```
import pandas as pd
import numpy as np
```

```
# Read the dataset
df = pd.read_csv('/Users/priyakundu/Documents/NYU Capstone WaterVue Files/Sample_Dataset.csv')
df
```

	Site #	Location	Sample Date	Analysis code	Analysis	DQC	Result	Units	Det
0	1.0	#1 HILLSBORO CANAL US 1	2/22/06	Chl a	Chlorophyll a	NaN	12.100	mg/m3	
1	1.0	#1 HILLSBORO CANAL US 1	2/22/06	Conductivity	Specific Conductance	NaN	31300.000	umho/cm	
2	1.0	#1 HILLSBORO CANAL US 1	2/22/06	DO	Dissolved Oxygen	NaN	6.980	mg/L	
3	1.0	#1 HILLSBORO CANAL US 1	2/22/06	Sal	Salinity	NaN	19.400	ppt	
		#1 HILLSBORO			Total				

```
# Pivot the dataframe to get unique values from "Analysis" column as columns
df_pivot = df.pivot_table(index=['Site #', 'Location', 'Sample Date'], columns='Analysis', values='Result').reset_index()

# Group by sample date and location
pivoted_df = df_pivot.groupby(['Sample Date', 'Location']).first().reset_index()

# Print the new dataframe
pivoted_df
```

Analysis	Sample Date	Location	Site #	Chlorophyll A	Chlorophyll a	Copper	Dissolved Oxygen	Sa
0	1/30/12	#1 HILLSBORO CANAL US 1	1.0	NaN	2.77	NaN	6.110	
1	1/30/12	#11 MIDDLE RIVER NW 21ST AVE	11.0	NaN	5.29	NaN	5.195	
2	1/30/12	#110 POMPANO CANAL AT DIXIE AN	110.0	NaN	5.53	NaN	7.130	
3	1/30/12	#111 S. FORK MID R. @ N.E. 15	111.0	NaN	NaN	NaN	6.170	
4	1/30/12	#112 N. FORK MID R. @ N.E. 16	112.0	NaN	NaN	NaN	5.980	
...	...	...	...	...	...	...	...	
3031	9/9/19	#2 HILLSBORO LOCK	2.0	NaN	15.00	NaN	6.250	
3032	9/9/19	#3 HILLSBORO CANAL US 441	3.0	NaN	9.77	NaN	2.120	
		#4						

```
# Renaming the columns using indexing
pivoted_df.columns = [*pivoted_df.columns[:-2], 'Turbidity1', 'Turbidity2']
```

```
# Merge the "Turbidity1" and "Turbidity2" columns into a single column named "Turbidity1"
pivoted_df['Turbidity'] = pivoted_df['Turbidity1'].combine_first(pivoted_df['Turbidity2'])

# Drop the "old" columns
pivoted_df.drop(columns=['Turbidity1' , 'Turbidity2'], inplace=True)

# Merge the "Chlorophyll A" and "Chlorophyll a" columns into a single column named "Chlorophyll A"
pivoted_df['Chlorophyll A'] = pivoted_df['Chlorophyll A'].combine_first(pivoted_df['Chlorophyll a'])

# Drop the "Chlorophyll a" column
pivoted_df.drop(columns=['Chlorophyll a'], inplace=True)

# Drop 'Copper', 'Site #' column
pivoted_df.drop(['Copper', 'Site #'], axis=1, inplace=True)

# Print the cleaned dataframe
pivoted_df
```

	Sample Date	Location	Chlorophyll A	Dissolved Oxygen	Salinity	Specific Conductance	Total Nitrogen	Total Phosphorus	Turbidity
0	1/30/12	#1 HILLSBORO CANAL US 1	2.77	6.110	24.80	39000.0	0.3600	0.0770	1.70
1	1/30/12	#11 MIDDLE RIVER NW 21ST AVE	5.29	5.195	5.39	9485.0	0.7025	0.0345	1.00
2	1/30/12	#110 POMPANO CANAL AT DIXIE AN	5.53	7.130	0.20	424.0	1.1600	0.0330	2.00
3	1/30/12	#111 S. FORK MID R. @ N.E. 15	NaN	6.170	20.80	33300.0	0.5460	0.0490	1.00
4	1/30/12	#112 N. FORK MID R. @ N.E. 16	NaN	5.980	21.20	34000.0	0.4840	0.0530	0.90
...	...	...	...	...	...	...	...	...	...
3031	9/9/19	#2 HILLSBORO LOCK	15.00	6.250	0.29	598.0	1.1763	0.0810	1.80
3032	9/9/19	#3 HILLSBORO CANAL US 441	9.77	2.120	0.29	605.0	1.3565	0.0680	1.40

```
# Find null values in the DataFrame
null_values = pivoted_df.isnull()

# Count null values in each column
null_counts = null_values.sum()

print("Null values in each column:")
print(null_counts)
```

```
Null values in each column:
Sample Date               0
Location                  0
Chlorophyll A           336
Dissolved Oxygen        379
Salinity                379
Specific Conductance    376
Total Nitrogen          910
Total Phosphorus        748
Turbidity               303
dtype: int64
```

```
numerical_cols = pivoted_df.select_dtypes(include=['float64']).columns
```

```
# Calculate maximum and minimum values for each numerical column
max_values = pivoted_df[numerical_cols].max()
min_values = pivoted_df[numerical_cols].min()

# Print maximum and minimum values for each numerical column
for column in numerical_cols:
    print(f"Maximum value of {column}: {max_values[column]}")
    print(f"Minimum value of {column}: {min_values[column]}")
```

```
Maximum value of Chlorophyll A: 99.2
Minimum value of Chlorophyll A: -0.048
Maximum value of Dissolved Oxygen: 61.5
Minimum value of Dissolved Oxygen: 0.32
Maximum value of Salinity: 72.4
```

```
Minimum value of Salinity: 0.0
Maximum value of Specific Conductance: 427119.0
Minimum value of Specific Conductance: 37.5
Maximum value of Total Nitrogen: 2.5149999999999997
Minimum value of Total Nitrogen: 0.0
Maximum value of Total Phosphorus: 0.649
Minimum value of Total Phosphorus: -0.147
Maximum value of Turbidity: 23.0
Minimum value of Turbidity: 0.0
```

```
# Define IQR multiplier
k = 1.5

# Calculate Q1 and Q3
Q1 = pivoted_df[numerical_cols].quantile(0.25)
Q3 = pivoted_df[numerical_cols].quantile(0.75)

# Calculate IQR
IQR = Q3 - Q1

# Filter out rows where any value lies outside the range (Q1 - k*IQR, Q3 + k*IQR)
df_no_outliers = pivoted_df[~((pivoted_df[numerical_cols] < (Q1 - k * IQR)) | (pivoted_df[numerical_cols] > (Q3 + k * IQR))).any(axis=1)]

# Print df
df_no_outliers
```

	Sample Date	Location	Chlorophyll A	Dissolved Oxygen	Salinity	Specific Conductance	Total Nitrogen	Total Phosphorus	Turbidity
0	1/30/12	#1 HILLSBORO CANAL US 1	2.77	6.110	24.80	39000.0	0.3600	0.0770	1.70
1	1/30/12	#11 MIDDLE RIVER NW 21ST AVE	5.29	5.195	5.39	9485.0	0.7025	0.0345	1.00
2	1/30/12	#110 POMPANO CANAL AT DIXIE AN	5.53	7.130	0.20	424.0	1.1600	0.0330	2.00
3	1/30/12	#111 S. FORK MID R. @ N.E. 15	NaN	6.170	20.80	33300.0	0.5460	0.0490	1.00
4	1/30/12	#112 N. FORK MID R. @ N.E. 16	NaN	5.980	21.20	34000.0	0.4840	0.0530	0.90
...	...	...	...	...	...	...	...	...	...
3028	9/6/19	#40 ICW SHERIDAN ST	1.20	4.740	25.30	39900.0	0.4440	0.1180	1.40
3029	9/6/19	#41 ICW HALLANDALE BCH BLVD	2.79	5.840	19.90	31800.0	0.5822	0.0690	0.65

```
# Calculate maximum and minimum values for each numerical column
max_values = df_no_outliers[numerical_cols].max()
min_values = df_no_outliers[numerical_cols].min()

# Print maximum and minimum values for each numerical column
for column in numerical_cols:
    print(f"Maximum value of {column}: {max_values[column]}")
    print(f"Minimum value of {column}: {min_values[column]}")
```

```
Maximum value of Chlorophyll A: 14.6
Minimum value of Chlorophyll A: -0.048
Maximum value of Dissolved Oxygen: 9.39
Minimum value of Dissolved Oxygen: 0.82
Maximum value of Salinity: 37.3
Minimum value of Salinity: 0.0
Maximum value of Specific Conductance: 58800.0
Minimum value of Specific Conductance: 37.5
Maximum value of Total Nitrogen: 2.38
Minimum value of Total Nitrogen: 0.0
Maximum value of Total Phosphorus: 0.147
Minimum value of Total Phosphorus: -0.047
Maximum value of Turbidity: 3.4
Minimum value of Turbidity: 0.0
```

```
df_no_outliers['Specific Conductance'] = np.log(df_no_outliers['Specific Conductance'])

df_no_outliers
```

```
/var/folders/9n/nyfs9h7n2lsfs0vd2lq0589h0000gn/T/ipykernel_10894/4246764221.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df_no_outliers['Specific Conductance'] = np.log(df_no_outliers['Specific Conductance'])
```

	Sample Date	Location	Chlorophyll A	Dissolved Oxygen	Salinity	Specific Conductance	Total Nitrogen	Total Phosphorus	Turbidity
0	1/30/12	#1 HILLSBORO CANAL US 1	2.77	6.110	24.80	10.571317	0.3600	0.0770	1.70
1	1/30/12	#11 MIDDLE RIVER NW 21ST AVE	5.29	5.195	5.39	9.157467	0.7025	0.0345	1.00
2	1/30/12	#110 POMPANO CANAL AT DIXIE AN	5.53	7.130	0.20	6.049733	1.1600	0.0330	2.00
3	1/30/12	#111 S. FORK MID R. @ N.E. 15	NaN	6.170	20.80	10.413313	0.5460	0.0490	1.00
4	1/30/12	#112 N. FORK MID R. @ N.E. 16	NaN	5.980	21.20	10.434116	0.4840	0.0530	0.90
...	...	...	...	...	...	...	...	...	...
3028	9/6/19	#40 ICW SHERIDAN ST	1.20	4.740	25.30	10.594132	0.4440	0.1180	1.40
3029	9/6/19	#41 ICW HALLANDALE BCH BLVD	2.79	5.840	19.90	10.367222	0.5822	0.0690	0.65

```
# Convert 'Sample Date' column to datetime
df_no_outliers['Sample Date'] = pd.to_datetime(df_no_outliers['Sample Date'])

# Set 'Sample Date' as index
df_no_outliers.set_index('Sample Date', inplace=True)

# Group by 'Location' and resample yearly for each group
resampled_df = df_no_outliers.groupby('Location').resample('6M').mean().reset_index()

# Output the resampled data
resampled_df
```

```
/var/folders/9n/nyfs9h7n2lsfs0vd2lq0589h0000gn/T/ipykernel_10894/1046139978.py:2: UserWarning: Could not infer format, so ea
  df_no_outliers['Sample Date'] = pd.to_datetime(df_no_outliers['Sample Date'])
/var/folders/9n/nyfs9h7n2lsfs0vd2lq0589h0000gn/T/ipykernel_10894/1046139978.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  df_no_outliers['Sample Date'] = pd.to_datetime(df_no_outliers['Sample Date'])
/var/folders/9n/nyfs9h7n2lsfs0vd2lq0589h0000gn/T/ipykernel_10894/1046139978.py:8: FutureWarning: 'M' is deprecated and will
  resampled_df = df_no_outliers.groupby('Location').resample('6M').mean().reset_index()
```

	Location	Sample Date	Chlorophyll A	Dissolved Oxygen	Salinity	Specific Conductance	Total Nitrogen	Total Phosphorus	Turbidity
0	#1 HILLSBORO CANAL US 1	2006-02-28	12.100	6.980	19.40	10.351373	0.830	0.0860	2.40
1	#1 HILLSBORO CANAL US 1	2006-08-31	4.430	5.540	15.75	10.165844	0.981	0.1090	1.40
2	#1 HILLSBORO CANAL US 1	2007-02-28	2.605	6.195	23.20	10.501905	0.754	0.0835	1.65
3	#1 HILLSBORO CANAL US 1	2007-08-31	4.890	4.730	31.10	10.774781	0.777	0.0940	2.30
4	#1 HILLSBORO CANAL US 1	2008-02-29	5.925	5.770	12.50	9.893361	1.440	0.1020	2.05
...	...	...	...	...	...	...	...	...	...
1580	#90 FPL CANAL SOUTH F NEW RIV	2014-02-28	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1581	#90 FPL CANAL SOUTH F NEW RIV	2014-08-31	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	#90 FPL CANAL	2015-02-							

```
# Count the number of unique values in the column
num_unique_values = resampled_df['Location'].nunique()

# Count the occurrences of each value in the column
value_counts = resampled_df['Location'].value_counts()

print("Number of unique values:", num_unique_values)
print("Occurrences of each value:")
print(value_counts)
```

```
Number of unique values: 60
Occurrences of each value:
Location
#1 HILLSBORO CANAL US 1                     33
#38 ICW 17TH ST CAUSEWAY                    33
#22 N NEW RIVER SW 125 AVE                  33
#10 MIDDLE RIVER E SUNRISE                  33
#25 HOLLYWOOD CANAL STIRLING                33
#28 S NEW RIVER CANAL FLAMINGO              33
#31 SNAKE CRK CANAL FLAMINGO                33
#32 C-9 CANAL US 27                         33
#36 ICW COMMERCIAL BLVD                     33
#37 ICW SUNRISE BLVD                        33
#39 ICW MARKER 35                           33
#17 PLANTATION CANAL @ S-33                 33
#4 HILLSBORO CANAL SE GROWERS               33
#40 ICW SHERIDAN ST                         33
#41 ICW HALLANDALE BCH BLVD                 33
#5 POMPANO CANAL US1                        33
#6 CYPRESS CREEK DIXIE HWY                  33
#7 CYPRESS CREEK S. PALM AIRE               33
#8 POMPANO CANAL US 441                     33
#89 NOB HILL RD POMPANO CANAL               33
#19 NEW RIVER RIVER REACH                   33
#24 DANIA CUT-OFF US-1                      33
#110 POMPANO CANAL AT DIXIE AN              33
#111 S. FORK MID R. @ N.E. 15               33
#12 MIDDLE RIVER NW 31ST AVE                33
#15 NEW RIVER ANDREWS AVE                   33
#16 NORTH FORK BROWARD BLVD                 32
#33 ICW SOUTH OF HILLSBORO BRG              32
#14 MIDDLE RIVER UNIVERS. DRV               32
#11 MIDDLE RIVER NW 21ST AVE                32
#20 N NEW RIV BRADFORD MARINA               32
#23 N NEW RIVER E OF US 27                  32
#112 N. FORK MID R. @ N.E. 16               32
#26 DANIA CUT-OFF RAVENSWOOD                32
#29 S NEW RIVER CANAL US 27                 32
#3 HILLSBORO CANAL US 441                   32
#35 ICW NE 14TH ST POMPANO                  31
#34 ICW NORTH OF MARKER 71                  31
#2 HILLSBORO LOCK                           31
#40 ICW SHERIDAN ST                         27
#35 ICW NE 14TH ST POMPANO                  27
#32 SNAKE CRK CANAL US 27                   27
#64 NORTH FORK AT SISTRUNK                  26
#90 FPL CANAL SOUTH F NEW RIV               21
#122 N NEW RIVER CANAL @ UNIVERSITY DR      18
#121 PLANTATION CANAL @ US 441              18
#125 Las Olas Canal @ NE 21 Ave             18
#123 N NEW RIVER CANAL @ SECRET WOODS       18
#124 S NEW RIVER CANAL @ FL TPK             18
#120 HILLSBORO CANAL W OF DIXIE HWY         18
#49 HENDRICKS ISLE                          15
#21 N NEW RIVER SEWELL LOCKS                15
#125 LAS OLAS CANAL @ NE 21 Ave             12
#126 POMPANO CANAL @ AVONDALE PARK           7
#114 SHERIDAN E OF US 27                     6
```

```
# Find the mode (most frequent value count)
mode_value_count = value_counts.mode()[0]

mode_value_count
```

```
33
```

```
# Keep only the rows with the mode value count
filtered_df = resampled_df[resampled_df['Location'].map(resampled_df['Location'].value_counts()) == mode_value_count]

filtered_df
```

	Location	Sample Date	Chlorophyll A	Dissolved Oxygen	Salinity	Specific Conductance	Total Nitrogen	Total Phosphorus	Turbidity
0	#1 HILLSBORO CANAL US 1	2006-02-28	12.100000	6.980000	19.400000	10.351373	0.830000	0.086000	2.400000
1	#1 HILLSBORO CANAL US 1	2006-08-31	4.430000	5.540000	15.750000	10.165844	0.981000	0.109000	1.400000
2	#1 HILLSBORO CANAL US 1	2007-02-28	2.605000	6.195000	23.200000	10.501905	0.754000	0.083500	1.650000
3	#1 HILLSBORO CANAL US 1	2007-08-31	4.890000	4.730000	31.100000	10.774781	0.777000	0.094000	2.300000
4	#1 HILLSBORO CANAL US 1	2008-02-29	5.925000	5.770000	12.500000	9.893361	1.440000	0.102000	2.050000
...	...	...	...	...	...	...	...	...	...
1559	#89 NOB HILL RD POMPANO CANAL	2020-02-29	3.933333	5.703333	0.310000	6.451930	1.097133	0.014000	0.750000
1560	#89 NOB HILL RD POMPANO CANAL	2020-08-31	1.570000	7.760000	0.250000	6.265301	0.910000	0.011000	0.000000

```
# Calculate maximum and minimum values for each numerical column
max_values = filtered_df[numerical_cols].max()
min_values = filtered_df[numerical_cols].min()

# Print maximum and minimum values for each numerical column
for column in numerical_cols:
    print(f"Maximum value of {column}: {max_values[column]}")
    print(f"Minimum value of {column}: {min_values[column]}")
```

```
Maximum value of Chlorophyll A: 14.6
Minimum value of Chlorophyll A: 0.231
Maximum value of Dissolved Oxygen: 9.1
Minimum value of Dissolved Oxygen: 0.9
Maximum value of Salinity: 35.3
Minimum value of Salinity: 0.18
Maximum value of Specific Conductance: 10.889304347058893
Minimum value of Specific Conductance: 5.958424693029782
Maximum value of Total Nitrogen: 2.38
Minimum value of Total Nitrogen: 0.1634
Maximum value of Total Phosphorus: 0.143
Minimum value of Total Phosphorus: -0.0075
Maximum value of Turbidity: 3.1
Minimum value of Turbidity: 0.0
```

```
cleaned_df = filtered_df.fillna(method="ffill")
```

```
cleaned_df
```

```
/var/folders/9n/nyfs9h7n2lsfs0vd2lq0589h0000gn/T/ipykernel_10894/3588038165.py:1: FutureWarning: DataFrame.fillna with 'meth
  cleaned_df = filtered_df.fillna(method="ffill")
```

	Location	Sample Date	Chlorophyll A	Dissolved Oxygen	Salinity	Specific Conductance	Total Nitrogen	Total Phosphorus	Turbidity
0	#1 HILLSBORO CANAL US 1	2006-02-28	12.100000	6.980000	19.400000	10.351373	0.830000	0.086000	2.400000
1	#1 HILLSBORO CANAL US 1	2006-08-31	4.430000	5.540000	15.750000	10.165844	0.981000	0.109000	1.400000
2	#1 HILLSBORO CANAL US 1	2007-02-28	2.605000	6.195000	23.200000	10.501905	0.754000	0.083500	1.650000
3	#1 HILLSBORO CANAL US 1	2007-08-31	4.890000	4.730000	31.100000	10.774781	0.777000	0.094000	2.300000
4	#1 HILLSBORO CANAL US 1	2008-02-29	5.925000	5.770000	12.500000	9.893361	1.440000	0.102000	2.050000
...	...	...	...	...	...	...	...	...	...
1559	#89 NOB HILL RD POMPANO CANAL	2020-02-29	3.933333	5.703333	0.310000	6.451930	1.097133	0.014000	0.750000
1560	#89 NOB HILL RD POMPANO CANAL	2020-08-31	1.570000	7.760000	0.250000	6.265301	0.910000	0.011000	0.000000
	#89 NOB HILL RD	2021-02-							

```
# Convert DataFrame to CSV
cleaned_df.to_csv('Updated_Dataframe_WaterQual.csv', index=False)
```