

## ▼ Water Consumption Amount by District in Istanbul

**\*\* Organizing Dataset and Analysis \*\***

1. Reading dataset
2. Rearrange the dataset
3. Reading Data Tail
4. Data set Shape
5. Showing Columns
6. How many district are there?
7. Creating a column that includes total water consumption of every year would be better
8. Check Null
9. Data set info
10. Dataset mean
11. Dataset median
12. Checking kurtosis
13. Checking skewness
14. Checking correlation
15. Checking covariance
16. Descriptive Statistics
17. Finding out most water consumption

```
#importing libraries
import matplotlib.pyplot as plt
%matplotlib inline
import matplotlib.mlab as mlab
import seaborn as sns
import pandas as pd
import numpy as np
import plotly.express as px
from plotly.subplots import make_subplots
```

```
#reading dataset
url='https://raw.githubusercontent.com/Sagor96/ds/main/dataset/water-consumption.csv'
df = pd.read_csv(url)
df.head()
```

	District	2019 (Consumption- m3)	2018 (Consumption- m3)	2017 (Consumption- m3)	2016 (Consumption- m3)	2015 (Consumption- m3)
0	ADALAR	1371291	1399182	1366581	1472276	1432494
1	ARNAVUTKÖY	13818204	11404878	10176132	9178953	8002123
2	ATAŞEHİR	22428468	21496185	21205911	19974097	18597049
3	AVCILAR	19485453	18312736	17558403	17052253	16220476

```
# Let's rearrange the dataset with the pandas concat method to display the years in order
df = pd.concat((df[["District"]],df[df.columns[1:][::-1]]), axis=1)
```

```
# Now the dataset will look better
```

```
df.head()
```

	District	2015 (Consumption- m3)	2016 (Consumption- m3)	2017 (Consumption- m3)	2018 (Consumption- m3)	2019 (Consumption- m3)
0	ADALAR	1432494	1472276	1366581	1399182	1371291
1	ARNAVUTKÖY	8002123	9178953	10176132	11404878	13818204
2	ATAŞEHİR	18597049	19974097	21205911	21496185	22428468
3	AVCILAR	16220476	17052253	17558403	18312736	19485453

```
#Showing Columns
df.columns
```

```
Index(['District', '2015 (Consumption-m3)', '2016 (Consumption-m3)',
      '2017 (Consumption-m3)', '2018 (Consumption-m3)',
      '2019 (Consumption-m3)'],
      dtype='object')
```

```
#How many district are there?
print("Number of districts: {}".format(df.District.nunique()))
```

```
Number of districts: 39
```

```
#Data set Shape
df.shape
```

```
(39, 6)
```

```
# Creating a column that includes total water consumption of every year would be better
total = df.sum(numeric_only=True, axis=1).to_frame()
```

```
total.columns = ["Total"]
```

```
df = pd.concat([df,total], axis=1)
```

```
df.head()
```

	District	2019 (Consumption- m3)	2018 (Consumption- m3)	2017 (Consumption- m3)	2016 (Consumption- m3)	2015 (Consumption- m3)	Total
0	ADALAR	1371291	1399182	1366581	1472276	1432494	7041824
1	ARNAVUTKÖY	13818204	11404878	10176132	9178953	8002123	52580290
2	ATAŞEHİR	22428468	21496185	21205911	19974097	18597049	103701710
3	AVCILAR	19485453	18312736	17558403	17052253	16220476	88629321

```
#Reading Data Tail
df.tail(5)
```

	District	2015 (Consumption- m3)	2016 (Consumption- m3)	2017 (Consumption- m3)	2018 (Consumption- m3)	2019 (Consumption- m3)	Total
34	ÇEKMEKÖY	8951936	9705483	10205702	10808688	11658154	51329963
35	ÜMRANİYE	28085173	30065065	31061816	32858425	33918205	155988684
36	ÜSKÜDAR	25142059	26110542	26353986	27205893	28298739	133111219
37	ŞİLE	1835302	2098928	2196246	2522410	2965979	11618865

```
#Check Null
df.isnull().values.any()
```

```
False
```

```
#Data set info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39 entries, 0 to 38
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   District              39 non-null    object
1   2015 (Consumption-m3) 39 non-null    int64
2   2016 (Consumption-m3) 39 non-null    int64
3   2017 (Consumption-m3) 39 non-null    int64
4   2018 (Consumption-m3) 39 non-null    int64
5   2019 (Consumption-m3) 39 non-null    int64
6   Total                 39 non-null    int64
dtypes: int64(6), object(1)
memory usage: 2.3+ KB
```

```
#Dataset mean
df.mean()

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with at least one column having a FutureWarning: Dropping of nuisance columns in DataFrame reductions (with at least one column having a
```

2015 (Consumption-m3)	1.587994e+07
2016 (Consumption-m3)	1.665096e+07
2017 (Consumption-m3)	1.728420e+07
2018 (Consumption-m3)	1.844990e+07
2019 (Consumption-m3)	1.978694e+07
Total	8.805193e+07

dtype: float64

```
#Dataset median
df.median()

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with at least one column having a FutureWarning: Dropping of nuisance columns in DataFrame reductions (with at least one column having a
```

2015 (Consumption-m3)	16174934.0
2016 (Consumption-m3)	16584976.0
2017 (Consumption-m3)	17162651.0
2018 (Consumption-m3)	17804043.0
2019 (Consumption-m3)	19175326.0
Total	86581261.0

dtype: float64

```
#Checking kurtosis
df.kurtosis()

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with at least one column having a FutureWarning: Dropping of nuisance columns in DataFrame reductions (with at least one column having a
```

2015 (Consumption-m3)	-0.560966
2016 (Consumption-m3)	-0.327526
2017 (Consumption-m3)	-0.071568
2018 (Consumption-m3)	0.101722
2019 (Consumption-m3)	0.356175
Total	-0.118692

dtype: float64

```
#Checking skewness
df.skew()

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with at least one column having a FutureWarning: Dropping of nuisance columns in DataFrame reductions (with at least one column having a
```

2015 (Consumption-m3)	0.017916
2016 (Consumption-m3)	0.198999
2017 (Consumption-m3)	0.266959
2018 (Consumption-m3)	0.293341
2019 (Consumption-m3)	0.337764
Total	0.215699

dtype: float64

```
#Checking correlation
df.corr()
```

	2015 (Consumption-m3)	2016 (Consumption-m3)	2017 (Consumption-m3)	2018 (Consumption-m3)	2019 (Consumption-m3)	Total
2015 (Consumption-m3)	1.000000	0.983279	0.974886	0.967549	0.957332	0.981266
2016 (Consumption-m3)	0.983279	1.000000	0.997902	0.993500	0.987106	0.998189
2017 (Consumption-m3)	0.974886	0.997902	1.000000	0.997345	0.993400	0.998044
2018 (Consumption-m3)	0.967549	0.993500	0.997345	1.000000	0.996899	0.997924
2019 (Consumption-m3)	0.957332	0.987106	0.993400	0.996899	1.000000	0.997106
Total	0.981266	0.998189	0.998044	0.997924	0.997106	1.000000

```
#Checking covariance
df.cov()
```

	2015 (Consumption- m3)	2016 (Consumption- m3)	2017 (Consumption- m3)	2018 (Consumption- m3)	2019 (Consumption- m3)	Total
2015 (Consumption- m3)	5.318456e+13	5.573382e+13	5.749504e+13	6.028044e+13	6.329532e+13	2.899892e+14
2016 (Consumption- m3)	5.573382e+13	6.040851e+13	6.272214e+13	6.596712e+13	6.955511e+13	3.143867e+14
2017 (Consumption- m3)	5.749504e+13	6.272214e+13	6.596712e+13	6.955511e+13	7.368811e+13	3.487812e+14

```
# Descriptive Statistics
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
2015 (Consumption- m3)	39.0	1.587994e+07	7.292774e+06	1432494.0	11338749.5	16174934.0	20530603.5	29188510.0
2016 (Consumption- m3)	39.0	1.665096e+07	7.772291e+06	1472276.0	11552462.0	16584976.0	20894335.0	32799010.0
2017 (Consumption- m3)	39.0	1.728420e+07	8.086933e+06	1366581.0	11964930.0	17162651.0	21594798.5	36083010.0

```
# Finding out most water consumption
df.groupby('District')['Total'].mean().sort_values(ascending=False)
```

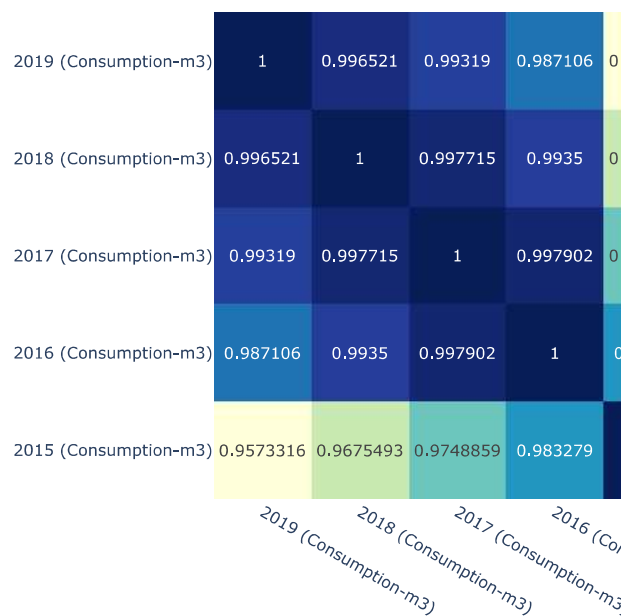
District	
ESENYURT	179597303.0
K.ÇEKMECE	161891430.0
ÜMRANIYE	155988684.0
PENDİK	150142554.0
BAĞCILAR	147871648.0
ÜSKÜDAR	133111219.0
KADIKÖY	128314577.0
BAHÇELİEVLER	126808760.0
MALTEPE	117599521.0
FATİH	115826376.0
ATAŞEHİR	103701710.0
KARTAL	103262043.0
BAŞAKŞEHİR	100765141.0
SARIYER	98248423.0
KAĞITHANE	97606023.0
SALTANGAZİ	96383158.0
GOP	95263805.0
EYÜPSULTAN	93823600.0
AVCILAR	88629321.0
ESENLER	86581261.0
ŞİŞLİ	83681702.0
SANCAKTEPE	76458060.0
BEYLİKDÜZÜ	75050593.0
GÜNGÖREN	71606564.0
TUZLA	69753877.0
ZEYTİNBURNU	68214183.0
BAKIRKÖY	67433576.0
BEŞİKTAŞ	65008521.0
BAYRAMPAŞA	62833200.0
BEYOĞLU	60393280.0
B.ÇEKMECE	57706187.0
BEYKOZ	57148082.0
SULTANBEYLİ	54240920.0
ARNAVUTKÖY	52580290.0
ÇEKMEKÖY	51329963.0
SİLİVRİ	43298547.0
ÇATALCA	17210496.0
ŞİLE	11618865.0
ADALAR	7041824.0
Name: Total, dtype: float64	

\*\* Visualization of Water Consumption in İstanbul \*\*

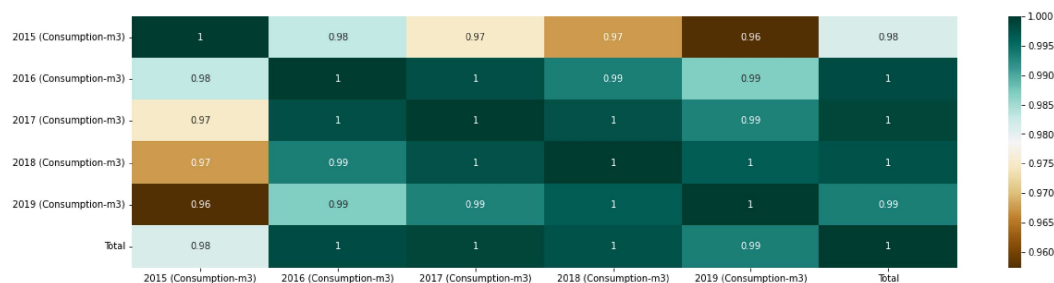
- 1. Heatmap of water consumption
- 2. District wise parameter checking
- 3. District wise Water Consumption in those years

4. District wise distribution
5. Water Consumption by Stacked Bar Chart
6. Total and Mean Water Consumption by Years
7. Box Plot of Total Water Consumption
8. Box Plots of Water Consumption
9. Histogram chart
10. Categorical Scaller Chart

```
# Heatmap of water consumption
px.imshow(df.corr(), text_auto=True, color_continuous_scale="ylgnbu")
```



```
correlation = df.corr()
plt.figure(figsize=(20, 5))
sns.heatmap(correlation, annot=True, cmap="BrBG")
plt.show()
```



```
#District wise parameter checking
tot=df.groupby('District')['Total','2015 (Consumption-m3)','2016 (Consumption-m3)', '2017 (Consumption-m3)', '2018 (Consumption-m3)','2019 (Consumption-m3)']
sns.distplot(tot)
```

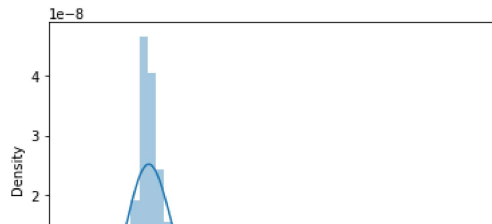
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning:
```

```
Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list in
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
```

```
`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe9c12cbc90>
```



```
#District wise Water Consumption in those years
```

```
import plotly.graph_objects as go
```

```
districts = df["District"].tolist()
```

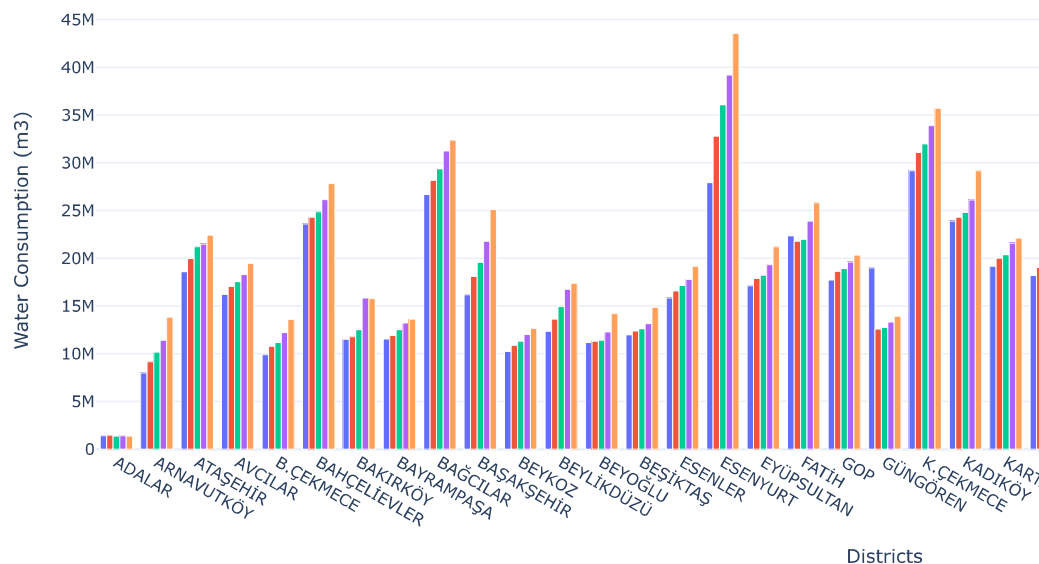
```
fig = go.Figure(data=[
    go.Bar(name="2015", x=districts, y=df["2015 (Consumption-m3)"]),
    go.Bar(name="2016", x=districts, y=df["2016 (Consumption-m3)"]),
    go.Bar(name="2017", x=districts, y=df["2017 (Consumption-m3)"]),
    go.Bar(name="2018", x=districts, y=df["2018 (Consumption-m3)"]),
    go.Bar(name="2019", x=districts, y=df["2019 (Consumption-m3)"])
])
```

```
# Change the bar mode
```

```
fig.update_layout(
    title={"text": "Water Consumption by Bar Chart",
           },
    template = "plotly_white",
    xaxis = dict(title="Districts"),
    yaxis = dict(title="Water Consumption (m3)")
)
```

```
fig.show()
```

Water Consumption by Bar Chart



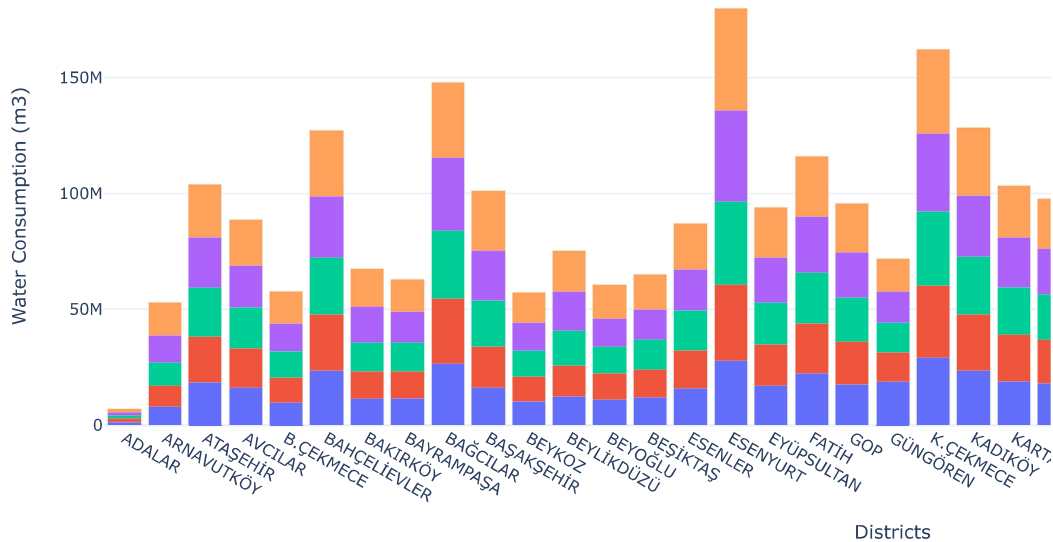
```
# Water Consumption by Stacked Bar Chart
```

```
fig = go.Figure(data=[
    go.Bar(name="2015", x=districts, y=df["2015 (Consumption-m3)"]),
    go.Bar(name="2016", x=districts, y=df["2016 (Consumption-m3)"]),
    go.Bar(name="2017", x=districts, y=df["2017 (Consumption-m3)"]),
    go.Bar(name="2018", x=districts, y=df["2018 (Consumption-m3)"]),
    go.Bar(name="2019", x=districts, y=df["2019 (Consumption-m3)"])
])
```

```
# Change the bar mode
fig.update_layout(barmode="stack",
                  title={"text": "Water Consumption by Stacked Bar Chart",
                        },
                  template = "plotly_white",
                  xaxis = dict(title="Districts"),
                  yaxis = dict(title="Water Consumption (m3)"))

fig.show()
```

## Water Consumption by Stacked Bar Chart



```
#Total and Mean Water Consumption by Years
fig = make_subplots(rows=1, cols=2)
#Total water consumption by years
df["2019 (Consumption-m3)"].sum()

fig.add_trace(go.Bar(
    x=df.columns[1:6],
    y=df[df.columns[1:6]].sum(),
    orientation="v",
    name="Total"), row=1, col=1)

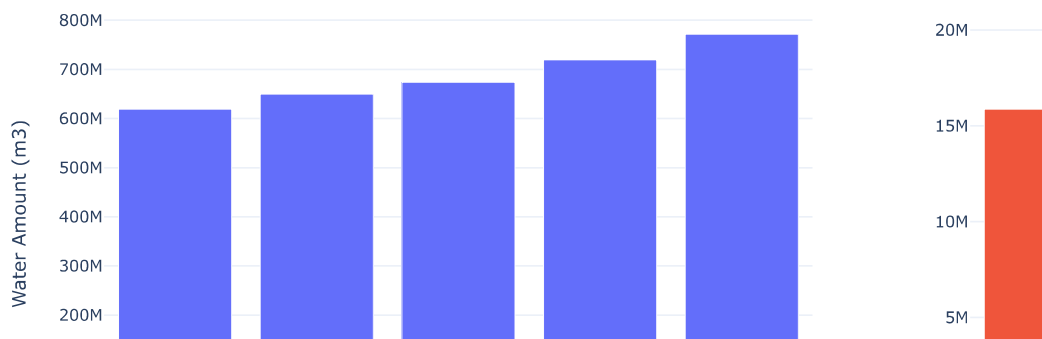
# Mean of water consumption by years
df["2019 (Consumption-m3)"].sum()

fig.add_trace(go.Bar(
    x=df.columns[1:6],
    y=df[df.columns[1:6]].mean(),
    orientation="v",
    name="Mean"), row=1, col=2)

fig.update_layout(title={"text": "Total and Mean Water Consumption by Years"},
    template = "plotly_white",
    xaxis = dict(title="Years"),
    yaxis = dict(title="Water Amount (m3)"))

fig.show()
```

Total and Mean Water Consumption by Years

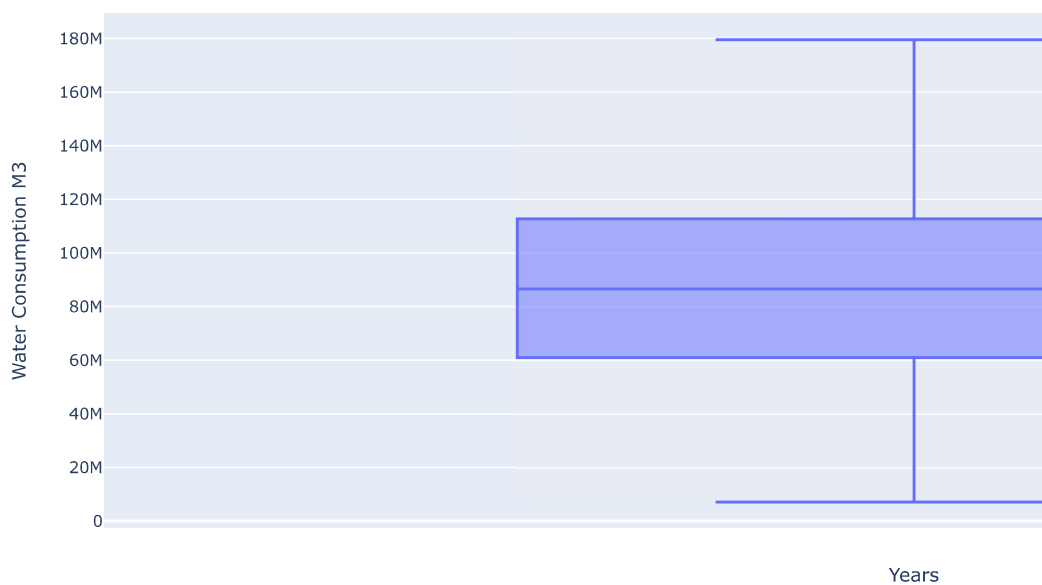


```
# Box Plot of Total Water Consumption
fig = px.box(df, y=df["Total"])

fig.update_layout(title="Box Plot of Total Water Consumption ",
                  xaxis_title="Years",
                  yaxis_title="Water Consumption M3")

fig.show()
```

Box Plot of Total Water Consumption



```
#Box Plots of Water Consumption
import plotly.graph_objects as go
y0 = df['2015 (Consumption-m3)']
y1 = df['2016 (Consumption-m3)']
y2 = df['2017 (Consumption-m3)']
y3 = df['2018 (Consumption-m3)']
y4 = df['2019 (Consumption-m3)']

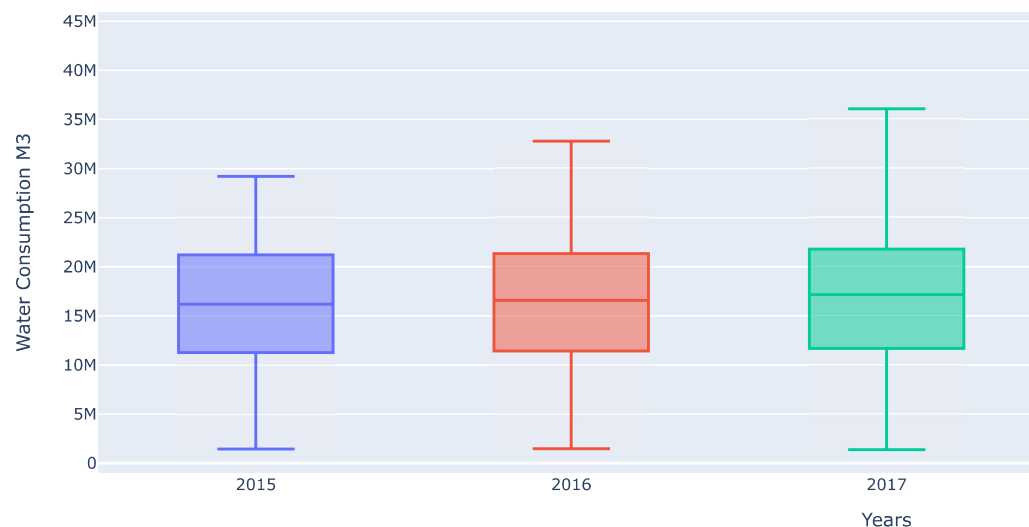
fig = go.Figure()
fig.add_trace(go.Box(y=y0, name="2015"))
fig.add_trace(go.Box(y=y1, name="2016"))
fig.add_trace(go.Box(y=y2, name="2017"))
fig.add_trace(go.Box(y=y3, name="2018"))
fig.add_trace(go.Box(y=y4, name="2019"))

fig.update_layout(title="Box Plots of Water Consumption (Year- 2015 to 2019) ",
                  xaxis_title="Years",
                  yaxis_title="Water Consumption M3")

fig.show()
```



Box Plots of Water Consumption (Year- 2015 to 2019)



```
#Box Plots of Water Consumption
```

```
import plotly.graph_objects as go
y0 = df['2015 (Consumption-m3)']
y1 = df['2016 (Consumption-m3)']
y2 = df['2017 (Consumption-m3)']
y3 = df['2018 (Consumption-m3)']
y4 = df['2019 (Consumption-m3)']
```

```
fig = go.Figure()
fig.add_trace(go.Box(x=y0, name="2015"))
fig.add_trace(go.Box(x=y1, name="2016"))
fig.add_trace(go.Box(x=y2, name="2017"))
fig.add_trace(go.Box(x=y3, name="2018"))
fig.add_trace(go.Box(x=y4, name="2019"))
```

```
fig.update_layout(title="Water Consumption Box Plots",
                  xaxis_title="Water Consumption M3",
                  yaxis_title="Years")
```

```
fig.show()
```

### Water Consumption Box Plot

```
#Histogram chart
import plotly.graph_objects as go
from plotly.subplots import make_subplots

x0 = df['2015 (Consumption-m3)']
x1 = df['2016 (Consumption-m3)']
x2 = df['2017 (Consumption-m3)']
x3 = df['2018 (Consumption-m3)']
x4 = df['2019 (Consumption-m3)']
x5 = df['Total']

fig = make_subplots(rows=3, cols=2)

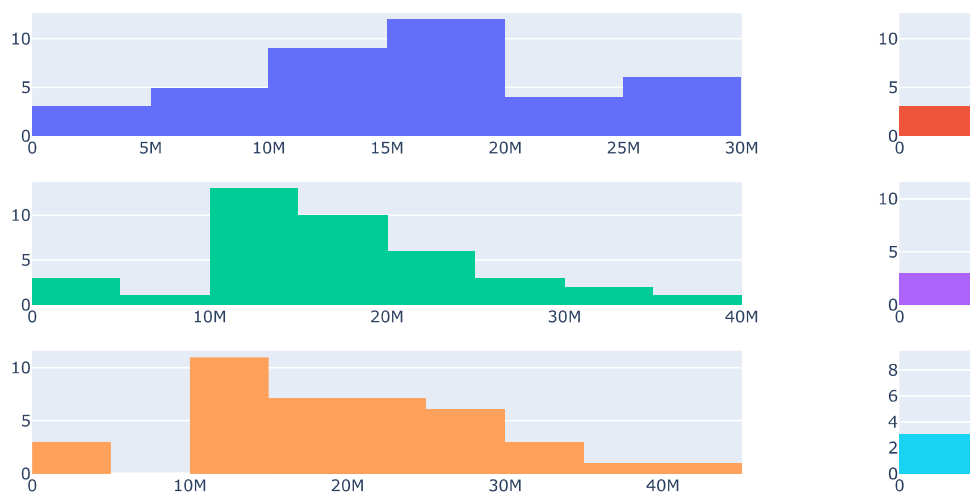
Y2015 = go.Histogram(x=x0, nbinsx=12, name="2015")
Y2016 = go.Histogram(x=x1, nbinsx=12, name="2016")
Y2017 = go.Histogram(x=x2, nbinsx=12, name="2017")
Y2018 = go.Histogram(x=x3, nbinsx=12, name="2018")
Y2019 = go.Histogram(x=x4, nbinsx=12, name="2019")
Total = go.Histogram(x=x5, nbinsx=12, name="Total")

fig.append_trace(Y2015, 1, 1)
fig.append_trace(Y2016, 1, 2)
fig.append_trace(Y2017, 2, 1)
fig.append_trace(Y2018, 2, 2)
fig.append_trace(Y2019, 3, 1)
fig.append_trace(Total, 3, 2)

fig.update_layout(title="Histogram of Water Consumption")

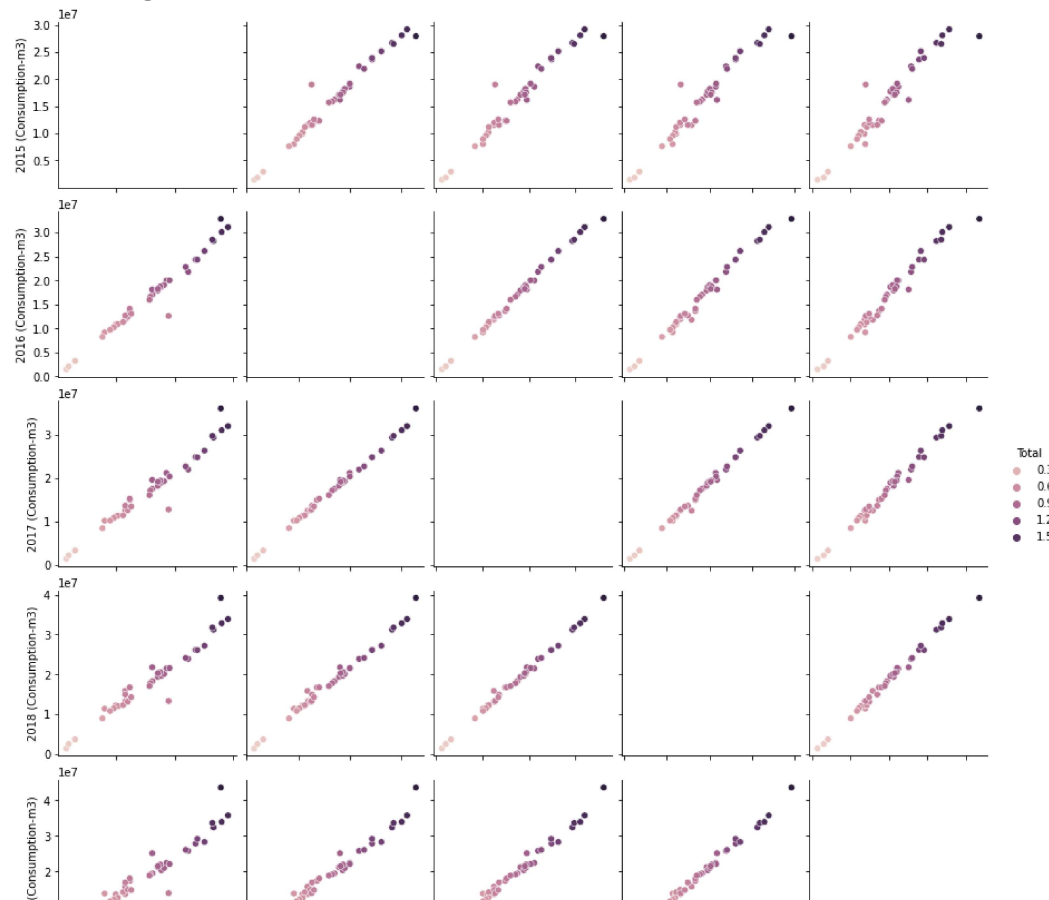
fig.show()
```

Histogram of Water Consumption



```
sns.pairplot(df[['District', '2015 (Consumption-m3)', '2016 (Consumption-m3)', '2017 (Consumption-m3)', '2018 (Consumption-m3)', '2019 (Consumption-m3)'])
```

```
<seaborn.axisgrid.PairGrid at 0x7f25e0fe7d50>
```



```
#Categorical Scaller Chart
```

```
#Data Normalization List
```

```
wc2015 = df["2015 (Consumption-m3)"].tolist()
wc2016 = df["2016 (Consumption-m3)"].tolist()
wc2017 = df["2017 (Consumption-m3)"].tolist()
wc2018 = df["2018 (Consumption-m3)"].tolist()
wc2019 = df["2019 (Consumption-m3)"].tolist()
total = df["Total"].tolist()
```

```
fig = go.Figure()
```

```
#Scaller Part
```

```
fig.add_trace(go.Scatter(
    x=wc2015,
    y=districts,
    marker=dict(color="#FFA500", size=8),
    mode="markers",
    name="2015",
))
```

```
fig.add_trace(go.Scatter(
    x=wc2016,
    y=districts,
    marker=dict(color="#800000", size=8),
    mode="markers",
    name="2016",
))
```

```
fig.add_trace(go.Scatter(
    x=wc2017,
    y=districts,
    marker=dict(color="#3B9C9C", size=8),
    mode="markers",
    name="2017",
))
```

```
fig.add_trace(go.Scatter(
    x=wc2018,
    y=districts,
```

```

marker=dict(color="#FF00FF", size=8),
mode="markers",
name="2018",
))

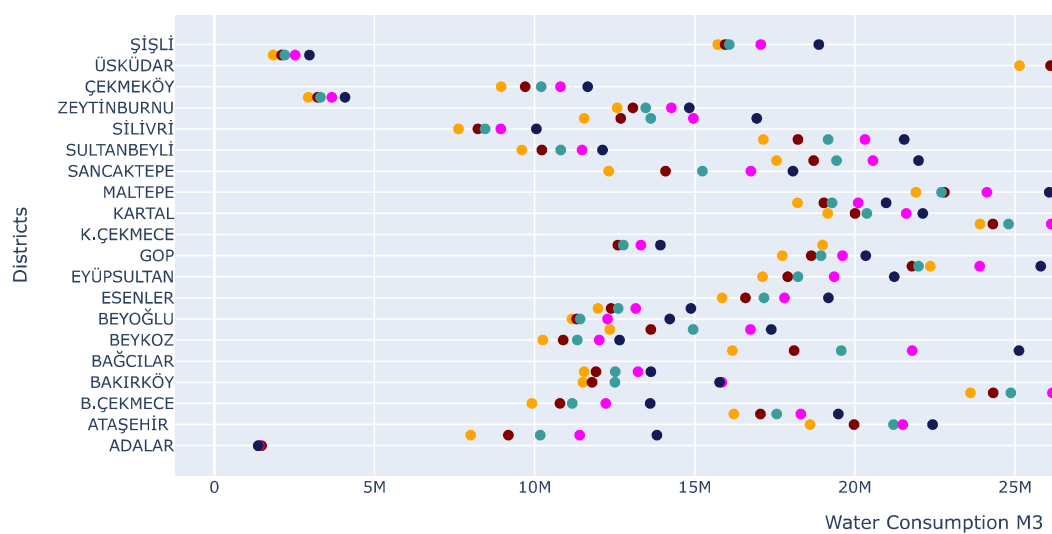
fig.add_trace(go.Scatter(
    x=wc2019,
    y=districts,
    marker=dict(color="#151B54", size=8),
    mode="markers",
    name="2019",
))

fig.update_layout(title="Water Consumption Scaller",
    xaxis_title="Water Consumption M3",
    yaxis_title="Districts")

fig.show()

```

Water Consumption Scaller



Created By MD. ASHRAFUL ALAM ---- ID # 0242220005343011