

# UN Dataset Analysis Program Report

## Purpose

The main purpose of the program designed by group 27 is for the user to obtain official UN statistics regarding fertility and population increase rate for a given UN sub-region and country within this sub-region

## Dataset summary

There were two datasets for the UN statistical data containing various statistical parameters shown below. The third dataset contained an index that categorized the country into UN regions and sub-regions.

Data set 1 contained the following parameters for the years 2001 - 2018 and many various UN regions, sub-regions, and countries:

- Population annual rate of increase (percent)
- Total fertility rate (children per women)
- Life expectancy at birth for both sexes (years)
- Life expectancy at birth for males (years)
- Life expectancy at birth for females (years)

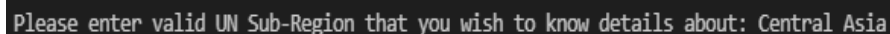
Data set 2 contained the following parameters for the years 2001, 2005, 2010, 2015, 2018 and many various UN regions, sub-regions, and countries:

- Capital city population (as a percentage of total population)
- Capital city population (as a percentage of total urban population)
- Capital city population (thousands)
- Urban population (percent)

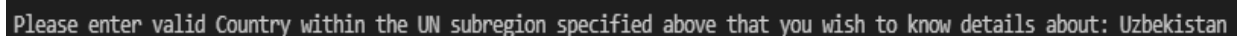
We wanted to focus on the fertility data and population increase as a way to determine when factors other than naturalized birth caused the population to change (immigration, war, brain drain etc.)

## User interface input/output

The user is tasked with inputting the UN sub-region and a country within this region to obtain UN sub-region and country fertility and population increase data. The following screenshots show this:



```
Please enter valid UN Sub-Region that you wish to know details about: Central Asia
```



```
Please enter valid Country within the UN subregion specified above that you wish to know details about: Uzbekistan
```

The user is then given fertility rate and population increase data for UN sub-region and country specified:

The following list shows fertility rates and population increase rate of the countries specified in UN Sub-Region Central Asia in descending order:

Series	Total fertility rate (children per women)	Population annual rate of increase (percent)
Country		
Tajikistan	3.7180	2.322
Turkmenistan	3.0000	1.796
Kazakhstan	2.6733	1.561
Uzbekistan	2.5126	1.625

The following statistics shows fertility and population increase rate for Uzbekistan:

Series	Total fertility rate (children per women)	Population annual rate of increase (percent)
UN Region		
UN Sub-Region		
Country		
Year		
Asia		
Central Asia		
Uzbekistan		
2005	2.5126	1.296
2010	2.4900	1.521
2015	2.4300	1.625

Following this 2 aggregate statistics, namely mean fertility rate for the UN sub-region and country as well as the dataset statistical parameter count, mean value, standard deviation, minimum value, 25 percentile, 50 percentile, 75 percentile, and maximum values are presented:

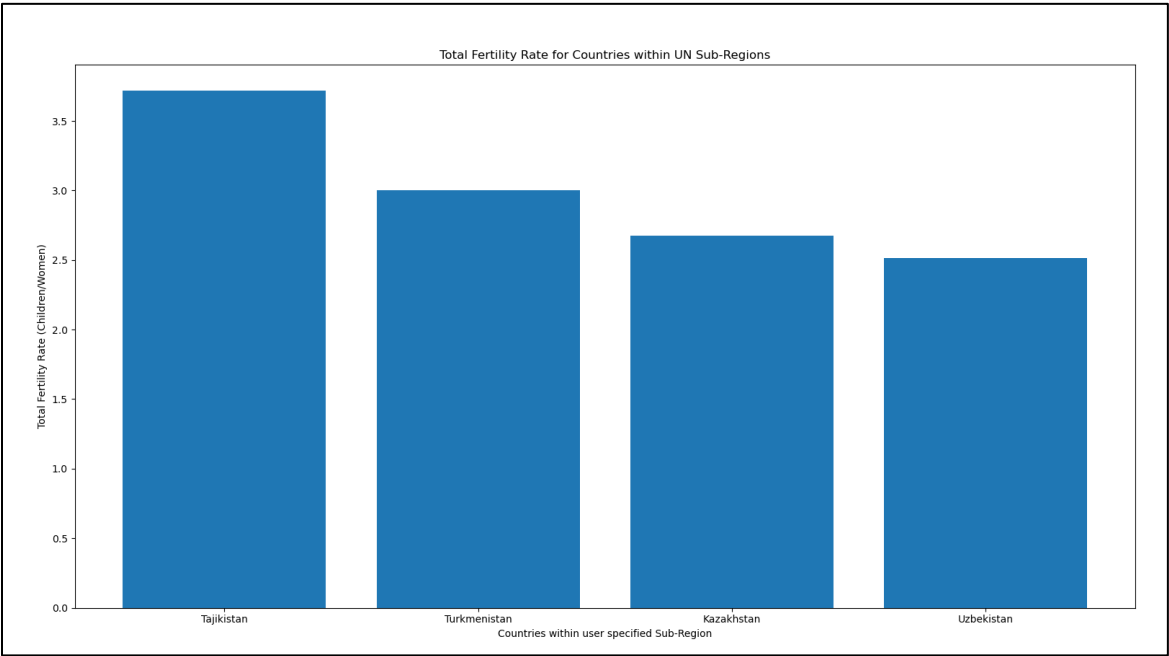
The mean fertility rate for the UN Sub-Region specified is 2.83

The mean fertility rate for the Country specified over 2005, 2010, 2015 and 2018 is 2.48

The following is the basic statistical data for the various series in the UN dataset:

Series	count	mean	std	min	25%	50%	75%	max
Capital city population (as a percentage of tot...	384.0	17.376562	14.136366	0.300000	9.100000	14.450000	22.300000	100.000000
Capital city population (as a percentage of tot...	384.0	30.714323	18.071066	1.100000	17.475000	29.050000	41.100000	100.000000
Capital city population (thousands)	552.0	3052.865942	4999.325862	3.000000	686.750000	1393.000000	2746.000000	37468.000000
Life expectancy at birth for both sexes (years)	485.0	68.741614	9.671050	42.031000	61.906000	71.446000	75.592000	83.321000
Life expectancy at birth for females (years)	492.0	71.197026	10.060906	42.710000	63.857500	74.545000	78.577500	86.470000
Life expectancy at birth for males (years)	492.0	66.446990	9.342212	40.640000	60.062500	68.355000	73.197500	81.719000
Population annual rate of increase (percent)	510.0	1.538302	1.572556	-1.580000	0.546500	1.339000	2.387500	15.263000
Total fertility rate (children per women)	495.0	3.066489	1.584348	1.145500	1.777300	2.540600	4.205000	7.650000
Urban population (percent)	672.0	57.152976	23.728971	9.400000	37.125000	57.400000	77.000000	100.000000
Total Urban Population (thousands)	384.0	22985.793974	70426.035841	526.645768	2533.584584	5465.071793	15688.741722	767541.666667

Finally, the code outputs the following plot to show the fertility rate of the various countries in the subregion in comparison to the user specified sub-region:



## Evidence of Program Criteria Specified

- Import data into data frame from excel
- Dropped column "Code" from merged dataset

```
world_data_uncodes = pd.read_excel("project-p21-group-27/UN Population Datasets/UN Codes.xlsx", # Importing the UN codes that categorize countries
                                   dtype = {'Country': str, 'UN Region': str, 'UN Sub-Region': str})
world_data1 = pd.read_excel("project-p21-group-27/UN Population Datasets/UN Population Dataset 1.xlsx",
                             dtype = {'Code': int, 'Region/Country/Area': str, 'Year': str, 'Series': str, 'Value': float}).drop('Code', axis = 1)
world_data2 = pd.read_excel("project-p21-group-27/UN Population Datasets/UN Population Dataset 2.xlsx",
                             dtype={'Code': int, 'Region/Country/Area': str, 'Year': str, 'Series': str, 'Value': float}).drop('Code', axis = 1)
```

- 2 merging/joining operations
- Hierarchical indexing and sorted
- Creation of pivot table

```
world_data3 = pd.concat([world_data1, world_data2]).sort_values(by=['Region/Country/Area', 'Year', 'Series']) # Concatating the 2 datasets in order to add them together
world_data = pd.merge(world_data_uncodes, world_data3, how='inner', left_on='Country', right_on='Region/Country/Area').drop('Region/Country/Area', axis=1)
world_data = world_data.set_index(['UN Region', 'UN Sub-Region', 'Country', 'Year', 'Series']) # Setting the hierarchical index according the column names
world_data_pivot = pd.pivot_table(world_data, values='Value', index = ['UN Region', 'UN Sub-Region', 'Country', 'Year'], columns='Series') # Making a pivot table
```

- Add 2 columns to merged data set (I added it to the pivot table since it did not make much sense to add additional columns to just the values for the various UN parameters)

```
# Add two new columns

# Calculating total urban population from capital city and capital city population as a percentage of the whole urban population
world_data_pivot['Total Urban Population (thousands)'] = world_data_pivot['Capital city population (thousands)']/(world_data_pivot['Capital city population (as a percentage of total urban population)']/100)

world_data_pivot['Gender with higher life expectancy'] = world_data_pivot['Life expectancy at birth for females (years)'] # Initializing column 'Life expectancy at birth for females (years)'

# Checking to see which gender has a longer life expectancy at birth for a given country for each year in the UN dataset
world_data_pivot['Gender with higher life expectancy'].astype(str) # Setting data type in column as string
idx = world_data_pivot.index[world_data_pivot['Life expectancy at birth for females (years)'].notnull()] # Using masking to find all indices in the 'Life expectancy at birth for females (years)' column that are not null
for i in idx: # Iterating over all the indexes that were found to contain values in the 'Life expectancy at birth for females (years)' column
    if world_data_pivot['Life expectancy at birth for females (years)'][i] > world_data_pivot['Life expectancy at birth for males (years)'][i]:
        world_data_pivot['Gender with higher life expectancy'][i] = 'Female'
    elif world_data_pivot['Life expectancy at birth for females (years)'][i] < world_data_pivot['Life expectancy at birth for males (years)'][i]:
        world_data_pivot['Gender with higher life expectancy'][i] = 'Male'
```

- Ask user to enter two inputs and perform error handling

```
# STAGE 3: User Entry

user_subregion = check_user_subregion(world_data) # Using the check_user_subregion function to do the error handling
sub_region_stats = world_data.loc[world_data.index.get_level_values('UN Sub-Region') == user_subregion] # Using the loc method to filter the data by the user's subregion
user_country = check_user_country(sub_region_stats) # Using the check_user_subregion function to do the error handling
country_stats = world_data.loc[world_data.index.get_level_values('Country') == user_country] # Assigning the country stats to a variable
```

- 2 user described functions to handle errors

```

'''
The check_user_sub-region function checks if the user inputted a sub-region that is contained within the merged UN dataset
'''

def check_user_subregion(data):

    while True:
        user_s = input('\nPlease enter valid UN Sub-Region that you wish to know details about: ')
        try:
            if user_s in data.index.get_level_values(1):
                break
            else:
                raise ValueError
        except ValueError:
            print('\nThe UN Sub-Region is not in the UN Database, please try again.')
    return user_s

'''
The check_user_country function checks if the user inputted country is contained within the UN subregion specified by the user earlier
'''

def check_user_country(data):

    while True:
        user_c = input('\nPlease enter valid Country within the UN subregion specified above that you wish to know details about: ')
        try:
            if user_c in data.index.get_level_values(2):
                break
            else:
                raise ValueError
        except ValueError:
            print('\nThe Country entered is not in the UN Sub-Region Specified, please try again.')
    return user_c

```

- Masking operation to find the indices for the user specified UN sub-region
- Create and print pivot plot for the fertility and population increase data

```

# Shows fertility rate (children/women) of the user specified UN Sub-Region
print('\n')
print('The following list shows fertility rates and population increase rate of the countries specified in UN Sub-Region ' + user_subregion + ' in descending order: ')
print('\n')

# This command assigns @parameter: fertility_rank to a ranked dataframe of all countries within the user specified UN sub-region by order of fertility of all years.
# It does this with a masking operation which return a boolean array of indices where the user specified UN Region is true and sorts in descending order base on the 'Total fertility rate (children per women)' column in the data
fertility_rank = world_data_pivot.loc[world_data_pivot.index.get_level_values('UN Sub-Region') == user_subregion].sort_values(by='Total fertility rate (children per women)', ascending=False)

#This find the maximum fertility and population increase rate values of each country regardless of year. This is done by grouping at year level (level=2) and applying the .apply(max).sort() methods to the data
fertility_max = fertility_rank[['Total fertility rate (children per women)', 'Population annual rate of increase (percent)']].groupby(level=2).apply(max).sort_values(by='Total fertility rate (children per women)', ascending=False)
print(fertility_max)

print('\n\n')

```

- Groupby function and aggregation function to calculate fertility mean

```

# Calculating the overall mean fertility data for the user specified UN subregion
world_subregion_mean_all = world_data_pivot.groupby(level='UN Sub-Region').mean()
world_subregion_mean_fertility = world_subregion_mean_all.loc[user_subregion, 'Total fertility rate (children per women)']
print('The mean fertility rate for the UN Sub-Region specified is ' + str(round(world_subregion_mean_fertility, 2)))
print('\n')

# Calculating the overall mean fertility data for the user specified country
country_mean_all = world_data_pivot.groupby(level='Country').mean()
country_mean_fertility = country_mean_all.loc[user_country, 'Total fertility rate (children per women)']
print('The mean fertility rate for the Country specified over 2005, 2010, 2015 and 2018 is ' + str(round(country_mean_fertility, 2)))

```

- Describe method to print many aggregate statical data from the dataset

```

# This command outputs a pivot table with a list of common statistical parameters using the describe
world_data_describe = world_data_pivot.describe()
print('The following is the basic statistical data for the various series in the UN dataset: \n')
print(world_data_describe.transpose()) # The table is transposed for easier comprehension of data

```

- Export file to working directory and make matplotlib bar graph

```
# Exporting pivot table with all merged UN data to the file "group27_output_export"
world_data_pivot.to_excel('project-p21-group-27/group27_output_export.xlsx')

# Converting countries within the user specified sub-region from indices to list and setting as the x-axis values of bar graph
# Converting fertility rates for the countries within the user specified sub-region from a dataframe to float list
plt.bar(fertility_max['Total fertility rate (children per women)'].keys().tolist(), fertility_max['Total fertility rate (children per women)'].reset_index()['Total fertility rate (children per women)'].values)
plt.xlabel('Countries within user specified Sub-Region') # Adding plot X-axis label
plt.ylabel('Total Fertility Rate (Children/Women)') # Adding plot Y-axis label
plt.title('Total Fertility Rate for Countries within UN Sub-Regions') # Adding plot title
plt.show()
```

- Exported excel screenshot

UN Region	UN Sub-Region	Country	Year	Capital city population (as a percentage of total population)	Capital city population (as a percentage of total urban population)	Capital city population (thousands)	Life expectancy at birth for both sexes (years)	Life expectancy at birth for females (years)	Life expectancy at birth for males (years)	Population annual rate of increase (percent)
Africa	Northern Africa	Algeria	2005	6.9	10.7	2282	71.827	73.24	70.47	1.314
			2010	6.7	10	2432	74.169	75.43	72.95	1.637
			2015	6.5	9.2	2592	75.513	76.71	74.36	1.983
			2018			2694				
		Egypt	2005	19.8	45.9	15174	68.984	71.41	66.65	1.856
			2010	20.1	46.7	16899	69.867	72.21	67.62	1.53
			2015	20.1	46.9	18820	70.834	73.05	68.71	2.213
			2018			20076				
		Libya	2005	18.3	23.7	1058	71.259	73.25	69.55	1.581
			2010	17.8	22.7	1095	72.335	74.86	70.14	1.331
			2015	18.2	22.9	1134	71.736	74.93	68.87	0.7
			2018			1158				
		Morocco	2005	5.4	9.7	1635	69.972	71.39	68.49	1.122
			2010	5.3	9.1	1714	73.413	74.7	72.06	1.203
			2015	5.2	8.5	1796	75.029	76.3	73.7	1.386
			2018			1847				
		Sudan	2005	12.9	39.3	3979	59.46	61.47	57.54	2.528
			2010	13.1	39.7	4517	61.624	63.5	59.81	2.198
			2015	13.3	39.2	5128	63.757	65.48	62.07	2.376
			2018			5534				
		Tunisia	2005	18.4	28.2	1899	73.722	76.3	71.44	0.804
			2010	18.9	28.4	2014	74.634	77.19	72.3	1.019
			2015	19.4	28.5	2183	75.452	77.59	73.4	0.999
			2018			2291				
	Sub-Saharan Africa	Angola	2005	19.8	35.4	3872	48.05	50.37	45.9	3.4
			2010	22.7	37.9	5300	52.686	55.26	50.29	3.677
			2015	25.2	39.7	7023	57.74	60.51	55.15	3.544

## IEEE Citation for Dataset Used

### UN Dataset 1:

*Population Growth, Fertility and Mortality Indicators*, United Nations Statistics Division, June 2021.

[Online] Available: <http://data.un.org/>

### Dataset 2:

*Population Growth Rates in Urban areas and Capital cities*, United Nations Statistics Division, June 2021.

[Online] Available: <http://data.un.org/>