Assignment -2
Python Programming

Assignment Date	19 September 2022
Student Name	LOGADHARSHINI M
Student Roll Number	2019115052
Maximum Marks	2 Marks

import required libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from matplotlib import rcParams

loading dataset

import pandas as pd

df=pd.read_csv('water.csv',encoding='latin-1')

df.head()

df.shape

df.info()

```
In [14]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1991 entries, 0 to 1990
Data columns (total 12 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Station_code           1991 non-null   object
 1   Locations              1991 non-null   object
 2   State                  1991 non-null   object
 3   Temp                   1991 non-null   object
 4   Do                     1991 non-null   object
 5   Ph                     1991 non-null   object
 6   Conductivity           1991 non-null   object
 7   Bod                    1991 non-null   object
 8   NITRATENAN N+ NITRITENANN  1991 non-null   object
 9   Fecal_coliform         1991 non-null   object
 10  Total_coliform         1991 non-null   object
 11  year                   1991 non-null   int64
dtypes: int64(1), object(11)
memory usage: 186.8+ KB
```

df.isnull().any()

```
In [15]: df.isnull().any()

Out[15]: Station_code               False
         Locations                  False
         State                      False
         Temp                       False
         Do                         False
         Ph                         False
         Conductivity               False
         Bod                        False
         NITRATENAN N+ NITRITENANN  False
         Fecal_coliform             False
         Total_coliform             False
         year                       False
         dtype: bool
```

df.Temp.value_counts()

```
In [16]: df.Temp.value_counts()

Out[16]: 28        241
         29        163
         27        161
         26        102
         NAN        88
                   ...
         25.667      1
         21.2        1
         31.1        1
         27.714      1
         14          1
         Name: Temp, Length: 179, dtype: int64
```

df.Ph.value_counts()

```
In [17]: df.Ph.value_counts()

Out[17]: 7.2     138
         7.4     127
         7.3     126
         7.1     118
         7       112
                ...
         7.22      1
         8.11      1
         8.07      1
         7.98      1
         110       1
         Name: Ph, Length: 266, dtype: int64
```

df.year.value_counts()

```
In [18]: df.year.value_counts()

Out[18]: 2012    292
         2013    261
         2014    245
         2011    231
         2010    188
         2009    181
         2008    159
         2007    120
         2005    119
         2006    105
         2003     88
         2004      2
         Name: year, dtype: int64
```
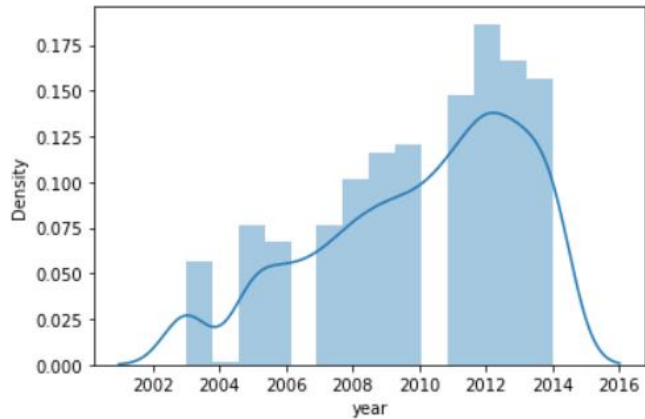
df.describe()

```
In [19]: df.describe()

Out[19]:
```

	year
count	1991.000000
mean	2010.038172
std	3.057333
min	2003.000000
25%	2008.000000
50%	2011.000000
75%	2013.000000
max	2014.000000

import seaborn as sns

sns.distplot(df.year) # univariate analysis

```
In [20]: import seaborn as sns
         sns.distplot(df.year) # univariate analysis
```

```
          warnings.warn(msg, FutureWarning)
```
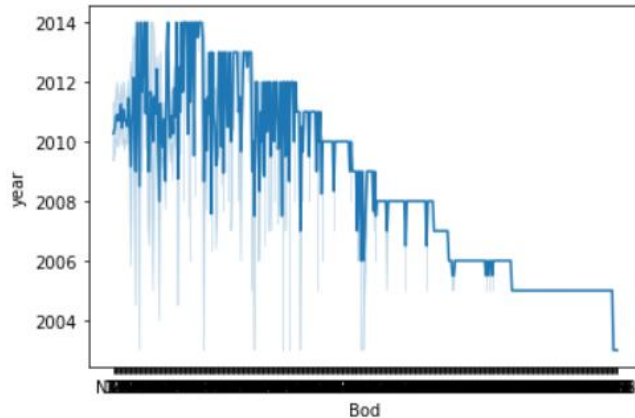
Out[20]: <AxesSubplot:xlabel='year', ylabel='Density'>

sns.lineplot(df.Bod,df.year) # bivariate analysis.

```
In [21]: sns.lineplot(df.Bod,df.year) # bivariate analysis.
```

```
          warnings.warn(
```

Out[21]: <AxesSubplot:xlabel='Bod', ylabel='year'>


sns.barplot(df.Ph.value_counts().index, df.Ph.value_counts())

```
In [24]: sns.barplot(df.Ph.value_counts().index, df.Ph.value_counts())

         D:\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
         the following variables as keyword args: x, y. From version 0.12, the only va
         lid positional argument will be `data`, and passing other arguments without a
         n explicit keyword will result in an error or misinterpretation.
           warnings.warn(

Out[24]: <AxesSubplot:ylabel='Ph'>
```
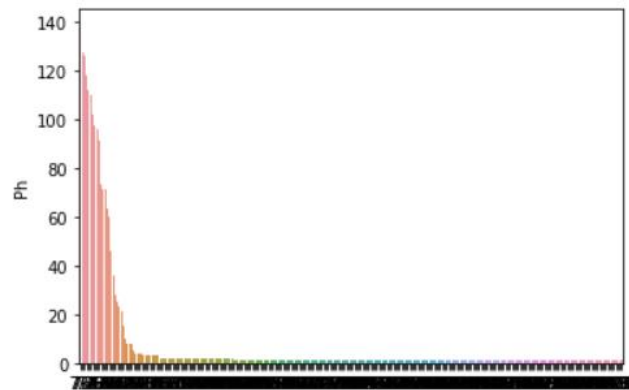

countplot

import seaborn as sns

import matplotlib.pyplot as plt

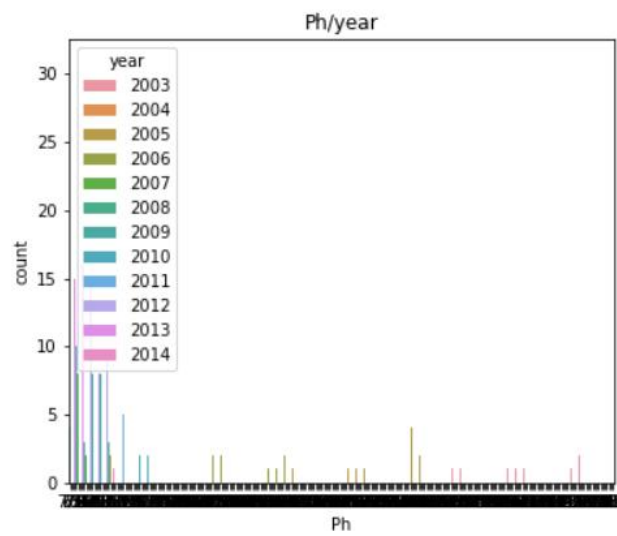def countplot_of_2(x,hue,title=None,figsize=(6,5)):

  plt.figure(figsize=figsize)

  sns.countplot(data=df[[x,hue]],x=x,hue=hue)

  plt.title(title)

  plt.show()

countplot_of_2('Ph','year','Ph/year')

```
In [15]: countplot_of_2('Ph','year','Ph/year')
```


countplot_of_2('Bod','year','Bod/year')

```
In [26]: countplot_of_2('Bod','year','Bod/year')
```
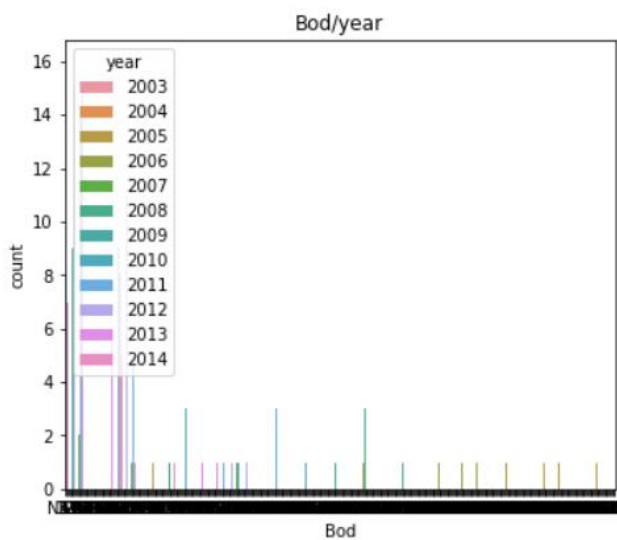

sns.pairplot(df)

```
In [28]: sns.pairplot(df)

Out[28]: <seaborn.axisgrid.PairGrid at 0x202b8ec27c0>
```
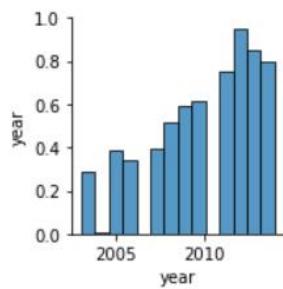

df.corr()

```
In [29]: df.corr()

Out[29]:
                year
        year    1.0
```

sns.heatmap(df.corr())

```
In [30]: sns.heatmap(df.corr())

Out[30]: <AxesSubplot:>
```
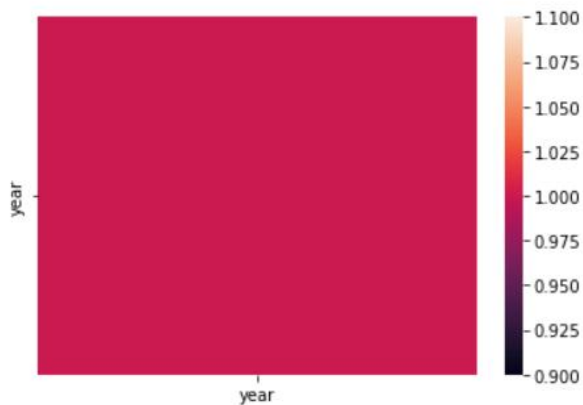

sns.heatmap(df.corr(),annot=True)

```
In [31]: sns.heatmap(df.corr(),annot=True)

Out[31]: <AxesSubplot:>
```
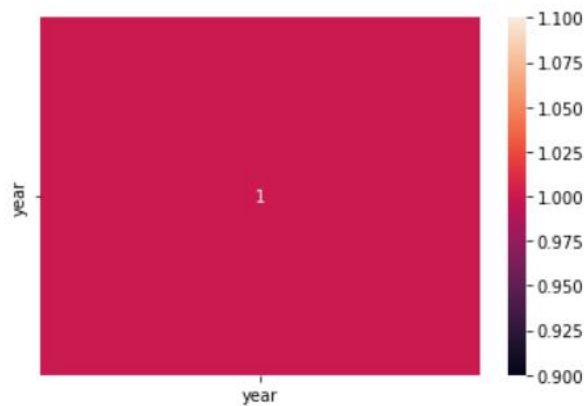

sns.boxplot(df.year)

```
In [32]: sns.boxplot(df.year)

D:\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pas
s the following variable as a keyword arg: x. From version 0.12, the only v
alid positional argument will be `data`, and passing other arguments withou
t an explicit keyword will result in an error or misinterpretation.
  warnings.warn(

Out[32]: <AxesSubplot:xlabel='year'>
```


outlier removal- IQR method

Q1= df.year.quantile(0.25)

Q3=df.year.quantile(0.75)

IQR=Q3-Q1

upper_limit =Q3 + 1.5*IQR

lower_limit =Q1 - 1.5*IQR

upper_limit

```
In [36]: upper_limit
Out[36]: 2020.5
```

df=df[df.year<upper_limit]

sns.boxplot(df.year)

```
In [38]: sns.boxplot(df.year)

         D:\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
         the following variable as a keyword arg: x. From version 0.12, the only valid
         positional argument will be `data`, and passing other arguments without an ex
         plicit keyword will result in an error or misinterpretation.
           warnings.warn(

Out[38]: <AxesSubplot:xlabel='year'>
```


df.shape

```
In [39]: df.shape
Out[39]: (1991, 12)
```
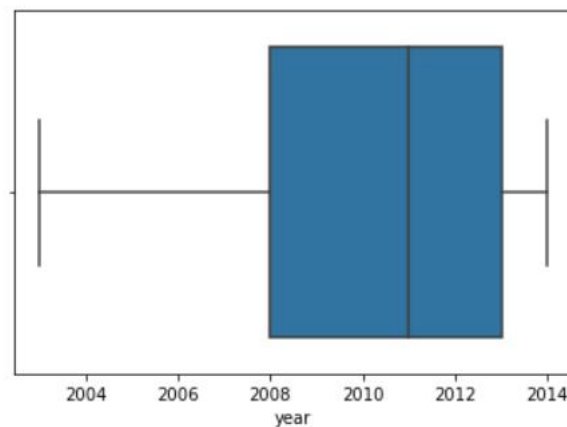
Outlier replacement using median

sns.boxplot(df.year)

```
In [40]: sns.boxplot(df.year)
```

D:\anaconda\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an ex
plicit keyword will result in an error or misinterpretation.
  warnings.warn(

Out[40]: <AxesSubplot:xlabel='year'>
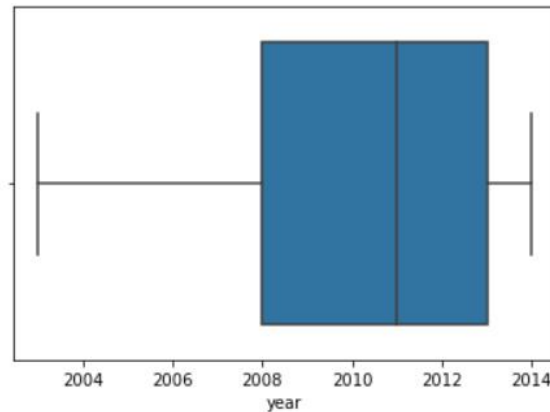
df.median()

```
In [41]: df.median()
```

C:\Users\KRISHN~1\AppData\Local\Temp/ipykernel_3196/530051474.py:1: FutureWar
ning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_onl
y=None') is deprecated; in a future version this will raise TypeError. Selec
t only valid columns before calling the reduction.
  df.median()

```
Out[41]: Station_code    1861.0
         Do                 6.7
         Ph                 7.3
         Conductivity     183.0
         year            2011.0
         dtype: float64
```

Q1= df.year.quantile(0.25)

Q3=df.year.quantile(0.75)

IQR=Q3-Q1

upper_limit =Q3 + 1.5*IQR

lower_limit =Q1 - 1.5*IQR
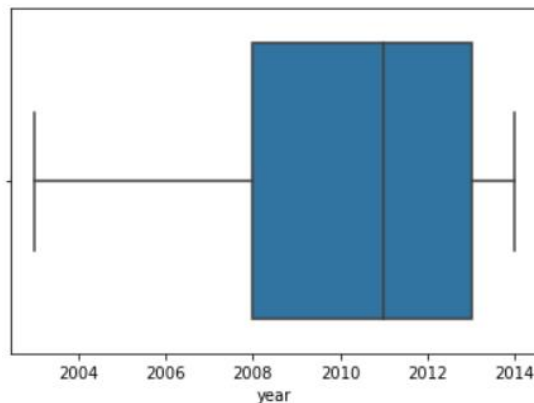
sns.boxplot(df.year)

```
In [45]: sns.boxplot(df.year)

         D:\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
         the following variable as a keyword arg: x. From version 0.12, the only valid
         positional argument will be `data`, and passing other arguments without an ex
         plicit keyword will result in an error or misinterpretation.
           warnings.warn(

Out[45]: <AxesSubplot:xlabel='year'>
```


df.shape

```
In [46]: df.shape
Out[46]: (1991, 12)
```

outlier removal- Percentile method

sns.boxplot(df.year)

```
In [47]: sns.boxplot(df.year)

D:\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an ex
plicit keyword will result in an error or misinterpretation.
  warnings.warn(

Out[47]: <AxesSubplot:xlabel='year'>
```
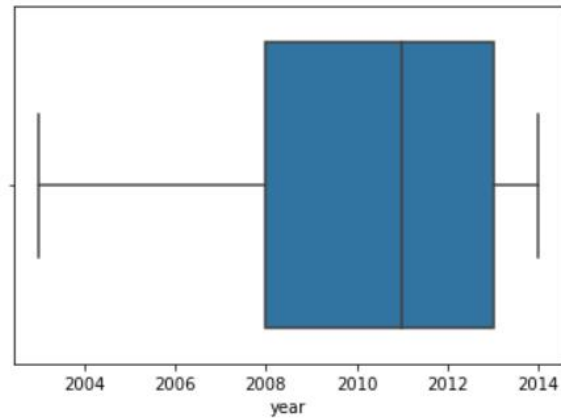

p99= df.year.quantile(0.99)

p99

```
In [48]: p99= df.year.quantile(0.99)
         p99
Out[48]: 2014.0
```

df = df[df.year<=p99]

sns.boxplot(df.year)

```
In [51]: sns.boxplot(df.year)
```

D:\anaconda\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an ex
plicit keyword will result in an error or misinterpretation.
  warnings.warn(

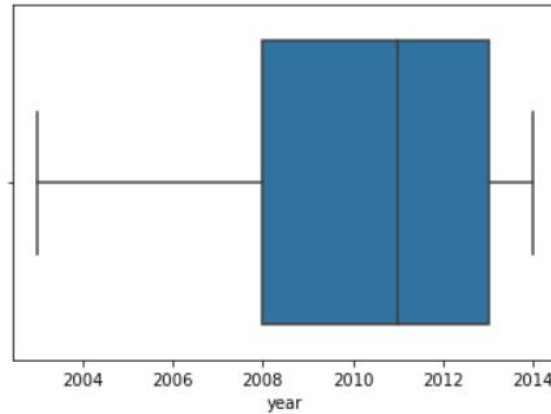Out[51]: <AxesSubplot:xlabel='year'>
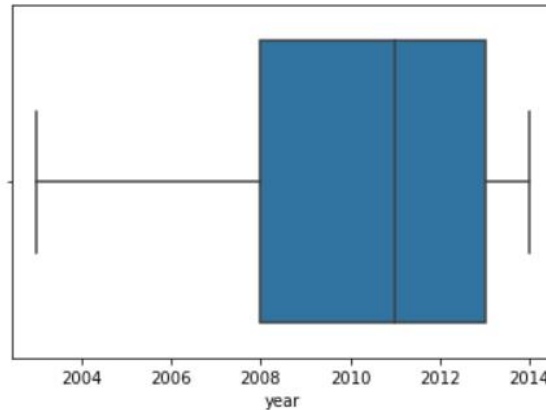
outlier removal- z-score

sns.boxplot(df.year)

```
In [52]: sns.boxplot(df.year)
```

D:\anaconda\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an ex
plicit keyword will result in an error or misinterpretation.
  warnings.warn(

Out[52]: <AxesSubplot:xlabel='year'>
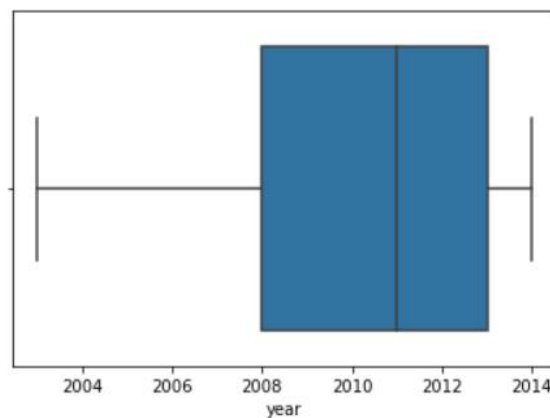
from scipy import stats

```
from scipy import stats

year_zscore = stats.zscore(df.year)

year_zscore
```

```
In [55]: year_zscore

Out[55]: 0        1.296170
         1        1.296170
         2        1.296170
         3        1.296170
         4        1.296170
                   ...
         1986    -2.302641
         1987    -2.302641
         1988    -2.302641
         1989    -2.302641
         1990    -2.302641
         Name: year, Length: 1991, dtype: float64
```

```
df_z = df[np.abs(year_zscore)<=3]

sns.boxplot(df_z.year)
```

```
In [57]: sns.boxplot(df_z.year)

D:\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an ex
plicit keyword will result in an error or misinterpretation.
  warnings.warn(

Out[57]: <AxesSubplot:xlabel='year'>
```
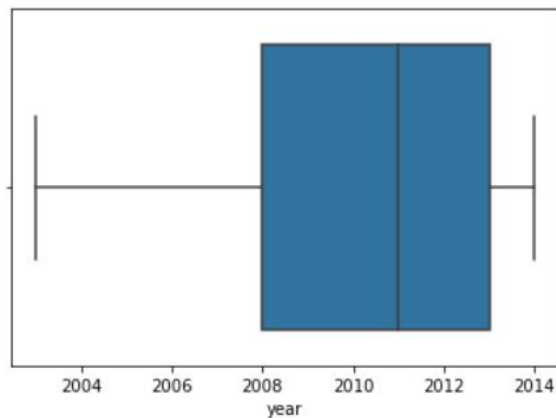


```
df.head()
```

```
In [58]: df.head()
```

Out[58]:

	Station_code	Locations	State	Temp	Do	Ph	Conductivity	Bod	NITRATENAN N+ NITRITENANN	Fec
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203	NAN	0.1	
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189	2	0.2	
2	1475	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179	1.7	0.1	
3	3181	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64	3.8	0.5	
4	3182	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	7.3	83	1.9	0.4	

Encoding Techniques

1.Label Encoding¶

from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

df.head()


```
In [61]: df.head()
```

Out[61]:

	Station_code	Locations	State	Temp	Do	Ph	Conductivity	Bod	NITRATENAN N+ NITRITENANN	Fec
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203	NAN	0.1	
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189	2	0.2	
2	1475	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179	1.7	0.1	
3	3181	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64	3.8	0.5	
4	3182	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	7.3	83	1.9	0.4	

2.One hot Encoding¶

df_main=pd.get_dummies(df,columns=['Station_code'])

df_main.head()

```
In [62]: df_main=pd.get_dummies(df,columns=['Station_code'])
         df_main.head()
```

Out[62]:

	Locations	State	Temp	Do	Ph	Conductivity	Bod	NITRATENAN N+ NITRITENANN	Fecal_coliform	T(
0	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203	NAN	0.1	11	
1	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189	2	0.2	4953	
2	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179	1.7	0.1	3243	
3	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64	3.8	0.5	5382	
4	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	7.3	83	1.9	0.4	3428	

5 rows × 332 columns

df_main.describe()

```
In [63]: df_main.describe()
```

Out[63]:

	year	Station_code_1023	Station_code_1024	Station_code_1025	Station_code_102(
count	1991.000000	1991.000000	1991.000000	1991.000000	1991.000000
mean	2010.038172	0.004520	0.004520	0.004520	0.004520
std	3.057333	0.067098	0.067098	0.067098	0.067098
min	2003.000000	0.000000	0.000000	0.000000	0.000000
25%	2008.000000	0.000000	0.000000	0.000000	0.000000
50%	2011.000000	0.000000	0.000000	0.000000	0.000000
75%	2013.000000	0.000000	0.000000	0.000000	0.000000
max	2014.000000	1.000000	1.000000	1.000000	1.000000

8 rows × 322 columns

df_main.corr()

```
In [64]: df_main.corr()
```

Out[64]:

	year	Station_code_1023	Station_code_1024	Station_code_1025	Station_
year	1.000000	-0.000842	-0.000842	-0.000842	
Station_code_1023	-0.000842	1.000000	-0.004541	-0.004541	
Station_code_1024	-0.000842	-0.004541	1.000000	-0.004541	
Station_code_1025	-0.000842	-0.004541	-0.004541	1.000000	
Station_code_1026	-0.000842	-0.004541	-0.004541	-0.004541	
...	...	...	...	...	
Station_code_3471	0.037643	-0.002618	-0.002618	-0.002618	
Station_code_3473	0.037643	-0.002618	-0.002618	-0.002618	
Station_code_42	-0.017159	-0.004788	-0.004788	-0.004788	
Station_code_43	-0.017159	-0.004788	-0.004788	-0.004788	
Station_code_NAN	-0.411536	-0.017216	-0.017216	-0.017216	

322 rows × 322 columns

plt.figure(figsize=(10,8))

sns.heatmap(df_main.corr(),annot=True)

```
In [74]: plt.figure(figsize=(10,8))
         sns.heatmap(df_main.corr(),annot=True)
```

Out[74]: <AxesSubplot:>
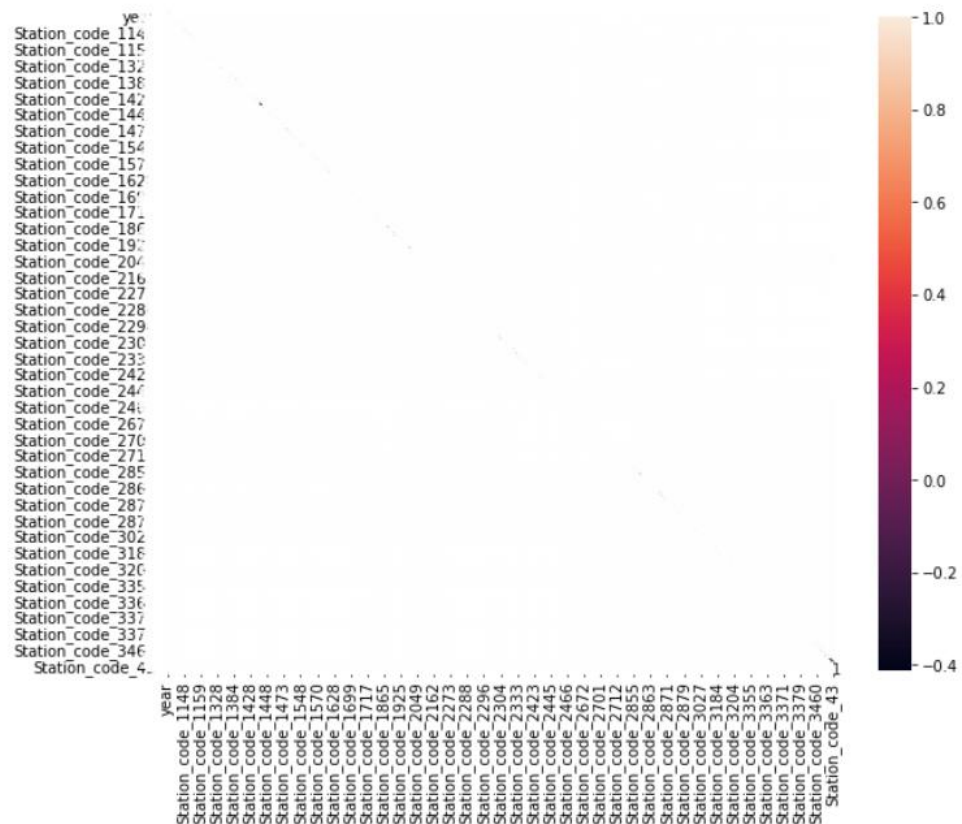
df_main.corr().year.sort_values(ascending=False)

```
In [66]: df_main.corr().year.sort_values(ascending=False)
```

Out[66]:
```
year                 1.000000
Station_code_3182    0.037643
Station_code_3186    0.037643
Station_code_3187    0.037643
Station_code_3181    0.037643
                        ...
Station_code_1246   -0.036832
Station_code_1438   -0.045161
Station_code_1861   -0.045161
Station_code_1435   -0.049874
Station_code_NAN    -0.411536
Name: year, Length: 322, dtype: float64
```

df_main.head()

```
In [67]: df_main.head()
```

Out[67]:

	Locations	State	Temp	Do	Ph	Conductivity	Bod	NITRATENAN N+ NITRITENANN	Fecal_coliform	Tc
0	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203	NAN	0.1	11	
1	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189	2	0.2	4953	
2	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179	1.7	0.1	3243	
3	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64	3.8	0.5	5382	
4	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	7.3	83	1.9	0.4	3428	

5 rows × 332 columns

X and y split

independent varibles-X

X=df_main.drop(columns=['Bod'],axis=1)

X.head()

```
In [75]: # independent varibles-X

         X=df_main.drop(columns=['Bod'],axis=1)
         X.head()
```

Out[75]:

	Locations	State	Temp	Do	Ph	Conductivity	NITRATENAN N+ NITRITENANN	Fecal_coliform	Total_co
0	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203	0.1	11	
1	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189	0.2	4953	
2	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179	0.1	3243	
3	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64	0.5	5382	
4	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	7.3	83	0.4	3428	

5 rows × 331 columns

y target-dependent variable

y=df_main.Bod

```
In [69]: # y target-dependent variable

         y=df_main.Bod
         y
```

```
Out[69]: 0        NAN
         1          2
         2        1.7
         3        3.8
         4        1.9
                 ...
         1986     2.7
         1987     2.6
         1988     1.2
         1989     1.3
         1990     1.1
         Name: Bod, Length: 1991, dtype: object
```

X.head()

```
In [71]: X.head()
```

Out[71]:

	Locations	State	Temp	Do	Ph	Conductivity	NITRATENAN N+ NITRITENANN	Fecal_coliform	Total_cc
0	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203	0.1	11	
1	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189	0.2	4953	
2	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179	0.1	3243	
3	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64	0.5	5382	
4	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	7.3	83	0.4	3428	

5 rows × 331 columns

scaling

from sklearn.preprocessing import scale

from sklearn.preprocessing import scale

X=pd.DataFrame(scale(X),columns=X.columns)

X.head()

Train test split

from sklearn.preprocessing import scale

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test =train_test_split(X_scaled,y, test_size=0.3,random_state=0)

X_train

X_train.shape

y_train.shape

X_test

X_test.shape