Hand written Notes

-On-

Exploratory Data Analysis (Python)

Tools used -
1. NumPy
2. Pandas
3. Matplotlib
4. Seaborn
5. Urge to learn

Divyanshu Vyas

Exploratory Data Analysis with Python.

List of Tasks to be covered:

1. Initial Data Exploration
2. Univariate Analysis
3. Bivariate Analysis
4. Dealing with duplicate rows and missing values.
5. Correlation Analysis.

* → libraries.

» import numpy as np , pandas as pd → Data Analysis

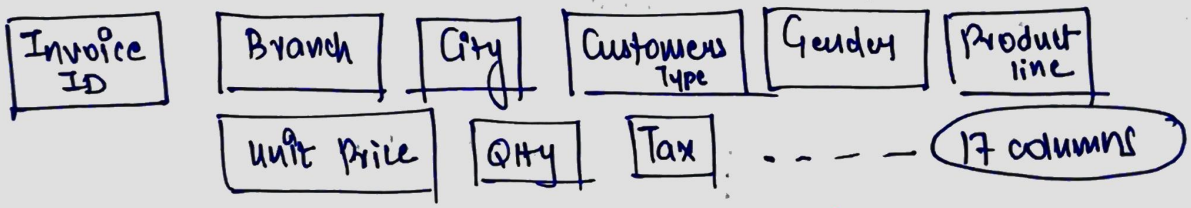
» import matplotlib.pyplot as plt ; » import seaborn as sns.

» import calmap :- Calender & heatmap library

» from pandas-profiling import Profile Report

↳ makes EDA structured.

* Dataset used → supermarket-sales data.



* To view the datatypes → df.dtypes ⟷

col_name	datatype
⋮	⋮

series.

* df['date'] = pd.to_datetime(df['date'])

* To set date('Date') column as the index →

→ df.set_index('Date', inplace=True)

Summary statistics

`df.describe()` →

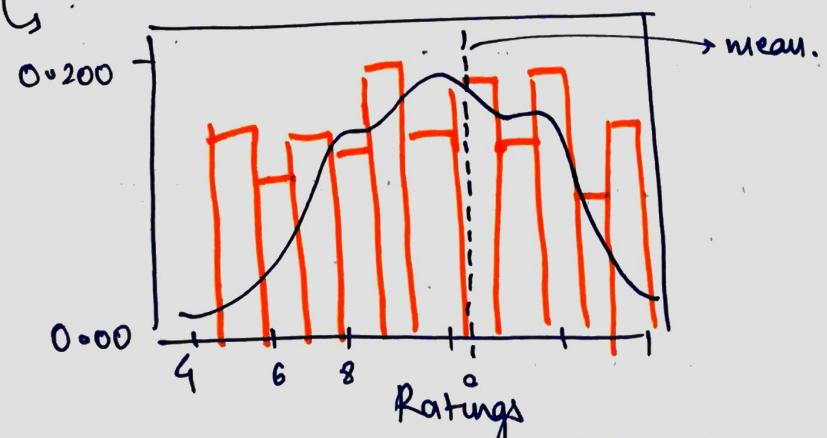
	← col-names →			
count	✓	✓	✓	✓
mean	✓	✓	✓	✓
std	✓	✓	✓	✓
min	✓	✓	✓	✓
25%	✓	✓	✓	✓
50%	✓	✓	—	—
75%	—	—	—	—
max	—	—	—	—

Univariate Analysis

"Ask questions & Answer them using your numpy / pandas / matplot / seaborn tools."

Q1 → what does customer ratings distribⁿ look like? Is it skewed?

Ans → `sns.distplot(df['Rating'])`



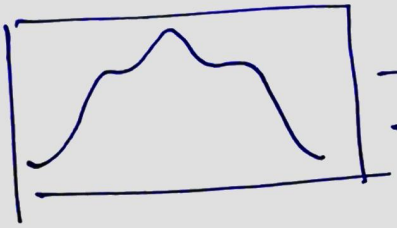
`plt.axvline(x= np.mean(df['Rating']), ls='--')`

Drawing v-lines for 25th & 75th percentile.

for 25th percentile →

```
plt.axvline(x= np.percentile(df['Ratings'], 25), lw='--')
```

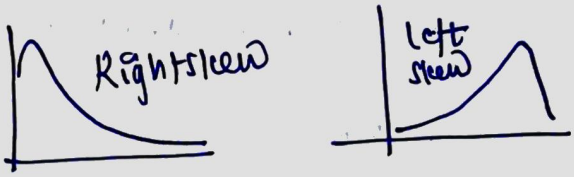
75 for 75th %ile.



→ looks like a Uniform Distribⁿ.
→ NO skew in left or Right.

Q2:- How to see 'visually' the distribution of all the Numerical variables?

Ans → `df.hist()` or `df.hist(figsize=(10,10))`

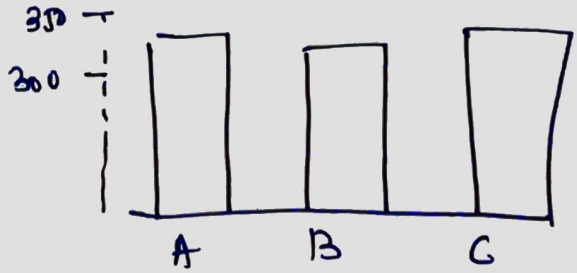
Note → 

(Trick. whenever you can write legend, that's the skew)

Note → Highly correlated stuff has almost identical correln.

Q3:- The column 'Branch' has 3 categories in it → 'A', 'B', 'C'. count the count of each.

Ans → `sns.countplot(df['Branch'])`



Not much difference.

Note → To get the exact count that the countplot displays

```
df['Brand'].value_counts()
```

```
A    342
B    333
C    328
Name: Brand, dtype: int64
```

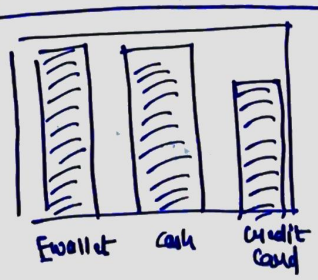
→ We can use countplot and value_counts for any categorical variable.

Example – Perform Count Analysis for 'Payment' column (this column has payment method categories)

∴

```
sns.countplot(df['Payments'])
df['payments'].value_counts
```

↳



Hence credit card is the least popular kind of payment method.

BIVARIATE ANALYSIS

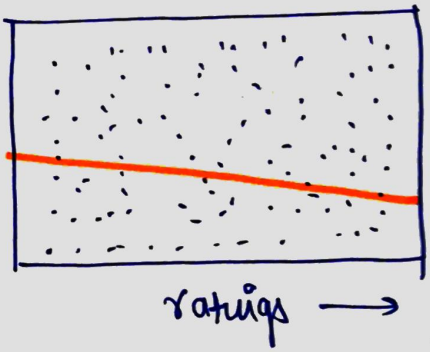
→ To check if there's a relationship b/w various columns (features)

Q:- Is there a relnship b/w Gross income & customer Ratings?

```
sns.scatterplot(df['Ratings'], df['gross income'])
```

↳ grossincome

```
sns.regplot(df['Ratings'], df['gross income'])
```



To see the trendline (orange)

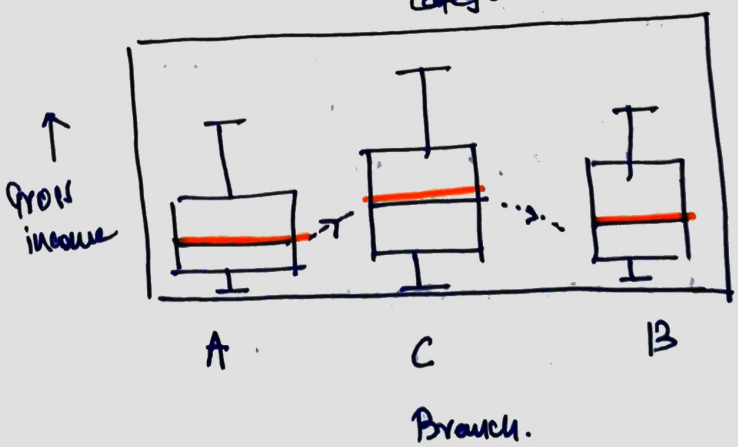
Flat-Trendline ⟷ No relnship.

Q:- To see the Relnship b/w Y (Gross Income) with a categorical variable X('Branch).

⇒ Boxplot.

→ `sns.boxplot(x = df['Branch'], y = df['Gross Income'])`

Categorical.



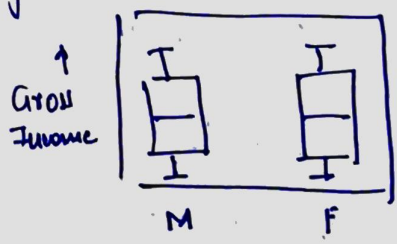
C > B > A

median Gross income

Similar Example → To see how gross income varies with gender.

sns.boxplot (x=df['Gender'], y=df['Gross Income'])

NOK → for a vertical box plot, make sure you put the reference categorical variable on the x-axis.



Q:- Time Trend Analysis → Is there any trend that the gross income (from the people) (Ratika is the earning of supermarket) follows?

NOK → ① First convert date column into DateTime Object.

② Cannot move directly. First aggregate the income for every day since each date appears multiple times (As expected).

↓

GROUP-BY

Or df['Date'] if it wasn't index

→ df.groupby(df.index).mean()

↳ or .sum()

↳ make it a new dataframe & perform the Analysis.

DEALING with Duplicate Rows & Missing Values.

Q1:- To find number of duplicate Rows →.

↳ `df.duplicated().sum()`

`df.duplicated()` → Returns T/F (Boolean)

`.sum()` → sums the No. of True's up.

Q2:- How to view the duplicated Rows?

↳ `df[df.duplicated()]`

Q3:- How to drop duplicate Rows - Permanently?

Ans - `df.drop_duplicates(inplace=True)`

↳ After this now if you run the `duplicated().sum()` ↳ Ans should be 0.

#Missing Values

Q1:- How to find out the No. of missing values in each column.

Soln - `df.isna().sum()` →

Series (output)

col-1	5
col-2	0
col-3	0
col-4	3
⋮	⋮

Q2:- How to visualize the No. of Missing values?

↳ `sns.heatmap(df.isnull())`

There are various ways to deal with Missing data.

2 Basic ways → (1) Fill with 0
(2) Fill with means.

Q3:- How to Fill the missing values with mean?

↳ `df.fillna(df.mean(), inplace=True)`

↳ But this only fills up the Numeric columns.

↳ Categorical Missing values still remain empty.

Q4:- How to Fill Categorical Missing values?
(Missing/NaNs in categorical columns)

↳ Fill in with Mode.

NOTE → To get the mode for each column, we do → `df.mode().iloc[0]`

↳ only first row gives mode.

∴ `df.fillna(df.mode().iloc[0], inplace=True)`

↳ Finally No missing values.

NOTE: Whatever we've done, the EDA till now can be nicely done in one command by the Pandas Profiler. (Recommended for small datasets).

```
→ data = pd.read_csv('xyz.csv')
→ profile = ProfileReport(data)
→ profile
```

↳ Detailed EDA report

CORRELATION ANALYSIS.

Method1 → `np.corrcoef(df['A'], df['B'])`

Method2 → `df.corr()` → Correl° matrix.

↳ `np.round(df.corr(), 2)`

↳ rounded to 2 decimal places.

Method3 → `sns.heatmap(df.corr())`

↳ `sns.heatmap(df.corr(), annot=True)`