AlImamMohammadIbnSaudIslamicUniversity
College of Computer and Information Sciences

InformationManagementDepartment

(IM371-DataMining)

Project Report

Instructor:Dr.Shakir&T.Thahab
amalbuhairi@imamu.edu.sa

Student Name (in English):	Student ID	Section No:
Nwara Aljoufi	440019357	371

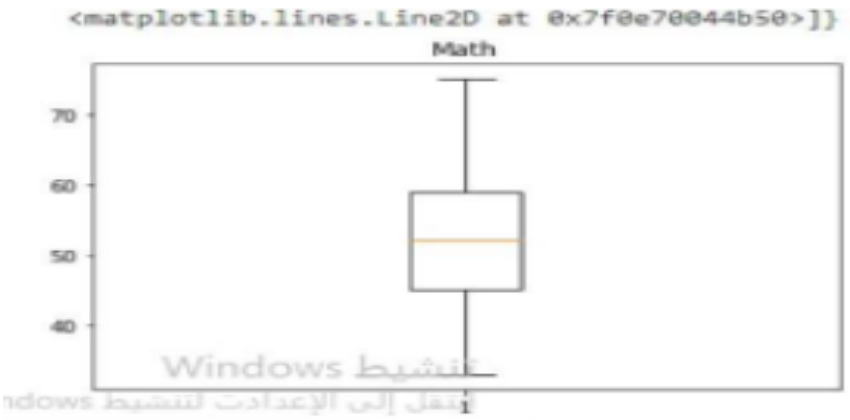
for our project, we are doing 2 jobs for each required package by using Google Colabs
At the beginning of the code we invoke each package we will use to produce 6 functions through the Seaborn ,Matplotlib, Plotly, packages.

Line plot and Box plot from Matplotlib library.

Here Box plot for math from Matplotlib library.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df=pd.read_csv('https://raw.githubusercontent.com/rlowrance/re/master/hsbdemo.csv')
fig1, ax1 = plt.subplots()
ax1.set_title('Math')
ax1.boxplot(df.math)
```

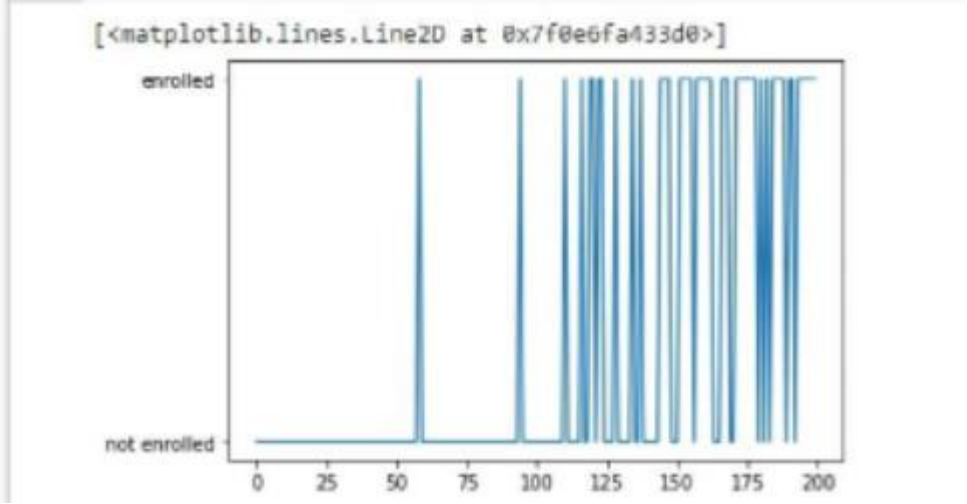
#In this code we imported matplotlib library firstly we determine the DataFrame that we want to plot ('math') from dataset in link, then we initialize the name one of the columns in that DataFrame that we want to plot (“Math”), then we set a name of the column in (ax1.set_title('Math')), write (ax1.boxplot(df.math)) to display the figure of the plot.



Here Line plot for student honors English from Matplotlib library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df=pd.read_csv('https://raw.githubusercontent.com/rlowrance/re/master/hsbdemo.csv')
fig1, ax1 = plt.subplots()
ax1.plot(df.honors)
```

#In this code we imported matplotlib library firstly we determine the DataFrame that we want to plot ('honors') from dataset in link, then we initialize the name one of the columns in that DataFrame that we want to plot ("honors"), then we set a name of the column and write ax1.plot(df.honors) to display the figure of the plot.

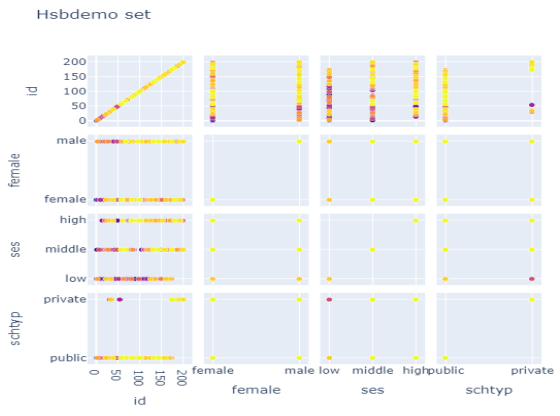


Scatter matrix and Histogram from Plotly library.

Function number one of Scatter matrix

```
import plotly.graph_objects as go
import pandas as pd
df = pd.read_csv('https://raw.githubusercontent.com/rlowrance/re/master/hsbdemo.csv')
index_vals = df['cid'].astype('category').cat.codes
fig = go.Figure(data=go.Splom(
dimensions=[dict(label='id',
values=df['id']),
dict(label='female',
values=df['female']),
dict(label='ses',
values=df['ses']),
dict(label='schtyp',
values=df['schtyp'])],
text=df['cid'],
marker=dict(color=index_vals,
showscale=False, # colors encode categorical variables
line_color='white', line_width=0.5)
                ))
fig.update_layout(
    title='Hsbdemo set',
    dragmode='select',
    width=600,
    height=600,
    hovermode='closest',
)
fig.show()
```

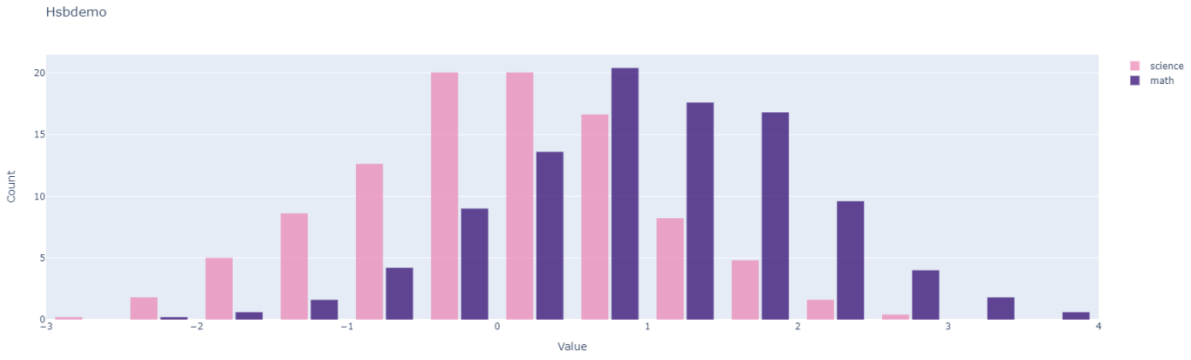
- #We start with implement the Plotly library which all scatter plot in the matrix visualizes the relationship between a pair of variables, allowing many relationships to be explored in one chart. then read the dataset by use the link , We put different lables on the code according to the dataset attributes and assign the colors using pandas label encoding in (index_vals) , Also identify width and height for that figure and finally display that figure using (fig.show()).



Function number one of Histogram.

```
import plotly.graph_objects as go
import numpy as np
x0 = np.random.randn(500)
x1 = np.random.randn(500) + 1
fig = go.Figure()
fig.add_trace(go.Histogram(
x=x0,
histnorm='percent',
name='science', # name used in legend and hover labels
xbins=dict( # bins used for histogram
start=-4.0,
end=3.0,
size=0.5),
marker_color='#EB89B5',
opacity=0.75 ))
fig.add_trace(go.Histogram(
x=x1,
histnorm='percent',
name='math',
xbins=dict(
start=-3.0,
end=4,
size=0.5 ),
marker_color='#330C73',
opacity=0.75))
fig.update_layout(
title_text='Hsbdemo', # title of plot
xaxis_title_text='Value', # xaxis label
yaxis_title_text='Count', # yaxis label
bargap=0.2, # gap between bars of adjacent location coordinates
bargroupgap=0.1 # gap between bars of the same location coordinates
)
fig.show()
```

- We imported plotly librarywhich contains functions that can create entire figures at once, and is referred to as Plotly Express or PX to make histogram plot, its color to be (#EB89B5),Also we need to display difference between(math) and (science) attribute and update figure layout where we set the gap between bars of adjacent location coordinates to be 0.2,Lastly we made sure to write fig.show() to display the figure of the plot.

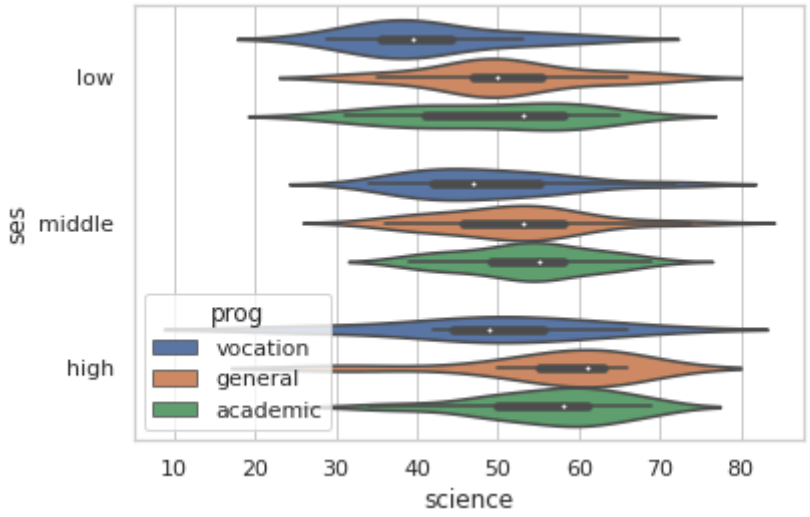


3-**Violin plot fromSeabornlibrary.**

Here violin plot for student science ,ses , prog from seaborn library.

```
import seaborn as sns
sns.set_theme(style="whitegrid")
ax = sns.violinplot(x="science", y="ses", hue="prog", data=df)
```

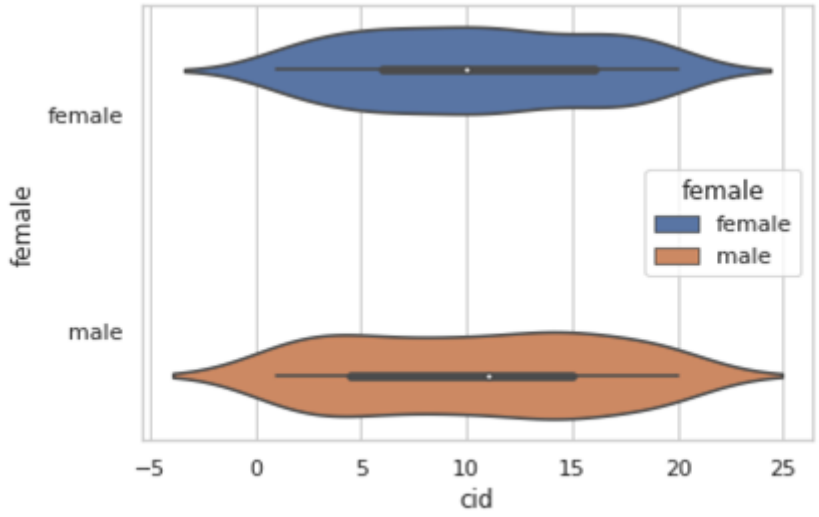
#A violin plot plays , It shows the distribution of quantitative data across several levels of one (or more) categorical variables
and Parameters is x, y, hue names of variables in data
x=science , y= sts , hue=prog
indicates scores in each course and socio economic status and educational choices



Here violin plot for student cid , female from seaborn library.

```
import seaborn as sns
sns.set_theme(style="whitegrid")
ax = sns.violinplot(x="cid", y="female", hue="female", data=df)
```

#A violin plot plays , It shows the distribution of quantitative data across several levels of one (or more) categorical variables
and Parameters is x, y, hue names of variables in data or vector data
x=cid , y= female, hue=female indicates cid and female.



4-Faceting

faceting technique that allows the user to split one

```
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

my_count=["id","female","schypt","ses","science","prog","awards","read","write","socst","honors","cid","schtyp(public)","schtyp(private)","honors(enrolled)","honors(not enrolled)"]
df = pd.DataFrame({
"attribute":np.repeat(my_count, 10),
"years":list(range(2000, 2010)) * 16,
"value":np.random.rand(160)
})
# Create a grid : initialize it
g = sns.FacetGrid(df, col='attribute', hue='attribute', col_wrap=4, )

# Fill the area with fill_between
g = g.map(plt.fill_between, 'years', 'value', alpha=0.2).set_titles("{col_name} attribute")

# Control the title of each facet
g = g.set_titles("{col_name}")

# Add a title for the whole plot
plt.subplots_adjust(top=0.92)
g = g.fig.suptitle('Analyze all attributes')

# Show the graph
plt.show()
```

We start with implement the libraries to help (Faceting) whichallows to quickly find out the different patterns existing in the data. Then createDateFrame using (attribute) and value, Also we put title that will describe the figure, Finally identify width and height for that figure and finally display that plots using (plt.show()).

Appendix

Line plot and Box plot from Matplotlib library.

https://github.com/rlowrance/re/blob/master/hsbdemo.csv

Violin plot fromSeabornlibrary.

Untitled3.ipynb