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 [21]:
```
import pandas as pd
```

Load the dataset into a dataframe.

In [31]:
```
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
```

Finding duplicates

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

Find how many duplicate rows exist in the dataframe.

In [41]:
```
# your code goes here
df.duplicated().sum()
```

Out[41]: 154

Removing duplicates

Remove the duplicate rows from the dataframe.

In [51]:
```
# your code goes here
df.drop_duplicates(inplace=True)
df.shape
```

Out[51]: (11398, 85)

Verify if duplicates were actually dropped.

In [61]:
```
# your code goes here
df.duplicated().sum()
```

Out[61]: 0

In [ ]:
```

```

Finding Missing values

Find the missing values for all columns.

In [71]:
```
# your code goes here
df.isnull().sum()
```

Out[71]:
```
Respondent            0
MainBranch            0
Hobbyist              0
OpenSourcer           0
OpenSource           81
                    ...
Sexuality           542
Ethnicity           675
Dependents          140
SurveyLength         19
SurveyEase           14
Length: 85, dtype: int64
```

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

In [81]:
```
# your code goes here
print ('There are', df['WorkLoc'].isnull().sum(),'rows missing in the column Workloc')
```

```
There are 32 rows missing in the column Workloc
```

Imputing missing values

Find the value counts for the column WorkLoc.

In [91]:
```
# your code goes here
print ('There are', df['WorkLoc'].nunique(), 'unique work stations')

print('\nWorkLoc                                    value count')
print(df['WorkLoc'].value_counts())

print('\n\nEmployment       value count')
print('----------          ----------')
print(df['Employment'].value_counts())

print('\n\nThere are', df['UndergradMajor'].nunique(), 'unique UndergradMajor values i

print('\n\nUndergradMajor                                                  value
print('---------------                                                 --------
print(df['UndergradMajor'].value_counts())
```

```
There are 3 unique work stations

WorkLoc                                value count
Office                                      6806
Home                                        3589
Other place, such as a coworking space or cafe       971
Name: WorkLoc, dtype: int64


Employment          value count
----------          ----------
Employed full-time    10968
Employed part-time      430
Name: Employment, dtype: int64


There are 12 unique UndergradMajor values in the survey:

UndergradMajor                                                  value count
---------------                                                 -----------
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
```

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

In [101]:
```
#make a note of the majority value here, for future reference
#office 6,806
```

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

In [111]:
```
import numpy as np

workloc_highest = 'Office'

missing_data = df.isnull()

df['WorkLoc'].replace(np.nan, workloc_highest, inplace=True)
```

In [121]:
```
df['WorkLoc'].isnull().sum()
```

Out[121]: 0

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

Verify if imputing was successful.

In [131]:
```
# your code goes here
df['WorkLoc'].isnull().sum()
```

Out[131]: 0

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.

List out the various categories in the column 'CompFreq'

In [141]:
```
# your code goes here
df['CompFreq'].value_counts()
```

Out[141]:
```
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 [151]:
```
# your code goes here
df.loc[df['CompFreq'] == 'Yearly', 'NormalizedAnnualCompensation']  = 1  * df['CompTot
df.loc[df['CompFreq'] == 'Monthly', 'NormalizedAnnualCompensation'] = 12 * df['CompTot
df.loc[df['CompFreq'] == 'Weekly', 'NormalizedAnnualCompensation']  = 52 * df['CompTot

df['NormalizedAnnualCompensation'].median()
```

Out[151]: 100000.0

In [161]:
```
df.head()
```

Out[161]:

	Respondent	MainBranch	Hobbyist	OpenSourcer	OpenSource	Employment	Country	Student
0	4	I am a developer by profession	No	Never	The quality of OSS and closed source software ...	Employed full-time	United States	No
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
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
3	16	I am a developer by profession	Yes	Never	The quality of OSS and closed source software ...	Employed full-time	United Kingdom	No
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

5 rows × 86 columns

In [ ]:
```

```

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