# charts.py modul működése

## Összefoglaló

A create\_chart kezeli az adatokat és a felhasználói beállításokat, létrehozza a diagrammokat.

Az add\_linear\_regression\_combined függvény pedig a lineráis regresszió kiszámítsáért és hozzáadásáért felel.

## Részletes magyarázat:

1. A megadott paraméterek alapján egy diagrammot hoz létre:
   * data: az adatok, amit a diagram megjelenítéséhez használunk
   * selected\_columns: a kiválasztott oszlopokat jelöli
   * income\_types: a jövedelem típusok, amit meg kell jeleníteni
   * chart\_types: amilyen grafikont szeretnénk megjeleníteni

def create\_chart(data, selected\_columns, income\_types, chart\_types):

1. Üres grafikon mellőzése, ha nincs elegendő paraméter megadva, ne jelenítsen meg grafikont

if not selected\_columns or not income\_types or not chart\_types:

1. grafikon létrehozása a Figure objektummal

        return go.Figure()

1. szűrés az adott jövedelem típusra

for income\_type in income\_types:

        for column in selected\_columns:

            filtered\_data = data[data['Jövedelem\_típus'] == income\_type] if 'Jövedelem\_típus' in data.columns else pd.DataFrame()

1. ellenőrizzük, hogy van-e adat, és ha nem üres, akkor folytatjuk

if filtered\_data.empty or column not in filtered\_data.columns:

                continue

            # Csak érvényes adatok megjelenítése

            filtered\_data\_nonan = filtered\_data[['Év', column]].dropna()

            if filtered\_data\_nonan.empty:

            continue

1. grafikon típusának feldolgozása, az x tengelyen az „Év” az y tengelyen az adott érték helyezkedik el. beállítjuk a diagram típusát, vonal, vagy pont, és a diagram jelmagyarázatának cimkéjét.

 for chart\_type in chart\_types:

                if chart\_type == 'Vonal':

                    fig.add\_trace(go.Scatter(

                        x=filtered\_data\_nonan['Év'],

                        y=filtered\_data\_nonan[column],

                        mode='lines',

                        name=f"{income\_type} - {column}",

                        showlegend=True

                    ))

                elif chart\_type == 'Pont':

                    fig.add\_trace(go.Scatter(

                        x=filtered\_data\_nonan['Év'],

                        y=filtered\_data\_nonan[column],

                        mode='markers',

                        name=f"{income\_type} - {column}",

                        showlegend=True

                    ))

1. amennyiben a felhasználó kiválasztotta a lineáris regresszió grafikont, akkor meghívjuk az add\_linear\_regression\_combined függvényt, ami kiszámítja a lineáris regressziót a kiválasztott oszlopok szerint.

if 'Lineáris regresszió' in chart\_types:

        for income\_type in income\_types:

            income\_filtered\_data = data[data['Jövedelem\_típus'] == income\_type]

            fig = add\_linear\_regression\_combined(

                fig,

                income\_filtered\_data,

                x\_column='Év',

                y\_columns=selected\_columns,

                label=income\_type

            )

1. frissíti a diagramot, beállítja a diagram címét, az x és y tengely feliratát, elhelyezi a diagramhoz tartozó magyarázatot, meghatározza a diagram és a magyarázat egymáshoz viszonyított helyzetét.

fig.update\_layout(

        title="Jövedelem Alakulása",

        xaxis\_title="Év",

        yaxis\_title="Jövedelem (Ft)",

        legend=dict(

            title="Típus",

            orientation="h",

            x=0.5,

            y=-0.2,

            xanchor="center",

            yanchor="top"

        ),

        showlegend=True,

        margin=dict(l=50, r=50, t=50, b=150)  # Extra alsó margó a magyarázat számára

    )

    return fig

1. hozzáad egy lineáris regresszió vonalat a megadott adatok alapján

def add\_linear\_regression\_combined(fig, filtered\_data, x\_column, y\_columns, label):

1. összevonja az adatokat, hogy ki tudja rajzolni a regressziós vonalat

def add\_linear\_regression\_combined(fig, filtered\_data, x\_column, y\_columns, label):

1. ha nincs elég adat, nem adunk hozzá regressziós vonalat

if combined\_data.empty or len(combined\_data) < 2:

        return fig

1. az x tengely adatait az Év oszlopból veszi, az y értékeket az összes kiválasztott oszlop átlagaként veszi, a LinearRegression() pedig kiszámítja a legjobban illeszkedő vonalat.

 x\_values = combined\_data[x\_column].values.reshape(-1, 1)

    y\_values = combined\_data[y\_columns].mean(axis=1).values

    model = LinearRegression()

    model.fit(x\_values, y\_values)

    y\_pred = model.predict(x\_values)

1. a lineáris regressziós grafikont hozzáadja a diagramhoz

fig.add\_trace(go.Scatter(

        x=combined\_data[x\_column],

        y=y\_pred,

        mode='lines',

        name=f"{label} - Lineáris Regresszió",

        line=dict(dash='dash')

    ))