**GOVERNMENT POLYTECHNIC NAGAMANGALA**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**“V”th Semester Diploma**

**Artificial Intelligence and Machine Learning** (20CS51)

**Assignment:03**

**NAME: YASHASWINI S**

**ROLL NO:158CS22057**

**AIML (20CS51)**

**ASSIGNMENT – WEEK 02**

1. Download any two datasets from the internet and perform the following operations.

a) Analyze the univariate dataset Ex- Mean, Mode, Median, Range, Std, and Variance and perform Univariate tests for the dataset.

b) Analyze the multivariate of the dataset Ex- co-variance, co-relation.

c) Visualize the univariate and multivariate with various plots.

d) Push the code to your GitHub Repository.

e)Perform any probability calculation.

1. Download any two datasets from the internet and perform the following operations.

a) Analyze the univariate dataset Ex- Mean, Mode, Median, Range, Std, and Variance and perform Univariate tests for the dataset.

MEAN:

import pandas as pd

path = ("//content/annual-enterprise-survey-2023-financial-year-provisional (3).csv")

df = pd.read\_csv(path)

df.mean(numeric\_only=True)

OUTPUT:

Year 2018.0

dtype: float64

MEDIAN:

df.median(numeric\_only=True)

**OUTPUT:**

Year 2018.0

dtype: float64

MODE:

df.mode(numeric\_only=True)

OUTPUT:

| **Year** |
| --- |
| **0** | 2013 |
| **1** | 2014 |
| **2** | 2015 |
| **3** | 2016 |
| **4** | 2017 |
| **5** | 2018 |
| **6** | 2019 |
| **7** | 2020 |
| **8** | 2021 |
| **9** | 2022 |
| **10** | 2023 |

RANGE

df.max(numeric\_only=True) - df.min(numeric\_only=True)

OUTPUT:

Year 10

dtype: int64

VARIANCE

df.var(numeric\_only=True)

**OUTPUT:**

Year 10.000196

dtype: float64

addCode

addText

STANDARD DEVIATION

df.std(numeric\_only=True)

**OUTPUT:**

Year 3.162309

dtype: float64

T-TEST:

import pandas as pd

import scipy.stats as stats

df=pd.read\_csv('/content/annual-enterprise-survey-2023-financial-year-provisional (3).csv')

Year\_df=df['Year'].values

t\_stat, p\_val=stats.ttest\_1samp(Year\_df, popmean=0)

print(f"One-sample t-test: t\_stat={t\_stat}, p\_val={p\_val}")

**OUTPUT:**

One-sample t-test: t\_stat=144091.41779301083, p\_val=0.0

CHI – SQUARE TEST:

import pandas as pd

from scipy.stats import chi2\_contingency

df=pd.read\_csv('/content/annual-enterprise-survey-2023-financial-year-provisional (3).csv')

contingency\_table=pd.crosstab(df['Variable\_name'],df['Value'])

chi2, p, dof, ex = chi2\_contingency(contingency\_table)

print(f"Chi-Square Test of Independent: chi2={chi2}, p={p}, dog={dof}, expected={ex}")

**OUTPUT:**

Chi-Square Test of Independent: chi2=795016.4394285574, p=0.0, dog=546880, expected=[[2.07119741e-01 2.58899676e-02 2.58899676e-02 ... 5.17799353e-02

5.91585761e+01 4.66019417e-01]

[2.39913700e-01 2.99892125e-02 2.99892125e-02 ... 5.99784250e-02

6.85253506e+01 5.39805825e-01]

[2.31283711e-01 2.89104639e-02 2.89104639e-02 ... 5.78209277e-02

6.60604099e+01 5.20388350e-01]

...

[2.39913700e-01 2.99892125e-02 2.99892125e-02 ... 5.99784250e-02

6.85253506e+01 5.39805825e-01]

[2.39913700e-01 2.99892125e-02 2.99892125e-02 ... 5.99784250e-02

6.85253506e+01 5.39805825e-01]

[2.19201726e-01 2.74002157e-02 2.74002157e-02 ... 5.48004315e-02

6.26094930e+01 4.93203883e-01]]

ANOVA:

import numpy as np

from scipy import stats

import pandas as pd

df = pd.read\_csv('/content/annual-enterprise-survey-2023-financial-year-provisional (3).csv')

df['Value'] = pd.to\_numeric(df['Value'], errors='coerce')

groups = df.groupby('Variable\_name')['Value'].apply(list)

f\_value, p\_value=stats.f\_oneway(\*groups)

print(f"F-value: {f\_value}")

print(f"P-value: {p\_value}")

**OUTPUT:**

F-value: nan

P-value: nan

b) Analyze the multivariate of the dataset Ex- co-variance, co-relation.

CO-VARIANCE

df.cov(numerical\_only=True)

**OUTPUT:**

|  | **Year** |
| --- | --- |
| **Year** | 10.000196 |

COR-RELATION

df.corr(numeric\_only=True)

**OUTPUT:**

|  | **Year** |
| --- | --- |
| **Year** | 1.0 |

import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

df=pd.read\_csv('//content/annual-enterprise-survey-2023-financial-year-provisional.csv')

plt.figure(figsize=(16, 4))

plt.subplot(1, 4, 1)

plt.scatter(df['Year'], df['Year'], color='green')

plt.xlabel('Year')

plt.ylabel('Year')

plt.title('Scatter Plot')

plt.subplot(1, 4, 2)

sns.pairplot(df, vars=['Units', 'Value', 'Year'])

df['Value'] = pd.to\_numeric(df['Value'], errors='coerce')

plt.subplot(1, 4, 3)

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

plt.title('Heatmap')

numeric\_df = df.select\_dtypes(include=['number'])

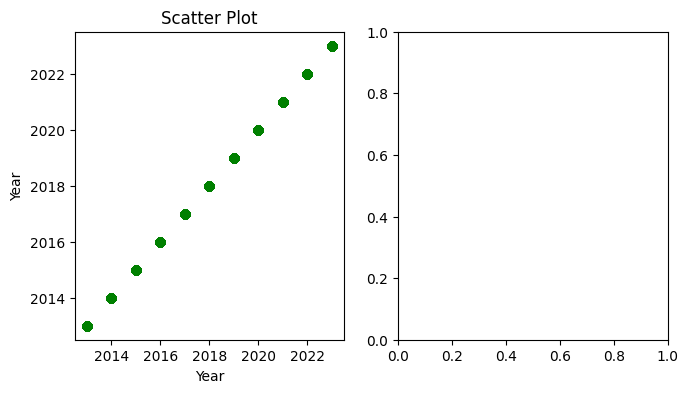
plt.subplot(1, 4, 4)

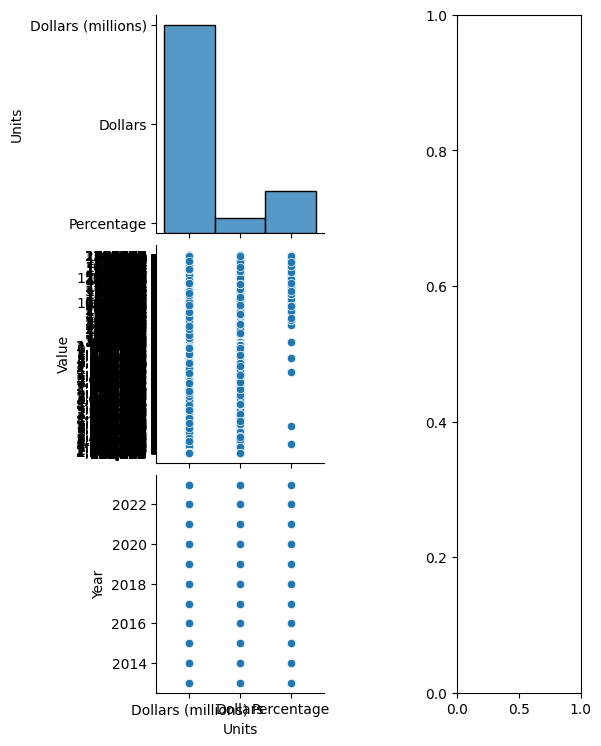
sns.jointplot(x='Value', y='Year', data=df, kind='hex', color='purple')

plt.tight\_layout()

plt.show()

**OUTPUT:**





UNIVARIATE PLOTS:

import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

data = pd.read\_csv("/content/annual-enterprise-survey-2023-financial-year-provisional (3).csv")

plt.figure(figsize=(16,14))

plt.subplot(1,4,1)

plt.hist(data['Year'],bins=10, color='blue')

plt.title('Histogram of year')

plt.xlabel('year')

plt.ylabel('Frequency')

plt.show

plt.subplot(1,4,2)

sns.boxplot(data=data, x='Year', color='salmon')

plt.xlabel('Year')

plt.title('Box plot')

plt.subplot(1,4,3)

sns.violinplot(data=data, x='Year', color ='green')

plt.xlabel('Year')

plt.title('Violinplot')

plt.subplot(1,4,4)

sns.displot(data['Year'], kde=True, color='purple')

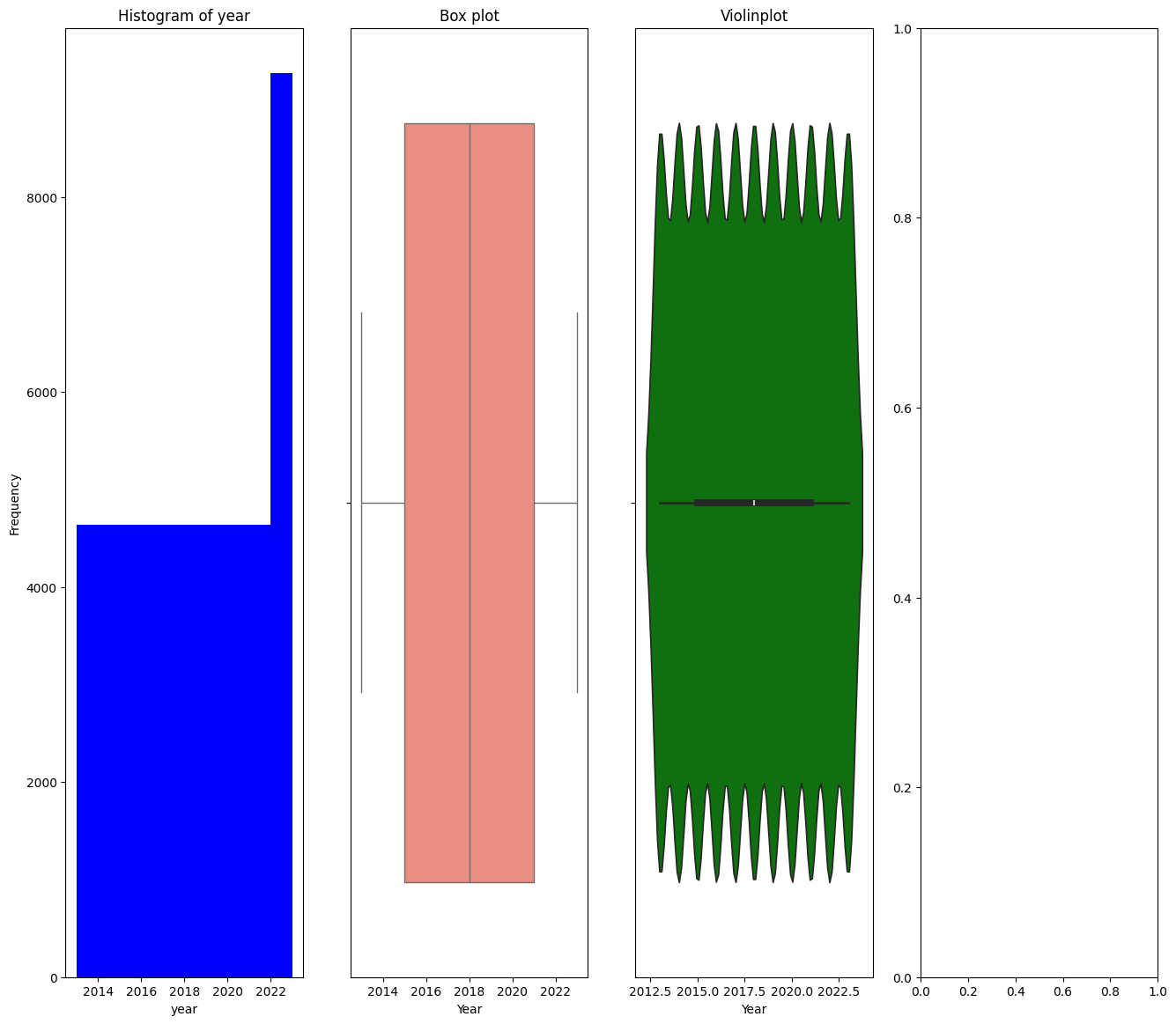
plt.xlabel('Year')

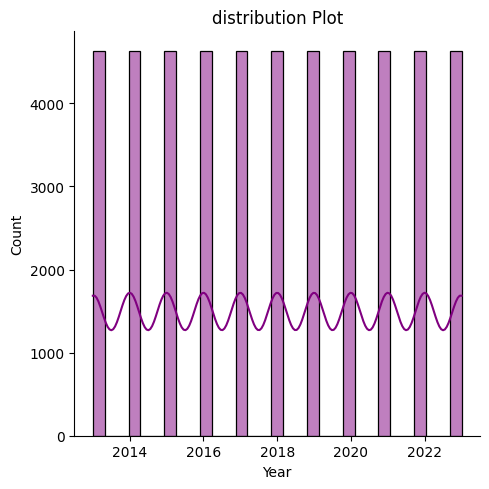
plt.title('distribution Plot')

plt.tight\_layout()

plt.show()

**OUTPUT:**

****



**DATASET – 2**

MEAN:

import pandas as pd

path = ('/content/tatauranga-umanga-maori-statistics-on-maori-businesses-march-2024-quarter.csv')

df = pd.read\_csv(path)

df.mean(numeric\_only=True)

**OUTPUT:**

Table\_Number 4.625000

Quarter 202021.137931

dtype: float64

MODE:

df.mode(numeric\_only=True)

**OUTPUT:**

|  | **Table\_Number** | **Quarter** |
| --- | --- | --- |
| **0** | 7.0 | 201703 |
| **1** | NaN | 201706 |
| **2** | NaN | 201709 |
| **3** | NaN | 201712 |
| **4** | NaN | 201803 |
| **5** | NaN | 201806 |
| **6** | NaN | 201809 |
| **7** | NaN | 201812 |
| **8** | NaN | 201903 |
| **9** | NaN | 201906 |
| **10** | NaN | 201909 |
| **11** | NaN | 201912 |
| **12** | NaN | 202003 |
| **13** | NaN | 202006 |
| **14** | NaN | 202009 |
| **15** | NaN | 202012 |
| **16** | NaN | 202103 |
| **17** | NaN | 202106 |
| **18** | NaN | 202109 |
| **19** | NaN | 202112 |
| **20** | NaN | 202203 |
| **21** | NaN | 202206 |
| **22** | NaN | 202209 |
| **23** | NaN | 202212 |
| **24** | NaN | 202303 |
| **25** | NaN | 202306 |
| **26** | NaN | 202309 |
| **27** | NaN | 202312 |
| **28** | NaN | 202403 |

MEDIAN:

df.median(numeric\_only=True)

**OUTPUT:**

Table\_Number 5.0

Quarter 202009.0

dtype: float64

RANGE:

df.max(numeric\_only=True) - df.min(numeric\_only=True)

**OUTPUT:**

Table\_Number 6

Quarter 700

dtype: int64

VARIANCE:

df.var(numeric\_only=True)

**OUTPUT:**

Table\_Number 3.695363

Quarter 43870.876720

dtype: float64

STANDARD DEVIATION:

df.std(numeric\_only=True)

**OUTPUT:**

Table\_Number 1.922333

Quarter 209.453758

dtype: float64

CO-VARIANCE:

df.cov(numeric\_only=True)

**OUTPUT:**

|  | **Table\_Number** | **Quarter** |
| --- | --- | --- |
| **Table\_Number** | 3.695363 | 0.00000 |
| **Quarter** | 0.000000 | 43870.87672 |

COR-RELATION:

df.corr(numeric\_only=True)

**OUTPUT:**

Table\_NumberQuarterTable\_Number1.000000e+001.164446e-12Quarter1.164446e-121.000000e+00

CHI-SQUARE TEST:

import pandas as pd

from scipy.stats import chi2\_contingency

df=pd.read\_csv('/content/tatauranga-umanga-maori-statistics-on-maori-businesses-march-2024-quarter.csv')

contingency\_table=pd.crosstab(df['Value'],df['Unit'])

chi2, p, dof, ex = chi2\_contingency(contingency\_table)

print(f"Chi-Square Test of Independent: chi2={chi2}, p={p}, dog={dof}, expected={ex}")

**OUTPUT:**

Chi-Square Test of Independent: chi2=1377.3473684210526, p=7.645825814782184e-49, dog=685, expected=[[7.125 1.875 ]

[0.79166667 0.20833333]

[0.79166667 0.20833333]

...

[1.58333333 0.41666667]

[0.79166667 0.20833333]

[6.33333333 1.66666667]]

ANOVA:

import numpy as np

from scipy import stats

import pandas as pd

df = pd.read\_csv('/content/tatauranga-umanga-maori-statistics-on-maori-businesses-march-2024-quarter.csv')

df['Value'] = pd.to\_numeric(df['Value'], errors='coerce')

groups = df.groupby('Unit')['Value'].apply(list)

f\_value, p\_value=stats.f\_oneway(\*groups)

print(f"F-value: {f\_value}")

print(f"P-value: {p\_value}")

**OUTPUT:**

F-value: nan

P-value: nan

mport pandas as pd

from scipy.stats import chi2\_contingency

df=pd.read\_csv('/content/tatauranga-umanga-maori-statistics-on-maori-businesses-march-2024-quarter.csv')

contingency\_table=pd.crosstab(df['Unit'],df['Value'])

chi2, p, dof, ex =i chi2\_contingency(contingency\_table)

print(f"Chi-Square Test of Independent: chi2={chi2}, p={p}, dog={dof}, expected={ex}")

**OUTPUT:**

Chi-Square Test of Independent: chi2=1377.3473684210526, p=7.645825814782184e-49, dog=685, expected=[[7.125 0.79166667 0.79166667 ... 1.58333333 0.79166667 6.33333333]

[1.875 0.20833333 0.20833333 ... 0.41666667 0.20833333 1.66666667]]

UNIVARIATE:

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv('/content/tatauranga-umanga-maori-statistics-on-maori-businesses-march-2024-quarter.csv')

plt.figure(figsize=(16, 4))

plt.subplot(1,4,1)

plt.hist(data['Unit'], bins=10, color='Skyblue', edgecolor='black')

plt.xlabel('Unit')

plt.ylabel('Value')

plt.title('Histogram')

plt.show()

plt.subplot(1, 4, 2)

sns.boxplot(data=df, x='Value', color='pink')

plt.xlabel('Value')

plt.title('Box Plot')

plt.subplot(1, 4, 3)

sns.violinplot(data=df, x='Unit', color='orange')

plt.xlabel('Unit')

plt.title('Violin Plot ')

plt.subplot(1, 4, 4)

sns.displot(df['Unit'], kde=True, color='blue')

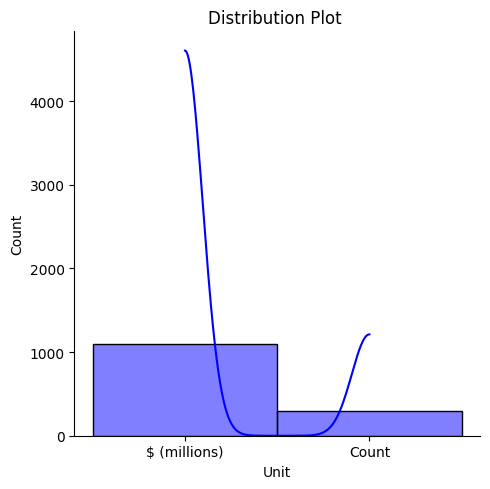
plt.xlabel('Unit')

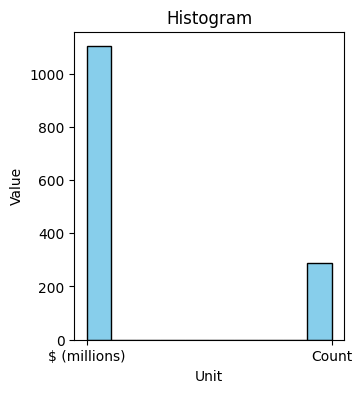
plt.title('Distribution Plot ')

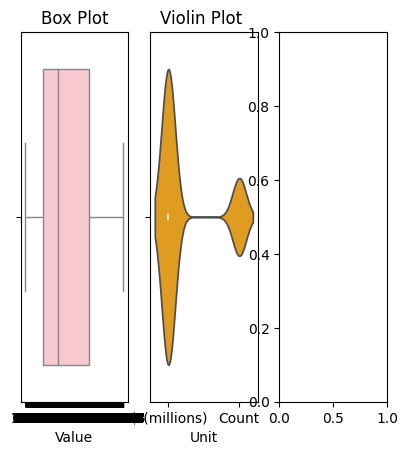
plt.tight\_layout()

plt.show()

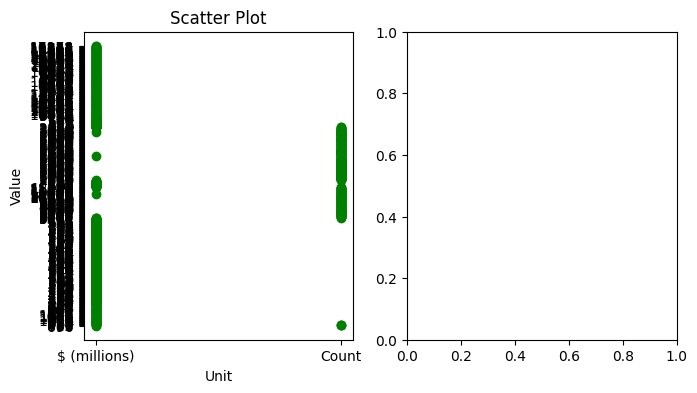
**OUTPUT:**

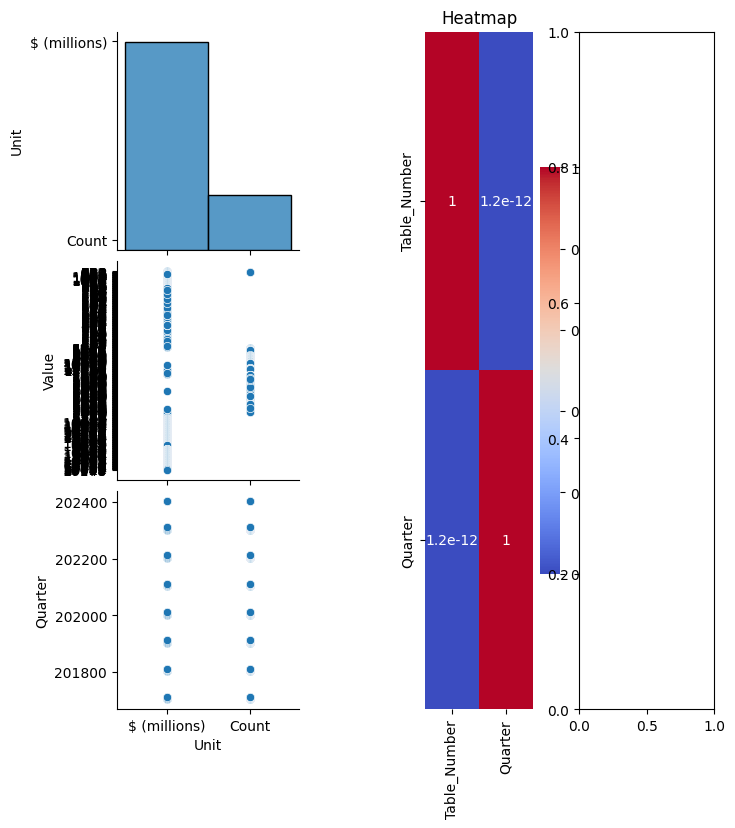






MULTIVARIATE PLOTS:





Probility calculation:

**DATASET-1**

import pandas as pd

df = pd.read\_csv('/content/annual-enterprise-survey-2023-financial-year-provisional (2).csv')

condition\_count = df['Variable\_name'].value\_counts()

probability = condition\_count / condition\_count.sum()

print(condition\_count)

print(probability)

print(f"The probability of getting values from {condition\_count.index[0]} to {condition\_count.index[-1]} are:")

for index, value in probability.items():

print(f"{index}: {value:.4f}")

**OUTPUT:**

Variable\_name

Total income 1529

Non-operating expenses 1529

Liabilities structure 1529

Return on total assets 1529

Return on equity 1529

Shareholders funds or owners equity 1529

Total assets 1529

Surplus before income tax 1529

Closing stocks 1529

Opening stocks 1529

Total equity and liabilities 1529

Interest and donations 1529

Indirect taxes 1529

Non-operating income 1529

Total expenditure 1529

Interest, dividends and donations 1529

Depreciation 1529

Salaries and wages paid 1529

Redundancy and severance 1529

Current assets 1474

Fixed tangible assets 1474

Other assets 1474

Current liabilities 1474

Other liabilities 1474

Quick ratio 1474

Current ratio 1474

Total income per employee count 1397

Surplus per employee count 1397

Disposals of fixed assets 1320

Additions to fixed assets 1320

Government funding, grants and subsidies 1221

Purchases and other operating expenses 957

Sales of goods and services 649

Sales of goods not further processed 572

Sales of other goods and services 572

Purchases of goods bought for resale 572

Margin on sales of goods for resale 572

Other Purchases and operating expenses 520

Sales, government funding, grants and subsidies 308

Salaries and wages to self employed commission agents 187

Other purchases and operating expenses 52

Name: count, dtype: int64

Variable\_name

Total income 0.029989

Non-operating expenses 0.029989

Liabilities structure 0.029989

Return on total assets 0.029989

Return on equity 0.029989

Shareholders funds or owners equity 0.029989

Total assets 0.029989

Surplus before income tax 0.029989

Closing stocks 0.029989

Opening stocks 0.029989

Total equity and liabilities 0.029989

Interest and donations 0.029989

Indirect taxes 0.029989

Non-operating income 0.029989

Total expenditure 0.029989

Interest, dividends and donations 0.029989

Depreciation 0.029989

Salaries and wages paid 0.029989

Redundancy and severance 0.029989

Current assets 0.028910

Fixed tangible assets 0.028910

Other assets 0.028910

Current liabilities 0.028910

Other liabilities 0.028910

Quick ratio 0.028910

Current ratio 0.028910

Total income per employee count 0.027400

Surplus per employee count 0.027400

Disposals of fixed assets 0.025890

Additions to fixed assets 0.025890

Government funding, grants and subsidies 0.023948

Purchases and other operating expenses 0.018770

Sales of goods and services 0.012729

Sales of goods not further processed 0.011219

Sales of other goods and services 0.011219

Purchases of goods bought for resale 0.011219

Margin on sales of goods for resale 0.011219

Other Purchases and operating expenses 0.010199

Sales, government funding, grants and subsidies 0.006041

Salaries and wages to self employed commission agents 0.003668

Other purchases and operating expenses 0.001020

Name: count, dtype: float64

The probability of getting values from Total income to Other purchases and operating expenses are:

Total income: 0.0300

Non-operating expenses: 0.0300

Liabilities structure: 0.0300

Return on total assets: 0.0300

Return on equity: 0.0300

Shareholders funds or owners equity: 0.0300

Total assets: 0.0300

Surplus before income tax: 0.0300

Closing stocks: 0.0300

Opening stocks: 0.0300

Total equity and liabilities: 0.0300

Interest and donations: 0.0300

Indirect taxes: 0.0300

Non-operating income: 0.0300

Total expenditure: 0.0300

Interest, dividends and donations: 0.0300

Depreciation: 0.0300

Salaries and wages paid: 0.0300

Redundancy and severance: 0.0300

Current assets: 0.0289

Fixed tangible assets: 0.0289

Other assets: 0.0289

Current liabilities: 0.0289

Other liabilities: 0.0289

Quick ratio: 0.0289

Current ratio: 0.0289

Total income per employee count: 0.0274

Surplus per employee count: 0.0274

Disposals of fixed assets: 0.0259

Additions to fixed assets: 0.0259

Government funding, grants and subsidies: 0.0239

Purchases and other operating expenses: 0.0188

Sales of goods and services: 0.0127

Sales of goods not further processed: 0.0112

Sales of other goods and services: 0.0112

Purchases of goods bought for resale: 0.0112

Margin on sales of goods for resale: 0.0112

Other Purchases and operating expenses: 0.0102

Sales, government funding, grants and subsidies: 0.0060

Salaries and wages to self employed commission agents: 0.0037

Other purchases and operating expenses: 0.0010

**DATASET-2**

**import pandas as pd**

**df = pd.read\_csv('/content/tatauranga-umanga-maori-statistics-on-maori-businesses-march-2024-quarter.csv')**

**condition\_count = df['Unit'].value\_counts()**

**probability = condition\_count / condition\_count.sum()**

**print(condition\_count)**

**print(probability)**

**print(f"The probability of getting values from {condition\_count.index[0]} to {condition\_count.index[-1]} are:")**

**for index, value in probability.items():**

**print(f"{index}: {value:.4f}")**

**OUTPUT:**

Unit

$ (millions) 1102

Count 290

Name: count, dtype: int64

Unit

$ (millions) 0.791667

Count 0.208333

Name: count, dtype: float64

The probability of getting values from $ (millions) to Count are:

$ (millions): 0.7917

Count: 0.2083