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.

In [36]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',85)
pd.set_option('display.width', 1000)
```

Load the dataset into a dataframe.

In [32]:
```
#df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeData/m2_survey_data.csv")

df = pd.read_csv("m2_survey_data.csv")
```

In [15]:
```
df.describe()
```

Out[15]:

	Respondent	CompTotal	ConvertedComp	WorkWeekHrs	CodeRevHrs	Age
count	11398.000000	1.058900e+04	1.058200e+04	11276.000000	8972.000000	11111.000000
mean	12490.392437	7.570477e+05	1.315967e+05	42.064606	4.781071	30.778895
std	7235.461999	9.705598e+06	2.947865e+05	24.672741	4.567060	7.393686
min	4.000000	0.000000e+00	0.000000e+00	3.000000	0.000000	16.000000
25%	6264.250000	2.500000e+04	2.686800e+04	40.000000	2.000000	25.000000
50%	12484.000000	6.500000e+04	5.774500e+04	40.000000	4.000000	29.000000
75%	18784.750000	1.200000e+05	1.000000e+05	43.000000	5.000000	35.000000
max	25142.000000	7.000000e+08	2.000000e+06	1012.000000	99.000000	99.000000

In [16]:
```
df.isna().sum().sum()
```

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

Distribution

Determine how the data is distributed

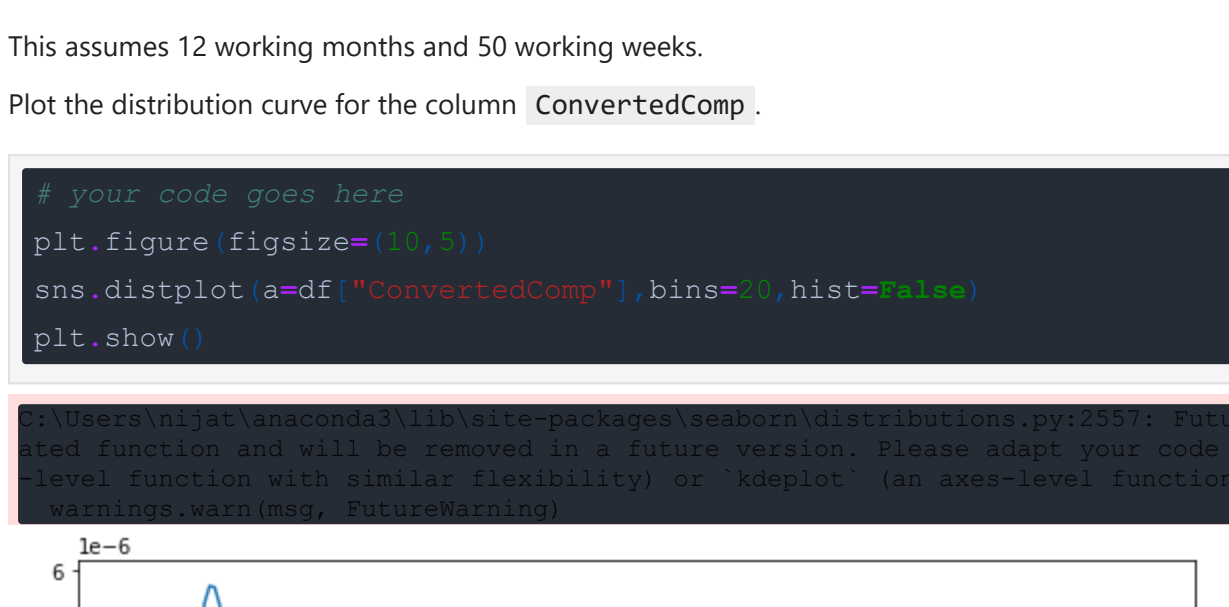
The column `ConvertedComp` contains Salary converted to annual USD salaries using the exchange rate on 2019-02-01.

This assumes 12 working months and 50 working weeks.

Plot the distribution curve for the column `ConvertedComp`.

In [37]:
```
# your code goes here
plt.figure(figsize=(10,5))
sns.distplot(a=df["ConvertedComp"],bins=20,hist=False)
plt.show()
```

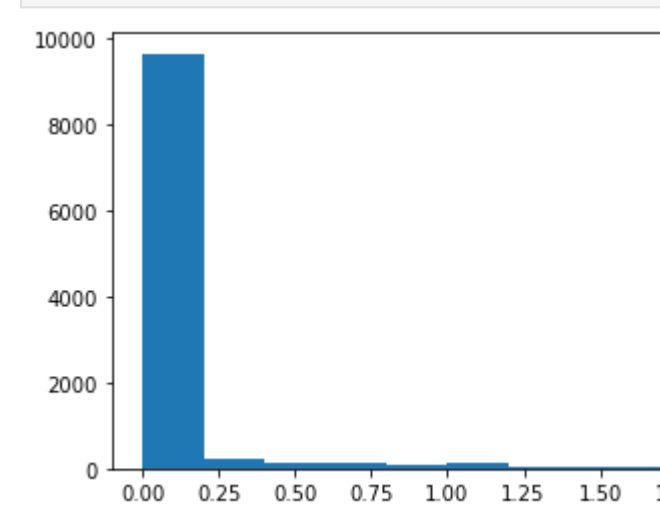
```
C:\Users\nijat\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```



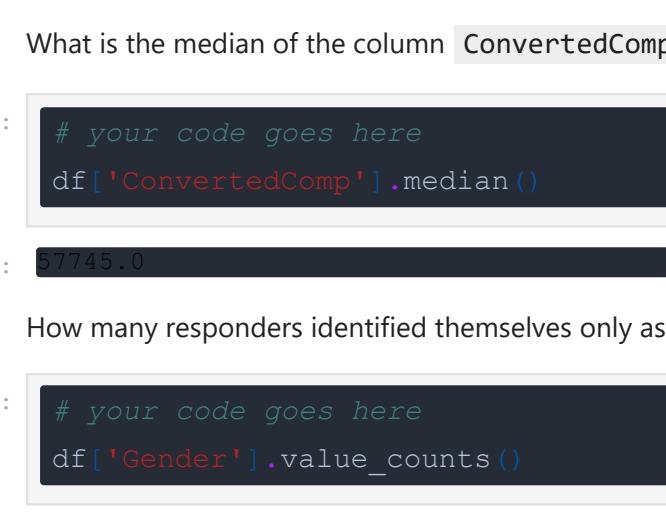
Plot the histogram for the column `ConvertedComp`.

In [35]:
```
# your code goes here
sns.histplot(df['ConvertedComp'])
```

Out[35]:
```
<AxesSubplot:xlabel='ConvertedComp', ylabel='Count'>
```



In [39]:
```
plt.hist(df['ConvertedComp'])
plt.show()
```



What is the median of the column `ConvertedComp`?

In [41]:
```
# your code goes here
df['ConvertedComp'].median()
```

Out[41]:
```
57745.0
```

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

In [51]:
```
# your code goes here
df['Gender'].value_counts()
```

Out[51]:
```
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: Gender, dtype: int64
```

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

In [54]:
```
# your code goes here
woman = df[df['Gender']=="Woman"]
woman['ConvertedComp'].median()
```

Out[54]:
```
57708.0
```

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

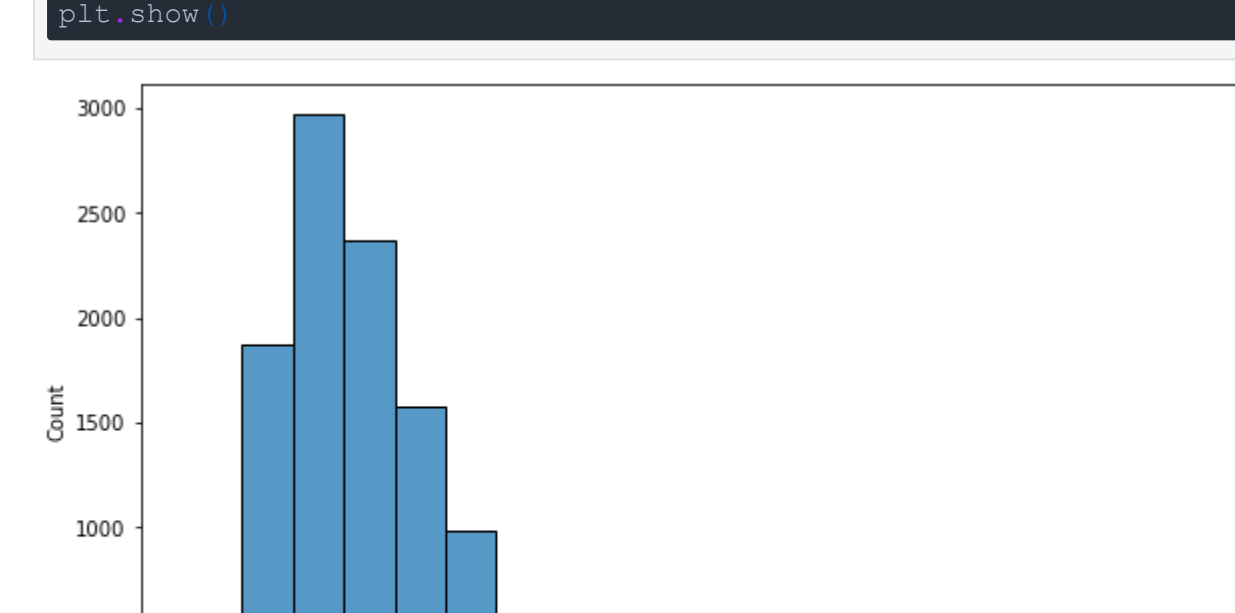
Double click here for hint.

In [55]:
```
# your code goes here
df['Age'].describe()
```

Out[55]:
```
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`.

In [63]:
```
# your code goes here
plt.figure(figsize=(10,6))
sns.histplot(df['Age'],bins=20)
plt.show()
```

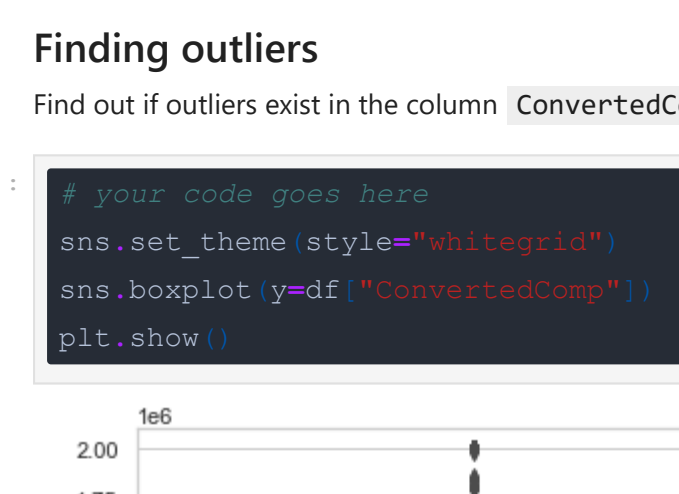


Outliers

Finding outliers

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

In [72]:
```
# your code goes here
sns.set_theme(style="whitegrid")
sns.boxplot(y=df["ConvertedComp"])
plt.show()
```



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

In [78]:
```
# your code goes here
q75,q25 = np.percentile(df['ConvertedComp'],[75,25])
inq = q75 - q25
inq
```

Out[78]:
```
nan
```

In [87]:
```
x = df['ConvertedComp'].describe()
inq = x['75%'] - x['25%']
print("The intergartil is :",inq)
```

```
The intergartil is : 73132.0
```

Second method

In [92]:
```
Q1 = df["ConvertedComp"].quantile(0.25)
Q3 = df["ConvertedComp"].quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

```
73132.0
```

Find out the upper and lower bounds.

In [91]:
```
# your code goes here
print("the lower bound is:",x['min'],"And the Upper bound is:",x['max'])
```

```
the lower bound is: 0.0 And the Upper bound is: 2000000.0
```

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

lets first find the IQR

In [94]:
```
Q1 = df["ConvertedComp"].quantile(0.25)
Q3 = df["ConvertedComp"].quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

```
73132.0
```

In [ ]:
```

```

In [93]:
```
# your code goes here
outlier = ((df['ConvertedComp'] < (Q1 - 1.5 * IQR)) | (df['ConvertedComp'] > (Q3 + 1.5 * IQR))).sum()
outlier
```

Out[93]:
```
879
```

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

In [99]:
```
# your code goes here
mask = ((df['ConvertedComp'] < (Q1 - 1.5 * IQR)) | (df['ConvertedComp'] > (Q3 + 1.5 * IQR)))
df[mask]=np.nan
```

In [100…
```
df.head()
```

Out[100…

	Respondent	MainBranch	Hobbyist	OpenSourcer	OpenSource	Employment	Country	Student	EdLevel	UndergradMajor	Edu
0	4.0	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...	Tak online c program or sc
1	9.0	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...	Tak online c program or sc
2	13.0	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...	Tak online c program or sc
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4	17.0	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...	Tak online c program or sc

Correlation

Finding correlation

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

In [101…
```
# your code goes here
df.corr()
```

Out[101…

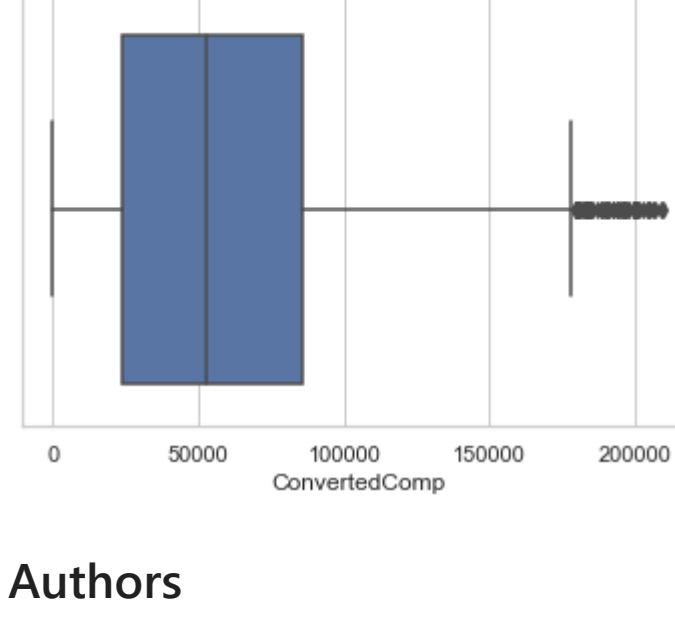
	Respondent	CompTotal	ConvertedComp	WorkWeekHrs	CodeRevHrs	Age
Respondent	1.000000	-0.019364	0.010878	-0.015275	0.002980	0.003950
CompTotal	-0.019364	1.000000	-0.063561	0.004975	0.017536	0.006371
ConvertedComp	0.010878	-0.063561	1.000000	0.034351	-0.088934	0.401821
WorkWeekHrs	-0.015275	0.004975	0.034351	1.000000	0.031963	0.037452
CodeRevHrs	0.002980	0.017536	-0.088934	0.031963	1.000000	-0.017961
Age	0.003950	0.006371	0.401821	0.037452	-0.017961	1.000000

In [102…
```
outlier = ((df['ConvertedComp'] < (Q1 - 1.5 * IQR)) | (df['ConvertedComp'] > (Q3 + 1.5 * IQR))).sum()
outlier
```

Out[102…
```
0
```

In [105…
```
sns.set_theme(style="whitegrid")
sns.boxplot(df['ConvertedComp'])
plt.show()
```

```
C:\Users\nijat\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



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