DS3030: Data Analytics Lab
Assignment 2

Date: Aug 11, 2025

Timing: 2:00 to 5:00 PM Max mark: 20

Instructions

- Submit one .ipynb file containing all answers named as [**student name**]_**assignment**[**number**]**.ipynb**
- Write the questions in **separate text blocks** before the answers.
- Write **justifications/comments** as required.

1 Part I

1. **Question 1: Data Generation using NumPy** (3)

   Set a random seed at the beginning: `np.random.seed(42)`.

   (a) Create the following NumPy arrays:

   1. `np1`: Integers from 1 to 100 (inclusive).
   2. `np2`: 100 samples from a normal distribution with mean = 170 and standard deviation = 10.
   3. `np3`: 100 samples from a uniform distribution between 60 and 90.
   4. `np4`: 100 random integers between 20,000 and 100,000 (inclusive).
   5. `np5`: Randomly assign one of the categories “HR”, “Sales”, “Tech”, or “Finance” to each element.

   Use appropriate NumPy functions for each step.

2. **Question 2: Create and Save a DataFrame** (2)

   Construct a pandas DataFrame using the arrays from Question 1, with columns:

   - `ID` (from `np1`)
   - `Height_Normal` (from `np2`)
   - `Weight_Uniform` (from `np3`)
   - `Salary_Random` (from `np4`)

- `Department` (from `np5`)

(a) Construct the DataFrame and display the first five rows.

(b) Save the DataFrame to `employee_data.csv` **without** the index column so the `ID` column remains the first column.

(c) Read back `employee_data.csv` into a new DataFrame named `df`. Show the code you used.

3. **Question 3: Data Analysis using pandas** (2)

Using the DataFrame `df` loaded from CSV:

(a) Print the mean and variance of all numeric columns.

(b) Display the count of each category in the `Department` column.

(c) Show the summary statistics of all numeric columns (i.e., using `df.describe()`).

(d) Identify which department has the highest average `Salary_Random`. (Show the code and the department name.)

2 Part II

COMPAS DATASET : compas-scores-two-years.csv

As quoted from Kaggle, "COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) is a popular commercial machine learning algorithm used by judges and parole officers for scoring criminal defendants' likelihood of reoffending (recidivism). It has been shown that the algorithm is biased in favor of white defendants and against black inmates, based on a 2 year follow up study.."

"Data contains variables used by the COMPAS algorithm in scoring defendants, along with their outcomes within 2 years of the decision"

For the tasks that follow, the variables '*age*', '*age_cat*', and '*race*' need to be used. The variable '*decile_score*' indicates the score indicating a person's likelihood of reoffending predicted by the algorithm, based on the values from other variables.

More Information can be found from
`https://www.kaggle.com/datasets/danofer/compass?resource=download`
`https://github.com/propublica/compas-analysis`

4. **Load the dataset to a pandas dataframe and print**
   a. Total Number of records (rows)
   b. Number of features (columns)
   c. Names of features and their types. **(2)**

5. **Perform Age Analysis Of Offenders by printing**
   a. Various age categories in the dataset
   b. Number of offenders above the age of 40 **(2)**

6. **Perform Race Analysis Of Offenders by printing**
   a. Count of offenders per race category
   b. Most frequent race in the dataset
   c. Average age per race **(3)**

3 Part III

Weather Data Analysis Using JSON
Dataset: `weather_data.json`

You are provided with a dataset containing a 16-day weather forecast for multiple cities. The data is stored in JSON format, where each entry corresponds to one city's weather data (including all 16 days).

7. **Load and Inspect Data** (2)
   (a) Load the JSON file containing multiple city weather records.
   (b) Create a DataFrame with the following columns: `city_name`, `country`, `longitude`, `latitude`, `date`, `day_temp`, `min_temp`, `max_temp`, `pressure`, `humidity`, `weather_main`.
   (c) Display the total number of rows in the DataFrame.
   (d) Display the count of unique cities present in the data.

8. **Filter and Subset Data** (2)
   (a) Filter the DataFrame to include only cities located within longitude 30° to 40° (European region).
   (b) For each city, retain data for only the first 7 days.

(c) Remove any rows where the `humidity` value is 0 (considered invalid readings).

(d) Create a subset consisting of exactly 2 cities and their corresponding 7-day data (total 14 rows).

9. **Temperature Conversion and Analysis** (2)

   (a) Convert the temperature columns (`day_temp`, `min_temp`, `max_temp`) from Kelvin to Celsius.

   (b) Calculate the daily temperature range (`max_temp` − `min_temp`) for each day.

   (c) Identify the city and date with the highest temperature range recorded.

   (d) Compute the mean day temperature (in Celsius) for each city over the 7-day period.

10. **Correlation Analysis (Not Evaluated)**

    (a) Select the numerical columns from the subset.

    (b) Compute the Pearson correlation matrix among these variables.

    (c) Identify which variable has the strongest correlation with the day temperature (in Celsius).

    (d) Report the correlation coefficient between `pressure` and `humidity`, rounded to three decimal places.