CITS3401 Data Warehousing

Mid-project submission

Student – GAURAV CHAKRAVERTY (22750993)

Date – April 24, 2020

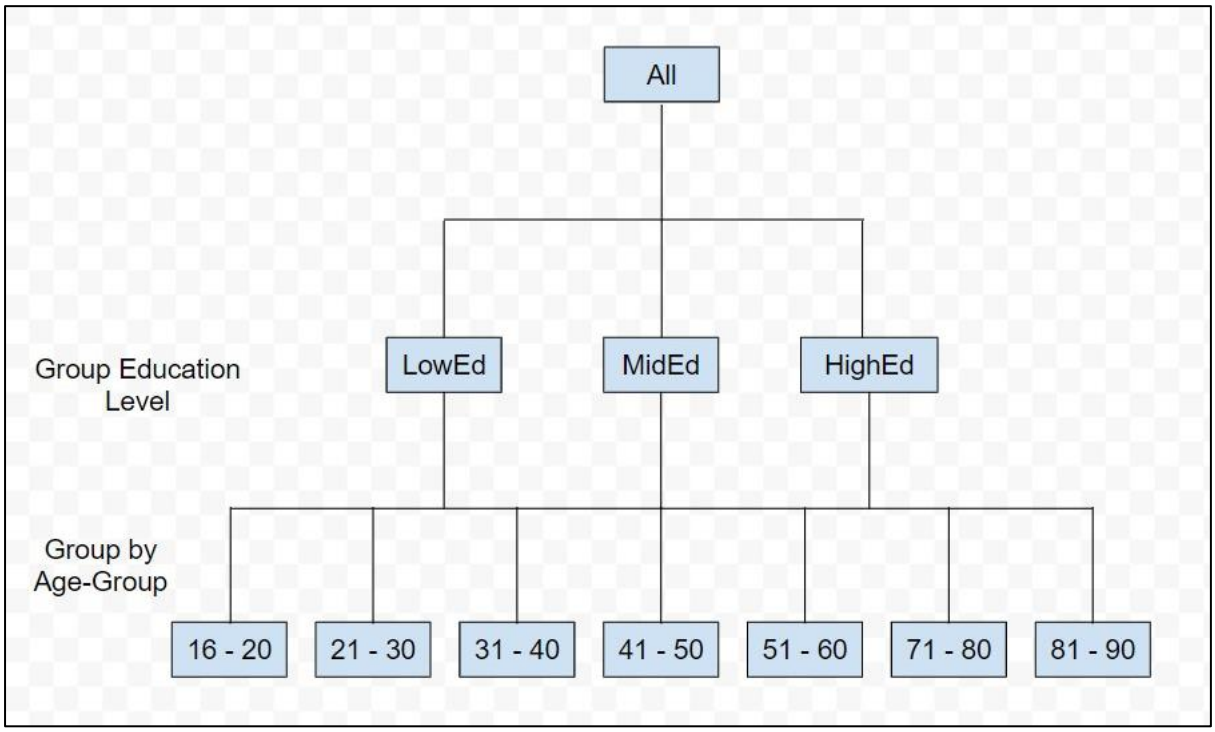
Index

Business Questions –

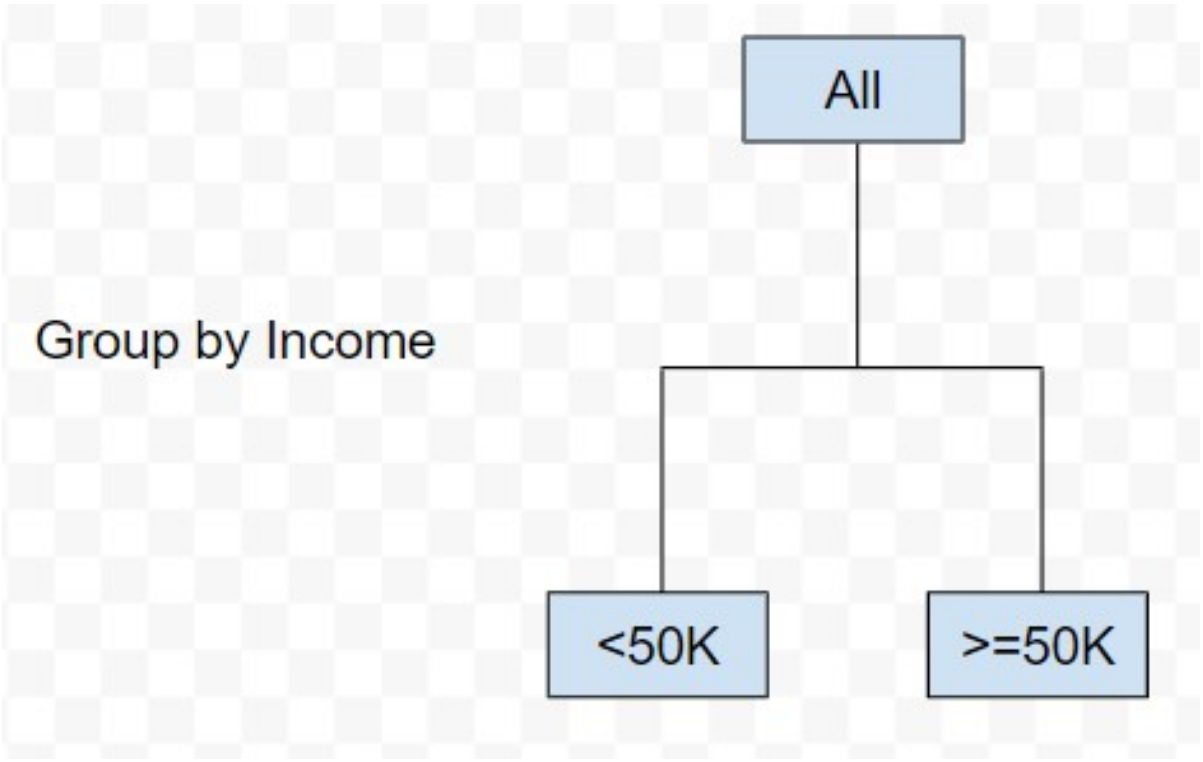
- How many people earn more than $50K across various occupations?
- Do these people mostly work in the public or private sector?
- Are there roughly equal number of males and females who earn more than $50K? If not - which gender tends to earn more?
- How is the income across various age groups and which age group earns the most money?
- What is the education level at with which people are earning the most money?

Concept Hierarchies

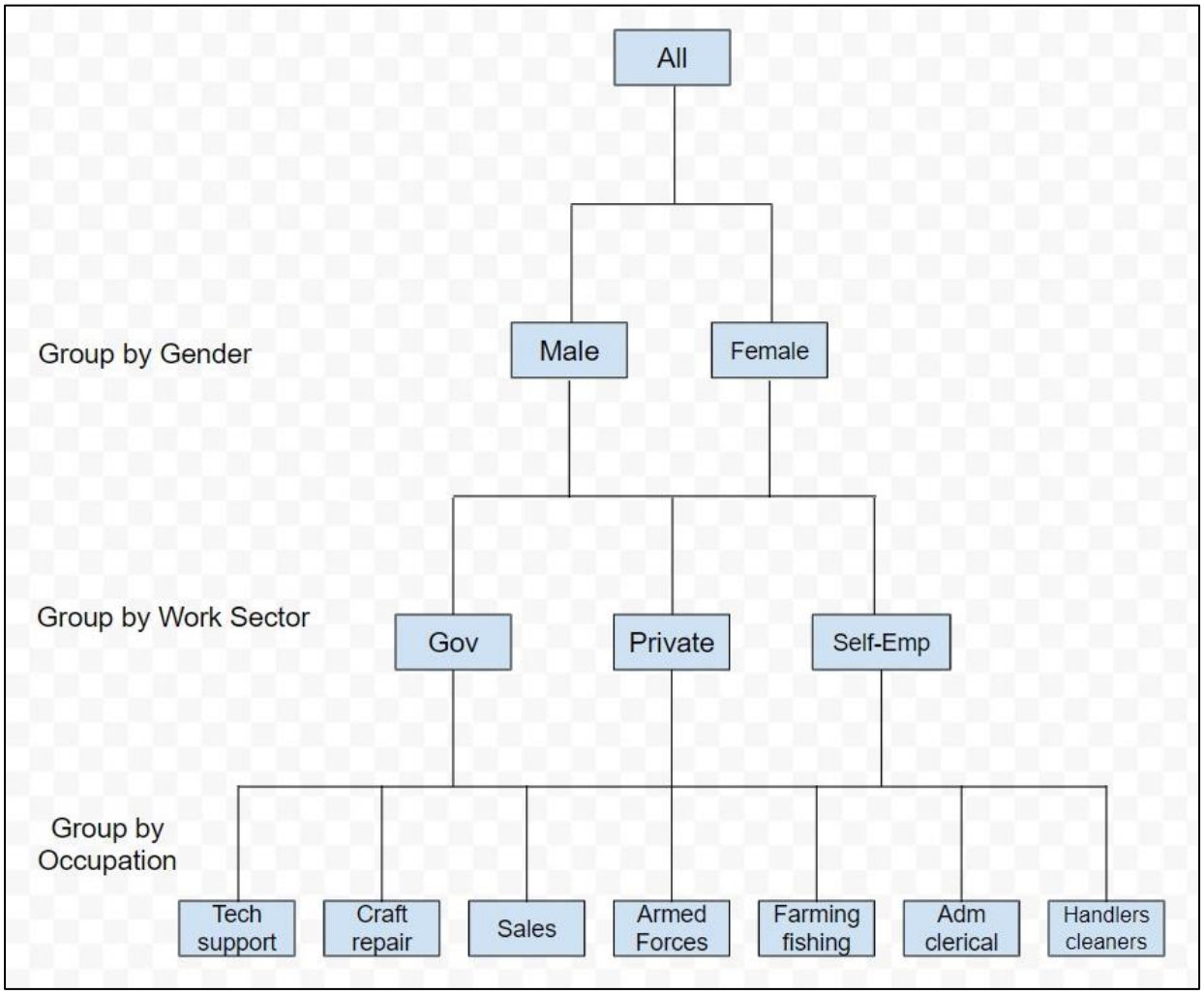
Education



Income

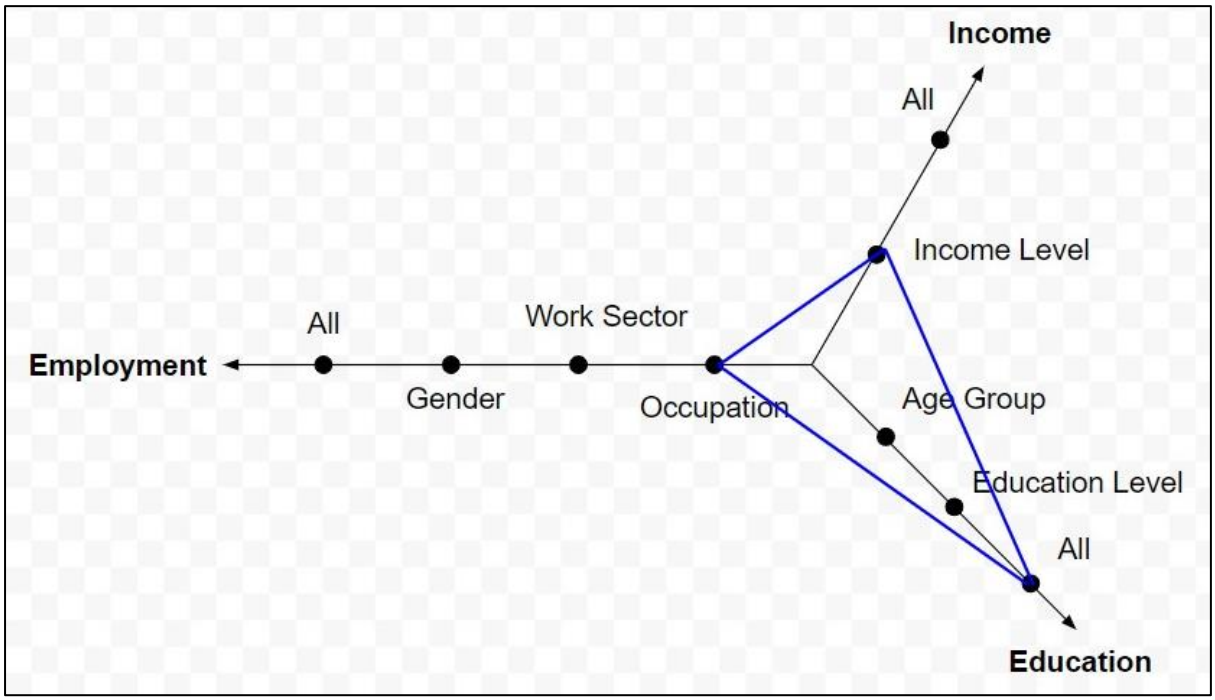


Employment

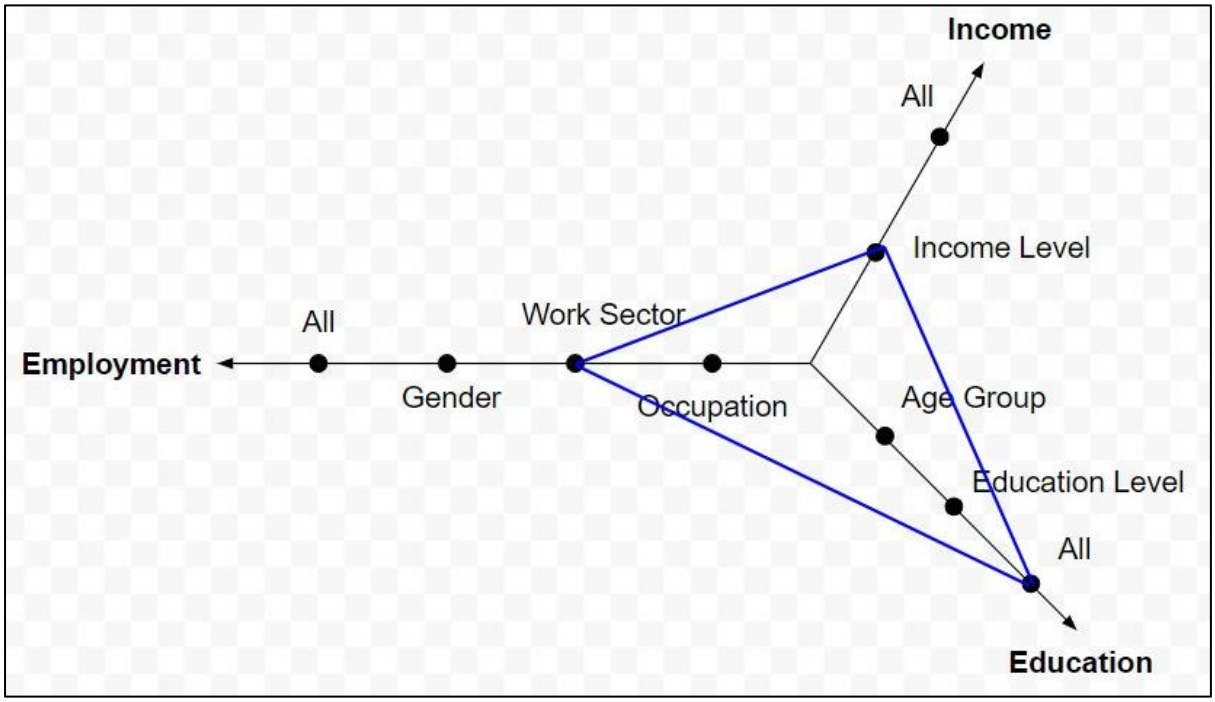

All
Group by Gender
Male
Female
Group by Work Sector
Gov
Private
Self-Emp
Group by Occupation
Tech support
Craft repair
Sales
Armed Forces
Farming fishing
Adm clerical
Handlers cleaners

StarNet Footprints

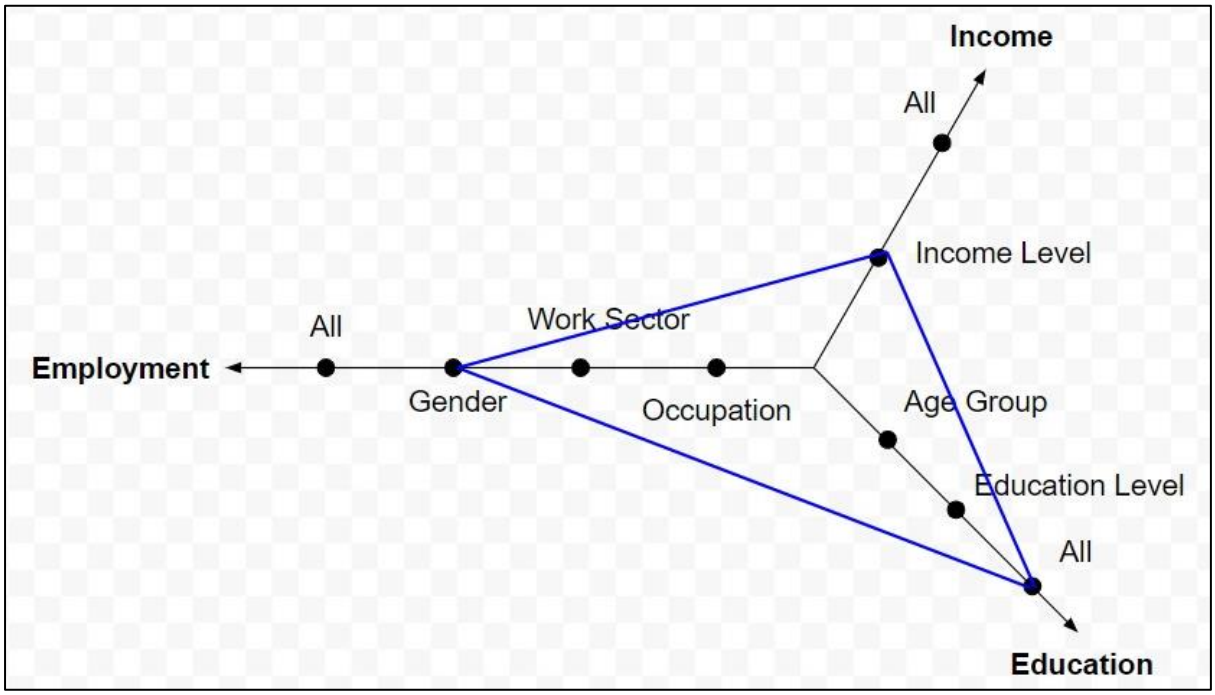
(How many people earn more than $50K across various occupations?)



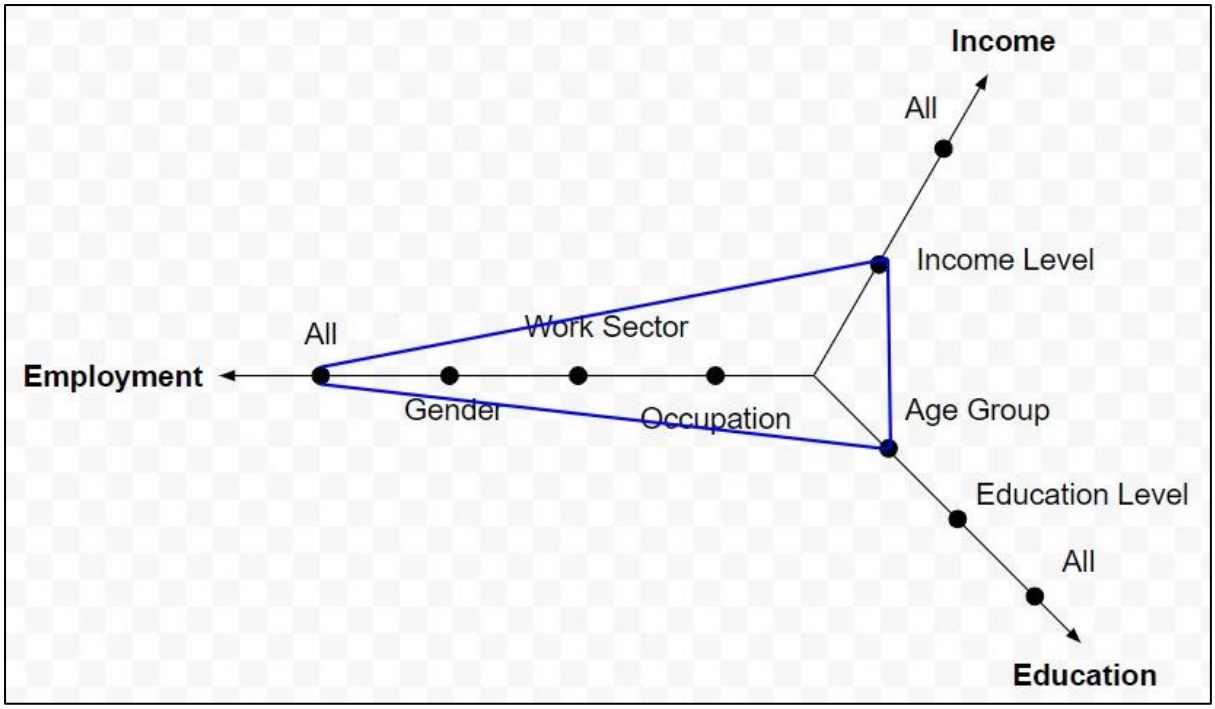
(Do these people mostly work in the public or private sector?)



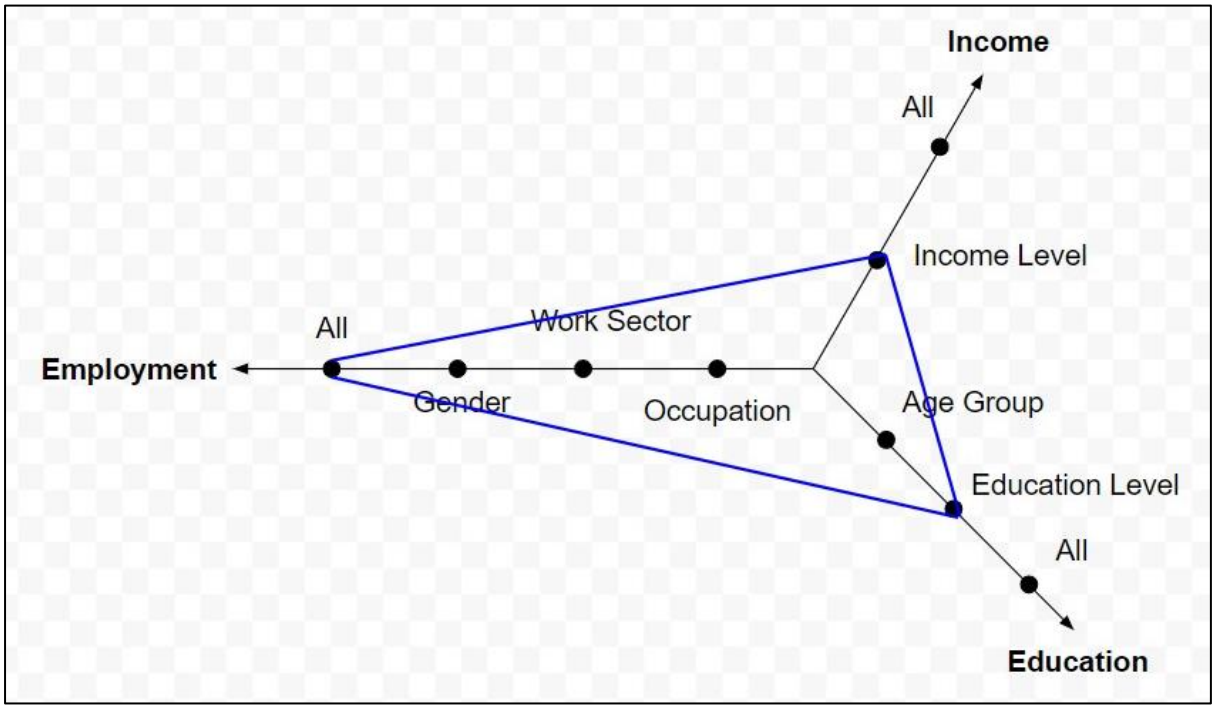
(Are there roughly equal number of males and females who earn more than $50K? If not - which gender tends to earn more?)



(How is the income across various age groups and which age group earns the most money?)



(What is the education level at with which people are earning the most money?)



Data Cleaning and Transformation

- I first downloaded the suggested "adult-training.csv" dataset from LMS.
- After that I noticed that the columns had no headers so after a bit of research I managed to find out the meaning and the possible values for each column. (reference - http://cseweb.ucsd.edu/classes/sp15/cse190-c/reports/sp15/048.pdf)
- When I first opened the provided dataset/csv file to examine it I noticed that it had a lot of "?" values. Also, there were a lot of columns that I did not need. I would also have to create separate csv/data files to later import the required information into my fact and dimension tables.
- I decided to use python (with numpy and pandas library) along with Jupiter Notebook to remove all the rows with "?" values.

```
import numpy as np
import pandas as pd
from numpy import nan
from pandas import read_csv

df=pd.read_csv('adult-training.csv',header=None, sep=',\s', na_values=["?"])
print(df.shape)
```

```
(32561, 15)
```

Had to include this as the csv file had space before the commas

```
df.replace('?', np.NaN, inplace=True)
df=df.dropna(axis='rows')
print(df.shape)
```

```
(30162, 15)
```

Managed to remove the rows with "?"/missing values.

```
df.to_csv("adult-trainingcleaned.csv", header=None);
```

- I was initially facing some problems while trying to read the csv file in python. After a bit of research online I found out that I had to use sep=',\s' because there is a space after each comma in the csv file. (reference - https://stackoverflow.com/questions/29247712/pandas-how-to-replace-with-nan-handling-non-standard-missing-values)
- I also added an index column at the start which I will later change all the values to "" – so the structure of my file looks similar to the files used in Lab4.
- I then proceeded to group and divide the data into smaller tables so it would be easy to import the values later into my database.

```
dfc=pd.read_csv('adult-trainingcleaned.csv',header=None, sep=',')
```

My cleaned file no longer had any space after the commas

```
print(dfc[[0]])
```

```
dfc[0] = ""
demp = dfc[[0,7,2,10]]
```

I set the first column as blank values and selected required columns.

- The above screenshot shows how I stored the columns corresponding to occupation, workclass and gender as required by my Employment dimension later. The first column is blank.

```
for x in range(0,(len(demp[2]))):
    if (demp[2][x] == 'Federal-gov') or (demp[2][x] == 'Local-gov') or (demp[2][x] == 'State-gov'):
        demp[2][x] = 'Gov-Sector'
    elif (demp[2][x] == 'Self-emp-not-inc') or (demp[2][x] == 'Self-emp-inc') or (demp[2][x] == 'Without-pay') or (demp[2][x] ==
'Never-worked'):
        demp[2][x] = 'Self-Emp/No-Pay'
    elif (demp[2][x] == 'Private'):
        demp[2][x] = 'Prv-Sector'
```

- I then wrote a small program to group the items in the workclass column by either Gov-Sector, Prv-Sector or Self-Emp/No-Pay. I saved the output as a new csv file.
- This is how the DimEmployeeFinal csv file output looks –

File Edit Format View Help

```
,Adm-clerical,Gov-Sector,Male
,Exec-managerial,Self-Emp/No-Pay,Male
,Handlers-cleaners,Prv-Sector,Male
,Handlers-cleaners,Prv-Sector,Male
,Prof-specialty,Prv-Sector,Female
,Exec-managerial,Prv-Sector,Female
```

- I followed a similar process to create the other three tables – DimEmploymentFinal, DimIncomeFinal, DimEducationFinal and IncomeFactFinal. (complete python code is in the Appendix at the end of this report.)
- Note that in DimEducationFinal – I have grouped the age column according to different age ranges and the education Level according to Low(HighSchool and below), Medium(ex.Bachelors Degree) and High(Masters, Phd etc).

```
dfc[0] = ""
dedu = dfc[[0,1,5]]
```

```
for x in range(0,(len(dedu[1]))):
    if (17<=dedu[1][x]<=20):
        dedu[1][x] = '17-20'
    elif (21<=dedu[1][x]<=30):
        dedu[1][x] = '21-30'
    elif (31<=dedu[1][x]<=40):
        dedu[1][x] = '31-40'
    elif (41<=dedu[1][x]<=50):
        dedu[1][x] = '41-50'
    elif (51<=dedu[1][x]<=60):
        dedu[1][x] = '51-60'
    elif (61<=dedu[1][x]<=70):
        dedu[1][x] = '61-70'
    elif (71<=dedu[1][x]<=80):
        dedu[1][x] = '71-80'
    elif (81<=dedu[1][x]<=90):
        dedu[1][x] = '81-90'

for x in range(0,(len(dedu[5]))):
    if (0<=dedu[5][x]<=9):
        dedu[5][x] = 'Low'
    elif (10<=dedu[5][x]<=13):
        dedu[5][x] = 'Medium'
    elif (14<=dedu[5][x]<=16):
        dedu[5][x] = 'High'
```

Age was grouped according to different ranges as seen in the python program on the left. Education Level was also grouped into three categories.

Output can be seen on the right.

```
,31-40,Medium
,41-50,Medium
,31-40,Low
,51-60,Low
,21-30,Medium
,31-40,High
,41-50,Low
,51-60,Low
,31-40,High
,41-50,Medium
,31-40,Medium
,21-30,Medium
,21-30,Medium
,31-40,Medium
,31-40,Low
,21-30,Low
```

- Next I proceeded to write SQL scripts to create and import my tables in SSMS.

SQL scripts and Star Schema Diagram from SSMS

- I referred to Lab 4 to write SQL scripts to firstly create my tables and then import the data from the csv files generated earlier. I modified the scripts provided in Lab4 named “CreateTables.sql” and “InsertData.sql”. (I wrote my own queries to create and populate the tables).

  (Some Screenshots of my Tables Creation)-

```
Create table DimEducation
(
EduID bigint primary key identity,
AgeGroup varchar(20) not null,
EduLevel varchar(20) not null,
)
Go

PRINT '';
PRINT '*** Creating Table DimEmployment';
GO

Create table DimEmployment
(
EmpID bigint primary key identity,
Occupation varchar(50) not null,
WorkSector varchar(25) not null,
Gender varchar(10) not null,
)
Go
```

```
Create Table IncomeFact
(
FactID bigint primary key identity,
EmpID bigint not null,
EduID bigint not null,
IncomeID bigint not null,
PeopleCount bigint not null,
)
Go

PRINT '';
PRINT '*** Add relation between fact table foreign keys to Primary keys
GO

AlTER TABLE IncomeFact ADD CONSTRAINT
FK_EmpID FOREIGN KEY (EmpID)REFERENCES DimEmployment(EmpID);
AlTER TABLE IncomeFact ADD CONSTRAINT
FK_EduID FOREIGN KEY (EduID)REFERENCES DimEducation(EduID);
AlTER TABLE IncomeFact ADD CONSTRAINT
FK_IncomeID FOREIGN KEY (IncomeID)REFERENCES DimIncome(IncomeID);
Go
```

  (Some Screenshots of my Bulk importing of Data)-

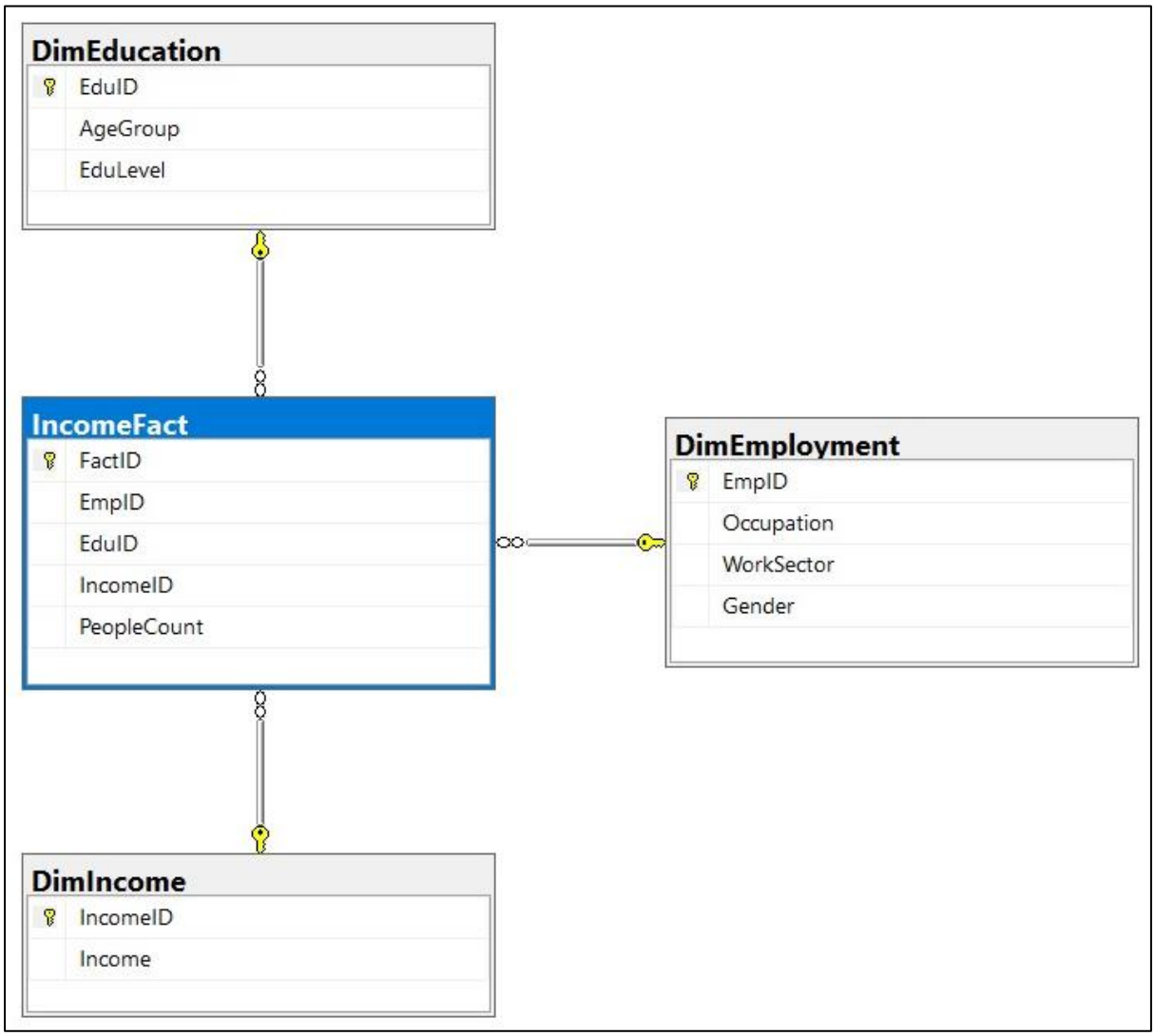
```
:setvar SqlSamplesSourceDataPath "D:\My Files\My Workspace\SQL Projects\MidSemProject\csv2import\"
:setvar DatabaseName "MidSemProject"
```

```
BULK INSERT [dbo].[DimEmployment] FROM '$(SqlSamplesSourceDataPath)DimEmploymentFinal.csv'
WITH (
    CHECK_CONSTRAINTS,
    --CODEPAGE='ACP',
    DATAFILETYPE='char',
    FIELDTERMINATOR=',',
    ROWTERMINATOR='\n',
    --KEEPIDENTITY,
    TABLOCK
);
```

```
BULK INSERT [dbo].[IncomeFact] FROM '$(SqlSamplesSourceDataPath)IncomeFactFinal.csv'
WITH (
    CHECK_CONSTRAINTS,
    --CODEPAGE='ACP',
    DATAFILETYPE='char',
    FIELDTERMINATOR=',',
    ROWTERMINATOR='\n',
    --KEEPIDENTITY,
    TABLOCK
);
```

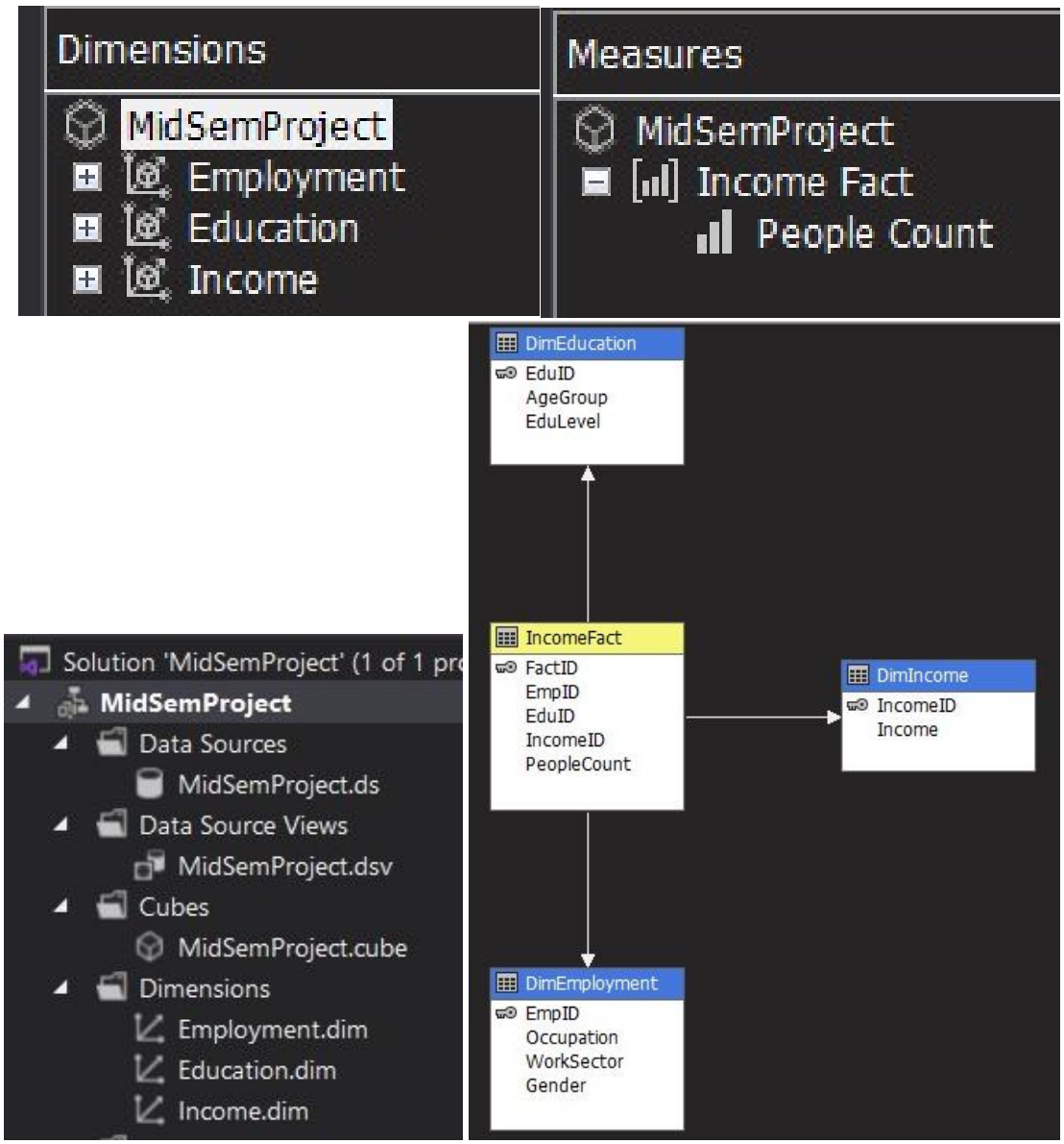
- (complete SQL code is in the Appendix at the end of this report.)

Star Schema Diagram from SSMS

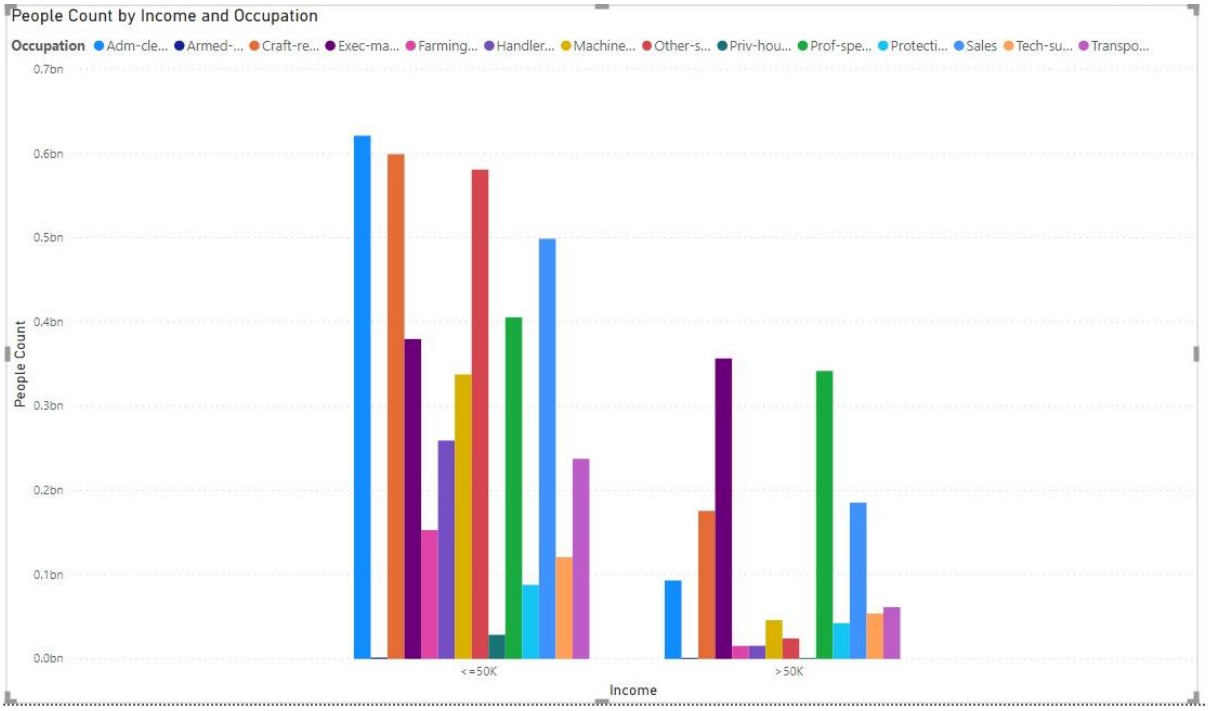


- After I made sure that my tables were properly created and all the data was correctly imported – I proceeded to use SQL Server Data Tools to build a multi-dimensional analysis service solution, with a cube designed to answer my business queries.

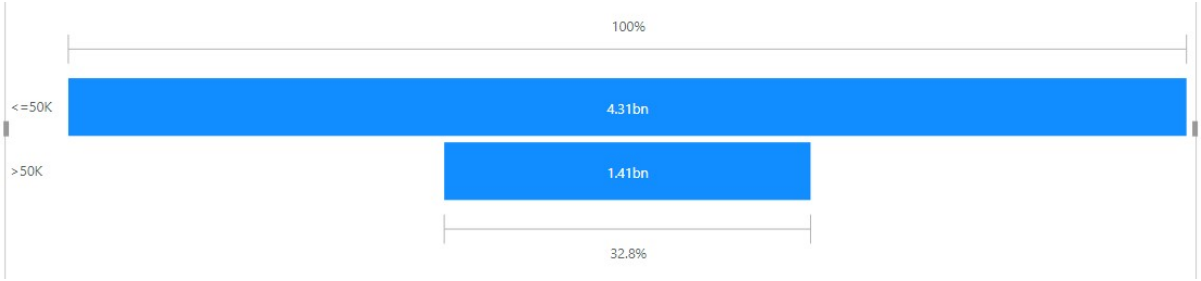
Cube Diagram



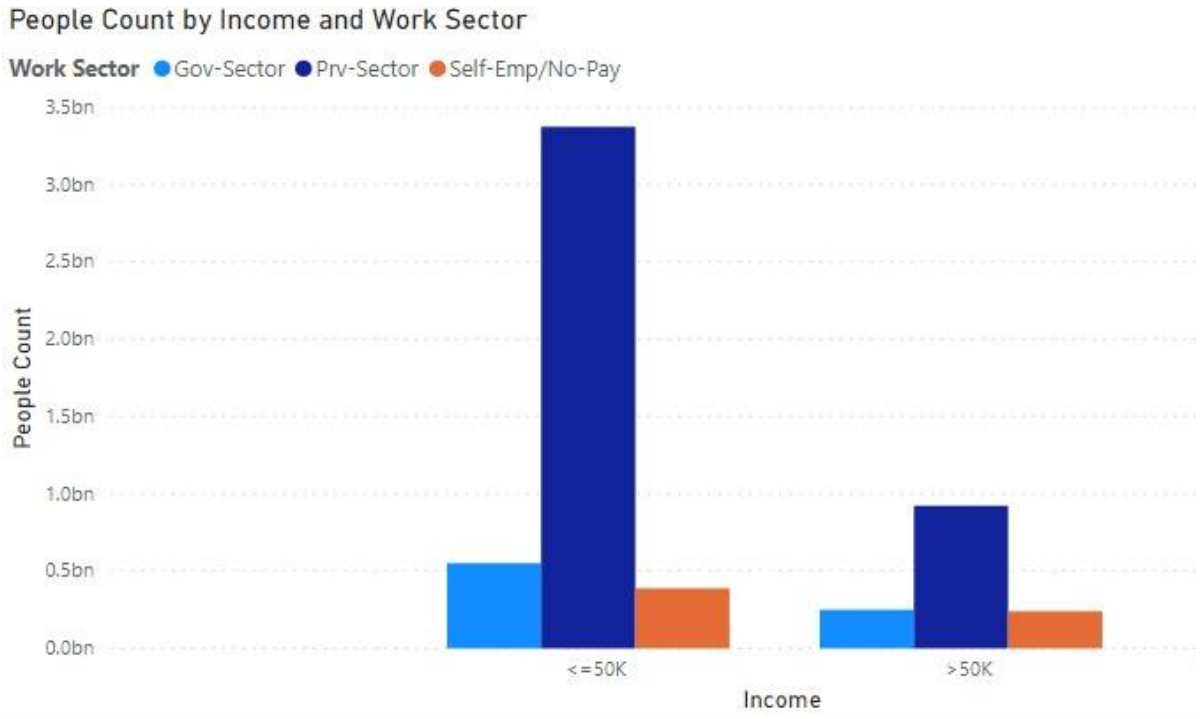
Q - How many people earn more than $50K across various occupations?



Answer – Seems like Adm-clerical, Craft-repair, “Other-Services” and Sales are the top 4 occupations that earn more than 50k per year. Also, 4.31bn people earn more than 50k per year compared to just 1.41bn people who earn under 50k.

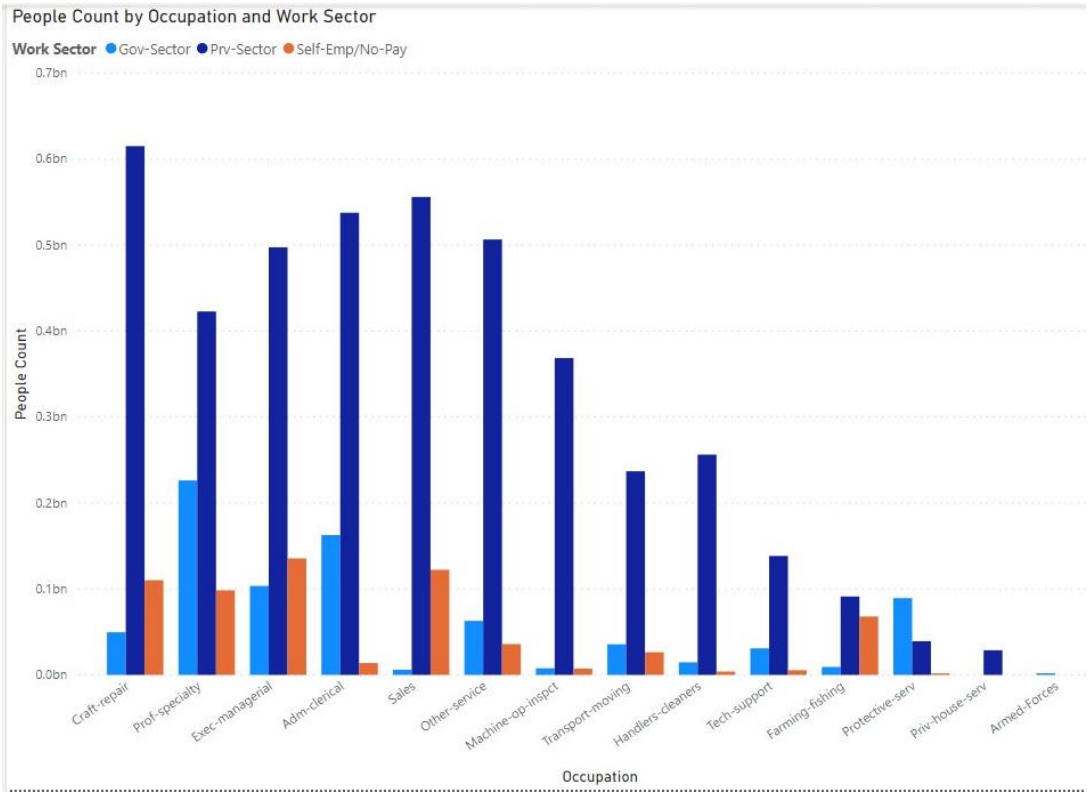


Q - Do these people mostly work in the public or private sector?

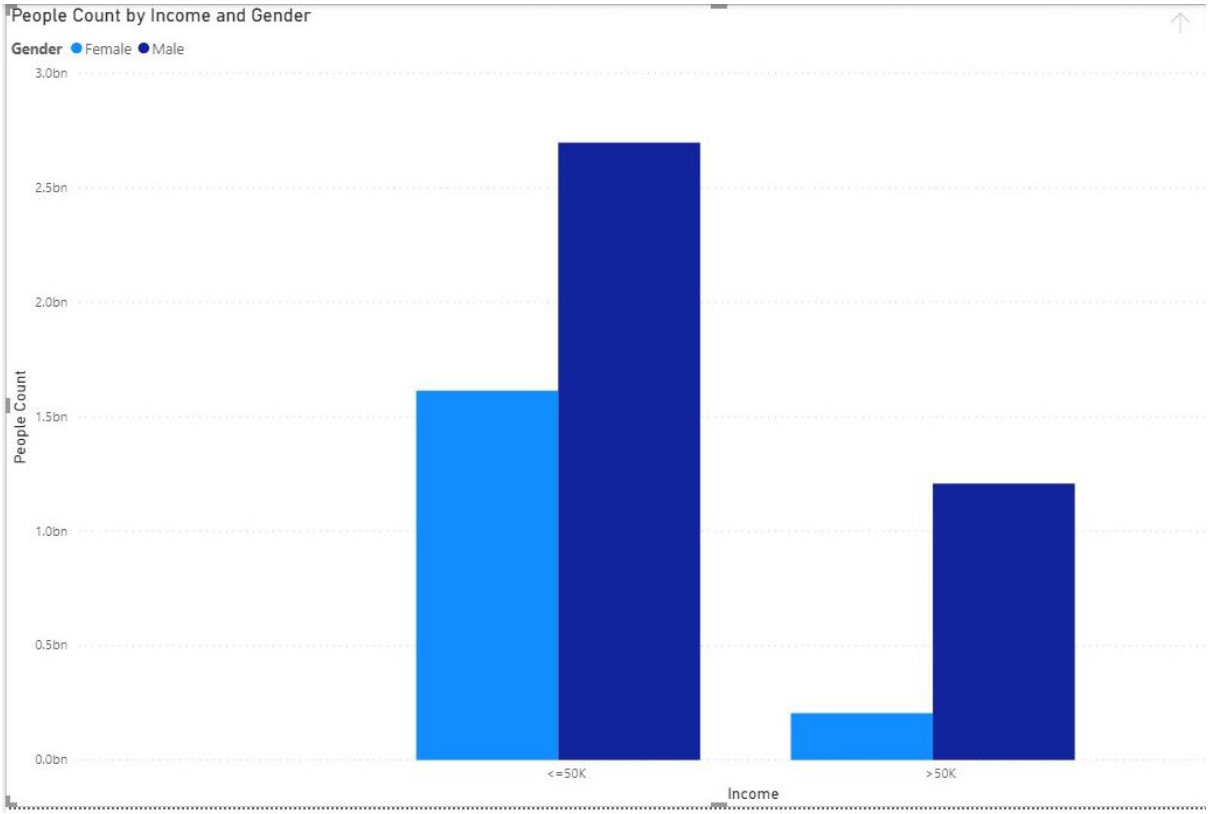


Answer – It seems that there is no doubt that most people who earn more than 50k per year are in the private sector.

We can also check how the people working in different sectors are distributed amongst the different occupations.

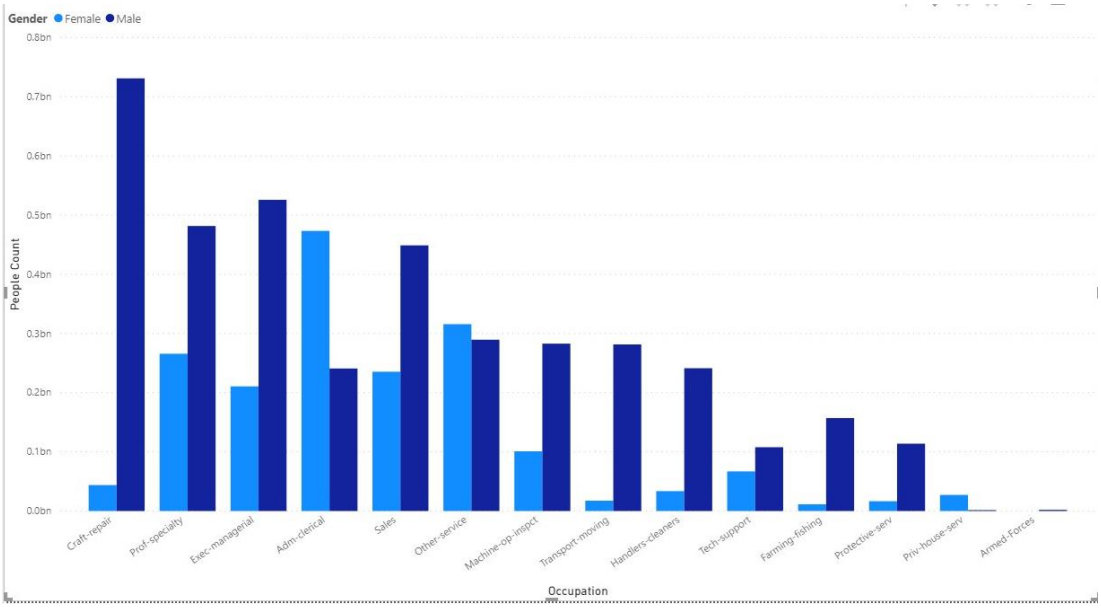


Q - (Are there roughly equal number of males and females who earn more than $50K? If not - which gender tends to earn more?)

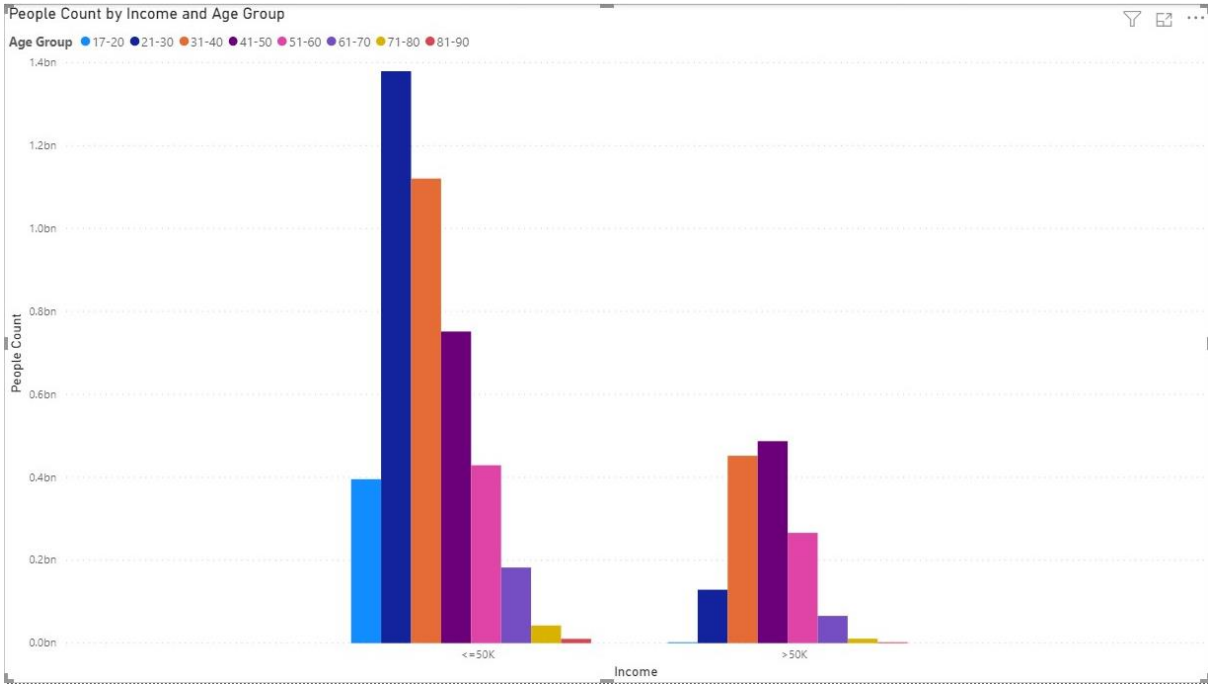


Answer – Here we can clearly see that the number of males that earn 50k is much higher than females. But – the number of males that earn lower than 50k is also higher which might mean that the number of females working is just less.

Below is an interesting graph that does not relate to income but tells us which professions are usually chosen by males or females more often.

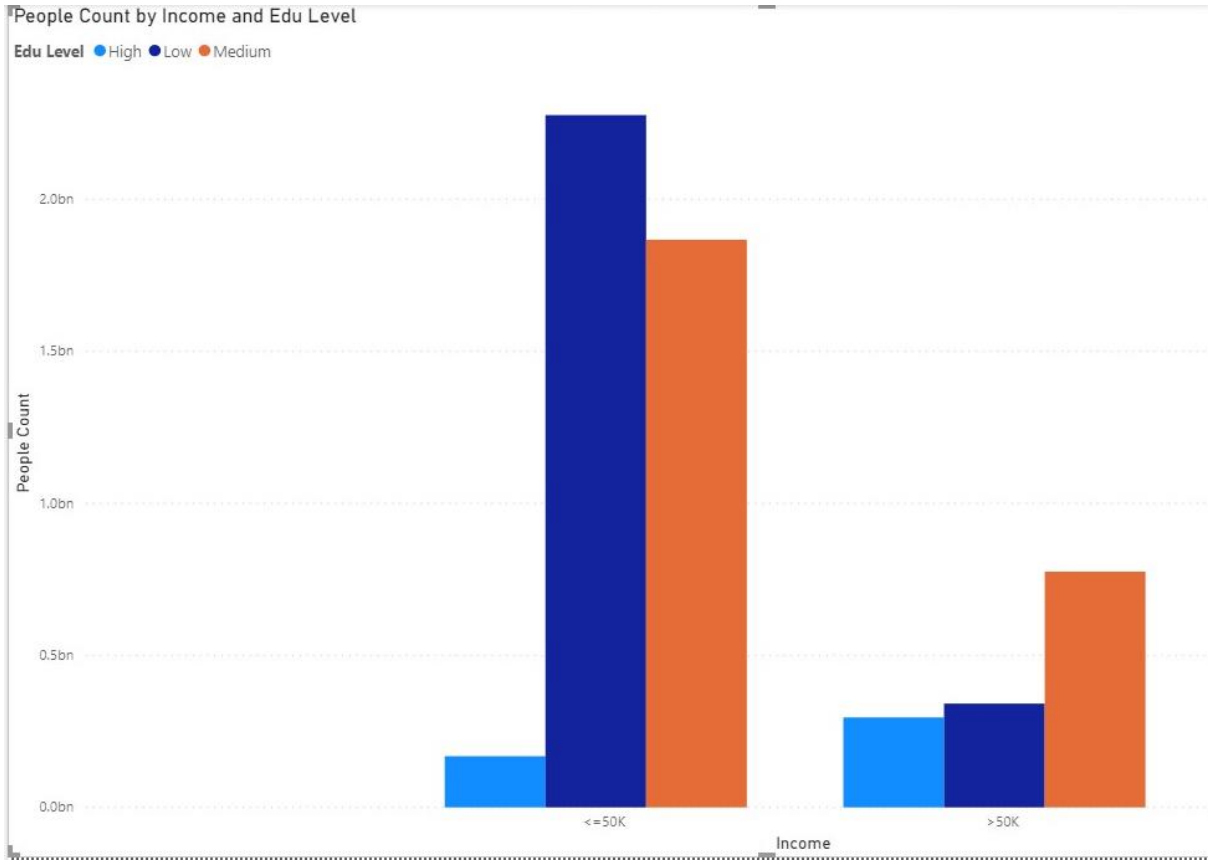


Q - How is the income across various age groups and which age group earns the most money?

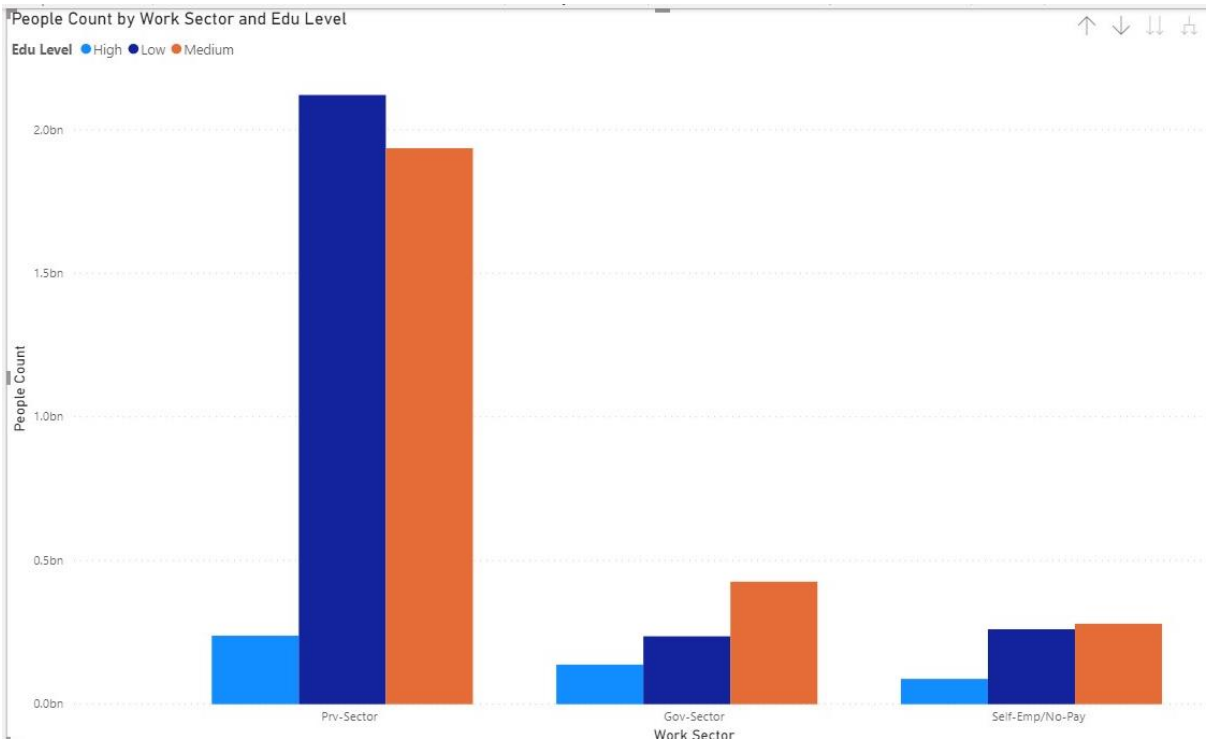


Answer – Seems like number of people who earn more than 50k is highest from ages 21-30. The number of people earning above 50k tends to slowly decline after this age range.

Q - What is the education level at with which people are earning the most money?



Answer – I was quite surprised to see that the majority of people who earn more than 50k per year have a low education qualification (High School diploma and below).



Another bonus observation is that if I add the Employment dimension to the above graph and group by work sector – I can see that most of these people who earn more than 50k and have a low educational qualification work in the Private sector.

APPENDIX

SQL codes used to create and populate Database

```
/*
 * File modified from Lab4 from LMS for MidSem Project.
 *
 * File modified by - GAURAV CHAKRAVERTY for MidSem Project.
 *
 */
Use master
Go
PRINT '';
PRINT '*** Dropping Database';
GO
IF EXISTS (SELECT [name] FROM [master].[sys].[databases] WHERE [name] = N'MidSemProject')
DROP DATABASE MidSemProject;
GO
PRINT '';
PRINT '*** Creating Database';
GO
Create database MidSemProject
Go
Use MidSemProject
Go
PRINT '';
PRINT '*** Creating Table DimEducation';
GO
Create table DimEducation
(
EduID bigint primary key identity,
AgeGroup varchar(20) not null,
EduLevel varchar(20) not null,
)
Go
PRINT '';
PRINT '*** Creating Table DimEmployment';
GO
Create table DimEmployment
(
```

```
EmpID bigint primary key identity,
Occupation varchar(50) not null,
WorkSector varchar(25) not null,
Gender varchar(10) not null,
)
Go
PRINT '';
PRINT '*** Creating Table DimIncome';
GO
Create table DimIncome
(
IncomeID bigint primary key identity,
Income varchar(10)not null,
)
Go
PRINT '';
PRINT '*** Creating Table IncomeFact';
GO
Create Table IncomeFact
(
FactID bigint primary key identity,
EmpID bigint not null,
EduID bigint not null,
IncomeID bigint not null,
PeopleCount bigint not null,
)
Go
PRINT '';
PRINT '*** Add relation between fact table foreign keys to Primary keys of Dimensions';
GO
AlTER TABLE IncomeFact ADD CONSTRAINT
FK_EmpID FOREIGN KEY (EmpID)REFERENCES DimEmployment(EmpID);
AlTER TABLE IncomeFact ADD CONSTRAINT
FK_EduID FOREIGN KEY (EduID)REFERENCES DimEducation(EduID);
AlTER TABLE IncomeFact ADD CONSTRAINT
FK_IncomeID FOREIGN KEY (IncomeID)REFERENCES DimIncome(IncomeID);
Go
```

Inserting Data into the Tables

```
/*
 * File modified from Lab4 from LMS for MidSem Project.
 *
 * File modified by - GAURAV CHAKRAVERTY for MidSem Project.
 *
 */
:setvar SqlSamplesSourceDataPath "D:\My Files\My Workspace\SQL Projects\MidSemProject\csv2import\"
:setvar DatabaseName "MidSemProject"
BULK INSERT [dbo].[DimEducation] FROM '$(SqlSamplesSourceDataPath)DimEducationFinal.csv'
WITH (
   CHECK_CONSTRAINTS,
   --CODEPAGE='ACP',
   DATAFILETYPE='char',
   FIELDTERMINATOR=',',
   ROWTERMINATOR='\n',
   --KEEPIDENTITY,
   TABLOCK
);
BULK INSERT [dbo].[DimEmployment] FROM '$(SqlSamplesSourceDataPath)DimEmploymentFinal.csv'
WITH (
   CHECK_CONSTRAINTS,
   --CODEPAGE='ACP',
   DATAFILETYPE='char',
   FIELDTERMINATOR=',',
   ROWTERMINATOR='\n',
   --KEEPIDENTITY,
   TABLOCK
);
BULK INSERT [dbo].[DimIncome] FROM '$(SqlSamplesSourceDataPath)DimIncomeFinal.csv'
WITH (
   CHECK_CONSTRAINTS,
   --CODEPAGE='ACP',
```

```
    DATAFILETYPE='char',
    FIELDTERMINATOR=',',
    ROWTERMINATOR='\n',
    --KEEPIDENTITY,
    TABLOCK
);
BULK INSERT [dbo].[IncomeFact] FROM '$(SqlSamplesSourceDataPath)IncomeFactFinal.csv'
WITH (
    CHECK_CONSTRAINTS,
    --CODEPAGE='ACP',
    DATAFILETYPE='char',
    FIELDTERMINATOR=',',
    ROWTERMINATOR='\n',
    --KEEPIDENTITY,
    TABLOCK
);
```

Python codes used for data cleaning and transformation

```
#Python Program to Clean and Transform Data from csv file for Data Warehousing Project.

#Author -> Gaurav Chakraverty

import numpy as np

import pandas as pd

from numpy import nan

from pandas import read_csv

df=pd.read_csv('adult-training.csv',header=None, sep=',\s', na_values=["?"])

df.replace('?', np.NaN, inplace=True)

df=df.dropna(axis='rows')

df.to_csv("adult-trainingcleaned.csv", header=None);

dfc=pd.read_csv('adult-trainingcleaned.csv',header=None, sep=',')

dfc[0] = ""

demp = dfc[[0,7,2,10]]

for x in range(0,(len(demp[2]))):

   if (demp[2][x] == 'Federal-gov') or (demp[2][x] == 'Local-gov') or (demp[2][x] == 'State-gov'):

      demp[2][x] = 'Gov-Sector'

   elif (demp[2][x] == 'Self-emp-not-inc') or (demp[2][x] == 'Self-emp-inc') or (demp[2][x] == 'Without-pay') or
(demp[2][x] == 'Never-worked'):

      demp[2][x] = 'Self-Emp/No-Pay'

   elif (demp[2][x] == 'Private'):

      demp[2][x] = 'Prv-Sector'

demp.to_csv("DimEmploymentFinal.csv", header=None, index=None);

dfc=pd.read_csv('adult-trainingcleaned.csv',header=None, sep=',')

dfc[0] = ""

dedu = dfc[[0,1,5]]

for x in range(0,(len(dedu[1]))):

   if (17<=dedu[1][x]<=20):

      dedu[1][x] = '17-20'

   elif (21<=dedu[1][x]<=30):

      dedu[1][x] = '21-30'

   elif (31<=dedu[1][x]<=40):

      dedu[1][x] = '31-40'
```

```
    elif (41<=dedu[1][x]<=50):
        dedu[1][x] = '41-50'
    elif (51<=dedu[1][x]<=60):
        dedu[1][x] = '51-60'
    elif (61<=dedu[1][x]<=70):
        dedu[1][x] = '61-70'
    elif (71<=dedu[1][x]<=80):
        dedu[1][x] = '71-80'
    elif (81<=dedu[1][x]<=90):
        dedu[1][x] = '81-90'
for x in range(0,(len(dedu[5]))):
    if (0<=dedu[5][x]<=9):
        dedu[5][x] = 'Low'
    elif (10<=dedu[5][x]<=13):
        dedu[5][x] = 'Medium'
    elif (14<=dedu[5][x]<=16):
        dedu[5][x] = 'High'
dedu.to_csv("DimEducationFinal.csv", header=None, index=None);
dfc=pd.read_csv('adult-trainingcleaned.csv',header=None, sep=',')
dfc[0] = ""
dinc = dfc[[0,15]]
dinc.to_csv("DimIncomeFinal.csv", header=None, index=None);
dfc=pd.read_csv('adult-trainingcleaned.csv',header=None, sep=',')
dfc[0] = ""
count = 0
for x in range(0,(len(dfc[1]))):
        dfc[1][x] = (count + 1)
        count = count+1
dfac = dfc[[0,1,1,1,3]]
dfac.to_csv("IncomeFactFinal.csv", header=None, index=None);
```

References –

- http://cseweb.ucsd.edu/classes/sp15/cse190-c/reports/sp15/048.pdf
- https://stackoverflow.com/questions/29247712/pandas-how-to-replace-with-nan-handling-non-standard-missing-values
- https://www.geeksforgeeks.org/python-creating-a-pandas-dataframe-column-based-on-a-given-condition/
- https://stackoverflow.com/questions/16327055/how-to-add-an-empty-column-to-a-dataframe
- https://stackoverflow.com/questions/26786960/remove-index-column-while-saving-csv-in-pandas
- I have also referred Lab4 from LMS and followed the instructions in the Labsheet to create and import data as well as create my cube.

-END OF REPORT-