

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
pd.set_option('display.max_columns', 50)
pd.set_option('display.max_rows', 50)
```

```
from utility import *
from trends import *
from plot import *
from kpi_computation import *
from forecast import *
```

1. The features in the dataset are continuous. They are as follows:
- Power consumed by different components
 - Factors influencing power consumption
 - Time series in an interval of 5 minutes for 2 vessels, spanning across a year. That makes it $1224365 = 105120 \times 12$ data points for each vessel.

```
# Read the data
df = pd.read_csv('data/data.csv', header = 0)
df.head()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	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)
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	27.3000	0.0000	0.0	19.0090
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	27.3000	47.7695	0.0	216.3180
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	27.3000	77.2034	0.0	439.4300
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	27.3076	60.6369	0.0	218.2797
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	27.3518	55.2184	0.0	0.0000

```
df.describe()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	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)
count	210224.000000	210224.000000	210222.000000	210033.000000	210033.000000	210033.000000	210224.000000	210224.000000	210224.000
mean	0.037829	0.118840	4.923284	0.182571	0.159086	0.154729	0.394463	18.304904	36.068139
std	0.164684	0.182357	1.014741	0.236498	0.221811	0.230015	0.298189	6.259071	65.686900
min	0.000000	0.000000	-0.040000	0.000000	0.000000	0.000000	0.000000	2.800000	0.000000
25%	0.007300	0.051600	4.239700	0.000000	0.000000	0.000000	0.155500	13.700000	0.000000
50%	0.025600	0.104300	4.916700	0.000000	0.000000	0.000000	0.321150	18.203900	0.000000
75%	0.055800	0.178500	5.526600	0.386400	0.348900	0.338900	0.763600	21.796525	58.300000
max	40.285400	40.305400	15.264000	0.826100	0.793000	6.305300	1.031500	31.611500	482.057000

```
# Check the data types and column names
df.dtypes
```

```
Start Time                object
End Time                  object
Vessel Name               object
Power Galley 1 (MW)       float64
Power Galley 2 (MW)       float64
Power Service (MW)        float64
HVAC Chiller 1 Power (MW) float64
HVAC Chiller 2 Power (MW) float64
HVAC Chiller 3 Power (MW) float64
Scrubber Power (MW)       float64
Sea Temperature (Celsius) float64
Boiler 1 Fuel Flow Rate (L/h) float64
Boiler 2 Fuel Flow Rate (L/h) float64
Incinerator 1 Fuel Flow Rate (L/h) float64
Diesel Generator 1 Power (MW) float64
Diesel Generator 2 Power (MW) float64
Diesel Generator 3 Power (MW) float64
Diesel Generator 4 Power (MW) float64
Latitude (Degrees)        float64
Longitude (Degrees)       float64
Relative Wind Angle (Degrees) float64
True Wind Angle (Degrees) float64
Depth (m)                 float64
Relative Wind Direction (Degrees) float64
True Wind Direction (Degrees) float64
Draft (m)                 float64
Speed Over Ground (knots) float64
True Wind Speed (knots)   float64
Relative Wind Speed (knots) float64
Speed Through Water (knots) float64
Local Time (h)            float64
Trim (m)                  float64
Propulsion Power (MW)     float64
Port Side Propulsion Power (MW) float64
Starboard Side Propulsion Power (MW) float64
Bow Thruster 1 Power (MW) float64
Bow Thruster 2 Power (MW) float64
Bow Thruster 3 Power (MW) float64
Stern Thruster 1 Power (MW) float64
Stern Thruster 2 Power (MW) float64
Main Engine 1 Fuel Flow Rate (kg/h) float64
Main Engine 2 Fuel Flow Rate (kg/h) float64
Main Engine 3 Fuel Flow Rate (kg/h) float64
Main Engine 4 Fuel Flow Rate (kg/h) float64
dtype: object
```

The features in the entire dataset are (without the Start time, End Time and the Vessel Name):

Column Name
Power Galley 1 (MW)

Column Name
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)

Vessel - Level Analysis

```
# dfv = pick_vessel(df, 'Vessel 1')
dfv = pick_vessel(df, 'Vessel 2')
# dfv = df
```

```
# Generates the distribution of the features in the dataset as histogram plots
feature_distribution(dfv)
```

Comments (Vessel 1):

1. Power Galleys: Mostly constant range of power consumption in operation. Less variation and wouldn't contribute much in forecast models
2. HVAC Chillers: A lot of variation, but extremely skewed in the sense that most of the operation range is fixed
3. Scrubber Power: Shows 4 peaks, denoting 4 levels of high intensity cleaning activities
4. Power Service: Service Power is more uniformly distributed but doesn't have much variation in the data
5. Sea Temperature: Primarily influenced by the season of the year, and equatorial position of the ship, is uniformly distributed except for a peak on the right side indicating slight skewness probably due to stronger summer season in one of the locations
6. Boiler Fuel Flow: Consists of a tall box and a tiny peak at the higher end of fuel flow rate. Most of the usual operations happen in between 0-10L/h fuel flow. Rarely, on days of extreme weather or rough operations
7. Incinerator: Majority of the waste removal operation happens with a lower fuel requirement, except for some extreme cases. These could be during docking for extra cleaning/occasional discharge cleaning.
8. Diesel Generators: Substantiate a 2-3 peaks highlighting a 2-3 modes of operation, including traveling against currents/winds requiring higher throughput
9. Latitude and Longitude: Positions indicate travel in the northern hemisphere, around Atlantic, with extra time spent close to the prime meridian
10. Relative Wind angle: Indicates that the cruise ship tries to move and orient itself in-line to the wind 0deg or 360deg for conserving energy by taking the momentum from wind and also by reducing drag
11. Depth: The cruise tries to maintain consistent depth throughout between 0-100 metres
12. Draft: maintained constantly between 7.5 and 8 metres, shows buoyancy control and build quality of the ship
13. Relative Wind Speed: The wind speed is aligned in favor to the ship movement by aligning the direction of sail
14. Speed through water/over land: Indicates 2 modes, one low range where the ship is either docking, anchoring or sailing in shallow waters. The second peak which is not so frequent indicates high speed motion. Among these, Speed through water has higher peak on greater end of speed spectrum because the ship has to move against the water many a times.
15. Trim: A uniform and normally distributed trim is maintained indicating well-planned design execution throughout the operation
16. Propulsion power has 4 peaks, indicating 4 ranges of operation. It is well spread out with variation, higher power being demanded probably when the ship is leaving from docks and is on shallow waters.
17. Thrusters: Limited operating ranges with 0-0.1 MW consumed in maneuvering primarily
18. Main Engine Fuel Flow: Is optimized to operate mostly at lower end without many fluctuations. This is indicated by a left skewed histogram, with only rare activities on the higher end

Comments (Vessel2): In terms of feature distribution, vessel 2 replicates the same distribution for almost all the features, except latitude and longitude positions. Which indicates a different route. Beyond that, to look at specific differences between the features of the two vessels: a boxplot would be ideal!

```
feature_boxplot(df) # Boxplot of the features for comparison between vessels
```

As we can see, there are a few differences in operational parameters for the two vessels. Majorly, the median propulsion power indicated by a line in the body of the candle is lower for Vessel 2. Meanwhile, there are many outliers in terms of Main engine fuel flow for vessel 2 and a larger variation within the body of the candle can be seen (1st quartile and 3rd quartile). The values are spread over a higher range of fuel flow. With a lower median of propulsion power, this directly implies that the Power Specific Fuel Consumption KPI, or the fuel required to produce a unit of power (efficiency) is much lower for vessel 2.

```
missing_values = dfv.isna().sum()
missing_values.plot(kind='bar', figsize=(12,5), title='Missing Values') # Plot the missing values
```

Impute the missing Values:

Comments Vessel 1:

1. For features that have <1% missing values, impute by interpolating them using closest non-missing values as the features are all usually smooth within the 5mins time intervals in which they are recorded
2. For features that have >20% missing values, impute by using median of the column/feature. The idea of using median is based on the skewed constant nature of depth maintained by the ship for most of the periods

Comments Vessel 2:

1. For features that have <1% missing values, impute by interpolating them using closest non-missing values as the features are all usually smooth within the 5mins time intervals in which they are recorded
2. For features that have >20% missing values, impute by using median of the column/feature. There are about 5-6 rows where all the values are empty except vessel name and timestamps. These rows can alternatively be removed from analysis. This is mostly a reporting error than a machine lapse, because a machine lapse and a sudden fluctuation between operational state and non-operational states is not possible within a 10-15 min time window. Also, in a later stage while resampling, they will have to be interpolated anyway. For the 'Depth' feature, a complete imputation with median/mean is necessary

```
# Imputing the missing values
missing_values
```

```
col_to_interpolate = dfv.columns.difference(['Depth (m)', 'Start Time', 'End Time', 'Vessel Name']) # Columns to interpolate
```

```
impute_time_series(dfv, col_to_interpolate) # Imputation via interpolation for the columns with < 1% missing values as they are likely to be continuous in time
dfv.isna().sum()
```

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)	0
Longitude (Degrees)	0
Relative Wind Angle (Degrees)	0
True Wind Angle (Degrees)	0
Depth (m)	29738
Relative Wind Direction (Degrees)	0
True Wind Direction (Degrees)	0
Draft (m)	0
Speed Over Ground (knots)	0
True Wind Speed (knots)	0
Relative Wind Speed (knots)	0
Speed Through Water (knots)	0
Local Time (h)	0
Trim (m)	0
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	

```
median_depth = dfv['Depth (m)'].median() # imputation via median value as more than 20% of the values are missing
dfv['Depth (m)'].fillna(median_depth, inplace=True)
dfv.isna().sum()
```

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)                 0
Longitude (Degrees)                 0
Relative Wind Angle (Degrees)      0
True Wind Angle (Degrees)           0
Depth (m)                          0
Relative Wind Direction (Degrees)  0
True Wind Direction (Degrees)      0
Draft (m)                          0
Speed Over Ground (knots)           0
True Wind Speed (knots)             0
Relative Wind Speed (knots)         0
Speed Through Water (knots)         0
Local Time (h)                     0
Trim (m)                           0
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
```

```
dfv.head()
```

Multi-Collinearity Check

```
# Correlation matrix
dfv_sub = dfv.iloc[:,3:]
corr = dfv_sub.corr()
```

```
high_corr = []
for i in corr.columns:
    high_corr.append(corr[(corr[i] > 0.8) | (corr[i] < -0.8)][i])
print(f'The highly correlated columns are:{high_corr}')
```

```
# Plot the correlation matrix
fig, ax = plt.subplots(figsize=(24,20))
sns.heatmap(data = corr[(corr > 0.8) | (corr < -0.8)], vmin=-1,vmax=1, cmap='coolwarm', ax = ax, annot= True)
ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
ax.set_title('Correlation Matrix')
plt.show()
```

Comments (Vessel 1):

- 1. Diesel Generators' Power are highly correlated to corresponding Main Engine's Fuel flow rate
- 2. Propulsion power is a linear combination of Port Side Propulsion Power and Starboard Side Propulsion Power
- 3. Speed through water and Speed over ground are positively correlated to the Propulsion Power
- 4. Sea water temperature reducing as the latitude is increasing completely makes sense. This is because, temperatures are lower as one moves towards the poles

Comments (Vessel 2):

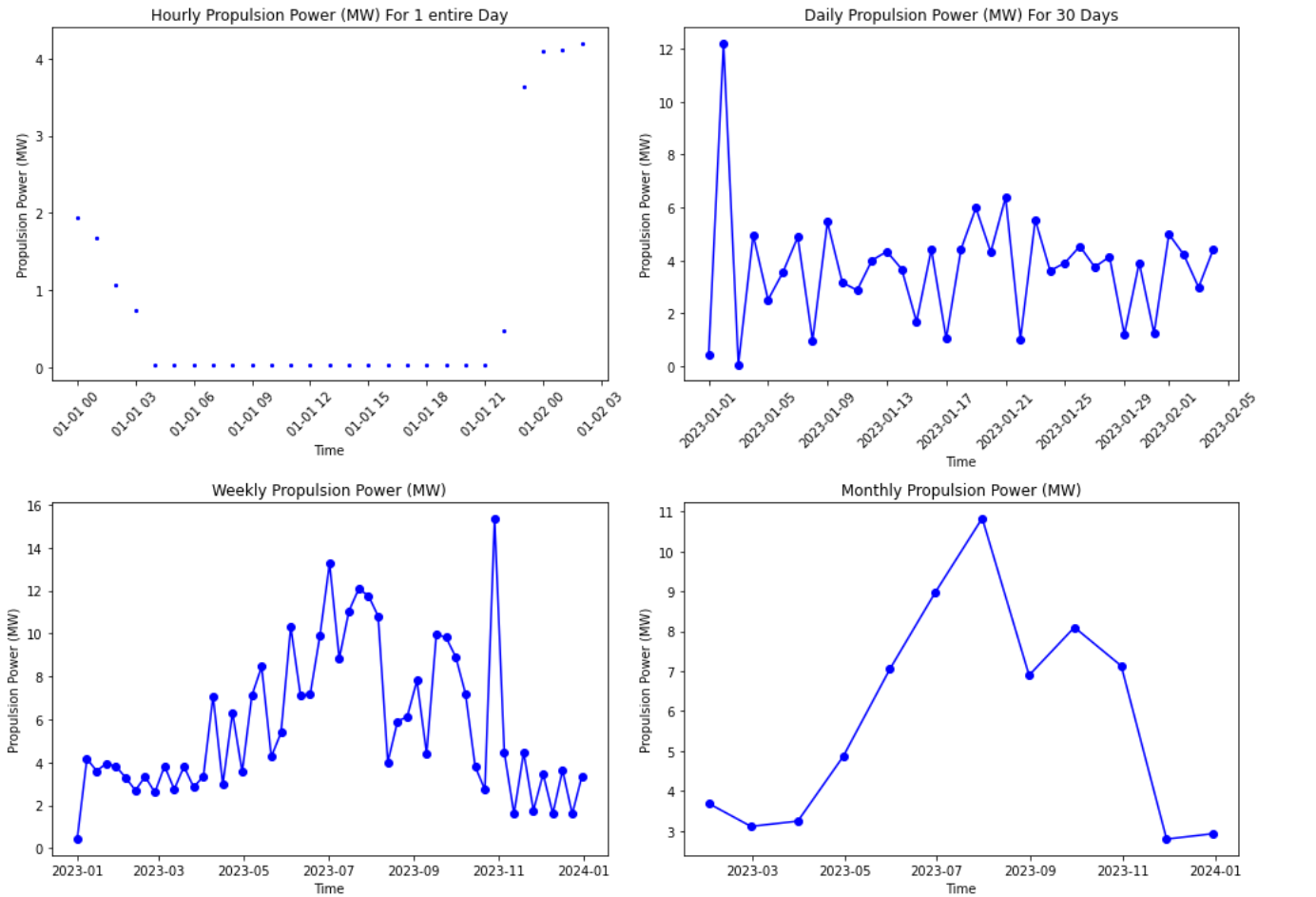
- 1. Diesel Generators' Power are highly correlated to corresponding Main Engine's Fuel flow rate (=1)
- 2. Propulsion power is a linear combination of Port Side Propulsion Power and Starboard Side Propulsion Power
- 3. Speed through water and Speed over ground are positively correlated to the Propulsion Power (=0.91)
- 4. Sea water temperature reducing as the latitude is increasing completely makes sense. This is because, temperatures are lower as one moves towards the poles. Longitude is also negatively correlated to sea temperatures

Trend and seasonality analysis

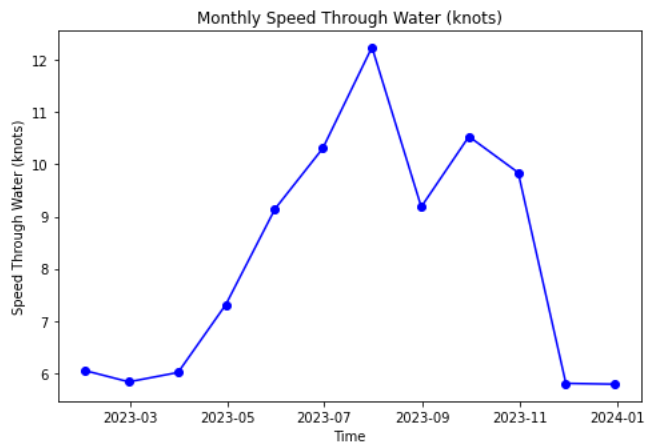
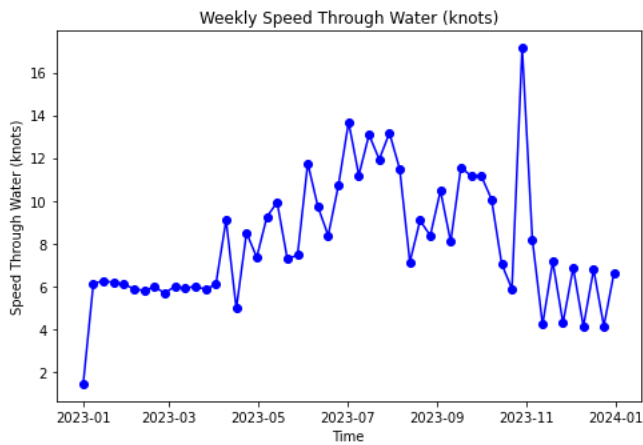
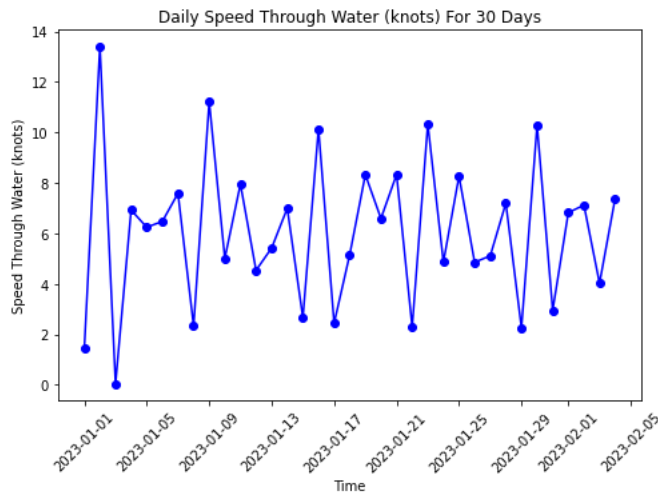
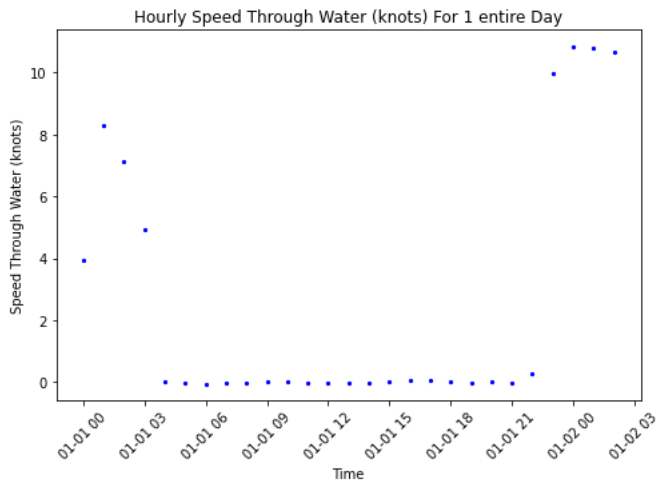
The outliers and skewness in the data are kept for analysis as this is recorded first hand. For forecasting the data will be standardized but now the analysis is done as is

```
# Resampling the data for hourly, daily, weekly and seasonal trends
hourly_df, daily_df, weekly_df, monthly_df = resample(dfv)
```

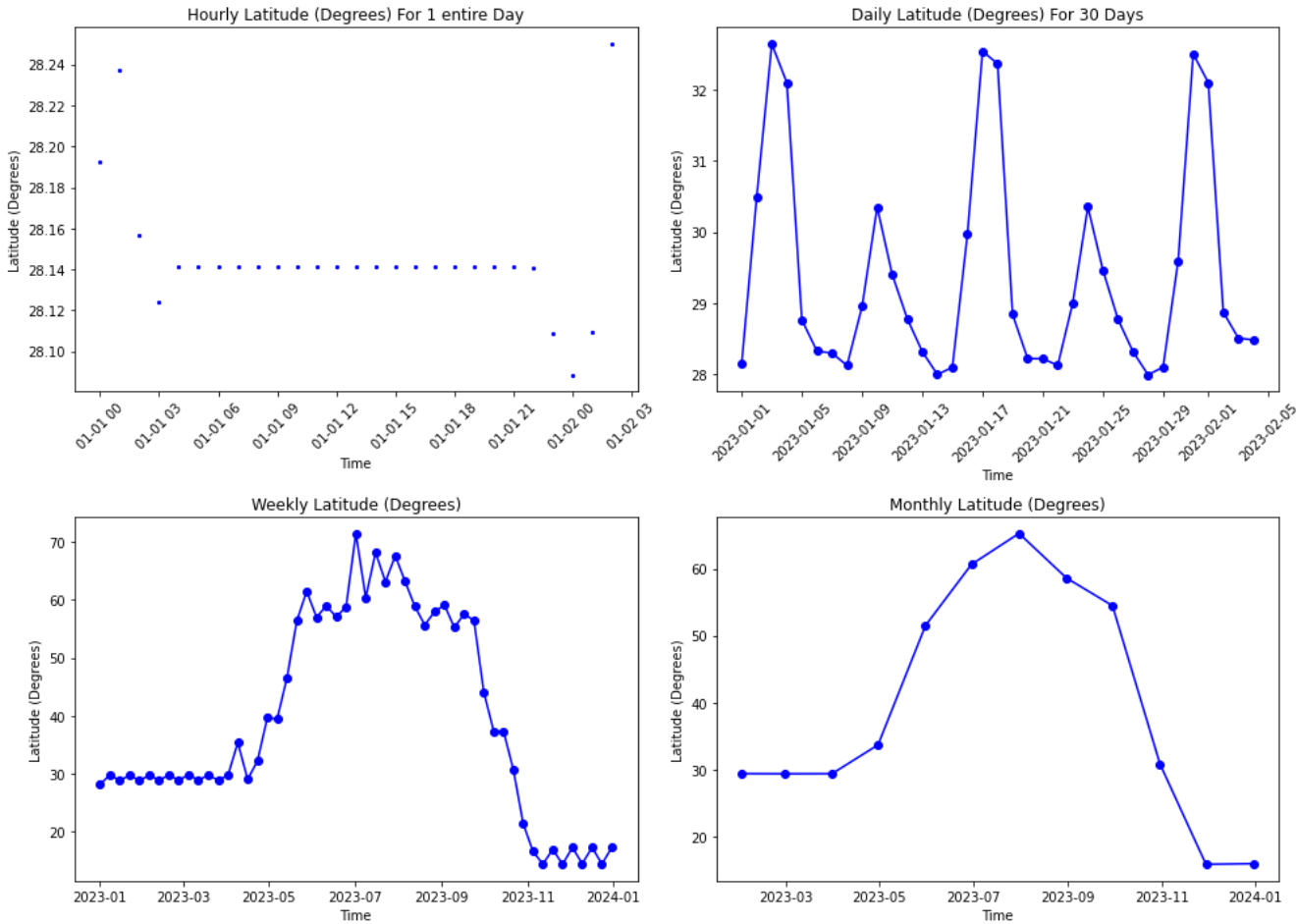
```
# Pick a feature from the list above to visualize the trend over
trend_plot(hourly_df, daily_df, weekly_df, monthly_df, 'Propulsion Power (MW)')
```



```
# Pick a feature from the list above to visualize the trend over
trend_plot(hourly_df, daily_df, weekly_df, monthly_df, 'Speed Through Water (knots)')
```



```
# Pick a feature from the list above to visualize the trend over
trend_plot(hourly_df, daily_df, weekly_df, monthly_df, 'Latitude (Degrees)')
```

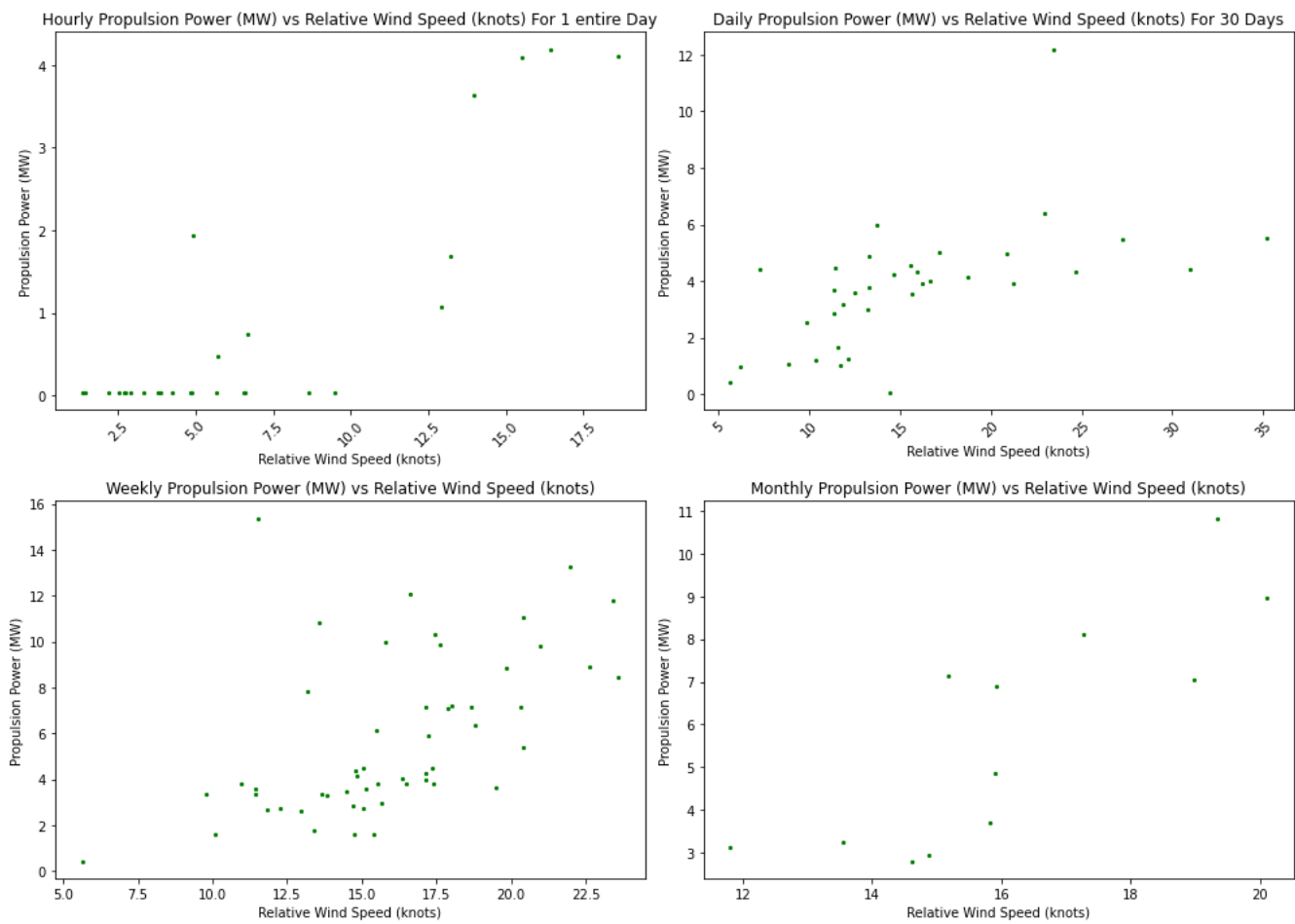
Comments (Vessel 1):

- 'Propulsion Power (MW)': Extreme Wind speeds demand more propulsion power, indicating the ship having to re-direct itself or move against the wind
- 'Latitude': No visible relationship between Latitude and Relative wind Speeds

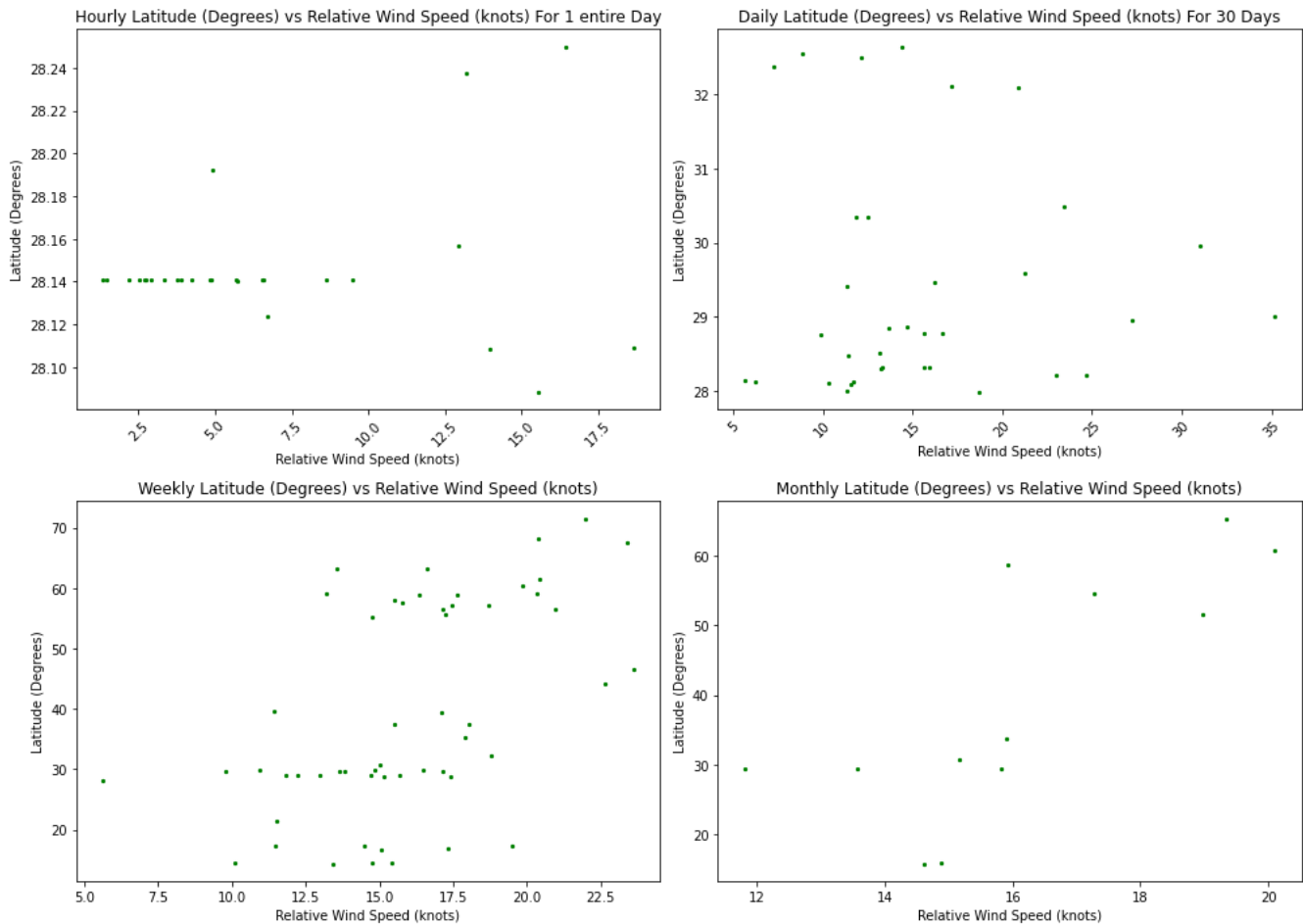
Comments (Vessel 2):

- 'Propulsion Power (MW)': In later part of the day, as seen in hourly plots there is a higher demand for propulsion power. This completely makes sense as the speed through water at nights are also the highest in a day
- 'Latitude': Gives an idea of the global position of the vessel. There is a 6-8 day cyclic pattern

```
# Pick 2 features from the above list to visualize their relationship with each other over time
pair_plot(hourly_df, daily_df, weekly_df, monthly_df, 'Propulsion Power (MW)', 'Relative Wind Speed (knots)')
```



```
# Pick 2 features from the above list to visualize their relationship with each other over time
pair_plot(hourly_df, daily_df, weekly_df, monthly_df, 'Latitude (Degrees)', 'Relative Wind Speed (knots)')
```



Comments (Vessel 1):

- 'Propulsion Power (MW)' vs 'Relative Wind Speed (knots)': Extreme Wind speeds demand more propulsion power, indicating the ship having to re-direct itself or move against the wind
- 'Latitude' vs 'Relative Wind Speed (knots)': No visible relationship between Latitude and Relative wind Speeds

Comments (Vessel 2):

- 'Propulsion Power (MW)' vs 'Relative Wind Speed (knots)': Extreme Wind speeds demand more propulsion power, which is clearly indicated in the plots as the ship having to re-direct itself or move against the wind
- 'Latitude' vs 'Relative Wind Speed (knots)': No visible relationship between Latitude and Relative wind Speeds

KPI Generation and Analysis

3 KPIs are generated based on domain knowledge and research (check the Energy Flow Diagram I built on the README):

1. Fuel Consumption per nautical mile (A measure of fuel efficiency)
2. Total Power Consumed
3. Power Specific Fuel Consumption

Method

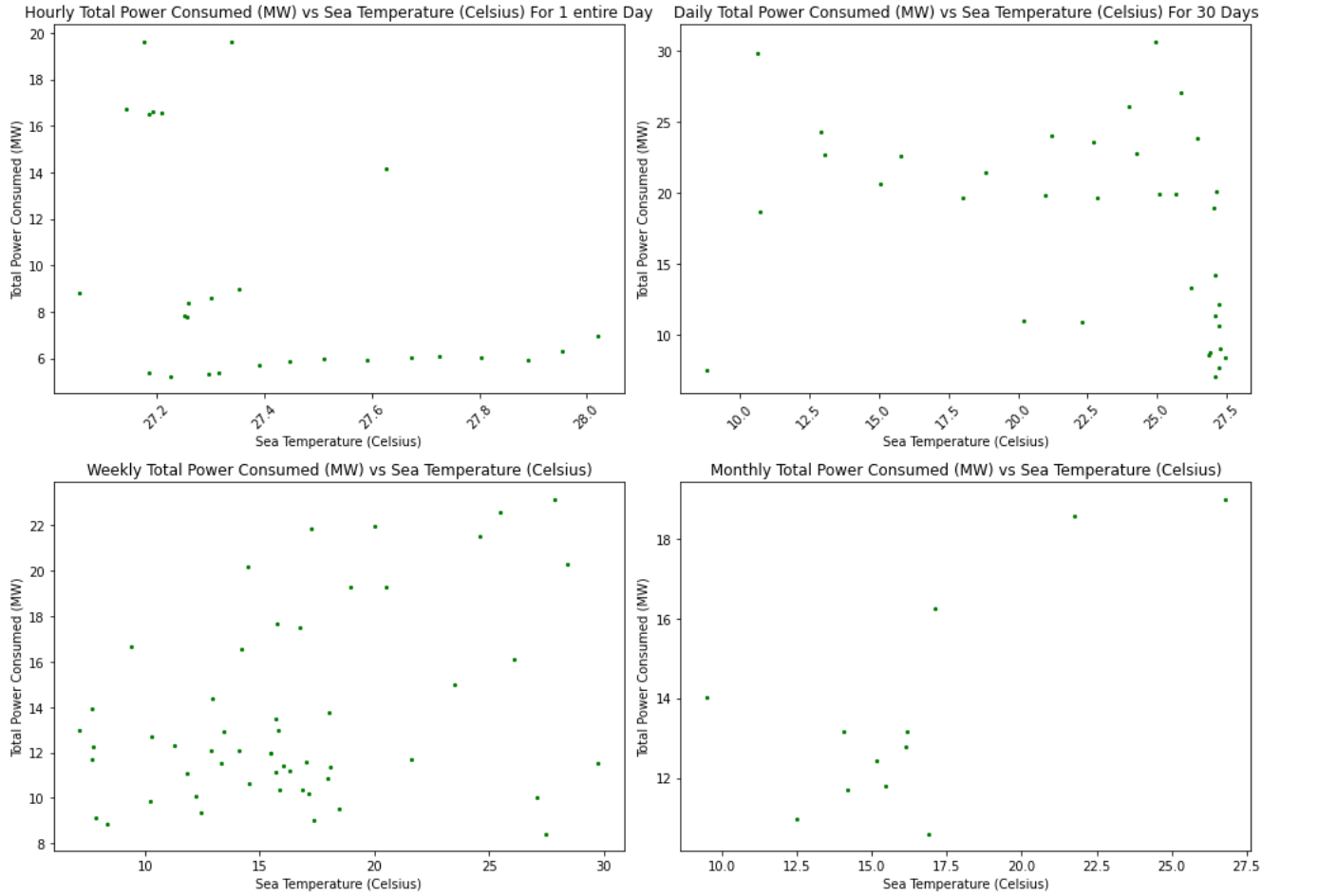
- Fuel Consumption per nautical mile = Sum of Engine Fuel Flow / Speed through water. Speed through water is used as the ship has to put in work to actually move relative to water. The KPI is capped at a maximum of 5000, because it goes to infinity when speed through is very close to 0. This happens while docking and anchoring
- Total Power Consumed = Sum power consumed by Power Galleys, HVAC Chillers, Power Service, Scrubber, Propulsion and Thrusters
- Power Specific Fuel Consumption = Fuel flow in engines, boilers and incinerator/ Total power consumed. Similar to the very commonly used fuel efficiency or brake specific fuel consumption (BSFC). It is used to assess the efficiency of any engine that burns fuel and produces rotational power, typically internal combustion engines. Assuming Fuel in Boiler and Incinerator to be Diesel and the density to be 0.85 kg/litre, for mass flow calculation.

```
# Compute the KPIs at hourly level
h_kpi = compute_kpis(hourly_df)
```

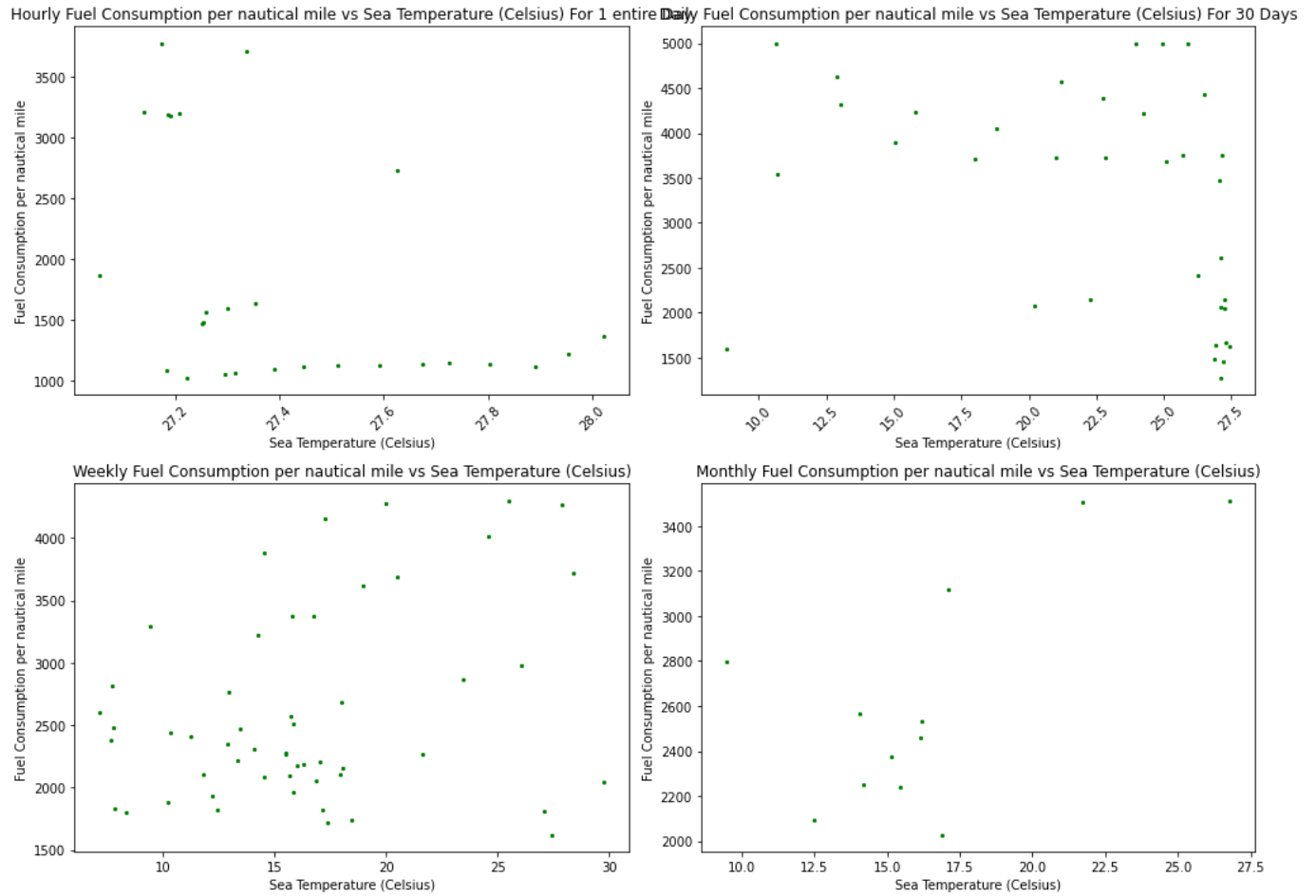
```
h_kpi['Fuel Consumption per nautical mile'].describe()
```

```
# Compute KPIS for daily, weekly and monthly granular data
d_kpi = compute_kpis(daily_df)
w_kpi = compute_kpis(weekly_df)
m_kpi = compute_kpis(monthly_df)
```

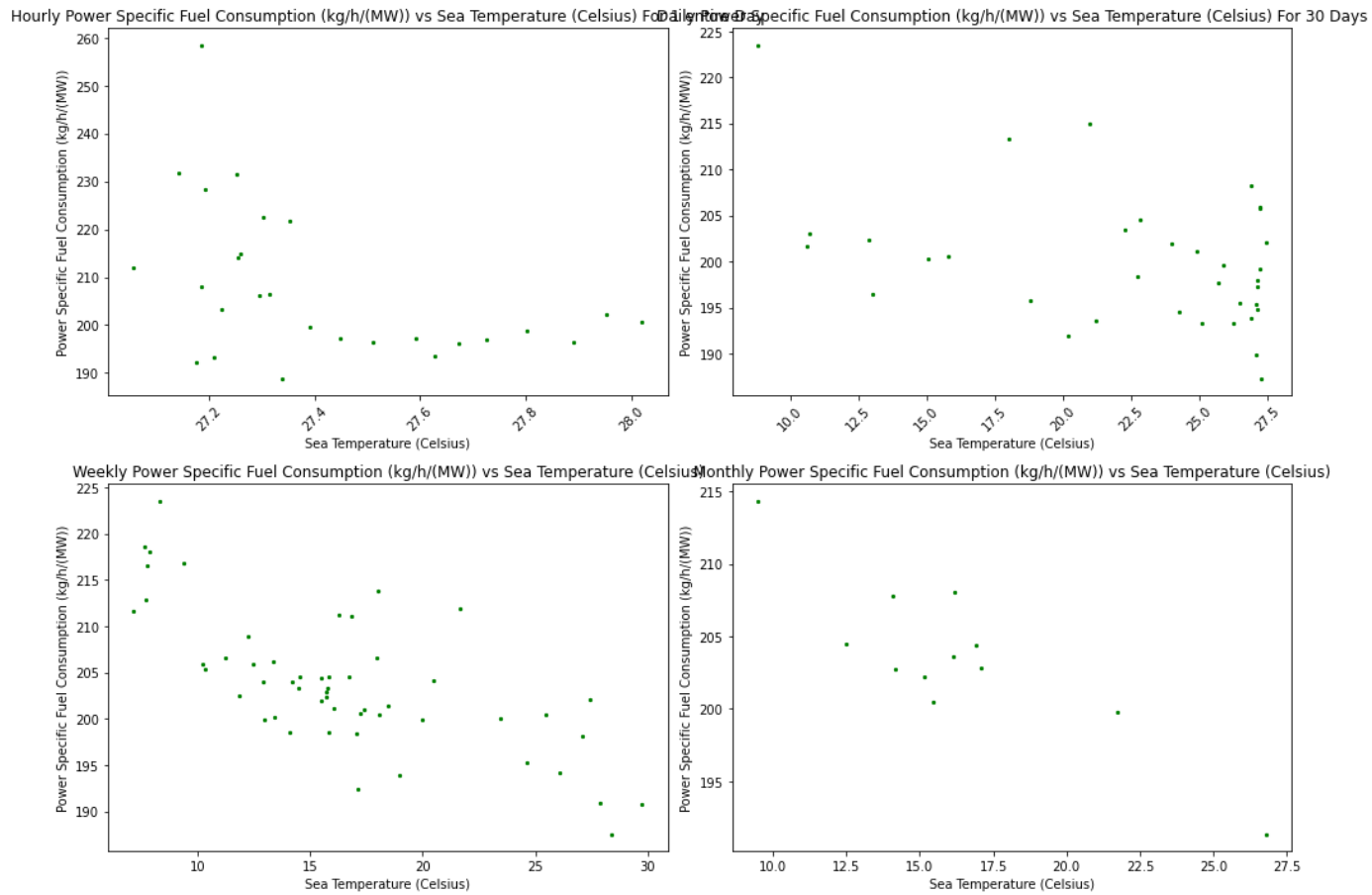
```
pair_plot(h_kpi, d_kpi, w_kpi, m_kpi, 'Total Power Consumed (MW)', 'Sea Temperature (Celsius)')
```



```
pair_plot(h_kpi, d_kpi, w_kpi, m_kpi, 'Fuel Consumption per nautical mile', 'Sea Temperature (Celsius)')
```



```
pair_plot(h_kpi, d_kpi, w_kpi, m_kpi, 'Power Specific Fuel Consumption (kg/h/(MW))', 'Sea Temperature (Celsius)')
```



1. KPIs vs Sea Temperature

- On the long-term front, with seasonality in play: we can observe from the monthly chart that for warmer conditions of water, more power is consumed.
- This totally makes sense as the water would be less dense and the ship would have to work generate thrust to move through it
- The effect is not fully observed in the Fuel Efficiency chart as lower density also means lower resistance to move through water.

2. KPIs vs Speed over Ground

- Total Power Required increases with Speed over Ground
- Fuel Consumption per nautical mile has two clusters where there is not much increase in Fuel required per nautical mile (between 6&8, 12&17)
- Coming to Power Specific Fuel Consumption, the optimal speeds again lie between 12 and 17 where the Fuel required to produce a certain power is flat and considerably low

Based on the scatter plots, the optimal speed of operation are between 7.5-8.5 and 15-17.5, where there is a horizontal spread of points: the fuel required to operate in these range of speeds seem to be the same and not fluctuate a lot

Comments(Vessel 2):

1. KPIs vs Sea Temperature

- On the long term monthly scale, with seasonality in play: we can observe from the monthly chart that for warmer conditions of water, more power is consumed.
- This totally makes sense as the water would be less dense and the ship would have to work generate thrust to move through it
- The effect is not fully observed in the Fuel Efficiency chart as lower density also means lower resistance to move through water.

2. KPIs vs Speed over Ground

- On a daily and weekly scale, total Power Required increases with Speed over Ground and vice versa
- Fuel Consumption per nautical mile also increases consistently with speed on daily, weekly and monthly scales (between 6&8, 12&17)
- Coming to Power Specific Fuel Consumption, no starkly visible pattern can be seen. But from daily and weekly plots, two possible clusters where fuel consumption is horizontal can be seen. This could account for an optimal speed of performance.

Time-Series Forecasts

1. I first remove the columns not necessary for analysis like 'End Time' and 'Vessel Name'
2. Further, from the above analysis it is also determined that some columns are very highly correlated. If two columns are correlated at > 0.9 , only one is kept for the analysis
3. If the distribution of a column is very narrow, it doesn't offer any variation as a feature. Hence we remove that also from the analysis
4. None of the features used to generate the KPI can be used for forecasting as it will be linearly dependant by inherent nature
5. Therefore, in the SARIMAX Seasonal modeling and forecast, I take an approach to try and forecast the total power from external factors that can be pre-planned before the cruise starts based on weather forecasts and other requirements
6. Finally, this is concluded with some simple hyperparameter tuning for better AIC value (Penalizing method for addition of a new feature to the model) to get the best performing forecast

Hourly will be too granular and computationally expensive. Monthly analysis would be sparse and not so informative since we only have 1 year data

External factors that can be pre-planned before the cruise starts

['Sea Temperature (Celsius)', '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)', 'Local Time (h)', 'Trim (m)', 'Total Power Consumed (MW)']

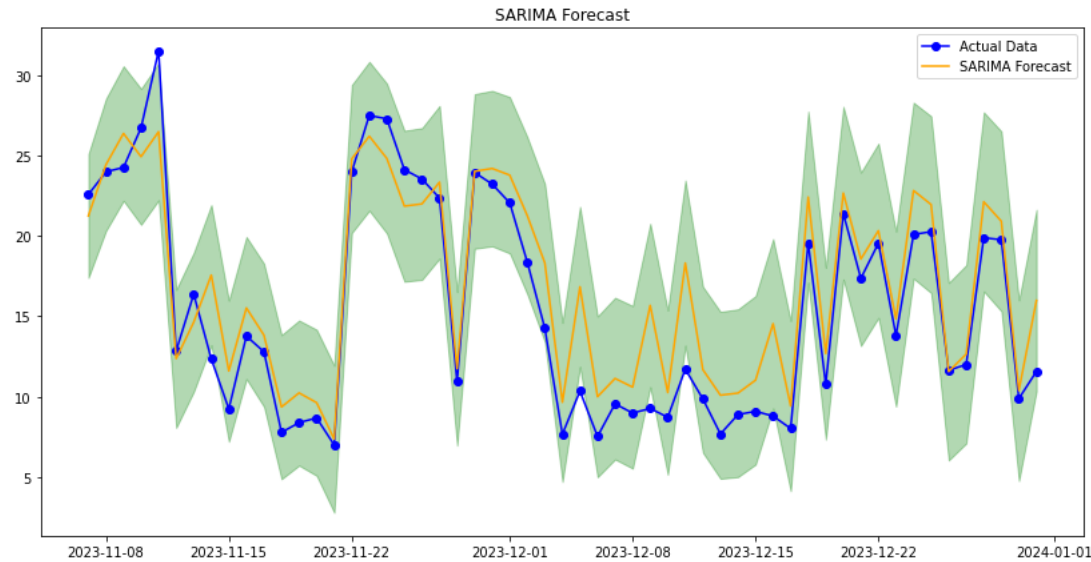
```
# Hourly forecast, the grid search for hyperparameters takes a long time to run, processor limitations
h_forecast = h_kpi[['Sea Temperature (Celsius)', 'Latitude (Degrees)', 'Longitude (Degrees)', 'Relative Wind Angle (Degrees)',
'True Wind Angle (Degrees)', 'Depth (m)', 'Relative Wind Direction (Degrees)', 'True Wind Direction (Degrees)', 'Draft (m)',
'Speed Over Ground (knots)', 'True Wind Speed (knots)', 'Relative Wind Speed (knots)', 'Local Time (h)', 'Trim (m)',
'Total Power Consumed (MW)']]
```

```
# Daily forecast
d_forecast = d_kpi[['Sea Temperature (Celsius)', '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)', 'Local Time (h)', 'Trim (m)',
'Total Power Consumed (MW)']]
```

```
sarimax_forecast(d_forecast, 'Total Power Consumed (MW)', 7) #7 day forecast
```

```
SARIMA(0, 0, 0)x(0, 0, 0, 7) - AIC:2524.534106193414
SARIMA(0, 0, 0)x(0, 0, 1, 7) - AIC:2203.8287094025195
SARIMA(0, 0, 0)x(0, 1, 0, 7) - AIC:1495.6689137248247
SARIMA(0, 0, 0)x(0, 1, 1, 7) - AIC:1381.6930851306543
SARIMA(0, 0, 0)x(1, 0, 0, 7) - AIC:1554.3164392783694
```

SARIMA(0, 0, 0)x(1, 0, 1, 7) - AIC:1440.4675582552318
SARIMA(0, 0, 0)x(1, 1, 0, 7) - AIC:1440.8541347430978
SARIMA(0, 0, 0)x(1, 1, 1, 7) - AIC:1381.5532990178663
SARIMA(0, 0, 1)x(0, 0, 0, 7) - AIC:2163.522797180879
SARIMA(0, 0, 1)x(0, 0, 1, 7) - AIC:1953.7048368439919
SARIMA(0, 0, 1)x(0, 1, 0, 7) - AIC:1480.748203983198
SARIMA(0, 0, 1)x(0, 1, 1, 7) - AIC:1356.532035470119
SARIMA(0, 0, 1)x(1, 0, 0, 7) - AIC:1533.6655203857724
SARIMA(0, 0, 1)x(1, 0, 1, 7) - AIC:1410.7956904684083
SARIMA(0, 0, 1)x(1, 1, 0, 7) - AIC:1423.6449679588536
SARIMA(0, 0, 1)x(1, 1, 1, 7) - AIC:1357.0080661219745
SARIMA(0, 1, 0)x(0, 0, 0, 7) - AIC:1417.1614173512103
SARIMA(0, 1, 0)x(0, 0, 1, 7) - AIC:1419.0744101440982
SARIMA(0, 1, 0)x(0, 1, 0, 7) - AIC:1578.1572093749292
SARIMA(0, 1, 0)x(0, 1, 1, 7) - AIC:1413.5585797519866
SARIMA(0, 1, 0)x(1, 0, 0, 7) - AIC:1419.0944385993396
SARIMA(0, 1, 0)x(1, 0, 1, 7) - AIC:1420.3892379002527
SARIMA(0, 1, 0)x(1, 1, 0, 7) - AIC:1518.9672706609258
SARIMA(0, 1, 0)x(1, 1, 1, 7) - AIC:1415.5420908756155
SARIMA(0, 1, 1)x(0, 0, 0, 7) - AIC:1337.99305600544
SARIMA(0, 1, 1)x(0, 0, 1, 7) - AIC:1339.6949213284652
SARIMA(0, 1, 1)x(0, 1, 0, 7) - AIC:1498.5847534599445
SARIMA(0, 1, 1)x(0, 1, 1, 7) - AIC:1339.4247999572608
SARIMA(0, 1, 1)x(1, 0, 0, 7) - AIC:1339.7699368738986
SARIMA(0, 1, 1)x(1, 0, 1, 7) - AIC:1339.2871913507195
SARIMA(0, 1, 1)x(1, 1, 0, 7) - AIC:1441.8920102570764
SARIMA(0, 1, 1)x(1, 1, 1, 7) - AIC:1341.2628219283417
SARIMA(1, 0, 0)x(0, 0, 0, 7) - AIC:1425.1160604183726
SARIMA(1, 0, 0)x(0, 0, 1, 7) - AIC:1426.9053862956748
SARIMA(1, 0, 0)x(0, 1, 0, 7) - AIC:1475.6631209066668
SARIMA(1, 0, 0)x(0, 1, 1, 7) - AIC:1343.1015926486082
SARIMA(1, 0, 0)x(1, 0, 0, 7) - AIC:1426.9623980342653
SARIMA(1, 0, 0)x(1, 0, 1, 7) - AIC:1394.4765352557101
SARIMA(1, 0, 0)x(1, 1, 0, 7) - AIC:1419.0929052089123
SARIMA(1, 0, 0)x(1, 1, 1, 7) - AIC:1343.9278744956437
SARIMA(1, 0, 1)x(0, 0, 0, 7) - AIC:1348.350567114666
SARIMA(1, 0, 1)x(0, 0, 1, 7) - AIC:1350.0651408508406
SARIMA(1, 0, 1)x(0, 1, 0, 7) - AIC:1476.5414219993563
SARIMA(1, 0, 1)x(0, 1, 1, 7) - AIC:1336.229878608054
SARIMA(1, 0, 1)x(1, 0, 0, 7) - AIC:1350.2525339597723
SARIMA(1, 0, 1)x(1, 0, 1, 7) - AIC:1352.3905059556037
SARIMA(1, 0, 1)x(1, 1, 0, 7) - AIC:1420.7313919852566
SARIMA(1, 0, 1)x(1, 1, 1, 7) - AIC:1337.8789967756884
SARIMA(1, 1, 0)x(0, 0, 0, 7) - AIC:1378.174872938987
SARIMA(1, 1, 0)x(0, 0, 1, 7) - AIC:1380.1733105012263
SARIMA(1, 1, 0)x(0, 1, 0, 7) - AIC:1541.5180719156613
SARIMA(1, 1, 0)x(0, 1, 1, 7) - AIC:1377.5584622385186
SARIMA(1, 1, 0)x(1, 0, 0, 7) - AIC:1380.173780859042
SARIMA(1, 1, 0)x(1, 0, 1, 7) - AIC:1379.5828648327788
SARIMA(1, 1, 0)x(1, 1, 0, 7) - AIC:1488.0021531164077
SARIMA(1, 1, 0)x(1, 1, 1, 7) - AIC:1379.5694659850183
SARIMA(1, 1, 1)x(0, 0, 0, 7) - AIC:1330.37243216749
SARIMA(1, 1, 1)x(0, 0, 1, 7) - AIC:1332.3376572590037
SARIMA(1, 1, 1)x(0, 1, 0, 7) - AIC:1478.7393210887928
SARIMA(1, 1, 1)x(0, 1, 1, 7) - AIC:1331.562339377681
SARIMA(1, 1, 1)x(1, 0, 0, 7) - AIC:1332.3450261667115
SARIMA(1, 1, 1)x(1, 0, 1, 7) - AIC:1333.5589045789138
SARIMA(1, 1, 1)x(1, 1, 0, 7) - AIC:1422.5167709527489
SARIMA(1, 1, 1)x(1, 1, 1, 7) - AIC:1333.5583192797858
Best SARIMA(1, 1, 1)x(0, 0, 0, 7) - AIC:1330.37243216749
RMSE: 2.616105733879915
R2 Score: 0.848555942512195



Comments Vessel 1: Power- Daily

- For the dataset on a daily level granularity, Total power consumption for the cruise ship can be forecasted within acceptable margin of errors. The 95% confidence range of forecast is also predicted. The true range almost always lies in the 95% confidence range of forecast.
- This is by using external factors only, without taking into account any operational component. Therefore it states that the external conditions can forecast the energy requirements for the ship, this can aid in early resource planning of cruises
- Best SARIMA(1, 1, 1)x(0, 0, 0, 7) - AIC:1330.37243216749
- RMSE: 2.616105733879915 (MW)
- R2 Score: 0.848555942512195

Comments Vessel 2: Power- Daily

- For the dataset on a daily level granularity, Total power consumption for the cruise ship can be forecasted better than for vessel 1 in terms of AIC and R2 values. The 95% confidence range of forecast is also predicted. The true range almost always lies in the 95% confidence range of forecast.
- This is by using external factors only, without taking into account any operational component. Therefore it states that the external conditions can forecast the energy requirements for the ship, this can aid in early resource planning of cruises
- Best SARIMA(1, 0, 1)x(0, 1, 1, 7) - AIC:1268.9562687277244
- RMSE: 1.5000476299191863 (MW)
- R2 Score: 0.8428713947731085

```
d_forecast_fuel = d_kpi[['Fuel Consumption per nautical mile', 'Total Power Consumed (MW)']]
```

```
d_forecast_fuel.head()
```

```
sarimax_forecast(d_forecast_fuel, 'Fuel Consumption per nautical mile', 7)
```

Comments for Vessel 1: Fuel Consumption per nautical mile

- For the dataset on a daily level granularity, Fuel Consumption per nautical mile for the cruise ship can be forecasted just using the Total power consumption data within acceptable range of errors. The 95% confidence range of forecast is also predicted. The true range almost always lies in the 95% confidence range of forecast. The 95% range in this case is also narrow, that shows more certainty and less variance in predictions.
- This means, if an accurate enough forecast can be made for the power requirements, the forecast can in turn can be used for fuel requirements of the cruise ship. Therefore it states that the external conditions can forecast the fuel planning requirements for the ship, this can aid in early fuel planning of cruises, with potential for entering trade positions.
- Best SARIMA(1, 1, 1)x(0, 1, 1, 7) - AIC:3924.477617510145
- RMSE: 161.68641766048893 (kg/h/knots)

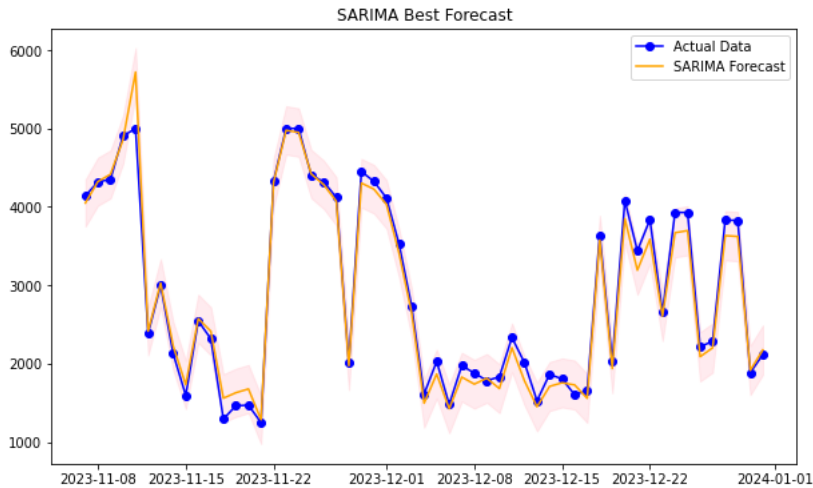
Comments for Vessel 2: Fuel Consumption per nautical mile

- For the dataset on a daily level granularity, Fuel Consumption per nautical mile for the cruise ship 2 can be forecasted just using the Total power consumption data within acceptable range of errors. The 95% confidence range of forecast is also predicted. The true range almost always lies in the 95% confidence range of forecast. The 95% range in this case is also narrow, that shows more certainty and less variance in predictions.
- In this case the forecast is better than for Vessel 1 in terms of the AIC and R2 values, the forecast can in turn can be used for fuel requirements of the cruise ship. Therefore it states that the external conditions can forecast the fuel planning requirements for the ship, this can aid in early fuel planning of cruises, with potential for entering trade positions.
- Best SARIMA(1, 1, 1)x(0, 1, 1, 7) - AIC:3931.2154674475537
- RMSE: 70.80014247972981

- R2 Score: 0.990756819854504

```
best_forecast(d_forecast_fuel, 'Fuel Consumption per nautical mile', [1,1,1],[0,1,1,7])
```

RMSE: 161.68641766048893
R2 Score: 0.9815103207432516



```
# Weekly Forecast model
w_forecast = w_kpi[['Sea Temperature (Celsius)', '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)', 'Local Time (h)', 'Trim (m)',
'Total Power Consumed (MW)']]
```

```
sarimax_forecast(w_forecast, 'Total Power Consumed (MW)', 4)
```

Vessel 1 Weekly Power Forecast The same deductions as daily forecasts apply for the weekly analyses too: Best SARIMA(0, 0, 0)x(0, 1, 0, 4) - AIC:140.56432053018307 RMSE: 2.291589529625523

Vessel 2: Weekly Power Forecast The weekly forecast has a very wide confidence interval indicating higher variance, and more uncertainty in forecast

- The true values and the forecasts are constantly offset Best SARIMA(1, 0, 1)x(0, 1, 0, 4) - AIC:102.17357626984798 RMSE: 2.1498565166984434 R2 Score: -1.8479834608544108

A negative R2 indicates a model misinterpretation

```
w_forecast_fuel = w_kpi[['Fuel Consumption per nautical mile', 'Total Power Consumed (MW)']]
```

```
sarimax_forecast(w_forecast_fuel, 'Fuel Consumption per nautical mile', 4) # Weekly forecast 4: weeks constituting a month
```

Vessel 1 Fuel Consumption per nautical mile: Weekly

- Acceptable performance by the model: Best SARIMA(0, 1, 0)x(1, 1, 1, 4) - AIC:452.4464831963399 RMSE: 158.0555408961258

Vessel 2 Fuel Consumption per nautical mile: Weekly

- The model and the true value and miles apart and don't seem to align at all. This could be because of a lack of seasonality in weekly data for the vessel 2

```
weekly_df.reset_index(inplace=True)
weekly_df.head()
```

```
hourly_df.to_csv("hourly.csv")
```

```
# Generate column Table
table = '| Column Name |\n|-----|\n'
for column in weekly_df.columns:
    table += f'| {column} |\n'
print(table)
```