Muley_Tushar_Exercises_6-2_Charts_Python_Week_11_12

November 11, 2021

Name: Tushar Muley

Assignment: Week 10-11 Exercises 6.2

Date:November 20, 20121

[1]:
```
# import libraries

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

[3]:
```
# update settings

pd.set_option('display.max_columns', None)
```

[13]:
```
# load data

# first file
file1 = 'crimeratesbystate-formatted.xlsx'
crime = pd.read_excel(file1)
```

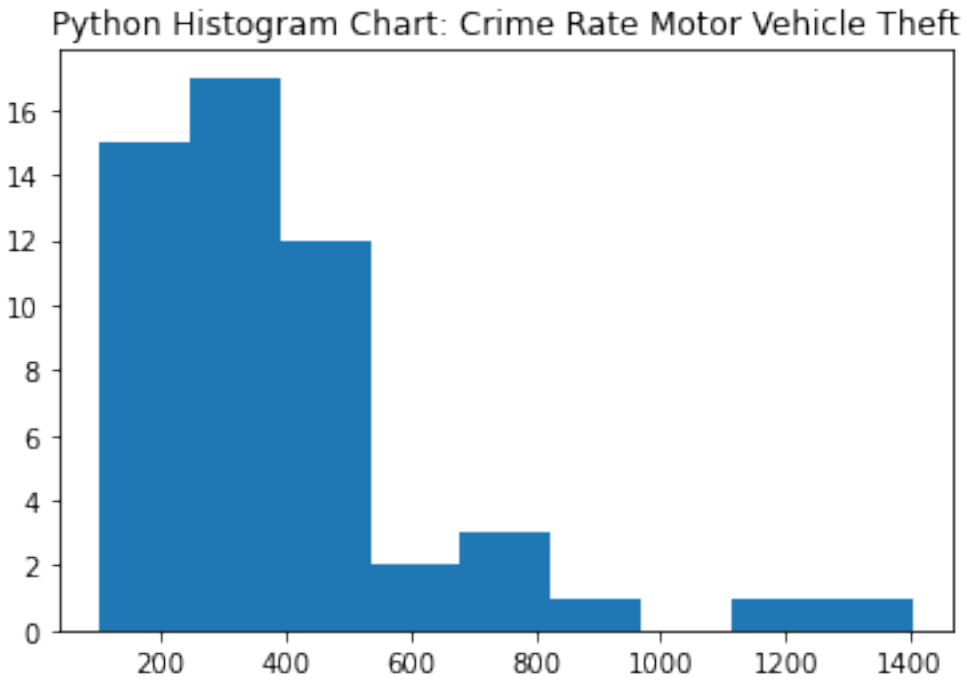
[14]:
```
# second file
file2 ='birth-rate.xlsx'
brate = pd.read_excel(file2)
```

[5]:
```
crime.columns
```

[5]:
```
Index(['state', 'murder', 'forcible_rape', 'robbery', 'aggravated_assault',
       'burglary', 'larceny_theft', 'motor_vehicle_theft'],
      dtype='object')
```

1 Histogram

[23]:
```
plt.hist(crime.motor_vehicle_theft,bins=9)
plt.title('Python Histogram Chart: Crime Rate Motor Vehicle Theft')
plt.show()
```

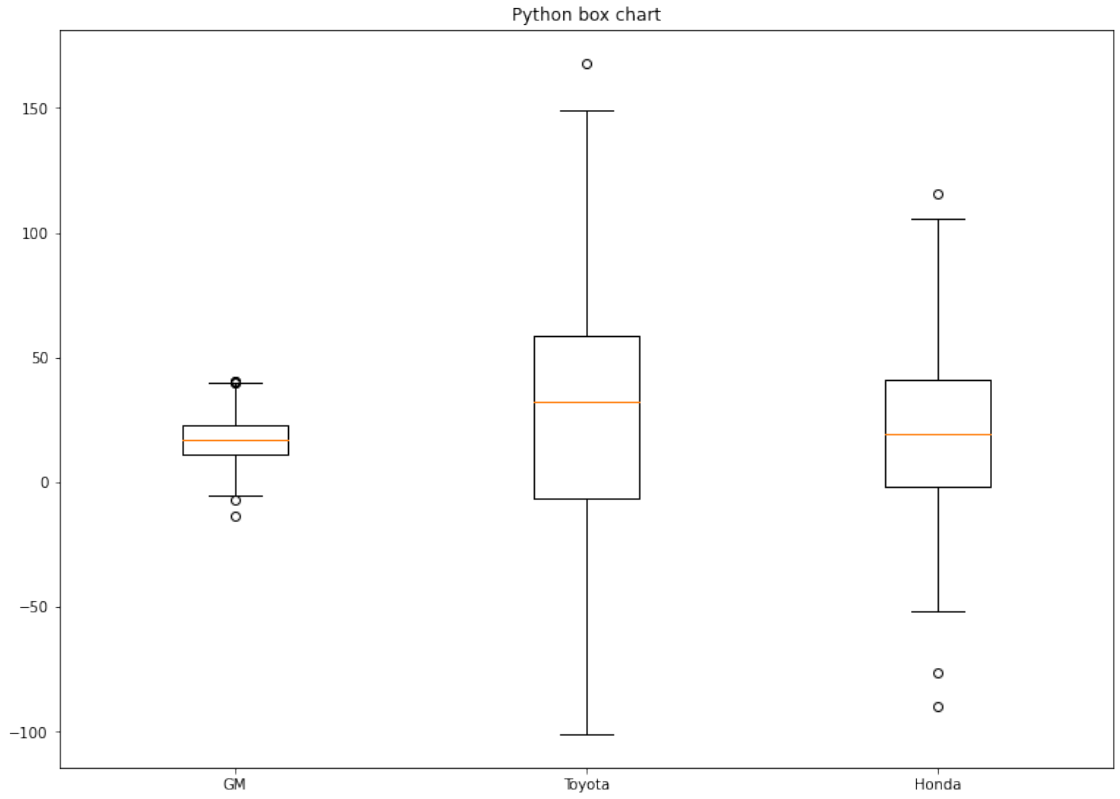


2 Box plot

[34]:
```
# Creating dataset
np.random.seed(10)

data_1 = np.random.normal(16, 10, 200)
data_2 = np.random.normal(26, 53, 200)
data_3 = np.random.normal(22, 35, 200)
data = [data_1, data_2, data_3]

fig = plt.figure(figsize =(10, 7))

# Creating axes instance
ax = fig.add_axes([0, 0, 1, 1])

# x-axis labels
ax.set_xticklabels(["GM", "Toyota",
```

```
                    "Honda"])

# Adding title
plt.title("Python box chart")

# Creating plot
bp = ax.boxplot(data)

# show plot
plt.show()
```

```
<ipython-input-34-81c7f6282c9d>:15: UserWarning:

FixedFormatter should only be used together with FixedLocator
```



[ ]:

3 Bullet chart

[31]:
```
# loaded library
import plotly.figure_factory as ff

# make up data

data = [
  {"label": "Toyota",
   "sublabel": "MPG",
   "range": [25, 32, 53],
   "performance": [27, 30],
   "point": [31]},

  {"label": "Honda",
   "sublabel": "MPG",
   "range": [20, 25, 30],
   "performance": [21, 23],
   "point": [26]},

  {"label": "GM",
   "sublabel": "MPG",
   "range": [16, 25, 42],
   "performance": [19, 23],
   "point": [22]},

]

# built the bullet chart

fig = ff.create_bullet(
    data, titles="label",
    subtitles="sublabel",
    markers="point",
    measures="performance",
    ranges="range",
    orientation="h",
    title="Python Bullet Charts of MPG"

)

fig.show()
```

4 My Choice Chart

[21]:
```
# libraries
import plotly.graph_objects as go
from sklearn.datasets import make_moons
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

mesh_size = .02
margin = 0.25

# load and split data
X, y = make_moons(noise=0.3, random_state=0)
X_train, X_test, y_train, y_test = train_test_split(
    X, y.astype(str), test_size=0.25, random_state=0)

# create a mesh grid on which we will run our model
x_min, x_max = X[:, 0].min() - margin, X[:, 0].max() + margin
y_min, y_max = X[:, 1].min() - margin, X[:, 1].max() + margin
xrange = np.arange(x_min, x_max, mesh_size)
yrange = np.arange(y_min, y_max, mesh_size)
xx, yy = np.meshgrid(xrange, yrange)

# create classifier, run predictions on grid
clf = KNeighborsClassifier(15, weights='uniform')
clf.fit(X, y)
Z = clf.predict_proba(np.c_[xx.ravel(), yy.ravel()])[:, 1]
Z = Z.reshape(xx.shape)


# plot the figure
fig = go.Figure(data=[
    go.Contour(
        x=xrange,
        y=yrange,
        z=Z,
        colorscale='RdBu'
    )
])

fig.update_layout(
    title='Python My Choice Contour Chart')
fig.show()
```

[22]:
```
# load library
from plotly.subplots import make_subplots
```

```
fig = make_subplots(rows=2, cols=2,
                    subplot_titles=('connectgaps = False',
                                                    'connectgaps = True'))
# data
z = [[None, None, None, 12, 13, 14, 15, 16],
     [None, 1, None, 11, None, None, None, 17],
     [None, 2, 6, 7, None, None, None, 18],
     [None, 3, None, 8, None, None, None, 19],
     [5, 4, 10, 9, None, None, None, 20],
     [None, None, None, 27, None, None, None, 21],
     [None, None, None, 26, 25, 24, 23, 22]]

fig.add_trace(go.Contour(z=z, showscale=False), 1, 1)
fig.add_trace(go.Contour(z=z, showscale=False, connectgaps=True), 1, 2)
fig.add_trace(go.Heatmap(z=z, showscale=False, zsmooth='best'), 2, 1)
fig.add_trace(go.Heatmap(z=z, showscale=False, connectgaps=True,␣
 ↪zsmooth='best'), 2, 2)

fig['layout']['yaxis1'].update(title='Contour map')
fig['layout']['yaxis3'].update(title='Heatmap')
fig.update_layout(
    title='Python My Choice Contour Chart')
fig.show()
```