

Inlämningsuppgift i kursen Grundläggande Programmering i Python

Syfte och mål

Syftet med inlämningsuppgiften är att du ska visa, inte minst för dig själv, att du kan skapa ett program i Python utifrån en definierad problembeskrivning där flera av de färdigheter som du fått i kursen kommer att användas. 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

All administrativ information om datum för inlämning, hur kamratgranskningen går till, betyg & rättning, osv finns på Canvas.

Uppgiften utförs i form av 5 st. deluppgifter. Varje deluppgift bedöms var för sig med "OK" eller "ej OK". För "OK" krävs att den är *acceptabelt* korrekt utförd, och för att inlämningsuppgiften i helhet ska bli godkänd vid lärarinlämningen måste alla deluppgifter vara bedömda "OK".

En grundregel i lösandet av denna inlämningsuppgift är att du inte får använda färdiga matematiska funktioner eller importerade moduler för att förenkla dina beräkningar. Det finns dock två moduler som du behöver använda:

- modulen CSV (se kapitel 10 i Canvas) för att läsa/skriva CSV-filer.
- modulen *matplotlib.pyplot* (se kapitel 11 i Canvas) för plottning.

Vi ställer inga krav på att "inmatningsfelkontroller" måste finnas.

Några allmänna riktlinjer som skall följas:

- koden ska vara genomtänkt, lättläst och välkommenterad
 - o skriv varför kodrader finns, och inte vad de tekniskt gör. Se i kursmaterialet för diskussion och exempel på detta (kapitel 2 i Canvas *Introduktion och ett första programexempel*)
- variabler ska ha meningsfulla namn.
- du får inte lämna in kod som genererar felmeddelande eller varningar.
- du får inte lämna kvar bortkommenterad kod i uppgiften.
- du får (givetvis) INTE använda dig av AI-generad information från ChatGPT eller liknande program för att lösa uppgifterna.

Obs:

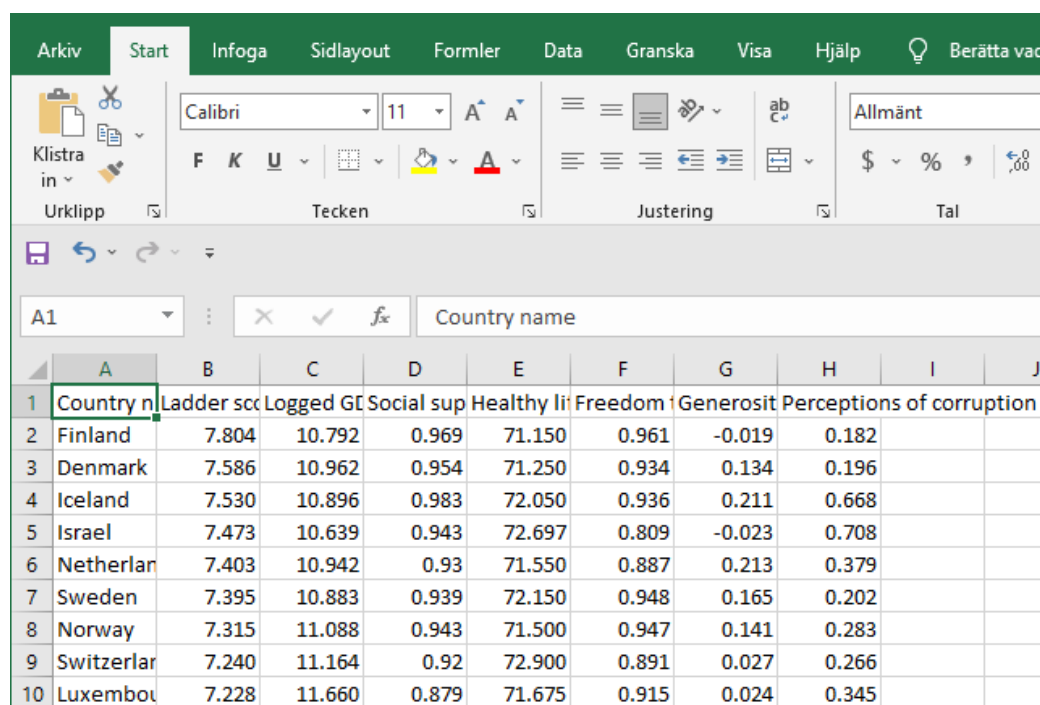
Programkoden skriver du i Jupyter Notebook och programkoden skriver du i den tillhandahållna svarsfilen "*Svarsfil_inluppg_0823_A237TG.ipynb*". Använd en kodcell per deluppgift. Det är endast svarsfilen som du ska ladda upp till Canvas. Ändra inte namn på svarsfilen.

Uppgiftsbeskrivning

Uppgiften går ut på att analysera, extrahera och presentera information som finns publicerad i FN:s "The World Happiness rapport. Rapporten (med tillhörande datafiler) presenterar rankningar av nationell lycka baserat på respondenternas uppfattningar avseende deras livssituation. Rapporten tar upp olika livsfaktorer såsom inkomst, hälsa, socialt stöd, att ha någon att lita på i tider av problem, ha känsla för frihet och att kunna fatta viktiga livsbeslut. Rapporten tar även upp mänskliga rättigheter för alla människor, oavsett ras, kön, nationalitet, etnicitet, språk, religion eller någon annan status. Generositet och frånvaro av korruption spelar en avgörande roll i utvärderingen av livssituationen. Dessutom belyser rapporten lyckotillståndet i världen idag och rankar länder i form av en livsutvärderingspoäng. En mera detaljerad information om undersökningen hittar du på rapportens hemsida <https://worldhappiness.report/>.

Den information du behöver för att lösa inlämningsuppgiften finns i de tre CSV-filerna **WHR2023.csv**, **WHR2020-2021.csv** och **WHR2005-2022.csv**. Nedan visas delar av innehållet i dessa CSV-filer.

WHR2023.csv:



| | A | B | C | D | E | F | G | H | I | J |
|----|--------------|--------------|------------|----------------|-------------------------|---------|------------|---------------------------|---|---|
| 1 | Country name | Ladder score | Logged GDP | Social support | Healthy life expectancy | Freedom | Generosity | Perceptions of corruption | | |
| 2 | Finland | 7.804 | 10.792 | 0.969 | 71.150 | 0.961 | -0.019 | 0.182 | | |
| 3 | Denmark | 7.586 | 10.962 | 0.954 | 71.250 | 0.934 | 0.134 | 0.196 | | |
| 4 | Iceland | 7.530 | 10.896 | 0.983 | 72.050 | 0.936 | 0.211 | 0.668 | | |
| 5 | Israel | 7.473 | 10.639 | 0.943 | 72.697 | 0.809 | -0.023 | 0.708 | | |
| 6 | Netherlands | 7.403 | 10.942 | 0.93 | 71.550 | 0.887 | 0.213 | 0.379 | | |
| 7 | Sweden | 7.395 | 10.883 | 0.939 | 72.150 | 0.948 | 0.165 | 0.202 | | |
| 8 | Norway | 7.315 | 11.088 | 0.943 | 71.500 | 0.947 | 0.141 | 0.283 | | |
| 9 | Switzerland | 7.240 | 11.164 | 0.92 | 72.900 | 0.891 | 0.027 | 0.266 | | |
| 10 | Luxembourg | 7.228 | 11.660 | 0.879 | 71.675 | 0.915 | 0.024 | 0.345 | | |

WHR2020-2021.csv:

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|-------------|--------|-----------|----------|------------|--------|----------|-----------|-----------|------|-----------|-----------------|-----------------|---|
| 1 | country | region | subregion | pop2023 | growthRate | area | landArea | density | densityMi | Rank | happiness | happiness score | Difference rank | |
| 2 | Finland | | Northern | 5545475 | 0.00085 | 338424 | 303940 | 182.453 | 472.553 | 118 | 7.842 | 7.809 | 0.033 | 1 |
| 3 | Denmark | Europe | Northern | 5910913 | 0.00487 | 43094 | 40000 | 1.477.728 | 3.827.316 | 115 | 7.62 | 7.646 | -0.026 | 2 |
| 4 | Switzerland | Europe | Western E | 8796669 | 0.00643 | 41284 | 39516 | 2.226.103 | 5.765.607 | 101 | 7.571 | 7.56 | 0.011 | 3 |
| 5 | Iceland | Europe | Northern | 375318 | 0.00649 | 103000 | 100830 | 37.223 | 96.407 | 179 | 7.554 | 7.504 | 0.05 | 4 |
| 6 | Netherlands | Europe | Western E | 17618299 | 0.00309 | 41850 | 33670 | 523.264 | ##### | 72 | 7.464 | 7.449 | 0.015 | 5 |
| 7 | Norway | Europe | Northern | 5474360 | 0.00737 | 323802 | 364285 | 150.277 | 389.217 | 119 | 7.392 | 7.488 | -0.096 | 6 |
| 8 | Sweden | Europe | Northern | 10612086 | 0.00595 | 450295 | 407283.5 | 260.558 | 674.844 | 87 | 7.363 | 7.353 | 0.01 | 7 |
| 9 | Luxembourg | Europe | Western E | 654768 | 0.01107 | 2586 | 2574.5 | 2.543.282 | 6.587.101 | 168 | 7.324 | 7.238 | 0.086 | 8 |

WHR2005-2022.csv:

| | A | B | C | D | E | F | G | H | I | J | K | L |
|----|-------------|------|------------|-----------|------------|------------|---------|-----------|-----------|------------|-----------------|---|
| 1 | country | year | Life Ladde | Log GDP p | Social sup | Healthy li | Freedom | Generosit | Perceptio | Positive a | Negative affect | |
| 2 | Afghanistan | 2008 | 3.724 | 7.350 | 0.451 | 50.500 | 0.718 | 0.168 | 0.882 | 0.414 | 0.258 | |
| 3 | Afghanistan | 2009 | 4.402 | 7.509 | 0.552 | 50.800 | 0.679 | 0.191 | 0.85 | 0.481 | 0.237 | |
| 4 | Afghanistan | 2010 | 4.758 | 7.614 | 0.539 | 51.100 | 0.6 | 0.121 | 0.707 | 0.517 | 0.275 | |
| 5 | Afghanistan | 2011 | 3.832 | 7.581 | 0.521 | 51.400 | 0.496 | 0.164 | 0.731 | 0.48 | 0.267 | |
| 6 | Afghanistan | 2012 | 3.783 | 7.661 | 0.521 | 51.700 | 0.531 | 0.238 | 0.776 | 0.614 | 0.268 | |
| 7 | Afghanistan | 2013 | 3.572 | 7.680 | 0.484 | 52.000 | 0.578 | 0.063 | 0.823 | 0.547 | 0.273 | |
| 8 | Afghanistan | 2014 | 3.131 | 7.671 | 0.526 | 52.300 | 0.509 | 0.106 | 0.871 | 0.492 | 0.375 | |
| 9 | Afghanistan | 2015 | 3.983 | 7.654 | 0.529 | 52.600 | 0.389 | 0.082 | 0.881 | 0.491 | 0.339 | |
| 10 | Afghanistan | 2016 | 4.220 | 7.650 | 0.559 | 52.925 | 0.523 | 0.044 | 0.793 | 0.501 | 0.348 | |
| 11 | Afghanistan | 2017 | 2.662 | 7.648 | 0.491 | 53.250 | 0.427 | -0.119 | 0.954 | 0.435 | 0.371 | |
| 12 | Afghanistan | 2018 | 2.694 | 7.631 | 0.508 | 53.575 | 0.374 | -0.091 | 0.928 | 0.385 | 0.405 | |

Uppgifter

Nedan följer uppgiftsbeskrivningarna som består av fem deluppgifter. Starta med att bekanta dig med den data som du ska analysera genom att öppna csv-filerna i Excel eller ett motsvarande program. Sätt dig in i vad det är för information som finns i respektive kolumner etc. Läs därefter igenom deluppgifterna och fundera på hur du ska lösa dessa. Ett bra sätt att börja lösa respektive deluppgift är att skapa ett enkelt flödesdiagram som hjälp att strukturera programkoden.

Ett bra sätt att strukturera programmet och samtidigt få en en mera lättläst kod är att använda en egen kodcell (den första kodcellen i lösningsfilen) där man anropar moduler som ska importeras och där programkoden finns till egendefinierade definitioner som ska användas i flera deluppgifter. Det räcker att man exekverar denna kodcell en gång för att programkoden ska bli 'synlig' i övriga kodceller.

När du är färdig med alla uppgifter bör du noggrant gå igenom din programkod och förvissa dig om att du får korrekta resultat. Kontrollera att programkoden inte bryter mot någon regel eller riktlinje som finns under *Förutsättningar*.

Innan inlämning bör du rensa alla variabler i minnet och testköra programkoden för alla deluppgifter en sista gång. Detta ska du göra för att Jupyter Notebook kan "komma ihåg" raderade variabler, vilket i vissa fall kan innebära att ett program som fungerar vid provkörning före rensning inte fungerar som tänkt nästa gång man laddar in programmet. Du rensar variablerna genom att antingen geom att välja alternativet "Restart & Clear Output" under menyalternativet "Kernel" i Jupyter Notebook eller starta om datorn och ladda in programmet igen.

Uppgift 1:

- a) Skriv en egendefinierad funktion `read_file(file_name)` som läser en csv-fil och och returnerar en lista med innehållet i filen `file_name`. Inargument till `read_file()` är alltså filnamnet (`file_name`) på csv-filen. Använd modulen CSV för att lösa uppgiften.

Observera: csv-filerna som används i denna inlämningsuppgift använder semikolon (;) som avgränsningstecken.

När du använder funktionen `open()` för att starta inläsning av filen, inkludera `encoding = 'UTF-8'` i argumentet för att säkerställa att inläsningen lyckas och att teckentolkningen avseende bl.a. åäö blir korrekt.

- b) Kontrollera att funktionen returnerar förväntad information genom att anropa funktionen `read_file()` och spara den returnerade listan i en lista med namnen **WHR1Data**, **WHR2Data** och **WHR3Data**. Skriv ut de tre första raderna från dessa listor:

WHR1Data

```
['Country name', 'Ladder score', 'Logged GDP per capita', 'Social support',  
'Healthy life expectancy', 'Freedom to make life choices', 'Generosity',  
'Perceptions of corruption']  
['Finland', '7.804', '10.792', '0.969', '71.150', '0.961', '-0.019',  
'0.182']  
['Denmark', '7.586', '10.962', '0.954', '71.250', '0.934', '0.134',  
'0.196']
```

WHR2Data

```
['country', 'region', 'subregion', 'pop2023', 'growthRate', 'area',  
'landAreaKm', 'density', 'densityMi', 'Rank', 'happiness2021',  
'happiness2020', 'scoreDifference', 'rank']  
['Finland', 'Europe', 'Northern Europe', '5545475.00', '0.00085', '338424',  
'303940', '182453.00', '472553.00', '118', '7.842', '7.809', '0.033', '1']  
['Denmark', 'Europe', 'Northern Europe', '5910913.00', '0.00487', '43094',  
'40000', '1477728.00', '3827316.00', '115', '7.62', '7.646', '-0.026', '2']
```

WHR3Data

```
['country', 'year', 'Life Ladder', 'Log GDP per capita', 'Social support',  
'Healthy life expectancy at birth', 'Freedom to make life choices',  
'Generosity', 'Perceptions of corruption', 'Positive affect', 'Negative  
affect']  
['Afghanistan', '2008', '3.724', '7.350', '0.451', '50.500', '0.718',  
'0.168', '0.882', '0.414', '0.258']  
['Afghanistan', '2009', '4.402', '7.509', '0.552', '50.800', '0.679',  
'0.191', '0.85', '0.481', '0.237']
```

Uppgift 2:

I denna uppgift ska du skriva ett program där du först matar in vilken kolumn (1–8) i listan **WHR1Data** som ska analyseras. Därefter ska de åtta länder med högst respektive lägst värde för år 2023 skrivas ut i en tabell med utseende enligt nedan. Resultatet ska också visas grafiskt i ett stapeldiagram med utseende enligt nedan.

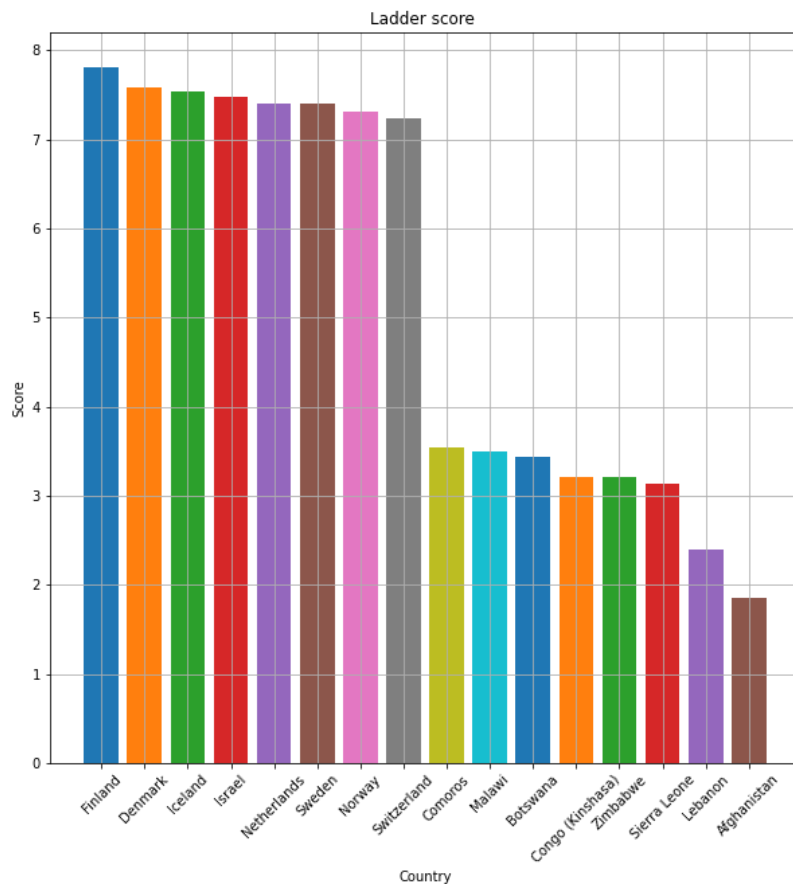
De 8 högsta värdena i kolumn 1 (Ladder score)

| Country name | Ladder score |
|--------------|--------------|
| Finland | 7.804 |
| Denmark | 7.586 |
| Iceland | 7.53 |
| Israel | 7.473 |
| Netherlands | 7.403 |
| Sweden | 7.395 |
| Norway | 7.315 |
| Switzerland | 7.24 |

De 8 lägsta värdena i kolumn 1 (Ladder score)

| | |
|------------------|-------|
| Comoros | 3.545 |
| Malawi | 3.495 |
| Botswana | 3.435 |
| Congo (Kinshasa) | 3.207 |

| | | |
|--------------|-------|--|
| Zimbabwe | 3.204 | |
| Sierra Leone | 3.138 | |
| Lebanon | 2.392 | |
| Afghanistan | 1.859 | |

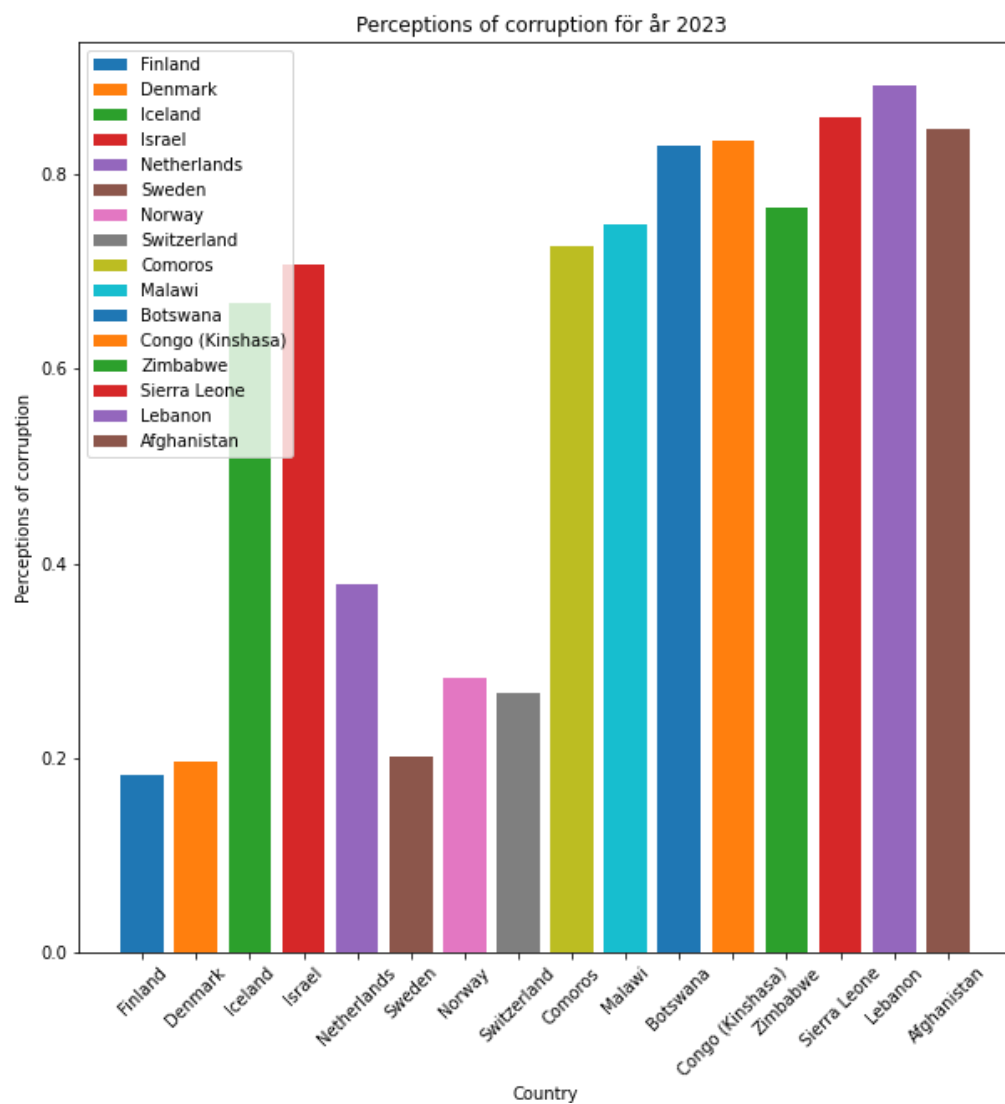


Uppgift 3:

I denna uppgift ska du skriva ett program som läser från listan **WHR1Data**. Skriv en egendefinierad funktion som beräknar och skriver ut min-, max-, medel- och medianvärdena för innehållet i samtliga kolumner för år 2023 i en tabell enligt nedan. Funktionen ska också skapa ett stapeldiagram över de åtta högsta och åtta lägsta värdena för varje kolumn. Totalt alltså sju stapeldiagram. Du ska själv skriva koden för beräkningarna av min-, max-, medel- och medianvärdena. Du får alltså inte använda Pythons inbyggda funktioner `min()`, `max()` och `sum()`. Medianen är det mittersta talet av de tal som finns i en kolumn efter att talen är sorterade i storleksordning. Om antalet tal är jämnt, definieras medianen som medelvärde av de två tal som är närmast mittvärdet. Du får använda funktionen `sorted()` för att sortera värdena i en kolumn.

| | | | | |
|------------------------------|------------|--------|--------|-------|
| År 2023 | | | | |
| Namn | medelvärde | median | min | max |
| Ladder score | 5.54 | 5.67 | 1.859 | 7.804 |
| Logged GDP per capita | 9.45 | 9.60 | 5.527 | 11.66 |
| Social support | 0.80 | 0.85 | 0.341 | 0.983 |
| Healthy life expectancy | 64.96 | 64.27 | 51.53 | 77.28 |
| Freedom to make life choices | 0.79 | 0.79 | 0.382 | 0.961 |
| Generosity | 0.02 | −0.05 | −0.254 | 0.531 |
| Perceptions of corruption | 0.73 | 0.82 | 0.146 | 0.929 |

Nedan visas stapeldiagrammet för kolumnen "Perceptions of corruption":



Uppgift 4.

I denna uppgift ska du skriva ett program som läser från listan **WHR2Data**. Användaren anger först den kontinent (Africa, Asia, Europe eller North America) som ska analyseras. Därefter ska programmet skriva ut "happiness index" och "rank" för länderna som ingår i denna kontinent i en tabell för åren 2020 och 2021. I tabellen ska länderna vara sorterade efter fallande "happiness index" (Python funktionen sorted() får användas). Även medelvärdet av "happiness index" baserat på samtliga länder som ingår i den angivna kontinenten ska finnas i tabellen. Programmet ska också rita ett stapeldiagram över "happiness index" och "rank" i form av cirklar för de fem länder med högsta respektive lägsta "happiness index" för åren 2020 och 2021. Nedan visas ett exempel på resultatet från en programkörning.

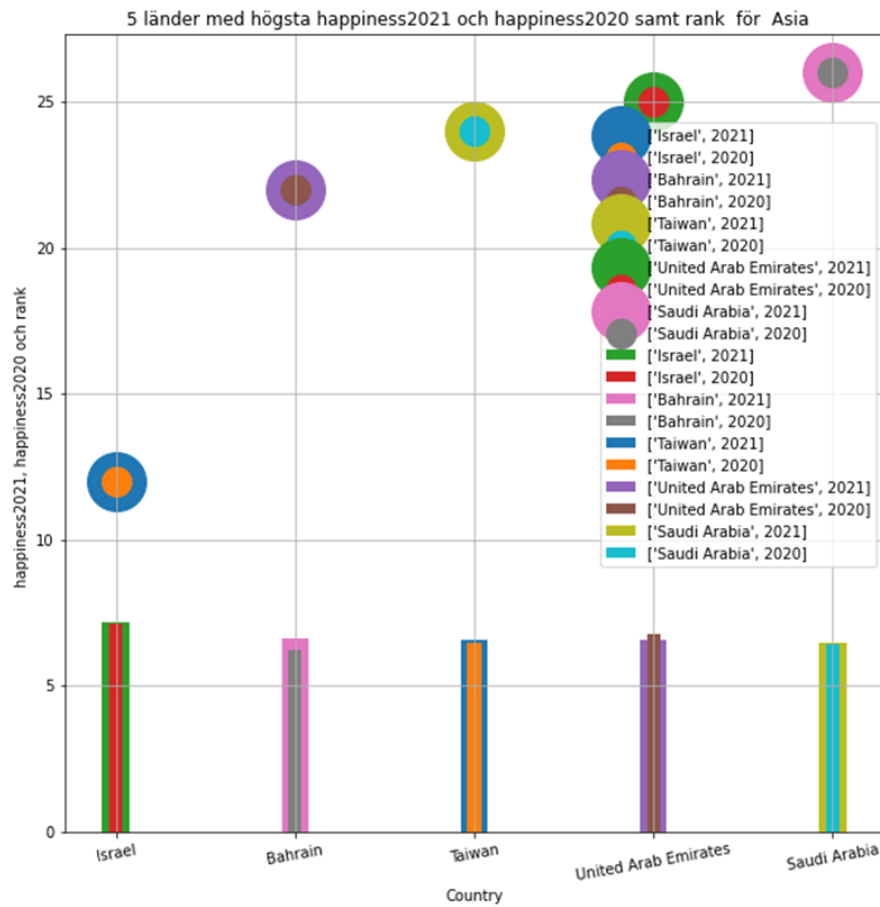
Ange region: Asia

Medelvärde 5.33 över happiness index för länder i Asia år 2021 och 2020

Länder i Asia:

| Land | Score för 2021 | Score för 2020 | Rank |
|----------------------|----------------|----------------|------|
| Israel | 7.157 | 7.129 | 12 |
| Bahrain | 6.647 | 6.227 | 22 |
| Taiwan | 6.584 | 6.455 | 24 |
| United Arab Emirates | 6.561 | 6.791 | 25 |
| Saudi Arabia | 6.494 | 6.406 | 26 |
| Singapore | 6.377 | 6.377 | 32 |
| Uzbekistan | 6.179 | 6.258 | 41 |
| Kazakhstan | 6.152 | 6.058 | 44 |
| Kuwait | 6.106 | 6.102 | 46 |
| Thailand | 5.985 | 5.999 | 53 |
| Japan | 5.94 | 5.871 | 55 |
| Philippines | 5.88 | 6.006 | 60 |
| South Korea | 5.845 | 5.872 | 61 |
| Kyrgyzstan | 5.744 | 5.542 | 66 |
| Mongolia | 5.677 | 5.456 | 69 |
| Hong Kong | 5.477 | 5.51 | 75 |
| Tajikistan | 5.466 | 5.556 | 76 |
| Vietnam | 5.411 | 5.353 | 77 |

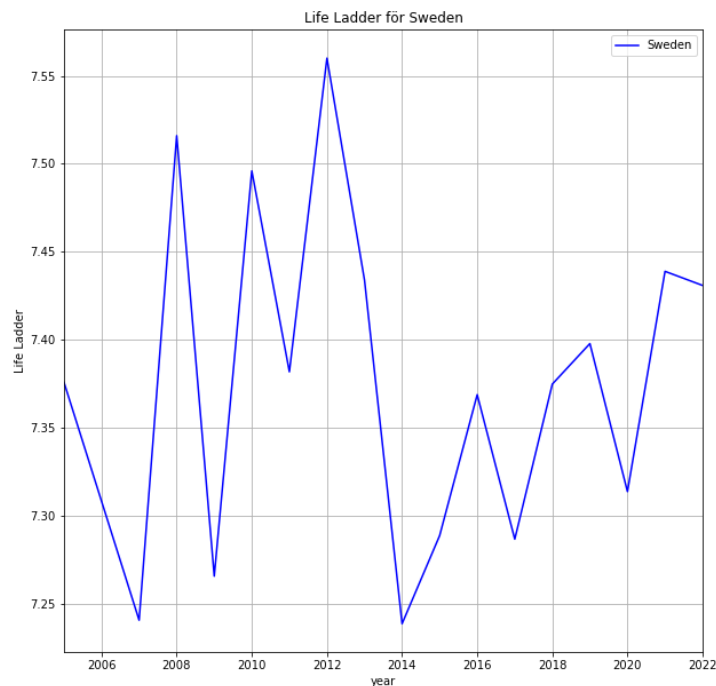
| | | | | |
|--------------|---------|---------|---------|---------|
| Malaysia | 5.384 | 5.384 | 79 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Indonesia | 5.345 | 5.286 | 80 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| China | 5.339 | 5.124 | 82 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Armenia | 5.283 | 4.677 | 84 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Nepal | 5.269 | 5.137 | 85 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Maldives | 5.198 | 5.198 | 87 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Azerbaijan | 5.171 | 5.165 | 88 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Turkmenistan | 5.066 | 5.119 | 95 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Laos | 5.03 | 4.889 | 98 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Bangladesh | 5.025 | 4.833 | 99 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Turkey | 4.948 | 5.132 | 102 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Pakistan | 4.934 | 5.693 | 103 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Georgia | 4.891 | 4.673 | 106 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Iraq | 4.854 | 4.785 | 109 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Cambodia | 4.83 | 4.848 | 112 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Iran | 4.721 | 4.672 | 116 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Lebanon | 4.584 | 4.772 | 121 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Myanmar | 4.426 | 4.308 | 123 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Jordan | 4.395 | 4.633 | 124 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Sri Lanka | 4.325 | 4.327 | 126 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| India | 3.819 | 3.573 | 136 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Yemen | 3.658 | 3.527 | 138 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| Afghanistan | 2.523 | 2.567 | 146 | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |



Uppgift 5:

I denna uppgift ska du skriva ett program som läser från listan **WHR3Data**. Användaren ska först välja ett land och därefter ska medelvärdet av landets 'Life Ladder' index beräknas och skrivas ut. Sedan ska landets 'Life Ladder' index under åren 2005-2022 presenteras i ett linjediagram. Slutligen ska programmet också beräkna förändringsfaktorn av 'Life Ladder' index för landet över åren 2005–2022 och presentera denna information i en tabell och i ett stapeldiagram. Se nedanstående utskrifter från en programkörning.

Ange landet som ska presenteras: *Sweden*
Medelvärde av Life Ladder är 7.4 för Sweden



| År | score för Sweden | Förändringsfaktor |
|------|------------------|-------------------|
| 2005 | 7.376 | -1.83 |
| 2007 | 7.241 | 3.80 |
| 2008 | 7.516 | -3.33 |
| 2009 | 7.266 | 3.17 |
| 2010 | 7.496 | -1.52 |
| 2011 | 7.382 | 2.41 |
| 2012 | 7.56 | -1.67 |
| 2013 | 7.434 | -2.62 |
| 2014 | 7.239 | 0.69 |
| 2015 | 7.289 | 1.10 |
| 2016 | 7.369 | -1.11 |
| 2017 | 7.287 | 1.21 |
| 2018 | 7.375 | 0.31 |
| 2019 | 7.398 | -1.14 |
| 2020 | 7.314 | 1.71 |
| 2021 | 7.439 | -0.11 |

