

```
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
43347	2025-06-09T12:37:45.852Z	1.749473e+09	52.7	52.7	43.508366	43.508364	43.508370	43.508376	NaN	125.520	...	09_06_Run5	1	Karl Maeder	SenseBoard	master	106.975
45297	2025-06-09T12:46:51.360Z	1.749473e+09	49.8	49.8	43.502480	43.502478	43.502484	43.502490	1.0	119.521	...	09_06_Run6	1	Karl Maeder	SenseBoard	slave	106.975
46164	2025-06-09T12:48:44.453Z	1.749473e+09	55.7	55.7	43.503545	43.503547	43.503541	43.503535	1.0	95.878	...	09_06_Run6	2	Karl Maeder	SenseBoard	slave	106.975
43685	2025-06-09T12:38:19.653Z	1.749473e+09	54.5	54.5	43.504905	43.504903	43.504909	43.504916	NaN	118.100	...	09_06_Run5	1	Karl Maeder	SenseBoard	master	106.975
45340	2025-06-09T12:46:55.652Z	1.749473e+09	56.2	56.2	43.502013	43.502011	43.502017	43.502023	1.0	143.230	...	09_06_Run6	1	Karl Maeder	SenseBoard	slave	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
53131	2025-06-09T13:19:30.072Z	1.749475e+09	49.5	49.5	43.507976	43.507978	43.507972	43.507966	NaN	89.7	...	09_06_Run9	2	Karl Maeder	SenseBoard	master	100.975
51509	2025-06-09T13:16:03.256Z	1.749475e+09	69.2	69.2	43.506181	43.506179	43.506185	43.506192	NaN	105.8	...	09_06_Run9	1	Karl Maeder	SenseBoard	master	100.975
49143	2025-06-09T13:05:33.255Z	1.749474e+09	61.9	61.9	43.507798	43.507797	43.507803	43.507809	NaN	137.9	...	09_06_Run8	1	Karl Maeder	SenseBoard	slave	100.975
56391	2025-06-09T13:34:58.161Z	1.749476e+09	63.0	63.0	43.503889	43.503887	43.503893	43.503899	NaN	158.2	...	09_06_Run11	1	Karl Maeder	SenseBoard	master	100.975
53837	2025-06-09T13:24:50.259Z	1.749475e+09	53.0	53.0	43.507257	43.507255	43.507261	43.507268	1.0	106.3	...	09_06_Run10	1	Karl Maeder	SenseBoard	slave	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: 0.00000000000004
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
44568	2025-06-09T12:38:39.153Z	1.749473e+09	56.0	56.0	43.503012	43.503010	43.503017	43.503023	1.0	4.912	...	09_06_Run5	1	SenseBoard	Karl Maeder	slave	109.09
44141	2025-06-09T12:37:56.456Z	1.749473e+09	56.9	56.9	43.507426	43.507424	43.507430	43.507436	1.0	7.900	...	09_06_Run5	1	SenseBoard	Karl Maeder	slave	109.09
45488	2025-06-09T12:46:02.360Z	1.749473e+09	63.5	63.5	43.507438	43.507436	43.507442	43.507448	1.0	3.500	...	09_06_Run6	1	SenseBoard	Karl Maeder	master	109.09
46926	2025-06-09T12:49:13.360Z	1.749473e+09	52.1	52.1	43.506898	43.506899	43.506893	43.506887	1.0	5.300	...	09_06_Run6	2	SenseBoard	Karl Maeder	master	109.09
45957	2025-06-09T12:46:49.256Z	1.749473e+09	60.0	60.0	43.502470	43.502468	43.502474	43.502481	1.0	6.100	...	09_06_Run6	1	SenseBoard	Karl Maeder	master	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
52162	2025-06-09T13:15:59.855Z	1.749475e+09	57.0	57.0	43.506579	43.506577	43.506583	43.506589	1.0	6.100	...	09_06_Run9	1	SenseBoard	Karl Maeder	slave	115.09
52521	2025-06-09T13:16:35.756Z	1.749475e+09	58.4	58.4	43.502819	43.502817	43.502823	43.502829	1.0	8.800	...	09_06_Run9	1	SenseBoard	Karl Maeder	slave	115.09
49707	2025-06-09T13:05:27.756Z	1.749474e+09	58.0	58.0	43.508255	43.508254	43.508260	43.508266	NaN	7.528	...	09_06_Run8	1	SenseBoard	Karl Maeder	master	115.09
53709	2025-06-09T13:19:33.959Z	1.749475e+09	38.3	38.3	43.508707	43.508709	43.508703	43.508697	NaN	4.978	...	09_06_Run9	2	SenseBoard	Karl Maeder	slave	115.09
52368	2025-06-09T13:16:20.453Z	1.749475e+09	63.2	63.2	43.504449	43.504447	43.504453	43.504459	1.0	5.200	...	09_06_Run9	1	SenseBoard	Karl Maeder	slave	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

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