Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа №1
по дисциплине
«Методы машинного обучения»
на тему

«Создание "истории о данных" (Data Storytelling)»

Выполнил: Ван Пэй
студент группы: ИУ5-22М

Москва — 2021 г.

1. Цель лабораторной работы

Изучение различных методов визуализация данных и создание истории на основе данных.

2. Задание

2.1 Выбрать набор данных (датасет).

2.2 Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:

История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.

На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.

Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.

Выбор графиков должен быть обоснован использованием методологии data-to-viz. Рекомендуется учитывать типичные ошибки построения выбранного вида графика по методологии data-to-viz. Если методология Вами отвергается, то просьба обосновать Ваше решение по выбору графика.

История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.

2.3 Сформировать отчет и разместить его в своем репозитории на github.

3. Ход выполнения работы

3.1. Текстовое описание набора данных

Доступ к безопасной питьевой воде имеет важное значение для здоровья, является основным правом человека и компонентом эффективной политики по охране здоровья. Это важно как вопрос здоровья и развития на национальном,

региональном и местном уровнях. В некоторых регионах было показано, что инвестиции в водоснабжение и санитарию могут принести чистую экономическую выгоду, поскольку сокращение неблагоприятных последствий для здоровья и затрат на здравоохранение перевешивают затраты на проведение мероприятий.

Цель работы: проверка безопасности питьевой воды по:

1) PH;

2) Жесткость воды;

3) Твердые вещества (общее количество растворенных твердых веществ - TDS);

4) Мутность и т.д.

3.2. Основные характеристики набора данных

Подключим все необходимые библиотеки:

In [43]:
```
import numpy as np
import pandas as pd
import os
import plotly
import plotly.express as px
```

In [44]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [47]:
```
import missingno as msno
msno.bar(data)
```

Загрузим непосредственно данные:

In [45]:
```
data=pd.read_csv("C:/Users/王沛/Desktop/water_potability.csv")
data
```

Показание информаций о температуре суши и океана на земле за последние несколько лет:

Out[45]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Orgar
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	
...	...	...	...	...	...	...	
3271	4.668102	193.681735	47580.991603	7.166639	359.948574	526.424171	
3272	7.808856	193.553212	17329.802160	8.061362	NaN	392.449580	
3273	9.419510	175.762646	33155.578218	7.350233	NaN	432.044783	
3274	5.126763	230.603758	11983.869376	6.303357	NaN	402.883113	
3275	7.874671	195.102299	17404.177061	7.509306	NaN	327.459760	

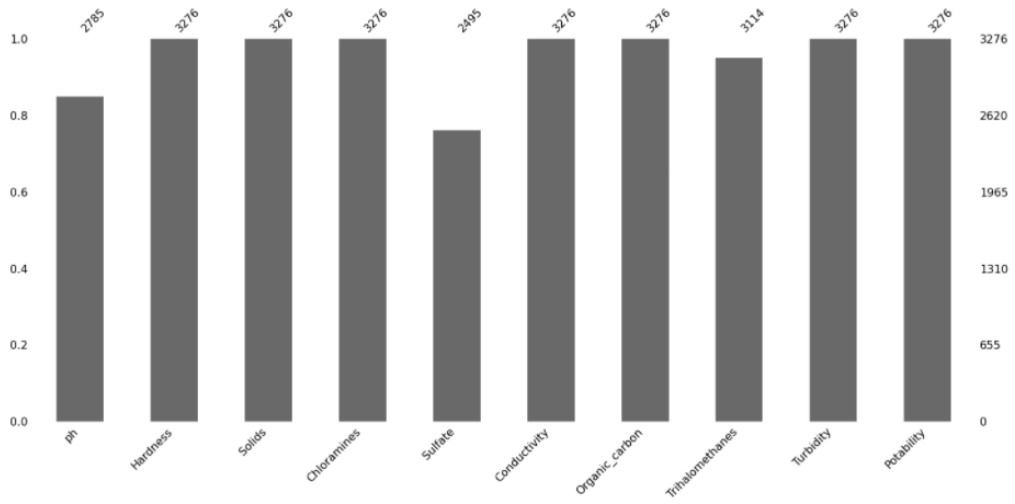
3276 rows × 10 columns

In [46]:
```
data.shape
```

Out[46]: (3276, 10)

In [47]:
```
import missingno as msno
msno.bar(data)
```

Out[47]: <AxesSubplot:>



In [48]:
```
data.isnull().sum()
```

Out[48]:
```
ph                491
Hardness            0
Solids              0
Chloramines         0
Sulfate           781
Conductivity        0
Organic_carbon      0
Trihalomethanes   162
Turbidity           0
Potability          0
dtype: int64
```

In [49]:
```
data.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
```

In [50]:
```
data['ph']=data['ph'].fillna(data['ph'].mean())
data['Sulfate']=data['Sulfate'].fillna(data['Sulfate'].mean())
data['Trihalomethanes']=data['Trihalomethanes'].fillna(data['Trihalomethanes'].mean())
```

In [51]:
```
data
```

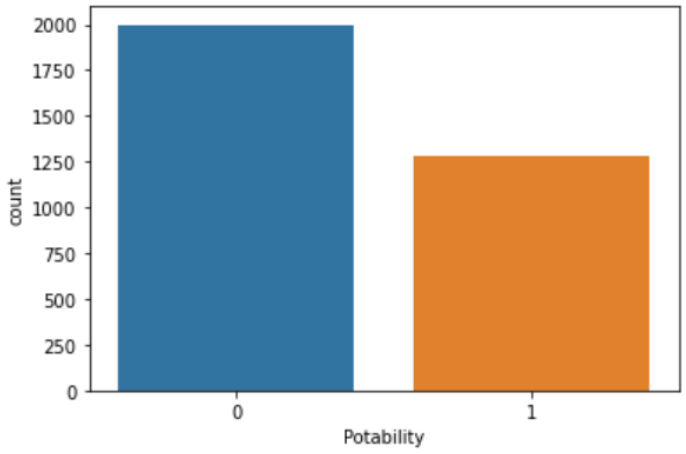
Out[51]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon
0	7.080795	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379783
1	3.716080	129.422921	18630.057858	6.635246	333.775777	592.885359	15.180013
2	8.099124	224.236259	19909.541732	9.275884	333.775777	418.606213	16.868637
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279
...	...	...	...	...	...	...	...
3271	4.668102	193.681735	47580.991603	7.166639	359.948574	526.424171	13.894419
3272	7.808856	193.553212	17329.802160	8.061362	333.775777	392.449580	19.903225
3273	9.419510	175.762646	33155.578218	7.350233	333.775777	432.044783	11.039070
3274	5.126763	230.603758	11983.869376	6.303357	333.775777	402.883113	11.168946
3275	7.874671	195.102299	17404.177061	7.509306	333.775777	327.459760	16.140368

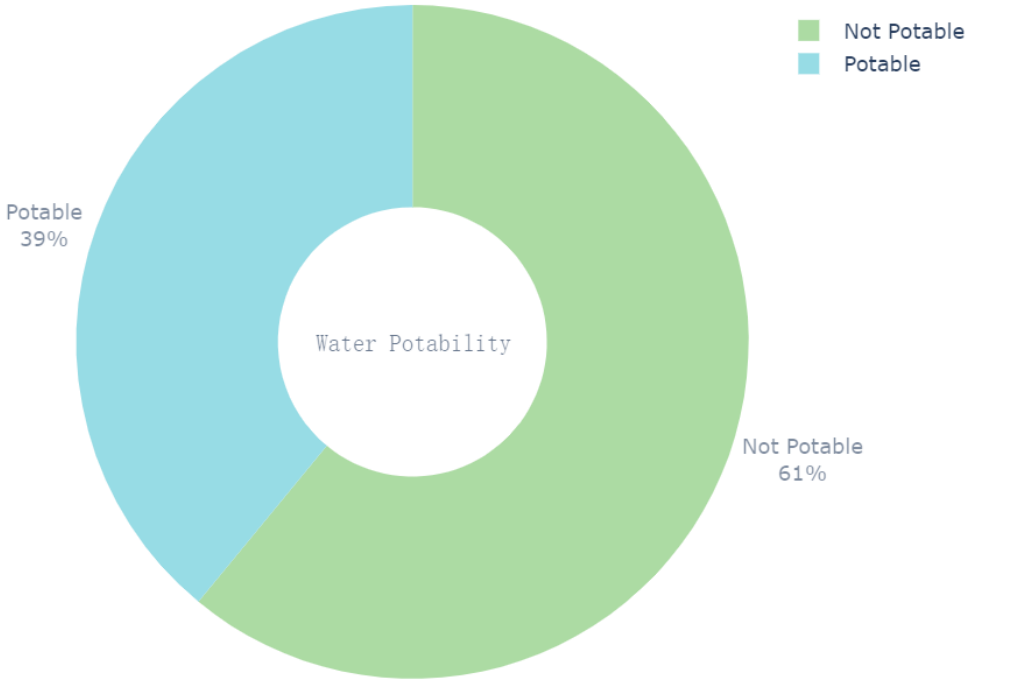
3276 rows × 10 columns

In [52]:
```
sns.countplot(x=data["Potability"])
```

Out[52]: <AxesSubplot:xlabel='Potability', ylabel='count'>



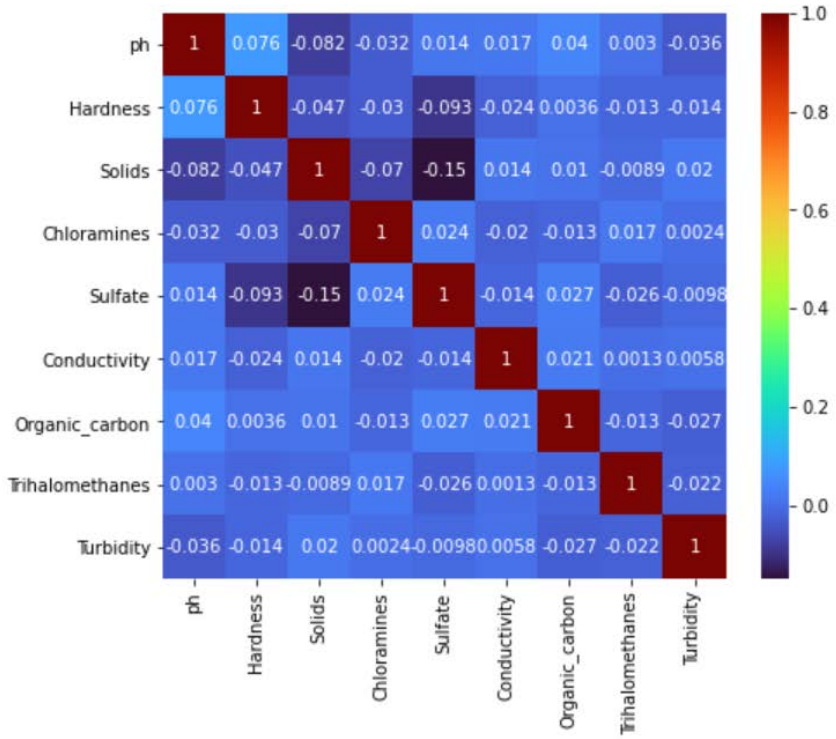
In [53]:
```
colors_blue = ["#132C33", "#264D58", '#17869E', '#51C4D3', '#B4DBE9']
colors_green = ['#01411C','#4B6F44','#4F7942','#74C365','#D0F0C0']
d= pd.DataFrame(data['Potability'].value_counts())
fig = px.pie(d,values='Potability',names=['Not Potable','Potable'],hole=0.4,opacity=0.6,
             color_discrete_sequence=[colors_green[3],colors_blue[3]],
              labels={'label':'Potability','Potability':'No. Of Samples'})

fig.add_annotation(text='Water Potability',
                   x=0.5,y=0.5,showarrow=False,font_size=14,opacity=0.7,font_family='mono

fig.update_traces(textposition='outside', textinfo='percent+label')

fig.show()
```



In [54]:
```
features_num = ['ph', 'Hardness', 'Solids', 'Chloramines', 'Sulfate',
                'Conductivity', 'Organic_carbon', 'Trihalomethanes',
                'Turbidity']
```

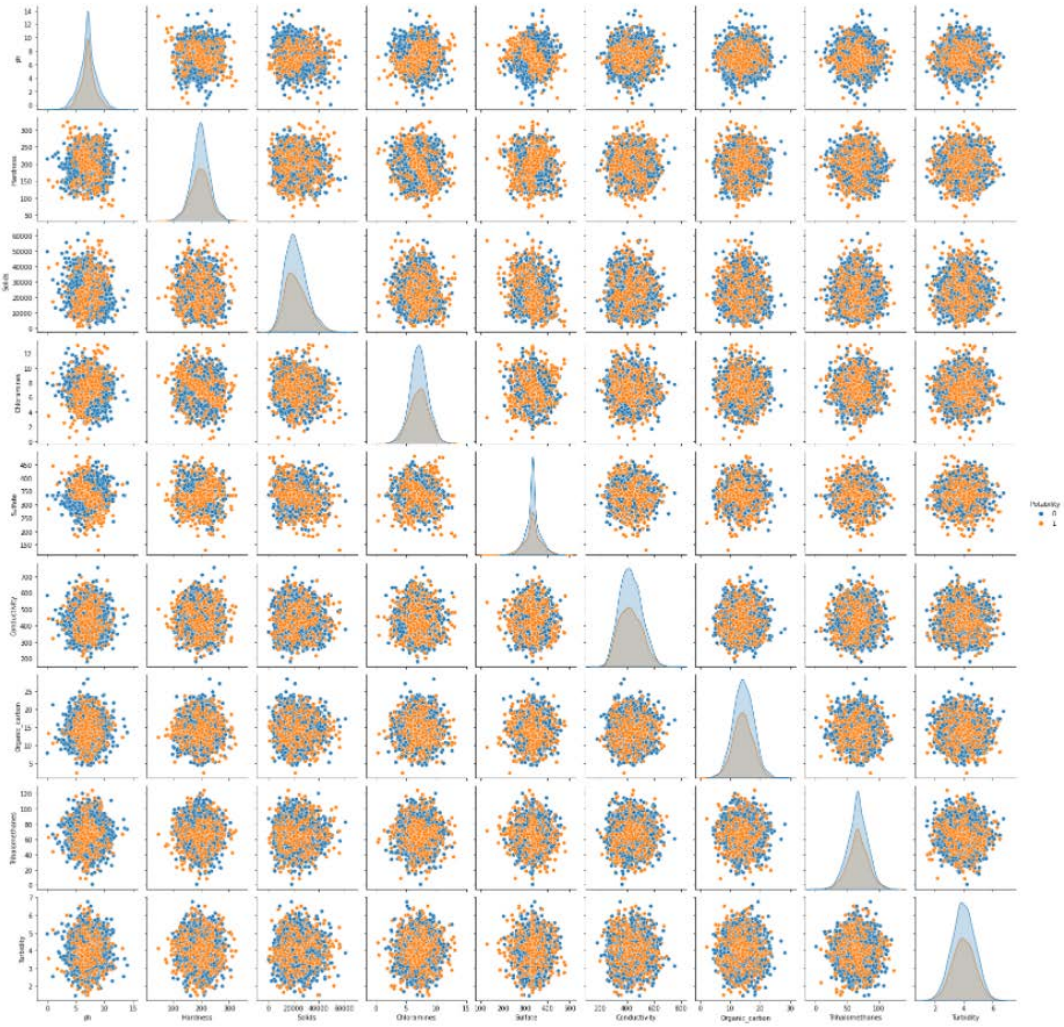
In [55]:
```
corr = data[features_num].corr()

plt.figure(figsize=(16,6))
ax1 = plt.subplot(1,2,1)
sns.heatmap(corr, annot=True, cmap='turbo')
```

Out[55]: <AxesSubplot:>

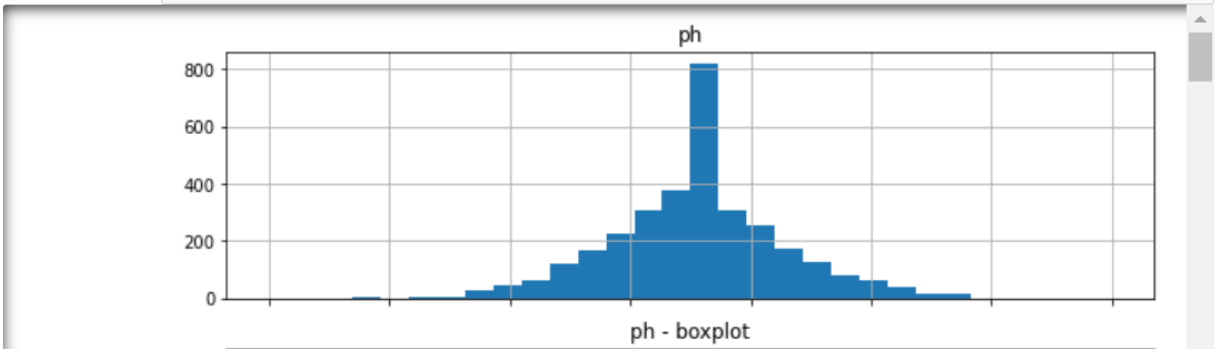


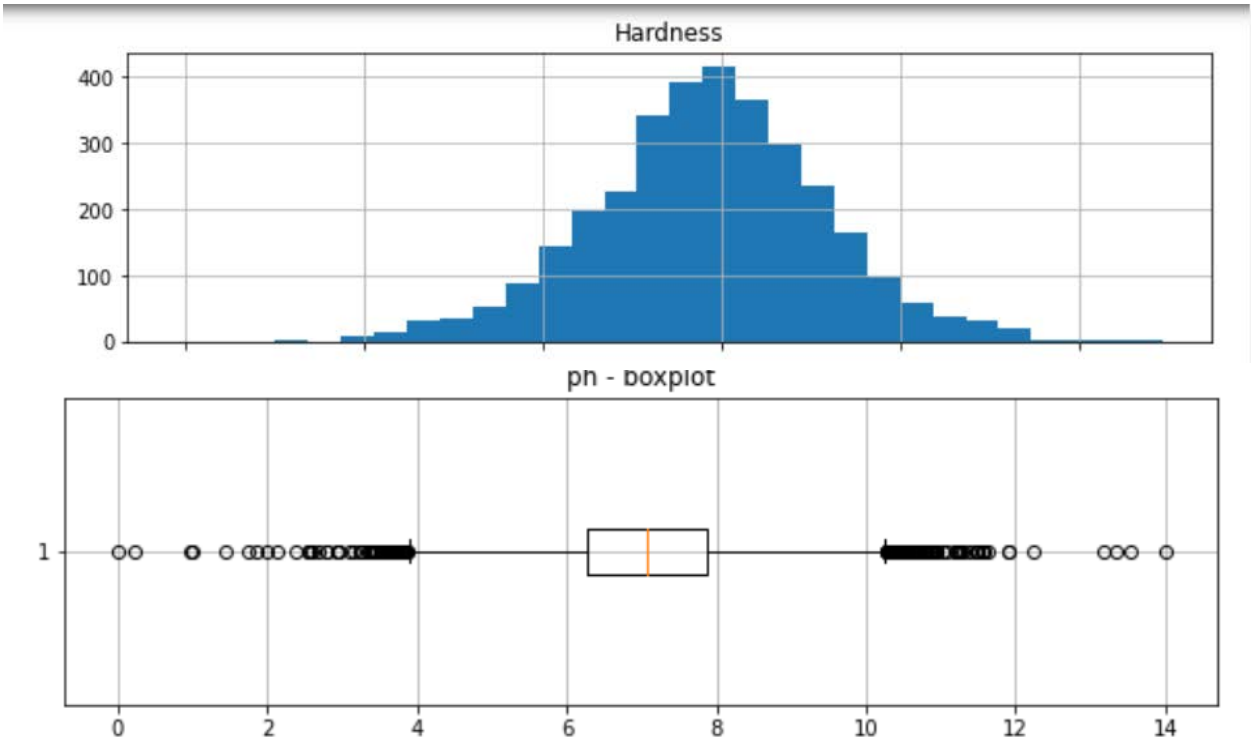
In [56]:
```
sns.pairplot(data=data,hue="Potability")
```

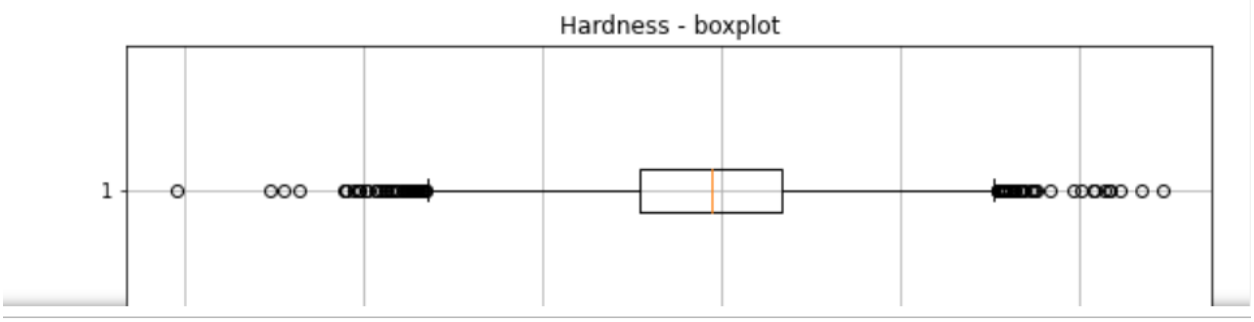
Out[56]: <seaborn.axisgrid.PairGrid at 0x21a532b1640>

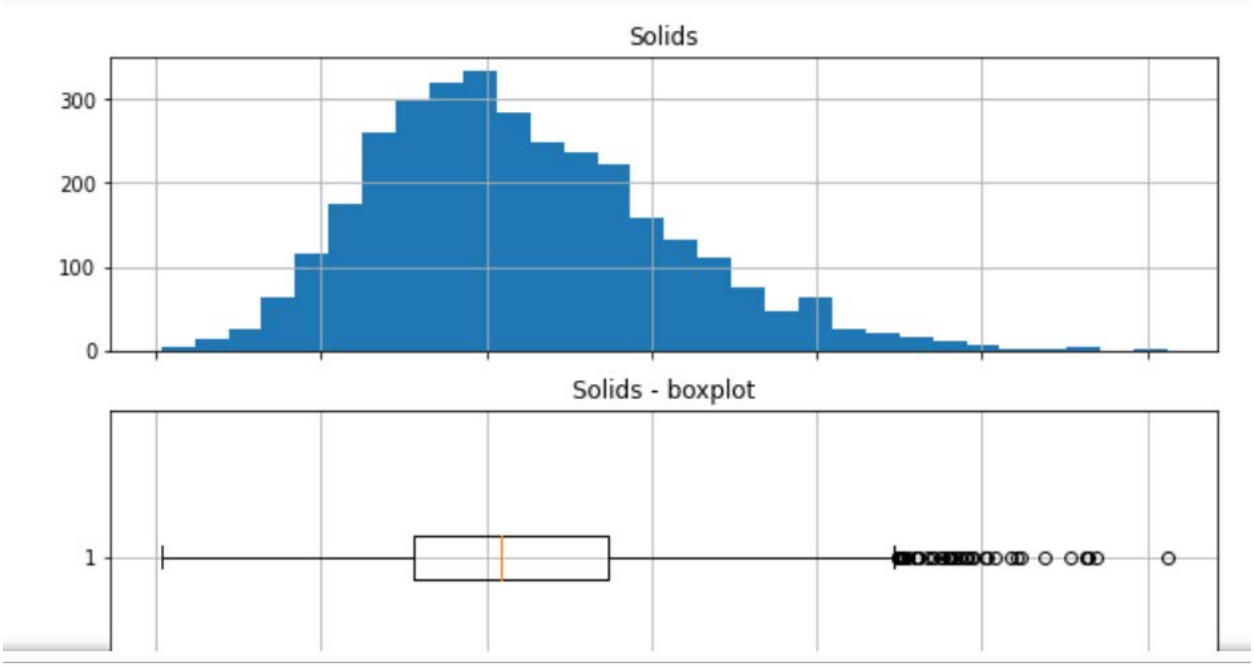


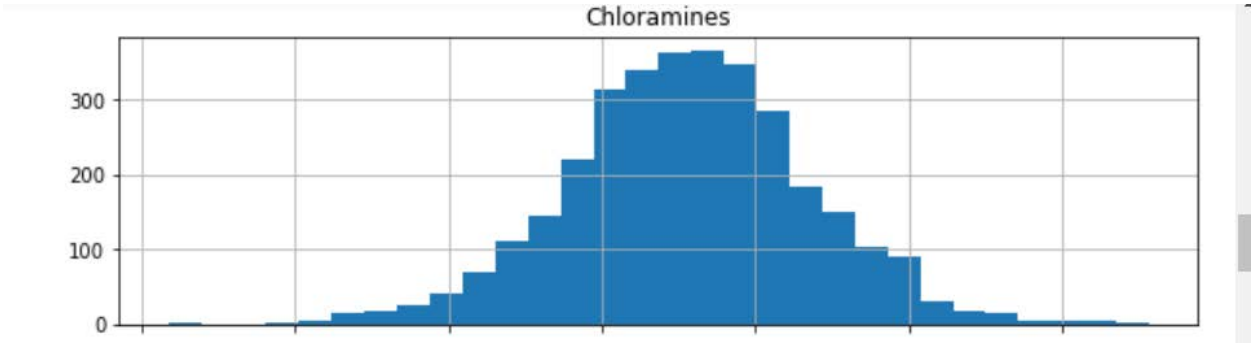
In [57]:
```
# plot distribution of numerical features
for f in features_num:
    fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10,6), sharex=True)
    ax1.hist(data[f], bins=30)
    ax1.grid()
    ax1.set_title(f)
    # for boxplot we need to remove the NaNs first
    feature_wo_nan = data[~np.isnan(data[f])][f]
    ax2.boxplot(feature_wo_nan, vert=False)
    ax2.grid()
    ax2.set_title(f + ' - boxplot')
    plt.show()
```

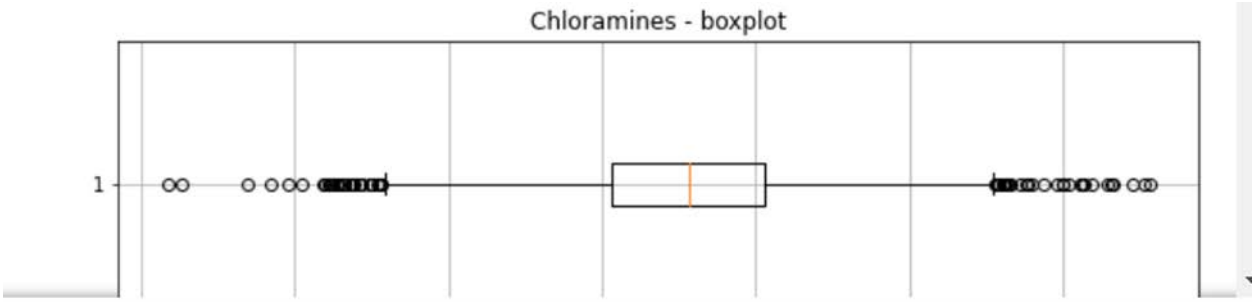


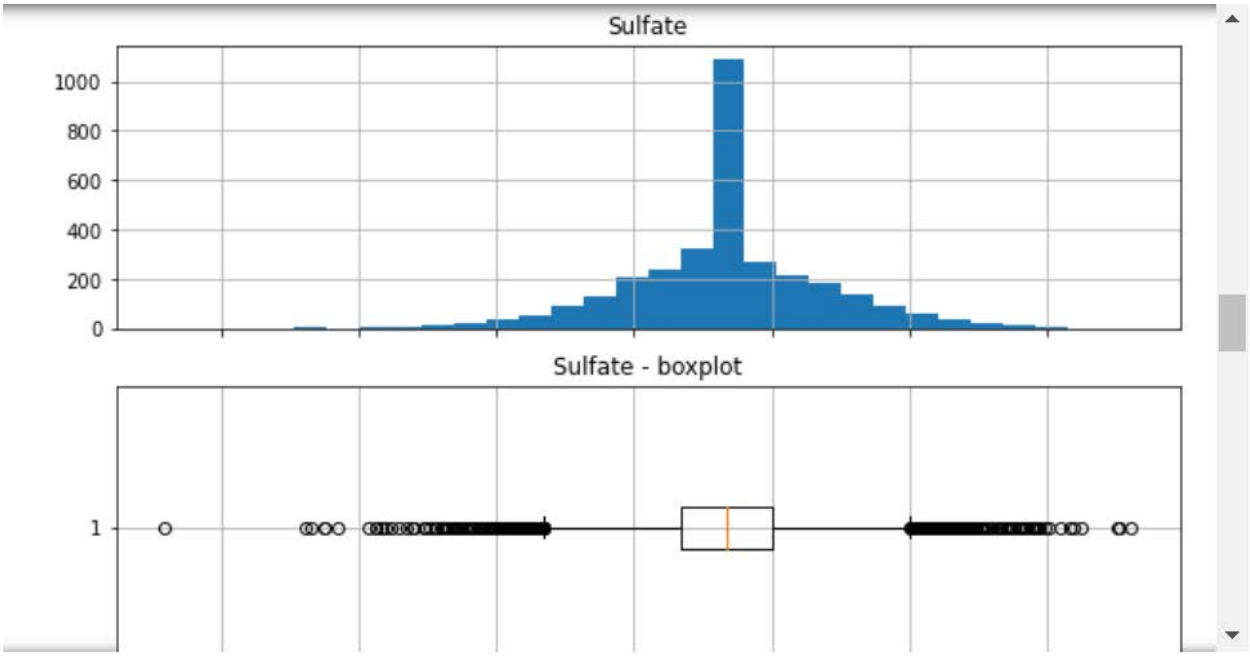


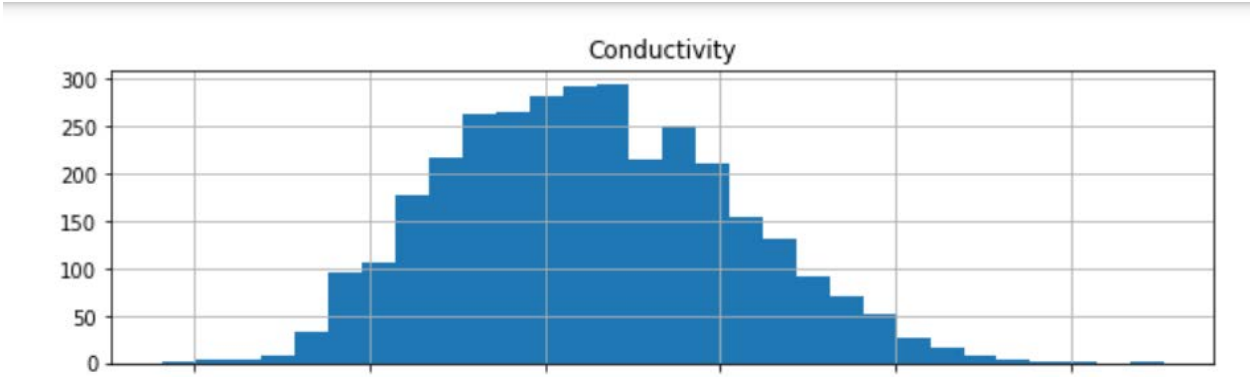

Hardness - boxplot
1

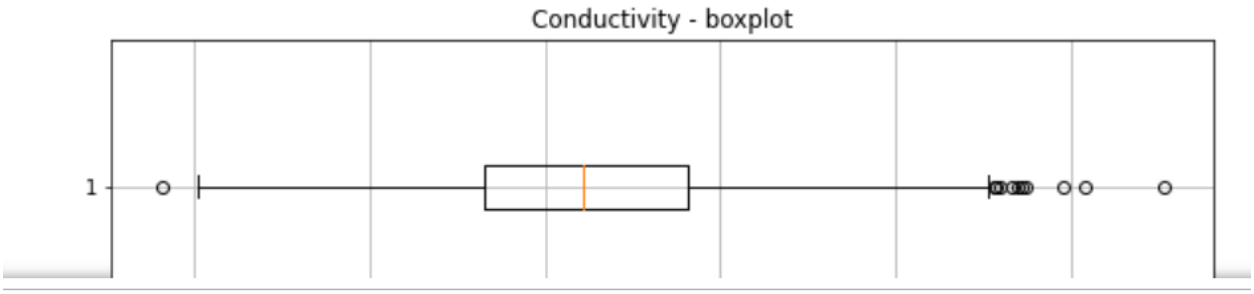

Solids
300
200
100
0
Solids - boxplot
1

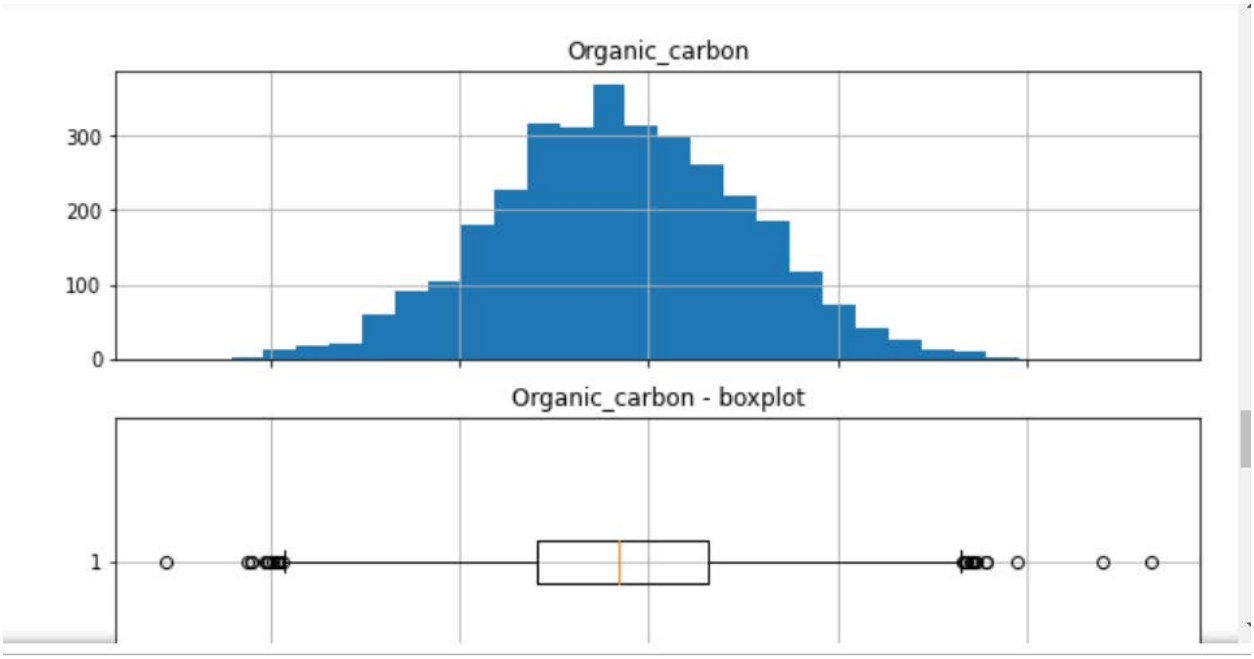

Chloramines
300
200
100
0

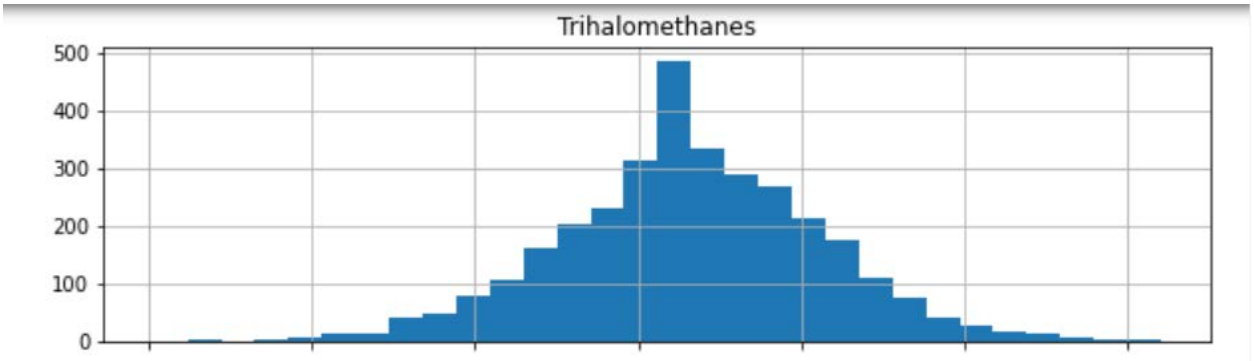

Chloramines - boxplot
1

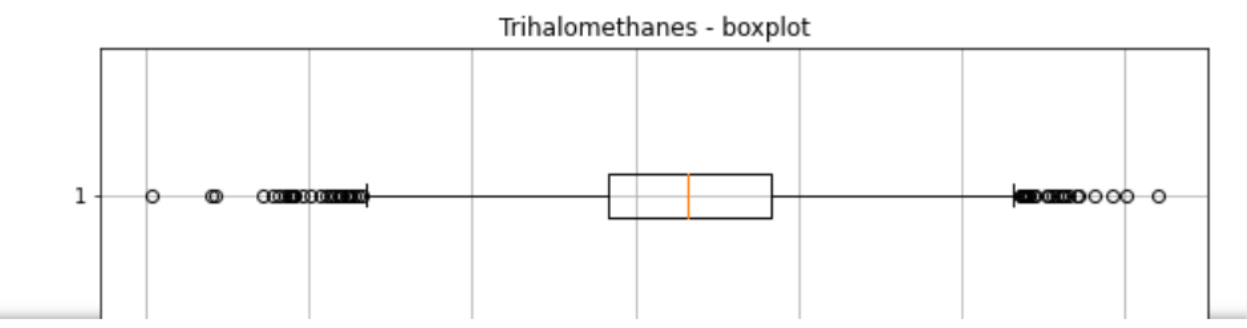

Sulfate
1000
800
600
400
200
0
Sulfate - boxplot
1

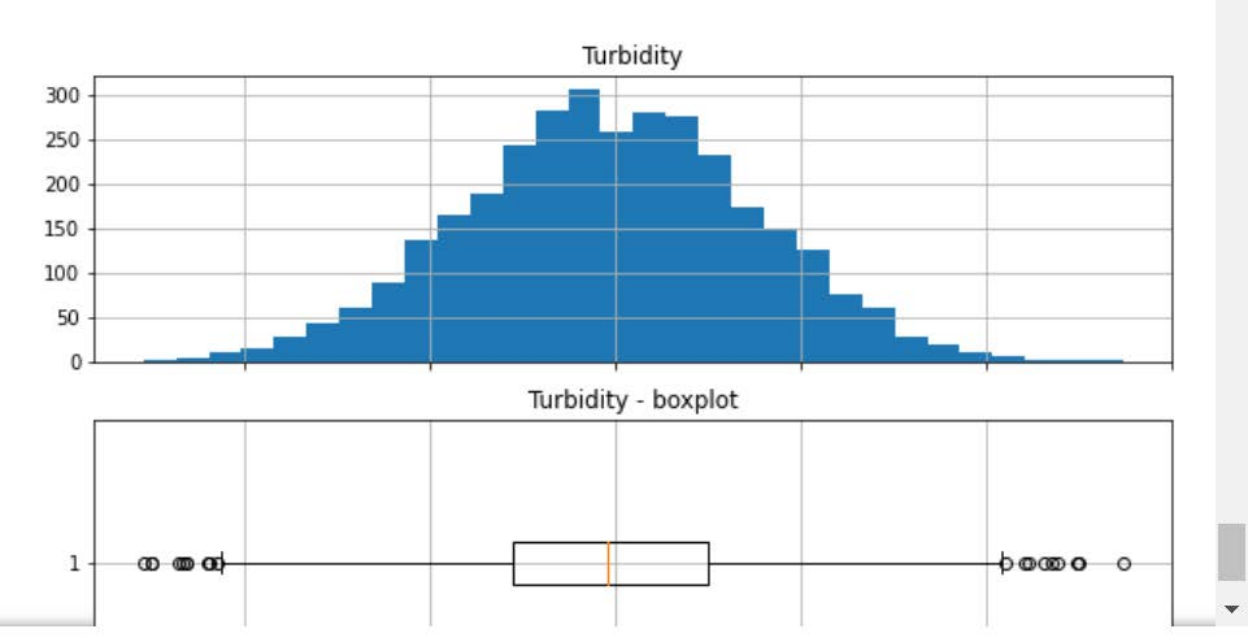

Conductivity
300
250
200
150
100
50
0


Conductivity - boxplot
1


Organic_carbon
300
200
100
0
Organic_carbon - boxplot
1


Trihalomethanes
500
400
300
200
100
0





Список литературы

[1] Гапанюк Ю. Е. Лабораторная работа «Разведочный анализ данных. Исследование и визуализация данных» [Электронный ресурс] // GitHub. — 2019. — Режим доступа: https://github.com/ugapanyuk/ml_course/wiki/LAB_EDA_VISUALIZATION (дата обращения: 13.02.2019)

[2] https://www.kaggle.com/datasets