MIDTERM SKILLS EXAM: DATA WRANGLING AND ANALYSIS

SUBMITTED BY: CASTILLO, RONEL CHRISTIAN V.

+ Code + Text

```
!pip install ucimlrepo
```

```
Requirement already satisfied: ucimlrepo in /usr/local/lib/python3.10/dist-packages (0.0.7)
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->ucimlrepo) (1.16.0)
```

```
from ucimlrepo import fetch_ucirepo

censusIncome = fetch_ucirepo(id=20)
gatherX = censusIncome.data.features
gatherY = censusIncome.data.targets

print(censusIncome.metadata)
print(censusIncome.variables)
```

```
{'uci_id': 20, 'name': 'Census Income', 'repository_url': 'https://archive.ics.uci.edu/dataset/20/census+income', 'data_url': 'https://archive.ics.uci.edu/static/public/20/data.csv', 'abstract': 'Predict whether income exceeds $50K/
              name     role         type      demographic  \
0              age  Feature      Integer              Age
1        workclass  Feature  Categorical           Income
2           fnlwgt  Feature      Integer             None
3        education  Feature  Categorical  Education Level
4    education-num  Feature      Integer  Education Level
5   marital-status  Feature  Categorical            Other
6       occupation  Feature  Categorical            Other
7     relationship  Feature  Categorical            Other
8             race  Feature  Categorical             Race
9              sex  Feature       Binary              Sex
10    capital-gain  Feature      Integer             None
11    capital-loss  Feature      Integer             None
12  hours-per-week  Feature      Integer             None
13  native-country  Feature  Categorical            Other
14          income   Target       Binary           Income

                                          description units missing_values
0                                                 N/A  None             no
1   Private, Self-emp-not-inc, Self-emp-inc, Feder...  None            yes
2                                                None  None             no
3    Bachelors, Some-college, 11th, HS-grad, Prof-...  None             no
4                                                None  None             no
5   Married-civ-spouse, Divorced, Never-married, S...  None             no
6   Tech-support, Craft-repair, Other-service, Sal...  None            yes
7   Wife, Own-child, Husband, Not-in-family, Other...  None             no
8   White, Asian-Pac-Islander, Amer-Indian-Eskimo,...  None             no
9                                       Female, Male.  None             no
10                                               None  None             no
11                                               None  None             no
12                                               None  None             no
13  United-States, Cambodia, England, Puerto-Rico,...  None            yes
14                                        >50K, <=50K.  None             no
```

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

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	39	Private	215419	Bachelors	13	Divorced	Prof-specialty	Not-in-family	White	Female	0	0	36	United-States
48838	64	NaN	321403	HS-grad	9	Widowed	NaN	Other-relative	Black	Male	0	0	40	United-States
48839	38	Private	374983	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	50	United-States
48840	44	Private	83891	Bachelors	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander	Male	5455	0	40	United-States
48841	35	Self-emp-inc	182148	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	60	United-States

48842 rows × 14 columns

Next steps: Generate code with gatherX | View recommended plots

```
gatherY
```

	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 gatherY | View recommended plots

```
newCensus= pd.concat([gatherX, gatherY], axis=1)
newCensus
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	39	Private	215419	Bachelors	13	Divorced	Prof-specialty	Not-in-family	White	Female	0	0	36	United-States	<=50K.
48838	64	NaN	321403	HS-grad	9	Widowed	NaN	Other-relative	Black	Male	0	0	40	United-States	<=50K.
48839	38	Private	374983	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	50	United-States	<=50K.
48840	44	Private	83891	Bachelors	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander	Male	5455	0	40	United-States	<=50K.
48841	35	Self-emp-inc	182148	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	60	United-States	>50K.

48842 rows × 15 columns

Next steps: Generate code with `newCensus` View recommended plots

```
# lets clean the dataset by removing any duplicated rows
newCensus.drop_duplicates(inplace=True)
```

```
newCensus.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 48813 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   age             48813 non-null  int64 
 1   workclass       47850 non-null  object
 2   fnlwgt          48813 non-null  int64 
 3   education       48813 non-null  object
 4   education-num   48813 non-null  int64 
 5   marital-status  48813 non-null  object
 6   occupation      47847 non-null  object
 7   relationship    48813 non-null  object
 8   race            48813 non-null  object
 9   sex             48813 non-null  object
 10  capital-gain    48813 non-null  int64 
 11  capital-loss    48813 non-null  int64 
 12  hours-per-week  48813 non-null  int64 
 13  native-country  48539 non-null  object
 14  income          48813 non-null  object
dtypes: int64(6), object(9)
memory usage: 6.0+ MB
```

```
# Determining workers employed in native-country column in the dataset.
newCensus['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)
```

```
# Converting categorical to numerical data and adding a dictionary to corresponding education column values.
educ= dict(zip(newCensus.education, newCensus['education-num']))
newCensus.drop(columns=['education'], inplace=True)
newCensus
```

	age	workclass	fnlwgt	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
0	39	State-gov	77516	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	39	Private	215419	13	Divorced	Prof-specialty	Not-in-family	White	Female	0	0	36	United-States	<=50K.
48838	64	NaN	321403	9	Widowed	NaN	Other-relative	Black	Male	0	0	40	United-States	<=50K.
48839	38	Private	374983	13	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	50	United-States	<=50K.
48840	44	Private	83891	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander	Male	5455	0	40	United-States	<=50K.
48841	35	Self-emp-inc	182148	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	60	United-States	>50K.

48813 rows × 14 columns

Next steps: Generate code with `newCensus` View recommended plots

```
educ
```

```
{'Bachelors': 13,
 'HS-grad': 9,
 '11th': 7,
 'Masters': 14,
 '9th': 5,
 'Some-college': 10,
 'Assoc-acdm': 12,
 'Assoc-voc': 11,
 '7th-8th': 4,
 'Doctorate': 16,
 'Prof-school': 15,
 '5th-6th': 3,
 '10th': 6,
 '1st-4th': 2,
 'Preschool': 1,
 '12th': 8}
```

```
columns= ['workclass', 'marital-status', 'occupation', 'relationship', 'race', 'sex', 'native-country', 'income']
uniqueVal= []

for column in columns:
  uniqueVal.append(newCensus[column].unique().tolist())

uniqueVal
```

```
  nan],
 ['Not-in-family',
  'Husband',
  'Wife',
  'Own-child',
  'Unmarried',
  'Other-relative'],
 ['White', 'Black', 'Asian-Pac-Islander', 'Amer-Indian-Eskimo', 'Other'],
 ['Male', 'Female'],
 ['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],
 ['<=50K', '>50K', '<=50K.', '>50K.']]
```

```python
# checking dictionaries

results= []

for data in uniqueVal:
  keys= [i for i in data]
  values= [i for i in range(1, len(data)+1)]
  results.append({keys[i]:values[i] for i in range(len(values))})

results
```

```
  'Priv-house-serv': 15,
  nan: 16},
 {'Not-in-family': 1,
  'Husband': 2,
  'Wife': 3,
  'Own-child': 4,
  'Unmarried': 5,
  'Other-relative': 6},
 {'White': 1,
  'Black': 2,
  'Asian-Pac-Islander': 3,
  'Amer-Indian-Eskimo': 4,
  'Other': 5},
 {'Male': 1, 'Female': 2},
 {'United-States': 1,
  'Cuba': 2,
  'Jamaica': 3,
  'India': 4,
  '?': 5,
  'Mexico': 6,
  'South': 7,
  'Puerto-Rico': 8,
  'Honduras': 9,
  'England': 10,
  'Canada': 11,
  'Germany': 12,
  'Iran': 13,
  'Philippines': 14,
  'Italy': 15,
  'Poland': 16,
  'Columbia': 17,
  'Cambodia': 18,
  'Thailand': 19,
  'Ecuador': 20,
  'Laos': 21,
  'Taiwan': 22,
  'Haiti': 23,
  'Portugal': 24,
  'Dominican-Republic': 25,
  'El-Salvador': 26,
  'France': 27,
  'Guatemala': 28,
  'China': 29,
  'Japan': 30,
  'Yugoslavia': 31,
  'Peru': 32,
  'Outlying-US(Guam-USVI-etc)': 33,
  'Scotland': 34,
  'Trinadad&Tobago': 35,
  'Greece': 36,
  'Nicaragua': 37,
  'Vietnam': 38,
  'Hong': 39,
  'Ireland': 40,
  'Hungary': 41,
  'Holand-Netherlands': 42,
  nan: 43},
 {'<=50K': 1, '>50K': 2, '<=50K.': 3, '>50K.': 4}]
```

Start coding or generate with AI.

	age	workclass	fnlwgt	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
0	39	State-gov	77516	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	39	Private	215419	13	Divorced	Prof-specialty	Not-in-family	White	Female	0	0	36	United-States	<=50K.
48838	64	Others	321403	9	Widowed	Others	Other-relative	Black	Male	0	0	40	United-States	<=50K.
48839	38	Private	374983	13	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	50	United-States	<=50K.
48840	44	Private	83891	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander	Male	5455	0	40	United-States	<=50K.
48841	35	Self-emp-inc	182148	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	60	United-States	>50K.

48813 rows × 14 columns

Next steps: Generate code with newCensus View recommended plots

```
newCensus['native-country'].value_counts()
```

```
native-country
United-States                 43810
Mexico                          947
Others                          856
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
Poland                           87
Guatemala                        86
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
```

```
newCensus.replace('<=50K.', '<=50K', inplace=True)
newCensus.replace('>50K.', '>50K', inplace=True)
```

```
newCensus['workclass'].value_counts()
```

```
workclass
Private             33879
Self-emp-not-inc     3861
Local-gov            3136
Others               2799
State-gov            1981
Self-emp-inc         1694
Federal-gov          1432
Without-pay            21
Never-worked           10
Name: count, dtype: int64
```

```
# data mapping
dataCensus= newCensus.copy()
for column in range(len(columns)):
  dataCensus.replace(results[column], inplace=True)
dataCensus
```

	age	workclass	fnlwgt	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
0	39	1.0	77516	13	1	1	1	1	1	2174	0	40	1	1
1	50	2.0	83311	13	2	2	2	1	1	0	0	13	1	1
2	38	3.0	215646	9	3	3	1	1	1	0	0	40	1	1
3	53	3.0	234721	7	2	3	2	2	1	0	0	40	1	1
4	28	3.0	338409	13	2	4	3	2	2	0	0	40	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	39	3.0	215419	13	3	4	1	1	2	0	0	36	1	3
48838	64	10.0	321403	9	7	10	6	2	1	0	0	40	1	3
48839	38	3.0	374983	13	2	4	2	1	1	0	0	50	1	3
48840	44	3.0	83891	13	3	1	4	3	1	5455	0	40	1	3
48841	35	7.0	182148	13	2	2	2	1	1	0	0	60	1	4

48813 rows × 14 columns

Next steps: Generate code with dataCensus View recommended plots

```
dataCensus['sex'].value_counts()
```

```
sex
1    32631
2    16182
Name: count, dtype: int64
```

```
# lets see what the values of the data are per sex, lets create new variables to do so
maleCensus= dataCensus.query('sex == 1')
femaleCensus= dataCensus.query('sex == 2')
```

```
# ave for males
maleCensus.mean()
```

```
age                   39.497594
workclass              3.362447
fnlwgt            191738.905795
education-num         10.095492
marital-status         1.928320
occupation             5.816984
relationship           2.262389
race                   1.191107
sex                    1.000000
capital-gain        1326.980509
capital-loss         100.468174
hours-per-week        42.419264
native-country         2.312617
income                 1.969262
dtype: float64
```

```
# ave for females
femaleCensus.mean()
```

```
age                   36.932827
workclass              3.447658
fnlwgt            185491.732172
education-num         10.044803
marital-status         2.398900
occupation             4.624027
relationship           3.096898
race                   1.279137
sex                    2.000000
capital-gain         581.085156
capital-loss          61.513472
hours-per-week        36.403720
native-country         2.313991
income                 1.779199
dtype: float64
```

Data Visualization Using Matplotlib and Seaborn

Now that we are done with Data wrangling, lets try and visualize our data using graphs

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

newCensus.replace(changeNaN, inplace = True)
newCensus.fillna('Others', inplace = True)
newCensus
```

	age	workclass	fnlwgt	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
0	39	State-gov	77516	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	39	Private	215419	13	Divorced	Prof-specialty	Not-in-family	White	Female	0	0	36	United-States	<=50K.
48838	64	Others	321403	9	Widowed	Others	Other-relative	Black	Male	0	0	40	United-States	<=50K.
48839	38	Private	374983	13	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	50	United-States	<=50K.
48840	44	Private	83891	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander	Male	5455	0	40	United-States	<=50K.
48841	35	Self-emp-inc	182148	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	60	United-States	>50K.

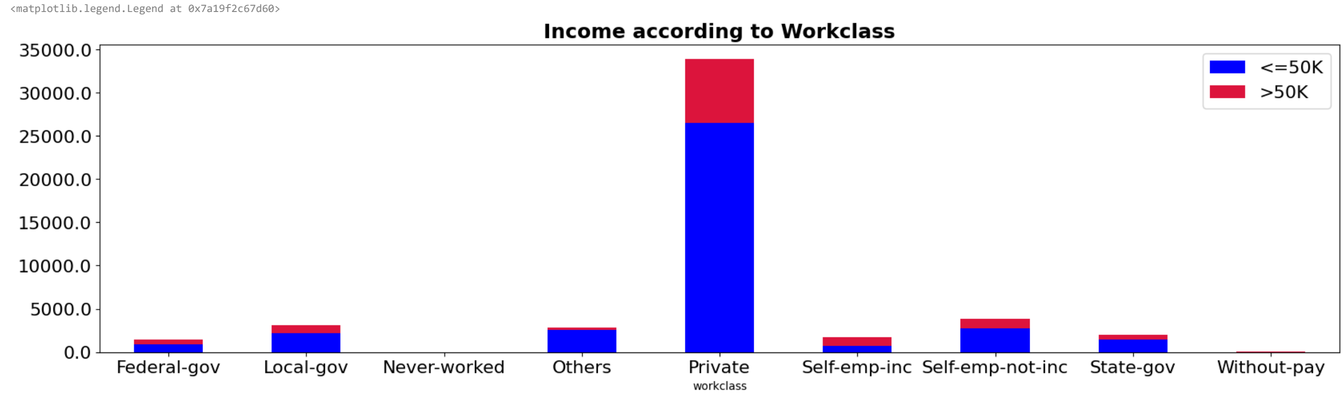
48813 rows × 14 columns

Next steps: Generate code with newCensus | View recommended plots

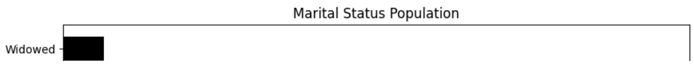
```
newCensus.replace('<=50K.', '<=50K', inplace=True)
newCensus.replace('>50K.', '>50K', inplace=True)

# we added these few lines of codes to better clean the data, if we didnt do this there would be a sandwich of data (example: red blue red)
```

```
# Using stacked bar graphs, plot each person's income according to work class
cen = newCensus.sample(300, random_state=0)
cen.groupby('workclass')['age'].describe()
crosstab = pd.crosstab(newCensus['workclass'], newCensus['income'])
fig, ax = plt.subplots(figsize=(20,5))
crosstab.plot(kind='bar', stacked=True, ax=ax, color=['blue', 'crimson'])
ax.set_title('Income according to Workclass', fontsize=18, fontweight='bold')
ax.set_xticklabels(ax.get_xticklabels(), fontsize=16, rotation=0)
ax.set_yticklabels(ax.get_yticks(), fontsize=16)
ax.legend(fontsize=16)
```

```
<ipython-input-115-8a4528483bed>:9: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.set_yticklabels(ax.get_yticks(), fontsize=16)
<matplotlib.legend.Legend at 0x7a19f2c67d60>
```

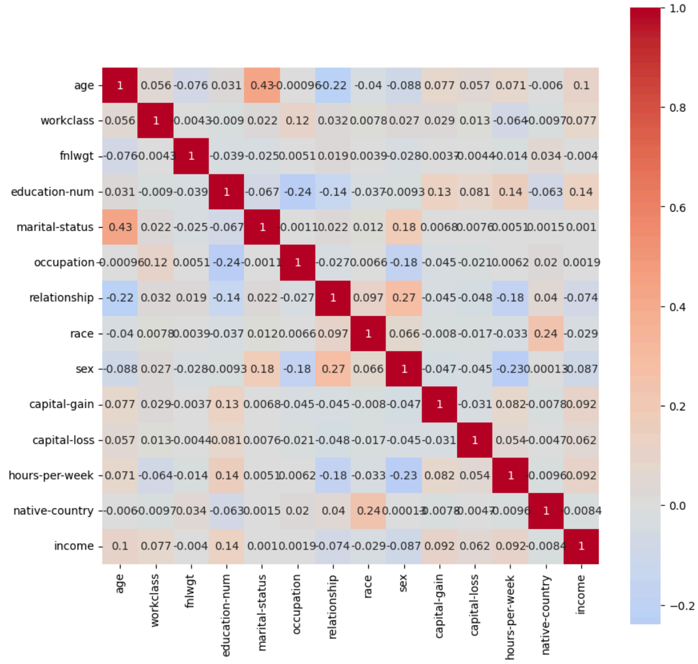


```
import matplotlib.pyplot as plt
import seaborn as sns

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

# marital-status
newCensus.groupby('marital-status').size().plot(kind='barh', ax = ax[0], color = ('black','pink'))
ax[0].set_title('Marital Status Population')
ax[0].set_xlabel('Number of People')


# relationship column
newCensus.groupby('relationship').size().plot(kind='barh', ax = ax[1], color = ('gray','green'))
ax[1].set_title('Relationship Situation in the Community')
ax[1].set_xlabel('Number of People')


# sex column
newCensus.groupby('sex').size().plot(kind='barh', ax = ax[2], color = ('crimson','orange'))
ax[2].set_title('Population\'s Sexual Orientations')
ax[2].set_xlabel('Number of People')


# race column
newCensus.groupby('race').size().plot(kind='barh', ax = ax[3], color = ('midnightblue','lightseagreen'))
ax[3].set_title('Population\'s Race')
ax[3].set_xlabel('Number of People')


fig.tight_layout()
```




```
# seaborn Heatmap
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10,10))
sns.heatmap(dataCensus.sort_index().corr(), annot=True, center=0, square=True, cmap= 'coolwarm')

# since we have 2 types of data sets, newCensus being the original one, and dataCensus being the numerical values one.
```

<Axes: >




```
# Graphing the categorical data via jointplot with Seaborn, relation of age and hours per week.
sns.jointplot(x='age', y='education-num', data=dataCensus, kind='hex')
```

<seaborn.axisgrid.JointGrid at 0x7a19f3e51720>