

Lab Assignment-3

Descriptive Statistics - Measures of Central Tendency and variability

Perform the following operations on any open source dataset (e.g., data.csv)

1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable.

```
import pandas as pd
import numpy as np
import statistics as st
```

```
df = pd.read_csv("loan.csv")
```

```
print(df.shape)
(500, 15)
```

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Unnamed: 0            500 non-null   int64  
1   Loan_ID               500 non-null   object  
2   Gender                491 non-null   object  
3   Married               497 non-null   object  
4   Dependents            488 non-null   object  
5   Education             500 non-null   object  
6   Self_Employed         473 non-null   object  
7   ApplicantIncome       500 non-null   int64  
8   CoapplicantIncome     500 non-null   float64 
9   LoanAmount            482 non-null   float64 
10  Loan_Amount_Term      486 non-null   float64
```

```
11 Credit_History    459 non-null    float64
12 Property_Area     500 non-null    object
13 Loan_Status       500 non-null    object
14 Total_Income      500 non-null    object
dtypes: float64(4), int64(2), object(9)
memory usage: 58.7+ KB
None
```

```
df.mean()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
```

```
"""Entry point for launching an IPython kernel.
```

```
Unnamed: 0          249.500000
ApplicantIncome     5493.644000
CoapplicantIncome   1506.307840
LoanAmount          144.020747
Loan_Amount_Term    342.543210
Credit_History      0.843137
dtype: float64
```

```
df.max()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
```

```
"""Entry point for launching an IPython kernel.
```

```
Unnamed: 0          499
Loan_ID            LP002602
Education          Not Graduate
ApplicantIncome     81000
CoapplicantIncome   20000.0
LoanAmount          700.0
Loan_Amount_Term    480.0
Credit_History      1.0
Property_Area       Urban
Loan_Status         Y
Total_Income        $9993.0
dtype: object
```

```
df.min()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of
```

nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.

"""Entry point for launching an IPython kernel.

```
Unnamed: 0      0
Loan_ID      LP001002
Education    Graduate
ApplicantIncome    150
CoapplicantIncome    0.0
LoanAmount      17.0
Loan_Amount_Term    12.0
Credit_History    0.0
Property_Area    Rural
Loan_Status      N
Total_Income    $10000.0
dtype: object
```

df.median()

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.

"""Entry point for launching an IPython kernel.

```
Unnamed: 0      249.5
ApplicantIncome    3854.0
CoapplicantIncome    1125.5
LoanAmount      126.5
Loan_Amount_Term    360.0
Credit_History      1.0
dtype: float64
```

df.std()

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.

"""Entry point for launching an IPython kernel.

```
Unnamed: 0      144.481833
ApplicantIncome    6515.668972
CoapplicantIncome    2134.432188
LoanAmount      82.344919
Loan_Amount_Term    63.834977
Credit_History      0.364068
dtype: float64
```

```
print(df.loc[:, 'LoanAmount'].mean())  
144.0207468879668
```

```
df.mean(axis = 1)[0:5]  
0    1242.000000  
1    1096.833333  
2     571.500000  
3     904.166667  
4    1084.333333  
dtype: float64
```

```
print(df.loc[:, 'LoanAmount'].median())  
126.5
```

```
df.median(axis = 1)[0:5]  
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of  
nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a  
future version this will raise TypeError. Select only valid columns before calling the reduction.  
    """Entry point for launching an IPython kernel.  
0     1.0  
1    244.0  
2     34.0  
3    240.0  
4     72.5  
dtype: float64
```

```
df.var()  
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of  
nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a  
future version this will raise TypeError. Select only valid columns before calling the reduction.  
    """Entry point for launching an IPython kernel.  
Unnamed: 0      2.087500e+04  
ApplicantIncome  4.245394e+07  
CoapplicantIncome 4.555801e+06  
LoanAmount      6.780686e+03  
Loan_Amount_Term 4.074904e+03  
Credit_History  1.325456e-01  
dtype: float64
```

```
df[["Gender","LoanAmount"]].groupby("Gender").mean()  
LoanAmount
```

Gender

Female 121.068182

Male 147.197403

```
df[["Gender","LoanAmount"]].groupby("Gender").median()  
LoanAmount
```

Gender

Female 115.5

Male 128.0

```
df[["Self_Employed","LoanAmount"]].groupby("Self_Employed").mean()  
LoanAmount
```

Self_Employed

No 141.836735

Yes 155.312500