A Deep Dive in Cycling Data Analytics

Drew Laird

Abstract

As the sport of cycling has continued to grow in popularity, an inseparable aspect of training and racing is the prevalence of data and methods of data collection and analysis. Modern competitive cycling entails collecting large amounts of data from power, heart rate, cadence, speed, distance, location, time, and temperature. While current technology allows the amateur cyclist to do this fairly easily, the processes of data analysis are still being developed by a variety of companies. Current offerings include estimated fitness charts, power curves, and dashboards to view statistics from each ride. This study aims to use data collected from my intense training cycle between December 28th, 2022 and March 5th, 2023. After organizing and cleaning the data, I explored the dataset using descriptive statistics and basic modeling to identify the days I did interval workouts. Next steps include using more complicated modeling techniques to measure growth and decline in fitness.

Introduction

Between December 28th and March 5th, I trained for an average of 10 hours a week for the collegiate road cycling season. With already two years of somewhat serious training experience, I was familiar with the methods and demands of reaching a high level of cycling fitness. As discussed in my prior presentation, fitness is determined by the interaction between three main factors: training, diet, and recovery. This study aims to focus on the data collected throughout the training process.

Over my years of research, training is mostly focused on the dichotomy between intensity and volume. Each cyclist tries to optimize performance with different

distributions of intensity within the time constraints of their life. Previous studies done by physiologists like Stephen Seiler have revealed that most world class athletes will spend the majority of their training volume at a low intensity, and the rest of the time at a very high intensity without much in between. This distinct difference between training intensities led him to coin this training philosophy as “polarized training.” More precisely, he observed that the best endurance athletes would spend about 80% of their time at low intensity and 20% of their time at very high intensity.

Further research has attempted to identify more specific ranges of intensity called zones. Traditionally, zones have been measured through “Rate of Perceived Exertion” or RPE. Essentially, it’s a scale from 1-10 where the athlete subjectively rates how hard they are working. For obvious reasons, this isn’t very accurate. Perceived intensity can vary day to day depending on diet, previous training, sleep, or stress. Thus, other physiological markers were sought out to measure intensity, one being heart rate. The technology used to measure heart rate during exercise has revolutionized training, but it has a few flaws. For instance, it can vary depending on temperature, stress levels, or sleep. Additionally, for exercises at higher intensities, there’s a lag between starting an interval and heart rate responding to it. Now, the most important cycling metric is power. The commercial availability of power meters has allowed both professional athletes and amateurs to measure their precise power output at a given second.

Power has allowed researchers to develop models for athlete-specific zones of power. For example, Dr. Andrew Coggan developed the seven zone model according to specific percentages of power and heart rate. The table below clarifies this (1).

Zone	Reference	% FTP	% Max HR	% Threshold HR	RPE*
1	Active recovery	<60%	50-60%	<68%	<3
2	Extensive aerobic	55-75%	60-70%	68-85%	3-4
3	Intensive aerobic	75-90%	70-80%	85-95%	5-6
4	Lactate threshold	90-105%	80-90%	95-105%	7
5	VO2max (aerobic capacity)	105-120%	90-100%	>105%	8
6	VLamax (anaerobic capacity)	120-130%	N/A	N/A	9
7	Neuromuscular power	>130%	N/A	N/A	10

The purpose of specifying ranges of power is to find the perfect prescription of exercise for a cyclist to maximize their fitness within a period of time. Herein lies the importance of systematic data collection. With enough historical data, theoretically, it is possible to develop an athlete specific training program to maximize results. By analyzing my data, I'm hoping to evaluate the efficacy of the work that I did, and, when I'm ready to begin training again, I want to use these lessons to further optimize my routine.

Contents

Data Collection

Cycling data is collected through a variety of methods. Heart rate is measured through a strap attached to my chest. There were multiple methods to measure power. When I began my training cycle, I mostly trained indoors to avoid the cold weather. I first started riding on a Peloton spin bike, and then later switched to a cycling smart trainer. A smart trainer is a device that attaches to my bike and simulates climbing and

measures power output. When the weather got better, most of my training was done outdoors, and power was measured using power meter pedals on my road bike. Once these indoor and outdoor activities were recorded, they were uploaded onto an athlete social media platform called Strava. This is where I downloaded the data files for my activities and saved them to my computer.

An issue with the activity data is that they were stored in different file types depending on the method of recording. Exercise files, whether running, cycling, or swimming, can be recorded in three forms: GPX, TCX, or FIT. GPX or TCX files are an older file type and are XML files. FIT files were created by Garmin and contain more information.

Because different activities were stored in different files, I had to find multiple methods to read these files into RStudio. To read TCX files, I used the readTCX() function from the trackeR package. To read FIT files, I used a package called FITfileR (2).

In the end, I was left with 46 objects corresponding to each of the activities I completed during my training cycle.

Data Cleaning

Once the data was read into R, a major issue was dealing with the different objects. The Peloton TCX files were stored as data frames. The trainer and outdoor rides were stored as tibbles–essentially a list of data frames. An important distinction between the outdoor and indoor tibbles is that outdoor rides would have multiple records for each time I paused or unpaused the activity (such as when I'd stop at a light). Below are figures of the three different cases.

```
> head(readTCX("/Users/drewlaird/Documents/Data Science/Cycling/Data/Peloton/4.tcx"))
                 time latitude longitude altitude distance heart_rate    speed cadence_running
1 2022-12-30 21:12:29       NA        NA       NA     6.12        134 0.000000              NA
2 2022-12-30 21:12:30       NA        NA       NA    14.40        134 4.140000              NA
3 2022-12-30 21:12:31       NA        NA       NA    22.80        133 5.560000              NA
4 2022-12-30 21:12:32       NA        NA       NA    31.19        133 6.267500              NA
5 2022-12-30 21:12:33       NA        NA       NA    39.40        133 6.656000              NA
6 2022-12-30 21:12:34       NA        NA       NA    47.74        134 6.936667              NA
  cadence_cycling power temperature
1              95   141          NA
2              94   139          NA
3              95   141          NA
4              95   141          NA
5              95   141          NA
6              96   144          NA
```

*a data frame from a TCX file

```
> head(records(readFitFile("/Users/drewlaird/Documents/Data Science/Cycling/Data/ZwiftOutside/19.fit")))
# A tibble: 6 × 15
  timestamp           position_lat position_long distance time_from_course compressed_speed_distance
  <dttm>                     <dbl>         <dbl>    <dbl>            <dbl> <list>
1 2023-01-28 00:22:28        -11.6          167.     4.01         2147484. <int [3]>
2 2023-01-28 00:22:29        -11.6          167.     5.85         2147484. <int [3]>
3 2023-01-28 00:22:30        -11.6          167.     7.92         2147484. <int [3]>
4 2023-01-28 00:22:31        -11.6          167.    10.3          2147484. <int [3]>
5 2023-01-28 00:22:32        -11.6          167.    13.1          2147484. <int [3]>
6 2023-01-28 00:22:33        -11.6          167.    16.3          2147484. <int [3]>
# ℹ 9 more variables: heart_rate <int>, altitude <dbl>, speed <dbl>, power <int>, grade <dbl>,
#   cadence <int>, resistance <int>, cycle_length <dbl>, temperature <int>
```

*a tibble from an indoor ride, note that there is only one tibble.

```
> records(readFitFile("/Users/drewlaird/Documents/Data Science/Cycling/Data/ZwiftOutside/20.fit"))
$record_1
# A tibble: 18 × 10
   timestamp           position_lat position_long distance accumulated_power altitude speed power
   <dttm>                     <dbl>         <dbl>    <dbl>             <dbl>    <dbl> <dbl> <int>
 1 2023-01-28 20:56:29         37.3         -76.7   79108.           1775292     22.2 0         0
 2 2023-01-28 20:56:30         37.3         -76.7   79108.           1775292     22.2 0         0
 3 2023-01-28 20:56:31         37.3         -76.7   79109.           1775292     22.2 0         0
 4 2023-01-28 20:56:32         37.3         -76.7   79109.           1775292     22   0.504     0
 5 2023-01-28 20:56:33         37.3         -76.7   79110.           1775292     22   1.21      0
 6 2023-01-28 20:56:34         37.3         -76.7   79112.           1775292     22   1.22      0
 7 2023-01-28 20:56:35         37.3         -76.7   79114.           1775292     21.8 2.01      0
 8 2023-01-28 20:56:36         37.3         -76.7   79116.           1775292     21.8 2.01      0
 9 2023-01-28 20:56:37         37.3         -76.7   79119.           1775292     21.8 2.26      0
10 2023-01-28 20:56:38         37.3         -76.7   79122.           1775292     21.6 2.86      0
11 2023-01-28 20:56:39         37.3         -76.7   79125.           1775292     21.6 2.86      0
12 2023-01-28 21:15:05         37.3         -76.7   86408.           1923392     23.4 0         0
13 2023-01-28 21:15:06         37.3         -76.7   86408.           1923392     23.4 0         0
14 2023-01-28 21:15:07         37.3         -76.7   86408.           1923392     23.4 0         0
15 2023-01-28 21:15:08         37.3         -76.7   86408.           1923392     23.4 0         0
16 2023-01-28 21:15:09         37.3         -76.7   86408.           1923392     23.4 0         0
17 2023-01-28 21:15:10         37.3         -76.7   86408.           1923392     23.4 0         0
18 2023-01-28 21:15:11         37.3         -76.7   86408.           1923392     23.4 0         0
# ℹ 2 more variables: heart_rate <int>, temperature <int>

$record_2
# A tibble: 2 × 17
  timestamp           position_lat position_long distance accumulated_power altitude speed power
  <dttm>                     <dbl>         <dbl>    <dbl>             <dbl>    <dbl> <dbl> <int>
1 2023-01-28 18:12:13         37.3         -76.7     794.             12659     18.2 10.3      0
2 2023-01-28 18:15:23         37.3         -76.7    2400.             48488     17.2  4.57     0
# ℹ 9 more variables: heart_rate <int>, cadence <int>, temperature <int>, left_right_balance <int>,
#   left_torque_effectiveness <dbl>, right_torque_effectiveness <dbl>, left_pedal_smoothness <dbl>,
#   right_pedal_smoothness <dbl>, fractional_cadence <dbl>
```

*a tibble from an outdoor ride, note that there is more than one record in this tibble.

On top of the different file types, there were also different variable names between the TCX and FIT files, as well as variables that I wasn't interested in. For example, I did not need data like temperature or latitude and longitude.

As for the ranges of values, I noticed that there were times when my power meter would register an impossibly high number, so these entries had to be imputed with the mean power in order to correct the data. I also made sure to delete any potential rows with NA values.

Thus, the cleaning process meant I needed to consolidate each activity into a data frame, standardize all the column names, and remove nonsensical values. The cleaning functions I used are below:

```
#clean Peloton TCX
clean.peloton = function(path) {
  #feature engineer
  subset = subset(readTCX(path), select = -c(longitude, latitude, altitude,cadence_running,temperature))
  clean = data.frame(subset, altitude = rep(0,nrow(subset)))
  #fix names
  colnames(clean)[5] = "cadence"
  final = add.metrics(na.omit(clean))
  return(final)
}
```

```
#clean fit file
clean.fit = function(path) {
  data = records(readFitFile(path))
 #turn tibbles into data frames
   if (length(data) == 1) {
      #feature engineer
     clean = subset(data.frame(data), select = c(timestamp, distance, heart_rate, altitude, speed, power,
cadence))
  }
  else {
    #combine tibbles into one tibble
    combined = data %>% bind_rows() %>% arrange(timestamp)
    #feature engineer
    clean = subset(data.frame(combined), select = c(timestamp, distance, heart_rate, altitude, speed,
power, cadence))
  }
  #change names
  colnames(clean)[1] = "time"
  #fix broken power values with imputed mean
  clean$power[clean$power == 65535] = mean(clean$power[clean$power != 65535])
  #feature engineer
  final = add.metrics(na.omit(clean))
  return(final)
}
```

Then, when feature engineering, I also created an additional column to identify what power zone I was in at the time. I did this with the function in the figure below. I also calculated change in power and change in heart rate over each second, and the ratio between heart rate and power output.

```
#zone detection
zone.test = function(val) {
  zone = NA
  if (val < 155) {
    zone = 1
  } else if(val >= 155 && val < 200) {
    zone = 2
  } else if(val >= 200 && val < 245) {
    zone = 3
  } else if(val >= 245 && val < 290) {
    zone = 4
  } else if(val >= 290 && val < 380) {
    zone = 5
  } else if(val >= 380) {
    zone = 6
  }
  return(zone)
}
```

In the end, I considered the following variables for each activity:

- Time
- Distance
- Altitude
- Heart rate
- Power
- Speed
- Cadence
- Change in heart rate
- Change in power
- Heart rate/power ratio
- Zone

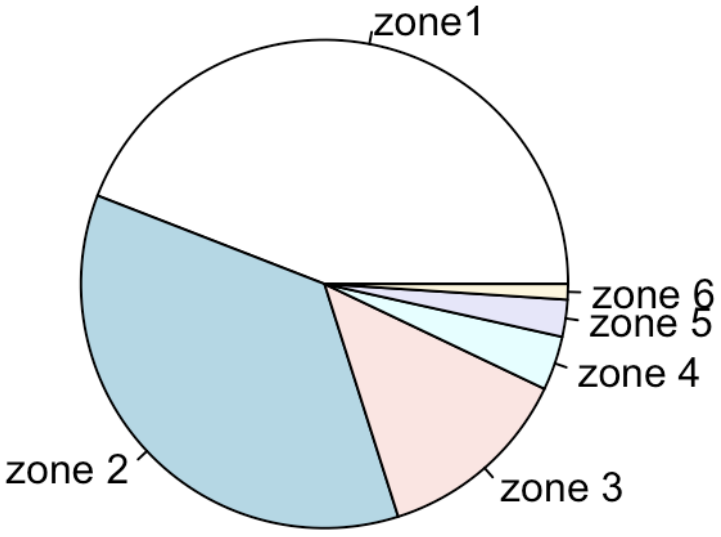
Below is an example of the data collected for an activity, each observation was collected for every second of the activity. Distance and other associated metrics were measured in meters, heart rate was in beats per minute, power was in watts, and cadence was in revolutions per minute. In total, I had 308,623 observations (or seconds) totaling to about 84 hours worth of activity.

```
> head(SD25)
                 time distance heart_rate altitude  speed power cadence     ratio d_heart d_power zone
1 2023-02-07 21:26:25     0.00        157     29.8  9.723   125      86 0.7961783       0       0    1
2 2023-02-07 21:26:26    10.29        157     28.0  9.835    27      86 0.1719745       0     -98    1
3 2023-02-07 21:26:27    20.63        157     28.4  9.872    27       0 0.1719745       0       0    1
4 2023-02-07 21:26:28    30.82        156     28.6  9.928    28      29 0.1794872      -1       1    1
5 2023-02-07 21:26:29    40.95        156     28.4 10.012   154      29 0.9871795       0     126    1
6 2023-02-07 21:26:30    51.13        155     28.2 10.012   154      83 0.9935484      -1       0    1
```

Data Analysis & Modelling

The first step towards getting the data ready for analysis was organizing the activity data frames into a list, and then sorting the list chronologically. I did this by arranging the list in terms of whatever the date and time was of the first entry of each data frame. Doing this allowed me to iterate through each of the data frames chronologically whenever I performed any kind of operation.

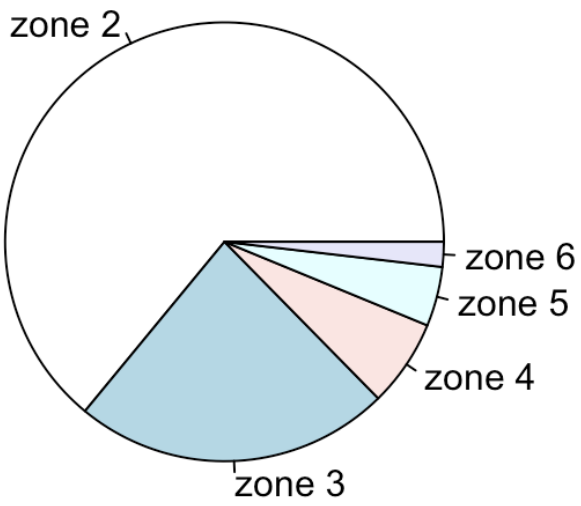
Next, I was curious what my overall zone distribution was after my training cycle. Because I included a column for the zone of power I was in, I could simply sum up the number of appearances of each zone in all the data frames, and divide it by the total number of observations. This would give me the percentage of time spent at each zone. After doing this, I organized it into a pie chart and got the results below.



What worked out nicely is that I spent 79.89% of my time in zone 1 and zone 2 (low intensity) and thus around 20% of my time at higher intensities. This would indicate almost textbook polarization. However, this isn't entirely accurate because I also specifically spent a lot of time at zone 3 at an intensity called sweetspot–a specific intensity that has shown to be highly effective at increasing fitness among more time-crunched athletes.

Another curious fact I noticed about this was that zone 1 took up the majority of the time (~44%). I deduced that this could be for two reasons: 1) as I got fitter, my zones increased, so what used to be zone 2 at the beginning became my zone 1 as I became faster and 2) any time I coasted, I was technically in zone 1 (which is anything less than 155 watts).

Therefore, I thought it might be more worthwhile to consider the zone distribution of my exercises not including my zone 1 time. When I did this, I found the following pie chart below.



In this case, zone 2 accounted for 64% of the time and zone 3 accounted for 23% of the time. This made more sense to me because I specifically designed my training to have more time at zone 3 than what traditional polarized training recommends. I spent the rest of the time (~13%) at the high intensity zones–zone 4, 5, and 6. Looking at my training subjectively, I think doing more middle intensity resulted in better performance gains than I had ever experienced in my life. At the same time, though, this could be due to other factors such as honing in on my diet, sleeping more, weight lifting, or simply being older and more physically developed.

As discussed previously, on a high level there are two types of activities in a training cycle: low intensity (zone 2) and high intensity (anything zone 3 and above). Each workout I knew what intensity I was targeting beforehand, so I manually went through my recorded activities and created a vector that indicated whether it was a high intensity (denoted by a value of 1) or low intensity (0) workout.

After that, I gathered summary statistics for each activity to use as variables in a classification model. I calculated the following variables:

- Average power
- Average heart rate
- Average ratio
- Variance of power
- Variance of heart rate
- Variance of ratio
- Percent of time spent at each zone (out of 100)

A sample of my data is as follows:

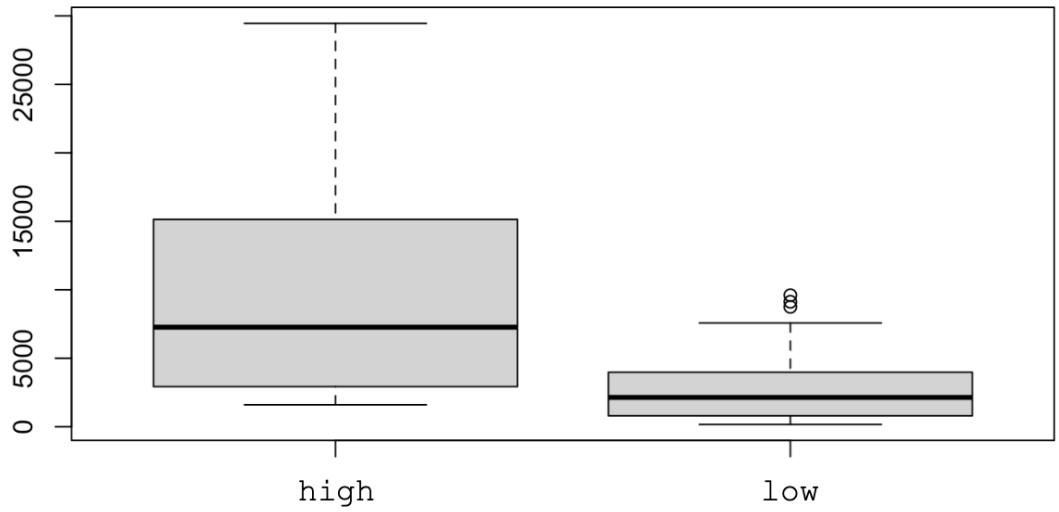
avg_pwr	avg_hr	avg_ratio	var_pwr	var_hr	var_ratio	zone1	zone2	zone3	zone4	zone5	zone6	intensity
133.7428339	151.6143187	0.879553346	605.432362	45.76821343	0.02005909992	95.27238147	2.295883711	2.092107051	0.339627768	0	0	0
137.3351016	150.1085779	0.914542886	295.8534917	19.24501051	0.01117903231	98.41986456	0.835214447	0.4514672686	0.02257336343	0.2708803612	0	0
141.6164858	150.2847118	0.9418997025	339.3924371	34.19917862	0.01255470648	97.20247006	1.787050898	0.1403443114	0.8701347305	0	0	0
165.8583141	162.906226	1.002052956	2344.353318	217.6319013	0.05597704221	38.68818857	20.34332565	40.81475788	0.1537279016	0	0	1
148.8210905	150.1739871	0.9913574419	167.6394832	21.27390056	0.007963839208	85.034078	14.91859144	0.04733055661	0	0	0	0
169.8112959	159.0934283	1.055183805	1940.390728	187.7589563	0.04932606748	36.84349433	21.6152019	41.54130377	0	0	0	1
149.6912921	151.30764	0.9885747485	424.9101271	22.73651347	0.01705100496	74.19667993	25.45383943	0.00970779536	0.00970779536	0.3300650422	0	0
150.0588871	158.3252296	0.9245648895	7109.824146	269.2430218	0.2088339701	70.15126958	10.56185845	10.10264722	6.537007023	1.782820097	0.8643976229	1
146.8714266	157.2158542	0.9235333686	3893.209295	165.7206657	0.1253933976	67.69783179	16.71039912	13.25783732	0.7319431018	0.8424250794	0.7595635962	1
155.7291013	147.0559144	1.059103171	4072.135164	130.0432001	0.1466970591	46.76139509	51.72541059	0.6089684444	0.03690717845	0.1291751246	0.7381435689	0
168.6563229	151.7377578	1.110326772	2081.068445	116.8013279	0.07305946937	20.44843049	67.83856502	9.883408072	0.7533632287	0.8609865471	0.2152466368	0

Next, I decided to use a logistic regression model to make a binary prediction on whether an activity was high or low intensity. Because there were only 45 observations, I did not want to use more than three or four variables, otherwise there may be an overfitting issue. After testing multiple models, I settled on using the variables var_ratio, avg_hr, and var_pwr to evaluate intensity. It yielded the model and confusion matrix below.

```
Call:
glm(formula = intensity ~ var_ratio + avg_hr + var_pwr, family = "binomial",
    data = intensity_data)

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-8.946e-05  -2.100e-08  -2.100e-08   2.100e-08   9.285e-05

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.285e+03  4.037e+05  -0.003    0.997
var_ratio   -2.752e+03  8.547e+05  -0.003    0.997
avg_hr       7.959e+00  2.551e+03   0.003    0.998
var_pwr      1.226e-01  3.569e+01   0.003    0.997

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6.0571e+01  on 44  degrees of freedom
Residual deviance: 2.5465e-08  on 41  degrees of freedom
AIC: 8

Number of Fisher Scoring iterations: 25
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 27  0
         1  0 18

               Accuracy : 1
                 95% CI : (0.9213, 1)
    No Information Rate : 0.6
    P-Value [Acc > NIR] : 1.039e-10

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0
            Specificity : 1.0
         Pos Pred Value : 1.0
         Neg Pred Value : 1.0
             Prevalence : 0.6
         Detection Rate : 0.6
   Detection Prevalence : 0.6
      Balanced Accuracy : 1.0
```

An important note is that none of the coefficients were significant. This is due to the multicollinearity between the variables (ratio is calculated by dividing heart rate by power). This isn't an issue, though, because the predictions are 100% accurate regardless of whether the interpretations make sense. 100% accuracy is an amazing result, which made me skeptical, but when you look at the difference in power variance between the two groups in the boxplot below, there is an obvious distinction between the groups which would lead to this perfect accuracy.



The benefit of having a model like this is that as I continue to train, I can use the activity values for power variance, average heart rate, and ratio variance to automatically identify high and low intensity workouts so I don't have to do it manually.

Conclusion

While the data analysis provided some interesting tools and insights, there is still room for continued exploration. A much more complicated phenomenon I was interested in observing was periods of improving, stagnating, and decreasing fitness. If I could quantify the quality and effects of my training, it would allow me to tailor a training plan to optimize my routine. To truly improve at something, you need timely feedback on the quality of your work, and the problem is that with cycling there's so much delay and so many variables that it is difficult to tell with just the eye test. I currently believe that there are three possible solutions for modeling fitness.

From my experience with cycling, my intuition about fitness is that the better shape I'm in, the more power I should be able to produce at the same heart rate. This is

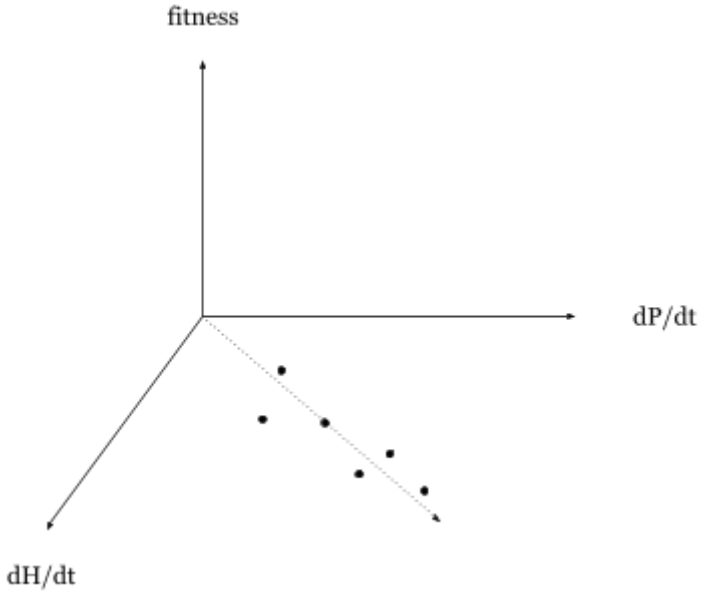
easy enough to calculate, but this is too small of a data point to encapsulate fitness and it also depends on the type of activity I was doing.

After my discussion during class, I want to investigate the relationship between decreases in power and decreases heart rate. Theoretically, if you're not producing as much power (not working as hard), your heart rate should respond by reaching a state closer to rest. In other words, heart rate should decrease when power decreases. The more fit you are, the faster your heart rate could reach rest levels, especially after large amounts of strenuous exercise.

To model this relationship, it could potentially be done graphically, where fitness is a value determined by a function of decrease in heart rate and decrease in power.

$f(dh,\ dP)\ =\ \ldots$

Theoretically, "perfect" fitness would be instantaneous heart rate decreases relative to decreases in power. The following picture from class is the current way I'm trying to visualize this phenomenon.



The issue with this approach is that I need to define some function that peaks across the line dP/dt = dH/dt. However, the range of the values of fitness is difficult to determine. Should it be 0 to 100? This approach will require more research in topological curves.

After talking with Professor Hunt and Sasinowska, another idea I'm considering is using time series analysis to look at the data. For each activity, all of the observations are taken at each second. If I want to model heart rate responsiveness to changes in power, I could evaluate time series cross correlation with covariates. Another issue is that heart rate decreases generally lag behind power decreases, thus, I'd need to model the relationship with lags as well. To do this, I am currently planning on teaching myself time series analysis to find the right model. This problem may be analogous to trying to find relationships between stock prices and other measurable phenomena.

The last method I came up with Professor Vinroot was aggregating professional cyclist data to use as a benchmark to my own performance. Using the same social media service I upload all my activities, I can find the heart rate and power data of the best athletes and construct a curve relating their changes in heart rate and power. Then, I can perform a Riemann sum for each second to calculate the total distance between my power/heart rate curve and a professional's. Using the average of many professional cyclists would allow me to evaluate how far I am, or how far anybody is, from the current genetic and technological ceiling. The fitter I got, the more the Riemann sum would decrease.

Regardless of whatever method I use (or perhaps if I used an ensemble of methods), I need to acquire new knowledge to create implementations for these ideas. Part of this process will also be reaching out to experts in the field as well as using a mix

of online resources and books to acquire the knowledge to construct these heart rate and power curves.

References

1. Dr Emma Wilkins & Tom Bell. (2022, August 12). *Cycling training zones: A detailed guide*. High North Performance. https://www.highnorth.co.uk/articles/cycling-training-zones
2. Smith, M. L. (2022, May 17). *Reading fit files*. FITfileR. https://msmith.de/FITfileR/articles/FITfileR.html