In [ ]:

```
# import pandas
import pandas as pd
# import numpy as np


#set all datasets as separate variables and dfs


df_gdp = pd.read_csv('datasets\gdppc.csv', on_bad_lines='skip')
df_happiness = pd.read_csv('../datasets/happiness.csv', on_bad_lines='skip')
df_hdi = pd.read_csv('../datasets/hdi.csv', on_bad_lines='skip')
df_population = pd.read_csv('../datasets/population.csv', on_bad_lines='skip')
df_landmass = pd.read_csv('../datasets/landmass.csv', on_bad_lines='skip')
df_qol = pd.read_csv('../datasets/qol_2021.csv', on_bad_lines='skip')
```

In [ ]:

```
# there's only a couple of countries in oecd  which are these:
# Austria, Australia, Belgium, Canada, Chile, Colombia, Costa Rica, Czech Republic, Deni
oecd_countries = ['Austria', 'Australia', 'Belgium', 'Canada', 'Chile', 'Colombia', 'Co
                  'Korea, Rep.', 'Latvia', 'Lithuania', 'Luxembourg', 'Mexico', 'Nether
# we don't care about the stats of other countries, so we'll just erase them.
```

In [ ]:

```
# we only want the oecd countries
#drop the indicator code etc
df_gdp.drop(['Indicator Code', 'Indicator Name', 'Country Code', '1960', '1961', '1962'
            '1987', '1988', '1989', '1990', '1991', '1992', '1993', '1994', '1995', '19
df_gdp = df_gdp.loc[df_gdp['Country Name'].isin(oecd_countries)]

#nice! we have all the countries
print(len(df_gdp))
# average the 2019 and 2020 data and write it as a new column

df_gdp['Combined'] = (df_gdp['2019'] + df_gdp['2020'])/2
df_gdp = df_gdp.drop(['2019', '2020'], axis=1)


df_gdp = df_gdp.round(2)
df_gdp = df_gdp.sort_values(by=['Combined'], ascending=False)
#find the middle of the list
# df_gdp_average = df_gdp.iloc[len(df_gdp)//2]
# df_gdp_average
```

Out[ ]:

```
38
Country Name      France
Combined         39804.5
Name: 77, dtype: object
```

In [ ]:

```
# happiness index have different names for the us and korea
oecd_countries += ['South Korea', 'United States', 'Slovakia']
df_happiness = df_happiness.loc[df_happiness['country'].isin(oecd_countries)]
print(len(df_happiness))

df_happiness.drop(['rank','pop2022'], axis=1, inplace=True)
# add the values in happiness2021 to happiness2020 and write it as a new column

df_happiness['Combined'] = (df_happiness['happiness2021'] + df_happiness['happiness2020
df_happiness.drop(['happiness2021', 'happiness2020'], axis=1, inplace=True)

df_happiness = df_happiness.round(2)

df_happiness.sort_values(by=['Combined'], ascending=False)


df_happiness_average = df_happiness.iloc[len(df_happiness)//2]

df_happiness_average
```

Out[ ]:

```
38
country     Belgium
Combined       6.85
Name: 19, dtype: object
```

In [ ]:

```
df_hdi = df_hdi[['HDI Rank', 'Country', '2019']]

# rank these based on hdi_rank column

df_hdi = df_hdi.sort_values(by=['HDI Rank'], ascending=True)

oecd_countries += ['Czechia']
df_hdi['Country'] = df_hdi['Country'].str.lstrip()
df_hdi = df_hdi.loc[df_hdi['Country'].isin(oecd_countries)]
print(len(df_hdi))

df_hdi['2019'] = df_hdi['2019'].astype(float)
# order in ascending way based on 2019

df_hdi = df_hdi.sort_values(by=['2019'], ascending=False)

# turn 2019 values into floating points



df_hdi_average = df_hdi.iloc[len(df_hdi)//2]

df_hdi_average
```

Out[ ]:

```
38
HDI Rank              23
Country      South Korea
2019               0.916
Name: 90, dtype: object
```

In [ ]:

```
df_landmass = df_landmass[['Country Name', '2020']]

df_landmass = df_landmass.sort_values(by=['2020'], ascending=False)

df_landmass = df_landmass = df_landmass.loc[df_landmass['Country Name'].isin(
    oecd_countries)]


df_landmass['2020'] = df_landmass['2020'].astype(float)
# order in ascending way based on 2019
# turn 2019 values into floating points


df_landmass_average = df_landmass.iloc[len(df_landmass)//2]

df_landmass_average
```

Out[ ]:

```
Country Name      Iceland
2020             100830.0
Name: 114, dtype: object
```

In [ ]:

```
df_population = df_population[['Location','Variant', 'Time', 'PopTotal']]
df_population

oecd_countries += ['Republic of Korea', 'United States of America']

df_population = df_population = df_population.loc[df_population['Location'].isin(
    oecd_countries)]

#convert time to string
df_population['Time'] = df_population['Time'].astype(str)

# only keep the years that are 2021
df_population = df_population[df_population.Variant == 'Constant fertility']
df_population = df_population[df_population.Time == '2021']

df_population['PopTotal'] = df_population['PopTotal'].astype(str)
#remove the . from the popTotal
df_population['PopTotal'] = df_population['PopTotal'].str.replace('.', '', regex=True)
df_population['PopTotal'] = df_population['PopTotal'].astype(int)

df_population = df_population.sort_values(by=['PopTotal'], ascending=False)

df_population_average = df_population.iloc[len(df_population)//2]
df_population_average
```

Out[ ]:

```
Location                Portugal
Variant       Constant fertility
Time                        2021
PopTotal                10167755
Name: 200582, dtype: object
```

In [ ]:

```
#qol data
#only LIFE_SATISFACTION
df_qol = df_qol[['Country', 'Indicator', 'Value', 'Inequality']]

oecd_countries += ['Korea']

df_qol = df_qol[df_qol.Indicator == 'Life satisfaction']
df_qol = df_qol[df_qol.Inequality == 'Total']

df_qol = df_qol.sort_values(by=['Value'], ascending=False)

df_qol = df_qol.loc[df_qol['Country'].isin(oecd_countries)]

#TODO costa rica unfortunately does not exist in the dataset
df_qol_average = df_qol.iloc[len(df_qol)//2]

df_qol_average
```

Out[ ]:

```
Country            United Kingdom
Indicator       Life satisfaction
Value                         6.8
Inequality                  Total
Name: 1750, dtype: object
```

In [ ]:

```
#final
print("GDP")
print(df_gdp_average.values)
print("Happiness")
print(df_happiness_average.values)
print("HDI")
print(df_hdi_average.values)
print("Landmass")
print(df_landmass_average.values)
print("Population")
print(df_population_average.values)
print("Quality of Life")
print(df_qol_average.values)
```

```
GDP
['France' 39804.5]
Happiness
['Belgium' 6.85]
HDI
['23' 'South Korea' 0.916]
Landmass
['Iceland' 100830.0]
Population
['Portugal' 'Constant fertility' '2021' 10167755]
Quality of Life
['United Kingdom' 'Life satisfaction' 6.8 'Total']
```