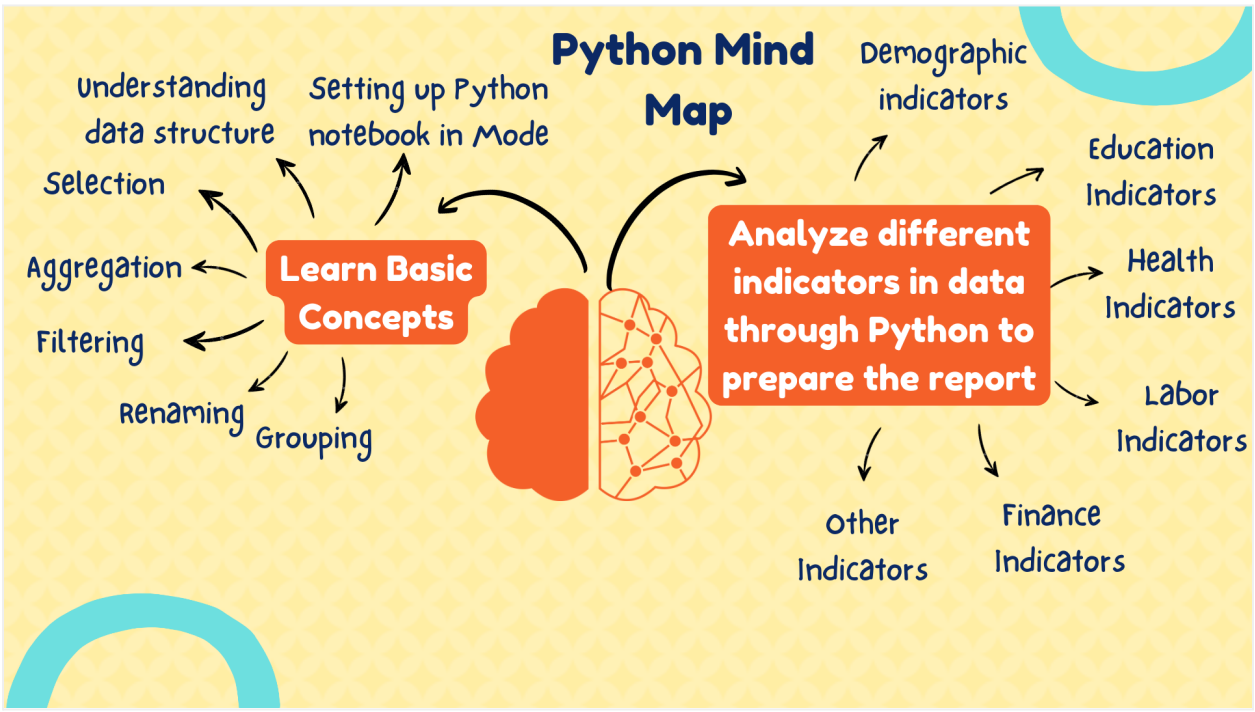
Gender Case Study through Python



We will first go through basic concepts in Python and then move to solve report questions.

Basic Python Concepts

Let's first learn some basic concepts for extracting and analyzing data in Python

Question 0:

Setting up Python Notebook and Getting the Dataset

Method:

- Import all required packages
  import pandas as pd
  import numpy as np
  import datetime as dt

- Step 1: Run the Query with all data
  "SELECT * FROM career_nub.gender_data;"

- *Make sure to remove the "Limit 100" button on the top to extract all rows*
- This will generate a query ID that you can see in the last section of the URL
    - Copy this query ID

Eg of query id:
*https://app.mode.com/editor/career_nub/reports/c2a33954975a/queries/**972e54b23781***
This is just an example. It will be different for each user when you run this command through your account.

- Step 2: Go to the left panel and click on the "Python Notebook" button to create a New Python Notebook

- Step 3: Use the query id from the URL that you copied above and use the following code to create a new dataframe
gender_df = datasets['query_id']

Refer to this video for a detailed step-by-step process to get your dataset in the Python notebook

Answer 0:

gender_df = datasets['query_id']

Question 1: Level: Easy

Get an overview of the dataset and output 100 rows from the dataset

Method:

- The function "**df.head(x)**" gives x number of rows and all columns of the dataset

Answer 1:

gender_df.head(100)

Output 1:

We extracted 100 rows of data with all 5 columns in the dataset.

Question 2: Level: Easy

Get specific columns from the data.
Get the Country, Year, Indicator, and Value from the dataset *(only 100 rows)*

Method 1:

- Create a new dataframe with a subset of columns using the
  new_df = df [['column_a','column_b','column_c']]
- Let's print out 100 rows of the above-created subset dataset

Answer 2:

```
subset_df = gender_df [ [ 'country', 'year', 'indicator', 'value' ] ]
subset_df.head(100)
```

Method 2:

- Using the loc function to Select specific columns
  - Loc Function lets us access a group of rows and columns by label(s)
- Select specific columns by their names using the loc function
  - **dataframe.loc[:, ['column_a','column_b','column_c']**
  - Here the colon ':' means that we want all the rows from the dataset
  - Specify the required column names within square brackets
- Save it as a new dataframe
- Print 100 rows of new dataframe

Answer 2:

```
subset_df = gender_df.loc [:,  [ 'country', 'year', 'indicator', 'value' ] ]
subset_df.head(100)
```

Output 2:

You can use either method to extract 100 rows of data with columns - country, year, indicator, and value - from the dataset.

Question 3: Level: Easy

Get specific columns from the data.
Get the Country, Year, Indicator, and Value from the dataset *(only 100 rows)*, and assign different column names to the Indicator and Value columns.

Method:

- Get specific columns from the dataset using either method 1 or method 2 given in question 2 above
- **Renaming columns**
    - Rename columns in a dataframe using
    - **dataframe.rename(columns = {'column': 'column_new_name'}, inplace=True)**
    - The 'inplace=True' parameter makes sure that the current column names are replaced and new columns are not created

  Note that this changes the column names in the dataframe and is not just for the output
- Print the dataframe with new column names

Answer 3:

```
subset_df = gender_df[['country','year','indicator','value']]
subset_df.rename(columns = {'indicator' : 'metric', 'value' : 'number'}, inplace=True)
subset_df
```

Output 3:

We extracted 100 rows of data with columns - country, year, indicator, and value - from the dataset while renaming some of the columns.

Question 4: Level: Easy

How many total rows are present in the dataset?

Method:

- Shape function is used to get the count of total rows and total columns in the dataset
    - Syntax is : **df.shape**

Answer 4:

```
gender_df.shape
```

Output 4:

We see that there are 52,653 rows in the dataset.

Question 5: Level: Easy

How many countries does our dataset contain?

Method:

- Write the dataframe name followed by column name for which unique count is required
    - The column name should be written inside single quotes within square brackets
    - Syntax is: dataframe[‘column a’]
- The function to get unique count is **nuinque()**
    - It is specified after the column name separated by a dot
    - Syntax is **df[‘column a’].nunique()**

Answer 5:

```
gender_df ['country'].nunique()
```

Output 5:

We see that there are 193 unique countries in the dataset.

Question 6: Level: Easy

List down all the countries present in our dataset.

Method:

- Write the dataframe name followed by column name for which unique count is required
    - The column name should be written inside single quotes within square brackets
    - Syntax is: dataframe[‘column a’]
- The function to get unique values is **uinque()**
    - It is specified after the column name separated by a dot
    - Syntax is **dataframe[‘column a’].unique()**

Answer 6:

```
gender_df ['country'].unique()
```

Output 6:

We see that there are 193 unique countries in the dataset and can see the names of each of these countries.

Question 7: Level: Easy

How many indicators does our dataset contain?

Method:

- Since each indicator will have multiple rows of data, we need to calculate the distinct number of indicators from the dataset
- **nunique()** followed by column name eliminates the repetitive appearance of the same data

Answer 7:

```
gender_df['indicator'].nunique()
```

Output 7:

We see that there are 63 unique indicators in the dataset.

Question 8: Level: Easy

List down all the indicators present in our dataset.

Method:

- Since each indicator will have multiple rows of data, we need to calculate the distinct number of indicators from the dataset
- **unique()** followed by column name eliminates the repetitive appearance of the same data

Answer 8:

```
gender_df['indicator'].unique()
```

Output 8:

We see that there are 63 unique indicators in the dataset and can see the names of each of these indicators.

CONDITIONAL FILTERING: Filter data based on a condition

Question 9: Level: Easy

List all indicators within the category of Education

Method:

- Dataframes can be filtered in multiple ways:
- Let's use boolean indexing method:
    - **Dataframe [dataframe [‘column name’] == ‘Condition’]**
- Create a subset dataframe with all data for category education
- Use unique() for column indicator of the subset dataset

Answer 9:

```
ed=gender_df [ gender_df ['category'] == 'Education']
ed['indicator'].unique()
```

Output 9:

We see that there are 12 indicators in the Education category. We get the list of all indicators within the category Education using this code.

Question 10: Level: Easy

Get all data post-2019

Method:

- Step 1: We can filter our dataset based on a condition using df[df['column']=='value']

Answer 10:

```
gender_df[gender_df['year']>2019]
```

Output 10:

We filter out the data for all years post-2019 and we see there are 3,159 rows of data.

Question 11: Level: Easy

Get all data from these five Nordic countries:

- Denmark, Finland, Iceland, Norway, and Sweden

Method:

- Step 1: We can filter our dataset based on a condition using
  df [df['column'].isin(['value1','value2'])]
  - Here the **isin** condition compares all the values in the list

Answer 11:

```
gender_df[gender_df['country'].isin(['Denmark', 'Finland', 'Iceland', 'Norway', 'Sweden'])]
```

Output 11:

We filter out the data for these Nordic countries and we see there are 1,492 rows of data.

MULTIPLE CONDITIONAL FILTERING: Filter data based on multiple conditions

Question 12: Level: Medium

Get all data from 2015 onwards for India

Method:

- Step 1.1: We can filter our dataset based on a condition using
  df[(df['column']=='value')]

- Step 1.2: We stack multiple conditions using logic gates (AND, OR etc.)

- Step 1.3: We can add another filtering condition using
  df[(df['column1']>='value1') & (df['column2'] == 'value2')]

- Here the **'&'** condition mentions that both conditions should be True to return an output

Answer 12:

```
gender_df [ (gender_df['year'] >= 2015) &
            (gender_df['country'] == 'India') ]
```

Output 12:

We filter out the data for all years post-2015 in India and we see there are 195 rows of data.

Question 13: Level: Medium

Get all rows of data which are either in Australia or from the Health category

Method:

- Step 1.1: We can filter our dataset based on a condition using
  df[(df['column']=='value')]
- Step 1.2: We stack multiple conditions using logic gates (AND, OR etc.)
- Step 1.3: We can add another filtering condition using
  df[(df['column1']=='value1') | (df['column2'] == 'value2')]
- Here the **'|'** condition mentions that both conditions should be True to return an output

Answer 13:

```
gender_df [ (gender_df['country'] == 'Australia') |
            (gender_df['category'] == 'Health') ]
```

Output 13:

We filter out the data for rows in either Australia or in the Health category and we see there are 13,790 rows of data.

Question 14: Level: Easy

What is the range of time period in the dataset?

Method:

- Step 2.1: We can find the maximum and mean values in a column using
  df['column'].max()
  df['column'].mean()
- Step 2: Print the Maximum and Average funding raised
  Use print(f'Output') function

Answer 14:

```
min_year = gender_df['year'].min()

max_year = gender_df['year'].max()

print(f'Start Year of the data is : {min_year}, and the end year is : {max_year} ')
```

Output 14:

We see that the data is from 2012 to 202

Report Questions

Let's now move to some of the advanced topics in Python and solve questions to prepare our report.

Report Question 1: Understanding demographic indicators

Report Question 1.1: Level: Easy

Find out the Total female population per year

- Indicator Name: Population, female

Method:

- Step 1: We can filter our dataset for a particular indicator and year using filter_df = df[(df['column']=='value')]

- The "**GROUPBY**" Function in Python Pandas is used to arrange identical data into groups
    - This provides pivot table-like functionality through Python and is extremely useful for performing analysis

- Step 2.1: Use **groupby** and mention the column name at which you want to group using df.groupby('column', as_index=False)
    - The 'as_index=False' parameter mentions that we want our groupby column not as an index

- Step 2.2: Use **agg** function to count values in another column using df.groupby('column', as_index=False).agg(new_column = ('column','aggregation function'))

Report Answer 1.1:

filter_df = gender_df [(gender_df['indicator'] == 'Population, female')]
filter_df.groupby('year', as_index=False).agg(total_female_population = ('value','sum'))

Report Output 1.1:

The total female population per year increased from 3.5Bn in 2012 to 3.8Bn in 2020

Report Question 1.2: Level: Medium

Find the yearly total and % of Female population across all countries

- Indicator Names:
  - Population, female
  - Population, male
  - *Total Population = (Population, female) + (Population, male)*

Method:

- Step 1: We can filter our dataset for particular indicators using
  filter_df = df[(df['column1'].isin(['value1','value2'])]

- Step 2: Since we want the sum of populations per year for each indicator we can use **groupby** and mention the column names at which you want to group using

  df.groupby(['column1, 'column2'], as_index=False)

- Step 3: We can pivot the results on the year column to get the population values of each indicator side-by-side for comparison using

  df.pivot(index ='column_to_pivot_on', columns =row_column', values = 'value_column')

- Step 4: Create a new columns for total population and % female population using the above calculated columns using

  df['new_column'] = df['column1'] + df['column2']
  df['new_column2'] = np.round(100.0 * df['column1'] / df['column2'],2)

Report Answer 1.2:

```
filter_df = gender_df [ (gender_df['indicator'].isin(['Population, female','Population, male'])) ]

agg_df = filter_df.groupby(['year','indicator'], as_index=False).agg(value = ('value','sum'))

pivot_df = agg_df.pivot(index ='year', columns ='indicator', values = 'value')

pivot_df['Total Population'] = pivot_df['Population, female'] + pivot_df['Population, male']
pivot_df['% Female Population'] = np.round(100.0 * pivot_df['Population, female'] / pivot_df['Total Population'],2)

pivot_df
```

Report Output 1.2:

We see that the global female population has increased from 3.4Bn or 49.57% of the total population in 2012 to 3.8 Bn or 49.58% of the overall population in 2020

Report Question 1.3: Level: Hard

Find the top 5 countries with the highest percentage of the female population in the year 2020

- Indicator Names:
  - Population, female
  - Population, male

Method:

- Step 1: We can filter our dataset for particular indicators in a year using
  `filter_df = df[ (df['column1'].isin(['value1','value2']) & df['column2'] == 'value' ]`

- Step 2: Since we want the sum of populations per country for each indicator we can use **groupby** and mention the column names at which you want to group using

  `df.groupby(['column1, 'column2'], as_index=False)`

- Step 3: We can pivot the results on the country column to get the population values of each indicator side-by-side for comparison using

  `df.pivot(index ='column_to_pivot_on', columns =row_column', values = 'value_column')`

- Step 4: Create a new columns for total population and % female population using the above calculated columns using

df[‘new_column’] = df[‘column1’] + df[‘column2’]
df[‘new_column2’] = np.round(100.0 * df[‘column1’] / df[‘column2’],2)

- Step 5: Since we want countries with highest % femal population, we can sort the final output using
  df.sort_values('column', ascending=False)

- Step 6: To get the top N countries we can get the first 5 rows of the above-sorted dataset using
  df.head(5)

Report Answer 1.3:

```
filter_df = gender_df [ (gender_df['indicator'].isin(['Population, female','Population, male'])) &
                        (gender_df['year'] == 2020) ]

agg_df = filter_df.groupby(['country','indicator'], as_index=False).agg(value = ('value','sum'))

pivot_df = agg_df.pivot(index ='country', columns ='indicator', values = 'value')

pivot_df['Total Population'] = pivot_df['Population, female'] + pivot_df['Population, male']
pivot_df['% Female Population'] = np.round(100.0 * pivot_df['Population, female'] / pivot_df['Total Population'],2)

sorted_df = pivot_df.sort_values('% Female Population', ascending=False)

sorted_df.head(5)
```

Report Output 1.3:

Countries with the highest percentage of female population are:
Nepal, Hong Kong, Latvia, Lithuania, and Ukraine

Report Question 1.4: Level: Medium

Which country had the highest sex ratio at birth in 2020?

- Indicator Name: Sex ratio at birth (male births per female births)

Method:

- Step 1: We can filter our dataset for a particular indicator and year using
  filter_df = df[(df['column1']=='value') & (df['column2']=='value')]

- Step 2: Since we want the highest value of sex ratio at birth, we can sort the final output in descending order by using
  df.sort_values('column', ascending = False)

- Step 3: To get the bottom N countries we can get the first row of the above-sorted dataset using
  df.head(1)

Report Answer 1.4:

```
filter_df = gender_df [ (gender_df['indicator'] == 'Sex ratio at birth (male births per female births)') & (gender_df['year'] == 2020)]

sorted_df = filter_df.sort_values('value', ascending = False)

sorted_df.head(1)
```

Report Output 1.4:

The sex ratio at birth (male births per female births) was highest for Azerbaijan in 2020

Report Question 2: Understanding Education indicators

Report Question 2.1: Level: Medium

Find the bottom 2 countries with the lowest adult female literacy rate in 2019

- Indicator Names:

- Literacy rate, adult female

Method:

- Step 1: We can filter our dataset for a particular indicator and year using
  filter_df = df[(df['column1']=='value') & (df['column2']=='value')]

- Step 2: Since we want the lowest value of adult female literacy rate, we can sort the final output using
  df.sort_values('column')

- Step 3: To get the bottom N countries we can get the first 2 rows of the above-sorted dataset using
  df.head(2)

Report Answer 2.1:

```
filter_df = gender_df [ (gender_df['indicator'] == 'Literacy rate, adult female') & (gender_df['year'] == 2019)]

sorted_df = filter_df.sort_values('value')

sorted_df.head(2)
```

Report Output 2.1:

We see that the countries with the lowest adult female literacy rate in the year 2019 are Pakistan and Togo.

Report Question 2.2: Level: Medium

Find the change in adult female literacy rate between 2012 and 2020 for Bangladesh

- Indicator Name: Literacy rate, adult female

Method:

- Step 1: We can filter our dataset for an indicator for a particular country and a range of years using

  filter_df = df[(df['column1'].between(value1,value2)) & (df['column2']=='value') & (df['column3']=='value')]

Report Answer 2.2:

```
filter_df = gender_df [ (gender_df['year'].between(2012,2019)) &
                        (gender_df['indicator'] == 'Literacy rate, adult female') &
                        (gender_df['country'] == 'Bangladesh') ]

filter_df
```

Report Output 2.2:

Between 2012 and 2019, the adult female literacy rate increased from 54% to 72% in Bangladesh

Report Question 2.3: Level: Hard

Find the number of Female Children out of primary school as a percentage of overall children out of primary school

- Indicator Names:
  - Children out of school, primary, female
  - Children out of school, primary, male

Method:

- Step 1: We can filter our dataset for particular indicators using
  filter_df = df[(df['column1'].isin([‘value1’,’value2’])]

- Step 2: Since we want the sum of children out of school per year we can use **groupby** and mention the column names at which you want to group using

  df.groupby(['column1, 'column2'], as_index=False)

- Step 3: We can pivot the results on the year column to get the population values of each indicator side-by-side for comparison using

  df.pivot(index ='column_to_pivot_on', columns =row_column', values = ‘value_column’)

- Step 4: Create new columns for total children out of school and % female children out of school using the above-calculated columns using

  df[‘new_column’] = df[‘column1’] + df[‘column2’]
  df[‘new_column2’] = np.round(100.0 * df[‘column1’] / df[‘column2’],2)

Report Answer 2.3:

```
filter_df = gender_df [ (gender_df['indicator'].isin(['Children out of school, primary, female','Children out of school, primary, male'])) &
                        (gender_df['country'] == 'Cameroon') ]

agg_df = filter_df.groupby(['year','indicator'], as_index=False).agg(value = ('value','sum'))

pivot_df = agg_df.pivot(index ='year', columns ='indicator', values = 'value')

pivot_df['Total Children out of primary school'] = pivot_df['Children out of school, primary, female'] + pivot_df['Children out of school, primary, male']
pivot_df['% Female Children out of primary school'] = np.round(100.0 * pivot_df['Children out of school, primary, female'] / pivot_df['Total Children out of primary school'],2)

pivot_df
```

Report Output 2.3:

The percentage of female children out of school (of the total children out of primary school) in Cameroon has improved from 92% in 2016 to 76% in 2019.

Report Question 3: Understanding Health indicators

Report Question 3.1: Level: Hard

Which are the best/worst countries based on the female & male Life expectancy among SAARC countries in 2020
Indicator Names:

- Life expectancy at birth, female
- Life expectancy at birth, male

SAARC countries: Afghanistan, Bangladesh, Bhutan, India, Maldives, Nepal, Pakistan, and Sri Lanka

Method:

- Step 1: We can filter our dataset for particular countries and indicators in a year using filter_df = df[(df['column1']=='value') & (df['column2] == 'value') & (df['column3] == 'value')]

- Step 2: Since we want the average values per country for each indicator we can use **groupby** and mention the column names at which you want to group using

  df.groupby(['column1, 'column2'], as_index=False).agg(new_column = ('column','aggregation function'))

- Step 3: We can pivot the results on the 'country' column to get the average values of each indicator side-by-side for comparison using
  df.pivot(index ='column_to_pivot_on', columns =row_column', values = 'value_column')

- Step 4: Since we want the highest value of Life expectance for females, we can sort the final output using
  df.sort_values('column', ascending=False)

Report Answer 3.1:

```
filter_df = gender_df [ (gender_df['country'].isin(['Afghanistan', 'Bangladesh', 'Bhutan', 'India', 'Maldives', 'Nepal', 'Pakistan', 'Sri Lanka'])) &
                (gender_df['indicator'].isin(['Life expectancy at birth, female','Life expectancy at birth, male'])) &
                (gender_df['year'] == 2020) ]

agg_df = filter_df.groupby(['country','indicator'], as_index=False).agg(avg_value = ('value','mean'))

pivot_df = agg_df.pivot(index ='country', columns ='indicator', values = 'avg_value')

sorted_df = pivot_df.sort_values('Life expectancy at birth, female', ascending=False)

sorted_df
```

Report Output 3.1:

We see that among SAARC countries, Maldives has the highest life expectancy for both females at 81 years and males at 77 yrs, with the lowest being in Afghanistan for both females (67) and males (64).

Report Question 3.2: Level: Hard

Compare the female and male Under-5 mortality rates in the year 2020 among SAARC countries
Indicator Names:

- Mortality rate, under-5, female
- Mortality rate, under-5, male

SAARC countries: Afghanistan, Bangladesh, Bhutan, India, Maldives, Nepal, Pakistan, and Sri Lanka

Method:

- Step 1: We can filter our dataset for particular countries and indicators using filter_df = df[(df['column1']=='value') & (df['column2] == 'value')]

- Step 2: Since we want the average values per country for each indicator we can use **groupby** and mention the column names at which you want to group using

  df.groupby(['column1, 'column2'], as_index=False).agg(new_column = ('column','aggregation function'))

- Step 3: We can pivot the results on the 'country' column to get the average values of each indicator side-by-side for comparison using
  df.pivot(index ='column_to_pivot_on', columns =row_column', values = 'value_column')

- Step 4: Since we want the highest value of under-5 mortality for females, we can sort the final output using
  df.sort_values('column', ascending=False)

Report Answer 3.2:

```
filter_df = gender_df [ (gender_df['country'].isin(['Afghanistan', 'Bangladesh', 'Bhutan', 'India', 'Maldives', 'Nepal', 'Pakistan', 'Sri Lanka'])) &
                        (gender_df['indicator'].isin(['Mortality rate, under-5, female','Mortality rate, under-5, male'])) &
                        (gender_df['year'] == 2020) ]

agg_df = filter_df.groupby(['country','indicator'], as_index=False).agg(avg_value = ('value','mean'))

pivot_df = agg_df.pivot(index ='country', columns ='indicator', values = 'avg_value')

sorted_df = pivot_df.sort_values('Mortality rate, under-5, female', ascending=False)

sorted_df
```

Report Output 3.2:

Pakistan has the worst under-5 mortality rate for both females and males with Maldives being the best for both genders.

Report Question 3.3: Level: Medium

Find the countries that had the highest Prevalence of HIV among females in the age group 15-24) in the year 2020:

- Indicator Name: Prevalence of HIV, female (% ages 15-24)

Method:

- Step 1: We can filter our dataset for a particular indicator and year using
  filter_df = df[(df['column1']=='value') & (df['column2] == 'value')]

- Step 2: Since we want the highest value of the Prevalence of HIV in females, we can sort the final output in a descending order using
  df.sort_values('column', ascending=False)

Report Answer 3.3:

```
filter_df = gender_df [ (gender_df['indicator'] == 'Prevalence of HIV, female (% ages 15-24)')  &
          (gender_df['year'] == 2020) ]

sorted_df = filter_df.sort_values('value', ascending=False)

sorted_df.head(5)
```

Report Output 3.3:

The bottom 5 countries in terms of the prevalence of HIV in females are South Africa, Lesotho, Botswana, Mozambique, and Zambia. We can share this analysis with the health vertical to focus on these countries and take steps to reduce HIV prevalence.

Report Question 4: Understanding Labor indicators

Report Question 4.1: Level: Hard

How has the percentage of female employment in Agriculture changed from 2012 to 2019

- Indicator Names:
  - Employment in agriculture, female

- Population employed, female

Method:

- Step 1: We can filter our dataset for particular indicators and years using
  filter_df = df[(df['column1'].isin([‘value1’,’value2’]) &
  (df['column1'].between(value1,value2))]

- Step 2: Since we want the sum of females employed in agriculture and total females employed for each year, we can use **groupby** and mention the column names at which you want to group using

  df.groupby(['column1, 'column2'], as_index=False)

- Step 3: We can pivot the results on the year column to get the values of each indicator

  df.pivot(index ='column_to_pivot_on', columns =row_column', values = ‘value_column’)

- Step 4: Create a new columns for % female employment in agriculture using the above calculated columns using

  df[‘new_column’] = np.round(100.0 * df[‘column1’] / df[‘column2’],2)

Report Answer 4.1:

```
filter_df = gender_df [ (gender_df['indicator'].isin(['Employment in agriculture, female','Population employed, female'])) &
                        (gender_df['year'].between(2012,2019)) ]

agg_df = filter_df.groupby(['year','indicator'], as_index=False).agg(value = ('value','sum'))

pivot_df = agg_df.pivot(index ='year', columns ='indicator', values = 'value')

pivot_df['% Female Employment in Agriculture'] = np.round(100.0 * pivot_df['Employment in agriculture, female'] / pivot_df['Population employed, female'],2)

pivot_df
```

Report Output 4.1:

Employment in agriculture for females has reduced from 32% to 27% between 2012 and 2019

Report Question 4.2: Level: Hard

How has the percentage of female employment in the Industry sector changed from 2012 to 2019

- Indicator Names:
    - Employment in industry, female
    - Population employed, female

Method:

- Step 1: We can filter our dataset for particular indicators and years using
  filter_df = df[(df['column1'].isin([‘value1’,’value2’]) &
  (df['column1'].between(value1,value2))]

- Step 2: Since we want the sum of females employed in industry and total females employed for each year, we can use **groupby** and mention the column names at which you want to group using

  df.groupby(['column1, 'column2'], as_index=False)

- Step 3: We can pivot the results on the year column to get the values of each indicator

  df.pivot(index ='column_to_pivot_on', columns =row_column', values = ‘value_column’)

- Step 4: Create a new columns for % female employment in industry using the above calculated columns using

  df[‘new_column’] = np.round(100.0 * df[‘column1’] / df[‘column2’],2)

Report Answer 4.2:

```
filter_df = gender_df [ (gender_df['indicator'].isin(['Employment in industry, female','Population employed, female'])) &
                        (gender_df['year'].between(2012,2019)) ]

agg_df = filter_df.groupby(['year','indicator'], as_index=False).agg(value = ('value','sum'))

pivot_df = agg_df.pivot(index ='year', columns ='indicator', values = 'value')

pivot_df['% Female Employment in Industry'] = np.round(100.0 * pivot_df['Employment in industry, female'] / pivot_df['Population employed, female'],2)

pivot_df
```

Report Output 4.2:

Employment in the industry sector for females has also dropped a bit from 16.3% to 15% between 2012 and 2019

Report Question 4.3: Level: Hard

How has the percentage of female employment in the Service sector changed from 2012 to 2019

- Indicator Names:
    - Employment in services, female
    - Population employed, female

Method:

- Step 1: We can filter our dataset for particular indicators and years using
  filter_df = df[(df['column1'].isin([‘value1’,’value2’]) &
  (df['column1'].between(value1,value2))]

- Step 2: Since we want the sum of females employed in services and total females employed for each year, we can use **groupby** and mention the column names at which you want to group using

  df.groupby(['column1, 'column2'], as_index=False)

- Step 3: We can pivot the results on the year column to get the values of each indicator

  df.pivot(index ='column_to_pivot_on', columns =row_column', values = ‘value_column’)

- Step 4: Create a new columns for % female employment in services using the above calculated columns using

  df[‘new_column’] = np.round(100.0 * df[‘column1’] / df[‘column2’],2)

Report Answer 4.3:

```
filter_df = gender_df [ (gender_df['indicator'].isin(['Employment in services, female','Population employed, female'])) &
                        (gender_df['year'].between(2012,2019)) ]

agg_df = filter_df.groupby(['year','indicator'], as_index=False).agg(value = ('value','sum'))

pivot_df = agg_df.pivot(index ='year', columns ='indicator', values = 'value')
```

pivot_df['% Female Employment in Services'] = np.round(100.0 * pivot_df['Employment in services, female'] / pivot_df['Population employed, female'],2)

pivot_df

Report Output 4.3:

Employment in the service sector for females has increased from 51% to 57% between 2012 and 2019

Overall we see a shift in female employment from Agriculture to the Services sector which now has a 57% share of female employment.

Report Question 5: Understanding Financial Indicators

Report Question 5.1: Level: Hard

How has female debit card ownership changed in 2017 compared to 2014?

- Indicator Names:
  - Debit card ownership, female, 15+
  - Population, female, 15+

Method:

- Step 1: We can filter our dataset for particular indicators and years using filter_df = df[(df['column1'].isin(['value1','value2']) & (df['column1'].isin(value1,value2))]

- Step 2: Since we want the sum of females who own a debit card and adult female population for each year, we can use **groupby** and mention the column names at which you want to group using

df.groupby(['column1, 'column2'], as_index=False)

- Step 3: We can pivot the results on the year column to get the values of each indicator

  df.pivot(index ='column_to_pivot_on', columns =row_column', values = 'value_column')

- Step 4: Create a new columns for % females who own a debit card using the above calculated columns using

  df['new_column'] = np.round(100.0 * df['column1'] / df['column2'],2)

Report Answer 5.1:

```
filter_df = gender_df [ (gender_df['indicator'].isin(['Debit card ownership, female, 15+','Population, female, 15+'])) &
                        (gender_df['year'].isin([2014,2017])) ]

agg_df = filter_df.groupby(['year','indicator'], as_index=False).agg(value = ('value','sum'))

pivot_df = agg_df.pivot(index ='year', columns ='indicator', values = 'value')

pivot_df['% Females who own a debit card'] = np.round(100.0 * pivot_df['Debit card ownership, female, 15+'] / pivot_df['Population, female, 15+'],2)

pivot_df
```

Report Output 5.1:

There is an increase in the percentage of females, owning a debit card from 36% in 2014 to 43% in 2017 indicating an increase in financial inclusion.

Report Question 5.2: Level: Medium

Find the five countries with the lowest female debit card ownership in 2017

- Indicator: Debit card ownership, female (% age 15+)

Method:

- Step 1: We can filter our dataset for a particular indicator and year using
  filter_df = df[(df['column1']=='value') & (df['column2']=='value')]

- Step 2: Since we want the lowest value of Female Debit Card Ownership, we can sort the final output using
  df.sort_values('column')

- Step 3: To get the bottom N countries we can get the first 5 rows of the above-sorted dataset using
  df.head(5)

Report Answer 5.2:

```
filter_df = gender_df [ (gender_df['indicator'] == 'Debit card ownership, female (% age 15+)')  &
(gender_df['year'] == 2017) ]

sorted_df = filter_df.sort_values('value')

sorted_df.head(5)
```

Report Output 5.2:

The countries with the lowest percentage of females (age 15+) with debit card ownership are: South Sudan, Afghanistan, Chad, Sierra Leone, and Liberia

Report Question 6: Understanding Other Indicators

Report Question 6.1: Level: Easy

How has the Human Capital Index for females in India changed from 2017 to 2020

- Indicator Names: Human Capital Index (HCI), Female

Method:

- Step 1: To filter the dataset for an indicator in a country across years, we can use
  filter_df = df[(df['column1'] == ‘value1’) &
  (df['column2'] == ‘value1’) &
  (df['column2'].isin([‘value1’,’value2’])]

Report Answer 6.1:

```
filter_df = gender_df [ (gender_df['indicator'] == 'Human Capital Index (HCI), Female') &
                (gender_df['country'] == 'India') &
                (gender_df['year'].isin([2017,2020])) ]

filter_df
```

Report Output 6.1:

The Human Capital Index for females in India has improved from 0.45 in 2017 to 0.5 in 2020

Report Question 6.2: Level: Medium

Which are the best 5 countries based on the Human Capital Index (HCI) for females in 2020

- Indicator Name: Human Capital Index (HCI), Female

Method:

- Step 1: We can filter our dataset for a particular indicator and year using
  filter_df = df[(df['column1']=='value') & (df['column2']=='value')]

- Step 2: Since we want the highest value of HCI, we can sort the final output using
  df.sort_values('column', ascending=False)

- Step 3: To get the top N countries we can get the first 5 rows of the above-sorted dataset using
  df.head(5)

Report Answer 6.2:

```
filter_df = gender_df [ (gender_df['indicator'] == 'Human Capital Index (HCI), Female')  &
(gender_df['year'] == 2020) ]

sorted_df = filter_df.sort_values('value', ascending=False)

sorted_df.head(5)
```

Report Output 6.2:

We see that the best 5 countries in terms of the Human Capital Index for females are Singapore, Hong Kong, Finland, Korea, and Estonia.

Report Question 6.3: Level: Medium

Which are the worst 5 countries based on the Human Capital Index (HCI) for females in 2020

- Indicator Name: Human Capital Index (HCI), Female

Method:

- Step 1: We can filter our dataset for a particular indicator and year using
  filter_df = df[(df['column1']=='value') & (df['column2']=='value')]

- Step 2: Since we want the lowest value of HCI, we can sort the final output using
  df.sort_values('column')

- Step 3: To get the bottom N countries we can get the first 5 rows of the above-sorted dataset using
  df.head(5)

Report Answer 6.3:

```
filter_df = gender_df [ (gender_df['indicator'] == 'Human Capital Index (HCI), Female')  &
(gender_df['year'] == 2020) ]

sorted_df = filter_df.sort_values('value')
sorted_df.head(5)
```

Report Output 6.3:

We see that the worst 5 countries in terms of the Human Capital Index for females are: Chad, Niger, Mali, Liberia, and Angola.

Report Question 6.4: Level: Medium

Find the number of countries each year where a woman cannot get a job the same way a man can

- Indicator Name: A woman can get a job in the same way as a man (1=yes; 0=no)

Method:

- Step 1: We can filter our dataset for a particular indicator using
  filter_df = df[(df['column']=='value')]

- Step 2.1: Since we want the count of countries per year, we can use **groupby** and mention the column name at which you want to group using
  df.groupby('column', as_index=False)

- Step 2.2: Use agg function to count or sum values in another column using
  df.groupby('column', as_index=False).agg(new_column1 = ('column','aggregation function'), new_column2 = ('column', 'aggregation function'))

- Step 3: Create a new column using the above calculated columns using
  df[‘new_column’] = df[‘column1’] - df[‘column2’]

Report Answer 6.4:

```
filter_df = gender_df [ (gender_df['indicator'] == 'A woman can get a job in the same way as a man (1=yes; 0=no)') ]

agg_df = filter_df.groupby('year', as_index=False).agg(countries_where_woman_can_get_job_same_way_as_man = ('value','sum'), total_countries = ('country','count'))

agg_df['countries_where_woman_cannot_get_job_same_way_as_man'] = agg_df['total_countries'] - agg_df['countries_where_woman_can_get_job_same_way_as_man']

agg_df
```

Report Output 6.4:

We can see that the number of countries where a woman cannot get the job in the same way as man has dropped from 23 in 2012 to 17 in 2021

Report Question 6.5: Level: Easy

List all countries where this still happens

- Indicator Name: A woman can open a bank account in the same way as a man (1=yes; 0=no)

Method:

- Step 1: We can filter our dataset for a particular indicator for the most recent year (2020) with a particular value (0) by using
  filter_df = df[(df['column1']=='value') & (df['column2] == 'value') & (df['column3] == 'value')]

Report Answer 6.5:

filter_df = gender_df [(gender_df['indicator'] == 'A woman can open a bank account in the same way as a man (1=yes; 0=no)') & (gender_df['year'] == 2020) & (gender_df['value'] == 0)]

filter_df

Report Output 6.5:

We see that there are still 5 countries where a woman cannot open a bank account in the same way as a man. These are Bhutan, Equatorial Guinea, Guinea-Bissau, Kenya, and Suriname.

→ Congratulations on completing the Python tutorial! You can now prepare the consolidated report.