Midtem-2022

- Name: Meet Hiteshkumar Trivedi
- Student Number: N01520331

Example1: (1 Mark)

1.create a dictionary for the consumer items which contains prices of 3 some items and names , convert the above dictionary into a series.

In [44]:

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

In [45]:

```
%matplotlib inline
```

In [4]:

```
items = dict()

items["item1"] = ["Item101", 29.99];
items["item2"] = ["Item102", 19.99];
items["item3"] = ["Item103", 9.99];

data = pd.DataFrame.from_dict(items)
data
```

Out[4]:

	item1	item2	item3
0	Item101	Item102	Item103
1	29.99	19.99	9.99

Example2: (1 Mark)

1.Create a 4x4 - 2 dimensional array and then slice it into half horizontally.

In [10]:

```
arr = np.array([[1,2,3,4],[5,6,7,8],[2,4,7,9],[3,5,8,6]])
print("Original array:\n",arr)
print("After splitting horizontally:")
print(np.hsplit(arr, [2, 6]))
```

```
Original array:
 [[1 2 3 4]
 [5 6 7 8]
 [2 4 7 9]
 [3 5 8 6]]
After splitting horizontally:
[array([[1, 2],
       [5, 6],
       [2, 4],
       [3, 5]]), array([[3, 4],
       [7, 8],
       [7, 9],
       [8, 6]]), array([], shape=(4, 0), dtype=int32)]
```

Example3:(3 Mark)

- Create a 3x9 data frame with some random nan values in it.
- Drop all nan values.
- Drop only those rows where all column values are nan.
- Once the row is dropped, replace the remaining nan values wih a value of 0.

In [16]:

```
nan_data = {'List1':  [1,2,3,4,5,np.nan,6,7,np.nan],
        'List2': [11,12,np.nan,13,14,np.nan,15,16,np.nan],
        'List3': [20,21,22,23,24,np.nan,np.nan,26,27]
       }
df1 = pd.DataFrame(nan_data)
print (df1)
print ("----------------")

df2 = df1.dropna()
print (df2)
print ("----------------")

df3 = df1.dropna(how="all")
print(df3)

print ("----------------")
df4 = df3.fillna(0)
print(df4)
```

```
   List1  List2  List3
0    1.0   11.0   20.0
1    2.0   12.0   21.0
2    3.0    NaN   22.0
3    4.0   13.0   23.0
4    5.0   14.0   24.0
5    NaN    NaN    NaN
6    6.0   15.0    NaN
7    7.0   16.0   26.0
8    NaN    NaN   27.0
----------------
   List1  List2  List3
0    1.0   11.0   20.0
1    2.0   12.0   21.0
3    4.0   13.0   23.0
4    5.0   14.0   24.0
7    7.0   16.0   26.0
----------------
   List1  List2  List3
0    1.0   11.0   20.0
1    2.0   12.0   21.0
2    3.0    NaN   22.0
3    4.0   13.0   23.0
4    5.0   14.0   24.0
6    6.0   15.0    NaN
7    7.0   16.0   26.0
8    NaN    NaN   27.0
----------------
   List1  List2  List3
0    1.0   11.0   20.0
1    2.0   12.0   21.0
2    3.0    0.0   22.0
3    4.0   13.0   23.0
4    5.0   14.0   24.0
6    6.0   15.0    0.0
7    7.0   16.0   26.0
8    0.0    0.0   27.0
```

Example4: (3 Mark)

1.Create a series with 5 elements.

2.Create a 2*5 data frame with some random values.

3.Subtract the series from the above data frame.

In [25]:

```
# 1.
data = np.array(['A','B','C','D','E'])

ser = pd.Series(data)
print(ser)

# 2.
df = pd.DataFrame(np.random.randint(0,50,size=(2, 5)), columns=list('ABCDE'))
print(df)

# 3.
df = df.subtract(ser)
print(df)
```

```
0    A
1    B
2    C
3    D
4    E
dtype: object
    A   B  C   D   E
0  15  22  5  30   1
1  21  29  4  35  25
     A    B    C    D    E    0    1    2    3    4
0  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
1  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
```

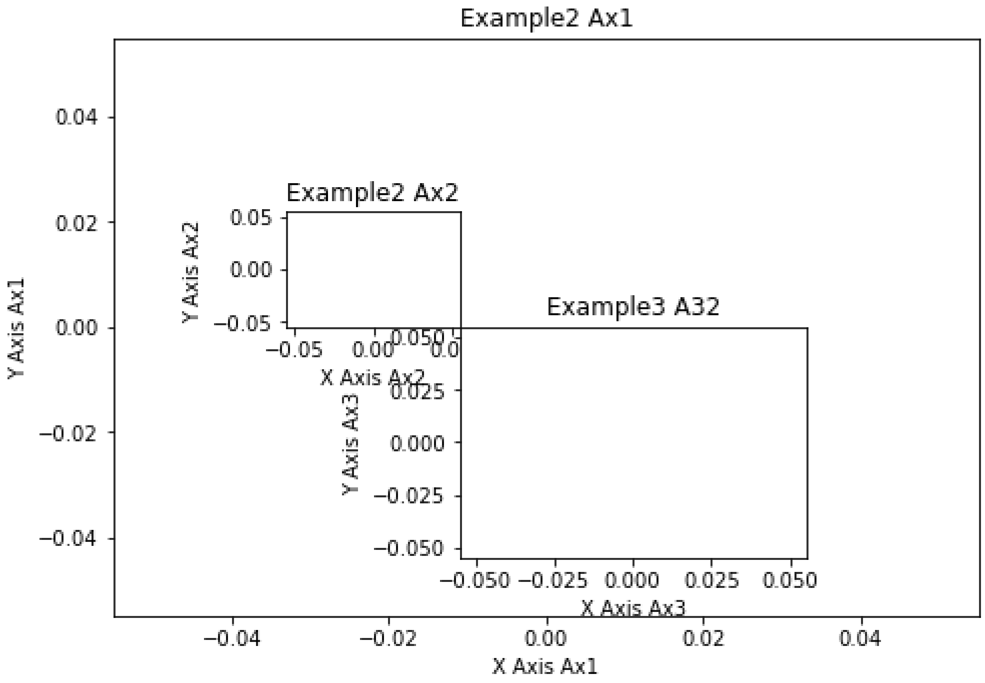
Example5:(3 Mark)

Create two figures.

1. First figure has only one subplot. Second figure should have two subplots.
2. Plot some random data into those subplots.

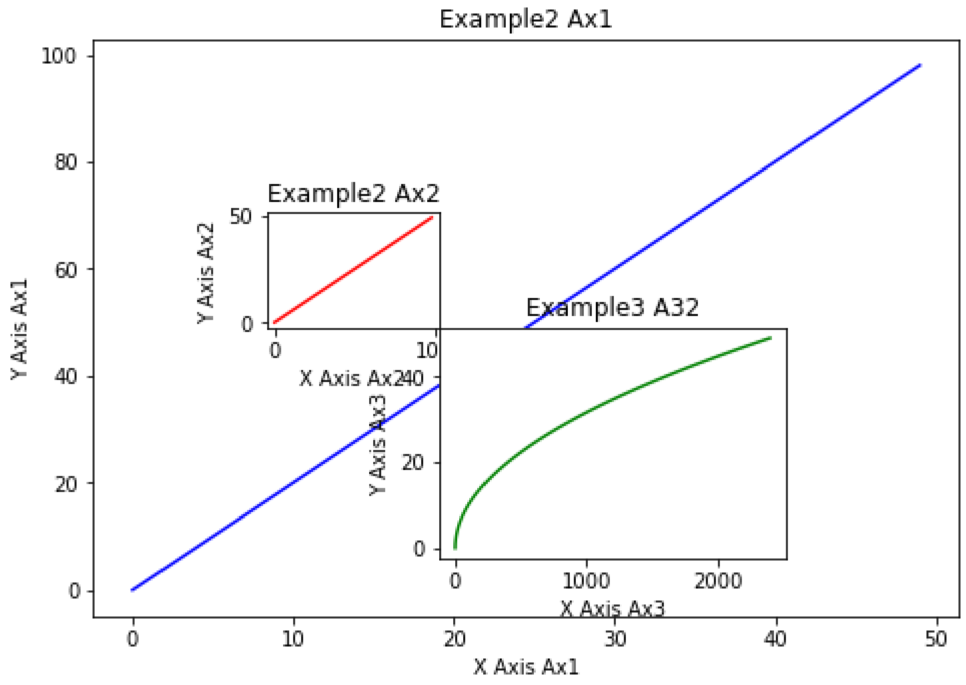
In [39]:

```
# 1.
# import matplotlib.pyplot as plt
x = np.arange(0,50)
y = x*2
z = x**2


fig = plt.figure()

ax1 = fig.add_axes([0, 0, 1, 1]) # main ax
ax2 = fig.add_axes([0.2, 0.5, 0.2, 0.2]) # inset ax
ax3 = fig.add_axes([0.4, 0.1, 0.4, 0.4]) # inset ax


# Larger Figure Ax 1
ax1.plot()
ax1.set_xlabel('X Axis Ax1')
ax1.set_ylabel('Y Axis Ax1')
ax1.set_title('Example2 Ax1')


# Insert Figure Ax 2
ax2.plot()
ax2.set_xlabel('X Axis Ax2')
ax2.set_ylabel('Y Axis Ax2')
ax2.set_title('Example2 Ax2');

# Insert Figure Ax 3
ax3.plot()
ax3.set_xlabel('X Axis Ax3')
ax3.set_ylabel('Y Axis Ax3')
ax3.set_title('Example3 A32');
```



In [43]:

```
# 2.
x = np.arange(0,50)
y = x*2
z = x**2


fig = plt.figure()

ax1 = fig.add_axes([0, 0, 1, 1]) # main ax
ax2 = fig.add_axes([0.2, 0.5, 0.2, 0.2]) # inset ax
ax3 = fig.add_axes([0.4, 0.1, 0.4, 0.4]) # inset ax

# Larger Figure Ax 1
ax1.plot(x, y, 'b')
ax1.set_xlabel('X Axis Ax1')
ax1.set_ylabel('Y Axis Ax1')
ax1.set_title('Example2 Ax1')


# Insert Figure Ax 2
ax2.plot(y, x, 'r')
ax2.set_xlabel('X Axis Ax2')
ax2.set_ylabel('Y Axis Ax2')
ax2.set_title('Example2 Ax2');

# Insert Figure Ax 3
ax3.plot(z, x, 'g')
ax3.set_xlabel('X Axis Ax3')
ax3.set_ylabel('Y Axis Ax3')
ax3.set_title('Example3 A32');
```



Example6: (4 Mark)

```
1. Read the dataset and create dataframe
2. Create a new column called 'Average' that contains the Average of 3 columns (Mi
ni_Exam1, Mini_Exam2, Mini_Exam3)
3. use the scatter plot and find correlation between all numerical features.
4. write your analysis on the relationship with different features?
5. Use pie plot to display information of Grade column.
```

In [46]:

```
# 1.
grade = pd.read_csv('E:\Programming\Humber college\Humber Sem 2\Data Analytics\Week-8/Grade
grade.head()
```

Out[46]:

	Name	Mini_Exam1	Mini_Exam2	Participation	Mini_Exam3	Final	Grade
0	Jake	19.5	20.0	1	10.0	33.0	A
1	Joe	20.0	16.0	1	14.0	32.0	A
2	Susan	19.0	19.0	1	10.5	33.0	A-
3	Sol	22.0	13.0	1	13.0	34.0	A
4	Chris	19.0	17.0	1	12.5	33.5	A

In [52]:

```
# 2.
grade["Average"] = grade[["Mini_Exam1","Mini_Exam2","Mini_Exam3"]].mean(axis=1)
grade.head()
```

Out[52]:

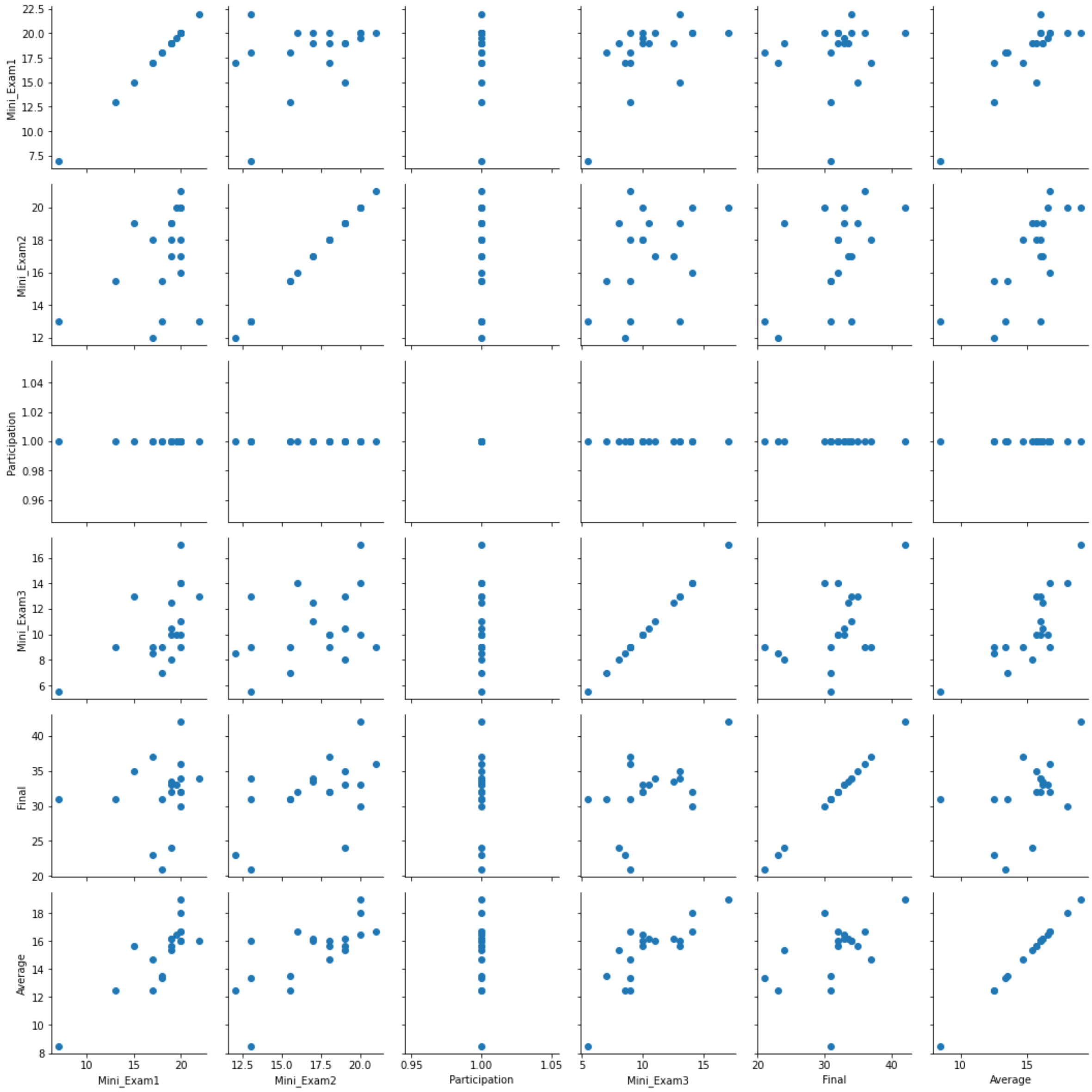
	Name	Mini_Exam1	Mini_Exam2	Participation	Mini_Exam3	Final	Grade	Average
0	Jake	19.5	20.0	1	10.0	33.0	A	16.500000
1	Joe	20.0	16.0	1	14.0	32.0	A	16.666667
2	Susan	19.0	19.0	1	10.5	33.0	A-	16.166667
3	Sol	22.0	13.0	1	13.0	34.0	A	16.000000
4	Chris	19.0	17.0	1	12.5	33.5	A	16.166667

In [53]:

```
# 3.
g = sns.PairGrid(grade)
g.map(plt.scatter)
```

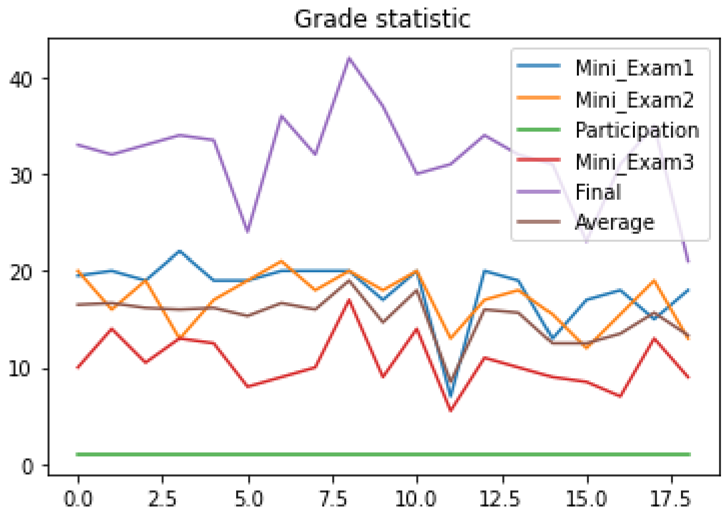
Out[53]:

```
<seaborn.axisgrid.PairGrid at 0x264c60f2d60>
```



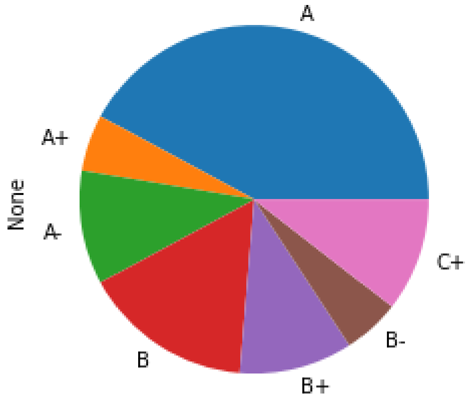
In [57]:

```
# 4.
grade.plot()
plt.title('Grade statistic')
plt.show()
```



In [58]:

```
# 5.
gr = grade['Grade']
gr.str.get_dummies(sep = ",").sum().plot.pie();
```



In [ ]:

```

```