COMP 333 — Lab Assignment 2

Motivation

The purpose of this assignment is develop Python code for Descriptive Data Analysis.

It builds on the lectures of Week 3–4 and handles quantitative and visual descriptions.

You are to extend the capabilities of the `describe()` function of `pandas` and `scipy`, and to produce a grid of plots with univariate plots down the diagonal of the grid and bivariate plots off the diagonal.

Assignment

Create a Jupyter notebook using Python code and any of its libraries, but especially pandas, to write and test code to carry out the Descriptive Data Analysis tasks below in (1)–(3).

- Write your own Python functions `quantDDA()` and `vizDDA()` within the notebook, and illustrate their use within the notebook.
- Structure your code and document your work.
- Test your code on at least three examples of data.

Organize your notebook to clearly separate and identify your work on parts (1), (2), and (3).

(1) Quantitative Descriptions (6 marks) Write your own Python function `quantDDA()` to extend the capabilities of the `describe()` function of `pandas` and `scipy`. Your function should take a dataframe as input, and for each feature, should report

- number of observations
- number of entries
- number of unique entries
- number of missing entries
- number of outliers
- number of extreme values
- mode, or modes
- mean
- standard deviation
- max
- min
- Q3
- Q2 (median)
- Q1
- skewness
- kurtosis

Show your code working on at least three examples of data.

(2) Visual Descriptions (3 marks) Write your own Python function `vizDDA()` to produce a grid of plots. The grid is a square grid indexed by the features of the dataframe in both dimensions. Down the diagonal is a univariate plot for each feature; and each off-diagonal grid entry is a bivariate plot for the pair of features.

Remember to choose the plot that is appropriate to the type of data in the feature, or the type of data in each of the two features.

You may begin by assuming that you have only continuous data, so that the histogram is an appropriate univariate plot, and that the scatter plot is an appropriate bivariate plot. And then work from there to improve your code to consider other types of data.

Show your code working on at least three examples of data.

(3) Missing Values per Observation (1 marks) In preparation for Data Wrangling you want to know how many missing values there are in each observation, and you wish for the DDA to show the distribution of the number of missing values across the observations.

For your dataframe, engineer a new feature `MissingValueCount` which records the number of missing values in each observation as a new column in the dataframe.

Run your function `quantDDA()` on the modified dataframe. It should show a description of the distribution of missing values of the observations in the column for `MissingValueCount`.

Also run your function `vizDDA()` on the modified dataframe.

Deliverable

Your deliverable is the completed ipynb notebook showing all computation, output, and plots.

Remember that your notebook should clearly identify your work on parts (1), (2), and (3).