
# 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 [24]:

```
import pandas as pd
```

Load the dataset into a dataframe.

In [25]:

```
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 [26]:

```
# your code goes here
df.duplicated().sum()
```

Out[26]:

```
154
```

In [28]:

```
df.duplicated(subset = "Respondent").sum()
```

Out[28]:

```
154
```

# Removing duplicates

Remove the duplicate rows from the dataframe.

In [34]:

```
# your code goes here
df = df.drop_duplicates()
```

Verify if duplicates were actually dropped.

In [35]:

```
# your code goes here
df.duplicated().sum()
```

Out[35]:

```
0
```

In [23]:

```

```

Out[23]:

```
0
```

# Finding Missing values

Find the missing values for all columns.

In [36]:

```
# your code goes here
df.columns[df.isnull().any()]
```

Out[36]:

```
Index(['OpenSource', 'Student', 'EdLevel', 'UndergradMajor', 'EduOther',
       'OrgSize', 'DevType', 'YearsCode', 'Age1stCode', 'YearsCodePro',
       'JobSat', 'MgrIdiot', 'MgrMoney', 'MgrWant', 'LastInt', 'FizzBuzz',
       'JobFactors', 'ResumeUpdate', 'CompTotal', 'CompFreq', 'ConvertedCo
mp',
       'WorkWeekHrs', 'WorkPlan', 'WorkChallenge', 'WorkRemote', 'WorkLo
c',
       'ImpSyn', 'CodeRev', 'CodeRevHrs', 'UnitTests', 'PurchaseHow',
       'PurchaseWhat', 'LanguageWorkedWith', 'LanguageDesireNextYear',
       'DatabaseWorkedWith', 'DatabaseDesireNextYear', 'PlatformWorkedWit
h',
       'PlatformDesireNextYear', 'WebFrameWorkedWith',
       'WebFrameDesireNextYear', 'MiscTechWorkedWith',
       'MiscTechDesireNextYear', 'DevEnviron', 'OpSys', 'Containers',
       'BlockchainOrg', 'BlockchainIs', 'BetterLife', 'ITperson', 'OffOn',
       'SocialMedia', 'Extraversion', 'ScreenName', 'SOVisit1st',
       'SOVisitFreq', 'SOVisitTo', 'SOFindAnswer', 'SOTimeSaved',
       'SOHowMuchTime', 'SOAccount', 'SOPartFreq', 'SOJobs', 'EntTeams',
       'WelcomeChange', 'SONewContent', 'Age', 'Gender', 'Trans', 'Sexuali
ty',
       'Ethnicity', 'Dependents', 'SurveyLength', 'SurveyEase'],
      dtype='object')
```

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

In [39]:

```
# your code goes here
df['Country'].isna().sum()
```

Out[39]:

```
0
```

# Imputing missing values

Find the value counts for the column WorkLoc.

In [17]:

```
# your code goes here
df['WorkLoc'].value_counts()
```

Out[17]:

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

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

In [41]:

```
#make a note of the majority value here, for future reference
majority = df['UndergradMajor'].value_counts().idxmin()
print(majority)
```

```
A health science (ex. nursing, pharmacy, radiology)
```

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

In [19]:

```
# 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 [20]:

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

Out[20]:

```
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 [46]:

```
# your code goes here
df.CompFreq.value_counts().index
```

Out[46]:

```
Index(['Yearly', 'Monthly', 'Weekly'], dtype='object')
```

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

In [49]:

```
df['CompFreq'].value_counts()
```

Out[49]:

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

Double click to see the **Hint**.

In [44]:

```
# your code goes here
anncomp=[]
def NAC():
    for x,y in zip(df['CompFreq'], df['CompTotal']):
        if x=='Monthly':
            anncomp.append(y*12)
        elif x=='Weekly':
            anncomp.append(y*52)
        else:
            anncomp.append(y)
NAC()

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

Out[44]:

```
100000.0
```

# 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 |