

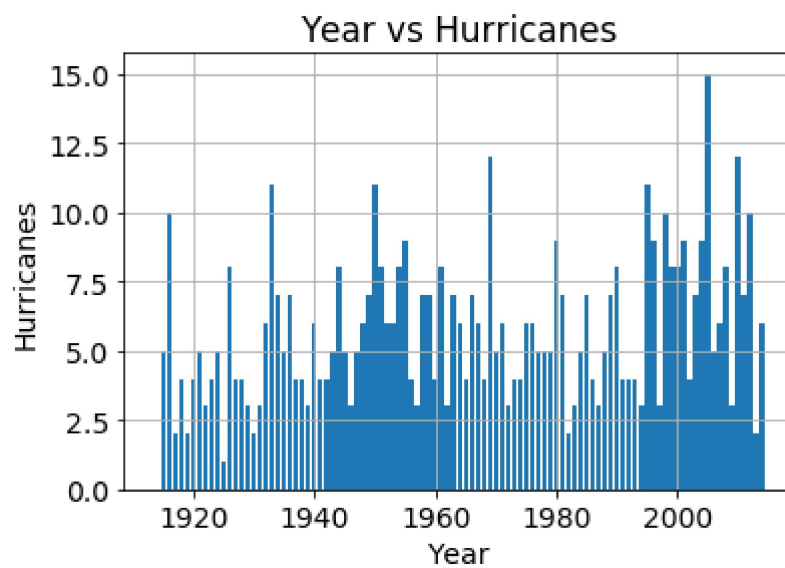
## Case Study 3

```
In [46]: import pandas as pd
import matplotlib.pyplot as plt
```

```
In [47]: #Task 1 solution-
df = pd.read_csv("Hurricanes.csv")
x = df["Year"]
y = df["Hurricanes"]

plt.bar(x,y)
plt.xlabel("Year")
plt.ylabel("Hurricanes")
plt.grid()
plt.title("Year vs Hurricanes")
```

```
Out[47]: Text(0.5, 1.0, 'Year vs Hurricanes')
```



```

In [48]: #Task 2 solution-
df_temp = pd.read_csv("CityTemps.csv")

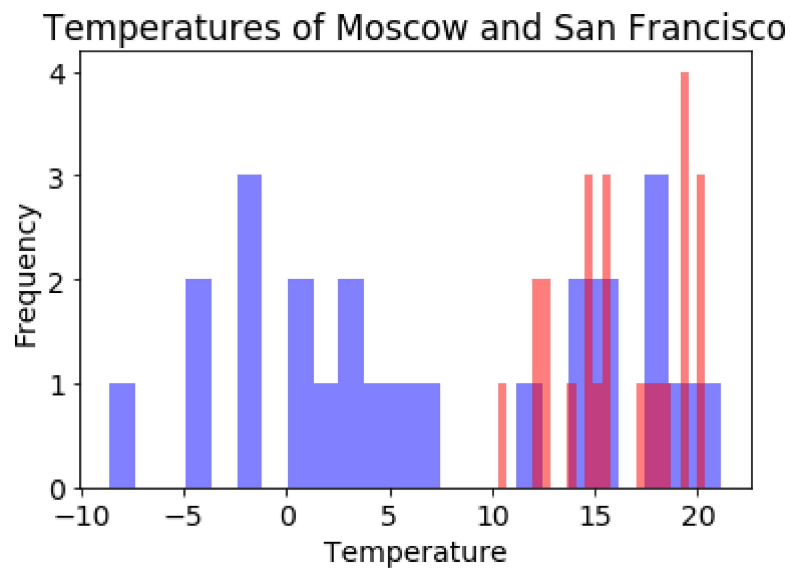
x_mosco = df_temp["Moscow"]
x_san_Francisco = df_temp["San Francisco"]

plt.hist(x_mosco, len(x_mosco), facecolor='blue', alpha=0.5)
plt.hist(x_san_Francisco, len(x_san_Francisco), facecolor='red', alpha=0.5)

plt.xlabel("Temperature")
plt.ylabel("Frequency")
plt.title("Temperatures of Moscow and San Francisco")

```

Out[48]: Text(0.5, 1.0, 'Temperatures of Moscow and San Francisco')



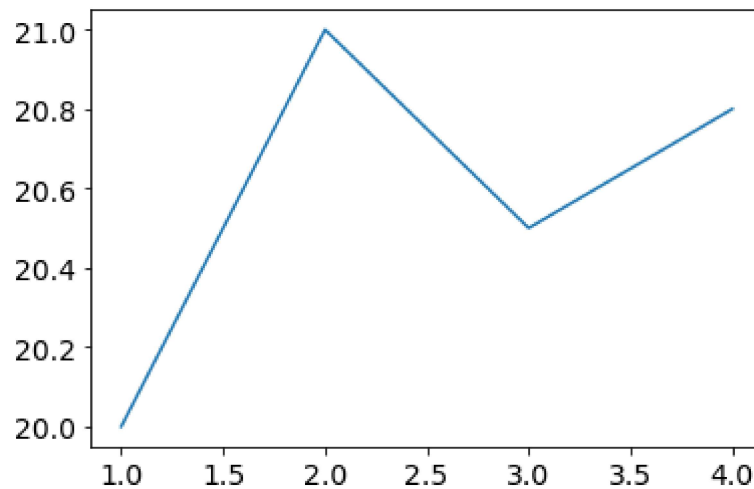
```

In [49]: #Task 3 solution-
M4_assign_dataset = pd.read_csv("data_file.txt")
M4_assign_dataset.to_csv("data_file.csv")

```

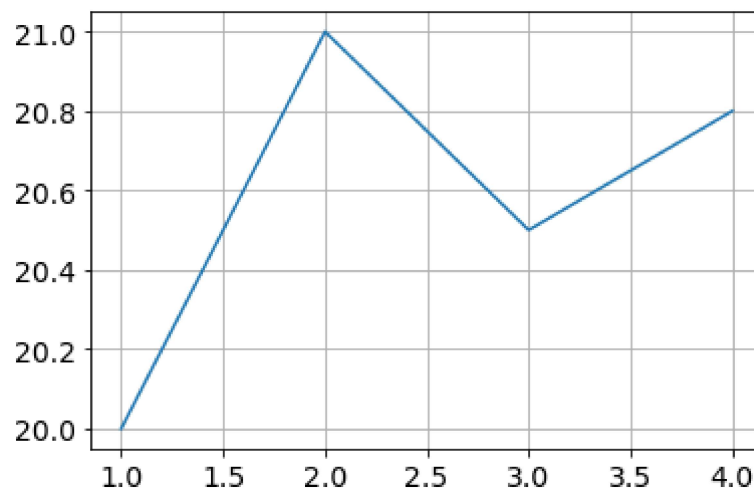
```
In [50]: #Task 4 solution-  
x = [1,2,3,4]  
y = [20, 21, 20.5, 20.8]  
  
#5.1 - draw a simple plot  
plt.plot(x,y)
```

Out[50]: [



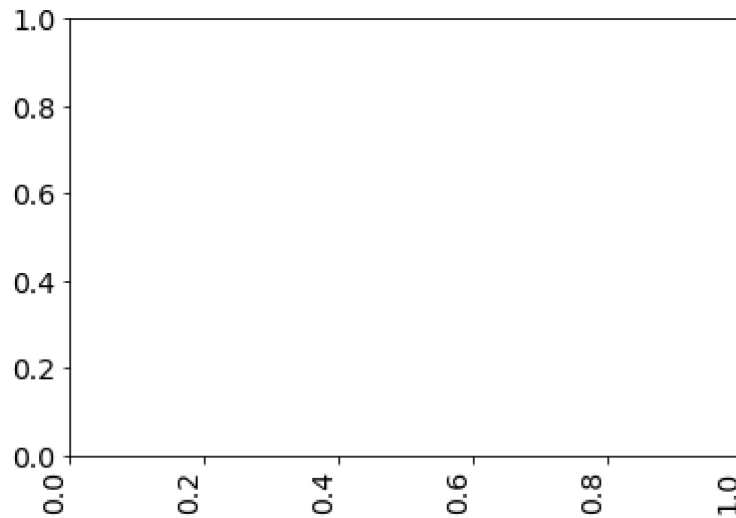
```
In [51]: #5.2 - Configure the line and markers in simple plot  
plt.grid()  
plt.plot(x,y)
```

Out[51]: [



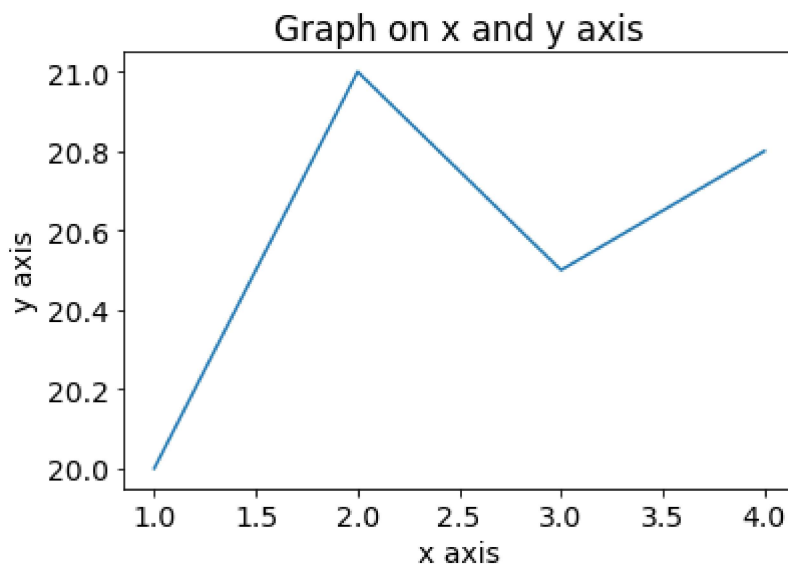
```
In [52]: #5.3 - configure the axes
plt.setp(plt.gca().get_xticklabels(), rotation=90, horizontalalignment='right')
) #rotate axis labels
```

Out[52]: [None, None, None, None, None, None, None, None, None, None, None, None]



```
In [53]: #5.4 - Give title of Graph & labels of x axis and y axis
plt.plot(x,y)
plt.title("Graph on x and y axis")
plt.xlabel("x axis")
plt.ylabel("y axis")
```

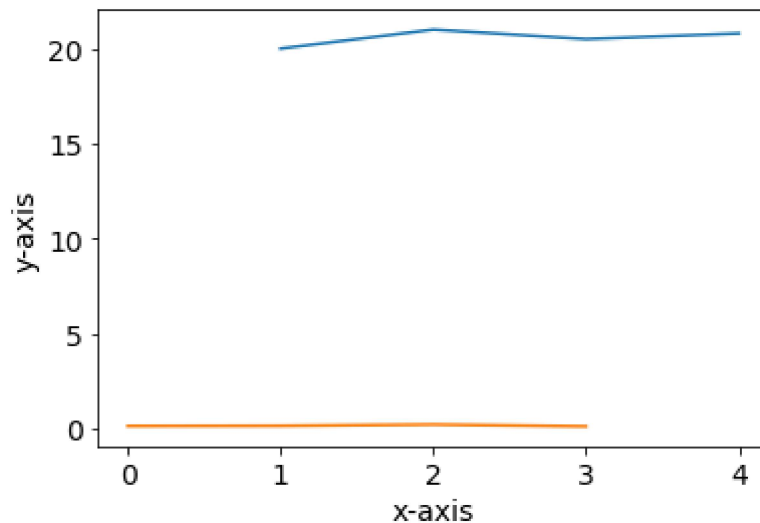
Out[53]: Text(0, 0.5, 'y axis')



In [54]: *#5.5- Give error bar if y\_error = [0.12, 0.13, 0.2, 0.1]*

```
y_error = [0.12, 0.13, 0.2, 0.1]
plt.plot(x,y,y_error)
plt.xlabel("x-axis")
plt.ylabel("y-axis")
```

Out[54]: Text(0, 0.5, 'y-axis')



In [36]: *#5.6- define width, height as figsize=(4,5) DPI and adjust plot dpi=100*  
plt.figure(figsize=(3,4))

Out[36]: <Figure size 216x288 with 0 Axes>

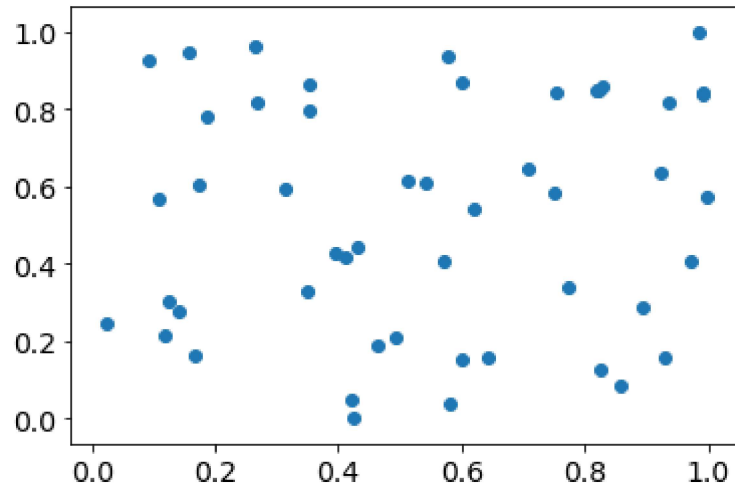
<Figure size 216x288 with 0 Axes>

In [55]: *#5.7- Give a font size of 14*  
plt.rcParams.update({'font.size':14})

```
In [56]: #5.8- Draw a scatter graph of any 50 random values of x and y axis
x=np.random.random(50)
y=np.random.random(50)

plt.scatter(x,y)
```

```
Out[56]: <matplotlib.collections.PathCollection at 0x1b0dd1a5c88>
```



```

In [58]: #5.9- Create a dataframe from following data
# 'first_name': ['Jason', 'Molly', 'Tina', 'Jake', 'Amy'],
# 'last_name': ['Miller', 'Jacobson', 'Ali', 'Milner', 'Cooze'],
# 'female': [0, 1, 1, 0, 1],
# 'age': [42, 52, 36, 24, 73],
# 'preTestScore': [4, 24, 31, 2, 3],
# 'postTestScore': [25, 94, 57, 62, 70]
# Draw a Scatterplot of preTestScore and postTestScore, with the size of each point determined by age

df = pd.DataFrame({
    'first_name': ['Jason', 'Molly', 'Tina', 'Jake', 'Amy'],
    'last_name': ['Miller', 'Jacobson', 'Ali', 'Milner', 'Cooze'],
    'female': [0, 1, 1, 0, 1],
    'age': [42, 52, 36, 24, 73],
    'preTestScore': [4, 24, 31, 2, 3],
    'postTestScore': [25, 94, 57, 62, 70]

})

plt.scatter(df["preTestScore"], df["postTestScore"])
plt.xlabel("preTestScore")
plt.ylabel("postTestScore")
plt.title("preTestScore vs postTestScore")
plt.legend(["preTestScore", "postTestScore"])

```

Out[58]: <matplotlib.legend.Legend at 0x1b0dd238908>



```

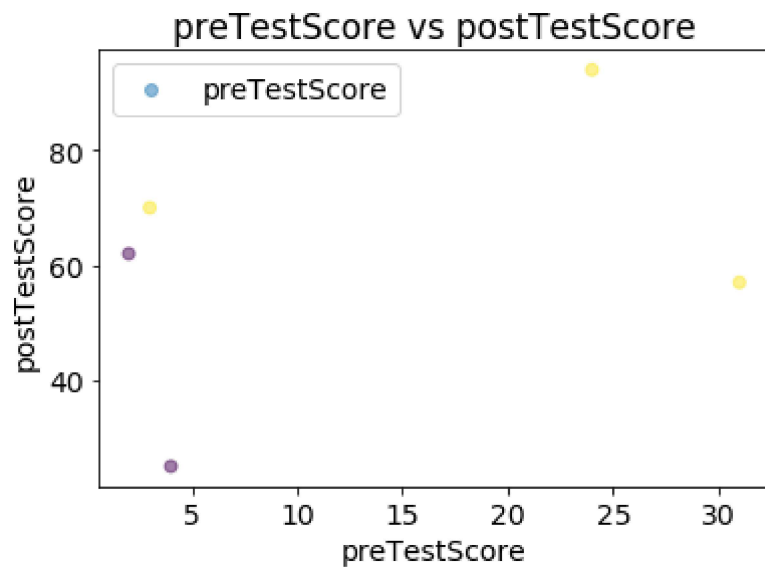
In [59]: #Draw a Scatterplot from the data in question 9 of preTestScore and postTestScore with the size = 300 and
#the color determined by sex
df = pd.DataFrame({
    'first_name': ['Jason', 'Molly', 'Tina', 'Jake', 'Amy'],
    'last_name': ['Miller', 'Jacobson', 'Ali', 'Milner', 'Cooze'],
    'female': [0, 1, 1, 0, 1],
    'age': [42, 52, 36, 24, 73],
    'preTestScore': [4, 24, 31, 2, 3],
    'postTestScore': [25, 94, 57, 62, 70]
})

x=df["preTestScore"]
y=df["postTestScore"]
colors = df["female"]

plt.scatter(x, y, c=colors, alpha=0.5)
plt.xlabel("preTestScore")
plt.ylabel("postTestScore")
plt.title("preTestScore vs postTestScore")
plt.legend(["preTestScore", "postTestScore"])

```

Out[59]: <matplotlib.legend.Legend at 0x1b0de274148>



In [ ]: