Midterm skill exam

Course: CPE 311	Program: BSCpE
Course Title: Computational Thinking with Python	**Date Performed:** April 13 , 2024
Section: BSCPE22S3	**Date Submitted:** April 14, 2024
Student Name: John Louie V. Adornado	**Instructor's Name:** Engr. Roman Richard

```
pip install ucimlrepo

Requirement already satisfied: ucimlrepo in
/usr/local/lib/python3.10/dist-packages (0.0.6)
```

pip install ucimlrepo: Installs a Python package called ucimlrepo, giving you access to datasets from the UCI Machine Learning Repository directly in Python.

```
import pandas as pd
import numpy as np
import seaborn as sb
#Importing all of the necessary
```

These libraries are often used together for data manipulation, analysis, and visualization in Python so we will import it.

```
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

# metadata
print(census_income.metadata)

# variable information
print(census_income.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/yr based on census data.  Also
known as Adult dataset.', 'area': 'Social Science', 'tasks':
['Classification'], 'characteristics': ['Multivariate'],
'num_instances': 48842, 'num_features': 14, 'feature_types':
['Categorical', 'Integer'], 'demographics': ['Age', 'Income',
```

```
'Education Level', 'Other', 'Race', 'Sex'], 'target_col': ['income'],
'index_col': None, 'has_missing_values': 'yes',
'missing_values_symbol': 'NaN', 'year_of_dataset_creation': 1996,
'last_updated': 'Thu Aug 10 2023', 'dataset_doi': '10.24432/C5GP7S',
'creators': ['Ron Kohavi'], 'intro_paper': None, 'additional_info':
{'summary': 'Extraction was done by Barry Becker from the 1994 Census
database.  A set of reasonably clean records was extracted using the
following conditions: ((AAGE>16) && (AGI>100) && (AFNLWGT>1)&&
(HRSWK>0))\r\n\r\nPrediction task is to determine whether a person
makes over 50K a year.', 'purpose': None, 'funded_by': None,
'instances_represent': None, 'recommended_data_splits': None,
'sensitive_data': None, 'preprocessing_description': None,
'variable_info': 'Listing of attributes:\r\n\r\n>50K, <=50K.\r\n\r\
nage: continuous.\r\nworkclass: Private, Self-emp-not-inc, Self-emp-
inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.\r\
nfnlwgt: continuous.\r\neducation: Bachelors, Some-college, 11th, HS-
grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters,
1st-4th, 10th, Doctorate, 5th-6th, Preschool.\r\neducation-num:
continuous.\r\nmarital-status: Married-civ-spouse, Divorced, Never-
married, Separated, Widowed, Married-spouse-absent, Married-AF-
spouse.\r\noccupation: Tech-support, Craft-repair, Other-service,
Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-
inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-
serv, Protective-serv, Armed-Forces.\r\nrelationship: Wife, Own-child,
Husband, Not-in-family, Other-relative, Unmarried.\r\nrace: White,
Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.\r\nsex: Female,
Male.\r\ncapital-gain: continuous.\r\ncapital-loss: continuous.\r\
nhours-per-week: continuous.\r\nnative-country: United-States,
Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-
USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras,
Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal,
Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti,
Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand,
Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-
Netherlands.', 'citation': None}}
```


```
              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
```

fetches a dataset from the UCI repository, extracts its features and targets, and then prints metadata and variable information about the dataset.

```
x #dataframe
```

{"summary":"{\n \"name\": \"x\",\n \"rows\": 48842,\n \"fields\": [\n   {\n   \"column\": \"age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 13,\n   \"min\": 17,\n   \"max\": 90,\n   \"num_unique_values\": 74,\n   \"samples\": [\n   28,\n   73,\n   35\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"workclass\",\n   \"properties\": {\n   \"dtype\": \"category\",\n

```
\"num_unique_values\": 9,\n        \"samples\": [\n
\"Without-pay\",\n          \"Self-emp-not-inc\",\n          \"?\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"fnlwgt\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 105604,\n
\"min\": 12285,\n        \"max\": 1490400,\n
\"num_unique_values\": 28523,\n        \"samples\": [\n
159077,\n          199450,\n          181773\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"education\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 16,\n        \"samples\": [\n
\"Bachelors\",\n          \"HS-grad\",\n          \"Some-college\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"education-num\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2,\n        \"min\": 1,\n        \"max\": 16,\n
\"num_unique_values\": 16,\n        \"samples\": [\n          13,\n
9,\n          10\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"marital-status\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 7,\n        \"samples\":
[\n          \"Never-married\",\n          \"Married-civ-spouse\",\n
\"Married-AF-spouse\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"occupation\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 15,\n
\"samples\": [\n          \"Machine-op-inspct\",\n          \"?\",\n
\"Adm-clerical\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"relationship\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 6,\n        \"samples\":
[\n          \"Not-in-family\",\n          \"Husband\",\n
\"Other-relative\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"race\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"Black\",\n          \"Other\",\n          \"Asian-Pac-Islander\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"sex\",\n      \"properties\": {\
n        \"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"Female\",\n          \"Male\"\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"capital-gain\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
7452,\n        \"min\": 0,\n        \"max\": 99999,\n
\"num_unique_values\": 123,\n        \"samples\": [\n          2176,\n
10520\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
```

```
\"capital-loss\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 403,\n        \"min\": 0,\n
\"max\": 4356,\n        \"num_unique_values\": 99,\n
\"samples\": [\n          1974,\n          419\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"hours-per-week\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
12,\n        \"min\": 1,\n        \"max\": 99,\n
\"num_unique_values\": 96,\n        \"samples\": [\n          97,\n
88\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"native-country\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 42,\n
\"samples\": [\n          \"El-Salvador\",\n          \"Philippines\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"x"}
```


```
y #dataframe
```


```
{"summary":"{\n  \"name\": \"y\",\n  \"rows\": 48842,\n  \"fields\":
[\n    {\n      \"column\": \"income\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 4,\n
\"samples\": [\n          \">50K\",\n          \">50K.\",\n
\"<=50K\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"y"}
```


```
concatenated_df = pd.concat([x, y], axis=1)
concatenated_df.to_csv('df.csv', index=False)
```

combining features and targets into a single DataFrame, which can be saved as a CSV file for further analysis

```
df = pd.read_csv('df.csv')
df
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 48842,\n  \"fields\":
[\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 13,\n        \"min\": 17,\n
\"max\": 90,\n        \"num_unique_values\": 74,\n        \"samples\":
[\n          28,\n          73,\n          35\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"workclass\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 9,\n        \"samples\": [\n
\"Without-pay\",\n          \"Self-emp-not-inc\",\n          \"?\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"fnlwgt\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 105604,\n
```

```
\"min\": 12285,\n        \"max\": 1490400,\n
\"num_unique_values\": 28523,\n        \"samples\": [\n
159077,\n          199450,\n          181773\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"education\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 16,\n        \"samples\": [\n
\"Bachelors\",\n          \"HS-grad\",\n          \"Some-college\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"education-num\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2,\n        \"min\": 1,\n        \"max\": 16,\n
\"num_unique_values\": 16,\n        \"samples\": [\n          13,\n
9,\n          10\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"marital-status\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 7,\n        \"samples\":
[\n          \"Never-married\",\n          \"Married-civ-spouse\",\n
\"Married-AF-spouse\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"occupation\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 15,\n
\"samples\": [\n          \"Machine-op-inspct\",\n          \"?\",\n
\"Adm-clerical\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"relationship\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 6,\n        \"samples\":
[\n          \"Not-in-family\",\n          \"Husband\",\n
\"Other-relative\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"race\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"Black\",\n          \"Other\",\n          \"Asian-Pac-Islander\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"sex\",\n      \"properties\": {\
n        \"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"Female\",\n          \"Male\"\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"capital-gain\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
7452,\n        \"min\": 0,\n        \"max\": 99999,\n
\"num_unique_values\": 123,\n        \"samples\": [\n          2176,\n
10520\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"capital-loss\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 403,\n        \"min\": 0,\n
\"max\": 4356,\n        \"num_unique_values\": 99,\n
\"samples\": [\n          1974,\n          419\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
```

```
n    },\n    {\n      \"column\": \"hours-per-week\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
12,\n        \"min\": 1,\n        \"max\": 99,\n
\"num_unique_values\": 96,\n        \"samples\": [\n          97,\n
88\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"native-country\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 42,\n
\"samples\": [\n          \"El-Salvador\",\n          \"Philippines\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"income\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \">50K\",\n          \">50K.\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```


```
df.head(20)
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 48842,\n  \"fields\":
[\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 13,\n        \"min\": 17,\n
\"max\": 90,\n        \"num_unique_values\": 74,\n        \"samples\":
[\n          28,\n          73,\n          35\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"workclass\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 9,\n        \"samples\": [\n
\"Without-pay\",\n          \"Self-emp-not-inc\",\n          \"?\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"fnlwgt\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 105604,\n
\"min\": 12285,\n        \"max\": 1490400,\n
\"num_unique_values\": 28523,\n        \"samples\": [\n
159077,\n          199450,\n          181773\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"education\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 16,\n        \"samples\": [\n
\"Bachelors\",\n          \"HS-grad\",\n          \"Some-college\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"education-num\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2,\n        \"min\": 1,\n        \"max\": 16,\n
\"num_unique_values\": 16,\n        \"samples\": [\n          13,\n
9,\n          10\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"marital-status\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 7,\n        \"samples\":
[\n          \"Never-married\",\n          \"Married-civ-spouse\",\n
```

\"Married-AF-spouse\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"occupation\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 15,\n
\"samples\": [\n   \"Machine-op-inspct\",\n   \"?\",\n
\"Adm-clerical\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"relationship\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 6,\n   \"samples\":
[\n   \"Not-in-family\",\n   \"Husband\",\n
\"Other-relative\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"race\",\n   \"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n   \"samples\": [\n
\"Black\",\n   \"Other\",\n   \"Asian-Pac-Islander\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"sex\",\n   \"properties\": {\
n   \"dtype\": \"category\",\n   \"num_unique_values\": 2,\n
\"samples\": [\n   \"Female\",\n   \"Male\"\n   ],\
n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"capital-gain\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
7452,\n   \"min\": 0,\n   \"max\": 99999,\n
\"num_unique_values\": 123,\n   \"samples\": [\n   2176,\n
10520\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"capital-loss\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 403,\n   \"min\": 0,\n
\"max\": 4356,\n   \"num_unique_values\": 99,\n
\"samples\": [\n   1974,\n   419\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"hours-per-week\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
12,\n   \"min\": 1,\n   \"max\": 99,\n
\"num_unique_values\": 96,\n   \"samples\": [\n   97,\n
88\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"native-country\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 42,\n
\"samples\": [\n   \"El-Salvador\",\n   \"Philippines\"\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"income\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 4,\n   \"samples\":
[\n   \">50K\",\n   \">50K.\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe","variable_name":"df"}

```
df.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
```

Here are the data types of the columns in our DataFrame. You can see that some columns are labeled as object types. We can change these object type columns into categorical data types. Doing this could make our data use less memory and work more efficiently. But, it might be hard to plot the DataFrame after we change it. So, we need to be careful. We want to clean up and make our data better, but we also need to think about whether we can still plot it easily. Also, not all object columns should be changed to categorical. It depends on what we're analyzing and what we need from the data.

```
df.isnull().sum()
```

```
age               0
workclass         0
fnlwgt            0
education         0
education-num     0
marital-status    0
occupation        0
relationship      0
race              0
sex               0
capital-gain      0
capital-loss      0
hours-per-week    0
native-country    0
income            0
dtype: int64
```

```
df.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
```

```
df.describe()
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 8,\n \"fields\": [\n
{\n   \"column\": \"age\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 17254.515015865374,\n
\"min\": 13.710509934443555,\n   \"max\": 48842.0,\n
\"num_unique_values\": 8,\n   \"samples\": [\n
38.64358543876172,\n   37.0,\n   48842.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"fnlwgt\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\": 487684.321495278,\
n   \"min\": 12285.0,\n   \"max\": 1490400.0,\n
\"num_unique_values\": 8,\n   \"samples\": [\n
189664.13459727284,\n   178144.5,\n   48842.0\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"education-num\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 17265.19214458616,\n   \"min\":
1.0,\n   \"max\": 48842.0,\n   \"num_unique_values\": 8,\n
\"samples\": [\n   10.078088530363212,\n   10.0,\n
48842.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"capital-gain\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 36540.175993736855,\n   \"min\":
0.0,\n   \"max\": 99999.0,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   1079.0676262233324,\n   99999.0,\n
7452.019057655394\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"capital-loss\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 17089.590809028763,\n   \"min\":

0.0,\n   \"max\": 48842.0,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   87.50231358257237,\n   4356.0,\n
403.00455212435907\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"hours-per-week\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 17254.246950179113,\n   \"min\":
1.0,\n   \"max\": 48842.0,\n   \"num_unique_values\": 7,\n
\"samples\": [\n   48842.0,\n   40.422382375824085,\n
45.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
df['workclass'].unique()
```

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

```
df['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)
```

```
df['occupation'].unique()
```

```
array(['Adm-clerical', 'Exec-managerial', 'Handlers-cleaners',
       'Prof-specialty', 'Other-service', 'Sales', 'Craft-repair',
       'Transport-moving', 'Farming-fishing', 'Machine-op-inspct',
       'Tech-support', '?', 'Protective-serv', 'Armed-Forces',
       'Priv-house-serv', nan], dtype=object)
```

```
df.replace("?", pd.NA, inplace=True)
```

```
df.isna().any()
```

```
age                 False
workclass            True
fnlwgt              False
education           False
education-num       False
marital-status      False
occupation           True
```

```
relationship      False
race              False
sex               False
capital-gain      False
capital-loss      False
hours-per-week    False
native-country     True
income            False
dtype: bool
```

This resulted in returning true therefore lets replace the NaN into "Others", we may replace it with much more longer label but Others fits more and it is not specified.

```
df.replace(pd.NA, "Others", inplace=True)
df
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 48842,\n \"fields\": [\n   {\n   \"column\": \"age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 13,\n   \"min\": 17,\n   \"max\": 90,\n   \"num_unique_values\": 74,\n   \"samples\": [\n   28,\n   73,\n   35\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"workclass\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 9,\n   \"samples\": [\n   \"Without-pay\",\n   \"Self-emp-not-inc\",\n   \"Others\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"fnlwgt\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 105604,\n   \"min\": 12285,\n   \"max\": 1490400,\n   \"num_unique_values\": 28523,\n   \"samples\": [\n   159077,\n   199450,\n   181773\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"education\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 16,\n   \"samples\": [\n   \"Bachelors\",\n   \"HS-grad\",\n   \"Some-college\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"education-num\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 2,\n   \"min\": 1,\n   \"max\": 16,\n   \"num_unique_values\": 16,\n   \"samples\": [\n   13,\n   9,\n   10\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"marital-status\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 7,\n   \"samples\": [\n   \"Never-married\",\n   \"Married-civ-spouse\",\n   \"Married-AF-spouse\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\":

```
\"occupation\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 15,\n
\"samples\": [\n          \"Machine-op-inspct\",\n
\"Others\",\n          \"Adm-clerical\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"relationship\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 6,\n        \"samples\": [\n          \"Not-in-
family\",\n          \"Husband\",\n          \"Other-relative\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"race\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
5,\n        \"samples\": [\n          \"Black\",\n
\"Other\",\n          \"Asian-Pac-Islander\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"sex\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"Female\",\n          \"Male\"\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"capital-gain\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
7452,\n        \"min\": 0,\n        \"max\": 99999,\n
\"num_unique_values\": 123,\n        \"samples\": [\n          2176,\n
10520\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"capital-loss\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 403,\n        \"min\": 0,\n
\"max\": 4356,\n        \"num_unique_values\": 99,\n
\"samples\": [\n          1974,\n          419\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"hours-per-week\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
12,\n        \"min\": 1,\n        \"max\": 99,\n
\"num_unique_values\": 96,\n        \"samples\": [\n          97,\n
88\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"native-country\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 42,\n
\"samples\": [\n          \"El-Salvador\",\n          \"Philippines\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"income\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \">50K\",\n          \">50K.\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

We notice that many of the data entries contain "?". It's up to us whether we want to get rid of these missing or unidentified values or replace them with "Others". In this case, we'll choose to replace them. This is because it could affect our analysis. For example, some entries might not have information about occupation but still have income data. This could affect our analysis because each sample is important data.

- In this dataset where each row represents information, some entries have missing or unidentified occupation data, indicated by "?" or empty. Simply removing these rows could lead to the loss of valuable income information associated with them. To preserve all available data for analysis, it's important to replace missing occupation entries with "Others". This ensures that income data, even from entries with missing details, remains intact and contributes to the analysis.

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

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

```
# Replace '<=50K.' with '<=50K' and '>50K.' with '>50K'
df['income'] = df['income'].replace({'<=50K.': '<=50K', '>50K.':
'>50K'})
```

```
# Checking for duplicate values
duplicates = df.duplicated().sum()
print("Number of Duplicate Rows:", duplicates)
```

```
Number of Duplicate Rows: 0
```

```
# Drop duplicate rows
df.drop_duplicates(inplace=True)
```

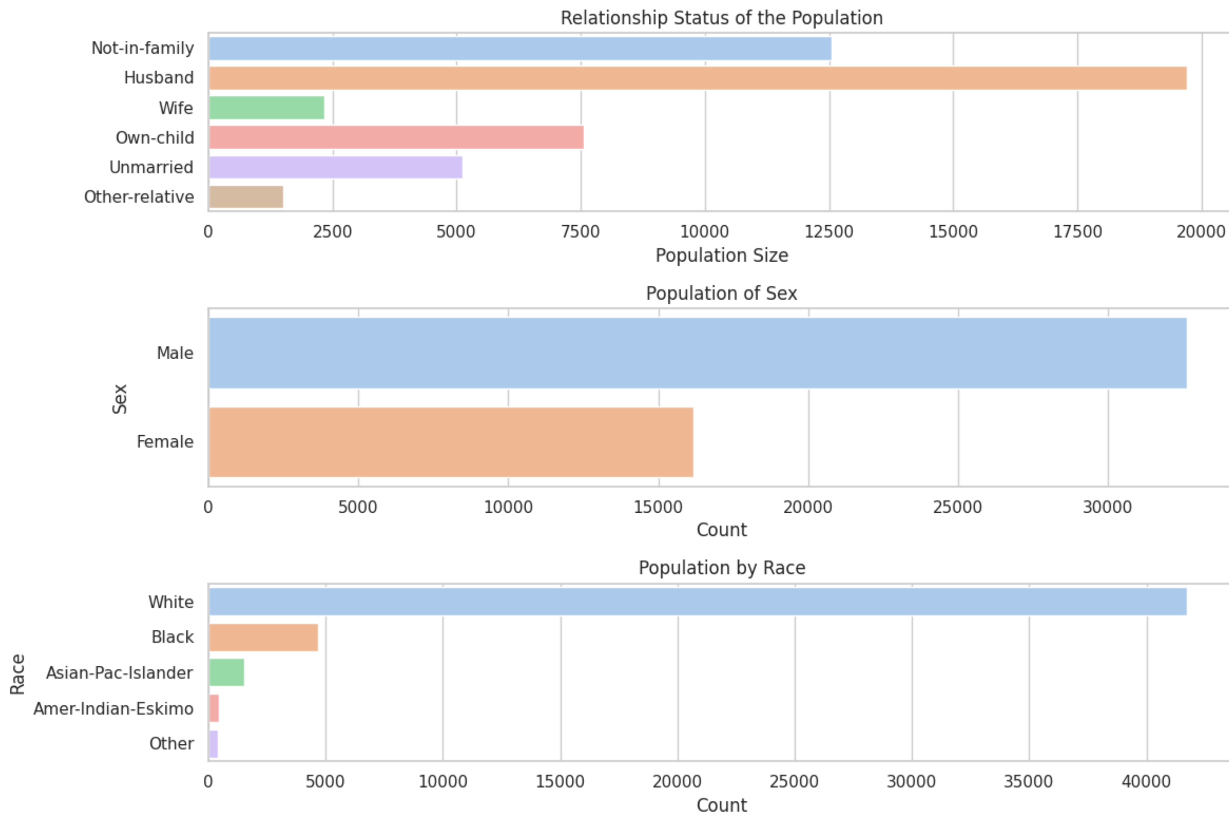
```
df.columns
```

```
Index(['age', 'workclass', 'fnlwgt', 'education', 'education-num',
       'marital-status', 'occupation', 'relationship', 'race', 'sex',
       'capital-gain', 'capital-loss', 'hours-per-week', 'native-
country',
       'income'],
      dtype='object')
```

Data Analysis and Data Exploratory

```
# Creating subplots
fig, ax = plt.subplots(3, figsize=[12, 8])

# Plotting relationship column
sns.countplot(y='relationship', hue='relationship', data=df, ax=ax[0],
palette='pastel')
ax[0].set_title('Relationship Status of the Population')
ax[0].set_xlabel('Population Size')
ax[0].set_ylabel('')

# Plotting sex column
```

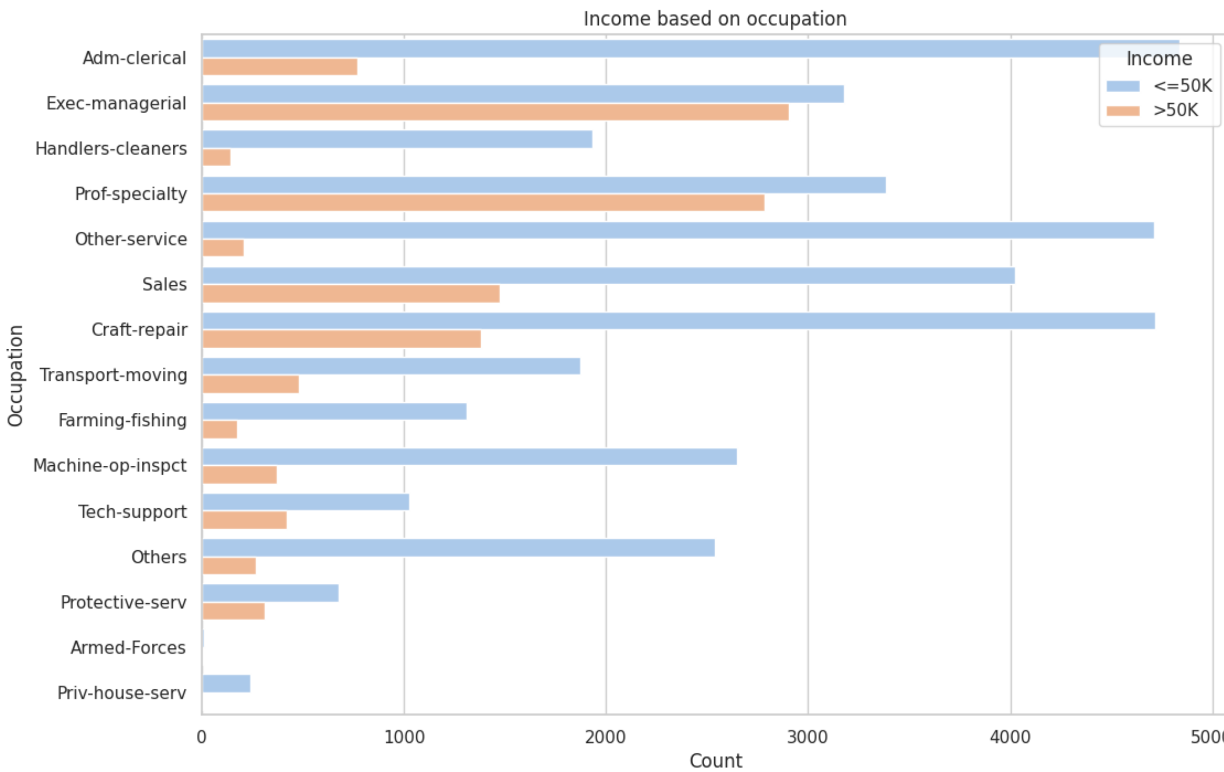
```
sns.countplot(y='sex',hue='sex', data=df, ax=ax[1], palette='pastel')
ax[1].set_title('Population of Sex')
ax[1].set_xlabel('Count')
ax[1].set_ylabel('Sex')

# Plotting race column
sns.countplot(y='race',hue='race', data=df, ax=ax[2],
palette='pastel')
ax[2].set_title('Population by Race')
ax[2].set_xlabel('Count')
ax[2].set_ylabel('Race')

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



We're plotting the relationship status, gender, and race to understand the population distribution. This helps us see how many people are in relationships versus others. It also shows the distribution between males and females, as well as the racial composition of the population. These insights are valuable for understanding our dataset.

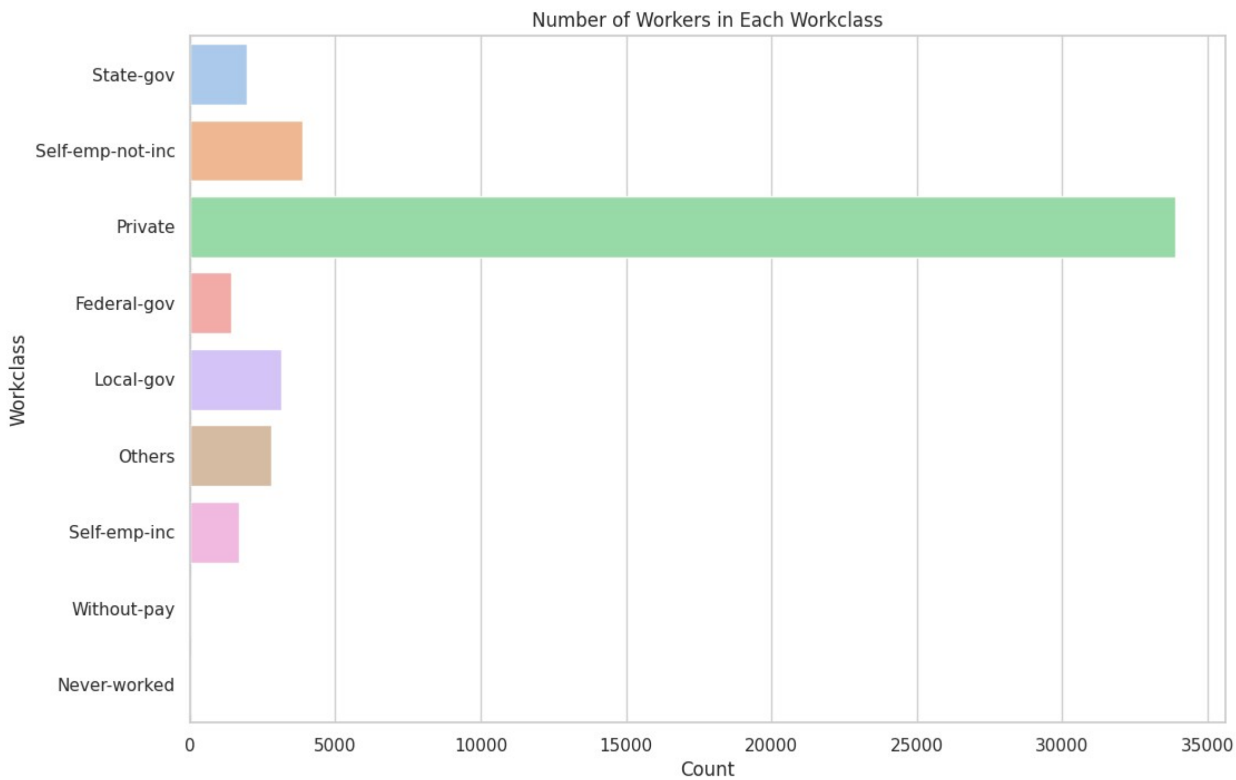
```
import seaborn as sns
import matplotlib.pyplot as plt

# Plotting
```

```
plt.figure(figsize=(12, 8))
sns.countplot(y='occupation', hue='income', data=df, palette='pastel')
plt.title('Income based on occupation')
plt.xlabel('Count')
plt.ylabel('Occupation')
plt.legend(title='Income', loc='upper right')
plt.show()
```



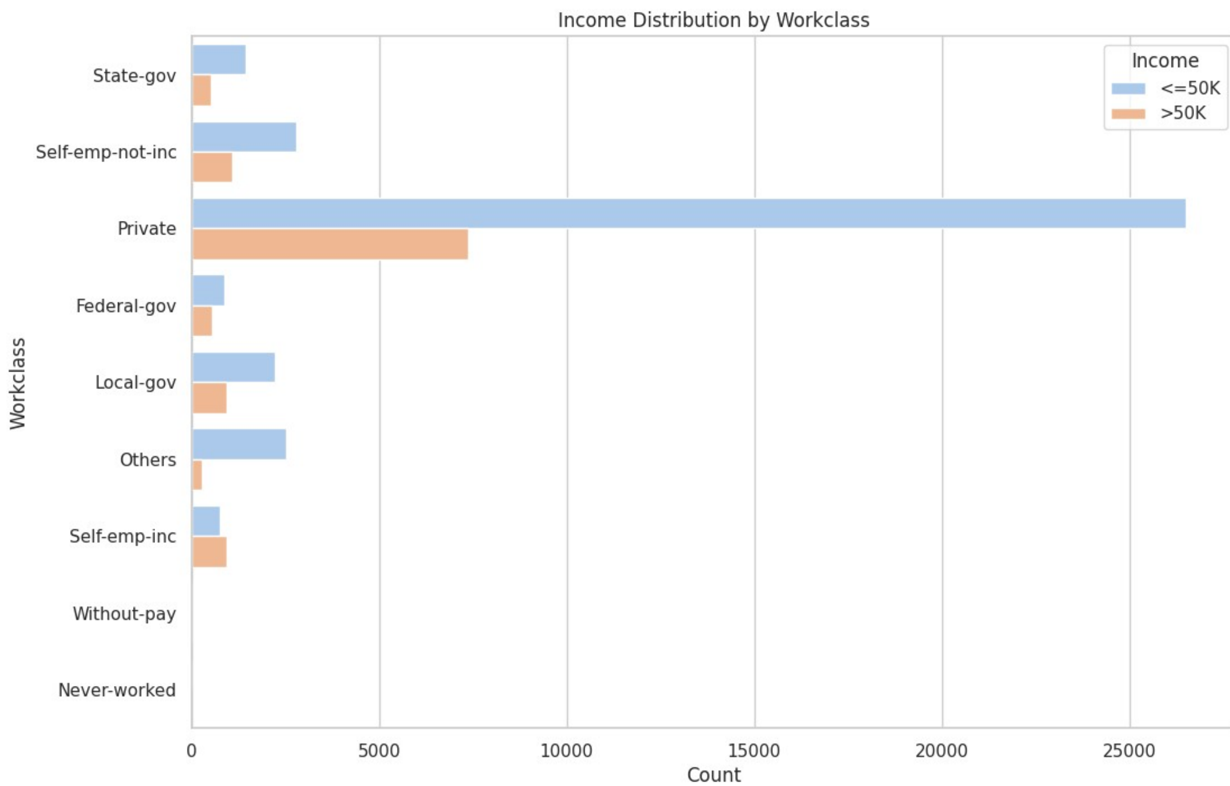
We're examining income across different occupations. It's evident that a majority earn less than or equal to 50k, outnumbering those earning more than 50k. For instance, in farming-fishing, a considerable number of individuals earn less than 50k, with only a small percentage earning above this threshold. This suggests a significant portion of individuals in each occupation earn lower incomes

```
plt.figure(figsize=(12, 8))
sns.countplot(y='workclass', hue="workclass", data=df,
palette='pastel')
plt.title('Number of Workers in Each Workclass')
plt.xlabel('Count')
plt.ylabel('Workclass')
plt.show()
```



A large proportion of workers across various work classes are employed in the private sector. This indicates that a significant number of individuals, regardless of their specific work class, are engaged in employment within private companies or organizations.

```
plt.figure(figsize=(12, 8))
sns.countplot(y='workclass', hue='income', data=df, palette='pastel')
plt.title('Income Distribution by Workclass')
plt.xlabel('Count')
plt.ylabel('Workclass')
plt.legend(title='Income', loc='upper right')
plt.show()
```



We're examining income distribution across different work classes. It's clear that individuals in the private sector dominate in numbers. Most of them earn less than 50k, but there's also a notable portion earning more than that. This higher-income group in the private sector still outnumbers those in other work classes.

```
# Grouping the data by workclass and income and getting the count
workclass_income_counts = df.groupby(['workclass',
'income']).size().reset_index(name='count')

# Displaying the resulting DataFrame
print(workclass_income_counts)
```

```
           workclass income  count
0        Federal-gov  <=50K    871
1        Federal-gov   >50K    561
2          Local-gov  <=50K   2209
3          Local-gov   >50K    927
4       Never-worked  <=50K     10
5             Others  <=50K   2534
6             Others   >50K    265
7            Private  <=50K  26519
8            Private   >50K   7387
9       Self-emp-inc  <=50K    757
10      Self-emp-inc   >50K    938
11  Self-emp-not-inc  <=50K   2785
12  Self-emp-not-inc   >50K   1077
```

```
13         State-gov  <=50K   1451
14         State-gov   >50K    530
15       Without-pay  <=50K     19
16       Without-pay   >50K      2
```

Here is the income distribution across various work classes. The private sector stands out as the largest group, with a significant number of individuals earning both less than and more than $50,000. While most work classes have a majority earning less than or equal to $50,000, there are notable exceptions, such as the federal government and self-employed (both incorporated and not incorporated), where a considerable proportion earn higher incomes.

```
# Grouping the data by occupation, education, and country, and getting
the count
occupation_education_country_counts = df.groupby(['occupation',
'education', 'native-country']).size().reset_index(name='count')

# Displaying the resulting DataFrame
print(occupation_education_country_counts)

            occupation     education              native-country
count
0         Adm-clerical          10th                     Germany
1
1         Adm-clerical          10th                     Jamaica
1
2         Adm-clerical          10th                      Mexico
2
3         Adm-clerical          10th               United-States
55
4         Adm-clerical          11th                      Canada
1
...                ...           ...                         ...    ..
.
2024  Transport-moving  Some-college                      Others
7
2025  Transport-moving  Some-college  Outlying-US(Guam-USVI-etc)
1
2026  Transport-moving  Some-college                        Peru
1
2027  Transport-moving  Some-college                 Puerto-Rico
2
2028  Transport-moving  Some-college               United-States
390

[2029 rows x 4 columns]

import seaborn as sns
import matplotlib.pyplot as plt

# Setting the plot size
```

```python
plt.figure(figsize=(12, 8))

# Creating the violin plot
sns.violinplot(x='income', y='occupation', data=df, palette='pastel')

# Adding title and labels
plt.title('Income Distribution by Occupation')
plt.xlabel('Income')
plt.ylabel('Occupation')

# Displaying the plot
plt.show()
```

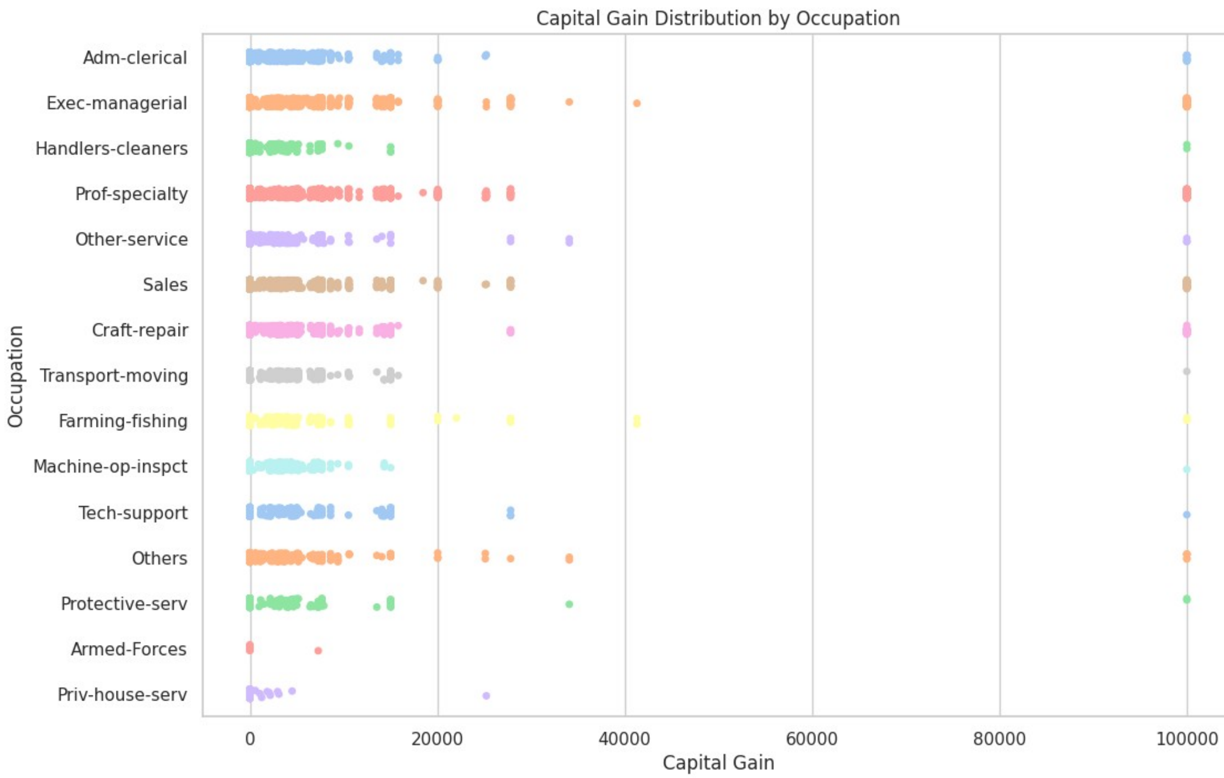
```
<ipython-input-137-cfb98158961c>:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.violinplot(x='income', y='occupation', data=df,
palette='pastel')
```

We've created a violin plot displaying the income distribution across different occupations. The violin plot provides a visual representation of the median income and other statistical measures, resulting in a more accurate depiction of the income density for each occupation. While it may be slightly more complex to interpret, it allows us to observe the density of income distribution within each occupation more effectively

```
# Counting the number of individuals in each income category for each
occupation
occupation_income_counts = df.groupby(['occupation',
'income']).size().unstack(fill_value=0)

# Calculating the total number of individuals in each occupation
occupation_totals = occupation_income_counts.sum(axis=1)

# Calculating the percentage of individuals with income <=50K and >50K
for each occupation
occupation_income_percentages =
occupation_income_counts.divide(occupation_totals, axis=0) * 100

# Displaying the result
print(occupation_income_percentages)

income                  <=50K       >50K
occupation
Adm-clerical        86.300392  13.699608
Armed-Forces        66.666667  33.333333
Craft-repair        77.351688  22.648312
Exec-managerial     52.219665  47.780335
Farming-fishing     88.350168  11.649832
Handlers-cleaners   93.336552   6.663448
Machine-op-inspct   87.703016  12.296984
Other-service       95.852816   4.147184
Others              90.552585   9.447415
Priv-house-serv     98.750000   1.250000
Prof-specialty      54.874290  45.125710
Protective-serv     68.635438  31.364562
Sales               73.186693  26.813307
Tech-support        70.934256  29.065744
Transport-moving    79.575372  20.424628


# Setting the plot size
plt.figure(figsize=(12, 8))

# Creating the strip plot
sns.stripplot(x='capital-gain', y='occupation', data=df,
palette='pastel', jitter=True)

# Adding title and labels
plt.title('Capital Gain Distribution by Occupation')
plt.xlabel('Capital Gain')
plt.ylabel('Occupation')
```

```
# Displaying the plot
plt.show()
```

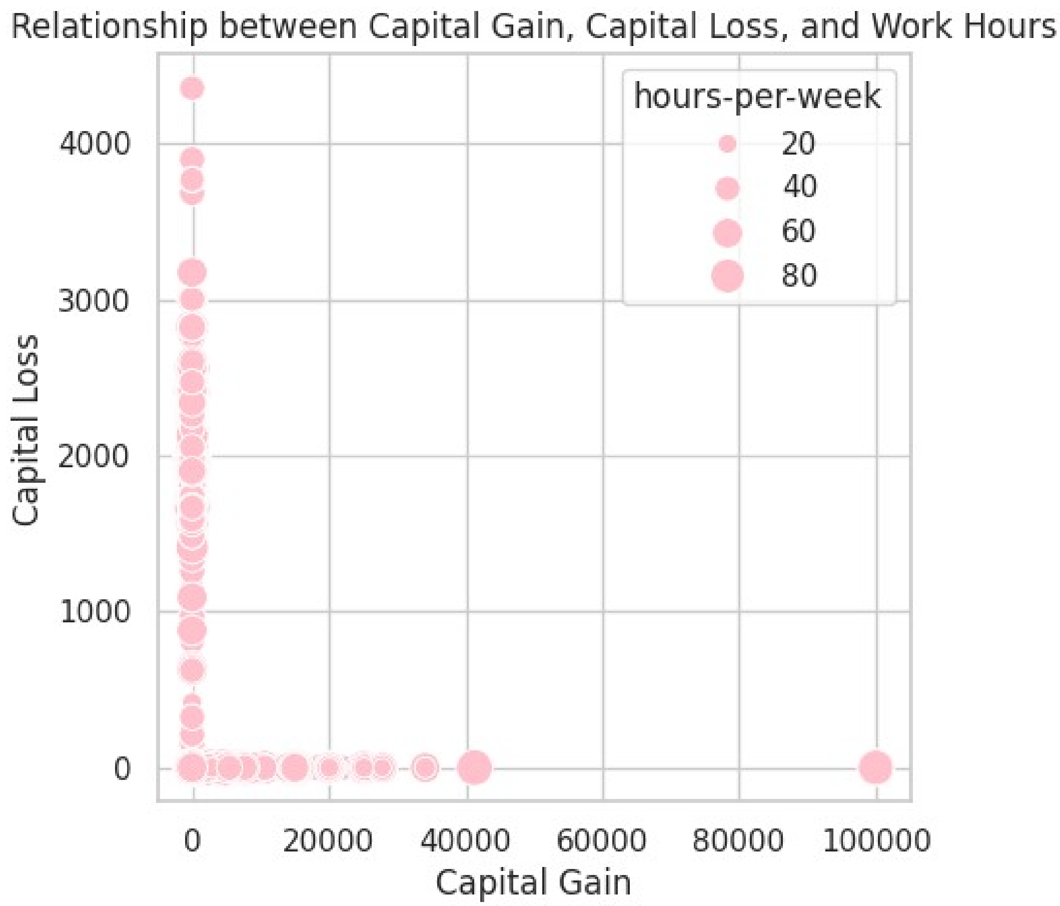
```
<ipython-input-144-e7547e58b3c4>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.stripplot(x='capital-gain', y='occupation', data=df,
palette='pastel', jitter=True)
```



We've generated a strip plot to visualize how capital gains are distributed across different occupations. This plot offers a clear depiction of the range and distribution of capital gains within each occupation, providing insights how these gains vary across different professional fields.

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

# Setting the plot size
plt.figure(figsize=(5, 5))

# Creating the scatter plot
sns.scatterplot(x='capital-gain', y='capital-loss', size='hours-per-
week', sizes=(20, 200), data=df, color='pink')
```

```
# Adding title and labels
plt.title('Relationship between Capital Gain, Capital Loss, and Work
Hours')
plt.xlabel('Capital Gain')
plt.ylabel('Capital Loss')

# Displaying the plot
plt.show()
```

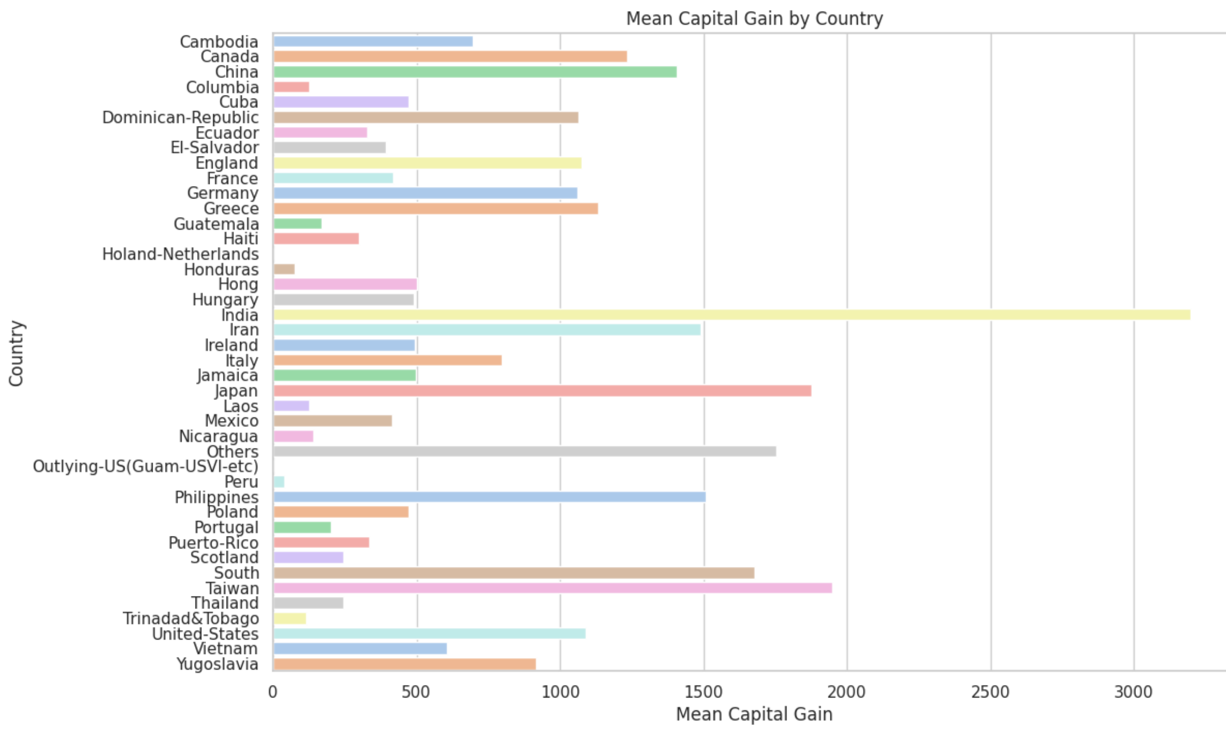


We've presented a scatter plot illustrating the relationship between capital gain and capital loss, along with their respective counts, alongside the number of hours worked per week. The rationale behind exploring this relationship lies in understanding how work hours influence individuals' capacity to generate the capital used for investment. Many individuals devote their time and effort to their careers or businesses to earn income, which forms the basis for their investments. The number of work hours directly impacts the amount of income one can generate, thereby influencing their ability to invest and subsequently realize capital gains or losses. By examining these variables together, we gain insights into the interplay between work hours, income generation, and outcomes.

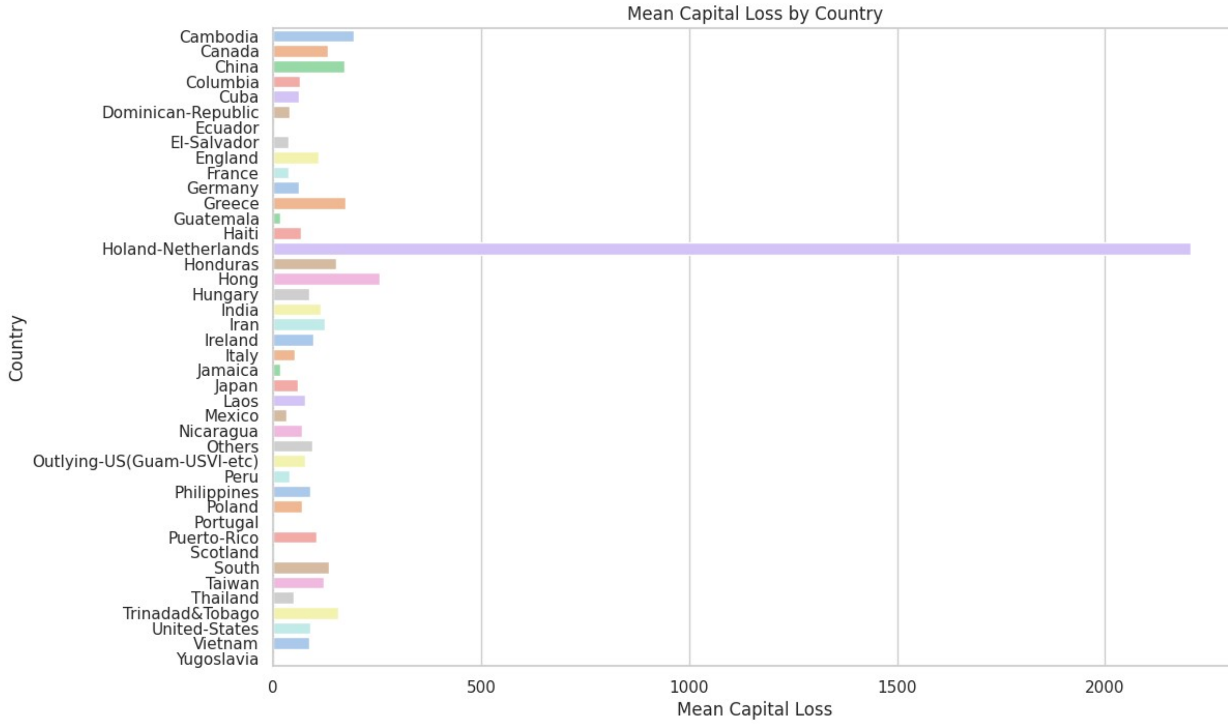
```
import seaborn as sns
import matplotlib.pyplot as plt

# Calculating mean capital gain, capital loss, and work hours for each
country
country_stats = df.groupby('native-country').agg({'capital-
gain':'mean', 'capital-loss':'mean', 'hours-per-
week':'mean'}).reset_index()

# Plotting
plt.figure(figsize=(12, 8))

# Creating the bar plot for mean capital gain
sns.barplot(x='capital-gain', y='native-country', data=country_stats,
palette='pastel')
plt.title('Mean Capital Gain by Country')
plt.xlabel('Mean Capital Gain')
plt.ylabel('Country')

# Displaying the plot
plt.show()

# Plotting
plt.figure(figsize=(12, 8))

# Creating the bar plot for mean capital loss
sns.barplot(x='capital-loss', y='native-country', data=country_stats,
palette='pastel')
plt.title('Mean Capital Loss by Country')
plt.xlabel('Mean Capital Loss')
plt.ylabel('Country')

# Displaying the plot
plt.show()

# Plotting
plt.figure(figsize=(12, 8))

# Creating the bar plot for mean work hours
sns.barplot(x='hours-per-week', y='native-country',
data=country_stats, palette='pastel')
plt.title('Mean Work Hours by Country')
plt.xlabel('Mean Work Hours')
plt.ylabel('Country')

# Displaying the plot
plt.show()
```

```
<ipython-input-148-093283930419>:11: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
```

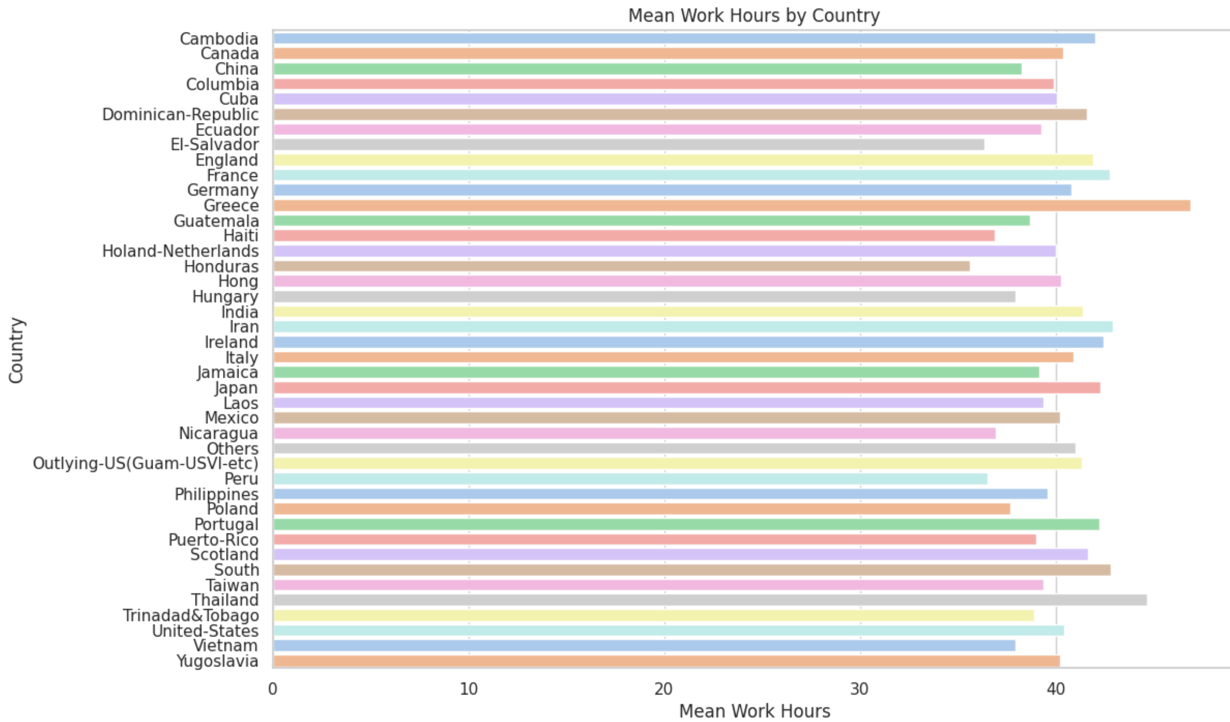
```
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='capital-gain', y='native-country',
data=country_stats, palette='pastel')
```




```
<ipython-input-148-093283930419>:23: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='capital-loss', y='native-country',
data=country_stats, palette='pastel')
```

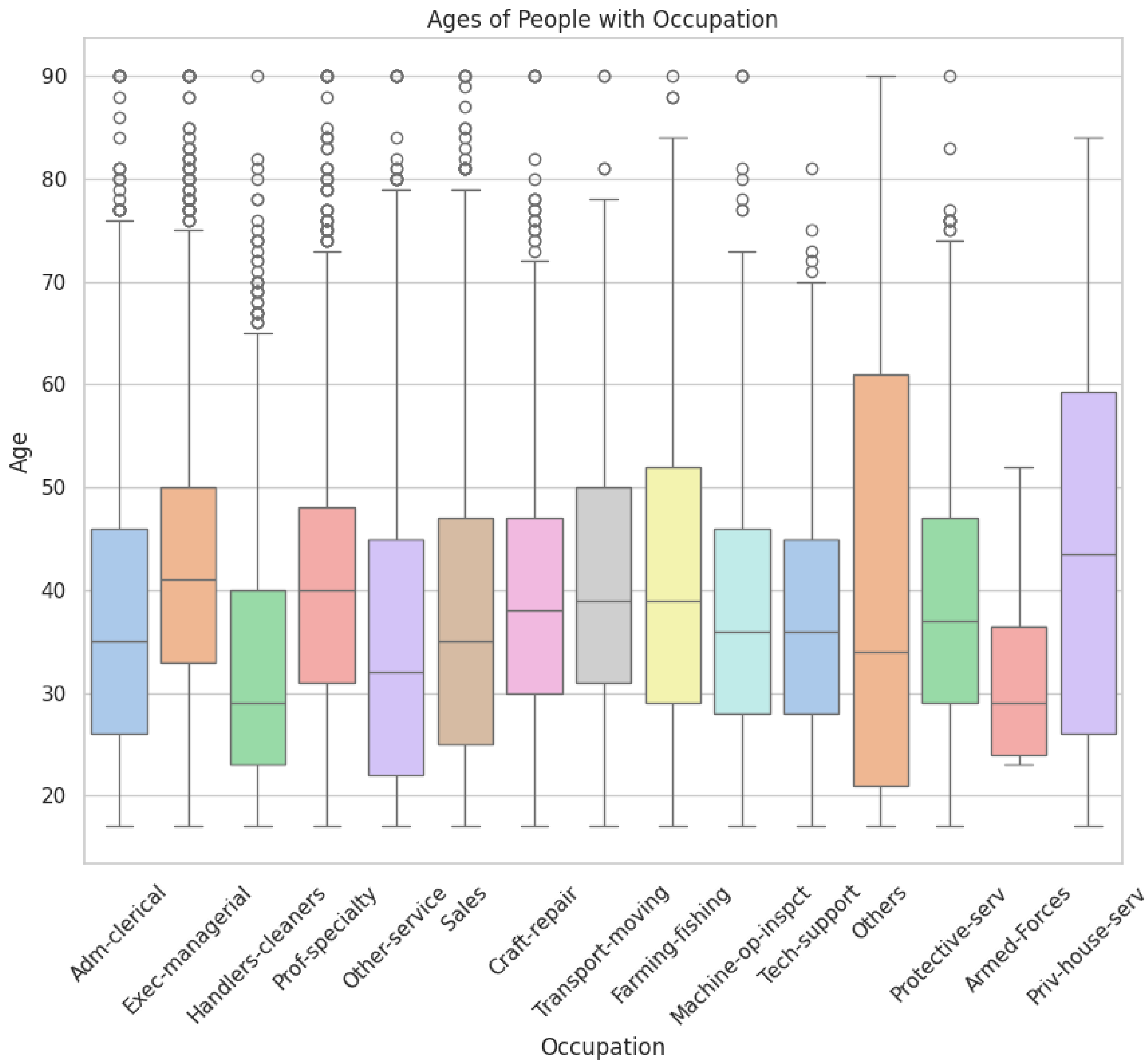



```
<ipython-input-148-093283930419>:35: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='hours-per-week', y='native-country',
data=country_stats, palette='pastel')
```



Upon analyzing the plot, it becomes evident that India has the highest mean capital gain among the observed countries. Additionally, the mean capital loss for each country highlights that the Netherlands surpasses all others in this aspect. Furthermore, in terms of work hours, the mean across nearly all countries hovers around 40.

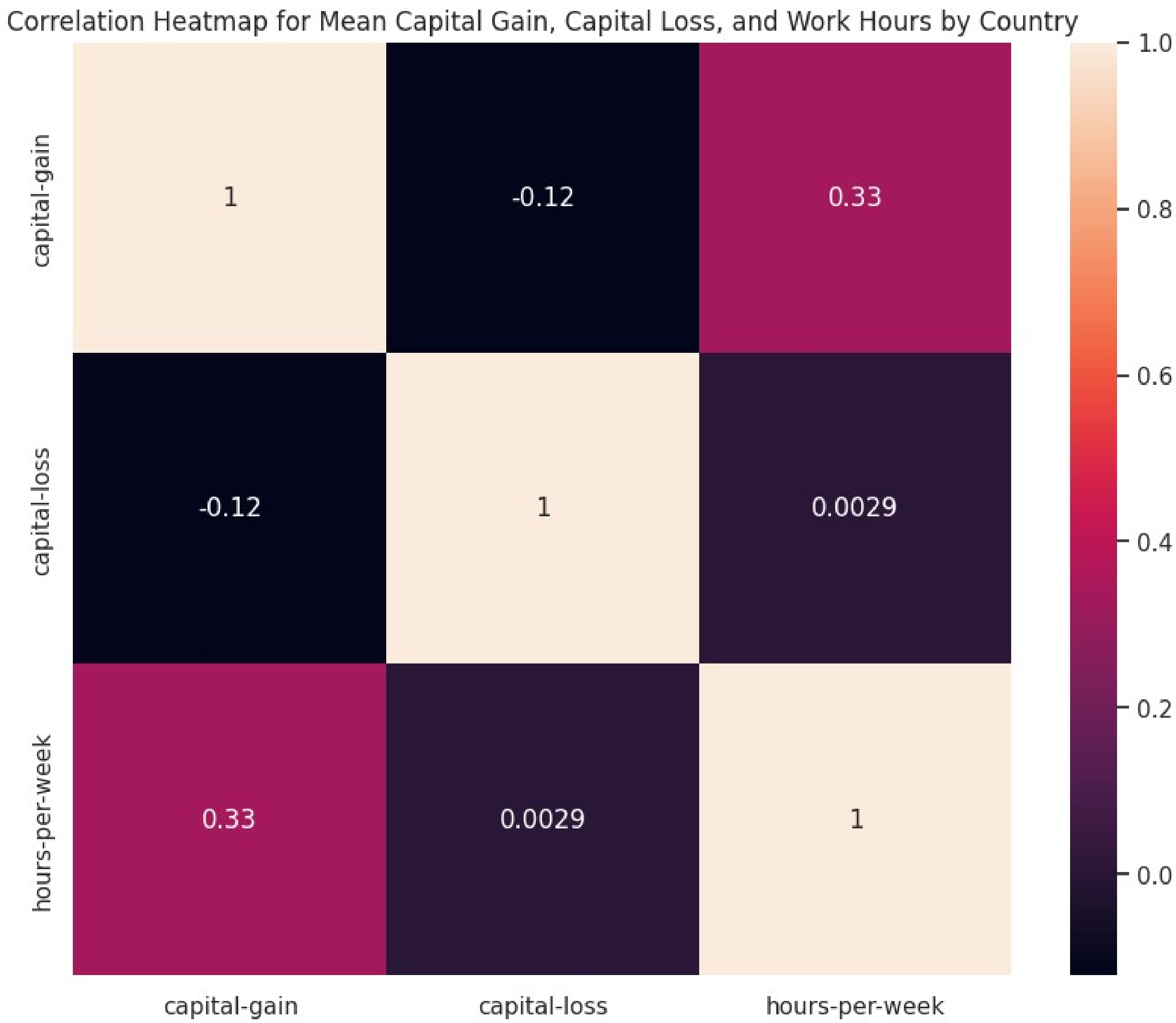
```python
import seaborn as sns
import matplotlib.pyplot as plt

# Plotting
plt.figure(figsize=(10, 8))

# Creating the box plot
sns.boxplot(x='occupation', y='age', hue='occupation', data=df,
palette='pastel')
plt.title('Ages of People with Occupation')
plt.xlabel('Occupation')
plt.ylabel('Age')

# Displaying the plot
plt.xticks(rotation=45)
plt.show()
```



With these box plot of ages of people with occupation we see that there is many outliers and the others have the max value surpasses others and it also surpasses everyone in the 75th percentile and the mean percentile of the armed forces is not the same as others while others are in the below 20 the armed forces is at above 20 below 30

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

# Calculating the correlation matrix
correlation_matrix = df.groupby('native-country').agg({'capital-gain':'mean', 'capital-loss':'mean', 'hours-per-week':'mean'}).corr()

# Plotting the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True)
```

```python
plt.title('Correlation Heatmap for Mean Capital Gain, Capital Loss,
and Work Hours by Country')
plt.show()
```

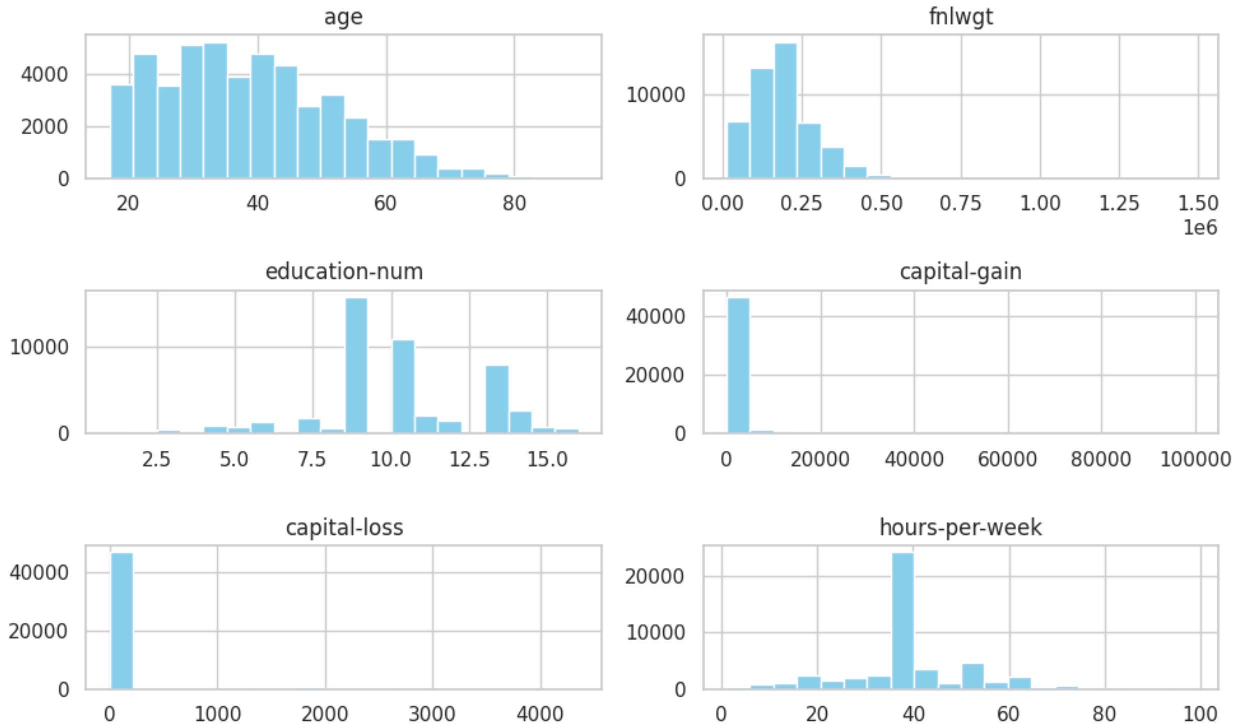


- The correlation coefficient between Hours Per Week and Capital Gain is 0.33. This indicates that there is a weak positive correlation between the number of hours worked per week and the amount of capital gain. In other words, as the number of hours worked per week increases, there tends to be a slight increase in capital gain, although the relationship is not very strong.

- The correlation coefficient between Hours Per Week and Capital Loss is 0.0029. This suggests that there is a very weak positive correlation between the number of hours worked per week and the amount of capital loss. The correlation is almost negligible, indicating that there is little to no relationship between these two variables.

- The correlation coefficient between Capital Gain and Capital Loss is -0.12. This indicates a very weak negative correlation between the amount of capital gain and the amount of capital loss. In other words, as capital gain increases, there is a slight decrease in capital loss, and vice versa. However, the correlation is quite weak.

```
import matplotlib.pyplot as plt
df.hist(bins=20, figsize=(10, 6), color='skyblue')
plt.tight_layout()
plt.show()
```



In conclusion, the analysis of various visualizations and statistical measures provides valuable insights economics of different occupations. Across the dataset, the distribution of income, capital gains, and work hours varies significantly among occupations, with notable differences observed across countries as well. While some occupations exhibit stronger correlations between certain variables, others show weaker or negligible relationships. Moreover, the presence of outliers in age distributions highlights the diversity within each occupation. Overall, these findings underscore the complexity economic dynamics, which shows the importance of considering multiple factors when analyzing work dynamics.