Part -1

1) Write a python program to compute distance between Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):

(a) Compute the Euclidean distance between the two objects.

(b) Compute the Manhattan distance between the two objects.

(c) Compute the Minkowski distance between the two objects, using q = 3.

(d) Compute the supremum distance between the two objects.

```
import math as m
x = (20,2,30,10)
y = (20,5,34,6)
sum = 0

for i in range(len(x)):
    sum += (x[i] - y[i])**2
print("Euclidean Distance:: ", m.sqrt(sum))

for i in range(len(x)):     sum +=
(abs(x[i] - y[i]))
print("Manhattan Distance:: ",sum)

for i in range(len(x)):     sum
+= (abs(x[i] - y[i]))**3
print("Minkowski Distance:: ",(sum)**(1/3))

for i in range(len(x)):     sum
=max(sum,abs(x[i] - y[i]))
print("Supremum Distance:: ",sum)
```

```
Euclidean Distance::  6.4031242374328485
Manhattan Distance::  52
Minkowski Distance::  5.915481699700716
Supremum Distance::  207
```

2) Perform Preprocessing on Titanic Data set Using Orange Tools

3) Kindly Perform Data Exploration on New Restaurant Data Set

Link - https://github.com/guipsamora/pandas_exercises/blob/master/01_Getting_%26_Knowing_Your_Data/Chipotle/Exercises.ipynb

PART - 2

```
import pandas as pd
```

1) First, you need to read the titanic dataset from local disk and display Last five records

```
df = pd.read_csv('titanicC.csv')
df
```

2) Handle Missing Values in data set [use dropna(), fillna(), and interpolate]

```
df.count()


df.isnull().sum()

df1 =df.dropna()
df1

df1=  df.fillna({'Age':'0','Cabin':'notdifined','Embarked':'notgive'})
df1

df1.isnull().sum()

df.Age.interpolate(method='linear',limit_direction='forward',axis=0)

df4=df
df4['Age']=df4['Age'].interpolate(method='linear',limit_direction='for
ward',axis=0) df4
```

3) Write programs to perform the following tasks of preprocessing.

Equal Width Binning Equal Frequency/Depth Binning

```
import numpy as np

data=[5,10,11,13,15,35,50,55,72,92,204,215]
temp = pd.DataFrame(data,columns=['Value'])
num_bins=3 bin_edges=
np.linspace(temp['Value'].min(),temp['Value'].max(),num_bins+1)
temp['Equal_Width_Bin'] =
```

```
pd.cut(temp['Value'],bins=bin_edges,labels=False,include_lowest=True)
print("Equal width Binning:: \n",temp)

data=[5,10,11,13,15,35,50,55,72,92,204,215]
num_bins=3
no_of_data = len(data)
points_in_bin = no_of_data / num_bins

for i  in range (0,len(data),int(points_in_bin)):
print(data[i:i+int(points_in_bin)])
```

4) Apply Scaling to AGE attribute with min max, decimal scaling and z score.

```
import pandas as pd
import numpy as np

##MinMax
df5=df1
new_max=1
new_min=0
df5['Age_MinMax'] = (((df1['Age'] - df1['Age'].min()) /
(df1['Age'].max()) - df1['Age'].min())) df5

##Decimal
df5['Decimal_scall'] = df['Age']/10**len(str(int(df['Age'].max())))
df5 ##ZScore

df5['age_zscore'] = (df1['Age'] - df1['Age'].mean()) /
df1['Age'].std() df5
```