Running Workout Statistics and Graphs

Aaron Palmer

2023-08-30

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import time
from datetime import datetime
import warnings

plt.style.use('ggplot')
warnings.filterwarnings('ignore')

start = time.perf_counter()
```

```
AttributeError: module 'numpy' has no attribute 'round'
```

Only counting activities post Europe, export HeartWatch Workouts from Jul 17 - Today and add the csv to the wd

```
df = pd.read_csv('HeartWatch-Workouts-20230717-to-20230902.csv')
```

```
df = df.drop(['Date', '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)

# Drop Run w/ Bear
df = df.drop(19)
```

```
# Fix Datetime Columns
df['ISO'] = pd.to_datetime(df['ISO'])
df['Duration'] = pd.to_timedelta(df['Duration'])
df['/km'] = pd.to_timedelta(df['/km'])
df['Date'] = df['ISO'].dt.date

df.set_index('ISO', inplace=True)

wklySUM = pd.DataFrame(df[df['Type'] == 'Running'].groupby(pd.Grouper(freq='W-SUN')).agg('
wklyAVG = pd.DataFrame(df[df['Type'] == 'Running'].groupby(pd.Grouper(freq='W-SUN')).agg('

# print(wklySUM.head())
# print(wklyAVG.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())
```

```
# 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
```

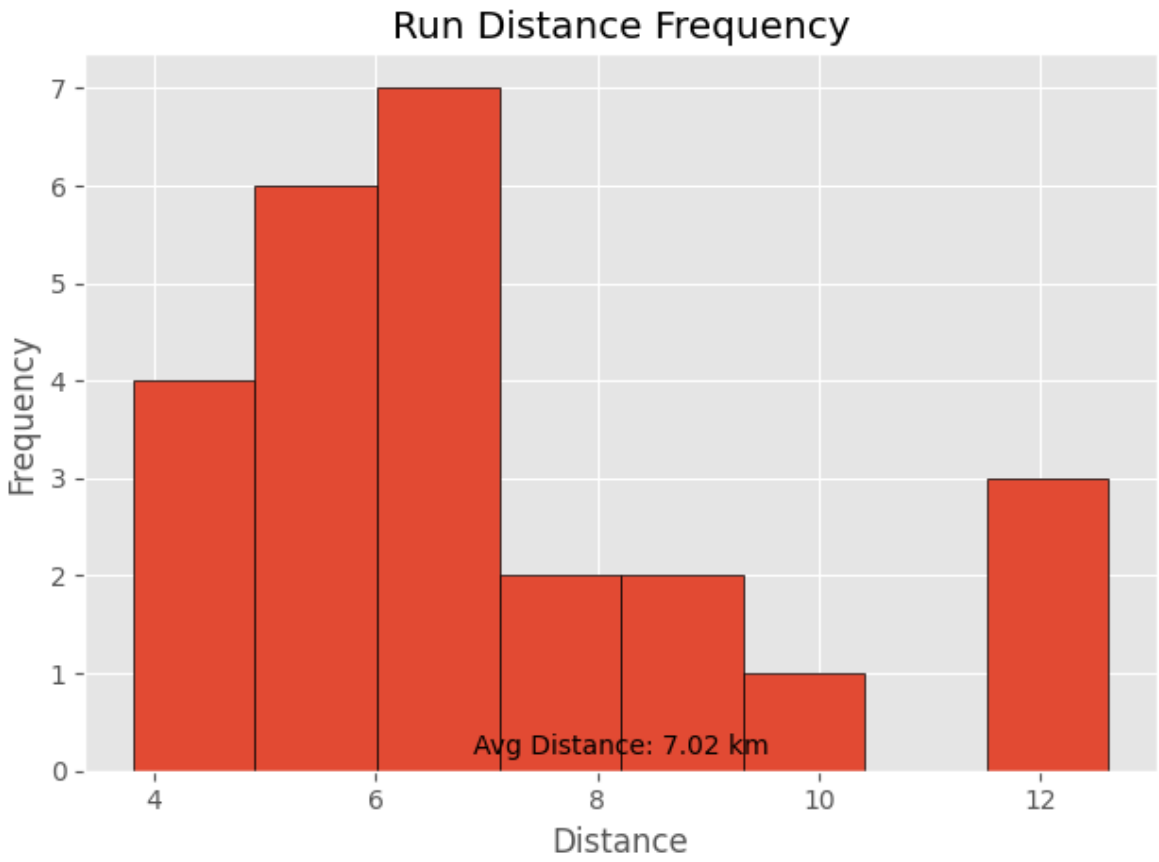
```
# Convert the time delta to decimal hours and create a new column
dfRunWght["Duration"] = dfRunWght["Duration"].apply(lambda x: x.total_seconds() / 3600)
dfRunWght['Weighted HR'] = dfRunWght['Duration'] * dfRunWght['bpm-Avg.']

# HR/Speed Decimal
dfRun['HR/Speed'] = (dfRun['bpm-Avg.'] / dfRun['km/h'])

# Cumulative Sum
dfRun['CumKM'] = dfRun['km'].cumsum()
```

```
# dfRun.head()
```

Graphs

Running Distance Frequency

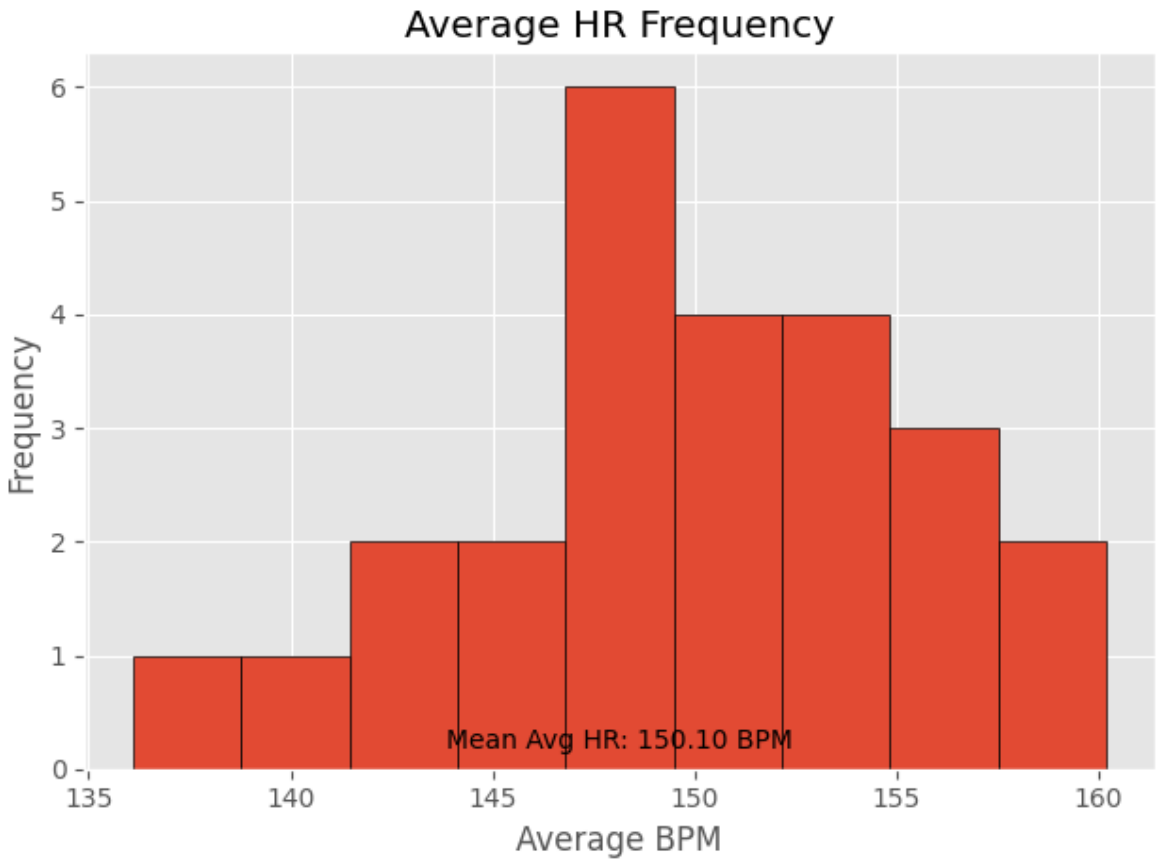
```
rdf_bins = int(dfRun['km'].max() - dfRun['km'].min())

plt.hist(dfRun['km'], edgecolor='black', bins=rdf_bins)

# Average Distance Text
avg_distance = dfRun['km'].mean()
mid_dist = (dfRun['km'].max() + dfRun['km'].min()) / 2
plt.text(mid_dist, 0.25, f'Avg Distance: {avg_distance:.2f} km',
         ha='center', va='center', fontsize=10, color='black')

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()
```



Average Heart Rate Frequency

```
hrf_bins = int((dfRun['bpm-Avg.'].max() - dfRun['bpm-Avg.'].min())/2.5)

plt.hist(dfRun['bpm-Avg.'], edgecolor='black', bins=hrf_bins)

# Average HR Text
avg_hr = dfRun['bpm-Avg.'].mean()
mid_hr = (dfRun['bpm-Avg.'].max() + dfRun['bpm-Avg.'].min()) / 2
plt.text(mid_hr, .25, f'Mean Avg HR: {avg_hr:.2f} BPM',
         ha='center', va='center', fontsize=10, color='black', transform=plt.gca().transDa

plt.title('Average HR Frequency')
plt.xlabel('Average BPM')
plt.ylabel('Frequency')
```

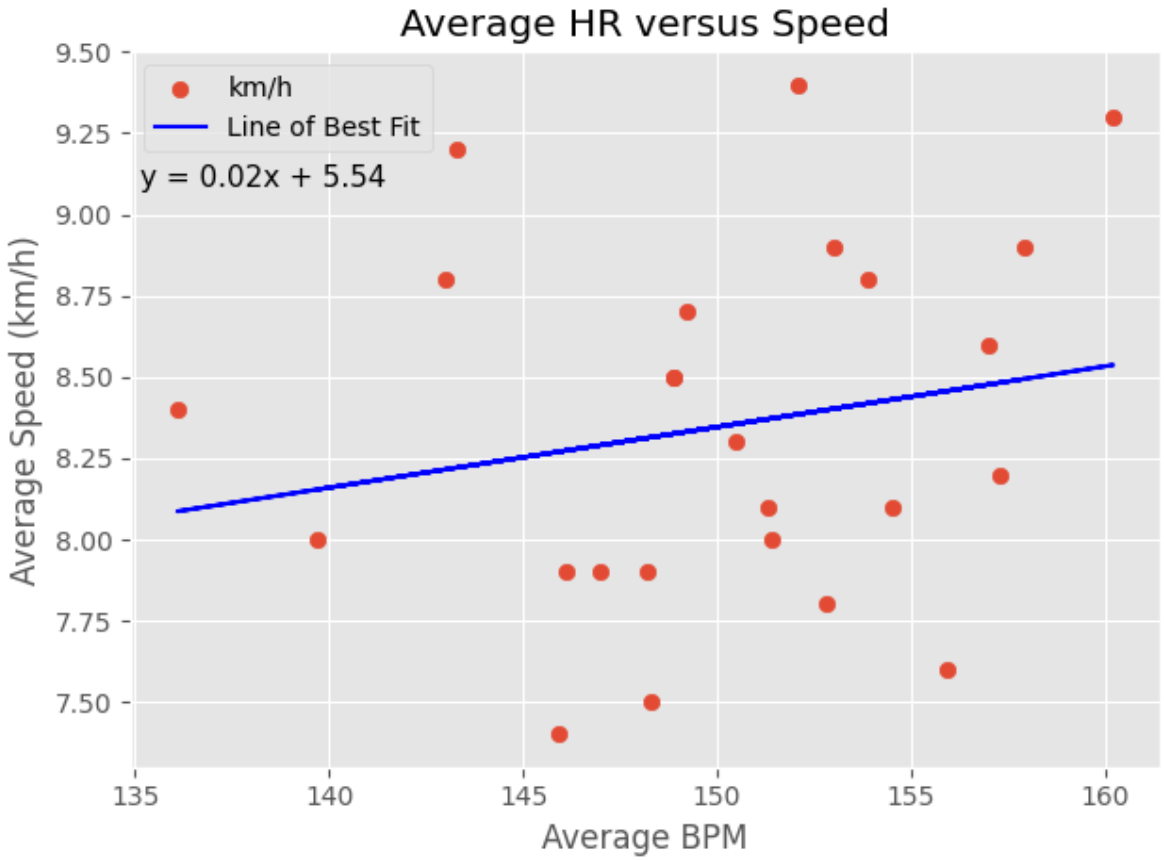
```
# plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('Graphs/HR Frequency.png', dpi=300)
plt.show()
```



Average Heart Rate versus Average Speed (km/h)

```
# Line of Best Fit
# Fit a linear regression line to the data
degree = 1
coefficients = np.polyfit(dfRun['bpm-Avg.'], dfRun['km/h'], degree)
slope = coefficients[0]
intercept = coefficients[1]
# Calculate the predicted y-values using the line equation
```

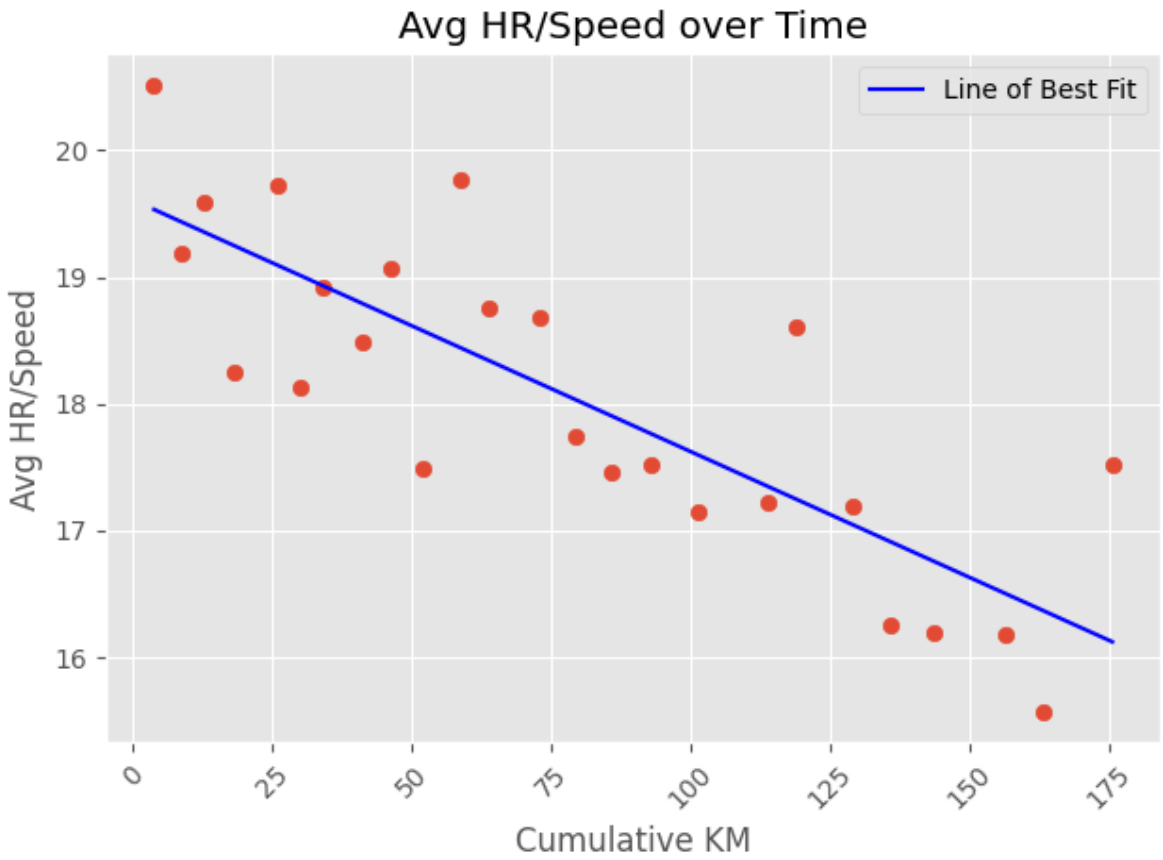
```
predicted_y = slope * dfRun['bpm-Avg.'] + intercept
equation = f'y = {slope:.2f}x + {intercept:.2f}'

# Plot
plt.scatter('bpm-Avg.', 'km/h', data=dfRun)
plt.plot(dfRun['bpm-Avg.'], predicted_y, color='blue', label='Line of Best Fit')
plt.xlabel('Average BPM')
plt.ylabel('Average Speed (km/h)')
plt.title('Average HR versus Speed')
plt.text(0.01, 0.81, equation, fontsize=11, transform=plt.gca().transAxes)
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.savefig('Graphs/SpeedvsHR.png', dpi=300)
plt.show()
```



Heart Rate / Speed over Time

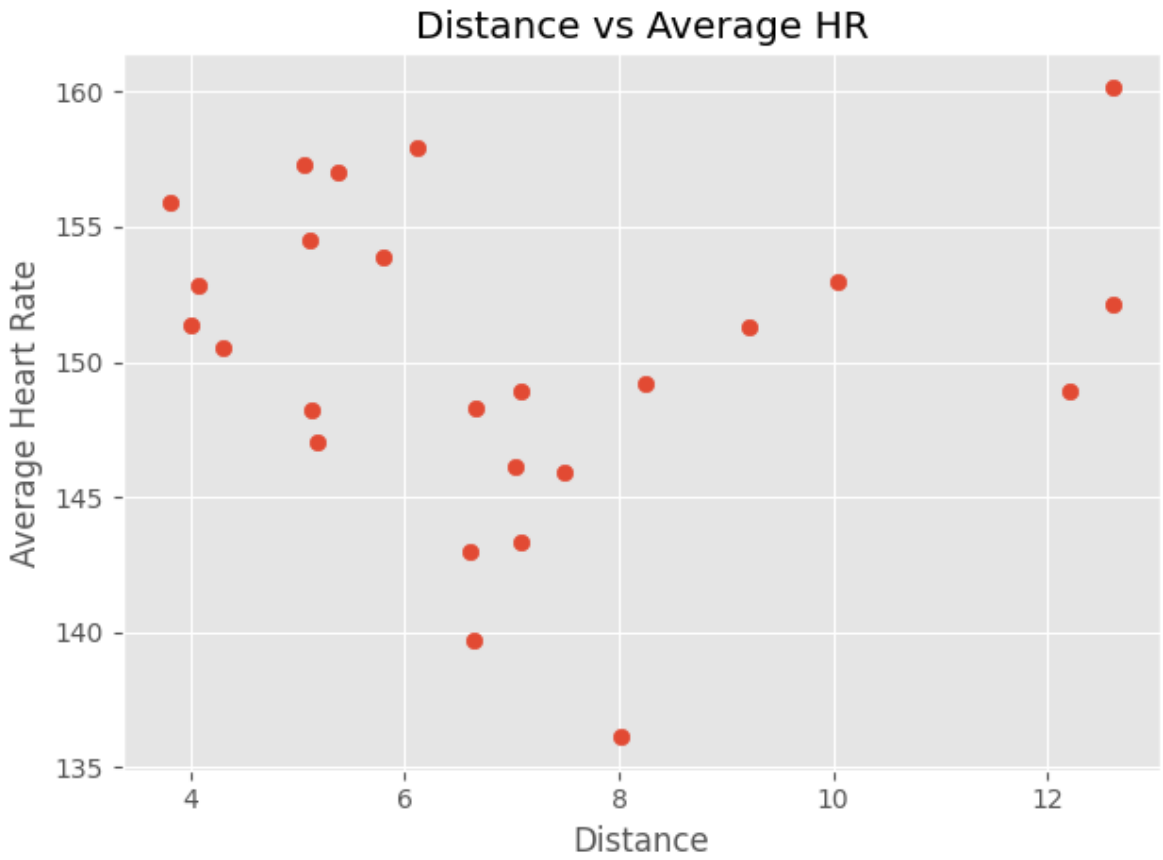
Lower is better

```
degree1 = 1
coefficients1 = np.polyfit(dfRun['CumKM'], dfRun['HR/Speed'], degree)
slope1 = coefficients1[0]
intercept1 = coefficients1[1]
# Calculate the predicted y-values using the line equation
predicted_y1 = slope1 * dfRun['CumKM'] + intercept1
equation1 = f'y = {slope1:.2f}x + {intercept1:.1f}'

plt.scatter(dfRun['CumKM'], y= dfRun['HR/Speed'])
plt.plot(dfRun['CumKM'], predicted_y1, color='blue', label='Line of Best Fit')
# plt.text(0.75, 0.86, equation1, fontsize=10, transform=plt.gca().transAxes)
plt.xlabel('Cumulative KM')
plt.xticks(rotation=45)
plt.ylabel('Avg HR/Speed')
plt.title('Avg HR/Speed over Time')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.savefig('Graphs/HR-Speed over Time.png', dpi=300)
plt.show()
```



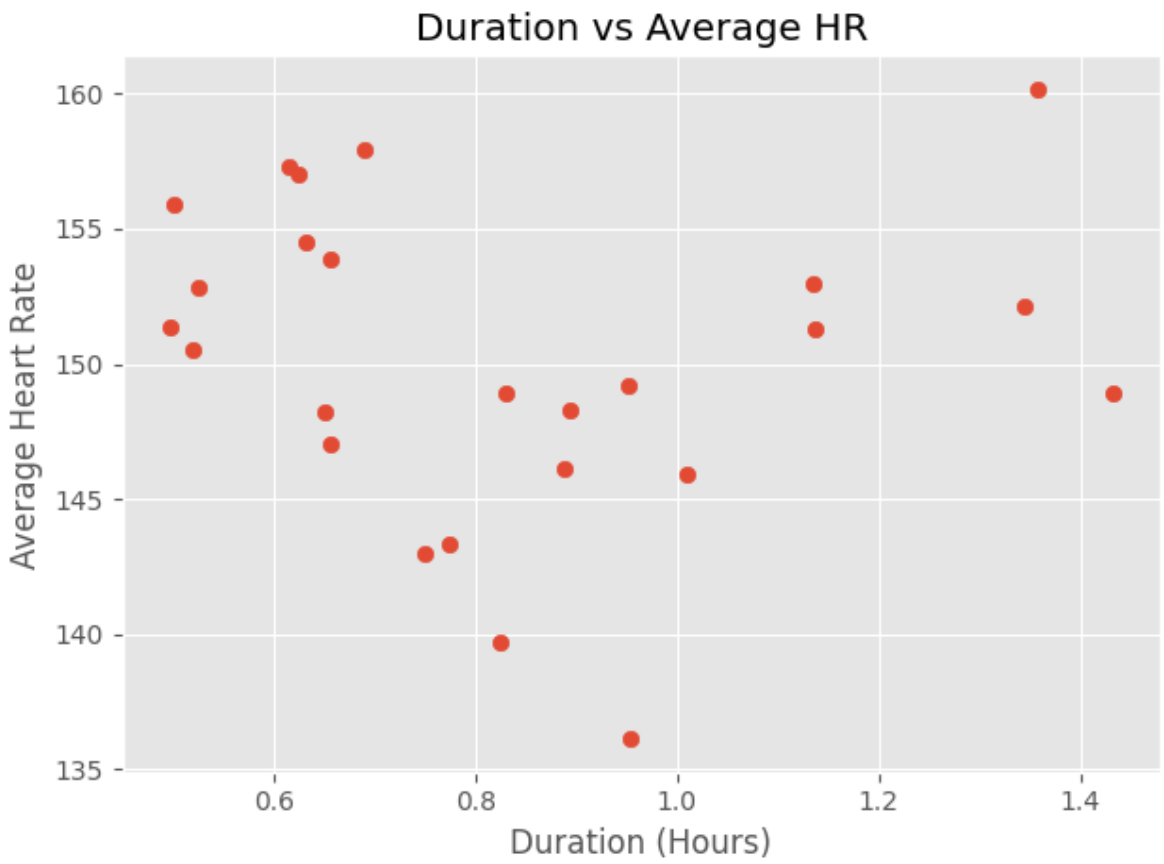
Distance vs Average Heart Rate

```
plt.scatter('km', 'bpm-Avg.', data=dfRun)
plt.xlabel('Distance')
plt.ylabel('Average Heart Rate')
plt.title('Distance vs Average HR')
plt.grid(True)
plt.tight_layout()
plt.savefig('Graphs/Distance vs Avg HR.png', dpi=300)
```



Duration vs Average Heart Rate

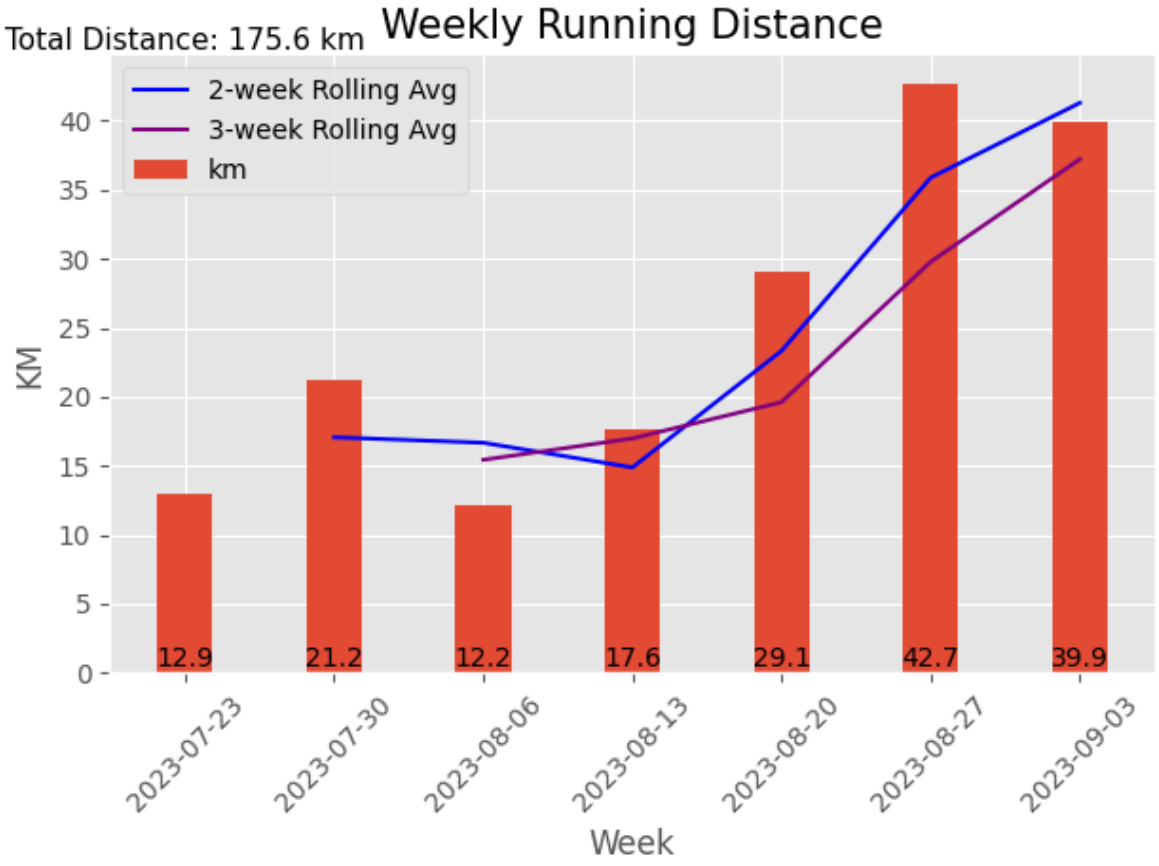
```
plt.scatter('Duration', 'bpm-Avg.', data=dfRun)
plt.xlabel('Duration (Hours)')
plt.ylabel('Average Heart Rate')
plt.title('Duration vs Average HR')
plt.grid(True)
plt.tight_layout()
plt.savefig('Graphs/Duration vs Avg HR.png', dpi=300)
```



Weekly Distance

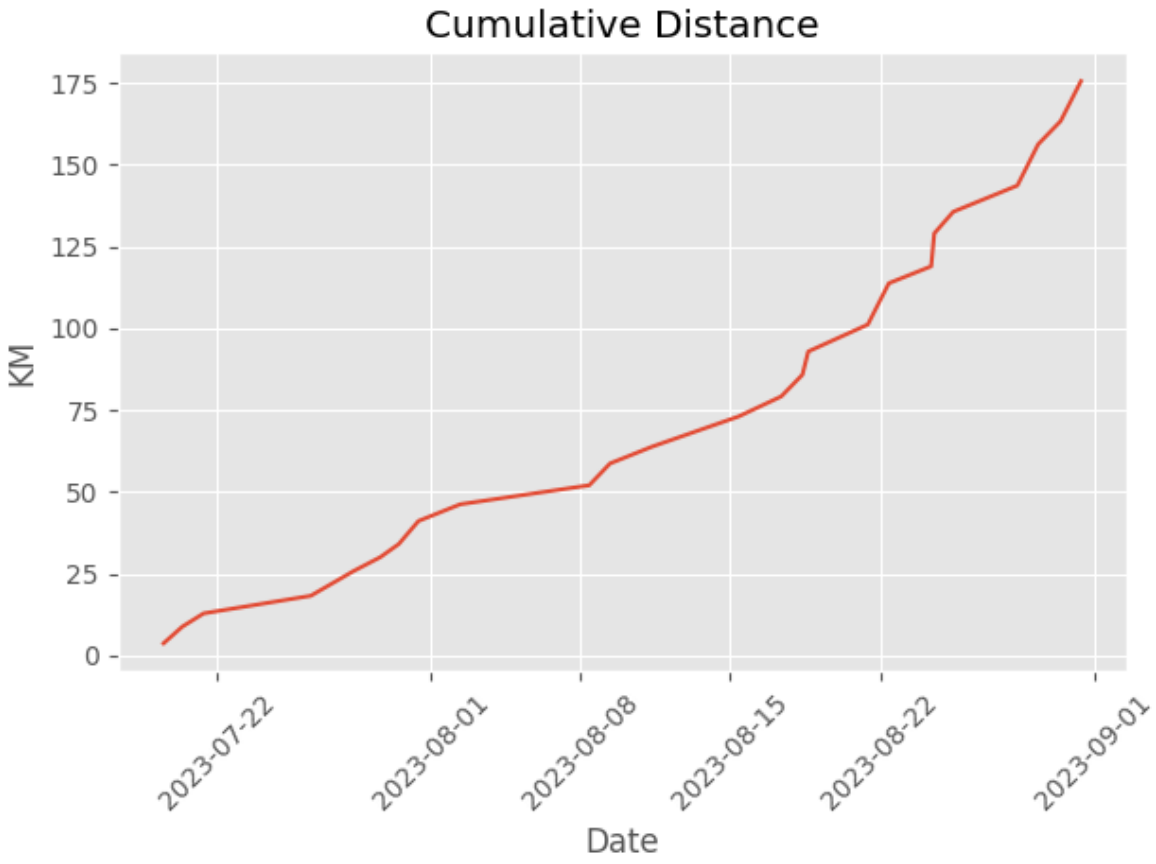
```
# plt.figure(figsize=(10,6))

# Rolling Average
rolling_average2 = wklySUM['km'].rolling(window=2).mean()
plt.plot(wklySUM.index, rolling_average2, label='2-week Rolling Avg', color='blue')
rolling_average3 = wklySUM['km'].rolling(window=3).mean()
plt.plot(wklySUM.index, rolling_average3, label='3-week Rolling Avg', color='purple')

# Toal Distance Text
total_distance = wklySUM['km'].sum()
plt.text(wklySUM.index[0], wklySUM['km'].max() + 2, f'Total Distance: {total_distance:.1f}
         ha='center', va='bottom', fontsize=11, color='black', transform=plt.gca().transDa
```

```
plt.bar(wklySUM.index, wklySUM['km'], label='km', width=2.6)
for idx, value in enumerate(wklySUM['km']):
    plt.text(wklySUM.index[idx], 0, f'{value:.1f}', ha='center', va='bottom',
             fontsize=10, color='black')

plt.xlabel('Week')
plt.ylabel('KM')
plt.title('Weekly Running Distance', fontsize=15)
plt.legend()
plt.xticks(wklySUM.index, wklySUM.index.strftime('%Y-%m-%d'), rotation=45)
plt.tight_layout()
plt.savefig('Graphs/Weekly Distance.png', dpi=300)
plt.show()
```



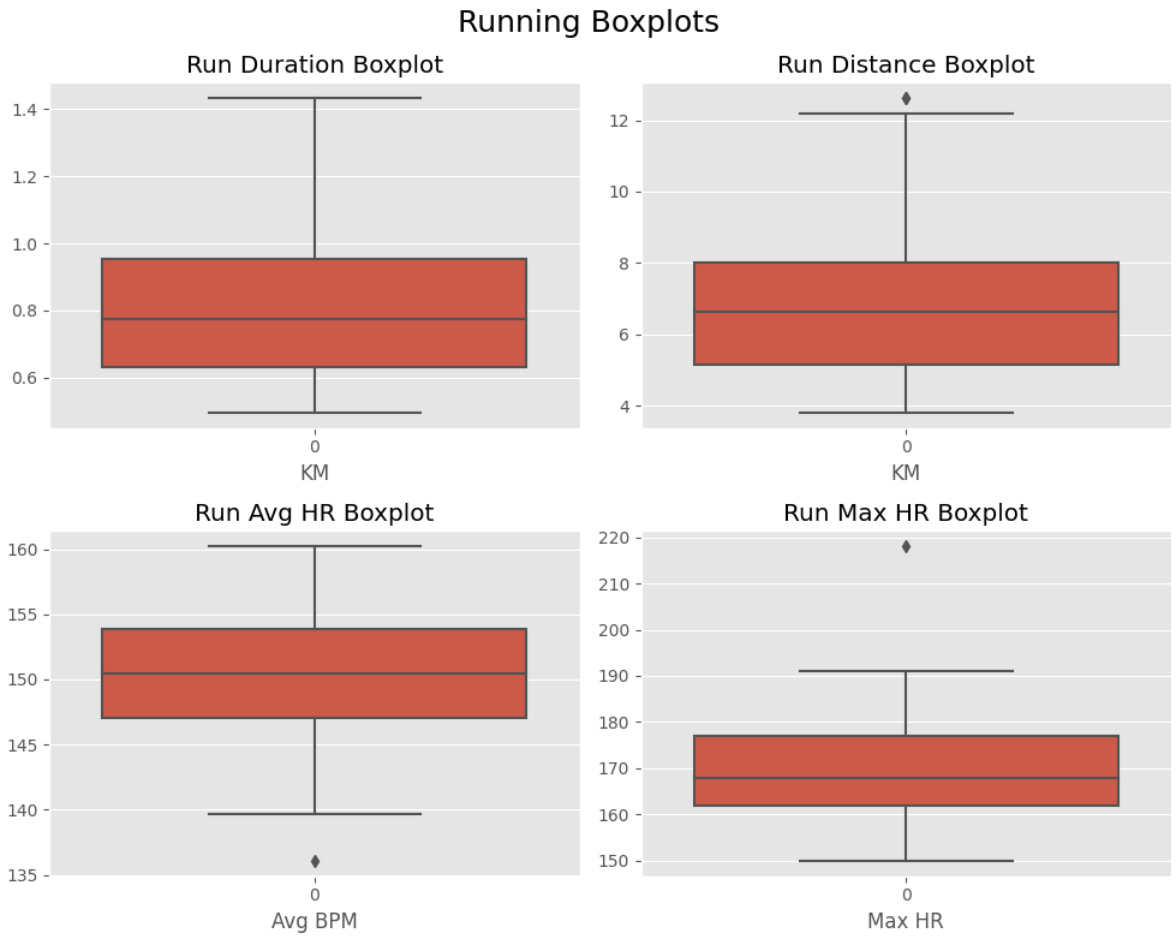
Cumulative Distance

```
plt.plot(dfRun.index, dfRun['CumKM'])
plt.title('Cumulative Distance')
plt.xlabel('Date')
plt.ylabel('KM')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('Graphs/Cumulative Distance.png', dpi=300)
plt.show()
```



Running Boxplots

```
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10, 8))

# Run Duration Boxplot
sns.boxplot(dfRun['Duration'], ax=axes[0, 0])
axes[0, 0].set_title('Run Duration Boxplot')
axes[0, 0].set_xlabel('KM')

# Run Distance Boxplot
sns.boxplot(dfRun['km'], ax=axes[0, 1])
axes[0, 1].set_title('Run Distance Boxplot')
axes[0, 1].set_xlabel('KM')

# Run Average HR Boxplot
sns.boxplot(dfRun['bpm-Avg.'], ax=axes[1, 0])
axes[1, 0].set_title('Run Avg HR Boxplot')
axes[1, 0].set_xlabel('Avg BPM')

# Run Max HR Boxplot
sns.boxplot(dfRun['bpm-hi'], ax=axes[1, 1])
axes[1, 1].set_title('Run Max HR Boxplot')
axes[1, 1].set_xlabel('Max HR')

fig.suptitle('Running Boxplots', fontsize=18)

plt.tight_layout()
plt.savefig('Graphs/Run Boxplots.png', dpi=300)
plt.show()
```




```
# 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()
avg_spd = round(dfRun['km/h'].mean(),2)
avg_ride = round(dfBike['km'].mean())
avg_walk = round(dfOther[dfOther['Type'] == 'Walking']['km'].mean(),2)
avg_hr_ride = round(dfBike['bpm-Avg.'].mean(),2)
avg_max_hr_ride = round(dfBike['bpm-hi'].mean(),2)
avg_hr_walk = round(dfOther[dfOther['Type'] == 'Walking']['bpm-Avg.'].mean(),2)
avg_max_hr_walk = round(dfOther[dfOther['Type'] == 'Walking']['bpm-hi'].mean(),2)
```

```
avg_cals_bike = round(dfBike['Cals'].mean(),)
avg_cals_walk = round(dfOther[dfOther['Type'] == 'Walking']['Cals'].mean(),)
avg_speed_bike = dfBike['km/h'].mean()

# Count Runs
num_runs = dfRun['km'].count()
num_rides = dfBike['km'].count()
num_walks = dfOther[dfOther['Type'] == 'Walking']['km'].count()
avghr_u155 = dfRun[dfRun['bpm-Avg.'] < 155].count()['bpm-Avg.']
pavghr_u155 = round(avghr_u155 / num_runs *100,2)
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)
runs_per_day = round(num_runs / dfRun['Date'].nunique(),3)
date_cts = dfRun['Date'].value_counts()
dbl_days = date_cts[date_cts > 1].count()

# 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()
max_bike = dfBike['km'].max()
max_walk = dfOther[dfOther['Type'] == 'Walking']['km'].max()
max_bike_cals = round(dfBike['Cals'].max(),)
max_walk_cals = round(dfOther[dfOther['Type'] == 'Walking']['Cals'].max(),)

# Totals
tot_dist = round(dfRun['km'].sum(),2)
tot_dur = dfRun['Duration'].sum()
tot_cals = round(dfRun['Cals'].sum(),)
tot_cals = f'{tot_cals:,}'
tot_bike = dfBike['km'].sum()
tot_walk = round(dfOther[dfOther['Type'] == 'Walking']['km'].sum(),2)
tot_bike_cals = round(dfBike['Cals'].sum(),)
tot_walk_cals = round(dfOther[dfOther['Type'] == 'Walking']['Cals'].sum(),)

# Medians
med_dist = round(dfRun['km'].median(),2)
```

```
med_avg_hr = round(dfRun['bpm-Avg.'].median(),2)
med_max_hr = round(dfRun['bpm-hi'].median(),2)
med_cals = round(dfRun['Cals'].median(),)
med_ride = round(dfBike['km'].median(),2)
med_walk = round(dfOther[dfOther['Type'] == 'Walking']['km'].median(),2)
med_avg_hr_ride = round(dfBike['bpm-Avg.'].median(),2)
med_max_hr_ride = round(dfBike['bpm-hi'].median(),2)
med_avg_hr_walk = round(dfOther[dfOther['Type'] == 'Walking']['bpm-Avg.'].median(),2)
med_max_hr_walk = round(dfOther[dfOther['Type'] == 'Walking']['bpm-hi'].median(),2)
med_cals_bike = round(dfBike['Cals'].median(),)
med_cals_walk = round(dfOther[dfOther['Type'] == 'Walking']['Cals'].median(),)
med_speed_bike = dfBike['km/h'].median()

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

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

avg_dur_m = int(avg_dur_m_dec)
max_dur_m = int(max_dur_m_dec)
tot_dur_m = int(tot_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)
tot_dur_s = int((tot_dur_m_dec - tot_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)
tot_dur_f = str(tot_dur_h) + ':' + str(tot_dur_m) + ':' + str(tot_dur_s)
```

Running Stats

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

```
print(f'Runs < 155bpm(%): {avghr_u155} ({pavghr_u155}%)')
print(f'Double Days: {dbl_days}')
print(f'Runs per Day: {runs_per_day}')

print('----------------------------')
print(f'Average Duration: {avg_dur_f}')
print(f'Average Distance: {avg_dist}')
print(f"Average Pace: {integer_part}:{decimal_minutes}")
print(f'Average Speed: {avg_spd}')
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('----------------------------')
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('----------------------------')
print(f'Median Distance: {med_dist}')
print(f'Median Avg HR: {med_avg_hr}')
print(f'Median Max HR: {med_max_hr}')
print(f'Median Calories: {med_cals}')

print('----------------------------')
print(f'Total Duration: {tot_dur_f}')
print(f'Total Distance: {tot_dist}')
print(f'Total Running Calories: {tot_cals}')
```

```
----------------------------
Runs: 25
Runs > 5k(%): 21 (84.0%)
Runs > 10k(%): 4 (16.0%)
Runs < 155bpm(%): 20 (80.0%)
Double Days: 2
Runs per Day: 1.087
----------------------------
Average Duration: 0:50:1
Average Distance: 7.02
```

```
Average Pace: 7:7.0
Average Speed: 8.35
Average Weighted HR: 149.95
Average HR: 150.1
Average Max HR: 170.4
Average Calories: 575.9
---------------------------
Max Duration: 1:25:56
Max Distance: 12.61
Max Average HR: 160.2
Max Max HR: 218.0
Max Calories: 1032.0
---------------------------
Median Distance: 6.64
Median Avg HR: 150.5
Median Max HR: 168.0
Median Calories: 554
---------------------------
Total Duration: 20:50:46
Total Distance: 175.57
Total Running Calories: 14,398
```

Riding/Walking Stats

```
print(f'Rides: {num_rides}')
print(f'Average Distance: {avg_ride}')
print(f'Average HR: {avg_hr_ride}')
print(f'Average Max HR: {avg_max_hr_ride}')
print(f'Average Calories: {avg_cals_bike}')
print(f'Average Speed: {avg_speed_bike}')
print(f'Median Distance: {med_ride}')
print(f'Median Avg HR: {med_avg_hr_ride}')
print(f'Median Max HR: {med_max_hr_ride}')
print(f'Median Calories: {med_cals_bike}')
print(f'Median Speed: {med_speed_bike}')
print('---------------------------')
print(f'Walks: {num_walks}')
print(f'Average Distance: {avg_walk}')
print(f'Average HR: {avg_hr_walk}')
print(f'Average Max HR: {avg_max_hr_walk}')
print(f'Average Calories: {avg_cals_walk}')
```

```
print(f'Median Distance: {med_walk}')
print(f'Median Avg HR: {med_avg_hr_walk}')
print(f'Median Max HR: {med_max_hr_walk}')
print(f'Median Calories: {med_cals_walk}')
print('----------------------------')
print('Maximums')
print(f'Max Bike Distance: {max_bike}')
print(f'Max Bike Calories: {max_bike_cals}')
print(f'Max Walk Distance: {max_walk}')
print(f'Max Walk Calories: {max_walk_cals}')
print('----------------------------')
print("Totals")
print(f'Total Bike Distance: {tot_bike}')
print(f'Total Bike Calories: {tot_bike_cals}')
print(f'Total Walk Distance: {tot_walk}')
print(f'Total Walk Calories: {tot_walk_cals}')
print('----------------------------')
print(f'Runtime: {round(time.perf_counter() - start,2)}s')
```

```
Rides: 5
Average Distance: 29
Average HR: 140.42
Average Max HR: 179.0
Average Calories: 612
Average Speed: 17.82
Median Distance: 31.5
Median Avg HR: 140.9
Median Max HR: 182.0
Median Calories: 661
Median Speed: 19.0
----------------------------
Walks: 9
Average Distance: 3.98
Average HR: 111.94
Average Max HR: 129.0
Average Calories: 231
Median Distance: 3.83
Median Avg HR: 113.3
Median Max HR: 127.0
Median Calories: 208
----------------------------
Maximums
```

```
Max Bike Distance: 35.16
Max Bike Calories: 829
Max Walk Distance: 6.42
Max Walk Calories: 406
----------------------------
Totals
Total Bike Distance: 143.8
Total Bike Calories: 3061
Total Walk Distance: 35.85
Total Walk Calories: 2079
----------------------------
Runtime: 3.08s
```