Skills Network

(https://skills.network/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDA0321ENSkillsNetwork928-2022-01-01)

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 [3]:
```
import pandas as pd
```

Load the dataset into a dataframe.

In [4]:
```
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeData/m1_surv
```

In [5]:
```
df.head()
```

Out[5]:

	Respondent	MainBranch	Hobbyist	OpenSourcer	OpenSource	Employment	Country	Student	EdLevel	UndergradMajor	...	WelcomeChange	SON
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...	...	Just as welcome now as I felt last year	T writ develo
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...	...	Just as welcome now as I felt last year	
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...	...	Somewhat more welcome now than last year	T writ develo
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	...	Just as welcome now as I felt last year	T writ develo
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...	...	Just as welcome now as I felt last year	T writ develo

5 rows × 85 columns

```
In [29]: df1[["WorkLoc"]]
```

Out[29]:

	WorkLoc
0	Home
1	Office
2	Home
3	Home
4	Other place, such as a coworking space or cafe
...	...
11547	Home
11548	Home
11549	Office
11550	Home
11551	Office

11398 rows × 1 columns

Finding duplicates

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

Find how many duplicate rows exist in the dataframe.

```
In [6]:  # your code goes here
         df.duplicated().sum()

Out[6]:  154
```

Removing duplicates

Remove the duplicate rows from the dataframe.

```
In [9]:  # your code goes here
         dropped_duplicates = df.drop_duplicates()
```

Verify if duplicates were actually dropped.

```
In [11]: # your code goes here
         dropped_duplicates.duplicated().sum()

Out[11]: 0
```

```
In [12]: df1 = dropped_duplicates
```

Finding Missing values

Find the missing values for all columns.

```
In [22]: # your code goes here
         df1.isnull().sum()

Out[22]: Respondent        0
         MainBranch        0
         Hobbyist          0
         OpenSourcer       0
         OpenSource       81
                        ... 
         Sexuality       542
         Ethnicity       675
         Dependents      140
         SurveyLength     19
         SurveyEase       14
         Length: 85, dtype: int64
```

In [23]:

```
df1.isnull().sum().sort_values(ascending = False)
```

Out[23]:

```
BlockchainIs          2610
CodeRevHrs            2426
BlockchainOrg         2322
MiscTechWorkedWith    2182
SONewContent          1965
                      ...
JobSeek                  0
MainBranch               0
LastHireDate             0
CurrencySymbol           0
Respondent               0
Length: 85, dtype: int64
```

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

In [24]:

```
# your code goes here
df1.WorkLoc.isnull().sum()
```

Out[24]:

```
32
```

Imputing missing values

Find the value counts for the column WorkLoc.

In [25]:

```
# your code goes here
print(df1['WorkLoc'].value_counts(dropna=False))
```

```
Office                                            6806
Home                                              3589
Other place, such as a coworking space or cafe     971
NaN                                                 32
Name: WorkLoc, dtype: int64
```

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

In [2]:

```
#make a note of the majority value here, for future reference
#Office 6806
```

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

In [36]:

```
# your code goes here
df1['WorkLoc'] = df1['WorkLoc'].fillna(df['WorkLoc'].mode()[0])
```

```
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
```

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

Verify if imputing was successful.

In [38]:

```
# your code goes here
print(df1['WorkLoc'].value_counts(dropna=False))
```

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

In [39]:

```
df1['WorkLoc'].isna().sum()
```

Out[39]:

```
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 [41]:

```
# your code goes here
df1['CompFreq'].value_counts()
```

Out[41]:

```
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 [40]:

```
# your code goes here
df1.reset_index(drop=True, inplace=True)

df1.loc[df_without_duplicates['CompFreq'] == 'Monthly', 'NormalizedAnnualCompensation'] = df_without_duplicates['CompTotal'] * 12
df_without_duplicates.loc[df_without_duplicates['CompFreq'] == 'Yearly', 'NormalizedAnnualCompensation'] = df_without_duplicates[
df_without_duplicates.loc[df_without_duplicates['CompFreq'] == 'Weekly', 'NormalizedAnnualCompensation'] = df_without_duplicates[
df_without_duplicates.loc[:,['CompFreq', 'CompTotal', 'NormalizedAnnualCompensation']]
```

Out[40]:

```
Yearly     6073
Monthly    4788
Weekly      331
Name: CompFreq, dtype: int64
```

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