

```

import pandas as pd
import matplotlib.pyplot as plt
import time

# plt.style.use('bmh')

start = time.perf_counter()

df = pd.read_csv('HeartWatch-Workouts-20230717-to-20230818.csv')

df = df.drop(['ISO', 'from', 'to', 'rpe', 'Load', 'bpm-lo', 'bpm-90%+-%', '90%+-mins',
             'bpm-80-90%-%', '80-90%-mins', 'bpm-70-80%-%', '70-80%-mins', 'bpm-60-70%-%',
             '60-70%-mins', 'bpm-50-60%-%', '50-60%-mins'], axis=1)

df['Duration'] = pd.to_timedelta(df['Duration'])
df['/km'] = pd.to_timedelta(df['/km'])

# df.head()

dfRun = df[df['Type'] == 'Running']
dfBike = df[df['Type'] == 'Cycling']
dfOther = df[~df['Type'].isin(['Running', 'Cycling'])]

# print(dfRun.info())
# print(dfBike.info())
# print(dfOther.head())

dfRun.tail()

```

	Date	Duration	Type	bpm-Avg.	bpm-%	bpm-hi	Cals	Cals/h	km	km/h
17	Friday, Aug 11	0 days 00:39:03	Running	148.2	77.4	161.0	434.2	667.1	5.14	7.5
19	Monday, Aug 14	0 days 00:47:58	Running	118.4	61.8	157.0	380.4	475.8	5.25	6.0
20	Tuesday, Aug 15	0 days 01:08:11	Running	151.3	79.0	177.0	792.8	697.6	9.22	8.0
23	Thursday, Aug 17	0 days 00:41:24	Running	157.9	82.4	189.0	498.9	723.0	6.12	8.0
24	Friday, Aug 18	0 days 00:49:30	Running	139.7	72.9	162.0	550.8	667.5	6.64	8.0

```

# Pace Calculation
# Calculate the total seconds of Duration column

```

```

total_seconds = dfRun['Duration'].dt.total_seconds()

# Average Pace from M/S
mps = dfRun['km'].sum()*1000 / total_seconds.sum()
kph = mps * 3.6
mpk = 60 / kph

integer_part = int(mpk)
decimal_part = mpk - integer_part

# Convert decimal part to minutes by dividing by 60
decimal_minutes = round(decimal_part * 60,0)

# Weighted HR
dfRunWght = dfRun

# Convert the time delta to decimal hours and create a new column
dfRunWght["Duration"] = dfRunWght["Duration"].apply(lambda x: x.total_seconds() / 3600)
dfRunWght['Weighted HR'] = dfRunWght['Duration'] * dfRunWght['bpm-Avg.']

```

/var/folders/y_/8cmvsg791ys9qqbzj_yxrlj40000gn/T/ipykernel_12821/2394796668.py:5: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide
dfRunWght["Duration"] = dfRunWght["Duration"].apply(lambda x: x.total_seconds() / 3600)
/var/folders/y_/8cmvsg791ys9qqbzj_yxrlj40000gn/T/ipykernel_12821/2394796668.py:6: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide
dfRunWght['Weighted HR'] = dfRunWght['Duration'] * dfRunWght['bpm-Avg.']

```

# Average Duration, Distance, Average HR, Average Max HR Average Calories
avg_dist = round(dfRun['km'].mean(),2)
avg_hr = round(dfRun['bpm-Avg.'].mean(),2)
avg_wght_hr = round(dfRunWght['Weighted HR'].sum() / dfRunWght['Duration'].sum(),2)
avg_maxhr = round(dfRun['bpm-hi'].mean(),2)
avg_cals = round(dfRun['Cals'].mean(),2)
avg_dur = dfRun['Duration'].mean()

```

```

# Count Runs
num_runs = dfRun['km'].count()
ovr_5k = dfRun[dfRun['km'] >=5].count()['km']
povr_5k = round(ovr_5k / num_runs *100,2)
ovr_10k = dfRun[dfRun['km'] >=10].count()['km']
povr_10k = round(ovr_10k / num_runs *100,2)

# Maximums
max_dur = dfRun['Duration'].max()
max_dist = dfRun['km'].max()
max_avghr = dfRun['bpm-Avg.'].max()
max_maxhr = dfRun['bpm-hi'].max()
max_cals = dfRun['Cals'].max()

# Durations to Time Format
avg_dur_h = int(avg_dur)
max_dur_h = int(max_dur)

avg_dur_m_dec = (avg_dur - avg_dur_h)*60
max_dur_m_dec = (max_dur - max_dur_h)*60

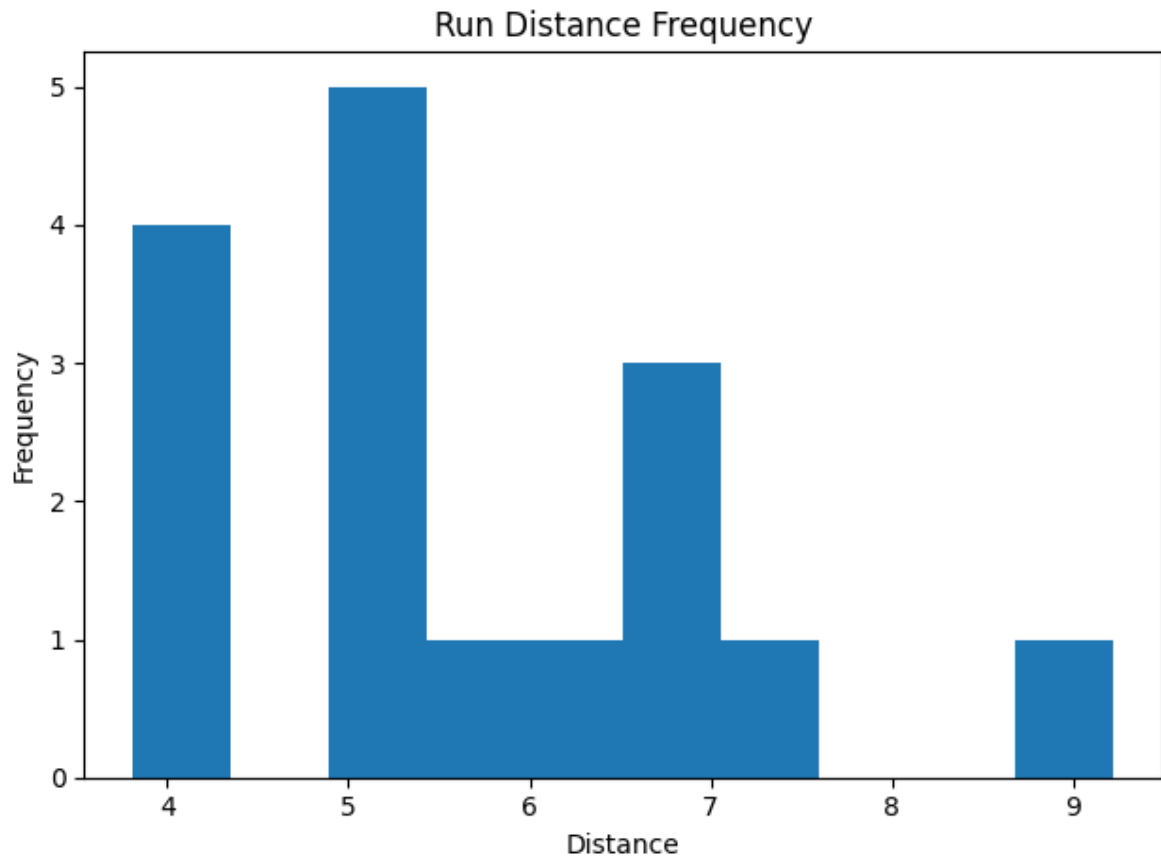
avg_dur_m = int(avg_dur_m_dec)
max_dur_m = int(max_dur_m_dec)

avg_dur_s = int((avg_dur_m_dec - avg_dur_m)*60)
max_dur_s = int((max_dur_m_dec - max_dur_m)*60)

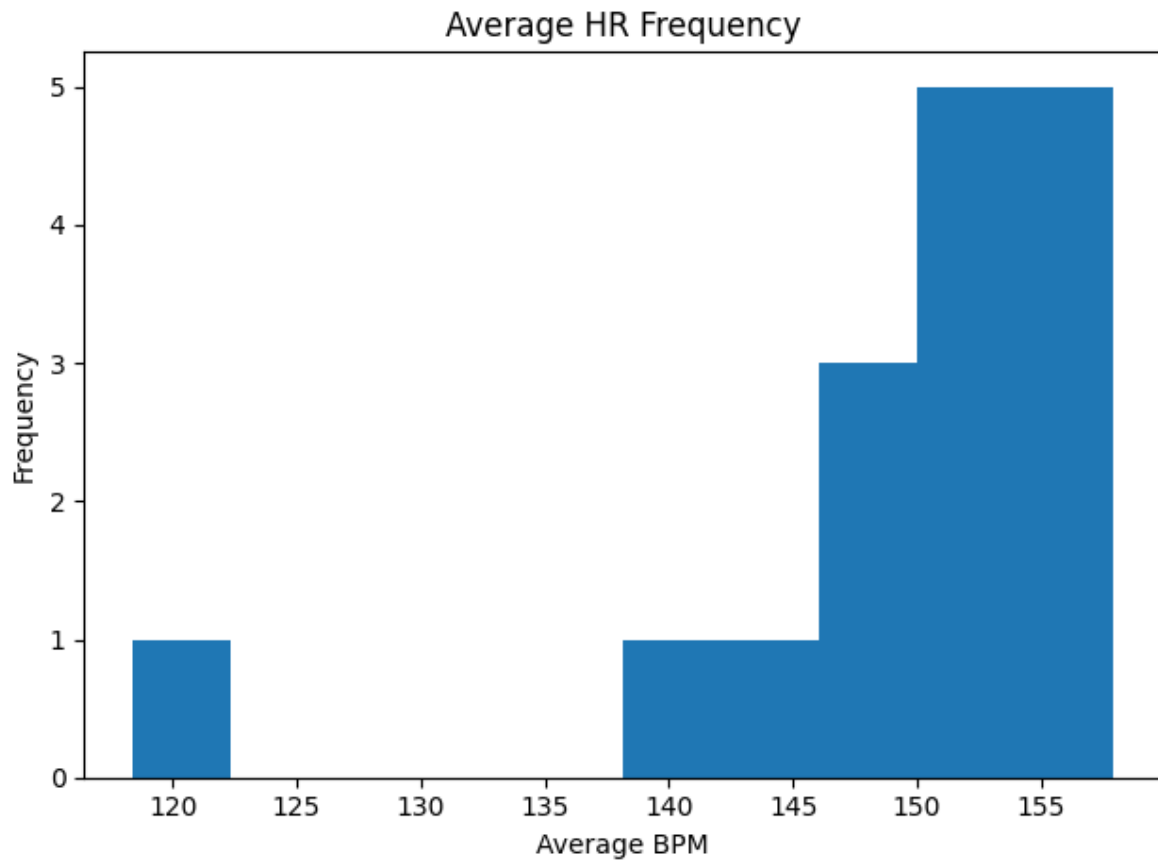
avg_dur_f = str(avg_dur_h) + ':' + str(avg_dur_m) + ':' + str(avg_dur_s)
max_dur_f = str(max_dur_h) + ':' + str(max_dur_m) + ':' + str(max_dur_s)

plt.hist(dfRun['km'])
plt.title('Run Distance Frequency')
plt.xlabel('Distance')
plt.ylabel('Frequency')
# plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('Graphs/Distance Frequency.png', dpi=300)
plt.show()

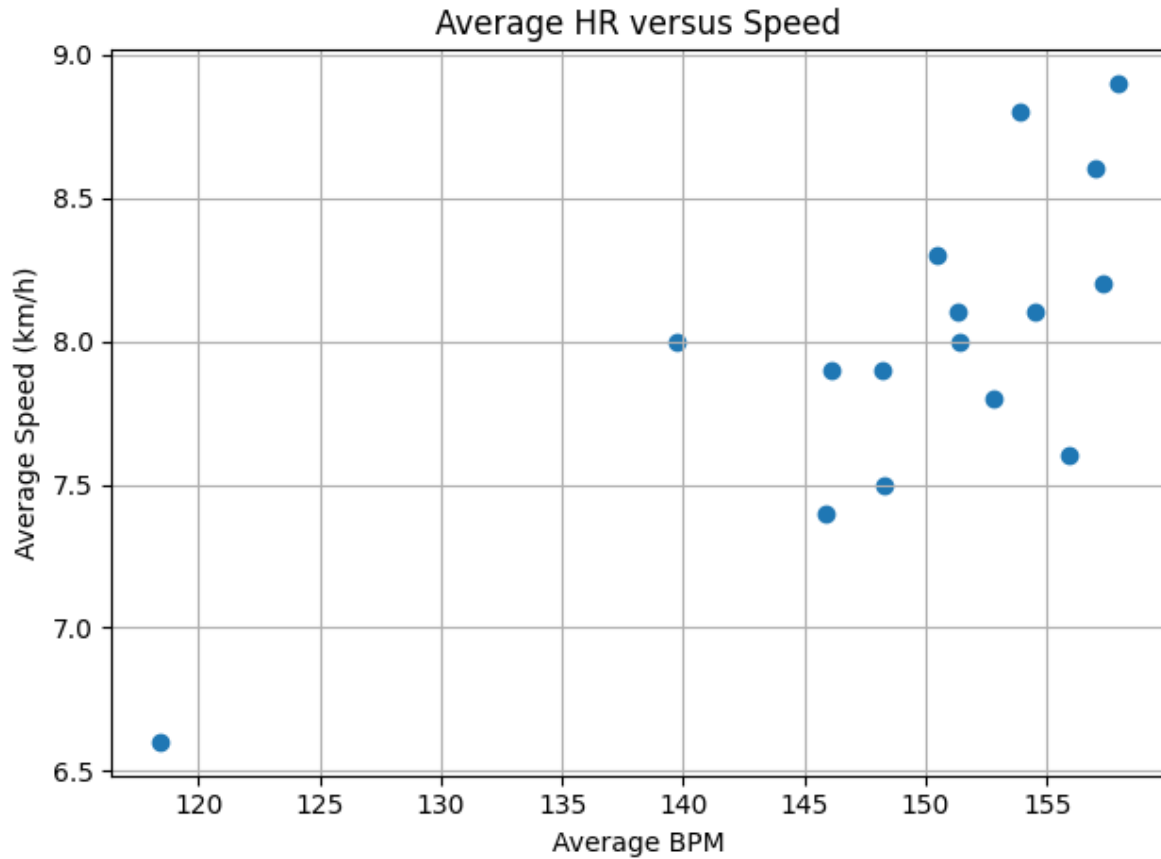
```



```
plt.hist(dfRun['bpm-Avg.'])  
plt.title('Average HR Frequency')  
plt.xlabel('Average BPM')  
plt.ylabel('Frequency')  
# plt.xticks(rotation=45)  
plt.tight_layout()  
plt.savefig('Graphs/HR Frequency.png', dpi=300)  
plt.show()
```



```
plt.scatter('bpm-Avg.', 'km/h', data=dfRun)
plt.xlabel('Average BPM')
plt.ylabel('Average Speed (km/h)')
plt.title('Average HR versus Speed')
plt.grid(True)
plt.tight_layout()
plt.savefig('Graphs/SpeedvsHR.png', dpi=300)
plt.show()
```



```

print(f'Runs: {num_runs}')
print(f'Runs over 5k(%): {ovr_5k} ({povr_5k}%)')
print(f'Runs over 10k(%): {ovr_10k} ({povr_10k}%)')

print('\n')
print(f'Average Duration: {avg_dur_f}')
print(f'Average Distance: {avg_dist}')
print(f"Average Pace: {integer_part}:{decimal_minutes}")
print(f'Average Weighted HR: {avg_wght_hr}')
print(f'Average HR: {avg_hr}')
print(f'Average Max HR: {avg_maxhr}')
print(f'Average Calories: {avg_cals}')

print('\n')
print(f'Max Duration: {max_dur_f}')

```

```
print(f'Max Distance: {max_dist}')
print(f'Max Average HR: {max_avghr}')
print(f'Max Max HR: {max_maxhr}')
print(f'Max Calories: {max_cals}')

print('\n')
print(f'Runtime: {round(time.perf_counter() - start,2)}s')
```

Runs: 16
Runs over 5k(%): 12 (75.0%)
Runs over 10k(%): 0 (0.0%)

Average Duration: 0:42:58
Average Distance: 5.7
Average Pace: 7:33.0
Average Weighted HR: 148.48
Average HR: 149.32
Average Max HR: 168.25
Average Calories: 462.42

Max Duration: 1:8:11
Max Distance: 9.22
Max Average HR: 157.9
Max Max HR: 189.0
Max Calories: 792.8

Runtime: 0.82s