Name: John Rome A. Belocora

Section: CPE22S3

```
pip install ucimlrepo
```

```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.6-py3-none-any.whl (8.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.6
```

```
from ucimlrepo import fetch_ucirepo

# fetch dataset
census_income = fetch_ucirepo(id=20)

# data (as pandas dataframes)
X = census_income.data.features
y = census_income.data.targets
```

Data Wrangling

```
import pandas as pd
import numpy as np
```

```
X
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife
...	...	...	...	...	...	...	...	...

Next steps: View recommended plots

```
y
```

	income
0	<=50K
1	<=50K
2	<=50K
3	<=50K
4	<=50K
...	...
48837	<=50K.
48838	<=50K.
48839	<=50K.
48840	<=50K.
48841	>50K.

48842 rows × 1 columns

Next steps: View recommended plots

```
#Concatinating the X and Y variable to merge them into One dataframe.
income_data = pd.concat([X, y], axis=1)
income_data
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife
...	...	...	...	...	...	...	...	...

Next steps: View recommended plots

```
#Renaming the columns with '-' and changing it into '_'
income_data.rename(columns = {'education-num' : 'education_num',
                  'marital-status' : 'marital_status',
                  'capital-gain' : 'capital_gain',
                  'capital-loss' : 'capital_loss',
                  'hours-per-week' : 'hours_per_week',
                  'native-country' : 'country'
                  }, inplace = True)
income_data
```

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	rel
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	N
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	N
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	
...	...	...	...	...	...	...	...	
48837	39	Private	215419	Bachelors	13	Divorced	Prof-specialty	N
48838	64	NaN	321403	HS-grad	9	Widowed	NaN	Ot

Next steps: View recommended plots

```
income_data.income.unique()
```

```
array(['<=50K', '>50K', '<=50K.', '>50K.'], dtype=object)
```

```
#Replacing duplicated values with the right variable name
income_data.income.replace({'<=50K.': '<=50K','>50K.' : '>50K'},  inplace = True)
income_data.income.unique()
```

```
array(['<=50K', '>50K'], dtype=object)
```

```
# New output
income_data.income.unique()
```

```
array(['<=50K', '>50K'], dtype=object)
```

```
income_data.age.unique()
```

```
array([39, 50, 38, 53, 28, 37, 49, 52, 31, 42, 30, 23, 32, 40, 34, 25, 43,
       54, 35, 59, 56, 19, 20, 45, 22, 48, 21, 24, 57, 44, 41, 29, 18, 47,
       46, 36, 79, 27, 67, 33, 76, 17, 55, 61, 70, 64, 71, 68, 66, 51, 58,
       26, 60, 90, 75, 65, 77, 62, 63, 80, 72, 74, 69, 73, 81, 78, 88, 82,
       83, 84, 85, 86, 87, 89])
```

```
# Dropping the fnlwgt column
income_data.drop('fnlwgt', axis=1, inplace = True)
income_data
```

	age	workclass	education	education_num	marital_status	occupation	relationshi
0	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-fami
1	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husban
2	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-fami
3	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husban
4	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Wif
...	...	...	...	...	...	...	.
48837	39	Private	Bachelors	13	Divorced	Prof-specialty	Not-in-fami
48838	64	NaN	HS-grad	9	Widowed	NaN	Other-relativ

Next steps: View recommended plots

Age Data

```
# Creating a new Dataframe for the data of the age
age_data = income_data.copy()
age_data
```

	age	workclass	education	education_num	marital_status	occupation	relationshi
0	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-fami
1	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husban
2	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-fami
3	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husban
4	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Wif
...	...	...	...	...	...	...	.
48837	39	Private	Bachelors	13	Divorced	Prof-specialty	Not-in-fami
48838	64	NaN	HS-grad	9	Widowed	NaN	Other-relativ

Next steps: View recommended plots

```
#Setting the age column as the index
age_data.set_index(('country'), inplace = True)
```

```
# Binning the age and creating a new column 'age_range' for the ranges of age
age_data['age_range'] = pd.cut(age_data.age, bins = 9,
                          labels = ['11-19', '20-29', '30-39', '40-49', '50-59',
                                    '60-69', '70-79', '80-89', '90-99'])
age_data
```

	age	workclass	education	education_num	marital_status	occupation	relations
country							
United-States	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-fa
United-States	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husb
United-States	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-fa
United-States	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husb
Cuba	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	\
...	...	...	...	...	...	...	
United-States	39	Private	Bachelors	13	Divorced	Prof-specialty	Not-in-fa
United-States	64	NaN	HS-grad	9	Widowed	NaN	Other-rela

Next steps: View recommended plots

```
age_data.age_range.value_counts()
```

```
age_range
30-39    10160
20-29    10079
11-19     9627
40-49     8302
50-59     5541
60-69     3330
70-79     1281
80-89      411
90-99      111
Name: count, dtype: int64
```

```
age_data.hours_per_week.unique()
```

```
array([40, 13, 16, 45, 50, 80, 30, 35, 60, 20, 52, 44, 15, 25, 38, 43, 55,
       48, 58, 32, 70,  2, 22, 56, 41, 28, 36, 24, 46, 42, 12, 65,  1, 10,
       34, 75, 98, 33, 54,  8,  6, 64, 19, 18, 72,  5,  9, 47, 37, 21, 26,
       14,  4, 59,  7, 99, 53, 39, 62, 57, 78, 90, 66, 11, 49, 84,  3, 17,
       68, 27, 85, 31, 51, 77, 63, 23, 87, 88, 73, 89, 97, 94, 29, 96, 67,
       82, 86, 91, 81, 76, 92, 61, 74, 95, 79, 69])
```

```
age_data.describe()
```

	age	education_num	capital_gain	capital_loss	hours_per_week
count	48842.000000	48842.000000	48842.000000	48842.000000	48842.000000
mean	38.643585	10.078089	1079.067626	87.502314	40.422382
std	13.710510	2.570973	7452.019058	403.004552	12.391444
min	17.000000	1.000000	0.000000	0.000000	1.000000
25%	28.000000	9.000000	0.000000	0.000000	40.000000
50%	37.000000	10.000000	0.000000	0.000000	40.000000
75%	48.000000	12.000000	0.000000	0.000000	45.000000
max	90.000000	16.000000	99999.000000	4356.000000	99.000000

Hours Data

```
# New data frame for the data of the 'hours-per-week'
hours_data = income_data.copy()
hours_data
```

	age	workclass	education	education_num	marital_status	occupation	relationshi
0	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-fami
1	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husban
2	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-fami
3	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husban
4	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Wif
...	...	...	...	...	...	...	.
48837	39	Private	Bachelors	13	Divorced	Prof-specialty	Not-in-fami
48838	64	NaN	HS-grad	9	Widowed	NaN	Other-relativ

Next steps: View recommended plots

```
#Setting the hours column as the index
hours_data.set_index(('country'), inplace = True)
hours_data
```

country	age	workclass	education	education_num	marital_status	occupation	relations
United-States	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-fa
United-States	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husb
United-States	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-fa
United-States	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husb
Cuba	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	\
...	...	...	...	...	...	...	
United-States	39	Private	Bachelors	13	Divorced	Prof-specialty	Not-in-fa
United-States	64	NaN	HS-grad	9	Widowed	NaN	Other-rela

Next steps: View recommended plots

```
# Binning the hours and creating a new column 'hours_range' for the ranges of hours
hours_data['hours_range'] = pd.cut(hours_data.hours_per_week, bins = 10,
                         labels = ['1-9', '11-19', '21-29', '31-39', '41-49',
                                 '51-59', '61-69', '71-79', '81-89', '91-99'])
hours_data
```

country	age	workclass	education	education_num	marital_status	occupation	relations
United-States	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-fa
United-States	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husb
United-States	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-fa
United-States	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husb
Cuba	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	\
...	...	...	...	...	...	...	
United-States	39	Private	Bachelors	13	Divorced	Prof-specialty	Not-in-fa
United-States	64	NaN	HS-grad	9	Widowed	NaN	Other-rela

Next steps: View recommended plots

```
income_data.workclass.unique()
```

```
array(['State-gov', 'Self-emp-not-inc', 'Private', 'Federal-gov',
       'Local-gov', '?', 'Self-emp-inc', 'Without-pay', 'Never-worked',
       nan], dtype=object)
```

```
hours_data.hours_range.value_counts()
```

```
hours_range
31-39    26639
41-49     8917
21-29     3398
11-19     3328
61-69     2642
51-59     1582
1-9       1125
71-79      683
```

```
81-89      315
91-99      213
Name: count, dtype: int64
```

```
income_data.country.value_counts()
```

```
country
United-States                 43832
Mexico                          951
?                               583
Philippines                     295
Germany                         206
Puerto-Rico                     184
Canada                          182
El-Salvador                     155
India                           151
Cuba                            138
England                         127
China                           122
South                           115
Jamaica                         106
Italy                           105
Dominican-Republic              103
Japan                            92
Guatemala                        88
Poland                           87
Vietnam                          86
Columbia                         85
Haiti                            75
Portugal                         67
Taiwan                           65
Iran                             59
Greece                           49
Nicaragua                        49
Peru                             46
Ecuador                          45
France                           38
Ireland                          37
Hong                             30
Thailand                         30
Cambodia                         28
Trinadad&Tobago                  27
Laos                             23
Yugoslavia                       23
Outlying-US(Guam-USVI-etc)       23
Scotland                         21
Honduras                         20
Hungary                          19
Holand-Netherlands                1
Name: count, dtype: int64
```

Visualizing Data

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

educ_income = income_data[['education_num', 'income']]
educ_income
```

	education_num	income
0	13	<=50K
1	13	<=50K
2	9	<=50K
3	7	<=50K
4	13	<=50K
...	...	...
48837	13	<=50K
48838	9	<=50K
48839	13	<=50K
48840	13	<=50K
48841	13	>50K

48842 rows × 2 columns

Next steps: View recommended plots

```
high_income = income_data[['occupation', 'income']]
high_income
```

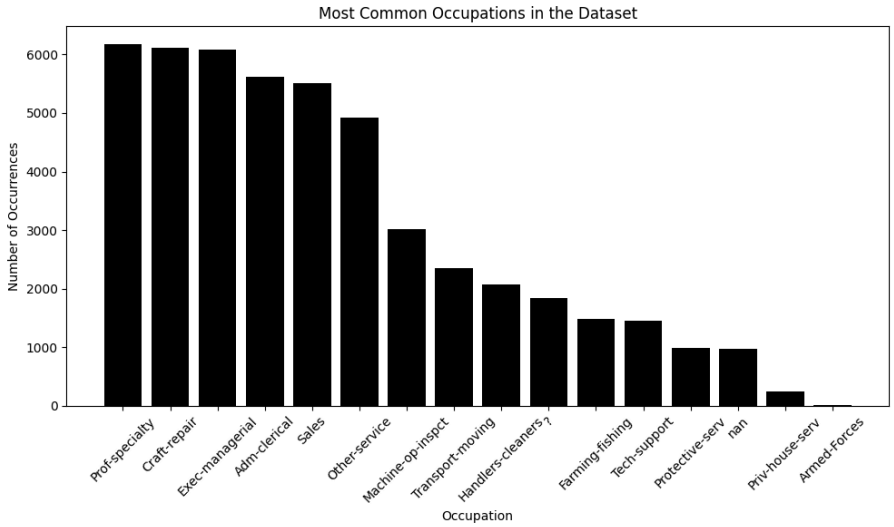
	occupation	income
0	Adm-clerical	<=50K
1	Exec-managerial	<=50K
2	Handlers-cleaners	<=50K
3	Handlers-cleaners	<=50K
4	Prof-specialty	<=50K
...	...	...
48837	Prof-specialty	<=50K
48838	NaN	<=50K
48839	Prof-specialty	<=50K
48840	Adm-clerical	<=50K
48841	Exec-managerial	>50K

48842 rows × 2 columns

Next steps: View recommended plots

Bar plot of Occupations that has more than 50k salary

```
import matplotlib.pyplot as plt

# Count the occurrences of each occupation
occupation_counts = income_data['occupation'].value_counts()

# Plot the number of each occupation
plt.figure(figsize=(10, 6))
plt.bar(occupation_counts.index, occupation_counts.values, color='black')
plt.xlabel('Occupation')
plt.ylabel('Number of Occurrences')
plt.title('Most Common Occupations in the Dataset')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.tight_layout()
plt.show()
```

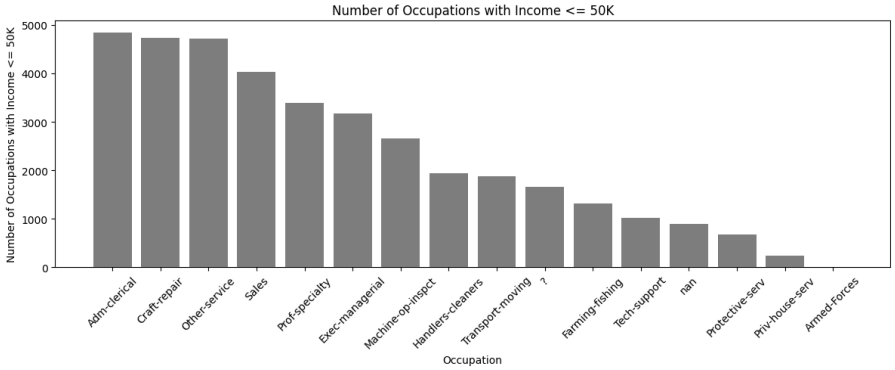


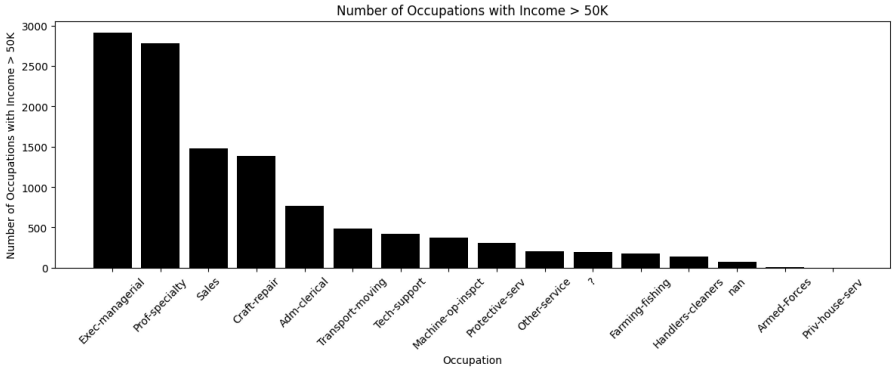

```
income_data.occupation.value_counts()
```

```
occupation
Prof-specialty       6172
Craft-repair         6112
Exec-managerial      6086
Adm-clerical         5611
Sales                5504
Other-service        4923
Machine-op-inspct    3022
Transport-moving     2355
Handlers-cleaners    2072
?                    1843
Farming-fishing      1490
Tech-support         1446
Protective-serv       983
nan                   966
Priv-house-serv       242
Armed-Forces           15
Name: count, dtype: int64
```

Bar plot of Occupations that has less than 50k salary

```
import matplotlib.pyplot as plt

# Filter the DataFrame for occupations with income less than or equal to 50k
low_income_data = income_data[income_data['income'] == '<=50K']
low_income_counts = low_income_data['occupation'].value_counts()

# Filter the DataFrame for occupations with income greater than 50k
high_income_data = income_data[income_data['income'] == '>50K']
high_income_counts = high_income_data['occupation'].value_counts()

fig, axs = plt.subplots(2, 1, figsize=(12, 10))

axs[0].bar(low_income_counts.index, low_income_counts.values, color='grey')
axs[0].set_xlabel('Occupation')
axs[0].set_ylabel('Number of Occupations with Income <= 50K')
axs[0].set_title('Number of Occupations with Income <= 50K')
axs[0].tick_params(axis='x', rotation=45)  # Rotate x-axis labels for better readability

# Plot for high income
axs[1].bar(high_income_counts.index, high_income_counts.values, color='black')
axs[1].set_xlabel('Occupation')
axs[1].set_ylabel('Number of Occupations with Income > 50K')
axs[1].set_title('Number of Occupations with Income > 50K')
axs[1].tick_params(axis='x', rotation=45)  # Rotate x-axis labels for better readability

plt.tight_layout()
plt.show()
```






```
high_income_counts = high_income_data['occupation'].value_counts()
high_income_counts
```

```
occupation
Exec-managerial      2908
Prof-specialty       2784
Sales                1475
Craft-repair         1383
Adm-clerical          768
Transport-moving      481
Tech-support          420
Machine-op-inspct     372
Protective-serv       308
Other-service         204
?                     191
Farming-fishing       173
Handlers-cleaners     138
nan                    74
Armed-Forces            5
Priv-house-serv         3
Name: count, dtype: int64
```

```
income_data.sex.unique()
```

```
array(['Male', 'Female'], dtype=object)
```

```
len(income_data.query('sex == "Male"'))
```

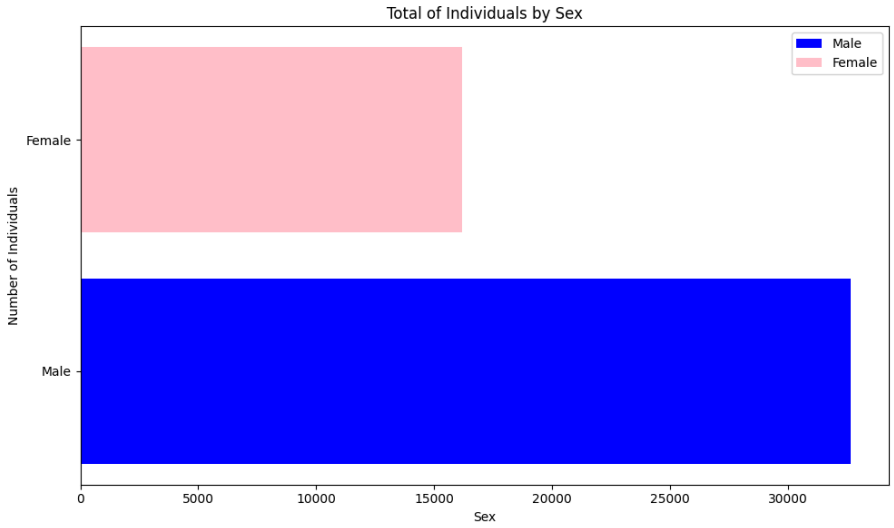
```
32650
```

```
len(income_data.query('sex == "Female"'))
```

```
16192
```

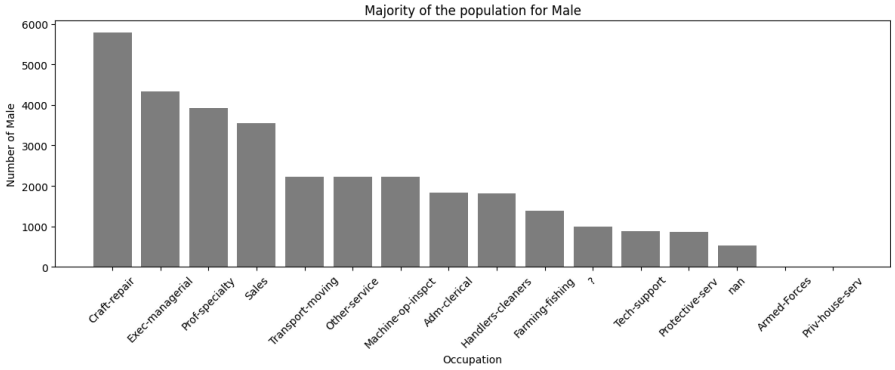
Plotting of Genders

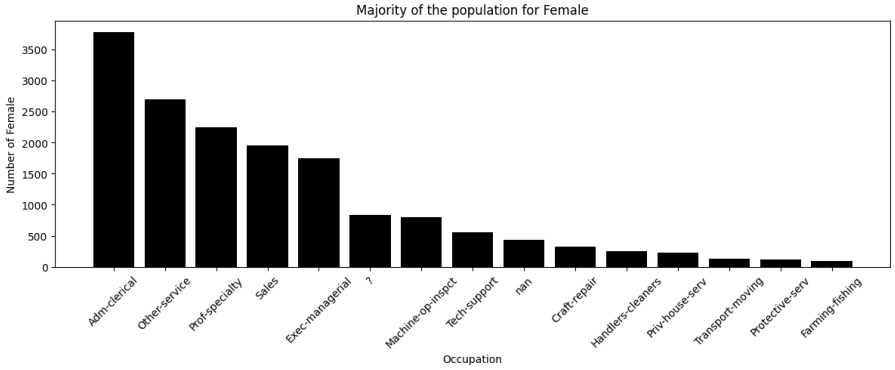
```
import matplotlib.pyplot as plt

# Count the total number of males and females
total_male_count = (income_data['sex'] == 'Male').sum()
total_female_count = (income_data['sex'] == 'Female').sum()

# Create subplots
fig, ax = plt.subplots(figsize=(10, 6))

# total number of males
ax.barh('Male', total_male_count, color='blue', label='Male')

# total number of females
ax.barh('Female', total_female_count, color='pink', label='Female')

ax.set_xlabel('Sex')
ax.set_ylabel('Number of Individuals')
ax.set_title('Total of Individuals by Sex')

# Adding a legend
ax.legend()

# Adjust layout
plt.tight_layout()
plt.show()
```



Highest Population of each gender in Occupations

```
import matplotlib.pyplot as plt

# Filter the DataFrame for occupations with income less than or equal to 50k
male_data = income_data[income_data['sex'] == 'Male']

# Count the occurrences of each occupation
male_counts = male_data['occupation'].value_counts()

# Filter the DataFrame for occupations with income greater than 50k
female_data = income_data[income_data['sex'] == 'Female']
female_counts = female_data['occupation'].value_counts()
fig, axs = plt.subplots(2, 1, figsize=(12, 10))

# Plot for Male
axs[0].bar(male_counts.index, male_counts.values, color='grey')
axs[0].set_xlabel('Occupation')
axs[0].set_ylabel('Number of Male')
axs[0].set_title('Majority of the population for Male')
axs[0].tick_params(axis='x', rotation=45)  # Rotate x-axis labels for better readability

# Plot for Female
axs[1].bar(female_counts.index, female_counts.values, color='black')
axs[1].set_xlabel('Occupation')
axs[1].set_ylabel('Number of Female')
axs[1].set_title('Majority of the population for Female')
axs[1].tick_params(axis='x', rotation=45)  # Rotate x-axis labels for better readability
plt.tight_layout()
plt.show()
```






```
male_counts
```

```
occupation
Craft-repair         5789
Exec-managerial      4338
Prof-specialty       3930
Sales                3557
Transport-moving     2228
Other-service        2225
Machine-op-inspct    2218
Adm-clerical         1842
Handlers-cleaners    1818
Farming-fishing      1395
?                    1002
Tech-support          884
Protective-serv       861
nan                   534
Armed-Forces           15
Priv-house-serv        14
Name: count, dtype: int64
```

```
female_counts
```

```
occupation
Adm-clerical         3769
Other-service        2698
Prof-specialty       2242
Sales                1947
Exec-managerial      1748
?                     841
Machine-op-inspct     804
Tech-support          562
nan                   432
Craft-repair          323
Handlers-cleaners     254
Priv-house-serv       228
Transport-moving      127
Protective-serv       122
Farming-fishing        95
Name: count, dtype: int64
```

```
income_data.education.unique()
```

```
array(['Bachelors', 'HS-grad', '11th', 'Masters', '9th', 'Some-college',
       'Assoc-acdm', 'Assoc-voc', '7th-8th', 'Doctorate', 'Prof-school',
       '5th-6th', '10th', '1st-4th', 'Preschool', '12th'], dtype=object)
```

```
import matplotlib.pyplot as plt
import pandas as pd

income_data['country'] = income_data['country'].astype(str)
```