Exploratory Data Analysis Lab

Estimated time needed: **30** minutes

In this module you get to work with the cleaned dataset from the previous module.

In this assignment you will perform the task of exploratory data analysis. You will find out the distribution of data, presence of outliers and also determine the correlation between different columns in the dataset.

Objectives

In this lab you will perform the following:

- Identify the distribution of data in the dataset.
- Identify outliers in the dataset.
- Remove outliers from the dataset.
- Identify correlation between features in the dataset.

Hands on Lab

Import the pandas module.

```
import pandas as pd
import matplotlib
from matplotlib import pyplot as plt
import numpy as np
%pip install seaborn
import seaborn as sns
```

Load the dataset into a dataframe.

Read Data

We utilize the `pandas.read_csv()` function for reading CSV files. However, in this version of the lab, which operates on JupyterLite, the dataset needs to be downloaded to the interface using the provided code below.

The functions below will download the dataset into your browser:

```
from pyodide.http import pyfetch

async def download(url, filename):
    response = await pyfetch(url)
    if response.status == 200:
        with open(filename, "wb") as f:
            f.write(await response.bytes())
```

```
file_path = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeData/m2_survey_data.csv"
```

To obtain the dataset, utilize the download() function as defined above:

```
await download(file_path, "m2_survey_data.csv")
file_name="m2_survey_data.csv"
```

Utilize the Pandas method read_csv() to load the data into a dataframe.

```
df = pd.read_csv(file_name)
```

> Note: This version of the lab is working on JupyterLite, which requires the dataset to be downloaded to the interface.While working on the downloaded version of this notebook on their local machines(Jupyter Anaconda), the learners can simply **skip the steps above,** and simply use the URL directly in the `pandas.read_csv()` function. You can uncomment and run the statements in the cell below.

Distribution

Determine how the data is distributed

The column `ConvertedComp` contains Salary converted to annual USD salaries using the exchange rate on 2019-02-01.

This assumes 12 working months and 50 working weeks.

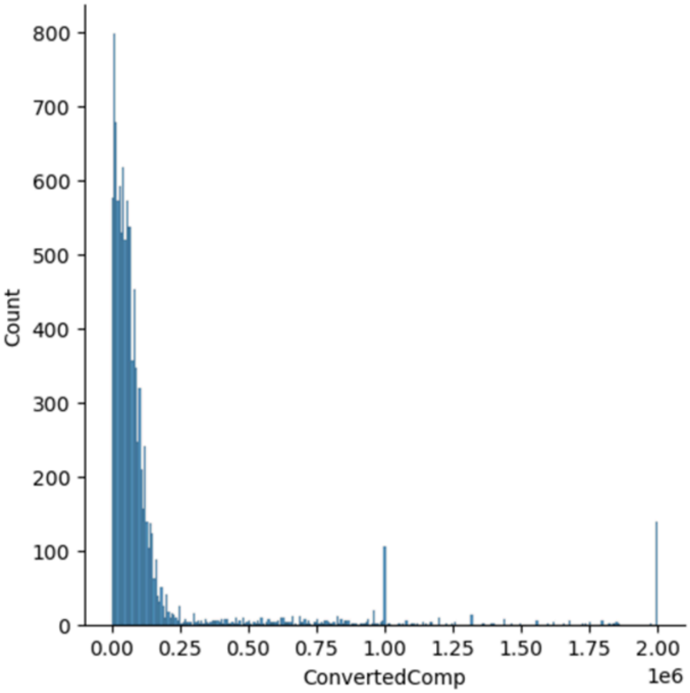
Plot the distribution curve for the column `ConvertedComp`.

[18]:

```
# your code goes here
sns.displot(df.ConvertedComp)
```

[18]:

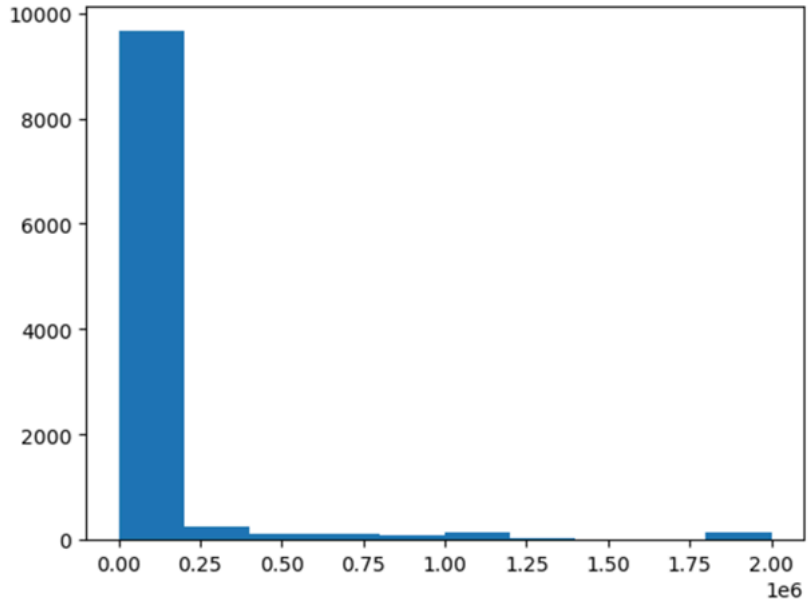
```
<seaborn.axisgrid.FacetGrid at 0x6bc83e8>
```



Plot the histogram for the column `ConvertedComp`.

```
[16]: # your code goes here
      plt.hist(df.ConvertedComp)
```

```
[16]: (array([9659.,  238.,  115.,  125.,   99.,  131.,   34.,   15.,   15.,
               151.]),
       array([      0.,  200000.,  400000.,  600000.,  800000., 1000000.,
              1200000., 1400000., 1600000., 1800000., 2000000.]),
       <BarContainer object of 10 artists>)
```



What is the median of the column `ConvertedComp` ?

```
[17]: # your code goes here
      df['ConvertedComp'].median()
```

```
[17]: 57745.0
```

How many responders identified themselves only as a **Man**?

```
[22]: # your code goes here
      df['Gender'].value_counts()
```

```
[22]: Gender
      Man                                                            10480
      Woman                                                            731
      Non-binary, genderqueer, or gender non-conforming                 63
      Man;Non-binary, genderqueer, or gender non-conforming             26
      Woman;Non-binary, genderqueer, or gender non-conforming           14
      Woman;Man                                                          9
      Woman;Man;Non-binary, genderqueer, or gender non-conforming        2
      Name: count, dtype: int64
```

Find out the median ConvertedComp of responders identified themselves only as a **Woman**?

```
[25]: # your code goes here
      df.ConvertedComp[df.Gender=='Woman'].median()
```

```
[25]: 57708.0
```

Give the five number summary for the column `Age` ?

Double click here for hint.

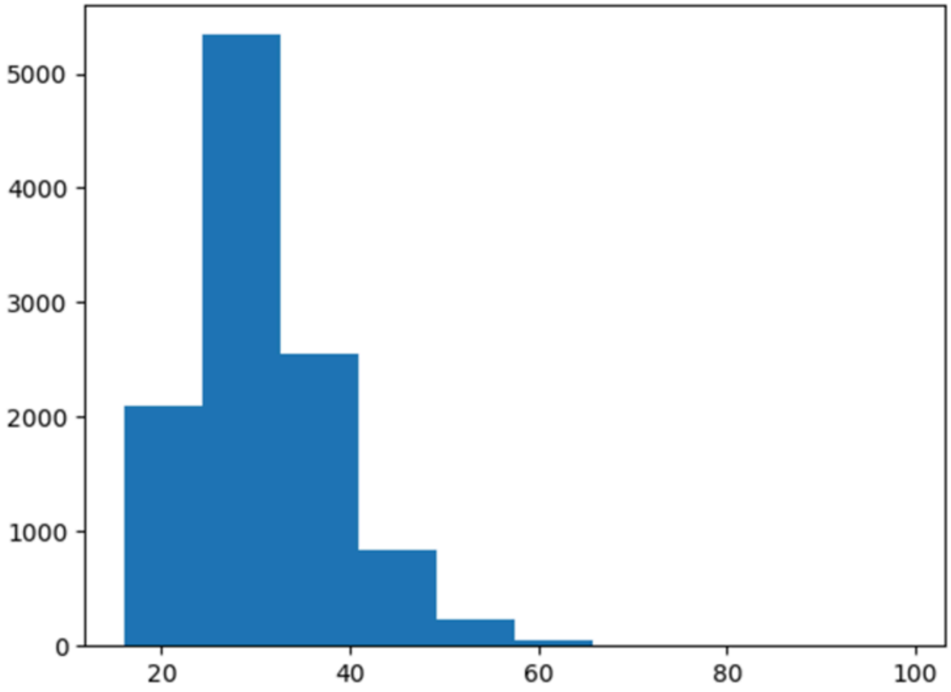
```
[28]: # your code goes here
      df['Age'].describe()
```

```
[28]: count    11111.000000
      mean        30.778895
      std          7.393686
      min         16.000000
      25%         25.000000
      50%         29.000000
      75%         35.000000
      max         99.000000
      Name: Age, dtype: float64
```

Plot a histogram of the column `Age`.

```
[29]: # your code goes here
      plt.hist(df.Age)
```

```
[29]: (array([2.094e+03, 5.337e+03, 2.557e+03, 8.420e+02, 2.250e+02, 4.900e+01,
              6.000e+00, 0.000e+00, 0.000e+00, 1.000e+00]),
       array([16. , 24.3, 32.6, 40.9, 49.2, 57.5, 65.8, 74.1, 82.4, 90.7, 99. ]),
       <BarContainer object of 10 artists>)
```



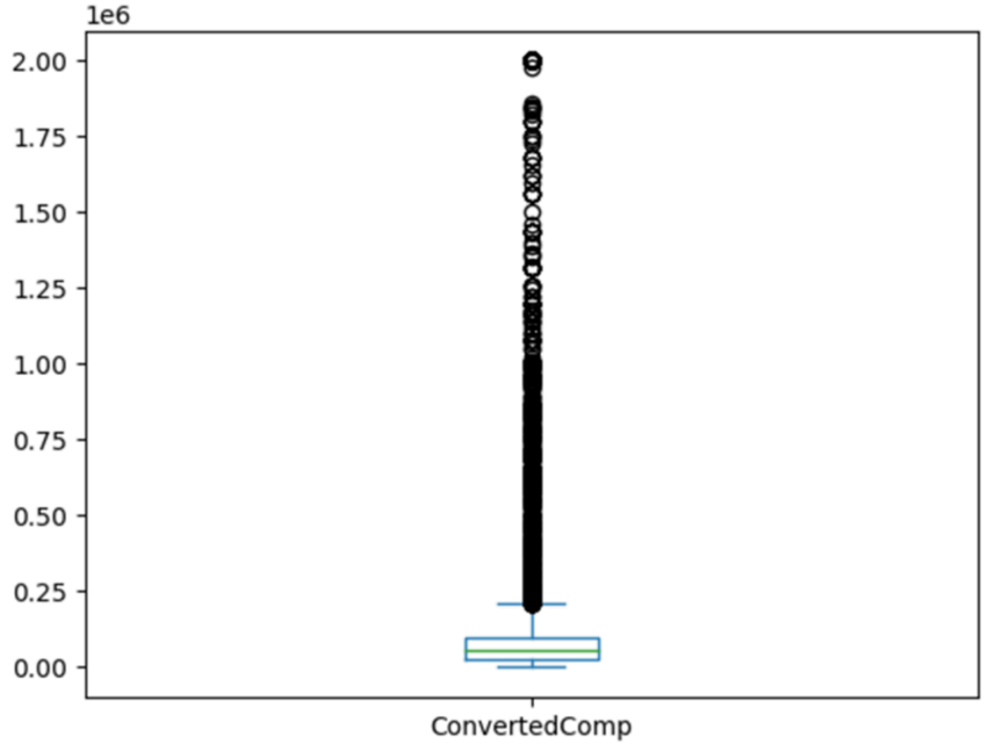
Outliers

Finding outliers

Find out if outliers exist in the column `ConvertedComp` using a box plot?

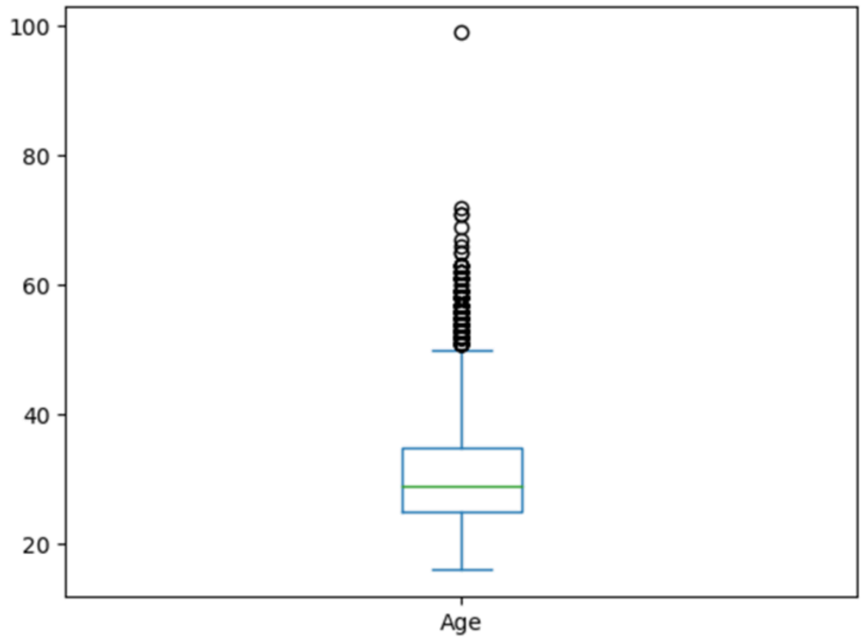
```
[50]: # your code goes here
      df['ConvertedComp'].plot(kind='box')
```

```
[50]: <AxesSubplot:>
```



```
[78]: df['Age'].plot(kind='box')
```

```
[78]: <AxesSubplot:>
```



Find out the Inter Quartile Range for the column `ConvertedComp`.

63]:

```python
# your code goes here
df['ConvertedComp'].dropna(axis=0, inplace=True)

q1 = df['ConvertedComp'].quantile(0.25)
q3 = df['ConvertedComp'].quantile(0.75)
IQR = q3 - q1
IQR
```

[63]:

```
73132.0
```

Find out the upper and lower bounds.

[65]:

```python
# your code goes here
Lower = q1 - (IQR*1.5)
Upper = q3 + (IQR*1.5)

print('Lower Bound: ', Lower)
print('Upper Bound: ', Upper)
```

```
Lower Bound:  -82830.0
Upper Bound:  209698.0
```

Identify how many outliers are there in the `ConvertedComp` column.

[71]:

```python
# your code goes here
Outliers_below = df['ConvertedComp'].lt(Lower).sum()
Outliers_above = df['ConvertedComp'].gt(Upper).sum()
print('Outliers below: ', Outliers_below)
print('Outliers above: ', Outliers_above)
```

```
Outliers below:  0
Outliers above:  879
```

Create a new dataframe by removing the outliers from the `ConvertedComp` column.

[77]:

```python
# your code goes here
df_remove_outliers = df[(df['ConvertedComp'] >= Lower) & (df['ConvertedComp'] <= Upper)]

df_remove_outliers['ConvertedComp'].describe()
```

```
[77]: count      9703.000000
      mean      59883.208389
      std       43394.336755
      min           0.000000
      25%       24060.000000
      50%       52704.000000
      75%       85574.500000
      max      209356.000000
      Name: ConvertedComp, dtype: float64
```

Correlation

Finding correlation

Find the correlation between `Age` and all other numerical columns.

```
[75]: # your code goes here
      df.corr(numeric_only=True)['Age']
```

```
[75]: Respondent       0.004041
      CompTotal        0.006970
      ConvertedComp    0.105386
      WorkWeekHrs      0.036518
      CodeRevHrs      -0.020469
      Age              1.000000
      Name: Age, dtype: float64
```