IBM Developer
SKILLS NETWORK

Data Wrangling Lab

Estimated time needed: **45 to 60** minutes

In this assignment you will be performing data wrangling.

Objectives

In this lab you will perform the following:

- Identify duplicate values in the dataset.
- Remove duplicate values from the dataset.
- Identify missing values in the dataset.
- Impute the missing values in the dataset.
- Normalize data in the dataset.

Hands on Lab

Import pandas module.

In [1]:
```
import pandas as pd
```

Load the dataset into a dataframe.

In [2]:
```
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeData/m1_survey_data.csv")
```

Finding duplicates

In this section you will identify duplicate values in the dataset.

Find how many duplicate rows exist in the dataframe.

In [9]:
```
# your code goes here
x = df[df.duplicated()]
len(x)
```

Out[9]:
```
154
```

In [11]:
```
y = df[df['Respondent'].duplicated()]
y
```

Out[11]:

	Respondent	MainBranch	Hobbyist	OpenSourcer	OpenSource	Employment	Country	Student	EdLevel	UndergradMajor	...
1168	2339	I am a developer by profession	Yes	Once a month or more often	OSS is, on average, of HIGHER quality than pro...	Employed full-time	United States	No	Some college/university study without earning ...	Computer science, computer engineering, or sof...	...
1169	2342	I am a developer by profession	Yes	Never	The quality of OSS and closed source software ...	Employed full-time	United Kingdom	No	Some college/university study without earning ...	Information systems, information technology, o...	...
1170	2343	I am a developer by profession	Yes	Less than once a month but more than once per ...	OSS is, on average, of LOWER quality than prop...	Employed full-time	Canada	No	Master's degree (MA, MS, M.Eng., MBA, etc.)	Computer science, computer engineering, or sof...	...
1171	2344	I am a developer by profession	Yes	Never	The quality of OSS and closed source software ...	Employed full-time	United States	No	Bachelor's degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...	...
1172	2347	I am a developer by profession	Yes	Never	OSS is, on average, of HIGHER quality than pro...	Employed full-time	United Kingdom	No	Master's degree (MA, MS, M.Eng., MBA, etc.)	Computer science, computer engineering, or sof...	...
...	...	...	...	...	...	...	...	...	...	...	...
2297	4674	I am not primarily a developer, but I write co...	Yes	Less than once per year	The quality of OSS and closed source software ...	Employed full-time	Bangladesh	No	Bachelor's degree (BA, BS, B.Eng., etc.)	NaN	...
2298	4675	I am a developer by profession	Yes	Never	OSS is, on average, of HIGHER quality than pro...	Employed full-time	United States	No	Bachelor's degree (BA, BS, B.Eng., etc.)	Information systems, information technology, o...	...
2299	4676	I am a developer by profession	Yes	Never	OSS is, on average, of HIGHER quality than pro...	Employed full-time	Finland	No	Master's degree (MA, MS, M.Eng., MBA, etc.)	Another engineering discipline (ex. civil, ele...	...
2300	4677	I am a developer by profession	Yes	Once a month or more often	OSS is, on average, of HIGHER quality than pro...	Employed full-time	United Kingdom	No	Bachelor's degree (BA, BS, B.Eng., etc.)	A natural science (ex. biology, chemistry, phy...	...
2301	4679	I am a developer by profession	Yes	Less than once a month but more than once per ...	The quality of OSS and closed source software ...	Employed full-time	United States	No	Master's degree (MA, MS, M.Eng., MBA, etc.)	Computer science, computer engineering, or sof...	...

154 rows × 85 columns

Removing duplicates

Remove the duplicate rows from the dataframe.

In [12]:
```
# your code goes here
data = df.drop_duplicates()
```

In [13]:
```
data.drop_duplicates(inplace=False)
```

Out[13]:

	Respondent	MainBranch	Hobbyist	OpenSourcer	OpenSource	Employment	Country	Student	EdLevel	UndergradMajor	...
0	4	I am a developer by profession	No	Never	The quality of OSS and closed source software ...	Employed full-time	United States	No	Bachelor's degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...	...
1	9	I am a developer by profession	Yes	Once a month or more often	The quality of OSS and closed source software ...	Employed full-time	New Zealand	No	Some college/university study without earning ...	Computer science, computer engineering, or sof...	...
2	13	I am a developer by profession	Yes	Less than once a month but more than once per ...	OSS is, on average, of HIGHER quality than pro...	Employed full-time	United States	No	Master's degree (MA, MS, M.Eng., MBA, etc.)	Computer science, computer engineering, or sof...	...
3	16	I am a developer by profession	Yes	Never	The quality of OSS and closed source software ...	Employed full-time	United Kingdom	No	Master's degree (MA, MS, M.Eng., MBA, etc.)	NaN	...
4	17	I am a developer by profession	Yes	Less than once a month but more than once per ...	The quality of OSS and closed source software ...	Employed full-time	Australia	No	Bachelor's degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...	...
...	...	...	...	...	...	...	...	...	...	...	...
11547	25136	I am a developer by profession	Yes	Never	OSS is, on average, of HIGHER quality than pro...	Employed full-time	United States	No	Master's degree (MA, MS, M.Eng., MBA, etc.)	Computer science, computer engineering, or sof...	...
11548	25137	I am a developer by profession	Yes	Never	The quality of OSS and closed source software ...	Employed full-time	Poland	No	Master's degree (MA, MS, M.Eng., MBA, etc.)	Computer science, computer engineering, or sof...	...
11549	25138	I am a developer by profession	Yes	Less than once per year	The quality of OSS and closed source software ...	Employed full-time	United States	No	Master's degree (MA, MS, M.Eng., MBA, etc.)	Computer science, computer engineering, or sof...	...
11550	25141	I am a developer by profession	Yes	Less than once a month but more than once per ...	OSS is, on average, of LOWER quality than prop...	Employed full-time	Switzerland	No	Secondary school (e.g. American high school, G...	NaN	...
11551	25142	I am a developer by profession	Yes	Less than once a month but more than once per ...	OSS is, on average, of HIGHER quality than pro...	Employed full-time	United Kingdom	No	Other doctoral degree (Ph.D, Ed.D., etc.)	A natural science (ex. biology, chemistry, phy...	...

11398 rows × 85 columns

In [14]:
```
df.drop_duplicates(inplace=True)
```

Verify if duplicates were actually dropped.

In [15]:
```
# your code goes here
df[df.duplicated()]
```

Out[15]:

Respondent	MainBranch	Hobbyist	OpenSourcer	OpenSource	Employment	Country	Student	EdLevel	UndergradMajor	...	WelcomeChan

0 rows × 85 columns

Finding Missing values

Find the missing values for all columns.

In [16]:
```
# your code goes here
df.isnull()

df.isnull().values.any()

df.isna().sum().sum()
```

Out[16]:
```
30967
```

Find out how many rows are missing in the column 'WorkLoc'

In [17]:
```
# your code goes here
df['WorkLoc'].isna().sum()
```

Out[17]:
```
32
```

After removing the duplicate rows, how many blank rows are there under the column EdLevel?

In [18]:
```
df['EdLevel'].isna().sum()
```

Out[18]:
```
112
```

After removing the duplicate rows, how many rows are missing under the column Country?

In [19]:
```
df['Country'].isna().sum()
```

Out[19]:
```
0
```

Imputing missing values

Find the value counts for the column WorkLoc.

- What is the majority category under the column Employment?

In [20]:
```
df['Employment'].value_counts()
```

Out[20]:
```
Employed full-time    10968
Employed part-time      430
Name: Employment, dtype: int64
```

- Under the column " UndergradMajor", which category has the minimum number of rows?

In [21]:
```
df['UndergradMajor'].value_counts()
```

Out[21]:
```
Computer science, computer engineering, or software engineering          6953
Information systems, information technology, or system administration     794
Another engineering discipline (ex. civil, electrical, mechanical)        759
Web development or web design                                             410
A natural science (ex. biology, chemistry, physics)                       403
Mathematics or statistics                                                 372
A business discipline (ex. accounting, finance, marketing)                244
A social science (ex. anthropology, psychology, political science)        210
A humanities discipline (ex. literature, history, philosophy)             207
Fine arts or performing arts (ex. graphic design, music, studio art)      161
I never declared a major                                                  124
A health science (ex. nursing, pharmacy, radiology)                        24
Name: UndergradMajor, dtype: int64
```

- Question 3 The column 'ConvertedComp' contains the annual compensation of the survey respondents. What is the best approach to impute the missing values in this column?

In [22]:
```
df['ConvertedComp'].value_counts()
```

Out[22]:
```
2000000.0    138
1000000.0    105
100000.0      99
150000.0      92
120000.0      86
            ...
53148.0        1
256000.0       1
54423.0        1
392.0          1
41280.0        1
Name: ConvertedComp, Length: 3515, dtype: int64
```

In [23]:
```
df['ConvertedComp'].mean()
```

Out[23]:
```
131596.7316197316
```

In [24]:
```
df['ConvertedComp'].median()
```

Out[24]:
```
57745.0
```

- Median is the best option to fill the missing values.

In [39]:
```
# your code goes here
df['WorkLoc'].count()
```

Out[39]:
```
11366
```

Identify the value that is most freqdescribet (majority) in the WorkLoc column.

In [40]:
```
#make a note of the majority value here, for future reference
# The most frequently is
df['WorkLoc'].value_counts()
```

Out[40]:
```
Office                                            6806
Home                                              3589
Other place, such as a coworking space or cafe     971
Name: WorkLoc, dtype: int64
```

Impute (replace) all the empty rows in the column WorkLoc with the value that you have identified as majority.

In [41]:
```
# your code goes here
df["WorkLoc"].fillna("Office", inplace = True)
```

After imputation there should ideally not be any empty rows in the WorkLoc column.

Verify if imputing was successful.

In [45]:
```
# your code goes here
print("Number of Missing Values are:",df['WorkLoc'].isna().sum())
df['WorkLoc'].value_counts()
```

```
Number of Missing Values are: 0
```

Out[45]:
```
Office                                            6838
Home                                              3589
Other place, such as a coworking space or cafe     971
Name: WorkLoc, dtype: int64
```

Normalizing data

There are two columns in the dataset that talk about compensation.

One is "CompFreq". This column shows how often a developer is paid (Yearly, Monthly, Weekly).

The other is "CompTotal". This column talks about how much the developer is paid per Year, Month, or Week depending upon his/her "CompFreq".

This makes it difficult to compare the total compensation of the developers.

In this section you will create a new column called 'NormalizedAnnualCompensation' which contains the 'Annual Compensation' irrespective of the 'CompFreq'.

Once this column is ready, it makes comparison of salaries easy.

Q1. How many unique values are there in the CompFreq column?

In [27]:
```
df['CompFreq'].unique()
```

Out[27]:
```
array(['Yearly', 'Monthly', 'Weekly', nan], dtype=object)
```

Q2. After removing the duplicate rows, how many respondents are being paid yearly?

In [40]:
```
xx = df[df['CompFreq'] == 'Yearly']
len(xx['CompFreq'])
```

Out[40]:
```
6073
```

List out the various categories in the column 'CompFreq'

In [49]:
```
# your code goes here
df['CompFreq'].value_counts()
```

Out[49]:
```
Yearly     6073
Monthly    4788
Weekly      331
Name: CompFreq, dtype: int64
```

Create a new column named 'NormalizedAnnualCompensation'. Use the hint given below if needed.

Double click to see the **Hint**.

In [41]:
```
# your code goes here
NormalizedAnnualCompensation = []
my_list=[]
x = df[['CompTotal','CompFreq']]

for column in x.columns:

    # Storing the rows of a column
    # into a temporary list
    li = x[column].tolist()

    # appending the temporary list
    my_list.append(li)


compFreq=my_list[1]
compTotal = my_list[0]
i=0
for j in compFreq:

    if compFreq[i] =='Yearly':
        NormalizedAnnualCompensation.append(compTotal[i])
    elif compFreq[i] == 'Monthly':
        NormalizedAnnualCompensation.append(compTotal[i]*12)
    elif compFreq[i] == 'Weekly':
        NormalizedAnnualCompensation.append(compTotal[i]*52)
    else:
        NormalizedAnnualCompensation.append(compTotal[i])
    i= i + 1


df['NormalizedAnnualCompensation']=NormalizedAnnualCompensation
df['NormalizedAnnualCompensation']
```

Out[41]:
```
0         61000.0
1        138000.0
2         90000.0
3        348000.0
4         90000.0
           ...
11547    130000.0
11548     74400.0
11549    105000.0
11550     80000.0
11551         NaN
Name: NormalizedAnnualCompensation, Length: 11398, dtype: float64
```

- What is the median NormalizedAnnualCompensation?

In [43]:
```
df['NormalizedAnnualCompensation'].median()
```

Out[43]:
```
100000.0
```

In [42]:
```
df['NormalizedAnnualCompensation'].describe()
```

Out[42]:
```
count    1.058900e+04
mean     6.133295e+06
std      9.838157e+07
min      0.000000e+00
25%      5.200000e+04
50%      1.000000e+05
75%      3.600000e+05
max      8.400000e+09
Name: NormalizedAnnualCompensation, dtype: float64
```

Authors

Ramesh Sannareddy

Other Contributors

Rav Ahuja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-10-17	0.1	Ramesh Sannareddy	Created initial version of the lab