practical-exam-05

May 23, 2023

[1]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

1 Problem Statement 5

Perform the following operations on any open source dataset —

- 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. Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris-versicolor' of iris.csv dataset.

[3]:
```
import pandas as pd
```

[75]:
```
data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/exam_datasets/7.
↪social_network_ads.csv')
```

```
<IPython.core.display.HTML object>
```

[76]:
```
data.describe()
```

```
<IPython.core.display.HTML object>
```

[76]:

	User ID	Age	EstimatedSalary	Purchased
count	4.000000e+02	400.000000	400.000000	400.000000
mean	1.569154e+07	37.655000	69742.500000	0.357500
std	7.165832e+04	10.482877	34096.960282	0.479864
min	1.556669e+07	18.000000	15000.000000	0.000000
25%	1.562676e+07	29.750000	43000.000000	0.000000
50%	1.569434e+07	37.000000	70000.000000	0.000000
75%	1.575036e+07	46.000000	88000.000000	1.000000

```
max    1.581524e+07   60.000000    150000.000000    1.000000
```

[77]:
```
data.info()
```

```
<IPython.core.display.HTML object>
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   User ID          400 non-null    int64
 1   Gender           400 non-null    object
 2   Age              400 non-null    int64
 3   EstimatedSalary  400 non-null    int64
 4   Purchased        400 non-null    int64
dtypes: int64(4), object(1)
memory usage: 15.8+ KB
```

[79]:
```
data = data.replace('Male', 0)
data = data.replace('Female', 1)
data.head()
```

```
<IPython.core.display.HTML object>
```

[79]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	0	19	19000	0
1	15810944	0	35	20000	0
2	15668575	1	26	43000	0
3	15603246	1	27	57000	0
4	15804002	0	19	76000	0

[84]:
```
data['EstimatedSalary'].min()
```

```
<IPython.core.display.HTML object>
```

[84]:
```
15000
```

[85]:
```
data['EstimatedSalary'].max()
```

```
<IPython.core.display.HTML object>
```

[85]:
```
150000
```

[86]:
```
data['EstimatedSalary'].std()
```

```
<IPython.core.display.HTML object>
```

[86]:
```
34096.960282424785
```

[87]:
```
data['EstimatedSalary'].mean()
```

```
<IPython.core.display.HTML object>
```

[87]:
```
69742.5
```

[88]:
```
data['EstimatedSalary'].median()
```

```
<IPython.core.display.HTML object>
```

[88]:
```
70000.0
```

[89]:
```
data['EstimatedSalary'].nunique()
```

```
<IPython.core.display.HTML object>
```

[89]:
```
117
```

[90]:
```
data.groupby(['EstimatedSalary', 'Age']).count()
```

```
<IPython.core.display.HTML object>
```

[90]:

EstimatedSalary	Age	User ID	Gender	Purchased
15000	26	2	2	2
	30	1	1	1
	31	1	1	1
16000	21	1	1	1
	26	1	1	1
…		…	…	…
148000	29	1	1	1
149000	33	1	1	1
	42	1	1	1
150000	32	1	1	1
	52	1	1	1

[365 rows x 3 columns]

2 2nd Dataset

[72]:
```
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/exam_datasets/
↪5-8-13-14.iris.csv')
```

```
<IPython.core.display.HTML object>
```

[73]:
```
df.describe()
```

```
<IPython.core.display.HTML object>
```

[73]:
```
       sepal.length  sepal.width  petal.length  petal.width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.057333      3.758000     1.199333
std        0.828066     0.435866      1.765298     0.762238
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000
```

[74]:
```
df.info()
```

```
<IPython.core.display.HTML object>
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal.length  150 non-null    float64
 1   sepal.width   150 non-null    float64
 2   petal.length  150 non-null    float64
 3   petal.width   150 non-null    float64
 4   variety       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

[93]:
```
grouped = df.groupby('variety')
stats = grouped.describe()
print(stats)
```

```
<IPython.core.display.HTML object>
```

```
           sepal.length                                               \
                  count   mean       std  min    25%  50%  75%  max
variety
Setosa             50.0  5.006  0.352490  4.3  4.800  5.0  5.2  5.8
Versicolor         50.0  5.936  0.516171  4.9  5.600  5.9  6.3  7.0
Virginica          50.0  6.588  0.635880  4.9  6.225  6.5  6.9  7.9

           sepal.width         … petal.length      petal.width         \
                 count   mean  …          75%  max       count   mean
variety                        …
Setosa            50.0  3.428  …        1.575  1.9        50.0  0.246
Versicolor        50.0  2.770  …        4.600  5.1        50.0  1.326
Virginica         50.0  2.974  …        5.875  6.9        50.0  2.026


                 std  min  25%  50%  75%  max
```

```
variety
Setosa      0.105386  0.1  0.2  0.2  0.3  0.6
Versicolor  0.197753  1.0  1.2  1.3  1.5  1.8
Virginica   0.274650  1.4  1.8  2.0  2.3  2.5

[3 rows x 32 columns]
```

[103]:
```
stats = grouped.describe()
formatted_stats = stats.style.format("{:.2f}")
formatted_stats
```

```
<IPython.core.display.HTML object>
```

[103]:
```
<pandas.io.formats.style.Styler at 0x7fd3655222c0>
```