Hands on Activity: Midterm Skills Exam: Data Wrangling and Analysis

Catulay, Weslie Jee L.

CPE22S2

Submitted to: Roman M. Richard

```
pip install ucimlrepo
```

```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.7-py3-none-any.whl (8.0 kB)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from ucimlrepo) (2.0.3)
Requirement already satisfied: certifi>=2020.12.5 in /usr/local/lib/python3.10/dist-packages (from ucimlrepo) (2024.6.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.0->ucimlrepo) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.0->ucimlrepo) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.0->ucimlrepo) (2024.1)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.0->ucimlrepo) (1.25.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas>=1.0.0->ucimlrep
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.7
```

```
from ucimlrepo import fetch_ucirepo

CE_IN = fetch_ucirepo(id=20)


Data_x = CE_IN.data.features
Data_y = CE_IN.data.targets

print(CE_IN.metadata)

print(CE_IN.variables)
```

```
{'uci_id': 20, 'name': 'Census Income', 'repository_url': 'https://archive.ics.uci.edu/dataset/20/census+income', 'data_url': 'https://a
                name     role  ... units missing_values
0                age  Feature  ...  None             no
1          workclass  Feature  ...  None            yes
2             fnlwgt  Feature  ...  None             no
3          education  Feature  ...  None             no
4      education-num  Feature  ...  None             no
5     marital-status  Feature  ...  None             no
6         occupation  Feature  ...  None            yes
7       relationship  Feature  ...  None             no
8               race  Feature  ...  None             no
9                sex  Feature  ...  None             no
10      capital-gain  Feature  ...  None             no
11      capital-loss  Feature  ...  None             no
12    hours-per-week  Feature  ...  None             no
13    native-country  Feature  ...  None            yes
14            income   Target  ...  None             no

[15 rows x 7 columns]
```

```
# Showing both Data to check

Data_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
...	...	...	...	...	...	...	...	...
48837	39	Private	215419	Bachelors	13	Divorced	Prof-specialty	Not-in-family
48838	64	NaN	321403	HS-grad	9	Widowed	NaN	Other-relative
48839	38	Private	374983	Bachelors	13	Married-civ-	Prof-specialty	Husband

Next steps: Generate code with Data_x View recommended plots

```
Data_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: Generate code with Data_y View recommended plots

```
# Concatenating Both Data

import pandas as pd
import numpy as np

data = pd.concat([Data_x, Data_y], axis = 1)

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
...	...	...	...	...	...	...	...	...
48837	39	Private	215419	Bachelors	13	Divorced	Prof-specialty	Not-in-family
48838	64	NaN	321403	HS-grad	9	Widowed	NaN	Other-relative
48839	38	Private	374983	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband
							Adm-	

Next steps: Generate code with data View recommended plots

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             48842 non-null  int64
 1   workclass       47879 non-null  object
 2   fnlwgt          48842 non-null  int64
 3   education       48842 non-null  object
 4   education-num   48842 non-null  int64
 5   marital-status  48842 non-null  object
 6   occupation      47876 non-null  object
 7   relationship    48842 non-null  object
 8   race            48842 non-null  object
 9   sex             48842 non-null  object
 10  capital-gain    48842 non-null  int64
 11  capital-loss    48842 non-null  int64
 12  hours-per-week  48842 non-null  int64
 13  native-country  48568 non-null  object
 14  income          48842 non-null  object
dtypes: int64(6), object(9)
memory usage: 5.6+ MB
```

```
# Show Datatypes
data.dtypes
```

```
age                int64
workclass         object
fnlwgt             int64
education         object
education-num      int64
marital-status    object
occupation        object
relationship      object
race              object
sex               object
capital-gain       int64
capital-loss       int64
hours-per-week     int64
native-country    object
income            object
dtype: object
```

```
data['native-country'].unique()
```

```
array(['United-States', 'Cuba', 'Jamaica', 'India', '?', 'Mexico',
       'South', 'Puerto-Rico', 'Honduras', 'England', 'Canada', 'Germany',
       'Iran', 'Philippines', 'Italy', 'Poland', 'Columbia', 'Cambodia',
       'Thailand', 'Ecuador', 'Laos', 'Taiwan', 'Haiti', 'Portugal',
       'Dominican-Republic', 'El-Salvador', 'France', 'Guatemala',
       'China', 'Japan', 'Yugoslavia', 'Peru',
       'Outlying-US(Guam-USVI-etc)', 'Scotland', 'Trinadad&Tobago',
       'Greece', 'Nicaragua', 'Vietnam', 'Hong', 'Ireland', 'Hungary',
       'Holand-Netherlands', nan], dtype=object)
```

```
change_nan = {"?" : "Others"}

data.replace(change_nan, inplace = True)
data.fillna('Others', inplace = True)
data
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	nat cou
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	U S
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	U S
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	U S
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	U S
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
48837	39	Private	215419	Bachelors	13	Divorced	Prof-specialty	Not-in-family	White	Female	0	0	36	U S
48838	64	Others	321403	HS-grad	9	Widowed	Others	Other-relative	Black	Male	0	0	40	U S
48839	38	Private	374983	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	50	U S
							Adm-		Asian-					U

Next steps: Generate code with data | View recommended plots

```
data.isna().any()
```

```
age               False
workclass         False
fnlwgt            False
education         False
education-num     False
marital-status    False
occupation        False
relationship      False
race              False
sex               False
capital-gain      False
capital-loss      False
hours-per-week    False
native-country    False
income            False
dtype: bool
```

```
data.rename(columns = {'native-country' : 'countries'}, inplace = True)
data.set_index(data['countries'],inplace = True)
data.drop('countries', axis = 1, inplace = True)
data
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relation
countries								
United-States	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-f
United-States	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Hus
United-States	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-f
United-States	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Hus
Cuba	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	
...	...	...	...	...	...	...	...	
United-States	39	Private	215419	Bachelors	13	Divorced	Prof-specialty	Not-in-f
United-States	64	Others	321403	HS-grad	9	Widowed	Others	Other-rel
United-States	38	Private	374983	Bachelors	13	Married-civ-spouse	Prof-specialty	Hus
United-	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Adm-	[illegible]

Next steps: Generate code with data | View recommended plots

```
data['income'].unique()
```

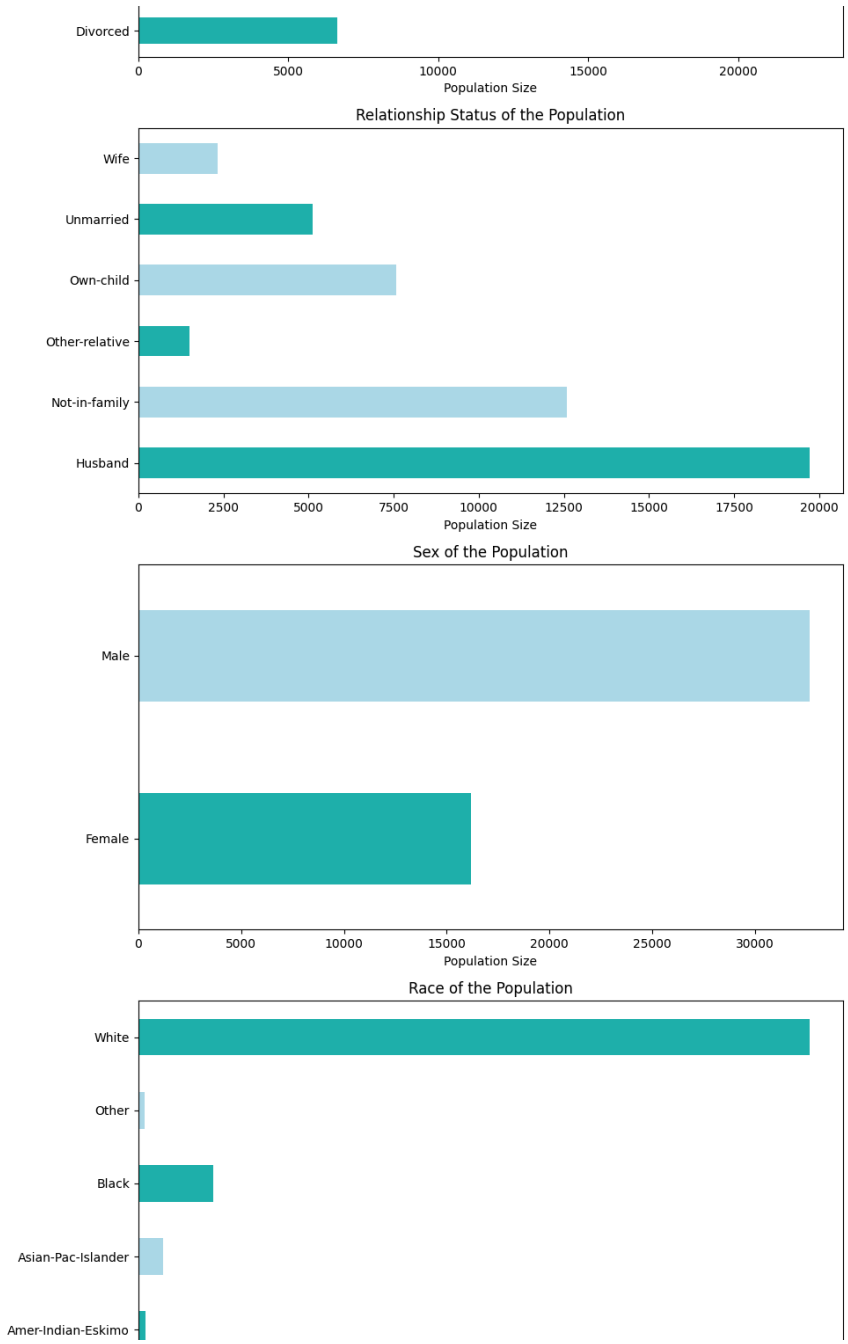
```
array(['<=50K', '>50K', '<=50K.', '>50K.'], dtype=object)
```

```
# Changing the and fixing the correct data output

row_change = {'<=50K.' : '<=50K',
              '>50K.' : '>50K'}

data = data.replace({'income' : row_change})
data
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relation
countries								
United-States	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-f
United-States	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Hus
United-States	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-f
United-States	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Hus
Cuba	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	
...	...	...	...	...	...	...	...	
United-States	39	Private	215419	Bachelors	13	Divorced	Prof-specialty	Not-in-f
United-States	64	Others	321403	HS-grad	9	Widowed	Others	Other-rel
United-States	38	Private	374983	Bachelors	13	Married-civ-spouse	Prof-specialty	Hus
United-	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Adm-	[illegible]

Next steps: Generate code with data View recommended plots

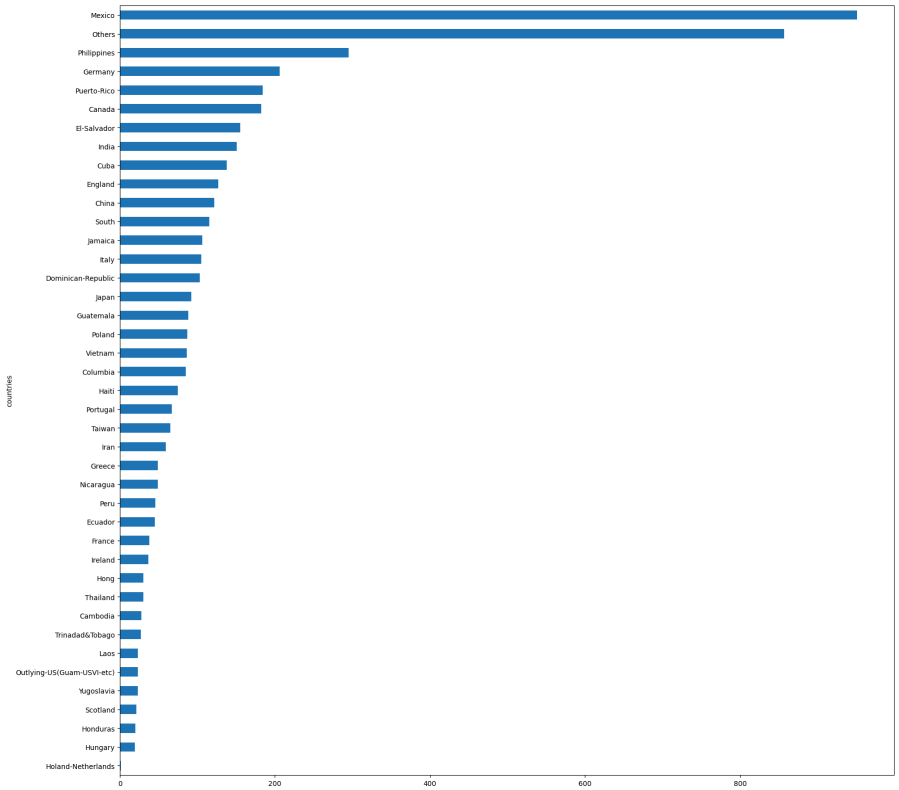
```
# Performing Data Plots and Perform exploratory data analysis

import matplotlib.pyplot as plt
import seaborn as sns

fig, ax = plt.subplots(4, figsize = [10,20])

#graph for marital-status column
data.groupby('marital-status').size().plot(kind='barh', ax = ax[0], color = ('lightseagreen','lightblue'))
ax[0].set_title('Population of Marital Status')
ax[0].set_xlabel('Population Size')
ax[0].set_ylabel('')

#graph for relationship column
data.groupby('relationship').size().plot(kind='barh', ax = ax[1], color = ('lightseagreen','lightblue'))
ax[1].set_title('Relationship Status of the Population')
ax[1].set_xlabel('Population Size')
ax[1].set_ylabel('')

#graph for sex column
data.groupby('sex').size().plot(kind='barh', ax = ax[2], color = ('lightseagreen','lightblue'))
ax[2].set_title('Sex of the Population')
ax[2].set_xlabel('Population Size')
ax[2].set_ylabel('')

#graph race column
data.groupby('race').size().plot(kind='barh', ax = ax[3], color = ('lightseagreen','lightblue'))
ax[3].set_title('Race of the Population')
ax[3].set_xlabel('Population Size')
ax[3].set_ylabel('')

fig.tight_layout()
```


Divorced
0
5000
10000
15000
20000
Population Size
Relationship Status of the Population
Wife
Unmarried
Own-child
Other-relative
Not-in-family
Husband
0
2500
5000
7500
10000
12500
15000
17500
20000
Population Size
Sex of the Population
Male
Female
0
5000
10000
15000
20000
25000
30000
Population Size
Race of the Population
White
Other
Black
Asian-Pac-Islander
Amer-Indian-Eskimo

```
# Checking the Data Population

data_population = data.groupby('countries').size().sort_values(ascending = False)
data_population = data_population.head()
data_population
```

```
countries
United-States    43832
Mexico             951
Others             857
Philippines        295
Germany            206
dtype: int64
```

```
data_population = data.reset_index()
data_population = data.drop('United-States')
data_population.groupby('countries').size().sort_values(ascending = True).plot(kind = 'barh', figsize = (20,20))
fig.tight_layout()
```



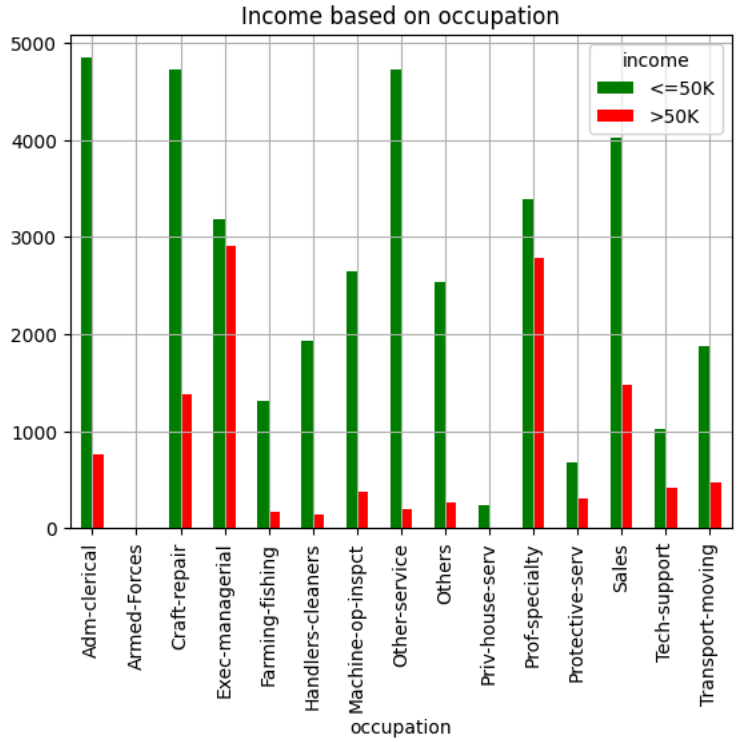

```
# Since the data has some uncessarry columns let's drop/dlete them

data.drop(['marital-status', 'relationship', 'race', 'sex'], axis = 1, inplace = True)
data
```

	age	workclass	fnlwgt	education	education-num	occupation	capital-gain	capital-loss
countries								
United-States	39	State-gov	77516	Bachelors	13	Adm-clerical	2174	0
United-States	50	Self-emp-not-inc	83311	Bachelors	13	Exec-managerial	0	0
United-States	38	Private	215646	HS-grad	9	Handlers-cleaners	0	0
United-States	53	Private	234721	11th	7	Handlers-cleaners	0	0
Cuba	28	Private	338409	Bachelors	13	Prof-specialty	0	0
...	...	...	...	...	...	...	...	...
United-States	39	Private	215419	Bachelors	13	Prof-specialty	0	0

Next steps: Generate code with data View recommended plots

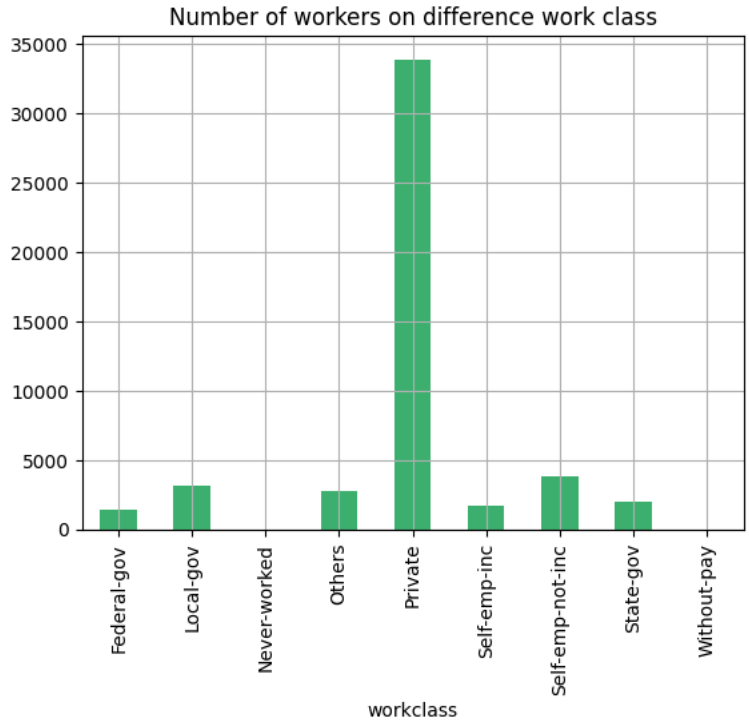
```
color_list = ['green', 'red']

data1=data.groupby(['occupation','income']).size()
data1=data1.unstack()
data1.plot(kind='bar', grid = True, title ='Income based on occupation', color =('green', 'red'))
```

```
<Axes: title={'center': 'Income based on occupation'}, xlabel='occupation'>
```




```
data2 = data.groupby(['workclass']).size()
data2.plot(kind = 'bar', grid = True, title = 'Number of workers on difference work class', color = 'mediumseagreen')
```

```
<Axes: title={'center': 'Number of workers on difference work class'},
xlabel='workclass'>
```




```
data2
# workclass counts
```

```
workclass
Federal-gov          1432
Local-gov            3136
Never-worked           10
Others               2799
Private             33906
Self-emp-inc         1695
Self-emp-not-inc     3862
State-gov            1981
Without-pay            21
dtype: int64
```

```
data3 = data.groupby(['workclass','income']).size()
data3=data3.unstack()
data3.plot(kind = 'barh', grid = True, title = 'Income based on Work Class', color = ('green', 'red'))
```

```
<Axes: title={'center': 'Income based on Work Class'}, ylabel='workclass'>
```


```
data3
```

income	<=50K	>50K
workclass		
Federal-gov	871.0	561.0
Local-gov	2209.0	927.0
Never-worked	10.0	NaN
Others	2534.0	265.0
Private	26519.0	7387.0
Self-emp-inc	757.0	938.0
Self-emp-not-inc	2785.0	1077.0
State-gov	1451.0	530.0
Without-pay	19.0	2.0

Next steps: Generate code with data3 View recommended plots

```
data_4 = data[['education-num', 'occupation']]
data_4.sort_values(by = 'education-num')
```

	education-num	occupation
countries		
Mexico	1	Other-service
United-States	1	Exec-managerial
Philippines	1	Priv-house-serv
El-Salvador	1	Other-service
United-States	1	Adm-clerical
...	...	...
United-States	16	Prof-specialty
Mexico	16	Prof-specialty
United-States	16	Prof-specialty
United-States	16	Exec-managerial
United-States	16	Prof-specialty

48842 rows × 2 columns

```
sns.stripplot(
x='occupation',
y='education-num',
hue='income',
data=data
)
plt.xticks(rotation=90)
```