Experiment No.4

Title: Applying and interpreting different plots

Batch: B4 Roll No.: 16010420133 Experiment No. 4

Aim: Applying and interpreting different plots

Resources needed: Any programming language/ Rapid Miner, any data source (RDBMS/Excel/CSV)

Results:

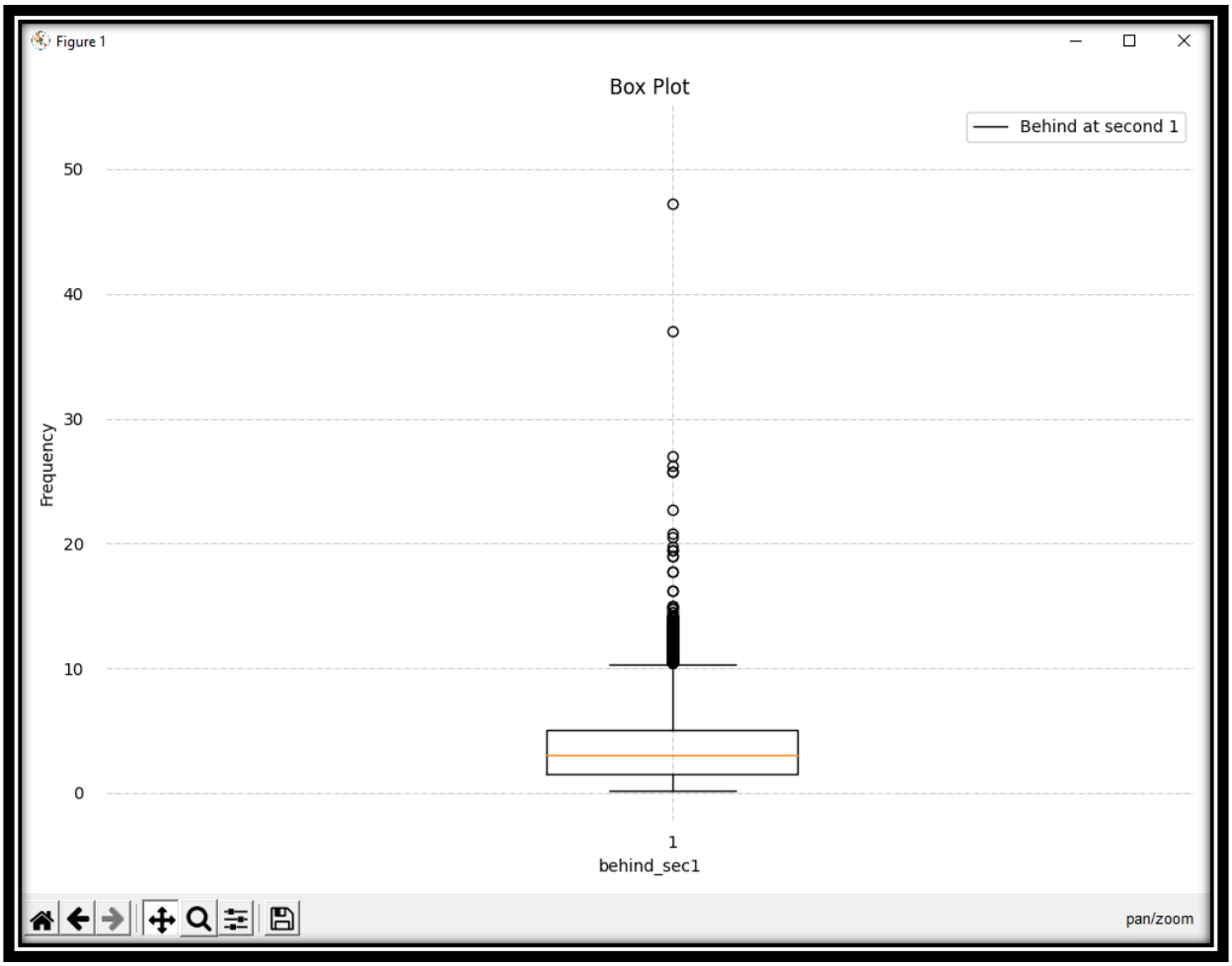
Code:

Box Plot:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
import scipy.stats as stats
from matplotlib import colors
from matplotlib.ticker import PercentFormatter

def Box_Plot():
    x_column = []
    csv_file = pd.read_csv('/Users/Soumen/Downloads/SEM_4
College_Stuff/HON/FDS/runs.csv')
    csv_file_x = csv_file[csv_file['behind_sec1'].notna()]
    for i in range(0,len(csv_file_x)):
        x_column.append(csv_file_x['behind_sec1'][i])

    fig, axs = plt.subplots(1, 1, figsize =(10, 7), tight_layout = True)
    for side in ['top', 'bottom', 'left', 'right']:
        axs.spines[side].set_visible(False)
    axs.xaxis.set_ticks_position('none')
    axs.yaxis.set_ticks_position('none')
    axs.xaxis.set_tick_params(pad = 5)
    axs.yaxis.set_tick_params(pad = 10)
    axs.grid(b = True, color ='grey', linestyle ='-.', linewidth = 0.5, alpha = 0.6)

    plt.boxplot(x_column)

    plt.xlabel('behind_sec1')
    plt.ylabel('Frequency')
    plt.legend(['Behind at second 1'])
    plt.title('Box Plot')
    plt.show()
```

Box_Plot()

Output:

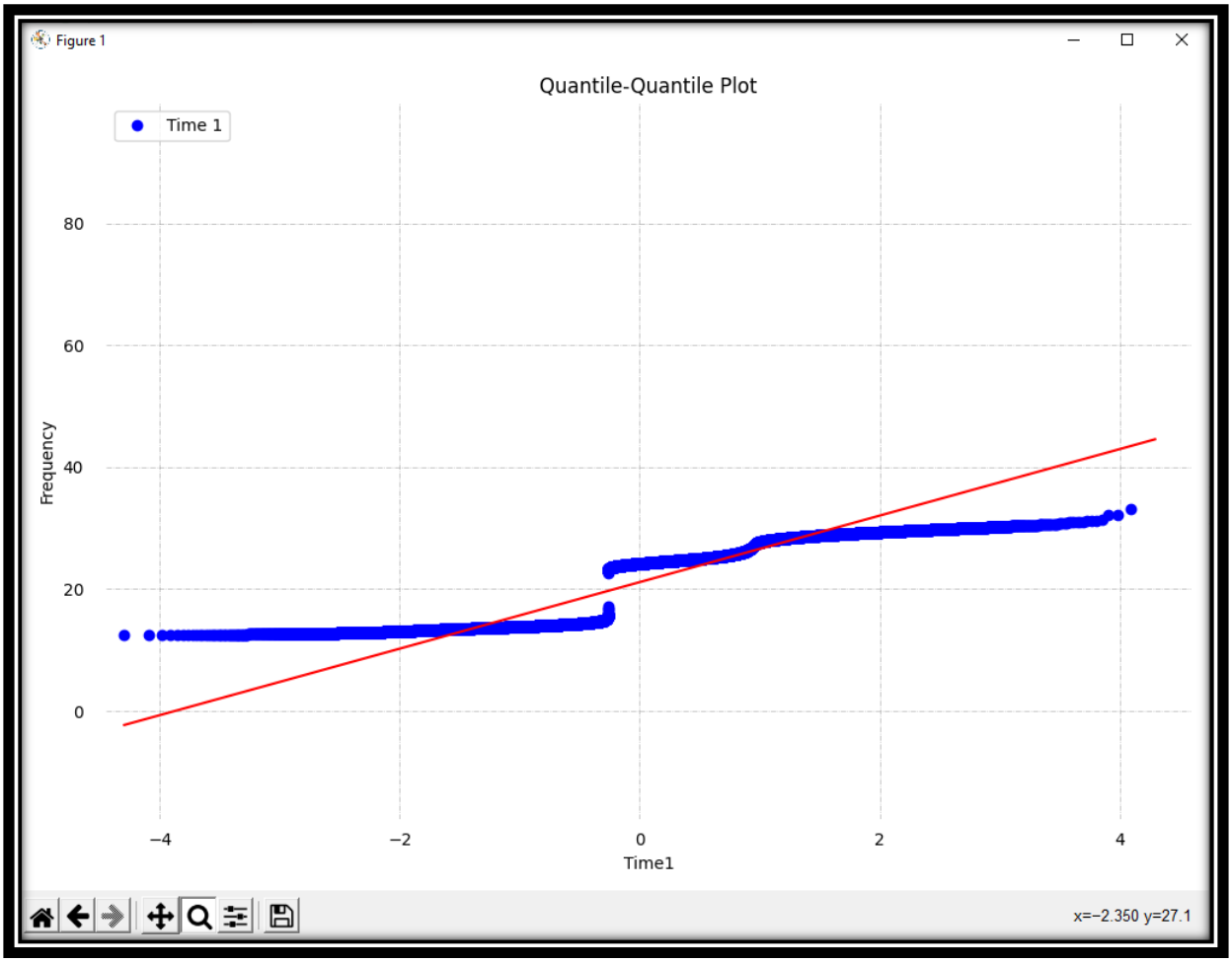


Quantile-Quantile Plot:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
import scipy.stats as stats
from matplotlib import colors
from matplotlib.ticker import PercentFormatter

def Quantile_Quantile_Plot():
    x_column = []
    csv_file = pd.read_csv('/Users/Soumen/Downloads/SEM_4
College_Stuff/HON/FDS/runs.csv')
    csv_file_x = csv_file[csv_file['time1'].notna()]
    for i in range(0,len(csv_file_x)):
```

```
        x_column.append(csv_file_x['time1'][i])

    fig, axs = plt.subplots(1, 1, figsize =(10, 7), tight_layout = True)
    for side in ['top', 'bottom', 'left', 'right']:
        axs.spines[side].set_visible(False)
    axs.xaxis.set_ticks_position('none')
    axs.yaxis.set_ticks_position('none')
    axs.xaxis.set_tick_params(pad = 5)
    axs.yaxis.set_tick_params(pad = 10)
    axs.grid(b = True, color ='grey', linestyle ='-.', linewidth = 0.5, alpha = 0.6)

    stats.probplot(x_column, dist="norm", plot = plt)
    plt.xlabel('Time1')
    plt.ylabel('Frequency')
    plt.legend(['Time 1'])
    plt.title('Quantile-Quantile Plot')
    plt.show()
Quantile_Quantile_Plot()
```

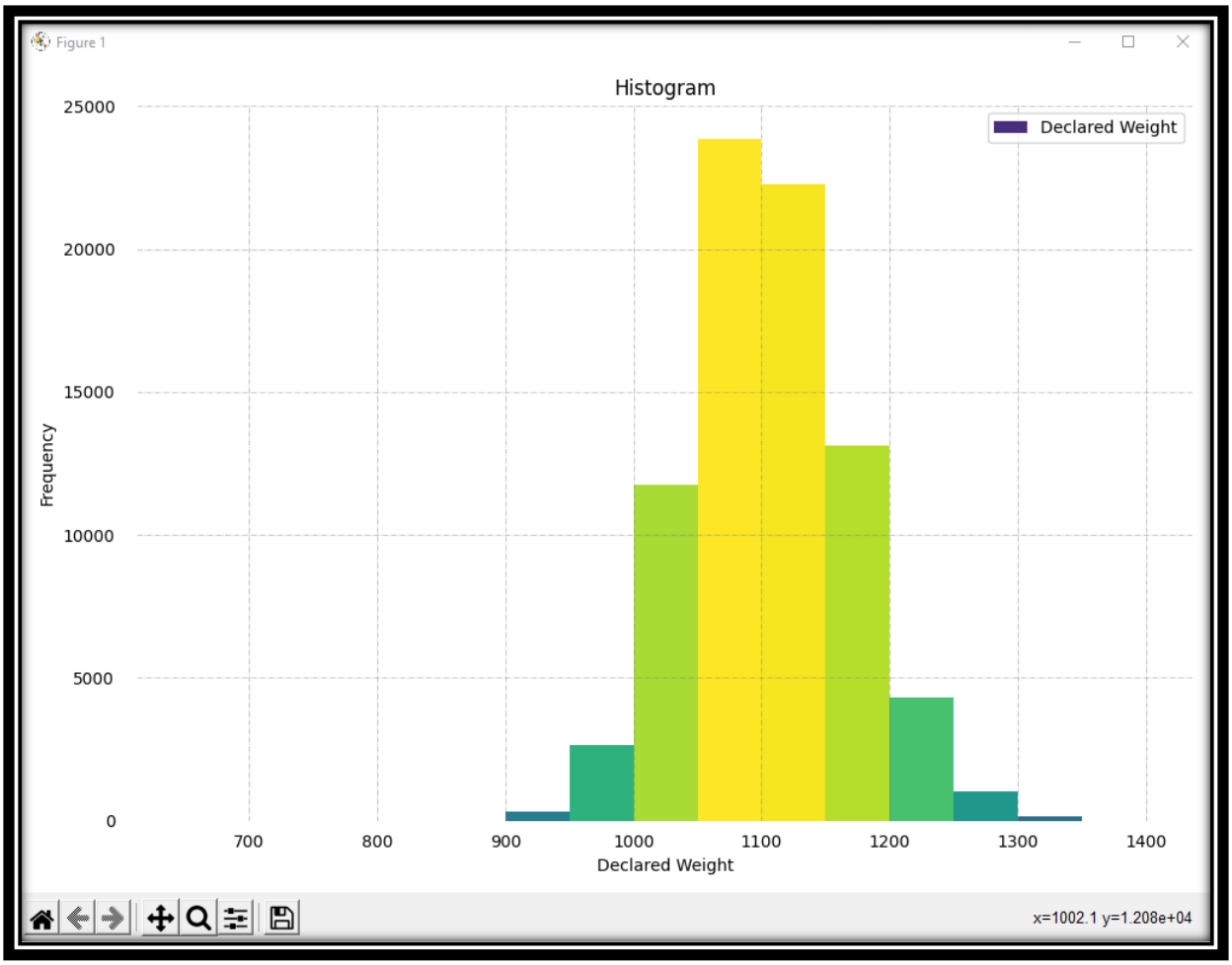
Output:



Box Plot:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
import scipy.stats as stats
from matplotlib import colors
from matplotlib.ticker import PercentFormatter

def Histogram():
    x_column = []
    csv_file = pd.read_csv('/Users/Soumen/Downloads/SEM_4
College_Stuff/HON/FDS/runs.csv')
    csv_file_x = csv_file[csv_file['declared_weight'].notna()]
    for i in range(0,len(csv_file_x)):
        x_column.append(csv_file_x['declared_weight'][i])

    bins = [650, 700, 750, 800, 850, 900, 950, 1000, 1050, 1100, 1150, 1200, 1250, 1300,
1350, 1399]

    fig, axs = plt.subplots(1, 1, figsize =(10, 7), tight_layout = True)
    for side in ['top', 'bottom', 'left', 'right']:
        axs.spines[side].set_visible(False)
    axs.xaxis.set_ticks_position('none')
    axs.yaxis.set_ticks_position('none')
    axs.xaxis.set_tick_params(pad = 5)
    axs.yaxis.set_tick_params(pad = 10)
    axs.grid(b = True, color ='grey', linestyle ='-.', linewidth = 0.5, alpha = 0.6)

    N, bins, patches = axs.hist(x_column, bins)
    fracs = ((N**(1 / 5)) / N.max())
    norm = colors.Normalize(fracs.min(), fracs.max())
    for thisfrac, thispatch in zip(fracs, patches):
        color = plt.cm.viridis(norm(thisfrac))
        thispatch.set_facecolor(color)

    plt.xlabel('Declared Weight')
    plt.ylabel('Frequency')
    plt.legend(['Declared Weight'])
    plt.title('Histogram')
    plt.show()
Histogram()
```

Output:

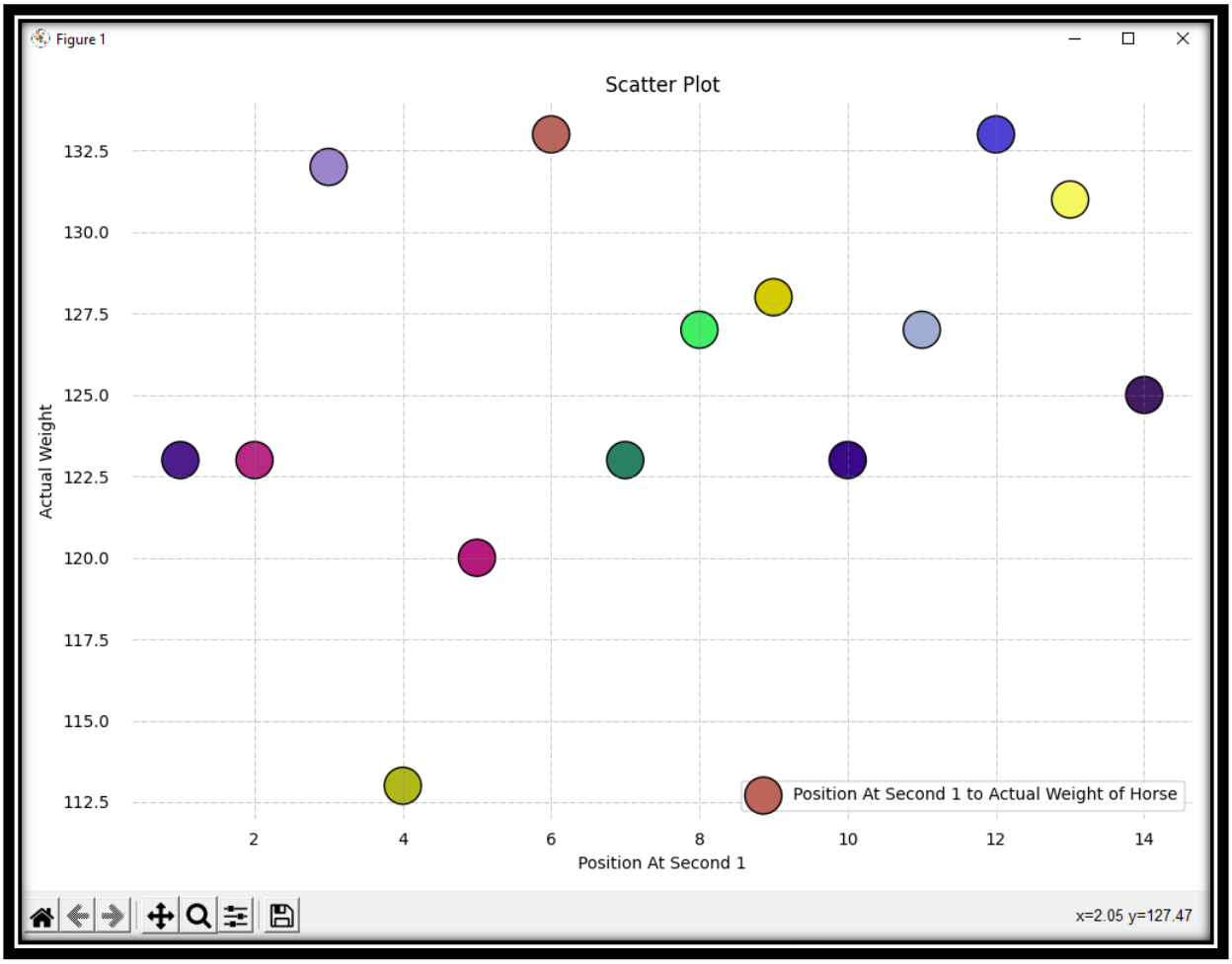


Quantile-Quantile Plot:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
import scipy.stats as stats
from matplotlib import colors
from matplotlib.ticker import PercentFormatter

def Scatter_Plot():
    x_column = []
    y_column = []
    csv_file = pd.read_csv('/Users/Soumen/Downloads/SEM_4
College_Stuff/HON/FDS/runs.csv')
    csv_file_x = csv_file[csv_file['position_sec1'].notna()]
    csv_file_y = csv_file[csv_file['actual_weight'].notna()]
    for i in range(0,14):
        x_column.append(csv_file_x['position_sec1'][i])
        y_column.append(csv_file_y['actual_weight'][i])
```

```
    fig, axs = plt.subplots(1, 1, figsize =(10, 7), tight_layout = True)
    for side in ['top', 'bottom', 'left', 'right']:
        axs.spines[side].set_visible(False)
    axs.xaxis.set_ticks_position('none')
    axs.yaxis.set_ticks_position('none')
    axs.xaxis.set_tick_params(pad = 5)
    axs.yaxis.set_tick_params(pad = 10)
    axs.grid(b = True, color ='grey', linestyle ='-.', linewidth = 0.5, alpha = 0.6)

    plt.scatter(x_column, y_column, c=np.random.rand(14,3), s = 500, edgecolor='black')

    plt.xlabel('Position At Second 1')
    plt.ylabel('Actual Weight')
    plt.legend(['Position At Second 1 to Actual Weight of Horse'])
    plt.title('Scatter Plot')
    plt.show()
Scatter_Plot()
```

Output:



Questions:

1. Why is it important to measure the dispersion in the dataset?

A) Measure of dispersion are important for describing the spread of the data, or its variation around a central value

2. Discuss the other purposes/advantages of the plots used in this experiment.

A) **Advantages of Boxplots**

Graphically display a variable's location and spread at a glance.
Provide some indication of the data's symmetry and skewness.
Unlike many other methods of data display, boxplots show outliers.

Advantages of the q-q plot are:

The sample sizes do not need to be equal.
Many distributional aspects can be simultaneously tested.

Advantages of a histogram are:

Simplicity and versatility.
It can be used in many different situations to offer an insightful look at frequency distribution.

Advantages of Scatter plots:

Show a relationship and a trend in the data relationship.
Show all data points, including minimum and maximum and outliers.
Can highlight correlations.

Outcomes:

CO2: Comprehend descriptive and proximity measures of data
CO4: Comprehend various data visualization techniques and its interpretation

---

Conclusion:

We were able to understand the concept of different plots and we understood their importance, advantages and disadvantages.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 3nd Edition