Assignment_3

January 27, 2026

[28]:
```
import pandas as pd

# Column names
column_names = [
    "age", "workclass", "fnlwgt", "education", "education_num",
    "marital_status", "occupation", "relationship", "race", "sex",
    "capital_gain", "capital_loss", "hours_per_week", "native_country", "income"
]

# Load the dataset using full path
adult = pd.read_csv(
    "/home/csl4/Desktop/adult.data",
    header=None,
    names=column_names,
    na_values=" ?"
)

# Check first few rows
print(adult.head())

# Check dataset info
print(adult.info())
```

```
   age          workclass  fnlwgt   education  education_num  \
0   39          State-gov   77516   Bachelors             13
1   50   Self-emp-not-inc   83311   Bachelors             13
2   38            Private  215646     HS-grad              9
3   53            Private  234721        11th              7
4   28            Private  338409   Bachelors             13

        marital_status          occupation    relationship    race      sex  \
0        Never-married        Adm-clerical   Not-in-family   White     Male
1   Married-civ-spouse     Exec-managerial         Husband   White     Male
2             Divorced   Handlers-cleaners   Not-in-family   White     Male
3   Married-civ-spouse   Handlers-cleaners         Husband   Black     Male
4   Married-civ-spouse      Prof-specialty            Wife   Black   Female

   capital_gain  capital_loss  hours_per_week  native_country  income
```

```
0          2174             0              40   United-States   <=50K
1             0             0              13   United-States   <=50K
2             0             0              40   United-States   <=50K
3             0             0              40   United-States   <=50K
4             0             0              40            Cuba   <=50K
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   workclass       30725 non-null  object
 2   fnlwgt          32561 non-null  int64
 3   education       32561 non-null  object
 4   education_num   32561 non-null  int64
 5   marital_status  32561 non-null  object
 6   occupation      30718 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital_gain    32561 non-null  int64
 11  capital_loss    32561 non-null  int64
 12  hours_per_week  32561 non-null  int64
 13  native_country  31978 non-null  object
 14  income          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
None
```

[56]:
```
bins = [0, 25, 45, 65, 100]
labels = ["Young", "Adult", "Middle", "Senior"]
adult["age_group"] = pd.cut(adult["age"], bins=bins, labels=labels)
print(adult[["age", "age_group"]].head(10))
```

```
   age age_group
0   39     Adult
1   50    Middle
2   38     Adult
3   53    Middle
4   28     Adult
5   37     Adult
6   49    Middle
7   52    Middle
8   31     Adult
9   42     Adult
```

[60]:
```
mean_hours = adult.groupby("age_group")["hours_per_week"].mean()
print("Mean hours per week by age group:\n", mean_hours)
```

```
Mean hours per week by age group:
 age_group
Young     33.893932
Adult     42.964292
Middle    42.020782
Senior    29.030225
Name: hours_per_week, dtype: float64
```

```
/tmp/ipykernel_6585/1451816249.py:1: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future version of
pandas. Pass observed=False to retain current behavior or observed=True to adopt
the future default and silence this warning.
  mean_hours = adult.groupby("age_group")["hours_per_week"].mean()
```

[34]:
```
median_hours = adult.groupby("age_group")["hours_per_week"].median()
print("Median hours per week by age group:\n", median_hours)
```

```
Median hours per week by age group:
 age_group
Young     40.0
Adult     40.0
Middle    40.0
Senior    30.0
Name: hours_per_week, dtype: float64
```

```
/tmp/ipykernel_6585/3580825962.py:1: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future version of
pandas. Pass observed=False to retain current behavior or observed=True to adopt
the future default and silence this warning.
  median_hours = adult.groupby("age_group")["hours_per_week"].median()
```

[36]:
```
min_hours = adult.groupby("age_group")["hours_per_week"].min()
print("Minimum hours per week by age group:\n", min_hours)
```

```
Minimum hours per week by age group:
 age_group
Young     1
Adult     1
Middle    1
Senior    1
Name: hours_per_week, dtype: int64
```

```
/tmp/ipykernel_6585/2663977078.py:1: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future version of
pandas. Pass observed=False to retain current behavior or observed=True to adopt
the future default and silence this warning.
```

```
  min_hours = adult.groupby("age_group")["hours_per_week"].min()
```

[38]:
```
max_hours = adult.groupby("age_group")["hours_per_week"].max()
print("Maximum hours per week by age group:\n", max_hours)
```

```
Maximum hours per week by age group:
 age_group
Young     99
Adult     99
Middle    99
Senior    99
Name: hours_per_week, dtype: int64
```

```
/tmp/ipykernel_6585/1693280925.py:1: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future version of
pandas. Pass observed=False to retain current behavior or observed=True to adopt
the future default and silence this warning.
  max_hours = adult.groupby("age_group")["hours_per_week"].max()
```

[40]:
```
std_hours = adult.groupby("age_group")["hours_per_week"].std()
print("Standard deviation of hours per week by age group:\n", std_hours)
```

```
Standard deviation of hours per week by age group:
 age_group
Young     12.431478
Adult     10.909220
Middle    11.603818
Senior    16.500961
Name: hours_per_week, dtype: float64
```

```
/tmp/ipykernel_6585/3456775204.py:1: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future version of
pandas. Pass observed=False to retain current behavior or observed=True to adopt
the future default and silence this warning.
  std_hours = adult.groupby("age_group")["hours_per_week"].std()
```

[42]:
```
mean_hours_list = adult.groupby("age_group")["hours_per_week"].mean().tolist()
print("List of mean hours per week by age group:", mean_hours_list)
```

```
List of mean hours per week by age group: [33.89393230385275,
42.964292198753256, 42.02078167434172, 29.030224525043177]
```

```
/tmp/ipykernel_6585/3582656014.py:1: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future version of
pandas. Pass observed=False to retain current behavior or observed=True to adopt
the future default and silence this warning.
  mean_hours_list = adult.groupby("age_group")["hours_per_week"].mean().tolist()
```

[44]:
```
# Group by age_group and calculate all required statistics
summary_stats = adult.groupby("age_group")["hours_per_week"].agg(
    mean="mean",
    median="median",
    minimum="min",
    maximum="max",
    std_dev="std"
)

print("Summary statistics of hours per week by age group:\n")
print(summary_stats)
```

```
Summary statistics of hours per week by age group:

                mean  median  minimum  maximum    std_dev
age_group
Young      33.893932    40.0        1       99  12.431478
Adult      42.964292    40.0        1       99  10.909220
Middle     42.020782    40.0        1       99  11.603818
Senior     29.030225    30.0        1       99  16.500961

/tmp/ipykernel_6585/51783839.py:2: FutureWarning: The default of observed=False
is deprecated and will be changed to True in a future version of pandas. Pass
observed=False to retain current behavior or observed=True to adopt the future
default and silence this warning.
  summary_stats = adult.groupby("age_group")["hours_per_week"].agg(
```

[ ]:
```
PART-II IRIS DATASET
```

[46]:
```
import pandas as pd

# Load iris.csv (update the path if necessary)
iris = pd.read_csv("/home/csl4/Desktop/iris.csv")
# Check first few rows
print(iris.head())
```

```
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
```

[48]:
```
# Filter by species
iris_setosa = iris[iris["species"] == "setosa"]
iris_versicolor = iris[iris["species"] == "versicolor"]
iris_virginica = iris[iris["species"] == "virginica"]
```

```
# Check the number of rows in each
print("Setosa samples:", len(iris_setosa))
print("Versicolor samples:", len(iris_versicolor))
print("Virginica samples:", len(iris_virginica))
```

```
Setosa samples: 50
Versicolor samples: 50
Virginica samples: 50
```

[54]:
```
# Function to display mean, std, and one percentile (50th) for numeric columns
def species_stats_single_percentile(df, species_name):
    numeric_df = df.select_dtypes(include="number")  # only numeric columns
    print(f"\nStatistics for {species_name}:\n")
    print("Mean:\n", numeric_df.mean())
    print("\nStandard Deviation:\n", numeric_df.std())
    print("\n50th Percentile (Median):\n", numeric_df.quantile(0.5))

# Setosa
species_stats_single_percentile(iris_setosa, "Iris-setosa")

# Versicolor
species_stats_single_percentile(iris_versicolor, "Iris-versicolor")

# Virginica
species_stats_single_percentile(iris_virginica, "Iris-virginica")
```

```
Statistics for Iris-setosa:

Mean:
 sepal_length    5.006
sepal_width     3.418
petal_length    1.464
petal_width     0.244
dtype: float64

Standard Deviation:
 sepal_length    0.352490
sepal_width     0.381024
petal_length    0.173511
petal_width     0.107210
dtype: float64

50th Percentile (Median):
 sepal_length    5.0
sepal_width     3.4
petal_length    1.5
petal_width     0.2
```

```
Name: 0.5, dtype: float64

Statistics for Iris-versicolor:

Mean:
 sepal_length    5.936
sepal_width     2.770
petal_length    4.260
petal_width     1.326
dtype: float64

Standard Deviation:
 sepal_length    0.516171
sepal_width     0.313798
petal_length    0.469911
petal_width     0.197753
dtype: float64

50th Percentile (Median):
 sepal_length    5.90
sepal_width     2.80
petal_length    4.35
petal_width     1.30
Name: 0.5, dtype: float64

Statistics for Iris-virginica:

Mean:
 sepal_length    6.588
sepal_width     2.974
petal_length    5.552
petal_width     2.026
dtype: float64

Standard Deviation:
 sepal_length    0.635880
sepal_width     0.322497
petal_length    0.551895
petal_width     0.274650
dtype: float64

50th Percentile (Median):
 sepal_length    6.50
sepal_width     3.00
petal_length    5.55
petal_width     2.00
Name: 0.5, dtype: float64
```

[ ]: