Data Visualization - Role of Females towards Economic Development

- Comparative analysis of the years 2000 and 2014
- Visualization based on data from World Bank

Reported by:
Anuparna Banerjee
UT EID: AB59958

Contents

1. Description

The project presents a comparative analysis of the role of females towards economy. Females in many countries do not play an active role in the economic development due to several reasons. However, with every passing year, the trend is improving with the spread of education and employment opportunities. This project compares data available for 2 years, 2000 and 2014 to understand the trend change and to figure out the countries which still need to work in this direction.

2. Workflow

The project utilizes data in the format of *comma-separated values (CSV)* from certain data sources and uses *Python* libraries to parse them. The parsed data is then inserted into a *MySQL database* and all the visualizations the project reports are based on reading the data from the database.

2.1. Tools Used

PhpMyAdmin - PHP Web tool to work with MySQL database
Python 3 interpreter
Python libraries - csv, pymysql, matplotlib, pandas

2.2. File Locations and Database Details

Data-source:

- /export/home/u16/anuparna/project/prototype/data-source/employment_to_population_ratio.csv
- /export/home/u16/anuparna/project/prototype/data-source/gender_sector_employment.csv
- /export/home/u16/anuparna/project/prototype/data-source/country_year_working_age_population.csv

Database: banerjee_a_world_economy_database

Python Script to Insert data into database:

- /export/home/u16/anuparna/project/prototype/insertDataInDatabase.py

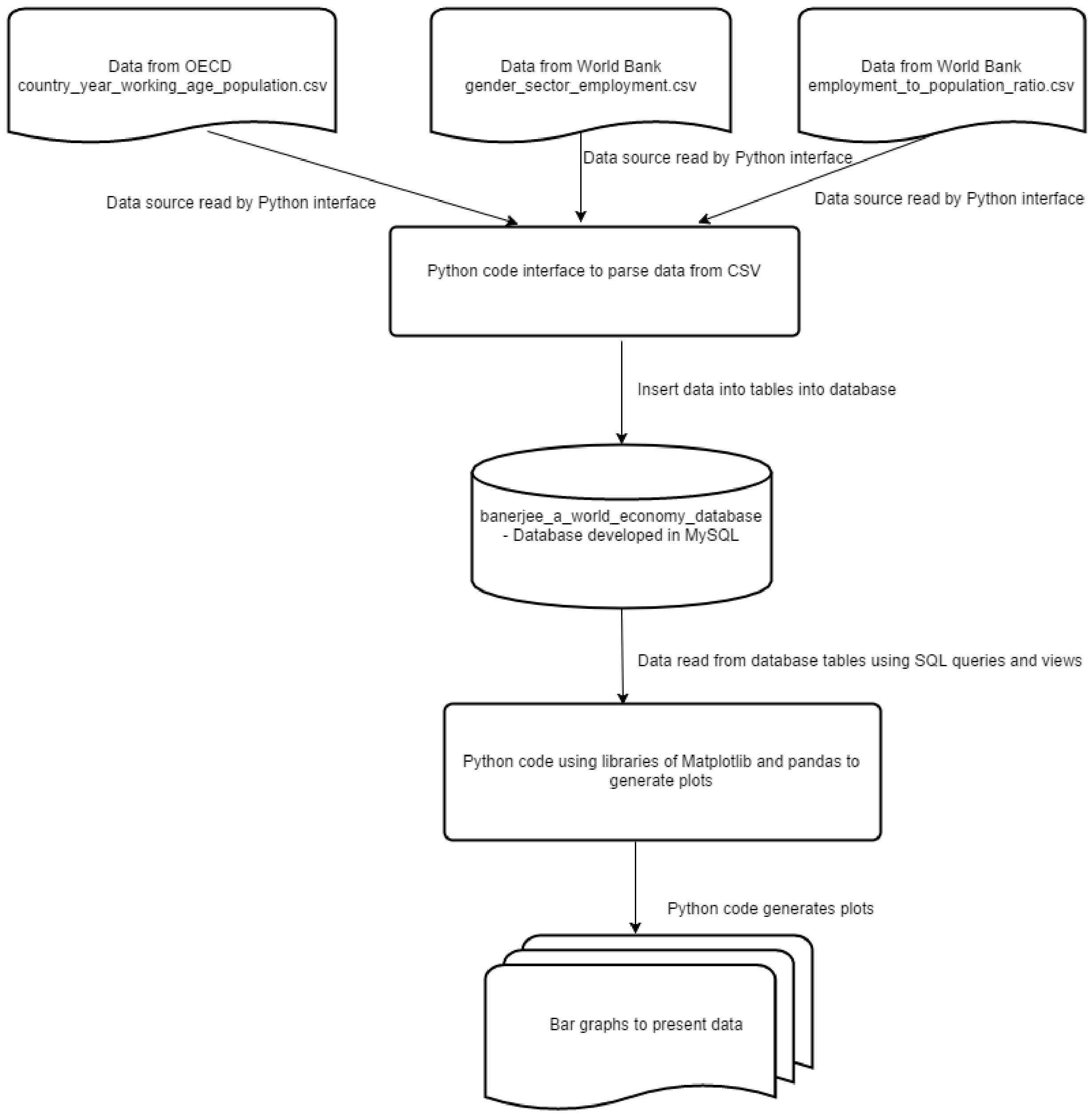
Data Analysis

- /export/home/u16/anuparna/project/prototype/plot1.py
- /export/home/u16/anuparna/project/prototype/plot2.py
- /export/home/u16/anuparna/project/prototype/plot3.py

Data Analysis Plots:

- /export/home/u16/anuparna/project/prototype/figures/sector_vs_females/*.png
- /export/home/u16/anuparna/project/prototype/figures/working_population_vs_females/*.png

2.3. Workflow View



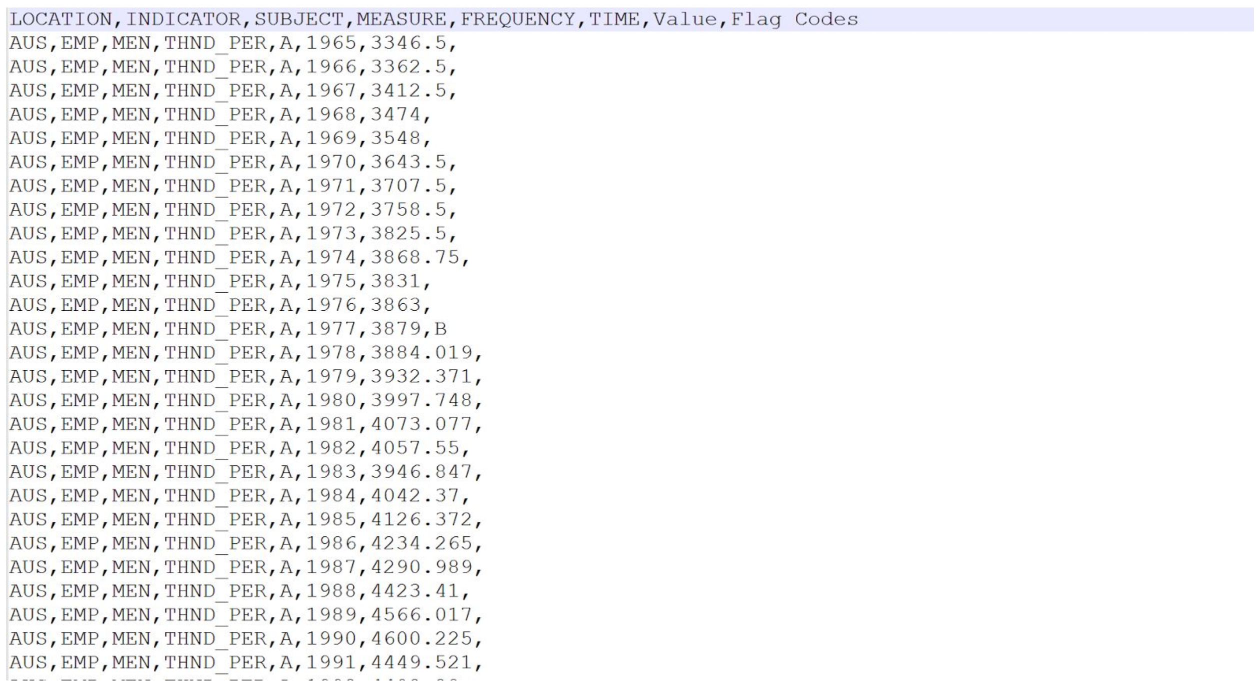
Data from OECD country_year_working_age_population.csv
Data from World Bank gender_sector_employment.csv
Data from World Bank employment_to_population_ratio.csv
Data source read by Python interface
Data source read by Python interface
Data source read by Python interface
Python code interface to parse data from CSV
Insert data into tables into database
banerjee_a_world_economy_database - Database developed in MySQL
Data read from database tables using SQL queries and views
Python code using libraries of Matplotlib and pandas to generate plots
Python code generates plots
Bar graphs to present data

3. Data Sources

The project utilizes data extracted from 2 different data sources. The data is available in the format of Comma-Separated Values (CSV).

3.1. The Organisation for Economic Co-operation and Development (OECD)

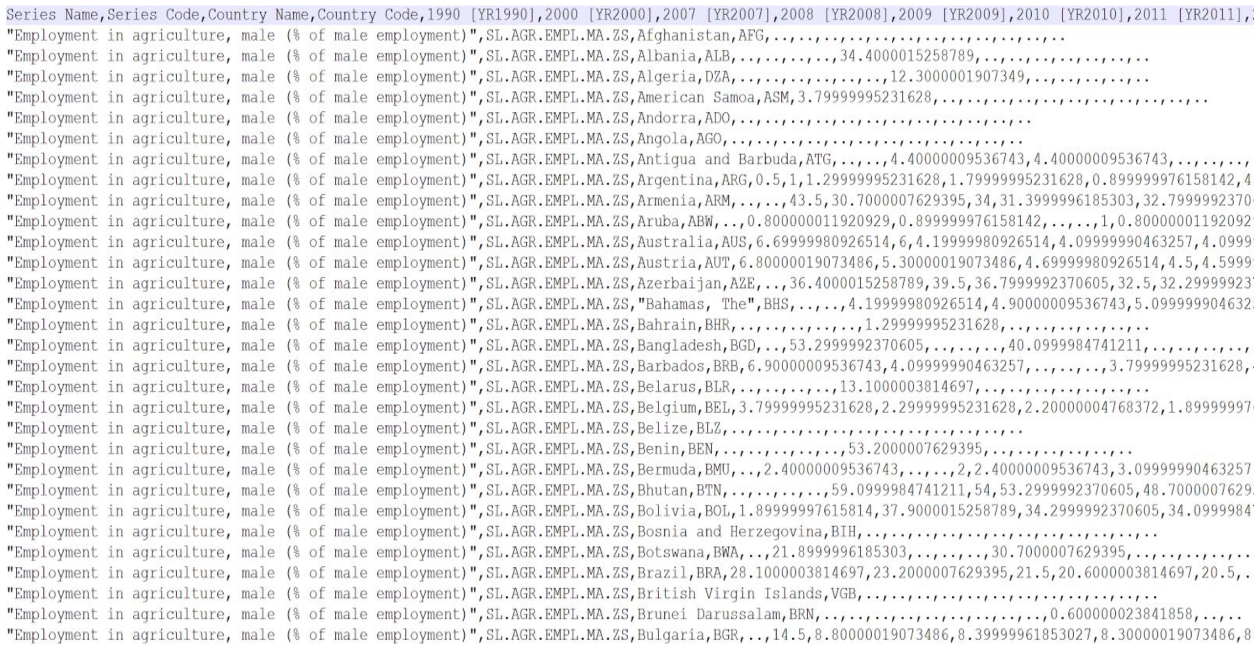
The Organisation for Economic Co-operation and Development (OECD) collects data across various verticals to improve the economic well-being of people across the world. It provides the data for employment rates i.e. the ratio of the employed to the working-age population at https://data.oecd.org/emp/employment-rate.htm#indicator-chart. This data can be downloaded in the form of CSV.



```
LOCATION,INDICATOR,SUBJECT,MEASURE,FREQUENCY,TIME,Value,Flag Codes
AUS,EMP,MEN,THND_PER,A,1965,3346.5,
AUS,EMP,MEN,THND_PER,A,1966,3362.5,
AUS,EMP,MEN,THND_PER,A,1967,3412.5,
AUS,EMP,MEN,THND_PER,A,1968,3474,
AUS,EMP,MEN,THND_PER,A,1969,3548,
AUS,EMP,MEN,THND_PER,A,1970,3643.5,
AUS,EMP,MEN,THND_PER,A,1971,3707.5,
AUS,EMP,MEN,THND_PER,A,1972,3758.5,
AUS,EMP,MEN,THND_PER,A,1973,3825.5,
AUS,EMP,MEN,THND_PER,A,1974,3868.75,
AUS,EMP,MEN,THND_PER,A,1975,3831,
AUS,EMP,MEN,THND_PER,A,1976,3863,
AUS,EMP,MEN,THND_PER,A,1977,3879,B
AUS,EMP,MEN,THND_PER,A,1978,3884.019,
AUS,EMP,MEN,THND_PER,A,1979,3932.371,
AUS,EMP,MEN,THND_PER,A,1980,3997.748,
AUS,EMP,MEN,THND_PER,A,1981,4073.077,
AUS,EMP,MEN,THND_PER,A,1982,4057.55,
AUS,EMP,MEN,THND_PER,A,1983,3946.847,
AUS,EMP,MEN,THND_PER,A,1984,4042.37,
AUS,EMP,MEN,THND_PER,A,1985,4126.372,
AUS,EMP,MEN,THND_PER,A,1986,4234.265,
AUS,EMP,MEN,THND_PER,A,1987,4290.989,
AUS,EMP,MEN,THND_PER,A,1988,4423.41,
AUS,EMP,MEN,THND_PER,A,1989,4566.017,
AUS,EMP,MEN,THND_PER,A,1990,4600.225,
AUS,EMP,MEN,THND_PER,A,1991,4449.521,
```

3.2. World Development Indicators (WDI) from World Bank

The World Bank provides World Development Indicators (WDI) to estimate the state of economy of differnt countries on the basis of economic parameters like *Employment by Sector* (http://wdi.worldbank.org/table/2.3) and *Decent Work and Productive employment* (http://wdi.worldbank.org/table/2.4). This project collects data from the World Bank data links in the form of CSV.



```
Series Name,Series Code,Country Name,Country Code,1990 [YR1990],2000 [YR2000],2007 [YR2007],2008 [YR2008],2009 [YR2009],2010 [YR2010],2011 [YR2011],
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Afghanistan,AFG,..,..,..,..,..,..,..,..,..,..,..,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Albania,ALB,..,..,..,..,34.4000015258789,..,..,..,..,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Algeria,DZA,..,..,..,..,..,..,12.3000001907349,..,..,..,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,American Samoa,ASM,3.79999995231628,..,..,..,..,..,..,..,..,..,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Andorra,ADO,..,..,..,..,..,..,..,..,..,..,..,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Angola,AGO,..,..,..,..,..,..,..,..,..,..,..,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Antigua and Barbuda,ATG,..,..,4.40000009536743,4.40000009536743,..,..,..,
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Argentina,ARG,0.5,1,1.29999995231628,1.79999995231628,0.899999976158142,4
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Armenia,ARM,..,..,43.5,30.7000007629395,34,31.3999996185303,32.7999992370
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Aruba,ABW,..,0.800000011920929,0.899999976158142,..,..,1,0.80000001192092
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Australia,AUS,6.69999980926514,6,4.19999980926514,4.09999990463257,4.0999
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Austria,AUT,6.80000019073486,5.30000019073486,4.69999980926514,4.5,4.5999
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Azerbaijan,AZE,..,36.4000015258789,39.5,36.7999992370605,32.5,32.2999992
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,"Bahamas, The",BHS,..,..,4.19999980926514,4.90000009536743,5.09999990463
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Bahrain,BHR,..,..,..,..,..,1.29999995231628,..,..,..,..,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Bangladesh,BGD,..,53.2999992370605,..,..,..,40.0999984741211,..,..,..,..,
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Barbados,BRB,6.90000009536743,4.09999990463257,..,..,..,3.79999995231628,
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Belarus,BLR,..,..,..,..,13.1000003814697,..,..,..,..,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Belgium,BEL,3.79999995231628,2.29999995231628,2.20000004768372,1.8999999
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Belize,BLZ,..,..,..,..,..,..,..,..,..,..,..,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Benin,BEN,..,..,..,..,..,53.2000007629395,..,..,..,..,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Bermuda,BMU,..,2.40000009536743,..,..,2,2.40000009536743,3.09999990463257
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Bhutan,BTN,..,..,..,..,59.0999984741211,54,53.2999992370605,48.700000762
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Bolivia,BOL,1.89999997615814,37.9000015258789,34.2999992370605,34.099998
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Bosnia and Herzegovina,BIH,..,..,..,..,..,..,..,..,..,..,..,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Botswana,BWA,..,21.8999996185303,..,..,..,30.7000007629395,..,..,..,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Brazil,BRA,28.1000003814697,23.2000007629395,21.5,20.6000003814697,20.5,.
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,British Virgin Islands,VGB,..,..,..,..,..,..,..,..,..,..,..,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Brunei Darussalam,BRN,..,..,..,..,..,..,..,..,..,0.600000023841858,..,..
"Employment in agriculture, male (% of male employment)",SL.AGR.EMPL.MA.ZS,Bulgaria,BGR,..,14.5,8.80000019073486,8.39999961853027,8.30000019073486,8
```

4. Database Design

The database for the project is developed on MySQL database using the PhpMyAdmin Web interface. The scope of the project requires data for 2 years, 2000 and 2014.

4.1. Field definitions

percentage_employment_to_population_ratio – It is the ratio of the employed to the country's population irrespective of their working-age (obtained from World Bank).

percentage_working_age_population – It is the ratio of the employed to the working-age population (obtained from OECD). Separate ratios are available for both genders.

percentage_sector_employment - It represents the ratio of people employed under a sector to the population of working-age (obtained from World bank)

gender – The data obtained from World bank consists of data for Males and Females. The database design of this project refers them as

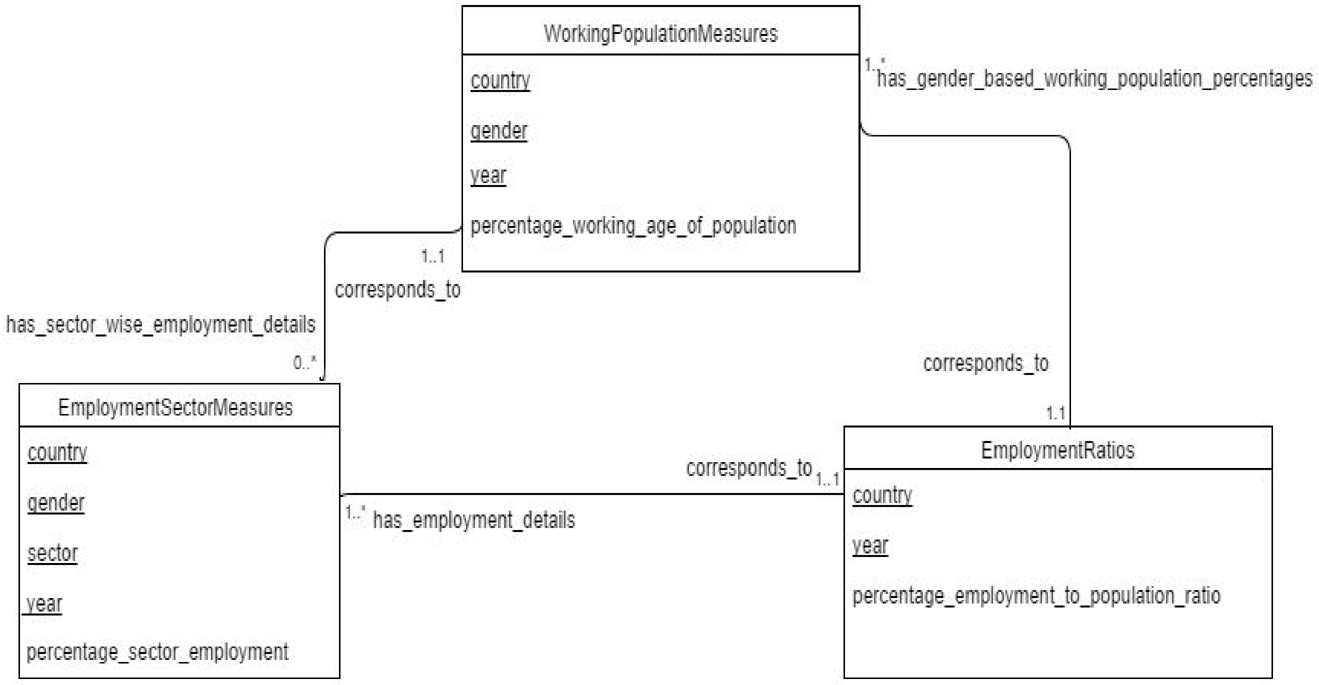
- Male (MA)
- Female (FE)

sector – There are 3 different sectors of employment referred in the World Bank data. These sectors of employment are represented by the following notation in the database.

- Services (SRV)
- Agriculture (AGR)
- Industry (IND)

country – All countries are represented by 3-character abbreviation.

4.2. Entity - Relationship Diagram


WorkingPopulationMeasures
country
gender
year
percentage_working_age_of_population
1..* has_gender_based_working_population_percentages
1..1
corresponds_to
has_sector_wise_employment_details
0..*
corresponds_to
1.1
EmploymentSectorMeasures
country
gender
sector
year
percentage_sector_employment
corresponds_to 1..1
1..* has_employment_details
EmploymentRatios
country
year
percentage_employment_to_population_ratio

4.3. Relationship Vocabulary

WorkingPopulationMeasures :has_many EmploymentSectorMeasures
EmploymentSectorMeasures :belongs_to WorkingPopulationMeasures

EmploymentSectorMeasures :belongs_to EmploymentRatios
EmploymentRatios :has_many EmploymentSectorMeasures

WorkingPopulationMeasures :belongs_to EmploymentRatios
EmploymentRatios :has_many WorkingPopulationMeasures

4.4. Sample Tables

employment_ratios

id	country	year	percentage_employment_to_population_ratio
257	FSM	2014	33.70
258	FSM	2000	33.90

employment_sector_measures

id	country	gender	sector	year	percentage_sector_employment
1	FSM	MA	SRV	2000	19.40
2	FSM	FE	SRV	2000	54.10

working_population_measures

id	country	year	gender	percentage_working_age_population
1	FSM	2014	MA	76.92
2	FSM	2000	FE	77.09

4.5. Database Table creation scripts

```
CREATE TABLE employment_ratios (
 'id' int(11) NOT NULL AUTO_INCREMENT,
 'country' text NOT NULL,
 'year' int(11) NOT NULL,
 'percentage_employment_to_population_ratio' decimal(10,2) NOT NULL,
 PRIMARY KEY ('id')
)

CREATE TABLE employment_sector_measures (
 'id' int(11) NOT NULL AUTO_INCREMENT,
 'country' text NOT NULL,
 'gender' varchar(2) NOT NULL,
 'sector' varchar(3) NOT NULL,
 'year' int(11) NOT NULL,
 'percentage_sector_employment' decimal(10,2) NOT NULL,
 PRIMARY KEY ('id')
)

CREATE TABLE working_population_measures (
 'id' int(11) NOT NULL AUTO_INCREMENT,
 'year' int(11) NOT NULL,
 'country' text NOT NULL,
 'gender' varchar(2) NOT NULL,
 'percentage_working_age_of_population' decimal(10,2) NOT NULL,
 PRIMARY KEY ('id')
)
```

4.6. Challenges

- The *country* data could be moved in a different table which could be utilized as a *lookup* between the other tables. This would result in one table with one column but the primary key of this table would still have to be placed as a foreign key in the other tables. This cannot be considered as a very efficient design since fetching data from the tables would require an extra join condition.
- Placing all the data in one single table required a lot of parsing and implementing join conditions within the Python code which is not an advisable strategy.

5. Python Interface

A Python script was written to import the data from the CSVs. As mentioned in the previous section, the earlier version of the script performed most of the join conditions within the Python code to insert all the relevant data in one table. However, the new version of code was developed after a revisited database design. The Python libraries of *csv* and *pymysql* was used to develop this interface.

5.1. Code

```python
import csv
import pymysql

# defined a map so that CSVs from different sources represent the same gender
gender_map={"MEN":"MA","WOMEN":"FE"}

# defined a map so that CSVs from different sources represent the same year
year_map={"2000":"2000 [YR2000]","2014":"2014 [YR2014]"}

#collect only percentages of working-age-population for the years 2000 and 2014 for
only men and women and not the total
country_year_working_age_population_reader =
open('data-source/country_year_working_age_population.csv')
country_year_working_age_population =
csv.DictReader(country_year_working_age_population_reader)
pc_working_population_rows=[]
for row in country_year_working_age_population:
    if row['MEASURE'] == 'PC_WKGPOP' and (row['TIME'] in year_map.keys()) and
row['SUBJECT'] != 'TOT':
        pc_working_population_rows.append(row)

working_population_rows=[]
for row in pc_working_population_rows:
    data={}
    country=row['LOCATION'] #fetch the country
    gender=row['SUBJECT'] #fetch the gender

    #Fill the dict to insert into database
    data["year"]=row['TIME']
    data["country"]=country
    data["gender"]=gender_map[gender]
    data["percentage_working_age_of_population"]=row['Value']
    working_population_rows.append(data)
```

```
#Collect those rows which correspond the country and the gender from the 3rd CSV
employment_ratios_reader = open('data-source/employment_to_population_ratio.csv')
employment_ratios = csv.DictReader(employment_ratios_reader)
employment_ratio_rows=[]
for employment_ratio in employment_ratios:
    if "SL.EMP.TOTL." in employment_ratio['Series Code']:
        for year,year_format in year_map.items():
            data={}
            data["country"]=employment_ratio['Country Code']
            data["year"]=year
            data["percentage_employment_to_population_ratio"]=
employment_ratio[year_format] if employment_ratio[year_format]!='..' else 0.00
            employment_ratio_rows.append(data)


# fetch those rows which correspond the country and the gender from the 2nd CSV
gender_sector_employment_reader = open('data-source/gender_sector_employment.csv')
gender_sector_employment = csv.DictReader(gender_sector_employment_reader)
employment_sector_rows=[]
for employment_detail in gender_sector_employment:
    code=employment_detail['Series Code']
    if code:
        split_string = code.split(".") #code is in the form of 'SL.SRV.EMPL.FE.ZS'
where FE denotes gender and SRV denotes sector
        for year,year_format in year_map.items():
            data={}
            data["country"]=employment_detail['Country Code']
            data["gender"]=split_string[3]
            data["sector"]=split_string[1]
            data["year"]=year
            data["percentage_sector_employment"]=employment_detail[year_format] if
employment_detail[year_format]!='..' else 0.00
            employment_sector_rows.append(data)

#Database operations start here
connection = pymysql.connect(host="localhost",
                             user="banerjee_a",
                             passwd="ab59958",
                             db="banerjee_a_world_economy_database",
                             autocommit=True,
                             cursorclass=pymysql.cursors.DictCursor)
cursor = connection.cursor()
# insert into database with the values in the list
sql_working_population='''
                    INSERT INTO working_population_measures
                    (country,gender,year,percentage_working_age_of_population)
                    VALUES

(%(country)s,%(gender)s,%(year)s,%(percentage_working_age_of_population)s)
                    '''
generated_working_population = [] #contains all the ids generated during the
execution of the insert statement
for data in working_population_rows:
    cursor.execute(sql_working_population, data)
    generated_working_population.append(cursor.lastrowid)
```

```
sql_employment_ratio='''
                    INSERT INTO employment_ratios
                    (country,year,percentage_employment_to_population_ratio)
                    VALUES

(%(country)s,%(year)s,%(percentage_employment_to_population_ratio)s)
                    '''
generated_employment_ratio_ids = [] #contains all the ids generated during the
execution of the insert statement
for data in employment_ratio_rows:
    cursor.execute(sql_employment_ratio, data)
    generated_employment_ratio_ids.append(cursor.lastrowid)

sql_employment_sector='''
                    INSERT INTO employment_sector_measures
                    (country,gender,sector,year,percentage_sector_employment)
                    VALUES

(%(country)s,%(gender)s,%(sector)s,%(year)s,%(percentage_sector_employment)s)
                    '''
generated_employment_sector_ids = [] #contains all the ids generated during the
execution of the insert statement
for data in employment_sector_rows:
    cursor.execute(sql_employment_sector, data)
    generated_employment_sector_ids.append(cursor.lastrowid)

cursor.close()
```

5.2. Challenges

- Ease of use vs readability: The *List comprehension* utility of Python provides a very easy way to collect appropriate data required for further use. This utility provides an interesting way of using conditional statements with List comprehension but this code affect the readability of the code. One of the code snippets used in the earlier version of the code was:

```
employment_to_population_ratios = [ employment_ratio for employment_ratio in
Employment_ratios if employment_ratio [ 'Country Code' ]== country and "SL.EMP.TOTL."
in employment_ratio [ 'Series Code' ]]
```

This code is not only difficult to read but also hard to maintain on the long run. The new version of the code tries to overcome this issue by using *for* loops.

6. Data Analysis

This project utilizes the Python libraries for Matplotlib and Pandas to generate bar plots to view the data. The appropriate data is fetched from a database view.

6.1. Database View

Database Views can hide complex SQL join queries and provide customized results with the desired format.
The View created for this project: *female_employment_vs_population* with the below definition:

```
SELECT wpm.country AS country, wpm.year as year,
         wpm.percentage_working_age_of_population AS
percentage_working_age_of_population,
         er.percentage_employment_to_population_ratio AS
percentage_employment_to_population_ratio
   FROM working_population_measures wpm, employment_ratios er
   WHERE wpm.country = er.country AND
         wpm.gender =  'FE' AND
         wpm.year = er.year
```

The query executed to generate the plot for the year 2000 from the python code is as follows:

```
SELECT f.country AS country,
         f.percentage_working_age_of_population AS
percentage_working_age_of_population,
         f.percentage_employment_to_population_ratio AS
percentage_employment_to_population_ratio
   FROM female_employment_vs_population f   WHERE f.year =2000
```

6.2. Python Matplotlib and Pandas Libraries

- Matplotlib library produces 2D plots and is used from Python scripts. There are many parameters which can be tweaked in order to produce bar plots, scatter plots, histograms, error-charts etc.
- Pandas is an open-source Python library used for data-analysis. The library provides advanced methods to parse CSVs as well as any kind of data source. The data populated using Pandas is helpful in creating plots with Matplotlib.

6.3. Code to generate plots

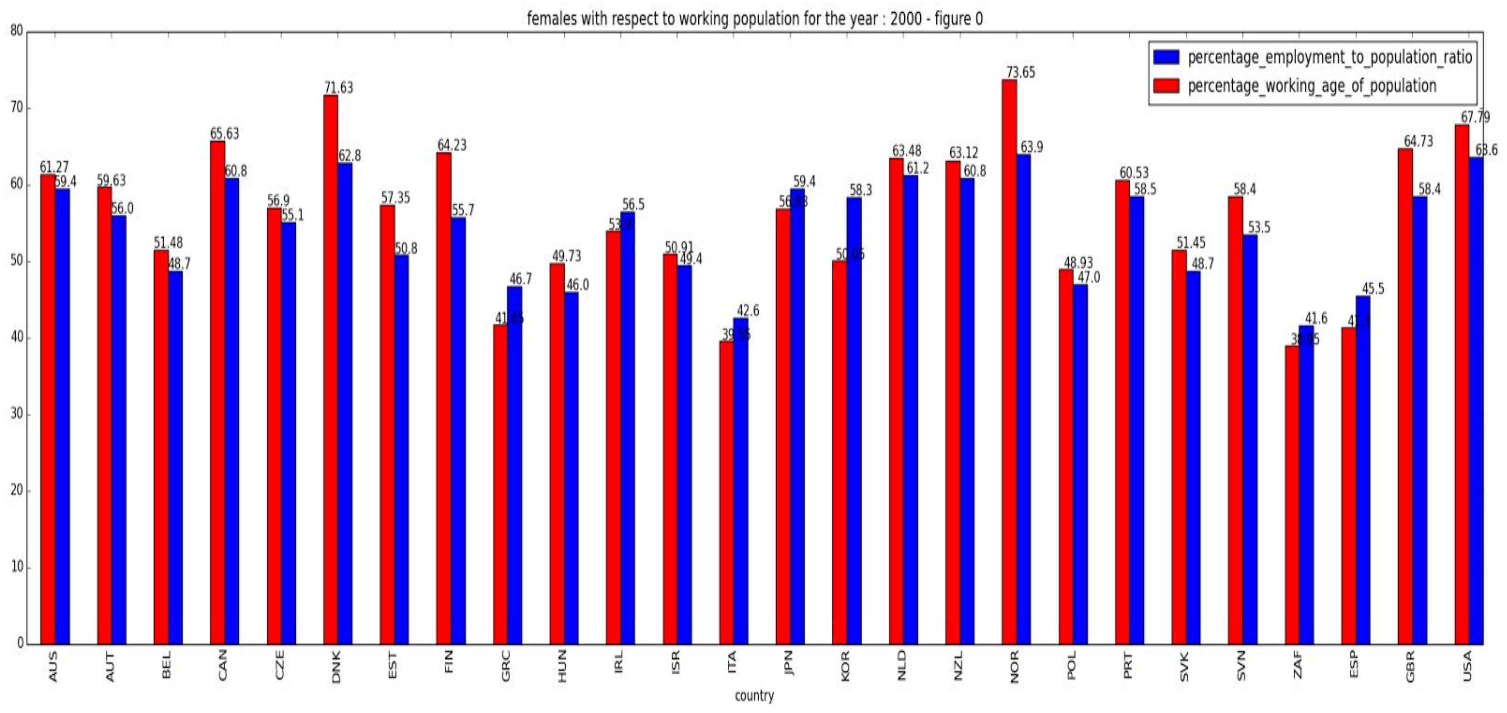
```
import pandas as pd
import matplotlib
matplotlib.use('Agg')

import matplotlib.pyplot as plt
from pylab import rcParams
import pymysql
import pprint

# Param to update the width,height of the figure produced
rcParams['figure.figsize'] = 30, 8

"""
Yield successive n-sized chunks from l.
@Param : l - list
         n - size of chunk
@Return: list splits

"""
def chunks(l, n):
    for i in xrange(0, len(l), n):
        yield l[i:i + n]



#Open connection to PhpMyAdmin
connection = pymysql.connect(host="localhost",
                             user="banerjee_a",
                             passwd="ab59958",
                             db="banerjee_a_world_economy_database",
                             autocommit=True,
                             cursorclass=pymysql.cursors.DictCursor)
cursor = connection.cursor()

#collect employment ratios for the year 2000
sql_employment_ratios='''
```

```
   SELECT f.country AS country,
              f.percentage_working_age_of_population AS
percentage_working_age_of_population,
              f.percentage_employment_to_population_ratio AS
percentage_employment_to_population_ratio
   FROM female_employment_vs_population f    WHERE f.year =2000
'''
cursor.execute(sql_employment_ratios)
result=cursor.fetchall()

# accumulate results to map to one key
# country : 'ABC','DEF',...
resultset={}
for row in result:
    for column_header,column_value in row.items():
        resultset.setdefault(column_header,[]).append(column_value)

# split list into chunks of 40 countries to make plots presentable
ratio_split = (list(chunks(resultset['percentage_employment_to_population_ratio'],
40)))
country_split = (list(chunks(resultset['country'], 40)))

# plot results
for row_num in xrange(len(ratio_split)):
    data = {'females with respect to working population for the year : 2000 - figure
'+str(row_num) : pd.Series(ratio_split[row_num], index=country_split[row_num])}
    df = pd.DataFrame(resultset)
    df = df.set_index(['country'])
    df=df.astype(float)
    fig, axes = plt.subplots(nrows=1, ncols=1)


    df['percentage_employment_to_population_ratio'].plot(kind='bar',
ax=axes,legend=True, title='females with respect to working population for the year :
2000 - figure '+str(row_num) , position=0,width=0.25)
    df['percentage_working_age_of_population'].plot(kind='bar',
color='red',legend=True, ax=axes, title='females with respect to working population
for the year : 2000 - figure '+str(row_num) , position=1,width=0.25)
    #axes.legend( (rects1[0], rects2[0], rects3[0]), ('y', 'z', 'k') )
    for p in axes.patches:
       axes.annotate(str(p.get_height()), (p.get_x() * 1.005, p.get_height() *
1.005))
#Save the generated plot at location : figures/working_population_vs_females/
plt.savefig('figures/working_population_vs_females/working_population_2000_'+str(row_
num)+'.png', bbox_inches='tight')
```

6.4. Plots and Analysis

6.4.1. Female employment for the year 2000

- *Source Code Location :* /export/home/u16/anuparna/project/prototype/plot1.py
- *File Location :* /export/home/u16/anuparna/project/prototype/figures/working_population_vs_females/working_population_2000_0.png


females with respect to working population for the year : 2000 - figure 0
percentage_employment_to_population_ratio
percentage_working_age_of_population
country
0
10
20
30
40
50
60
70
80
AUS 61.27 59.4
AUT 59.63 56.0
BEL 51.48 48.7
CAN 65.63 60.8
CZE 56.9 55.1
DNK 71.63 62.8
EST 57.35 50.8
FIN 64.23 55.7
GRC 46.7
HUN 49.73 46.0
IRL 56.5
ISR 50.91 49.4
ITA 42.6
JPN 59.4
KOR 58.3
NLD 63.48 61.2
NZL 63.12 60.8
NOR 73.65 63.9
POL 48.93 47.0
PRT 60.53 58.5
SVK 51.45 48.7
SVN 58.4 53.5
ZAF 41.6
ESP 45.5
GBR 64.73 58.4
USA 67.79 63.6

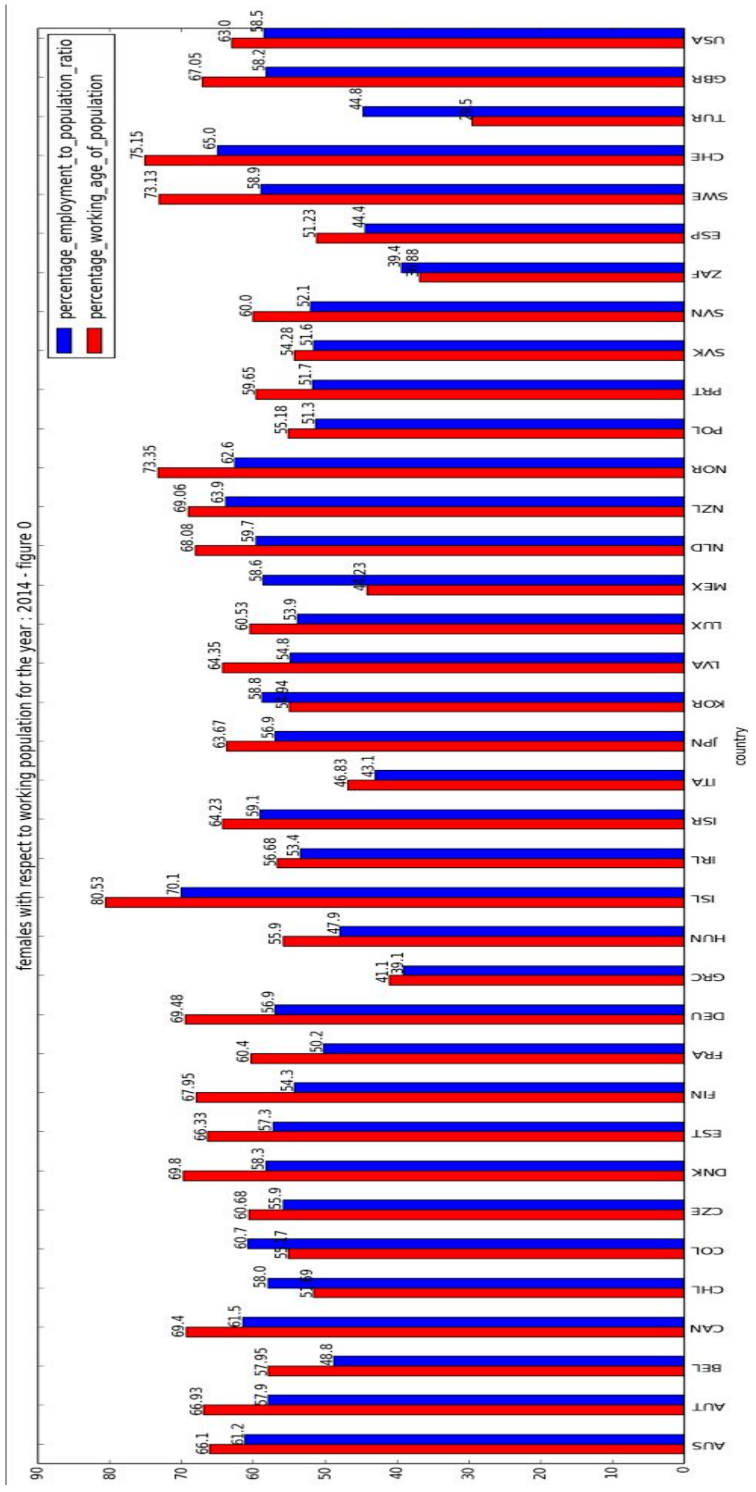
The above plot represents the data for females for the year 2000. Countries like Greece (abbreviated by GRC) show that 41.65% of the working-age population are females while only 46.7% of the population is employed. Although the total proportion of the population employed is quite low compared to countries like USA (63.9%), but the female contribution is reasonably high.

6.4.2. Female employment for the year 2014

- *Source Code Location :* /export/home/u16/anuparna/project/prototype/plot2.py
- *File Location :* /export/home/u16/anuparna/project/prototype/figures/working_population_vs_females/working_population_2014_0.png

The bar-plot below represents the data for female employment for the year 2014. Considering the example for Greece (GRC),

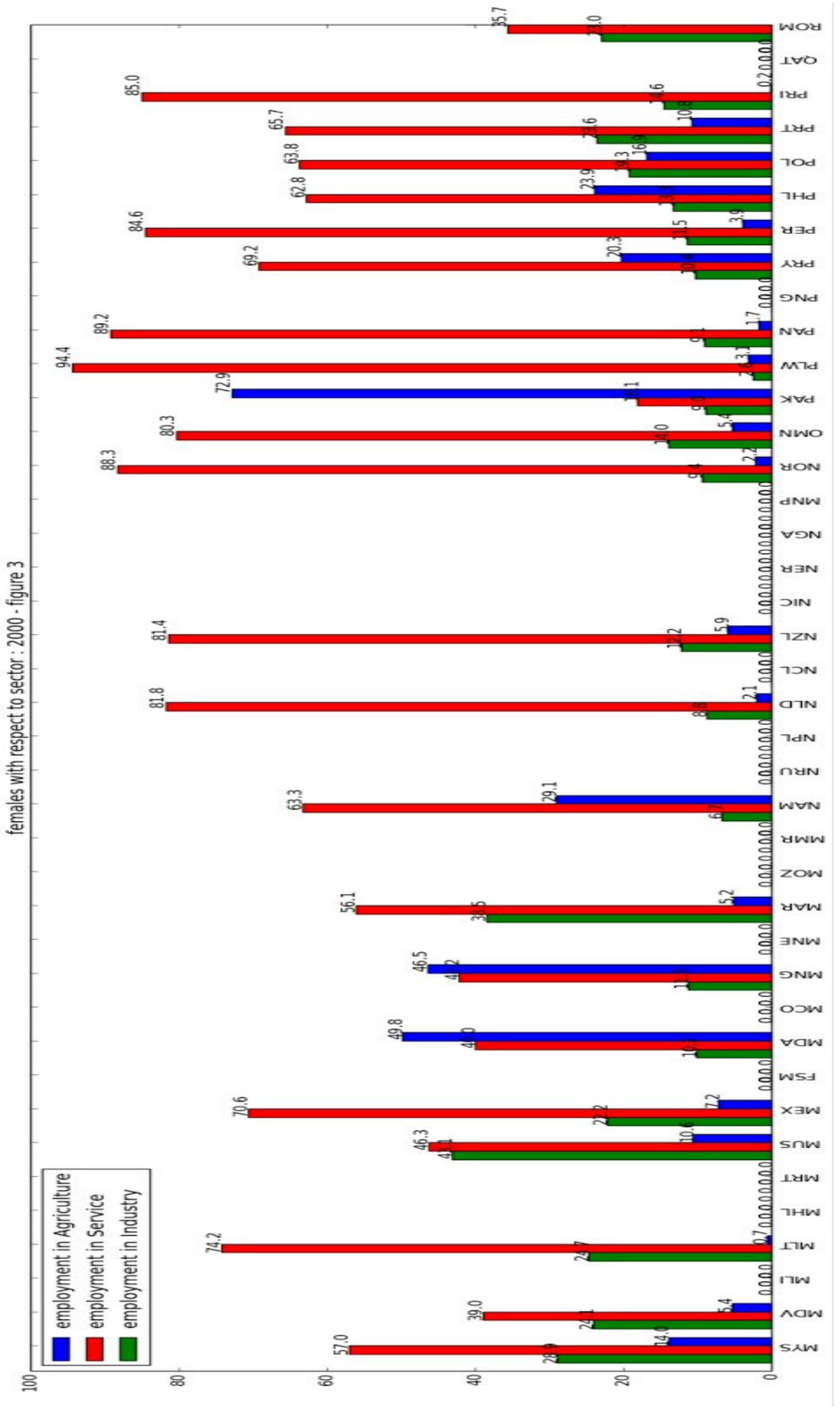
- the percentage of population employed has reduced from 46.7% in the year 2000 to 39.1% in the year 2014. This signifies that the employment opportunities have not developed with the increase in population from 2000 to 2014.
- There is not a significant difference between female employment from 2000 (41.65%) to 2014 (41.1%).
- The data is available for more countries in the year 2014 than that of the year 2000.


females with respect to working population for the year : 2014 - figure 0
percentage_employment_to_population_ratio
percentage_working_age_of_population
country
0
10
20
30
40
50
60
70
80
90
AUS 61.2 66.1
AUT 57.9 66.93
BEL 48.8 57.95
CAN 61.5 69.4
CHL 58.0 51.89
COL 60.7 55.7
CZE 55.9 60.68
DNK 58.3 69.8
EST 57.3 66.33
FIN 54.3 67.95
FRA 50.2 60.4
DEU 56.9 69.48
GRC 39.1 41.1
HUN 47.9 55.9
ISL 70.1 80.53
IRL 53.4 56.68
ISR 59.1 64.23
ITA 43.1 46.83
JPN 56.9 63.67
KOR 58.8 55.94
LVA 54.8 64.35
LUX 53.9 60.53
MEX 58.6 44.23
NLD 59.7 68.08
NZL 63.9 69.06
NOR 62.6 73.35
POL 51.3 55.18
PRT 51.7 59.65
SVK 51.6 54.28
SVN 52.1 60.0
ZAF 39.4 36.88
ESP 44.4 51.23
SWE 58.9 73.13
CHE 65.0 75.15
TUR 44.8 29.5
GBR 58.2 67.05
USA 58.5 63.0

6.4.3. Female employment comparison in various sectors for the year 2000

- *Source Code Location :* /export/home/u16/anuparna/project/prototype/plot3.py
- *File Location :* /export/home/u16/anuparna/project/prototype/figures/sector_vs_females/sector_2000_3.png

The below plot provides a pictorial representation of females engaged in 3 major employment sectors - *Agriculture*, *Services* and *Industry* during the year 2000. As evident from the graph, most of countries displayed a trend that females are involved in the Service sector more than in the other two occupational sectors.


females with respect to sector : 2000 - figure 3
employment in Agriculture
employment in Service
employment in Industry
0
20
40
60
80
100
MYS
MDV
MLI
MLT
MHL
MRT
MUS
MEX
FSM
MDA
MCO
MNG
MNE
MAR
MOZ
MMR
NAM
NRU
NPL
NLD
NCL
NZL
NIC
NER
NGA
MNP
NOR
OMN
PAK
PLW
PAN
PNG
PRY
PER
PHL
POL
PRT
PRI
QAT
ROM

6.5. Learning Experience

Many data analysis tools which we come across during development of data-oriented projects involve error-plotting which utilize the Matplotlib library. The library provides a mechanism to generate different plots and this project provides an introduction to such methods.
Pandas is another library which are used to import data and create quick visualizations. Learning these tools is the next step of Data Wrangling.

The project analysis was based on different online tutorials

- http://pandas.pydata.org/pandas-docs/stable/10min.html
- http://matplotlib.org/1.4.1/examples/api/barchart_demo.html

The major inspiration was drawn from one of the online source:
https://datasciencelab.wordpress.com/2013/12/21/beautiful-plots-with-pandas-and-matplotlib/
which describes how Pandas integrated with Matplotlib produces beautiful plots.

7. Future Work

- Currently, the project does not provide the full names of countries with their associated abbreviations. The part of the problem is that the data from different sources use different abbreviations for the country. All the abbreviations could be added into a single table with an associated long-name and the primary key of this table could be utilized as a lookup.
- The code for data-analysis could be further improved to provide stacked bar-graphs with year comparisons. Due to large number of countries with sector wise data, the graph was splitted into groups of 40 and the plots were saved separately. A better analysis tool is preferred to visualize data associated with countries at one go.