BVP prognozavimas

Tautvydas Lukšas

Užduotis

- 1. Surasti top 10 šalių, kurios labiausiai paaugo BVP atžvilgiu.
- 2. Nubrėžti grafikus, kurie iliustruotų, kaip keitėsi šalių populiacija iš The Organization for Economic Cooperation and Development (OECD).
- 3. Suskirstykite šalis į 5 klasterius naudodamiesi GDP ir "Volume of exports of goods".
- 4. Sukurkite modelį, kuris prognozuoja “Gross domestic product per capita”. Būkite atidūs ir nenaudokite laukų, kurie tiesiogiai susiję su GDP.
- 5. Supaprastinkite 4 punkte sukurtą modelį taip, kad jis būtų kuo tikslesnis ir turėtų ne daugiau 5 kintamųjų (features).

Duomenys Excel failas

https://www.imf.org/-/media/Files/Publications/WEO/WEO-Database/2022/WEOOct2022all.ashx

Duomenų apibrėžimas

https://www.imf.org/en/Publications/WEO/weo-database/2022/October/download-entire-database

Kas yra BVP ?

Bendrasis vidaus produktas (**BVP**) (angl. *gross domestic product – GDP*) – vienas iš pagrindinių rodiklių, rodančių šalies ekonomikos išsivystymo lygį.
Bendrasis vidaus produktas yra apibrėžiamas kaip galutinė prekių ir paslaugų sukurtų šalyje rinkos vertė per tam tikrą laiko tarpą.
Dažniausiai naudojamas būdas matuoti ir suprasti BVP yra išlaidų metodas:

BVP = vartojimas + investicijos + valstybės išlaidos + (eksportas – importas)

Duomenys

- Duomenys pateikti užduotyje
- Reikėjo rasti sprendimą, kad galima būtų įsikelti į jupyter notebook
- Pagal nuorodą parsisiunčia exel failas, netinkamo formatavimo
- Išsisaugojau kaip csv faila su UTF8 formatavimu ir „suveikė“
- Nusiskaičius duomenis, matome lentelę kur vienai šaliai yra apie 44 eilutės su įvairiais matavimo vienetais ir duomenimis nuo 1980 iki 2027 metų

Duomenys ir ju analizė

	WEO Country Code	ISO	WEO Subject Code	Country	Subject Descriptor	Subject Notes	Units	Scale	Country/Series-specific Notes	1980	...	2019	2020	2021	2022	2023	2024	2025	2026	2027	Estimates Start After
0	512	AFG	NGDP_R	Afghanistan	Gross domestic product, constant prices	Expressed in billions of national currency uni...	National currency	Billions	Source: National Statistics Office Latest actu...	NaN	...	1,319.90	1,288.87	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0
1	512	AFG	NGDP_RPCH	Afghanistan	Gross domestic product, constant prices	Annual percentages of constant price GDP are y...	Percent change	NaN	See notes for: Gross domestic product, consta...	NaN	...	3.912	-2.351	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0
2	512	AFG	NGDP	Afghanistan	Gross domestic product, current prices	Expressed in billions of national currency uni...	National currency	Billions	Source: National Statistics Office Latest actu...	NaN	...	1,469.60	1,547.29	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0
3	512	AFG	NGDPD	Afghanistan	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...	NaN	...	18.876	20.136	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0

- Išsitrinu paskutines eilutes, jokių duomenų jose nėra

```
dfcsv.reset_index()
dfcsv = dfcsv.drop([8624,8625])
dfcsv
```

8624	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8625	International Monetary Fund, World Economic Ou...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

- Pasitikrinam duomenų tipą ir non null reikšmes

```
# pasitikrinam lenteles reiksmiu tipus
dfcsv.info()
```

executed in 105ms, finished 08:36:42 2023-05-18

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8624 entries, 0 to 8623
Data columns (total 58 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   WEO Country Code               8624 non-null   object
 1   ISO                            8624 non-null   object
 2   WEO Subject Code               8624 non-null   object
 3   Country                        8624 non-null   object
 4   Subject Descriptor             8624 non-null   object
 5   Subject Notes                  8624 non-null   object
 6   Units                          8624 non-null   object
 7   Scale                          3920 non-null   object
 8   Country/Series-specific Notes  7641 non-null   object
 9   1980                           3886 non-null   object
 10  1981                           4007 non-null   object
 11  1982                           4050 non-null   object
 12  1983                           4091 non-null   object
 13  1984                           4117 non-null   object
 14  1985                           4194 non-null   object
 15  1986                           4241 non-null   object
 16  1987                           4265 non-null   object
```

- Atsirenkam reikšmes kur BVP yra procentinis pokytis

In [6]:
```
# issifiltruojam kur yra 'Gross domestic product, constant prices' reiksmes su Percent change
df_top = dfcsv[dfcsv['WEO Subject Code'].str.contains('NGDP_RPCH')]
df_top
```

executed in 69ms, finished 08:36:42 2023-05-18

	WEO Country Code	ISO	WEO Subject Code	Country	Subject Descriptor	Subject Notes	Units	Scale	Country/Series-specific Notes	1980	...	2019	2020	2021	2022	2023	2024	2025	2026	2027	Estimates Start After
1	512	AFG	NGDP_RPCH	Afghanistan	Gross domestic product, constant prices	Annual percentages of constant price GDP are y...	Percent change	NaN	See notes for: Gross domestic product, consta...	NaN	...	3.912	-2.351	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0
45	914	ALB	NGDP_RPCH	Albania	Gross domestic product, constant prices	Annual percentages of constant price GDP are y...	Percent change	NaN	See notes for: Gross domestic product, consta...	2.684	...	2.088	-3.482	8.516	4	2.5	3.2	3.2	3.4	3.4	2020.0
89	612	DZA	NGDP_RPCH	Algeria	Gross domestic product, constant prices	Annual percentages of constant price GDP are y...	Percent change	NaN	See notes for: Gross domestic product, consta...	-5.4	...	1	-5.1	3.5	4.665	2.568	1.968	1.853	1.884	1.692	2021.0
133	171	AND	NGDP_RPCH	Andorra	Gross domestic product, constant prices	Annual percentages of constant price GDP are y...	Percent change	NaN	See notes for: Gross domestic product, consta...	NaN	...	2.016	-11.184	8.949	6.575	2	2.4	2.1	1.7	1.5	2021.0
177	614	AGO	NGDP_RPCH	Angola	Gross domestic product, constant prices	Annual percentages of constant price GDP are y...	Percent change	NaN	See notes for: Gross domestic product, consta...	2.406	...	-0.702	-5.75	0.804	2.866	3.427	3.876	4.024	4.104	3.925	2021.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Kodėl procentinis pokytis?

- BVP pokytis yra matuojamas procentine išraiška lyginant su praėjusiais metais (BVP tikroji vertė yra skaičiuojama kaip visų prekių ir paslaugų suma per praėjusius metus ir iškaiciuojama inflacija, ir tai itakoja tikrąjį augimą, o ne kainų pokytį)
- Nes pvz BVP yra 100 mln 2020 metais ir 105 mln 2021 = augimas 5%,
- Bet tai neivertina infliacijos pokycio, nes sakykim BVP yra 100 mln 2020 metais ir 110 mln 2021 , bet infliacija 10% = 0% augimo, o imant tik vertes, turėtume 10% augimą.
- Todėl didžiausa problema yra pasirinkti tinkamas reikšmes augimo skaičiavimui. Ir jei duomenyse turime procentinį BVP pokytį, jį reikia ir naudoti.

- Pasiliekam reikalingus stulpelius

```
df_top1 = df_top.drop(['WEO Country Code', 'ISO', 'WEO Subject Code','Subject Descriptor', 'Subject Notes', 'Units', 'Scale', 'Country/Series-specific Notes', '2022', '20
df_top1
```

executed in 27ms, finished 08:36:42 2023-05-18

	Country	1980	1981	1982	1983	1984	1985	1986	1987	1988	...	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
1	Afghanistan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	13.968	5.683	2.697	0.988	2.164	2.647	1.189	3.912	-2.351	NaN
45	Albania	2.684	5.7	2.9	1.1	2	-1.5	5.6	-0.8	-1.4	...	1.418	1.002	1.774	2.219	3.315	3.802	4.019	2.088	-3.482	8.516
89	Algeria	-5.4	3	6.4	5.4	5.6	5.6	-0.2	-0.7	-1.9	...	3.4	2.8	3.8	3.7	3.2	1.4	1.2	1	-5.1	3.5
133	Andorra	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	-4.974	-3.548	2.504	1.434	3.71	0.346	1.589	2.016	-11.184	8.949
177	Angola	2.406	-4.4	--	4.2	6	3.5	2.9	4.083	6.129	...	8.542	4.955	4.823	0.944	-2.58	-0.15	-1.316	-0.702	-5.75	0.804
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8405	Vietnam	-3.497	5.797	8.15	7.093	8.397	5.619	3.357	2.549	5.1	...	5.505	5.554	6.422	6.987	6.69	6.94	7.197	7.15	2.944	2.576
8449	West Bank and Gaza	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	6.096	4.699	-0.158	3.721	8.865	1.419	1.227	1.363	-11.318	7.05
8493	Yemen	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	2.393	4.824	-0.189	-27.995	-9.375	-5.072	0.752	1.4	-8.5	-1
8537	Zambia	3.854	6.631	-2.912	-1.145	-1.718	1.237	1.698	1.491	9.271	...	7.598	5.057	4.698	2.92	3.777	3.504	4.035	1.441	-2.785	4.599
8581	Zimbabwe	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	16.658	1.975	2.394	1.794	0.478	4.983	4.732	-6.144	-5.156	7.159

196 rows × 43 columns

- užsipildom NaN reikšmes 0, šiuo atveju preziumuojam, kad BVP nekito ir buvo lygiai toks pat, kaip ir prieš tai metais
- Ir verčiam ‘object’ tipą į ‘float’ / aptikau, kad 1980 metų stulpelyje yra
- ,- -‘ reikšmė, ją pakeičiu irgi į 0

```
df_full = df_full.replace('--', 0)
listprint = df_full['1980'].tolist()
listprint
```

executed in 26ms, finished 08:36:42 2023-05-18

```
 '2.894',
 '2.314',
 0,
 '7.1',
 '7.494',
 '151.644',
 '4.371',
 0,
 '4.444',
 '5.013',
```

- ‚1980’ stulpelį susitvarkę verčiam į ‚float‘

```
# paverciam laikotarpio stulpelius i skaicius ir tuo paciu psitikrinam ar visos reiksmes yra
df_full.iloc[:, 1:43] = df_full.iloc[:, 1:43].apply(pd.to_numeric, errors='coerce')
df_full.info()
```

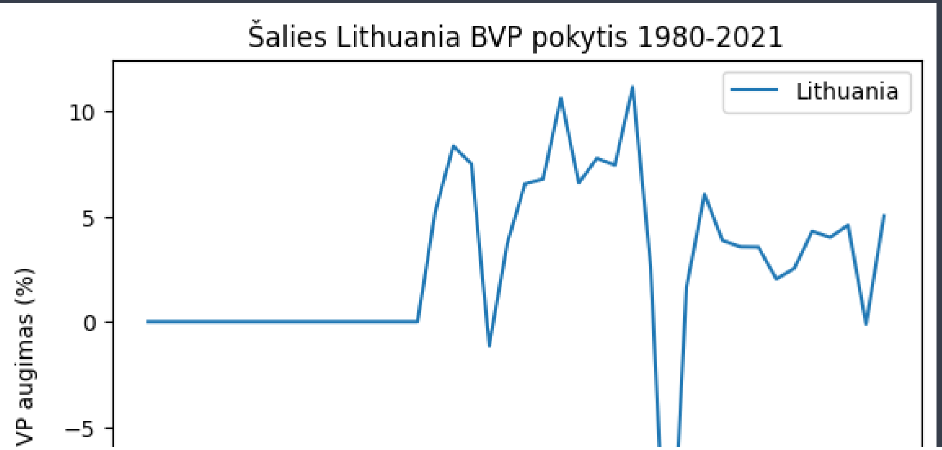
executed in 54ms, finished 08:36:42 2023-05-18

```
 8   1987     196 non-null    float64
 9   1988     196 non-null    float64
 10  1989     196 non-null    float64
 11  1990     196 non-null    float64
 12  1991     196 non-null    float64
 13  1992     196 non-null    float64
 14  1993     196 non-null    float64
 15  1994     196 non-null    float64
 16  1995     196 non-null    float64
 17  1996     196 non-null    float64
 18  1997     196 non-null    float64
 19  1998     196 non-null    float64
 20  1999     196 non-null    float64
 21  2000     196 non-null    float64
 22  2001     196 non-null    float64
 23  2002     196 non-null    float64
 24  2003     196 non-null    float64
 25  2004     196 non-null    float64
```

- Pasidarom kiekvienos šalies atskirai BVP procentini pokytį, įsivertinti duomenis

```
# pasidarom kiekvienos salies kitimo grafika, pasiziureti neatitikciu
for country in df_full['Country'].unique():
    country_data = df_full[df_full['Country'] == country]

    plt.plot(country_data.columns[1:43], country_data.iloc[0, 1:43], label=country)

    plt.xlabel('Metai')
    plt.ylabel('BVP augimas (%)')
    plt.title(f'Šalies {country} BVP pokytis 1980-2021')
    plt.legend()

    plt.xticks(rotation=90)

    plt.show()
```

executed in 1m 5.55s, finished 08:37:48 2023-05-18



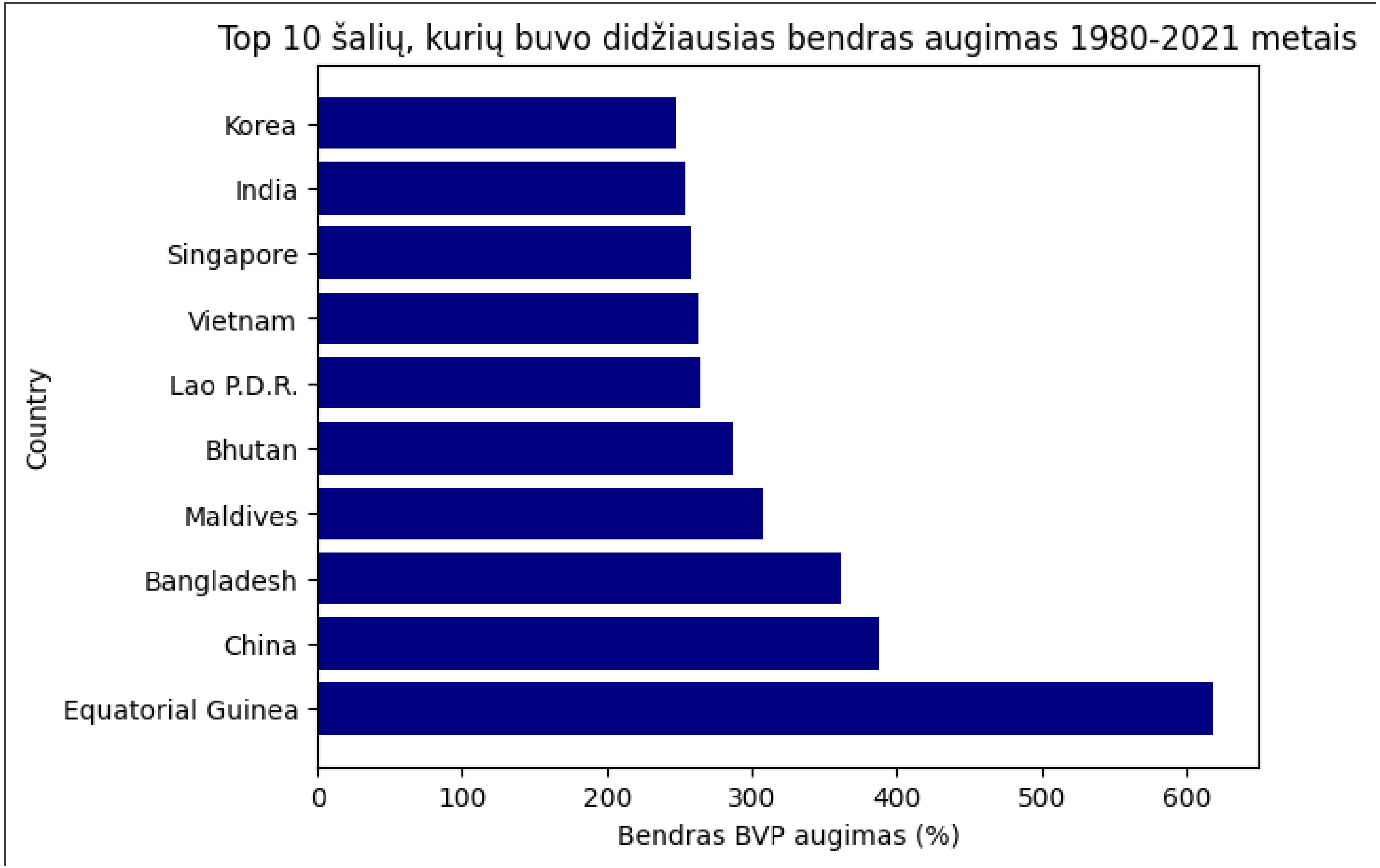
```python
# susumuojam BVP ktimo procentus kiekvienai šaliai
df_full['Total GDP Growth'] = df_full[df_full.columns[1:]].sum(axis=1)

# išsirūšiuojam mažėjančia tvarka ir išsitraukiam top 10
top_10 = df_full.sort_values(by='Total GDP Growth', ascending=False).head(10)

top_10
```

executed in 43ms, finished 08:37:49 2023-05-18

	Country	1980	1981	1982	1983	1984	1985	1986	1987	1988	...	2013	2014	2015	2016	2017	2018	2019	2020	2021	Total GDP Growth
2333	Equatorial Guinea	4.839	5.769	2.202	5.004	1.011	12.905	-2.331	4.437	2.655	...	-4.133	0.415	-9.110	-8.816	-5.668	-6.237	-5.482	-4.241	-3.187	618.780
1585	China	7.910	5.100	9.000	10.800	15.200	13.501	8.597	11.700	11.200	...	7.771	7.391	7.018	6.851	6.947	6.751	5.951	2.244	8.080	387.693
617	Bangladesh	151.644	3.802	2.376	4.016	5.181	3.223	4.249	3.732	2.159	...	6.014	6.061	6.553	7.114	6.590	7.319	7.882	3.448	6.939	361.843
4665	Maldives	18.803	7.886	7.466	4.414	17.379	13.801	8.596	8.866	8.722	...	7.281	7.330	2.885	6.338	7.210	8.123	6.884	-33.500	36.953	307.899
881	Bhutan	4.995	13.589	8.195	7.337	7.616	4.349	7.954	20.229	15.079	...	3.582	3.968	6.221	7.408	6.322	3.835	4.425	-2.348	-3.332	286.709
4137	Lao P.D.R.	10.004	15.332	4.715	3.000	6.438	9.122	4.829	-0.960	-2.100	...	8.026	7.612	7.270	7.023	6.851	6.289	4.652	-0.435	2.061	264.609
8405	Vietnam	-3.497	5.797	8.150	7.093	8.397	5.619	3.357	2.549	5.100	...	5.554	6.422	6.987	6.690	6.940	7.197	7.150	2.944	2.576	263.811
6733	Singapore	10.113	10.816	7.102	8.554	8.792	-0.623	1.343	10.798	11.264	...	4.818	3.936	2.977	3.562	4.661	3.661	1.096	-4.143	7.614	257.628
3389	India	5.281	6.006	3.476	7.289	3.821	5.254	4.777	3.965	9.628	...	6.386	7.410	7.996	8.256	6.795	6.454	3.738	-6.596	8.681	253.562
3961	Korea	-1.646	7.246	8.338	13.376	10.552	7.839	11.327	12.724	11.988	...	3.165	3.202	2.809	2.947	3.160	2.907	2.244	-0.709	4.145	247.758

10 rows × 44 columns

- TOP 10



Top 10 šalių, kurių buvo didžiausias bendras augimas 1980-2021 metais
Korea
India
Singapore
Vietnam
Lao P.D.R.
Bhutan
Maldives
Bangladesh
China
Equatorial Guinea
Country
0
100
200
300
400
500
600
Bendras BVP augimas (%)

2. Nubrėžti grafikus, kurie iliustruotų, kaip keitėsi šalių populiacija iš The Organization for Economic Cooperation and Development (OECD).

- Išsitraukiam duomenis iš Economic Cooperation and Development (OECD) tinklalapio / csv formate, įsikeliame

```
# parsisiunčiam populiacijos duomenis iš The Organization for Economic Cooperation and Development (OECD) tinklalapio
df_population = pd.read_csv("C:\\Users\\inves\\OneDrive\\Dokumentai\\DATA learning\\PROJEKTAS\\HISTPOP_10052023174013174.csv")
df_population
```

executed in 26ms, finished 08:37:49 2023-05-18

	LOCATION	Country	SEX	Sex	AGE	Age	TIME	Time	Value	Flag Codes	Flags
0	AUS	Australia	T	Total	TOTAL	Total	1960	1960	10275000.0	NaN	NaN
1	AUS	Australia	T	Total	TOTAL	Total	1961	1961	10508200.0	NaN	NaN
2	AUS	Australia	T	Total	TOTAL	Total	1962	1962	10700500.0	NaN	NaN
3	AUS	Australia	T	Total	TOTAL	Total	1963	1963	10906900.0	NaN	NaN
4	AUS	Australia	T	Total	TOTAL	Total	1964	1964	11121600.0	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...
3167	ZAF	South Africa	T	Total	TOTAL	Total	2016	2016	56020148.0	NaN	NaN

- Pasižiūrime kokios šalys yra

```python
unique_country = df_population['Country'].unique()
unique_country
```

executed in 13ms, finished 08:37:49 2023-05-18

```
array(['Australia', 'Austria', 'Belgium', 'Canada', 'Czech Republic',
       'Denmark', 'Finland', 'France', 'Germany', 'Greece', 'Hungary',
       'Iceland', 'Ireland', 'Italy', 'Japan', 'Korea', 'Luxembourg',
       'Mexico', 'Netherlands', 'New Zealand', 'Norway', 'Poland',
       'Portugal', 'Slovak Republic', 'Spain', 'Sweden', 'Switzerland',
       'Türkiye', 'United Kingdom', 'United States', 'Argentina',
       'Brazil', 'Bulgaria', 'Chile', "China (People's Republic of)",
       'Colombia', 'Costa Rica', 'Croatia', 'Cyprus', 'Estonia', 'India',
       'Indonesia', 'Israel', 'Latvia', 'Lithuania', 'Malta', 'Romania',
       'Russia', 'Saudi Arabia', 'Singapore', 'Slovenia', 'South Africa'],
      dtype=object)
```

- Pasiliekam tik reikalingus stulpelius

```
# issimetam nereiklaingus stulpelius
df_population_data = df_population.drop(['LOCATION', 'SEX', 'Sex', 'AGE', 'Age', 'TIME', 'Flag Codes', 'Flags' ], axis=1)
df_population_data
```

executed in 13ms, finished 08:37:49 2023-05-18

	Country	Time	Value
0	Australia	1960	10275000.0
1	Australia	1961	10508200.0
2	Australia	1962	10700500.0
3	Australia	1963	10906900.0
4	Australia	1964	11121600.0
...	...	...	...
3167	South Africa	2016	56020148.0
3168	South Africa	2017	56840036.0
3169	South Africa	2018	57673251.0
3170	South Africa	2019	58532857.0
3171	South Africa	2020	59352940.0

3172 rows × 3 columns

- Persikeliam eilušiu reikšmes į stulpelius

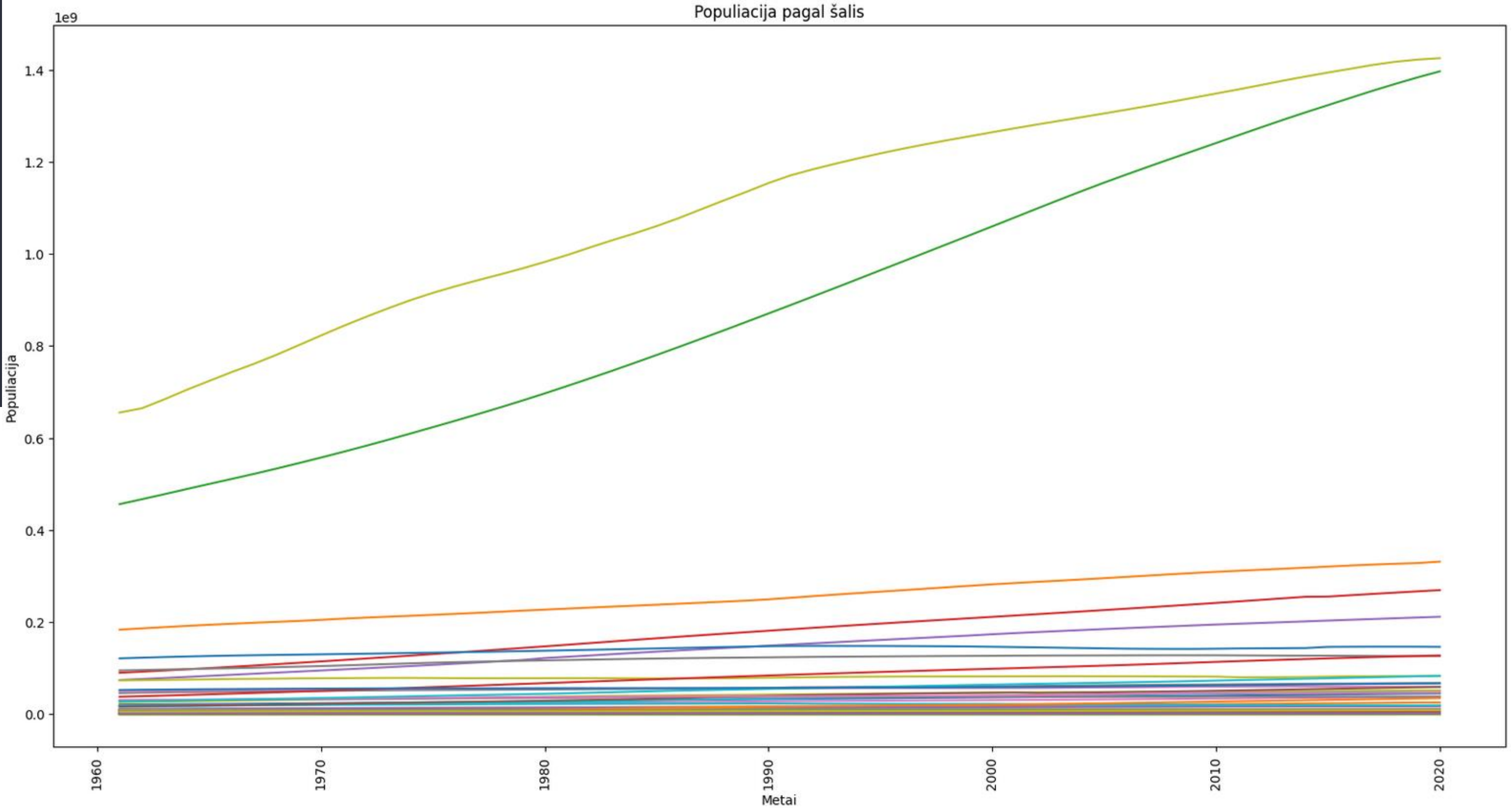
```
# Time stulpeli keičiam i eilutę, kad kiekviena šalis turėtų vieną eilutę su metų duomenimis
df_pop_pivot = df_population_data.pivot(index='Country', columns='Time', values='Value')
df_pop_pivot
```

executed in 91ms, finished 08:37:49 2023-05-18

Time	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	...	2011	2012	2013	2014	2015	2016	
Country																		
Argentina	20616010.0	20950583.0	21283784.0	21616406.0	21949246.0	22283102.0	22611643.0	22934338.0	23260684.0	23600177.0	...	4.126149e+07	4.173327e+07	4.220294e+07	4.266950e+07	4.313197e+07	4.359037e+07	4
Australia	10275000.0	10508200.0	10700500.0	10906900.0	11121600.0	11340900.0	11599498.0	11799078.0	12008635.0	12263014.0	...	2.234002e+07	2.273346e+07	2.312813e+07	2.347569e+07	2.381600e+07	2.419091e+07	2
Austria	7047539.0	7086299.0	7129864.0	7175811.0	7223801.0	7270889.0	7322066.0	7376998.0	7415403.0	7441055.0	...	8.388534e+06	8.426311e+06	8.477230e+06	8.543932e+06	8.629519e+06	8.739806e+06	8
Belgium	9153490.0	9183948.0	9220578.0	9289770.0	9378114.0	9463668.0	9527808.0	9580991.0	9618756.0	9646033.0	...	1.099361e+07	1.106775e+07	1.112503e+07	1.117978e+07	1.123847e+07	1.129500e+07	1
Brazil	72179235.0	74311338.0	76514329.0	78772647.0	81064572.0	83373533.0	85696502.0	88035815.0	90387079.0	92746607.0	...	1.966037e+08	1.983149e+08	2.000042e+08	2.017175e+08	2.034757e+08	2.051566e+08	2
Bulgaria	7867374.0	7943118.0	8012946.0	8078145.0	8144340.0	8204168.0	8258057.0	8310226.0	8369603.0	8434172.0	...	7.348328e+06	7.305888e+06	7.265114e+06	7.223938e+06	7.177991e+06	7.127822e+06	7
Canada	18256447.0	18634666.0	18989309.0	19347231.0	19719595.0	20084314.0	20420867.0	20795138.0	21129417.0	21439830.0	...	3.433933e+07	3.471422e+07	3.508295e+07	3.543744e+07	3.570291e+07	3.610949e+07	3
Chile	7643277.0	7843945.0	8044614.0	8245284.0	8445953.0	8646622.0	8831223.0	9015825.0	9200427.0	9385028.0	...	1.725416e+07	1.744349e+07	1.761190e+07	1.778762e+07	1.797142e+07	1.816715e+07	1
China (People's Republic of)	654170692.0	655260379.0	664614649.0	683903557.0	704593772.0	723846349.0	742948540.0	761006262.0	780371957.0	801430977.0	...	1.357095e+09	1.366561e+09	1.376100e+09	1.385190e+09	1.393715e+09	1.401890e+09	1
Colombia	15687691.0	16182414.0	16691286.0	17210952.0	17739753.0	18275813.0	18811407.0	19343964.0	19872505.0	20392263.0	...	4.604460e+07	4.658182e+07	4.712109e+07	4.766179e+07	4.820340e+07	4.874771e+07	4

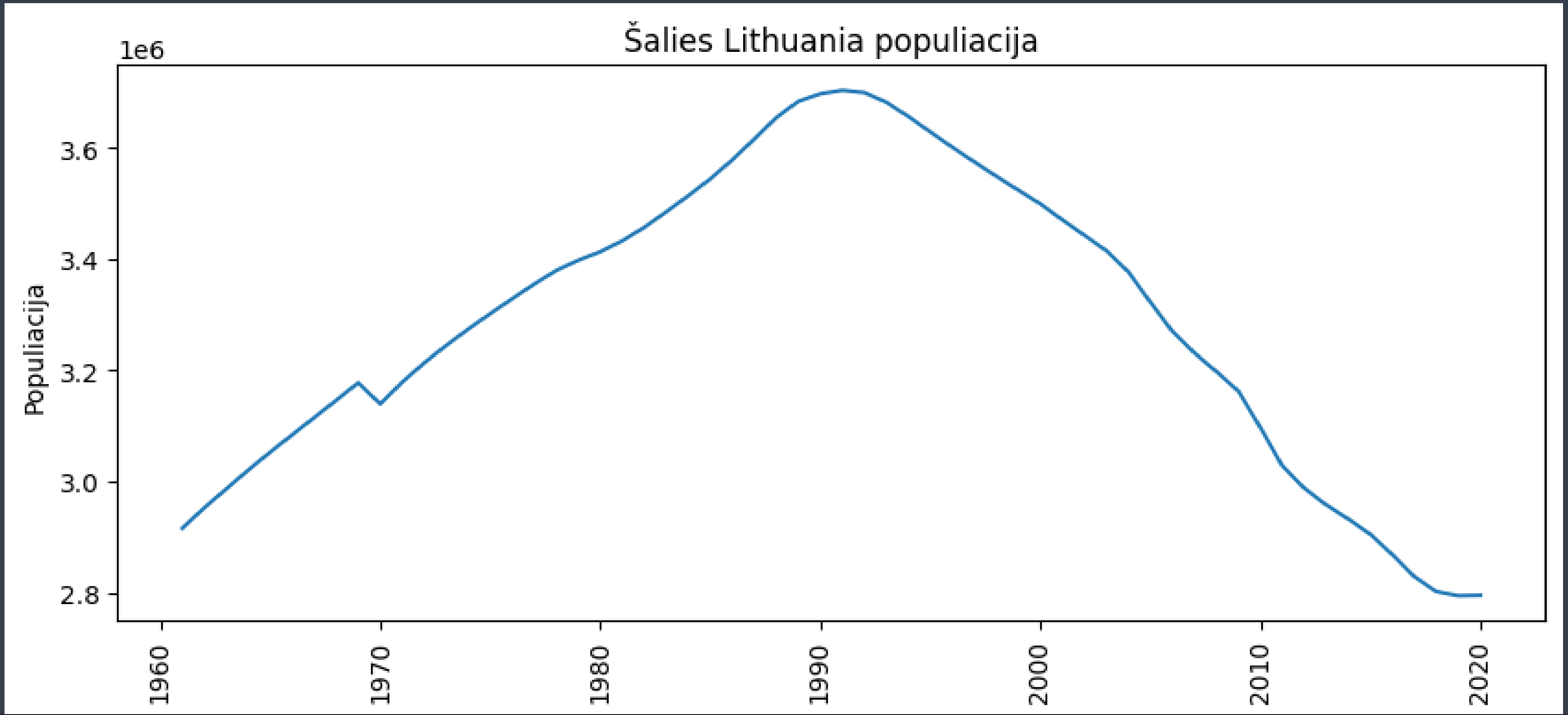
- Atsivaizduojam visas šalis viename grafike

```python
# atvaizduojam visas šalis viename grafike
countries = df_pop_pivot.index

fig, ax = plt.subplots(figsize=(20, 10))

for country in countries:
    country_data = df_pop_pivot[df_pop_pivot.index == country]
    ax.plot(country_data.columns[1:], country_data.iloc[0, 1:], label=country)

plt.xticks(rotation=90)

ax.set_xlabel('Metai')
ax.set_ylabel('Populiacija')
ax.set_title('Populiacija pagal šalis')

plt.show()
```



- Atsivaizduojam kiekvienos šalies populiacijos pokytį

```
# Atsivaizduojam grafike kiekvienos šakies populiacijos pokytį
countries = df_pop_pivot.index

for country in countries:
    fig, ax = plt.subplots(figsize=(10, 4))

    country_data = df_pop_pivot.loc[country].iloc[1:]

    ax.plot(country_data.index, country_data.values)

    plt.xticks(rotation=90)

    ax.set_xlabel('Metai')
    ax.set_ylabel('Populiacija')
    ax.set_title(f'Šalies {country} populiacija')

    # Show the plot
    plt.show()
```



Suskirstykite šalis į 5 klasterius naudodamiesi GDP ir "Volume of exports of goods"

- Užsikraunam duomenis

```
duomenys = "C:\\Users\\inves\\OneDrive\\Dokumentai\\DATA learning\\PROJEKTAS\\WEOOct2022.csv"

df = pd.read_csv(duomenys)
df
```

executed in 480ms, finished 14:43:30 2023-05-22

	WEO Country Code	ISO	WEO Subject Code	Country	Subject Descriptor	Subject Notes	Units	Scale	Country/Series-specific Notes	1980	...	2019	2020	2021	2022	2023	2024	2025	2026	2027	Estimates Start After
0	512	AFG	NGDP_R	Afghanistan	Gross domestic product, constant prices	Expressed in billions of national currency uni...	National currency	Billions	Source: National Statistics Office Latest actu...	NaN	...	1,319.90	1,288.87	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0
1	512	AFG	NGDP_RPCH	Afghanistan	Gross domestic product, constant prices	Annual percentages of constant price GDP are y...	Percent change	NaN	See notes for: Gross domestic product, consta...	NaN	...	3.912	-2.351	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0
2	512	AFG	NGDP	Afghanistan	Gross domestic product, current prices	Expressed in billions of national currency uni...	National currency	Billions	Source: National Statistics Office Latest actu...	NaN	...	1,469.60	1,547.29	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0
3	512	AFG	NGDPD	Afghanistan	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...	NaN	...	18.876	20.136	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0
4	512	AFG	PPPGDP	Afghanistan	Gross domestic product, current prices	These data form the basis for the country weig...	Purchasing power parity; international dollars	Billions	See notes for: Gross domestic product, curren...	NaN	...	81.873	80.912	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
					Gross domestic	Gross domestic			Source: Ministry of												

• Pasiliekam laikotarpį 1980 – 2021

```
df = df.drop(['2022', '2023', '2024', '2025', '2026', '2027', 'Estimates Start After'], axis=1)
df
```

executed in 83ms, finished 14:43:30 2023-05-22

	WEO Country Code	ISO	WEO Subject Code	Country	Subject Descriptor	Subject Notes	Units	Scale	Country/Series-specific Notes	1980	...	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
0	512	AFG	NGDP_R	Afghanistan	Gross domestic product, constant prices	Expressed in billions of national currency uni...	National currency	Billions	Source: National Statistics Office Latest actu...	NaN	...	1,092.12	1,154.18	1,185.31	1,197.01	1,222.92	1,255.29	1,270.22	1,319.90	1,288.87	NaN
1	512	AFG	NGDP_RPCH	Afghanistan	Gross domestic product, constant prices	Annual percentages of constant price GDP are y...	Percent change	NaN	See notes for: Gross domestic product, consta...	NaN	...	13.968	5.683	2.697	0.988	2.164	2.647	1.189	3.912	-2.351	NaN
2	512	AFG	NGDP	Afghanistan	Gross domestic product, current prices	Expressed in billions of national currency uni...	National currency	Billions	Source: National Statistics Office Latest actu...	NaN	...	1,033.59	1,116.83	1,183.04	1,226.57	1,222.92	1,285.46	1,327.69	1,469.60	1,547.29	NaN
3	512	AFG	NGDPD	Afghanistan	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...	NaN	...	20.293	20.17	20.616	20.057	18.02	18.883	18.401	18.876	20.136	NaN
4	512	AFG	PPPGDP	Afghanistan	Gross domestic product, current prices	These data form the basis for the country weig...	Purchasing power parity; international dollars	Billions	See notes for: Gross domestic product, curren...	NaN	...	59.945	63.784	69.444	72.056	70.098	74.712	77.406	81.873	80.912	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
					Gross domestic	Gross domestic			Source: Ministry of												

- Pasianalizuojam kokių duomenų turim

```
# pasiziurim turimus duomenis Subject Descriptor stulpelyje
reiksmes = df['Subject Descriptor'].tolist()
reiksmes
```

```
executed in 55ms, finished 14:43:30 2023-05-22
```

```
['Gross domestic product, constant prices',
 'Gross domestic product, constant prices',
 'Gross domestic product, current prices',
 'Gross domestic product, current prices',
 'Gross domestic product, current prices',
 'Gross domestic product, deflator',
 'Gross domestic product per capita, constant prices',
 'Gross domestic product per capita, constant prices',
 'Gross domestic product per capita, current prices',
 'Gross domestic product per capita, current prices',
 'Gross domestic product per capita, current prices',
 'Output gap in percent of potential GDP',
 'Gross domestic product based on purchasing-power-parity (PPP) share of world total',
 'Implied PPP conversion rate',
 'Total investment',
 'Gross national savings',
 'Inflation, average consumer prices',
 'Inflation, average consumer prices',
 'Inflation, end of period consumer prices',
 'Inflation, end of period consumer prices',
 'Volume of imports of goods and services',
 'Volume of Imports of goods',
```

- Išsifiltruojam reikalingas BVP ir export reikšmes

```
ieskomos_reiksmes = ['Gross domestic product, current prices', 'Volume of exports of goods', 'U.S. dollars', 'Percent change' ]

df = df[df['Subject Descriptor'].isin(ieskomos_reiksmes)]
df
```

executed in 88ms, finished 14:43:30 2023-05-22

	WEO Country Code	ISO	WEO Subject Code	Country	Subject Descriptor	Subject Notes	Units	Scale	Country/Series-specific Notes	1980	...	2012	2013	2014	2015	2016	2017	2018	2019	20
2	512	AFG	NGDP	Afghanistan	Gross domestic product, current prices	Expressed in billions of national currency uni...	National currency	Billions	Source: National Statistics Office Latest actu...	NaN	...	1,033.59	1,116.83	1,183.04	1,226.57	1,222.92	1,285.46	1,327.69	1,469.60	1,54
3	512	AFG	NGDPD	Afghanistan	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...	NaN	...	20.293	20.17	20.616	20.057	18.02	18.883	18.401	18.876	20.1
4	512	AFG	PPPGDP	Afghanistan	Gross domestic product, current prices	These data form the basis for the country weig...	Purchasing power parity; international dollars	Billions	See notes for: Gross domestic product, curren...	NaN	...	59.945	63.784	69.444	72.056	70.098	74.712	77.406	81.873	80.9
23	512	AFG	TXG_RPCH	Afghanistan	Volume of exports of goods	Percent change of volume of exports of goods r...	Percent change	NaN	Source: Various sources: Central Statistical O...	NaN	...	4.546	9.76	34.978	12.886	13.944	18.11	2.545	1.531	-16.6
46	914	ALB	NGDP	Albania	Gross domestic product, current prices	Expressed in billions of national currency uni...	National currency	Billions	Source: IMF Staff Estimates. Official national...	18.489	...	1,332.81	1,350.05	1,395.31	1,434.31	1,472.48	1,550.65	1,636.73	1,691.90	1,64
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8559	754	ZMB	TXG_RPCH	Zambia	Volume of exports of goods	Percent change of volume of exports of goods r...	Percent change	NaN	Source: Central Bank. Values from BOP, volumes...	NaN	...	28.072	23.131	-4	-11.392	-5.147	3.981	5.35	-11.722	9.23
8582	698	ZWE	NGDP	Zimbabwe	Gross domestic product, current prices	Expressed in billions of national currency uni...	National currency	Billions	Source: National Statistics Office Latest actu...	NaN	...	17.116	19.093	19.499	19.969	20.555	21.385	36.945	187.419	1,18
					Gross domestic	Values are based														

- Susiskirstom į 5 klasterius

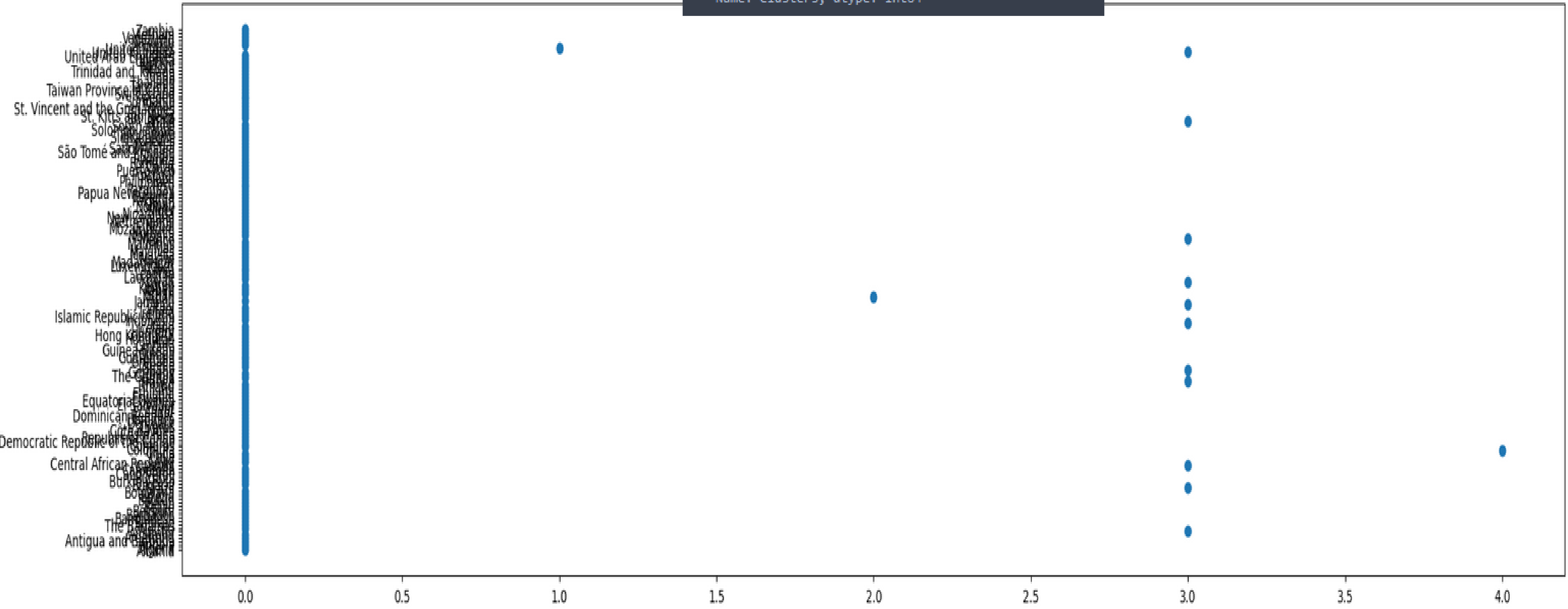
```
# kmeans clusterizavimas

metai = ['1980', '1981', '1982', '1983', '1984', '1985', '1986', '1987', '1988', '1989', '1990', '1991', '1992', '1993', '1994', '1995', '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003', '
# tukstantines reiksmes yra atskirtos kableliu, keiciam ir nepaliekma tarpo
df[metai] = df[metai].replace(',', '', regex=True)
# metu duomenis verciam i float duomenu tipa
df[metai] = df[metai].astype(float)
data = df[metai].values
#  atliekam clusterizavima
kmeans = KMeans(n_clusters=5, random_state=42)
clusters = kmeans.fit_predict(data)
# sukuriam stulpeli
df['Clusters'] = clusters
# atsivaizduojam kuri salis i koki clusteri patenka
df[['Country', 'Clusters']]
```

- Gauname toki pasiskirstymą

```
df['Clusters'].value_counts()
```

executed in 24ms, finished 15:05:47 2023-05-22

```
0    129
3     11
4      1
2      1
1      1
Name: Clusters, dtype: int64
```



4. Sukurkite modelį, kuris prognozuoja “GDP per capita”.

- Naudojamos bibliotekos

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
```

- Nusiskaitom duomenis

```
duomenys = "C:\\Users\\inves\\OneDrive\\Dokumentai\\DATA learning\\PROJEKTAS\\WEOOct2022.csv"

df = pd.read_csv(duomenys)
df
```

executed in 866ms, finished 15:09:04 2023-05-22

	WEO Country Code	ISO	WEO Subject Code	Country	Subject Descriptor	Subject Notes	Units	Scale	Country/Series-specific Notes	1980	...	2019	2020	2021	2022	2023	2024	2025	2026	2027	Estimates Start After
0	512	AFG	NGDP_R	Afghanistan	Gross domestic product, constant prices	Expressed in billions of national currency uni...	National currency	Billions	Source: National Statistics Office Latest actu...	NaN	...	1,319.90	1,288.87	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0
1	512	AFG	NGDP_RPCH	Afghanistan	Gross domestic product, constant prices	Annual percentages of constant price GDP are y...	Percent change	NaN	See notes for: Gross domestic product, consta...	NaN	...	3.912	-2.351	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0
2	512	AFG	NGDP	Afghanistan	Gross domestic product, current prices	Expressed in billions of national currency uni...	National currency	Billions	Source: National Statistics Office Latest actu...	NaN	...	1,469.60	1,547.29	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0
3	512	AFG	NGDPD	Afghanistan	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...	NaN	...	18.876	20.136	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0
4	512	AFG	PPPGDP	Afghanistan	Gross domestic product, current prices	These data form the basis for the country weig...	Purchasing power parity; international dollars	Billions	See notes for: Gross domestic product, curren...	NaN	...	81.873	80.912	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2020.0

- Pasiliekame reikalingus stulpelius

```
df = df.drop(['WEO Country Code', 'ISO', 'WEO Subject Code', 'Subject Notes', 'Scale', 'Country/Series-specific Notes', '2022', '2023', '2024', '2025', '2026', '2027', 'Estimates Start After' ], axis=1
df
```

executed in 103ms, finished 15:09:04 2023-05-22

	Country	Subject Descriptor	Units	1980	1981	1982	1983	1984	1985	1986	...	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
0	Afghanistan	Gross domestic product, constant prices	National currency	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	1,092.12	1,154.18	1,185.31	1,197.01	1,222.92	1,255.29	1,270.22	1,319.90	1,288.87	NaN
1	Afghanistan	Gross domestic product, constant prices	Percent change	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	13.968	5.683	2.697	0.988	2.164	2.647	1.189	3.912	-2.351	NaN
2	Afghanistan	Gross domestic product, current prices	National currency	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	1,033.59	1,116.83	1,183.04	1,226.57	1,222.92	1,285.46	1,327.69	1,469.60	1,547.29	NaN
3	Afghanistan	Gross domestic product, current prices	U.S. dollars	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	20.293	20.17	20.616	20.057	18.02	18.883	18.401	18.876	20.136	NaN
4	Afghanistan	Gross domestic product, current prices	Purchasing power parity; international dollars	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	59.945	63.784	69.444	72.056	70.098	74.712	77.406	81.873	80.912	NaN

- Susitvarkom duomenų tipą, išsitrinam kablelius, kurie keičia skaičių vertes

```
metai = df.columns[3:]
df[metai] = df[metai].replace(',', '', regex=True)
df[metai] = df[metai].apply(pd.to_numeric, errors='coerce')
df = df.interpolate(method='linear')
df = df.fillna(0)
df.head(60)
```

Random forest

```python
# pasirenkam duomenis
X = df.loc[:, '1980':'2020']

# pasirenkam spejimo duoemnu stulpeli
y = df['2021']

# Isdalinam i train ir test duoemnis 80/20
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Random Forest modelis
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

executed in 2m 34s, finished 15:44:42 2023-05-22

```
Mean Squared Error: 235667805820.47784
R-squared: 0.9803977488076824
```

```python
rf = RandomForestRegressor(n_estimators=10, random_state=68)
rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

executed in 16.2s, finished 16:36:57 2023-05-22

```
Mean Squared Error: 8.338941393324642e+24
R-squared: 0.033591019798275945
```

```python
rf = RandomForestRegressor(n_estimators=1000, random_state=42)
rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

executed in 23m 31s, finished 16:11:02 2023-05-22

```
Mean Squared Error: 3.470229402235161e+16
R-squared: -2885.4489233373633
```

Gradient boosting regressor

```python
# Gradient boosting regressor
X = df.loc[:, '1980':'2020']
y = df['2021']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

gb_model = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1, max_depth=3, subsample=0.8, min_samples_split=2, min_samples_leaf=1, max_features=None)
gb_model.fit(X_train, y_train)

y_pred = gb_model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

executed in 13.5s, finished 16:46:45 2023-05-22

```
Mean Squared Error: 2038488123370.2378
R-squared: 0.8304437209497398
```

```python
gb_model = GradientBoostingRegressor(n_estimators=50, learning_rate=0.05, max_depth=3, subsample=0.8, min_samples_split=2)
gb_model.fit(X_train, y_train)

y_pred = gb_model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

executed in 6.46s, finished 16:53:05 2023-05-22

```
Mean Squared Error: 5.335539770055529e+18
R-squared: -443795.68429934053
```

```python
gb_model = GradientBoostingRegressor(n_estimators=500, learning_rate=0.2)
gb_model.fit(X_train, y_train)

y_pred = gb_model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

executed in 2m 6s, finished 16:59:18 2023-05-22

```
Mean Squared Error: 61981614198.99792
R-squared: 0.9948445263170159
```

Linear regression

```python
X = df.loc[:, '1980':'2020']
y = df['2021']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

y_pred = lr_model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

executed in 50ms, finished 17:05:26 2023-05-22

```
Mean Squared Error: 3.507985133805792e+19
R-squared: -2917852.184586395
```

Tolimesni veiksmai

- Vystant toliau šią problematiką imčiausi tokių veiksmų:

  1. BVP duomenų paėmimą programuočiau tiesiai iš tinklalapio;

  2. Populiacijos duomenų paėmimą irgi programuočiau paėmima tiesiai

iš tinklalapio, gal net su galimybe atsirinkti reikalingus parametrus, pagal tai , kaip yra suprogramuota pačiame puslapyje;

  3. Ieškočiau papildomų šaltinių BVP duomenų, kad trūkstamos reikšmės būtų užpildytos kuo tiksliau;

  4. Sukurčiau TimeLine modelį;

  5. Giliau išanalizuočiau turimus duomenis ir jų įtaką kitiems esantiems toje pačioje lentelėje duomenims, jų koreliacijas ir priklausomybes.

Išvados

- BVP nuspėti sudėtinga, dėl labai didelių pašalinių įtakų:

  exporto/importo pokytis (muitai, sankcijos ir pan.)

  pandemijos

  infliacija

  darbo užimtumo

  ir kitų veiksnių.

- Viską Jupyter notebook formate galite rasti

https://github.com/TautvydasLuksas/data_course_finish_project


QUESTIONS