

```
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
46091	2025-06-09T12:48:37.160Z	1.749473e+09	48.7	48.7	43.502618	43.502620	43.502614	43.502608	1.0	108.711	...	09_06_Run6	2	Karl Maeder	SenseBoard	slave	106.975
47029	2025-06-09T12:54:05.950Z	1.749474e+09	57.1	57.1	43.508425	43.508423	43.508429	43.508435	NaN	98.489	...	09_06_Run7	1	Karl Maeder	SenseBoard	master	106.975
43349	2025-06-09T12:37:46.048Z	1.749473e+09	59.2	59.2	43.508346	43.508344	43.508350	43.508356	NaN	113.600	...	09_06_Run5	1	Karl Maeder	SenseBoard	master	106.975
46257	2025-06-09T12:48:53.760Z	1.749473e+09	55.1	55.1	43.504638	43.504640	43.504634	43.504628	1.0	103.012	...	09_06_Run6	2	Karl Maeder	SenseBoard	slave	106.975
44951	2025-06-09T12:46:16.757Z	1.749473e+09	53.6	53.6	43.506068	43.506066	43.506072	43.506078	1.0	119.100	...	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
50674	2025-06-09T13:08:49.655Z	1.749475e+09	46.1	46.1	43.506608	43.506610	43.506604	43.506598	NaN	84.2	...	09_06_Run8	2	Karl Maeder	SenseBoard	slave	100.975
54301	2025-06-09T13:25:36.655Z	1.749476e+09	53.4	53.4	43.502254	43.502252	43.502258	43.502264	1.0	94.7	...	09_06_Run10	1	Karl Maeder	SenseBoard	slave	100.975
51861	2025-06-09T13:16:38.454Z	1.749475e+09	55.5	55.5	43.502559	43.502557	43.502563	43.502569	NaN	131.1	...	09_06_Run9	1	Karl Maeder	SenseBoard	master	100.975
54101	2025-06-09T13:25:16.644Z	1.749476e+09	50.2	50.2	43.504420	43.504418	43.504424	43.504431	1.0	139.8	...	09_06_Run10	1	Karl Maeder	SenseBoard	slave	100.975
53157	2025-06-09T13:19:32.653Z	1.749475e+09	45.8	45.8	43.508272	43.508274	43.508268	43.508262	NaN	102.3	...	09_06_Run9	2	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: {data_9juin_first_runs['SOG'].mean()}")
print(f"Weight of Karl on the last runs: {only_karl_last_runs_light['boat_weight'].mean()}, average SOG: {data_9juin_first_runs['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: 106.9750000000001, average SOG: 24.09159415245388
Weight of Karl on the last runs: 100.975, average SOG: 24.09159415245388

```
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: 100.9750000000001, 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
44371	2025-06-09T12:38:19.456Z	1.749473e+09	42.4	42.4	43.504992	43.504990	43.504996	43.505003	1.0	1.1	...	09_06_Run5	1	SenseBoard	Karl Maeder	slave	109.09
45435	2025-06-09T12:45:57.055Z	1.749473e+09	60.4	60.4	43.507987	43.507985	43.507991	43.507997	1.0	7.4	...	09_06_Run6	1	SenseBoard	Karl Maeder	master	109.09
44602	2025-06-09T12:38:42.561Z	1.749473e+09	60.8	60.8	43.502660	43.502658	43.502664	43.502670	1.0	6.0	...	09_06_Run5	1	SenseBoard	Karl Maeder	slave	109.09
46762	2025-06-09T12:48:56.959Z	1.749473e+09	52.2	52.2	43.504982	43.504984	43.504978	43.504972	1.0	7.2	...	09_06_Run6	2	SenseBoard	Karl Maeder	master	109.09
46571	2025-06-09T12:48:37.855Z	1.749473e+09	52.0	52.0	43.502633	43.502635	43.502628	43.502622	1.0	3.8	...	09_06_Run6	2	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
52295	2025-06-09T13:16:13.160Z	1.749475e+09	60.0	60.0	43.505233	43.505231	43.505237	43.505244	1.0	7.363	...	09_06_Run9	1	SenseBoard	Karl Maeder	slave	115.09
54914	2025-06-09T13:25:36.654Z	1.749476e+09	54.1	54.1	43.502050	43.502048	43.502054	43.502061	NaN	5.700	...	09_06_Run10	1	SenseBoard	Karl Maeder	master	115.09
57809	2025-06-09T13:37:16.656Z	1.749476e+09	55.1	55.1	43.502413	43.502415	43.502408	43.502402	NaN	1.700	...	09_06_Run11	2	SenseBoard	Karl Maeder	slave	115.09
53373	2025-06-09T13:19:00.358Z	1.749475e+09	47.3	47.3	43.504768	43.504770	43.504764	43.504758	NaN	4.400	...	09_06_Run9	2	SenseBoard	Karl Maeder	slave	115.09
58073	2025-06-09T13:37:43.056Z	1.749476e+09	44.8	44.8	43.505549	43.505551	43.505544	43.505538	NaN	5.000	...	09_06_Run11	2	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

```
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