

```
In [1]: import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings(action='ignore')
```

```
In [2]: df=pd.read_csv(r"C:\Users\Shree\Downloads\A2.csv")
df
```

```
Out[2]:
```

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	NaN	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	NaN	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

```
In [3]: df.head()
```

```
Out[3]:
```

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	NaN	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	NaN	84.0	86
4	5	Student_5	97	70	84.0	70.0	86

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1001 entries, 0 to 1000
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Roll No     1001 non-null   int64
1   Name        1001 non-null   object
2   Subject 1   1001 non-null   int64
3   Subject 2   1001 non-null   int64
4   Subject 3   1000 non-null   float64
5   Subject 4   1000 non-null   float64
6   Attendance  1001 non-null   int64
dtypes: float64(2), int64(4), object(1)
memory usage: 54.9+ KB
```

In [5]: `df.dtypes`

```
Out[5]: Roll No      int64
Name         object
Subject 1     int64
Subject 2     int64
Subject 3     float64
Subject 4     float64
Attendance    int64
dtype: object
```

In [6]: `df.isnull()`

```
Out[6]:
```

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	False	False	False	False	False	False	False
1	False	False	False	False	False	True	False
2	False	False	False	False	False	False	False
3	False	False	False	False	True	False	False
4	False	False	False	False	False	False	False
...
996	False	False	False	False	False	False	False
997	False	False	False	False	False	False	False
998	False	False	False	False	False	False	False
999	False	False	False	False	False	False	False
1000	False	False	False	False	False	False	False

1001 rows × 7 columns

In [7]: `df.isnull().sum()`

```
Out[7]: Roll No      0
        Name        0
        Subject 1    0
        Subject 2    0
        Subject 3     1
        Subject 4     1
        Attendance   0
        dtype: int64
```

```
In [8]: df.isnull().sum().sum()
```

```
Out[8]: 2
```

```
In [9]: df.ffill()
```

```
Out[9]:
```

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	92.0	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	71.0	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

```
In [10]: df=pd.read_csv(r"C:\Users\Shree\Downloads\A2.csv")
         df.bfill()
```

Out[10]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	99.0	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	84.0	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

```
In [11]: df=pd.read_csv(r"C:\Users\Shree\Downloads\A2.csv")
df.fillna({"Subject 3":df["Subject 3"].mean(),inplace=True)#nan replace by mean
df.fillna({"Subject 4":df["Subject 4"].mean(),inplace=True)#nan=sub 4 mean
df
```

Out[11]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.00	92.000	96
1	2	Student_2	72	97	82.00	80.545	78
2	3	Student_3	100	88	71.00	99.000	-94
3	4	Student_4	72	99	79.89	84.000	86
4	5	Student_5	97	70	84.00	70.000	86
...
996	997	Student_997	88	68	84.00	66.000	98
997	998	Student_998	61	96	62.00	84.000	83
998	999	Student_999	72	76	90.00	72.000	90
999	1000	Student_1000	68	87	100.00	76.000	79
1000	1	Student_1	100	62	73.00	92.000	96

1001 rows × 7 columns

```
In [12]: df=pd.read_csv(r"C:\Users\Shree\Downloads\A2.csv")
df.fillna({"Subject 3":df["Subject 3"].median(),inplace=True)#apply for median
df.fillna({"Subject 4":df["Subject 4"].mode(),inplace=True)
df
```

Out[12]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	95.0	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	80.0	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

In [13]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1001 entries, 0 to 1000
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Roll No     1001 non-null  int64
1   Name        1001 non-null  object
2   Subject 1   1001 non-null  int64
3   Subject 2   1001 non-null  int64
4   Subject 3   1001 non-null  float64
5   Subject 4   1001 non-null  float64
6   Attendance  1001 non-null  int64
dtypes: float64(2), int64(4), object(1)
memory usage: 54.9+ KB
```

```
In [14]: df=pd.read_csv(r"C:\Users\Shree\Downloads\A2.csv")
df.fillna(0,inplace=True)
df
```

Out[14]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	0.0	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	0.0	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

```
In [15]: df=pd.read_csv(r"C:\Users\Shree\Downloads\A2.csv")
df.fillna({"Subject 3":3000},inplace=True)
df
```

Out[15]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	NaN	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	3000.0	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

```
In [16]: df=pd.read_csv(r"C:\Users\Shree\Downloads\A2.csv")
nc=['Subject 1','Subject 2','Subject 3','Subject 4','Attendance']
for c in nc:
    df.fillna({c:df[c].mean()},inplace=True)
df
```

Out[16]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.00	92.000	96
1	2	Student_2	72	97	82.00	80.545	78
2	3	Student_3	100	88	71.00	99.000	-94
3	4	Student_4	72	99	79.89	84.000	86
4	5	Student_5	97	70	84.00	70.000	86
...
996	997	Student_997	88	68	84.00	66.000	98
997	998	Student_998	61	96	62.00	84.000	83
998	999	Student_999	72	76	90.00	72.000	90
999	1000	Student_1000	68	87	100.00	76.000	79
1000	1	Student_1	100	62	73.00	92.000	96

1001 rows × 7 columns

In [17]:

```
df=pd.read_csv(r"C:\Users\Shree\Downloads\A2.csv")
nc=['Subject 1','Subject 2','Subject 3','Subject 4','Attendance']
for c in nc:
    df.fillna({c:df[c].median()}),inplace=True)
df
```

Out[17]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	81.0	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	80.0	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

In [18]:

```
df=pd.DataFrame(df)
df=df.apply(pd.to_numeric,errors='coerce')
df.interpolate(method='linear',limit_direction='forward')
```

Out[18]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	NaN	100	62	73.0	92.0	96
1	2	NaN	72	97	82.0	81.0	78
2	3	NaN	100	88	71.0	99.0	-94
3	4	NaN	72	99	80.0	84.0	86
4	5	NaN	97	70	84.0	70.0	86
...
996	997	NaN	88	68	84.0	66.0	98
997	998	NaN	61	96	62.0	84.0	83
998	999	NaN	72	76	90.0	72.0	90
999	1000	NaN	68	87	100.0	76.0	79
1000	1	NaN	100	62	73.0	92.0	96

1001 rows × 7 columns

In [19]:

```
df=pd.read_csv(r"C:\Users\Shree\Downloads\A2.csv")
df1=pd.DataFrame(df)
df1[df1[["Subject 1","Subject 2","Subject 3","Subject 4","Attendance"]]<0]
```

Out[19]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	-94.0
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...
996	NaN	NaN	NaN	NaN	NaN	NaN	NaN
997	NaN	NaN	NaN	NaN	NaN	NaN	NaN
998	NaN	NaN	NaN	NaN	NaN	NaN	NaN
999	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1000	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1001 rows × 7 columns

In [20]:

```
df=pd.read_csv(r"C:\Users\Shree\Downloads\A2.csv")
df1=pd.DataFrame(df)
df1.drop_duplicates(inplace=True)
df
```


Out[20]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	NaN	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	NaN	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

In [21]: df1.shape

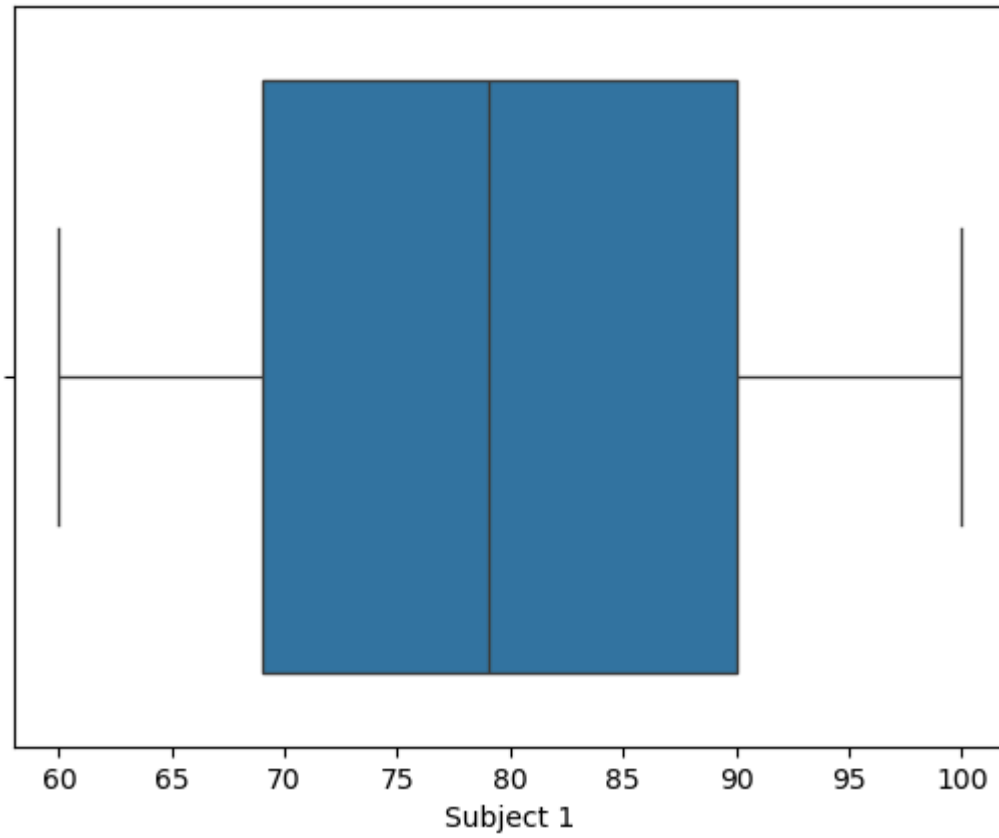
Out[21]: (1000, 7)

```

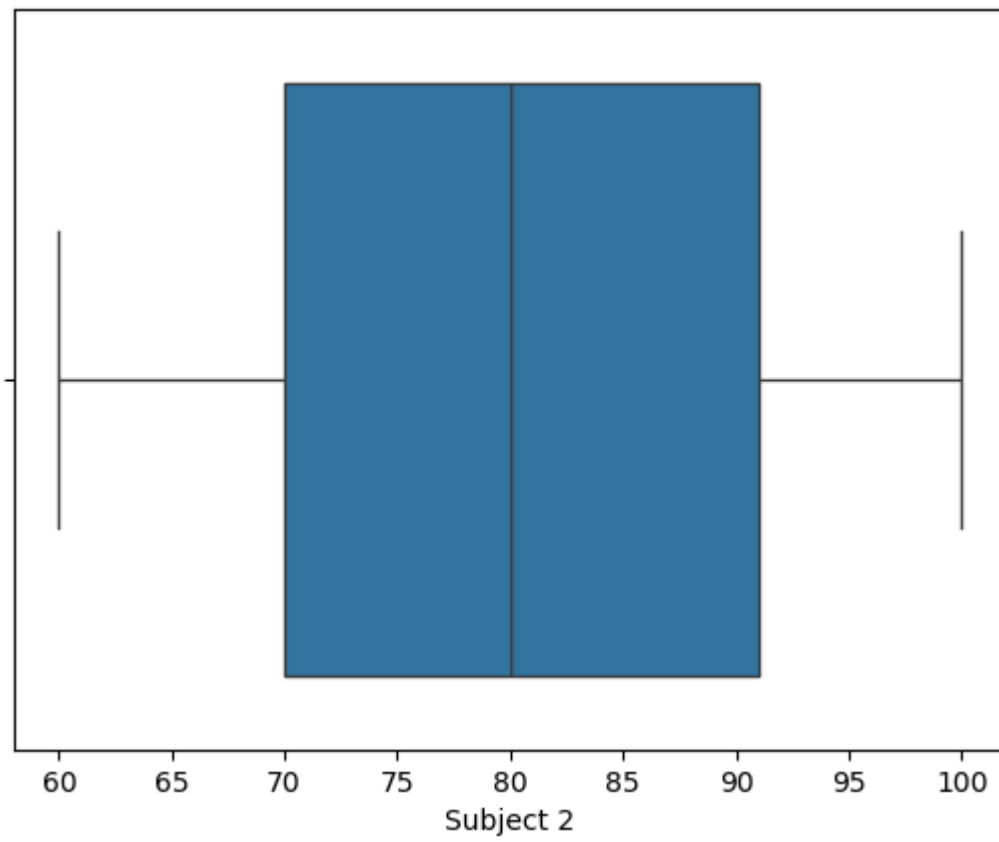
In [22]: import seaborn as sns
import matplotlib.pyplot as plt
nc=["Subject 1","Subject 2","Subject 3","Subject 4","Attendance"]
for c in nc:
    sns.boxplot(x=df[c])
    plt.title(f'box plot of {c}')
    plt.show()

```

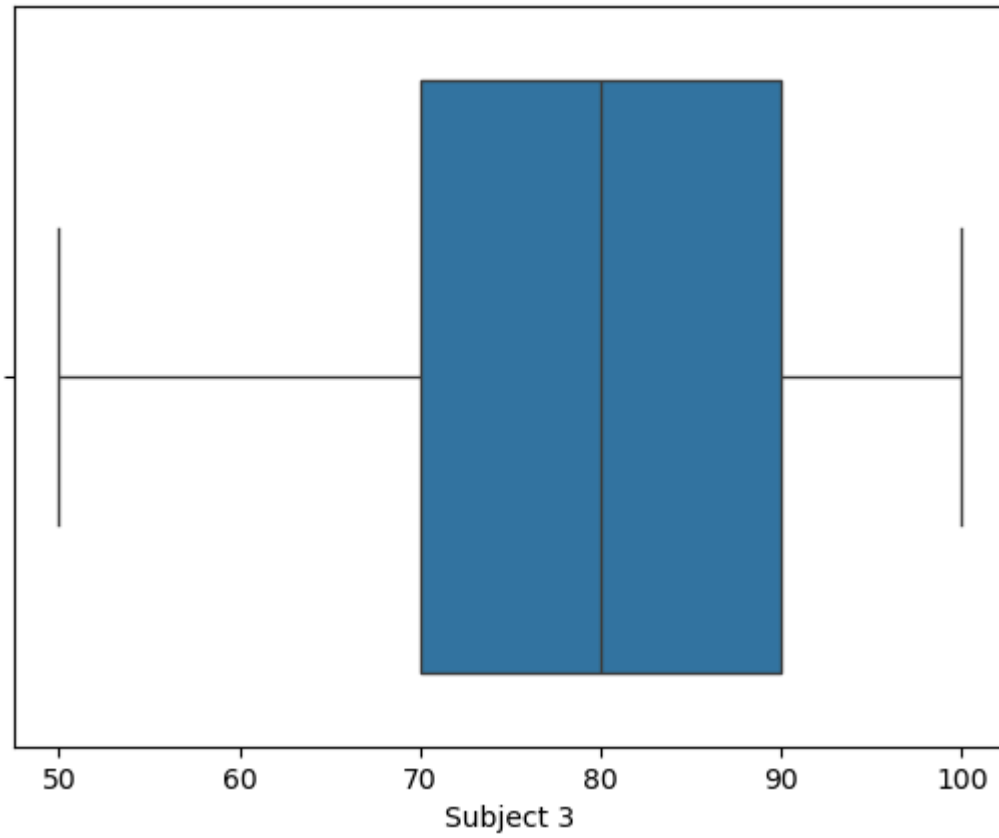
box plot of Subject 1



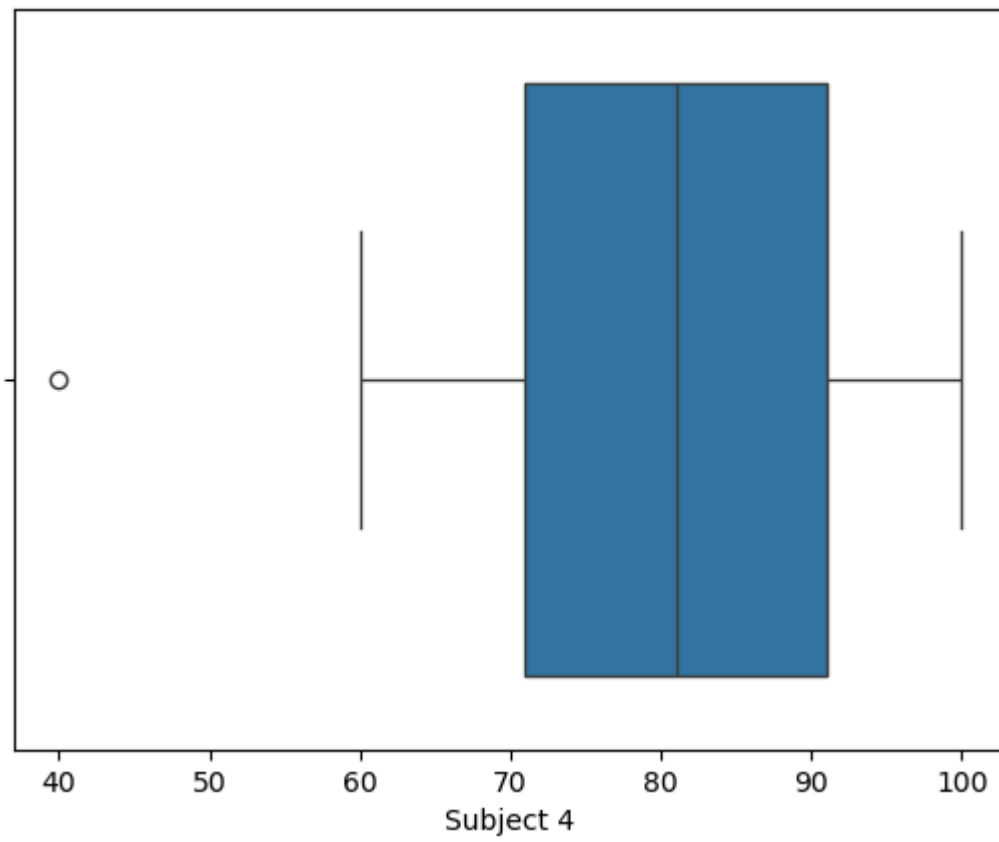
box plot of Subject 2

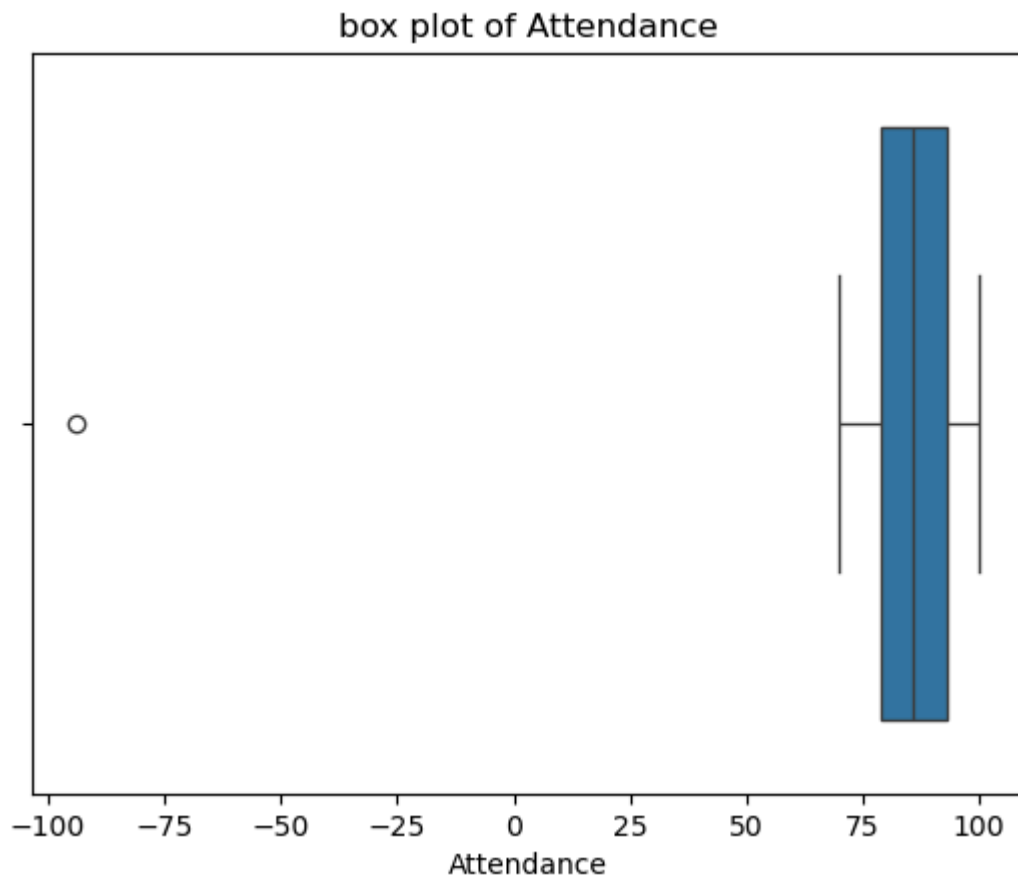


box plot of Subject 3



box plot of Subject 4





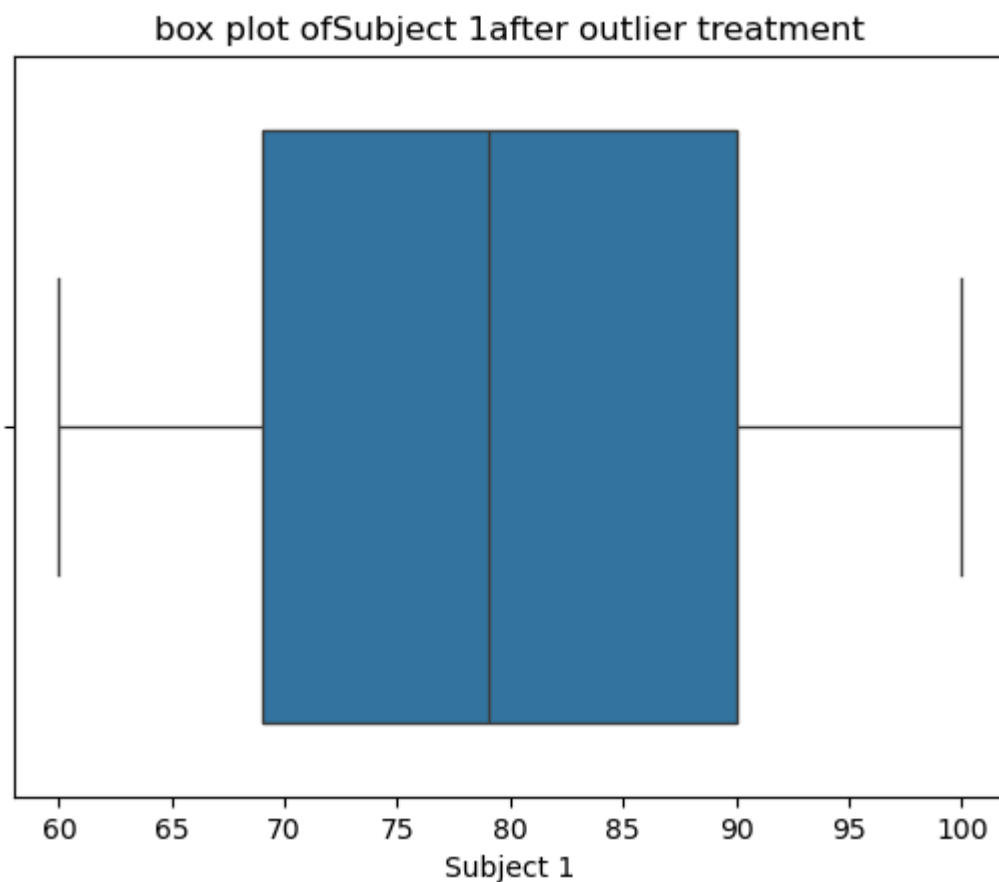
```
In [23]: for c in nc:
          Q1=df[c].quantile(0.25)
          Q3=df[c].quantile(0.75)
          IQR=Q3-Q1
          lowerbound=Q1-1.5*IQR
          upperbound=Q3+1.5*IQR
          df[c]=np.where((df[c]<lowerbound)|(df[c]>upperbound),df[c].median(),df[c])
df
```

Out[23]:

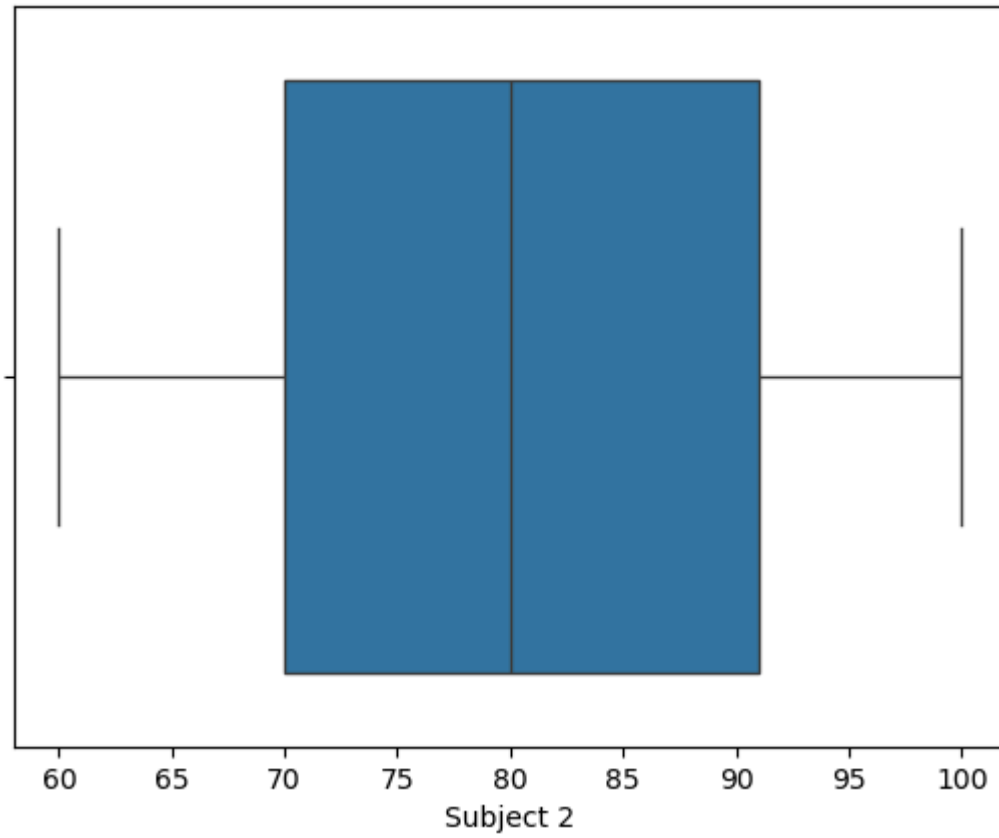
	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100.0	62.0	73.0	92.0	96.0
1	2	Student_2	72.0	97.0	82.0	NaN	78.0
2	3	Student_3	100.0	88.0	71.0	99.0	86.0
3	4	Student_4	72.0	99.0	NaN	84.0	86.0
4	5	Student_5	97.0	70.0	84.0	70.0	86.0
...
996	997	Student_997	88.0	68.0	84.0	66.0	98.0
997	998	Student_998	61.0	96.0	62.0	84.0	83.0
998	999	Student_999	72.0	76.0	90.0	72.0	90.0
999	1000	Student_1000	68.0	87.0	100.0	76.0	79.0
1000	1	Student_1	100.0	62.0	73.0	92.0	96.0

1001 rows × 7 columns

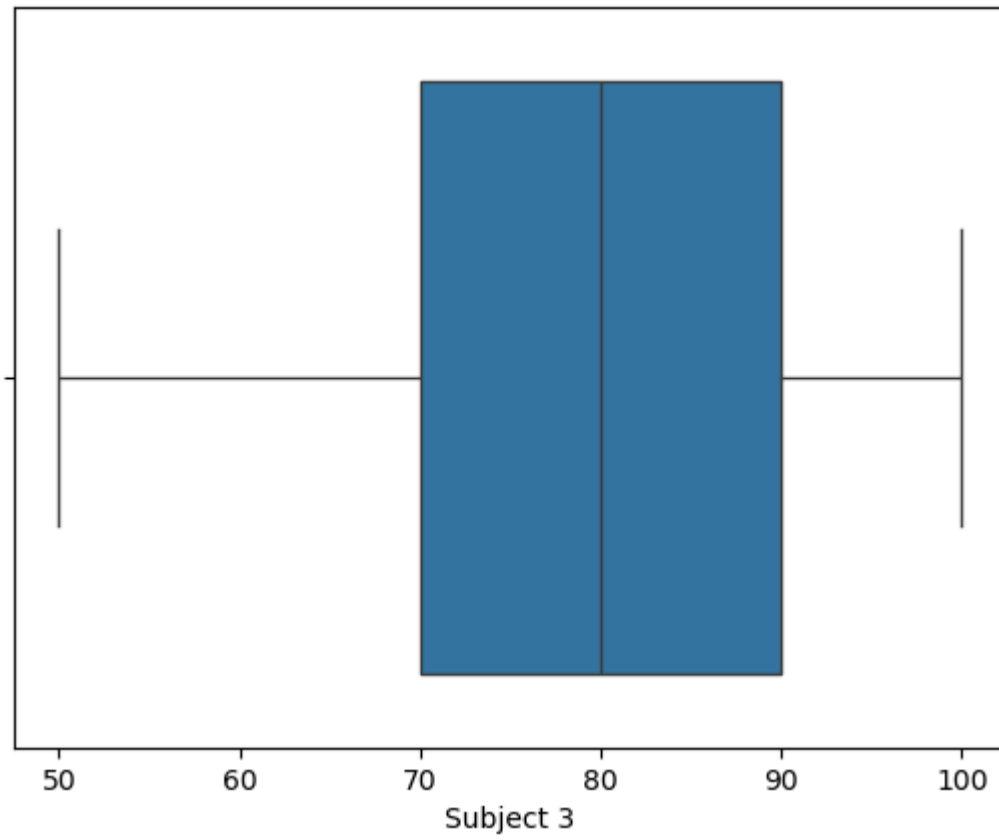
```
In [24]: for c in nc:
sns.boxplot(x=df[c])
plt.title(f'box plot of{c}after outlier treatment')
plt.show()
```



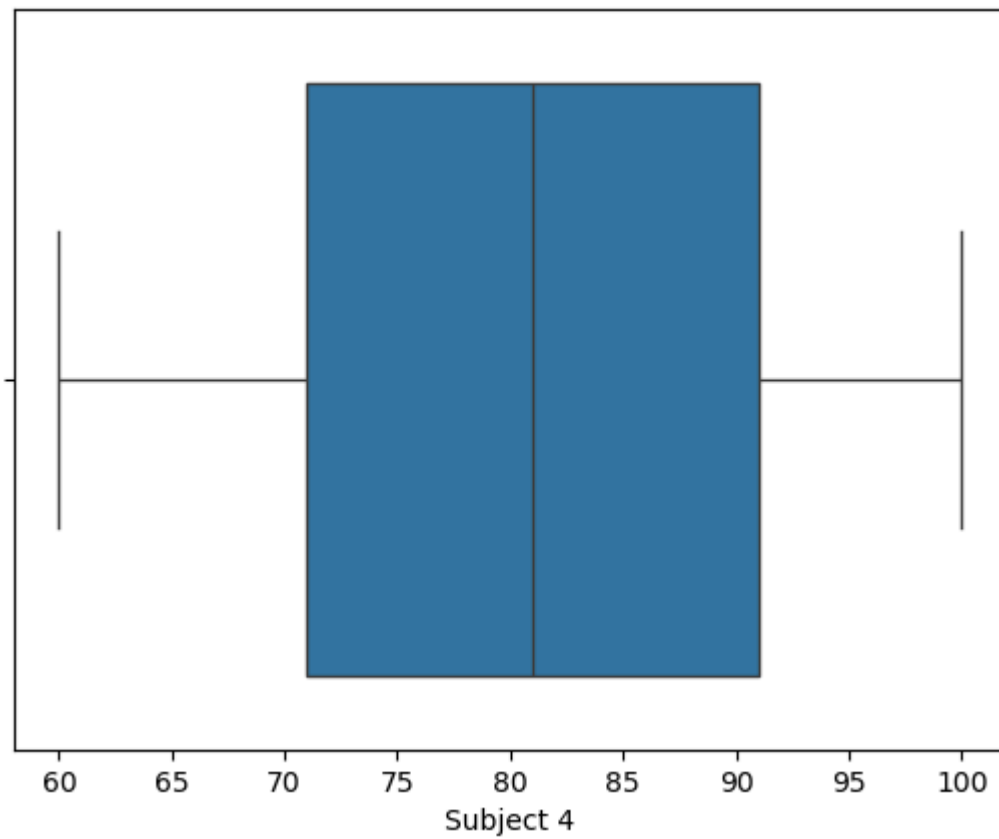
box plot of Subject 2 after outlier treatment



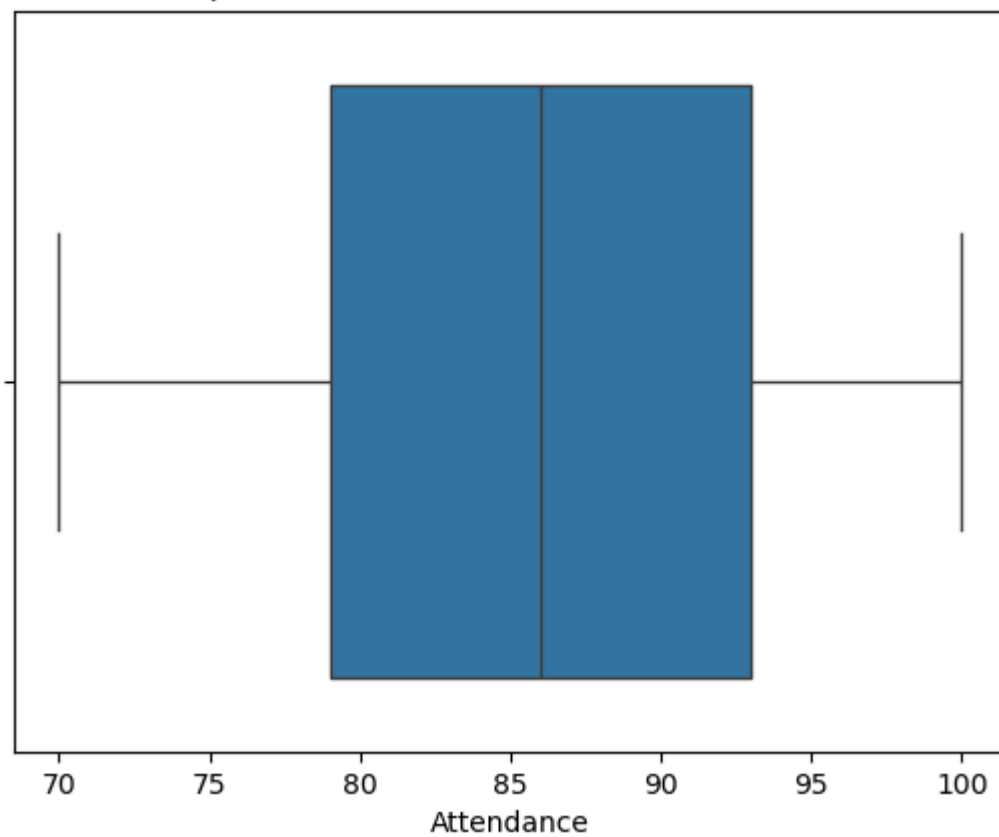
box plot of Subject 3 after outlier treatment



box plot ofSubject 4after outlier treatment



box plot ofAttendanceafter outlier treatment



```
In [25]: from scipy import stats
z=np.abs(stats.zscore(df["Subject 2"]))
z
```

```
Out[25]: 0      1.518270
         1      1.423310
         2      0.666904
         3      1.591400
         4      0.845909
         ...
        996     1.013999
        997     1.339265
        998     0.341638
        999     0.582858
       1000     1.518270
Name: Subject 2, Length: 1001, dtype: float64
```

```
In [26]: threshold=1
         print(np.where(z>1))
         z
```

```
(array([ 0,  1,  3,  6, 10, 12, 13, 16, 17, 18, 22,
        28, 34, 36, 38, 41, 42, 43, 44, 45, 46, 47,
        48, 49, 51, 52, 53, 57, 65, 66, 67, 68, 70,
        72, 73, 75, 76, 77, 79, 80, 82, 86, 87, 92,
        97, 106, 107, 108, 109, 110, 111, 114, 119, 122, 125,
       126, 127, 128, 130, 132, 133, 134, 135, 144, 145, 147,
       148, 150, 154, 156, 157, 159, 165, 166, 167, 171, 172,
       174, 177, 183, 184, 189, 190, 191, 193, 194, 195, 196,
       198, 201, 203, 204, 207, 209, 211, 212, 214, 216, 217,
       218, 221, 222, 226, 227, 229, 230, 231, 232, 233, 235,
       238, 242, 244, 245, 246, 248, 250, 254, 256, 257, 258,
       263, 265, 270, 272, 276, 280, 281, 282, 284, 287, 293,
       294, 298, 299, 303, 309, 313, 314, 320, 322, 323, 326,
       327, 329, 331, 332, 333, 334, 336, 338, 340, 341, 343,
       345, 346, 347, 350, 352, 356, 359, 360, 362, 364, 365,
       367, 369, 372, 373, 374, 378, 380, 382, 384, 385, 388,
       389, 390, 392, 396, 399, 400, 401, 402, 404, 405, 406,
       412, 414, 415, 418, 419, 421, 427, 428, 430, 432, 441,
       450, 451, 452, 455, 456, 463, 469, 471, 475, 476, 477,
       478, 481, 487, 488, 490, 491, 493, 497, 501, 502, 503,
       505, 506, 507, 510, 512, 513, 514, 515, 519, 523, 525,
       527, 531, 535, 538, 539, 540, 542, 543, 544, 545, 547,
       549, 551, 552, 553, 554, 555, 557, 558, 559, 562, 563,
       564, 565, 566, 569, 570, 572, 573, 574, 576, 584, 585,
       586, 587, 589, 590, 591, 592, 593, 594, 600, 601, 604,
       605, 609, 611, 614, 616, 618, 619, 621, 623, 624, 627,
       628, 629, 631, 634, 637, 641, 644, 645, 648, 650, 654,
       656, 657, 660, 661, 663, 666, 668, 669, 674, 675, 677,
       679, 681, 682, 684, 685, 690, 691, 692, 694, 696, 697,
       698, 700, 701, 703, 704, 705, 706, 707, 708, 710, 717,
       718, 723, 724, 725, 727, 731, 732, 737, 738, 740, 746,
       749, 750, 757, 759, 764, 765, 766, 767, 768, 769, 771,
       772, 775, 779, 782, 783, 785, 786, 788, 789, 790, 792,
       793, 794, 795, 799, 801, 805, 806, 807, 809, 810, 811,
       812, 813, 814, 818, 820, 822, 824, 826, 827, 832, 834,
       838, 839, 840, 844, 846, 848, 850, 851, 855, 856, 857,
       859, 862, 864, 869, 870, 871, 873, 875, 876, 880, 881,
       885, 887, 890, 891, 892, 894, 896, 900, 901, 905, 909,
       911, 913, 919, 922, 924, 927, 928, 929, 930, 933, 934,
       944, 945, 952, 953, 956, 961, 962, 964, 966, 967, 968,
       970, 975, 976, 977, 987, 988, 989, 991, 992, 993, 996,
       997, 1000], dtype=int64),)
```



```
Out[26]: 0      1.518270
          1      1.423310
          2      0.666904
          3      1.591400
          4      0.845909
          ...
          996    1.013999
          997    1.339265
          998    0.341638
          999    0.582858
          1000    1.518270
          Name: Subject 2, Length: 1001, dtype: float64
```

```
In [27]: df.drop_duplicates(inplace=True)
          df['log_attendance']=np.log1p(df['Attendance'])
          df
```

```
Out[27]:
```

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance	log_attendance
0	1	Student_1	100.0	62.0	73.0	92.0	96.0	4.57471
1	2	Student_2	72.0	97.0	82.0	NaN	78.0	4.36944
2	3	Student_3	100.0	88.0	71.0	99.0	86.0	4.46590
3	4	Student_4	72.0	99.0	NaN	84.0	86.0	4.46590
4	5	Student_5	97.0	70.0	84.0	70.0	86.0	4.46590
...
995	996	Student_996	74.0	89.0	85.0	71.0	87.0	4.47733
996	997	Student_997	88.0	68.0	84.0	66.0	98.0	4.59512
997	998	Student_998	61.0	96.0	62.0	84.0	83.0	4.43081
998	999	Student_999	72.0	76.0	90.0	72.0	90.0	4.51086
999	1000	Student_1000	68.0	87.0	100.0	76.0	79.0	4.38202

1000 rows × 8 columns



```
In [28]: df1['sqrt_attendance']=np.sqrt(df['Attendance'])
          df1
```

Out[28]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance	sqrt_attendanc
0	1	Student_1	100	62	73.0	92.0	96	9.79795
1	2	Student_2	72	97	82.0	NaN	78	8.83176
2	3	Student_3	100	88	71.0	99.0	-94	9.27361
3	4	Student_4	72	99	NaN	84.0	86	9.27361
4	5	Student_5	97	70	84.0	70.0	86	9.27361
...
995	996	Student_996	74	89	85.0	71.0	87	9.32735
996	997	Student_997	88	68	84.0	66.0	98	9.89949
997	998	Student_998	61	96	62.0	84.0	83	9.11043
998	999	Student_999	72	76	90.0	72.0	90	9.48683
999	1000	Student_1000	68	87	100.0	76.0	79	8.88819

1000 rows × 8 columns



```
In [29]: numeric_cols=["Subject 1","Subject 2","Subject 3","Subject 4","Attendance"]
df[numeric_cols]=df[numeric_cols].fillna(method='bfill')
df
#normalization
df[numeric_cols]=(df[numeric_cols]-df[numeric_cols].min())/(df[numeric_cols].max
print(df)
```

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	\
0	1	Student_1	1.000	0.050	0.46	0.800	
1	2	Student_2	0.300	0.925	0.64	0.975	
2	3	Student_3	1.000	0.700	0.42	0.975	
3	4	Student_4	0.300	0.975	0.68	0.600	
4	5	Student_5	0.925	0.250	0.68	0.250	
..	
995	996	Student_996	0.350	0.725	0.70	0.275	
996	997	Student_997	0.700	0.200	0.68	0.150	
997	998	Student_998	0.025	0.900	0.24	0.600	
998	999	Student_999	0.300	0.400	0.80	0.300	
999	1000	Student_1000	0.200	0.675	1.00	0.400	

	Attendance	log_attendance
0	0.866667	4.574711
1	0.266667	4.369448
2	0.533333	4.465908
3	0.533333	4.465908
4	0.533333	4.465908
..
995	0.566667	4.477337
996	0.933333	4.595120
997	0.433333	4.430817
998	0.666667	4.510860
999	0.300000	4.382027

[1000 rows x 8 columns]

```
In [33]: numeric_cols = ['Subject 1', 'Subject 2', 'Subject 3', 'Subject 4', 'Attendance']
for col in numeric_cols:
    df1[f'sqrt_{col}']=np.sqrt(df1[col])
    print(df1)
```

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	\
0	1	Student_1	100	62	73.0	92.0	
1	2	Student_2	72	97	82.0	NaN	
2	3	Student_3	100	88	71.0	99.0	
3	4	Student_4	72	99	NaN	84.0	
4	5	Student_5	97	70	84.0	70.0	
..	
995	996	Student_996	74	89	85.0	71.0	
996	997	Student_997	88	68	84.0	66.0	
997	998	Student_998	61	96	62.0	84.0	
998	999	Student_999	72	76	90.0	72.0	
999	1000	Student_1000	68	87	100.0	76.0	

	Attendance	sqrt_attendance	sqrt_Subject 1	sqrt_Subject 2	\
0	96	9.797959	10.000000	7.874008	
1	78	8.831761	8.485281	9.848858	
2	-94	9.273618	10.000000	9.380832	
3	86	9.273618	8.485281	9.949874	
4	86	9.273618	9.848858	8.366600	
..	
995	87	9.327379	8.602325	9.433981	
996	98	9.899495	9.380832	8.246211	
997	83	9.110434	7.810250	9.797959	
998	90	9.486833	8.485281	8.717798	
999	79	8.888194	8.246211	9.327379	

	sqrt_Subject 3	sqrt_Subject 4	sqrt_Attendance
0	8.544004	9.591663	9.797959
1	9.055385	NaN	8.831761
2	8.426150	9.949874	NaN
3	NaN	9.165151	9.273618
4	9.165151	8.366600	9.273618
..
995	9.219544	8.426150	9.327379
996	9.165151	8.124038	9.899495
997	7.874008	9.165151	9.110434
998	9.486833	8.485281	9.486833
999	10.000000	8.717798	8.888194

[1000 rows x 13 columns]

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	\
0	1	Student_1	100	62	73.0	92.0	
1	2	Student_2	72	97	82.0	NaN	
2	3	Student_3	100	88	71.0	99.0	
3	4	Student_4	72	99	NaN	84.0	
4	5	Student_5	97	70	84.0	70.0	
..	
995	996	Student_996	74	89	85.0	71.0	
996	997	Student_997	88	68	84.0	66.0	
997	998	Student_998	61	96	62.0	84.0	
998	999	Student_999	72	76	90.0	72.0	
999	1000	Student_1000	68	87	100.0	76.0	

	Attendance	sqrt_attendance	sqrt_Subject 1	sqrt_Subject 2	\
0	96	9.797959	10.000000	7.874008	
1	78	8.831761	8.485281	9.848858	
2	-94	9.273618	10.000000	9.380832	
3	86	9.273618	8.485281	9.949874	
4	86	9.273618	9.848858	8.366600	
..	

995	87	9.327379	8.602325	9.433981
996	98	9.899495	9.380832	8.246211
997	83	9.110434	7.810250	9.797959
998	90	9.486833	8.485281	8.717798
999	79	8.888194	8.246211	9.327379

	sqrt_Subject 3	sqrt_Subject 4	sqrt_Attendance
0	8.544004	9.591663	9.797959
1	9.055385	NaN	8.831761
2	8.426150	9.949874	NaN
3	NaN	9.165151	9.273618
4	9.165151	8.366600	9.273618
..
995	9.219544	8.426150	9.327379
996	9.165151	8.124038	9.899495
997	7.874008	9.165151	9.110434
998	9.486833	8.485281	9.486833
999	10.000000	8.717798	8.888194

[1000 rows x 13 columns]

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	\
0	1	Student_1	100	62	73.0	92.0	
1	2	Student_2	72	97	82.0	NaN	
2	3	Student_3	100	88	71.0	99.0	
3	4	Student_4	72	99	NaN	84.0	
4	5	Student_5	97	70	84.0	70.0	
..	
995	996	Student_996	74	89	85.0	71.0	
996	997	Student_997	88	68	84.0	66.0	
997	998	Student_998	61	96	62.0	84.0	
998	999	Student_999	72	76	90.0	72.0	
999	1000	Student_1000	68	87	100.0	76.0	

	Attendance	sqrt_attendance	sqrt_Subject 1	sqrt_Subject 2	\
0	96	9.797959	10.000000	7.874008	
1	78	8.831761	8.485281	9.848858	
2	-94	9.273618	10.000000	9.380832	
3	86	9.273618	8.485281	9.949874	
4	86	9.273618	9.848858	8.366600	
..	
995	87	9.327379	8.602325	9.433981	
996	98	9.899495	9.380832	8.246211	
997	83	9.110434	7.810250	9.797959	
998	90	9.486833	8.485281	8.717798	
999	79	8.888194	8.246211	9.327379	

	sqrt_Subject 3	sqrt_Subject 4	sqrt_Attendance
0	8.544004	9.591663	9.797959
1	9.055385	NaN	8.831761
2	8.426150	9.949874	NaN
3	NaN	9.165151	9.273618
4	9.165151	8.366600	9.273618
..
995	9.219544	8.426150	9.327379
996	9.165151	8.124038	9.899495
997	7.874008	9.165151	9.110434
998	9.486833	8.485281	9.486833
999	10.000000	8.717798	8.888194

[1000 rows x 13 columns]

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	\
0	1	Student_1	100	62	73.0	92.0	
1	2	Student_2	72	97	82.0	NaN	
2	3	Student_3	100	88	71.0	99.0	
3	4	Student_4	72	99	NaN	84.0	
4	5	Student_5	97	70	84.0	70.0	
..	
995	996	Student_996	74	89	85.0	71.0	
996	997	Student_997	88	68	84.0	66.0	
997	998	Student_998	61	96	62.0	84.0	
998	999	Student_999	72	76	90.0	72.0	
999	1000	Student_1000	68	87	100.0	76.0	

	Attendance	sqrt_attendance	sqrt_Subject 1	sqrt_Subject 2	\
0	96	9.797959	10.000000	7.874008	
1	78	8.831761	8.485281	9.848858	
2	-94	9.273618	10.000000	9.380832	
3	86	9.273618	8.485281	9.949874	
4	86	9.273618	9.848858	8.366600	
..	
995	87	9.327379	8.602325	9.433981	
996	98	9.899495	9.380832	8.246211	
997	83	9.110434	7.810250	9.797959	
998	90	9.486833	8.485281	8.717798	
999	79	8.888194	8.246211	9.327379	

	sqrt_Subject 3	sqrt_Subject 4	sqrt_Attendance
0	8.544004	9.591663	9.797959
1	9.055385	NaN	8.831761
2	8.426150	9.949874	NaN
3	NaN	9.165151	9.273618
4	9.165151	8.366600	9.273618
..
995	9.219544	8.426150	9.327379
996	9.165151	8.124038	9.899495
997	7.874008	9.165151	9.110434
998	9.486833	8.485281	9.486833
999	10.000000	8.717798	8.888194

[1000 rows x 13 columns]

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	\
0	1	Student_1	100	62	73.0	92.0	
1	2	Student_2	72	97	82.0	NaN	
2	3	Student_3	100	88	71.0	99.0	
3	4	Student_4	72	99	NaN	84.0	
4	5	Student_5	97	70	84.0	70.0	
..	
995	996	Student_996	74	89	85.0	71.0	
996	997	Student_997	88	68	84.0	66.0	
997	998	Student_998	61	96	62.0	84.0	
998	999	Student_999	72	76	90.0	72.0	
999	1000	Student_1000	68	87	100.0	76.0	

	Attendance	sqrt_attendance	sqrt_Subject 1	sqrt_Subject 2	\
0	96	9.797959	10.000000	7.874008	
1	78	8.831761	8.485281	9.848858	
2	-94	9.273618	10.000000	9.380832	
3	86	9.273618	8.485281	9.949874	
4	86	9.273618	9.848858	8.366600	
..	

995	87	9.327379	8.602325	9.433981
996	98	9.899495	9.380832	8.246211
997	83	9.110434	7.810250	9.797959
998	90	9.486833	8.485281	8.717798
999	79	8.888194	8.246211	9.327379

	sqrt_Subject 3	sqrt_Subject 4	sqrt_Attendance
0	8.544004	9.591663	9.797959
1	9.055385	NaN	8.831761
2	8.426150	9.949874	NaN
3	NaN	9.165151	9.273618
4	9.165151	8.366600	9.273618
..
995	9.219544	8.426150	9.327379
996	9.165151	8.124038	9.899495
997	7.874008	9.165151	9.110434
998	9.486833	8.485281	9.486833
999	10.000000	8.717798	8.888194

[1000 rows x 13 columns]

In [37]: `df1['zscore_Attendance']=(df1['Attendance']-df1['Attendance'].mean())/(df1['Attendance'].std())`

Out[37]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance	sqrt_attendance
0	1	Student_1	100	62	73.0	92.0	96	9.797959
1	2	Student_2	72	97	82.0	NaN	78	8.831761
2	3	Student_3	100	88	71.0	99.0	-94	9.273618
3	4	Student_4	72	99	NaN	84.0	86	9.273618
4	5	Student_5	97	70	84.0	70.0	86	9.273618
...
995	996	Student_996	74	89	85.0	71.0	87	9.327379
996	997	Student_997	88	68	84.0	66.0	98	9.899495
997	998	Student_998	61	96	62.0	84.0	83	9.110434
998	999	Student_999	72	76	90.0	72.0	90	9.486833
999	1000	Student_1000	68	87	100.0	76.0	79	8.888194

1000 rows x 14 columns



In []: