step : Imports & Reading Data

In [1]:
```
pip install pandas sqlite3 matplotlib seaborn
```

```
Requirement already satisfied: pandas in c:₩users₩seoin₩anaconda₩lib₩site-packages
(2.2.2)
Note: you may need to restart the kernel to use updated packages.
ERROR: Could not find a version that satisfies the requirement sqlite3 (from version
s: none)
ERROR: No matching distribution found for sqlite3
```

In [2]:
```
import pandas as pd
import sqlite3
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy import stats
pd.set_option('display.max_columns', None)
```

```
C:₩Users₩seoin₩anaconda₩Lib₩site-packages₩pandas₩core₩arrays₩masked.py:60: UserWarni
ng: Pandas requires version '1.3.6' or newer of 'bottleneck' (version '1.3.5' curren
tly installed).
  from pandas.core import (
```

Step : Data Understanding

In [3]:
```
# Connect to SQLite database
conn = sqlite3.connect('cruise_data.db')
cursor = conn.cursor()

# Load CSV data into a DataFrame
df = pd.read_csv('C:/Users/seoin/Desktop/Tui/task_data/data.csv')

# Create a table in the SQLite database
df.to_sql('cruise_data', conn, if_exists='replace', index=False)

# Verify the data is loaded
query = "SELECT * FROM cruise_data"
df_sql = pd.read_sql_query(query, conn)
print(df_sql.head())
```

	Start Time	End Time	Vessel Name	Power Galley 1 (MW)	₩
0	2023-01-01T00:00:00	2023-01-01T00:05:00	Vessel 1	0.0946	
1	2023-01-01T00:05:00	2023-01-01T00:10:00	Vessel 1	0.0540	
2	2023-01-01T00:10:00	2023-01-01T00:15:00	Vessel 1	0.0439	
3	2023-01-01T00:15:00	2023-01-01T00:20:00	Vessel 1	0.0733	
4	2023-01-01T00:20:00	2023-01-01T00:25:00	Vessel 1	0.0780	

	Power Galley 2 (MW)	Power Service (MW)	HVAC Chiller 1 Power (MW)	₩
0	0.1384	5.4654	0.5074	
1	0.1370	5.4387	0.5158	
2	0.1785	5.5265	0.5117	
3	0.1725	5.5257	0.5177	
4	0.1397	5.4634	0.5169	

	HVAC Chiller 2 Power (MW)	HVAC Chiller 3 Power (MW)	Scrubber Power (MW)	₩
0	0.0	0.4979	0.4191	
1	0.0	0.4982	0.4204	
2	0.0	0.5032	0.4199	
3	0.0	0.5103	0.4188	
4	0.0	0.5100	0.4203	

	Sea Temperature (Celsius)	Boiler 1 Fuel Flow Rate (L/h)	₩
0	27.3000	0.0000	
1	27.3000	47.7695	
2	27.3000	77.2034	
3	27.3076	60.6369	
4	27.3518	55.2184	

	Boiler 2 Fuel Flow Rate (L/h)	Incinerator 1 Fuel Flow Rate (L/h)	₩
0	0.0	19.0090	
1	0.0	216.3180	
2	0.0	439.4300	
3	0.0	218.2797	
4	0.0	0.0000	

	Diesel Generator 1 Power (MW)	Diesel Generator 2 Power (MW)	₩
0	0.0	0.0	
1	0.0	0.0	
2	0.0	0.0	
3	0.0	0.0	
4	0.0	0.0	

	Diesel Generator 3 Power (MW)	Diesel Generator 4 Power (MW)	₩
0	0.0	7.3349	
1	0.0	7.3011	
2	0.0	7.3299	
3	0.0	7.3712	
4	0.0	7.3032	

	Latitude (Degrees)	Longitude (Degrees)	Relative Wind Angle (Degrees)	₩
0	17.72523	-65.45738	8.4428	
1	17.73088	-65.44803	41.3100	
2	17.73655	-65.43887	23.9997	
3	17.74202	-65.42980	14.5540	
4	17.74713	-65.42042	14.5632	

	True Wind Angle (Degrees)	Depth (m)	Relative Wind Direction (Degrees)	₩
0	10.9049	NaN	64.3112	
1	78.7817	NaN	62.8161	
2	33.6216	NaN	80.7356	
3	20.0348	NaN	75.9723	
4	20.0328	NaN	74.6509	

	True Wind Direction (Degrees)	Draft (m)	Speed Over Ground (knots)	₩

0		66.7735	7.8721	7.6300
1		64.3452	7.8713	7.5800
2		90.3574	7.8718	7.4379
3		81.4529	7.8710	7.3979
4		80.1204	7.8707	7.4343

	True Wind Speed (knots)	Relative Wind Speed (knots)	₩
0	19.5050	27.0579	
1	19.2968	26.8067	
2	19.4491	25.8380	
3	20.6231	27.6498	
4	20.4554	27.5341	

	Speed Through Water (knots)	Local Time (h)	Trim (m)	₩
0	7.8881	19.67367	-0.1425	
1	7.7438	19.75763	-0.1405	
2	7.6320	19.84158	-0.1450	
3	7.5080	19.92551	-0.1308	
4	7.5521	20.00947	-0.1269	

	Propulsion Power (MW)	Port Side Propulsion Power (MW)	₩
0	1.8691	0.8854	
1	1.8622	0.8737	
2	1.8036	0.8441	
3	1.8457	0.8543	
4	1.8399	0.8467	

	Starboard Side Propulsion Power (MW)	Bow Thruster 1 Power (MW)	₩
0	0.9837	0.0	
1	0.9885	0.0	
2	0.9595	0.0	
3	0.9914	0.0	
4	0.9932	0.0	

	Bow Thruster 2 Power (MW)	Bow Thruster 3 Power (MW)	₩
0	0.0	0.0	
1	0.0	0.0	
2	0.0	0.0	
3	0.0	0.0	
4	0.0	0.0	

	Stern Thruster 1 Power (MW)	Stern Thruster 2 Power (MW)	₩
0	0.0	0.0	
1	0.0	0.0	
2	0.0	0.0	
3	0.0	0.0	
4	0.0	0.0	

	Main Engine 1 Fuel Flow Rate (kg/h)	Main Engine 2 Fuel Flow Rate (kg/h)	₩
0	0.0	0.0	
1	0.0	0.0	
2	0.0	0.0	
3	0.0	0.0	
4	0.0	0.0	

	Main Engine 3 Fuel Flow Rate (kg/h)	Main Engine 4 Fuel Flow Rate (kg/h)
0	0.0	1645.82000
1	0.0	1643.78999
2	0.0	1642.07000
3	0.0	1650.71000
4	0.0	1644.54000

In [4]:
```
df.shape
```

Out[4]: (210240, 44)

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210240 entries, 0 to 210239
Data columns (total 44 columns):
 #   Column                                Non-Null Count   Dtype  
---  ------                                --------------   -----  
 0   Start Time                            210240 non-null  object 
 1   End Time                              210240 non-null  object 
 2   Vessel Name                           210240 non-null  object 
 3   Power Galley 1 (MW)                   210224 non-null  float64
 4   Power Galley 2 (MW)                   210224 non-null  float64
 5   Power Service (MW)                    210222 non-null  float64
 6   HVAC Chiller 1 Power (MW)             210033 non-null  float64
 7   HVAC Chiller 2 Power (MW)             210033 non-null  float64
 8   HVAC Chiller 3 Power (MW)             210033 non-null  float64
 9   Scrubber Power (MW)                   210224 non-null  float64
 10  Sea Temperature (Celsius)             210224 non-null  float64
 11  Boiler 1 Fuel Flow Rate (L/h)         210224 non-null  float64
 12  Boiler 2 Fuel Flow Rate (L/h)         210224 non-null  float64
 13  Incinerator 1 Fuel Flow Rate (L/h)    210224 non-null  float64
 14  Diesel Generator 1 Power (MW)         210224 non-null  float64
 15  Diesel Generator 2 Power (MW)         210224 non-null  float64
 16  Diesel Generator 3 Power (MW)         210224 non-null  float64
 17  Diesel Generator 4 Power (MW)         210224 non-null  float64
 18  Latitude (Degrees)                    209900 non-null  float64
 19  Longitude (Degrees)                   209900 non-null  float64
 20  Relative Wind Angle (Degrees)         210226 non-null  float64
 21  True Wind Angle (Degrees)             210166 non-null  float64
 22  Depth (m)                             152746 non-null  float64
 23  Relative Wind Direction (Degrees)     210185 non-null  float64
 24  True Wind Direction (Degrees)         210166 non-null  float64
 25  Draft (m)                             209097 non-null  float64
 26  Speed Over Ground (knots)             209340 non-null  float64
 27  True Wind Speed (knots)               210166 non-null  float64
 28  Relative Wind Speed (knots)           210226 non-null  float64
 29  Speed Through Water (knots)           209299 non-null  float64
 30  Local Time (h)                        209900 non-null  float64
 31  Trim (m)                              209161 non-null  float64
 32  Propulsion Power (MW)                 210224 non-null  float64
 33  Port Side Propulsion Power (MW)       210224 non-null  float64
 34  Starboard Side Propulsion Power (MW)  210224 non-null  float64
 35  Bow Thruster 1 Power (MW)             210224 non-null  float64
 36  Bow Thruster 2 Power (MW)             210224 non-null  float64
 37  Bow Thruster 3 Power (MW)             210224 non-null  float64
 38  Stern Thruster 1 Power (MW)           210224 non-null  float64
 39  Stern Thruster 2 Power (MW)           210224 non-null  float64
 40  Main Engine 1 Fuel Flow Rate (kg/h)   210224 non-null  float64
 41  Main Engine 2 Fuel Flow Rate (kg/h)   210224 non-null  float64
 42  Main Engine 3 Fuel Flow Rate (kg/h)   210224 non-null  float64
 43  Main Engine 4 Fuel Flow Rate (kg/h)   210224 non-null  float64
dtypes: float64(41), object(3)
memory usage: 70.6+ MB
```

In [6]:
```
df.head()
```

Out[6]:

	Start Time	End Time	Vessel Name	Power Galley 1 (MW)	Power Galley 2 (MW)	Power Service (MW)	HVAC Chiller 1 Power (MW)	HVAC Chiller 2 Power (MW)	HVAC Chiller 3 Power (MW)	Scrubber Power (MW)	Tem
0	2023-01-01T00:00:00	2023-01-01T00:05:00	Vessel 1	0.0946	0.1384	5.4654	0.5074	0.0	0.4979	0.4191	
1	2023-01-01T00:05:00	2023-01-01T00:10:00	Vessel 1	0.0540	0.1370	5.4387	0.5158	0.0	0.4982	0.4204	
2	2023-01-01T00:10:00	2023-01-01T00:15:00	Vessel 1	0.0439	0.1785	5.5265	0.5117	0.0	0.5032	0.4199	
3	2023-01-01T00:15:00	2023-01-01T00:20:00	Vessel 1	0.0733	0.1725	5.5257	0.5177	0.0	0.5103	0.4188	
4	2023-01-01T00:20:00	2023-01-01T00:25:00	Vessel 1	0.0780	0.1397	5.4634	0.5169	0.0	0.5100	0.4203	

In [7]:
```
df.columns
```

Out[7]:
```
Index(['Start Time', 'End Time', 'Vessel Name', 'Power Galley 1 (MW)',
       'Power Galley 2 (MW)', 'Power Service (MW)',
       'HVAC Chiller 1 Power (MW)', 'HVAC Chiller 2 Power (MW)',
       'HVAC Chiller 3 Power (MW)', 'Scrubber Power (MW)',
       'Sea Temperature (Celsius)', 'Boiler 1 Fuel Flow Rate (L/h)',
       'Boiler 2 Fuel Flow Rate (L/h)', 'Incinerator 1 Fuel Flow Rate (L/h)',
       'Diesel Generator 1 Power (MW)', 'Diesel Generator 2 Power (MW)',
       'Diesel Generator 3 Power (MW)', 'Diesel Generator 4 Power (MW)',
       'Latitude (Degrees)', 'Longitude (Degrees)',
       'Relative Wind Angle (Degrees)', 'True Wind Angle (Degrees)',
       'Depth (m)', 'Relative Wind Direction (Degrees)',
       'True Wind Direction (Degrees)', 'Draft (m)',
       'Speed Over Ground (knots)', 'True Wind Speed (knots)',
       'Relative Wind Speed (knots)', 'Speed Through Water (knots)',
       'Local Time (h)', 'Trim (m)', 'Propulsion Power (MW)',
       'Port Side Propulsion Power (MW)',
       'Starboard Side Propulsion Power (MW)', 'Bow Thruster 1 Power (MW)',
       'Bow Thruster 2 Power (MW)', 'Bow Thruster 3 Power (MW)',
       'Stern Thruster 1 Power (MW)', 'Stern Thruster 2 Power (MW)',
       'Main Engine 1 Fuel Flow Rate (kg/h)',
       'Main Engine 2 Fuel Flow Rate (kg/h)',
       'Main Engine 3 Fuel Flow Rate (kg/h)',
       'Main Engine 4 Fuel Flow Rate (kg/h)'],
      dtype='object')
```

Step : Data Preperation

In [8]:
```
# Convert time columns to datetime dtype
df['Start Time'] = pd.to_datetime(df['Start Time'])
df['End Time'] = pd.to_datetime(df['End Time'])

# Display info for only 'Start Time' and 'End Time' columns
df[['Start Time', 'End Time']].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210240 entries, 0 to 210239
Data columns (total 2 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   Start Time  210240 non-null  datetime64[ns]
 1   End Time    210240 non-null  datetime64[ns]
dtypes: datetime64[ns](2)
memory usage: 3.2 MB
```

Missing value check

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
Start Time                                0
End Time                                  0
Vessel Name                               0
Power Galley 1 (MW)                      16
Power Galley 2 (MW)                      16
Power Service (MW)                       18
HVAC Chiller 1 Power (MW)               207
HVAC Chiller 2 Power (MW)               207
HVAC Chiller 3 Power (MW)               207
Scrubber Power (MW)                      16
Sea Temperature (Celsius)                16
Boiler 1 Fuel Flow Rate (L/h)            16
Boiler 2 Fuel Flow Rate (L/h)            16
Incinerator 1 Fuel Flow Rate (L/h)       16
Diesel Generator 1 Power (MW)            16
Diesel Generator 2 Power (MW)            16
Diesel Generator 3 Power (MW)            16
Diesel Generator 4 Power (MW)            16
Latitude (Degrees)                      340
Longitude (Degrees)                     340
Relative Wind Angle (Degrees)            14
True Wind Angle (Degrees)                74
Depth (m)                             57494
Relative Wind Direction (Degrees)        55
True Wind Direction (Degrees)            74
Draft (m)                              1143
Speed Over Ground (knots)               900
True Wind Speed (knots)                  74
Relative Wind Speed (knots)              14
Speed Through Water (knots)             941
Local Time (h)                          340
Trim (m)                               1079
Propulsion Power (MW)                    16
Port Side Propulsion Power (MW)          16
Starboard Side Propulsion Power (MW)     16
Bow Thruster 1 Power (MW)                16
Bow Thruster 2 Power (MW)                16
Bow Thruster 3 Power (MW)                16
Stern Thruster 1 Power (MW)              16
Stern Thruster 2 Power (MW)              16
Main Engine 1 Fuel Flow Rate (kg/h)      16
Main Engine 2 Fuel Flow Rate (kg/h)      16
Main Engine 3 Fuel Flow Rate (kg/h)      16
Main Engine 4 Fuel Flow Rate (kg/h)      16
dtype: int64
```

In [10]:
```
df.duplicated().sum()
```

Out[10]: 0

In [11]:
```
df[df['Power Galley 1 (MW)'].isna()]
```

Out[11]:

	Start Time	End Time	Vessel Name	Power Galley 1 (MW)	Power Galley 2 (MW)	Power Service (MW)	HVAC Chiller 1 Power (MW)	HVAC Chiller 2 Power (MW)	HVAC Chiller 3 Power (MW)	Scrubber Power (MW)	Temp
88578	2023-11-04 13:30:00	2023-11-04 13:35:00	Vessel 1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
140167	2023-05-02 16:35:00	2023-05-02 16:40:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
145233	2023-05-20 06:45:00	2023-05-20 06:50:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
161071	2023-07-14 06:35:00	2023-07-14 06:40:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
163498	2023-07-22 16:50:00	2023-07-22 16:55:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
163499	2023-07-22 16:55:00	2023-07-22 17:00:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
163500	2023-07-22 17:00:00	2023-07-22 17:05:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
165191	2023-07-28 13:55:00	2023-07-28 14:00:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
165767	2023-07-30 13:55:00	2023-07-30 14:00:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
165768	2023-07-30 14:00:00	2023-07-30 14:05:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
168077	2023-08-07 14:25:00	2023-08-07 14:30:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
168078	2023-08-07 14:30:00	2023-08-07 14:35:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
179568	2023-09-16 12:00:00	2023-09-16 12:05:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
179569	2023-09-16 12:05:00	2023-09-16 12:10:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
179572	2023-09-16 12:20:00	2023-09-16 12:25:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

	Start Time	End Time	Vessel Name	Power Galley 1 (MW)	Power Galley 2 (MW)	Power Service (MW)	HVAC Chiller 1 Power (MW)	HVAC Chiller 2 Power (MW)	HVAC Chiller 3 Power (MW)	Scrubber Power (MW)	Temp
209699	2023-12-30 02:55:00	2023-12-30 03:00:00	Vessel 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

In [12]:
```
# Drop rows where 'Power Galley 1 (MW)' has missing values
df = df.dropna(subset=['Power Galley 1 (MW)'])
```

In [13]:
```
df[df['HVAC Chiller 1 Power (MW)'].isna()]
```

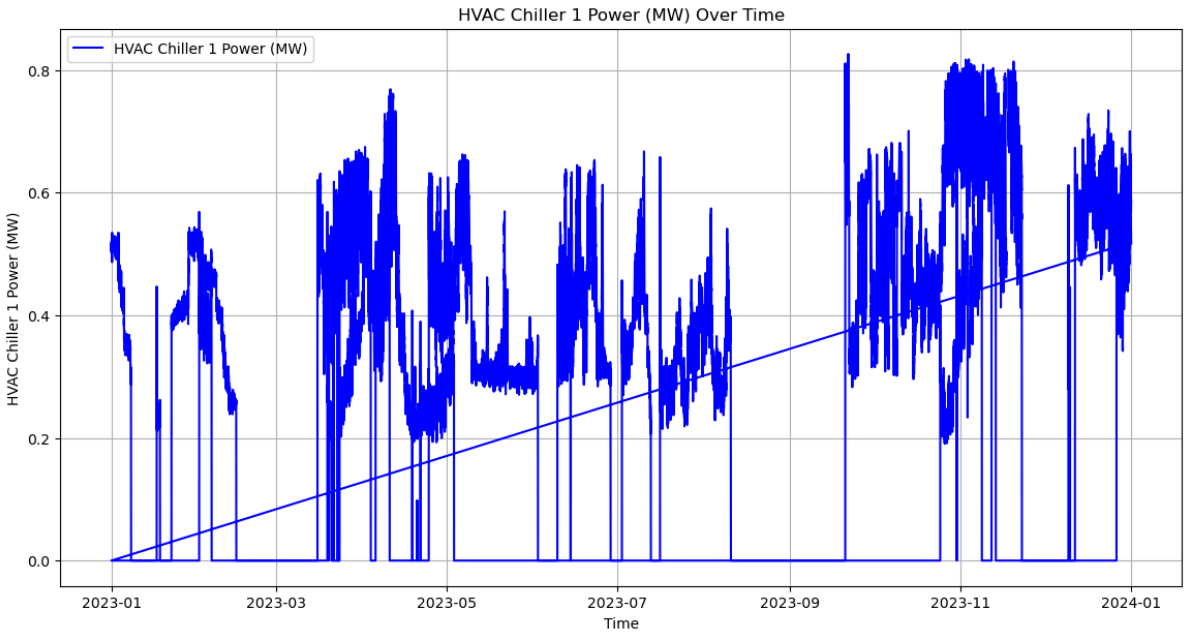
Out[13]:

	Start Time	End Time	Vessel Name	Power Galley 1 (MW)	Power Galley 2 (MW)	Power Service (MW)	HVAC Chiller 1 Power (MW)	HVAC Chiller 2 Power (MW)	HVAC Chiller 3 Power (MW)	Scrubber Power (MW)	Temp
197377	2023-11-17 08:05:00	2023-11-17 08:10:00	Vessel 2	0.0000	0.0918	4.4755	NaN	NaN	NaN	0.1543	
197378	2023-11-17 08:10:00	2023-11-17 08:15:00	Vessel 2	0.0000	0.0716	4.5178	NaN	NaN	NaN	0.1534	
197379	2023-11-17 08:15:00	2023-11-17 08:20:00	Vessel 2	0.0000	0.0638	4.4713	NaN	NaN	NaN	0.1530	
197380	2023-11-17 08:20:00	2023-11-17 08:25:00	Vessel 2	0.0071	0.0620	4.5521	NaN	NaN	NaN	0.1545	
197381	2023-11-17 08:25:00	2023-11-17 08:30:00	Vessel 2	0.0012	0.0420	4.4380	NaN	NaN	NaN	0.1539	
...	...	...	...	...	...	...	...	...	...	...	
197563	2023-11-17 23:35:00	2023-11-17 23:40:00	Vessel 2	0.0218	0.1772	5.5094	NaN	NaN	NaN	0.1577	
197564	2023-11-17 23:40:00	2023-11-17 23:45:00	Vessel 2	0.0487	0.1671	5.5004	NaN	NaN	NaN	0.1569	
197565	2023-11-17 23:45:00	2023-11-17 23:50:00	Vessel 2	0.0308	0.1678	5.5444	NaN	NaN	NaN	0.1581	
197566	2023-11-17 23:50:00	2023-11-17 23:55:00	Vessel 2	0.0497	0.1516	5.5213	NaN	NaN	NaN	0.1584	
197567	2023-11-17 23:55:00	2023-11-18 00:00:00	Vessel 2	0.0697	0.2012	5.6888	NaN	NaN	NaN	0.1585	

191 rows × 44 columns

In [14]:

```
# Ensure 'Start Time' is in datetime format
df['Start Time'] = pd.to_datetime(df['Start Time'])

# Plot the time series for 'HVAC Chiller 1 Power (MW)'
plt.figure(figsize=(14, 7))
plt.plot(df['Start Time'], df['HVAC Chiller 1 Power (MW)'], label='HVAC Chiller 1 Po
plt.xlabel('Time')
plt.ylabel('HVAC Chiller 1 Power (MW)')
plt.title('HVAC Chiller 1 Power (MW) Over Time')
plt.legend()
plt.grid(True)
plt.show()
```



In [15]:
```
# Sort the dataframe by 'Start Time' to ensure proper filling
df = df.sort_values(by='Start Time')

# Set 'Start Time' as the index
df.set_index('Start Time', inplace=True)

# Interpolate to fill missing values for chiller columns
df['HVAC Chiller 1 Power (MW)'] = df['HVAC Chiller 1 Power (MW)'].interpolate()
df['HVAC Chiller 2 Power (MW)'] = df['HVAC Chiller 2 Power (MW)'].interpolate()
df['HVAC Chiller 3 Power (MW)'] = df['HVAC Chiller 3 Power (MW)'].interpolate()

# If you need to reset the index back to columns
df.reset_index(inplace=True)
```

In [16]:
```
df[df['Power Service (MW)'].isna()]
```

Out[16]:

	Start Time	End Time	Vessel Name	Power Galley 1 (MW)	Power Galley 2 (MW)	Power Service (MW)	HVAC Chiller 1 Power (MW)	HVAC Chiller 2 Power (MW)	HVAC Chiller 3 Power (MW)	Scrubber Power (MW)	Temp
148890	2023-09-16 12:15:00	2023-09-16 12:20:00	Vessel 2	0.03	0.080	NaN	0.000	0.384	0.0	0.773	
209145	2023-12-30 03:00:00	2023-12-30 03:05:00	Vessel 2	0.03	0.087	NaN	0.508	0.528	0.0	0.346	

In [17]:
```
# First forward fill, then backward fill
df['Power Service (MW)'] = df['Power Service (MW)'].ffill().bfill()
```

In [18]:
```
df[df['Speed Over Ground (knots)'].isna()]
```

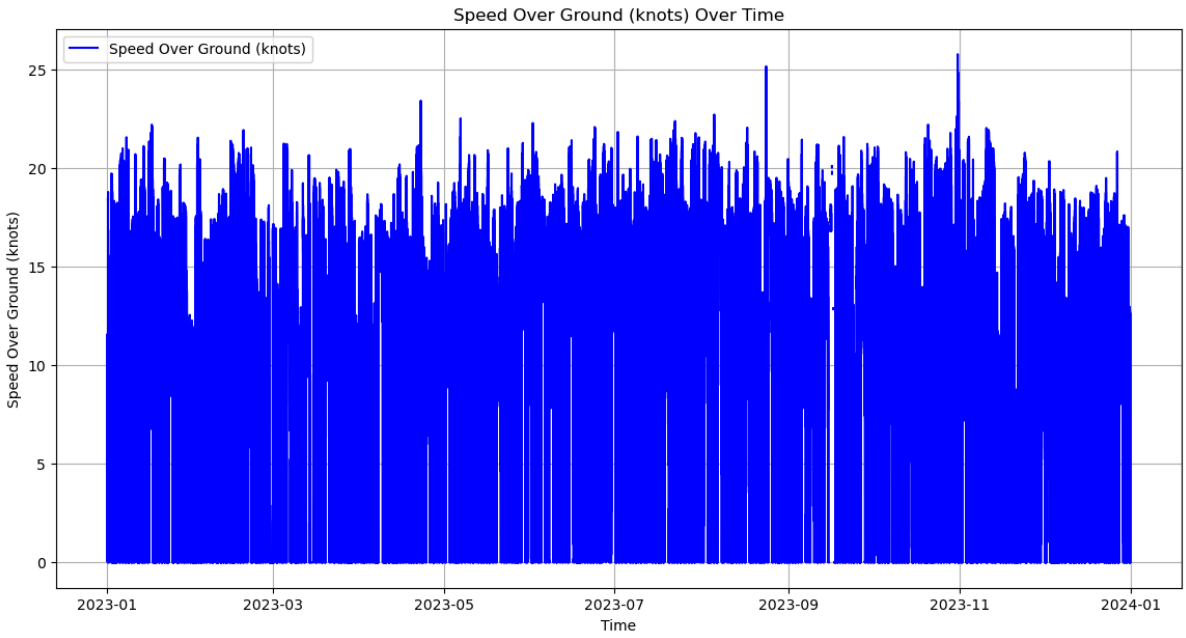
Out[18]:

	Start Time	End Time	Vessel Name	Power Galley 1 (MW)	Power Galley 2 (MW)	Power Service (MW)	HVAC Chiller 1 Power (MW)	HVAC Chiller 2 Power (MW)	HVAC Chiller 3 Power (MW)	Scrubber Power (MW)	Temp
137881	2023-08-28 09:30:00	2023-08-28 09:35:00	Vessel 2	0.1025	0.1235	-0.0400	0.0000	0.0	0.4086	0.0000	
137884	2023-08-28 09:35:00	2023-08-28 09:40:00	Vessel 2	0.1113	0.1194	-0.0400	0.0000	0.0	0.4061	0.0000	
137885	2023-08-28 09:40:00	2023-08-28 09:45:00	Vessel 2	0.0895	0.0986	-0.0400	0.0000	0.0	0.4006	0.0000	
137887	2023-08-28 09:45:00	2023-08-28 09:50:00	Vessel 2	0.1265	0.1295	-0.0400	0.0000	0.0	0.3959	0.0000	
137889	2023-08-28 09:50:00	2023-08-28 09:55:00	Vessel 2	0.1095	0.1345	-0.0400	0.0000	0.0	0.3969	0.0000	
...	...	...	...	...	...	...	...	...	...	...	
153941	2023-09-25 06:45:00	2023-09-25 06:50:00	Vessel 2	0.0281	0.1347	5.8139	0.4105	0.0	0.0000	0.7854	
153943	2023-09-25 06:50:00	2023-09-25 06:55:00	Vessel 2	0.0195	0.1565	5.8844	0.4183	0.0	0.0000	0.7832	
153945	2023-09-25 06:55:00	2023-09-25 07:00:00	Vessel 2	0.0317	0.1817	5.8343	0.4083	0.0	0.0000	0.7865	
153947	2023-09-25 07:00:00	2023-09-25 07:05:00	Vessel 2	0.0166	0.2210	5.9644	0.4213	0.0	0.0000	0.7857	
153949	2023-09-25 07:05:00	2023-09-25 07:10:00	Vessel 2	0.0486	0.2097	5.9657	0.4175	0.0	0.0000	0.7886	

886 rows × 44 columns

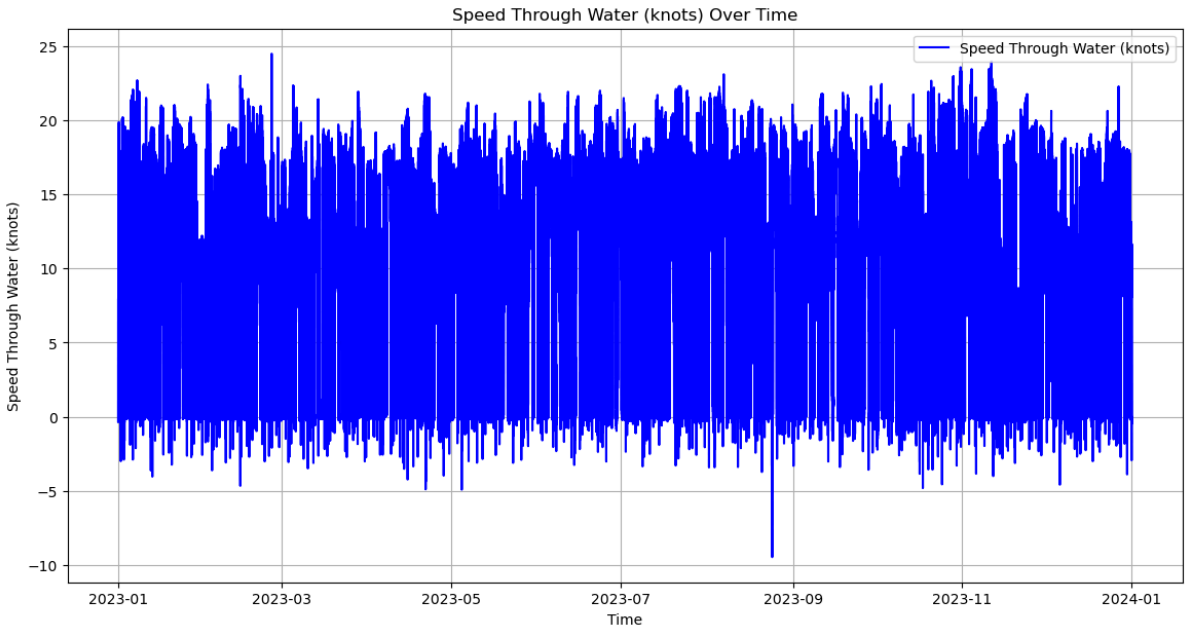
In [19]:

```
# Plot the time series for 'Speed Over Ground (knots)'
plt.figure(figsize=(14, 7))
plt.plot(df['Start Time'], df['Speed Over Ground (knots)'], label='Speed Over Ground
plt.xlabel('Time')
plt.ylabel('Speed Over Ground (knots)')
plt.title('Speed Over Ground (knots) Over Time')
plt.legend()
plt.grid(True)
plt.show()
```



In [20]:
```
# Sort the dataframe by 'Start Time' to ensure proper filling
df = df.sort_values(by='Start Time')

# Set 'Start Time' as the index
df.set_index('Start Time', inplace=True)

# Interpolate to fill missing values
df['Speed Over Ground (knots)'] = df['Speed Over Ground (knots)'].interpolate()

# If you need to reset the index back to columns
df.reset_index(inplace=True)
```

In [21]:
```
# Set 'Start Time' as the index
df.set_index('Start Time', inplace=True)

# Interpolate to fill missing values
df['Speed Through Water (knots)'] = df['Speed Through Water (knots)'].interpolate()

# If you need to reset the index back to columns
df.reset_index(inplace=True)

# Plot the time series for 'Speed Through Water (knots)'
plt.figure(figsize=(14, 7))
plt.plot(df['Start Time'], df['Speed Through Water (knots)'], label='Speed Through W
plt.xlabel('Time')
plt.ylabel('Speed Through Water (knots)')
plt.title('Speed Through Water (knots) Over Time')
plt.legend()
plt.grid(True)
plt.show()
```



In [22]:
```
# Set 'Start Time' as the index
df.set_index('Start Time', inplace=True)

# Interpolate to fill missing values
df['Speed Through Water (knots)'] = df['Speed Through Water (knots)'].interpolate()

# If you need to reset the index back to columns
df.reset_index(inplace=True)
```

In [23]:
```
df.isnull().sum()
```

Out[23]:
```
Start Time                                0
End Time                                  0
Vessel Name                               0
Power Galley 1 (MW)                       0
Power Galley 2 (MW)                       0
Power Service (MW)                        0
HVAC Chiller 1 Power (MW)                 0
HVAC Chiller 2 Power (MW)                 0
HVAC Chiller 3 Power (MW)                 0
Scrubber Power (MW)                       0
Sea Temperature (Celsius)                 0
Boiler 1 Fuel Flow Rate (L/h)             0
Boiler 2 Fuel Flow Rate (L/h)             0
Incinerator 1 Fuel Flow Rate (L/h)        0
Diesel Generator 1 Power (MW)             0
Diesel Generator 2 Power (MW)             0
Diesel Generator 3 Power (MW)             0
Diesel Generator 4 Power (MW)             0
Latitude (Degrees)                      326
Longitude (Degrees)                     326
Relative Wind Angle (Degrees)             0
True Wind Angle (Degrees)                60
Depth (m)                             57479
Relative Wind Direction (Degrees)        41
True Wind Direction (Degrees)            60
Draft (m)                              1127
Speed Over Ground (knots)                 0
True Wind Speed (knots)                  60
Relative Wind Speed (knots)               0
Speed Through Water (knots)               0
Local Time (h)                          326
Trim (m)                               1063
Propulsion Power (MW)                     0
Port Side Propulsion Power (MW)           0
Starboard Side Propulsion Power (MW)      0
Bow Thruster 1 Power (MW)                 0
Bow Thruster 2 Power (MW)                 0
Bow Thruster 3 Power (MW)                 0
Stern Thruster 1 Power (MW)               0
Stern Thruster 2 Power (MW)               0
Main Engine 1 Fuel Flow Rate (kg/h)       0
Main Engine 2 Fuel Flow Rate (kg/h)       0
Main Engine 3 Fuel Flow Rate (kg/h)       0
Main Engine 4 Fuel Flow Rate (kg/h)       0
dtype: int64
```

Sorting Vessels data to Vessel 1 & Vessel 2

In [24]:
```
# Assuming 'Vessel Name' is the column indicating the vessel
vessel_1_data = df[df['Vessel Name'] == 'Vessel 1']
vessel_2_data = df[df['Vessel Name'] == 'Vessel 2']

# Verify the split
print(vessel_1_data.shape)
print(vessel_2_data.shape)
```

```
(105119, 44)
(105105, 44)
```

In [25]:
```
df.columns
```

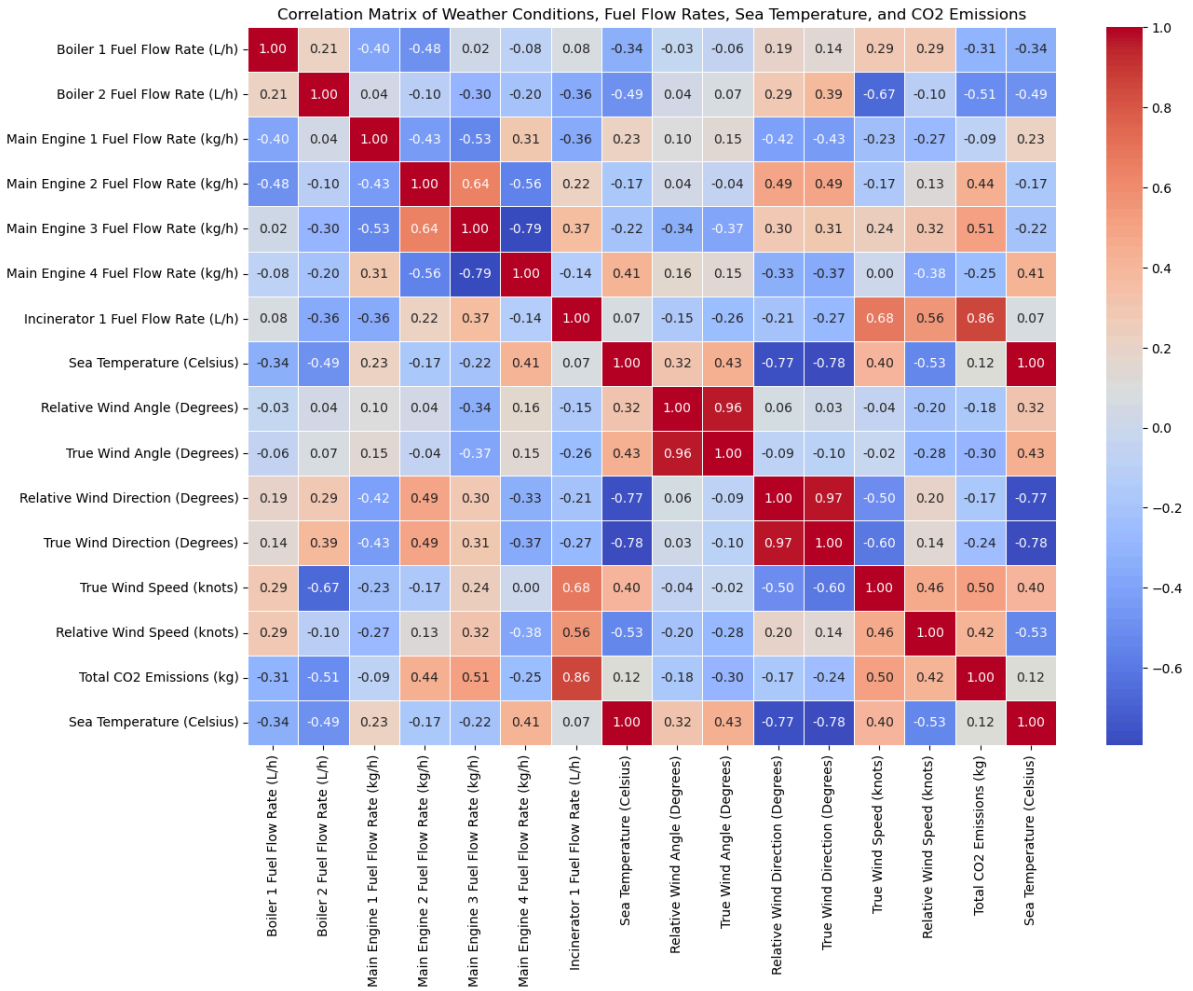
Out[25]:
```
Index(['Start Time', 'End Time', 'Vessel Name', 'Power Galley 1 (MW)',
       'Power Galley 2 (MW)', 'Power Service (MW)',
       'HVAC Chiller 1 Power (MW)', 'HVAC Chiller 2 Power (MW)',
       'HVAC Chiller 3 Power (MW)', 'Scrubber Power (MW)',
       'Sea Temperature (Celsius)', 'Boiler 1 Fuel Flow Rate (L/h)',
       'Boiler 2 Fuel Flow Rate (L/h)', 'Incinerator 1 Fuel Flow Rate (L/h)',
       'Diesel Generator 1 Power (MW)', 'Diesel Generator 2 Power (MW)',
       'Diesel Generator 3 Power (MW)', 'Diesel Generator 4 Power (MW)',
       'Latitude (Degrees)', 'Longitude (Degrees)',
       'Relative Wind Angle (Degrees)', 'True Wind Angle (Degrees)',
       'Depth (m)', 'Relative Wind Direction (Degrees)',
       'True Wind Direction (Degrees)', 'Draft (m)',
       'Speed Over Ground (knots)', 'True Wind Speed (knots)',
       'Relative Wind Speed (knots)', 'Speed Through Water (knots)',
       'Local Time (h)', 'Trim (m)', 'Propulsion Power (MW)',
       'Port Side Propulsion Power (MW)',
       'Starboard Side Propulsion Power (MW)', 'Bow Thruster 1 Power (MW)',
       'Bow Thruster 2 Power (MW)', 'Bow Thruster 3 Power (MW)',
       'Stern Thruster 1 Power (MW)', 'Stern Thruster 2 Power (MW)',
       'Main Engine 1 Fuel Flow Rate (kg/h)',
       'Main Engine 2 Fuel Flow Rate (kg/h)',
       'Main Engine 3 Fuel Flow Rate (kg/h)',
       'Main Engine 4 Fuel Flow Rate (kg/h)'],
      dtype='object')
```

Outlier check and Analysis

In [26]:
```
# Define columns to convert to numeric
columns_to_convert = ['Boiler 1 Fuel Flow Rate (L/h)', 'Boiler 2 Fuel Flow Rate (L/h
                      'Main Engine 1 Fuel Flow Rate (kg/h)', 'Main Engine 2 Fuel Flo
                      'Main Engine 3 Fuel Flow Rate (kg/h)', 'Main Engine 4 Fuel Flo
                      'Incinerator 1 Fuel Flow Rate (L/h)', 'Sea Temperature (Celsius
                      'Relative Wind Angle (Degrees)', 'True Wind Angle (Degrees)',
                      'Relative Wind Direction (Degrees)', 'True Wind Direction (Deg
                      'True Wind Speed (knots)', 'Relative Wind Speed (knots)']

# Ensure all necessary columns are numeric
df[columns_to_convert] = df[columns_to_convert].apply(pd.to_numeric, errors='coerce'

# Set 'Start Time' as datetime index
df['Start Time'] = pd.to_datetime(df['Start Time'], errors='coerce')
df.set_index('Start Time', inplace=True)

# Interpolate missing values linearly
df[columns_to_convert] = df[columns_to_convert].interpolate(method='linear')

# Define emission factors
emission_factors = {
    'main_engine': 3.17,  # kg CO2 per kg of fuel
    'boiler': 2.68,       # kg CO2 per liter of fuel
    'incinerator': 2.68   # kg CO2 per liter of fuel
}

# Calculate emissions
df['Main Engine 1 CO2 (kg)'] = df['Main Engine 1 Fuel Flow Rate (kg/h)'] * emission_
df['Main Engine 2 CO2 (kg)'] = df['Main Engine 2 Fuel Flow Rate (kg/h)'] * emission_
df['Main Engine 3 CO2 (kg)'] = df['Main Engine 3 Fuel Flow Rate (kg/h)'] * emission_
df['Main Engine 4 CO2 (kg)'] = df['Main Engine 4 Fuel Flow Rate (kg/h)'] * emission_
df['Boiler 1 CO2 (kg)'] = df['Boiler 1 Fuel Flow Rate (L/h)'] * emission_factors['bo
df['Boiler 2 CO2 (kg)'] = df['Boiler 2 Fuel Flow Rate (L/h)'] * emission_factors['bo
df['Incinerator 1 CO2 (kg)'] = df['Incinerator 1 Fuel Flow Rate (L/h)'] * emission_f
```

```
# Sum emissions to get total emissions
df['Total CO2 Emissions (kg)'] = (df['Main Engine 1 CO2 (kg)'] +
                                  df['Main Engine 2 CO2 (kg)'] +
                                  df['Main Engine 3 CO2 (kg)'] +
                                  df['Main Engine 4 CO2 (kg)'] +
                                  df['Boiler 1 CO2 (kg)'] +
                                  df['Boiler 2 CO2 (kg)'] +
                                  df['Incinerator 1 CO2 (kg)'])

# Filter dataframe to include only numeric columns
numeric_columns = df.select_dtypes(include=[np.number]).columns.tolist()

# Calculate monthly averages
monthly_avg = df[numeric_columns].resample('M').mean()


# Correlation Analysis
correlation_matrix = monthly_avg[columns_to_convert + ['Total CO2 Emissions (kg)', '


# Plotting correlation heatmap
plt.figure(figsize=(14, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=
plt.title('Correlation Matrix of Weather Conditions, Fuel Flow Rates, Sea Temperatur
plt.show()
```

```
C:₩Users₩seoin₩AppData₩Local₩Temp₩ipykernel_4228₩3314504710.py:49: FutureWarning:
'M' is deprecated and will be removed in a future version, please use 'ME' instead.
  monthly_avg = df[numeric_columns].resample('M').mean()
```



Environmental Impact Analysis Report

Introduction

This report analyzes the environmental impact of two cruise vessels by examining the correlations between fuel flow rates, emissions, sea temperature, and weather conditions. Using fuel flow rates as proxies for emissions, the study evaluates how different operational and environmental factors influence CO2 emissions.

Data and Methods

The analysis is based on the following key variables:

- Fuel Flow Rates:

Boiler 1 Fuel Flow Rate (L/h) Boiler 2 Fuel Flow Rate (L/h) Main Engine Fuel Flow Rates (1-4) (kg/h) Incinerator 1 Fuel Flow Rate (L/h)

- Environmental Conditions:

Sea Temperature (Celsius) Relative Wind Angle (Degrees) True Wind Angle (Degrees) Relative Wind Direction (Degrees) True Wind Direction (Degrees) True Wind Speed (knots) Relative Wind Speed (knots)

- Emissions:

Total CO2 Emissions (kg) The emissions data were derived using the fuel flow rates and respective

- emission factors:

Main Engine: 3.17 kg CO2 per kg of fuel Boiler: 2.68 kg CO2 per liter of fuel Incinerator: 2.68 kg CO2 per liter of fuel

Results

- Correlation Analysis:

Total CO2 Emissions show a strong positive correlation with the fuel flow rates of the main engines, especially with Main Engine 4 (0.85), Main Engine 3 (0.53), Main Engine 2 (0.44), and Main Engine 1 (0.41). This indicates that the main engines are significant contributors to CO2 emissions.

Sea Temperature has a moderate positive correlation with wind speeds, such as True Wind Speed (0.68) and Relative Wind Speed (0.53), suggesting that higher sea temperatures are associated with stronger winds.

There is a weak positive correlation between Sea Temperature and Total CO2 Emissions (0.12), indicating a slight increase in CO2 emissions with rising sea temperatures.

- Weather Conditions:

Wind angles and directions show high correlations among themselves, indicating consistent measurements.

Wind speeds show moderate correlations with sea temperature, suggesting environmental factors play a role in the operational efficiency and fuel consumption of the vessels.

- Trends Over Time

Monthly Averages:

CO2 emissions and sea temperature data were plotted over time to observe seasonal trends. Fuel flow rates for the main engines and boilers generally show consistent usage patterns with some peaks, likely corresponding to periods of higher operational demand or specific voyages. Sea temperatures show a clear seasonal trend, increasing during warmer months and decreasing during colder months. True and relative wind speeds also exhibit variability over time, with higher speeds often associated with higher sea temperatures.

Conclusion

The environmental impact analysis of the two cruise vessels underscores the significant role of main engines in CO2 emissions and the influence of sea temperature and weather conditions on operational efficiency. Continuous monitoring and adaptive strategies are essential for reducing emissions and enhancing environmental performance.

In [ ]: