

Adatbányászat a Gyakorlatban

3. Gyakorlat: Dash diagramok

Kuknyó Dániel
Budapesti Gazdasági Egyetem

2024/25
1.félév

- 1 Bevezetés
- 2 Grafikonok szerkesztése
- 3 Plotly express

- 1 Bevezetés
- 2 Grafikonok szerkesztése
- 3 Plotly express

A Figure objektum

A Plotly-ban a Figure objektum egy magasszintű adatstruktúra, amely egyesíti a diagram adatait és elrendezését egyetlen objektumban. Olyan attribútumokat tartalmaz, mint a data és layout.

```
1 import plotly.graph_objects as go
2
3 fig = go.Figure()
```

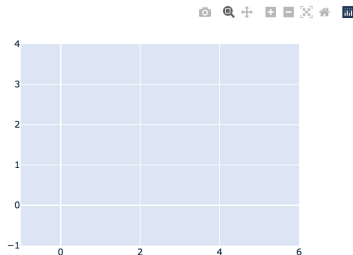


Figure objektumok attribútumai

data

Ez az attribútum tartalmazza a diagramon megjelenítendő adatokat. Ez egy lista, amely különböző nyomokat (trace) tartalmaz.

trace

Egy trace vagy nyom egy adatcsoportot képvisel a diagramon belül. Minden nyomnak megvan a maga típusa (pl. kör, szórás, oszlop) és különböző tulajdonságokkal rendelkezik, mint az x és y tengely értékei, színek, név stb...

layout

Ez az attribútum határozza meg a diagram elrendezését és stílusát. Ide tartoznak a tengelyek címkéi, a diagram címe, a háttérszínek, a margók, a legendák és egyéb vizuális elemek. A layout attribútum egy szótár, amely különböző kulcs-érték párokat tartalmaz.

A data attribútum

A Figure objektumot a `plotly.graph_objects` könyvtár tartalmazza. Példányosítás után az `add_scatter()` függvény meghívásával hozzáadódik egy új nyom a vászonhoz. A nyom az x és y koordinátákat tartalmazza a pontdiagramhoz.

```
1 import plotly.graph_objects as go
2
3 fig = go.Figure()
4 fig.add_scatter(x=[1, 2, 3], y=[4, 2, 3])
5 fig.show()
```

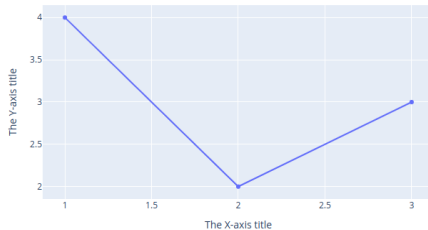


A layout attribútum

A layout attribútum módosítása közvetlen hozzáféréssel lehetséges. Ez egy fastruktúrájú adatszerkezet, ahol az attribútumoknak alárendelt attribútumai vannak.

```
1 fig.layout.title = 'The Figure Title'
2 fig.layout.xaxis.title = 'The X-axis
  title'
3 fig.layout.yaxis.title = 'The Y-axis
  title'
```

The Figure Title



- 1 Bevezetés
- 2 **Grafikonok szerkesztése**
- 3 Plotly express

Grafikon renderelése json formátumba

A `show()` függvény több módot is biztosít egy függvény megjelenítésére. Az egyik ilyen a `fig.show('json')`.

Ennek segítségével meg lehet vizsgálni a diagram renderelése közben létrejövő fa struktúrát.

```
1 fig.show('json')  
▼ root:  
  ► data: [] 2 items  
  ► layout:
```

```
1 fig.show('json')  
▼ root:  
  ► data: [] 2 items  
  ▼ layout:  
    ► template:  
      ▼ title:  
        text: "The Figure Title"  
    ▼ xaxis:  
      ▼ title:  
        text: "The X-axis title"  
    ▼ yaxis:  
      ▼ title:  
        text: "The Y-axis title"
```

```
1 fig.show('json')  
▼ root:  
  ▼ data: [] 2 items  
    ▼ 0:  
      type: "scatter"  
      ▼ x: [] 3 items  
        0: 1  
        1: 2  
        2: 3  
      ► y: [] 3 items  
        ► 1:  
          layout:  
            ► template:  
              ▼ title:  
                text: "The Figure Title"  
            ► xaxis:  
            ► yaxis:
```

Grafikon konfigurálása

A config paraméter egy dict objektumot vár el, és több tulajdonságát is képes vezérelni a diagramnak:

```
1 fig.show(  
2     config={  
3         'displaylogo': False,  
4         'modeBarButtonsToAdd': ['drawrect',  
5                                 'drawcircle', 'eraseshape']  
6     }  
7 )
```

- **displayModeBar**: A teljes menüsáv mutatása, alapértelmezése True
- **responsive**: Változzon-e a diagram mérete a böngésző méretnek megfelelően. Alapértelmezése a True.
- **toImageButtonOptions**: A kép letöltésének alapértelmezett formátumát adja meg.
- **modeBarButtonsToRemove**: Azon menügombok listája, amiket ne jelenítsen meg a diagram.

Vezérlőkkel összekapcsolt diagramok

Egy interaktív diagram létrehozása az elrendezésben:

```
1 app.layout = html.Div([
2     dcc.Dropdown(
3         id='year_dropdown',
4         value='2010',
5         options=[{'label': year, 'value':
6                     str(year)} for year in range
7                     (1974, 2019)]
8     ),
9     dcc.Graph(id='population_chart')
```

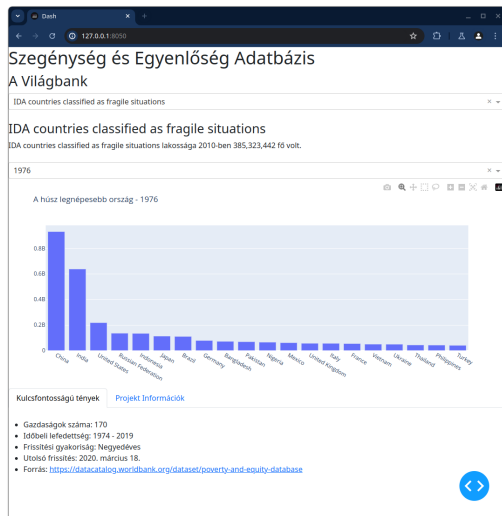
A Dropdown komponensben definiálva vannak a lehetséges értékek, és a Graph egy egyedi id adattaggal van ellátva az interaktivitás miatt.

Az ehhez a diagramhoz tartozó callback függvény a legördülő menü értékét kapja meg paraméterül, az implementációjában leszűri a táblát, majd egy Figure objektumot térít vissza, ami felülírja a population_chart figure attribútumát:

```
1 @app.callback(
2     Output('population_chart', 'figure'),
3     Input('year_dropdown', 'value')
4 )
5 def plot_countries_by_population(year):
6     year_df = ...
7     fig = go.Figure()
8     ...
9     return fig
```

Alkalmazás felépítése interaktív diagrammal (app_v2_1.py)

- 1 Pandas importálása, és a szegénységi adatokat tartalmazó fájl megnyitása.
- 2 Régiók kizárása az adatkészletből, hogy csak országok maradjanak.
- 3 Egy DataFrame létrehozása, amely csak az országok teljes népességét tartalmazza.
- 4 Egy legördülő menü segítségével ki lehet választani az évet, amely leszűri az adathalmazt, hogy az abból az évből legnépesebb országokat reprezentálja.
- 5 Oszlopdiagram létrehozása, amely a legnépesebb országokat tartalmazza.



Dash vizuális debugger

A vizuális debugger a Dash-ben egy eszköz, amely segít a fejlesztőknek nyomon követni és hibakeresni a Dash alkalmazásaikat. A vizuális debugger lehetővé teszi, hogy valós időben lássuk az alkalmazás állapotát, a komponensek közötti adatáramlást és az eseményeket:

- Komponensek hierarchiája
- Állapot és tulajdonságok
- Események és változások
- Hibák és figyelmeztetések

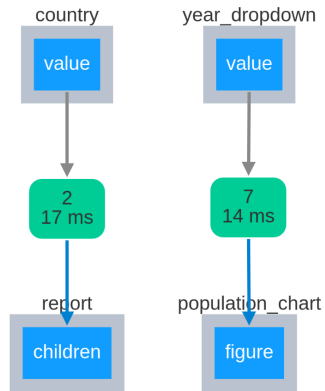
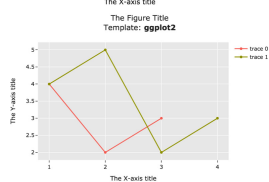
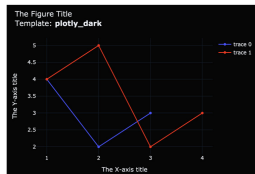
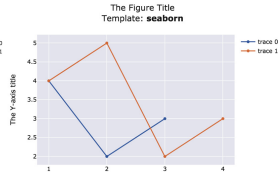
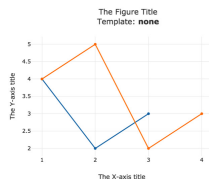


Diagram témák szerkesztése

A diagramok témájának szerkesztése nagyon sok időt megspórolhat, és egy általános megoldást nyújt arra, hogy minden diagram témáját egyszerre lehessen változtatni.

Ez a layout alatt a `template` attribútum módosításával érhető el.

```
1 fig.layout.template = template_name
```



- 1 Bevezetés
- 2 Grafikonok szerkesztése
- 3 Plotly express

Példa adathalmaz definiálása

A következőkben a következő egyszerű adattáblával készült diagramok lesznek láthatók:

```
1 df = pd.DataFrame({
2     'numbers': [1, 2, 3, 4, 5, 6,
3                 7, 8],
4     'colors': ['blue', 'green', 'orange', 'yellow', 'black',
5               'gray', 'pink', 'white'],
6     'floats': [1.1, 1.2, 1.3, 2.4, 2.1, 5.6, 6.2, 5.3],
7     'shapes': ['rectangle', 'circle', 'triangle', 'rectangle', 'circle',
8               'triangle', 'rectangle', 'circle'],
9     'letters': list('AAABBCCC')
10 })
```

	numbers	colors	floats	shapes	letters	
1	0	1	blue	1.1	rectangle	A
2	1	2	green	1.2	circle	A
3	2	3	orange	1.3	triangle	A
4	3	4	yellow	2.4	rectangle	B
5	4	5	black	2.1	circle	B
6	5	6	gray	5.6	triangle	C
7	6	7	pink	6.2	rectangle	C
8	7	8	white	5.3	circle	C

Pontdiagram

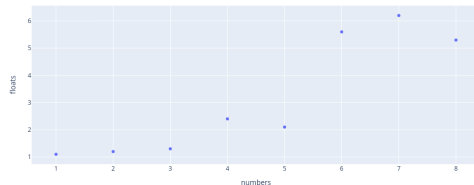
Egy plotly express diagramnak kétféleképpen is át lehet adni az adathalmazt.

Az első esetben a DataFrame kerül átadásra, és az x, y paraméterek a DataFrame oszlopaira hivatkoznak:

```
1 px.scatter(data_frame=df, x='numbers',  
             y='floats')
```

A másik esetben pedig közvetlenül vannak hivatkozva az oszlopok:

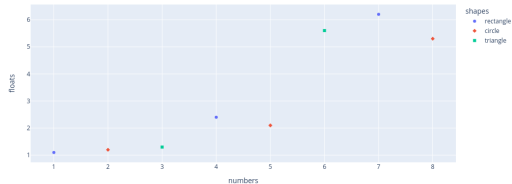
```
1 px.scatter(x=df['numbers'], y=df['  
             floats'])
```



Pontdiagram kategóriákkal

Ebben az esetben minden adatosztály egy külön nyomként jelenik meg.

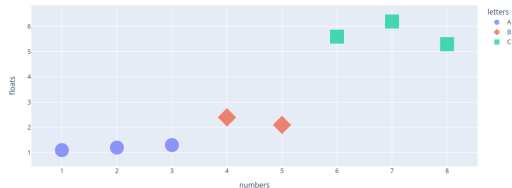
```
1 px.scatter(df, x='numbers', y='floats',  
            color='shapes', symbol='shapes')
```



Pontdiagram jelölőkkel

Minden (x, y) adatpont jelölőjét lehetséges külön állítani. Ezeknek testre lehet szabni a színét, méretét:

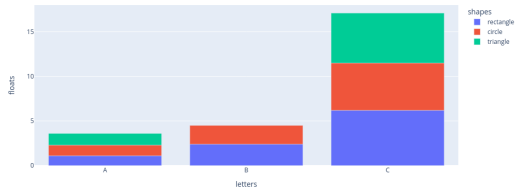
```
1 px.scatter(df, x='numbers', y='floats',  
             color='letters', symbol='letters',  
             size=[35] * 8)
```



Rakott oszlopdiagram

A rakott oszlopdiagram több oszlopdiagram együttese. Ebben az esetben is minden adatcsoport egy külön nyomként jelenik meg az adatszerkezetben. A csoportosítási változót a color attribútum adja meg.

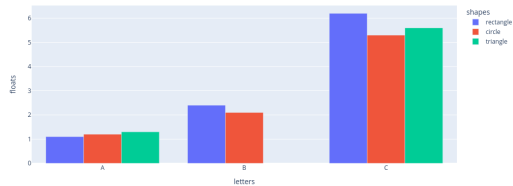
```
1 px.bar(df, x='letters', y='floats',  
         color='shapes')
```



Csoportosított oszlopdiagram

Csoportosítás esetén az adatcsoportok nem egymáson, hanem egymás mellett foglalnak helyet. A csoportosítás attribútuma itt is a `color`, és a csoportosítási típust a `barmode=color` adja meg.

```
1 px.bar(df, x='letters', y='floats',  
        color='shapes', barmode='group')
```



Összetett plotly express diagram létrehozása (px_app.py)

- 1 Adathalmazok beolvasása, transzformációja és megfelelő formára hozása
- 2 Változók létrehozása, amik megadják a szűrési kritériumokat: year, indicator, grouper
- 3 Adathalmaz leszűrése a változók alapján
- 4 A `px.scatter()` meghívása egy Dash objektum Div komponensén

