Python for Data Analysis
By: Atefeh Khazaei
Week#7 – Practicals[1]

1- How to count the number of missing values in each column?

Count the number of missing values in each column of df. Which column has the maximum number of missing values?

Input:

```
df = pd.read_csv('https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv')
```

2- How to replace missing values of multiple numeric columns with the mean?

Replace missing values in Min.Price and Max.Price columns with their respective mean.

Input:

```
df = pd.read_csv('https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv')
```

3- How to replace missing spaces in a string with the least frequent character?

Replace the spaces in my_str with the least frequent character.

Input:

```
my_str = 'dbc deb abed gadeg'
```

Desired Output:

```
'dbccdebcabedcgadeg'  # least frequent is 'c'
```

4- How to import only every nth row from a csv file to create a dataframe?

Import every 50th row of "BostonHousing dataset" as a dataframe.

BostonHousing dataset:

https://raw.githubusercontent.com/selva86/datasets/master/BostonHousing.csv

[1] Reference: https://www.machinelearningplus.com/python/101-pandas-exercises-python/

5- How to create one-hot encodings of a categorical variable (dummy variables)?

Get one-hot encodings for column 'countries' in the dataframe df and append it as columns.
Input: (df):

	Ids	Countries
0	11	Spain
1	22	France
2	33	Spain
3	44	Germany
4	55	France

Output:

	Country_France	Country_Germany	Country_Spain
0	0	0	1
1	1	0	0
2	0	0	1
3	0	1	0
4	1	0	0

6- How to normalize all columns in a dataframe?

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information. Normalization is also required for some algorithms to model the data correctly.

For example, assume your input dataset contains one column with values ranging from 0 to 1, and another column with values ranging from 10,000 to 100,000. The great difference in the scale of the numbers could cause problems when you attempt to combine the values as features during modeling.

Normalization avoids these problems by creating new values that maintain the general distribution and ratios in the source data, while keeping values within a scale applied across all numeric columns used in the model.[2]

There are different mathematical methods to normalize values. **Zscore** and B are two of the most common normalization methods.

- **Zscore**: Converts all values to a z-score. The values in the column are transformed using the following formula:

[2] https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/normalize-data

$$z = \frac{x - mean(x)}{stdev(x)}$$

Mean and standard deviation are computed for each column separately. Population standard deviation is used.

- **MinMax**: The min-max normalizer linearly usually rescales every feature to the [0,1] interval.

  Rescaling to the [0,1] interval is done by shifting the values of each feature so that the minimal value is 0, and then dividing by the new maximal value (which is the difference between the original maximal and minimal values).

  The values in the column are transformed using the following formula:

$$z = \frac{x - min(x)}{[\max(x) - min(x)]}$$

Question:

1- Normalize all columns of *df* by subtracting the column mean and divide by standard deviation.
2- Range all columns of *df* such that the minimum value in each column is 0 and max is 1.

Don't use external packages like sklearn.

Input:

```
df = pd.DataFrame(np.random.randint(1,100, 80).reshape(8, -1))
```