DataToolkit

August 10, 2024

[49]:
```
import pandas as pd
import numpy as np
```

Question 1

[3]:
```
# 1st Method
np.eye(2)
```

[3]:
```
array([[1., 0.],
       [0., 1.]])
```

[3]:
```
# 2nd method
np.identity(2)
```

[3]:
```
array([[1., 0.],
       [0., 1.]])
```

[5]:
```
# 3rd Method
identity_matrix = [[1 if i == j else 0 for j in range(2)] for i in range(2)]
print(identity_matrix)
```

```
[[1, 0], [0, 1]]
```

Question 2

[12]:
```
res = np.linspace(1,10,100).reshape(10,10)
```

Question 3

[13]:
```
# np.array always creates a new array, np.asarray avoids making a copy if␣
 ↪possible, and np.asanyarray is like asarray but preserves subclasses.
# Shallow Copy replicates the outer object but keeps references to inner␣
 ↪objects, while Deep Copy duplicates everything, creating a completely␣
 ↪independent copy.
```

[14]:
```
arr = np.array([1, 2, 3])
```

[15]:
```
lst = [1, 2, 3]
arr1 = np.asarray(lst)  # No copy is made
```

[16]:
```python
mat = np.matrix([[1, 2], [3, 4]])
arr = np.asanyarray(mat)  # Keeps the matrix subclass
```

[17]:
```python
import copy
original = [[1, 2, 3], [4, 5, 6]]
shallow_copy = copy.copy(original)
shallow_copy[0][0] = 10  # This will change `original` as well
```

[19]:
```python
import copy
original = [[1, 2, 3], [4, 5, 6]]
deep_copy = copy.deepcopy(original)
deep_copy[0][0] = 10  # This will NOT change `original`
```

Question 4

[24]:
```python
matrix = np.random.uniform(5, 20, (3, 3))
```

[26]:
```python
print(matrix)
```

```
[[16.01334847 18.81406809 10.48418529]
 [10.84828341  8.34293861 16.12302257]
 [17.18508209  9.56653152 10.16766299]]
```

[28]:
```python
final = np.round(matrix,2)
final
```

[28]:
```
array([[16.01, 18.81, 10.48],
       [10.85,  8.34, 16.12],
       [17.19,  9.57, 10.17]])
```

Question 5

[32]:
```python
res = np.random.randint(1,10,30).reshape(5,6)
print(res)
```

```
[[5 1 2 8 8 7]
 [2 9 7 2 5 8]
 [5 7 9 2 8 4]
 [4 4 2 9 2 9]
 [6 1 9 6 3 5]]
```

[48]:
```python
# part a
even = []
for row in res :
    for val in row :
        if val%2==0:
            even.append(val)
```

```python
print(even)
```

```
[2, 8, 8, 2, 2, 8, 2, 8, 4, 4, 4, 2, 2, 6, 6]
```

[50]:

```python
odd = []
for row in res :
    for val in row :
        if val%2==1:
            odd.append(val)

print(odd)
```

```
[5, 1, 7, 9, 7, 5, 5, 7, 9, 9, 9, 1, 9, 3, 5]
```

Question 6

[55]:

```python
arr = np.random.randint(1,10,27).reshape(3,3,3)
```

[56]:

```python
print(arr)
```

```
[[[3 7 5]
  [9 7 2]
  [4 9 2]]

 [[9 5 2]
  [4 7 8]
  [3 1 4]]

 [[2 8 4]
  [2 6 6]
  [4 6 2]]]
```

[75]:

```python
max_indices = np.argmax(arr, axis=2)
max_indices
```

[75]:

```
array([[1, 0, 1],
       [0, 2, 2],
       [1, 1, 1]])
```

[76]:

```python
arr2 = arr = np.random.randint(1,10,27).reshape(3,3,3)
```

[ ]:

```python

```

Question 7

[4]:

```python
df = pd.read_csv('People_Data.csv')
```

[5]:

```python
df
```

[5]:

```
     Index          User Id First Name Last Name  Gender  \
0        1  8717bbf45cCDbEe     Shelia   Mahoney    Male
1        2  3d5AD30A4cD38ed         Jo    Rivers  Female
2        3  810Ce0F276Badec     Sheryl    Lowery  Female
3        4  BF2a889C00f0cE1    Whitney    Hooper    Male
4        5  9afFEafAe1CBBB9    Lindsey      Rice  Female
..     …              …        …       …     …
995    996  fedF4c7Fd9e7cFa       Kurt    Bryant  Female
996    997  ECddaFEDdEc4FAB      Donna     Barry  Female
997    998  2adde51d8B8979E      Cathy  Mckinney  Female
998    999  Fb2FE369D1E171A   Jermaine    Phelps    Male
999   1000  8b756f6231DDC6e        Lee      Tran  Female

                             Email                  Phone Date of birth  \
0              pwarner@example.org           857.139.8239    27-01-2014
1    fergusonkatherine@example.net                    NaN    26-07-1931
2              fhoward@example.org          (599)782-0605    25-11-2013
3            zjohnston@example.com                    NaN    17-11-2012
4                 elin@example.net     (390)417-1635x3010    15-04-1923
..                             …                    …            …
995         lyonsdaisy@example.net           021.775.2933    05-01-1959
996        dariusbryan@example.com   001-149-710-7799x721    06-10-2001
997         georgechan@example.org  +1-750-774-4128x33265    13-05-1918
998            wanda04@example.net          (915)292-2254    31-08-1971
999        deannablack@example.org     079.752.5424x67259    24-01-1947

                           Job Title  Salary
0                  Probation officer   90000
1                             Dancer   80000
2                               Copy   50000
3           Counselling psychologist   65000
4                Biomedical engineer  100000
..                               …      …
995                Personnel officer   90000
996          Education administrator   50000
997  Commercial/residential surveyor   60000
998                 Ambulance person  100000
999       Nurse, learning disability   90000

[1000 rows x 10 columns]
```

[6]:

```python
def valid_number(number):
    val = str(number)
    res = ""
    for i in val :
        if i.isdigit():
            res += i
    return res[:10]
```

[7]:
```
df['Phone'] = df['Phone'].dropna()
```

[26]:
```
df['Phone'] = df['Phone'].apply(valid_number)
```

[27]:
```
df
```

[27]:
```
     Index          User Id First Name Last Name  Gender  \
0        1  8717bbf45cCDbEe     Shelia   Mahoney    Male
1        2  3d5AD30A4cD38ed         Jo    Rivers  Female
2        3  810Ce0F276Badec     Sheryl    Lowery  Female
3        4  BF2a889C00f0cE1    Whitney    Hooper    Male
4        5  9afFEafAe1CBBB9    Lindsey      Rice  Female
..     …              …        …       …     …
995     996  fedF4c7Fd9e7cFa       Kurt    Bryant  Female
996     997  ECddaFEDdEc4FAB      Donna     Barry  Female
997     998  2adde51d8B8979E      Cathy  Mckinney  Female
998     999  Fb2FE369D1E171A   Jermaine    Phelps    Male
999    1000  8b756f6231DDC6e        Lee      Tran  Female

                             Email       Phone Date of birth  \
0              pwarner@example.org  8571398239    27-01-2014
1    fergusonkatherine@example.net           0    26-07-1931
2              fhoward@example.org  5997820605    25-11-2013
3            zjohnston@example.com           0    17-11-2012
4                 elin@example.net  3904171635    15-04-1923
..                             …         …           …
995         lyonsdaisy@example.net   217752933    05-01-1959
996        dariusbryan@example.com    11497107    06-10-2001
997         georgechan@example.org  1750774412    13-05-1918
998            wanda04@example.net  9152922254    31-08-1971
999        deannablack@example.org   797525424    24-01-1947

                           Job Title  Salary
0                  Probation officer   90000
1                             Dancer   80000
2                               Copy   50000
3           Counselling psychologist   65000
4                Biomedical engineer  100000
..                               …     …
995                Personnel officer   90000
996          Education administrator   50000
997  Commercial/residential surveyor   60000
998                 Ambulance person  100000
999       Nurse, learning disability   90000
```

```
[1000 rows x 10 columns]
```

[18]:

```
df['Phone'] = df['Phone'].replace('',0).astype(int)
```

[20]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Index          1000 non-null   int64
 1   User Id        1000 non-null   object
 2   First Name     1000 non-null   object
 3   Last Name      1000 non-null   object
 4   Gender         1000 non-null   object
 5   Email          1000 non-null   object
 6   Phone          1000 non-null   int64
 7   Date of birth  1000 non-null   object
 8   Job Title      1000 non-null   object
 9   Salary         1000 non-null   int64
dtypes: int64(3), object(7)
memory usage: 78.2+ KB
```

Question 8

[31]:

```
df
```

[31]:

```
     Index          User Id First Name Last Name  Gender  \
0        1  8717bbf45cCDbEe     Shelia   Mahoney    Male
1        2  3d5AD30A4cD38ed         Jo    Rivers  Female
2        3  810Ce0F276Badec     Sheryl    Lowery  Female
3        4  BF2a889C00f0cE1    Whitney    Hooper    Male
4        5  9afFEafAe1CBBB9    Lindsey      Rice  Female
..     …              …        …       …     …
995    996  fedF4c7Fd9e7cFa       Kurt    Bryant  Female
996    997  ECddaFEDdEc4FAB      Donna     Barry  Female
997    998  2adde51d8B8979E      Cathy  Mckinney  Female
998    999  Fb2FE369D1E171A   Jermaine    Phelps    Male
999   1000  8b756f6231DDC6e        Lee      Tran  Female

                             Email       Phone Date of birth  \
0              pwarner@example.org  8571398239    27-01-2014
1    fergusonkatherine@example.net           0    26-07-1931
2              fhoward@example.org  5997820605    25-11-2013
3            zjohnston@example.com           0    17-11-2012
4                 elin@example.net  3904171635    15-04-1923
..                             …         …           …
```

```
995         lyonsdaisy@example.net   217752933    05-01-1959
996        dariusbryan@example.com    11497107    06-10-2001
997         georgechan@example.org  1750774412    13-05-1918
998            wanda04@example.net  9152922254    31-08-1971
999        deannablack@example.org   797525424    24-01-1947

                           Job Title  Salary
0                  Probation officer   90000
1                             Dancer   80000
2                               Copy   50000
3           Counselling psychologist   65000
4                Biomedical engineer  100000
..                               …       …
995                Personnel officer   90000
996          Education administrator   50000
997  Commercial/residential surveyor   60000
998                 Ambulance person  100000
999       Nurse, learning disability   90000

[1000 rows x 10 columns]
```

[32]:
```
df[50:]
```

[32]:
```
     Index          User Id First Name Last Name  Gender  \
50      51  CccE5DAb6E288e5         Jo    Zavala    Male
51      52  DfBDc3621D4bcec     Joshua     Carey  Female
52      53  f55b0A249f5E44D     Rickey     Hobbs  Female
53      54  Ed71DcfaBFd0beE      Robyn    Reilly    Male
54      55  FDaFD0c3f5387EC  Christina    Conrad    Male
..     …              …         …       …     …
995    996  fedF4c7Fd9e7cFa       Kurt    Bryant  Female
996    997  ECddaFEDdEc4FAB      Donna     Barry  Female
997    998  2adde51d8B8979E      Cathy  Mckinney  Female
998    999  Fb2FE369D1E171A   Jermaine    Phelps    Male
999   1000  8b756f6231DDC6e        Lee      Tran  Female

                          Email       Phone Date of birth  \
50          pamela64@example.net    18594489    23-11-1992
51     dianashepherd@example.net    12747398    07-01-1915
52     ingramtiffany@example.org  2411799509    01-07-1910
53    carriecrawford@example.org  2077978345    27-07-1982
54    fuentesclaudia@example.net    15990427    06-01-1998
..                          …         …           …
995      lyonsdaisy@example.net   217752933    05-01-1959
996     dariusbryan@example.com    11497107    06-10-2001
997      georgechan@example.org  1750774412    13-05-1918
998         wanda04@example.net  9152922254    31-08-1971
```

```
999     deannablack@example.org   797525424    24-01-1947

                           Job Title  Salary
50                      Nurse, adult   80000
51               Seismic interpreter   70000
52                         Barrister   60000
53              Engineer, structural  100000
54                   Producer, radio   50000
..                               …       …
995                Personnel officer   90000
996          Education administrator   50000
997  Commercial/residential surveyor   60000
998                 Ambulance person  100000
999       Nurse, learning disability   90000

[950 rows x 10 columns]
```

```
[34]: df[['Last Name' , 'Gender','Email','Phone','Salary']]
```

```
[34]:     Last Name  Gender                          Email       Phone  Salary
      0     Mahoney    Male            pwarner@example.org  8571398239   90000
      1      Rivers  Female  fergusonkatherine@example.net           0   80000
      2      Lowery  Female            fhoward@example.org  5997820605   50000
      3      Hooper    Male          zjohnston@example.com           0   65000
      4        Rice  Female               elin@example.net  3904171635  100000
      ..        …      …                            …         …      …
      995    Bryant  Female         lyonsdaisy@example.net   217752933   90000
      996     Barry  Female        dariusbryan@example.com    11497107   50000
      997  Mckinney  Female         georgechan@example.org  1750774412   60000
      998    Phelps    Male            wanda04@example.net  9152922254  100000
      999      Tran  Female        deannablack@example.org   797525424   90000

      [1000 rows x 5 columns]
```

```
[35]: df.head(10)
```

```
[35]:    Index          User Id First Name Last Name  Gender  \
      0      1  8717bbf45cCDbEe     Shelia   Mahoney    Male
      1      2  3d5AD30A4cD38ed         Jo    Rivers  Female
      2      3  810Ce0F276Badec     Sheryl    Lowery  Female
      3      4  BF2a889C00f0cE1    Whitney    Hooper    Male
      4      5  9afFEafAe1CBBB9    Lindsey      Rice  Female
      5      6  aF75e6dDEBC5b66     Sherry  Caldwell    Male
      6      7  efeb05c7Cc94EA3     Ernest   Hoffman    Male
      7      8  fb1BF3FED57E9d7      Doris  Andersen    Male
      8      9  421fAB9a3b98F30     Cheryl      Mays    Male
      9     10  4A42Fe10dB717CB      Harry  Mitchell    Male
```

```
                          Email       Phone Date of birth  \
0             pwarner@example.org  8571398239    27-01-2014
1  fergusonkatherine@example.net           0    26-07-1931
2             fhoward@example.org  5997820605    25-11-2013
3           zjohnston@example.com           0    17-11-2012
4                elin@example.net  3904171635    15-04-1923
5           kaitlin13@example.net  8537800927    06-08-1917
6          jeffharvey@example.com   936557480    22-12-1984
7            alicia33@example.org  4709522945    02-12-2016
8               jake50@example.com   138204758    16-12-2012
9      lanechristina@example.net  5609035068    29-06-1953

                   Job Title  Salary
0          Probation officer   90000
1                     Dancer   80000
2                       Copy   50000
3   Counselling psychologist   65000
4        Biomedical engineer  100000
5  Higher education lecturer   50000
6             Health visitor   60000
7                 Air broker   65000
8       Designer, multimedia   50000
9  Insurance account manager   50000
```

[36]:
```
s = df['Salary']
```

[38]:
```
s.tail(5)
```

[38]:
```
995     90000
996     50000
997     60000
998    100000
999     90000
Name: Salary, dtype: int64
```

Question 9

[53]:
```
df[(df['Last Name'] == 'Duke') & (df['Gender'] == 'Female') & (df['Salary'] <␣
 ↪85000)]
```

[53]:
```
     Index          User Id First Name Last Name  Gender  \
45      46  99A502C175C4EBd     Olivia      Duke  Female
210    211  DF17975CC0a0373    Katrina      Duke  Female
457    458  dcE1B7DE83c1076      Traci      Duke  Female
729    730  c9b482D7aa3e682     Lonnie      Duke  Female
```

```
                           Email       Phone Date of birth        Job Title  \
45         diana26@example.net    13664758    13-10-1934          Dentist
210        robin78@example.com  7404340212    21-09-1935  Producer, radio
457  perryhoffman@example.org  1903596099    11-02-1997        Herbalist
729   kevinkramer@example.net  9826926257    12-05-2015     Nurse, adult

     Salary
45    60000
210   50000
457   50000
729   70000
```

Question 10

```
[57]: s = pd.Series(np.random.randint(1,6,35))
```

```
[68]: df = pd.DataFrame(s.values.reshape(7,5))
```

```
[70]: df
```

```
[70]:    0  1  2  3  4
      0  5  5  1  4  1
      1  5  4  1  5  1
      2  4  2  1  2  1
      3  3  3  1  3  4
      4  4  4  2  2  5
      5  2  3  2  2  1
      6  1  3  3  1  2
```

Question 11

```
[80]: s1 = pd.Series(np.random.randint(10,50,50))
```

```
[81]: s2 = pd.Series(np.random.randint(100,1000,50))
```

```
[89]: s1 = s1.reset_index()
```

```
[90]: s2 = s2.reset_index()
```

```
[92]: df = pd.concat([s1,s2])
```

```
[93]: df
```

```
[93]:     index    0
      0       0   44
      1       1   17
      2       2   42
      3       3   23
```

```
4        4   22
..     …   …
45      45  484
46      46  585
47      47  603
48      48  734
49      49  390

[100 rows x 2 columns]
```

[96]:
```
df = df.rename(columns = {'index':'col1',0 : 'col2'})
```

[98]:
```
df
```

[98]:
```
    col1  col2
0      0    44
1      1    17
2      2    42
3      3    23
4      4    22
..   …   …
45    45   484
46    46   585
47    47   603
48    48   734
49    49   390

[100 rows x 2 columns]
```

Question 12

[99]:
```
temp = pd.read_csv('People_Data.csv')
```

[102]:
```
temp = temp.drop(columns={'Phone','Email','Date of birth'})
```

[105]:
```
temp = temp.dropna()
```

[107]:
```
temp.isnull().sum()
```

[107]:
```
Index          0
User Id        0
First Name     0
Last Name      0
Gender         0
Job Title      0
Salary         0
dtype: int64
```

```
[108]: temp
```

```
[108]:      Index          User Id First Name Last Name  Gender  \
       0        1  8717bbf45cCDbEe     Shelia   Mahoney    Male
       1        2  3d5AD30A4cD38ed         Jo    Rivers  Female
       2        3  810Ce0F276Badec     Sheryl    Lowery  Female
       3        4  BF2a889C00f0cE1    Whitney    Hooper    Male
       4        5  9afFEafAe1CBBB9    Lindsey      Rice  Female
       ..     …              …        …       …     …
       995    996  fedF4c7Fd9e7cFa       Kurt    Bryant  Female
       996    997  ECddaFEDdEc4FAB      Donna     Barry  Female
       997    998  2adde51d8B8979E      Cathy  Mckinney  Female
       998    999  Fb2FE369D1E171A   Jermaine    Phelps    Male
       999   1000  8b756f6231DDC6e        Lee      Tran  Female

                                Job Title  Salary
       0                Probation officer   90000
       1                           Dancer   80000
       2                             Copy   50000
       3          Counselling psychologist   65000
       4               Biomedical engineer  100000
       ..                              …     …
       995              Personnel officer   90000
       996          Education administrator   50000
       997   Commercial/residential surveyor   60000
       998                 Ambulance person  100000
       999        Nurse, learning disability   90000

       [1000 rows x 7 columns]
```

Question 13

```
[110]: x = np.random.rand(100)
       x
```

```
[110]: array([0.68283223, 0.7742662 , 0.93042642, 0.33537671, 0.15106733,
              0.53811948, 0.31860737, 0.69718363, 0.17563345, 0.28320342,
              0.4784762 , 0.00574081, 0.6009724 , 0.62761752, 0.89344351,
              0.37207842, 0.28735862, 0.92858348, 0.72118536, 0.03645687,
              0.09472015, 0.06675247, 0.15475199, 0.80234411, 0.81596194,
              0.85199368, 0.83176122, 0.34619308, 0.41424557, 0.71677558,
              0.84527571, 0.36663083, 0.48886355, 0.98188895, 0.42057512,
              0.40128584, 0.73896538, 0.78708205, 0.18783617, 0.77054402,
              0.27268818, 0.09842655, 0.09394982, 0.03743849, 0.25636771,
              0.23556302, 0.83188251, 0.85711207, 0.21482643, 0.90863407,
              0.22257415, 0.78368257, 0.63429631, 0.53332433, 0.63734655,
              0.70534306, 0.20217152, 0.77570713, 0.4978114 , 0.49854905,
              0.45617168, 0.84128331, 0.44114193, 0.70633994, 0.90475016,
              0.45547705, 0.78257876, 0.51244434, 0.74358936, 0.7694398 ,
              0.20368766, 0.9683631 , 0.0701125 , 0.54532232, 0.95251461,
              0.37154376, 0.9136258 , 0.18063711, 0.01911665, 0.69325282,
              0.66511956, 0.45108795, 0.28644158, 0.46837888, 0.29683414,
              0.66913354, 0.336633  , 0.07785765, 0.41573168, 0.28104705,
              0.77836306, 0.62667832, 0.62807129, 0.88232958, 0.51800141,
              0.35452214, 0.35786747, 0.67590734, 0.19490344, 0.27139726])
```

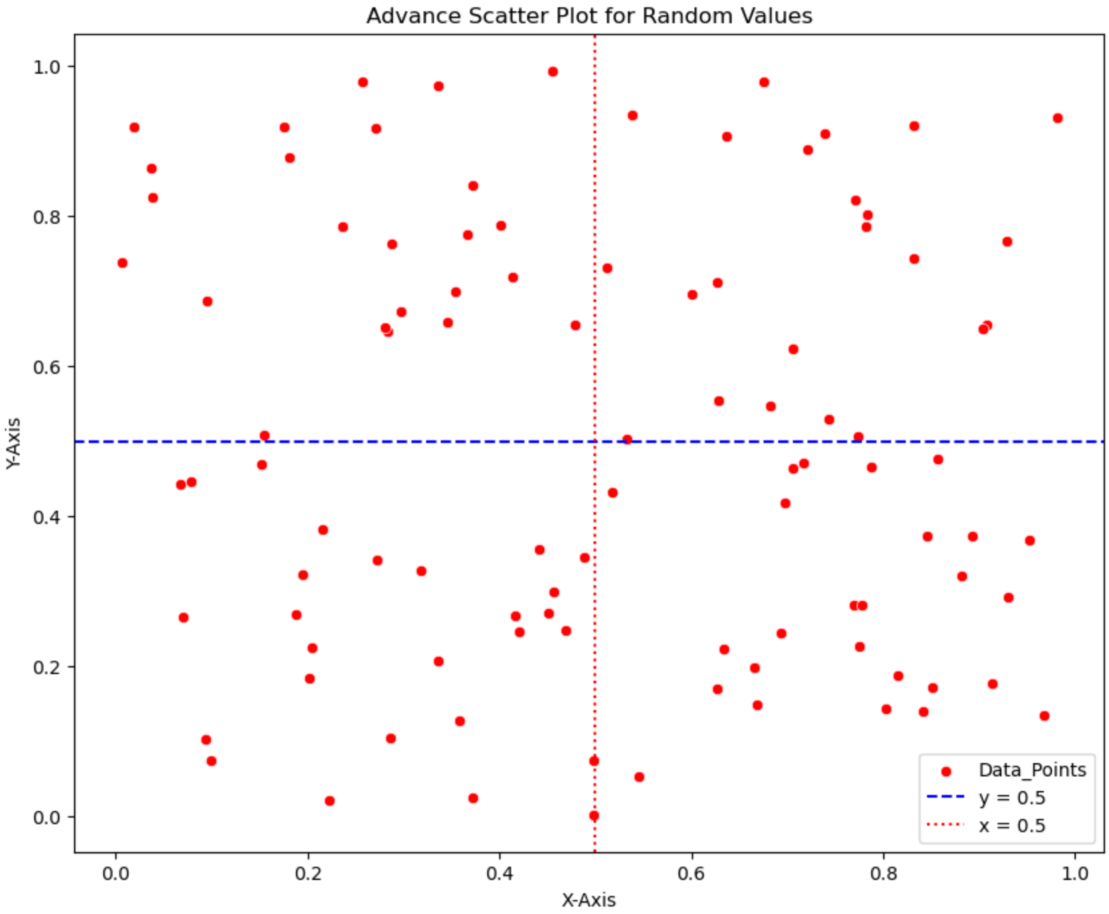
[111]:
```
y = np.random.rand(100)
y
```

[111]:
```
array([0.54730879, 0.50702882, 0.29220412, 0.20716146, 0.4696364 ,
       0.9344318 , 0.32702859, 0.41721708, 0.91785411, 0.64593846,
       0.65573738, 0.7384585 , 0.69537817, 0.71082097, 0.37329974,
       0.84024767, 0.76263914, 0.76596845, 0.88777102, 0.86286782,
       0.68651066, 0.4427033 , 0.50777339, 0.14334222, 0.18865536,
       0.17158105, 0.74329589, 0.65906989, 0.71875204, 0.47093779,
       0.37324842, 0.77518002, 0.34516024, 0.930242  , 0.24601382,
       0.78705294, 0.90985675, 0.46510196, 0.26967486, 0.82093614,
       0.34223977, 0.07454383, 0.10403379, 0.82495646, 0.97884391,
       0.78541929, 0.91974956, 0.47591218, 0.38347293, 0.65525943,
       0.02236711, 0.80250182, 0.22364381, 0.50320197, 0.90573638,
       0.46361371, 0.18413392, 0.22742863, 0.0751404 , 0.00291441,
       0.29885058, 0.14103675, 0.35525604, 0.62363092, 0.64957502,
       0.99327377, 0.78607194, 0.73132836, 0.53007764, 0.28101855,
       0.22518238, 0.13491665, 0.26553488, 0.05380922, 0.36816321,
       0.02510106, 0.17701549, 0.87695465, 0.91797224, 0.2444551 ,
       0.19788417, 0.27132456, 0.10487962, 0.24878845, 0.67240083,
       0.14945953, 0.97333815, 0.44702851, 0.26791115, 0.65166266,
       0.28096401, 0.17081332, 0.5547249 , 0.32045536, 0.43268031,
       0.69890367, 0.12744143, 0.97799694, 0.32330735, 0.91600505])
```

[2]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

[126]:
```
figure = plt.figure(figsize=(10,8))
sns.scatterplot(x=x,y=y,color='red',marker='o',label='Data_Points')
plt.xlabel('X-Axis')
plt.ylabel('Y-Axis')
plt.axhline(y=0.5, color='blue', linestyle='--', label='y = 0.5')
plt.axvline(x=0.5, color='red', linestyle=':', label='x = 0.5')
plt.title('Advance Scatter Plot for Random Values')
plt.legend()
```

[126]:
```
<matplotlib.legend.Legend at 0x79ceee316b60>
```



Question 14

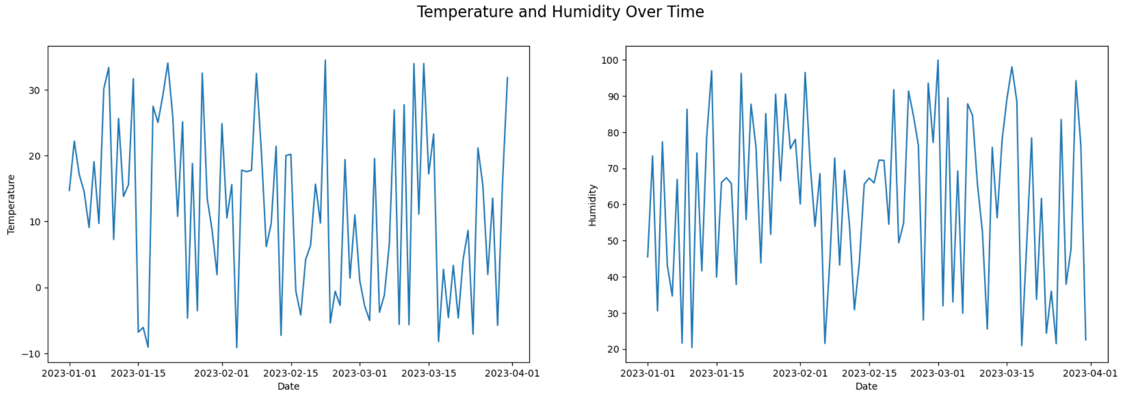
```
[20]: date_range = pd.date_range(start='2023-01-01', end='2023-03-31', freq='D')

      np.random.seed(0)
      temperature = np.random.uniform(low=-10, high=35, size=len(date_range))
      humidity = np.random.uniform(low=20, high=100, size=len(date_range))

      df = pd.DataFrame({
          'Date': date_range,
          'Temperature': temperature,
          'Humidity': humidity
      })
```

```
[21]: df
```

```
[21]:          Date  Temperature   Humidity
      0  2023-01-01    14.696608  45.485516
      1  2023-01-02    22.183521  73.392830
```

```
2  2023-01-03    17.124352  30.543829
3  2023-01-04    14.519743  77.306176
4  2023-01-05     9.064466  43.152487
..         …           …          …
85 2023-03-27     1.942527  37.913975
86 2023-03-28    13.546162  47.628134
87 2023-03-29    -5.772677  94.246503
88 2023-03-30    15.917592  76.353152
89 2023-03-31    31.818329  22.547114

[90 rows x 3 columns]
```

[28]:
```python
fig,ax = plt.subplots(1,2,figsize=(20,6))
ax[0].plot(df['Date'],df['Temperature'])
ax[0].set_xlabel('Date')
ax[0].set_ylabel('Temperature')
ax[1].plot(df['Date'] , df['Humidity'])
ax[1].set_xlabel('Date')
ax[1].set_ylabel('Humidity')
fig.suptitle('Temperature and Humidity Over Time', fontsize=16)
plt.show()
```



Question 15 and Question 16

[29]:
```python
data = np.random.randn(1000)
print(data)
```

```
[ 3.96006713e-01 -1.09306151e+00 -1.49125759e+00  4.39391701e-01
  1.66673495e-01  6.35031437e-01  2.38314477e+00  9.44479487e-01
 -9.12822225e-01  1.11701629e+00 -1.31590741e+00 -4.61584605e-01
 -6.82416053e-02  1.71334272e+00 -7.44754822e-01 -8.26438539e-01
 -9.84525244e-02 -6.63478286e-01  1.12663592e+00 -1.07993151e+00
 -1.14746865e+00 -4.37820045e-01 -4.98032451e-01  1.92953205e+00
  9.49420807e-01  8.75512414e-02 -1.22543552e+00  8.44362976e-01
```

```
-1.00021535e+00 -1.54477110e+00  1.18802979e+00  3.16942612e-01
 9.20858824e-01  3.18727653e-01  8.56830612e-01 -6.51025593e-01
-1.03424284e+00  6.81594518e-01 -8.03409664e-01 -6.89549778e-01
-4.55532504e-01  1.74791590e-02 -3.53993911e-01 -1.37495129e+00
-6.43618403e-01 -2.22340315e+00  6.25231451e-01 -1.60205766e+00
-1.10438334e+00  5.21650793e-02 -7.39562996e-01  1.54301460e+00
-1.29285691e+00  2.67050869e-01 -3.92828182e-02 -1.16809350e+00
 5.23276661e-01 -1.71546331e-01  7.71790551e-01  8.23504154e-01
 2.16323595e+00  1.33652795e+00 -3.69181838e-01 -2.39379178e-01
 1.09965960e+00  6.55263731e-01  6.40131526e-01 -1.61695604e+00
-2.43261244e-02 -7.38030909e-01  2.79924599e-01 -9.81503896e-02
 9.10178908e-01  3.17218215e-01  7.86327962e-01 -4.66419097e-01
-9.44446256e-01 -4.10049693e-01 -1.70204139e-02  3.79151736e-01
 2.25930895e+00 -4.22571517e-02 -9.55945000e-01 -3.45981776e-01
-4.63595975e-01  4.81481474e-01 -1.54079701e+00  6.32619942e-02
 1.56506538e-01  2.32181036e-01 -5.97316069e-01 -2.37921730e-01
-1.42406091e+00 -4.93319883e-01 -5.42861476e-01  4.16050046e-01
-1.15618243e+00  7.81198102e-01  1.49448454e+00 -2.06998503e+00
 4.26258731e-01  6.76908035e-01 -6.37437026e-01 -3.97271814e-01
-1.32880578e-01 -2.97790879e-01 -3.09012969e-01 -1.67600381e+00
 1.15233156e+00  1.07961859e+00 -8.13364259e-01 -1.46642433e+00
 5.21064876e-01 -5.75787970e-01  1.41953163e-01 -3.19328417e-01
 6.91538751e-01  6.94749144e-01 -7.25597378e-01 -1.38336396e+00
-1.58293840e+00  6.10379379e-01 -1.18885926e+00 -5.06816354e-01
-5.96314038e-01 -5.25672963e-02 -1.93627981e+00  1.88778597e-01
 5.23891024e-01  8.84220870e-02 -3.10886172e-01  9.74001663e-02
 3.99046346e-01 -2.77259276e+00  1.95591231e+00  3.90093323e-01
-6.52408582e-01 -3.90953375e-01  4.93741777e-01 -1.16103939e-01
-2.03068447e+00  2.06449286e+00 -1.10540657e-01  1.02017271e+00
-6.92049848e-01  1.53637705e+00  2.86343689e-01  6.08843834e-01
-1.04525337e+00  1.21114529e+00  6.89818165e-01  1.30184623e+00
-6.28087560e-01 -4.81027118e-01  2.30391670e+00 -1.06001582e+00
-1.35949701e-01  1.13689136e+00  9.77249677e-02  5.82953680e-01
-3.99449029e-01  3.70055888e-01 -1.30652685e+00  1.65813068e+00
-1.18164045e-01 -6.80178204e-01  6.66383082e-01 -4.60719787e-01
-1.33425847e+00 -1.34671751e+00  6.93773153e-01 -1.59573438e-01
-1.33701560e-01  1.07774381e+00 -1.12682581e+00 -7.30677753e-01
-3.84879809e-01  9.43515893e-02 -4.21714513e-02 -2.86887192e-01
-6.16264021e-02 -1.07305276e-01 -7.19604389e-01 -8.12992989e-01
 2.74516358e-01 -8.90915083e-01 -1.15735526e+00 -3.12292251e-01
-1.57667016e-01  2.25672350e+00 -7.04700276e-01  9.43260725e-01
 7.47188334e-01 -1.18894496e+00  7.73252977e-01 -1.18388064e+00
-2.65917224e+00  6.06319524e-01 -1.75589058e+00  4.50934462e-01
-6.84010898e-01  1.65955080e+00  1.06850940e+00 -4.53385804e-01
-6.87837611e-01 -1.21407740e+00 -4.40922632e-01 -2.80355495e-01
-3.64693544e-01  1.56703855e-01  5.78521498e-01  3.49654457e-01
-7.64143924e-01 -1.43779147e+00  1.36453185e+00 -6.89449185e-01
-6.52293600e-01 -5.21189312e-01 -1.84306955e+00 -4.77974004e-01
```

```
-4.79655814e-01  6.20358298e-01  6.98457149e-01  3.77088909e-03
 9.31848374e-01  3.39964984e-01 -1.56821116e-02  1.60928168e-01
-1.90653494e-01 -3.94849514e-01 -2.67733537e-01 -1.12801133e+00
 2.80441705e-01 -9.93123611e-01  8.41631264e-01 -2.49458580e-01
 4.94949817e-02  4.93836776e-01  6.43314465e-01 -1.57062341e+00
-2.06903676e-01  8.80178912e-01 -1.69810582e+00  3.87280475e-01
-2.25556423e+00 -1.02250684e+00  3.86305518e-02 -1.65671510e+00
-9.85510738e-01 -1.47183501e+00  1.64813493e+00  1.64227755e-01
 5.67290278e-01 -2.22675101e-01 -3.53431749e-01 -1.61647419e+00
-2.91837363e-01 -7.61492212e-01  8.57923924e-01  1.14110187e+00
 1.46657872e+00  8.52551939e-01 -5.98653937e-01 -1.11589699e+00
 7.66663182e-01  3.56292817e-01 -1.76853845e+00  3.55481793e-01
 8.14519822e-01  5.89255892e-02 -1.85053671e-01 -8.07648488e-01
-1.44653470e+00  8.00297949e-01 -3.09114445e-01 -2.33466662e-01
 1.73272119e+00  6.84501107e-01  3.70825001e-01  1.42061805e-01
 1.51999486e+00  1.71958931e+00  9.29505111e-01  5.82224591e-01
-2.09460307e+00  1.23721914e-01 -1.30106954e-01  9.39532294e-02
 9.43046087e-01 -2.73967717e+00 -5.69312053e-01  2.69904355e-01
-4.66845546e-01 -1.41690611e+00  8.68963487e-01  2.76871906e-01
-9.71104570e-01  3.14817205e-01  8.21585712e-01  5.29264630e-03
 8.00564803e-01  7.82601752e-02 -3.95228983e-01 -1.15942052e+00
-8.59307670e-02  1.94292938e-01  8.75832762e-01 -1.15107468e-01
 4.57415606e-01 -9.64612014e-01 -7.82629156e-01 -1.10389299e-01
-1.05462846e+00  8.20247837e-01  4.63130329e-01  2.79095764e-01
 3.38904125e-01  2.02104356e+00 -4.68864188e-01 -2.20144129e+00
 1.99300197e-01 -5.06035410e-02 -5.17519043e-01 -9.78829859e-01
-4.39189522e-01  1.81338429e-01 -5.02816701e-01  2.41245368e+00
-9.60504382e-01 -7.93117363e-01 -2.28862004e+00  2.51484415e-01
-2.01640663e+00 -5.39454633e-01 -2.75670535e-01 -7.09727966e-01
 1.73887268e+00  9.94394391e-01  1.31913688e+00 -8.82418819e-01
 1.12859406e+00  4.96000946e-01  7.71405949e-01  1.02943883e+00
-9.08763246e-01 -4.24317621e-01  8.62596011e-01 -2.65561909e+00
 1.51332808e+00  5.53132064e-01 -4.57039607e-02  2.20507656e-01
-1.02993528e+00 -3.49943365e-01  1.10028434e+00  1.29802197e+00
 2.69622405e+00 -7.39246663e-02 -6.58552967e-01 -5.14233966e-01
-1.01804188e+00 -7.78547559e-02  3.82732430e-01 -3.42422805e-02
 1.09634685e+00 -2.34215801e-01 -3.47450652e-01 -5.81268477e-01
-1.63263453e+00 -1.56776772e+00 -1.17915793e+00  1.30142807e+00
 8.95260273e-01  1.37496407e+00 -1.33221165e+00 -1.96862469e+00
-6.60056320e-01  1.75818953e-01  4.98690275e-01  1.04797216e+00
 2.84279671e-01  1.74266878e+00 -2.22605681e-01 -9.13079218e-01
-1.68121822e+00 -8.88971358e-01  2.42117961e-01 -8.88720257e-01
 9.36742464e-01  1.41232771e+00 -2.36958691e+00  8.64052300e-01
-2.23960406e+00  4.01499055e-01  1.22487056e+00  6.48561063e-02
-1.27968917e+00 -5.85431204e-01 -2.61645446e-01 -1.82244784e-01
-2.02896841e-01 -1.09882779e-01  2.13480049e-01 -1.20857365e+00
-2.42019830e-01  1.51826117e+00 -3.84645423e-01 -4.43836093e-01
 1.07819730e+00 -2.55918467e+00  1.18137860e+00 -6.31903758e-01
 1.63928572e-01  9.63213559e-02  9.42468119e-01 -2.67594746e-01
-6.78025782e-01  1.29784579e+00 -2.36417382e+00  2.03341817e-02
-1.34792542e+00 -7.61573388e-01  2.01125668e+00 -4.45954265e-02
 1.95069697e-01 -1.78156286e+00 -7.29044659e-01  1.96557401e-01
 3.54757693e-01  6.16886554e-01  8.62789892e-03  5.27004208e-01
 4.53781913e-01 -1.82974041e+00  3.70057219e-02  7.67902408e-01
 5.89879821e-01 -3.63858810e-01 -8.05626508e-01 -1.11831192e+00
-1.31054012e-01  1.13307988e+00 -1.95180410e+00 -6.59891730e-01
-1.13980246e+00  7.84957521e-01 -5.54309627e-01 -4.70637658e-01
-2.16949570e-01  4.45393251e-01 -3.92388998e-01 -3.04614305e+00
 5.43311891e-01  4.39042958e-01 -2.19541028e-01 -1.08403662e+00
 3.51780111e-01  3.79235534e-01 -4.70032883e-01 -2.16731471e-01
-9.30156503e-01 -1.78589092e-01 -1.55042935e+00  4.17318821e-01
-9.44368491e-01  2.38103148e-01 -1.40596292e+00 -5.90057646e-01
-1.10489405e-01 -1.66069981e+00  1.15147873e-01 -3.79147563e-01
-1.74235620e+00 -1.30324275e+00  6.05120084e-01  8.95555986e-01
-1.31908640e-01  4.04761812e-01  2.23843563e-01  3.29622982e-01
 1.28598401e+00 -1.50699840e+00  6.76460732e-01 -3.82008956e-01
-2.24258934e-01 -3.02249730e-01 -3.75147117e-01 -1.22619619e+00
 1.83339199e-01  1.67094303e+00 -5.61330204e-02 -1.38504274e-03
-6.87299037e-01 -1.17474546e-01  4.66166426e-01 -3.70242441e-01
-4.53804041e-01  4.03264540e-01 -9.18004770e-01  2.52496627e-01
 8.20321797e-01  1.35994854e+00 -9.03820073e-02  1.36759724e+00
 1.03440989e+00 -9.96212640e-01 -1.21793851e+00 -3.04963638e-01
 1.02893549e+00 -7.22870076e-02 -6.00657558e-01  1.55224318e+00
 2.86904488e-01 -2.32059428e+00  3.17160626e-01  5.20040615e-01
 2.25608654e-01  4.49712100e-01 -6.72756089e-02 -1.31839587e+00
-3.70704003e-01 -9.45615796e-01 -9.32740911e-01 -1.26306835e+00
 4.52489093e-01  9.78961454e-02 -4.48165363e-01 -6.49337928e-01
-2.34231050e-02  1.07919473e+00 -2.00421572e+00  3.76876521e-01
-5.45711974e-01 -1.88458584e+00 -1.94570308e+00 -9.12783494e-01
 2.19509556e-01  3.93062934e-01 -9.38981573e-01  1.01702099e+00
 1.42298350e+00  3.96086585e-01 -5.91402668e-01  1.12441918e+00
 7.55395696e-01  8.67407411e-01 -6.56463675e-01 -2.83455451e+00
 2.11679102e+00 -1.61087840e+00 -3.57680719e-02  2.38074535e+00
 3.30576756e-01  9.49246474e-01 -1.50239657e+00 -1.77766695e+00
-5.32702792e-01  1.09074973e+00 -3.46249448e-01 -7.94636321e-01
 1.97967290e-01  1.08193522e+00 -1.44494020e+00 -1.21054299e+00
-7.88669255e-01  1.09463837e+00  2.34821526e-01  2.13215341e+00
 9.36445726e-01 -3.50951769e-02  1.26507784e+00  2.11497013e-01
-7.04921353e-01  6.79974844e-01 -6.96326654e-01 -2.90397101e-01
 1.32778270e+00 -1.01281486e-01 -8.03141387e-01 -4.64337691e-01
 1.02179059e+00 -5.52540673e-01 -3.86870847e-01 -5.10292740e-01
 1.83925494e-01 -3.85489760e-01 -1.60183605e+00 -8.87180942e-01
-9.32789042e-01  1.24331938e+00  8.12674042e-01  5.87259379e-01
-5.05358317e-01 -8.15791542e-01 -5.07517602e-01 -1.05188010e+00
 2.49720039e+00 -2.24532165e+00  5.64008535e-01 -1.28455230e+00
-1.04343491e-01 -9.88001942e-01 -1.17762896e+00 -1.14019630e+00
```

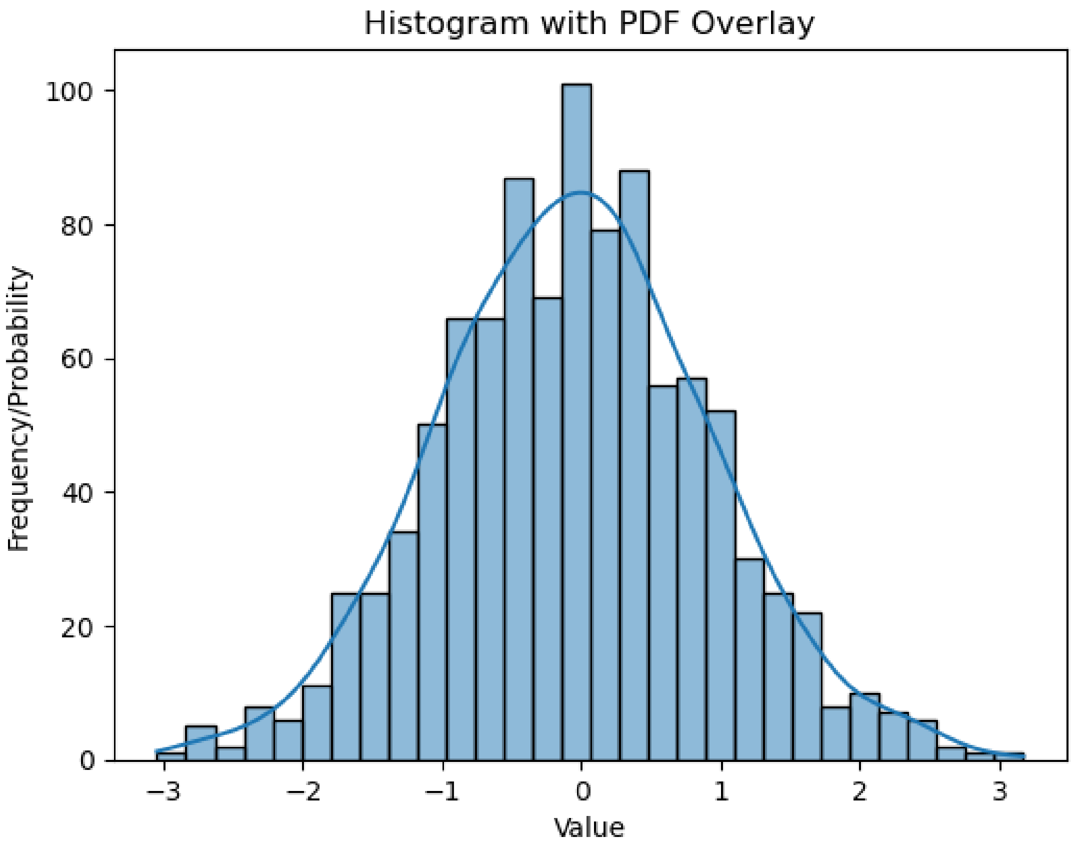
```
 1.75498615e+00 -1.32988422e-01 -7.65702194e-01  5.55786964e-01
 1.03493146e-02  7.20033759e-01 -1.82425666e+00  3.03603904e-01
 7.72694837e-01 -1.66159829e+00  4.48195284e-01  1.69618157e+00
-1.48577034e-02  8.21405937e-01  6.70570450e-01 -7.07505698e-01
 3.97667346e-02 -1.56699471e+00 -4.51303037e-01  2.65687975e-01
 7.23100494e-01  2.46121252e-02  7.19983730e-01 -1.10290621e+00
-1.01697275e-01  1.92793845e-02  1.84959125e+00 -2.14166656e-01
-4.99016638e-01  2.13512238e-02 -9.19113445e-01  1.92753849e-01
-3.65055217e-01 -1.79132755e+00 -5.85865511e-02 -3.17543094e-01
-1.63242330e+00 -6.71341546e-02  1.48935596e+00  5.21303748e-01
 6.11927193e-01 -1.34149673e+00  4.76898369e-01  1.48449581e-01
 5.29045238e-01  4.22628622e-01 -1.35978073e+00 -4.14008116e-02
-7.57870860e-01 -5.00840943e-02 -8.97400927e-01  1.31247037e+00
-8.58972388e-01 -8.98942156e-01  7.45864065e-02 -1.07709907e+00
-4.24663302e-01 -8.29964598e-01  1.41117206e+00  7.85803827e-01
-5.74695185e-02 -3.91217052e-01  9.40917615e-01  4.05204080e-01
 4.98052405e-01 -2.61922373e-02 -1.68823003e+00 -1.12465983e-01
-5.32489919e-01  6.45055273e-01  1.01184243e+00 -6.57951045e-01
 4.68385234e-01  1.73587900e+00 -6.67712721e-01  1.68192174e+00
-8.52585847e-01  2.29597556e-02 -1.11456118e-02  1.14988999e-02
-8.37678042e-01 -5.91183104e-01 -6.67720286e-01  3.26962595e-01
 3.30035115e-01  2.22594433e+00  1.37098901e+00 -5.09843242e-01
 3.24869616e-01  9.97117981e-01  3.06018243e-02 -6.96415784e-02
 5.15749428e-02  8.67276629e-01 -8.48320523e-01 -3.25669469e-01
 4.70433145e-01  3.11447072e-01  2.39582760e-01 -3.69801166e-01
 9.72535789e-01  2.13386825e+00  4.06415494e-01 -1.93176702e-01
 7.55740289e-01 -5.39132637e-01 -7.49690345e-01  3.28087476e-02
-2.58279663e+00 -1.15395036e+00 -3.47961856e-01 -1.35338886e+00
-1.03264310e+00 -4.36748337e-01 -1.64296529e+00 -4.06071796e-01
-5.35270165e-01  2.54052084e-02  1.15418403e+00  1.72504416e-01
 2.10620213e-02  9.94544570e-02  2.27392775e-01 -1.01673865e+00
-1.14775325e-01  3.08751242e-01 -1.37075998e+00  8.65652923e-01
 1.08137603e+00 -6.31375988e-01 -2.41337791e-01 -8.78190343e-01
 6.99380484e-01 -1.06122229e+00 -2.22477010e-01 -8.58919908e-01
 5.09542770e-02 -1.79422927e+00  1.32646164e+00 -9.64606424e-01
 5.98946831e-02 -2.12523045e-01 -7.62114512e-01 -8.87780137e-01
 9.36398544e-01 -5.25640593e-01  2.71170185e-01 -8.01496885e-01
-6.47181432e-01  4.72247150e-01  9.30408496e-01 -1.75316402e-01
-1.42191987e+00  1.99795608e+00 -8.56549308e-01 -1.54158740e+00
 2.59442459e+00 -4.04032294e-01 -1.46173269e+00 -6.83439767e-01
 3.67544896e-01  1.90311558e-01 -8.51729197e-01  1.82272360e+00
-5.21579678e-01 -1.18468659e+00  9.60693398e-01  1.32906285e+00
-8.17493098e-01 -1.40134729e+00  1.03043827e+00 -2.04732361e+00
-1.22662166e+00  9.67446150e-01 -5.53525480e-02 -2.63937349e-01
 3.52816606e-01 -1.52774424e-01 -1.29868672e+00  1.27607535e+00
 1.32501405e+00  2.05332564e-01  4.51340154e-02  2.33962481e+00
-2.76432845e-01 -2.59576982e-01  3.64481249e-01  1.47132196e+00
 1.59277075e+00 -2.58572632e-01  3.08331246e-01 -1.37808347e+00
```

```
-3.11976108e-01 -8.40290395e-01 -1.00683175e+00  1.68157672e+00
-7.92286662e-01 -5.31605908e-01  3.65848788e-01  1.29782527e+00
 4.81115126e-01  2.75935511e+00 -7.46679783e-02  2.58716440e-01
 2.75600674e-01  1.43504939e+00  5.07238951e-01 -1.16229700e-01
-9.47488595e-01  2.44443456e-01  1.40134483e+00 -4.10381794e-01
 5.28943618e-01  2.46147789e-01  8.63519658e-01 -8.04753741e-01
 2.34664703e+00 -1.27916111e+00 -3.65551090e-01  9.38092541e-01
 2.96733172e-01  8.29986159e-01 -4.96102334e-01 -7.48049827e-02
 1.22319836e-02  1.56925961e+00  6.90429024e-01  7.96672108e-01
-6.57926093e-01  9.68882639e-01  2.25581664e-01  1.38914532e+00
 2.01406015e+00 -3.06765776e-01 -4.06303130e-01 -8.64044991e-01
-1.43579512e-01 -3.82025449e-01  3.59504400e-01 -1.44566817e-01
-3.61599281e-01  1.06458514e+00 -9.37880231e-01  4.33107953e-01
-4.05941727e-01  7.24368505e-01  1.38526155e+00 -3.03098253e-01
 4.41032907e-01  1.78792866e-01 -7.99422400e-01  2.40787510e-01
 2.89120505e-01  4.12870820e-01 -1.98398897e-01  9.41923003e-02
-1.14761094e+00 -3.58114075e-01  5.55962680e-01  8.92473887e-01
-4.22314824e-01  1.04714029e-01  2.28053325e-01  2.01479947e-01
 5.40773585e-01 -1.81807763e+00 -4.93240701e-02  2.39033601e-01
-1.00033035e+00  1.67398571e+00  1.61559267e-01  1.56340475e+00
-7.90523022e-01 -9.07300122e-01  2.24252221e-01 -1.67868836e+00
 2.14965591e-01  9.72192320e-02  1.01566528e+00  7.01041341e-01
-4.17477350e-01 -1.09749665e+00  1.71230522e+00 -7.92115021e-01
-1.04552456e+00 -1.08485606e+00  1.11730532e+00 -5.18900204e-01
-7.53704466e-01  1.37689826e-01 -2.06944711e-01 -6.78095461e-01
 7.53991467e-01  1.06531549e+00  9.85317509e-01  7.66919670e-01
 4.02625531e-01 -1.77588800e+00  1.66925081e+00  3.01989210e-01
 6.08156428e-01  1.11496232e+00  1.43335250e+00  4.18398011e-01
 4.35546159e-01 -5.99224277e-01  3.30897511e-02 -8.54161261e-01
-7.19940532e-01 -8.93574402e-01 -1.56023891e-01  1.04909319e+00
 3.17097477e+00  1.89499638e-01 -1.34841309e+00  1.26498333e+00
-3.00783876e-01 -6.60608594e-01  2.09849478e-01 -1.24062460e+00
 2.22463164e-01 -8.83755232e-02  9.83779068e-02  3.81416254e-01
 6.74922572e-02  1.63380841e-02  2.84314519e-01  4.15400626e-01
-1.03148246e+00 -1.42999126e+00 -6.16380522e-02 -1.43273549e+00
 8.75314709e-02  9.38746876e-01  6.07111672e-01 -1.04817041e+00
-8.60262452e-01  3.28301295e-01 -4.01297805e-01 -3.16655295e-01
 5.96906481e-01 -9.87286693e-01 -4.01234710e-01 -8.00082476e-01
-1.04312950e+00 -8.57078189e-01  6.77462169e-01  5.18203895e-02
-8.79160629e-01 -2.31101608e-01 -1.63880731e+00 -7.33312808e-01
 2.14957453e+00 -9.02438497e-02  7.31658927e-01 -6.54883751e-02
 3.48169235e-01  6.63258090e-01 -1.10461660e+00 -3.09362573e-02
 1.57886519e+00 -7.95500550e-01 -5.66439854e-01 -3.07691277e-01
 2.69024073e-01  5.24917864e-01  1.26741165e+00  4.99498233e-01
-6.20531258e-02  1.25916713e+00  7.04111022e-01 -1.49567952e+00
 2.52636824e+00  1.76992139e+00 -1.68214223e-01  3.77910102e-01
 1.32435875e+00 -1.72200793e-01  7.30351790e-01  1.10457847e+00
-1.01482591e+00 -6.02331854e-01  9.21408398e-01  4.60814477e-01
```

```
  9.23796560e-01 -1.32568015e-01 -2.89005211e-01 -1.99863948e+00
 -1.14600043e+00  4.70660947e-02  8.24557220e-01  5.31178367e-01
 -1.28241974e-01 -2.71771566e-01  2.17179633e-01  7.82111811e-02]
```

[35]:

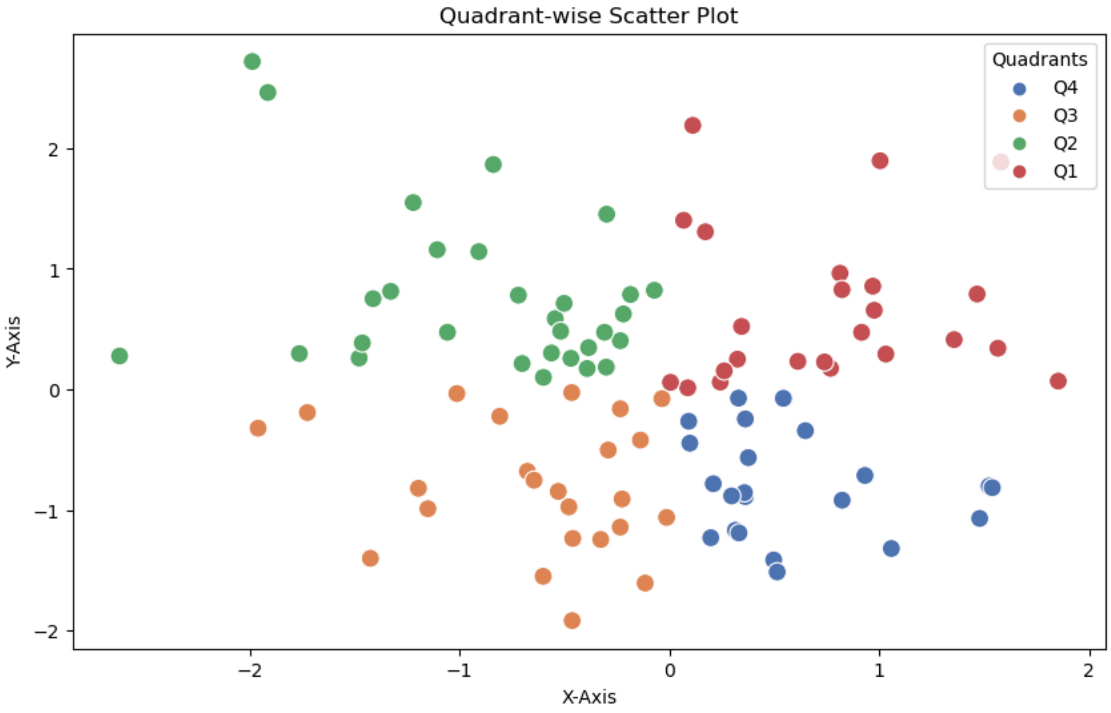
```python
sns.histplot(x=data,bins=30,kde=True)
plt.xlabel('Value')
plt.ylabel('Frequency/Probability')
plt.title('Histogram with PDF Overlay')
plt.show()
```



Question 17

[36]:

```python
np.random.seed(42)
x = np.random.randn(100)
y = np.random.randn(100)

# Determine the quadrant for each point
def get_quadrant(x, y):
    if x >= 0 and y >= 0:
        return 'Q1'
    elif x < 0 and y >= 0:
        return 'Q2'
    elif x < 0 and y < 0:
        return 'Q3'
    else:
        return 'Q4'

df = pd.DataFrame({'x': x, 'y': y})
df['Quadrant'] = df.apply(lambda row: get_quadrant(row['x'], row['y']), axis=1)
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='x', y='y', hue='Quadrant', palette='deep', s=100)
plt.xlabel('X-Axis')
plt.ylabel('Y-Axis')
plt.title('Quadrant-wise Scatter Plot')
plt.legend(title='Quadrants', loc='upper right')
plt.show()
```



Question 18

[37]:
```
from bokeh.plotting import figure, show
from bokeh.io import output_notebook
import numpy as np

output_notebook()
```

```
x = np.linspace(0, 2 * np.pi, 100)
y = np.sin(x)

p = figure(title="Sine Wave Function", x_axis_label='X-Axis (radians)',␣
 ↪y_axis_label='Y-Axis (amplitude)')

p.line(x, y, legend_label="Sine Wave", line_width=2)

p.grid.grid_line_color = 'gray'
p.grid.grid_line_alpha = 0.5
show(p)
```

Question 19

[39]:

```
import pandas as pd
import numpy as np
from bokeh.io import show, output_notebook
from bokeh.plotting import figure
from bokeh.models import ColumnDataSource, HoverTool
from bokeh.transform import factor_cmap


output_notebook()


np.random.seed(42)
categories = ['A', 'B', 'C', 'D', 'E']
values = np.random.randint(10, 100, size=len(categories))


data = pd.DataFrame({
    'Category': categories,
    'Value': values
})


source = ColumnDataSource(data=data)


colors = factor_cmap('Category', palette='Spectral11', factors=categories)


p = figure(x_range=categories, height=350, title='Random Categorical Bar Chart',
           toolbar_location=None, tools="")
```

```python
p.vbar(x='Category', top='Value', width=0.9, source=source, color=colors)

hover = HoverTool()
hover.tooltips = [("Category", "@Category"), ("Value", "@Value")]
p.add_tools(hover)


p.xaxis.axis_label = 'Category'
p.yaxis.axis_label = 'Value'


p.xgrid.grid_line_color = None
p.y_range.start = 0
p.yaxis.axis_label_standoff = 12
p.xaxis.major_label_orientation = "vertical"

show(p)
```

Question 20

[45]:

```python
pip install plotly
```

```
Requirement already satisfied: plotly in /opt/conda/lib/python3.10/site-packages
(5.23.0)
Requirement already satisfied: tenacity>=6.2.0 in
/opt/conda/lib/python3.10/site-packages (from plotly) (9.0.0)
Requirement already satisfied: packaging in /opt/conda/lib/python3.10/site-
packages (from plotly) (22.0)
Note: you may need to restart the kernel to use updated packages.
```

[46]:

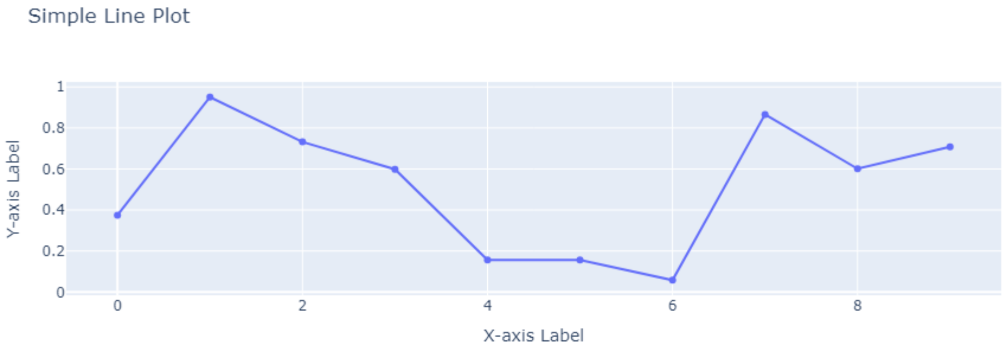
```python
import plotly
print(plotly.__version__)
```

```
5.23.0
```

[47]:

```python
import plotly.graph_objects as go
import numpy as np

# Generate random data
np.random.seed(42)  # For reproducibility
x = np.arange(10)  # X-axis values
y = np.random.random(10)  # Random Y-axis values

# Create a line plot
fig = go.Figure()

# Add a line trace
fig.add_trace(go.Scatter(x=x, y=y, mode='lines+markers', name='Random Data'))
```

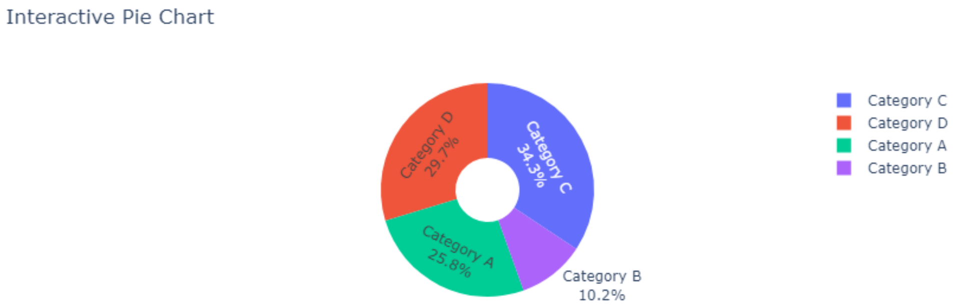
```
# Update the layout with title and axis labels
fig.update_layout(
    title='Simple Line Plot',
    xaxis_title='X-axis Label',
    yaxis_title='Y-axis Label'
)

# Show the plot
fig.show()
```



Question 21

[48]:

```
import plotly.graph_objects as go
import numpy as np

# Generate random data
np.random.seed(42)  # For reproducibility
labels = ['Category A', 'Category B', 'Category C', 'Category D']
values = np.random.randint(10, 100, size=len(labels))

# Create a pie chart
fig = go.Figure(data=[go.Pie(
    labels=labels,
    values=values,
    textinfo='label+percent',  # Show both label and percentage
    hole=0.3  # Optional: makes it a donut chart if set to a value between 0␣
 ↪and 1
)])

# Update layout with title
fig.update_layout(
    title='Interactive Pie Chart'
)

# Show the plot
fig.show()
```


Interactive Pie Chart
Category D
Category C
Category A
Category B
Category C
Category D
Category A
Category B