Assignment 1

# Aim:- Write a python script to find basic descriptive statistics using summary, str, quartile function on a dataset.

Theory:-

Mean

The Mean value of a dataset is the average value i.e. a number around which a whole data is spread out. All values used in calculating the average are weighted equally when defining the Mean.

Median

Median is the middle value of the dataset i.e if we sort the data from smallest to biggest (or biggest to smallest) and then take the value in the middle of the set: that’s the Median.

Mode

The mode is the value that appears most frequently in a data set. A set of data may have one mode, more than one mode, or no mode at all. Other popular measures of central tendency include the mean, or the average of a set, and the median, the middle value in a set.

Variance

Variance is another number that indicates how spread out the values are.

In fact, if you take the square root of the variance, you get the standard deviation. Or the other way around, if you multiply the standard deviation by itself, you get the variance!

Standard Deviation

Standard deviation is a measure of uncertainty.

A low standard deviation means that most of the numbers are close to the mean (average) value.

A high standard deviation means that the values are spread out over a wider range.

Interquartile Range

IQR (Interquartile Range) is the difference between the third and the first quartile of a distribution (or the 75th percentile minus the 25th percentile). It is a measure of how wide our distribution is since this range contains half of the points of the dataset. It’s very useful to make an idea of the shape of the distribution. For example, it is the width of the boxes in the boxplot.

Mean Absolute Error

In the context of machine learning, absolute error refers to the magnitude of difference between the prediction of an observation and the true value of that observation. MAE takes the average of absolute errors for a group of predictions and observations as a measurement of the magnitude of errors for the entire group. MAE can also be referred as L1 loss function.

#Importing the required libraries import numpy as np

import pandas as pd import statistics

from google.colab import drive drive.mount('/content/drive')

#Loading the dataset

data = pd.read\_csv('/content/drive/MyDrive/Ass1/ustraffic.csv')

#Displaying the information of the dataset data.info()

 Drive already mounted at /content/drive; to attempt to forcibly remount, call

<class 'pandas.core.frame.DataFrame'> RangeIndex: 336 entries, 0 to 335 Data columns (total 35 columns):

# Column Non-Null Count Dtype

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 |  | Unnamed: 0 | 336 | non-null |  | int64 |
| 1 |  | state | 336 | non-null |  | object |
| 2 |  | year | 336 | non-null |  | int64 |
| 3 |  | spirits | 336 | non-null |  | float64 |
| 4 |  | unemp | 336 | non-null |  | float64 |
| 5 |  | income | 336 | non-null |  | float64 |
| 6 |  | emppop | 336 | non-null |  | float64 |
| 7 |  | beertax | 336 | non-null |  | float64 |
| 8 |  | baptist | 336 | non-null |  | float64 |
| 9 |  | mormon | 336 | non-null |  | float64 |
| 10 |  | drinkage | 336 | non-null |  | float64 |
| 11 |  | dry | 336 | non-null |  | float64 |
| 12 |  | youngdrivers | 336 | non-null |  | float64 |
| 13 |  | miles | 336 | non-null |  | float64 |
| 14 |  | breath | 336 | non-null |  | object |
| 15 |  | jail | 335 | non-null |  | object |
| 16 |  | service | 335 | non-null |  | object |
| 17 |  | fatal | 336 | non-null |  | int64 |
| 18 |  | nfatal | 336 | non-null |  | int64 |
| 19 |  | sfatal | 336 | non-null |  | int64 |
| 20 |  | fatal1517 | 336 | non-null |  | int64 |
| 21 |  | nfatal1517 | 336 | non-null |  | int64 |
| 22 |  | fatal1820 | 336 | non-null |  | int64 |
| 23 |  | nfatal1820 | 336 | non-null |  | int64 |
| 24 |  | fatal2124 | 336 | non-null |  | int64 |
| 25 |  | nfatal2124 | 336 | non-null |  | int64 |
| 26 |  | afatal | 336 | non-null |  | float64 |
| 27 |  | pop | 336 | non-null |  | float64 |
| 28 |  | pop1517 | 336 | non-null |  | float64 |
| 29 |  | pop1820 | 336 | non-null |  | float64 |
| 30 |  | pop2124 | 336 | non-null |  | float64 |
| 31 |  | milestot | 336 | non-null |  | float64 |
| 32 |  | unempus | 336 | non-null |  | float64 |
| 33 |  | emppopus | 336 | non-null |  | float64 |
| 34 |  | gsp | 336 | non-null |  | float64 |

dtypes: float64(20), int64(11), object(4) memory usage: 92.0+ KB

#Display number of columns and names of the columns present in the dataset data.columns

Index(['Unnamed: 0', 'state', 'year', 'spirits', 'unemp', 'income', 'emppop', 'beertax', 'baptist', 'mormon', 'drinkage', 'dry', 'youngdrivers', 'miles', 'breath', 'jail', 'service', 'fatal', 'nfatal', 'sfatal', 'fatal1517', 'nfatal1517', 'fatal1820', 'nfatal1820', 'fatal2124', 'nfatal2124', 'afatal', 'pop', 'pop1517', 'pop1820', 'pop2124',

'milestot', 'unempus', 'emppopus', 'gsp'], dtype='object')

#Finding the measures of central tendency of the whole dataset print(data.mean())

print(data.median()) print(data.mode())

mormon drinkage dry

youngdrivers miles

fatal nfatal sfatal fatal1517 nfatal1517 fatal1820 nfatal1820 fatal2124 nfatal2124 afatal

pop

pop1517

3.931110e-01

2.100000e+01

8.681250e-02

1.853875e-01

7.796219e+03

7.010000e+02

1.350000e+02

8.100000e+01

4.900000e+01

1.000000e+01

8.200000e+01

2.400000e+01

9.750000e+01

3.000000e+01

2.115940e+02

3.310503e+06

1.630002e+05

#Finding measures of central tendency for specific columns print(data['unemp'].mean())

print(data['unemp'].median())

print(data['unemp'].mode())

7.346726204667773

7.0

0 6.3

dtype: float64

#Finding measures of variablility

print("Variance of unemp column is", data['unemp'].var()) print("Standard deviation of unemp column is", data['unemp'].std()) print("Meannabsolute deviation of unemp column is", data['unemp'].mad())

#Finding range of columns

print("Range of fatal column is", (max(data['fatal'])-min(data['fatal']))) print("Range of nfatal column is", (max(data['nfatal'])-min(data['nfatal']))) print("Range of sfatal column is", (max(data['sfatal'])-min(data['sfatal'])))

Variance of unemp column is 6.418138471719675

Standard deviation of unemp column is 2.533404521926902 Meannabsolute deviation of unemp column is 2.0235561961118056 Range of fatal column is 5425

Range of nfatal column is 1036 Range of sfatal column is 595

#Finding the Inter Quatile range of individual columns Q1 = np.percentile(data['fatal'], 25)

Q3 = np.percentile(data['fatal'], 75) print("First Quartile of fatal column is", Q1) print("Third Quartile of fatal column is", Q3)

IQR1 = Q3-Q1

print("Inter Quartile range of fatal column is", IQR1) print("\n")

R1 = np.percentile(data['nfatal'], 25) R3 = np.percentile(data['nfatal'], 75)

print("First Quartile of nfatal column is", R1) print("Third Quartile of nfatal column is", R3)

IQR2 = R3-R1

print("Inter Quartile range of nfatal column is", IQR2) print("\n")

S1 = np.percentile(data['sfatal'], 25) S3 = np.percentile(data['sfatal'], 75)

print("First Quartile of sfatal column is", S1) print("Third Quartile of sfatal column is", S3)

IQR3 = S3-S1

print("Inter Quartile range of sfatal column is", IQR3) print("\n")

First Quartile of fatal column is 293.75 Third Quartile of fatal column is 1063.5

Inter Quartile range of fatal column is 769.75

First Quartile of nfatal column is 53.75 Third Quartile of nfatal column is 212.0

Inter Quartile range of nfatal column is 158.25

First Quartile of sfatal column is 35.0 Third Quartile of sfatal column is 131.0

Inter Quartile range of sfatal column is 96.0

#Using the statistics library to find mean and variance import statistics

print("Mean of unemp column is", statistics.mean(data['unemp'])) print("Mean of income column is", statistics.mean(data['income'])) print("Mean of emppop column is", statistics.mean(data['emppop'])) print("Mean of beertax column is", statistics.mean(data['beertax']))

Mean of unemp column is 7.346726204667772 Mean of income column is 13880.184532528832 Mean of emppop column is 60.80567565418425 Mean of beertax column is 0.513