In [1]:

```python
# Library Imports

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

In [2]:

```python
# Dataset Imports

data = pd.read_csv('Lab1-Data.csv')
data.head(10)
```

Out[2]:

	job_category	race_ethnicity	gender	count	percentage
0	All workers	White	Male	268883	41.257252
1	All workers	White	Female	105560	16.197065
2	All workers	Black_or_African American	Male	17508	2.686417
3	All workers	Black_or_African American	Female	11479	1.761331
4	All workers	Asian	Male	125347	19.233171
5	All workers	Asian	Female	58049	8.907005
6	All workers	Hispanic_or_Latino	Male	32201	4.940903
7	All workers	Hispanic_or_Latino	Female	15512	2.380152
8	All workers	All	Male	454813	69.786244
9	All workers	All	Female	196910	30.213756

In [3]:

```python
# Viewing Dataset Statistically

data.describe()
```

Out[3]:

	count	percentage
count	44.000000	44.000000
mean	75269.954545	27.084140
std	133360.750559	31.904184
min	53.000000	0.427075
25%	3452.250000	2.320972
50%	16510.000000	15.423271
75%	91864.500000	39.309757
max	651723.000000	100.000000

In [4]:

```python
# dropping duplicate values

data = data.drop_duplicates()
data
```

Out[4]:

	job_category	race_ethnicity	gender	count	percentage
0	All workers	White	Male	268883	41.257252
1	All workers	White	Female	105560	16.197065
2	All workers	Black_or_African American	Male	17508	2.686417
3	All workers	Black_or_African American	Female	11479	1.761331
4	All workers	Asian	Male	125347	19.233171
5	All workers	Asian	Female	58049	8.907005
6	All workers	Hispanic_or_Latino	Male	32201	4.940903
7	All workers	Hispanic_or_Latino	Female	15512	2.380152
8	All workers	All	Male	454813	69.786244
9	All workers	All	Female	196910	30.213756
10	All workers	Totals	Both	651723	100.000000
11	Executives	White	Male	7282	58.678485
12	Executives	White	Female	1818	14.649476
13	Executives	Black_or_African American	Male	120	0.966962
14	Executives	Black_or_African American	Female	53	0.427075
15	Executives	Asian	Male	2023	16.301370
16	Executives	Asian	Female	556	4.500000
17	Executives	Hispanic_or_Latino	Male	266	2.143433
18	Executives	Hispanic_or_Latino	Female	103	0.829976
19	Executives	All	Male	9824	79.161966
20	Executives	All	Female	2586	20.838034
21	Executives	Totals	Both	12410	100.000000
22	Managers	White	Male	48311	46.479253
23	Managers	White	Female	18935	18.217065
24	Managers	Black_or_African American	Male	1575	1.515283
25	Managers	Black_or_African American	Female	978	0.940918
26	Managers	Asian	Male	18563	17.859170
27	Managers	Asian	Female	8084	7.777489
28	Managers	Hispanic_or_Latino	Male	3741	3.599157
29	Managers	Hispanic_or_Latino	Female	1642	1.579742
30	Managers	All	Male	73526	70.738207
31	Managers	All	Female	30415	29.261793
32	Managers	Totals	Both	103941	100.000000
33	Professionals	White	Male	133311	38.660592
34	Professionals	White	Female	47505	13.776593
35	Professionals	Black_or_African American	Male	6301	1.827309
36	Professionals	Black_or_African American	Female	3756	1.089251
37	Professionals	Asian	Male	89365	25.916120
38	Professionals	Asian	Female	39902	11.571700
39	Professionals	Hispanic_or_Latino	Male	11820	3.427836
40	Professionals	Hispanic_or_Latino	Female	5533	1.604587
41	Professionals	All	Male	245461	71.184430
42	Professionals	All	Female	99363	28.815570
43	Professionals	Totals	Both	344824	100.000000

In [5]:

```python
# Checking the null values

data.isnull()
```

Out[5]:

	job_category	race_ethnicity	gender	count	percentage
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False
9	False	False	False	False	False
10	False	False	False	False	False
11	False	False	False	False	False
12	False	False	False	False	False
13	False	False	False	False	False
14	False	False	False	False	False
15	False	False	False	False	False
16	False	False	False	False	False
17	False	False	False	False	False
18	False	False	False	False	False
19	False	False	False	False	False
20	False	False	False	False	False
21	False	False	False	False	False
22	False	False	False	False	False
23	False	False	False	False	False
24	False	False	False	False	False
25	False	False	False	False	False
26	False	False	False	False	False
27	False	False	False	False	False
28	False	False	False	False	False
29	False	False	False	False	False
30	False	False	False	False	False
31	False	False	False	False	False
32	False	False	False	False	False
33	False	False	False	False	False
34	False	False	False	False	False
35	False	False	False	False	False
36	False	False	False	False	False
37	False	False	False	False	False
38	False	False	False	False	False
39	False	False	False	False	False
40	False	False	False	False	False
41	False	False	False	False	False
42	False	False	False	False	False
43	False	False	False	False	False

In [6]:

```python
# Statistically checking the missing values

data.isnull().sum()
```

Out[6]:

```
job_category      0
race_ethnicity    0
gender            0
count             0
percentage        0
dtype: int64
```

In [7]:

```python
data.dropna(how='any',inplace=True)
```

Splitting the Data

In [8]:

```python
# Splitting dataset

X = data[['job_category', 'count', "percentage"]].values
Y = data['gender'].values
Z = data['race_ethnicity'].values
print(X)
print(Y)
print(Z)
```

```
[['All workers' 268883 41.25725193]
 ['All workers' 105560 16.19706532]
 ['All workers' 17508 2.686417389]
 ['All workers' 11479 1.761331118]
 ['All workers' 125347 19.23317115]
 ['All workers' 58049 8.90700497]
 ['All workers' 32201 4.940902807]
 ['All workers' 15512 2.380152304]
 ['All workers' 454813 69.78624354]
 ['All workers' 196910 30.21375646]
 ['All workers' 651723 100.0]
 ['Executives' 7282 58.67848509]
 ['Executives' 1818 14.64947623]
 ['Executives' 120 0.966962127]
 ['Executives' 53 0.42707494]
 ['Executives' 2023 16.30136986]
 ['Executives' 556 4.5]
 ['Executives' 266 2.143432716]
 ['Executives' 103 0.829975826]
 ['Executives' 9824 79.16196616]
 ['Executives' 2586 20.83803384]
 ['Executives' 12410 100.0]
 ['Managers' 48311 46.47925265]
 ['Managers' 18935 18.21706545]
 ['Managers' 1575 1.515282708]
 ['Managers' 978 0.940918406]
 ['Managers' 18563 17.85917011]
 ['Managers' 8084 7.777489152]
 ['Managers' 3741 3.599157214]
 ['Managers' 1642 1.579742354]
 ['Managers' 73526 70.73820725]
 ['Managers' 30415 29.26179275]
 ['Managers' 103941 100.0]
 ['Professionals' 133311 38.66059207]
 ['Professionals' 47505 13.77659328]
 ['Professionals' 6301 1.827309004]
 ['Professionals' 3756 1.089251328]
 ['Professionals' 89365 25.91611953]
 ['Professionals' 39902 11.57170035]
 ['Professionals' 11820 3.427835649]
 ['Professionals' 5533 1.604586688]
 ['Professionals' 245461 71.18443032]
 ['Professionals' 99363 28.81556968]
 ['Professionals' 344824 100.0]]
['Male' 'Female' 'Male' 'Female' 'Male' 'Female' 'Male' 'Female' 'Male'
 'Female' 'Both' 'Male' 'Female' 'Male' 'Female' 'Male' 'Female' 'Male'
 'Female' 'Male' 'Female' 'Both' 'Male' 'Female' 'Male' 'Female' 'Male'
 'Female' 'Male' 'Female' 'Male' 'Female' 'Both' 'Male' 'Female' 'Male'
 'Female' 'Male' 'Female' 'Male' 'Female' 'Male' 'Female' 'Both']
['White' 'White' 'Black_or_African American' 'Black_or_African American'
 'Asian' 'Asian' 'Hispanic_or_Latino' 'Hispanic_or_Latino' 'All' 'All'
 'Totals' 'White' 'White' 'Black_or_African American'
 'Black_or_African American' 'Asian' 'Asian' 'Hispanic_or_Latino'
 'Hispanic_or_Latino' 'All' 'All' 'Totals' 'White' 'White'
 'Black_or_African American' 'Black_or_African American' 'Asian' 'Asian'
 'Hispanic_or_Latino' 'Hispanic_or_Latino' 'All' 'All' 'Totals' 'White'
 'White' 'Black_or_African American' 'Black_or_African American' 'Asian'
 'Asian' 'Hispanic_or_Latino' 'Hispanic_or_Latino' 'All' 'All' 'Totals']
```

Categorical Data Encodding

In [9]:

```python
# Handling Categorical Data

from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
ct = ColumnTransformer(transformers=[('enconder', OneHotEncoder(), [0])], remainder='passthrough')
```

In [10]:

```python
# One-Hot Encoding Categorical Data

X = np.array(ct.fit_transform(X))
print(X)
```

```
[[1.0 0.0 0.0 0.0 268883 41.25725193]
 [1.0 0.0 0.0 0.0 105560 16.19706532]
 [1.0 0.0 0.0 0.0 17508 2.686417389]
 [1.0 0.0 0.0 0.0 11479 1.761331118]
 [1.0 0.0 0.0 0.0 125347 19.23317115]
 [1.0 0.0 0.0 0.0 58049 8.90700497]
 [1.0 0.0 0.0 0.0 32201 4.940902807]
 [1.0 0.0 0.0 0.0 15512 2.380152304]
 [1.0 0.0 0.0 0.0 454813 69.78624354]
 [1.0 0.0 0.0 0.0 196910 30.21375646]
 [1.0 0.0 0.0 0.0 651723 100.0]
 [0.0 1.0 0.0 0.0 7282 58.67848509]
 [0.0 1.0 0.0 0.0 1818 14.64947623]
 [0.0 1.0 0.0 0.0 120 0.966962127]
 [0.0 1.0 0.0 0.0 53 0.42707494]
 [0.0 1.0 0.0 0.0 2023 16.30136986]
 [0.0 1.0 0.0 0.0 556 4.5]
 [0.0 1.0 0.0 0.0 266 2.143432716]
 [0.0 1.0 0.0 0.0 103 0.829975826]
 [0.0 1.0 0.0 0.0 9824 79.16196616]
 [0.0 1.0 0.0 0.0 2586 20.83803384]
 [0.0 1.0 0.0 0.0 12410 100.0]
 [0.0 0.0 1.0 0.0 48311 46.47925265]
 [0.0 0.0 1.0 0.0 18935 18.21706545]
 [0.0 0.0 1.0 0.0 1575 1.515282708]
 [0.0 0.0 1.0 0.0 978 0.940918406]
 [0.0 0.0 1.0 0.0 18563 17.85917011]
 [0.0 0.0 1.0 0.0 8084 7.777489152]
 [0.0 0.0 1.0 0.0 3741 3.599157214]
 [0.0 0.0 1.0 0.0 1642 1.579742354]
 [0.0 0.0 1.0 0.0 73526 70.73820725]
 [0.0 0.0 1.0 0.0 30415 29.26179275]
 [0.0 0.0 1.0 0.0 103941 100.0]
 [0.0 0.0 0.0 1.0 133311 38.66059207]
 [0.0 0.0 0.0 1.0 47505 13.77659328]
 [0.0 0.0 0.0 1.0 6301 1.827309004]
 [0.0 0.0 0.0 1.0 3756 1.089251328]
 [0.0 0.0 0.0 1.0 89365 25.91611953]
 [0.0 0.0 0.0 1.0 39902 11.57170035]
 [0.0 0.0 0.0 1.0 11820 3.427835649]
 [0.0 0.0 0.0 1.0 5533 1.604586688]
 [0.0 0.0 0.0 1.0 245461 71.18443032]
 [0.0 0.0 0.0 1.0 99363 28.81556968]
 [0.0 0.0 0.0 1.0 344824 100.0]]
```

In [11]:

```python
Y = le.fit_transform(Y)
print(Y)
```

```
[2 1 2 1 2 1 2 1 2 1 0 2 1 2 1 2 1 2 1 2 1 0 2 1 2 1 2 1 2 1 2 1 0 2 1 2 1
 2 1 2 1 2 1 0]
```

In [12]:

```python
Z = le.fit_transform(Z)
print(Z)
```

```
[5 5 2 2 1 1 3 3 0 0 4 5 5 2 2 1 1 3 3 0 0 4 5 5 2 2 1 1 3 3 0 0 4 5 5 2 2
 1 1 3 3 0 0 4]
```

Splitting into Test and Train

In [13]:

```python
from sklearn.model_selection import train_test_split
```

In [14]:

```python
# Splitting Test and Train Data on Y as dependent data

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

In [15]:

```python
# Splitting Test and Train Data on Z as dependent data

X_train, X_test, Z_train, Z_test = train_test_split(X, Z, test_size=0.2, random_state=2)
```

In [16]:

```python
print(X_train)
print(Y_train)
print(Z_train)
print(X_test)
print(Y_test)
print(Z_test)
```

```
[[0.0 0.0 0.0 1.0 11820 3.427835649]
 [0.0 0.0 1.0 0.0 18935 18.21706545]
 [0.0 0.0 0.0 1.0 245461 71.18443032]
 [0.0 0.0 1.0 0.0 8084 7.777489152]
 [0.0 0.0 1.0 0.0 978 0.940918406]
 [0.0 1.0 0.0 0.0 1818 14.64947623]
 [0.0 1.0 0.0 0.0 266 2.143432716]
 [1.0 0.0 0.0 0.0 105560 16.19706532]
 [0.0 0.0 0.0 1.0 344824 100.0]
 [0.0 0.0 1.0 0.0 1575 1.515282708]
 [0.0 0.0 1.0 0.0 103941 100.0]
 [0.0 1.0 0.0 0.0 9824 79.16196616]
 [0.0 0.0 0.0 1.0 3756 1.089251328]
 [1.0 0.0 0.0 0.0 651723 100.0]
 [1.0 0.0 0.0 0.0 125347 19.23317115]
 [1.0 0.0 0.0 0.0 32201 4.940902807]
 [1.0 0.0 0.0 0.0 11479 1.761331118]
 [0.0 0.0 0.0 1.0 89365 25.91611953]
 [1.0 0.0 0.0 0.0 58049 8.90700497]
 [0.0 0.0 1.0 0.0 73526 70.73820725]
 [0.0 0.0 1.0 0.0 3741 3.599157214]
 [0.0 1.0 0.0 0.0 2586 20.83803384]
 [0.0 0.0 1.0 0.0 18563 17.85917011]
 [0.0 0.0 0.0 1.0 6301 1.827309004]
 [0.0 1.0 0.0 0.0 12410 100.0]
 [0.0 0.0 0.0 1.0 39902 11.57170035]
 [0.0 0.0 1.0 0.0 30415 29.26179275]
 [0.0 0.0 0.0 1.0 47505 13.77659328]
 [1.0 0.0 0.0 0.0 15512 2.380152304]
 [0.0 1.0 0.0 0.0 7282 58.67848509]
 [0.0 1.0 0.0 0.0 103 0.829975826]
 [0.0 0.0 1.0 0.0 48311 46.47925265]
 [1.0 0.0 0.0 0.0 454813 69.78624354]
 [0.0 1.0 0.0 0.0 2023 16.30136986]
 [0.0 0.0 0.0 1.0 5533 1.604586688]]
[2 1 2 1 1 1 2 1 0 2 0 2 1 0 2 2 1 2 1 2 2 1 2 2 0 1 1 1 1 2 1 2 2 2 1]
[3 5 0 1 2 5 3 5 4 2 4 0 2 4 1 3 2 1 1 0 3 0 1 2 4 1 0 5 3 5 3 5 0 1 3]
[[0.0 0.0 1.0 0.0 1642 1.579742354]
 [0.0 0.0 0.0 1.0 133311 38.66059207]
 [0.0 1.0 0.0 0.0 53 0.42707494]
 [1.0 0.0 0.0 0.0 196910 30.21375646]
 [0.0 0.0 0.0 1.0 99363 28.81556968]
 [1.0 0.0 0.0 0.0 268883 41.25725193]
 [1.0 0.0 0.0 0.0 17508 2.686417389]
 [0.0 1.0 0.0 0.0 556 4.5]
 [0.0 1.0 0.0 0.0 120 0.966962127]]
[1 2 1 1 1 2 2 1 2]
[3 5 2 0 0 5 2 1 2]
```

Standardization

In [17]:

```python
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
```

In [18]:

```python
X_train[:, 5:] = sc.fit_transform(X_train[:, 5:])
```

In [19]:

```python
print(X_test)
```

```
[[0.0 0.0 1.0 0.0 1642 1.579742354]
 [0.0 0.0 0.0 1.0 133311 38.66059207]
 [0.0 1.0 0.0 0.0 53 0.42707494]
 [1.0 0.0 0.0 0.0 196910 30.21375646]
 [0.0 0.0 0.0 1.0 99363 28.81556968]
 [1.0 0.0 0.0 0.0 268883 41.25725193]
 [1.0 0.0 0.0 0.0 17508 2.686417389]
 [0.0 1.0 0.0 0.0 556 4.5]
 [0.0 1.0 0.0 0.0 120 0.966962127]]
```

In [20]:

```python
print(X_train)
```

```
[[0.0 0.0 0.0 1.0 11820 -0.7795949892757811]
 [0.0 0.0 1.0 0.0 18935 -0.342214376417952]
 [0.0 0.0 0.0 1.0 245461 1.2242565500625042]
 [0.0 0.0 1.0 0.0 8084 -0.6509571805149282]
 [0.0 0.0 1.0 0.0 978 -0.8531437408001161]
 [0.0 1.0 0.0 0.0 1818 -0.44772320821412]
 [0.0 1.0 0.0 0.0 266 -0.8175802629885438]
 [1.0 0.0 0.0 0.0 105560 -0.40195439714388287]
 [0.0 0.0 0.0 1.0 344824 2.076455866461907]
 [0.0 0.0 1.0 0.0 1575 -0.83615733827661]
 [0.0 0.0 1.0 0.0 103941 2.076455866461907]
 [0.0 1.0 0.0 0.0 9824 1.460186315286557]
 [0.0 0.0 0.0 1.0 3756 -0.8487569035405473]
 [1.0 0.0 0.0 0.0 651723 2.076455866461907]
 [1.0 0.0 0.0 0.0 125347 -0.31216379638272373]
 [1.0 0.0 0.0 0.0 32201 -0.7348471389978479]
 [1.0 0.0 0.0 0.0 11479 -0.8288806372038322]
 [0.0 0.0 0.0 1.0 89365 -0.11452050477980388]
 [1.0 0.0 0.0 0.0 58049 -0.6175525795075081]
 [0.0 0.0 1.0 0.0 73526 1.2110598303919027]
 [0.0 0.0 1.0 0.0 3741 -0.774528279778144]
 [0.0 1.0 0.0 0.0 2586 -0.2647011606437482]
 [0.0 0.0 1.0 0.0 18563 -0.35279886832554114]
 [0.0 0.0 0.0 1.0 6301 -0.8269293896674803]
 [0.0 1.0 0.0 0.0 12410 2.076455866461907]
 [0.0 0.0 0.0 1.0 39902 -0.538746170109162]
 [0.0 0.0 1.0 0.0 30415 -0.015574675749093906]
 [0.0 0.0 0.0 1.0 47505 -0.47353808057470326]
 [1.0 0.0 0.0 0.0 15512 -0.8105794549770345]
 [0.0 1.0 0.0 0.0 7282 0.8544024019652187]
 [0.0 1.0 0.0 0.0 103 -0.8564247861476669]
 [0.0 0.0 1.0 0.0 48311 0.4936190594383802]
 [1.0 0.0 0.0 0.0 454813 1.182906202616773]
 [0.0 1.0 0.0 0.0 2023 -0.3988696669555691]
 [0.0 0.0 0.0 1.0 5533 -0.8335162386366273]]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: