In [1]:

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

import scipy.stats as stats

def t_test(df1, df2, target="SOG"):
    t_stat, p_value = stats.ttest_ind(df1[target].dropna(), df2[target].dropna())
    print(f"T-statistic: {t_stat:.3f}, p-value: {p_value:.15f}")

    # If p-value is less than 0.05, the difference is statistically significant
    if p_value < 0.05:
        print("The difference is statistically significant, keeping data split.")
    else:
        print("The difference is not statistically significant, keeping data combined.")
```

In [2]:

```
df = pd.read_csv("all_data.csv")
```

In [3]:

```
data_9juin = df[df["ISODateTimeUTC"].str.startswith("2025-06-09")]
```

T test on the TWS between runs 5,6,7 where Karl holds the weights and runs 8,9,10,11 where Gian holds 6kgs

In [4]:

```
first_runs = ["09_06_Run5","09_06_Run6","09_06_Run7"]
data_9juin_first_runs = data_9juin[data_9juin["run"].isin(first_runs) ]
```

In [5]:

```
last_runs = ["09_06_Run8","09_06_Run9","09_06_Run10","09_06_Run11"]
data_9juin_last_runs = data_9juin[data_9juin["run"].isin(last_runs) ]
```

In [6]:

```
t_test(data_9juin_first_runs,data_9juin_last_runs, target="TWS")
print(data_9juin_first_runs["TWS"].mean(),data_9juin_last_runs["TWS"].mean())
print(f"Average TWS in Group 1: {data_9juin_first_runs['TWS'].mean()}")
print(f"Average TWS in Group 2: {data_9juin_last_runs['TWS'].mean()}")
```

```
T-statistic: -20.004, p-value: 0.000000000000000
The difference is statistically significant, keeping data split.
7.2590247128437175 7.541569498486814
Average TWS in Group 1: 7.2590247128437175
Average TWS in Group 2: 7.541569498486814
```

t test karl heavy vs karl not heavy

In [7]:

```
only_karl_first_runs_heavy = data_9juin_first_runs[
    (data_9juin_first_runs["boat_name"] == "Karl Maeder") |
    ((data_9juin_first_runs["boat_name"] == "Senseboard") &
     (data_9juin_first_runs["opponent_name"] == "Gian Stragiotti"))
]
only_karl_first_runs_heavy.sample(5)
```

Out[7]:

	ISODateTimeUTC	SecondsSince1970	Heel_Abs	Heel_Lwd	Lat	LatBow	LatCenter	LatStern	Leg	Line_C	...	run	interval_id	boat_name	opponent_name	boat_role	boat_weight
44954	2025-06-09T12:46:17.072Z	1.749473e+09	54.8	54.8	43.506037	43.506035	43.506041	43.506047	1.0	121.7	...	09_06_Run6	1	Karl Maeder	SenseBoard	slave	106.975
47215	2025-06-09T12:54:24.561Z	1.749474e+09	58.4	58.4	43.506571	43.506569	43.506575	43.506581	NaN	126.3	...	09_06_Run7	1	Karl Maeder	SenseBoard	master	106.975
47016	2025-06-09T12:54:04.653Z	1.749474e+09	59.1	59.1	43.508555	43.508553	43.508559	43.508566	NaN	114.5	...	09_06_Run7	1	Karl Maeder	SenseBoard	master	106.975
43653	2025-06-09T12:38:16.452Z	1.749473e+09	63.1	63.1	43.505239	43.505237	43.505243	43.505250	NaN	122.5	...	09_06_Run5	1	Karl Maeder	SenseBoard	master	106.975
47694	2025-06-09T12:55:12.452Z	1.749474e+09	47.1	47.1	43.501741	43.501739	43.501745	43.501751	NaN	121.3	...	09_06_Run7	1	Karl Maeder	SenseBoard	master	106.975

5 rows × 48 columns

In [8]:

```
only_karl_last_runs_light = data_9juin_last_runs[
    (data_9juin_last_runs["boat_name"] == "Karl Maeder") |
    ((data_9juin_last_runs["boat_name"] == "SenseBoard") &
     (data_9juin_last_runs["opponent_name"] == "Gian Stragiotti"))
]
only_karl_last_runs_light.sample(5)
```

Out[8]:

	ISODateTimeUTC	SecondsSince1970	Heel_Abs	Heel_Lwd	Lat	LatBow	LatCenter	LatStern	Leg	Line_C	...	run	interval_id	boat_name	opponent_name	boat_role	boat_weight
53020	2025-06-09T13:19:18.961Z	1.749475e+09	37.4	37.4	43.506734	43.506736	43.506730	43.506724	NaN	84.70	...	09_06_Run9	2	Karl Maeder	SenseBoard	master	100.975
50357	2025-06-09T13:08:17.953Z	1.749474e+09	42.4	42.4	43.502838	43.502840	43.502834	43.502828	NaN	87.66	...	09_06_Run8	2	Karl Maeder	SenseBoard	slave	100.975
50596	2025-06-09T13:08:41.855Z	1.749475e+09	54.2	54.2	43.505670	43.505672	43.505666	43.505660	NaN	87.30	...	09_06_Run8	2	Karl Maeder	SenseBoard	slave	100.975
54153	2025-06-09T13:25:21.854Z	1.749476e+09	57.5	57.5	43.503861	43.503859	43.503865	43.503871	1.0	121.10	...	09_06_Run10	1	Karl Maeder	SenseBoard	slave	100.975
51896	2025-06-09T13:16:41.966Z	1.749475e+09	58.5	58.5	43.502199	43.502197	43.502203	43.502209	NaN	105.80	...	09_06_Run9	1	Karl Maeder	SenseBoard	master	100.975

5 rows × 48 columns

In [9]:

```
t_test(only_karl_first_runs_heavy,only_karl_last_runs_light) #general

print("\nUpwind and downwind for Karl:")
print(f"\nWeight of Karl on the first runs: {only_karl_first_runs_heavy['boat_weight'].mean()}, average SOG: {only_karl_first_runs_heavy['SOG'].mean()}")
print(f"Weight of Karl on the last runs: {only_karl_last_runs_light['boat_weight'].mean()}, average SOG: {only_karl_last_runs_light['SOG'].mean()}")
```

```
T-statistic: -7.879, p-value: 0.000000000000004
The difference is statistically significant, keeping data split.

Upwind and downwind for Karl:

Weight of Karl on the first runs: 106.97500000000001, average SOG: 23.94727146332986
Weight of Karl on the last runs: 100.975, average SOG: 24.320151187904965
```

In [10]:

```
only_karl_first_runs_heavy_upwind = only_karl_first_runs_heavy[only_karl_first_runs_heavy["TWA"]>0]
only_karl_last_runs_light_upwind = only_karl_last_runs_light[only_karl_last_runs_light["TWA"]>0]

t_test(only_karl_first_runs_heavy_upwind,only_karl_last_runs_light_upwind) #upwind

print("\nUpwind for Karl:")
print(f"Weight of Karl on the first runs: {only_karl_first_runs_heavy_upwind['boat_weight'].mean()}, average SOG: {only_karl_first_runs_heavy_upwind['SOG'].mean()}")
print(f"Weight of Karl on the last runs: {only_karl_last_runs_light_upwind['boat_weight'].mean()}, average SOG: {only_karl_last_runs_light_upwind['SOG'].mean()}")
```

```
T-statistic: -1.138, p-value: 0.255372507601836
The difference is not statistically significant, keeping data combined.

Upwind for Karl:
Weight of Karl on the first runs: 106.97499999999997, average SOG: 22.72190383681399
Weight of Karl on the last runs: 100.97500000000001, average SOG: 22.74386574074074
```

In [11]:

```
only_karl_first_runs_heavy_downwind = only_karl_first_runs_heavy[only_karl_first_runs_heavy["TWA"] <= 0]
only_karl_last_runs_light_downwind = only_karl_last_runs_light[only_karl_last_runs_light["TWA"] <= 0]
t_test(only_karl_first_runs_heavy_downwind, only_karl_last_runs_light_downwind)  # downwind

print("\nDownwind for Karl:")
print(f"Weight of Karl on the first runs: {only_karl_first_runs_heavy_downwind['boat_weight'].mean()}, average SOG: {only_karl_first_runs_heavy_downwind['SOG'].mean()}")
print(f"Weight of Karl on the last runs: {only_karl_last_runs_light_downwind['boat_weight'].mean()}, average SOG: {only_karl_last_runs_light_downwind['SOG'].mean()}")
```

```
T-statistic: 18.894, p-value: 0.000000000000000
The difference is statistically significant, keeping data split.

Downwind for Karl:
Weight of Karl on the first runs: 106.97499999999998, average SOG: 27.031662591687045
Weight of Karl on the last runs: 100.97499999999998, average SOG: 26.32492639842983
```

t test Gian heavy vs karl not heavy

In [12]:

```
only_gian_first_runs_light = data_9juin_first_runs[
    (data_9juin_first_runs["boat_name"] == "Gian Stragiotti") |
    ((data_9juin_first_runs["boat_name"] == "SenseBoard") &
     (data_9juin_first_runs["opponent_name"] == "Karl Maeder"))
]
only_gian_first_runs_light.sample(5)
```

Out[12]:

	ISODateTimeUTC	SecondsSince1970	Heel_Abs	Heel_Lwd	Lat	LatBow	LatCenter	LatStern	Leg	Line_C	...	run	interval_id	boat_name	opponent_name	boat_role	boat_weight
46650	2025-06-09T12:48:45.762Z	1.749473e+09	58.6	58.6	43.503634	43.503636	43.503630	43.503624	1.0	3.585	...	09_06_Run6	2	SenseBoard	Karl Maeder	master	109.09
48053	2025-06-09T12:54:39.454Z	1.749474e+09	59.2	59.2	43.505140	43.505138	43.505144	43.505150	1.0	6.300	...	09_06_Run7	1	SenseBoard	Karl Maeder	slave	109.09
48183	2025-06-09T12:54:52.453Z	1.749474e+09	52.5	52.5	43.503842	43.503840	43.503846	43.503852	1.0	5.775	...	09_06_Run7	1	SenseBoard	Karl Maeder	slave	109.09
46925	2025-06-09T12:49:13.256Z	1.749473e+09	49.8	49.8	43.506886	43.506887	43.506881	43.506875	1.0	5.900	...	09_06_Run6	2	SenseBoard	Karl Maeder	master	109.09
48209	2025-06-09T12:54:55.052Z	1.749474e+09	56.5	56.5	43.503584	43.503582	43.503588	43.503593	1.0	4.200	...	09_06_Run7	1	SenseBoard	Karl Maeder	slave	109.09

5 rows × 48 columns

In [13]:

```
only_gian_last_runs_heavy = data_9juin_last_runs[
    (data_9juin_last_runs["boat_name"] == "Gian Stragiotti") |
    ((data_9juin_last_runs["boat_name"] == "SenseBoard") &
     (data_9juin_last_runs["opponent_name"] == "Karl Maeder"))
]
only_gian_last_runs_heavy.sample(5)
```

Out[13]:

	ISODateTimeUTC	SecondsSince1970	Heel_Abs	Heel_Lwd	Lat	LatBow	LatCenter	LatStern	Leg	Line_C	...	run	interval_id	boat_name	opponent_name	boat_role	boat_weight
54934	2025-06-09T13:25:38.654Z	1.749476e+09	60.9	60.9	43.501833	43.501831	43.501838	43.501844	NaN	7.600	...	09_06_Run10	1	SenseBoard	Karl Maeder	master	115.09
51051	2025-06-09T13:08:40.158Z	1.749475e+09	56.2	56.2	43.505549	43.505551	43.505545	43.505539	NaN	3.100	...	09_06_Run8	2	SenseBoard	Karl Maeder	master	115.09
56908	2025-06-09T13:34:42.756Z	1.749476e+09	56.1	56.1	43.505610	43.505609	43.505615	43.505621	1.0	5.100	...	09_06_Run11	1	SenseBoard	Karl Maeder	slave	115.09
56750	2025-06-09T13:34:26.956Z	1.749476e+09	49.9	49.9	43.507254	43.507252	43.507258	43.507265	1.0	4.282	...	09_06_Run11	1	SenseBoard	Karl Maeder	slave	115.09
54768	2025-06-09T13:25:22.056Z	1.749476e+09	50.9	50.9	43.503654	43.503652	43.503658	43.503665	NaN	9.300	...	09_06_Run10	1	SenseBoard	Karl Maeder	master	115.09

5 rows × 48 columns

In [14]:

```
t_test(only_gian_first_runs_light,only_gian_last_runs_heavy) #general

print("\nUpwind and downwind for Gian:")
print(f"\nWeight of Gian on the first runs: {only_gian_first_runs_light['boat_weight'].mean()}, average SOG: {only_gian_first_runs_light['SOG'].mean()}")
print(f"Weight of Gian on the last runs: {only_gian_last_runs_heavy['boat_weight'].mean()}, average SOG: {only_gian_last_runs_heavy['SOG'].mean()}")
```

```
T-statistic: -11.650, p-value: 0.000000000000000
The difference is statistically significant, keeping data split.

Upwind and downwind for Gian:

Weight of Gian on the first runs: 109.08999999999999, average SOG: 24.236319275008714
Weight of Gian on the last runs: 115.08999999999999, average SOG: 24.825573344872346
```

In [15]:

```
only_gian_first_runs_light_upwind = only_gian_first_runs_light[only_gian_first_runs_light["TWA"]>0]
only_gian_last_runs_heavy_upwind = only_gian_last_runs_heavy[only_gian_last_runs_heavy["TWA"]>0]

t_test(only_gian_first_runs_light_upwind,only_gian_last_runs_heavy_upwind) #upwind

print("\nUpwind for Gian:")
print(f"Weight of Gian on the first runs: {only_gian_first_runs_light_upwind['boat_weight'].mean()}, average SOG: {only_gian_first_runs_light_upwind['SOG'].mean()}")
print(f"Weight of Gian on the last runs: {only_gian_last_runs_heavy_upwind['boat_weight'].mean()}, average SOG: {only_gian_last_runs_heavy_upwind['SOG'].mean()}")
```

```
T-statistic: -8.572, p-value: 0.000000000000000
The difference is statistically significant, keeping data split.

Upwind for Gian:
Weight of Gian on the first runs: 109.08999999999997, average SOG: 22.98097323600973
Weight of Gian on the last runs: 115.09000000000002, average SOG: 23.180486862442038
```

In [16]:

```
only_gian_first_runs_light_downwind = only_gian_first_runs_light[only_gian_first_runs_light["TWA"] <= 0]
only_gian_last_runs_heavy_downwind = only_gian_last_runs_heavy[only_gian_last_runs_heavy["TWA"] <= 0]
t_test(only_gian_first_runs_light_downwind, only_gian_last_runs_heavy_downwind)  # downwind

print("\nDownwind for Gian:")
print(f"Weight of Gian on the first runs: {only_gian_first_runs_light_downwind['boat_weight'].mean()}, average SOG: {only_gian_first_runs_light_downwind['SOG'].mean()}")
print(f"Weight of Gian on the last runs: {only_gian_last_runs_heavy_downwind['boat_weight'].mean()}, average SOG: {only_gian_last_runs_heavy_downwind['SOG'].mean()}")
```

```
T-statistic: 10.288, p-value: 0.000000000000000
The difference is statistically significant, keeping data split.

Downwind for Gian:
Weight of Gian on the first runs: 109.08999999999999, average SOG: 27.405528255528253
Weight of Gian on the last runs: 115.08999999999997, average SOG: 26.918731563421826
```