In [5]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.ticker import StrMethodFormatter
import seaborn as sns
```

In [6]:
```
ships = pd.read_csv("C:/Users/ijuzumai/Downloads/AIS.csv")
```

```
C:\Users\ijuzumai\AppData\Local\Temp\ipykernel_12160\198884166.py:1: DtypeWarning: C
olumns (11) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  ships = pd.read_csv("C:/Users/ijuzumai/Downloads/AIS.csv")
```

In [7]:
```
ships
```

Out[7]:

| | Unnamed: 0 | # Timestamp | Type of mobile | MMSI | Latitude | Longitude | Navigational status | R |
|---|---|---|---|---|---|---|---|---|
| **0** | 32 | 01/11/2024 00:00:00 | Class A | 219000429 | 54.654167 | 11.350667 | Under way using engine | |
| **1** | 34 | 01/11/2024 00:00:00 | Class A | 219000429 | 54.654167 | 11.350667 | Under way using engine | |
| **2** | 743 | 01/11/2024 00:00:03 | Class A | 211188000 | 54.599917 | 11.287717 | Under way using engine | |
| **3** | 788 | 01/11/2024 00:00:03 | Class A | 219000431 | 54.506467 | 11.231333 | Under way using engine | 2 |
| **4** | 791 | 01/11/2024 00:00:03 | Class A | 219000431 | 54.506467 | 11.231333 | Under way using engine | 2 |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **354809** | 16995459 | 04/11/2024 23:59:53 | Class A | 211188000 | 54.606033 | 11.293117 | Under way using engine | |
| **354810** | 16995460 | 04/11/2024 23:59:53 | Class A | 211188000 | 54.606033 | 11.293117 | Under way using engine | |
| **354811** | 16995461 | 04/11/2024 23:59:53 | Class A | 211188000 | 54.606033 | 11.293117 | Under way using engine | |
| **354812** | 16996321 | 04/11/2024 23:59:58 | Class A | 219000431 | 54.652917 | 11.350267 | Under way using engine | |
| **354813** | 16996322 | 04/11/2024 23:59:58 | Class A | 219000431 | 54.652917 | 11.350267 | Under way using engine | |

354814 rows × 27 columns

In [8]:
```
pd.options.display.max_columns = None
ships
```

Out[8]:

| | Unnamed: 0 | # Timestamp | Type of mobile | MMSI | Latitude | Longitude | Navigational status | R |
|---|---|---|---|---|---|---|---|---|
| **0** | 32 | 01/11/2024 00:00:00 | Class A | 219000429 | 54.654167 | 11.350667 | Under way using engine | |
| **1** | 34 | 01/11/2024 00:00:00 | Class A | 219000429 | 54.654167 | 11.350667 | Under way using engine | |
| **2** | 743 | 01/11/2024 00:00:03 | Class A | 211188000 | 54.599917 | 11.287717 | Under way using engine | |
| **3** | 788 | 01/11/2024 00:00:03 | Class A | 219000431 | 54.506467 | 11.231333 | Under way using engine | 2 |
| **4** | 791 | 01/11/2024 00:00:03 | Class A | 219000431 | 54.506467 | 11.231333 | Under way using engine | 2 |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **354809** | 16995459 | 04/11/2024 23:59:53 | Class A | 211188000 | 54.606033 | 11.293117 | Under way using engine | |
| **354810** | 16995460 | 04/11/2024 23:59:53 | Class A | 211188000 | 54.606033 | 11.293117 | Under way using engine | |
| **354811** | 16995461 | 04/11/2024 23:59:53 | Class A | 211188000 | 54.606033 | 11.293117 | Under way using engine | |
| **354812** | 16996321 | 04/11/2024 23:59:58 | Class A | 219000431 | 54.652917 | 11.350267 | Under way using engine | |
| **354813** | 16996322 | 04/11/2024 23:59:58 | Class A | 219000431 | 54.652917 | 11.350267 | Under way using engine | |

354814 rows × 27 columns

In [9]:
```
ships.isna().sum()
```

Out[9]:
```
Unnamed: 0                             0
# Timestamp                            0
Type of mobile                         0
MMSI                                   0
Latitude                               0
Longitude                              0
Navigational status                    0
ROT                                    3
SOG                                    0
COG                                   41
Heading                                3
IMO                                    0
Callsign                               0
Name                                 735
Ship type                              0
Cargo type                        247836
Width                                735
Length                               735
Type of position fixing device         0
Draught                           107713
Destination                            0
ETA                                  735
Data source type                       0
A                                    735
B                                    735
C                                    735
D                                    735
dtype: int64
```

In [10]:
```
ships.columns
```

Out[10]:
```
Index(['Unnamed: 0', '# Timestamp', 'Type of mobile', 'MMSI', 'Latitude',
       'Longitude', 'Navigational status', 'ROT', 'SOG', 'COG', 'Heading',
       'IMO', 'Callsign', 'Name', 'Ship type', 'Cargo type', 'Width', 'Length',
       'Type of position fixing device', 'Draught', 'Destination', 'ETA',
       'Data source type', 'A', 'B', 'C', 'D'],
      dtype='object')
```

In [11]:
```
ships.dtypes
```

Out[11]:
```
Unnamed: 0                          int64
# Timestamp                        object
Type of mobile                     object
MMSI                                int64
Latitude                          float64
Longitude                         float64
Navigational status                object
ROT                               float64
SOG                               float64
COG                               float64
Heading                           float64
IMO                                object
Callsign                           object
Name                               object
Ship type                          object
Cargo type                         object
Width                             float64
Length                            float64
Type of position fixing device     object
Draught                           float64
Destination                        object
ETA                                object
Data source type                   object
A                                 float64
B                                 float64
C                                 float64
D                                 float64
dtype: object
```

In [12]:
```
ships['Length'] = ships['A'] + ships['B']
ships['Width'] = ships['C'] + ships['D']
```

In [13]:
```
ships.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 354814 entries, 0 to 354813
Data columns (total 27 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   Unnamed: 0                      354814 non-null  int64  
 1   # Timestamp                     354814 non-null  object 
 2   Type of mobile                  354814 non-null  object 
 3   MMSI                            354814 non-null  int64  
 4   Latitude                        354814 non-null  float64
 5   Longitude                       354814 non-null  float64
 6   Navigational status             354814 non-null  object 
 7   ROT                             354811 non-null  float64
 8   SOG                             354814 non-null  float64
 9   COG                             354773 non-null  float64
 10  Heading                         354811 non-null  float64
 11  IMO                             354814 non-null  object 
 12  Callsign                        354814 non-null  object 
 13  Name                            354079 non-null  object 
 14  Ship type                       354814 non-null  object 
 15  Cargo type                      106978 non-null  object 
 16  Width                           354079 non-null  float64
 17  Length                          354079 non-null  float64
 18  Type of position fixing device  354814 non-null  object 
 19  Draught                         247101 non-null  float64
 20  Destination                     354814 non-null  object 
 21  ETA                             354079 non-null  object 
 22  Data source type                354814 non-null  object 
 23  A                               354079 non-null  float64
 24  B                               354079 non-null  float64
 25  C                               354079 non-null  float64
 26  D                               354079 non-null  float64
dtypes: float64(13), int64(2), object(12)
memory usage: 73.1+ MB
```

In [14]:
```
ships.rename(columns={'# Timestamp': 'Timestamp'}, inplace=True)
```

In [15]:
```
ships["Timestamp"]
```

Out[15]:
```
0         01/11/2024 00:00:00
1         01/11/2024 00:00:00
2         01/11/2024 00:00:03
3         01/11/2024 00:00:03
4         01/11/2024 00:00:03
                 ...
354809    04/11/2024 23:59:53
354810    04/11/2024 23:59:53
354811    04/11/2024 23:59:53
354812    04/11/2024 23:59:58
354813    04/11/2024 23:59:58
Name: Timestamp, Length: 354814, dtype: object
```

In [16]:
```
ships["ETA"]
```

Out[16]:
```
0                       NaN
1                       NaN
2                       NaN
3                       NaN
4                       NaN
                  ...
354809    31/12/2024 12:00:00
354810    31/12/2024 12:00:00
354811    31/12/2024 12:00:00
354812    31/12/2024 00:00:00
354813    31/12/2024 00:00:00
Name: ETA, Length: 354814, dtype: object
```

In [17]:
```
ships['ROT'].value_counts()
```

Out[17]:
```
ROT
 0.0     115915
-1.1      41305
 1.1      39282
-2.2      25094
 2.2      23997
-2.9      14813
 2.9      14045
-3.6       8708
 3.6       8635
-7.5       6779
 7.5       6494
-5.4       5833
 5.4       5626
 6.4       4585
-6.4       4494
-11.4      3720
 11.4      3227
-8.7       2623
-10.0      2360
 8.7       2344
 10.0      2119
 14.5      1848
-14.5      1730
-12.9      1538
 12.9      1330
 17.9       904
 19.7       678
 16.1       677
 21.6       550
-16.1       520
-17.9       513
 23.6       422
 27.9       396
 25.7       317
 35.0       277
 30.2       257
-19.7       179
 32.5       160
-21.6        97
-23.6        97
 37.5        89
 45.7        48
 40.2        44
 42.9        42
-25.7        40
 48.6        17
-30.2        15
-27.9        13
 51.6         9
 54.7         3
 57.9         3
Name: count, dtype: int64
```

In [18]:
```
ships[ships['ROT'].isnull()]
```

Out[18]:

| | Unnamed: 0 | Timestamp | Type of mobile | MMSI | Latitude | Longitude | Navigational status | RO |
|---|---|---|---|---|---|---|---|---|
| **45611** | 7387740 | 01/11/2024 12:10:14 | Class A | 219000431 | 54.502648 | 11.226785 | Under way using engine | Na |
| **45612** | 7387741 | 01/11/2024 12:10:14 | Class A | 219000431 | 54.502648 | 11.226785 | Under way using engine | Na |
| **45613** | 7387742 | 01/11/2024 12:10:14 | Class A | 219000431 | 54.502648 | 11.226785 | Under way using engine | Na |

In [19]:
```
ships['ROT'] = ships['ROT'].fillna(0)
```

In [20]:
```
ships['Heading'].value_counts()
```

Out[20]:
```
Heading
230.0    15492
35.0     13890
211.0    11955
215.0    11618
50.0      9860
         ...
171.0       13
56.0         9
243.0        6
176.0        5
57.0         3
Name: count, Length: 146, dtype: int64
```

In [21]:
```
ships[ships['Heading'].isnull()]
```

Out[21]:

| | Unnamed: 0 | Timestamp | Type of mobile | MMSI | Latitude | Longitude | Navigational status | RC |
|---|---|---|---|---|---|---|---|---|
| **45611** | 7387740 | 01/11/2024 12:10:14 | Class A | 219000431 | 54.502648 | 11.226785 | Under way using engine | 0 |
| **45612** | 7387741 | 01/11/2024 12:10:14 | Class A | 219000431 | 54.502648 | 11.226785 | Under way using engine | 0 |
| **45613** | 7387742 | 01/11/2024 12:10:14 | Class A | 219000431 | 54.502648 | 11.226785 | Under way using engine | 0 |

In [22]:
```
ships.iloc[45609:45615]
```

Out[22]:

| | Unnamed: 0 | Timestamp | Type of mobile | MMSI | Latitude | Longitude | Navigational status | RC |
|---|---|---|---|---|---|---|---|---|
| **45609** | 7387541 | 01/11/2024 12:10:13 | Class A | 211188000 | 54.561617 | 11.274817 | Under way using engine | 0 |
| **45610** | 7387544 | 01/11/2024 12:10:13 | Class A | 211188000 | 54.561617 | 11.274817 | Under way using engine | 0 |
| **45611** | 7387740 | 01/11/2024 12:10:14 | Class A | 219000431 | 54.502648 | 11.226785 | Under way using engine | 0 |
| **45612** | 7387741 | 01/11/2024 12:10:14 | Class A | 219000431 | 54.502648 | 11.226785 | Under way using engine | 0 |
| **45613** | 7387742 | 01/11/2024 12:10:14 | Class A | 219000431 | 54.502648 | 11.226785 | Under way using engine | 0 |
| **45614** | 7388517 | 01/11/2024 12:10:19 | Class A | 211188000 | 54.561167 | 11.274567 | Under way using engine | 1 |

In [23]:
```
ships.drop(
    ships[(ships['Heading'].isna()) & (ships['COG'].isna()) & (ships['SOG'] == 0) &
    inplace=True)
```

In [217…
```
ships.isna().sum()
```

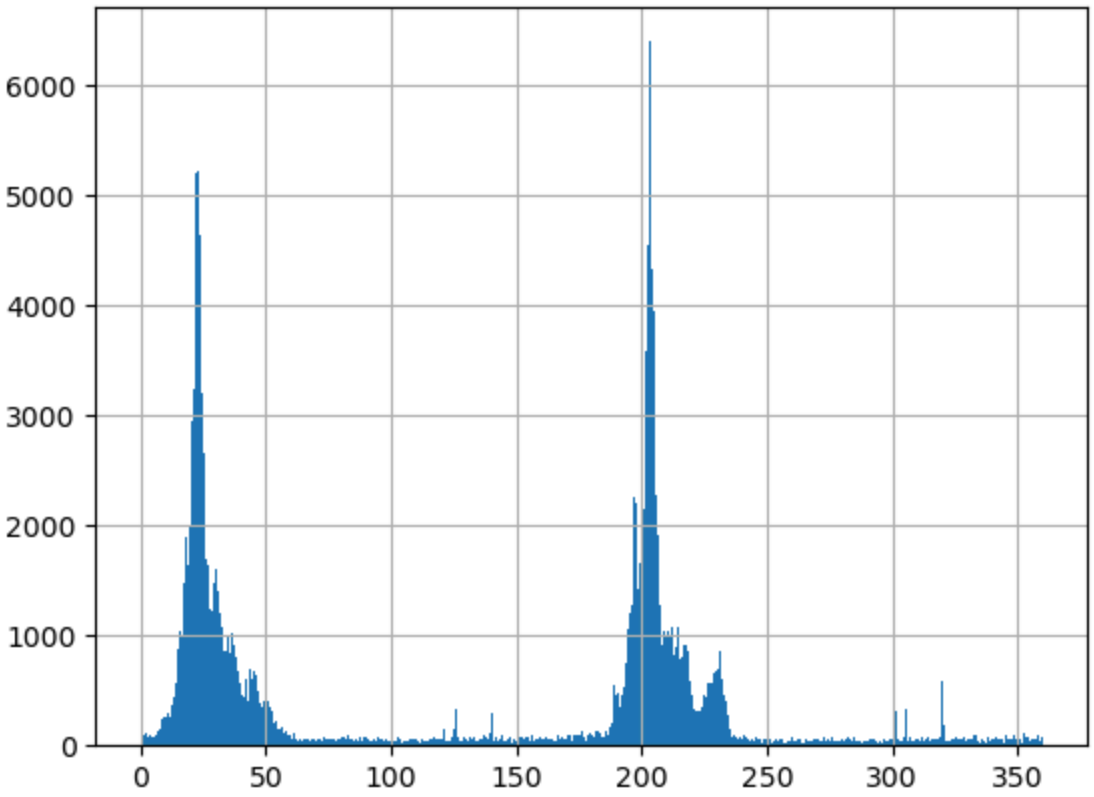
Out[217…
```
Timestamp                            0
Type of mobile                       0
MMSI                                 0
Latitude                             0
Longitude                            0
ROT                                  0
SOG                                  0
COG                                 16
Heading                              0
IMO                                  0
Callsign                             0
Name                                 0
Width                                0
Length                               0
Type of position fixing device       0
Draught                          44470
Destination                          0
ETA                                  0
A                                    0
B                                    0
C                                    0
D                                    0
dtype: int64
```

In [25]:
```
ships['COG'].value_counts()
```

Out[25]:
```
COG
203.3    1657
203.1    1624
203.2    1561
203.0    1559
202.9    1517
         ...
293.1       2
290.2       2
104.0       1
264.5       1
327.0       1
Name: count, Length: 3591, dtype: int64
```

In [26]:
```
ships['COG'].hist(bins=1000)
```

Out[26]:
```
<Axes: >
```




In [27]:
```
ships[ships['COG'].isnull()]
```

Out[27]:

| | Unnamed: 0 | Timestamp | Type of mobile | MMSI | Latitude | Longitude | Navigational status | R |
|---|---|---|---|---|---|---|---|---|
| **45617** | 7389420 | 01/11/2024 12:10:24 | Class A | 219000431 | 54.502648 | 11.226785 | Under way using engine | |
| **45618** | 7389422 | 01/11/2024 12:10:24 | Class A | 219000431 | 54.502648 | 11.226785 | Under way using engine | |
| **45619** | 7389424 | 01/11/2024 12:10:24 | Class A | 219000431 | 54.502648 | 11.226785 | Under way using engine | |
| **45627** | 7390975 | 01/11/2024 12:10:33 | Class A | 219000431 | 54.502650 | 11.226787 | Under way using engine | |
| **45628** | 7390976 | 01/11/2024 12:10:33 | Class A | 219000431 | 54.502650 | 11.226787 | Under way using engine | |
| **45629** | 7390977 | 01/11/2024 12:10:33 | Class A | 219000431 | 54.502650 | 11.226787 | Under way using engine | |
| **45637** | 7392789 | 01/11/2024 12:10:44 | Class A | 219000431 | 54.502650 | 11.226788 | Under way using engine | |
| **45638** | 7392791 | 01/11/2024 12:10:44 | Class A | 219000431 | 54.502650 | 11.226788 | Under way using engine | |
| **45639** | 7392792 | 01/11/2024 12:10:44 | Class A | 219000431 | 54.502650 | 11.226788 | Under way using engine | |
| **45643** | 7394425 | 01/11/2024 12:10:53 | Class A | 219000431 | 54.502652 | 11.226790 | Under way using engine | |
| **45644** | 7394427 | 01/11/2024 12:10:53 | Class A | 219000431 | 54.502652 | 11.226790 | Under way using engine | |
| **45645** | 7394644 | 01/11/2024 12:10:54 | Class A | 219000431 | 54.502652 | 11.226790 | Under way using engine | |
| **45646** | 7394645 | 01/11/2024 12:10:54 | Class A | 219000431 | 54.502652 | 11.226790 | Under way using engine | |

| | Unnamed: 0 | Timestamp | Type of mobile | MMSI | Latitude | Longitude | Navigational status | R |
|---|---|---|---|---|---|---|---|---|
| **264964** | 15568569 | 03/11/2024 23:26:47 | Class A | 219000429 | 54.502677 | 11.226883 | Under way using engine | |
| **264965** | 15568570 | 03/11/2024 23:26:47 | Class A | 219000429 | 54.502677 | 11.226883 | Under way using engine | |
| **264974** | 15570560 | 03/11/2024 23:26:58 | Class A | 219000429 | 54.502678 | 11.226887 | Under way using engine | |
| **264975** | 15570561 | 03/11/2024 23:26:58 | Class A | 219000429 | 54.502678 | 11.226887 | Under way using engine | |
| **264983** | 15572198 | 03/11/2024 23:27:06 | Class A | 219000429 | 54.502678 | 11.226893 | Under way using engine | |
| **264984** | 15572199 | 03/11/2024 23:27:06 | Class A | 219000429 | 54.502678 | 11.226893 | Under way using engine | |
| **264985** | 15572200 | 03/11/2024 23:27:06 | Class A | 219000429 | 54.502678 | 11.226893 | Under way using engine | |
| **295345** | 5065654 | 04/11/2024 07:06:46 | Class A | 219000431 | 54.653453 | 11.349960 | Under way using engine | |
| **295346** | 5065657 | 04/11/2024 07:06:46 | Class A | 219000431 | 54.653453 | 11.349960 | Under way using engine | |
| **295347** | 5065659 | 04/11/2024 07:06:46 | Class A | 219000431 | 54.653453 | 11.349960 | Under way using engine | |
| **295348** | 5065747 | 04/11/2024 07:06:46 | Class A | 219000431 | 54.653453 | 11.349960 | Under way using engine | |
| **295349** | 5065750 | 04/11/2024 07:06:46 | Class A | 219000431 | 54.653453 | 11.349960 | Under way using engine | |
| **295350** | 5065751 | 04/11/2024 07:06:46 | Class A | 219000431 | 54.653453 | 11.349960 | Under way using engine | |
| **295357** | 5067857 | 04/11/2024 07:06:56 | Class A | 219000431 | 54.653452 | 11.349962 | Under way using engine | |
| **295358** | 5067858 | 04/11/2024 07:06:56 | Class A | 219000431 | 54.653452 | 11.349962 | Under way using engine | |

| | Unnamed: 0 | Timestamp | Type of mobile | MMSI | Latitude | Longitude | Navigational status | R |
|---|---|---|---|---|---|---|---|---|
| **295359** | 5067860 | 04/11/2024 07:06:56 | Class A | 219000431 | 54.653452 | 11.349962 | Under way using engine | |
| **295366** | 5069744 | 04/11/2024 07:07:05 | Class A | 219000431 | 54.653452 | 11.349960 | Under way using engine | |
| **295372** | 5071791 | 04/11/2024 07:07:15 | Class A | 219000431 | 54.653453 | 11.349960 | Under way using engine | |
| **295373** | 5071793 | 04/11/2024 07:07:15 | Class A | 219000431 | 54.653453 | 11.349960 | Under way using engine | |
| **295380** | 5074257 | 04/11/2024 07:07:27 | Class A | 219000431 | 54.653458 | 11.349957 | Under way using engine | |
| **295381** | 5074258 | 04/11/2024 07:07:27 | Class A | 219000431 | 54.653458 | 11.349957 | Under way using engine | |
| **295382** | 5074259 | 04/11/2024 07:07:27 | Class A | 219000431 | 54.653458 | 11.349957 | Under way using engine | |
| **295389** | 5076377 | 04/11/2024 07:07:36 | Class A | 219000431 | 54.653458 | 11.349953 | Under way using engine | |
| **295396** | 5077849 | 04/11/2024 07:07:43 | Class A | 219000431 | 54.653458 | 11.349953 | Under way using engine | |
| **295397** | 5077850 | 04/11/2024 07:07:43 | Class A | 219000431 | 54.653458 | 11.349953 | Under way using engine | |

In [28]:
```
ships = ships.drop(columns=['Unnamed: 0'])
```

In [29]:
```
ships = ships.drop_duplicates()
```

In [30]:
```
print(ships['IMO'].dtype)
```

```
object
```

In [31]:
```
ships['IMO'] = ships['IMO'].astype(str)
ships['Callsign'] = ships['IMO'].astype(str)
```

In [32]:
```
ships[ships['COG'].isnull()]
```

Out[32]:

| | Timestamp | Type of mobile | MMSI | Latitude | Longitude | Navigational status | ROT | SOG | C |
|---|---|---|---|---|---|---|---|---|---|
| **45617** | 01/11/2024 12:10:24 | Class A | 219000431 | 54.502648 | 11.226785 | Under way using engine | 0.0 | 0.0 | N |
| **45627** | 01/11/2024 12:10:33 | Class A | 219000431 | 54.502650 | 11.226787 | Under way using engine | 0.0 | 0.0 | N |
| **45637** | 01/11/2024 12:10:44 | Class A | 219000431 | 54.502650 | 11.226788 | Under way using engine | 0.0 | 0.0 | N |
| **45643** | 01/11/2024 12:10:53 | Class A | 219000431 | 54.502652 | 11.226790 | Under way using engine | 0.0 | 0.0 | N |
| **45645** | 01/11/2024 12:10:54 | Class A | 219000431 | 54.502652 | 11.226790 | Under way using engine | 0.0 | 0.0 | N |
| **264964** | 03/11/2024 23:26:47 | Class A | 219000429 | 54.502677 | 11.226883 | Under way using engine | 0.0 | 0.0 | N |
| **264974** | 03/11/2024 23:26:58 | Class A | 219000429 | 54.502678 | 11.226887 | Under way using engine | 0.0 | 0.0 | N |
| **264983** | 03/11/2024 23:27:06 | Class A | 219000429 | 54.502678 | 11.226893 | Under way using engine | 0.0 | 0.0 | N |
| **295345** | 04/11/2024 07:06:46 | Class A | 219000431 | 54.653453 | 11.349960 | Under way using engine | 0.0 | 0.0 | N |
| **295348** | 04/11/2024 07:06:46 | Class A | 219000431 | 54.653453 | 11.349960 | Under way using engine | 0.0 | 0.0 | N |
| **295357** | 04/11/2024 07:06:56 | Class A | 219000431 | 54.653452 | 11.349962 | Under way using engine | 0.0 | 0.0 | N |
| **295366** | 04/11/2024 07:07:05 | Class A | 219000431 | 54.653452 | 11.349960 | Under way using engine | 0.0 | 0.0 | N |
| **295372** | 04/11/2024 07:07:15 | Class A | 219000431 | 54.653453 | 11.349960 | Under way using engine | 0.0 | 0.0 | N |
| **295380** | 04/11/2024 07:07:27 | Class A | 219000431 | 54.653458 | 11.349957 | Under way using engine | 0.0 | 0.0 | N |

| | Timestamp | Type of mobile | MMSI | Latitude | Longitude | Navigational status | ROT | SOG | C |
|---|---|---|---|---|---|---|---|---|---|
| **295389** | 04/11/2024 07:07:36 | Class A | 219000431 | 54.653458 | 11.349953 | Under way using engine | 0.0 | 0.0 | N |
| **295396** | 04/11/2024 07:07:43 | Class A | 219000431 | 54.653458 | 11.349953 | Under way using engine | 0.0 | 0.0 | N |

In [33]:
```
ships.isna().sum()
```

Out[33]:
```
Timestamp                          0
Type of mobile                     0
MMSI                               0
Latitude                           0
Longitude                          0
Navigational status                0
ROT                                0
SOG                                0
COG                               16
Heading                            0
IMO                                0
Callsign                           0
Name                             304
Ship type                          0
Cargo type                     90533
Width                            304
Length                           304
Type of position fixing device     0
Draught                        44774
Destination                        0
ETA                              304
Data source type                   0
A                                304
B                                304
C                                304
D                                304
dtype: int64
```

In [34]:
```
ships['Cargo type'].value_counts()
```

Out[34]:
```
Cargo type
No additional information    44470
Name: count, dtype: int64
```

In [35]:
```
ships['Cargo type'].isna().sum()
```

Out[35]:
```
90533
```

In [36]:
```
ships = ships.drop(columns=['Cargo type'])
```

In [37]:
```
ships['Data source type'].value_counts()
```

Out[37]:
```
Data source type
AIS    135003
Name: count, dtype: int64
```

In [38]:
```
ships = ships.drop(columns=['Data source type'])
```

In [39]:
```
ships['Draught'].value_counts()
```

Out[39]:
```
Draught
5.6    45229
5.2    45000
Name: count, dtype: int64
```

In [40]:
```
ships['Draught'].isna().sum()
```

Out[40]:
```
44774
```

In [41]:
```
ships.isna().sum()
```

Out[41]:
```
Timestamp                          0
Type of mobile                     0
MMSI                               0
Latitude                           0
Longitude                          0
Navigational status                0
ROT                                0
SOG                                0
COG                               16
Heading                            0
IMO                                0
Callsign                           0
Name                             304
Ship type                          0
Width                            304
Length                           304
Type of position fixing device     0
Draught                        44774
Destination                        0
ETA                              304
A                                304
B                                304
C                                304
D                                304
dtype: int64
```

In [42]:
```
ships['Destination'].value_counts()
```

Out[42]:
```
Destination
DEPUT <-> DKROD        45229
ROEDBY-PUTTGARTEN      45000
ROEDBY<->PUTTGARDEN    44470
Unknown                  304
Name: count, dtype: int64
```

In [43]:
```
ships['Destination'] = ships['Destination'].replace({
    'ROEDBY<->PUTTGARDEN': 'ROEDBY-PUTTGARTEN',
    'ROEDBY-PUTTGARTEN': 'ROEDBY-PUTTGARTEN',
    'DEPUT <-> DKROD' : 'DEPUT-DKROD' })
```

In [44]:
```
ships['Destination'].value_counts()
```

Out[44]:
```
Destination
ROEDBY-PUTTGARTEN    89470
DEPUT-DKROD          45229
Unknown                304
Name: count, dtype: int64
```

In [45]:
```
ships['MMSI'].value_counts()
```

Out[45]:
```
MMSI
211188000    45335
219000429    45086
219000431    44582
Name: count, dtype: int64
```

In [46]:
```
ships['Name'].value_counts()
```

Out[46]:
```
Name
DEUTSCHLAND            45229
PRINS RICHARD          45000
PRINSESSE BENEDIKTE    44470
Name: count, dtype: int64
```

In [47]:
```
ships.isna().sum()
```

Out[47]:
```
Timestamp                            0
Type of mobile                       0
MMSI                                 0
Latitude                             0
Longitude                            0
Navigational status                  0
ROT                                  0
SOG                                  0
COG                                 16
Heading                              0
IMO                                  0
Callsign                             0
Name                               304
Ship type                            0
Width                              304
Length                             304
Type of position fixing device       0
Draught                          44774
Destination                          0
ETA                                304
A                                  304
B                                  304
C                                  304
D                                  304
dtype: int64
```

In [48]:
```
ships['Navigational status'].value_counts()
```

Out[48]:
```
Navigational status
Under way using engine    135003
Name: count, dtype: int64
```

In [49]:
```
ships = ships.drop(columns=['Navigational status'])
```

In [50]:
```
ships['IMO'].value_counts()
```

Out[50]:
```
IMO
9151541    45229
9144419    45000
9144421    44470
Unknown      304
Name: count, dtype: int64
```

In [51]:
```
304/134997
```

Out[51]:
```
0.0022519018941161654
```

In [52]:
```
ships = ships.dropna(subset=['Width'])
```

In [53]:
```
ships.isna().sum()
```

Out[53]:
```
Timestamp                            0
Type of mobile                       0
MMSI                                 0
Latitude                             0
Longitude                            0
ROT                                  0
SOG                                  0
COG                                 16
Heading                              0
IMO                                  0
Callsign                             0
Name                                 0
Ship type                            0
Width                                0
Length                               0
Type of position fixing device       0
Draught                          44470
Destination                          0
ETA                                  0
A                                    0
B                                    0
C                                    0
D                                    0
dtype: int64
```

In [54]:
```
ships['Name'].value_counts()
```

Out[54]:
```
Name
DEUTSCHLAND            45229
PRINS RICHARD          45000
PRINSESSE BENEDIKTE    44470
Name: count, dtype: int64
```

In [55]:
```
89466+45227
```

Out[55]:
```
134693
```

In [56]:
```
ships = ships.drop(columns=['Ship type'])
```

In [57]:
```
ships
```

Out[57]:

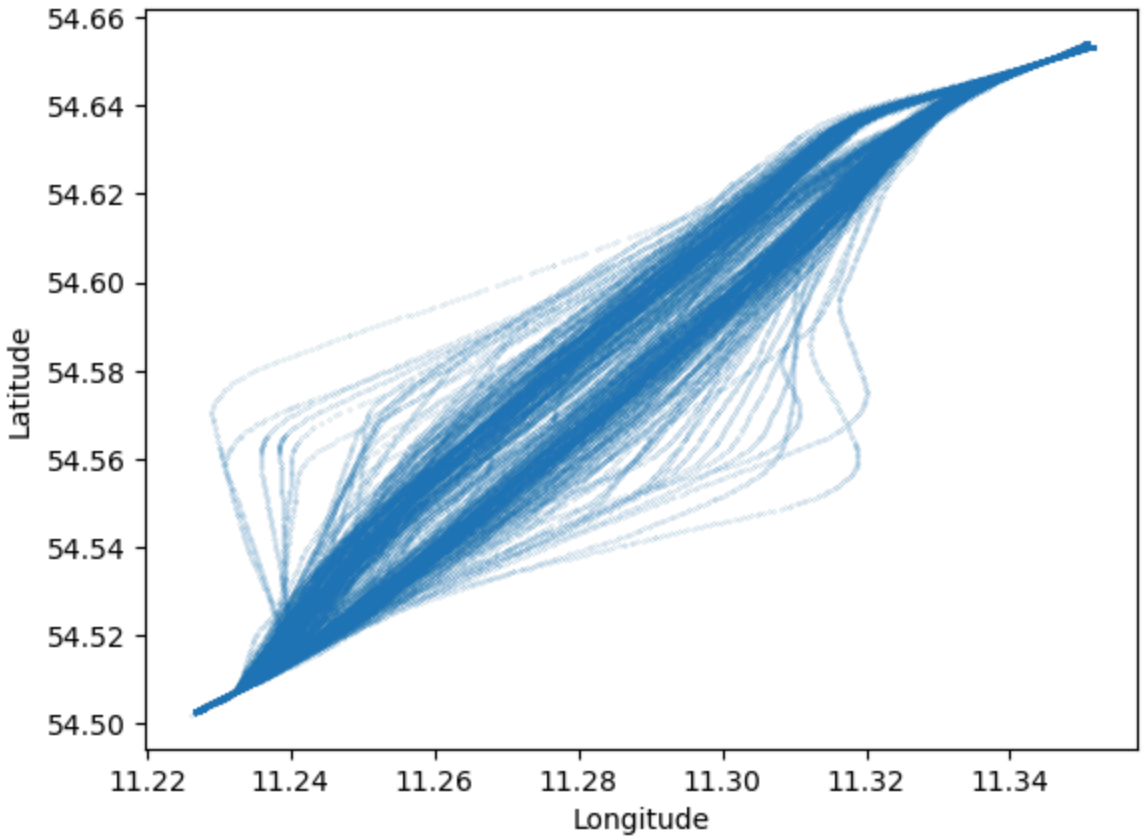
| | Timestamp | Type of mobile | MMSI | Latitude | Longitude | ROT | SOG | COG | Heading |
|---|---|---|---|---|---|---|---|---|---|
| **5** | 01/11/2024 00:00:07 | Class A | 219000431 | 54.506467 | 11.231333 | 25.7 | 10.2 | 213.4 | 213.0 |
| **12** | 01/11/2024 00:00:10 | Class A | 219000431 | 54.506200 | 11.231033 | 21.6 | 9.7 | 215.6 | 215.0 |
| **17** | 01/11/2024 00:00:17 | Class A | 219000431 | 54.505950 | 11.230733 | 14.5 | 9.2 | 216.2 | 217.0 |
| **21** | 01/11/2024 00:00:20 | Class A | 219000431 | 54.505883 | 11.230650 | 10.0 | 8.9 | 217.3 | 217.0 |
| **23** | 01/11/2024 00:00:20 | Class A | 219000429 | 54.654200 | 11.350700 | 0.0 | 0.3 | 43.5 | 31.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **354801** | 04/11/2024 23:59:43 | Class A | 219000429 | 54.654167 | 11.350700 | 0.0 | 0.4 | 34.4 | 31.0 |
| **354804** | 04/11/2024 23:59:47 | Class A | 219000431 | 54.652917 | 11.350267 | 0.0 | 0.0 | 74.9 | 230.0 |
| **354807** | 04/11/2024 23:59:53 | Class A | 219000429 | 54.654183 | 11.350717 | 0.0 | 0.3 | 30.9 | 31.0 |
| **354809** | 04/11/2024 23:59:53 | Class A | 211188000 | 54.606033 | 11.293117 | 0.0 | 13.4 | 204.8 | 205.0 |
| **354812** | 04/11/2024 23:59:58 | Class A | 219000431 | 54.652917 | 11.350267 | 0.0 | 0.0 | 192.8 | 230.0 |

134699 rows × 22 columns

In [58]:
```
print(ships[['Longitude', 'Latitude']].dtypes)
```

```
Longitude    float64
Latitude     float64
dtype: object
```

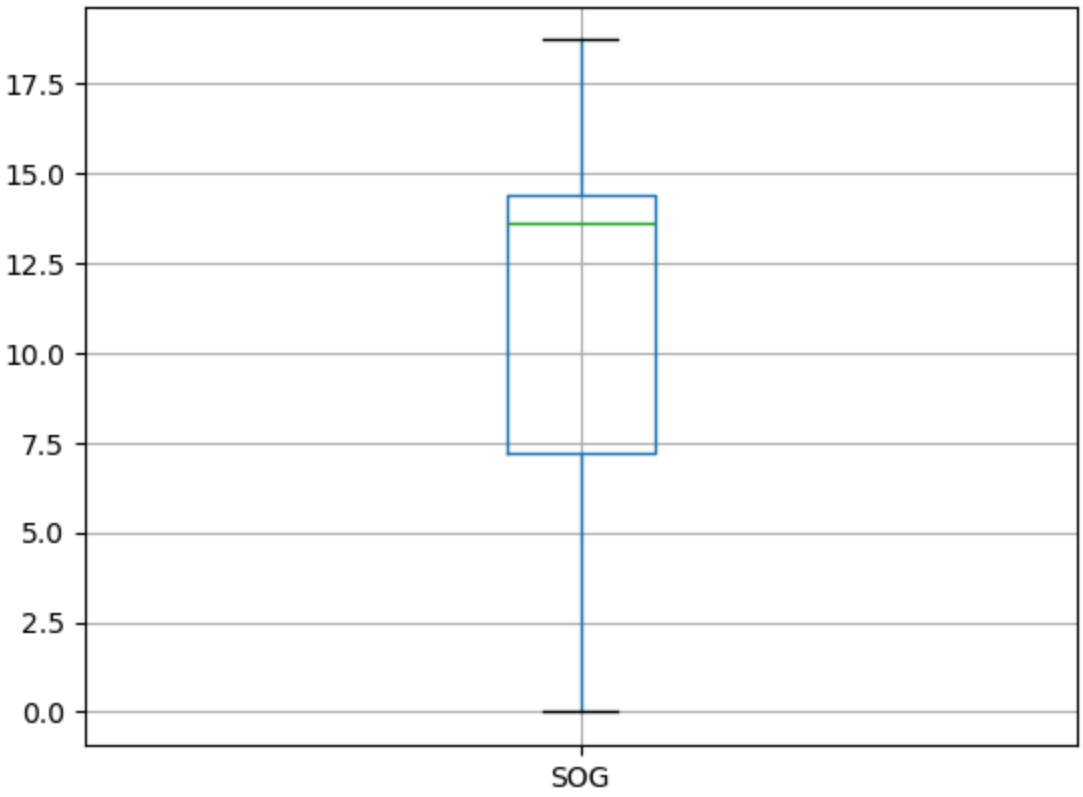
In [59]:
```
ship_locations = ships.plot.scatter(x='Longitude', y='Latitude', s=0.01)
```




In [60]:
```
ships["Timestamp"] = pd.to_datetime(ships["Timestamp"], dayfirst=True)
ships["ETA"] = pd.to_datetime(ships["ETA"], dayfirst=True)
```

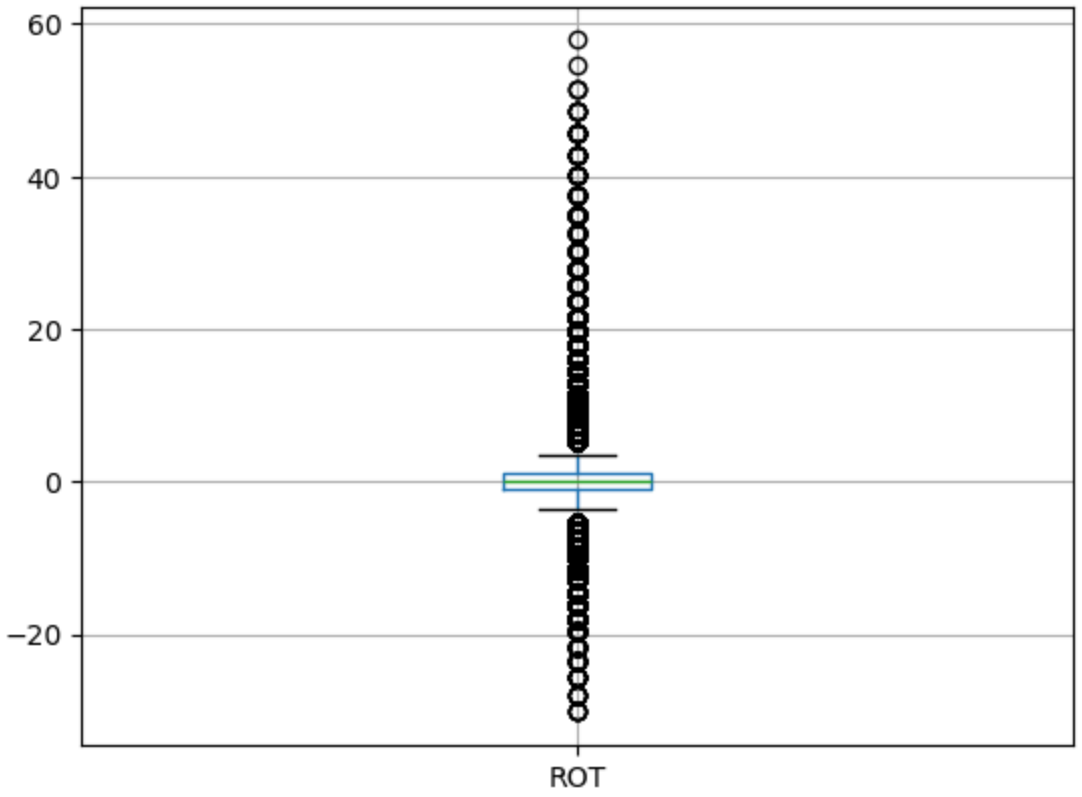
In [61]:
```
ships.boxplot(column='SOG')
```

Out[61]: <Axes: >




In [62]:
```
ships.boxplot(column='ROT')
```

Out[62]: <Axes: >

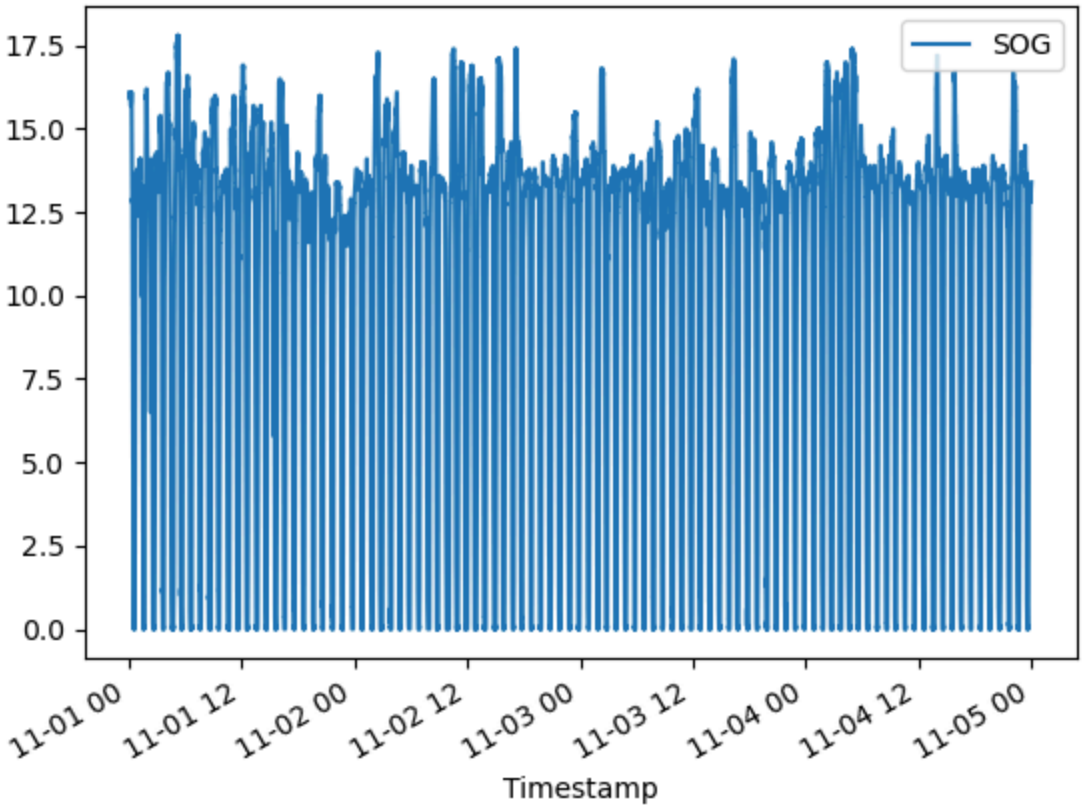



In [63]:
```
ships['SOG'] = pd.to_numeric(ships['SOG'])
```

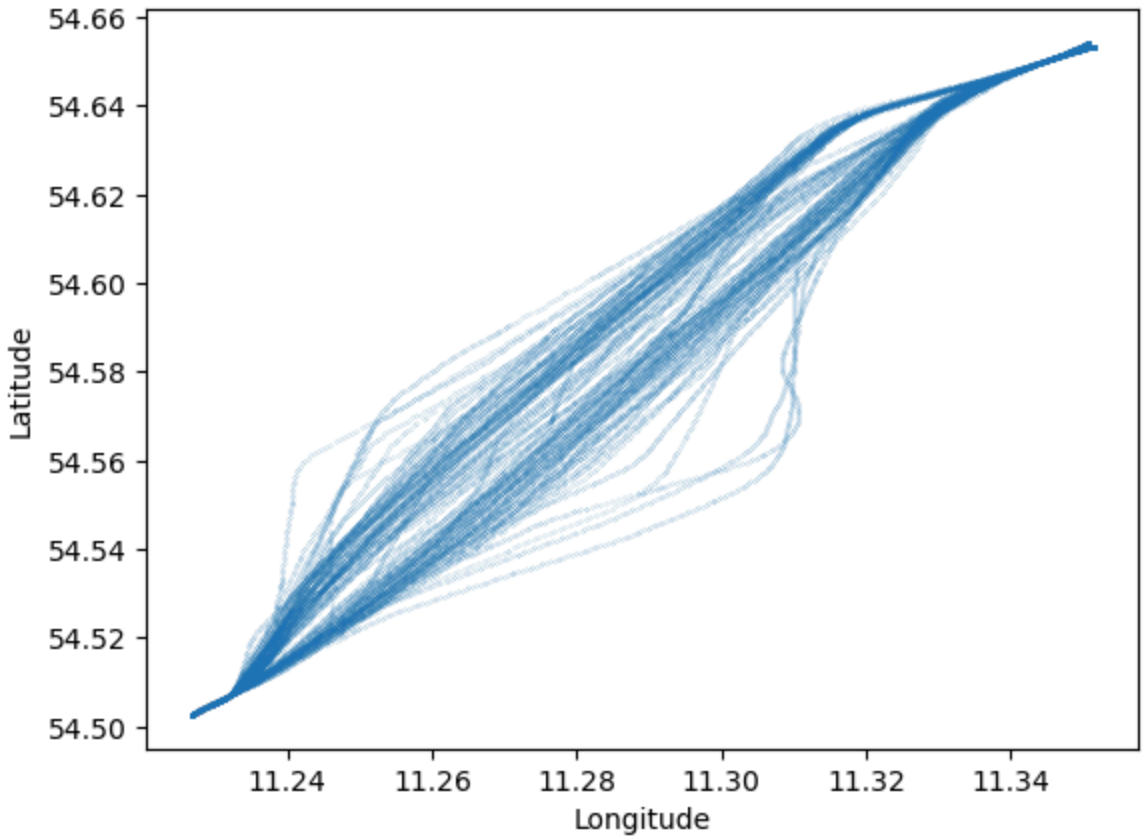
In [64]:
```
ship1_mmsi = 211188000
essential_cols = ['Timestamp', 'MMSI', 'Latitude', 'Longitude', 'ROT', 'SOG','COG',
ship1 = ships[ships["MMSI"] == ship1_mmsi][essential_cols].sort_values(by="Timestam
```

In [65]:
```
ship1.plot.line(x='Timestamp', y='SOG')
```

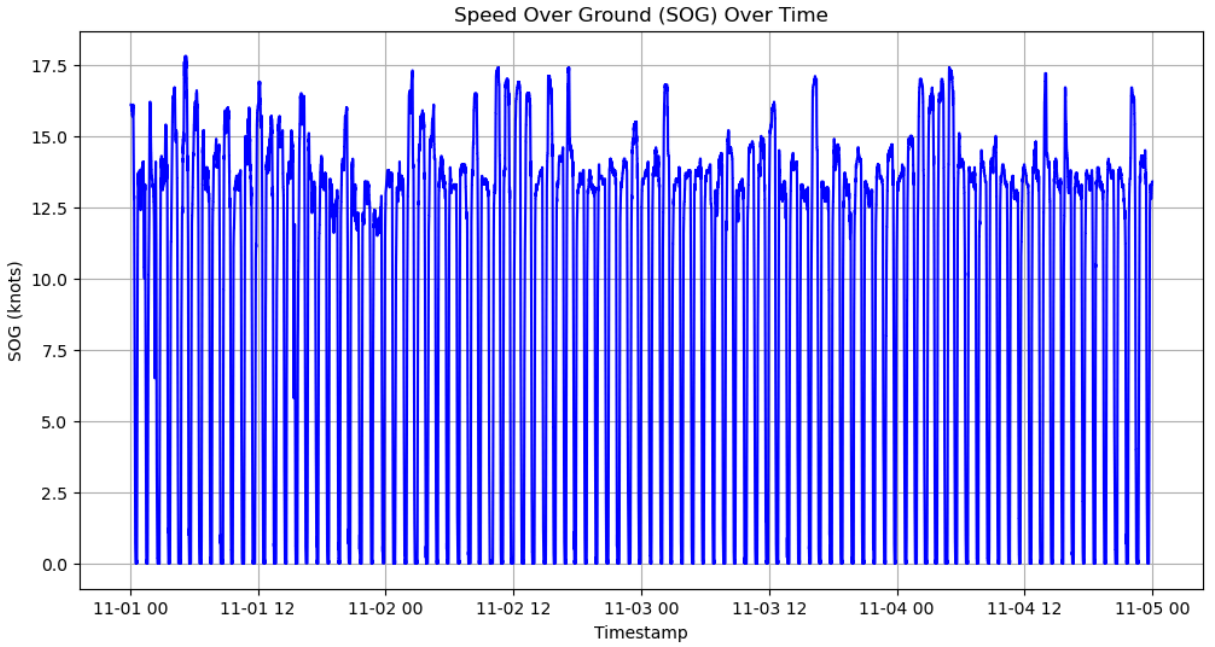
Out[65]:
```
<Axes: xlabel='Timestamp'>
```



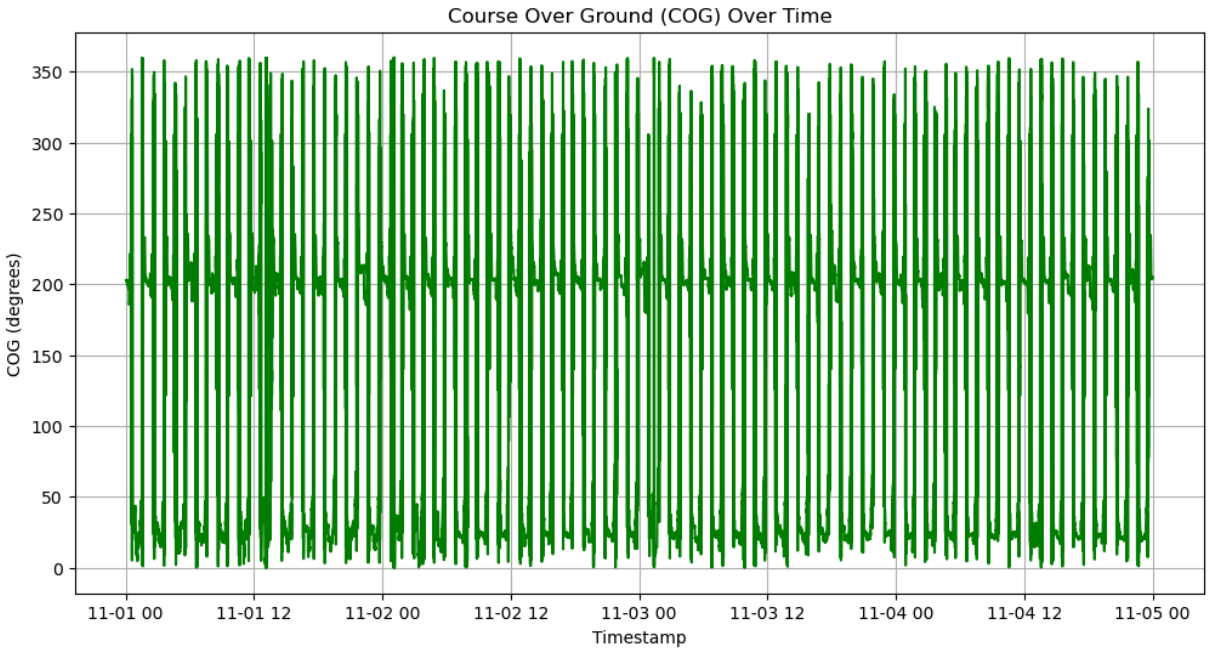

In [66]:
```
ship1_location = ship1.plot.scatter(x='Longitude', y='Latitude', s=0.01)
```




In [67]:
```
plt.figure(figsize=(12, 6))
plt.plot(ship1['Timestamp'], ship1['SOG'], label='Speed Over Ground', color='blue')
plt.title('Speed Over Ground (SOG) Over Time')
plt.xlabel('Timestamp')
plt.ylabel('SOG (knots)')
plt.grid(True)
plt.show()
```



In [68]:
```
plt.figure(figsize=(12, 6))
plt.plot(ship1['Timestamp'], ship1['COG'], label='Course Over Ground', color='green
plt.title('Course Over Ground (COG) Over Time')
plt.xlabel('Timestamp')
plt.ylabel('COG (degrees)')
plt.grid(True)
plt.show()
```




In [69]:
```
ship1.isna().sum()
```

Out[69]:
```
Timestamp      0
MMSI           0
Latitude       0
Longitude      0
ROT            0
SOG            0
COG            0
Heading        0
Destination    0
ETA            0
dtype: int64
```

In [70]:
```
print(ship1.dtypes)
```

```
Timestamp      datetime64[ns]
MMSI                    int64
Latitude              float64
Longitude             float64
ROT                   float64
SOG                   float64
COG                   float64
Heading               float64
Destination            object
ETA            datetime64[ns]
dtype: object
```

In [71]:
```
ship1['ETA'].value_counts()
```

Out[71]:
```
ETA
2024-12-31 12:00:00    45229
Name: count, dtype: int64
```

In [72]:
```
ship1.head()
```

Out[72]:

| | Timestamp | MMSI | Latitude | Longitude | ROT | SOG | COG | Heading | Destinatior |
|---|---|---|---|---|---|---|---|---|---|
| **222** | 2024-11-01 00:03:51 | 211188000 | 54.584183 | 11.276583 | -1.1 | 16.1 | 202.8 | 212.0 | DEPUT DKROI |
| **232** | 2024-11-01 00:03:56 | 211188000 | 54.583767 | 11.276283 | 0.0 | 16.1 | 202.8 | 212.0 | DEPUT DKROI |
| **242** | 2024-11-01 00:04:02 | 211188000 | 54.583367 | 11.275983 | 1.1 | 16.1 | 202.8 | 212.0 | DEPUT DKROI |
| **251** | 2024-11-01 00:04:09 | 211188000 | 54.582950 | 11.275683 | 0.0 | 16.1 | 202.9 | 212.0 | DEPUT DKROI |
| **259** | 2024-11-01 00:04:14 | 211188000 | 54.582533 | 11.275400 | -2.9 | 16.1 | 202.7 | 212.0 | DEPUT DKROI |

In [73]:
```
ship1.columns
```

Out[73]:
```
Index(['Timestamp', 'MMSI', 'Latitude', 'Longitude', 'ROT', 'SOG', 'COG',
       'Heading', 'Destination', 'ETA'],
      dtype='object')
```

In [74]:
```
ship1 = ship1.reset_index()
ship1
```

Out[74]:

| | index | Timestamp | MMSI | Latitude | Longitude | ROT | SOG | COG | Heading |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 222 | 2024-11-01 00:03:51 | 211188000 | 54.584183 | 11.276583 | -1.1 | 16.1 | 202.8 | 212.0 |
| **1** | 232 | 2024-11-01 00:03:56 | 211188000 | 54.583767 | 11.276283 | 0.0 | 16.1 | 202.8 | 212.0 |
| **2** | 242 | 2024-11-01 00:04:02 | 211188000 | 54.583367 | 11.275983 | 1.1 | 16.1 | 202.8 | 212.0 |
| **3** | 251 | 2024-11-01 00:04:09 | 211188000 | 54.582950 | 11.275683 | 0.0 | 16.1 | 202.9 | 212.0 |
| **4** | 259 | 2024-11-01 00:04:14 | 211188000 | 54.582533 | 11.275400 | -2.9 | 16.1 | 202.7 | 212.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **45224** | 354782 | 2024-11-04 23:59:23 | 211188000 | 54.607783 | 11.294517 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45225** | 354788 | 2024-11-04 23:59:27 | 211188000 | 54.607783 | 11.294517 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45226** | 354794 | 2024-11-04 23:59:34 | 211188000 | 54.607150 | 11.294017 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45227** | 354799 | 2024-11-04 23:59:42 | 211188000 | 54.606650 | 11.293600 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45228** | 354809 | 2024-11-04 23:59:53 | 211188000 | 54.606033 | 11.293117 | 0.0 | 13.4 | 204.8 | 205.0 |

45229 rows × 11 columns

In [75]:
```
ship1['Destination'].value_counts()
```

Out[75]:
```
Destination
DEPUT-DKROD    45229
Name: count, dtype: int64
```

In [76]:
```
def haversine(lat1, lon1, lat2, lon2, to_radians=True, earth_radius_nm=3440):

    if to_radians:
        lat1, lon1, lat2, lon2 = np.radians([lat1, lon1, lat2, lon2])
```

```
    a = np.sin((lat2 - lat1) / 2.0) ** 2 + \
        np.cos(lat1) * np.cos(lat2) * np.sin((lon2 - lon1) / 2.0) ** 2

    return earth_radius_nm * 2 * np.arcsin(np.sqrt(a))

# coordinates for ports from the ShipXplorer website
deput_coords = (54.506, 11.232)
dkrod_coords = (54.651, 11.348)

# extracting latitude and longitude for both ports
deput_lat, deput_lon = deput_coords
dkrod_lat, dkrod_lon = dkrod_coords

# distance between DEPUT and DKROD
distance_nm = haversine(deput_lat, deput_lon, dkrod_lat, dkrod_lon)
distance_nm
```

Out[76]: 9.595993037899353

In [77]:
```
ship1['Distance'] = haversine(ship1['Latitude'].shift(), ship1['Longitude'].shift()
ship1
```

Out[77]:

| | index | Timestamp | MMSI | Latitude | Longitude | ROT | SOG | COG | Heading |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 222 | 2024-11-01 00:03:51 | 211188000 | 54.584183 | 11.276583 | -1.1 | 16.1 | 202.8 | 212.0 |
| **1** | 232 | 2024-11-01 00:03:56 | 211188000 | 54.583767 | 11.276283 | 0.0 | 16.1 | 202.8 | 212.0 |
| **2** | 242 | 2024-11-01 00:04:02 | 211188000 | 54.583367 | 11.275983 | 1.1 | 16.1 | 202.8 | 212.0 |
| **3** | 251 | 2024-11-01 00:04:09 | 211188000 | 54.582950 | 11.275683 | 0.0 | 16.1 | 202.9 | 212.0 |
| **4** | 259 | 2024-11-01 00:04:14 | 211188000 | 54.582533 | 11.275400 | -2.9 | 16.1 | 202.7 | 212.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **45224** | 354782 | 2024-11-04 23:59:23 | 211188000 | 54.607783 | 11.294517 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45225** | 354788 | 2024-11-04 23:59:27 | 211188000 | 54.607783 | 11.294517 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45226** | 354794 | 2024-11-04 23:59:34 | 211188000 | 54.607150 | 11.294017 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45227** | 354799 | 2024-11-04 23:59:42 | 211188000 | 54.606650 | 11.293600 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45228** | 354809 | 2024-11-04 23:59:53 | 211188000 | 54.606033 | 11.293117 | 0.0 | 13.4 | 204.8 | 205.0 |

45229 rows × 12 columns

In [78]:
```
ship1['Distance'].sum()
```

Out[78]: 978.5696977483747

In [79]:
```
def is_near_port(lat, lon, port_lat, port_lon, threshold_nm=0.5):
    dist = haversine(lat, lon, port_lat, port_lon)
    return dist <= threshold_nm
```

In [80]:
```
ship1['Near_DEPUT'] = ship1.apply(lambda row: is_near_port(row['Latitude'], row['Lo

ship1['Near_DKROD'] = ship1.apply(lambda row: is_near_port(row['Latitude'], row['Lo
```

In [81]:
```
depart_idx = ship1[ship1['Near_DEPUT']].index.max()
arrive_idx = ship1[(ship1.index > depart_idx) & (ship1['Near_DKROD'])].index.min()
single_journey = ship1.loc[depart_idx:arrive_idx]
```

In [82]:
```
single_journey
```

Out[82]:

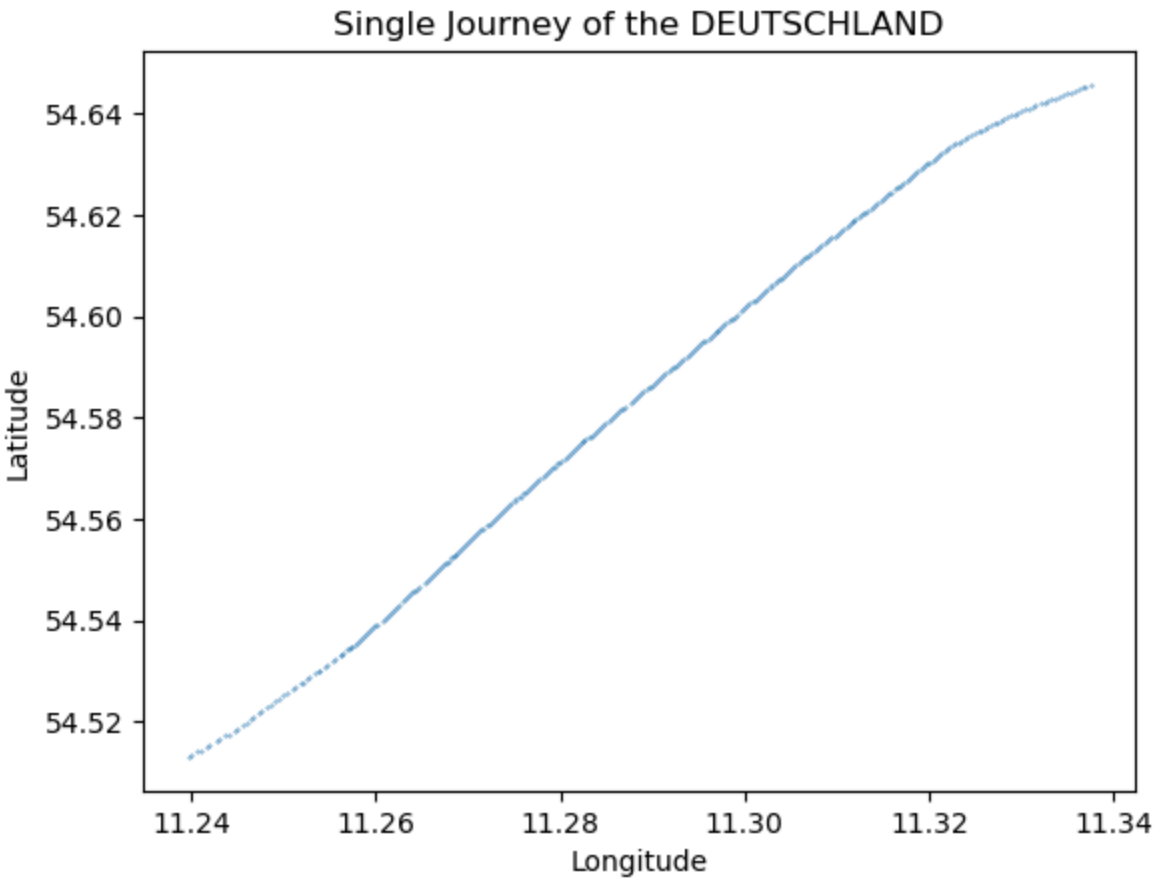
| | index | Timestamp | MMSI | Latitude | Longitude | ROT | SOG | COG | Heading |
|---|---|---|---|---|---|---|---|---|---|
| **44646** | 350810 | 2024-11-04 22:48:00 | 211188000 | 54.513033 | 11.239667 | -2.9 | 12.2 | 30.4 | 30.0 |
| **44647** | 350814 | 2024-11-04 22:48:07 | 211188000 | 54.513383 | 11.240000 | -2.2 | 12.3 | 30.3 | 30.0 |
| **44648** | 350829 | 2024-11-04 22:48:20 | 211188000 | 54.514067 | 11.240683 | 0.0 | 12.6 | 29.8 | 30.0 |
| **44649** | 350838 | 2024-11-04 22:48:27 | 211188000 | 54.514383 | 11.240983 | 0.0 | 12.7 | 29.9 | 30.0 |
| **44650** | 350851 | 2024-11-04 22:48:40 | 211188000 | 54.515050 | 11.241650 | 0.0 | 12.9 | 29.8 | 30.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **44993** | 353057 | 2024-11-04 23:24:58 | 211188000 | 54.644450 | 11.335750 | 2.9 | 13.5 | 39.1 | 43.0 |
| **44994** | 353063 | 2024-11-04 23:25:05 | 211188000 | 54.644783 | 11.336233 | 5.4 | 13.5 | 39.6 | 43.0 |
| **44995** | 353069 | 2024-11-04 23:25:11 | 211188000 | 54.645083 | 11.336650 | 6.4 | 13.5 | 40.1 | 44.0 |
| **44996** | 353074 | 2024-11-04 23:25:16 | 211188000 | 54.645317 | 11.337000 | 7.5 | 13.6 | 41.1 | 44.0 |
| **44997** | 353080 | 2024-11-04 23:25:23 | 211188000 | 54.645650 | 11.337500 | 2.9 | 13.6 | 40.4 | 45.0 |

352 rows × 14 columns

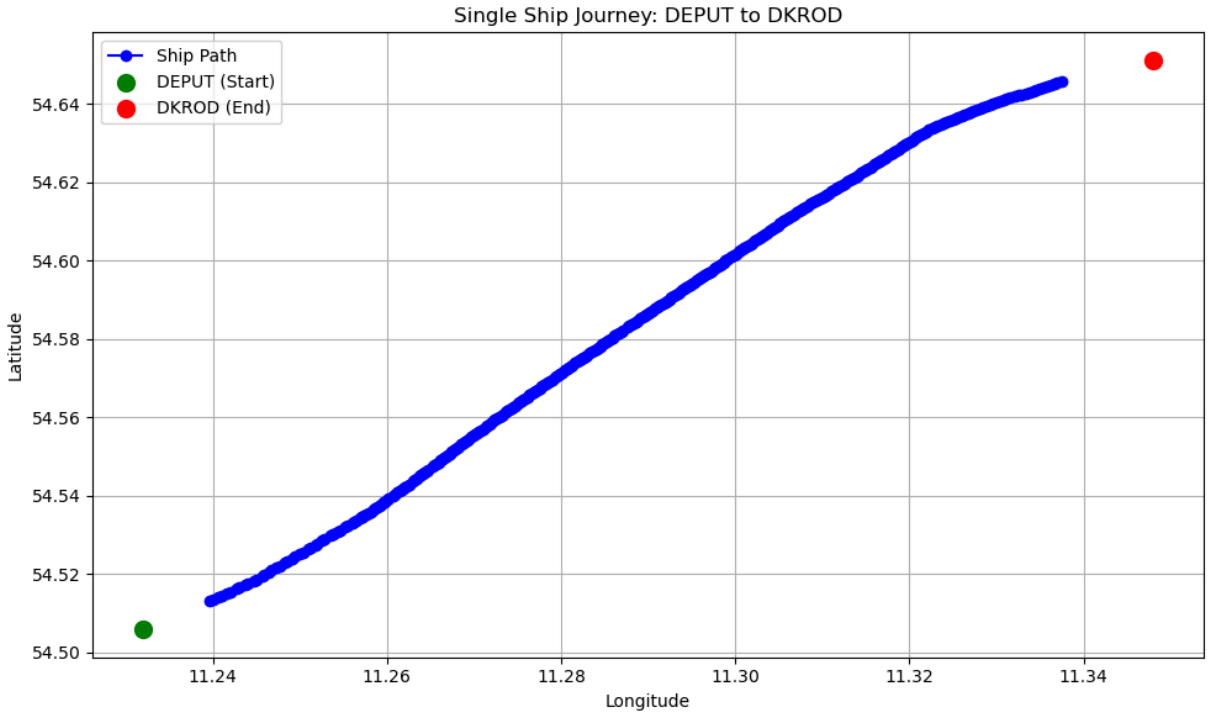
In [83]:
```
single_journey['Distance'].sum()
```

Out[83]: 8.73654354369189

In [84]:
```
single_journeyloc = single_journey.plot.scatter(x='Longitude', y='Latitude', s=0.1,
```




In [85]:
```
plt.figure(figsize=(10, 6))

plt.plot(single_journey['Longitude'], single_journey['Latitude'], marker='o', lines

plt.scatter(deput_coords[1], deput_coords[0], color='green', s=100, label='DEPUT (S
plt.scatter(dkrod_coords[1], dkrod_coords[0], color='red', s=100, label='DKROD (End

plt.title("Single Ship Journey: DEPUT to DKROD")
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.legend()
plt.grid(True)
plt.tight_layout()
```



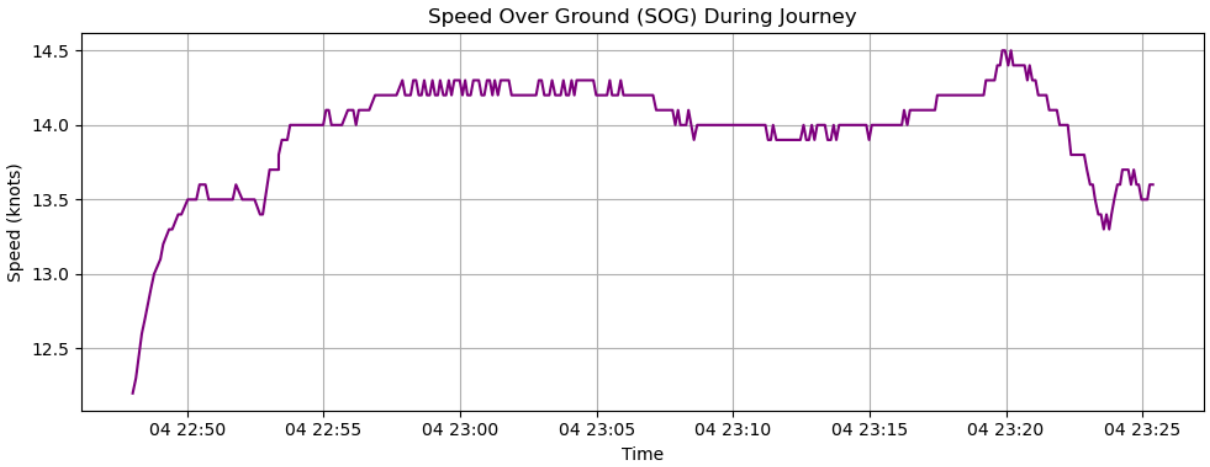

In [86]:
```
total_distance = single_journey['Distance'].sum()
print(f"Total distance traveled: {total_distance:.2f} nautical miles")
```

```
Total distance traveled: 8.74 nautical miles
```

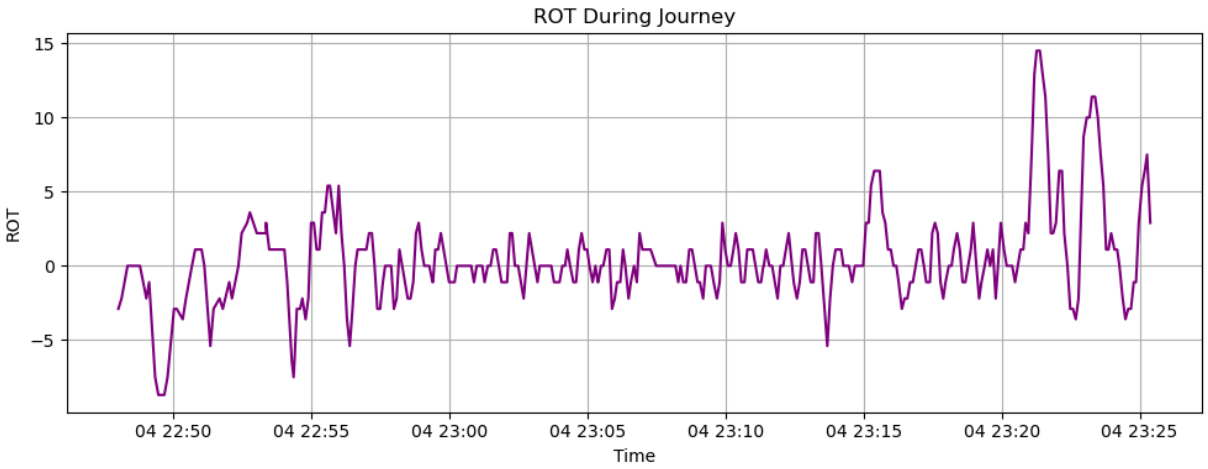
In [87]:
```
duration = single_journey['Timestamp'].iloc[-1] - single_journey['Timestamp'].iloc[
print(f"Journey duration: {duration}")
```

```
Journey duration: 0 days 00:37:23
```

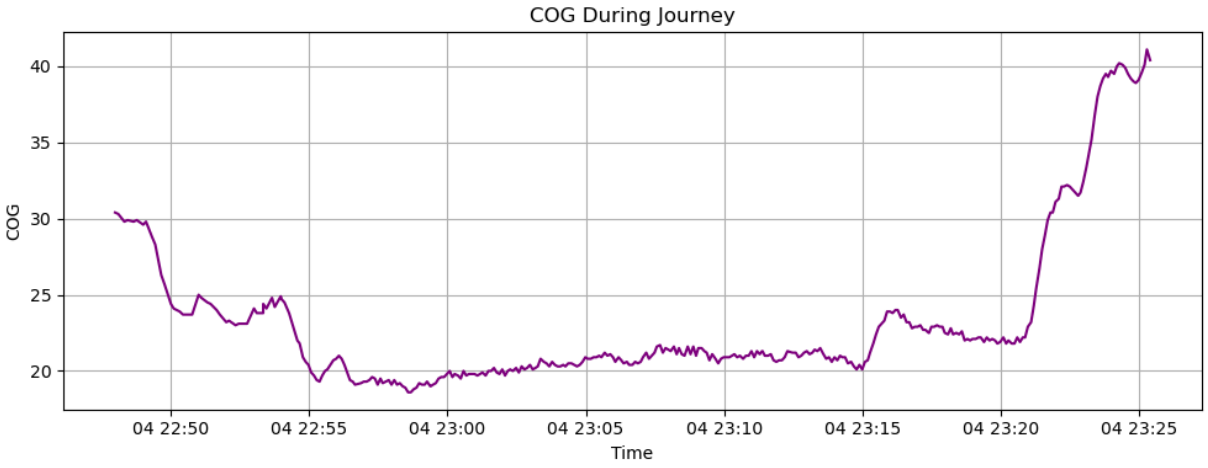
In [88]:
```
plt.figure(figsize=(10, 4))
plt.plot(single_journey['Timestamp'], single_journey['SOG'], color='purple')
plt.title("Speed Over Ground (SOG) During Journey")
plt.xlabel("Time")
plt.ylabel("Speed (knots)")
plt.grid(True)
plt.tight_layout()
```



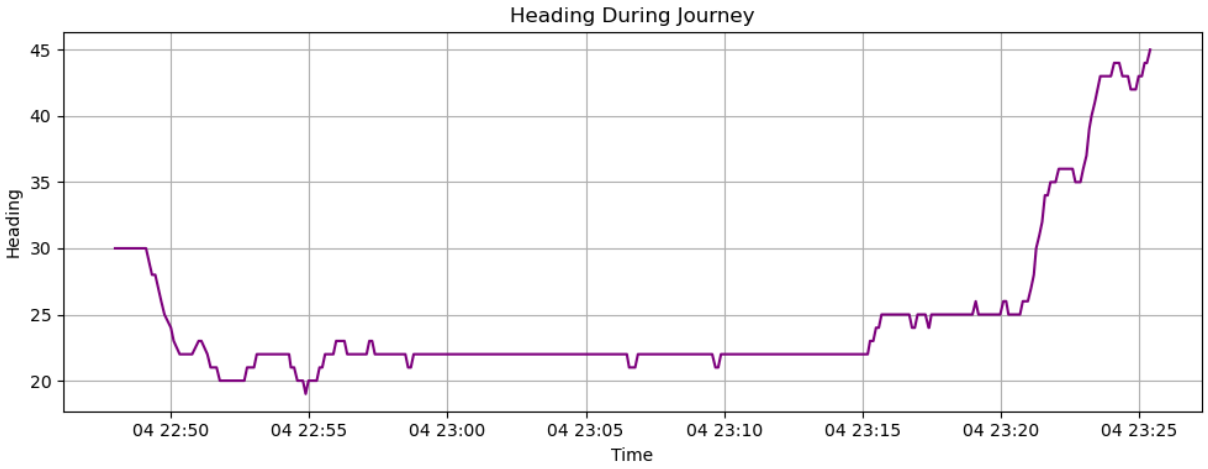
In [89]:
```
plt.figure(figsize=(10, 4))
plt.plot(single_journey['Timestamp'], single_journey['ROT'], color='purple')
plt.title("ROT During Journey")
plt.xlabel("Time")
plt.ylabel("ROT")
plt.grid(True)
plt.tight_layout()
```




In [90]:
```
plt.figure(figsize=(10, 4))
plt.plot(single_journey['Timestamp'], single_journey['COG'], color='purple')
plt.title("COG During Journey")
plt.xlabel("Time")
plt.ylabel("COG")
plt.grid(True)
plt.tight_layout()
```




In [91]:
```
plt.figure(figsize=(10, 4))
plt.plot(single_journey['Timestamp'], single_journey['Heading'], color='purple')
plt.title("Heading During Journey")
plt.xlabel("Time")
plt.ylabel("Heading")
plt.grid(True)
plt.tight_layout()
```

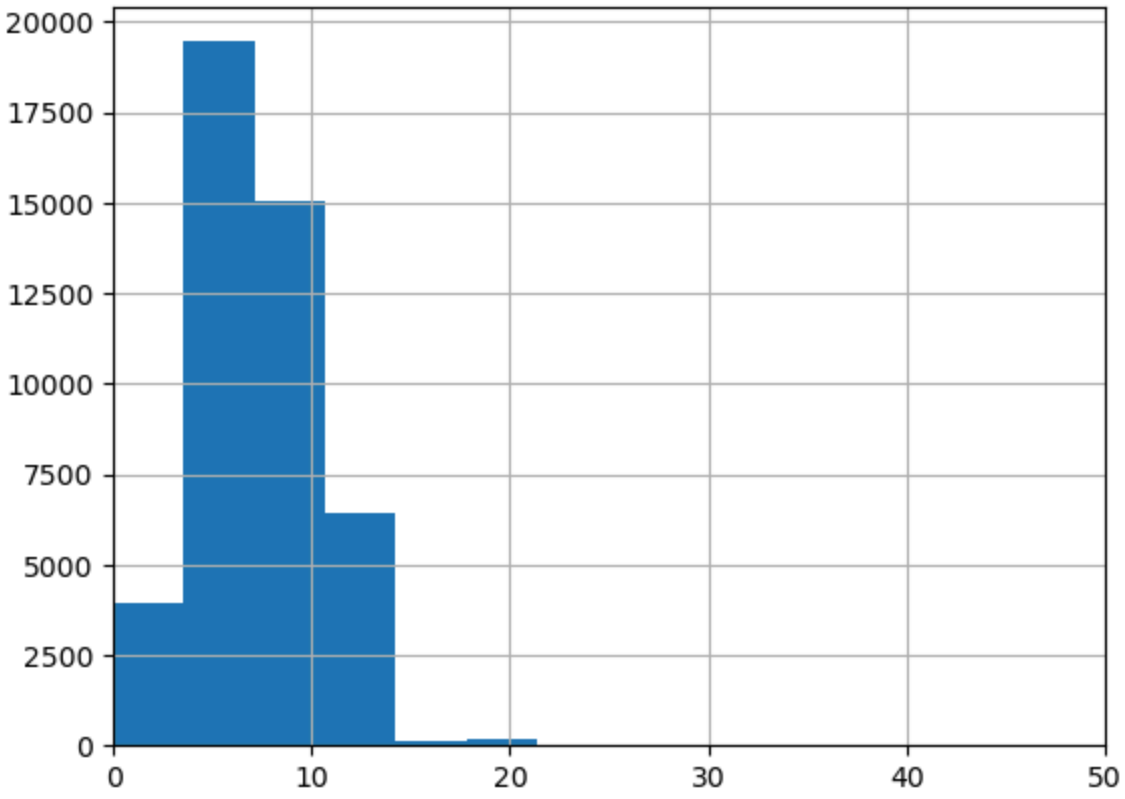



In [92]:
```
ship1['time_diff'] = ship1['Timestamp'].diff().dt.total_seconds()
ship1['time_diff'].head()
```

Out[92]:
```
0    NaN
1    5.0
2    6.0
3    7.0
4    5.0
Name: time_diff, dtype: float64
```

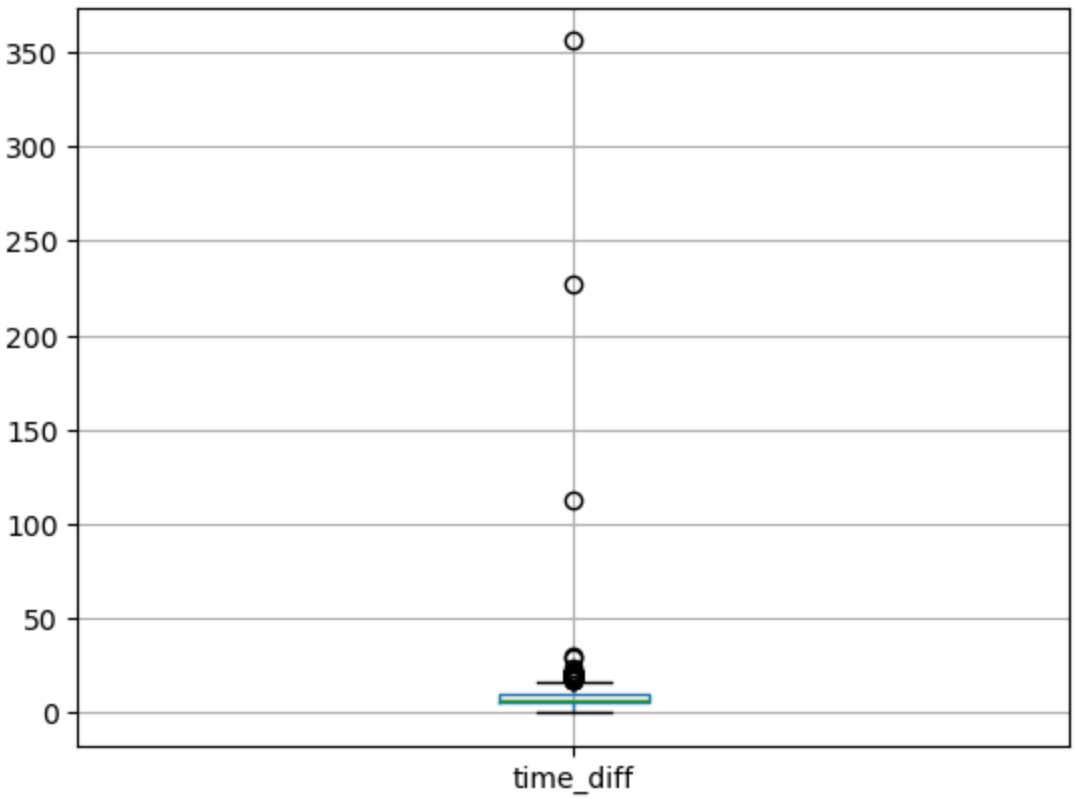
In [93]:
```
ship1['time_diff'].hist(bins=100)
plt.xlim(0, 50)
```

Out[93]:
```
(0.0, 50.0)
```




In [94]:
```
ship1.boxplot(column='time_diff')
```

Out[94]: <Axes: >




In [95]:
```
large_time_diff_rows = ship1[ship1['time_diff'] > 50]
large_time_diff_rows
```

Out[95]:

| | index | Timestamp | MMSI | Latitude | Longitude | ROT | SOG | COG | Heading |
|---|---|---|---|---|---|---|---|---|---|
| **11549** | 87603 | 2024-11-02 00:05:52 | 211188000 | 54.617750 | 11.305667 | -2.9 | 13.5 | 212.4 | 215.0 |
| **22842** | 174585 | 2024-11-03 00:01:42 | 211188000 | 54.645950 | 11.336400 | 6.4 | 12.2 | 230.6 | 229.0 |
| **34055** | 267312 | 2024-11-04 00:03:37 | 211188000 | 54.638333 | 11.320033 | -16.1 | 13.0 | 227.2 | 227.0 |

In [96]:
```
ship1['time_diff'].describe()
```

Out[96]:
```
count    45228.000000
mean         7.636022
std          3.535042
min          0.000000
25%          6.000000
50%          7.000000
75%         10.000000
max        356.000000
Name: time_diff, dtype: float64
```

In [97]:
```
ship1[ship1['Near_DEPUT']]
```

Out[97]:

| | index | Timestamp | MMSI | Latitude | Longitude | ROT | SOG | COG | Heading |
|---|---|---|---|---|---|---|---|---|---|
| **174** | 1230 | 2024-11-01 00:21:21 | 211188000 | 54.514233 | 11.233917 | -6.4 | 12.9 | 185.9 | 190.0 |
| **175** | 1239 | 2024-11-01 00:21:32 | 211188000 | 54.513583 | 11.233783 | 3.6 | 12.9 | 186.2 | 190.0 |
| **176** | 1244 | 2024-11-01 00:21:39 | 211188000 | 54.513167 | 11.233717 | 2.9 | 12.9 | 186.1 | 190.0 |
| **177** | 1253 | 2024-11-01 00:21:45 | 211188000 | 54.512817 | 11.233633 | -2.9 | 12.9 | 186.5 | 190.0 |
| **178** | 1256 | 2024-11-01 00:21:47 | 211188000 | 54.512700 | 11.233617 | -2.9 | 12.9 | 186.1 | 190.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **44642** | 350770 | 2024-11-04 22:47:21 | 211188000 | 54.511217 | 11.237700 | -7.5 | 11.4 | 35.5 | 36.0 |
| **44643** | 350779 | 2024-11-04 22:47:27 | 211188000 | 54.511483 | 11.238000 | -10.0 | 11.6 | 34.2 | 35.0 |
| **44644** | 350790 | 2024-11-04 22:47:40 | 211188000 | 54.512083 | 11.238667 | -10.0 | 11.9 | 32.4 | 32.0 |
| **44645** | 350798 | 2024-11-04 22:47:47 | 211188000 | 54.512400 | 11.239017 | -7.5 | 12.0 | 31.6 | 31.0 |
| **44646** | 350810 | 2024-11-04 22:48:00 | 211188000 | 54.513033 | 11.239667 | -2.9 | 12.2 | 30.4 | 30.0 |

8526 rows × 15 columns

In [98]:
```
ship1
```

Out[98]:

| | index | Timestamp | MMSI | Latitude | Longitude | ROT | SOG | COG | Heading |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 222 | 2024-11-01 00:03:51 | 211188000 | 54.584183 | 11.276583 | -1.1 | 16.1 | 202.8 | 212.0 |
| **1** | 232 | 2024-11-01 00:03:56 | 211188000 | 54.583767 | 11.276283 | 0.0 | 16.1 | 202.8 | 212.0 |
| **2** | 242 | 2024-11-01 00:04:02 | 211188000 | 54.583367 | 11.275983 | 1.1 | 16.1 | 202.8 | 212.0 |
| **3** | 251 | 2024-11-01 00:04:09 | 211188000 | 54.582950 | 11.275683 | 0.0 | 16.1 | 202.9 | 212.0 |
| **4** | 259 | 2024-11-01 00:04:14 | 211188000 | 54.582533 | 11.275400 | -2.9 | 16.1 | 202.7 | 212.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **45224** | 354782 | 2024-11-04 23:59:23 | 211188000 | 54.607783 | 11.294517 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45225** | 354788 | 2024-11-04 23:59:27 | 211188000 | 54.607783 | 11.294517 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45226** | 354794 | 2024-11-04 23:59:34 | 211188000 | 54.607150 | 11.294017 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45227** | 354799 | 2024-11-04 23:59:42 | 211188000 | 54.606650 | 11.293600 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45228** | 354809 | 2024-11-04 23:59:53 | 211188000 | 54.606033 | 11.293117 | 0.0 | 13.4 | 204.8 | 205.0 |

45229 rows × 15 columns

In [99]:

```
ship1['journey_id'] = np.nan
journey_id = 0
in_journey = False
current_journey_rows = []

for i in range(len(ship1)):
    near_deput = ship1.iloc[i]['Near_DEPUT']
    near_dkrod = ship1.iloc[i]['Near_DKROD']

    if not in_journey:
```

```
        if near_deput or near_dkrod:
            continue
        else:
            in_journey = True
            current_journey_rows = [i]
    else:
        if near_deput or near_dkrod:
            current_journey_rows.append(i)
            for row in current_journey_rows:
                ship1.at[row, 'journey_id'] = journey_id
            journey_id += 1
            in_journey = False
        else:
            current_journey_rows.append(i)

ship1['journey_id'] = ship1['journey_id'].ffill().bfill()
```

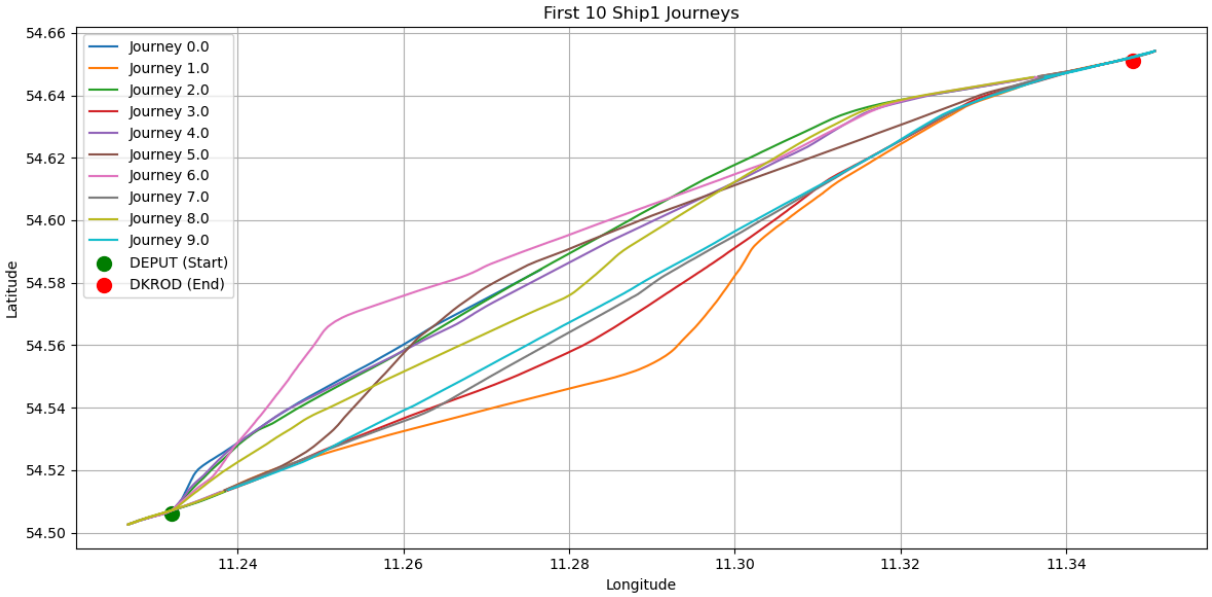
In [100…

```
print("Null journey_id count:", ship1['journey_id'].isna().sum())
```

```
Null journey_id count: 0
```

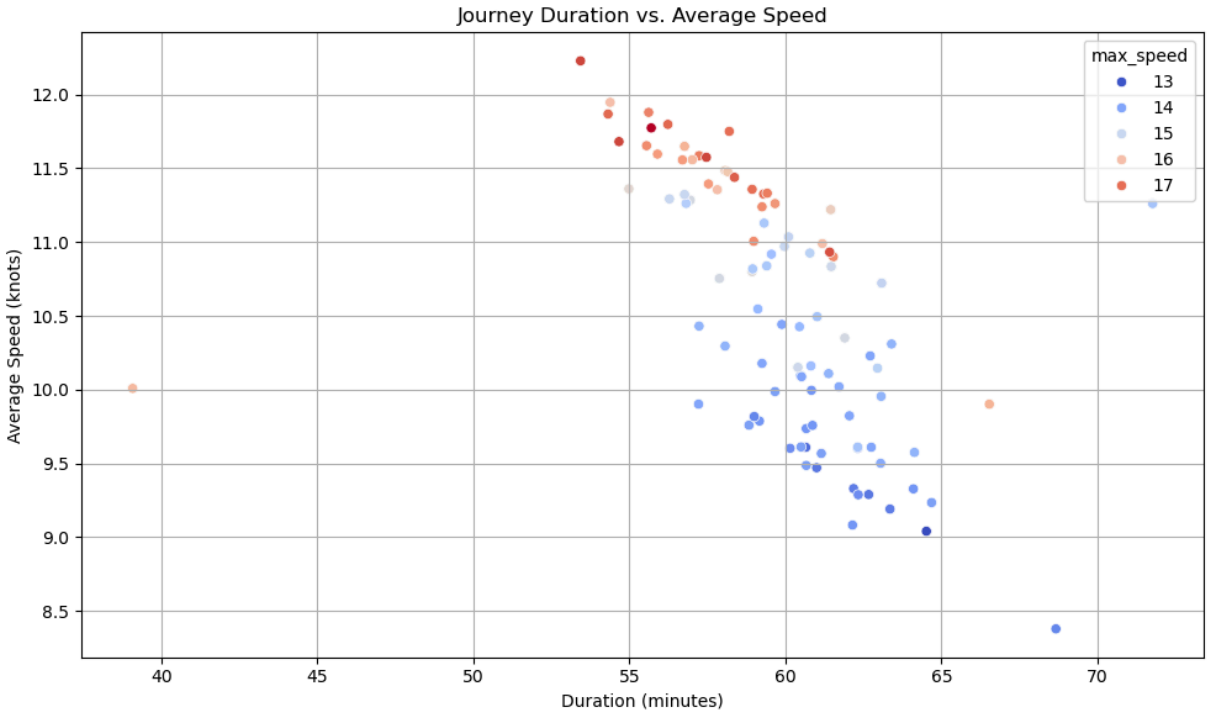
In [103…

```
first_10_journeys = ship1['journey_id'].unique()[:10]

plt.figure(figsize=(12, 6))
for journey in first_10_journeys:
    journey_data = ship1[ship1['journey_id'] == journey]
    plt.plot(journey_data['Longitude'], journey_data['Latitude'], label=f'Journey {

plt.scatter(deput_coords[1], deput_coords[0], color='green', s=100, label='DEPUT (S
plt.scatter(dkrod_coords[1], dkrod_coords[0], color='red', s=100, label='DKROD (End

plt.title('First 10 Ship1 Journeys')
plt.xlabel('Longitude')
plt.ylabel('Latitude')

plt.legend(loc='best')
plt.grid(True)
plt.tight_layout()
plt.show()
```



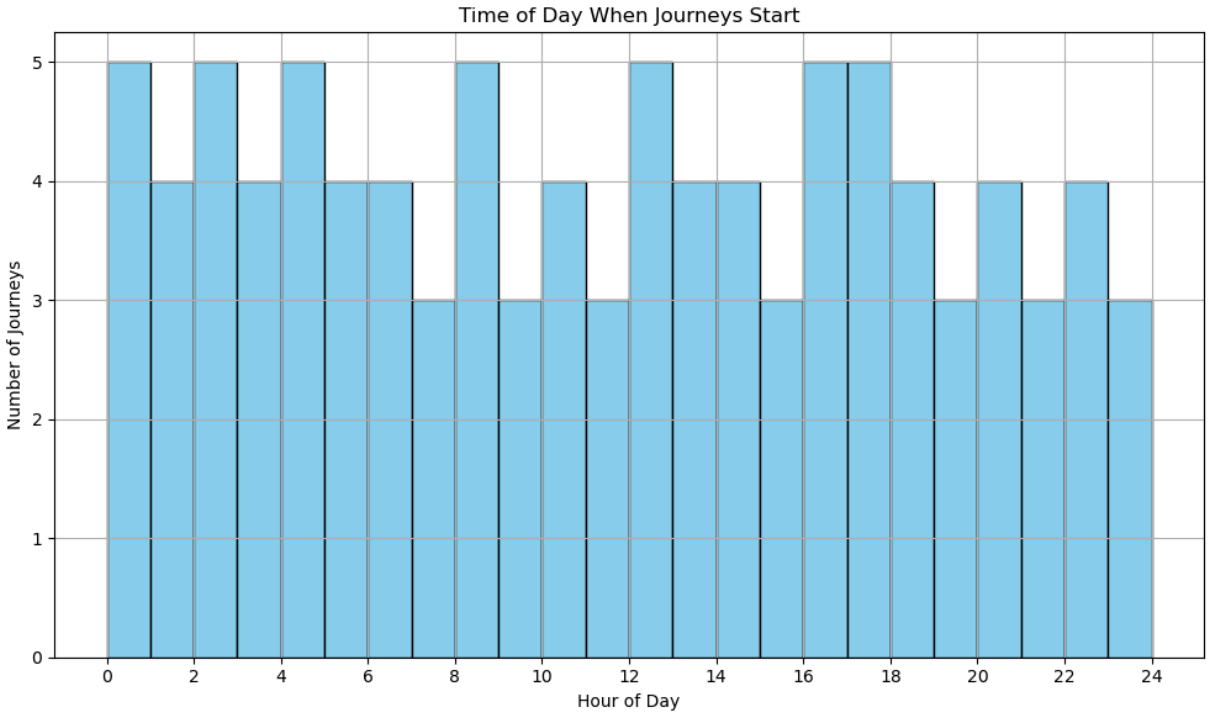

In [105…
```
journey_stats = ship1.dropna(subset=['journey_id']).groupby('journey_id').agg({
    'SOG': ['mean', 'max'],
    'Timestamp': ['min', 'max']
})
journey_stats.columns = ['avg_speed', 'max_speed', 'start_time', 'end_time']
journey_stats['duration'] = journey_stats['end_time'] - journey_stats['start_time']
```

In [109…
```
journey_stats['duration_mins'] = journey_stats['duration'].dt.total_seconds() / 60

plt.figure(figsize=(10, 6))
sns.scatterplot(data=journey_stats, x='duration_mins', y='avg_speed', hue='max_spee
plt.title("Journey Duration vs. Average Speed")
plt.xlabel("Duration (minutes)")
plt.ylabel("Average Speed (knots)")
plt.grid(True)
plt.tight_layout()
plt.show()
```




In [111…

```
ship1['Timestamp'] = pd.to_datetime(ship1['Timestamp'])

journey_starts = ship1.groupby('journey_id')['Timestamp'].first()

journey_times = journey_starts.dt.time
journey_hours = journey_starts.dt.hour + journey_starts.dt.minute / 60

plt.figure(figsize=(10, 6))
plt.hist(journey_hours, bins=24, range=(0, 24), color='skyblue', edgecolor='black')
plt.title('Time of Day When Journeys Start')
plt.xlabel('Hour of Day')
plt.ylabel('Number of Journeys')
plt.xticks(range(0, 25, 2))
plt.grid(True)
plt.tight_layout()
plt.show()
```




In [115…
```
# Journey duration
ship1['journey_start_time'] = ship1.groupby('journey_id')['Timestamp'].transform('m
ship1['journey_duration'] = (ship1['Timestamp'] - ship1['journey_start_time']).dt.t
```

In [117…
```
ship1
```

Out[117…

| | index | Timestamp | MMSI | Latitude | Longitude | ROT | SOG | COG | Heading |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 222 | 2024-11-01 00:03:51 | 211188000 | 54.584183 | 11.276583 | -1.1 | 16.1 | 202.8 | 212.0 |
| **1** | 232 | 2024-11-01 00:03:56 | 211188000 | 54.583767 | 11.276283 | 0.0 | 16.1 | 202.8 | 212.0 |
| **2** | 242 | 2024-11-01 00:04:02 | 211188000 | 54.583367 | 11.275983 | 1.1 | 16.1 | 202.8 | 212.0 |
| **3** | 251 | 2024-11-01 00:04:09 | 211188000 | 54.582950 | 11.275683 | 0.0 | 16.1 | 202.9 | 212.0 |
| **4** | 259 | 2024-11-01 00:04:14 | 211188000 | 54.582533 | 11.275400 | -2.9 | 16.1 | 202.7 | 212.0 |
| **…** | … | … | … | … | … | … | … | … | … |
| **45224** | 354782 | 2024-11-04 23:59:23 | 211188000 | 54.607783 | 11.294517 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45225** | 354788 | 2024-11-04 23:59:27 | 211188000 | 54.607783 | 11.294517 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45226** | 354794 | 2024-11-04 23:59:34 | 211188000 | 54.607150 | 11.294017 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45227** | 354799 | 2024-11-04 23:59:42 | 211188000 | 54.606650 | 11.293600 | 0.0 | 13.4 | 204.9 | 205.0 |
| **45228** | 354809 | 2024-11-04 23:59:53 | 211188000 | 54.606033 | 11.293117 | 0.0 | 13.4 | 204.8 | 205.0 |

45229 rows × 18 columns

In [141…
```
'''ship1['SOG_rolling'] = ship1['SOG'].rolling(window=5, min_periods=1).mean()
ship1['ROT_rolling'] = ship1['ROT'].rolling(window=5, min_periods=1).mean()
```

In [121…
```
'''ship1['SOG_diff'] = ship1['SOG'].diff().fillna(0)
```

In [123…
```
'''ship1['journey_progress'] = ship1.groupby('journey_id').cumcount() / ship1.group
```

In [139…
```
'''# remaining time
ship1['max_time'] = ship1.groupby('journey_id')['Timestamp'].transform('max')
```

```
ship1['remaining_time'] = (ship1['max_time'] - ship1['Timestamp']).dt.total_seconds
```

In [127…
```
'''model_data = ship1.dropna(subset=['journey_id', 'remaining_time'])
```

In [167…
```
'''features = ['SOG', 'ROT', 'Heading', 'COG', 'SOG_rolling', 'ROT_rolling', 'SOG_d

X = model_data[features]
y = model_data['remaining_time']
```

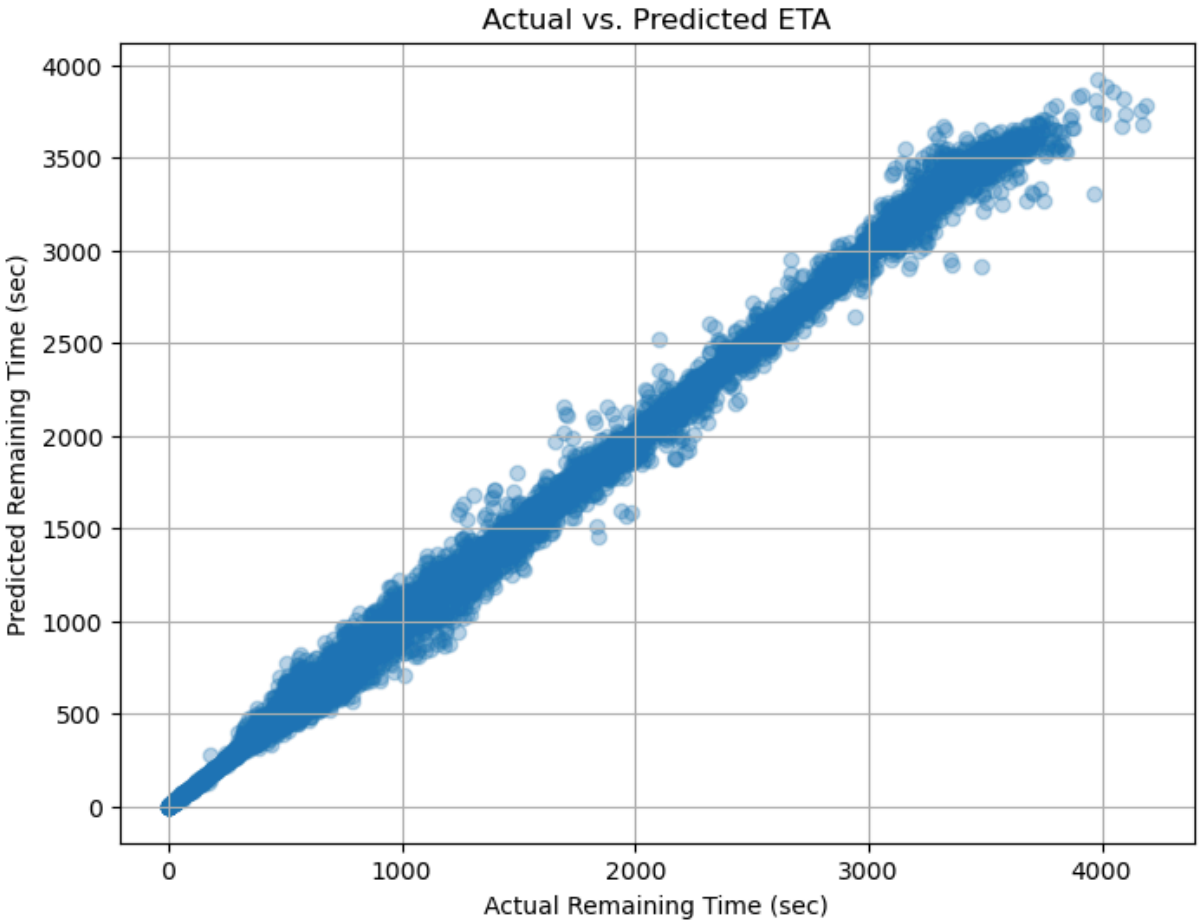
In [169…
```
'''from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

In [171…
```
'''from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor(n_estimators=100, random_state=25)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

In [134…
```
'''from sklearn.metrics import mean_absolute_error, r2_score, root_mean_squared_err

rmse = root_mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"RMSE: {rmse:.2f} sec")
print(f"MAE: {mae:.2f} sec")
print(f"R² Score: {r2:.2f}")
```

```
RMSE: 70.13 sec
MAE: 45.07 sec
R² Score: 1.00
```

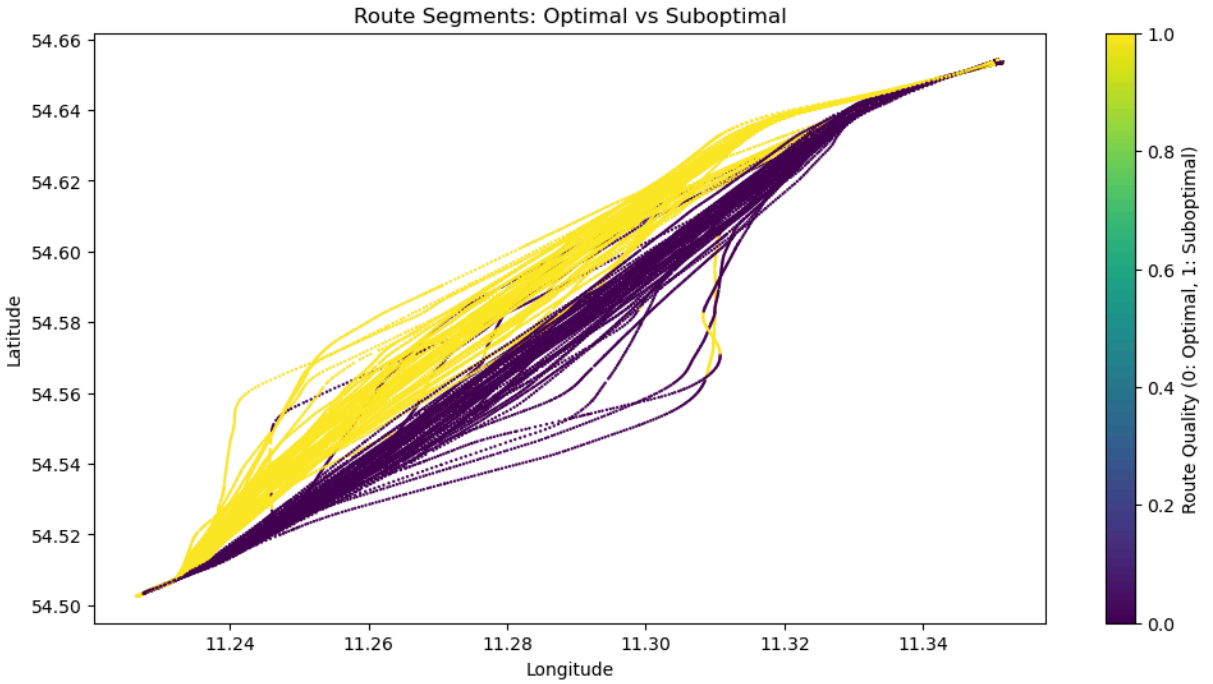
In [135…
```
'''plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, alpha=0.3)
plt.xlabel('Actual Remaining Time (sec)')
plt.ylabel('Predicted Remaining Time (sec)')
plt.title('Actual vs. Predicted ETA')
plt.grid(True)
plt.show()
```




In [209…
```
# creating smoothed values for SOG, COG, Heading, ROT to capture overall trend
# SOG difference for seeing sudden changes
ship1['SOG_smoothed'] = ship1['SOG'].rolling(window=5, min_periods=1).mean()
ship1['COG_smoothed'] = ship1['COG'].rolling(window=5, min_periods=1).mean()
ship1['SOG_diff'] = ship1['SOG'].diff().fillna(0)

# journey progress
journey_sizes = ship1['journey_id'].value_counts()
# each row's position in the journey
ship1['point_index'] = ship1.groupby('journey_id').cumcount()
# mapping each row to the total size of its journey
ship1['journey_size'] = ship1['journey_id'].map(journey_sizes)
ship1['journey_progress'] = ship1['point_index'] / (ship1['journey_size'] - 1)

ship1['Heading_smoothed'] = ship1['Heading'].rolling(window=5, min_periods=1).mean(
ship1['ROT_smoothed'] = ship1['ROT'].rolling(window=5, min_periods=1).mean()
```

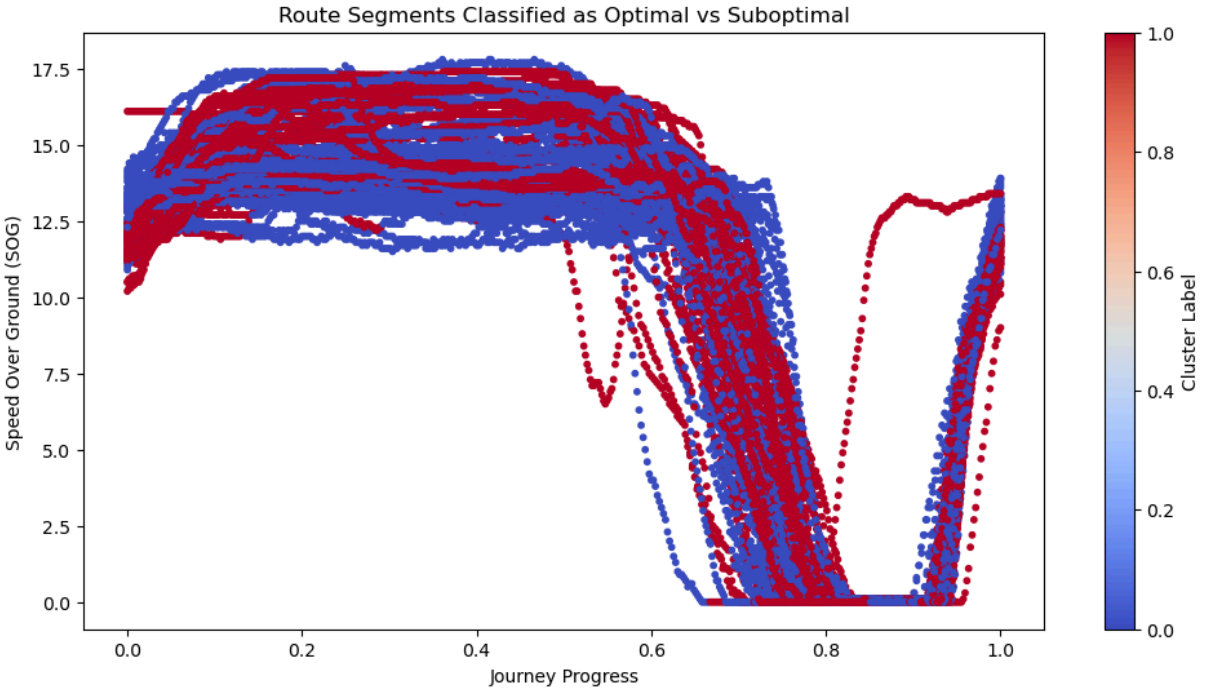
In [215…
```
from sklearn.cluster import KMeans

X = ship1[['SOG_smoothed', 'COG_smoothed', 'SOG_diff', 'journey_progress', 'Heading

# KMeans clustering to classify as optimal or suboptimal
kmeans = KMeans(n_clusters=2, random_state=42)
ship1['route_quality'] = kmeans.fit_predict(X)

plt.figure(figsize=(12, 6))
plt.scatter(ship1['Longitude'], ship1['Latitude'], c=ship1['route_quality'], cmap='
```

```
plt.title('Route Segments: Optimal vs Suboptimal')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.colorbar(label='Route Quality (0: Optimal, 1: Suboptimal)')
plt.show()
```




In [203…

```
plt.figure(figsize=(12, 6))
plt.scatter(ship1['journey_progress'], ship1['SOG'], c=ship1['route_quality'], cmap
plt.xlabel('Journey Progress')
plt.ylabel('Speed Over Ground (SOG)')
plt.title('Route Segments Classified as Optimal vs Suboptimal')
plt.colorbar(label='Cluster Label')
plt.show()
```



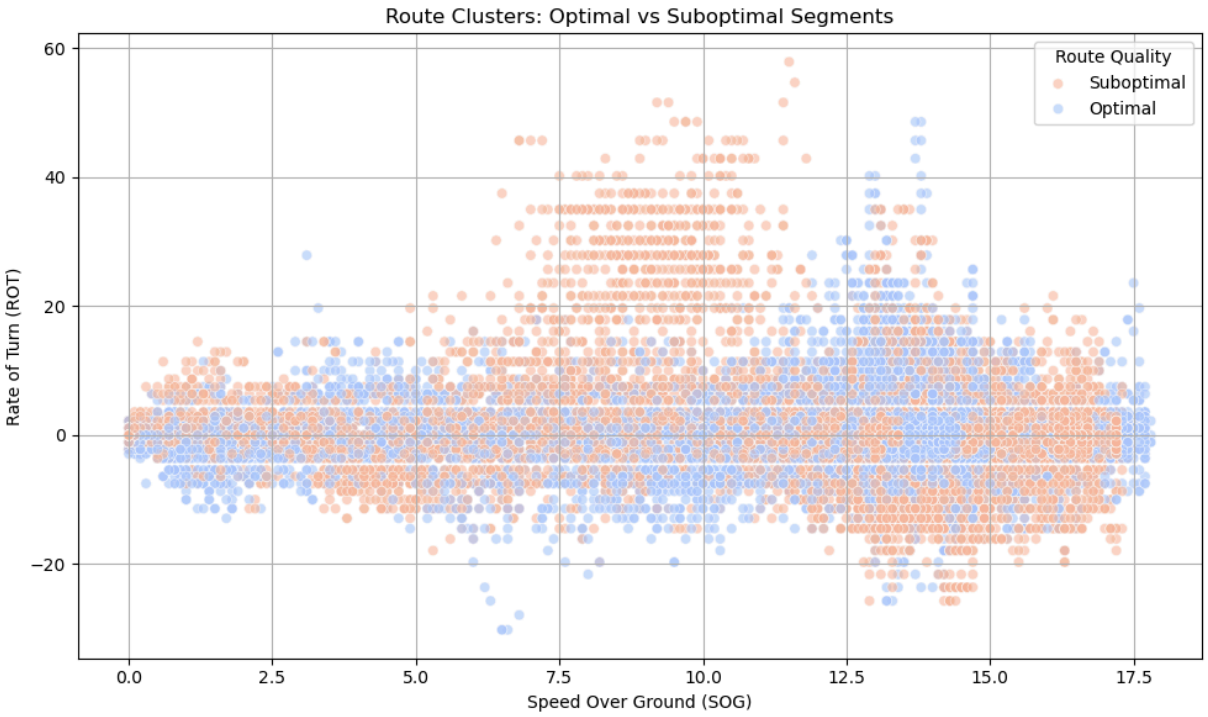
In [205…
```
cluster_summary = ship1.groupby('route_quality')[['SOG', 'SOG_diff', 'ROT', 'journe
print(cluster_summary)
```

```
                     SOG  SOG_diff       ROT  journey_progress
route_quality
0              10.985189 -0.000144  0.253066          0.486221
1              10.327005  0.000012  0.349283          0.511842
```

In [207…
```
from sklearn.metrics import silhouette_score

score = silhouette_score(X, ship1['route_quality'])
print(f'Silhouette Score: {score:.2f}')
```

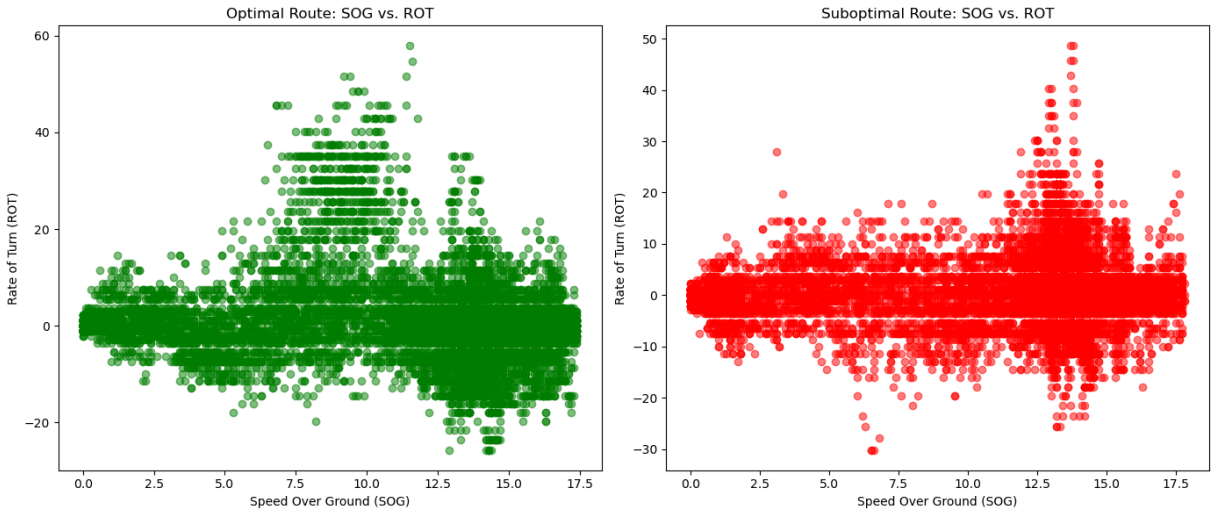
```
Silhouette Score: 0.81
```

In [179…
```
features_to_plot = ['SOG', 'ROT']

plt.figure(figsize=(10, 6))
sns.scatterplot(data=ship1, x=features_to_plot[0], y=features_to_plot[1], hue='rout

plt.title('Route Clusters: Optimal vs Suboptimal Segments')
plt.xlabel('Speed Over Ground (SOG)')
plt.ylabel('Rate of Turn (ROT)')
plt.legend(title='Route Quality', labels=['Suboptimal', 'Optimal'])
plt.grid(True)
plt.tight_layout()
plt.show()
```




In [189…
```
optimal_data = ship1[ship1['route_quality'] == 1]
suboptimal_data = ship1[ship1['route_quality'] == 0]

fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# optimal
```

```
axes[0].scatter(optimal_data['SOG'], optimal_data['ROT'], c='green', alpha=0.5)
axes[0].set_title('Optimal Route: SOG vs. ROT')
axes[0].set_xlabel('Speed Over Ground (SOG)')
axes[0].set_ylabel('Rate of Turn (ROT)')

# suboptimal
axes[1].scatter(suboptimal_data['SOG'], suboptimal_data['ROT'], c='red', alpha=0.5)
axes[1].set_title('Suboptimal Route: SOG vs. ROT')
axes[1].set_xlabel('Speed Over Ground (SOG)')
axes[1].set_ylabel('Rate of Turn (ROT)')

plt.tight_layout()
plt.show()
```




In [ ]: