```
import pandas as pd
import numpy as py
df = pd.read_csv('house_hold.csv')
```

```
#water usage for one year
df
```

	Year	Month	Household	Bathing(L)	Cooking(L)	Washing(L)	Gardening(L)	Drinking(L)	Total_Usage(L)
0	2023	1	Household_1	1274	432	1412.0	774.0	311.0	4203
1	2023	1	Household_2	1219	590	1844.0	1185.0	331.0	5169
2	2023	1	Household_3	1076	589	NaN	1023.0	238.0	4310
3	2023	1	Household_4	1486	465	1766.0	992.0	NaN	5103
4	2023	1	Household_5	1422	470	1721.0	615.0	236.0	4464
...	...	...	...	...	...	...	...	...	...
115	2023	12	Household_6	1032	480	1802.0	1200.0	259.0	4773
116	2023	12	Household_7	1270	560	1421.0	900.0	301.0	4452
117	2023	12	Household_8	1461	442	1992.0	1190.0	345.0	5430
118	2023	12	Household_9	964	536	1662.0	1021.0	291.0	4474
119	2023	12	Household_10	1431	505	1777.0	943.0	258.0	4914

120 rows × 9 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
#water usage for one month
df1=df[df['Month']==1]
df1
```

	Year	Month	Household	Bathing(L)	Cooking(L)	Washing(L)	Gardening(L)	Dr
0	2023	1	Household_1	1274	432	1412.0	774.0	

1	2023	1	Household_2	1219	590	1844.0	1185.0
2	2023	1	Household_3	1076	589	NaN	1023.0
3	2023	1	Household_4	1486	465	1766.0	992.0
4	2023	1	Household_5	1422	470	1721.0	615.0
5	2023	1	Household_6	1249	447	1832.0	759.0
6	2023	1	Household_7	1095	582	1377.0	1060.0
7	2023	1	Household_8	1455	557	1960.0	890.0
8	2023	1	Household_9	975	456	1480.0	NaN
9	2023	1	Household_10	1378	574	1854.0	722.0

Next steps: Generate code with df1 View recommended plots New interactive sheet

```
df1.head()
```

	Year	Month	Household	Bathing(L)	Cooking(L)	Washing(L)	Gardening(L)	Dri
0	2023	1	Household_1	1274	432	1412.0	774.0	
1	2023	1	Household_2	1219	590	1844.0	1185.0	
2	2023	1	Household_3	1076	589	NaN	1023.0	
3	2023	1	Household_4	1486	465	1766.0	992.0	
4	2023	1	Household_5	1422	470	1721.0	615.0	

Next steps: Generate code with df1 View recommended plots New interactive sheet

```
 #checking is null values is there are not ->to clean the data
df1.isnull()
```

	Year	Month	Household	Bathing(L)	Cooking(L)	Washing(L)	Gardening(L)	Dri
0	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	
2	False	False	False	False	False	True	False	
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	
5	False	False	False	False	False	False	False	
6	False	False	False	False	False	False	False	
7	False	False	False	False	False	False	False	
8	False	False	False	False	False	False	True	
9	False	False	False	False	False	False	False	

```
df1=df1.drop(columns=["Total_Usage(L)"])
df1
```

	Year	Month	Household	Bathing(L)	Cooking(L)	Washing(L)	Gardening(L)	Dr
0	2023	1	Household_1	1274	432	1412.0	774.0	
1	2023	1	Household_2	1219	590	1844.0	1185.0	
2	2023	1	Household_3	1076	589	NaN	1023.0	
3	2023	1	Household_4	1486	465	1766.0	992.0	
4	2023	1	Household_5	1422	470	1721.0	615.0	
5	2023	1	Household_6	1249	447	1832.0	759.0	
6	2023	1	Household_7	1095	582	1377.0	1060.0	
7	2023	1	Household_8	1455	557	1960.0	890.0	
8	2023	1	Household_9	975	456	1480.0	NaN	
9	2023	1	Household_10	1378	574	1854.0	722.0	

Next steps: Generate code with df1 | View recommended plots | New interactive sheet

```
df1.isnull().sum()
```

	0
Year	0
Month	0
Household	0
Bathing(L)	0
Cooking(L)	0
Washing(L)	1
Gardening(L)	1
Drinking(L)	2

dtype: int64

```
#fill the null values
df1= df1.fillna(df1.mean(numeric_only=True))
df1
```

	Year	Month	Household	Bathing(L)	Cooking(L)	Washing(L)	Gardening(L)	Dr
0	2023	1	Household_1	1274	432	1412.0	774.000000	
1	2023	1	Household_2	1219	590	1844.0	1185.000000	
2	2023	1	Household_3	1076	589	1694.0	1023.000000	
3	2023	1	Household_4	1486	465	1766.0	992.000000	
4	2023	1	Household_5	1422	470	1721.0	615.000000	
5	2023	1	Household_6	1249	447	1832.0	759.000000	
6	2023	1	Household_7	1095	582	1377.0	1060.000000	
7	2023	1	Household_8	1455	557	1960.0	890.000000	
8	2023	1	Household_9	975	456	1480.0	891.111111	
9	2023	1	Household_10	1378	574	1854.0	722.000000	

Next steps: Generate code with df1 View recommended plots New interactive sheet

```
#add the total_usage coloumn
df1["total_usage"] = df1["Bathing(L)"] + df1["Cooking(L)"] + df1["Washing(L)"] + df1["Gardening(L)"] + df1["Drinking(L)"]
df1
```

	Year	Month	Household	Bathing(L)	Cooking(L)	Washing(L)	Gardening(L)	Dr
0	2023	1	Household_1	1274	432	1412.0	774.000000	
1	2023	1	Household_2	1219	590	1844.0	1185.000000	
2	2023	1	Household_3	1076	589	1694.0	1023.000000	
3	2023	1	Household_4	1486	465	1766.0	992.000000	
4	2023	1	Household_5	1422	470	1721.0	615.000000	
5	2023	1	Household_6	1249	447	1832.0	759.000000	
6	2023	1	Household_7	1095	582	1377.0	1060.000000	
7	2023	1	Household_8	1455	557	1960.0	890.000000	
8	2023	1	Household_9	975	456	1480.0	891.111111	
9	2023	1	Household_10	1378	574	1854.0	722.000000	

Next steps: Generate code with df1 View recommended plots New interactive sheet

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, 0 to 9
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Year          10 non-null     int64
 1   Month         10 non-null     int64
 2   Household     10 non-null     object
 3   Bathing(L)    10 non-null     int64
 4   Cooking(L)    10 non-null     int64
 5   Washing(L)    10 non-null     float64
 6   Gardening(L)  10 non-null     float64
 7   Drinking(L)   10 non-null     float64
 8   total_usage   10 non-null     float64
```

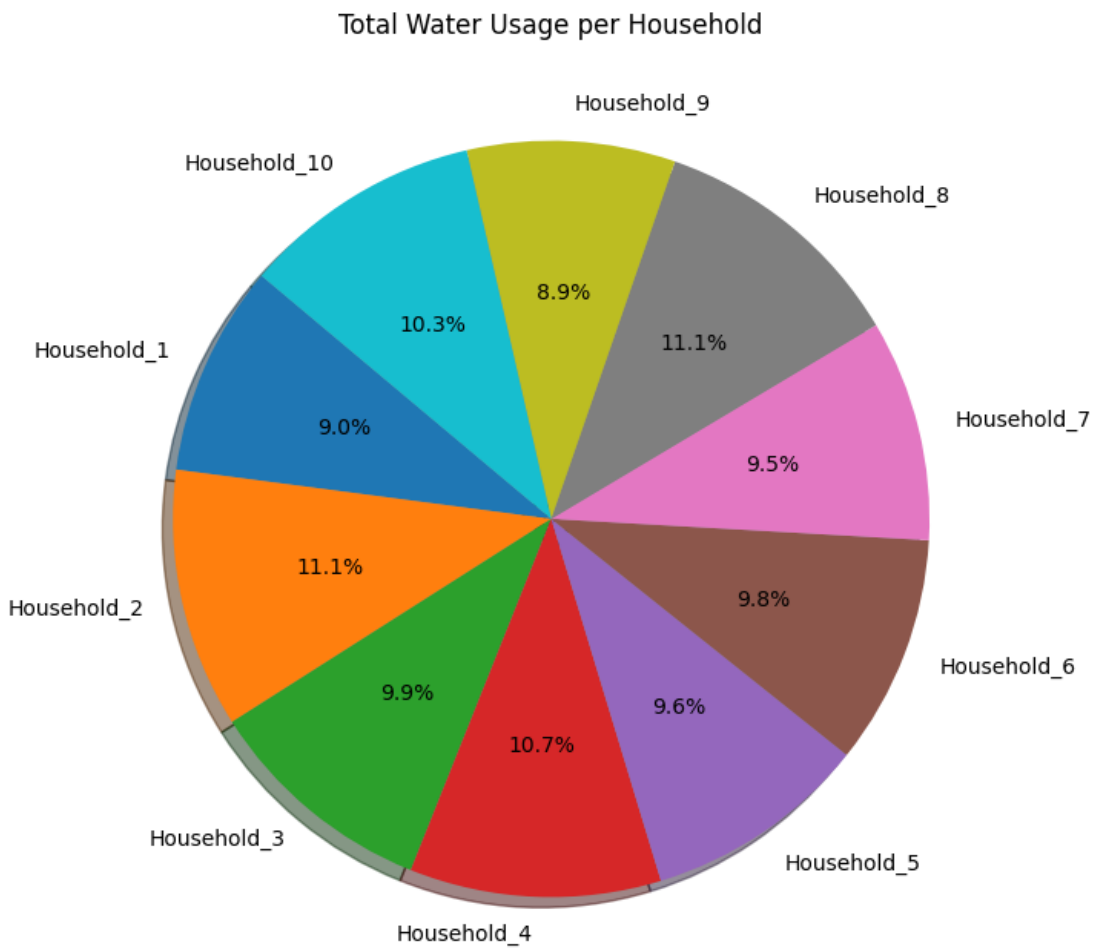
```
dtypes: float64(4), int64(4), object(1)
memory usage: 800.0+ bytes
```

```
df1.describe()
```

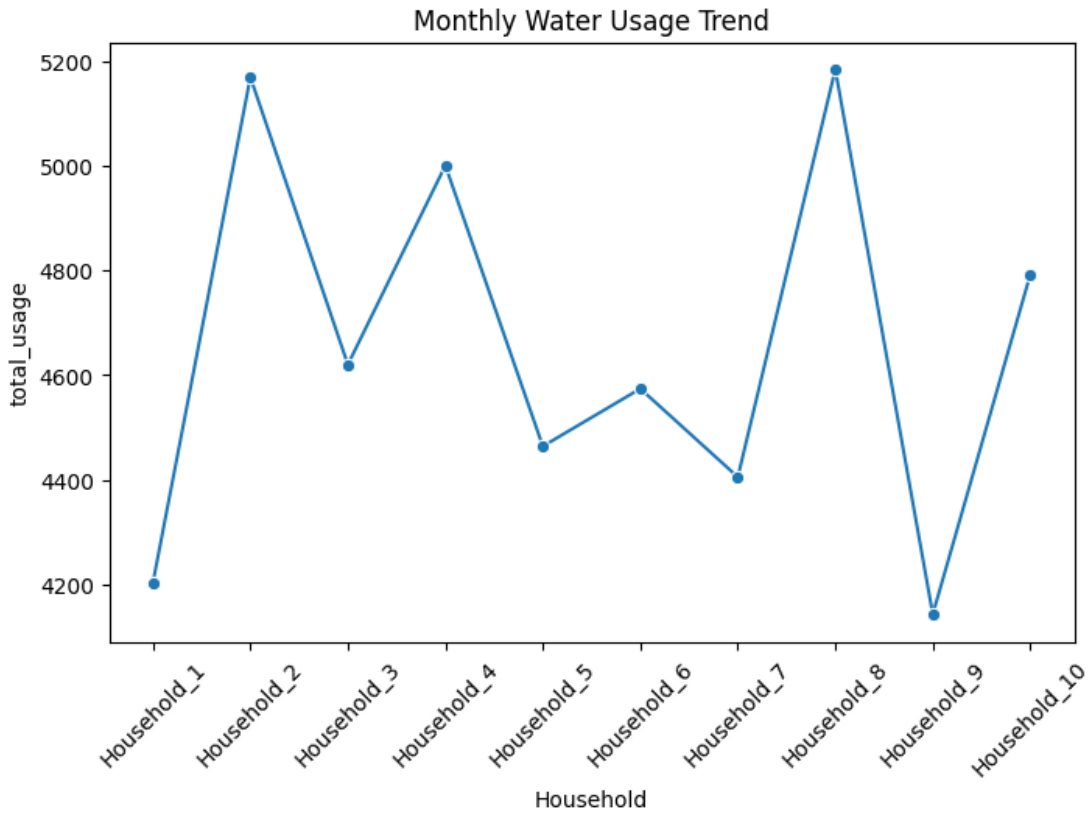
	Year	Month	Bathing(L)	Cooking(L)	Washing(L)	Gardening(L)	Drinking(l
count	10.0	10.0	10.000000	10.000000	10.000000	10.000000	10.0000
mean	2023.0	1.0	1262.900000	516.200000	1694.000000	891.111111	291.2500
std	0.0	0.0	174.320554	66.949069	202.655153	175.901768	36.5748
min	2023.0	1.0	975.000000	432.000000	1377.000000	615.000000	236.0000
25%	2023.0	1.0	1126.000000	458.250000	1533.500000	762.750000	269.7500
50%	2023.0	1.0	1261.500000	513.500000	1743.500000	890.555556	291.2500
75%	2023.0	1.0	1411.000000	580.000000	1841.000000	1015.250000	319.2500
max	2023.0	1.0	1486.000000	590.000000	1960.000000	1185.000000	341.0000

```
import matplotlib.pyplot as plt
import seaborn as sns
```

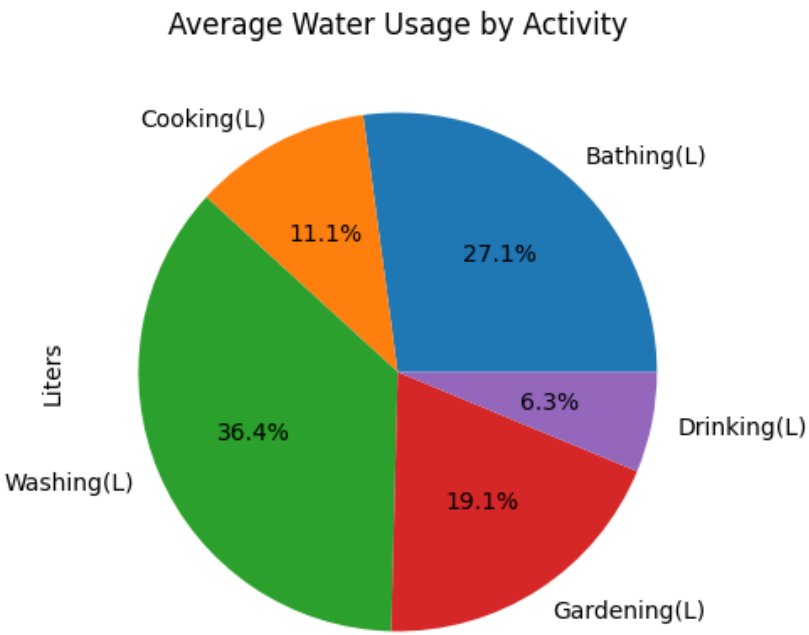
```
#Total water usage for every houshold that represent in pie chart
plt.figure(figsize=(8,8))
plt.pie( df1["total_usage"],labels=df1["Household"],autopct='%1.1f%%',startangle=140,shadow=True)
plt.title("Total Water Usage per Household")
plt.show()
```




```
plt.figure(figsize=(8,5))
sns.lineplot(x="Household", y="total_usage", data=df1, marker="o")
plt.xticks(rotation=45)
plt.title("Monthly Water Usage Trend")
plt.show()
```




```
#Average Water usage for every activity
activity_cols = ['Bathing(L)','Cooking(L)','Washing(L)','Gardening(L)','Drinking(L)']
df1[activity_cols].mean().plot(kind="pie", figsize=(8,5),autopct='%1.1f%%')
plt.title("Average Water Usage by Activity")
plt.ylabel("Liters")
plt.show()
```




```
#Data Normalization
from sklearn.preprocessing import MinMaxScaler, StandardScaler
activity_cols = ['Bathing(L)','Cooking(L)','Washing(L)','Gardening(L)','Drinking(L)','total_usage']
minmax_scaler = MinMaxScaler()
df_minmax = df1.copy()
df_minmax[activity_cols] = minmax_scaler.fit_transform(df1[activity_cols])
print("Min-Max Normalized Data:")
print(df_minmax)
standard_scaler = StandardScaler()
df_standard = df1.copy()
df_standard[activity_cols] = standard_scaler.fit_transform(df1[activity_cols])
print("\nStandardized Data:")
print(df_standard)
```

```
Min-Max Normalized Data:
   Year  Month     Household  Bathing(L)  Cooking(L)  Washing(L)  \
0  2023      1   Household_1    0.585127    0.000000    0.060034
1  2023      1   Household_2    0.477495    1.000000    0.801029
2  2023      1   Household_3    0.197652    0.993671    0.543739
```

```
3  2023      1   Household_4    1.000000    0.208861    0.667238
4  2023      1   Household_5    0.874755    0.240506    0.590051
5  2023      1   Household_6    0.536204    0.094937    0.780446
6  2023      1   Household_7    0.234834    0.949367    0.000000
7  2023      1   Household_8    0.939335    0.791139    1.000000
8  2023      1   Household_9    0.000000    0.151899    0.176672
9  2023      1  Household_10    0.788650    0.898734    0.818182

   Gardening(L)  Drinking(L)  total_usage
0      0.278947     0.714286     0.057536
1      1.000000     0.904762     0.985589
2      0.715789     0.019048     0.458155
3      0.661404     0.526190     0.823468
4      0.000000     0.000000     0.308284
5      0.252632     0.485714     0.413962
6      0.780702     0.526190     0.251841
7      0.482456     0.819048     1.000000
8      0.484405     1.000000     0.000000
9      0.187719     0.266667     0.623399

Standardized Data:
   Year  Month     Household  Bathing(L)  Cooking(L)  Washing(L)  \
0  2023      1   Household_1    0.067120   -1.325703   -1.466798
1  2023      1   Household_2   -0.265457    1.161958    0.780212
2  2023      1   Household_3   -1.130159    1.146214    0.000000
3  2023      1   Household_4    1.349055   -0.806128    0.374502
4  2023      1   Household_5    0.962056   -0.727405    0.140438
5  2023      1   Household_6   -0.084051   -1.089533    0.717795
6  2023      1   Household_7   -1.015268    1.036001   -1.648847
7  2023      1   Household_8    1.161602    0.642383    1.383575
8  2023      1   Household_9   -1.740892   -0.947831   -1.113102
9  2023      1  Household_10    0.695994    0.910043    0.832226

   Gardening(L)  Drinking(L)  total_usage
0     -0.701789     0.569197    -1.277213
1      1.761131     1.145600     1.449624
2      0.790345    -1.534671    -0.100100
3      0.604577     0.000000     0.973274
4     -1.654598    -1.592312    -0.540459
5     -0.791677    -0.122486    -0.229949
6      1.012068     0.000000    -0.706299
7     -0.006658     0.886219     1.491966
8      0.000000     1.433801    -1.446268
9     -1.013399    -0.785348     0.385424
```

```
plt.figure(figsize=(10,6))
sns.barplot(x="Household", y="total_usage", data=df1)
plt.xticks(rotation=45)
```

```
plt.title("Total Water Usage per Household")
plt.show()
```