Name : Tanishq Thuse

Div : SY CSAI - B

Roll no. : 60

PRN : 12310237

Subject : DV Assignment - 2

Assignment - 2

Problem Statement : Data Wrangling, IN Perform the following operations using Python on any open source dataset (e.g., data.csv)

a. Import all the required Python Libraries.

b. Locate open-source data from the web (e.g., https://www.kaggle.com). Provide a clear description of the data and its source (i.e., URL of the web site).

c. Load the Dataset into pandas data frame.

d. Data Preprocessing: check for missing values in the data using pandas isnull(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame.

e. Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.

f. Turn categorical variables into quantitative variables in Python. Practical based on Data Loading, Storage and File Formats

Dataset Link : https://www.kaggle.com/datasets/joebeachcapital/carbon-majors-emissions-data

The DataSet has 3 files : 1)emissions_high_granularity.csv 2)emissions_low_granularity.csv 3)emissions_medium_granularity.csv

For this Assignment I am going to refer to 1)emissions_high_granularity.csv

Importing Libraries

```
# a) Import all the required Python Libraries.
import pandas as pd
```

```
import numpy as np
from matplotlib import pyplot as plt
```

```
file = "/content/emissions_high_granularity.csv"
df = pd.read_csv(file)
#I have to upload the file manually before running the cells
```

Exploratory Data Analysis of Kaggle Dataset

```
df.head()
```

	year	parent_entity	parent_type	reporting_entity	commodity	production_value	prod
0	1962	Abu Dhabi National Oil Company	State-owned Entity	Abu Dhabi	Oil & NGL	0.9125	
1	1963	Abu Dhabi National Oil Company	State-owned Entity	Abu Dhabi	Oil & NGL	1.8250	
2	1964	Abu Dhabi National Oil Company	State-owned Entity	Abu Dhabi	Oil & NGL	7.3000	
3	1965	Abu Dhabi National Oil Company	State-owned Entity	Abu Dhabi	Oil & NGL	10.9500	
4	1966	Abu Dhabi National Oil Company	State-owned Entity	Abu Dhabi	Oil & NGL	13.5050	

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15797 entries, 0 to 15796
Data columns (total 16 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
```

```
 0   year                                15797 non-null  int64
 1   parent_entity                       15797 non-null  object
 2   parent_type                         15797 non-null  object
 3   reporting_entity                    15797 non-null  object
 4   commodity                           15797 non-null  object
 5   production_value                    15797 non-null  float64
 6   production_unit                     15797 non-null  object
 7   product_emissions_MtCO2             15797 non-null  float64
 8   flaring_emissions_MtCO2             15797 non-null  float64
 9   venting_emissions_MtCO2             15797 non-null  float64
 10  own_fuel_use_emissions_MtCO2        15797 non-null  float64
 11  fugitive_methane_emissions_MtCO2e   15797 non-null  float64
 12  fugitive_methane_emissions_MtCH4    15797 non-null  float64
 13  total_operational_emissions_MtCO2e  15797 non-null  float64
 14  total_emissions_MtCO2e              15797 non-null  float64
 15  source                              15797 non-null  object
dtypes: float64(9), int64(1), object(6)
memory usage: 1.9+ MB
```

```
df.describe()
```

	year	production_value	product_emissions_MtCO2	flaring_emissions_MtCO2
count	15797.000000	15797.000000	15797.000000	15797.000000
mean	1985.827942	327.879634	79.391514	0.517226
std	28.664256	1188.625001	261.984080	1.783744
min	1854.000000	0.000000	0.000000	0.000000
25%	1970.000000	11.800000	5.996490	0.000000
50%	1993.000000	59.970871	21.502409	0.015913
75%	2007.000000	246.375000	62.191954	0.197253
max	2022.000000	27192.000000	7769.222235	27.026872

```
df.isnull()
```

	year	parent_entity	parent_type	reporting_entity	commodity	production_value
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...	...	...	...	...	...	...
15792	False	False	False	False	False	False
15793	False	False	False	False	False	False
15794	False	False	False	False	False	False
15795	False	False	False	False	False	False
15796	False	False	False	False	False	False

15797 rows × 16 columns

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

```
year                                  0
parent_entity                         0
parent_type                           0
reporting_entity                      0
commodity                             0
production_value                      0
production_unit                       0
product_emissions_MtCO2               0
flaring_emissions_MtCO2               0
venting_emissions_MtCO2               0
own_fuel_use_emissions_MtCO2          0
fugitive_methane_emissions_MtCO2e     0
fugitive_methane_emissions_MtCH4      0
total_operational_emissions_MtCO2e    0
total_emissions_MtCO2e                0
source                                0
dtype: int64
```

Object Creation

```
# df = pd.Series()
# print(df)

a = [1,2,3,4,5,6,7,8,9]
```

```
DF = pd.Series(a)
print(DF)
```

```
0    1
1    2
2    3
3    4
4    5
5    6
6    7
7    8
8    9
dtype: int64
```

```
dataFrame = {'Roll no': [1,2,3,4,5] ,
             'Name': ['Tanishq', 'Ayush', 'Aditya', 'Shantanu', 'Yash'],
             'DV': ['100', '98', '97', '91','95'],
            'OS' : ['98', '95', '90', '89', '83'],
             'AI' : ['98', '95', '90', '89', '83'],
             'ADS' : ['98', '95', '90', '89', '83']
}

df = pd.DataFrame(dataFrame)
print(df)

# calories = {"day1": 420, "day2": 380, "day3": 390}

# myvar = pd.Series(calories, index = ["day1", "day2"])

# print(myvar)
```

```
   Roll no      Name   DV  OS  AI ADS
0        1   Tanishq  100  98  98  98
1        2     Ayush   98  95  95  95
2        3    Aditya   97  90  90  90
3        4  Shantanu   91  89  89  89
4        5      Yash   95  83  83  83
```

```
df.dtypes
```

```
Roll no     int64
Name       object
DV         object
OS         object
AI         object
ADS        object
dtype: object
```

Viewing Data

```
df.head()
```

	Roll no	Name	DV	OS	AI	ADS
0	1	Tanishq	100	98	98	98
1	2	Ayush	98	95	95	95
2	3	Aditya	97	90	90	90
3	4	Shantanu	91	89	89	89
4	5	Yash	95	83	83	83

```
print(df.loc[0])
```

```
Roll no          1
Name       Tanishq
DV             100
OS              98
AI              98
ADS             98
Name: 0, dtype: object
```

```
print(df.iloc[ 0 : 1])
dataFrame = df
```

```
   Roll no     Name   DV  OS  AI ADS
0        1  Tanishq  100  98  98  98
```

```
# print(df.loc['0'])
```

```
dataFrame.isnull()
```

	Roll no	Name	DV	OS	AI	ADS
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False

```
dataFrame.dropna()
```

	Roll no	Name	DV	OS	AI	ADS
0	1	Tanishq	100	98	98	98
1	2	Ayush	98	95	95	95
2	3	Aditya	97	90	90	90
3	4	Shantanu	91	89	89	89
4	5	Yash	95	83	83	83

```
print(dataFrame.loc[0])
```

```
Roll no          1
Name       Tanishq
DV             100
OS              98
AI              98
ADS             98
Name: 0, dtype: object
```

```
dataFrame
```

	Roll no	Name	DV	OS	AI	ADS
0	1	Tanishq	100	98	98	98
1	2	Ayush	98	95	95	95
2	3	Aditya	97	90	90	90
3	4	Shantanu	91	89	89	89
4	5	Yash	95	83	83	83

```
dates = pd.date_range("20130101", periods=6)
dates
```

```
DatetimeIndex(['2013-01-01', '2013-01-02', '2013-01-03', '2013-01-04',
               '2013-01-05', '2013-01-06'],
              dtype='datetime64[ns]', freq='D')
```

```
dataFrame
```

	Roll no	Name	DV	OS	AI	ADS
0	1	Tanishq	100	98	98	98
1	2	Ayush	98	95	95	95
2	3	Aditya	97	90	90	90
3	4	Shantanu	91	89	89	89
4	5	Yash	95	83	83	83

```
dataFrame.dtypes
```

```
Roll no     int64
Name       object
DV         object
OS         object
AI         object
ADS        object
dtype: object
```

```
df = dataFrame
df.dtypes
```

```
Roll no     int64
Name       object
DV         object
OS         object
AI         object
ADS        object
dtype: object
```

```
df.T
```

	0	1	2	3	4
Roll no	1	2	3	4	5
Name	Tanishq	Ayush	Aditya	Shantanu	Yash
DV	100	98	97	91	95
OS	98	95	90	89	83
AI	98	95	90	89	83
ADS	98	95	90	89	83

```
df.describe()
```

	Roll no
count	5.000000
mean	3.000000
std	1.581139
min	1.000000
25%	2.000000
50%	3.000000
75%	4.000000
max	5.000000

```
df.isnull()
```

	Roll no	Name	DV	OS	AI	ADS
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False

Selection

```
#TO sort by values in DV Subject
df.sort_values(by='DV')
```

	Roll no	Name	DV	OS	AI	ADS
0	1	Tanishq	100	98	98	98
3	4	Shantanu	91	89	89	89
4	5	Yash	95	83	83	83
2	3	Aditya	97	90	90	90
1	2	Ayush	98	95	95	95

```
df['DV']
```

```
0    100
1     98
2     97
3     91
4     95
Name: DV, dtype: object
```

```
df[0:3]
```

	Roll no	Name	DV	OS	AI	ADS
0	1	Tanishq	100	98	98	98
1	2	Ayush	98	95	95	95
2	3	Aditya	97	90	90	90

Selection by label

```
df.loc[:, ['Name','DV' ]]
```

	Name	DV
0	Tanishq	100
1	Ayush	98
2	Aditya	97
3	Shantanu	91
4	Yash	95

```
df.loc[0:2, ['Name','DV', 'ADS']]
```

	Name	DV	ADS
0	Tanishq	100	98
1	Ayush	98	95
2	Aditya	97	90

Selection by position

```
df.iloc[3]
```

```
Roll no           4
Name       Shantanu
DV               91
OS               89
AI               89
ADS              89
Name: 3, dtype: object
```

```
df.iloc[0:1,:]
```

	Roll no	Name	DV	OS	AI	ADS
0	1	Tanishq	100	98	98	98

```
df
```

	Roll no	Name	DV	OS	AI	ADS
0	1	Tanishq	100	98	98	98
1	2	Ayush	98	95	95	95
2	3	Aditya	97	90	90	90
3	4	Shantanu	91	89	89	89
4	5	Yash	95	83	83	83

```
#A python function to assign DV Subject grade
def grade_dv(marks):
    marks = int(marks)
    if marks > 95:
        return 'A+'
    elif marks > 90 :
        return 'A'
    else:
        return 'B'

# Apply the grading function to the DV marks
df['DV Grade'] = df['DV'].apply(grade_dv)

# Display the updated DataFrame
print(df)
```

```
   Roll no      Name   DV  OS  AI ADS DV Grade
0        1   Tanishq  100  98  98  98       A+
1        2     Ayush   98  95  95  95       A+
2        3    Aditya   97  90  90  90       A+
3        4  Shantanu   91  89  89  89        A
4        5      Yash   95  83  83  83        A
```

Getting

```
df["Name"]
```

```
0     Tanishq
1       Ayush
2      Aditya
3    Shantanu
4        Yash
Name: Name, dtype: object
```

```
df[0:3]
```

	Roll no	Name	DV	OS	AI	ADS	DV Grade
0	1	Tanishq	100	98	98	98	A+
1	2	Ayush	98	95	95	95	A+
2	3	Aditya	97	90	90	90	A+

```
df[0:4]
```

	Roll no	Name	DV	OS	AI	ADS	DV Grade
0	1	Tanishq	100	98	98	98	A+
1	2	Ayush	98	95	95	95	A+
2	3	Aditya	97	90	90	90	A+
3	4	Shantanu	91	89	89	89	A

Selection by label

```
df.head() #having a look at the dataframe
```

```
df.loc[:, ["Name", "DV"]]
```

	Name	DV
0	Tanishq	100
1	Ayush	98
2	Aditya	97
3	Shantanu	91
4	Yash	95

```
df.loc[0:2, ["Name", "DV"]]
```