DAC-phase3-water-quality-analysis

October 18, 2023

DAC_Phase3 : Water Quality Analysis Project

The goal of the "Water Quality Analysis Project" in Phase 3, is to perform preprocessing and Exploratory Data Analysis by plotting graphs and getting insights.

Our approach involves,

1. finding correlation between the attributes of the dataset provided,
2. Handling missing values,
3. Getting comparative insights by using necessary plots for further processing and clear

Python Libraries

[1]:
```
#importing necessary libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# For visualizing Decision Tree
from sklearn import tree
```

Reading Dataset

[3]:
```
# Creating DataFrame by using .csv file
df = pd.read_csv("archive/water_potability.csv")
```

[4]:
```
df.head()
```

[4]:
```
         ph    Hardness        Solids  Chloramines     Sulfate  Conductivity  \
0       NaN  204.890455  20791.318981     7.300212  368.516441    564.308654
1  3.716080  129.422921  18630.057858     6.635246         NaN    592.885359
2  8.099124  224.236259  19909.541732     9.275884         NaN    418.606213
3  8.316766  214.373394  22018.417441     8.059332  356.886136    363.266516
```

```
4   9.092223  181.101509  17978.986339     6.546600  310.135738    398.410813

   Organic_carbon  Trihalomethanes  Turbidity  Potability
0       10.379783        86.990970   2.963135           0
1       15.180013        56.329076   4.500656           0
2       16.868637        66.420093   3.055934           0
3       18.436524       100.341674   4.628771           0
4       11.558279        31.997993   4.075075           0
```

[7]:
```python
# Descriptive Statistics
df.describe()
```

[7]:
```
                ph     Hardness        Solids  Chloramines      Sulfate  \
count  2785.000000  3276.000000   3276.000000  3276.000000  2495.000000
mean      7.080795   196.369496  22014.092526     7.122277   333.775777
std       1.594320    32.879761   8768.570828     1.583085    41.416840
min       0.000000    47.432000    320.942611     0.352000   129.000000
25%       6.093092   176.850538  15666.690297     6.127421   307.699498
50%       7.036752   196.967627  20927.833607     7.130299   333.073546
75%       8.062066   216.667456  27332.762127     8.114887   359.950170
max      14.000000   323.124000  61227.196008    13.127000   481.030642

       Conductivity  Organic_carbon  Trihalomethanes    Turbidity   Potability
count   3276.000000     3276.000000      3114.000000  3276.000000  3276.000000
mean     426.205111       14.284970        66.396293     3.966786     0.390110
std       80.824064        3.308162        16.175008     0.780382     0.487849
min      181.483754        2.200000         0.738000     1.450000     0.000000
25%      365.734414       12.065801        55.844536     3.439711     0.000000
50%      421.884968       14.218338        66.622485     3.955028     0.000000
75%      481.792304       16.557652        77.337473     4.500320     1.000000
max      753.342620       28.300000       124.000000     6.739000     1.000000
```

[8]:
```python
# Information about dataframe
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3276 entries, 0 to 3275
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   ph               2785 non-null   float64
 1   Hardness         3276 non-null   float64
 2   Solids           3276 non-null   float64
 3   Chloramines      3276 non-null   float64
 4   Sulfate          2495 non-null   float64
 5   Conductivity     3276 non-null   float64
 6   Organic_carbon   3276 non-null   float64
```

```
 7   Trihalomethanes  3114 non-null   float64
 8   Turbidity        3276 non-null   float64
 9   Potability       3276 non-null   int64
```

dtypes: float64(9), int64(1)
memory usage: 256.1 KB

Correlation Between Features

[10]:
```python
#correlation table
df.corr()
```

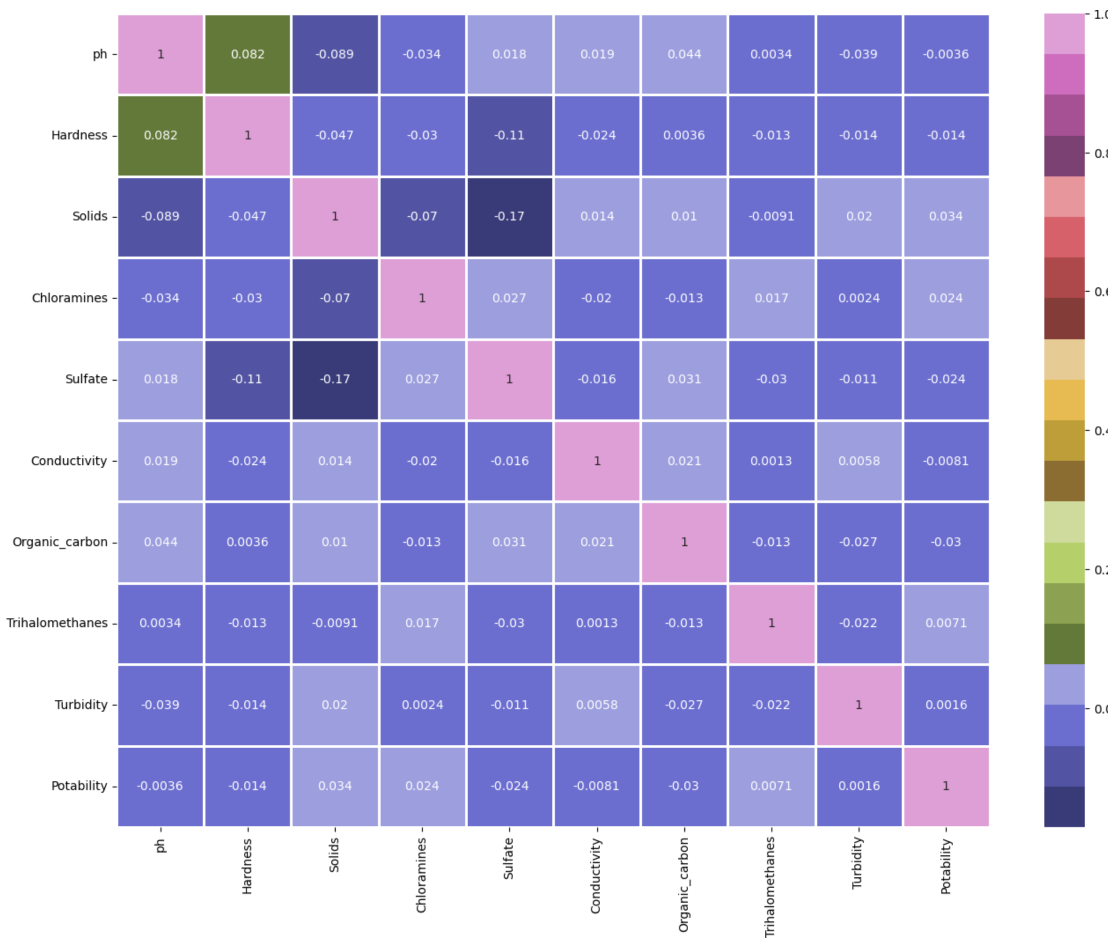
[10]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
ph	1.000000	0.082096	-0.089288	-0.034350	0.018203	0.018614	0.043503	0.003354	-0.039057	-0.003556
Hardness	0.082096	1.000000	-0.046899	-0.030054	-0.106923	-0.023915	0.003610	-0.013013	-0.014449	-0.013837
Solids	-0.089288	-0.046899	1.000000	-0.070148	-0.171804	0.013831	0.010242	-0.009143	0.019546	0.033743
Chloramines	-0.034350	-0.030054	-0.070148	1.000000	0.027244	-0.020486	-0.012653	0.017084	0.002363	0.023779
Sulfate	0.018203	-0.106923	-0.171804	0.027244	1.000000	-0.016121	0.030831	-0.030274	-0.011187	-0.023577
Conductivity	0.018614	-0.023915	0.013831	-0.020486	-0.016121	1.000000	0.020966	0.001285	0.005798	-0.008128
Organic_carbon	0.043503	0.003610	0.010242	-0.012653	0.030831	0.020966	1.000000	-0.013274	-0.027308	-0.030001
Trihalomethanes	0.003354	-0.013013	-0.009143	0.017084	-0.030274	0.001285	-0.013274	1.000000	-0.022145	0.007130
Turbidity	-0.039057	-0.014449	0.019546	0.002363	-0.011187	0.005798	-0.027308	-0.022145	1.000000	0.001581
Potability	-0.003556	-0.013837	0.033743	0.023779	-0.023577	-0.008128	-0.030001	0.007130	0.001581	1.000000

[64]:
```python
#correlation by using clustermap
#sns.heatmap(df.corr(), cmap='flag')

fig, ax = plt.subplots(figsize=(16, 12))
sns.heatmap(df.corr(), cmap='tab20b',annot=True,linewidths='0.8',ax=ax)
```

[64]: <Axes: >



Preprocessing: Missing Value

[65]:
```python
#missing value counts
df.isnull().sum()
```

[65]:
```
ph                 491
 Hardness            0
 Solids              0
 Chloramines         0
 Sulfate           781
```

```
Conductivity         0
Organic_carbon       0
Trihalomethanes    162
Turbidity            0
Potability           0
dtype: int64
```

[67]:
```python
df['ph'].fillna(value = df['ph'].mean(), inplace = True)
```

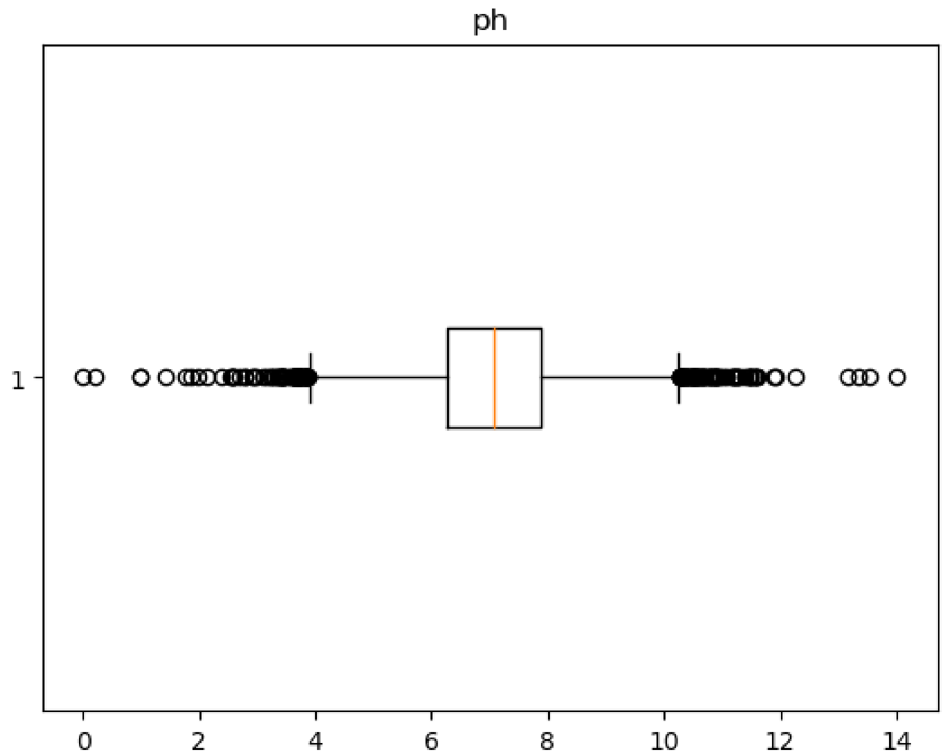
[69]:
```python
df['Sulfate'].fillna(value = df['Sulfate'].mean(), inplace = True)
df['Trihalomethanes'].fillna(value = df['Trihalomethanes'].mean(), inplace =␣
↪True)
```

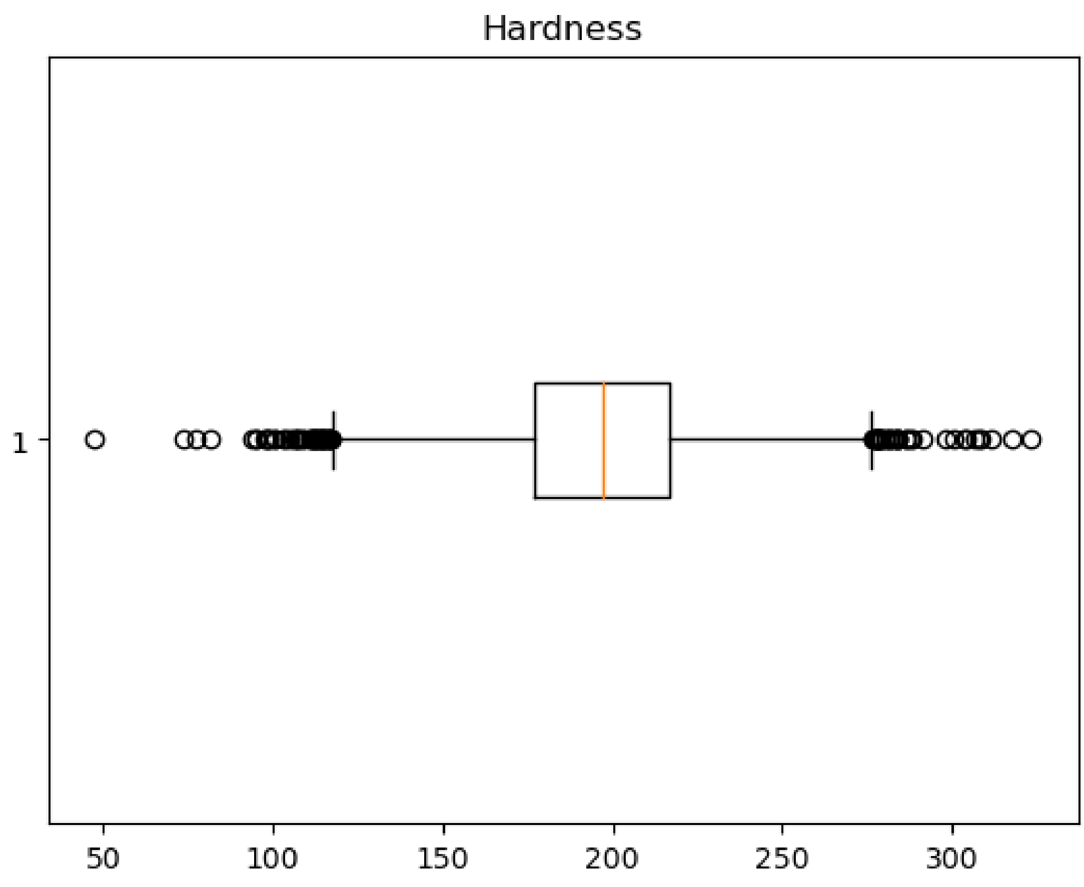
[70]:
```python
# Check again the missing values
df.isnull().sum()
```

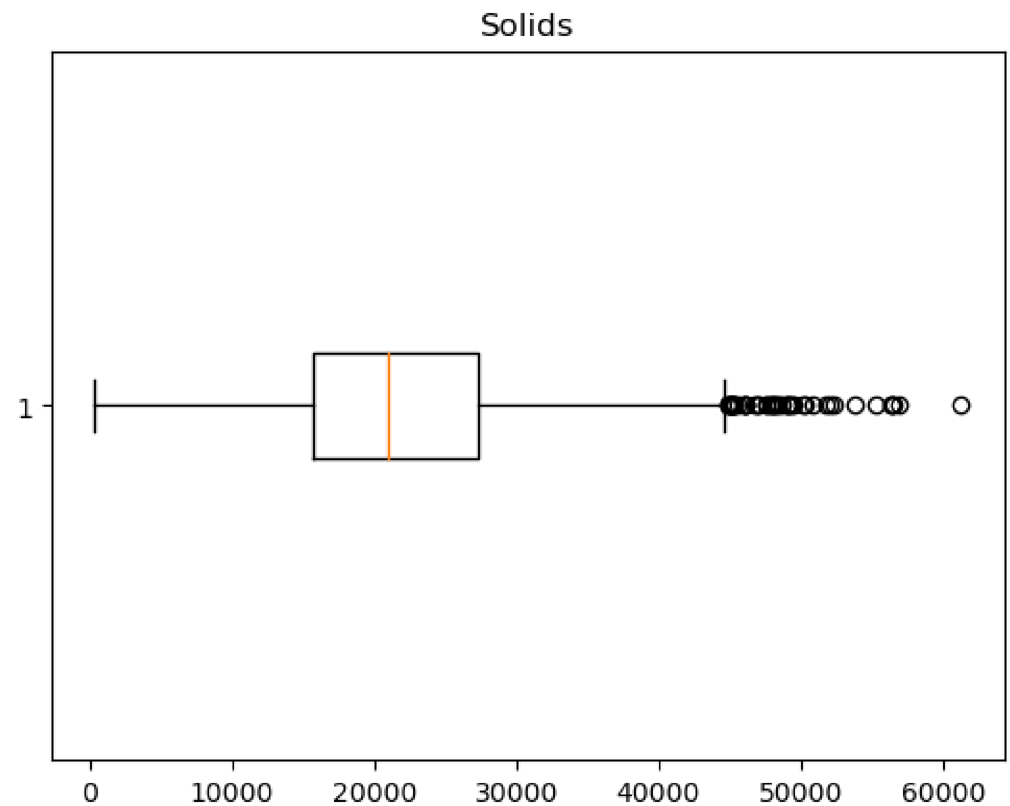
[70]:
```
ph                 0
Hardness           0
Solids             0
Chloramines        0
Sulfate            0
Conductivity       0
Organic_carbon     0
Trihalomethanes    0
Turbidity          0
Potability         0
dtype: int64
```

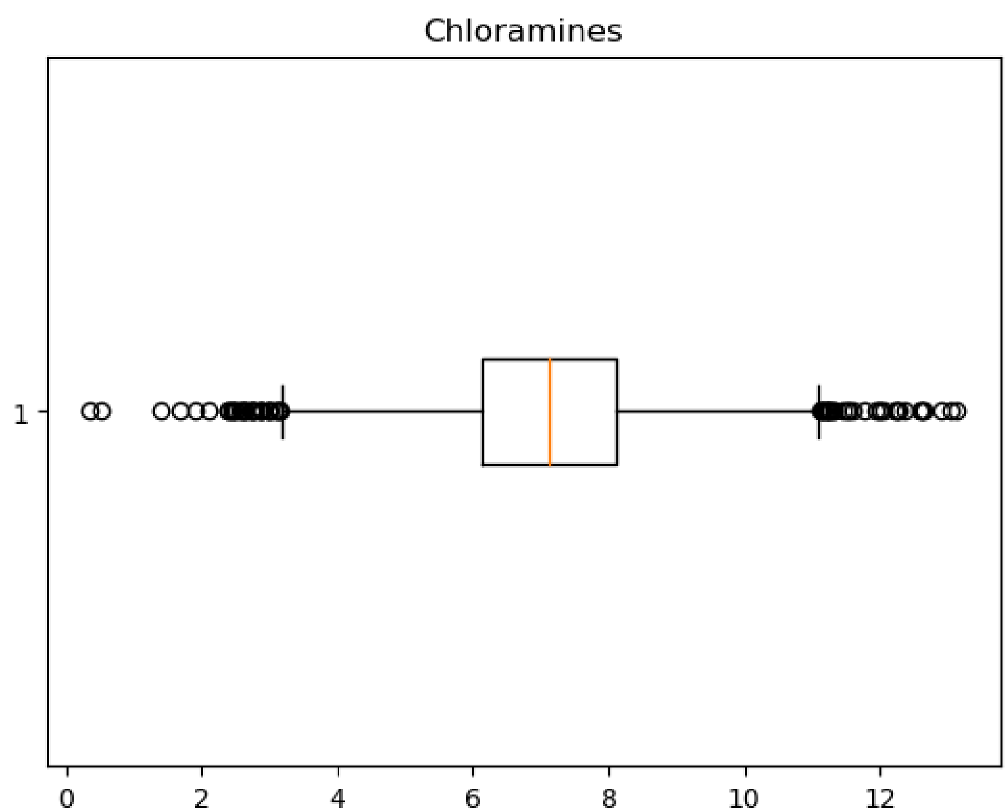
3.1 Checking for outliers using boxplot

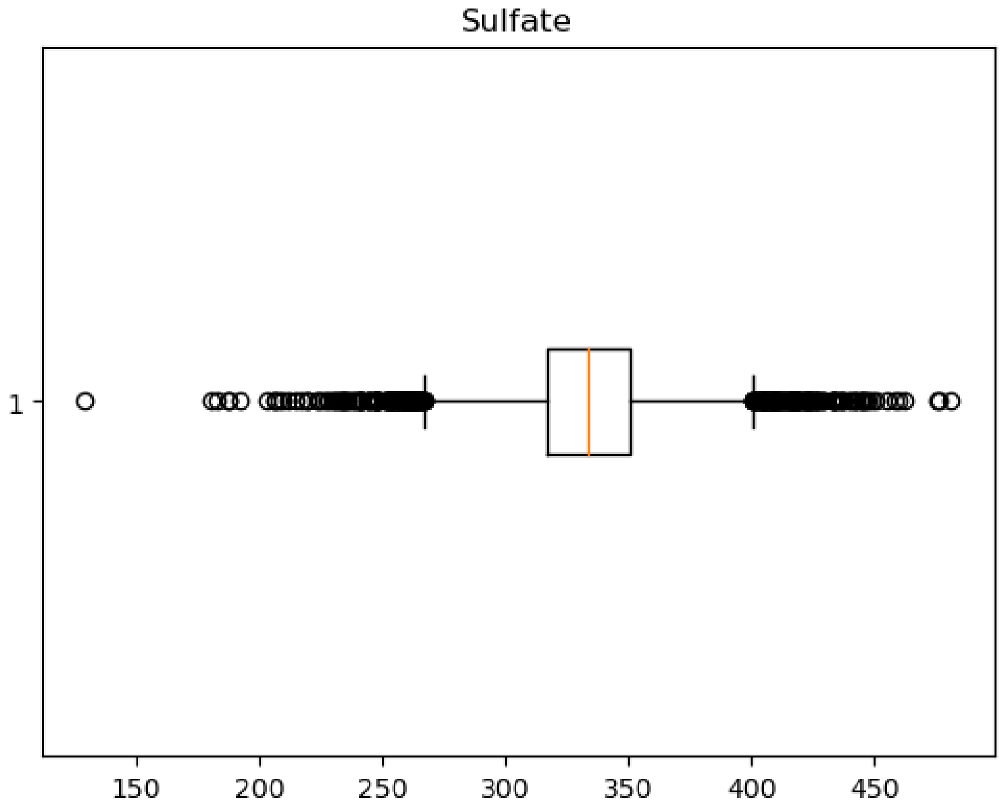
[106]:
```python
for col in df.columns:
    plt.boxplot(df[col], vert=False)
    plt.title(col)
    plt.show()
```

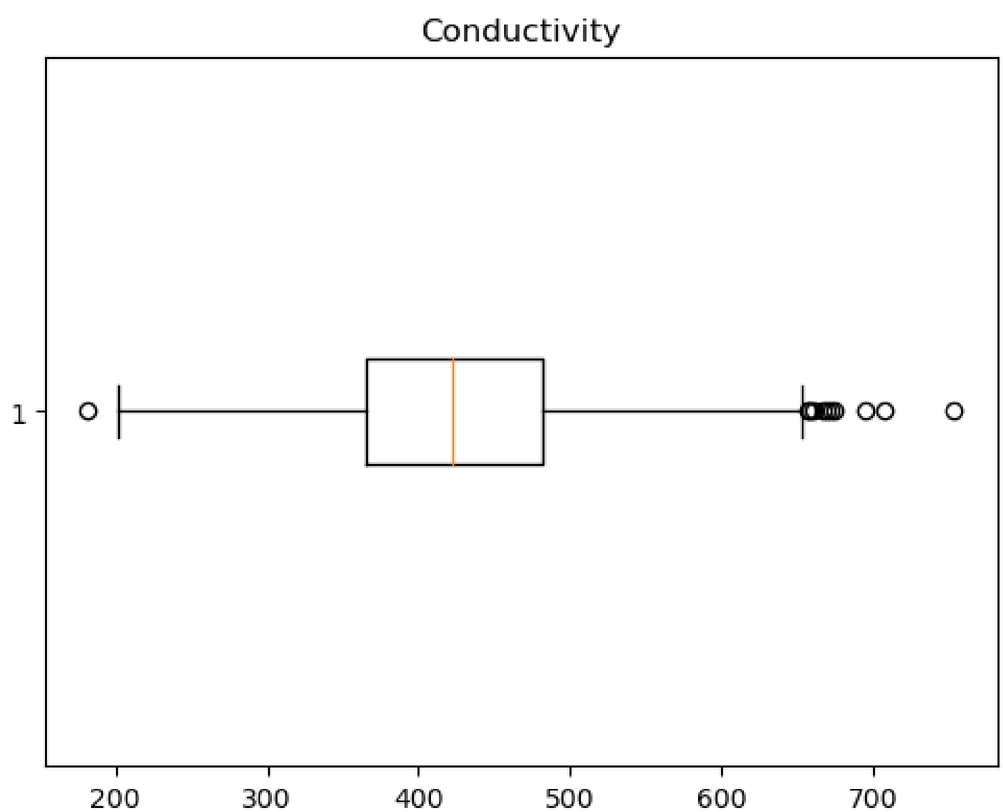

ph
1
0
2
4
6
8
10
12
14

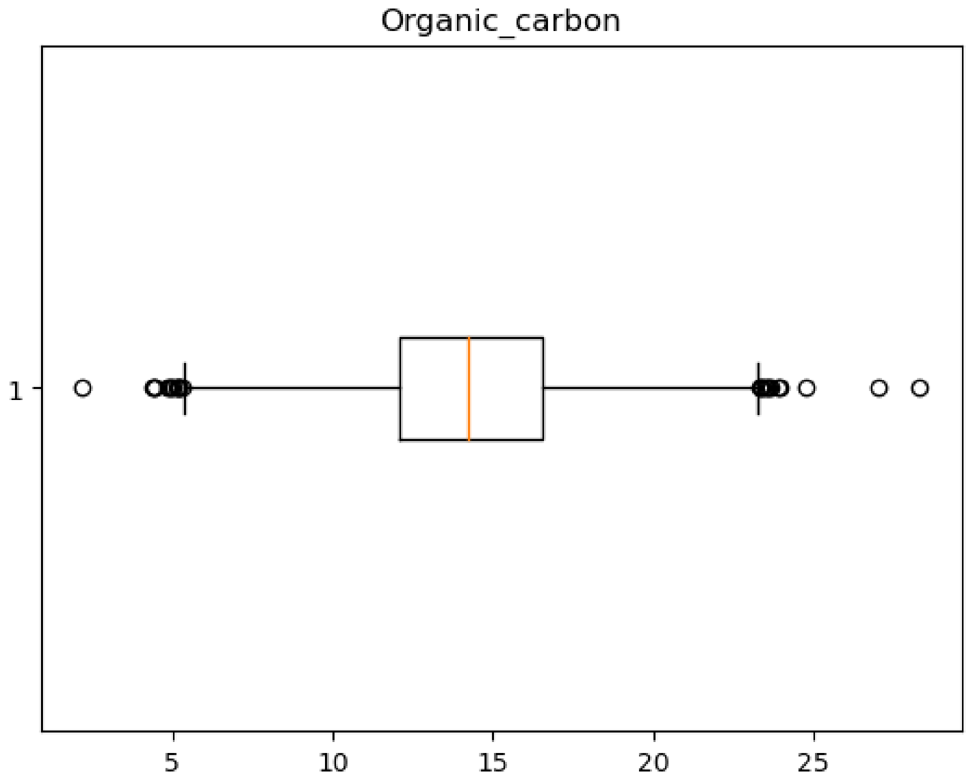

Hardness
1
50
100
150
200
250
300

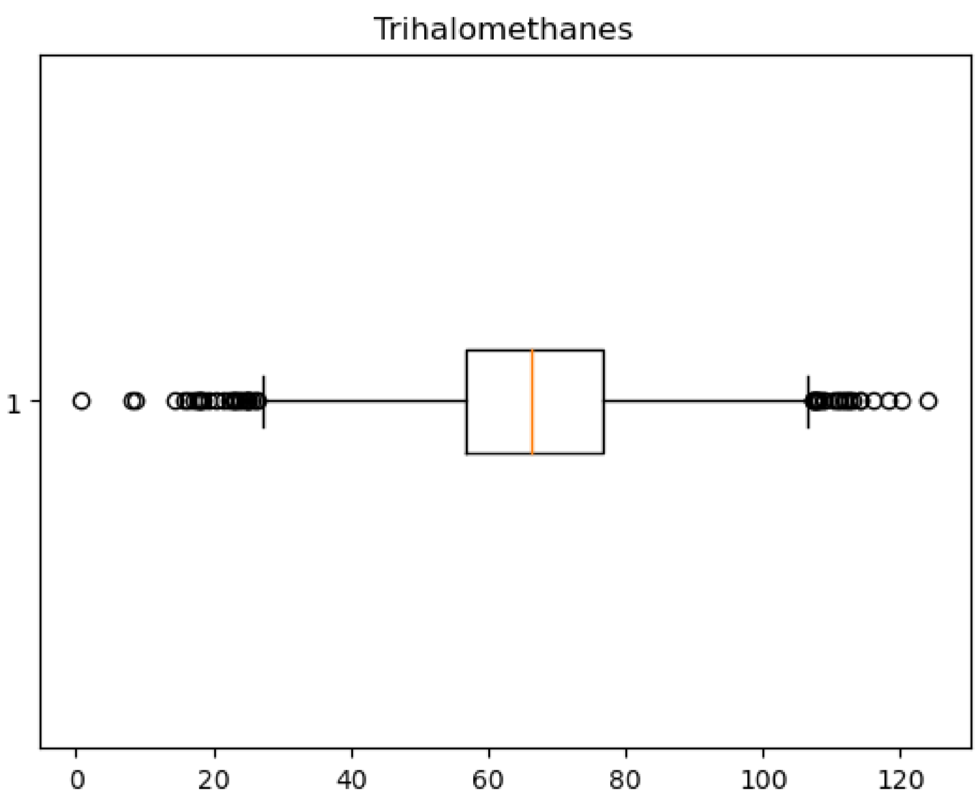

Solids
1
0
10000
20000
30000
40000
50000
60000

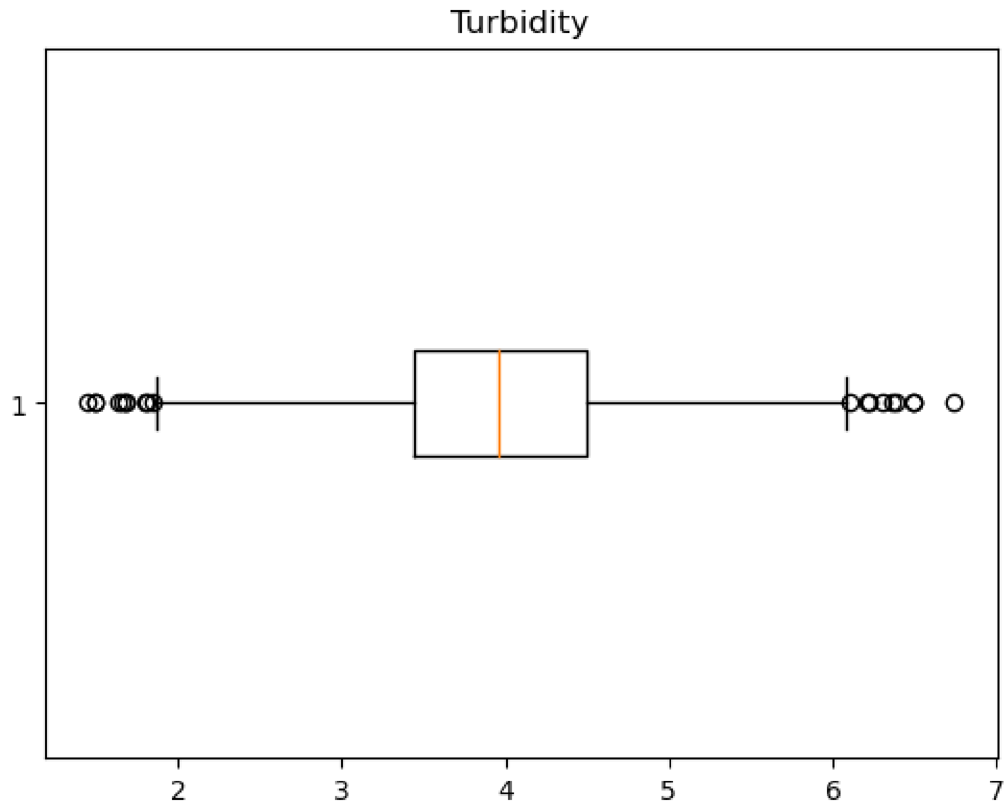

Chloramines
1
0
2
4
6
8
10
12


Sulfate
1
150
200
250
300
350
400
450


Conductivity
1
200
300
400
500
600
700

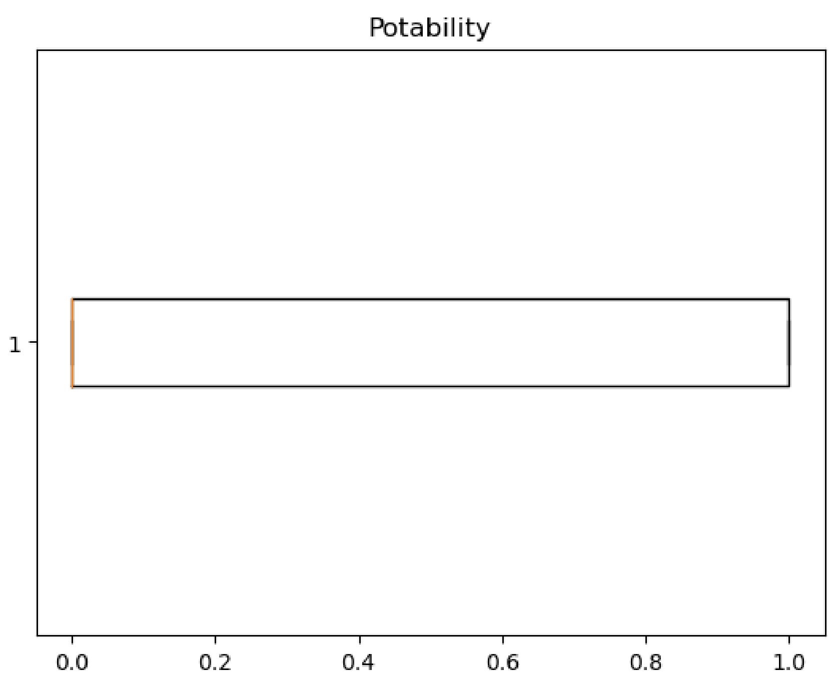

Organic_carbon
1
5
10
15
20
25

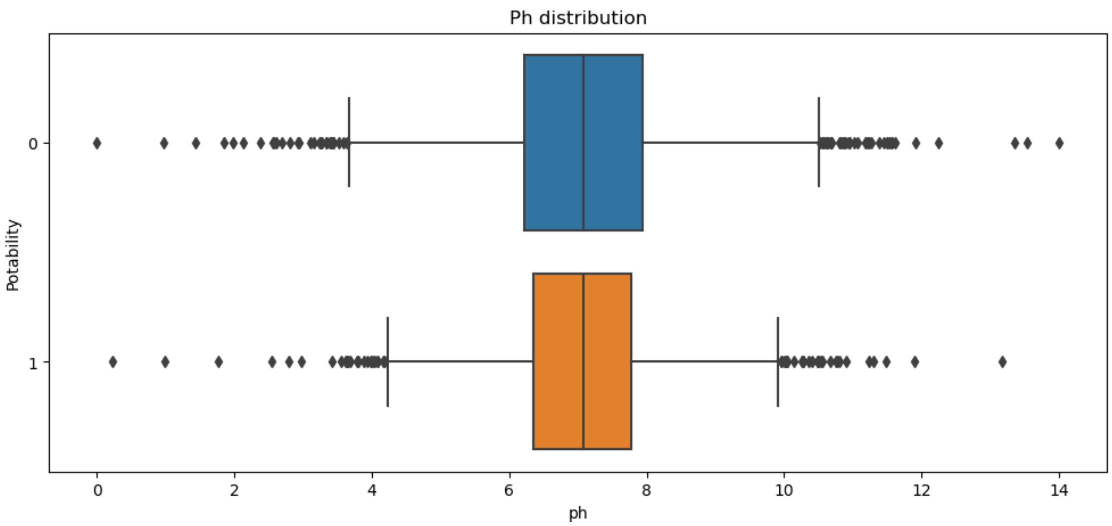

Trihalomethanes
1
0
20
40
60
80
100
120



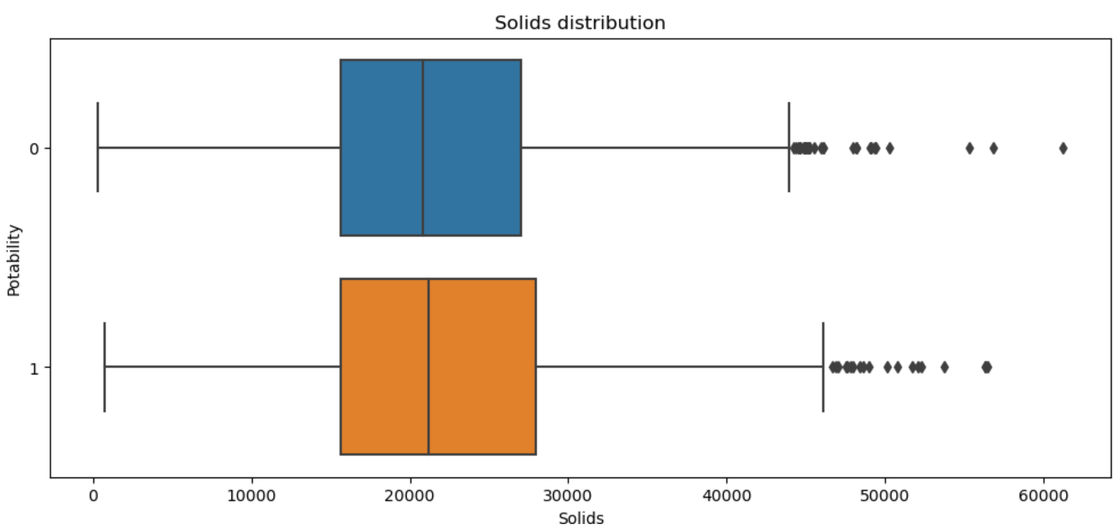
Checking for other relations

```
fig,ax   = plt.subplots(figsize = (12,5))
sns.boxplot(data =df, x = 'ph', y = 'Potability', orient = 'h').set(title = 'Ph␣
↪distribution');
```

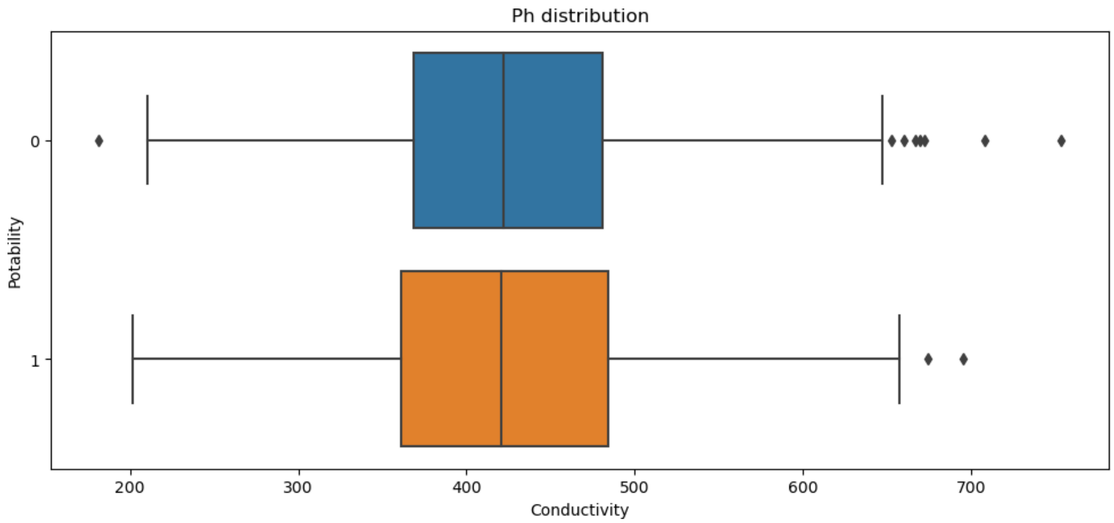





```
fig,ax  = plt.subplots(figsize = (12,5))
sns.boxplot(data =df, x = 'Solids', y = 'Potability', orient = 'h').set(title =␣
↪'Solids distribution');
```




```
fig,ax  = plt.subplots(figsize = (12,5))
sns.boxplot(data =df, x = 'Conductivity', y = 'Potability', orient = 'h').
↪set(title = 'Ph distribution');
```



Conclusions

-> *From the correlation heatmap plotted earlier, its clear that the pf level of the water and the hardness of the water are highly correlated.*

-> The Outliers of each attribute in the dataset is properly visualized using boxplot,

-> *Sulfate has so many outliers as well as less correlated with most other attributes, thus it can be deleted if not needed.*

-> *ph, Chloramine, solids also have many outliers*

From other three comparative boxplot using ph and probability, it is clearthat water which harmful for drinking and water which safe for drinking are almost slightly equally distributed in this samples