Assignment 01

Questions 01 and Answers

1. List all professions and products in the dataset.

```
# Import CSV file check the column and data inside of it
```

In [1]:
```
#import all the needed packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re
import csv
```

In [2]:
```
#import the CSV file into the script to analyze
insurance_data = pd.read_csv("Insurance_Data_and_Description_CW1 (1).csv")

```

In [3]:
```
# check CSV file with its's initial file colum and data
insurance_data
```

Out[3]:

	ID	Product	Region	Profession	Age	First Rate Paid	Contract sum	Closing Date
0	1	Life	150	Employee	26.0	Y	1,000	Jul-90
1	2	Life	150	Farmer	31.0	N	2,000	Jul-90
2	3	Life	150	Farmer	33.0	Y	2,000	Jul-90
3	4	Life	150	Farmer	33.0	Y	2,000	Jul-90
4	5	Life	120	Self-employed	34.0	Y	2,000	Jul-90

```
# Answering Assignment questions
```

```
# A) List all professions and products
```

In [4]:
```
# a) List all professions and products in the dataset
insurance_product_and_profession = insurance_data[["Product", "Profession"]]
insurance_product_and_profession
```

Out[4]:

	Product	Profession
0	Life	Employee
1	Life	Farmer
2	Life	Farmer
3	Life	Farmer
4	Life	Self-employed
...	...	...
14840	Household	Civil servant
14841	Life	Civil servant
14842	Health	Civil servant
14843	Health	Civil servant
14844	Household	Civil servant

Import all the libraries on the top and develop the script to get output for product and profession.

2. Show any problem with the data etc.

The main problem in this data set is Wrong data types and N/A, nan values – which mean Null values. The solutions are provided, then transformed into to correct data type also removed Null values from the table.

```
# B) Show problems & Handling problems
```

In [5]:
```
# Show any problem with the data etc.
# 1. check whether the csv file have any null values
missing_value = ["N/a", "na", np.nan]
insurance_data = pd.read_csv("Insurance_Data_and_Description_CW1 (1).csv", na_values = missing_value)
insurance_data.isnull().sum()
insurance_data = insurance_data.dropna()
insurance_data.isnull().sum()
```

Out[5]:
```
ID                0
Product           0
Region            0
Profession        0
Age               0
First Rate Paid   0
Contract sum      0
Closing Date      0
dtype: int64
```

3. Clean and transform the data into the correct data types.

The abovementioned problem was sorted here.

```
# C) Clean and transform
```

In [8]:
```

```

In [6]:
```
#As this result there are so many Null values in both of the columns
#In order to clean these problem removing all the null values from dataset

insurance_data['Contract sum'] = insurance_data['Contract sum'].str.replace(',','0')
insurance_data['Contract sum'] = insurance_data['Contract sum'].str.replace('K','0')
insurance_data['Contract sum'] = insurance_data['Contract sum'].str.replace('k','0')


insurance_data['Contract sum'] = insurance_data['Contract sum'].replace(',',"0")
insurance_data['Contract sum'] = insurance_data['Contract sum'].astype(int)

cleaned_insurance_data = insurance_data.dropna()

cleaned_insurance_data
```

4. Add a new column “Age Group” and fill up its data according to below criteria.

here I am using between and comparison operators to check the age and created the colum for Age group.

D) Add a new column “Age Group"

In [8]:
```
cleaned_insurance_data.loc[cleaned_insurance_data['Age']<=16, 'Age Group'] = 'AG-1'
cleaned_insurance_data.loc[cleaned_insurance_data['Age'].between(17,24), 'Age Group'] = 'AG-2'
cleaned_insurance_data.loc[cleaned_insurance_data['Age'].between(25,29), 'Age Group'] = 'AG-3'
cleaned_insurance_data.loc[cleaned_insurance_data['Age'].between(20,39), 'Age Group'] = 'AG-4'
cleaned_insurance_data.loc[cleaned_insurance_data['Age'].between(40,49), 'Age Group'] = 'AG-5'
cleaned_insurance_data.loc[cleaned_insurance_data['Age'].between(50,59), 'Age Group'] = 'AG-6'
cleaned_insurance_data.loc[cleaned_insurance_data['Age']>=60, 'Age Group'] = 'AG-7'

# check the edited file
cleaned_insurance_data[["Age","Age Group"]]
```

Out[8]:

	Age	Age Group
0	26.0	AG-4
1	31.0	AG-4
2	33.0	AG-4
3	33.0	AG-4
4	34.0	AG-4
...	...	...
14840	78.0	AG-7
14841	78.0	AG-7
14842	82.0	AG-7

5. Rank region by total insurance product.

using the count function to for region and group by the to identify the total insurance product for each Region.

E) Rank region by total insurance product

In [9]:
```
Rank = cleaned_insurance_data.groupby('Region',sort=False).count()
Rank['Product']
```

Out[9]:
```
Region
150    12281
120       16
160     2265
999      195
130       36
Name: Product, dtype: int64
```

6. Find the mean, median, standard deviation of Age and Contract Sum.

Finding the summary of Age group and Contract sum in the dataset, which Is showing up mean, median etc.

```
# F) Find the mean, median, standard deviation of Age and Contract Sum
```

In [10]:
```
# 1. For Age
cleaned_insurance_data["Age"].describe()
```

Out[10]:
```
count    14793.00000
mean        44.93578
std         14.66991
min          1.00000
25%         34.00000
50%         45.00000
75%         57.00000
max         95.00000
Name: Age, dtype: float64
```

In [11]:
```
# 2. For Contract Sum
insurance_data['Contract sum'].astype(str).astype(int)
insurance_data["Contract sum"].describe()
```

Out[11]:
```
count    1.479300e+04
mean     2.892438e+05
std      2.584105e+05
min      4.000000e+01
25%      1.500000e+05
50%      2.000000e+05
75%      3.300000e+05
max      4.250000e+06
Name: Contract sum, dtype: float64
```

7. Filter the rows of Age < 25 and Contract Sum between 50k and 100k and insurance type of Health or Life insurance.

Filtering the data with above condition, I used single line comparison to done the filtering.

```
# G) filtering age, contract sum and insurance type
```

In [13]:
```
# selecting rows based on condition
filderd_data = cleaned_insurance_data[(cleaned_insurance_data['Age'] < 25) &  ((cleaned_insurance_data['Product'] == 'Life')
filderd_data
```

Out[13]:

	ID	Product	Region	Profession	Age	First Rate Paid	Contract sum	Closing Date	Age Group
29	30	Health	150	Student	21.0	Y	50000	Aug-90	AG-4
196	197	Health	150	Student	16.0	N	50000	Sep-90	AG-1
1100	1101	Life	150	Executive emplyee	7.0	Y	100000	Nov-90	AG-1
1164	1165	Health	150	Student	23.0	N	100000	Dec-90	AG-4
1165	1166	Life	160	Student	24.0	Y	100000	Dec-90	AG-4
...	...	...	...	...	...	...	...	...	...
14506	14507	Health	150	Student	23.0	N	1000000	Dec-96	AG-4
14507	14508	Health	150	Student	23.0	N	1000000	Dec-96	AG-4
14509	14510	Health	160	Farmer	24.0	N	1000000	Dec-96	AG-4
14510	14511	Health	150	Student	24.0	N	1000000	Dec-96	AG-4
14511	14512	Health	150	Student	24.0	N	1000000	Dec-96	AG-4

8. Find total insurance product by profession.

Finding product over the profession, by grouping profession and calculate product in the professions.

Find total insurance product by profession

In [13]:
```
total_insurance_product = cleaned_insurance_data.groupby('Profession').count()
total_insurance_product['Product']
```

Out[13]:
```
Profession
Civil servant        2044
Employee             5602
Executive emplyee     575
Farmer               3686
Pupil                 190
Self-employed          20
Student              2552
Worker                124
Name: Product, dtype: int64
```

9. Find correlations between product, profession, and contract sum. What are your conclusions?

   The correlation is can only calculate with numerical values, so other variable could not do for the correlation.

 # Find correlations between product, profession and contract sum

In [14]:
```
dataset = cleaned_insurance_data[['Product','Profession','Contract sum']]

print( dataset.corr())

```

```
             Contract sum
Contract sum          1.0
```

10. Calculate and Visualize products on age group. Describe your findings

Calculate and Visualise products on age group

In [24]:

```
x = []
y = []

visualize_data = total_insurance_product = cleaned_insurance_data.groupby('Profession').count()
visualize_data['Product'] = visualize_data['Product'].astype(int)


new_data = visualize_data[['Product', 'Age Group']]
new_data = new_data.to_csv('newfile.csv')

with open('newfile.csv','r') as csvfile:
    plots = csv.reader(csvfile, delimiter = ',')

    for row in plots:
        x.append(int(row[0]))
        y.append(row[1])


plt.bar(x, y, color = 'g', width = 0.72, label = "Age")
plt.xlabel('Products')
plt.ylabel('Age group')
plt.title('Ages of different product')
plt.legend()
plt.show()

visualize_data
```
