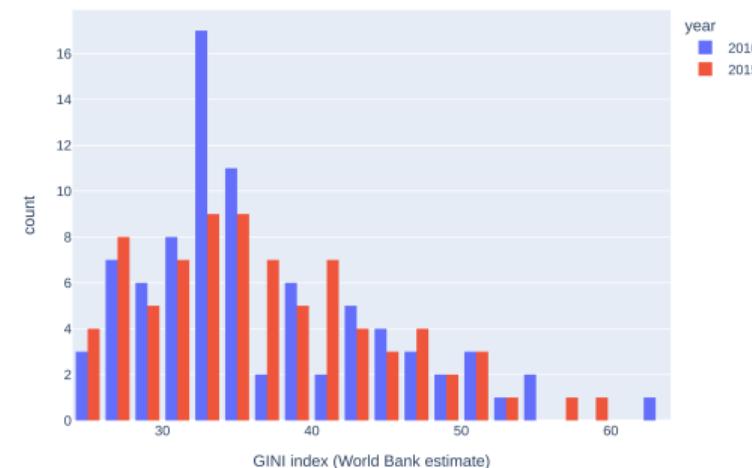
```
1 px.histogram(data_frame=df, x=gini,  
    color='Income Group',  
    color_discrete_sequence=px.colors.  
    qualitative.Set1)
```



Csoportosított hisztogramok

Vannak olyan esetek, amikor egy változónak több csoportját egymás mellett szükséges megmutatni. Ekkor a hisztogramokat lehetséges csoportosítani adott értékek szerint, a `color` és a `barmode='group'` paraméterek állításával.

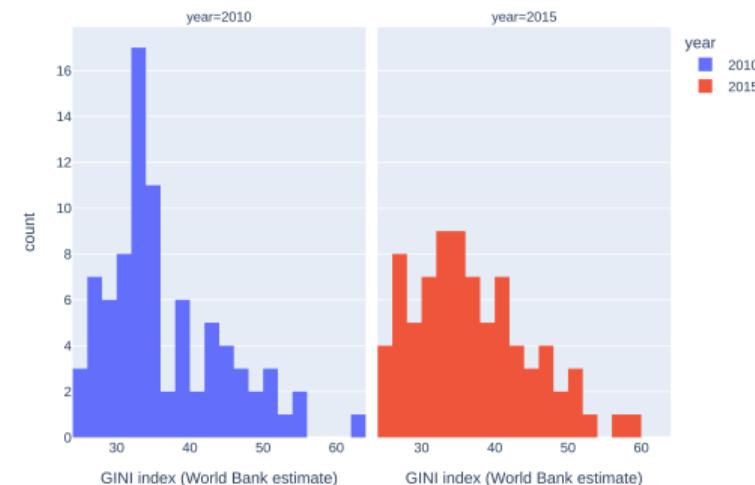
```
1 px.histogram(df, x=gini, color='year',  
   barmode='group')
```



Hasított hisztogramok

A diagramok hasítása adott változó értékei szerint lehetséges úgy is, hogy minden, a változóhoz tartozó értékre szűrt adathalmaz egy külön diagramon jelenik meg, a `facet_col` paraméter állításával.

```
1 px.histogram(df, x=gini, color='year',  
   facet_col='year')
```

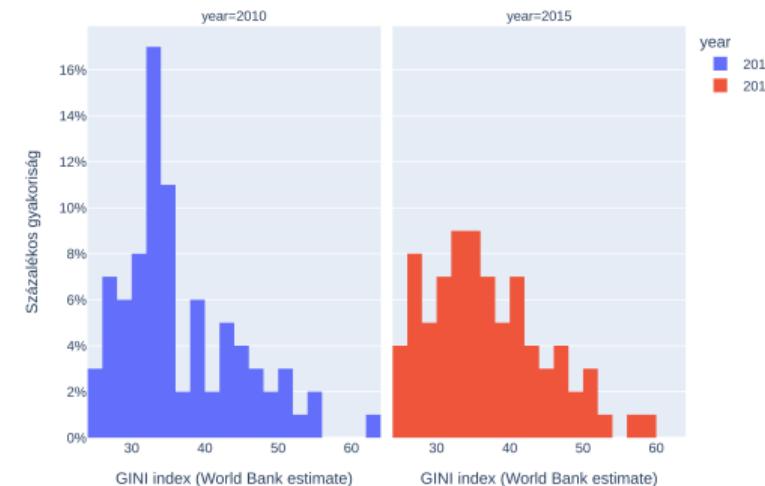


Hisztorogramok normalizálása

Normalizált hisztorogram

Olyan grafikon, ahol az egyes oszlopok az adott intervallumba eső adatok gyakoriságát jelzi olyan módon, hogy az oszlopok összege 1 legyen.

```
1 fig = px.histogram(df, x=gini, color='year', facet_col='year')
2 fig.layout.yaxis.ticksuffix = '%'
3 fig.layout.yaxis.title = 'Százalékos gyakoriság'
4 fig.show()
```

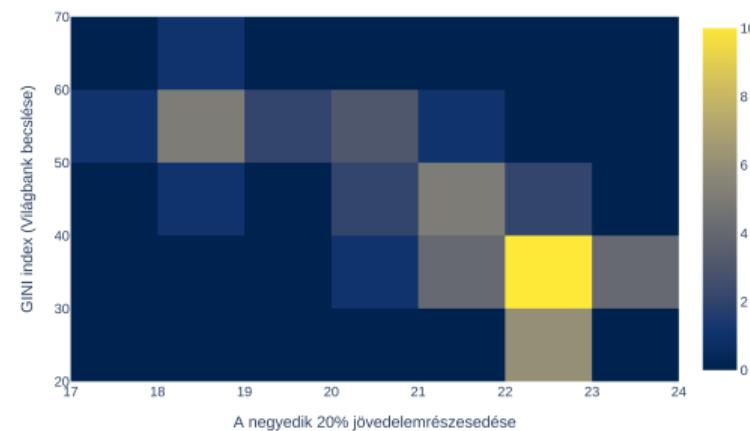


Hisztogramok több dimenzióban

2D hisztogram

Két dimenzióban osztja fel az adatokat, és minden cella (osztályköz) azt mutatja meg, hogy hány adatpont esik az adott tartományba mindkét változó esetében.

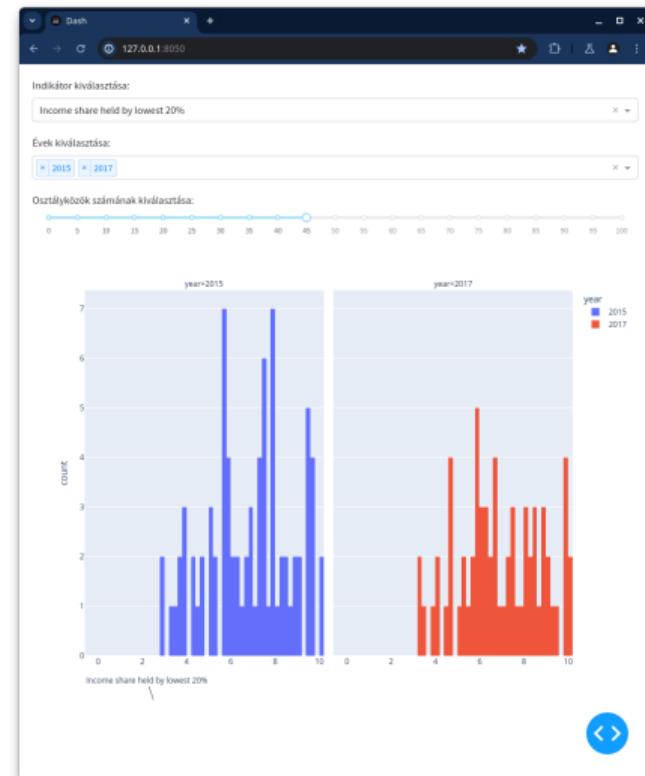
```
1 fig = go.Figure()
2 fig.add_histogram2d(
3     x=df['Income share held by fourth 20%'],
4     y=df['GINI index (World Bank estimate)'],
5     colorscale='cividis'
6 )
```



Alkalmazás interaktív hisztogramokkal (freq_app_v1.py)

A callback függvények a felhasználói interakciók alapján frissítik a grafikonokat. A `display_histogram` függvény három bemeneti elemet figyel (`years`, `indicator`, `bins`), és ezek alapján frissíti a hisztogram ábrát.

Ha nincs kiválasztott év vagy indikátor, a függvény nem frissíti az ábrát (PreventUpdate). Az adatok szűrése után a Plotly Express segítségével készül el a hisztogram.



Interaktív hisztogramok beépítése az alkalmazásba (app_v4_1.py)



1 Bevezetés

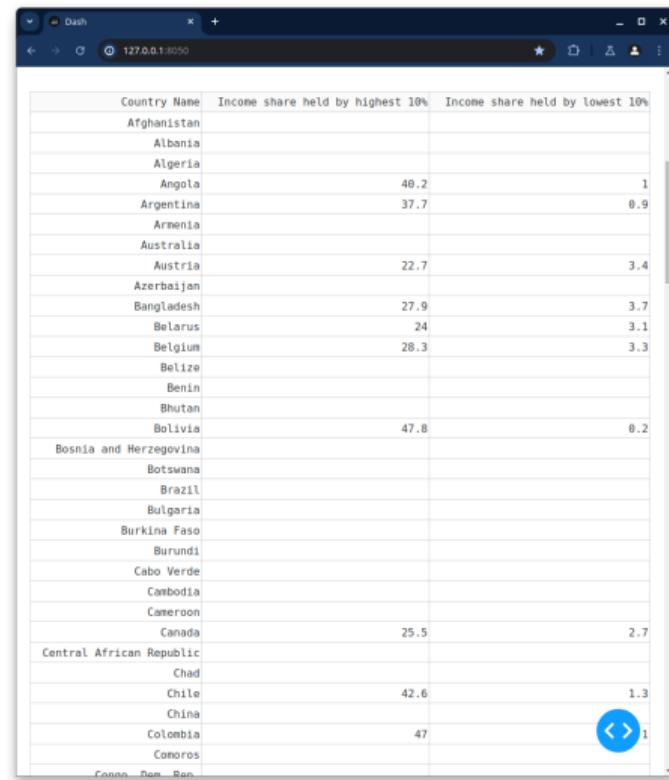
2 Adattáblák

3 Gépi tanulás

Adattábla létrehozása

A Dash keretrendszerben interaktív táblázatokat a dash_table könyvtárral lehet létrehozni.

```
1 from dash import html, dash_table
2
3 app.layout = html.Div([
4     ...
5     dash_table.DataTable(
6         data=pov_df.to_dict('records'),
7         columns=[{'name': col, 'id': col}
8                  for col in pov_df.columns]
9     )
10    ...
11])
```

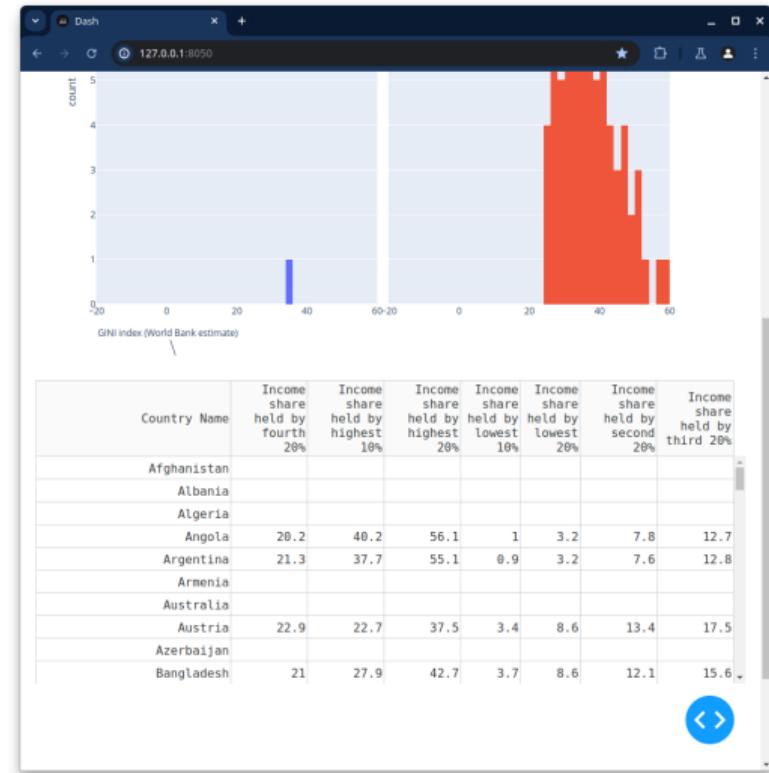


The screenshot shows a web browser window titled 'Dash' with the URL '127.0.0.1:8050'. The page displays an interactive table with three columns: 'Country Name', 'Income share held by highest 10%', and 'Income share held by lowest 10%'. The table lists various countries with their corresponding income share percentages. At the bottom right of the table, there is a blue circular button with a white double-headed arrow icon.

Country Name	Income share held by highest 10%	Income share held by lowest 10%
Afghanistan		
Albania		
Algeria		
Angola	48.2	1
Argentina	37.7	0.9
Armenia		
Australia		
Austria	22.7	3.4
Azerbaijan		
Bangladesh	27.9	3.7
Belarus	24	3.1
Belgium	28.3	3.3
Belize		
Benin		
Bhutan		
Bolivia	47.8	0.2
Bosnia and Herzegovina		
Botswana		
Brazil		
Bulgaria		
Burkina Faso		
Burundi		
Cabo Verde		
Cambodia		
Cameroon		
Canada	25.5	2.7
Central African Republic		
Chad		
Chile	42.6	1.3
China		
Colombia		
Comoros	47	
Conor. Dem. Rep.		

Adattábla személyre szabása

```
1 dash_table.DataTable(  
2     data=pov_df.to_dict('records'),  
3     columns=[{'name': col, 'id': col} for  
4         col in pov_df.columns],  
5     style_header={'whiteSpace': 'normal'  
6         },  
7     fixed_rows={'headers': True},  
8     style_table={'height': '400px'},  
9     virtualization=True,  
10    )
```



1 Bevezetés

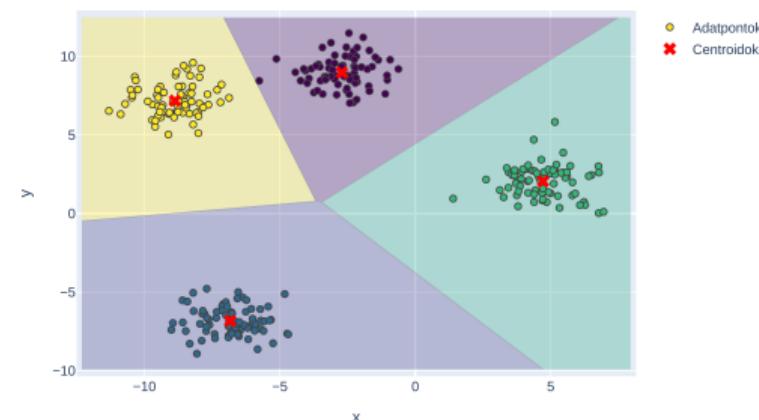
2 Adattáblák

3 Gépi tanulás

K-közép klaszterezés

Az algoritmus eljárása:

- ❶ K számú klaszter centroid inicializálása véletlenszerűen:
 $\mu_1, \mu_2, \dots, \mu_K$
- ❷ minden x_i adatpont a hozzá legközelebb eső klaszterhez rendelése az euklideszi távolságot használva:
 $c_i = \arg \min_j \|x_i - \mu_j\|^2$
- ❸ Klaszterközéppontok újraszámítása úgy, hogy az adott klaszterhez tartozó pontok várható értékét tükrözzék:
 $\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$
- ❹ Ismétlés a kilépési kritériumig



Optimális klaszterszám megtalálása

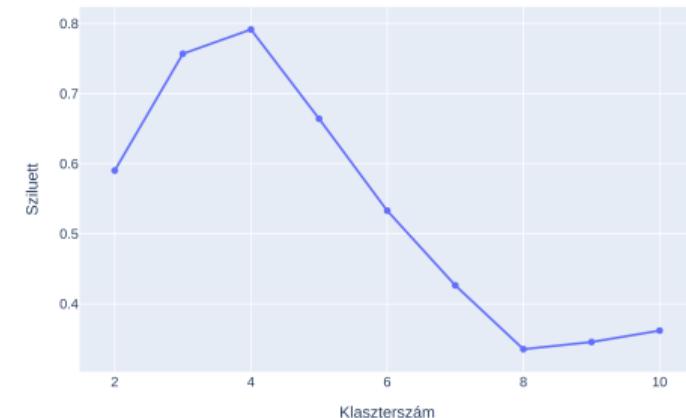
Egy klaszter konfiguráció annál jobb, minél szorosabban helyezkednek el az egy klaszterben lévő egyedek, és minél jobban elkülönülnek a más klaszterben lévő egyedektől. Ezt tükrözi a sziluett együttható:

Sziluett

$$S(x) \in [-1, 1] = \frac{a(x) - b(x)}{\max\{a(x), b(x)\}}$$

Ahol:

- $a(x)$: x mintaegyed és minden vele nem egy klaszterben lévő mintaegyed távolsága
- $b(x)$: x mintaegyed és minden vele egy klaszterben lévő mintaegyed távolsága



Scikit-learn K -közép

K -közép modul importálása és tanítása a make_blobs adathalmazon:

```
1 from sklearn.cluster import KMeans
2
3 kmeans = KMeans(n_clusters=n_clusters,
4   random_state=random_state)
5 kmeans.fit(X)
6
7 labels = kmeans.labels_
centroids = kmeans.cluster_centers_
```

Klaszter centroidok x és y koordinátái:

```
1 In [1]: print(kmeans.cluster_centers_)
2 Out [1]: [[-2.70981136  8.97143336]
3           [-6.83235205 -6.83045748]
4           [ 4.7182049   2.04179676]
5           [-8.87357218  7.17458342]]
```

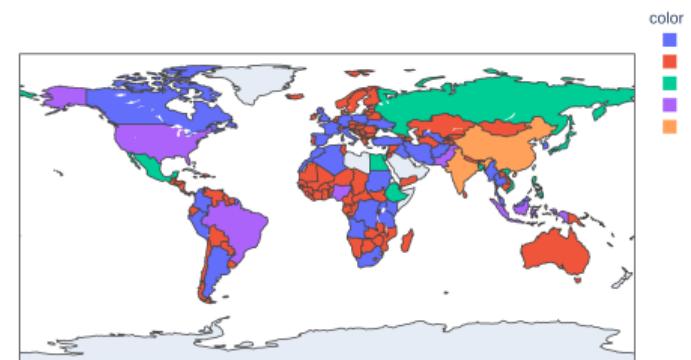
Becsült címkék az adathalmaz első 10 elemére:

```
1 In [2]: print(kmeans.labels_)
2 Out [2]: [3 3 0 1 3 1 2 1 0 2]
```

Országok klaszterezése

A következő program az országokat 5 klaszterbe sorolja be, majd ennek a kimenetét felhasználva hoz létre egy tematikus térképet:

```
1 year = 2018
2 indicators = ['Population', 'total']
3 kmeans = KMeans(n_clusters=5)
4 df = poverty[poverty['year'].eq(year) &
               poverty['is_country']]
5 data = df[indicators].values
6 kmeans.fit(data)
7
8 fig = px.choropleth(
9     df,
10    locations='Country Name',
11    locationmode='country names',
12    color=[str(x) for x in kmeans.labels_
13        ])
```



Hiányzó értékek kezelése

Az `sklearn.impute.SimpleImputer` lehetővé teszi a hiányzó értékek pótlását különböző stratégiák alkalmazásával.

Néhány gyakori stratégia:

- `mean` (átlag): A hiányzó értékeket az adott oszlop átlagával helyettesíti
- `median` (medián): A hiányzó értékeket az adott oszlop mediánjával helyettesíti
- `most_frequent` (leggyakoribb): A hiányzó értékeket az adott oszlop leggyakoribb értékével
- `constant` (állandó): Egy megadott állandó értékkel helyettesíti a hiányzó értékeket.

SimpleImputer használata:

```
1 from sklearn.impute import
   SimpleImputer
2
3 data = np.array([1, 2, 1, 2, np.nan]).
   reshape(-1, 1)
4 imp = SimpleImputer(strategy='mean')
5 imp.fit(data)
6
7 print(imp.transform(data))
```

```
1 [[1. ]]
2 [2. ]
3 [1. ]
4 [2. ]
5 [1.5]]
```

Adatok méretezése

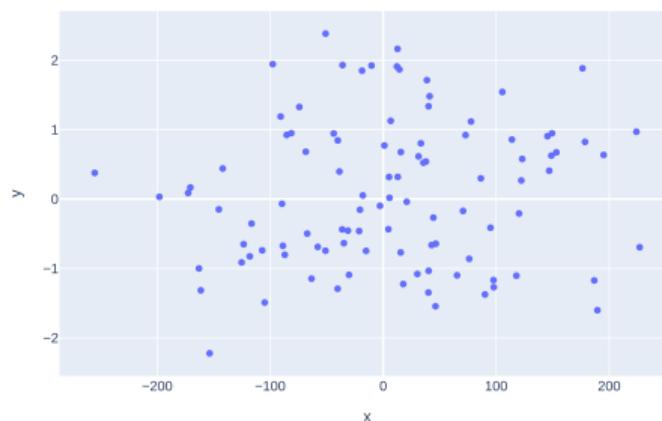
```
1 from sklearn.preprocessing import StandardScaler  
2  
3 data = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)  
4  
5 scaler = StandardScaler()  
6 print(scaler.fit_transform(data))
```

```
1 [[-1.41421356]  
2 [-0.70710678]  
3 [ 0. ]  
4 [ 0.70710678]  
5 [ 1.41421356]]
```

Méretezés

Adat előkészítési technika, mely során az adatok értékei egy adott tartományba transzformálódnak.

Méretezés előtt



Adatok méretezése

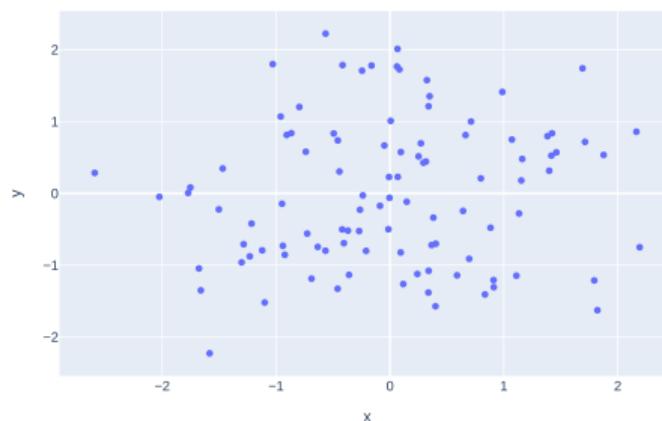
```
1 from sklearn.preprocessing import StandardScaler  
2  
3 data = np.array([1, 2, 3, 4, 5]).  
    reshape(-1, 1)  
4  
5 scaler = StandardScaler()  
6 print(scaler.fit_transform(data))
```

```
1 [[-1.41421356]  
2 [-0.70710678]  
3 [ 0. ]  
4 [ 0.70710678]  
5 [ 1.41421356]]
```

Méretezés

Adat előkészítési technika, mely során az adatok értékei egy adott tartományba transzformálódnak.

Méretezés után



Alkalmazás interaktív térképpel, K-közép klaszterezéssel (kmeans_app.py)

A callback függvény frissíti a térképet a kiválasztott év, klaszterszám és indikátor alapján.

A hiányzó értékeket az átlaggal pótolja, majd az adatokat skálázza. A K-Közép algoritmust alkalmazza a transzformált adatokra, és a klaszterszámot a rendelkezésre álló adatok alapján korlátozza. Végül létrehozza a térképet, ahol az országokat a klaszter címkék alapján színezi, és finomhangolja a megjelenést.

