

## Inlämningsuppgift i kursen Grundläggande Programmering i Python, LP3-2022.

### Syfte och mål

Syftet med inlämningsuppgiften är att du ska visa, inte minst för dig själv, att du kan skapa ett Pythonprogram utifrån en definierad problembeskrivning där flera av de färdigheter som du fått i kursen används. Målet med uppgiften är att du, efter slutförd uppgift, har fått ytterligare ökad förståelse för programmeringens grunder och att du fått insikt om hur man bryter ned en programmeringsuppgift i mindre delar.

### Förutsättningar

Uppgiften utförs i form av 6 st. fristående uppgifter där den sista uppgiften blir att sätta ihop programmen från de olika uppgifterna och skapa ett sammanhållet program via ett enkelt menysystem.

Varje deluppgift kommer att bedömas var för sig. För att inlämningsuppgiften ska bli godkänd måste varje deluppgift vara korrekt utförd. Beräkningar som ska utföras måste du skapa själv. Du får alltså inte använda dig av färdiga funktioner i moduler som går att importera. Du ska däremot använda modulen *CSV* (se kapitel 10 i Canvas) för att hantera CSV-filer och modulen *matplotlib.pyplot* (se kapitel 11 i Canvas) för plottnig. Vi ställer inga krav på avancerad felkontroll i programmen.

Några regler som skall följas:

- koden ska vara genomtänkt, lättläst och välkommenterad
  - o skriv varför en kodrad finns och inte vad den gör. Se kursmaterialet i Canvas för exempel (modul 2 Introduktion och ett första programexempel).
- variabler ska ha meningsfulla namn.
- du får inte lämna in kod som genererar felmeddelande eller varningar. Då blir det omedelbart retur.
- du får inte lämna in bortkommenterad kod.

Programkoden skriver du i Jupyter Notebookfilen som finns att ladda ned. För att få en hanterbar och smidig rättning av uppgifterna måste de anvisningar som finns i Jupyter Notebook filen följas. Avsteg från dessa innebär omedelbar retur.

- Jupyter Notebookfilen med dina lösningar är det enda material som du ska ladda upp.
  - o Filen skall behålla sin namnstruktur. Ladda alltså upp den med samma namn som den hade när du hämtade den.

## Uppgiften.



Eurostat är EU:s statistikkontor som publicerar europeiska statistikuppgifter och indikatorer, så att man kan jämföra olika länder och regioner inom en mängd områden. På deras hemsida (<https://ec.europa.eu/eurostat>) finns all denna statistik fritt tillgänglig för nedladdning utan kostnad. I denna inlämningsuppgift ska du använda tillgänglig statistik från Eurostat för att analysera EU-ländernas förväntade befolkningsutveckling fram till år 2100. Nedanstående figur visar en delmängd av denna data. Filen som du ska använda och som innehåller den fullständiga datan heter 'befolkningsdata.csv' och är lagrad som en csv-fil med semikolon (;) som avgränsningstecken.

COUNTRIES	2019	2020	2025	2030	2035	2040	2045	2050
Belgium	11 455 519	11 507 338	11 661 206	11 757 990	11 835 820	11 894 881	11 927 324	11 926 987
Bulgaria	7 000 039	6 949 549	6 690 388	6 450 296	6 224 049	6 016 719	5 828 425	5 655 026
Czechia	10 649 800	10 693 861	10 789 806	10 762 174	10 694 436	10 625 402	10 571 354	10 530 251
Denmark	5 806 081	5 811 651	5 884 275	5 963 578	6 018 816	6 055 503	6 080 087	6 098 190
Germany	83 019 213	83 135 181	83 482 307	83 453 697	83 318 670	83 178 426	82 983 422	82 669 724
Estonia	1 324 820	1 329 916	1 322 440	1 308 435	1 294 043	1 281 555	1 269 633	1 256 223
Ireland	4 904 240	4 966 879	5 272 930	5 504 390	5 714 371	5 904 540	6 073 416	6 213 191
Greece	10 724 599	10 696 535	10 510 196	10 303 200	10 104 622	9 910 798	9 713 851	9 503 127
Spain	46 937 060	47 321 434	48 310 619	48 746 399	49 110 869	49 377 094	49 479 880	49 348 530
France	67 012 883	67 197 367	68 036 808	68 749 400	69 354 321	69 802 409	70 015 780	70 010 903
Croatia	4 076 246	4 056 285	3 936 509	3 828 089	3 721 032	3 612 487	3 502 067	3 392 559
Italy	60 359 546	60 286 529	60 088 529	59 942 512	59 709 982	59 375 006	58 870 898	58 125 032
Cyprus	875 899	887 331	928 295	962 854	990 814	1 012 858	1 030 807	1 046 219
Latvia	1 919 968	1 907 094	1 815 550	1 712 746	1 618 827	1 536 108	1 462 550	1 395 039
Lithuania	2 794 184	2 793 592	2 707 915	2 575 553	2 452 313	2 339 698	2 236 142	2 137 939
Luxembourg	613 894	626 031	662 364	692 722	718 104	739 137	756 215	769 048
Hungary	9 772 756	9 771 975	9 697 220	9 619 020	9 532 830	9 441 139	9 350 754	9 270 352
Malta	493 559	506 951	557 426	588 691	614 214	634 910	652 604	668 373
Netherlands	17 282 163	17 404 793	17 751 055	17 969 884	18 119 551	18 185 792	18 185 796	18 142 292
Austria	8 858 775	8 904 262	9 029 008	9 149 001	9 232 708	9 292 363	9 332 840	9 345 829
Poland	37 972 812	37 941 122	37 567 248	37 018 453	36 369 328	35 661 656	34 897 373	34 102 204
Portugal	10 276 617	10 291 457	10 220 601	10 089 138	9 948 994	9 786 632	9 593 258	9 375 347
Romania	19 414 458	19 281 118	18 507 547	17 808 000	17 169 288	16 576 187	16 018 918	15 502 837
Slovenia	2 080 908	2 095 314	2 114 603	2 106 316	2 094 654	2 081 622	2 065 380	2 043 751
Slovakia	5 450 421	5 457 679	5 467 891	5 440 730	5 384 612	5 312 439	5 232 249	5 147 215
Finland	5 517 919	5 527 189	5 537 441	5 519 298	5 480 971	5 426 143	5 361 229	5 290 709
Sweden	10 230 185	10 322 613	10 746 886	11 099 033	11 405 423	11 693 373	11 979 369	12 254 064

Figur 1. Delmängd av innehållet i filen befolkningsdata.csv.

Ett tips innan du börjar lösa nedanstående uppgifter är att du först (noga) läser igenom alla uppgifter som du ska utföra för att du ska få en klar uppfattning om vad inlämningsuppgiften går ut på. Öppna sedan *befolkningsdata.csv* i Excel eller i ett motsvarande program och ta reda på hur data i filen är strukturerad. Därefter är du redo att börja lösa uppgifterna.

## Uppgifter.

Nedanstående uppgifter utför du i Jupyter Notebookfilen som finns att ladda ner i inlämningsuppgiften i Canvas.

### Uppgift 1:

Skapa en egendefinerad funktion `read_file(file_name)` som öppnar en csv-fil med namnet 'file\_name' och läser in dess innehåll i en tvådimensionell lista. När du kallar på denna funktion med filnamnet `befolkningsdata.csv` ska innehållet i denna fil hamna i en tvådimensionell lista med namnet `befolkning`. Avsluta uppgiften med att skriva ut de två första raderna i listan.

### Uppgift 2:

Skapa en egendefinerad funktion `analysera_data(lista)` som tar listan `befolkning` som argument och returnerar en tvådimensionell lista där varje rad innehåller en utförd analys per land och innehåller följande information:

kolumn 0: land

kolumn 1: lägsta förväntade befolkningstalet under tidsperioden 2019-2100.

kolumn 2: året för lägsta förväntade befolkningstalet.

kolumn 3: högsta förväntade befolkningstalet under tidsperioden 2019-2100.

kolumn 4: året för högsta förväntade befolkningstalet.

kolumn 5: förväntad befolkningsutveckling uttryckt i % under tidsperioden 2019-2100 och ska beräknas enligt formeln:  $(\text{befolkning}_{2100} - \text{befolkning}_{2019}) / \text{befolkning}_{2019} * 100$ .

Förtydligande: kolumn 0 ska innehålla alla länder som finns i listan `befolkning` från och med 'Belgium' till och med 'Sweden'. Den returnerade listan får därmed följande principiella utseende (värdena i listan är bara exempelvärden):

Belgium	3623500	2035	3823500	2065	5.5
...	...	...	...	...	...
...	...	...	...	...	...
Bulgaria	6456984	2075	7145694	2025	-4.4

Avsluta uppgiften med att anropa funktionen och skriv ut de två första raderna i listan som returneras.

### Uppgift 3.

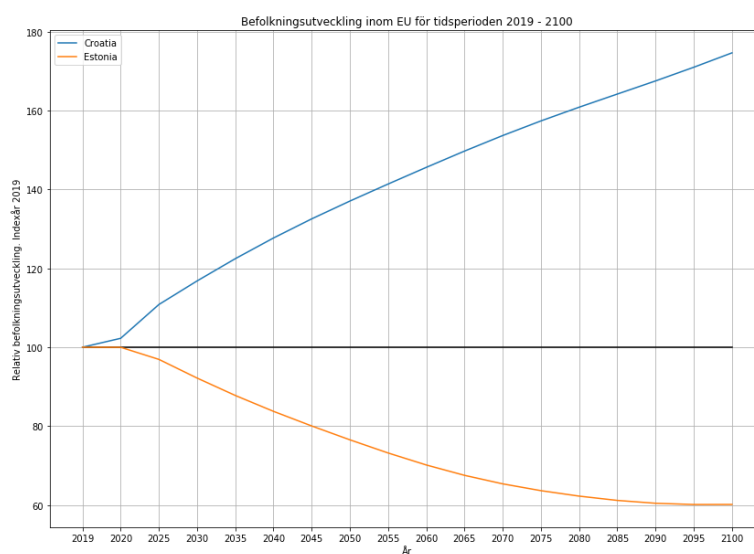
Använd informationen som finns i listan som returnerades från den egendefinerade funktionen i uppgift 2 och skriv ett program som skapar en tabell innehållande de fem länder i listan *befolkning* där befolkningen ökar mest och de fem länder där befolkningen minskar mest. I tabellen ska också information om dessa länders lägsta och högsta befolkningstal under perioden 2019-2100 finnas och vilka år dessa förväntas ske. Nedan visas tabellens principiella utseende (innehållet i tabellen är bara exempel).

```
=====
```

Förväntad befolkningsutveckling för tio länder inom EU under åren 2019 -- 2100					
(Tabellen visar de fem länder med störst respektive minst förväntad befolkningsökning)					
-----					
Estimerad befolkning					
-----					
Land	Lägst befolkningsantal Befolkning	År	Högst befolkningsantal Befolkning	År	Förändring [%] 2019-2100
-----					
Sweden	10768900	2019	13475900	2100	25.14
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
Poland	35975687	2100	37843035	2019	-5.19
=====					

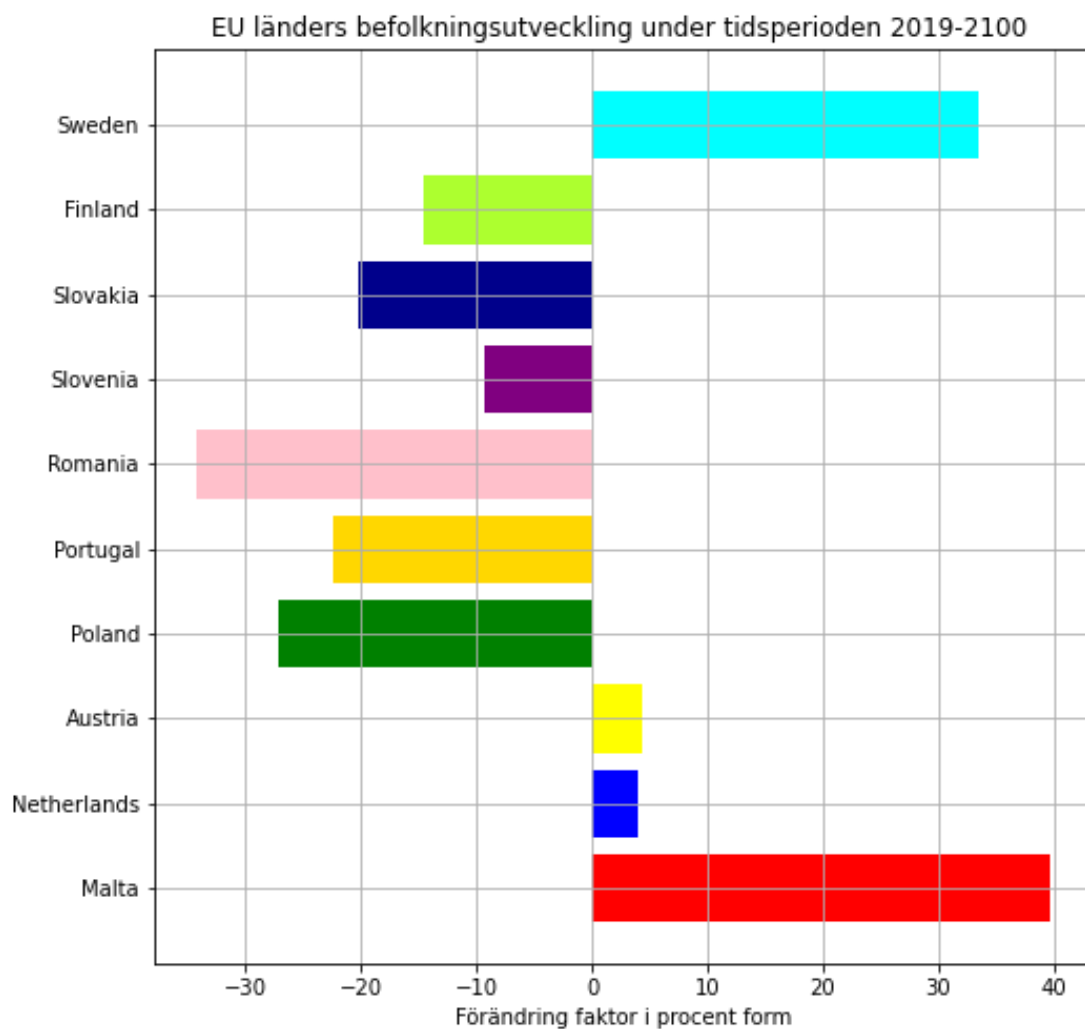
### Uppgift 4.

Skriv ett program som analyserar länderna i uppgift 3 ytterligare ett steg genom att normera respektive lands befolkningsutveckling med år 2019 som bas. År 2019 ska alltså ha värdet 100 och övriga år ska relateras till år 2019. Skapa därefter ett diagram där x-axeln innehåller årtalen som finns i listan *befolkning* och y-axeln visar den relativa befolkningsutvecklingen. Diagrammet ska innehålla ett rutnät och ett rakt streck som indikerar normvärdet 100. Varje land ska ha olika färg och ha en etikett som visas i diagrammet. Nedan visas diagrammets principiella utseende med två länder (obs! värdena och länderna i grafen är bara exempel).



### Uppgift 5:

Skapa ett horisontellt stapeldiagram innehållande samtliga länders befolkningsutveckling under tidsperioden 2019-2100 (dvs innehållet i kolumn 6 i uppgift 2). Diagrammets principiella utseende visas nedan.



---

Här finns den officiella dokumentation om horisontell stapeldiagram (horizontal bar plot) `barh()`:

[https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.barh.html#matplotlib.pyplot.barh](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.barh.html#matplotlib.pyplot.barh)

### Uppgift 6:

Skapa ett sammanhållet program med hjälp av nedanstående menyalternativ.

Användaren anger ett menyalternativ. När detta är utfört ska användaren kunna ange ett nytt menyalternativ osv. Först när användaren anger menyalternativ 3 avslutas programmet.

1. Hämta data från fil
2. Analysera data
3. Avsluta

Välj menyalternativ (1-3):

Beskrivning av menyalternativen:

1. Frågar efter en datafil (*befolkningsdata.csv*) och sparar dess innehåll i en lista med namnet *befolkning*.
2. Innehållet i listan *befolkning* analyseras och resultaten presenteras på skärmen i form av tabellen (uppgift 3) det normerade diagrammet (uppgift 4) och stapeldiagrammet (uppgift 5).
3. Avslutar programmet.

### Innan du laddar upp din färdiga källkod:

När du är färdig med alla uppgifter bör du lägga lite extra tid för att en sista gång kontrollera att din källkod uppfyller kraven i respektive uppgift. Innan du laddar upp din färdiga Jupyter Notebook-fil bör du rensa alla variabler i minnet och köra programmet en gång till för att förvissa dig om att programmen inte 'kraschar'. Du rensar variablerna antingen genom att välja alternativet 'Restart & Clear Output' under menyalternativet 'Kernel' i Jupyter Notebook eller starta om datorn och starta Jupyter Notebook och ladda in källkoden igen. När du kört programmet efter att du rensat variablerna och allt fungerar korrekt är det klart för uppladdning till Canvas.

- Jupyter Notebookfilen med dina lösningar är det enda material som du ska ladda upp.
  - Filen skall behålla sin namnstruktur. Ladda alltså upp den med samma namn som den hade när du hämtade den.

Lycka till!