Name : Tanishq Thuse

Div : SY CSAI - B

Roll no. : 60

PRN : 12310237

Subject : DV Assignment - 3

Date : 29/07/2024

Assignment 3

Descriptive Statistics - Measures of Central Tendency and variability perform the following operations on any open-source dataset (e.g., data.csv) 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(Here in this project, I have done by Profession). Create a list that contains a numeric value for each response to the categorical variable.

Dataset Link : https://www.kaggle.com/datasets/datascientistanna/customers-dataset

```
import pandas as pd

path = "/content/Customers.csv"

df = pd.read_csv(path)

# Display original DataFrame
print("Original DataFrame:")
df.head()
```

	CustomerID	Gender	Age	Annual Income ($)	Spending Score (1-100)	Profession	Work Experience	Family Size
0	1	Male	19	15000	39	Healthcare	1	4
1	2	Male	21	35000	81	Engineer	3	3
2	3	Female	20	86000	6	Engineer	1	1
3	4	Female	23	59000	77	Lawyer	0	2

```
df.tail()
```

	CustomerID	Gender	Age	Annual Income ($)	Spending Score (1-100)	Profession	Work Experience	Family Size
1995	1996	Female	71	184387	40	Artist	8	7
1996	1997	Female	91	73158	32	Doctor	7	7
1997	1998	Male	87	90961	14	Healthcare	9	2
1998	1999	Male	77	182109	4	Executive	7	2

```
df.shape
```

```
(2000, 8)
```

```
df.describe()
```

	CustomerID	Age	Annual Income ($)	Spending Score (1-100)	Work Experience	Family Size
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	1000.500000	48.960000	110731.821500	50.962500	4.102500	3.768500
std	577.494589	28.429747	45739.536688	27.934661	3.922204	1.970749
min	1.000000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	500.750000	25.000000	74572.000000	28.000000	1.000000	2.000000
50%	1000.500000	48.000000	110045.000000	50.000000	3.000000	4.000000
75%	1500.250000	73.000000	149092.750000	75.000000	7.000000	5.000000

```
# see how many null values in data
df.isnull()
```

	CustomerID	Gender	Age	Annual Income ($)	Spending Score (1-100)	Profession	Work Experience	Family Size
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
1995	False	False	False	False	False	False	False	False
1996	False	False	False	False	False	False	False	False
1997	False	False	False	False	False	False	False	False
1998	False	False	False	False	False	False	False	False
1999	False	False	False	False	False	False	False	False

```
df.isnull().sum()
#Below we can see that there are 35 null
```

```
CustomerID                0
Gender                    0
Age                       0
Annual Income ($)         0
Spending Score (1-100)    0
Profession               35
Work Experience           0
Family Size               0
dtype: int64
```

```
#Method 1 of finding most common profession
df['Profession'].value_counts().idxmax()
```

```
#Method 2
mode_value = df['Profession'].mode()
print(mode_value)
```

```
0    Artist
Name: Profession, dtype: object
```

```
mode_value = df['Profession'].mode()[0]
```

```
df['Profession'].fillna(mode_value, inplace=True)
```

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

```
CustomerID              0
Gender                  0
Age                     0
Annual Income ($)       0
Spending Score (1-100)  0
Profession              0
Work Experience         0
Family Size             0
dtype: int64
```

Central Tendencies individually

Central Tendency for Annual Income & Spending Score

```
# Function to calculate central tendency for a given column
def central_tendency(column):
    mean = column.mean()
    median = column.median()
    mode = column.mode().iloc[0]
    return mean, median, mode

# List of numeric columns
numeric_columns = ['Annual Income ($)', 'Spending Score (1-100)']

# Calculate central tendency for each numeric column
for col in numeric_columns:
    mean, median, mode = central_tendency(df[col])
    print(f"\nCentral Tendency for {col}:")
    print(f"Mean: {mean}")
    print(f"Median: {median}")
    print(f"Mode: {mode}")
```

```
Central Tendency for Annual Income ($):
Mean: 110731.8215
Median: 110045.0
Mode: 9000

Central Tendency for Spending Score (1-100):
Mean: 50.9625
Median: 50.0
Mode: 49
```

Central Tendecies for Income & Spending Score (old Method)

```
# # Central Tendency for 'Annual Income ($)'
# central_tendency_income = df.groupby('Age')['Annual Income ($)'].agg(['mean', 'median'])
# central_tendency_income['mode'] = df.groupby('Age')['Annual Income ($)'].agg(lambda x: x.mode()[0])
# print("\nCentral Tendency for Annual Income ($) grouped by Age:")
# print(central_tendency_income)

# Central Tendency for 'Annual Income ($)'
central_tendency_income = df.groupby('Profession')['Annual Income ($)'].agg(['mean', 'median'])
central_tendency_income['mode'] = df.groupby('Profession')['Annual Income ($)'].agg(lambda x: x.mode()[0])
print("\nCentral Tendency for Annual Income ($) grouped by Profession:")
print(central_tendency_income)
```

```
Central Tendency for Annual Income ($) grouped by Profession:
                         mean    median   mode
Profession
Artist         109234.081917  106633.0  84000
Doctor         111573.217391  111871.0  35000
Engineer       111161.240223  112766.0  97000
Entertainment  110650.333333  109446.0   9000
Executive      113770.130719  112957.0   4000
Healthcare     112574.041298  111717.0  31000
Homemaker      108758.616667  100387.0  79000
Lawyer         110995.838028  113338.5  50000
Marketing      107994.211765  120899.0   5000
```

```
# Central Tendency for 'Spending Score (1-100)'
central_tendency_spending = df.groupby('Profession')['Spending Score (1-100)'].agg(['mean', 'median'])
central_tendency_spending['mode'] = df.groupby('Profession')['Spending Score (1-100)'].agg(lambda x: x.mode()[0])
print("\nCentral Tendency for Spending Score (1-100) grouped by Profession:")
print(central_tendency_spending)
```

```
Central Tendency for Spending Score (1-100) grouped by Profession:
                     mean  median  mode
Profession
Artist          52.231839    52.0    55
Doctor          51.900621    50.0    42
Engineer        48.966480    47.0    45
Entertainment   52.940171    53.0    49
Executive       49.901961    49.0    88
Healthcare      50.516224    51.0    14
Homemaker       46.383333    45.5    32
Lawyer          48.859155    49.0    46
Marketing       48.717647    46.0     3
```

```
# Variability for 'Annual Income ($)'
variability_income = df.groupby('Profession')['Annual Income ($)'].agg(['std', 'var'])
print("\nVariability for Annual Income ($) grouped by Profession:")
print(variability_income)
```

```
Variability for Annual Income ($) grouped by Profession:
                          std           var
Profession
Artist          45172.541695  2.040559e+09
Doctor          48261.233502  2.329147e+09
Engineer        46503.822115  2.162605e+09
Entertainment   45001.884572  2.025170e+09
Executive       45434.149328  2.064262e+09
Healthcare      45426.143104  2.063534e+09
Homemaker       40393.442633  1.631630e+09
Lawyer          47793.706749  2.284238e+09
Marketing       48772.573140  2.378764e+09
```

```
# Variability for 'Spending Score (1-100)'
variability_spending = df.groupby('Profession')['Spending Score (1-100)'].agg(['std', 'var'])
print("\nVariability for Spending Score (1-100) grouped by Profession:")
print(variability_spending)
```

```
Variability for Spending Score (1-100) grouped by Profession:
                      std         var
Profession
Artist          28.271386  799.271245
Doctor          27.437703  752.827562
Engineer        27.733868  769.167409
Entertainment   26.455985  699.919152
Executive       28.102202  789.733746
Healthcare      28.344492  803.410239
Homemaker       28.394373  806.240395
Lawyer          27.718594  768.320448
Marketing       28.924208  836.609804
```

For all together

```
# Calculate measures of central tendency and variability for each group
central_tendency_variability = df.groupby('Profession')['Annual Income ($)'].agg(['mean', 'median', 'std', 'var'])
print("\nCentral Tendency and Variability for Income grouped by Profession:")
print(central_tendency_variability)
```

```
Central Tendency and Variability for Income grouped by Profession:
                           mean    median           std           var
Profession
Artist          109234.081917  106633.0  45172.541695  2.040559e+09
Doctor          111573.217391  111871.0  48261.233502  2.329147e+09
Engineer        111161.240223  112766.0  46503.822115  2.162605e+09
Entertainment   110650.333333  109446.0  45001.884572  2.025170e+09
Executive       113770.130719  112957.0  45434.149328  2.064262e+09
Healthcare      112574.041298  111717.0  45426.143104  2.063534e+09
Homemaker       108758.616667  100387.0  40393.442633  1.631630e+09
Lawyer          110995.838028  113338.5  47793.706749  2.284238e+09
Marketing       107994.211765  120899.0  48772.573140  2.378764e+09
```

```
# Create a list of numeric values for each response to the categorical variable
income_by_agegroup = df.groupby('Profession')['Annual Income ($)'].apply(list).to_dict()
print("\nList of Income values for each Profession:")
print(income_by_agegroup)
```

```
List of Income values for each Profession:
{'Artist': [58000, 98000, 62000, 42000, 71000, 52000, 78000, 18000, 20000, 39000, 9000, 69000, 25000, 22000, 33000, 52000, 88000, 97000,
```

Start coding or generate with AI.

Insights

Key Insights:

Business Value Insights:

Highest Income: Executives have the highest average annual income ($113,770).

Lowest Income: Homemakers have the lowest average income, indicating price sensitivity.

Income Variability: Marketing professionals show the highest income variability, suggesting diverse income levels.

Spending Insights: Highest Spending: Entertainment professionals have the highest average spending score (52.94).

Lowest Spending: Homemakers have the lowest average spending score (46.38). Spending Variability: Marketing professionals exhibit the highest variability in spending scores.