Intended Learning Outcome

Perform descriptive and correlation analysis to to analyze the dataset. Interpret the results of descriptive and correlation analysis Resources

- Personal Computer
- Jupyter Notebook
- Internet Connection

Instruction

1. Gather a dataset regarding your identified problem for the ASEAN Data Science Explorer. Make sure that the dataset includes multiple variables.
2. Load the dataset into pandas dataframe.
3. Prepare the data by applying appropriate data preprocessing techniques.
4. Analyze the data using descriptive analysis.
5. Perform correlation analysis.
6. Interpret the results based on the descriptive and correlation analysis.
7. Submit the PDF file.

Instruction

1. Gather a dataset regarding your identified problem for the ASEAN Data Science Explorer. Make sure that the dataset includes multiple variables.
2. Load the dataset into pandas dataframe.
3. Prepare the data by applying appropriate data preprocessing techniques.
4. Analyze the data using descriptive analysis.
5. Perform correlation analysis.
6. Interpret the results based on the descriptive and correlation analysis.
7. Submit the PDF file.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
groundwater = pd.read_csv('/content/gwl-daily.csv')
```

```
groundwater
```

	STATION	MSMT_DATE	WLM_RPE	WLM_RPE_QC	WLM_GSE	WLM_GSE_QC	RPE_WS
0	14N01E35P001M	12/27/2022	48.740	1	46.880	1	42.44
1	14N01E35P004M	12/27/2022	47.620	1	46.880	1	20.78
2	16N03W14H004M	12/27/2022	68.210	1	65.700	1	19.73
3	14N01E35P001M	12/26/2022	48.740	1	46.880	1	42.55
4	14N01E35P004M	12/26/2022	47.620	1	46.880	1	20.82
...	...	...	...	...	...	...	.
1048570	09N03E08C004M	3/26/1992	32.056	2	30.016	2	22.16
1048571	09N03E08C001M	3/25/1992	30.386	2	30.016	2	9.26
1048572	09N03E08C002M	3/25/1992	30.226	2	30.016	2	9.43
1048573	09N03E08C003M	3/25/1992	30.186	2	30.016	2	16.27
1048574	09N03E08C004M	3/25/1992	32.056	2	30.016	2	22.18

1048575 rows × 12 columns

```
# Identifying missing values
print(groundwater.isnull().sum())

# Dropping columns not needed
# groundwater.drop(['column_name'], axis=1, inplace=True)

# Filling missing values (numerical data)
# groundwater['numerical_column'] = groundwater['numerical_column'].fillna(groundwater['numerical_column'].mean())

# Filling missing values (categorical data)
# groundwater['categorical_column'] = groundwater['categorical_column'].fillna('Unknown')

# Correcting data types
# groundwater['date_column'] = pd.to_datetime(groundwater['date_column'])
```

```
STATION           0
MSMT_DATE         0
WLM_RPE        6573
WLM_RPE_QC        0
WLM_GSE        8851
WLM_GSE_QC        0
RPE_WSE       85352
RPE_WSE_QC        0
GSE_WSE       94130
GSE_WSE_QC        0
WSE           91925
WSE_QC            0
dtype: int64
```

```
descriptive_stats = groundwater.describe()

correlation_matrix = groundwater.corr()

descriptive_stats, correlation_matrix
```

```
<ipython-input-86-0c4f7f51067b>:5: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future
  correlation_matrix = groundwater.corr()
(            WLM_RPE    WLM_RPE_QC       WLM_GSE    WLM_GSE_QC        RPE_WSE  \
 count  1.048575e+06  1.048575e+06  1.039724e+06  1.048575e+06  963223.000000
 mean   4.991109e+01  1.242263e+01  4.812979e+01  1.217537e+01      21.899209
 std    4.892829e+01  3.096056e+01  4.899384e+01  3.135395e+01      19.311263
 min   -5.700000e-01  1.000000e+00 -3.170000e+00  1.000000e+00     -12.492000
 25%    2.645000e+01  1.000000e+00  2.500000e+01  1.000000e+00       9.922000
 50%    3.684100e+01  1.000000e+00  3.554600e+01  1.000000e+00      16.798000
 75%    6.821000e+01  2.000000e+00  6.570000e+01  1.000000e+00      27.683000
 max    5.226500e+02  2.550000e+02  5.200000e+02  2.550000e+02     187.804000

          RPE_WSE_QC        GSE_WSE    GSE_WSE_QC            WSE        WSE_QC  \
 count  1.048575e+06  954445.000000  1.048575e+06  956650.000000  1.048575e+06
 mean   2.214869e+01      20.253326  3.277009e+01      27.523056  3.239656e+01
 std    6.805131e+01      19.477138  7.157474e+01      49.853360  7.140478e+01
 min    1.000000e+00     -15.292000  1.000000e+00    -170.798000  1.000000e+00
 25%    1.000000e+00       8.360000  1.000000e+00       2.888000  1.000000e+00
 50%    1.000000e+00      15.187000  1.000000e+00      19.063000  1.000000e+00
 75%    1.000000e+00      26.195000  2.000000e+00      48.565000  2.000000e+00
 max    2.550000e+02     186.534000  2.550000e+02     514.804000  2.550000e+02

        WSE_WLM_RPE_Ratio    WSE_scaled
 count       9.566500e+05  9.566500e+05
 mean                -inf  6.036995e-17
 std                  NaN  1.000001e+00
 min                 -inf -3.978090e+00
 25%         1.441464e-01 -4.941506e-01
 50%         5.248795e-01 -1.696989e-01
 75%         7.450432e-01  4.220770e-01
 max         1.727895e+01  9.774290e+00  ,
                       WLM_RPE  WLM_RPE_QC   WLM_GSE  WLM_GSE_QC   RPE_WSE  \
 WLM_RPE              1.000000   -0.020829  0.999032   -0.032895  0.087920
 WLM_RPE_QC          -0.020829    1.000000 -0.023044    0.958962  0.041957
 WLM_GSE              0.999032   -0.023044  1.000000   -0.045758  0.089491
 WLM_GSE_QC          -0.032895    0.958962 -0.045758    1.000000  0.032320
 RPE_WSE              0.087920    0.041957  0.089491    0.032320  1.000000
 RPE_WSE_QC           0.026859   -0.003679  0.026497   -0.004100 -0.026018
 GSE_WSE              0.077733    0.109300  0.084903    0.096798  0.993460
 GSE_WSE_QC           0.022391    0.394250  0.022021    0.398543  0.031862
 WSE                  0.922254   -0.056759  0.921202   -0.067085 -0.303836
 WSE_QC               0.020878    0.397092  0.022384    0.379989  0.031916
 WSE_WLM_RPE_Ratio    0.247334   -0.039260  0.249044   -0.041110 -0.362052
 WSE_scaled           0.922254   -0.056759  0.921202   -0.067085 -0.303836

                      RPE_WSE_QC   GSE_WSE  GSE_WSE_QC       WSE    WSE_QC  \
 WLM_RPE                0.026859  0.077733    0.022391  0.922254  0.020878
 WLM_RPE_QC            -0.003679  0.109300    0.394250 -0.056759  0.397092
 WLM_GSE                0.026497  0.084903    0.022021  0.921202  0.022384
 WLM_GSE_QC            -0.004100  0.096798    0.398543 -0.067085  0.379989
 RPE_WSE               -0.026018  0.993460    0.031862 -0.303836  0.031916
 RPE_WSE_QC             1.000000 -0.028263    0.905367 -0.001260  0.909147
```

```
GSE_WSE            -0.028263  1.000000    0.090302 -0.309467  0.091879
GSE_WSE_QC          0.905367  0.090302    1.000000 -0.042537  0.995342
WSE                -0.001260 -0.309467   -0.042537  1.000000 -0.048603
WSE_QC              0.909147  0.091879    0.995342 -0.048603  1.000000
```

```
# Extracting only the numerical columns for a cleaner correlation analysis
numerical_columns = groundwater.select_dtypes(include='number')
correlation_matrix = numerical_columns.corr()

correlation_matrix
```

	WLM_RPE	WLM_RPE_QC	WLM_GSE	WLM_GSE_QC	RPE_WSE	RPE_WSE_(
WLM_RPE	1.000000	-0.020829	0.999032	-0.032895	0.087920	0.0268
WLM_RPE_QC	-0.020829	1.000000	-0.023044	0.958962	0.041957	-0.0036
WLM_GSE	0.999032	-0.023044	1.000000	-0.045758	0.089491	0.0264
WLM_GSE_QC	-0.032895	0.958962	-0.045758	1.000000	0.032320	-0.0041(
RPE_WSE	0.087920	0.041957	0.089491	0.032320	1.000000	-0.0260
RPE_WSE_QC	0.026859	-0.003679	0.026497	-0.004100	-0.026018	1.0000(
GSE_WSE	0.077733	0.109300	0.084903	0.096798	0.993460	-0.0282(
GSE_WSE_QC	0.022391	0.394250	0.022021	0.398543	0.031862	0.9053(
WSE	0.922254	-0.056759	0.921202	-0.067085	-0.303836	-0.0012(
WSE_QC	0.020878	0.397092	0.022384	0.379989	0.031916	0.9091(
WSE_WLM_RPE_Ratio	0.247334	-0.039260	0.249044	-0.041110	-0.362052	0.0039(
WSE_scaled	0.922254	-0.056759	0.921202	-0.067085	-0.303836	-0.0012(

```
# Check for missing values
print(groundwater.isnull().sum())

# Fill missing numeric values with the mean or median
groundwater['WLM_RPE'] = groundwater['WLM_RPE'].fillna(groundwater['WLM_RPE'].mean())

# For categorical data, you might want to fill with the most common value or a placeholder like 'Unknown'
# groundwater['category_column'] = groundwater['category_column'].fillna('Unknown')

# Alternatively, if a column has too many missing values, you might decide to drop it
# groundwater.drop(['column_with_many_missing_values'], axis=1, inplace=True)

# Or drop rows with any missing values
# groundwater.dropna(inplace=True)
```

```
STATION                 0
MSMT_DATE               0
WLM_RPE              6573
WLM_RPE_QC              0
WLM_GSE              8851
WLM_GSE_QC              0
RPE_WSE             85352
RPE_WSE_QC              0
GSE_WSE             94130
GSE_WSE_QC              0
WSE                 91925
WSE_QC                  0
WSE_WLM_RPE_Ratio   91925
dtype: int64
```

```
# Descriptive statistics
print(groundwater.describe())

# Data normalization or scaling (if required)
# from sklearn.preprocessing import StandardScaler
# scaler = StandardScaler()
# groundwater[['numerical_column']] = scaler.fit_transform(groundwater[['numerical_column']])
```

```
            WLM_RPE    WLM_RPE_QC       WLM_GSE    WLM_GSE_QC        RPE_WSE  \
count  1.042002e+06  1.048575e+06  1.039724e+06  1.048575e+06  963223.000000
mean   4.991109e+01  1.242263e+01  4.812979e+01  1.217537e+01      21.899209
std    4.908236e+01  3.096056e+01  4.899384e+01  3.135395e+01      19.311263
min   -5.700000e-01  1.000000e+00 -3.170000e+00  1.000000e+00     -12.492000
25%    2.645000e+01  1.000000e+00  2.500000e+01  1.000000e+00       9.922000
50%    3.669200e+01  1.000000e+00  3.554600e+01  1.000000e+00      16.798000
75%    6.821000e+01  2.000000e+00  6.570000e+01  1.000000e+00      27.683000
max    5.226500e+02  2.550000e+02  5.200000e+02  2.550000e+02     187.804000

         RPE_WSE_QC        GSE_WSE    GSE_WSE_QC            WSE        WSE_QC
```

```
count  1.048575e+06  954445.000000  1.048575e+06  956650.000000  1.048575e+06
mean   2.214869e+01      20.253326  3.277009e+01      27.523056  3.239656e+01
std    6.805131e+01      19.477138  7.157474e+01      49.853360  7.140478e+01
min    1.000000e+00     -15.292000  1.000000e+00    -170.798000  1.000000e+00
25%    1.000000e+00       8.360000  1.000000e+00       2.888000  1.000000e+00
50%    1.000000e+00      15.187000  1.000000e+00      19.063000  1.000000e+00
75%    1.000000e+00      26.195000  2.000000e+00      48.565000  2.000000e+00
max    2.550000e+02     186.534000  2.550000e+02     514.804000  2.550000e+02
```

```
# Remove duplicate rows
groundwater.drop_duplicates(inplace=True)
```

```
# Convert data types if necessary
groundwater['MSMT_DATE'] = pd.to_datetime(groundwater['MSMT_DATE'])

# Convert a numeric column to float, if not already
groundwater['WLM_RPE'] = groundwater['WLM_RPE'].astype(float)
```

```
# Identify outliers using Z-score for the 'WSE' column (as previously discussed)
from scipy import stats

z_scores = stats.zscore(groundwater['WSE'])
abs_z_scores = np.abs(z_scores)
outliers = (abs_z_scores > 3)

# Remove outliers
groundwater = groundwater[~outliers]
```

```
# Standardizing a column
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
groundwater['WSE_scaled'] = scaler.fit_transform(groundwater[['WSE']])
```

```
groundwater.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 12 columns):
 #   Column      Non-Null Count    Dtype
---  ------      --------------    -----
 0   STATION     1048575 non-null  object
 1   MSMT_DATE   1048575 non-null  object
 2   WLM_RPE     1042002 non-null  float64
 3   WLM_RPE_QC  1048575 non-null  int64
 4   WLM_GSE     1039724 non-null  float64
 5   WLM_GSE_QC  1048575 non-null  int64
 6   RPE_WSE     963223 non-null   float64
 7   RPE_WSE_QC  1048575 non-null  int64
 8   GSE_WSE     954445 non-null   float64
 9   GSE_WSE_QC  1048575 non-null  int64
 10  WSE         956650 non-null   float64
 11  WSE_QC      1048575 non-null  int64
dtypes: float64(5), int64(5), object(2)
memory usage: 96.0+ MB
```

```
# Creating a new feature by calculating the ratio between 'WSE' and 'WLM_RPE'
groundwater['WSE_WLM_RPE_Ratio'] = groundwater['WSE'] / groundwater['WLM_RPE']
```

```
# Display the first few rows to verify the new column
print(groundwater[['WSE', 'WLM_RPE', 'WSE_WLM_RPE_Ratio']].head())
```

```
     WSE  WLM_RPE  WSE_WLM_RPE_Ratio
0   6.298    48.74           0.129216
1  26.834    47.62           0.563503
2  48.475    68.21           0.710673
3   6.188    48.74           0.126959
4  26.791    47.62           0.562600
```

```
import numpy as np

# Calculate Z-scores of the 'WSE' column
z_scores = stats.zscore(groundwater['WLM_GSE_QC'])
abs_z_scores = np.abs(z_scores)

# Identify outliers (where Z-score is greater than 3)
outliers = (abs_z_scores > 3)

# Print outliers
outliers_df = groundwater[outliers]
```

```
print(outliers_df)
```

```
                STATION  MSMT_DATE    WLM_RPE  WLM_RPE_QC  WLM_GSE  WLM_GSE_QC  \
36      03N04E11L001M 2022-12-15  49.911091         255      NaN         255
39      04N04E10Q001M 2022-12-15  49.911091         255      NaN         255
40      04N04E10Q002M 2022-12-15  49.911091         255      NaN         255
43      04N04E13A001M 2022-12-15  49.911091         255      NaN         255
44      04N04E13A002M 2022-12-15  49.911091         255      NaN         255
...               ...        ...        ...         ...      ...         ...
952711  19N01W22D004M 2006-04-05  89.680000         130    87.38         130
952782  19N01W22D004M 2006-04-04  89.680000         130    87.38         130
952853  19N01W22D004M 2006-04-03  89.680000         130    87.38         130
952924  19N01W22D004M 2006-04-02  89.680000         130    87.38         130
952995  19N01W22D004M 2006-04-01  89.680000         130    87.38         130

        RPE_WSE  RPE_WSE_QC  GSE_WSE  GSE_WSE_QC     WSE  WSE_QC  \
36        8.804           1      NaN         255     NaN     255
39        9.124           1      NaN         255     NaN     255
40        8.695           1      NaN         255     NaN     255
43        7.440           1      NaN         255     NaN     255
44        8.884           1      NaN         255     NaN     255
...         ...         ...      ...         ...     ...     ...
952711    3.529           1    1.229         130  86.151     130
952782    3.659           1    1.359         130  86.021     130
952853    3.777           1    1.477         130  85.903     130
952924    3.897           1    1.597         130  85.783     130
952995    3.944           1    1.644         130  85.736     130

        WSE_WLM_RPE_Ratio  WSE_scaled
36                    NaN         NaN
39                    NaN         NaN
40                    NaN         NaN
43                    NaN         NaN
44                    NaN         NaN
...                   ...         ...
952711           0.960649    1.176008
952782           0.959199    1.173401
952853           0.957884    1.171034
952924           0.956545    1.168627
952995           0.956021    1.167684

[12981 rows x 14 columns]
```

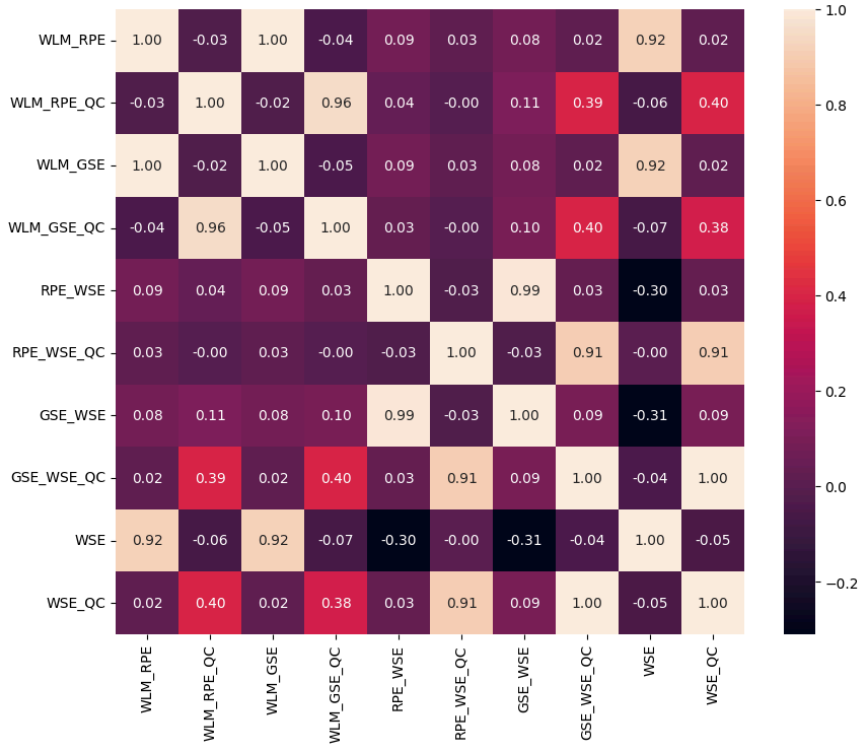
```
# Correlation matrix
corr_matrix = groundwater.corr()
print(corr_matrix)
```

```
<ipython-input-41-3f27efdbb4ed>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
  corr_matrix = groundwater.corr()
             WLM_RPE  WLM_RPE_QC   WLM_GSE  WLM_GSE_QC   RPE_WSE  RPE_WSE_QC  \
WLM_RPE     1.000000   -0.026609  0.999032   -0.041721  0.088094    0.026868
WLM_RPE_QC -0.026609    1.000000 -0.023044    0.958962  0.041957   -0.003679
WLM_GSE     0.999032   -0.023044  1.000000   -0.045758  0.089491    0.026497
WLM_GSE_QC -0.041721    0.958962 -0.045758    1.000000  0.032320   -0.004100
RPE_WSE     0.088094    0.041957  0.089491    0.032320  1.000000   -0.026018
RPE_WSE_QC  0.026868   -0.003679  0.026497   -0.004100 -0.026018    1.000000
GSE_WSE     0.077733    0.109300  0.084903    0.096798  0.993460   -0.028263
GSE_WSE_QC  0.023105    0.394250  0.022021    0.398543  0.031862    0.905367
WSE         0.922254   -0.056759  0.921202   -0.067085 -0.303836   -0.001260
WSE_QC      0.021549    0.397092  0.022384    0.379989  0.031916    0.909147

              GSE_WSE  GSE_WSE_QC       WSE    WSE_QC
WLM_RPE     0.077733    0.023105  0.922254  0.021549
WLM_RPE_QC  0.109300    0.394250 -0.056759  0.397092
WLM_GSE     0.084903    0.022021  0.921202  0.022384
WLM_GSE_QC  0.096798    0.398543 -0.067085  0.379989
RPE_WSE     0.993460    0.031862 -0.303836  0.031916
RPE_WSE_QC -0.028263    0.905367 -0.001260  0.909147
GSE_WSE     1.000000    0.090302 -0.309467  0.091879
GSE_WSE_QC  0.090302    1.000000 -0.042537  0.995342
WSE        -0.309467   -0.042537  1.000000 -0.048603
WSE_QC      0.091879    0.995342 -0.048603  1.000000
```

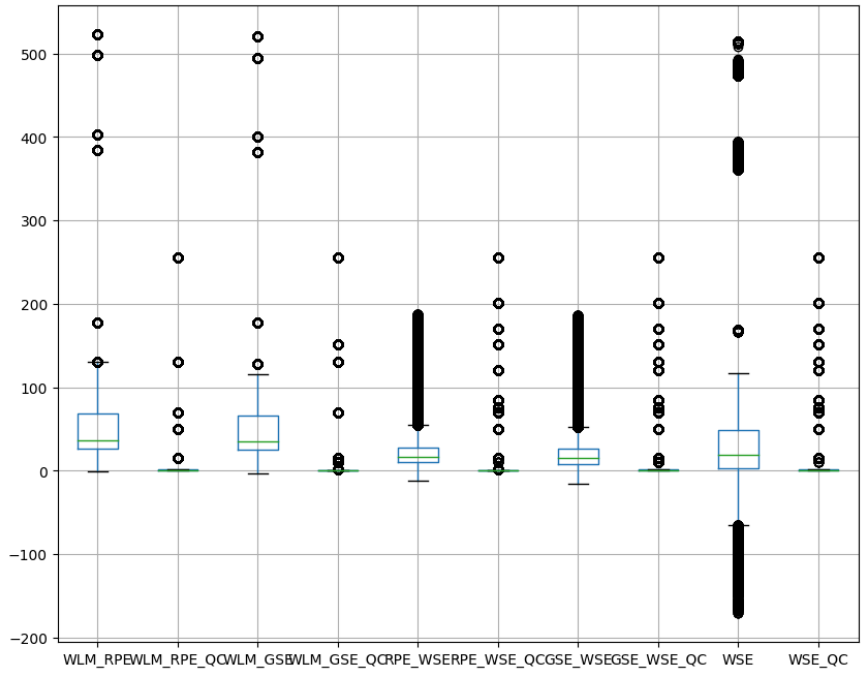
```
# Descriptive statistics for numerical columns
descriptive_stats = groundwater.describe()
print(descriptive_stats)
```

```
            WLM_RPE    WLM_RPE_QC       WLM_GSE    WLM_GSE_QC        RPE_WSE  \
count  1.048575e+06  1.048575e+06  1.039724e+06  1.048575e+06  963223.000000
mean   4.991109e+01  1.242263e+01  4.812979e+01  1.217537e+01      21.899209
std    4.892829e+01  3.096056e+01  4.899384e+01  3.135395e+01      19.311263
min   -5.700000e-01  1.000000e+00 -3.170000e+00  1.000000e+00     -12.492000
25%    2.645000e+01  1.000000e+00  2.500000e+01  1.000000e+00       9.922000
50%    3.684100e+01  1.000000e+00  3.554600e+01  1.000000e+00      16.798000
75%    6.821000e+01  2.000000e+00  6.570000e+01  1.000000e+00      27.683000
max    5.226500e+02  2.550000e+02  5.200000e+02  2.550000e+02     187.804000

          RPE_WSE_QC        GSE_WSE    GSE_WSE_QC            WSE        WSE_QC  \
count  1.048575e+06  954445.000000  1.048575e+06  956650.000000  1.048575e+06
mean   2.214869e+01      20.253326  3.277009e+01      27.523056  3.239656e+01
std    6.805131e+01      19.477138  7.157474e+01      49.853360  7.140478e+01
min    1.000000e+00     -15.292000  1.000000e+00    -170.798000  1.000000e+00
25%    1.000000e+00       8.360000  1.000000e+00       2.888000  1.000000e+00
50%    1.000000e+00      15.187000  1.000000e+00      19.063000  1.000000e+00
75%    1.000000e+00      26.195000  2.000000e+00      48.565000  2.000000e+00
max    2.550000e+02     186.534000  2.550000e+02     514.804000  2.550000e+02

       WSE_WLM_RPE_Ratio    WSE_scaled
count       9.566500e+05  9.566500e+05
mean                -inf  6.036995e-17
std                  NaN  1.000001e+00
min                 -inf -3.978090e+00
25%         1.441464e-01 -4.941506e-01
50%         5.248795e-01 -1.696989e-01
75%         7.450432e-01  4.220770e-01
max         1.727895e+01  9.774290e+00
```

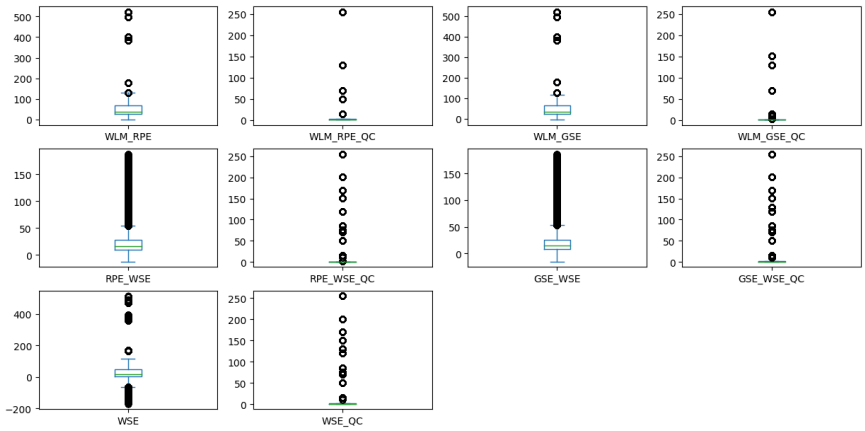
```
# Heatmap for correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, fmt=".2f")
plt.show()
```



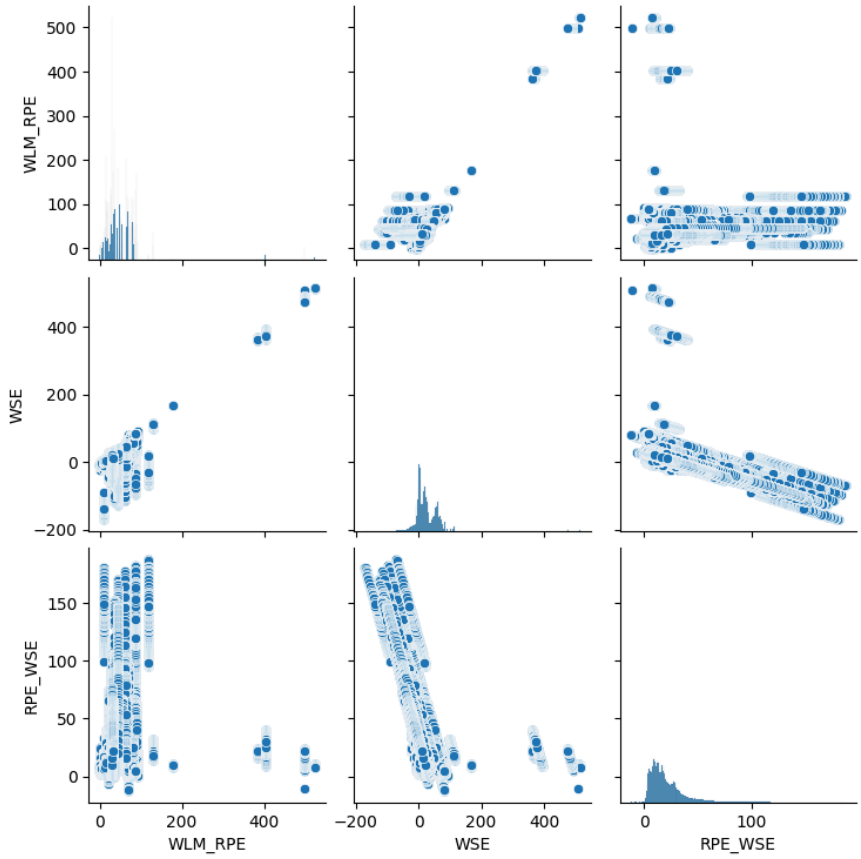

```
# Box plots for numerical data distributions and outliers
groundwater.boxplot(figsize=(10, 8))
plt.show()
```



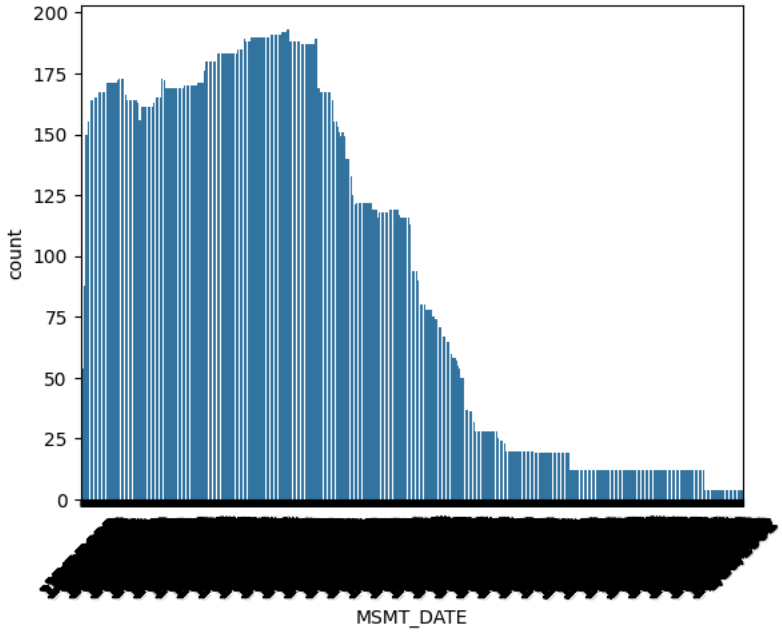

```
# Boxplots for each numerical feature to identify outliers
groundwater.plot(kind='box', subplots=True, layout=(4,4), figsize=(15, 10))
plt.show()
```



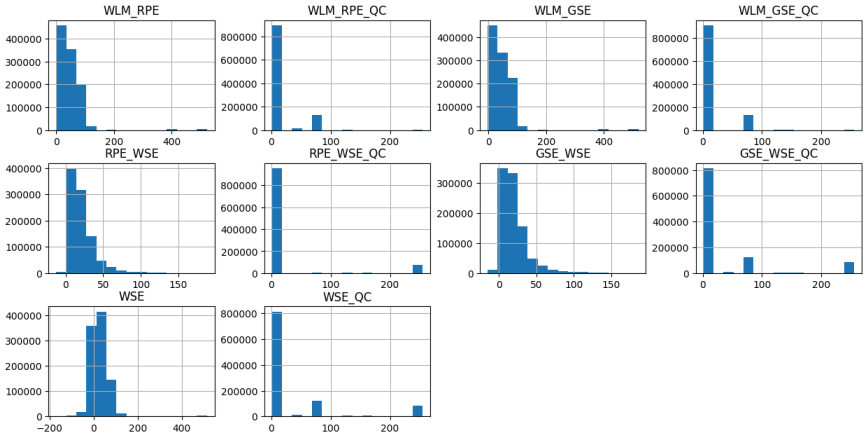

```
import seaborn as sns

# Replace 'WLM_RPE', 'WSE', and 'RPE_WSE' with the actual columns you're interested in
sns.pairplot(groundwater[['WLM_RPE', 'WSE', 'RPE_WSE']])
plt.show()
```



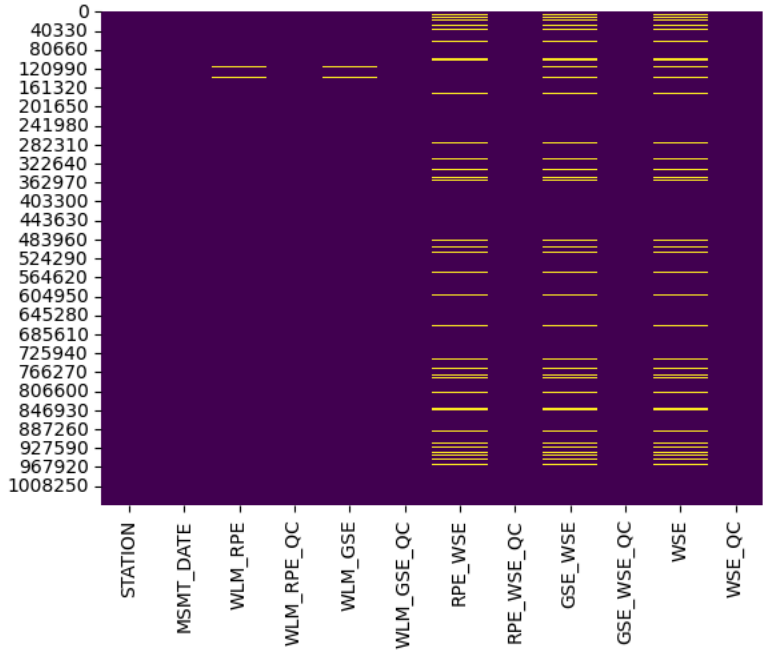

```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'category' is a categorical column in your dataset
sns.countplot(x='MSMT_DATE', data=groundwater)
plt.xticks(rotation=45)
plt.show()
```




```
# Histograms for each numerical feature
groundwater.hist(bins=15, figsize=(15, 10), layout=(4, 4))
plt.show()
```




```
# Visualize missing values
sns.heatmap(groundwater.isnull(), cbar=False, cmap='viridis')
plt.show()
```



The descriptive statistics reveal a wide range in groundwater measurements, indicating diverse levels across locations and times. Mean and median values offer insight into general groundwater levels, with standard deviation indicating variability. High correlation between **WLM_RPE and WLM_GSE** suggests they provide similar elevation information. **RPE_WSE and GSE_WSE** show consistent water surface elevation measurements. Positive correlation of **WSE** with elevation measures indicates higher ground elevations coincide with higher water levels. Quality control metrics demonstrate consistent assessment approaches. Overall, these analyses inform environmental studies, aiding in groundwater management and conservation policies.