ASSIGNMENT 3

PROGRAM 1

import numpy as np import pandas as pd import math from scipy import stats

In [3]:

```
x=np.random.sample(200)
x
```

Out[3]:

```
array([0.9637415 , 0.20039626, 0.67248269, 0.12058807, 0.80911636,
       0.76405038, 0.01126666, 0.357481  , 0.1343382 , 0.90564926,
       0.87244251, 0.9091042 , 0.5678616 , 0.39887118, 0.33246707,
       0.43307178, 0.97072831, 0.26927224, 0.01157733, 0.23547777,
       0.247737  , 0.61491822, 0.59293563, 0.53483127, 0.88733183,
       0.68001448, 0.87809398, 0.49677541, 0.9290637 , 0.92733582,
       0.07857915, 0.76686127, 0.1580078 , 0.33530162, 0.12316588,
       0.59076507, 0.53555693, 0.5073795 , 0.1718071 , 0.60693477,
       0.11125503, 0.22348865, 0.04928585, 0.61530126, 0.84779738,
       0.16537727, 0.12585024, 0.56454434, 0.46813816, 0.36816886,
       0.26626047, 0.39567694, 0.07631512, 0.9188296 , 0.01502077,
       0.63598486, 0.61392013, 0.41818139, 0.133945  , 0.62598833,
       0.24042823, 0.12500523, 0.57988015, 0.86355138, 0.2295279 ,
       0.40146491, 0.05426558, 0.43161135, 0.50957927, 0.12390576,
       0.15472474, 0.54956839, 0.79214412, 0.36723833, 0.85310114,
       0.59625243, 0.11750267, 0.11493269, 0.82272231, 0.34197666,
       0.54371221, 0.1287351 , 0.63026069, 0.33244175, 0.44016932,
       0.11912384, 0.55192706, 0.10774067, 0.94664624, 0.4779483 ,
       0.83949141, 0.18255968, 0.76108135, 0.46505725, 0.8439175 ,
       0.50608939, 0.38753264, 0.47261945, 0.41624681, 0.41637014,
       0.93194244, 0.4379085 , 0.49614132, 0.21737328, 0.04457875,
       0.90079608, 0.78515653, 0.84684093, 0.81844033, 0.44102101,
       0.80660896, 0.31844372, 0.34510169, 0.38458502, 0.72546685,
       0.32319211, 0.92632193, 0.88216166, 0.70352835, 0.5286202 ,
       0.29412447, 0.14550106, 0.60306935, 0.02598791, 0.15761612,
       0.57146222, 0.39876915, 0.13592853, 0.20905704, 0.28281508,
       0.08619144, 0.23682981, 0.97612048, 0.51144694, 0.16662551,
       0.15693694, 0.19885465, 0.38680181, 0.58716778, 0.23966462,
       0.7894308 , 0.05130857, 0.73526626, 0.13084165, 0.440171  ,
       0.13022762, 0.01875594, 0.85860638, 0.77256088, 0.48938158,
       0.33432439, 0.7897252 , 0.82669687, 0.61911241, 0.39209156,
       0.94106117, 0.11500512, 0.04066638, 0.10409113, 0.75767915,
       0.94296928, 0.95257581, 0.65204626, 0.37597823, 0.06126636,
       0.41008756, 0.15204315, 0.03281421, 0.1984741 , 0.91677158,
       0.57072072, 0.54445202, 0.93755896, 0.61131163, 0.73341077,
       0.2775896 , 0.08951451, 0.77399204, 0.25616799, 0.22521273,
       0.14744975, 0.18672178, 0.63783692, 0.31677667, 0.29435621,
       0.89058709, 0.30748865, 0.01160985, 0.39075039, 0.02411658,
       0.74139447, 0.62009745, 0.19514914, 0.00388252, 0.80978318,
       0.55444888, 0.53091377, 0.11641803, 0.98986983, 0.10012348])
```

In [4]:

```
n=int(input("satisfied customrers:"))
```

```
satisfied customrers:120
```

In [5]:

```
samplesiz=200
samplemean=x.mean()
samplemean
```

Out[5]:

Out[6]:

```
0.4561935061209889
```

In [7]:

```
criticalval=stats.norm.ppf(q=0.90)
pop=x.std()
merror=criticalval*(pop/math.sqrt(samplesiz))
criticalval
```

Out[7]:

```
1.2815515655446004
```

In [8]:

```
pop
```

Out[8]:

```
0.29022832529200826
```

In [9]:

```
merror
```

Out[9]:

```
0.026300310967123614
```

In [11]:

```
confidence=(samplemean-merror,samplemean+merror)
confidence
```

Out[11]:

```
(0.4298931951538653, 0.4824938170881125)
```

In [ ]:

```

```

In [1]:

```
#PROGRAM3
import numpy as np
import pandas as pd
import math
```

In [2]:

```
data=pd.read_csv("kerala.csv")
data
```

Out[2]:

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL RAINFALL	FLOOI
0	KERALA	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	197.7	266.9	350.8	48.4	3248.6	YI
1	KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	491.6	358.4	158.3	121.5	3326.6	YI
2	KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.1	157.0	59.0	3271.2	YI
3	KERALA	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	222.7	328.1	33.9	3.3	3129.7	YI
4	KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	217.2	383.5	74.4	0.2	2741.6	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
113	KERALA	2014	4.6	10.3	17.9	95.7	251.0	454.4	677.8	733.9	298.8	355.5	99.5	47.2	3046.4	YI
114	KERALA	2015	3.1	5.8	50.1	214.1	201.8	563.6	406.0	252.2	292.9	308.1	223.6	79.4	2600.6	N
115	KERALA	2016	2.4	3.8	35.9	143.0	186.4	522.2	412.3	325.5	173.2	225.9	125.4	23.6	2176.6	N
116	KERALA	2017	1.9	6.8	8.9	43.6	173.5	498.5	319.6	531.8	209.5	192.4	92.5	38.1	2117.1	N
117	KERALA	2018	29.1	52.1	48.6	116.4	183.8	625.4	1048.5	1398.9	423.6	356.1	125.4	65.1	4473.0	YI

118 rows × 16 columns

In [11]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 118 entries, 0 to 117
Data columns (total 22 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   SUBDIVISION       118 non-null    object 
 1   YEAR              118 non-null    int64  
 2   JAN               118 non-null    float64
 3   FEB               118 non-null    float64
 4   MAR               118 non-null    float64
 5   APR               118 non-null    float64
 6   MAY               118 non-null    float64
 7   JUN               118 non-null    float64
 8   JUL               118 non-null    float64
 9   AUG               118 non-null    float64
 10  SEP               118 non-null    float64
 11  OCT               118 non-null    float64
 12  NOV               118 non-null    float64
 13  DEC               118 non-null    float64
 14   ANNUAL RAINFALL  118 non-null    float64
 15  FLOODS            118 non-null    object 
 16  JUN_GT_500        118 non-null    bool   
 17  JUL_GT_500        118 non-null    bool   
 18  FLOODJUNE         118 non-null    bool   
 19  FLOODJUL          118 non-null    bool   
 20  JUN>500           118 non-null    bool   
 21  JUL>500           118 non-null    bool
 21  JUL>500               118 non-null    bool
dtypes: bool(6), float64(13), int64(1), object(2)
memory usage: 15.6+ KB
```

In [20]:

```
data["JUNGRT500"]=data["JUN"]>500
data["JULGRT500"]=data["JUL"]>500
data
```

Out[20]:

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	...	ANNUAL RAINFALL	FLOODS	JUN_GT_500	JUL_
0	KERALA	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	...	3248.6	YES	True	
1	KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	...	3326.6	YES	False	
2	KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	...	3271.2	YES	True	
3	KERALA	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	...	3129.7	YES	True	
4	KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	...	2741.6	NO	True	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
113	KERALA	2014	4.6	10.3	17.9	95.7	251.0	454.4	677.8	733.9	...	3046.4	YES	False	
114	KERALA	2015	3.1	5.8	50.1	214.1	201.8	563.6	406.0	252.2	...	2600.6	NO	True	
115	KERALA	2016	2.4	3.8	35.9	143.0	186.4	522.2	412.3	325.5	...	2176.6	NO	True	
116	KERALA	2017	1.9	6.8	8.9	43.6	173.5	498.5	319.6	531.8	...	2117.1	NO	False	
117	KERALA	2018	29.1	52.1	48.6	116.4	183.8	625.4	1048.5	1398.9	...	4473.0	YES	True	

118 rows × 24 columns

In [17]:

```
data["FLOODJUNE"]=data["JUN"]>500
data["FLOODJUL"]=data["JUL"]>500
data
```

Out[17]:

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	...	NOV	DEC	ANNUAL RAINFALL	FLOODS	JUN_
0	KERALA	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	...	350.8	48.4	3248.6	YES	
1	KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	...	158.3	121.5	3326.6	YES	
2	KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	...	157.0	59.0	3271.2	YES	
3	KERALA	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	...	33.9	3.3	3129.7	YES	
4	KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	...	74.4	0.2	2741.6	NO	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
113	KERALA	2014	4.6	10.3	17.9	95.7	251.0	454.4	677.8	733.9	...	99.5	47.2	3046.4	YES	
114	KERALA	2015	3.1	5.8	50.1	214.1	201.8	563.6	406.0	252.2	...	223.6	79.4	2600.6	NO	
115	KERALA	2016	2.4	3.8	35.9	143.0	186.4	522.2	412.3	325.5	...	125.4	23.6	2176.6	NO	
116	KERALA	2017	1.9	6.8	8.9	43.6	173.5	498.5	319.6	531.8	...	92.5	38.1	2117.1	NO	
117	KERALA	2018	29.1	52.1	48.6	116.4	183.8	625.4	1048.5	1398.9	...	125.4	65.1	4473.0	YES	

118 rows × 22 columns

In [21]:

```
newdata=data[["FLOODS","YEAR"]].copy()
```

```
newdata["JUNGRT500"]=(data["JUN"]>500).astype(int)
newdata["JULGRT500"]=(data["JUL"]>500).astype(int)
newdata["FLOODS"]=(data["FLOODS"]=="YES").astype(int)
```

In [22]:

```
newdata.tail(10)
```

Out[22]:

	FLOODS	YEAR	JUNGRT500	JULGRT500
108	0	2009	0	1
109	1	2010	1	1
110	1	2011	1	1
111	0	2012	0	0
112	1	2013	1	1
113	1	2014	0	1
114	0	2015	1	0
115	0	2016	1	0
116	0	2017	0	0
117	1	2018	1	1

In [23]:

```
pd.crosstab(data.FLOODS,data.JUNGRT500)
```

Out[23]:

JUNGRT500	False	True
FLOODS		
NO	19	39
YES	6	54

In [24]:

```
proboffflood=60/(19+39+6+54)
probofhighraininjune=(39+54)/(6+54+19+39)
intsct=54/(6+54+39+19)
```

In [25]:

```
"PROBABILITY OF FLOODING="
proboffflood
```

Out[25]:

```
0.5084745762711864
```

In [26]:

```
"PROBABILITY OF HIGH RAIN IN JUNE="
probofhighraininjune
```

Out[26]:

```
0.788135593220339
```

In [27]:

```
"INTERSECTION OF HIGH RAIN AND FLOODING="
intsct
```

Out[27]:

```
0.4576271186440678
```

In [28]:

```
"POBABILITY OF HIGH RAIN AND FLOODING"
intsct/probofhighraininjune
```

Out[28]:

```
0.5806451612903226
```

In [29]:

```
pd.crosstab(data.FLOODS,data.JULGRT500)
```

Out[29]:

JULGRT500	False	True
FLOODS		
NO	19	39
YES	3	57

In [32]:

```
probofflood=60/(19+39+3+57)
probofhighraininjuly=(39+57)/(3+57+19+39)
intsct=57/(3+57+39+19)
```

In [33]:

```
"PROBABILITY OF FLOODING="
probofflood
```

Out[33]:

```
0.5084745762711864
```

In [34]:

```
"PROBABILITY OF HIGH RAIN IN JULY="
probofhighraininjuly
```

Out[34]:

```
0.8135593220338984
```

In [35]:

```
"INTERSECTION OF HIGH RAIN AND FLOODING="
intsct
```

Out[35]:

```
0.4830508474576271
```

In [36]:

```
"POBABILITY OF HIGH RAIN AND FLOODING"
intsct/probofhighraininjuly
```

Out[36]:

```
0.59375
```

In [ ]:

```

```

In [20]:

```
#PROGRAM4
import pandas as pd
import numpy as np
import math
from scipy import stats
from sklearn.datasets import load_wine
wine=load_wine()
wine
```

Out[20]:

```
{'data': array([[1.423e+01, 1.710e+00, 2.430e+00, ..., 1.040e+00, 3.920e+00,
         1.065e+03],
        [1.320e+01, 1.780e+00, 2.140e+00, ..., 1.050e+00, 3.400e+00,
         1.050e+03],
        [1.316e+01, 2.360e+00, 2.670e+00, ..., 1.030e+00, 3.170e+00,
         1.185e+03],
        ...,
        [1.327e+01, 4.280e+00, 2.260e+00, ..., 5.900e-01, 1.560e+00,
         8.350e+02],
        [1.317e+01, 2.590e+00, 2.370e+00, ..., 6.000e-01, 1.620e+00,
         8.400e+02],
        [1.413e+01, 4.100e+00, 2.740e+00, ..., 6.100e-01, 1.600e+00,
         5.600e+02]]),
 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2]),
 'frame': None,
 'target_names': array(['class_0', 'class_1', 'class_2'], dtype='<U7'),
 'DESCR': '.. _wine_dataset:\n\nWine recognition dataset\n------------------------\n\n**
Data Set Characteristics:**\n\n    :Number of Instances: 178 (50 in each of three classes
)\n    :Number of Attributes: 13 numeric, predictive attributes and the class\n    :Attri
bute Information:\n \t\t- Alcohol\n \t\t- Malic acid\n \t\t- Ash\n\t\t- Alcalinity of ash
\n \t\t- Magnesium\n\t\t- Total phenols\n \t\t- Flavanoids\n \t\t- Nonflavanoid phenols\n
\t\t- Proanthocyanins\n\t\t- Color intensity\n \t\t- Hue\n \t\t- OD280/OD315 of diluted w
ines\n \t\t- Proline\n\n    - class:\n            - class_0\n            - class_1\n
- class_2\n\t\t\t\n    :Summary Statistics:\n    \n    ============================= ==== =
==== ======= =====\n                                   Min   Max   Mean     SD\n    ====
========================= ==== ===== ======= =====\n    Alcohol:                      11.
0  14.8    13.0   0.8\n    Malic Acid:                    0.74  5.80    2.34  1.12\n    A
sh:                           1.36  3.23    2.36  0.27\n    Alcalinity of Ash:
10.6  30.0    19.5   3.3\n    Magnesium:                     70.0 162.0    99.7  14.3\n
Total Phenols:                0.98  3.88    2.29  0.63\n    Flavanoids:
0.34  5.08    2.03  1.00\n    Nonflavanoid Phenols:          0.13  0.66    0.36  0.12\n
Proanthocyanins:              0.41  3.58    1.59  0.57\n    Colour Intensity:
1.3  13.0     5.1   2.3\n    Hue:                           0.48  1.71    0.96  0.23\n
OD280/OD315 of diluted wines: 1.27  4.00    2.61  0.71\n    Proline:
278  1680     746   315\n    ============================= ==== ===== ======= =====\n\n
:Missing Attribute Values: None\n    :Class Distribution: class_0 (59), class_1 (71), cla
ss_2 (48)\n    :Creator: R.A. Fisher\n    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.n
asa.gov)\n    :Date: July, 1988\n\nThis is a copy of UCI ML Wine recognition datasets.\nh
ttps://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data\n\nThe data is th
e results of a chemical analysis of wines grown in the same\nregion in Italy by three dif
ferent cultivators. There are thirteen different\nmeasurements taken for different consti
tuents found in the three types of\nwine.\n\nOriginal Owners: \n\nForina, M. et al, PARVU
S - \nAn Extendible Package for Data Exploration, Classification and Correlation. \nInsti
tute of Pharmaceutical and Food Analysis and Technologies,\nVia Brigata Salerno, 16147 Ge
noa, Italy.\n\nCitation:\n\nLichman, M. (2013). UCI Machine Learning Repository\n[https:/
/archive.ics.uci.edu/ml]. Irvine, CA: University of California,\nSchool of Information an
d Computer Science. \n\n.. topic:: References\n\n  (1) S. Aeberhard, D. Coomans and O. de
Vel, \n  Comparison of Classifiers in High Dimensional Settings, \n  Tech. Rep. no. 92-02
  (1992), Dept. of Computer Science and Dept. of  \n  Mathematics and Statistics, James C
, (1992), Dept. of Computer Science and Dept. of  \n Mathematics and Statistics, James C
ook University of North Queensland. \n  (Also submitted to Technometrics). \n\n  The data
was used with many others for comparing various \n  classifiers. The classes are separabl
e, though only RDA \n  has achieved 100% correct classification. \n  (RDA : 100%, QDA 99.
4%, LDA 98.9%, 1NN 96.1% (z-transformed data)) \n  (All results using the leave-one-out t
echnique) \n\n  (2) S. Aeberhard, D. Coomans and O. de Vel, \n  "THE CLASSIFICATION PERFO
RMANCE OF RDA" \n  Tech. Rep. no. 92-01, (1992), Dept. of Computer Science and Dept. of \
n  Mathematics and Statistics, James Cook University of North Queensland. \n  (Also submi
tted to Journal of Chemometrics).\n',
 'feature_names': ['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline']}
```

In [9]:

```
data=pd.DataFrame(wine.data,columns=wine["feature_names"])
data
```

Out[9]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82
...	...	...	...	...	...	...	...	...	...
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52	1.06
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43	1.41
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43	1.35
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53	1.46
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56	1.35

178 rows × 13 columns

In [13]:

```
newdata=data["alcohol"]
newdata
```

Out[13]:

```
0      14.23
1      13.20
2      13.16
3      14.37
4      13.24
       ...  
173    13.71
174    13.40
175    13.27
176    13.17
177    14.13
Name: alcohol, Length: 178, dtype: float64
```

In [15]:

```
samplesiz=50
sample=newdata.sample(n=50,random_state=100)
sample
```

Out[15]:

```
88     11.64
159    13.48
11     14.12
74     11.96
158    14.34
149    13.08
99     12.29
96     11.81
90     12.08
95     12.47
134    12.51
65     12.37
171    12.77
165    13.73
169    13.40
15     13.63
145    13.16
7      14.06
77     11.84
41     13.41
150    13.50
32     13.68
118    12.77
92     12.69
40     13.56
1      13.20
75     11.66
114    12.08
64     12.17
163    12.96
147    12.87
69     12.21
26     13.39
97     12.29
146    13.88
151    12.79
111    12.52
119    12.00
170    12.20
142    13.52
29     14.02
152    13.11
136    12.25
167    12.82
46     14.38
174    13.40
177    14.13
139    12.84
20     14.06
31     13.58
Name: alcohol, dtype: float64
```

In [16]:

```
samplemean=sample.mean()
samplemean
```

Out[16]:

```
12.973600000000001
```

In [21]:

```
criticalvalue=stats.norm.ppf(q=0.90)
```

```
pop=sample.std()
print(criticalvalue,pop)
```

```
1.2815515655446004 0.7686663775657162
```

In [23]:

```
merror=criticalvalue*(pop/math.sqrt(samplesiz))
merror
```

Out[23]:

```
0.13931214149832302
```

In [24]:

```
confidenceitrvl=(samplemean-merror,samplemean+merror)
confidenceitrvl
```

Out[24]:

```
(12.834287858501678, 13.112912141498324)
```

In [ ]:

```

```