**Expt. No.:03 Date: 29/09/2020**

**READING AND SUMMARIZING DATA USING PYTHON**

**AIM:**

1. **Reading and Evaluating Datasets:**
   1. Read csv files
   2. Summarize the dataset
   3. Count unique values in each column
2. **Computing Summary statistics:** Compute mean, median, mode, range, variance of every column

**THEORY:** Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.

Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Key Features of Pandas

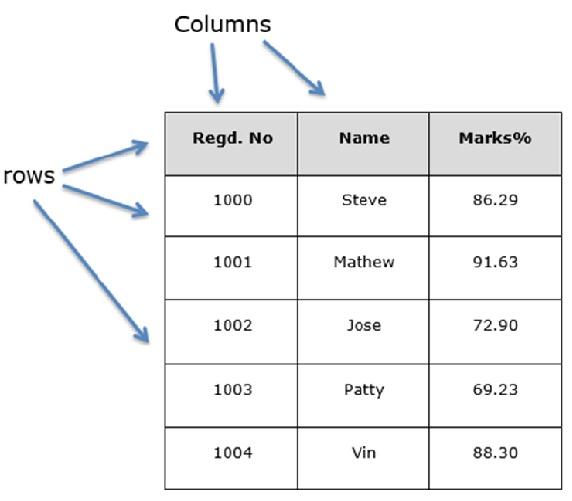
* Fast and efficient DataFrame object with default and customized indexing.
* Tools for loading data into in-memory data objects from different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of date sets.
* Label-based slicing, indexing and subsetting of large data sets.
* Columns from a data structure can be deleted or inserted.
* Group by data for aggregation and transformations.
* High performance merging and joining of data.
* Time Series functionality.

A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.

### Features of DataFrame:

* Potentially columns are of different types
* Size – Mutable
* Labeled axes (rows and columns)
* Can Perform Arithmetic operations on rows and columns

### Structure:



You can think of it as an SQL table or a spreadsheet data representation.

## pandas.DataFrame

A pandas DataFrame can be created using the following constructor −

pandas.DataFrame( data, index, columns, dtype, copy)

The parameters of the constructor are as follows −

|  |  |
| --- | --- |
| **Sr.No** | **Parameter & Description** |
| 1 | **data**  data takes various forms like ndarray, series, map, lists, dict, constants and also another DataFrame. |
| 2 | **index**  For the row labels, the Index to be used for the resulting frame is Optional Default np.arange(n) if no index is passed. |
| 3 | **columns**  For column labels, the optional default syntax is - np.arange(n). This is only true if no index is passed. |
| 4 | **dtype**  Data type of each column. |
| 5 | **copy**  This command (or whatever it is) is used for copying of data, if the default is False. |

**Functions & Description**

Following is a list of methods available for descriptive statistics:

|  |  |  |
| --- | --- | --- |
| Sr.No. | Function | Description |
| 1 | count() | Number of non-null observations |
| 2 | sum() | Sum of values |
| 3 | mean() | Mean of Values |
| 4 | median() | Median of Values |
| 5 | mode() | Mode of values |
| 6 | std() | Standard Deviation of the Values |
| 7 | min() | Minimum Value |
| 8 | max() | Maximum Value |
| 9 | abs() | Absolute Value |
| 10 | prod() | Product of Values |
| 11 | cumsum() | Cumulative Sum |
| 12 | cumprod() | Cumulative Product |

Note − Since DataFrame is a Heterogeneous data structure. Generic operations don’t work with all functions.

* Functions like sum(), cumsum() work with both numeric and character (or) string data elements without any error. Though n practice, character aggregations are never used generally, these functions do not throw any exception.
* Functions like abs(), cumprod() throw exception when the DataFrame contains character or string data because such operations cannot be performed.

**Summarizing Data**

The describe() function computes a summary of statistics pertaining to the DataFrame columns. This function gives the mean, std and IQR values. And, function excludes the character columns and given summary about numeric columns. 'include' is the argument which is used to pass necessary information regarding what columns need to be considered for summarizing. Takes the list of values; by default, 'number'.

* **object** − Summarizes String columns
* **number** − Summarizes Numeric columns
* **all** − Summarizes all columns together (Should not pass it as a list value)

**PROGRAM:**

**i)**

import numpy as np

import pandas as pd

cereal\_data = pd.read\_csv("cereal.csv") #read the dataset

cereal\_data.head()

cereal\_data.isna().sum()

cereal\_data.head()

cereal\_data.dropna(inplace=True)

print("Summary of the basic information about this DataFrame and its data:")

print(cereal\_data.info()) #summarize the dataset

cereal\_data.nunique(axis=0) # number of unique values in every column

**ii)**

print("\n----------- Calculate Mean -----------\n")

print(cereal\_data.mean())

print("\n----------- Calculate Median -----------\n")

print(cereal\_data.median())

print("\n----------- Calculate Mode -----------\n")

mode\_data.fillna('-')

mode\_data=cereal\_data.iloc[:,1:15].mode()

print("\n----------- Calculate Range -----------\n")

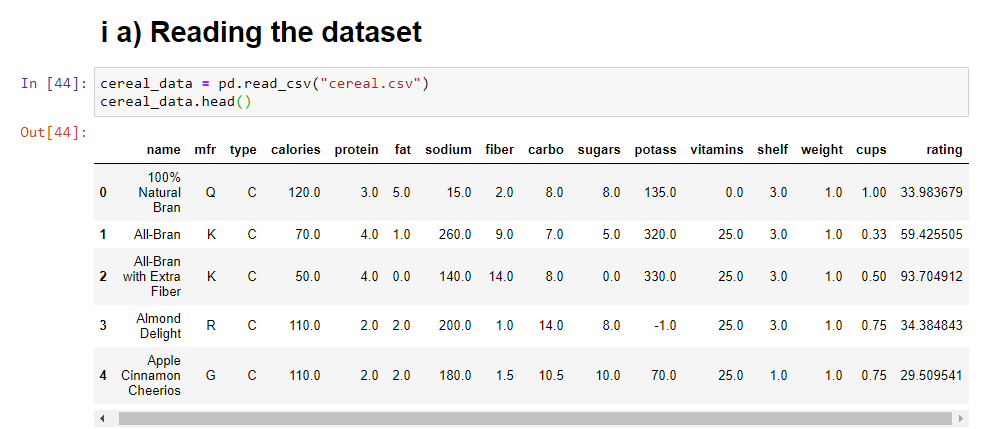
cereal\_data\_non\_cat = cereal\_data.drop(columns=['name','mfr','type'])

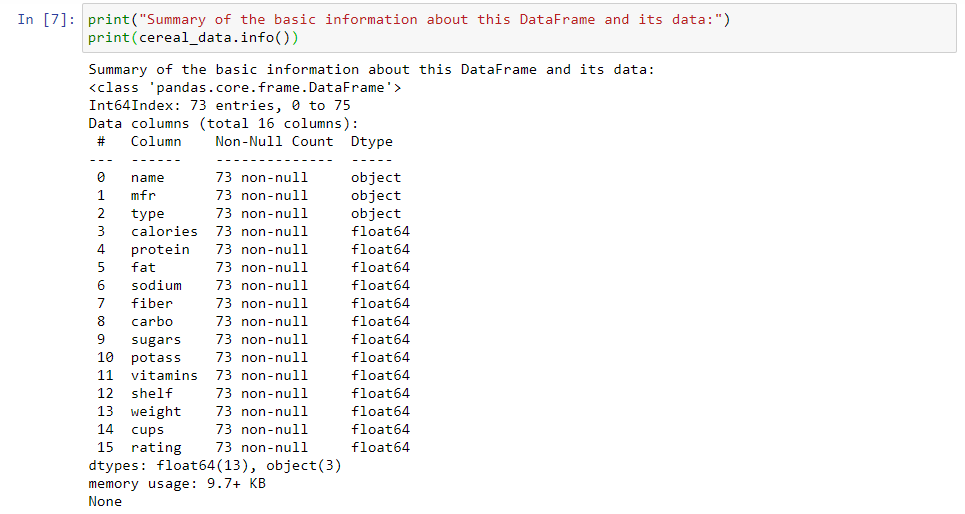
print(cereal\_data\_non\_cat.max()-cereal\_data\_non\_cat.min())

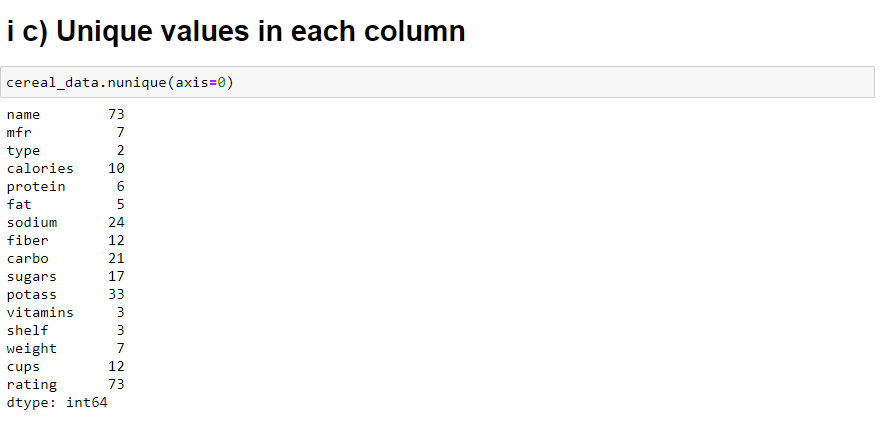
print("\n----------- Calculate Variance -----------\n")

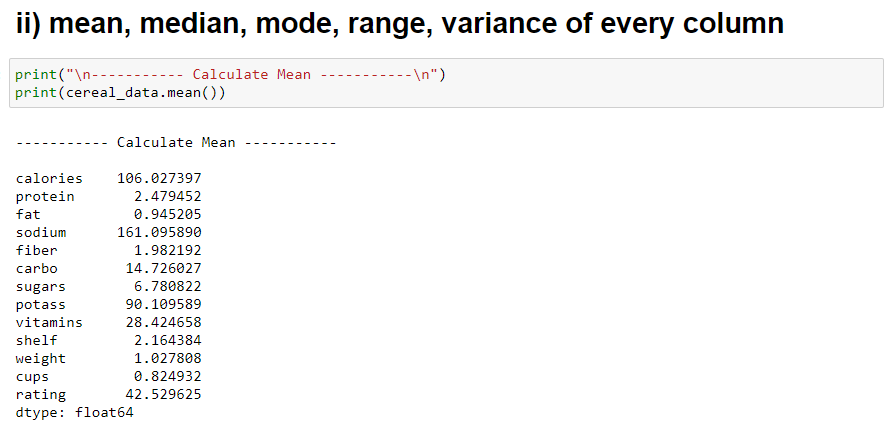
cereal\_data.var()

**OUTPUT:**

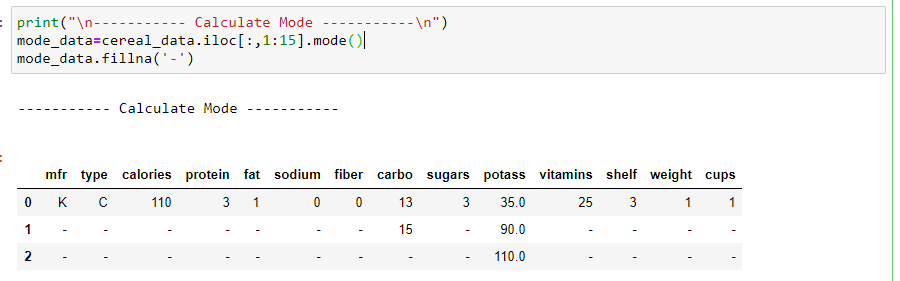
****

****

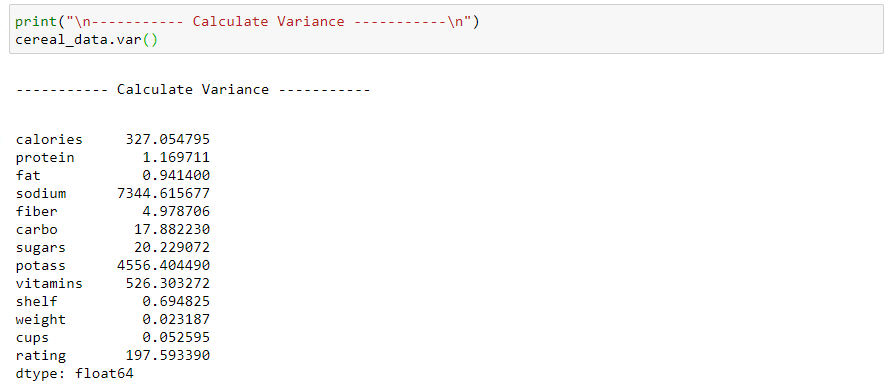
****

****

****

****

****

****

**CONCLUSION:**

Reading and summarizing of cereal dataset was completed successfully using python.