University of Strasbourg

Master 1 CSMI

Data analysis, modeling and simulation of professional road cyclist

Laurène Guidet
Céline Lorentz
Jimmy Mousel
Colin Holler
Mélissa Untz

Supervisor : M. Prud'homme Christophe
Supervisor : M. Mazenot Olivier

Academic year 2020-2021

Contents

1 Introduction.

1.1 General context.

Since every gain is potentially synonymous with a competitive advantage, everything can be measured and analysed. And this goes much further than you might think. Data Science is a hot topic that is impacting a range of diverse areas from business to sport. With so many cyclists collecting and uploading their data, there is plenty of raw material from which to draw interesting insights. Today, the analysis of heart rate, power or pedaling frequency is one of the basic parameters of any cycling training program. It is now possible for coaches to accurately analyse the performance of professional athletes in order to determine their strengths and weaknesses and to exploit their full potential. In particular, this will allow us to **analyse** various cyclists' trainings and even to **model** them in order to be able to give better decisions and optimise their trainings. We will work on this subject in our project and internship who are a practical application of the knowledge acquired during our first year of CSMI (Scientific Computing and Mathematics of Information) Master's degree, at the UFR Mathématiques-Informatique in Strasbourg.

Project.

The subject is proposed by Olivier Mazenot, data scientist working for the French cycling team Groupama-FDJ. His mission is to analyse and exploit the data collected during cyclists' training. So, our project is in cooperation with the Groupama-FDJ Cycling Team.

Internship.

The subject of Untz Mélissa's internship was proposed by the Cemosis laboratory, in collaboration with the Groupama-FDJ cycling team. This is the continuity of the project described in sections **4** and **5**. Part **5.3** on the comparison between real and simulated data was completed in collaboration with Mousel Jimmy. However, the main focus was on the nutritional aspect of the Pulse software and on cycling nutrition. The objectives of the internship and the results of research can be found in section **6**.

1.2 Companies presentation.

Groupama-FDJ Cycling Team.

At the beginning of the 1997 season, thanks to the passion of Marc and Yvon Madiot, the cycling team "Française des Jeux" was born. 2018 marks a turning point in the team's history, with the arrival of Groupama, the new title partner. The association of these two major players in French sport gives birth to the **Groupama-FDJ Cycling Team** [14].

Figure 1: Groupama-FDJ logo (Source : Groupama-FDJ Equipe Cycliste [14])

The current team leaders are Thibault Pinot (climber) and Arnaud Démarre (sprinter). In 24 years, the team has won more than 500 victories, including 11 French Champion jerseys, 3 cycling Monuments and 44 stages of the Grand Tours. The team is part of the UCI WorldTour and therefore among the 19 best teams in the world. In particular, in 2020 it was the 9th world team and the 1st French team. Moreover, recently David Gaudu, a climber from Groupama-FDJ, won the sixth and last stage of the Tour of the Basque Country.

Cemosis.

Figure 2: Cemosis logo (Source : Cemosis [67])

Cemosis (Centre for modeling and simulation in Strasbourg) is the technological platform in mathematics at the University of Strasbourg. It was created in January 2013 and is hosted by the **IRMA** (Institute of Advanced Mathematical Research). Cemosis works on modeling simulation, in data science or high performance computing [67].

1.3 Data from Groupama-FDJ.

They use data from GPS computers. In particular, at each competition and each training session, data is acquired thanks to Garmin computers and sensors. About fifteen variables are recorded by the computers every second: elapsed time, GPS coordinates, speed, altitude, slope, temperature, power, cadence, heart rate. This gives us almost 3 billion values per year for the team. There is also perceptual data on runners (scale /100): perceived difficulty, feeling of performance, feeling of exhaustion or sleep. Moreover, there is also the possibility of having runners' follow-up data: weight evolution, HRV tests (cardiac variability to measure a state of form), medical and biological data (blood tests).

1.4 Explanation of project subjects.

We can then briefly present our project. The objective of the project is to study existing data and to explore software compatible with these data. It is therefore divided into two interdependent themes :

Project 1: The objective is to **measure** and **model** the impact of different **factors** on the **performance** of the cyclist in competition. Performance is measured using the athlete's power data, as this is the only reliable measure. It can be influenced by many environmental, physical and moral factors, which we will study. The aim of this work would be to find a measure of performance that is not correlated with the duration of the activity or the course. To do this, we will have a large amount of cyclist data stored in .csv.

Project 2: The aim is to use and complement the existing physiological models on the PulsePhysiology Engine framework to **model cyclist physiological effort** from our data. Among the available models, we will focus on the **cardiovascular system** and the **energy system**. We will understand the parameterisation of these systems so that they can be modified and used with different physiological data. This will allow us to integrate them into virtual models, which will be useful tools for to understand the impact factors.

2 Project management.

2.1 Road map.

Our road map defines what we want to achieve. It combines our **issues**, **milestones** and **projects**.

The **issues** define the various tasks to be carried out during the project. They are assigned to the different collaborators of the project in question. Everyone must therefore know what their job consists in during the project. Consequently, efficient and precise collaboration between different collaborators is one of our objectives.

Then come the **milestones**. Their purpose is to represent a kind of sub-objectives to be achieved in order to have some stages in our projects. In fact, we can easily link our issues with our milestones on GitHub. This has the advantage of being able to easily check the different issues assigned to the milestone by clicking on it, and thus have a clear view on these stages. Consequently, we have set up milestones along the entire length of the projects.

Next we created **projects**, which group together milestones and their associated issues. The first is on the subject of **Performance Factors** and the other on **Data informed physiological models**. These contain different columns including the issues related to the milestones. So for example we have a column «V0 (In progress)» with all the issues to be achieved in relation to this project.

Furthermore, these features allow us to implement the Kanban Method and thus have automated columns. Consequently, when a new issue is created, it is automatically added to the «V0 (In progress)» column, and then once this issue is closed, it will be automatically moved to the «V0 (Done)» column which collects the completed issues. This same implementation is done for the versions V1 and V2.

Finally, in order to have an overview of all the activities and issues related to our projects, we have set up a **Gantt chart**. This, and our objectives, are obviously subject to changes and adaptations over time. In fact, we can't predict for sure how and how fast we will progress in our projects.

In order to achieve all this, we need to use different tools. Firstly, we use as host platform **GitHub** where all our files and codes will be stored and manipulated. For the data project, we program in **python**, which nowadays is a strong programming language for manipulating data. Then, the project related to the physiological models consists in exploring and adapting the **Pulse** platform, which will be explained in a following section. Not to forget that we communicate between collaborators on different platforms such as GitHub, Slack and Zoom.

2.2 Update of Objectives and Road map.

It should be noted that the first assignment of issues for this project took place on April 6th and that the defenses are scheduled for the 27th and 28th of May.
However, at the beginning of May, we realised, together with our supervisors, that the objectives we had set for the **Performance factors** project were a little too ambitious in view of the data at our disposal. In fact, we would need to have all the data from several riders over several seasons to be able to achieve these. However, we did get some other data which will be discussed in more detail in sections **3.1** and **3.2**. The two new objectives for this project are thus to **quantify the level of performance/intensity** for each of the activities, and to **plot different performance level curves** of the cyclist and to **describe** the **observations**.
But also for the project on **Data informed physiological models** we had to change and adapt the work plan. In fact, we also recognised that we are limited with the data we have, as there are many models on Pulse which have many parameters. Thus, we continued to explore the existing possibilities with Pulse. Therefore, we set up different descriptions, for example of the nutrition and environment parameters, explaining the implementation of these, as well as explaining different scenarios. However, we managed to get more data on a specific cyclist, and so we were able to simulate the effort of a cyclist over a certain period of time (section **5**).

This change of objectives leads us to adapt our road map. To do this, we will delete, modify and create new issues that will be adapted to our new objectives. In addition, our Milestone regarding the V1 version of our projects has been moved in time. In order to get a clearer idea of the different changes, we will use visualisations of our Gantt chart before and after these changes.

Let us first look at our initial Gantt chart shown in figure **3**.

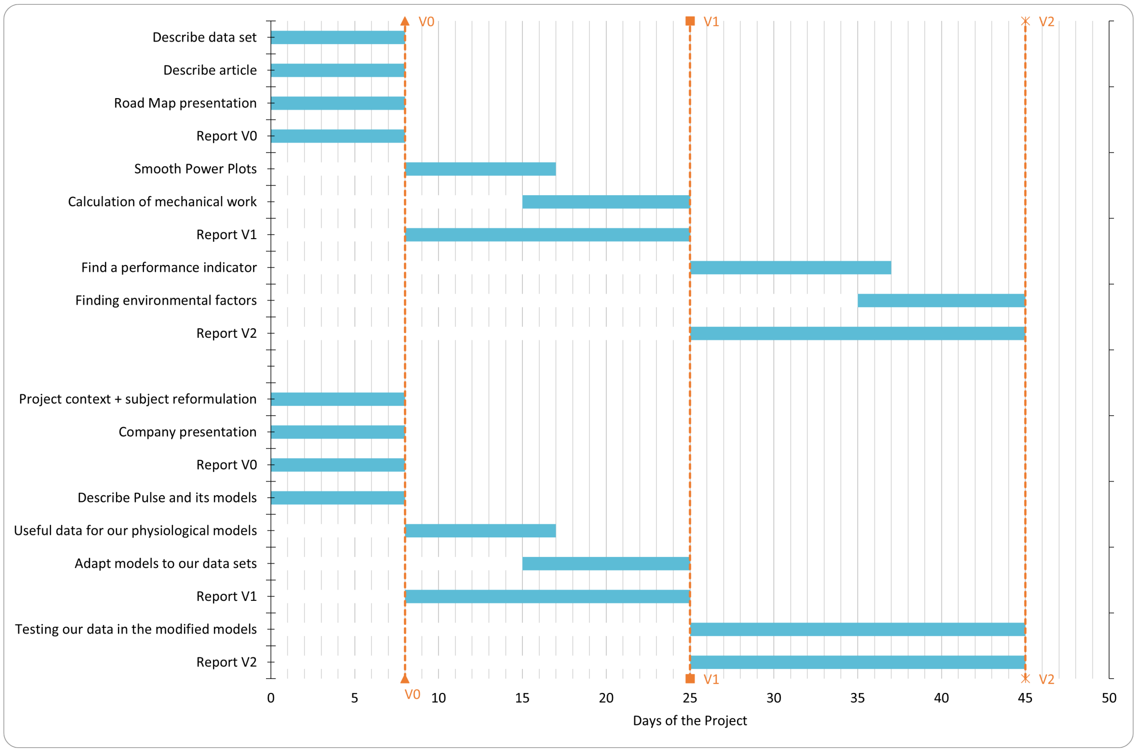

Figure 3: Initial Gantt Chart

The first part always represents the issues of the project on **Performance factors** and the second part those of the project on **Data informed physiological models**.

Note that our current Gantt chart is accessible via a link located in the «README» section of our repository. This is a link to the Google sheet containing the chart and the distribution of issues, which can also be seen on GitHub.
After the above mentioned changes, our new Gantt chart can be seen on figure **4**.

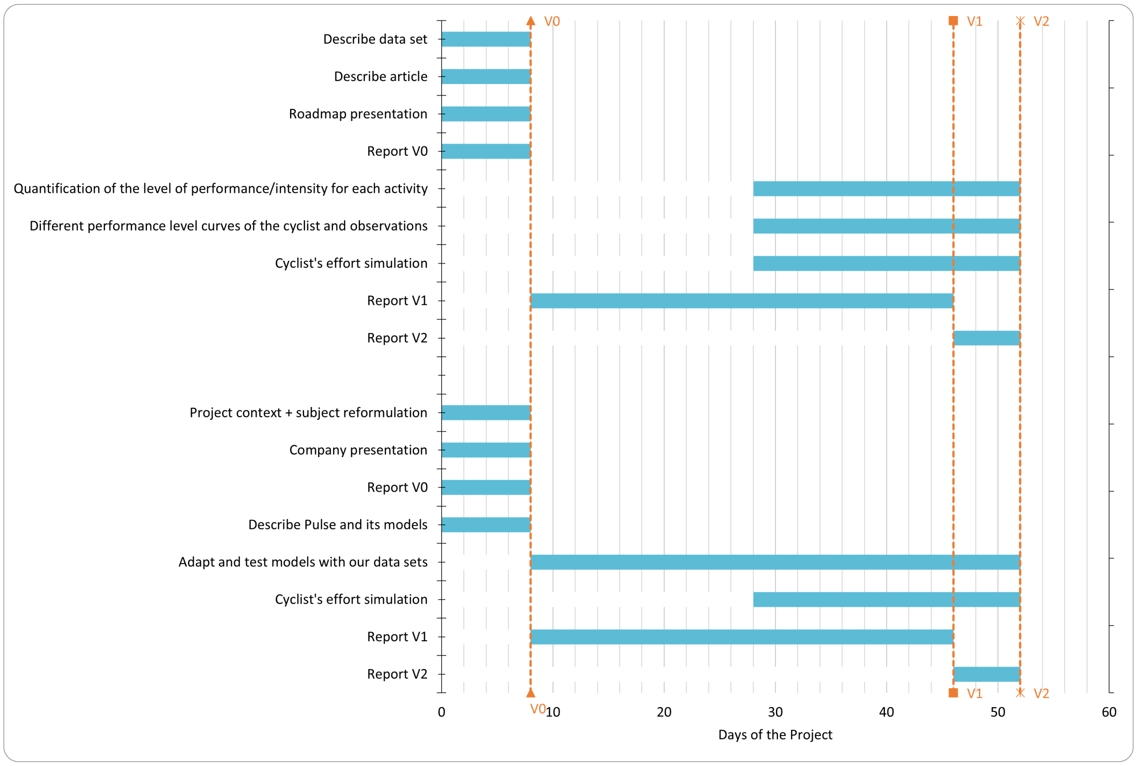

Figure 4: Current Gantt Chart

It is easy to see the new issues that have been implemented, which we began around the 4th of May, as well as the shift in time from the V1 version which now is one week ahead of the last version.

2.3 GitHub.

In order to have a clearly arranged GitHub repository, we created different folders and files so that we could work as efficiently as possible.
First of all, we created a branch named **develop**, where we commit our changes. This branch is intended to somehow secure the **main** branch, where we only want to have content that works without problems. Once all the checks in the **main** branch are done, we can make a pull request of the **develop** branch and push it into the **main** branch.
Thereafter, the repository contains several folders:

- .devcontainer, .vscode, .github, docs and src which contain configurations for the execution of Pulse, but also for python files of the data part.
- V0_Folder, V1_Folder and V2_Folder which contain the files and images used for our different versions of reports, beamers and description sheets.
- data storing different files associated with the various data sets. In fact, we have set up different python notebooks which are used for the **Performance factors** project in order to have more adapted visualisations. But this folder also contains .csv and .json files created the python file which links both projects. We will detail this in the section **5 Cyclist's effort simulation**.

- different files associated with the C++ compilations needed for Pulse, but also files such as .gitignore, README.adoc (containing the link to our Gantt Chart) and the PDF's of the latest version of our report and beamer.

Note that we have already explained the setting up of the issues and projects, which represent important elements of our repository, in the section **2.1 Road map**.

2.4 Internship's management

The researches made during the internships of Lorentz Céline, Guidet Laurène and Untz Mélissa are stored in the same GitHub repository as the project. We created a folder named rapport stage in which the new pictures and the completed report are stored. The attachments are stored in data, the same folder as the one used in the project.

3 Performance factors.

3.1 Data description.

We got a first data set which contains 9 cyclists. We got these data on March the 26th. This data set contains **144 680 lines** which all represent a measure at a different time. We will explain in detail what are the different data which are measured, so we have to explain the **9 columns** :

- the ID of each cyclist, this number is between 0 and 8. It is unknown for us which cyclist of the team corresponds to which ID.
- the num of the file, this number is between 0 and 11 and is linked to the activity of the cyclist, each activity (whether training or competition) has a number.
- the time (in **seconds**), which is the time that has passed since the beginning when a measure is done.
- the distance (in **meter**), which is the number of meters the cyclist has ridden since the beginning of the training.
- the altitude (in **meter**) which is the height of the cyclist in relation to the sea level at the time of the measure.
- the speed (in **meter per seconds**) which is the number of meters the cyclist has ridden per seconds.
- the power (in **Watt**) which is the rate at which energy, expressed in terms of work (in **Joules**), is used.
- the cadence (in **Rate per Minute**) which is the number of pedal revolutions per minute.
- the temperature (in **Celsius degree**) which is the temperature outside at one time.

Let's display the beginning of the data set shown in figure **5**.

coureur	activity	time_seconds	distance	altitude	speed	power	heart_rate	cadence	temperature	n_segment
17	0	0	0	656	0	0	67	0	24	0
17	0	1	0.08	655.8	0	0	67	0	24	0
17	0	12	0.08	649.6	1.978		77	0	23	1
17	0	13	2.08	649.6	2.211	0	76	0	23	1
17	0	14	4.36	649	2.445	0	77	0	23	1
17	0	15	6.88	648.4	2.678	0	78	0	23	1
17	0	16	9.09	648	2.071	0	80	0	23	1
17	0	17	10.84	647.4	1.67	0	84	0	23	1
17	0	18	12.1	646.8	1.418	0	84	0	23	1
17	0	19	12.84	646	1.073	0	84	0	23	1
17	0	20	14.61	645.4	2.071	0	83	0	23	1
17	0	21	17.39	645	3.023	0	82	0	23	1
17	0	22	20.8	644.2	3.602	196	84	27	23	1
17	0	23	24.53	643.4	3.583	156	86	26	23	1
17	0	24	27.88	643.2	3.116	76	89	26	23	1
17	0	25	30.47	642.8	2.062	76	90	26	23	1
17	0	26	31.94	642.4	1.138	76	89	26	23	1

Figure 5: First data set - March 26th

Let's also add some visualisations in the form of histograms, illustrated in the figure **6**.

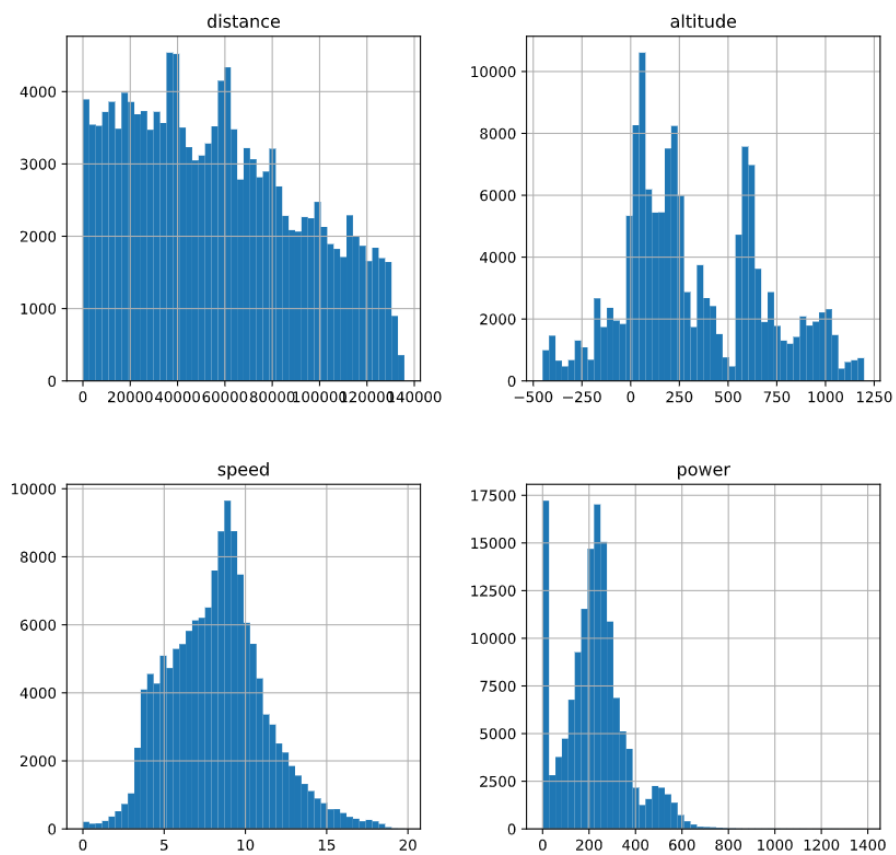

Figure 6: First data set - March 26th

It can be seen that there are generally more trips over short distances than over long distances. For the altitude, we observe that most of the trips are done on attitudes above the sea level, but that there are also some below. Moreover, we can see that they also made trips on higher attitudes, notably at more than 1000 m above sea level. Furthermore, we can see that the speed at which the cyclists ride the most is around 8-9 m/s and that the most often exercised power is around 210-220 W. However, there are also regularly times when they produce more power, which is above 400 W.

3.2 Data sets update.

Then we received a second data set which is more precise and contains more columns. We obtained this data set on April the 9th. It contains more lines (**152 801**). It contains the same columns than before, one modification is that the column numfile is now called activity so that the name is more telling than before. Moreover, two new measures were added :

- the heart rate which is the speed at which the heart **beats per minute**.
- the nsegment which represents the temporal segments per activities.

On May the 11th, we received the two last data sets. These are a bit different in comparison of the others. The first difference is that this new data set is focused only on one cyclist. It is composed of 77 activities : 11 against the clock races and 66 in line races. It is decomposed in two : "data.cycling.4.csv" contains the data which are measured and "data.cycling.4.sensations.csv" contains some information of feelings of the cyclist. First, we can focus on the first data set. It contains **13 columns** and **1 165 931 lines**. 11 columns are in common with the last version ("data.cycling.1.csv") and two more columns were added, which are:

- the position.lat which are the coordinates of the latitude of the place where the cyclist is.
- the position.long which are the coordinates of the longitude of the place where the cyclist is.

Let's do some more visualisations here to get a clearer idea of the data we are using. These can be seen in the figure **7**.

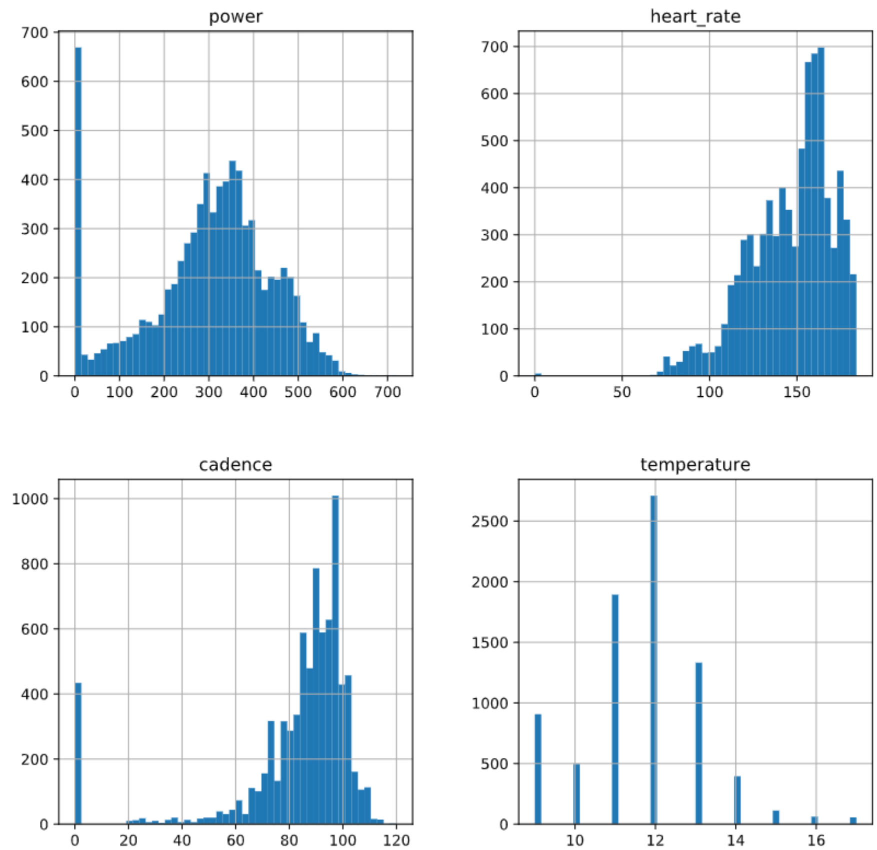

Figure 7: Newest data set - May 11th

It is noticeable that the power values are more distributed than in the previous histogram. The heart rates are mostly around 170 bpm, while the cadences rarely exceed 100 rpm. Finally, it can be observed that cyclists ride most of the time at an ambient temperature of 12°.

Then the second data set contains the feelings of the cyclist. It has 7 columns which are :

- the date which is the date of the activity.
- the type of activity which is whether an in line race or an against the clock race.
- the difficulty feeling which is the percentage of difficulty of the activity evaluated by the cyclist.
- the performance feeling which is the percentage of performance that the cyclist think has done.
- the exhaustion feeling which is the percentage of exhaustion that the cyclist feels on his body.
- the sleep which is the percentage of sleeping that the cyclist has done the night before the activity.
- the activity which is the number of the activity, this column is the common key between the feeling data set and the other. So we can switch from one data set to the other through this column.

In the figure **8** will be represented these data concerning the sensations of the cyclist.

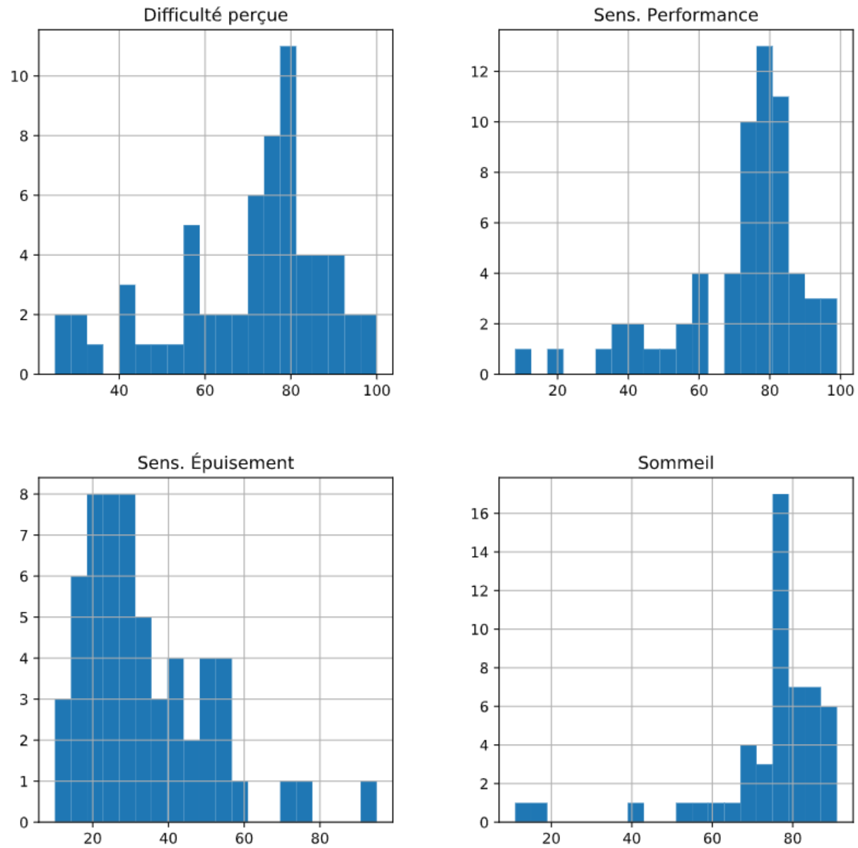

Figure 8: Sensations data set - May 11th

Most often, cyclists report percentages above 70% for perceived difficulty. The same is true for the cyclists' feeling of performance. However, we note that their feelings of exhaustion are much lower. They generally seem to stay below half of the maximum exhaustion feeling. Finally, the sleep percentages indicate that the cyclists claim to have had enough sleep before starting the different rides.

Note that all our data is stored in the "data" folder on GitHub.

3.3 Quantify the level of performance for each activity

3.3.1 Introduction of some notions

First of all, we will explain some notions that will be useful later. We have the record power for different durations, which represents the maximum power that the cyclist can deliver on a certain period of time. Moreover, we will look at the power data after different work done (which correspond to the integral of the power curve along the time divided by the weight : $\omega = \int_I P\,\mathrm{d}t$ with ω the work et $C = \frac{\omega}{p}$ with C the work done level and p the weight of the cyclist), it allows us to measure the cyclist's performance after a certain level of fatigue. Finally, we will make a short summary of the article given by our tutor:
this article [1] deals with the factors which can influence the MMP (Maximal Mean Power) and then the performance of cyclists. First, they separate cyclists in 2 groups: climbers and sprinters, and then we divide each group in category 1 and category 2 according to the races results of the athlete (the best athletes are in cat 1). In fact, the profile climber or sprinter influences the MMP : climbers can produce high MMP during

a long time whereas sprinters have a higher MMP in a short time. Moreover, cat 2 sprinters are generally less powerful than the athletes of cat 1, and we have the same results for the climbers. The decline of MMP can be explained by different factors like mental fatigue, nutrition or hydration intake, muscular fiber type or fatigue resistance which we will study in a future work. The limits of this study is that they take different brands of power meter and the zero-calibration (the calibration of the power meter) done by the cyclists is not controlled.

3.3.2 Percentage of achievement of record power

Firstly, to measure the cyclist's performance, we will plot the percentages of achievement of record power for time trial rides. We take time trials because the rider is giving it his all, which will give a good estimate of the rider's performance level in the race. We have the time trial data of one rider on a year. First, after looking at the power data, we noticed that for some times the associated power value was not recorded and we chose to set all its values to 0. Secondly, we performed smoothing on the power curves for different durations depending on what we want using convolution product ($(a * v)[n] = \sum_{m=\infty}^{\infty} a[m]v[n-m]$). We plotted the percentages of achievement of record powers for the times corresponding to the times for which we had the associated record powers and which were less than the duration of the race. Then we calculated the percentages of achievement of record power for each time trial and for each durations that we want ($per = \frac{puismaxt}{prt} * 100$ with puismaxt the maximal mean power for a duration t and prt the record power for a duration t). We used plotly library to plot these histograms. Here are some examples of the histograms we obtained:

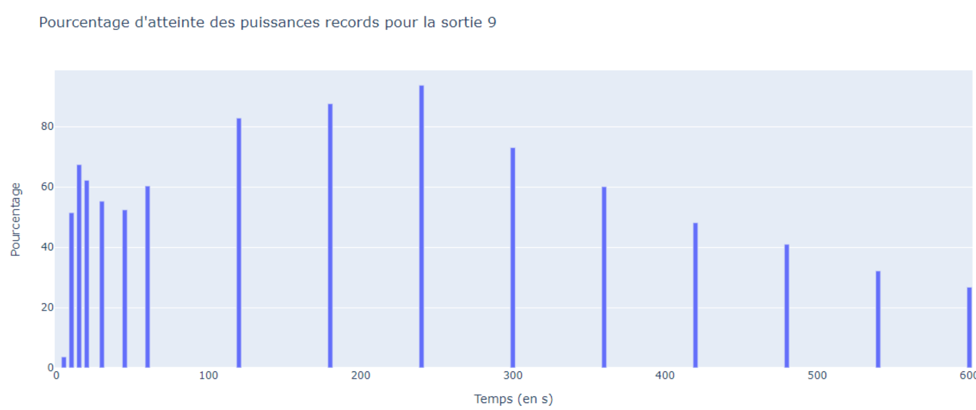

Figure 9: Percentage of achievement of power record for the activity 9

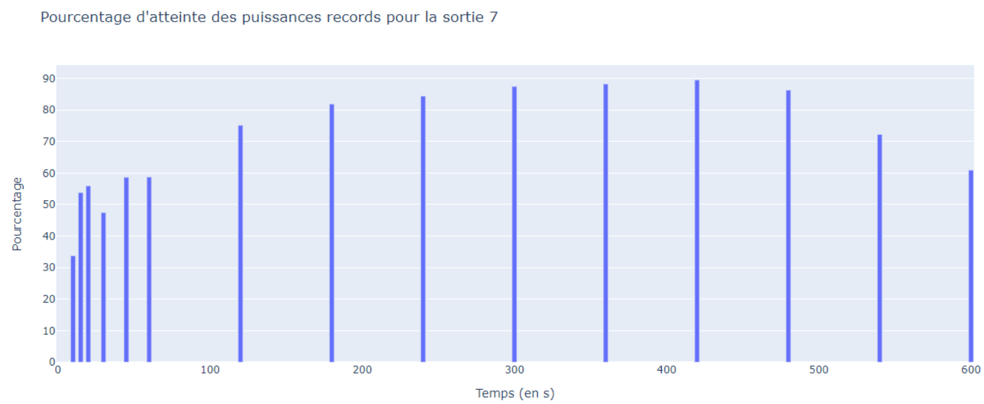

Figure 10: Percentage of achievement of power record for the activity 7

For example for the activity 9, We noticed that the percentage of achievement of record power at 3 minutes is nearly 87 %. The highest percentage is at 4 minutes (93 %). Globally, the cyclist need some time to start the race and it is difficult to maintain high power during the time due to the fatigue.

3.3.3 Percentage of achievement of record power for different work done levels

Secondly, we were interested in calculating percentages of achievement of record power but after different work done levels and for road races. We measured the level of performance taking into account the level of fatigue of the cyclist. We cut the power curves from the moment the rider exceeds the desired work done levels. We performed the calculations for workloads of 0,10,20,30,40,50 kJ/kg and to limit the number of values to plot, we calculate for the following durations: 5,10,20 minutes and 1 hour. We only took the maximum value of the percentages of achievement of the record powers for these 4 durations in order to obtain only 1 value for each work done levels. These calculations were done for all road races. Here are some examples of histograms:

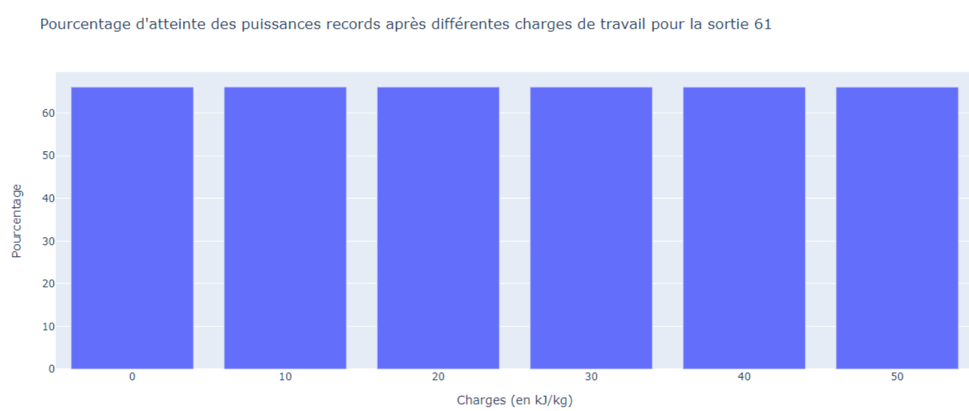

Figure 11: Percentage of achievement of power record for different work done levels for the activity 61

For the activity 61, the percentage of achievement of record power is nearly 66 % .

We notice that histograms have the same value for all the work done levels. The problem is that we have taken the absolute record power to calculate these percentages when we should have the record powers after different work done levels, but we will not have this data because it has not been measured. However, we could estimate them with existing data by looking at the maximum power that the rider reached after a certain work done level.

3.3.4 Improvement of percentage of achievement of record power record for different work done levels

During the early part of the internship, we tried to improve the code that calculated the percentages of achievement of record power by using the numpy library more. This saved a few seconds at runtime. Then, we calculated the average record power after different work done levels with the race data we have (we take the maximum power reached after a certain work done level for a race). Then, we can calculate the percentages of achievement of record power. Finally, we will not continue to develop this line of work because cyclists make maximum efforts at the end of the race, so we will not be able to determine the influence of fatigue on performance with this method.

3.4 Find correlations between our calculated performance level and the cyclist feelings.

3.4.1 Introduction.

In all of this part, we worked with a Google Colab Notebook and we wrote functions in Python language. In this section, we have plotted several curves with the help of the quantification of performance levels made in the first objective. First, we took a look at time trial races and we plotted the performance levels depending on the date. It allowed us to observe how performances evolved through the time. Then, we made the same plot for in line races. It was a bit different, because for in line races, performance levels were not only quantified depending on the date but also depending on work done levels (in kJ/kg).
After that, our second goal was to observe the correlations between the performance levels and the feelings of the cyclist. To do that, we received a data set which contains several subjective informations on the cyclist (for example, the exhaustion feeling).

3.4.2 Performance level for time trial races.

In the first objective, we wrote functions that gives us a percentage of performance for each activities. But this percentage was not unique for each activity. Indeed, the percentage of achievement of the record power was calculated for different times. So to get a unique indicator of performance for each time trial race, we have calculated the mean of each values (for each times) for each activity. By doing that, we obtain a unique level of performance for each date. Now, we are ready to see the plot of these performance levels (calculated with a mean) for each date. To calculate the mean, we used this formula : we take k which is the number of percentage calculated for one activity, which correspond to different times. To do the mean, we calculated $\frac{\sum_{i=0}^{k} percentage[i]}{k}$.

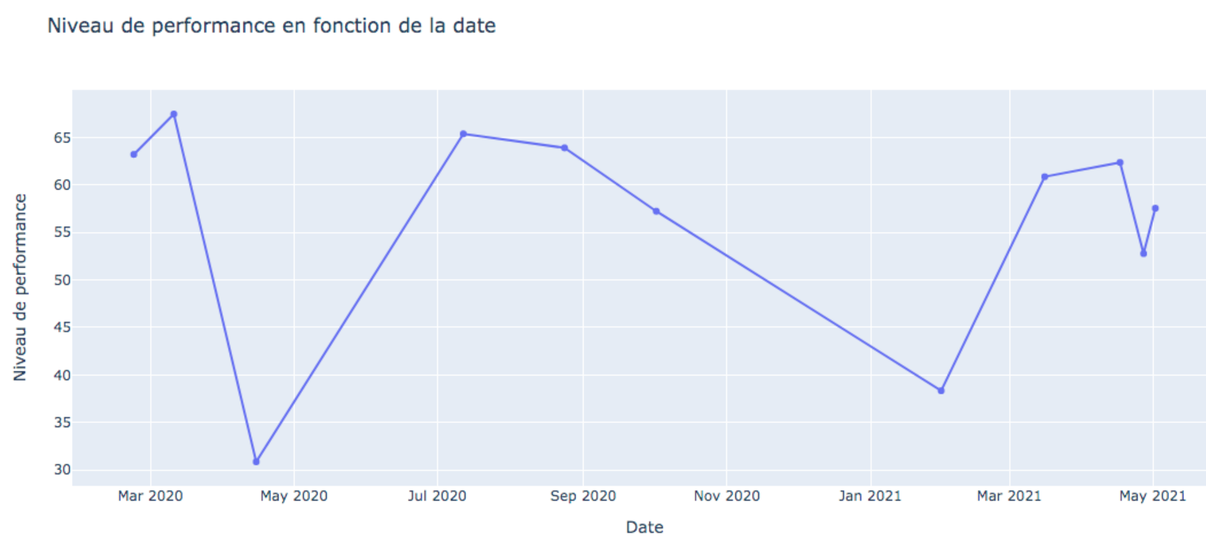

Figure 12: Plot of performance indicators depending on the date, for time trial races

We can observe that the performance level of the cyclist fluctuates a lot depending on the date. Overall, this level tends to increase when several activities are realised in a row. It could be explained easily: when the cyclist has done several races, he is trained so he is in a good shape to do better and better performances. Then, this increasing is always followed by a decreasing on the performance level. The cyclist is probably exhausted.

3.4.3 Performance level for in line races.

Now, in this part, let's observe the evolution of performance levels during in line races. It will allow us to do more precise analysis because we will not only calculate the performance for each activity, but we will also compare the level of performance for different work done levels. The different work done levels are 0,10,20,30,40 and 50 kJ/kg. We worked similarly that with time trial races. But, because there are different power levels for different times, we calculated means. But it was a bit different because we have to calculated the mean for each work done levels. Then we plot performance level depending on the date for each work done levels. That's why we have 6 different plots. Unfortunately, the indicators calculated in the first objective are probably false for in line races (they are almost the same for each work done) and we know what is the error (I will explain it later in the possible amelioration section). But for now, let's only observe the plots for 0 kJ/kg and for 10 kJ/kg.

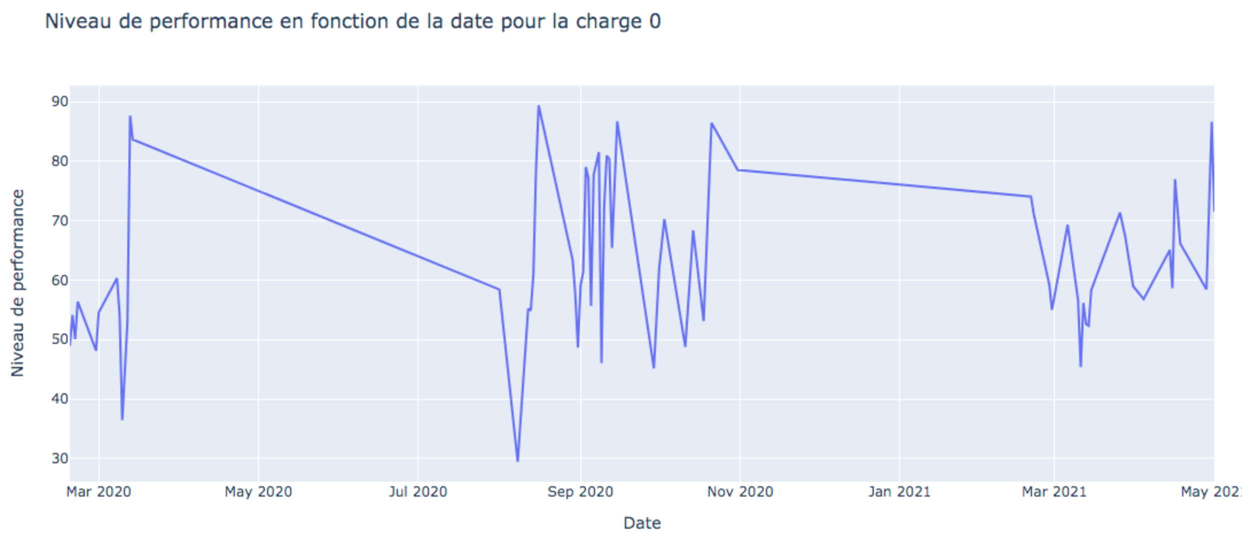

Figure 13: Plot of performance indicators depending on the date, for in line races, after the work done level 0 kJ/kg

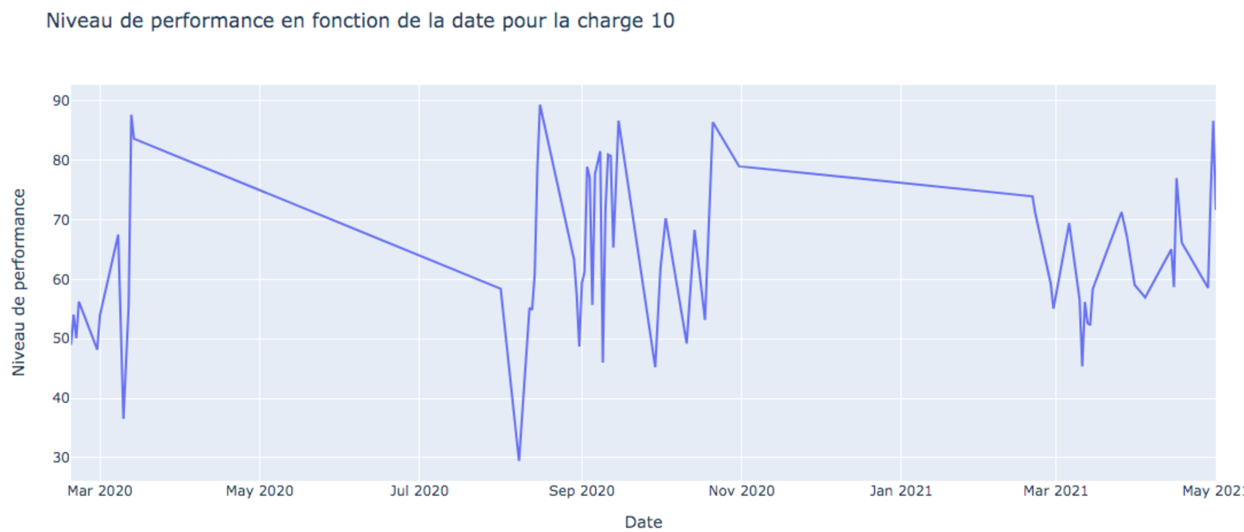

Figure 14: Plot of performance indicators depending on the date, for in line races, after the work done level 10 kJ/kg

Analysis : We can observe that our performance levels oscillates much, it is probably not correct. And moreover, it is almost the same for each work done level which is not right.

3.4.4 Correlating performance indicators and feelings.

We have to measure correlations between the performance indicators and the feelings of the cyclist. Initially, it was decided that we will calculate these correlations only for in line races, but because the results are not good for these races, we will focus on time trial races, because the results are coherent.

To calculate the correlations, we used some implemented Python functions, here is an extract of the code we wrote to have the correlations :

```
corr_matrix=df.corr()
plt.figure(1,figsize=(14,14))
sns.heatmap(np.round(corr_matrix,2), annot=True,cmap="jet");
```

Before doing that, we have done an observation of correlations calculating the matrix of correlation only for the feeling data-set without adding our calculated performance levels. Let's observe this in figure 14:

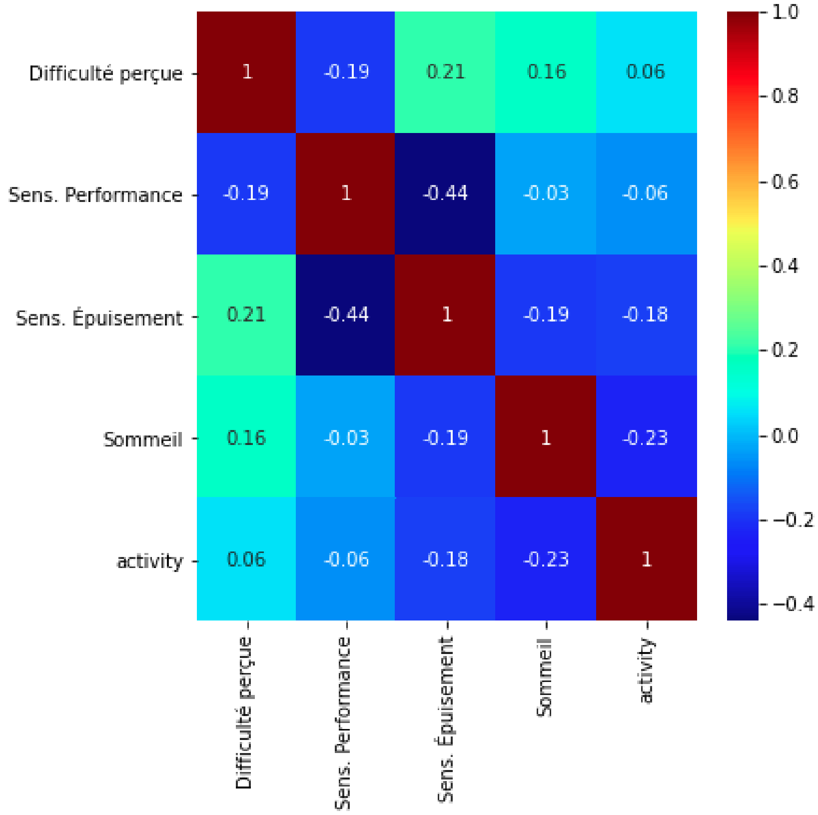

Figure 15: Matrix of correlations for the sensation data-set

Analysis : we see a correlation between performance feeling and exhaustion feeling. Indeed, it seems logical that when a cyclist felt tired, then he also felt less performing. So, with the performance levels, we will be able to verify this correlation. We will see if the performance level is really linked to exhaustion feeling.

To do that, we will compare :

- performance level vs difficulty feeling
- performance level vs performance feeling
- performance level vs exhaustion feeling
- performance level vs sleep

So, we had to create a new CSV file, which contain the same columns than the sensation CSV file but with one additionnal column : the column which contains the performance level for each activity. By adding this column in the data-set and calculating the matrix of correlations, we got this table of correlation (table 1) :

Comparison	Correlation
Performance level vs Difficulty feeling	-0.68
Performance level vs Performance feeling	0.68
Performance level vs Exhaustion feeling	-0.11
Performance level vs Sleep	-0.32

Table 1: Table of correlations

Analysis : we observe one correlation between performance level and difficulty feeling, and also one between performance level and performance feeling. With these correlations, we can see that our performance indicators seemed pretty reliable. Indeed, it is logical that when a race seemed very difficult to the cyclist, then his performance level decrease. Moreover, the cyclist knows himself well : his performance feeling which is only an estimation is linked to the performance level that we calculated. So our estimator is probably pretty reliable.

Only one correlation is missing : in fact, the data indicates that exhaustion feeling may not be correlated to performance level. Exhausted cyclists think they are doing a bad performance but the data does not seem to show that.

3.4.5 Amelioration and work which could be done this summer during the internship.

The first amelioration which could be done is for in line races. Indeed, for those races we didn't get precise performance levels. This could be explained with one easy reason : in this part, we had to estimated the percentage of power after reaching different work done levels. But, our data-set of record powers doesn't take the different work done levels into account. That's why all of the plots were the same. To change that, we will have to calculate by our self (by hand) the record power after reaching the different work done levels. After that, we will be able to get a reliable performance level, as for time trial races.

Then, an other improvement will probably be to optimize the Python functions (less loop, vectorization) because at the moment our functions are pretty slow to execute (approximately 30 seconds to execute the notebook).

Now, the project is finished and we will follow our work during the summer in the internship. Let's see this part.

3.5 Quantify performance indicators and measure correlations

In all of this part, we worked with a Google Colab Notebook and we wrote functions in Python language.

3.5.1 Introduction and link with the previous project.

We will calculate performance indicators for different kind of activities. There are road races, which are races that last longer than time trials and whose intensity varies greatly depending on the conditions and the progress of the race. Then we have time trial races which corresponds to a constant and continuous effort and during those races cyclists are going faster. We have already begin to calculate some indicators during the project this year but there were many problems and now new goals which are more realistic were set by analysing the previous results. We have now many thing to do.

In the first part, we will focus on time trials. For time trials, it will be quite simple. We will only calculate a performance indicator by looking at the percentage of reaching the record power. Then it will be already finished, we will not calculate a correlation matrix because there is too much missing data for these races. Then, in a second step, we will calculate performance indicators for road races. We will proceed differently than during the project, I will explain all this later. And finally, we will calculate the correlation matrix.

3.5.2 Explanation of our Python functions.

I will start by presenting you the code that will be common to all types of races. These are functions that will be very useful to us. First of all, there is the smoothing function which allows us to perform the smoothing of a function.

```
def smoothing(X, n, mode="same"):
    box = np.ones(n)
    return np.convolve(X, box, mode=mode)/n
```

Then, we make a function that calculates the percentage of reaching a record power for the different times of the record power dataframe.

```
def pourcentage_record(mat_conv,temps,record):
  ycomp = np.zeros(len(temps))
  ycomp[0] = ((mat_conv[0,:])[0])/record[0]*100
  j = 1
  for i in temps:
    if i!=temps[0]:
      y = mat_conv[j,:]
      ycomp[j] = np.max(y[:i])/record[j]*100
      j +=1
  return ycomp
```

Finally, we write a function that returns a matrix containing the smoothed power vectors for all possible duration of the dataframe.

```
def matrice_lissage(xi,yi,puissance_record):
    duree_total = len(xi)
    temps = puissance_record["Temps"].values
    intervalle = temps[temps<=duree_total]
    nb_int = len(temps)
    mat_conv = np.zeros((nb_int,duree_total))
    i = 0
    for j in intervalle:
      mat_conv[i,:] = smoothing(yi,j)
      i+=1
    return mat_conv,intervalle
```

We have also added two functions that allow you to select only road races or only time trials.

```
def clm(df):
  clm_or_cel = df_sensation.groupby("type_activity")
  return clm_or_cel.get_group("clm")

def cel(df):
  clm_or_cel = df_sensation.groupby("type_activity")
  return clm_or_cel.get_group("cel")
```

Before we start, we must keep in mind that no matter what calculations and methods we use, our performance indicators will not be 100 % reliable. Indeed, these indicators are hard to define because it is difficult to reduce all the performances of a cyclist to a single number. So it is normal that our performance indicator does not manage to capture all the performances.

3.5.3 Performance level for time trial races.

First our objective for time trial races is pretty simple and almost identical to the project. We have to calculate the performance indicators with the power records. We already have done that before but our calculated indicators were not reliable because for professional cyclists, it should be almost always up to 60 % and it was not the case but now when we have done it again, we got the good estimators. Here is a look at the Python code of this part. We have created a function that returns an array containing the record powers per activity. This will be our performance indicator which we will still have to average. This average is done in lines 14 and 15 of the same function.

```
1  def moyenne_perf(nb_clm,num_activity,num_clm,puissance_record):
2    indic_perf = np.zeros(nb_clm)
3    k = 0
4    for i in range(nb_clm):
5      dfi = num_activity.get_group(num_clm[i])
6      xi = dfi["time_seconds"].values
7      yi = dfi["power"].values
8
9      num_coureur = np.max(dfi["coureur"])
10     puissance_coureur = puissance_record[num_coureur].values
11     mat_conv,intervalle = matrice_lissage(xi,yi,puissance_record)
12     pourcentage = pourcentage_record(mat_conv,intervalle,puissance_coureur)
13     #boucle sur la longueur de pourcentage pour calculer la moyenne de puissance
14     indic_perf[k]=np.mean(pourcentage)
15     k+=1
16   return indic_perf
```

Then, we made a function that plots this performance level.

```
def courbes_niveau_perf(chemin_df,chemin_puissance_record,chemin_sensations,plot=True):
  #on importe les datas
  df = pd.read_csv(chemin_df)
  puissance_record = pd.read_excel(chemin_puissance_record)
  sensations = pd.read_csv(chemin_sensations)

  df["power"].fillna(0, inplace=True)

  clm = sensations[sensations["type_activity"]=="clm"]

  num_clm = clm["activity"].values
  nb_clm = len(num_clm)
  #on recupere les activites
  num_activity = df.groupby("activity")

  x_clm = clm["date"].values
  y_clm = moyenne_perf(nb_clm,num_activity,num_clm,puissance_record)
```

```
    x_clm_activity = clm["activity"].values

    if plot:
      fig = px.line(x=x_clm, y=y_clm,title="Niveau de performance en fonction de la date",
       labels={'x':'Date', 'y':'Niveau de performance'})
      fig.update_traces(mode='markers+lines')
      fig.show()

    return y_clm
```

And here in the Figure 16, you can see the curve we obtain.

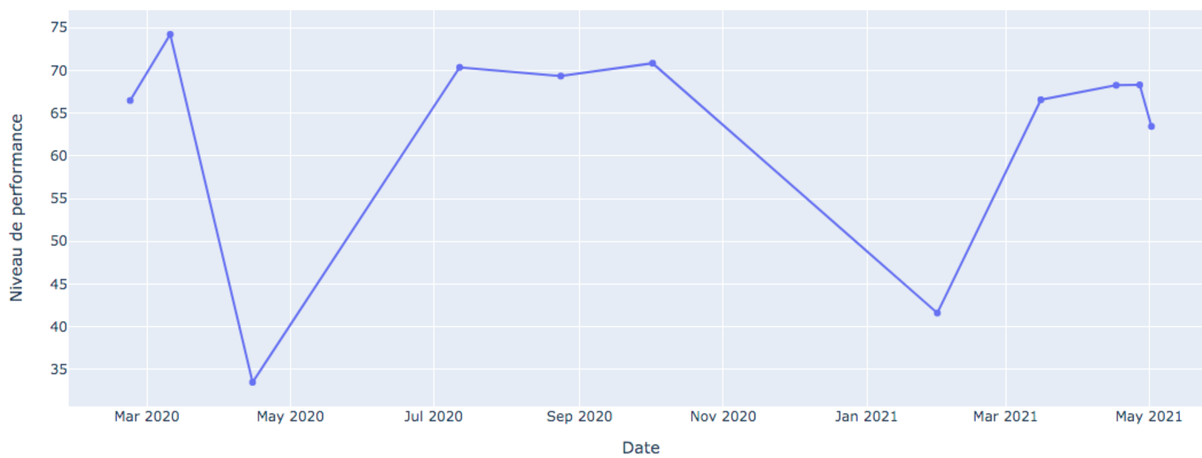

Figure 16: Plot of performance indicators depending on the date, for time trial races

We can see that the results seem consistent. Indeed, all but two values are above 60 %. This is coherent with the performance of the cyclists considered during the time trials.

For time trial races, we won't compare the indicators with the sensations because there are too many missing data so it won't be reliable.

3.5.4 Performance level for road races.

Then we have a second part which is longer than the first, it is for road races. With road races, we will do many things. First of all, we calculated the maximum of the percentages of reaching the record powers. But we didn't do the same as for the project with work done levels because it didn't give good results.

So we used work classes, which correspond to different duration of the race. These classes were divided into 4: the first one from 10 seconds to 45 seconds, the second one from 1 minute to 9 minutes, the third one from 10 minutes to 45 minutes and the last one from 1 hour to 5 hours. All these classes will allow us to compare the performance indicators at different times of the race, and therefore necessarily at different intensities of effort (it is logical that at the beginning, in the middle or at the end of the race, the cyclist does not give the same effort all the time). Within each time class, we selected the maximum percentage for all durations that compose the class.

When we have the performance indicators for each time class, we will display a graph for each class and also a graph containing the 4 classes superposed. Then, we will also superimpose the sensation curves to see if they fit with our calculated performance indicators. Finally, we will calculate the correlation matrix to see if interesting correlations appear.

First of all, to select these different classes of time, we had to choose a method. The method I chose is to create 4 new dataframes, which correspond to extracts of the record powers dataframe for each selected duration. To do this, I used the iloc command which allows to select only some lines of a dataframe. So I obtained 4 new dataframes that I called $class\_1$, $class\_2$, $class\_3$ and $class\_4$, and each one corresponds to one of the classes I described before. Let's have a look of the code which allows us to do that.

```
pd_record = pd.read_excel(puissance_record)
class_1 = pd_record.iloc[2:7,:]
class_2 = pd_record.iloc[7:16,:]
class_3 = pd_record.iloc[16:21,:]
class_4 = pd_record.iloc[21:26,:]
```

After having obtained all these dataframes, it was not complicated anymore, I just had to use the function that calculates the performance indicators and to enter these dataframe samples as parameters instead of putting all the dataframe of the record powers. Here is what we got:

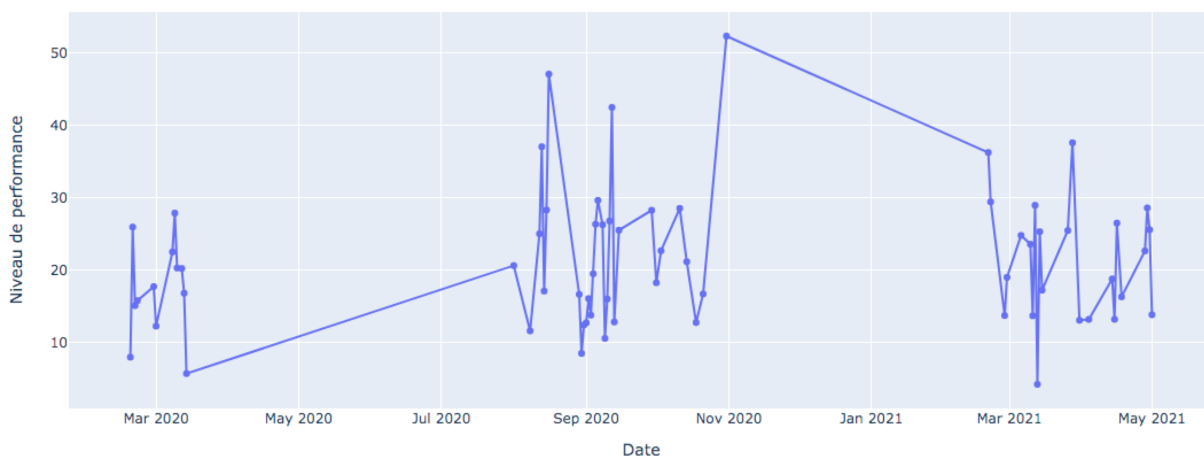

Figure 17: Plot of performance indicators depending on the date, for class 1

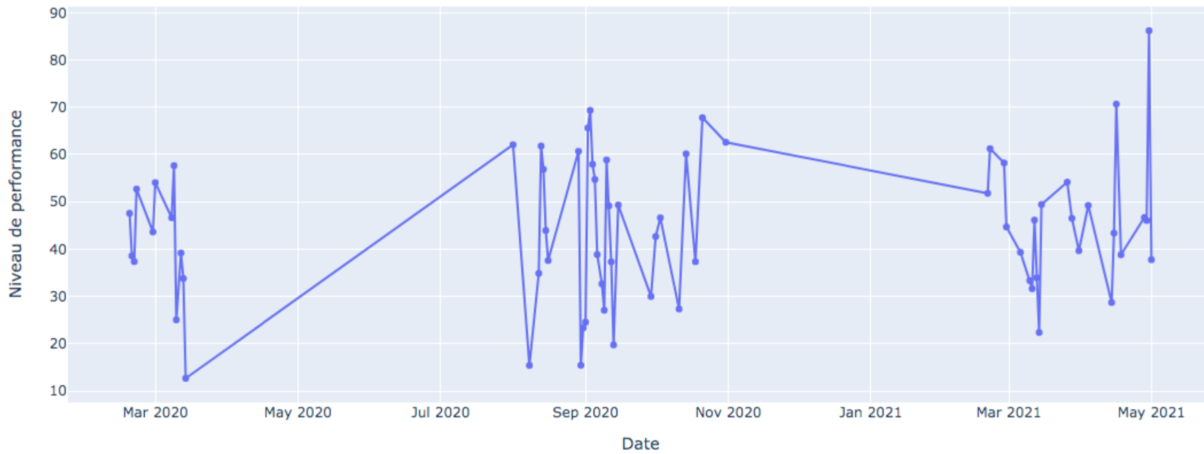

Figure 18: Plot of performance indicators depending on the date, for class 2

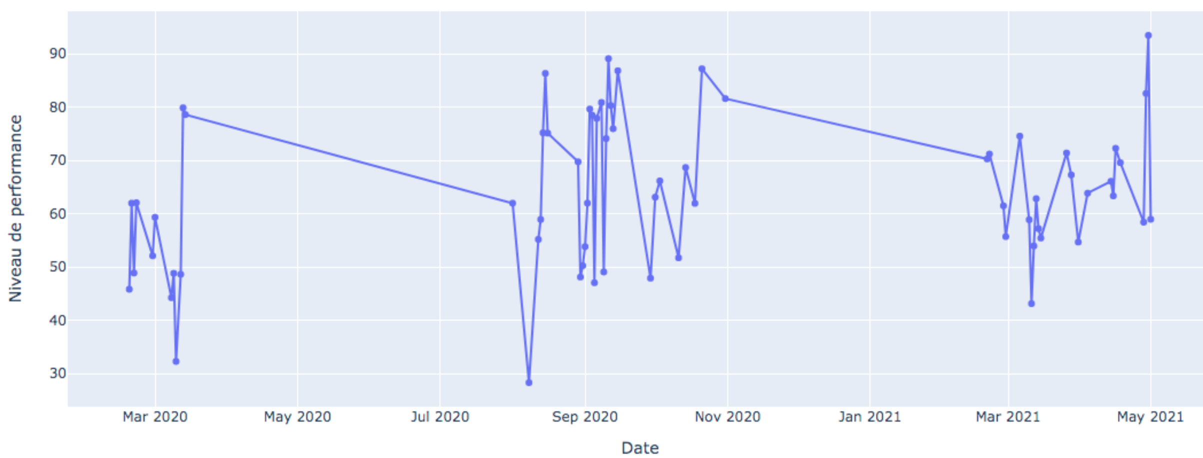

Figure 19: Plot of performance indicators depending on the date, for class 3

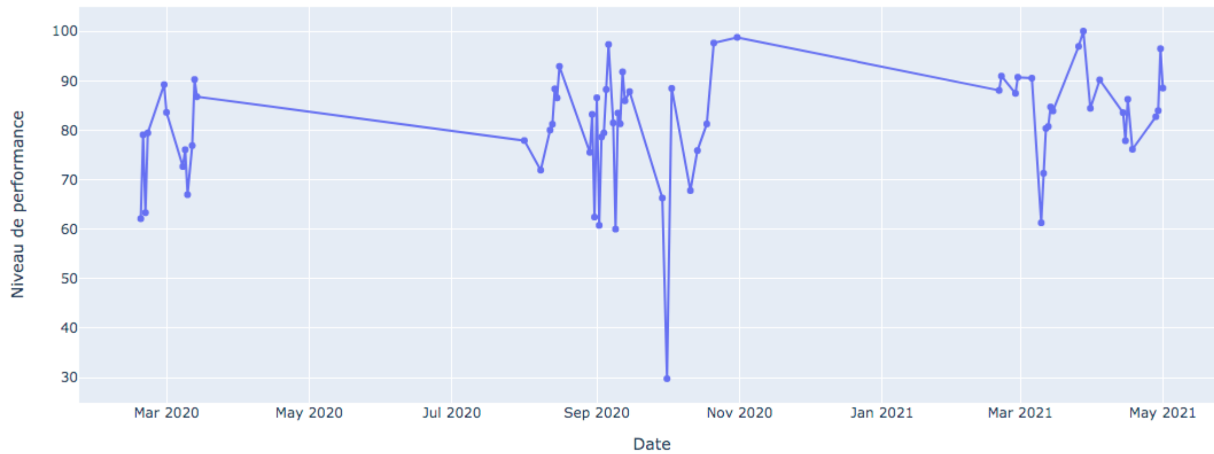

Figure 20: Plot of performance indicators depending on the date, for class 4

This is coherent, in fact, we see that for class 1, the performance indicator does not exceed 50 % and is often close to 30 %. This seems logical because road races are long races, so the cyclist starts the race calmly. Then, for the other 3 classes, this indicator increases more and more, it is almost always above 60 % and it even reaches 90 %.

When the performance indicator is 100 %, which is sometimes the case for class 4, it means that a record has been broken by the cyclist. The first class is a sprint so it is normal that no records are broken during this class, while class 4 is a long race so it is logical to break records.

3.5.5 Superposition of our plotting with sensations.

Then we superposed the feeling curves with our performance indicators, and we can see that the curves are very close so it seems that our indicator is reliable. This is good because during the project, the curves for road races were not correct, they oscillated a lot and they did not seem to give the right values. You can see the superposed curves in the next figure :

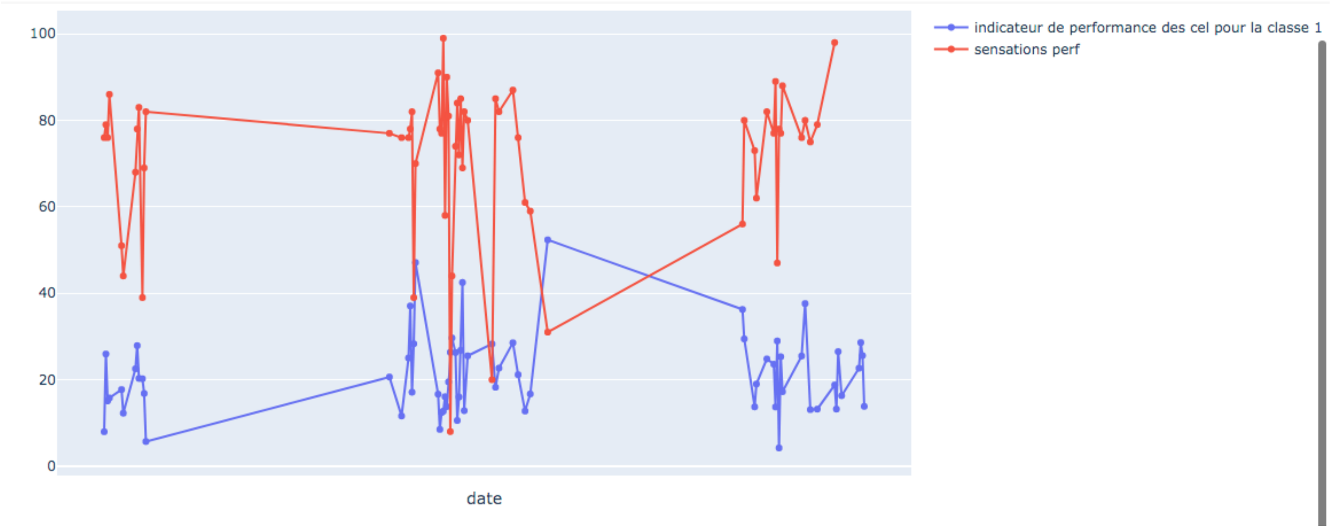

Figure 21: Superposition of the curves for class 1 with the sensation curves

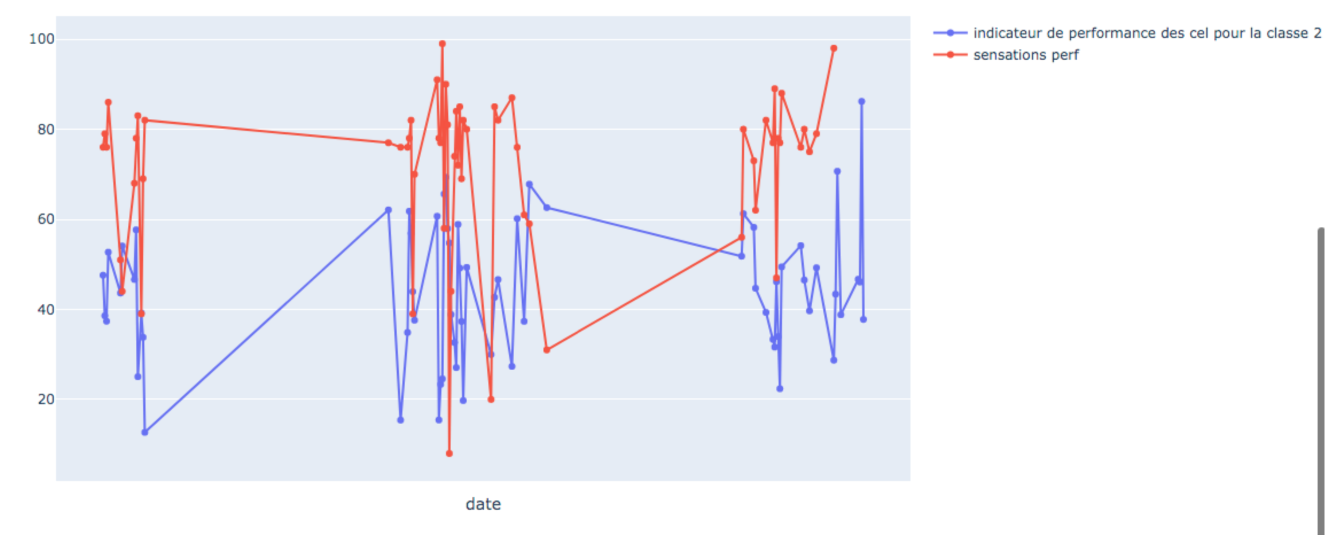

Figure 22: Superposition of the curves for class 2 with the sensation curves

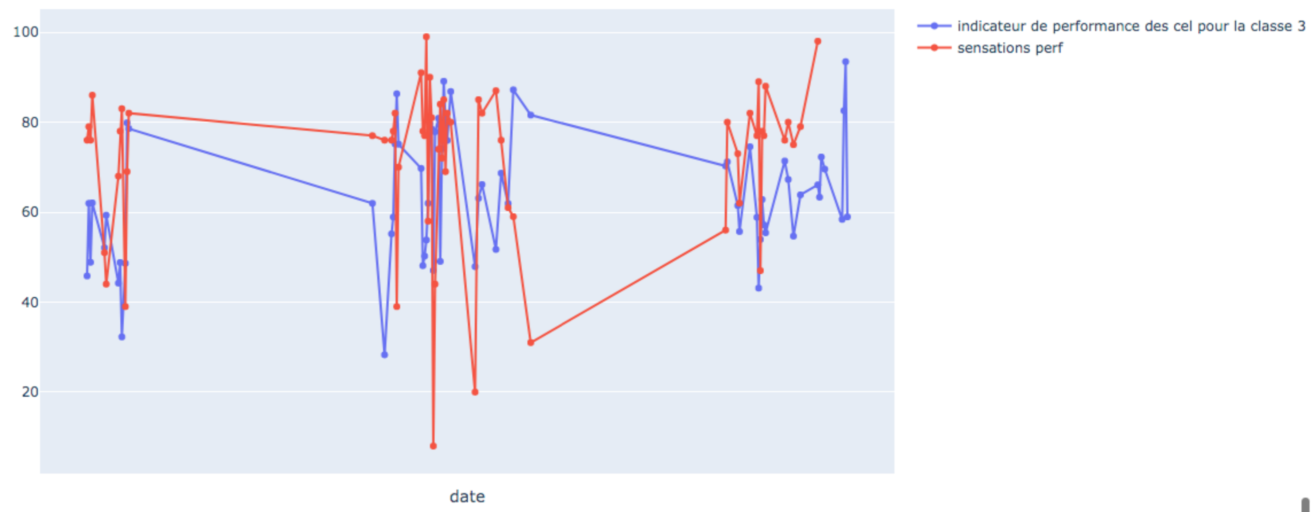

Figure 23: Superposition of the curves for class 3 with the sensation curves

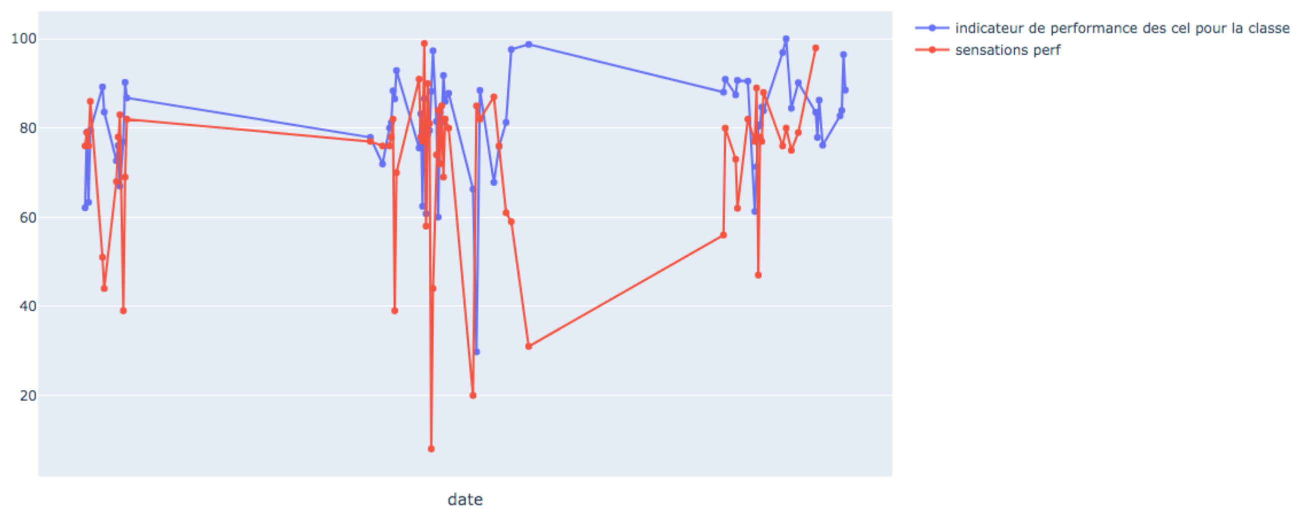

Figure 24: Superposition of the curves for class 4 with the sensation curves

We see that the performance indicator we calculated seems to superpose itself rather well with the cyclist's performance sensations. This is especially true for classes 3 and 4, for which our estimator seems very consistent. We will check later in another section if indeed there are correlations between these two data.

3.5.6 Superposition of our 4 plots.

Then, we also found it wise to overlay the performance levels for the 4 classes on the same graph to see what we were getting. This allows us to see the evolution of the performance level at different times of the road races. It is interesting to make these observations because road races are long races during which the cyclist is in different phases of effort and intensity.

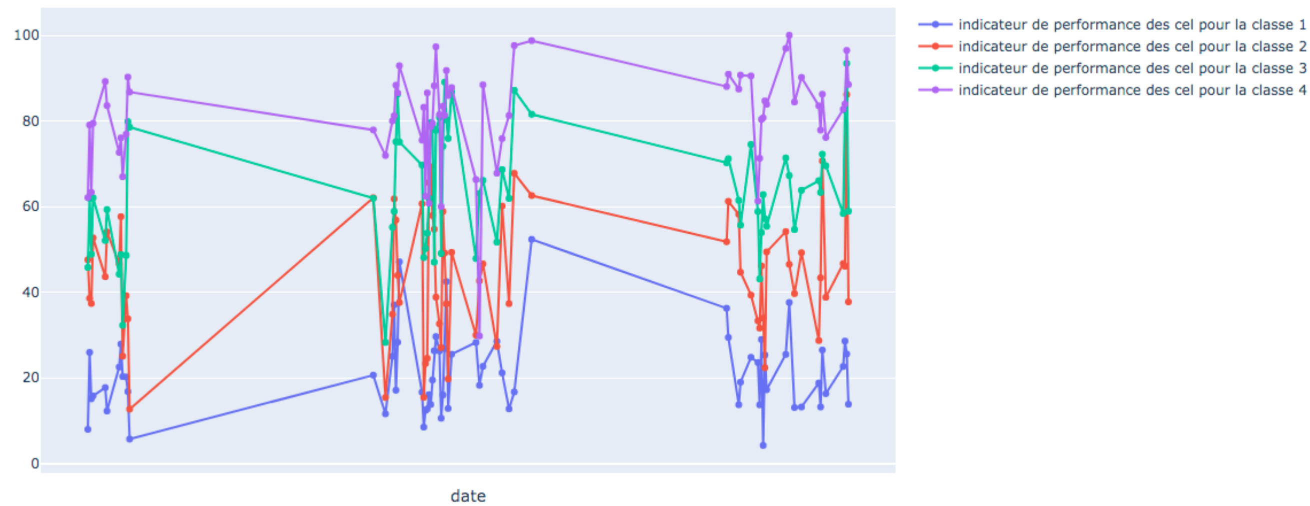

Figure 25: Superposition of the curves of all the classes

We see that the higher the time class is (which corresponds to being at a later point in the race), the higher the performance level is. This seems logical, as the cyclist knows that the race is long so at the beginning he starts slowly. Gradually he goes faster and faster and his performance is better. When he enters class 4, which is the last class, his performance level reaches peaks of almost 100 %.

3.5.7 Correlating performance indicators and feelings.

Finally, we calculated the correlations between the performance indicators calculated for the different classes and the cyclist's sensations. To do this, we created a new dataframe that contains all the sensations dataframe, to which we added 4 columns that correspond to the performance indicators for the 4 classes. I will show you the code I used to add the columns. First I created new columns with default values. Then I assigned the right values to these columns with the iloc command.

```
new_df_cel = cel.assign(Niv_Performance_Classe_1=0)
shape=new_df_cel.shape
niveau_perf = courbes_niveau_perf(path,class_1,path_sensa,1,plot=False)

for i in range(shape[0]):
  new_df_cel.iloc[i,shape[1]-1]=niveau_perf[i]
```

Then we calculated the correlation matrix. Here it is:

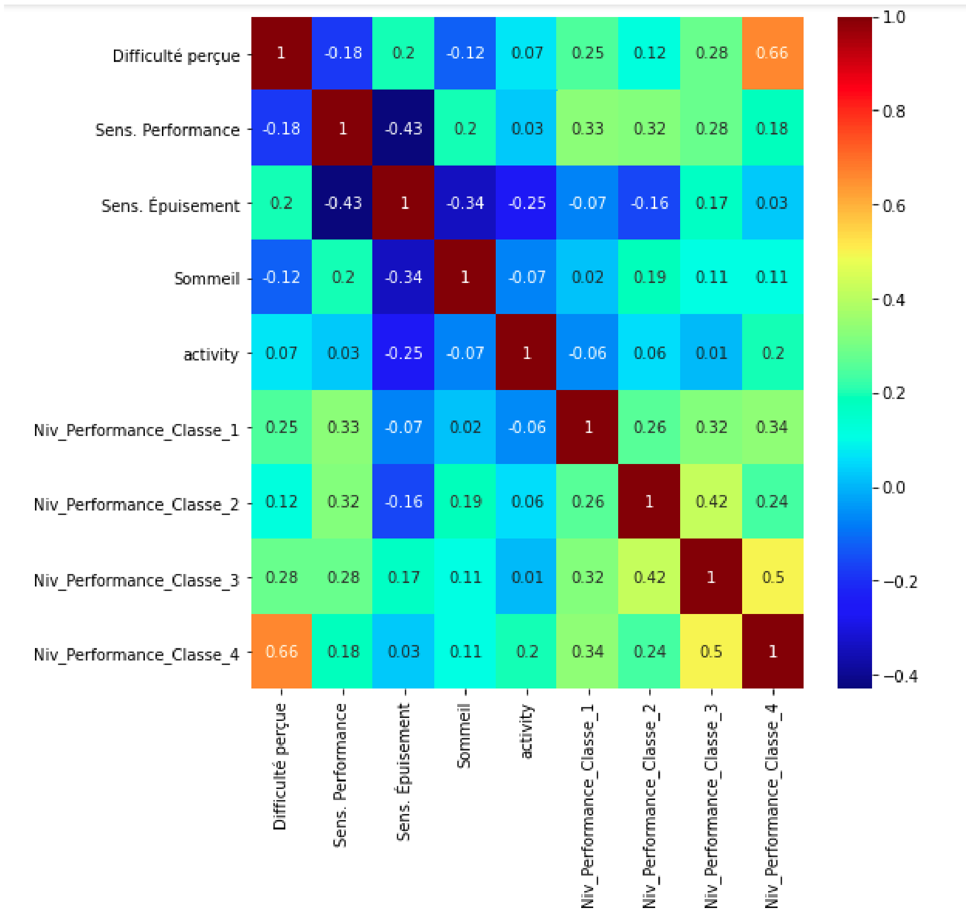

Figure 26: Matrix of correlations

Here is an extract of the table of correlation that we got with the more interesting correlations.

Comparison	Correlation
Performance level (class 1) vs Performance feeling	0.33
Performance level (class 2) vs Performance feeling	0.32
Performance level (class 3) vs Performance feeling	0.28
Performance level (class 4) vs Performance feeling	0.18
Performance level (class 4) vs Difficulty feeling	0.66

Table 2: Table of correlations

Analysis : We see that the correlations between the cyclist's performance sensation and the performance levels for the different times are around 0.3. This indicates that there is a beginning of correlation, the correlation is not zero but it is not proven either. This seems logical because given our calculations, which may contain some errors, we cannot have a very reliable indicator, but still it seems to hold. Furthermore, we observe that the performance level for class 3 is correlated with the performance level for class 4 (correlation of 0.5). This is rather logical because these two classes are the two longest classes, and during these time periods the cyclist's performance fluctuates little.

Finally, the most important correlation is the one between the performance level of class 4 and the feeling of difficulty (correlation of 0.66). This is very interesting because first of all it tells us that the performance indicator for class 4 is the most convincing of the 4. This is normal because for the other classes, we expect the records to be broken during the race and not during training. Furthermore, this high correlation with the sensation of difficulty and not the sensation of performance indicates that the record power levels are more representative of the perceived difficulty of the effort than of the sensation of having performed. The cyclist may have the sensation of having performed well but not having broken a record because the race was not suitable because of its difficulty or its profile. Our indicator therefore captures the intensity of the effort rather than the actual performance.

3.5.8 Conclusion.

To conclude, following the project where our estimators were not all reliable, especially for road races, we now wanted to finally obtain consistent results. It is now done thanks to this internship which allowed us to calculate performance indicators for all the different types of races. From now on, the only improvement that could be made is to vectorize the code a little more to make it faster because the execution is still a little long. Indeed, we could factorize the calculations and use the numpy library because vector calculations are extremely powerful and fast with this library. It would make a huge difference.

3.6 Critical power model, work prime and work prime balance

3.6.1 Critical power and prime work

The critical power model can allow us to predict performance and prepare for a race. This model is essentially based on 2 parameters that we will calculate later :

The **critical power** (noted PC) and the **prime work** (noted W'). The **critical power** is the maximum power that the cyclist can maintain in a stable physiological state. For power levels below the critical power, the cyclist can maintain the effort "indefinitely". However, as soon as the cyclist's power exceeds the critical power, the cyclist can only maintain the effort for a certain time. The **prime work** is the "anaerobic" work reserve, which decreases when the cyclist is at powers above the critical power. The reserve can be built up again by cycling at powers below the critical power but the greater the effort above the critical power, the longer the recovery. The W' can be identified with the level of available battery and a new concept can be introduced: the **W' balance** (often called **Wbal**) which corresponds to the energy expenditure.
We will calculate the **critical power** and **work prime** from the record power data. By plotting the record powers as a function of T, for durations ranging from 2 to 20 minutes, we see a hyperbolic relationship between power and time. The record powers can therefore be approximated by an affine function of the inverse of time.

Figure 27: Power record as a function of inverse of the time

Let's break down the calculations to find the critical power and the prime work.

We have : $P(T) = a \times \frac{1}{T} + b$
We multiply by T :
$a = P(T) \times T - b \times T$
$= W_{P(T)} - W_b(T)$

where $W_{P(T)}(T)$ is the work done by the cyclist riding at power P for a time T and b correspond to the critical power.

Therefore, we have :
$P(T) = W' \times \frac{1}{T} + PC$

To determine PC and W', it is therefore sufficient to perform a linear regression, here we used the scikit-learn library of python. We find that the critical power of the rider is therefore about 479.8 W and the prime work is about 21503.8 J. Here is the algorithm of the linear regression :

```
#Fonction de selection du df sur des durees [start, stop]
def select_df(df, start, stop):
    mask = (df["Temps"] >= start) & (df["Temps"] <= stop)#changement selection
    return df[mask]

#on calcule la puissance critique et le travail critique pour le coureur 13
def calcul_val_critique(puissance_record):
    df_ppr = puissance_record
    df_ppr["inverse_duree"] = 1/df_ppr["Temps"]

    df = select_df(df_ppr, 2*60, 20*60)#on choisit ses durees car c'est la periode sur
     laquelle on peut faire une regression lineaire
    X = np.array(df.inverse_duree).reshape(-1,1)

    #pas besoin d'utiliser une liste on considere uniquement un coureur
    y = df[13]
    reg = LinearRegression().fit(X, y)

    PC = reg.intercept_
    W_prime = reg.coef_[0]
    return PC,W_prime
```

3.6.2 W' balance

We will now look at the **W' balance** more often called **Wbal or W'bal**. This is actually the energy expenditure (we can see it as the available battery level), with the W'bal we can see the variations of the "battery" level, so we can see when the "battery" level increases or decreases.
To calculate W'bal, we will use the differential algorithm which uses PC and W'. **The W'bal differential algorithm** is the closest to reality and the simplest to implement, which is why it is chosen. It uses the following formula for the calculations :

$$W'bal(t) = \sum_0^t (PC - P(T)) \times \begin{cases} 1 \; where \; P(T) > PC \\ \frac{W' - W'bal(t-1)}{W'} \; else \end{cases}$$

Here is the code for the differential W'bal algorithm:

```
#on calcule W_bal
def W_bal_differential(PC,W_prime,power):
  W_bal = []
  W_bal_val = W_prime
  for p in power:
    if p<PC:
      W_bal_val += (PC-p)*(W_prime - W_bal_val)/W_prime
    else:
      W_bal_val += PC-p
    W_bal.append(W_bal_val)
  return np.array(W_bal)
```

We will see an example of W'bal for the time trial 4 :

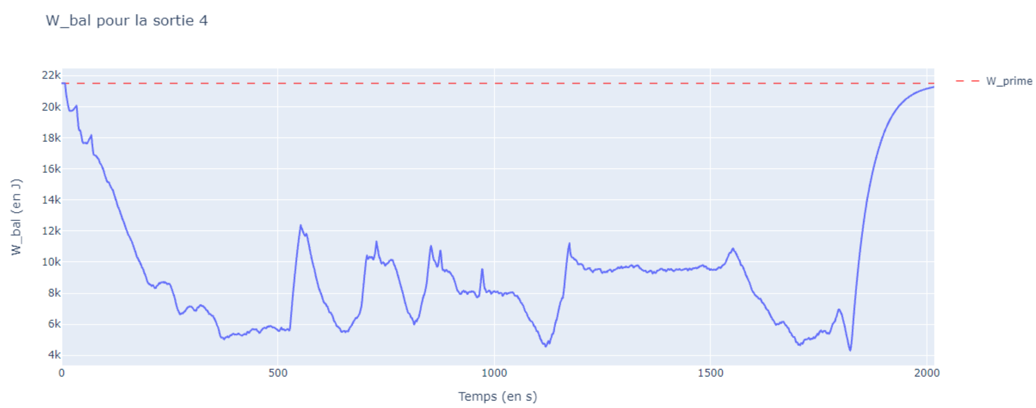

Figure 28: W'bal obtained by the differential algorithm for the time trial race 4

It can be observed that the rider first drains his "battery", so he is at powers above the critical power, then he goes below this critical power and thus recharges the "battery". He repeats this operation several times. The closer you get to 0, the longer it takes to recharge the "battery". The graph seems coherent because the W'bal is equal to the W' at the beginning of the race.

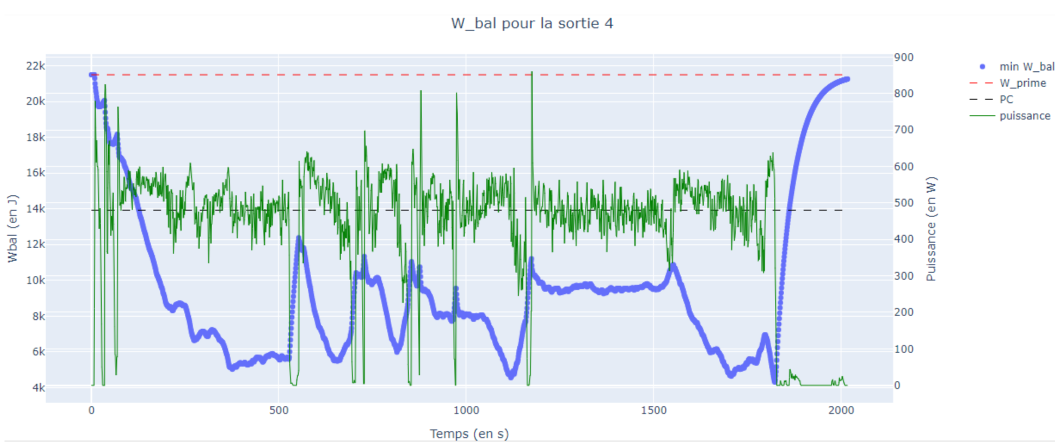

Figure 29: W'bal obtained by the differential algorithm for the time trial race 4 superposed to the power data

With the power curve, it can be seen that the W'bal increases when the power decreases sharply (i.e. it is below the critical power) and the W'bal decreases when the powers are above the critical power. This is therefore consistent with the expected result.

3.6.3 Searching for a link to performance sensations

We will now look at two pieces of data: **the minimum of W'bal** and **the mean integral of W'-W'bal over time**. With the minimum of W'bal, we expect it to be as small as possible if the rider has performed well. And for the mean integral W'-W'bal which reflects the intensity of a race we expect to have quite high value the better the performance. We plot these indicators depending on the date for all time trial races and just the minimum of W'bal for the road races because the intensity of the race depends on others factors (groups of cyclists …) and not just of the difficulty of the race. Then, we superposed to these plots the sensations of performance or the difficulty that the cyclist noticed. Here some examples of what we find :

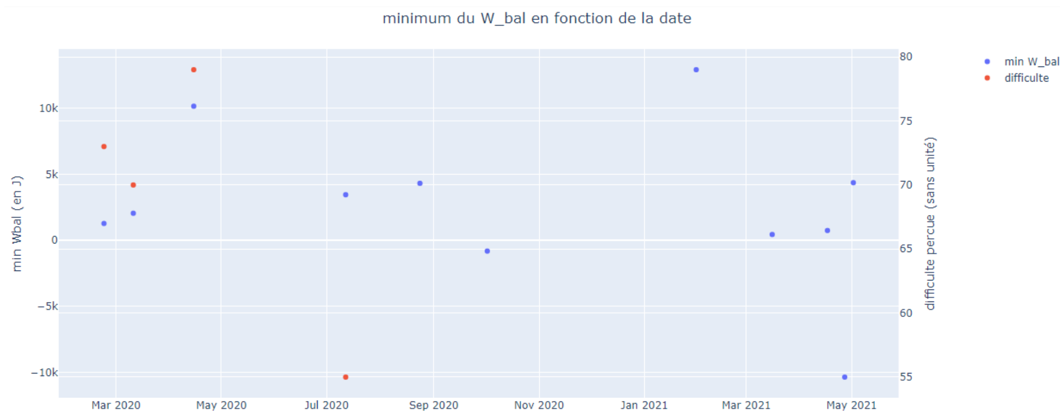

Figure 30: Minimum of W'bal for different time trial races superposed to the difficulty

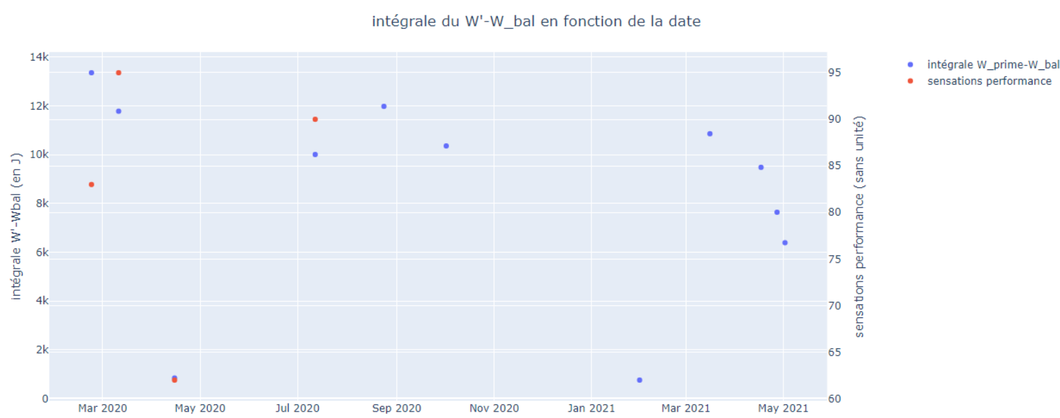

Figure 31: Mean integrale of W'-W'bal for different time trial races superposed to the sensation of performance

We have little data on performance and difficulty for time trials. However, by observing the data according to the periods, we can observe some vague similarities between the feeling of performance and the minimum of the W'bal and between the perceived difficulty and the mean integral of the W'-W'bal related to the time. It is difficult to draw conclusions from these graphs because the number of data is quite low, there could be a link between the 2 but this would have to be verified with more data.

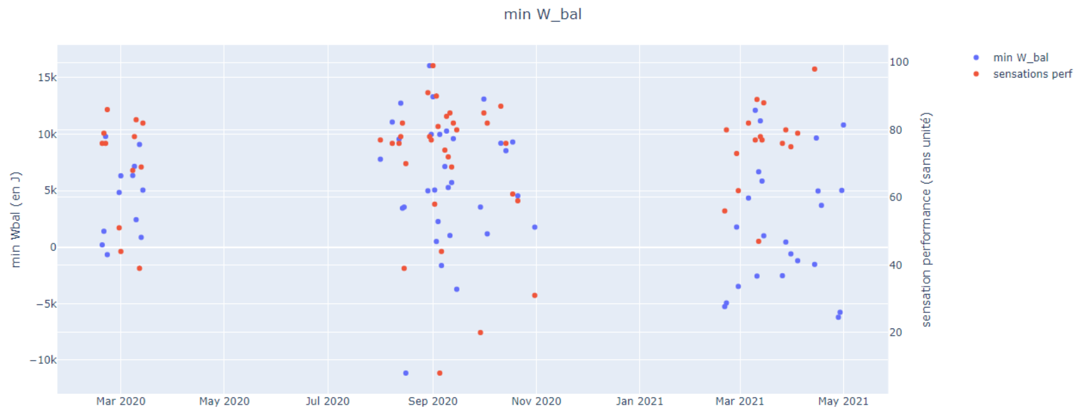

Figure 32: Minimum of W'bal for different road races superposed to the sensation of performance

For road races, there is much more data available on performance sensations, which gives better results. Indeed, it can be observed that the performance data coincide rather well with the W'bal minimum.

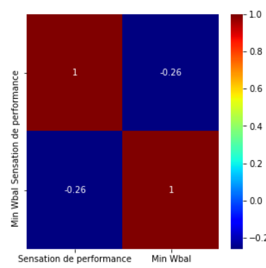

Figure 33: Correlation between the minimum of W'bal and the sensation of performance for road races

There is a correlation of 0.26 between the feeling of performance and the minimum of the W'bal for road races. This shows the beginning of a correlation even if it is not fully proven. Our indicator does not completely reflect the cyclist's performance as it also depends on other factors which may explain this level of correlation.

4 Data informed physiological models.

4.1 Pulse Physiology Engine framework.

The **Pulse Physiology Engine** is a C++ based, comprehensive human physiology simulator that drives medical education, research, and training technologies. The engine enables accurate and consistent **physiology simulation** across the medical community.

Figure 34: Pulse Physiology Engine logo (Source : Pulse [5])

The engine is a **body physiology model** that combines **physics-based parameter models** and **control system mechanisms** to model real-time system level physiologic behaviors. Parameter models use electrical circuit analogs to represent the major physiologic systems. For example, the "Environment" circuit is represented on the figure **35**.

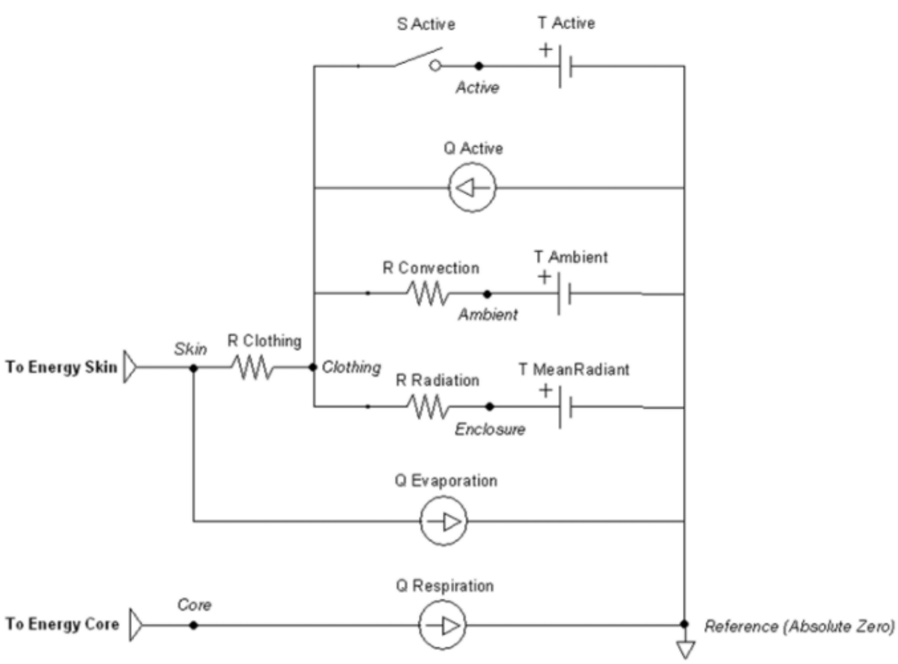

Figure 35: The Environment circuit consists of 6 nodes that are connected via 8 paths (Source : Pulse [12])

Closed loop circuits can be defined easily within the Common Data Model [6]. It is a specification of all data and relationships associated with the software [6]. It provides a data exchange format between models of physiological engine, but also between any application and physiological engine. Since the language is common to all, this makes it easier to implement new models in the application. Closed loop circuits fully dynamic, and can be manipulated and modified at each time step. The solver can be used to analyse electrical, fluid, and thermal circuits using unit in the international system. The solver outputs were validated using the outputs from an existing known third party circuit solver. All results matched the validation data, confirming this is a sound approach for the engine. Here with the environment system below, the Ambient node contains both thermal properties (temperature), fluid properties (pressure), and substance properties (volume fraction, aerosol concentration). It is assigned as the Respiratory and Anesthesia Machine circuit's reference node, and therefore, interacts with them directly. Any changes to the Ambient node properties automatically propagates through the other systems.

Pulse has been developed through a combination of funding by many government and private entities, and is a significantly improved and extended fork from the DoD-funded BioGears program. The Pulse repository is maintained by the Kitware team that includes the original core BioGears creators. Pulse has recently been incorporated into a number of commercial, research, and academic tools for medical simulation. Moreover, models and results are validated using a combination of peer-reviewed publications and subject matter expertise.

4.2 Models selection.

Our **objective** is to model the **physiological effort** of cyclists during their competitions and training. The **cardiovascular** and **energy** models are quite well suited to our data, so we will focus on them as a priority.

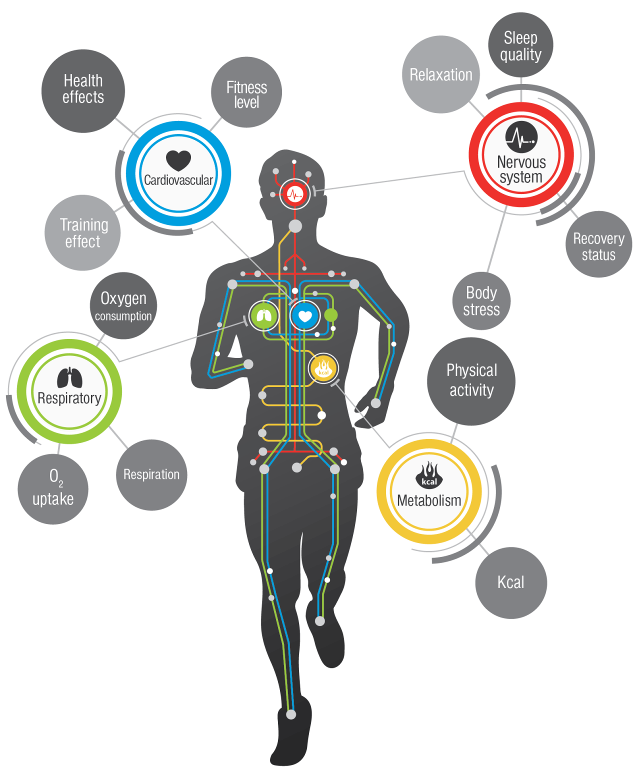

Figure 36: Diagram of some models from Pulse Physiology Engine framework (Source : Pulse [5])

Below is the composition and general functioning of these models:

- The cardiovascular system is a large organ system comprised of the heart and the blood vessels. It serves as the body's primary transport and distribution system. The cardiovascular system is sometimes described as two separate circulations: the systemic circulation and the pulmonary circulation. In the systemic circulation, oxygenated blood leaves the left side of the heart, travels through arteries and into the capillaries, and then returns as deoxygenated blood through the veins to the right side of the heart. From the right side of the heart, the deoxygenated blood travels in the pulmonary circulation through the pulmonary arteries, is re-oxygenated in the pulmonary capillaries, and then returns to the left side of the heart through the pulmonary veins. The state of the cardiovascular system is determined in a 3-step process [9]:

  - **Pre-processing** : Cardiac cycle calculations include methodology for updating the driving force (heart contraction and relaxation) of the CV System throughout the duration of a CV cycle (a single heart beat). This includes a set of systolic calculations that updates contractility at the beginning of the cycle to represent a heart contraction. Modifications to heart rate and heart compliance are calculated and applied for the remainder of the current cardiac cycle. Changes to things like heart rate and heart contractility can only occur at the top of the current cardiac cycle after the last cardiac cycle has completed. This helps to avoid discontinuous behavior such as the complete suspension of heart function mid-contraction.

  - **Process** : The function **Calculate Vital Signs** takes the current timestep's circuit quantities to calculate important system-level quantities for the current time step. The system pressures and flow rates related to shunting are calculated here. In addition, different cardiac events are triggered.

  - **Post-processing** : The Post-process step moves everything calculated in Process from the next time step calculation to the current time step calculation. This allows all other systems access to the information when completing their Preprocess analysis for the next time step.

- The energy system simulates metabolism and regulates internal body temperature. Thus, metabolism is simulated by the consumption of nutrients in tissues. Temperature regulation can be achieved by varying heat production (chills) or heat exchange (perspiration). As before, the state of the energy system is determined in a 3-step process [11] :

  - **Pre-processing** : The exercise function adds to the body's basal metabolic rate a value that is specified by the exercise action. The metabolic rate set-point is specified by the action but limited by the amount of energy available. The energy limit is computed, and the actual metabolic rate is ramped to the limited set-point. The body's actual work rate as a fraction of the total possible work rate and the body's actual work rate as a fraction of the action-requested work rate are set in this method.

  - **Process** : The function Calculate Vital Signs takes the current timestep's circuit quantities to calculate important system-level quantities for the current time step. In particular, the core and skin temperatures are recorded in this function.

  - **Post-processing** : The Post-process step moves everything calculated in Process from the next time step calculation to the current time step calculation. This allows all other systems access to the information when completing their Preprocess analysis for the next time step.

Figure **37** shows the steps used at each time step to determine each circuit state.

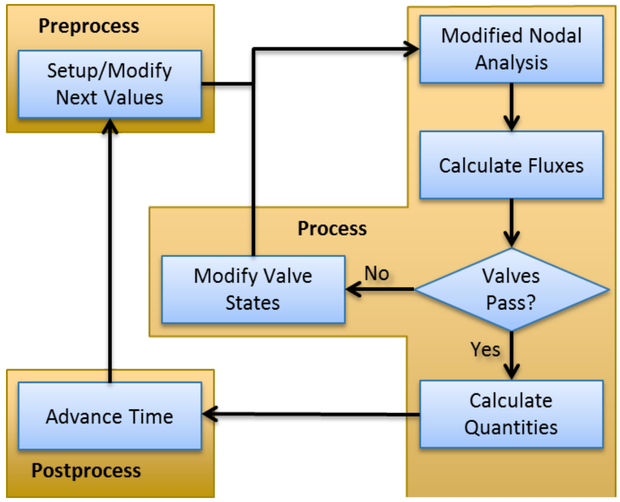

Figure 37: Steps to determine each circuit state (Source : Pulse [10])

4.3 First steps with Pulse.

The engine is driven by a set of instructions written in a scenario file. The structure of a scenario must respect the Common Data Model.
The scenario is stored in a JSON file and contains the following execution information [8]:

- Patient File and optional conditions.
- A list of values to return from the engine.
- A list of actions to execute over the course of the run.

As described in Listing **1**, the JSON file starts with a name and description, but this is not used in execution.

```
{
"Name": "Vitals Monitor",
"Description": "Data associated with a vitals monitor.",
...
```

Listing 1: Begin of VitalsMonitor.json.

Then, it is necessary to specify which file is used to initialize the patient's data. At the time of the simulation, this data is checked to ensure that the patient is stable before any action is taken. In Listing **2**, the patient's file is named *StandardMale.json*. Below, the *ExerciseEnvironment.json* file initializes the data from the patient's environment. In section **4.4** we will see in detail what the two files are made up of, as well as the units of the data. Then we see the initial nutritional conditions of the patient. This allows us to specify which nutrients he consumed before the simulation, and how long it has elapsed since that meal. We will see what these parameters correspond to in the **4.4** section.

```
...
"PatientConfiguration": {
  "PatientFile": "StandardMale.json",
  "Conditions": { "AnyCondition": [{
    "EnvironmentCondition": {
      "InitialEnvironmentalConditions": {
```

```
        "EnvironmentalConditionsFile": "./environments/ExerciseEnvironment.json"
        }
      }
    },{
      "PatientCondition" : {
        "ConsumeMeal": { "Meal": { "Nutrition": {
          "Carbohydrate": {
            "ScalarMass": { "Value": 200.0, "Unit": "g" }
          },
          "CarbohydrateDigestionRate": {
            "ScalarMassPerTime": { "Value": 1.083, "Unit": "g/min" }
          },
          "Fat": {
            "ScalarMass": { "Value": 20.0, "Unit": "g" }
          },
          "FatDigestionRate": {
            "ScalarMassPerTime": { "Value": 0.055, "Unit": "g/min" }
          },
          "Protein": {
            "ScalarMass": { "Value": 30.0, "Unit": "g" }
          },
          "ProteinDigestionRate": {
            "ScalarMassPerTime": { "Value": 0.071, "Unit": "g/min" }
          },
          "Calcium": {
            "ScalarMass": { "Value": 521, "Unit": "mg" }
          },
          "Sodium": {
            "ScalarMass": { "Value": 1604, "Unit": "mg" }
          },
          "Water": {
            "ScalarVolume": { "Value": 1.0, "Unit": "L"}
          }},
        "ElapsedTime": { "ScalarTime": { "Value": 2.0, "Unit": "hr"}}}}}}
      ]}},
...
```

Listing 2: Configuration of initial data.

Listing 3 describes Data Requests that correspond to the output data after simulation, stored in a CSV file. Currently, there are four supported types of data requests : Physiology System Data (useful for the heart rate of our runners), Environment Data, Substance Data, Equipment Data and Compartment Data. First, we will look at the physiological output data of the cyclist, in order to compare them with our own data. For each data, we have to choose the precision of the rounded values and give the unit of measurement.

```
...
"DataRequestManager": {
  "DataRequest": [
    {
      "DecimalFormat":{"Precision":2}, "Category":"Physiology",
      "PropertyName":"HeartRate", "Unit":"1/min"
    },
    {
      "DecimalFormat":{"Precision":3}, "Category":"Physiology",
      "PropertyName":"OxygenSaturation", "Unit":"unitless"
    },
    {
      "DecimalFormat":{"Precision":1}, "Category":"Physiology",
      "PropertyName":"SkinTemperature", "Unit":"degC"
    },
  ]},
...
```

Listing 3: Data requests for outputs.

Then, conditions give instructions to the engine to apply certain changes to the engine to simulate the specified conditions. For example in our case, we can set environmental conditions during the cyclists' outings. We can change the environment completely by including a file that stores this data, or we can change all environment settings manually, as in Listing **4**. To change the amount of nutrients consumed, you can also change all parameters as in the example.

```
...
"AnyAction": [{
  "AdvanceTime": {
    "Time": {
      "ScalarTime": {
        "Value": 50.0,
          "Unit": "s"
        }
      }
    }
  },
  { "EnvironmentAction": {
      "ChangeEnvironmentalConditions": {
        "EnvironmentalConditionsFile": "./environments/Hypobaric4000m.json"
      }
    }
  },
  { "PatientAction": {
      "ConsumeNutrients": {
        "Nutrition": {
          "Carbohydrate": {
            "ScalarMass": { "Value":415.0, "Unit":"g" }
          },
          "CarbohydrateDigestionRate": {
            "ScalarMassPerTime": { "Value":0.5, "Unit":"g/min" }
          },
          "Fat": {
            "ScalarMass": { "Value":83.0, "Unit":"g" }
          },
          "FatDigestionRate": {
            "ScalarMassPerTime": { "Value":0.055, "Unit":"g/min" }
          },
          "Protein": {
            "ScalarMass": { "Value":99.6, "Unit":"g"}
          },
          "ProteinDigestionRate": {
            "ScalarMassPerTime": { "Value":0.071, "Unit":"g/min" }
          },
          "Calcium": {
            "ScalarMass": { "Value":1.0, "Unit":"g"}
          },
          "Sodium": {
            "ScalarMass": { "Value":1.5, "Unit":"g"}
          },
          "Water": {
            "ScalarVolume": { "Value":3.7, "Unit":"L"}
          }
        }
      }
    }
  }
  ]
```

Listing 4: Patient actions and environment actions.

Among the other patient actions we will use is *Exercise*. It consists of increase the patient's metabolic rate leading to an increase in core temperature, cardiac output, respiration rate and tidal volume. This will therefore allow us to model the effort of a cyclist, we will describe it in more detail below.

Eventually we could look at the "Respiratory Fatigue" action. This action leads to respiratory muscle weakness caused by excessive effort relative to the strength and endurance of the muscles. Especially useful during a very long race that requires a lot of effort from a cyclist.

4.4 Parameters description.

4.4.1 Patient file.

We will first start with the patient description which is built using the patient description file. In particular, the patient file contains all the parameters that describe his state of health.

```json
{
  "Name": "StandardMale",
  "Age": {
    "ScalarTime": {
      "Value": 44.0,
      "Unit": "yr"
    }
  },
  "Weight": {
    "ScalarMass": {
      "Value": 170.0,
      "Unit": "lb"
    }
  },
  "Height": {
    "ScalarLength": {
      "Value": 71.0,
      "Unit": "in"
    }
  },
  "BodyFatFraction": {
    "Scalar0To1": {
      "Value": 0.21
    }
  },
  "DiastolicArterialPressureBaseline": {
    "ScalarPressure": {
      "Value": 73.5,
      "Unit": "mmHg"
    }
  },
  "SystolicArterialPressureBaseline": {
    "ScalarPressure": {
      "Value": 114.0,
      "Unit": "mmHg"
    }
  }
}
```

Listing 5: Initial Patient data

In Listing **5**, we describe how to implement this data.

<u>Age</u> : Age of the cyclist in **years**.

<u>Weight</u> : Weight of the cyclist in pound-mass, **lb**.

<u>Height</u> : Height of the cyclist in inches, **in**.

BodyFatFraction : Number between 0 and 1 associated with body fat.

DiastolicArterialPressureBaseline : Blood pressure is traditionally measured using auscultation with a mercury-tube sphygmomanometer. The diastolic blood pressure is the minimum pressure experienced in the aorta when the heart is relaxing before ejecting blood into the aorta from the left ventricle, the unit is **mmHg** and generally the value is approximated at 80 mmHg.

HeartRateBaseline : Heart rate is the speed of the heartbeat measured by the number of contractions (beats) of the heart per minute. The unit is in Hertz i.e. **1/min**.

RespirationRateBaseline : The respiratory rate is the rate at which breathing occurs. The unit is also in Hertz i.e. **1/min**.

SystolicArterialPressureBaseline : Systolic pressure refers to the maximum pressure within the large arteries when the heart muscle contracts to propel blood through the body, the unit is therefore **mmHg**.

4.4.2 Nutrients data.

In Listing **6**, we describe how to implement the amount of nutrients consumed by the patient initially [7]. These are the same data that we can change during the simulation. We can modify the nutrient data for each patient action. In our case, we can change these values at any time during a cyclist's race. Section **4.5.2** shows how to configure nutrient consumption at different points in a scenario.

```json
{
  "Carbohydrate": {
    "ScalarMass": {
      "Value": 390.0,
      "Unit": "g"
    }
  },
  "CarbohydrateDigestionRate": {
    "ScalarMassPerTime": {
      "Value": 0.5,
      "Unit": "g/min"
    }
  },
  "Fat": {
    "ScalarMass": {
      "Value": 90.0,
      "Unit": "g"
    }
  },
  "FatDigestionRate": {
    "ScalarMassPerTime": {
      "Value": 0.055,
      "Unit": "g/min"
    }
  },
  "Protein": {
    "ScalarMass": {
      "Value": 56.0,
      "Unit": "g"
    }
  },
  "ProteinDigestionRate": {
    "ScalarMassPerTime": {
      "Value": 0.071,
      "Unit": "g/min"
```

```
    }
  },
  "Calcium": {
    "ScalarMass": {
      "Value": 1.0,
      "Unit": "g"
    }
  },
  "Sodium": {
    "ScalarMass": {
      "Value": 1.5,
      "Unit": "g"
    }
  },
  "Water": {
    "ScalarVolume": {
      "Value": 3.7,
      "Unit": "L"
    }
  }
}
```

Listing 6: Initial Nutrients data

Endurance athletes such as cyclists have caloric needs that can range from $3,000$ to over $6,000$ kcal per day. Giving global advice on caloric bases makes no sense as these are individual needs depending on several factors including: your weight, your body composition, your energy situation, your physical activity... However, there are recommendations on the quantities of macronutrients to be provided to each individual, and in particular to sportsmen and women, which should be adapted to their physical activity. The information was found mainly on the website [23].

Carbohydrate : The quantity of carbohydrates to be provided to the athlete should be correlated to his or her training program. Carbohydrates will be the "effort fuel" in particular. In fact, carbohydrates are synthesised into glycogen in our body, its storage form. The recommendation according to body mass is :
- 5g to 7g / kg / day in case of moderate physical exercise.
- 7g to 10g / kg / day in case of intense training or during competitions.
- 10g to 12g / kg / day in case of a long competition or an intensive or difficult training block.

Proteins : As a source of energy, proteins play a major role in cell recovery, muscle and tissue strengthening. The recommendation according to body mass is, for the endurance athlete of : 1.2 to 2 g / kg / day (i.e. 84g to 140g per day for a 70kg person).
`"Protein": ScalarMass": "Value": 30.0 "Unit": "g" .`

Fat : Lipids are our body's main source of stored energy and are necessary for its proper functioning. The recommendation according to body mass is : 1 to 1.5g / kg / day (i.e. 70g to 105g per day for a 70kg person).
`"Fat": ScalarMass": "Value": 30.0 "Unit": "g" .`

We can choose how long the body consumes all the nutrients with the command ElapsedTime. We often hear about the "3-hour rule" in the world of sport and the pre-race or pre-match meal: 3 hours is the average time for the gastric emptying to have taken place for a large part of a sports meal. In fact, after 3 hours, the most energy-consuming part of the digestion process has passed and the blood flow to the stomach is no longer as great.
`"ScalarTime": "Value": 3.0 "Unit": "hr"`.

4.4.3 Environment file.

We can place ourselves in an initial environment, for example when climbing a slope (for climbers) or simply training in a straight line. This condition is implemented in Listing 7.

```json
{
  "SurroundingType": "Air",
  "AirVelocity": {
    "ScalarLengthPerTime": {
      "Value": 2.5, "Unit": "m/s"
    }},
  "AmbientTemperature": {
  "ScalarTemperature": {
    "Value": 35.0, "Unit": "degC"
    }},
  "AtmosphericPressure": {
    "ScalarPressure": {
      "Value": 760.0, "Unit": "mmHg"
    }},
  "ClothingResistance": {
    "ScalarHeatResistanceArea": {
      "Value": 3.0, "Unit": "clo"
    }},
  "Emissivity": {
    "Scalar0To1": {
      "Value": 0.9
    }},
  "MeanRadiantTemperature": {
    "ScalarTemperature": {
      "Value": 35.0, "Unit": "degC"
    }},
  "RelativeHumidity": {
    "Scalar0To1": {
      "Value": 0.45
    }},
  "RespirationAmbientTemperature": {
    "ScalarTemperature": {
      "Value": 35.0, "Unit": "degC"
    }},
  "AmbientGas": [
    {"Name": "Nitrogen", "Amount": {
      "Scalar0To1": { "Value": 0.7901
    }}},
    {"Name": "Oxygen", "Amount": {
      "Scalar0To1": { "Value": 0.2095
    }}},
    {"Name": "CarbonDioxide", "Amount": {
      "Scalar0To1": { "Value": 4.0E-4
    }}}
  ]
}
```

Listing 7: Initial Environment data

Airvelocity : In meteorology, wind is the horizontal movement of air. Its measurement consists of two parameters: its direction and its speed or force. The speed is commonly expressed in **km/h** or **m/s** [24]. This air flow is an integral part of cycling. When a weather report indicates the wind strength, this is taken at 10 meters above the ground. A cyclist can halve the real effect of the headwind on his progress. In a conventional position with the hands on the brake levers, you already have to produce 14 more watts to maintain a speed of 30 km/h to fight a 10 km/h wind than you would without the wind. Then 57 watts more if the wind is blowing at 20 km/h, or 129 watts at 30 km/h, 229 watts at 40 km/h, and 357 watts at 50 km/h. For some people, this is more than doubling the power they are able to sustain for only a few minutes, if they try to maintain this 30 km/h speed. They are then forced to slow down, or expend much more energy. We will consider several cases for the wind speed.

Ambiant Temperature : Ambient temperature is the air temperature of any object or environment where equipment is stored. The adjective ambient means "relating to the immediate surroundings." We will consider 15 degrees as a pleasant temperature, and go up to 25 on very hot days.

AtmosphericPressure : The actually measured atmospheric pressure varies around the normal atmospheric pressure, which is defined as $1,013.25$ hPa (760 mmHg) at "sea level" (mean level) and a temperature of 15°C.

Clothing Resistance : Clothing insulation is the thermal insulation provided by clothing and is measured in **clo**. One of the roles of clothing is to protect against the weather, especially the cold. We are all more or less sensitive to heat and cold, their effects are not supported by each of us in the same way, after a few years of practice it is possible to set an approximate scale that will guide in the choice of the equipment to wear [21]:
- Below 7°: use shoe covers and a helmet. (~ 1.10 clo)
- Below 10°C: winter gloves and a jersey under the fleece jacket. (~ 0.90 clo)
- Below 15°C: use a fleece jacket and long tights (never wear shorts below this temperature). (~ 0.76 clo)
- Above 15° in good weather, the shorts can be used, and the fleece jacket replaced by a long-sleeved jersey or equivalent. (~ 0.50 clo)
- Above 20°: short shorts and short-sleeved jersey. (~ 0.36 clo)

Mean Radiant Temperature : Mean radiant temperature (MRT) is a measure of the average temperature of the surfaces that surround a particular point, with which it will exchange thermal radiation. If the point is exposed to the outside, this may include the sky temperature and solar radiation.

Our **next objective** is to parameterise our model for a cyclist with given characteristics doing an exercise over a given period of time. We will obtain results using a .csv file, and we will analyse these data.

4.5 Using Pulse.

4.5.1 How to configure an exercise scenario.

Before we could use our data on Pulse, we have to create scenarios that vary over time. Indeed, to model the rides of a cyclist, it is necessary to simulate the different stages of the exercise. For this, we have to change the *AnyAction* part of *VitalsMonitor.json* by adding the fields described in Listing **8**.

```
    "AnyAction": [{
        "PatientAction": {
            "Exercise": {
                "Intensity": {
                    "Scalar0To1": { "Value":0.5 }
                }
            }
        }
    },
    {
        "AdvanceTime":{
            "Time":{
                "ScalarTime":{"Value":92.0,"Unit":"s"}
            }
        }
    }]
```

Listing 8: Exercise field in a scenario file

If the training consists of a series of different exercises, it is sufficient to repeat the field **8**. Let's decrypt the *Exercise* and *Time* fields to understand how to use them.

We have already introduced that *Exercise* is used to simulate the patient in physical activity, with different intensities.
The Pulse documentation gives Table **3** that shows approximate intensity of the exercise for some routine activities and the corresponding approximate mechanical power. The values in the table correspond to a body with a maximum operating capacity of 1200 watts.

Activity	Exercise Intensity	Mechanical Power (W)
All muscles in the body working all-out	1.0	1200
All-out cycling with resistance, sustainable for approx. 8 seconds	0.8	1000
Rowing 6.1 m/s on an ergometer	0.54	650
Running at about 7.1 m/s (16 mph)	0.36	430
'Typical' Level Cycling	0.1	120
Jogging at about 2.2 m/s (5 mph)	0.06	70
Rest	0.0	0

Table 3: Approximate intensity of exercise for certain routine activities and the corresponding approximate mechanical power (Source : Pulse [11])

The exercise capacity of the body is physiologically and psychologically limited. However, this limitation is purely physiological and is due to the fatigue model [11]. For the time being, the maximum energy storage values and maximum filling rates of the fatigue model do not change with physical fitness, dietary levels and body composition. The Pulse team has several projects underway and implementing this part is one of them. As long as we do not know these data precisely, the intensity of exercise takes real values ranging from 0.0 to 0.5. Otherwise, some of the patient's parameters take negative values and the simulation stops because the patient is "unhealthy".

Then, *Time* field is used to set the duration of the exercise described previously. The unit of measurement may be second (s) or minute (min).

Our objective will be to reliably simulate a cyclist for two exercises : a climb of the pass (variation of atmospheric pressure) and a split training (alternation of strong and low effort intensity).

4.5.2 How to configure nutrient consumption in a scenario.

The cyclist may drink or eat during a race, which affects his or her abilities. In order to take into account the nutrients consumed during a simulation, we can use the field described in Listing **9**. This is an action performed by the "patient", to be added to the scenario file.

```
"AnyAction": [{
    "PatientAction": {
        "ConsumeNutrients": {
            "Nutrition": {
              "Carbohydrate": {
                "ScalarMass": { "Value":415.0, "Unit":"g" }
              },
              "CarbohydrateDigestionRate": {
                "ScalarMassPerTime": { "Value":0.5, "Unit":"g/min" }
              },
              "Fat": {
                "ScalarMass": { "Value":83.0, "Unit":"g" }
              },
```

```
                "FatDigestionRate": {
                  "ScalarMassPerTime": { "Value":0.055, "Unit":"g/min" }
                },
                "Protein": {
                  "ScalarMass": { "Value":99.6, "Unit":"g"}
                },
                "ProteinDigestionRate": {
                  "ScalarMassPerTime": { "Value":0.071, "Unit":"g/min" }
                },
                "Calcium": {
                  "ScalarMass": { "Value":1.0, "Unit":"g"}
                },
                "Sodium": {
                  "ScalarMass": { "Value":1.5, "Unit":"g"}
                },
                "Water": {
                  "ScalarVolume": { "Value":3.7, "Unit":"L"}
                }
              }
          }
      }
  },
  {
      "AdvanceTime": {
          "Time": {
              "ScalarTime": { "Value": 60.0, "Unit": "min" }
          }
      }
  }]
```

Listing 9: Nutrient consumption in a scenario file

4.5.3 How to execute a scenario.

After implementing the desired scenario in a JSON file, it must be stored in the folder */install/bin/*, which is the location of the *PulseScenarioDriver* executable. The scenario can then be executed using the Linux command described in Listing **10**.

```
./PulseScenarioDriver VitalsMonitor.json
```

Listing 10: Execution command

The software then checks that the initialized patient is stable before starting the simulation. After confirmation of the stable state, the simulation displays in real time in the terminal what is happening. For example, Listing **11** describes the actions from Listing **4** and changes in the patient's body (we just added 60 minutes after conditions changes).

```
...
[INFO] : [0(s)] Executing Scenario
[INFO] : [0(s)] [Action]
        Advance Time : 50(s)
[INFO] : [50.02(s)] [Action] 50.02(s), Environment Action : Change Environmental
    Conditions
        Environmental Conditions File: ./environments/Hypobaric4000m.json
[INFO] : [50.02(s)] [Action] 50.02(s), Patient Action : Consume Nutrients
        Charbohydrates: 415(g)
        Charbohydrates Digestion Rate: 0.5(g/min)
        Fat: 83(g)
        Fat Digestion Rate: 0.055(g/min)
        Protein: 99.6(g)
        Protein Digestion Rate: 0.071(g/min)
        Calcium: 1(g)
        Sodium: 1.5(g)
```

```
        Water: 3.7(L)
[INFO] : [50.02(s)] [Action]
        Advance Time : 60(min)
...
[INFO] : [60.74(s)] [Event] 60.74(s),  Patient has Hypoxia
...
[INFO] : [3650.04(s)] It took 486.788s to run this simulation
```

Listing 11: Display on the terminal during execution

The selected output data are stored in a CSV file generated by the simulation. The name of file is like the scenario file by adding “Results”. For example, the results presented in Listing **12** are stored in *VitalsMonitorResults.csv*.

```
Time(s),HeartRate(1/min),OxygenSaturation,SkinTemperature(degC)
...
55.60,75.00,0.966,32.7
55.62,75.00,0.966,32.7
55.64,76.92,0.966,32.7
55.66,76.92,0.965,32.7
55.68,76.92,0.965,32.7
...
```

Listing 12: Results in CSV file

4.6 Numerical experiments.

For the moment, we have considered two interesting scenarios in the field of cycling: an exercise with progressive intensity and a VO2max exercise.
These simulations were performed by a high level cyclist profile described in Table **5**, with environmental parameters described in Table **4**.

Finally, in these examples we considered that a cyclist consumes between 60 and 90 grams of carbohydrate per hour, between 0.3 and 0.6 grams of sodium per hour, and between 0.5 and 0.75 liters of water per hour. The minimum amount of nutrients is chosen when the intensity of effort is low, and the maximum amount is set when the effort is high.

Parameters	**Values**
AirVelocity (m/s)	2.5
AmbientTemperature (degC)	15
AtmosphericPressure (mmHg)	760
ClothingResistance (clo)	0.50
Emissivity (Scalar between 0 and 1)	0.9
MeanRadiantTemperature (degC)	15
RelativeHumidity (Scalar between 0 and 1)	0.45
RespirationAmbientTemperature (degC)	15
Nitrogen (Scalar between 0 and 1)	0.7901
Oxygen (Scalar between 0 and 1)	0.2095
CarbonDioxide (Scalar between 0 and 1)	4.0E-4

Table 4: Parameters associated to an exercise environment

4.6.1 Progressive intensity scenario.

The first training consisted of the sequence of several effort-rest steps with an effort that increases progressively:
- 30 seconds of rest (intensity 0.0)
- 4 minutes of effort with intensity 0.05
- 2 minutes rest (intensity 0.01)
- 4 minutes of effort with intensity 0.15
- 2 minutes rest (intensity 0.01)
- 4 minutes of effort with intensity 0.25
- 2 minutes rest (intensity 0.01)
- 4 minutes of effort with intensity 0.35
- 2 minutes rest (intensity 0.01)

To perform this simulation, we use the *ExerciseStages.json* file stored in the *data* folder.

Figure **38** shows similarities between heart rates and exercise intensity. First, the heart rate increases at about the same time as the intensity of exercise. We see the same phenomenon when the intensity of exercise decreases. Finally, these curves show a strong correlation between the intensity of physical effort and heart rate. This result is what we expected.

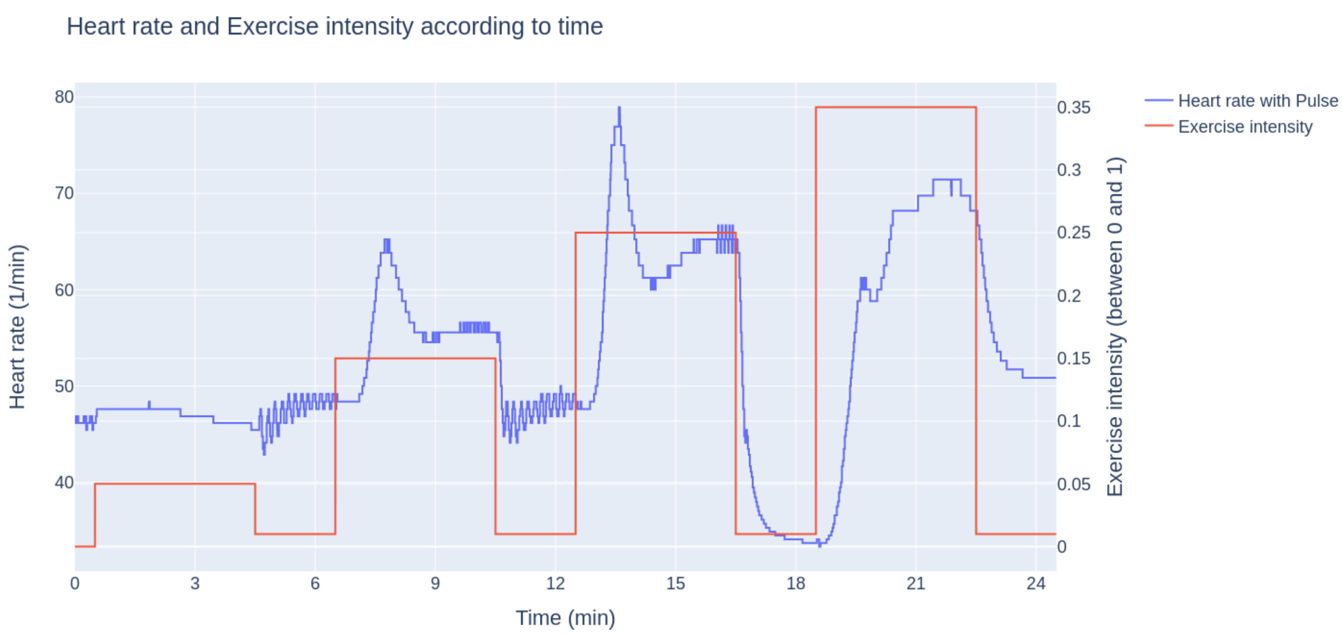

Figure 38: Progressive training : Heart rate and Exercise intensity according to time

4.6.2 VO2max scenario.

The second training is a VO2max exercise. This is the maximum amount of oxygen that the body is able to extract from the air and then carry to the muscle fibers. The VO2max is a performance indicator because it gives an indication of the level of fitness and endurance of an athlete [15].

We defined the scenario like this:
- 30 seconds of rest
- 6 minutes of effort with intensity 0.5
- 7 minutes rest (intensity 0.0).

The scenario's implementation is described in Listing **13** but we can find it in the file *ExerciseVO2max.json* stored in *data* folder.

```
"AnyAction": [{
  "AdvanceTime": {
    "Time": {
      "ScalarTime": {
        "Value": 30.0,
        "Unit": "s"
      }
    }
  }
}, {
  "PatientAction": {
    "Exercise": {
      "Intensity": {
        "Scalar0To1": {
          "Value": 0.5
        }
      }
    }
  }
}, {
  "AdvanceTime": {
    "Time": {
      "ScalarTime": {
        "Value": 6.0,
        "Unit": "min"
      }
    }
  }
}, {
  "PatientAction": {
    "Exercise": {
      "Intensity": {
        "Scalar0To1": {
          "Value": 0.0
        }
      }
    }
  }
}, {
  "AdvanceTime": {
    "Time": {
      "ScalarTime": {
        "Value": 7.0,
        "Unit": "min"
      }
    }
  }
}]
```

Listing 13: Exercise VO2max

Let's look at the figure **39**. We notice a peak increase in heart rate during the increase in exercise intensity. However, the heart rate drops immediately afterwards while the physical effort is very high (0.5). This may be due to the fact that the body gets used to the effort, which helps to lower the heart rate. At the end of the exercise we observe an almost linear descent of the frequency. Finally, these results are consistent with reality.

We can also note that the simulated cyclist reaches his heart rate baseline from about 12 minutes, i.e. after 5 minutes 30 seconds of rest.

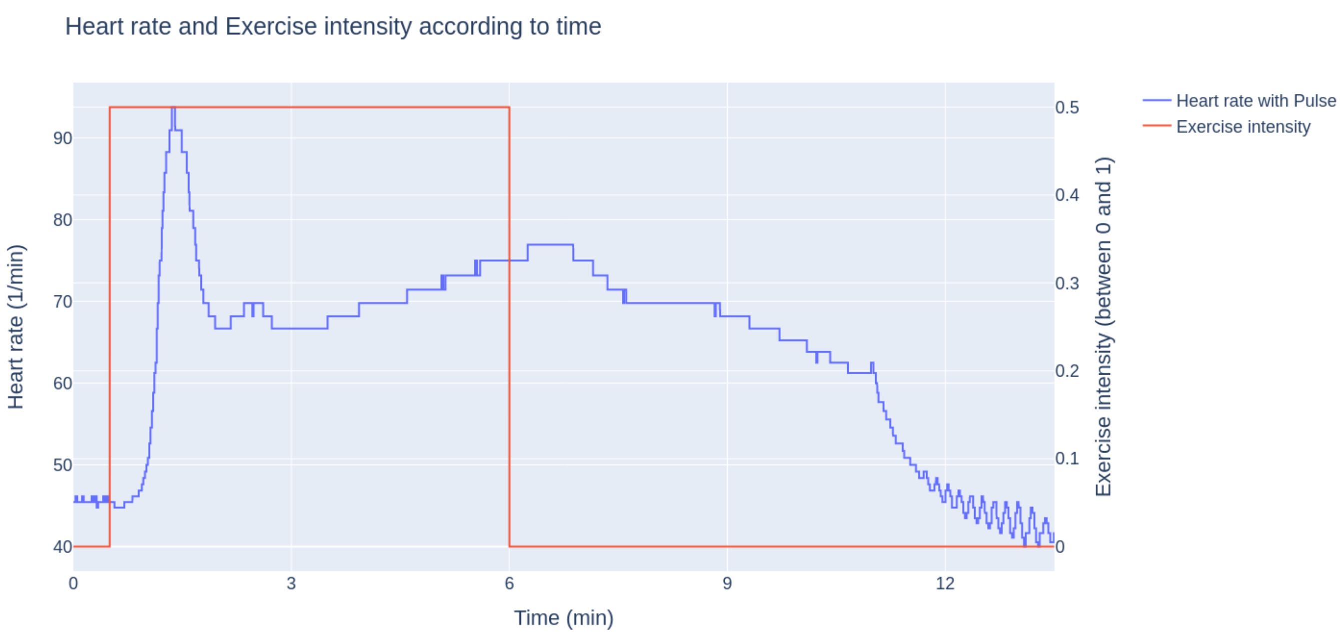

Figure 39: VO2 max : Heart rate and Exercise intensity according to time

5 Cyclist's effort simulation.

5.1 Context.

This part aims to make a link between the two projects. In fact, we will compare the results obtained by the Pulse simulations and compare them with the data we actually have on the cyclist. We would like to know to what **extent** the **simulations** and **modelling** of cyclists' **efforts with Pulse are consistent or not** with those of the data part. To do this, we consider a unique cyclist described in Table **5** to which correspond the most current data sets and on which we perform the simulations.

Therefore, we set up a python file *create_json.py*, stored in the "data" folder, that includes some functions implemented in the data parts and allows us to do different operations. The main goal is to **generate a JSON file** from the percentages of attaining a record power for the different times, and which can be directly evaluated by Pulse models.
We show in the listing **14** the command to execute this file and its different options.

```
create(trip=2, max_percentage=40, plot_diagrams=True, plot_heart_rates=False, export=
    False)
```

Listing 14: Command to execute the file «create_json.py».

Firstly, the variable *trip* represents the choice of activity, in this case it is the trip nr. 2. Then, *max_percentage* allows us to fix the maximum percentage above which we stop the simulation. In fact, we have noticed that from a certain percentage onwards, the simulations with Pulse crashed. This has been explained in details in **4.5.1** part. Thus, in this section we have focused on scenarios with a maximum intensity of less than 0.5. In order to select scenarios from our data, we consider the beginnings of the exercises, because these often contain the lowest percentages, including those below 50%.

At this point, we can already create the JSON file. Listing **15** contains an extract from trip nr. 0, created by our python file. The resulting file is therefore called *new_scenario.json*.


```
{
  "Name": "ExerciseStages",
  "Description": "Tests exercise physiology by increasing intensity.",
  "PatientConfiguration": {
    "PatientFile": "HighPerformanceCyclist.json",
    "Conditions": { "AnyCondition": [{
      "EnvironmentCondition": {
        "InitialEnvironmentalConditions": {
          "EnvironmentalConditionsFile": "./environments/ExerciseEnvironment.json"
        }
      }
    }]}
  },
  "DataRequestManager": {
    "DataRequest": [
      {
        "DecimalFormat":{"Precision":2}, "Category":"Physiology",
        "PropertyName":"HeartRate", "Unit":"1/min"
      },
      {
        "DecimalFormat":{"Precision":2}, "Category":"Physiology",
        "PropertyName":"TotalWorkRateLevel", "Unit": "unitless"
      },
      {
        "DecimalFormat":{"Precision":2}, "Category":"Physiology",
        "PropertyName":"FatigueLevel", "Unit": "unitless"
      },
      {
        "DecimalFormat":{"Precision":2}, "Category":"Physiology",
        "PropertyName":"AchievedExerciseLevel", "Unit": "unitless"
      }
    ]},
  "AnyAction": [{
    "PatientAction": {
      "Exercise": {
        "Intensity": {
          "Scalar0To1": {
            "Value": 0.000000
          }
        }
      }
    }
  }, {
    "AdvanceTime": {
      "Time": {
        "ScalarTime": {
          "Value": 1.0,
          "Unit": "s"
        }
      }
    }
  }, {
    "PatientAction": {
      "Exercise": {
        "Intensity": {
          "Scalar0To1": {
            "Value": 0.000000
          }
```

```
        }
      }
    }
  }, {
    "AdvanceTime": {
      "Time": {
        "ScalarTime": {
          "Value": 4.0,
          "Unit": "s"
        }
      }
    }
  }, {
    "PatientAction": {
      "Exercise": {
        "Intensity": {
          "Scalar0To1": {
            "Value": 0.137222
          }
        }
      }
    }
  }, {
    "AdvanceTime": {
      "Time": {
        "ScalarTime": {
          "Value": 5.0,
          "Unit": "s"
        }
      }
    }
  }, {
    "PatientAction": {
      "Exercise": {
        "Intensity": {
          "Scalar0To1": {
            "Value": 0.328826
          }
        }
      }
    }
  }, {
    "AdvanceTime": {
      "Time": {
        "ScalarTime": {
          "Value": 5.0,
          "Unit": "s"
        }
      }
    }
  }]
}
```

Listing 15: new_scenario.json for trip nr. 0

Furthermore, the arguments *plot_diagrams* and *plot_heart_rates* allow us to generate the diagrams of the first part of the project **Performance factors**, respectively the heart rate in relation to time on the interval which we will consider for this exercise.
These graphs will be used in the upcoming section **5.3**.

Finally, by choosing *export* as *True*, the python file exports the heart rates for this time interval as a .csv file. In the case of the trip nr. 0 the file will be named «heart_rates_0.csv». This can be useful for us in order to plot these.

5.2 Example of a high-performance cyclist.

Thanks to Olivier Mazenot, we collected data that accurately portrays a high performance cyclist. This data allowed us to create the profile described in Table **5**. We note that **DiastolicArterialPressureBaseline** and **SystolicArterialPressureBaseline** are much lower than for non-sports people (on average 120 and 80 mmHg respectively).

Parameters	**Values**
Age (year)	27
Weight (kg)	83
Height (cm)	193
BodyFatFraction (Scalar between 0 and 1)	0.08
HeartRateBaseline (1/min)	45
RespirationRateBaseline (1/min)	8
DiastolicArterialPressureBaseline (mmHg)	60
SystolicArterialPressureBaseline (mmHg)	100

Table 5: Description of a high-level cyclist (Source : Olivier Mazenot)

5.3 Pulse and data comparison. (Completed during Mélissa's internship)

In this section, we have generated three scenarios associated with the second, sixth and fifty-third activities, stored in our data. The first two are time trials and the last one is an in-line race. Let us compare the graphs obtained from our data with the results simulated by the Pulse model.

In order to compare the real data with the data generated by Pulse, we use MAE (Mean Absolute Error) and MSE (Mean Square Error) errors.

The MAE is defined as: $\dfrac{1}{n}\sum_{i}^{n}|y_i - y_{i_{Pulse}}|$

The MSE is defined as follows: $\dfrac{1}{n}\sum_{i}^{n}(y_i - y_{i_{Pulse}})^2$

These two calculations express the model's mean prediction error in units of the variable of interest. Both metrics can range from 0 to ∞ and are indifferent to the direction of errors [25].

Since errors are squared before averaging, the MSE gives relatively high weight to significant errors. This means that the MSE is more useful when large errors are particularly undesirable.

5.3.1 Activity nr. 2 (time-trial race).

To simulate the second activity with the Pulse model, we initialized the environmental parameters we knew from the real data.
Thus, we obtained parameters described in Table **6**. The corresponding JSON file is named **CyclistEnvironment_2.json** and is stored in data folder.

Parameters	Values
AirVelocity (m/s)	2.5
AmbientTemperature (degC)	11.7
AtmosphericPressure (mmHg)	722.3
ClothingResistance (clo)	0.513
Emissivity (Scalar between 0 and 1)	0.95
MeanRadiantTemperature (degC)	11.7
RelativeHumidity (Scalar between 0 and 1)	0.45
RespirationAmbientTemperature (degC)	11.7
Nitrogen (Scalar between 0 and 1)	0.7901
Oxygen (Scalar between 0 and 1)	0.2095
CarbonDioxide (Scalar between 0 and 1)	4.0E-4

Table 6: Environment parameters associated to activity nr. 2

The graph in Figure **40** was obtained from the data and functions used in the **3.3.2** section. It represents the percentage of record power achieved for activity nr. 2. These powers allow us to model the intensity of exercise in our scenarios. For the simulations, we selected the data of the first 10 minutes in order not to get too close to the maximum exercise intensity of 0.5. Note that sticks number 5 to 7 are averages obtained from values of the previous and next sticks. In fact, the cyclist's power data had not been recorded over that period.

Figure 40: Percentage of record power from real data

Figure **41** represents the exercise intensity (in green), the heart rate of the cyclist with Pulse model (in blue) and the heart rate stored with real data (in red). We see similarities between the intensity curve and the heart rate curve with our data. Indeed, the peaks of increased physical effort are mostly represented by an increased heart rate. We note that two peaks fall between 120 and 180 seconds, and between 450 and 630 seconds, this is due to a lack of data on these periods.

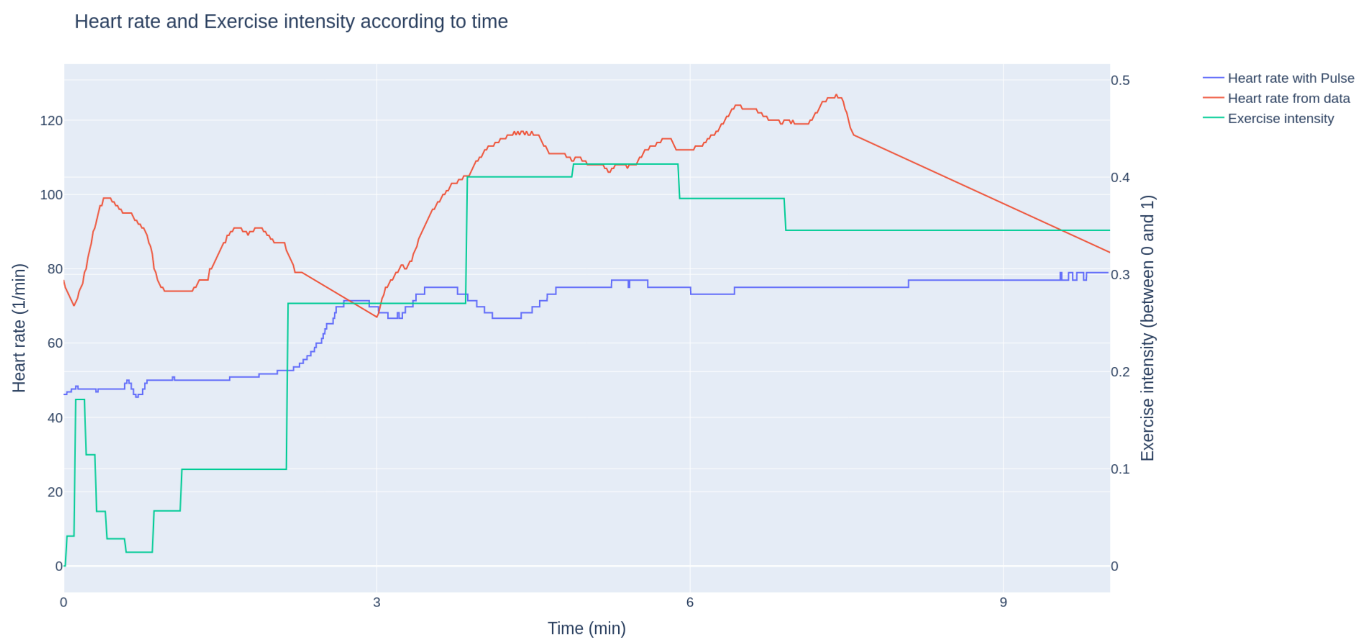

Figure 41: Heart rate baseline set at 45 for the Pulse model

In the case of Figure **41**, we get an MAE equal to 32.311468168038104 and an MSE equal to 1204.659164633374. Errors are particularly high.

Let us look at the results of the simulation with the Pulse model. We notice two peaks in heart rate increase between 150 and 250 seconds. Between 190 and 290 seconds we see that the two heart rate curves are similar if we translate Pulse's to the right a little bit. With the model, we see a small peak increase in heart rate at the start of the scenario, contrary to the real data which show a large peak. However, the remainder of the heart rate values are fully consistent with the exercise intensity values.

Analysis.

Finally, we find some similarities between the data obtained by the model and our real data. However, the maximum heart rate for our data is about 128 beats per minute, compared to about 79 for the Pulse model. We also see that the minimum frequency is 70 beats per minute for the real data compared to 46 for the simulation. The high frequency with our data may be explained by the fact that the cyclist did not start training with his usual resting heart rate (45), but rather 74 instead.

Now, let's look at the Pulse heart rate curve described in Figure **42**. This time, the heart rate baseline is set at 74, as in our data. Note that the phenomenon is reversed compared to Figure **41** : the Pulse data takes much higher values than our data. Indeed, the Pulse model reaches heart rates up to 176 beats per minute, compared to 128 for our data. On the other hand, the overall trend of the two heart rate curves is now more similar than in Figure **41**.

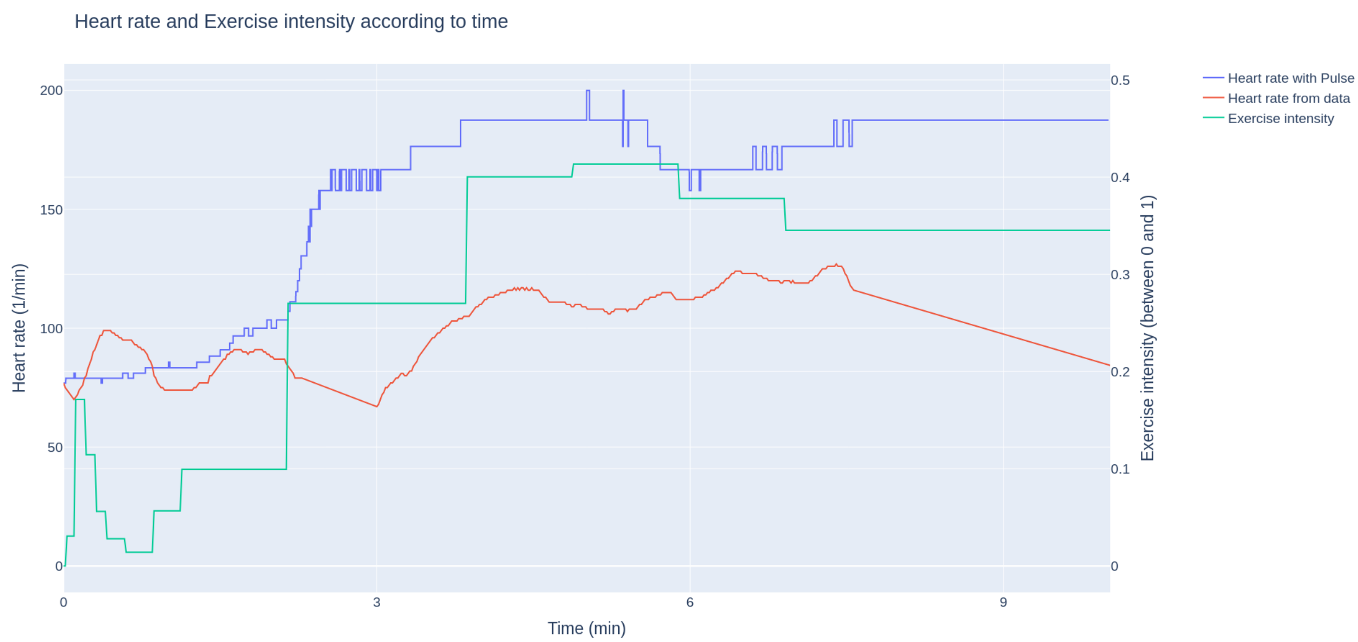

Figure 42: Heart rate baseline set at 74 for the Pulse model

In the case of Figure **42**, we get an MAE equal to 60.553317120298495 and an MSE equal to 4699.17399465636. Errors are even higher than in the previous case, so focus on a lower HeartRateBaseline.

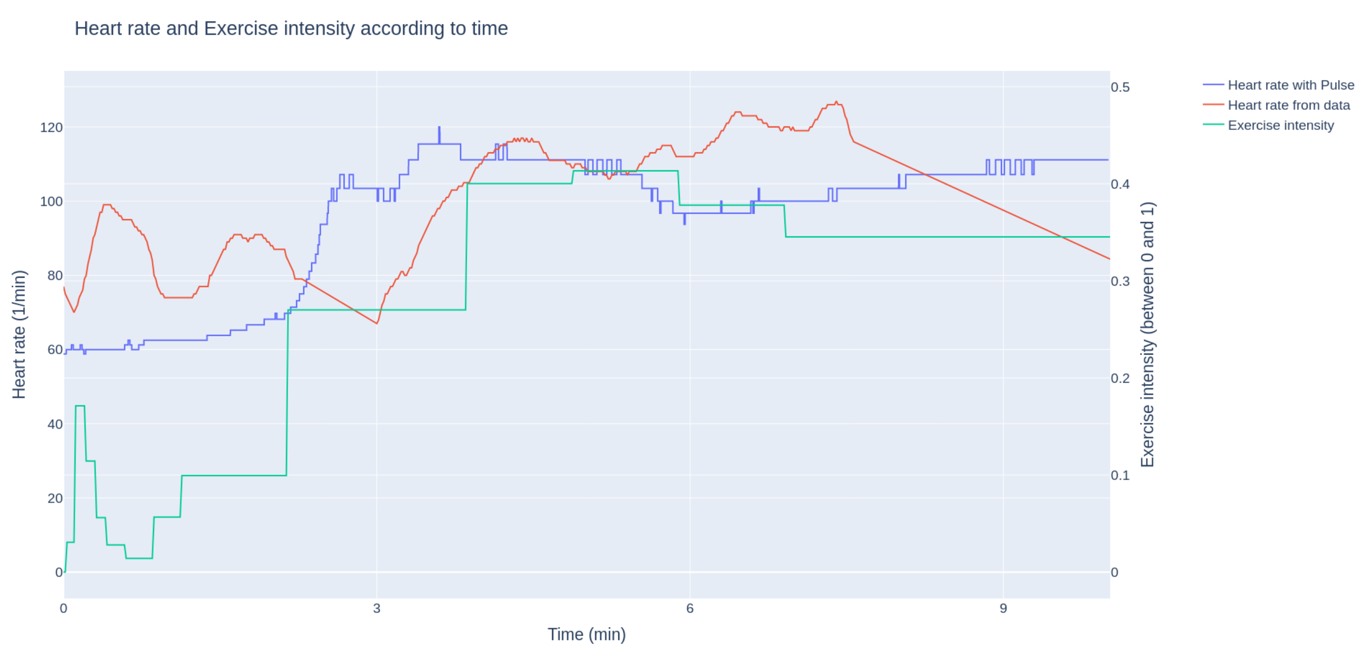

Figure 43: Heart rate baseline set at 57 for the Pulse model

Finally, we note that the lowest error is reached for a HeartRateBaseline equal to 57 since we get an MAE of 13.744604390845199 and an MSE of 334.8286537062331. Figure **43** lets you see that the values of the two graphs are closer to each other than the other ones. However, these results are not satisfactory.

5.3.2 Activity nr. 6 (time-trial race).

As with the previous outing, we use the real data to initialize the environment parameters of the Pulse model. Table **7** describes these parameters. The corresponding JSON file is named **CyclistEnvironment_6.json** and is stored in data folder.

Parameters	Values
AirVelocity (m/s)	2.5
AmbientTemperature (degC)	10.1
AtmosphericPressure (mmHg)	692.0
ClothingResistance (clo)	0.513
Emissivity (Scalar between 0 and 1)	0.95
MeanRadiantTemperature (degC)	10.1
RelativeHumidity (Scalar between 0 and 1)	0.45
RespirationAmbientTemperature (degC)	10.1
Nitrogen (Scalar between 0 and 1)	0.7901
Oxygen (Scalar between 0 and 1)	0.2095
CarbonDioxide (Scalar between 0 and 1)	4.0E-4

Table 7: Environment parameters associated to activity nr. 6

As before, the graph in Figure **44** was obtained from the data and functions used in the **3.3.2** section. It allows us to select the first 10 minutes of activity nr. 6 to obtain power data less than 0.5.

Figure 44: Percentage of record power from real data

Figure **45** shows that training is an exercise with progressive intensity. For real data, the peak that drops to 120 seconds should not be considered as it is due to a lack of data for heart rate at that time. In this curve we see a general upward trend in heart rate, but between 200 and 300 seconds the rate curve decreases. This observation is surprising because the intensity of effort increases linearly over this period.

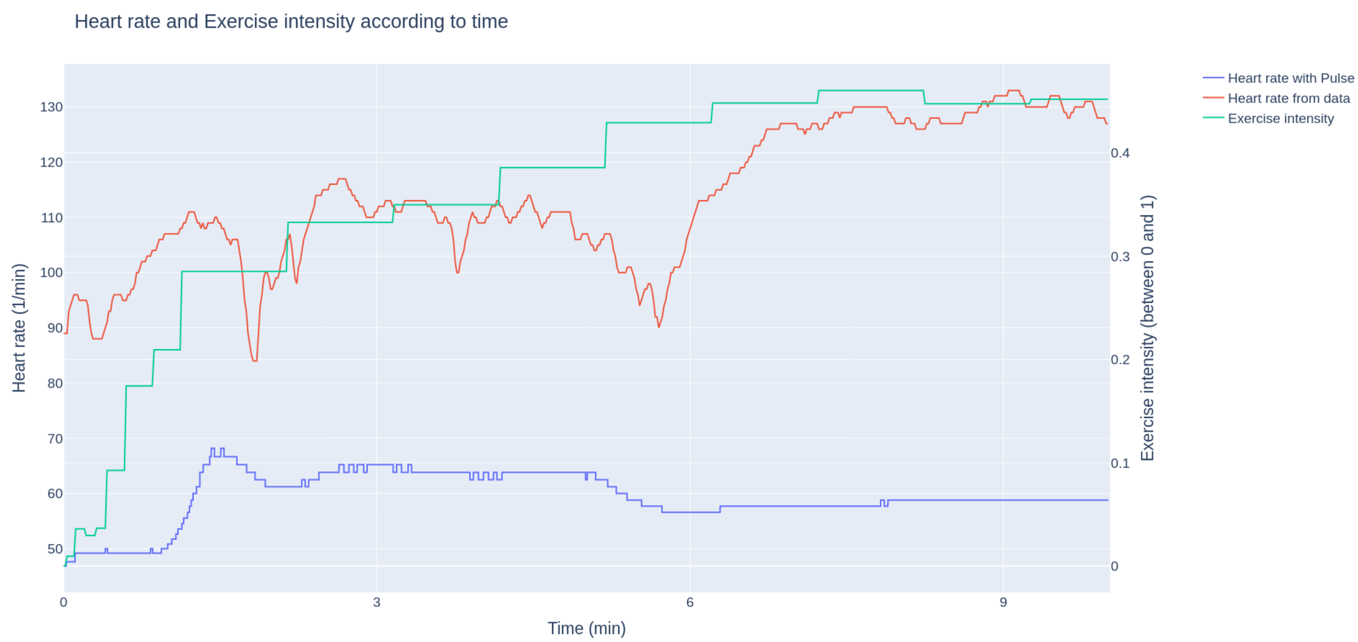

Figure 45: Heart rate baseline set at 45 for the Pulse model

In the case of heart rate baseline set at 45, we get an MAE equal to 54.318360262517906 and an MSE equal to 3113.7389208981576. We see that errors are particularly high.

Analysis.

An interesting observation with the Pulse curve is the decrease in heart rate over the period 200-350 seconds because it is the same as previously observed.
The peak at about 100 seconds could be explained by a sudden increase in exercise intensity. However, it is surprising to observe a heart rate of about 60 beats per minute from 350 seconds, as the effort expended by the cyclist is greater than 0.4. As before, we see a large difference in the extremum values of the heart rate. Indeed, heart rates belong to the range $[90, 135]$ for the real data, compared to $[45, 85]$ for the simulation.

Now, let's look at the Pulse heart rate curve described in Figure **46**. This time, the heart rate baseline is set at 85, as in our data. We note that the phenomenon is reversed compared to Figure **45** because the Pulse data takes much higher values than our data. Indeed, the Pulse model reaches heart rates up to 200 beats per minute, compared to 133 for our data. However, it is interesting to note that the general trend of the two curves is more similar than in Figure **45**.

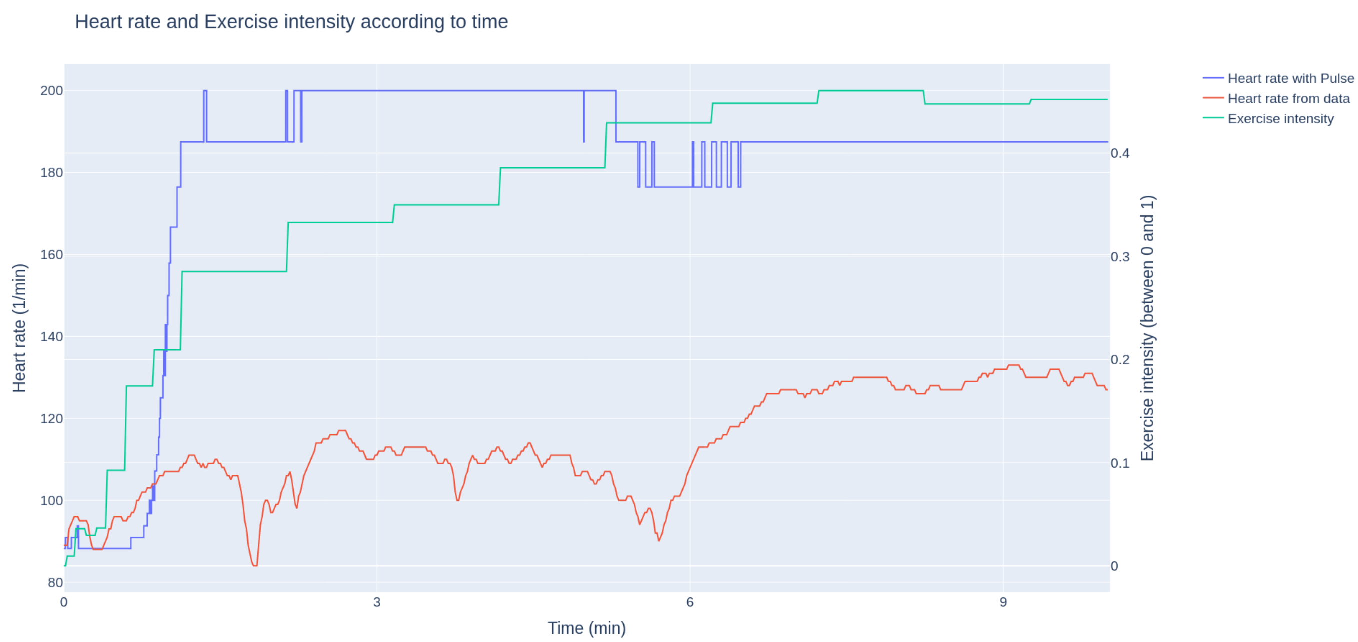

Figure 46: Heart rate baseline set at 85 for the Pulse model

In the case of heart rate baseline set at 85, we get an MAE equal to 68.22068028117401 and an MSE equal to 5271.120388393244. Errors are even higher than in the previous case, so focus on a lower HeartRate-Baseline.

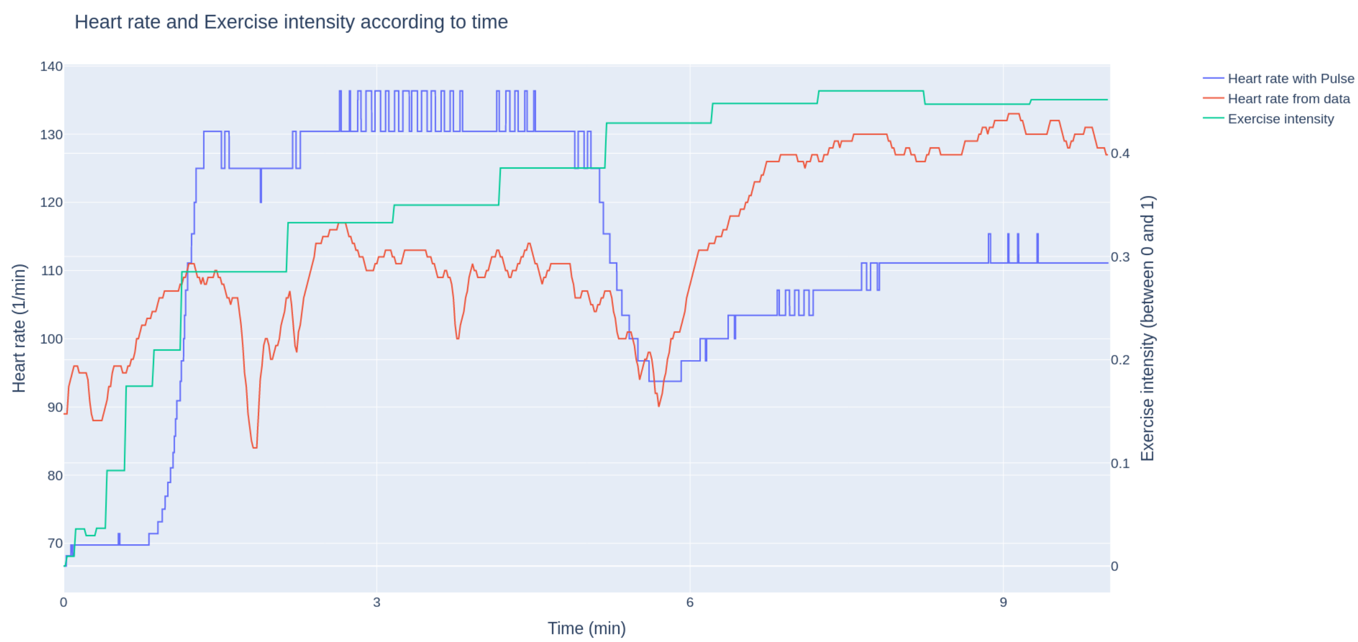

Figure 47: Heart rate baseline set at 61 for the Pulse model

Finally, we note that the lowest error is reached for a HeartRateBaseline equal to 61 since we get an MAE of 19.19520704933871 and an MSE of 413.23051006096557. Figure **47** shows some similarities between the two graphs but it is not satisfactory.

5.3.3 Activity nr. 53 (in-line race).

The 53th activity is an in-line race, but it may be interesting to see if the simulation gives more satisfactory results than the time trials.

The average temperature over the entire race is 13.1 degrees Celsius and the average altitude is 699.0 mmHg.

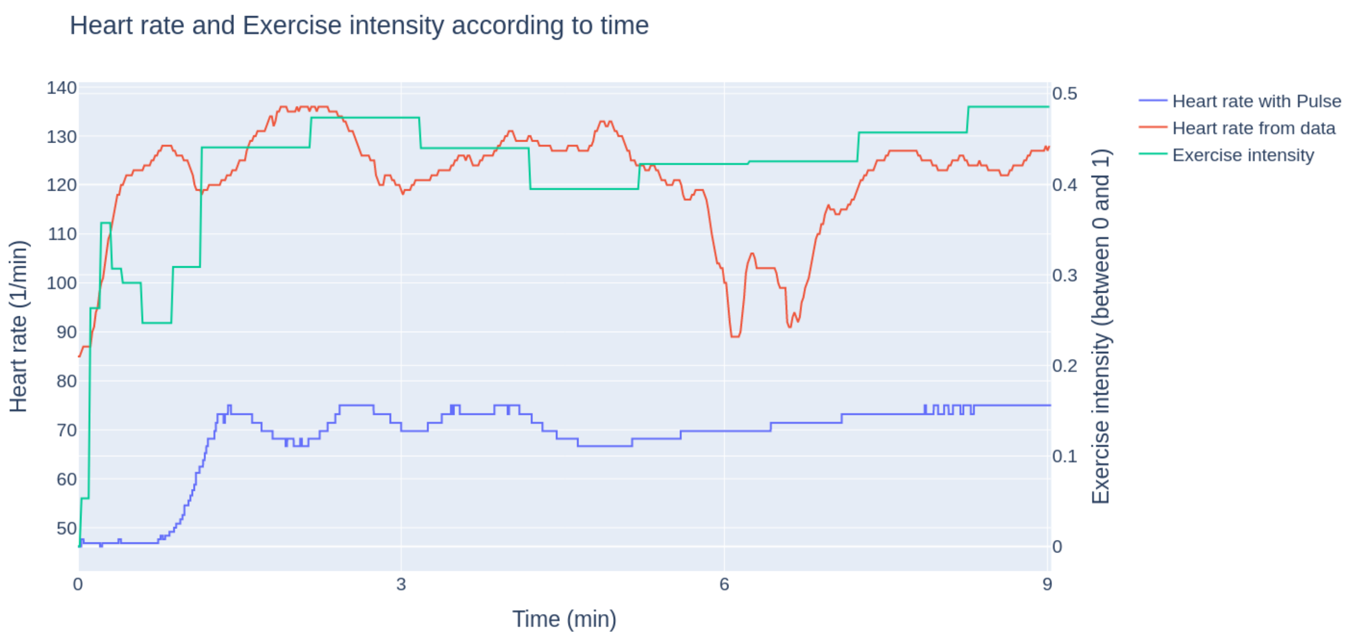

Figure 48: Heart rate baseline set at 45 for the Pulse model

In the case of Figure **48**, we get an MAE equal to 52.64772014787431 and an MSE equal to 2915.846259560074. We see that the errors are not smaller than for the previous releases.

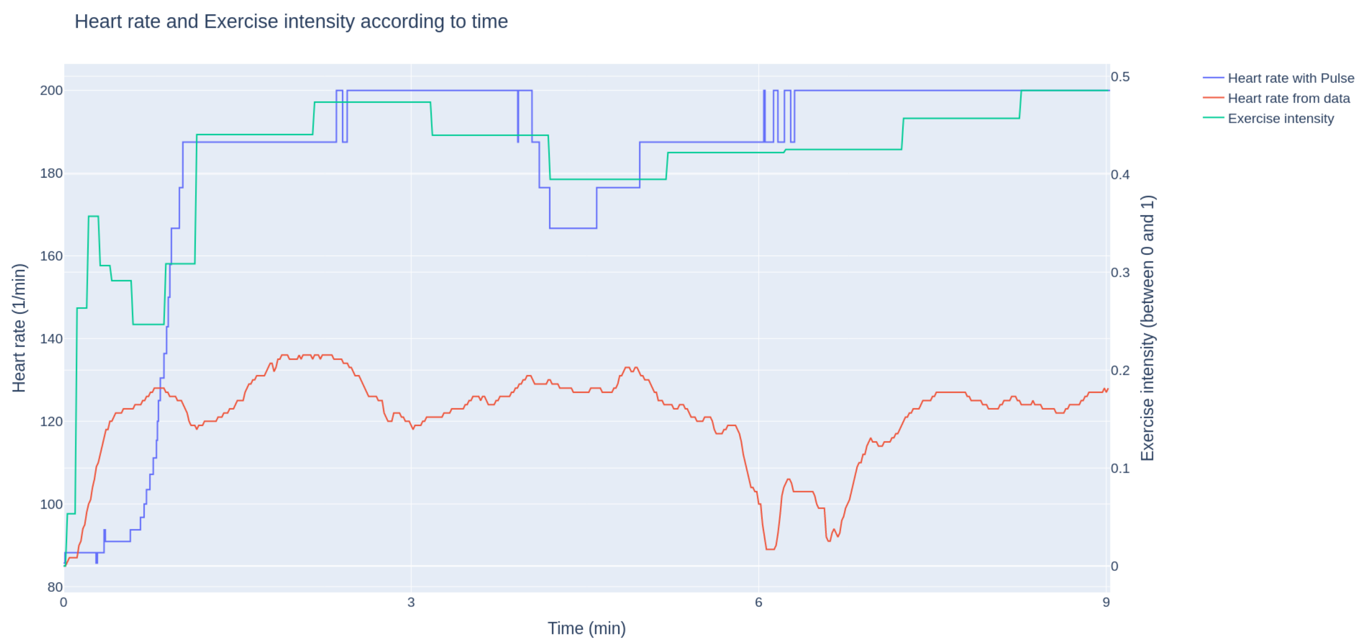

Figure 49: Heart rate baseline set at 82 for the Pulse model

In the case of heart rate baseline set at 82 (see Figure **49**), we get an MAE equal to 65.00489722735675 and an MSE equal to 4707.005288425139. Errors are even higher than before so we should try to lower the value of HeartRateBaseline.

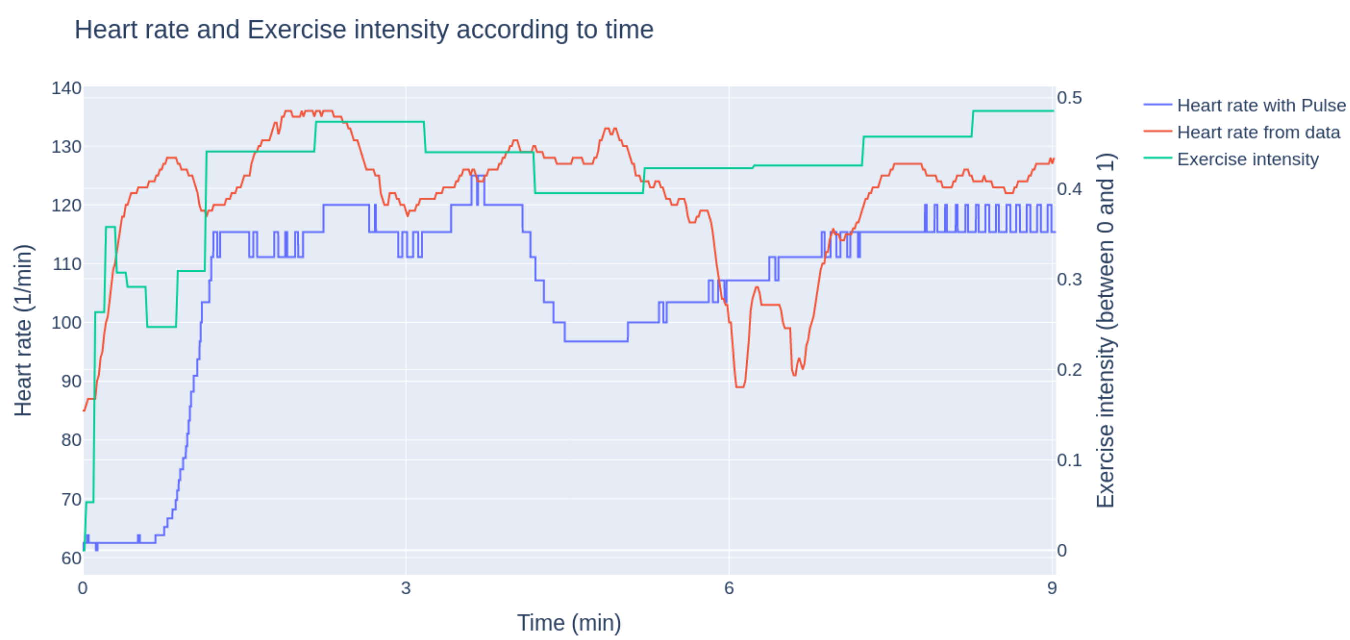

Figure 50: Heart rate baseline set at 59 for the Pulse model

Finally, we note that the lowest error is reached for a heart rate baseline equal to 59 since we get an MAE of 16.976271349353055 and an MSE of 508.87151058780046. Figure **50** shows that the values of the two graphs are closer to each other than the other ones but these results are not satisfactory.

5.4 Conclusion.

Finally, the simulation of training based on our data allowed us to show some similar graphic trends. However, we have not yet obtained a satisfactory scale for Pulse heart rate values.

5.5 Future work.

The next goal will be to study simulations with maximum and average exercise intensity, and to compare output data for different scenarios. We could also look more closely at what point in a false simulation the cyclist is "alive".

6 Role of nutrition in the cyclist's performance. (Mélissa's internship)

This section deals with two different lines of research.

Objective 1

- Reconstitute information on cyclists' nutrition. It will be a question of understanding the value of supervised nutrition to prepare for a race, and knowing the recommended amounts to achieve the best performance. The focus will be on the entire food process, i.e. before, during and after effort.
- Studying the nutritional aspect of Pulse models : to understand what the functions of these systems are and what principles are used to construct them. It will also be necessary to show that the models are interrelated and sometimes dependent on one another.

Objective 2

- Simulate scenarios different from the classic scenario, by changing the energy drink consumption or changing the amount of food before the race. The objective is to compare Pulse output data from these unconventional scenarios with a classic high performance cyclist race. This will allow us to observe the effects of a change in nutrient consumption on the cyclist's vital data.

6.1 The cyclist's diet

Getting hydrated and eating well is part of a cyclist's habit. Nutritional monitoring is as important before a race as during and after it. This is necessary to make the most of training outings and improve your performance during events. With experience, they manage to manage well, namely when to eat according to the intensity and length of the effort, their feelings, the weather... However, the teams of high performance cyclists are advised by a specialized nutritionist.

Table **8** corresponds to a list of the different nutrients and their role in the human body.

Nutrient	Role in body
Carbohydrate	Maintains energy reserves (particularly glycogenic)
Protein	Maintains tissue structures (including muscles)
Fat	Aid to the energy input
Calcium	Helps for proper muscular function
Magnesium	Helps to anticipate magnesium losses included in sweat
Sodium	Helps to compensate for sodium losses included in sweat
Potassium	Helps for proper neuromuscular function
Citrate	Helps to limit the muscle acidity
Vitamins B1 + B2	Vitamin intake for energy metabolism
Vitamins C + E	Antioxidant properties
Caffeine	Favors the use fats during exercise to delay the depletion of glycogen stores

Table 8: Nutrients and function in human body (Source : Nicolas Aubineau [38])

In our simulations, the models take into account carbohydrates, proteins, lipids, calcium and sodium. So we're going to focus on how to consume these nutrients, before, during and after a run.

Before the effort

On the eve of a race, it is advisable to have specific intakes of carbohydrates (fruit and vegetables cooked to avoid digestive problems).

It is advisable to finish breakfast 2 hours before departure but the most important point is to take the time to eat it, to chew (minimum 15 minutes). Here is the average nutritional intake of a cyclist's breakfast: 120-200g carbohydrates, 20-30g proteins, 20-40g lipids, 475 -520mg calcium, 880-1600mg sodium [38].

The drink between the last meal and the race is an integral part of the approach nutritional of the athlete upstream of an effort. It integrates the prevention of digestive disorders, functional disorders, possible glycemic disorders and promotes proper hydration. It will be consumed in small sips during the waiting time of 3 to 4 hours without exceeding 500 ml per hour, volume to be determined according to environmental conditions (temperature, humidity...) [38].

During the effort

First of all, we need to know that every cyclist is unique, so there is no standard dietary protocol for everyone. However, we can describe the average habits to be adopted for a race to take place at Better.

Runners have a continuous intake of carbohydrates during exercise, thanks to the powder put in the cans. The exercise drink should contain about 50 to 70g carbohydrates per liter (fructose, glucose, maltodextrins) in order to cover most of the energy needs. The recommended dose of carbohydrates is 60 to 90g per hour of exercise. The exercise drink also contains sodium, see other minerals (magnesium, potassium), vitamins (especially B1, B2, B3, B6, C, E) [36].

It is important that the energy drink is properly dosed with sodium during exercise (minimum 300 mg per hour). There is no specific value for sodium intake per hour of effort, but an intake of approximately 400-500 mg sodium (1.2-1.5 g sodium chloride = "salt") per hour or more is considered to be adequate for long-term and/or warm-environment tests [35].

The amount of drink ingested depends on the duration and intensity of the test, environmental conditions (temperature, wind, humidity, etc.). It varies between 0.5 L and 1 L per hour (avoid drinking less than 500 mL per hour). This amount is then taken in small sips but regularly. It is advisable to take 2 to 3 sips every 7 to 10 minutes [37]. The idea is to do everything possible to avoid cravings (hypoglycemia), without overloading the stomach.

On the same principle as energy drinks, energy gel is another way to provide carbohydrates, sodium and vitamin B1 [36]. Simultaneous water intake will allow easier and faster absorption. The advantage is that it can be easily stored in a pocket.

For long exertions (more than 4-5 hours), the cyclist needs protein intake not in order to replenish the reserves (which is useless during the exertion, but unavoidable at the end of the race) but in order to delay the onset of nervous fatigue. The energy bar contains proteins and provides a complement in nutritional support, especially carbohydrate. It allows you to bring variety in the texture of the catch (cracky, soft, elastic...), to "chew" a little, to vary tastes and to boost the consumption of "liquid" (effort drinks) [36].

After the effort

Recovery is a key moment to bounce back on the next workout or step. Carbohydrate intake after exercise is particularly important. It is directly involved in the resynthesis of glycogen stocks. In practice, a supply of glucose and fructose is recommended.

The consumption of proteins after an effort, especially long-term and/or intense (marathon type, ultra, strength/resistance training), is essential to limit catabolism (destruction) and promote the anabolism (construction) of proteins within the different tissues (muscles, bones, viscera...). An average protein intake of 10 to 30g appears to be optimal, depending on the intensity and duration of the exercise. The grip must be preferred immediately at the end of the race. The faster the catch, the more the overcompensation phenomena are favoured [39].

To combat acidosis, it is recommended to consume alkalizing drinks (e.g. a drink rich in bicarbonates) or natural citrate-based dietary supplements, which are better tolerated from the digestive point of view. In the hours following the end of an effort, eat solid foods that promote the overall reconstruction of the body (dried fruits, protein-seed fruits, fresh, ripe or cooked fruits, but also cereals) [39].

Finally, complete recovery drinks composed of high-quality proteins, carbohydrates and fats (particularly omega 3), minerals and relevant vitamins (vitamins B, potassium, sodium...) help to optimize recovery processes. The liquid texture, the nutritional mix of the intake as well as the proximity to the event (maximum within 30 minutes after the end of the exercise) allows to optimize the processes of regeneration of muscle and joint tissues [39].

6.2 Nutritional aspect of Pulse models

In this section, we look at the nutritional aspects of Pulse. We focus on the energy, tissue, endocrine and gastrointestinal systems.
Indeed, the energy system is connected to all other systems, but the nutritional aspect depends on the renal and gastrointestinal systems, as these are the methods by which nutrients enter the body and metabolic waste are removed from the body. As for the tissue system, it is mechanically linked to the cardiovascular and respiratory systems, but it interacts with energy and drug systems. How do they interact ? During simulations, the Post-process step updates the data at every step of time, allowing all other systems to access the information.

6.2.1 Energy system

One of the functions of the energy system is to simulate metabolism, i.e. the set of coupled reactions occurring in the cells of the body [60]. This metabolism is simulated by the consumption of nutrients in the tissues because they are the "fuel" of the human body. The production and metabolic consumption of the energy system are represented by various calculations that determine the rate of change of nutrients, ions and gases in the tissue. These calculations are obtained using the principles of advection and diffusion of transport methods. Advection is the movement of a material dissolved or suspended in a fluid [61], while diffusion is an irreversible transport phenomenon that tends to homogenize the composition of the environment [62].

Oxygen is used to metabolize these nutrients during aerobic metabolism. It is simply a metabolism that requires oxygen (extract from air or water) to function, and it consists of the breakdown of carbohydrates and fats and secondarily of proteins [63].
Aerobic metabolism is activated during endurance sports. After entering the pulmonary capillaries, oxygen is transported through the circulatory system and diffuses into the tissues. The resulting by-product, carbon dioxide, diffuses out of the tissues, travels through the circulation before leaving the body via the lung capillaries [11].
On the contrary, anaerobic metabolism is defined as those metabolic processes that do not require oxygen

to function. This mode of operation is active during intense but relatively short efforts. In these situations, glucose can also be used to produce energy, and lactate (lactic acid) is the resulting by-product [11].

During periods of prolonged starvation, the liver produces ketoacids. Acetoacetate is the predominantly synthesized ketone. Once produced, acetoacetate can be used as an energy source for the brain, myocardium and muscles. Note that the "famine" event is not yet activated in the Pulse models [11].

In the part on the gastrointestinal system we will see how macronutrients are stored and processed. It is assumed that all carbohydrates are stored in the tissues as glucose. Glycogen stores are not taken into account since the global representation of carbohydrates is glucose.

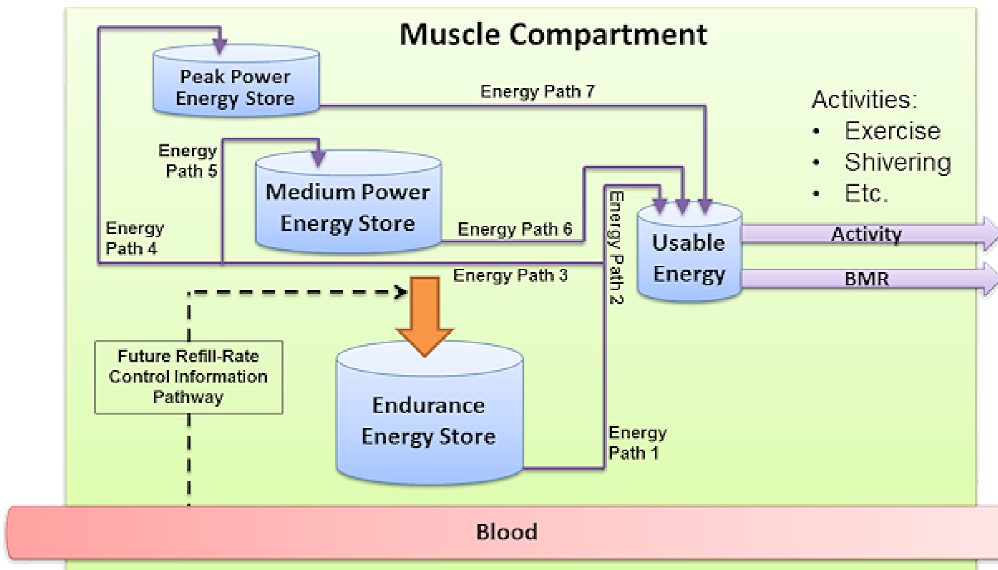

Figure 51: Energy storage (Source : Pulse [11])

The graph **51** shows that the filling rate of the endurance energy store from the substances circulating in the blood is not defined at the moment. At present, the energy flow into this reservoir is independent of the substances carried by the blood. This will be confirmed when studying the output data associated with the different simulations, in section **6.3**. This level will be coupled with blood nutrient levels in a future release [11].

Similarly, high- and medium-intensity energy reserves are not replenished according to the nutrients consumed. In practice, this means that increasing nutrient consumption will not increase the physical capacity of the patient. We will see this later in the simulations. However, we already know that the most far-reaching results will be those for endurance, because in practice it is necessary to hydrate and eat in this case. In fact, these drinks allow the cyclist to recharge his energy, while in Pulse only breaks are a way to replenish the reserves.

Energy flows are bounded at the upper boundary by a proportion of the maximum work rate and limited at the lower boundary by zero. The various events associated with the energy system that can occur during a simulation are hypothermia, hyperthermia, dehydration, fatigue and fasciculation. However, dehydration is not yet active in the models [11].

6.2.2 Tissue system

Metabolism is simulated by the production and consumption of substances. The basis for metabolic production and consumption calculations is the non-protein respiratory quotient (RQ). This value is used to determine the fraction of metabolic energy produced by the consumption of carbohydrates (glucose) [57].

This allows us to link to the tissue system, which has the function of controlling the transport of substances (nutrients, gases, ions) between tissues and blood, it is inseparable from the energy system. There are several types of transport depending on the nature of the substances [57] :

- Instant diffusion allows gases to be transported through biological membranes.
- Simple diffusion is based on a linear relationship between the flux of the substance and the concentration gradient of the substance.
- In facilitated diffusion, the flow of substance is asymptotic. This dissemination uses the kinetics of an enzyme-catalyzed reaction acting on a single substrate to produce a product.
  Catalyzed reactions are used to increase the speed of a specific chemical process. The role of the catalyst is to allow a different course of the reaction, limiting the energy required. To do this, the catalytic agent interacts with the reagent to form an intermediate compound [64].

The three diffusions described above are gradient modes of transport, while active transport is a model that requires energy (for example sodium-potassium pump). The active transport feature is currently inactive. However it will be based on feedback on consumption rates of energy-containing substances, to move substances across barriers [57].

6.2.3 Endocrine system

Then, it is necessary to release hormones to maintain homeostasis and regulate body functions, including metabolism. This is the main function of the endocrine system [58]. In the biological field, homeostasis is defined as the natural regulation of the body to keep the biological parameters of the human body constant in the face of changes in the external environment (examples: human body temperature, blood glucose, blood composition) [65].

In the endocrine system, different endocrine glands respond to stimuli by synthesizing, releasing or inhibiting the release of hormones. Generally, the release of hormones is described by a negative feedback mechanism, which means that the effects of the hormone on physiology result in a halt in the release of hormones [58].

Pulse's endocrine system currently contains only two hormones: epinephrine and insulin [58]. The rate of insulin release is altered by the concentration of glucose in the blood, and the rate of epinephrine release is disturbed by two actions: exercise and acute stress [58]. In our case, we are only interested in exercise actions.

Pulse does not currently model the physiological effects of epinephrine deficiency, so a reduced blood level of epinephrine will have no effect [58]. In addition, the response to insulin currently depends only on blood glucose concentration, whereas it is expected to depend on both blood protein and blood glucose concentration [58].

6.2.4 Gastrointestinal system

Finally, the gastrointestinal system is connected to extravascular tissue and cardiovascular system. It models the ingestion of water and macronutrients, and their subsequent digestion and transport into the cardiovascular system [59].

Currently, the gastrointestinal model reproduces the behaviour of the stomach and small intestine. After eating, food is stored in the stomach, progressively digested, and then released into the intestinal chyma. At this stage, nutrients are either rapidly absorbed into the bloodstream through the intestinal wall or broken down before absorption [59]. The rate of digestion of macronutrients is highly dependent on the nature of the source food.

Table **9** describes how macronutrients are converted into substances in the system.

Macronutrient	**Resultant Substance**	**Technique**
Fat	Tristearin	emulsified by the small intestine and then quickly broken down by pancreatic enzymes to absorb them
Protein	Urea	currently decomposed into urea to simulate amino acid deamination in the liver
Calcium	Calcium	regulated by parathyroid hormone
Sodium	Sodium	quickly absorbed thanks to active transport (not yet active in the model)
Water	Water	absorbed through the osmotic gradient between intestinal chyma and blood OR absorbed through active transport via the intestinal wall

Table 9: How macronutrients are converted into substances (Source : Pulse [59])

The stomach is initialized with configurable amounts of each macronutrient, sodium, calcium and water. Stomach contents can be changed in two ways [59] :

- The *ConsumeMeal* condition is used to specify a meal ingested before a simulation. This condition is currently inactive because it is being developed by research teams.
- The *ConsumeNutrients* action allows the patient to consume nutrients, with a configurable digestion rate for macronutrients (proteins, fats, carbohydrates).

In Pulse, a digested amount is calculated individually for each substance, and this calculated mass is decremented from the stomach, multiplied by its associated absorption factor, and the result is added to the gastrointestinal circuit [59]. Table **10** describes the digestion rates associated with nutrients.

Macronutrient	**Digestion Rate**
Carbohydrate	0.5 - 0.625 g/min
Fat	0.055 g/min
Protein	0.071 - 0.157 g/min
Calcium	2.7 mg/min
Sodium	N/A
Water	0.417 mL/s

Table 10: Configuration of macronutrients in Pulse models (Source : Pulse [59])

6.3 Study of output data through different simulations

6.3.1 Maximum capacity without nutrients.

Let us observe the maximum capacity of a high level cyclist according to the model currently available. We simulated a single exercise with a maximum intensity of 0.5 during 60 minutes, in order to observe his

state of health. During this simulation, the cyclist did not eat or drink before or during the race.
Listing **16** contains an extract of the terminal display, during the execution.

```
...
[INFO] : [1926.08(s)] [Event] 1926.08(s),  Patient has Bradypnea
...
[INFO] : [2522.82(s)] [Event] 2522.82(s),  Oxygen tension in the brain is dangerously low
...
[INFO] : [2851.4(s)] [Event] 2851.4(s),  Patient has low blood pressure and the
    vasculature has collapsed
...
[INFO] : [3527.22(s)] [Event] 3527.22(s),  The patient‘s heart is not receiving enough
    oxygen
...
[INFO] : [3600.12(s)] It took 420.958s to run this simulation
```

Listing 16: Display on the terminal

We observe that from 32 minutes of exercise, the cyclist "patient" has an abnormal slowing of breathing (Bradypnea). At the 41th minute his brain is dangerously lacking oxygen. He then has the vasculature collapsed, and finally the simulation ends with an error due to lack of oxygen. The results of this scenario suggest that an effort of 0.5 intensity must last less than 32 minutes for the cyclist to not be "unhealthy".

Let us analyse simulation graphically, using different output data of Pulse model.
Figure **52** describes heart rate according to time.

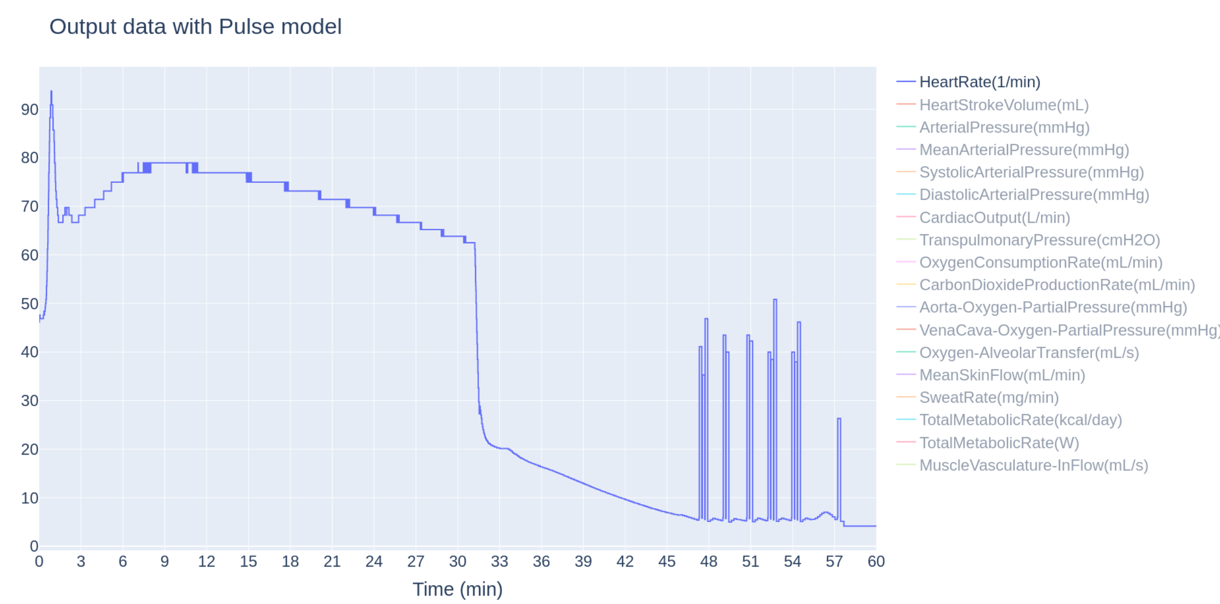

Figure 52: Output data with Pulse model

We notice a sharp decrease in heart rate between the minute 31 and the minute 32. Indeed, during this period the cyclist goes from 64 beats per minute to about 28 beats per minute. This phenomenon corresponds to bradypnea, as shown in the terminal during the simulation (see Listing **16**).
However, it should be noted that the French Academy of Medicine defines bradypnea as a breathing rate below 16/min for adults [26], i.e. from 41 minutes in our case. This can be caused by cold weather or by a high concentration of carbon dioxide (lack of oxygen). If we look at the terminal extract displayed in Listing **16**, it is from the minute 42 that the amount of oxygen in the brain is dangerously low. This is

completely consistent with the definition of bradypnea.

Another interesting part of the chart is from the minute 47. Figure **52** first shows several heart rate peaks and then becomes constant and very low (less than 5 beats per minute). If we refer to Listing **16**, abrupt increases occur when the blood pressure of the cyclist is weak and the vasculature is collapsing. Then, the constant low heart rate corresponds to the lack of oxygen in the cyclist's heart.

Let us take a closer look at what happens during the two periods described above.

Zoom on the time interval [29,33] (min)

The time period of interest here is between 29 and 33 minutes because the model indicates that the patient has sharp decrease in heart rate during this time period.

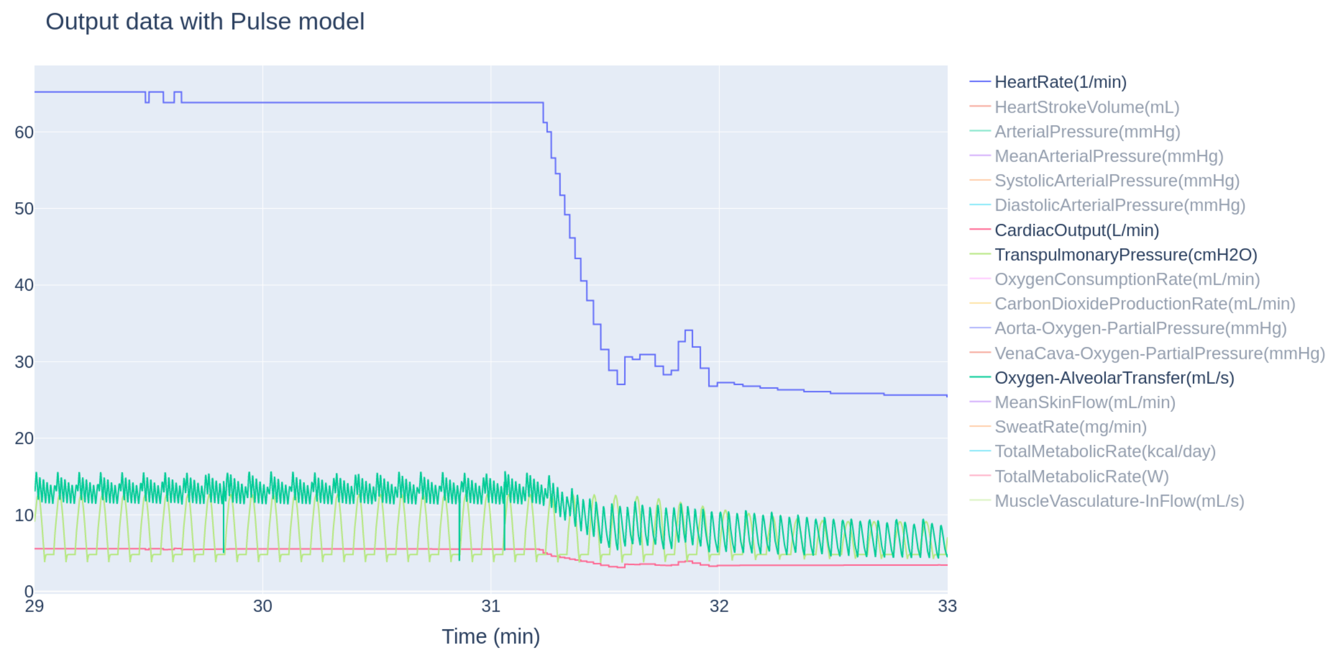

Figure 53: Output data with Pulse model

Figure **53** shows three new data:

- **CardiacOutput** corresponds to the volume of blood (in liters) supplied by the heart every minute. It is the amount of work performed by the heart in response to oxygen needs of the body [27].
- **Oxygen-AlveolarTransfer** corresponds to the amount of oxygen (in milliliters) diffused from the alveoli into the pulmonary capillaries every second. Alveoli are tiny air sacs in our lungs that take up the oxygen we breathe in. We have about 480 million alveoli, located at the end of bronchial tubes. When we breathe in, the alveoli expand to take in oxygen and when we breathe out, the alveoli shrink to expel carbon dioxide [29].
- **TranspulmonaryPressure** is the difference between alveolar pressure (inside the pulmonary alveoli) and intrapleural pressure (inside the pleural cavity). It should be noted that the intrapleural

pressure is slightly lower than the atmospheric pressure [28]. The unit of transpulmonary pressure is the centimetre of water (cmH2O), which is equal to 0.7355 mmHg.

On Figure **53** we observed that the cyclist's heart rate decreased sharply when the values associated with the three data decreased. This is normal because a fall in the volume of blood supplied by the heart and the volume of oxygen delivered to the lungs leads to a decrease in heart rate.

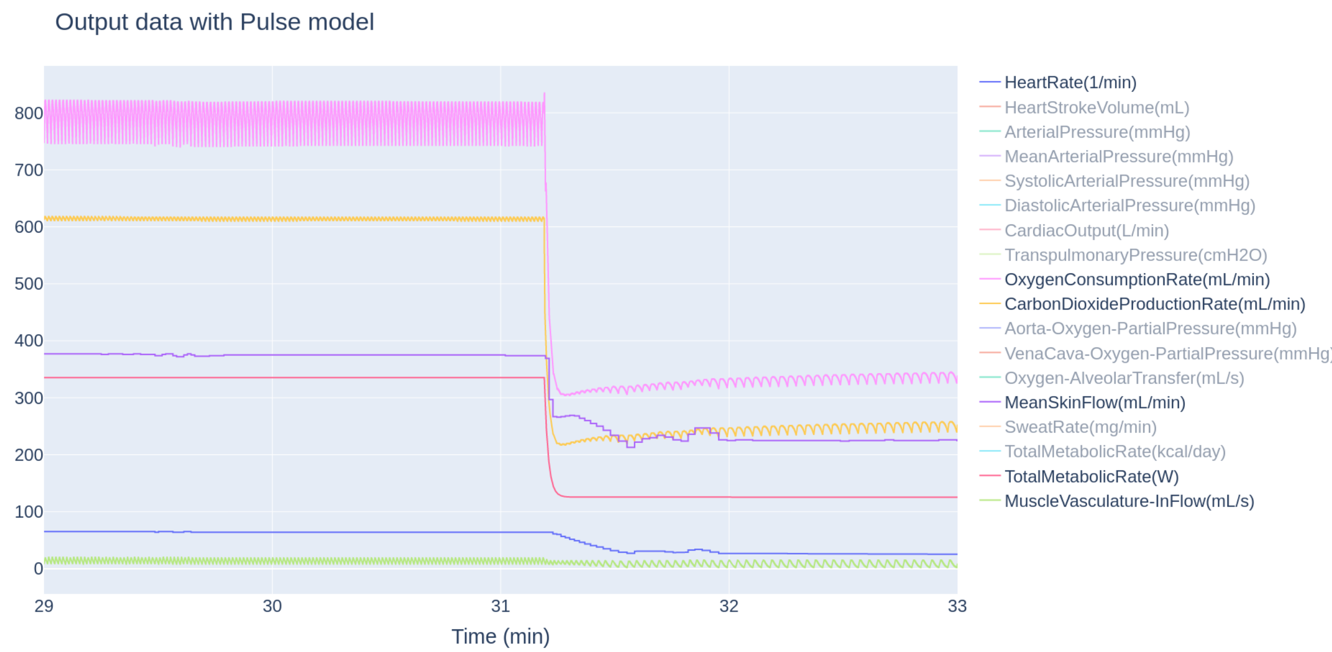

Figure 54: Output data with Pulse model

Let's describe the five new data appearing in Figure **54**:

- **OxygenConsumptionRate** measures the volume of oxygen (in milliliters) the body takes in and uses over a minute of time.
- **CarbonDioxideProductionRate** measures the volume of carbon dioxide (in milliliters) produced each minute in the body as a result of cellular respiration.
- **MeanSkinFlow** measures the average flow of blood (in milliliters) under the skin. Skin blood flow is important in maintaining nutrition, regional and whole body temperature, and healing traumatized skin. There are two different types of skin. The apical (glabrous) skin is present on the palmar surface of the hand, the plantar surface of the foot, and the face. The nonapical (hairy) skin is present over most of the body surface [30].
- **TotalMetabolicRate** is the total amount of energy used by the body per day. It determines our ability to lose weight, gain weight or remain the same based on our food intake and physical activity throughout the day [31]. The unit of measurement used is watts or kcal/day.
- **MuscleVasculature-InFlow** it measures the volume of blood (in milliliters) circulating every second in the blood vessels of the muscles.

Carbon dioxide production varies with oxygen consumption so we have to see a decrease in the **OxygenConsumptionRate** and **CarbonDioxideProductionRate** data as the heart rate drops. We should also notice a decrease in blood volumes for **MeanSkinFlow** and **MuscleVasculature -InFlow**. If you look at Figure **54**, you can see, as expected, a decrease in the values at the same time as the heart rate. Note that the **TotalMetabolicRate** of the cyclist also decreases, which is normal because the cyclist becomes weaker and consumes less energy.

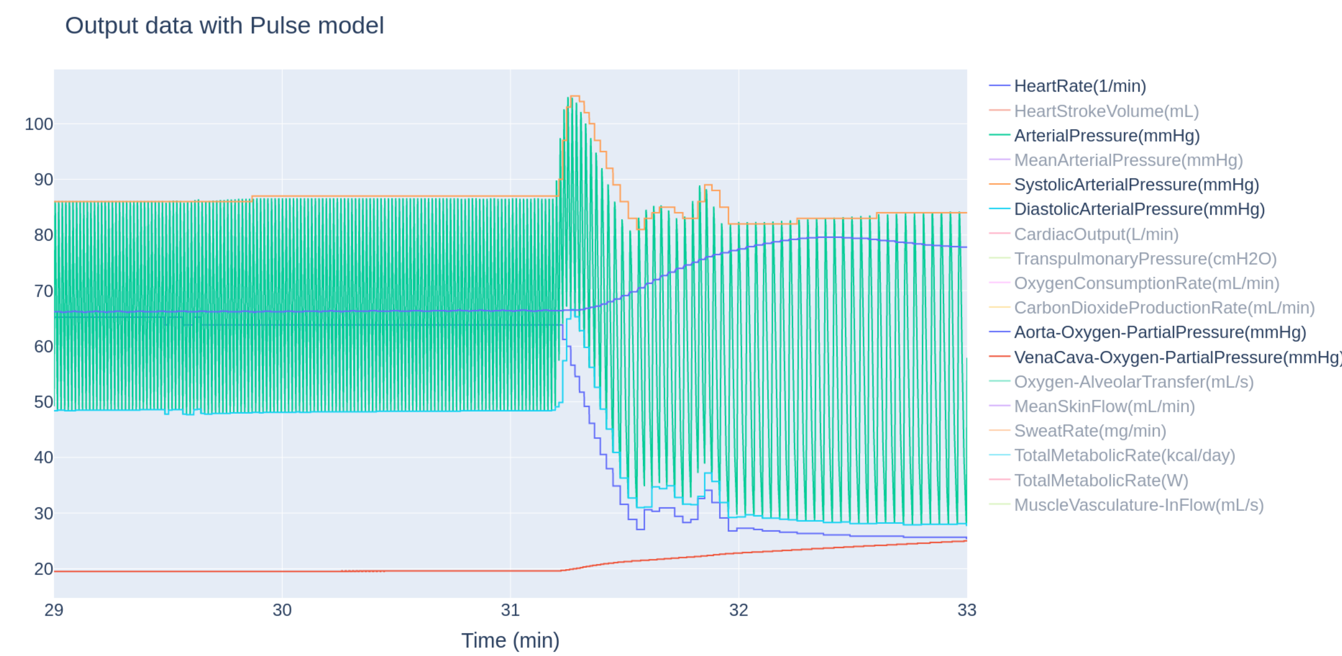

Figure 55: Output data with Pulse model

In Figure **55** we see two new data.
Let's note first that in a mixture of gases, each constituent gas has a partial pressure which is the notional pressure of that constituent gas if it alone occupied the entire volume of the original mixture at the same temperature [32].

- **Aorta-Oxygen-PartialPressure** : The aorta is the largest vessel in the human body and serves to transport oxygenated blood in the lungs from the body to the rest of the body. This function is vital, providing each cell with oxygen and energy, and then carrying away metabolic degradation products and carbon dioxide [33]. Finally, **Aorta-Oxygen-PartialPressure** measures the partial pressure (in mmHg) exerted by oxygen molecules in the aorta.
- **VenaCava-Oxygen-PartialPressure** : In air-breathing vertebrates (such as humans), Vena Cava is the anterior and posterior cava veins, which release oxygen-depleted blood to the right side of the heart. The anterior vena cava, also called the precave, drain the head of the body, while the posterior vena cava, or postcave, drains the tail, or hind extremity [34]. Finally, **VenaCava-Oxygen- Partial-Pressure** measures the partial pressure (in mmHg) exerted by oxygen molecules in the anterior and posterior vena cava.

As can be seen in Figure **55**, the decrease in the data described above is caused at the time of a peak of blood pressure increase. This explains the sharp drop in heart rate during this period. Then the pressure decreases and seems to stabilize up to the minute 33, as with the other data.
We note that the partial pressure of oxygen in the aorta and vena cava also increases, but less abruptly.

Finally, all these observations point to the reasons for the cyclist's decreased heart rate.

Zoom on the time interval [47,60] (min)

Let us observe what happens in the cyclist's body between 47 and 60 minutes.

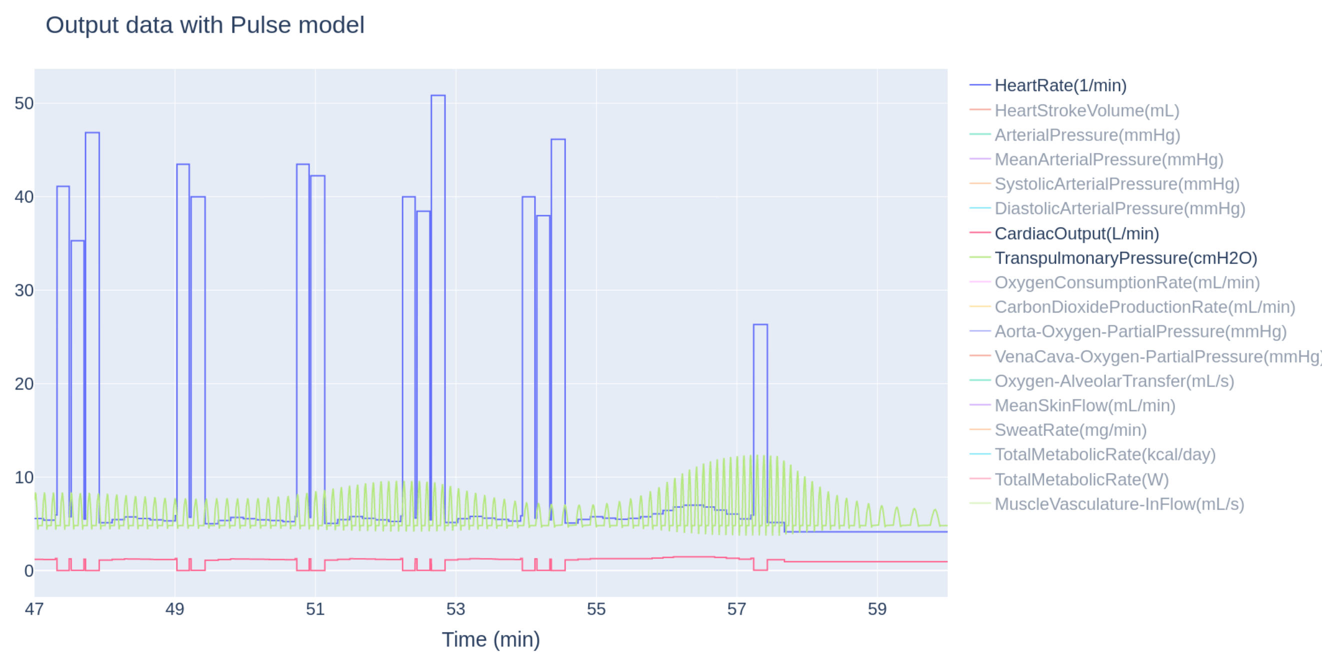

Figure 56: Output data with Pulse model

Figure **56** shows that during peak heart rate increases, the **CardiacOutput** data becomes zero, i.e. the heart stops providing blood. We also see that transpulmonary pressure tends to increase during peaks.
If we look at the terminal extract described in Listing **16**, when the heart rate increases sharply there is a low blood pressure. That's consistent, and it's explained by the zero blood flow at the same time.

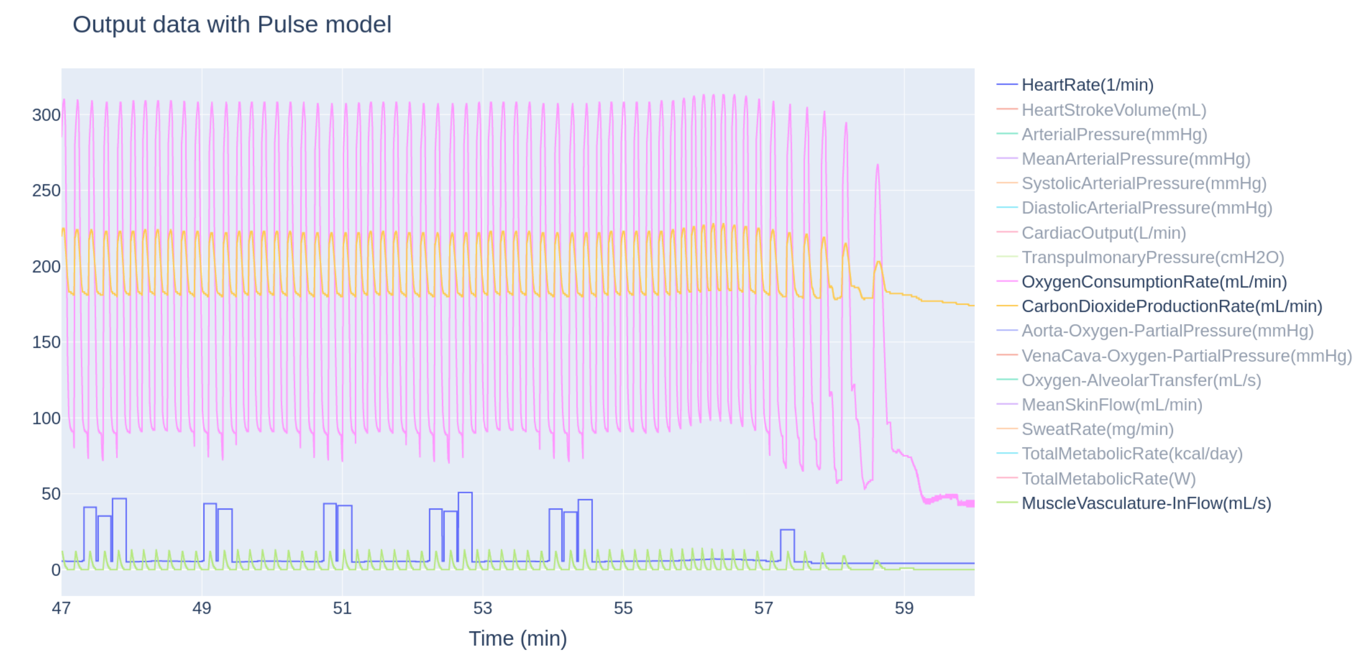

Figure 57: Output data with Pulse model

On Figure **57** we can see the drop in oxygen consumption during peak heart rate increases. This is related to the zero blood flow which leads to a sudden lack of oxygen. In addition, we notice a sharp decrease in the **OxygenConsumptionRate** data starting from the minute 58. The next minute the consumption remains very low, which is explained by a decrease in blood flow. This information is observable by a zero blood circulation in the muscular vascularization. We know that the production of carbon dioxide is directly linked to the consumption of oxygen, so it is normal to observe the same phenomenon.
Finally, these observations were described in Listing **16** by the fact that the heart lacks oxygen from the minute 58.

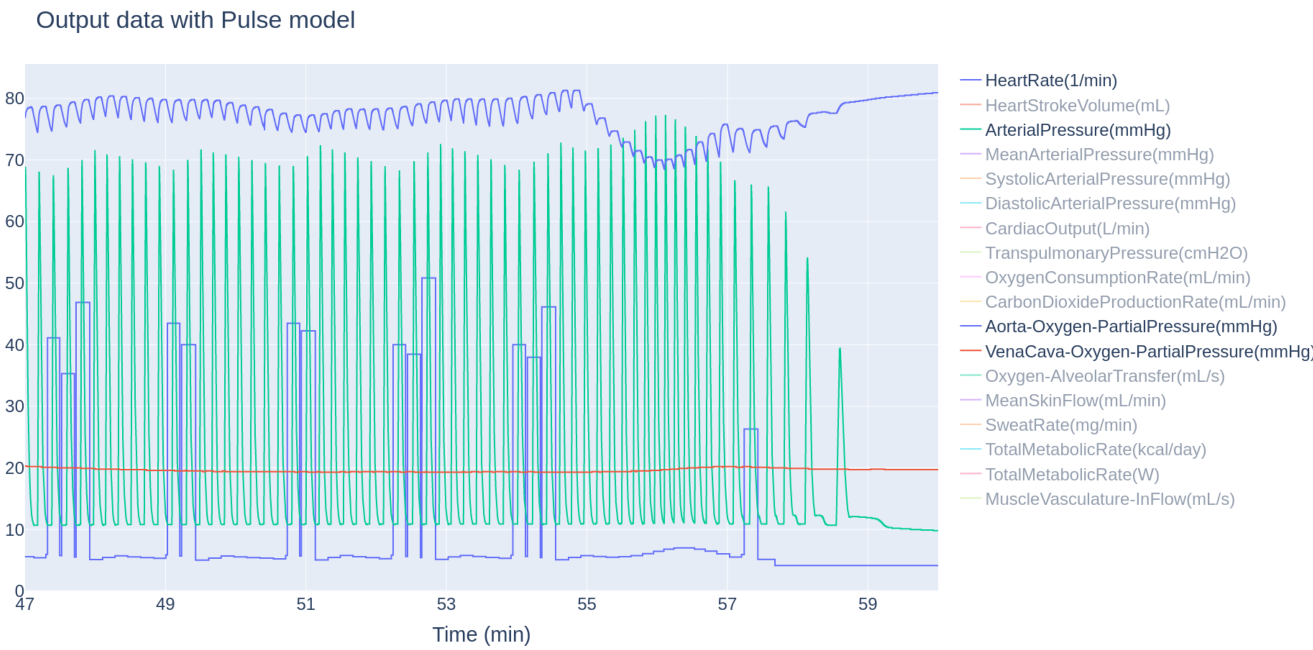

Figure 58: Output data with Pulse model

The phenomenon observed above is also described in Figure **58**. Indeed, we notice a sharp decrease in blood pressure from 57 minutes until it is zero. On the other hand, the partial pressure of oxygen in the aorta increases and the pressure in the vena cava changes very little.

Finally, all the data observed from 47 minutes show a strong deterioration in the cyclist's health. We find that from the minute about 57 the cyclist is dying, then the end of the simulation shows almost zero vital data.

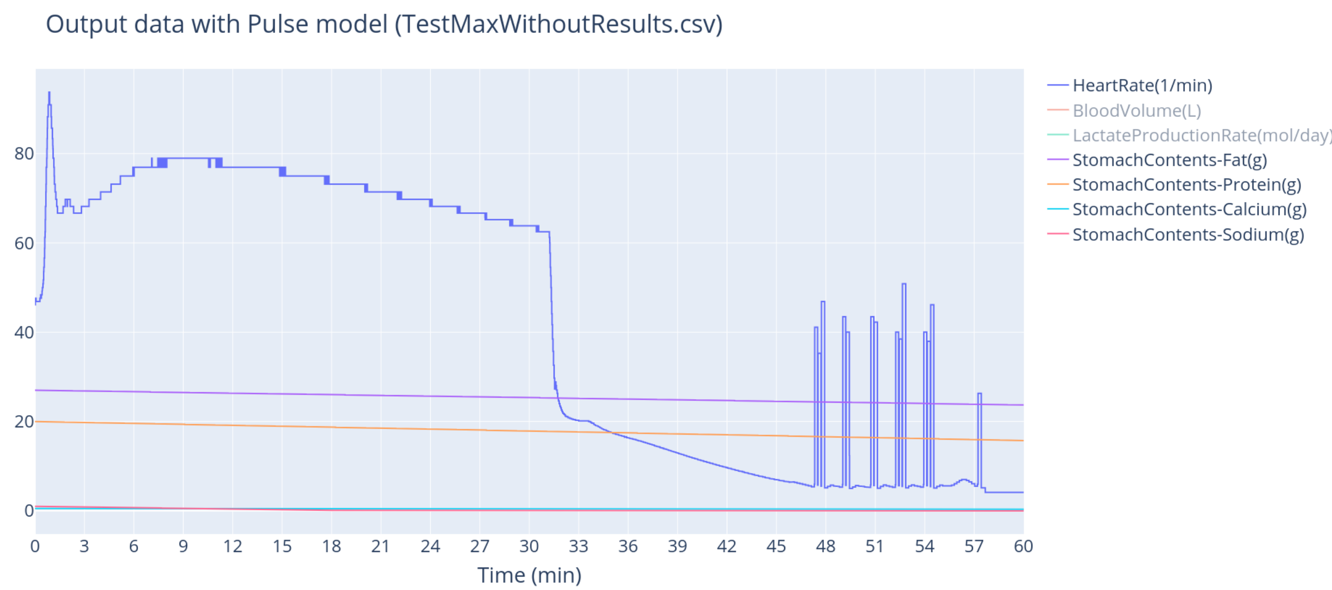

Figure 59: Output data with Pulse model

In Figure **59** we see four new data :

- **StomachContents-Fat** is the amount of fat in the patient's stomach (in grams). Fat transport fat-soluble vitamins for use by the body and build cell membranes [42].
- **StomachContents-Protein** is the amount of protein in the patient's stomach (in grams). They have many roles, but they are primarily an energy fuel for the body in the event of prolonged fasting or when there is a shortage of glucose available to the body. Proteins are also needed for muscle contraction [43].
- **StomachContents-Calcium** is the amount of calcium in the patient's stomach (in grams). Calcium intervenes in particular within the heart, nerve and muscle cells, and is mainly localized in the skeleton, teeth and blood [44].
- **StomachContents-Sodium** is the amount of sodium in the patient's stomach (in grams). Sodium plays an important role in water balance, influencing the distribution of water within our body. It is also involved in regulating blood pressure: high sodium levels in the blood lead to hypertension, while too low levels lead to hypotension. In addition, sodium also ensures the transmission of nerve impulses, and the contraction of muscles [45].

On the graph **59** we observe how the contents of the cyclist's stomach change during exercise. If unspecified, the amount of nutrients in the stomach is set by default, so that the simulation starts without error. The amount of nutrients in the stomach is decreasing in a linear trend.
The amount of fat in the stomach is initialized at 27 g and decreases to 23.7 g after 1 hour of simulation.
The cyclist starts the exercise with 20 g of protein in the stomach and finishes the exercise with 15.74 g of protein.
Similarly, the amount of calcium at the beginning of the exercise is 0.5 g and reaches 0.34 at the end of the simulation.
Finally, the initial amount of sodium was 1 g, then decreased to 0 g after 20 minutes of simulation.

Finally, we find that when nutrients are initalized by default, they are not compatible with the lifestyle of a high level cyclist. In fact, we saw in the **6.1** section that cyclists should consume sodium during even a short exercise to compensate for the loss of salt. According to the same section, the amount of fat, protein and calcium in the athlete's stomach before the race is also too low.
For example, in the subsection below it will be considered whether the cyclist has eaten a meal before running.

6.3.2 Maximum capacity with consumption of nutrients

The above observations led me to add the cyclist breakfast below [38]:

- 200g of bread (a baguette)
- 40g sweet butter (four individual trays)
- 60g of honey or jam (two small individual pots)
- 250mL of black coffee (a large bowl)
- 300mL of pure orange juice (two glasses or two oranges)
- 250g of plain yogurt (two units)

consumed by the cyclist 2 hours before exercise. This corresponds to the maximum quantities that the cyclist can consume during the breakfast that precedes the effort.
The **ConsumeMeal** option is used to define the meal eaten before the simulation. However, this command is not activated in the software yet, so I included the meal in the script and let it go 2 hours before starting the exercise.

The nutritional benefits of this breakfast are described in Table **11**. Then, to make the simulation realistic, we added a consumption of carbohydrates, water and sodium every 10 minutes. Knowing the consumption per hour of exercise for a cyclist, we divided the data by 6 and obtained about 15 grams of carbohydrate, 0.2 L of water, and 100 mg of sodium. However, we observed that the cyclist feels a lack of sodium from the start of the simulation. Since we did not find any information in the literature, we assumed that sodium actually corresponds to sodium chloride (salt), i.e. that the amount should be about three times higher.

The simulations data I was comparing were not necessarily of the same length because the cyclist's health could lead to premature termination of the simulations. Thus, for the MAE and MSE error calculations, I considered the shortest simulation and compared the data over the shortest interval.

Nutrients	**Quantities**
Energie (kcal)	1300
Glucides (g)	201
Protides (g)	31
Lipides (g)	39
Calcium (mg)	521
Sodium (mg)	1604

Table 11: Nutritional benefits of the cyclist breakfast (Source: Nicolas Aubineau [38])

The JSON file corresponding to scenario **TestMaxIntensity** is stored in the data folder on GitHub.
The simulation was limited to 2 hours 51 minutes, an additional time would have resulted in an error due

to the cyclist's poor health. In this period are included 2 hours corresponding to the intake of the meal as well as the digestion before the race. In the end, the race lasted only 51 minutes. The execution of the full script took approximately 18 minutes.

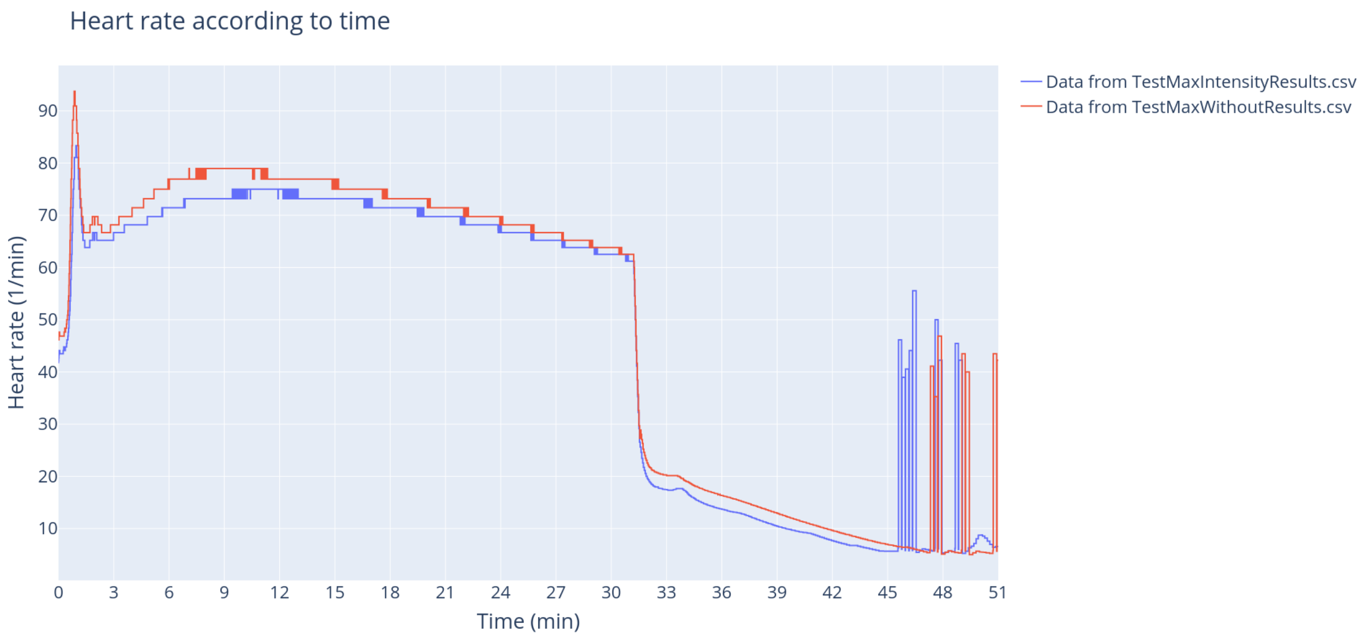

Figure 60: Output data with Pulse model

Figure **60** represents the heart rate curves for a maximum intensity exercise (0.5) with food and energy drink (TestMaxIntensity) and for the same exercise without food and energy drink (TestMaxWithout). The two simulations did not last the same time depending on the cyclist's state of health, so I truncated the end of the results. This allows me to compare the same amount of data and it's not a problem because I cut the frequency peaks at a time when the cyclist's health is critical.

It is observed that from the start of the simulation the heart rates are different. In fact, eating nutrients before the race allowed the cyclist to have a lower peak heart rate than when he was not eating. This is a difference of less than 10 beats per minute.
It is noticeable that throughout the exercise the cyclist has a slightly lower heart rate when he consumes nutrients. However, it is not known if this is due to the earlier meal that shifted the starting heart rate, or if it is due to the consumption of energy drinks every 10 minutes.
The major inconsistency is that the same decreasing peak of heart rate is observed at the 31st minute, while the intensity of exercise is high and constant. As the cyclist's health deteriorates at the same time, in the two different scenarios, this leads to the ingestion of meals and nutrients not being included in the models.

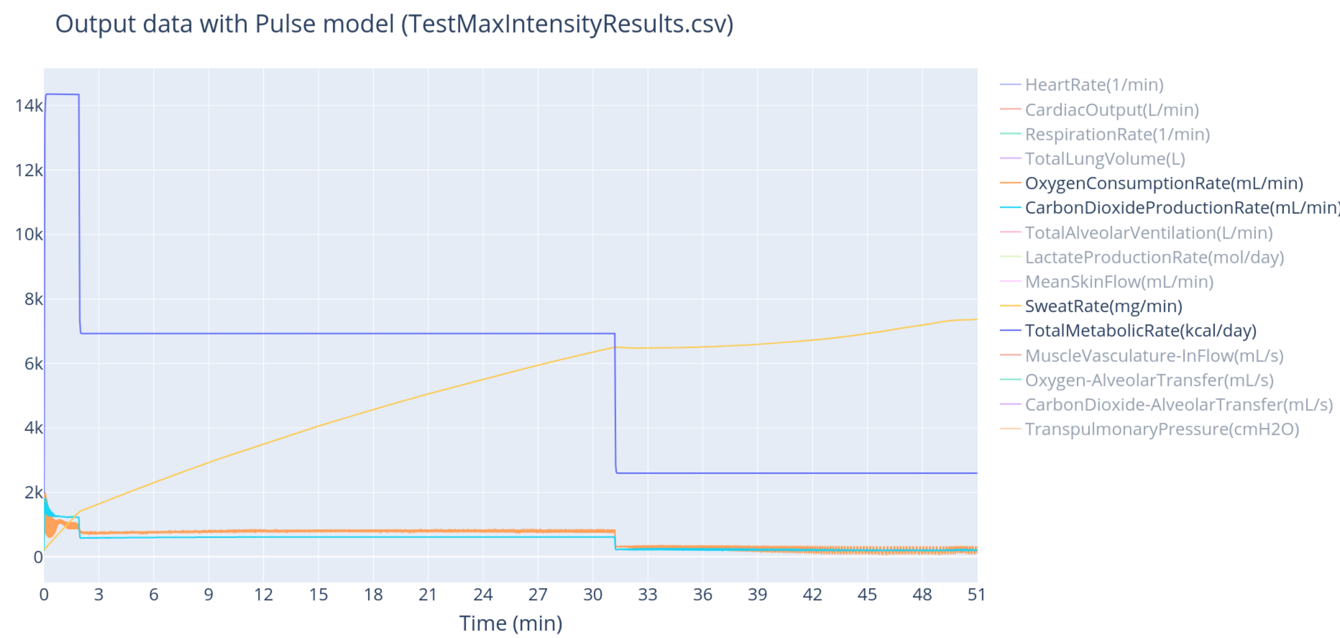

Figure 61: Output data with Pulse model

In Figure **61** we see one new data :

- **SweatRate** : The sweat rate calculates the amount of fluid lost due to sweating during exercise. It is an indicator of fatigue, linked to the cardiac activity, vasodilation and internal temperature. The unit of measurement is the milligram per minute (mg/min). Sweating is a very important thermoregulatory defence of our body to prevent overheating and heat-related diseases. An average person sweats between 0.8 to 1.4 liters per hour during exercise [40].

Figure **61** shows two decreasing peaks for the **TotalMetabolicRate** data, at 2 and 31 minutes. It is noted that at the time of these peaks, the rate of sweating changes in its rate of variation. In fact, this rate increases rapidly between 0 and 2 minutes, then we see a slower increase between 2 and 31 minutes, and from 31 minutes the increase is even slower.

The two observations are consistent with each other because if the body uses less energy, it will produce less sweat. However, this overall decrease may show that the cyclist's health deteriorates over time, as the simulation ends when he has very little energy. Both peaks of **TotalMetabolicRate** occur at times when the rate of oxygen consumption drops sharply. This supports the fact that the cyclist is in poor health.

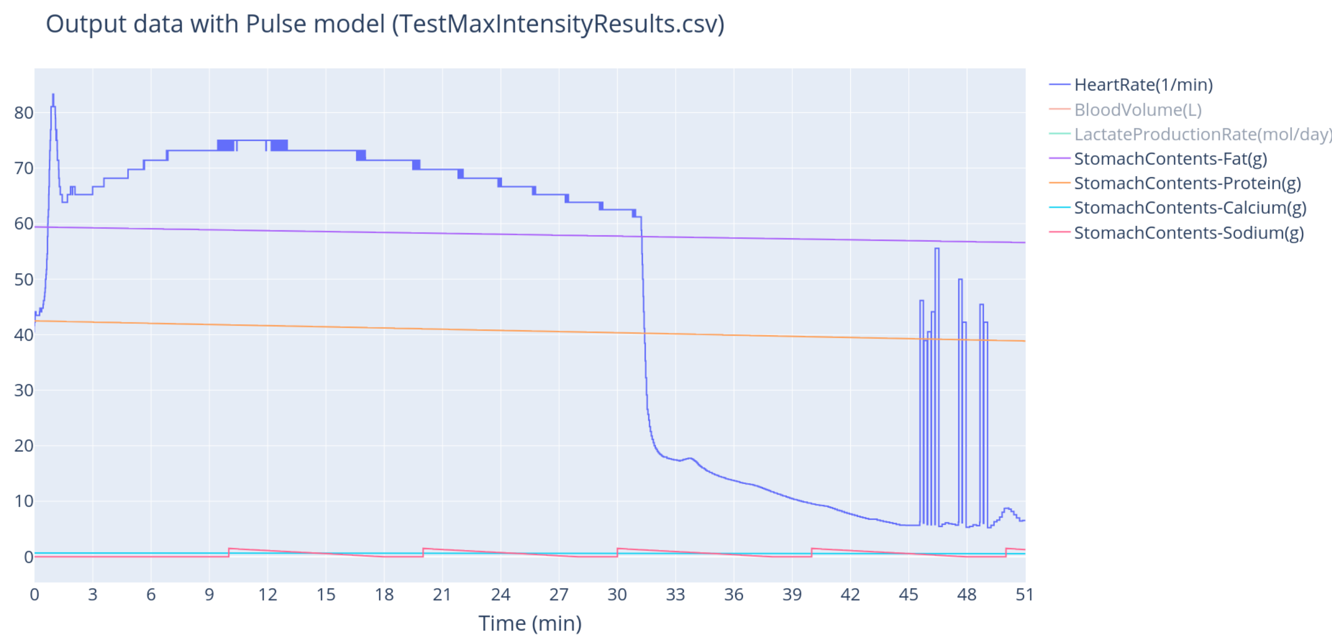

Figure 62: Output data with Pulse model

On the graph **62** we observe how the contents of the cyclist's stomach change during the simulation. The amount of fat in the stomach is initialized at 59.4 g and decreases to 56.59 g after one hour of simulation. This is when we reach an error.
The cyclist starts the exercise with 42.48 g of protein in the stomach and finishes the exercise with 38.86 g of protein.
Similarly, the amount of calcium at the beginning of the exercise is 0.7 g and reaches 0.56 g at the end of the simulation.
Finally, the amount of sodium starts at 0 g. It increases with small peaks of 1.5g every 10 minutes. These peaks correspond to the time when I set a consumption of 15 g of carbohydrate, 1500 mg of sodium and 0.2 L of water. This simulates the moments when the cyclist consumes an energy bar or snackes a small meal during exercise. However, we notice that this high amount of sodium is still consumed entirely before the next 10 minutes interval...

According to Table **12**, the two scenarios simulate very different amounts of fat and protein in the cyclist's stomach.

Output data	MAE	MSE
StomachContents-Fat(g)	32.39999830073133	1049.7598899043835
StomachContents-Protein(g)	22.48000032678244	505.3504146954063
StomachContents-Calcium(g)	0.19695013953609966	0.03881055441907888
StomachContents-Sodium(g)	0.6011679204219414	0.553520431091388

Table 12: Errors between TestMaxIntensityResults.csv and TestMaxWithoutResults.csv

Finally, the results for a maximum intensity exercise appear to be unsatisfactory. However, even a high-level cyclist is rarely at maximum power during a race. Then it would be better to study simulations with a lower intensity of exercise.

6.3.3 Medium capacity : with nutrients vs without nutrients

During a Tour de France, cyclists produce on average between 220 and 320 watts per stage, i.e. for three to five hours [66]. But this is far from being the maximum power, as the example of cyclist Andre Greipel shows: he reached 1900 watts of power in a [66]. It should be noted that this power can only be maintained during a short and intense effort.

Based on the above information, the average power of cyclists can be considered to be 270 watts. Table **3** does not contain the intensity match of this power, but it is easily retrievable. Taking the equidistant value of 430 and 120 watts gives a power of 275 watts. The associated exercise intensity is then 0.24.

The scenario **TestMediumIntensity** is the same as **TestMaxIntensity** but this time the forces are medium (0.24). The feeding of a cyclist described in the table **11** is also used to define the pre-race meal. Similarly, we use the data described in **6.1** to fix the recommended amounts of carbohydrates, sodium and water consumed every 10 minutes during exercise. The amount of sodium is higher than the recommended amount because during the simulations the terminal displayed alerts about too low sodium.
The **TestMediumWithout** scenario corresponds to a run with 0. 24 intensity, but the cyclist did not consume any food or energy drink.

The JSON files corresponding to the scenarios **TestMediumIntensity** and **TestMediumWithout** are stored in the data folder on GitHub.

The simulation of **TestMediumIntensity.json** was limited to 2 hours 56 minutes, an additional time would have resulted in an error due to the cyclist's poor health. In this period are included 2 hours corresponding to the intake of the meal as well as the digestion before the race. In the end, the race lasted only 56 minutes. The execution of the full script took about 20 minutes.
The simulation of **TestMediumWithout.json** was limited to 45 minutes, an additional time would have resulted in an error due to the cyclist's poor health. This scenario does not take into account meals before the race and the execution of the scenario lasted approximately 6 minutes. It was noted that this second simulation took less time before it experienced an error, which could be explained by the lack of food and nutrient consumption.

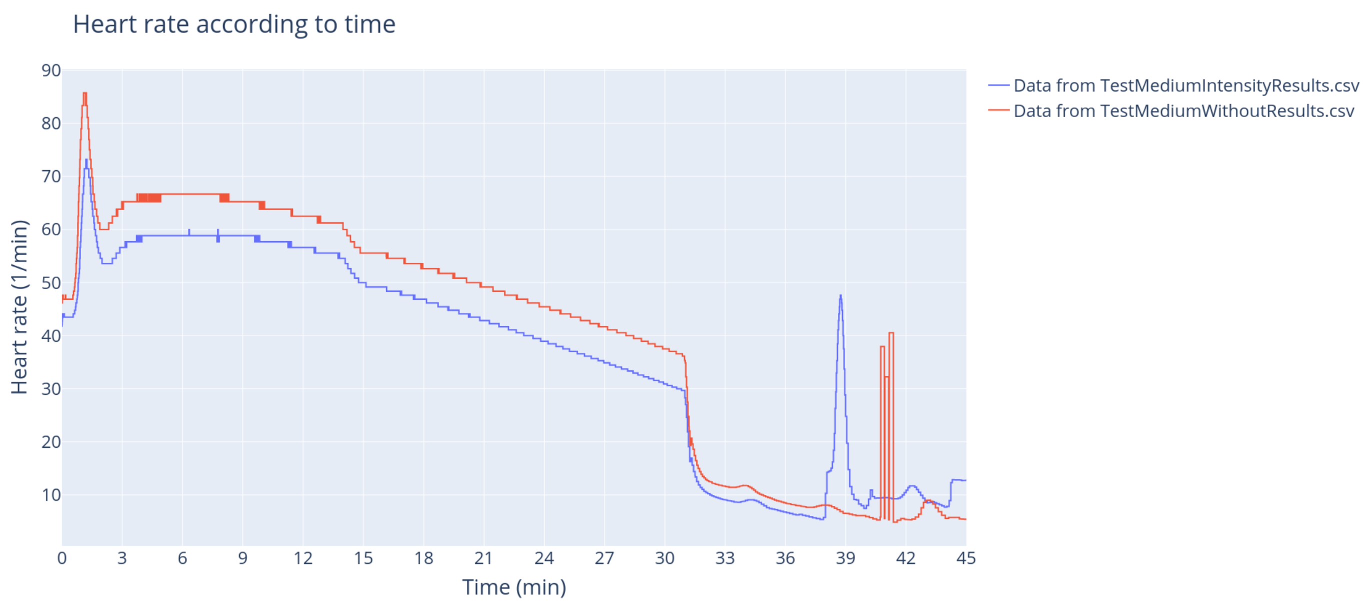

Figure 63: Output data with Pulse model

Figure **63** shows the heart rate curves for a medium intensity exercise with food and energy drink (TestMediumIntensity) and for the same exercise without food and energy drink (TestMediumWithout). The two simulations did not last the same time depending on the cyclist's state of health, so I truncated the end of the results. This allows me to compare the same amount of data and it's not a problem because I cut the frequency peaks at a time when the cyclist's health is critical.

It is observed that from the start of the simulation the heart rates are different. In fact, eating nutrients before the race allowed the cyclist to have a lower peak heart rate than when he was not eating. However, this is a difference of less than 10 beats per minute.

It is noticeable that throughout the exercise the cyclist has a slightly lower heart rate when he consumes nutrients. However, it is difficult to say that it is the digested nutrients that cause this phenomenon.

Let's describe the three new data appearing in Figure **64** :

- **SkinTemperature** : Skin temperature is the temperature of the outermost surface of the body. Normal human skin temperature on the trunk of the body varies between 33.5 and 36.9 °C [46].
- **LeftAlveoli-Oxygen-PartialPressure** corresponds to the partial pressure (defined in **6.3.1**) exerted by the oxygen molecules in the left-hand cells (defined in **6.3.1**). The unit of measurement is mmHg.
- **RightAlveoli-Oxygen-PartialPressure** corresponds to the partial pressure (defined in **6.3.1**) exerted by the oxygen molecules in the right-hand cells (defined in **6.3.1**). The unit of measurement is mmHg.

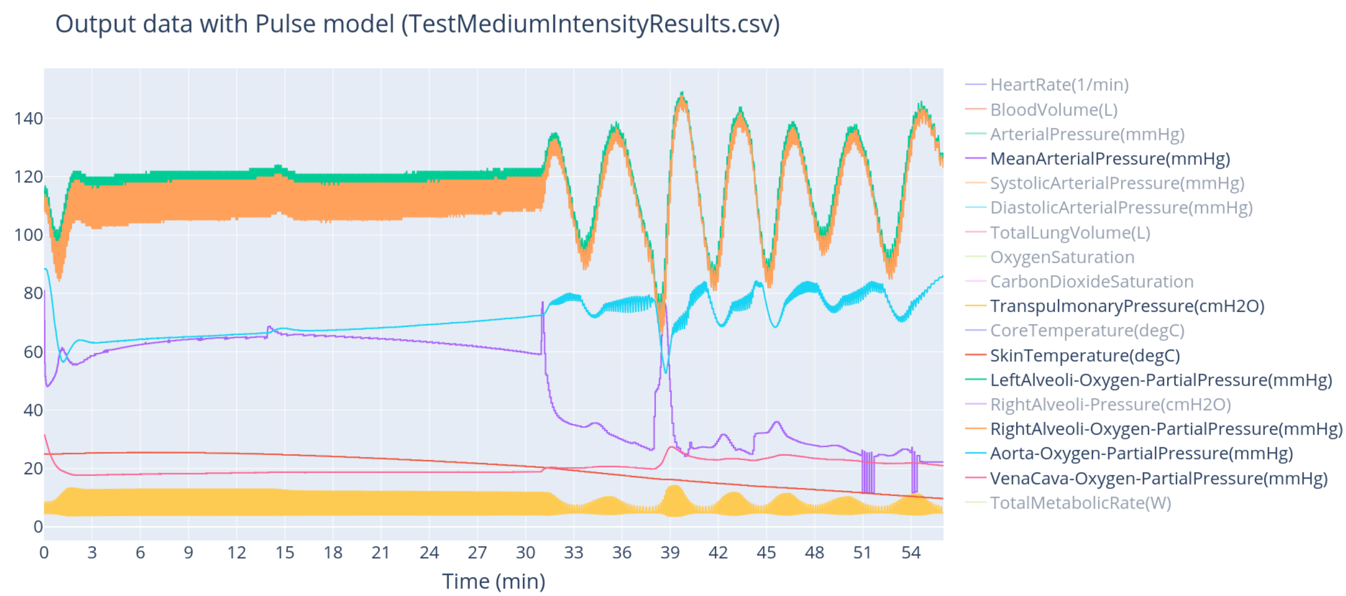

Figure 64: Output data with Pulse model

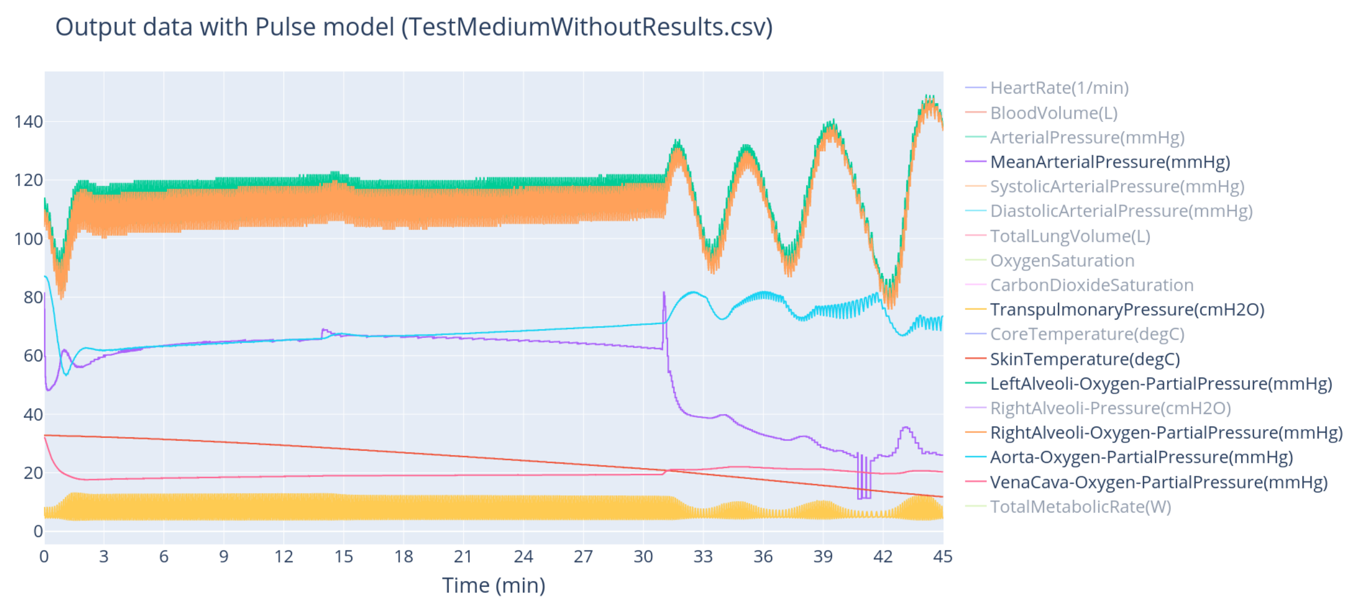

Figure 65: Output data with Pulse model

Looking at the graphs **64** and **65** we see that the first 45 minutes of simulation are very similar. Variations in output data follow the same trend, except for heart rate, which has irregular peaks at over 33 minutes of effort. Frequency peaks imply a change in blood pressure and partial pressures from that point on.

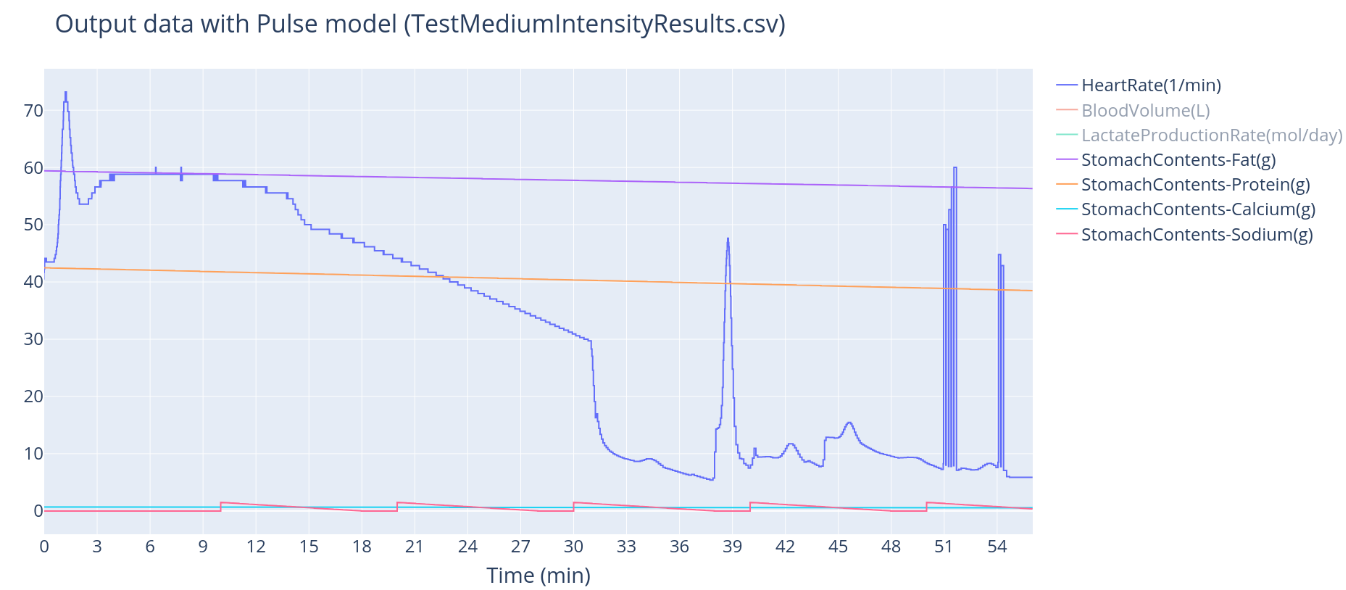

Figure 66: Output data with Pulse model

On the graph **66** we observe how the contents of the cyclist's stomach change during the simulation **TestMediumIntensity**.
The amount of fat in the stomach was initialized at 59.4 g and decreased to 56.32 g after 56 minutes of simulation. This is when we reach an error.
The cyclist starts the exercise with 42.48 g of protein in the stomach and finishes the exercise with 38.5 g of protein.
Similarly, the amount of calcium at the beginning of the exercise is 0.7 g and reaches 0.55 at the end of the simulation.

Finally, the amount of sodium starts at 0 g. It increases with small peaks of 1.5g every 10 minutes. These peaks correspond to the time when I set a consumption of 15 g of carbohydrate, 1500 mg of sodium and 0.2 L of water. This simulates the moments when the cyclist consumes an energy drink during exercise. However, sodium is still digested too quickly...

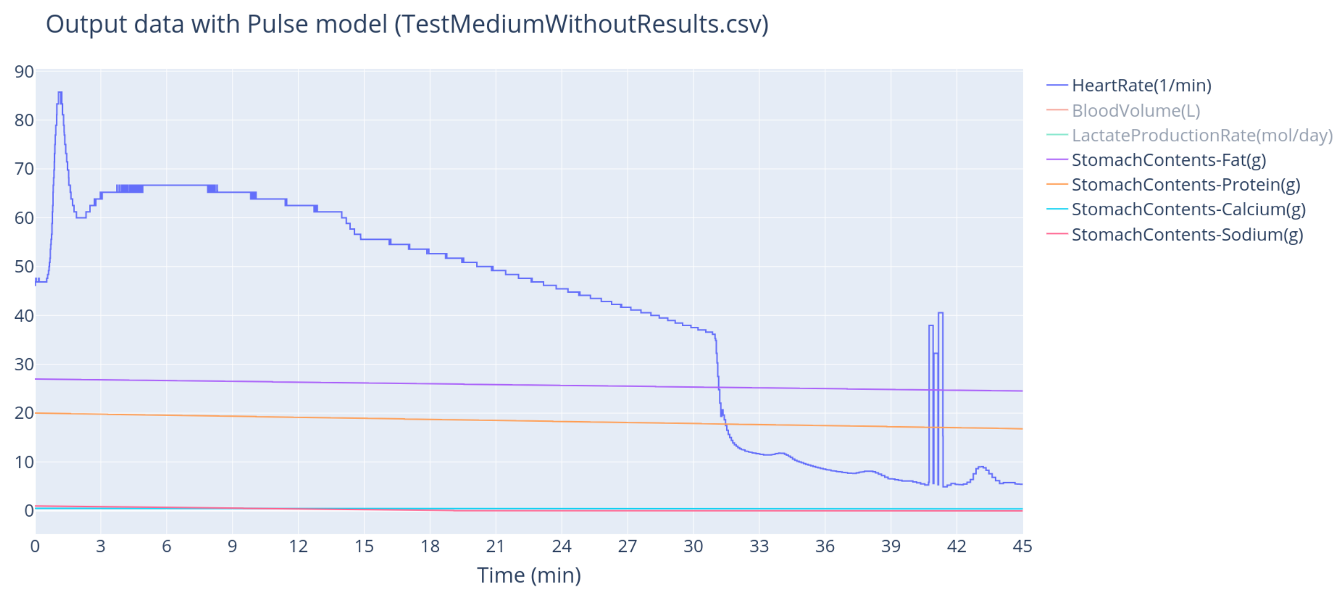

Figure 67: Output data with Pulse model

On the graph **67** we observe how the contents of the cyclist's stomach change during the **TestMediumWithout** simulation. During the simulation, the amount of nutrients in the stomach decreases linearly. The amount of fat in the stomach is initialized at 27 g and decreases to 24.53 g after 45 minutes of simulation.
The cyclist starts the exercise with 20 g of protein in the stomach and finishes the exercise with 16.81 g of protein.
Similarly, the amount of calcium at the beginning of the exercise is 0.5 g and reaches 0.38 at the end of the simulation.
Finally, the initial amount of sodium was 1 g, then decreased to 0 g after 20 minutes of simulation.

According to Table **13**, the two scenarios simulate very different amounts of fat and protein in the cyclist's stomach. Errors are identical to cases with maximum exercise intensity.

Output data	MAE	MSE
StomachContents-Fat(g)	32.39999829635939	1049.7598896211252
StomachContents-Protein(g)	22.480000370356663	505.3504166549389
StomachContents-Calcium(g)	0.19703722084367245	0.03884451612903227
StomachContents-Sodium(g)	0.6313909855190548	0.5763155586830119

Table 13: Errors between TestMediumIntensityResults.csv and TestMediumWithoutResults.csv

Let's describe the five new data appearing in Figure **68** :

- **Epinephrine-BloodConcentration** : Epinephrine is a hormone produced by the adrenal medulla; called also adrenaline (British). Its function is to aid in the regulation of the sympathetic branch of the autonomic nervous system. Epinephrine is a powerful vasopressor that increases blood pressure and increases the heart rate and cardiac output. It also increases glycogenolysis and the release of glucose from the liver, so that a person has a suddenly increased feeling of muscular strength and aggressiveness [41].

- **Albumin-BloodConcentration** : Albumin is an important component of the proper functioning of the human body. It transports essential fatty acids, otherwise known as fat, from adipose tissue to muscle tissue. Albumin also helps regulate osmosis, helping to transport hormones, drugs and other substances through the blood. The range for a normal albumin result is between 3.4 and 5.4 grams per deciliter [53].
- **Chloride-BloodConcentration** : Chloride is an electrolyte that helps keep a proper fluid and acid-base balance in your body. It also helps in maintaining your blood pressure, blood volume, and pH levels. Chloride is also important to help the muscles and heart contract and to help our nerve cells carry messages (nerve impulses) between the brain and the body. More so, this mineral is needed to help red blood cells exchange oxygen and carbon dioxide in both the lungs. Lastly, chloride also plays a role in the digestion of foods, by supporting the production and release of hydrochloric acid (HCl) in the stomach [54].
- **Globulin-BloodConcentration** : Globulin is a blood protein. Grouping four families of proteins - alpha 1 globulins, alpha 2 globulins, beta-globulins and gamma globulins, it allows to transport lipids, ions and other vitamins. Globulin contributes to the formation of the blood clot and also helps to defend the body in that it contains antibodies [55].
- **Insulin-BloodConcentration** : Insulin is a hormone that is naturally produced by the pancreas in response to a high level of sugar (glucose) in the blood. It lowers blood sugar, lowers blood sugar. In fact, it "commands" the body's cells to take up glucose, which helps to limit the amount of glucose in the blood [56].

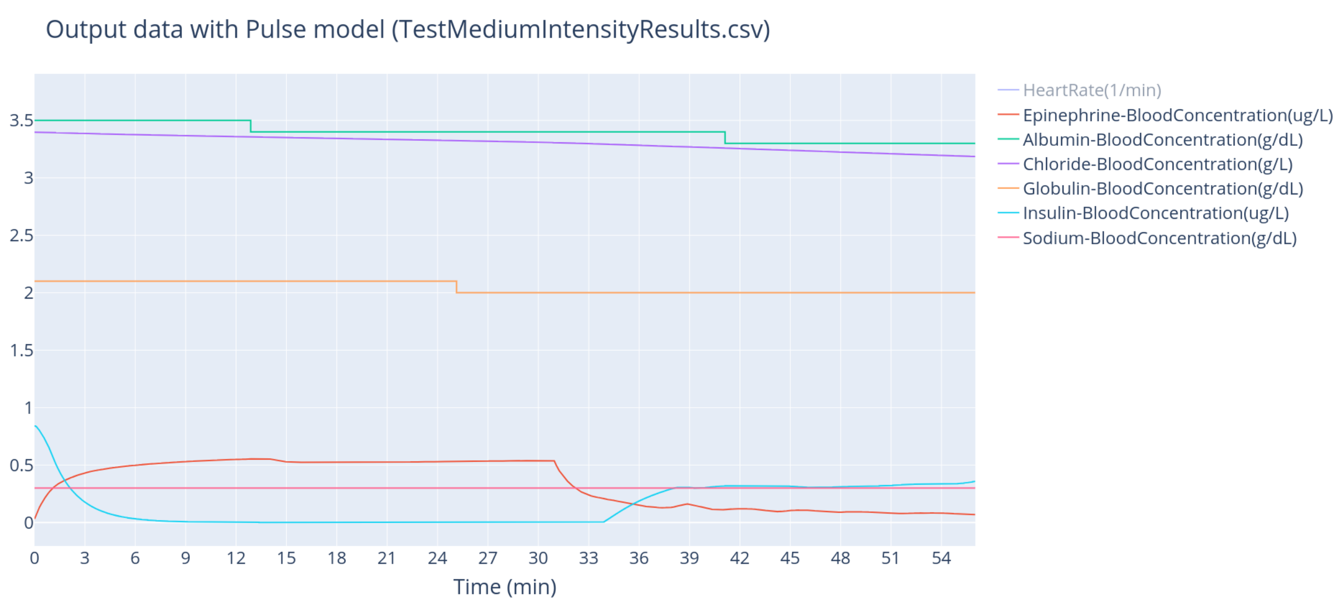

Figure 68: Output data with Pulse model

The graph **68** describes some of the nutrients and substances present in the cyclist's blood during the scenario **TestMediumIntensity**.
The blood concentration of epinephrine at the start of exercise is 0. 034 micrograms per liter. This amount increases rapidly during the first three minutes and then stabilizes at about 0. 54 micrograms per liter of blood until the 31st minute. This is followed by a sharp decrease to 0. 068 micrograms per liter of blood at the end of the simulation. It is noted that the fall of epinephrine occurs at a time when the heart rate also

decreases sharply.
The initial amount of albumin in the cyclist's blood is 3 5 grams per deciliter, which is the standard according to my research. There was then a decrease of 0.1 grams per deciliter at the 13th and 41st minute of exercise, then a stabilization until the end. A concentration of 3.3 grams per deciliter of blood is slightly below the ideal amount.
The chloride concentration in the cyclist's blood at the beginning of exercise is 3,398 grams per liter. This quantity decreases linearly until the end of the simulation, when it reaches 3.186 grams per liter.
The cyclist's blood initially contains 2.1 grams of globulin per deciliter. This amount drops to 2.0 grams per deciliter at the 25th minute and then remains constant until the end of the exercise.
It is interesting to note that the level of insulin in the cyclist's blood has almost a reversed trend relative to the amount of epinephrine. In fact, the blood initially contains 0. 842 micrograms per liter and then it decreases rapidly during the first 9 minutes to 0.007 micrograms per liter. This concentration remains stable until the 34th minute, then rises sharply and reaches 0.358 at the end of the simulation.
Finally, an astonishing observation is that the sodium concentration in the cyclist's blood remains constant at 0.3 grams per deciliter throughout the race. It is impossible that this amount will not vary, since research in the subsection **6.1** has shown the importance of regular use during exercise. This highlights the fact that the model is incomplete and does not yet consider the variation in blood sodium during exercise.

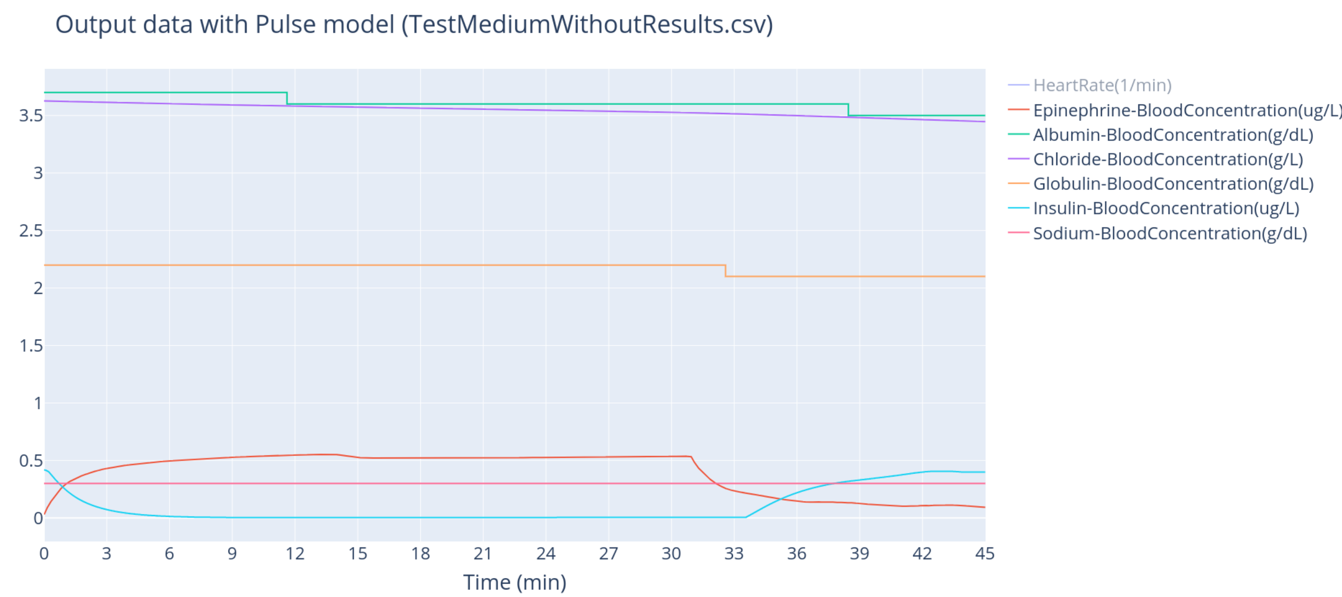

Figure 69: Output data with Pulse model

The graph **69** describes some of the nutrients and substances present in the cyclist's blood during the **TestMediumWithout** scenario.
Changes in blood epinephrine levels are almost identical to those observed in the previous simulation. On the other hand, the initial amount of albumin in the cyclist's blood is higher than before: it is 3.7 grams per deciliter. There was then a decrease of 0.1 grams per deciliter at the 11th and 38th minutes of exercise, then a stabilization until the end. In this case, the level of albumin in the blood is normal.
The amount of chloride in the cyclist's blood at the start of the exercise is 3,627 grams per liter, which is higher than in the graph **68**. However, it decreases at the same rate until the end of the simulation, when it reaches 3.447 grams per liter.
The cyclist's blood initially contains 2.2 grams per deciliter and this amount drops to 2. 1 grams per deciliter at the 32nd minute. This is a slightly higher concentration than if the cyclist consumes a meal and nutrients.

As with the **TestMediumIntensity** scenario, it is interesting to note that the level of insulin in the cyclist's blood has almost a reversed trend relative to the amount of epinephrine. Thus, the changes are almost the same as those observed earlier.
Finally, we see the same phenomenon as in the graph **68**: the sodium concentration in the cyclist's blood remains constant at 0.3 grams per deciliter throughout the race.

Output data	MAE	MSE
Epinephrine-BloodConcentration(ug/L)	0.004523096033480241	4.46936326173401e-05
Albumin-BloodConcentration(g/dL)	0.19129217436391266	0.03738765230917381
Chloride-BloodConcentration(g/L)	0.21960289692974336	0.048254384735009814
Globulin-BloodConcentration(g/dL)	0.11655642383615433	0.014966927150846292
Insulin-BloodConcentration(ug/L)	0.03427364912410652	0.00664453258768194
Sodium-BloodConcentration(g/dL)	0.0	0.0

Table 14: Errors between TestMediumIntensityResults.csv and TestMediumWithoutResults.csv

Let's describe the six new data appearing in Figure **70** :

- **Acetoacetate-BloodConcentration** : Acetoacetic acid (acetoacetate) is a ketone body that acts as an intermediate in metabolism. It is of particular importance for catabolism during hunger or diet [47].
- **Creatinine-BloodConcentration** : Creatinine comes from the breakdown of creatine, which is synthesized by the liver and stored in the muscles where it plays an important role in energy production. The use of creatine by muscles produces waste, the most notable being creatinine. The This is carried by the blood, filtered by the kidneys and excreted in the urine [48].
- **Hemoglobin-BloodConcentration** : Hemoglobin is a protein found inside red blood cells, which are produced in the bone marrow. It's responsible for the red colour of the blood. Its main function is to transport oxygen from the lungs to organs, muscles and all tissues through the arterial circulation. After delivering oxygen to the tissues, it returns to the lungs, for example by carrying carbon dioxide from the body. Hemoglobin contains 65% of the body's iron. The normal amount of haemoglobin is between 13 and 18 g/dL of blood in humans [49].
- **Potassium-BloodConcentration** : Potassium is an essential mineral for the transmission of nerve impulses and muscle contraction, including that of the heart muscle. Normal blood levels of potassium are between 3.5 and 5 mmol/L, or between 630.56 and 900.8 mg/L [50].
- **Glucose-BloodConcentration** : Glucose is the basic component of most carbohydrates. It is a simple sugar, or monosaccharide, which plays a central role in the proper functioning of cells by providing them with energy. It is stored in the liver and muscles as glycogen and is synthesized as energy needs increase [51]. The level of glucose in the blood is called blood sugar.
- **Lactate-BloodConcentration** : Lactate is a biological molecule and corresponds to the ionized form of lactic acid. It is the end product of anaerobic glycolysis (fermentation). Lactate in the blood is taken up by various organs or cells of the human body. In the liver, it is used to synthesize glucose through gluconeogenesis. In the heart, it is oxidized to carbon dioxide (CO_2) at the same time as energy is released [52].

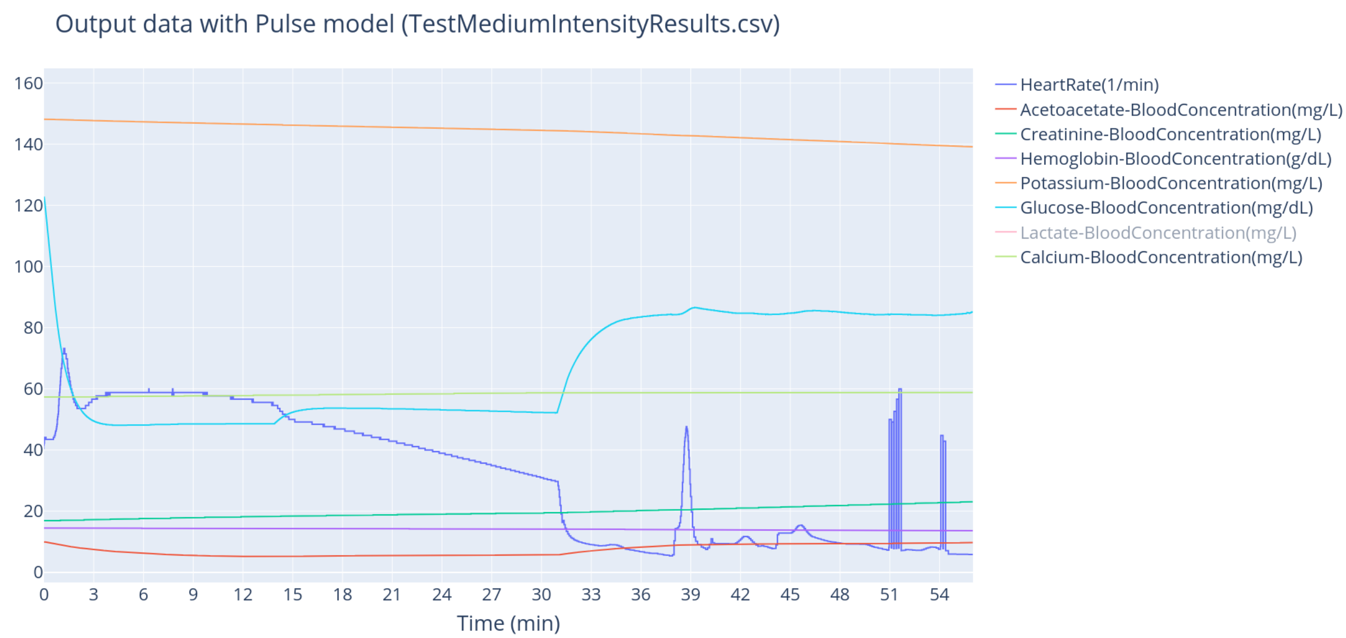

Figure 70: Output data with Pulse model

The **70 figure** represents some nutrients and substances contained in the cyclist's blood during the simulation **TestMediumIntensity**. The amount of acetoacetate is 10 milligrams per liter of blood at the beginning of exercise. We see that it decreases during the first 10 minutes and then the concentration increases from the 31st minute. Note that this moment corresponds to the peak of the decrease in heart rate.

Blood creatinine levels were initially 16.8 milligrams per liter and increased slowly until the end of the simulation, when they reached 23.1 milligrams per liter. Similarly, the amount of hemoglobin in the cyclist's blood varies little, from 14.5 grams per deciliter to 13.62 when the scenario ends.

Cyclist's blood initially contains 148 milligrams of potassium per liter and decreases almost linearly to 139 milligrams per liter.

Changes in the amount of glucose in the blood occur mostly when heart rate changes. In fact, we initially see 122. 7 milligrams per deciliter, but this amount decreases sharply during the first peak of heart rate, until the 4th minute. Stabilization was observed at 48. 5 milligrams per deciliter as long as the heart rate was stable, followed by an increase at the 14th minute when the heart rate dropped. Similarly, from the 31st minute onwards the heart rate drops dangerously, which corresponds to the time when the glucose concentration rises sharply. Blood contains 85. 1 milligrams of glucose per deciliter when the simulation is over.

The amount of calcium in the cyclist's blood drops from 57.3 milligrams per liter to 58.8 milligrams per liter at the end of the exercise.

The amount of lactate in the blood is not shown on the graph because it made the other data illegible. However, this concentration increases continuously during the simulation, between 163 and 257 milligrams per liter of blood. It was noticeable that the increase slowed slightly from the 31st minute.

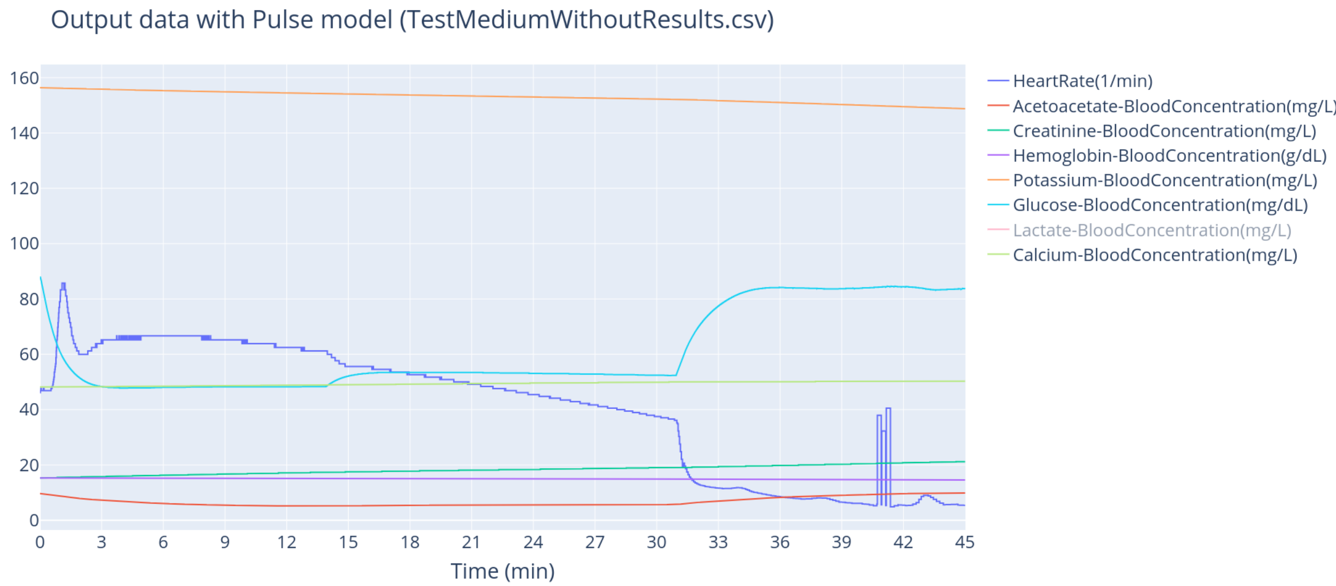

Figure 71: Output data with Pulse model

Figure **71** shows some of the nutrients and substances in the cyclist's blood during the simulation **TestMediumWithout**. The amount of acetoacetate in the blood was found to be almost the same as in the scenario **TestMediumIntensity**.
Blood creatinine concentration show the same changes as previously observed, but with slightly lower values: 15.3 milligrams per liter initially.
Similarly, the amount of hemoglobin in the cyclist's blood varies in the same way as in the previous simulation, but the initial concentration is higher: 15.27 grams per deciliter of blood.
Cyclist's blood initially contains 156 milligrams of potassium per liter and decreases almost linearly to 148 milligrams per liter. On the other hand, the blood glucose concentration is initially much lower when the cyclist has not eaten meals or nutrients: 88 milligrams per deciliter compared to 122.7 in the previous scenario. This is the only big difference because during the rest of the exercise the glucose changes are very similar to those seen in Figure **70**.
The cyclist's blood contains almost 10 milligrams of calcium per liter less than in the scenario **TestMediumIntensity**. This calcium concentration drops to 50.3 milligrams per liter of blood at the end of the exercise.
The amount of lactate in the blood is also lower than in the previous simulation: between 146 and 242 milligrams per liter of blood. It was noticeable that the increase also slowed slightly from the 31st minute.

Output data	**MAE**	**MSE**
Acetoacetate-BloodConcentration(mg/L)	0.10147576756416431	0.03088241994000222
Creatinine-BloodConcentration(mg/L)	0.745425724973149	0.7155997185289433
Hemoglobin-BloodConcentration(g/dL)	0.7406934557979333	0.5490242050294434
Potassium-BloodConcentration(mg/L)	7.811040257768229	61.05746736120885
Glucose-BloodConcentration(mg/dL)	1.1892811377356396	15.855571052923967
Lactate-BloodConcentration(mg/L)	13.369492981741416	181.472737602311
Calcium-BloodConcentration(mg/L)	8.86141328098959	78.55388696714934

Table 15: Errors between TestMediumIntensityResults.csv and TestMediumWithoutResults.csv

6.3.4 Medium capacity : cyclist meal vs normal meal

It was noted that eating meals/nutrients resulted in variations in some output data. Now it is a question of simulating a scenario in which the cyclist eats a meal from a non-sporting individual. The aim is to compare it to a classic scenarion **CyclistMeal**, i.e. a suitable pre-race meal and energy drink during exercise. So I added a normal breakfast below :

- 50g of bread
- 10g sweet butter
- 15g of honey or jam
- 250mL of black coffee (a large bowl)
- 300mL of pure orange juice (two glasses or two oranges)
- 250g of plain yogurt (two units)

This is the same as a cyclist's breakfast, but the quantities are lower, as a non-athlete does not need as many nutrients and calories.

Nutrients	**Quantities**
Energie (kcal)	470
Glucides (g)	68
Protides (g)	18
Lipides (g)	12
Calcium (mg)	452
Sodium (mg)	524

Table 16: Nutritional benefits of a normal breakfast

The **CyclistMeal** scenario is exactly the same as **TestMediumIntensity**, I renamed it because it focuses on comparing a sports meal with a regular meal.

JSON files corresponding to the **CyclistMeal** and **NormalMeal** scenarios are stored in the data folder on GitHub.

The simulations of **CyclistMeal.json** and **NormalMeal.json** were limited to 2 hours 56 minutes, an additional time would have resulted in an error due to the cyclist's poor health. In this period are included 2 hours corresponding to the intake of the meal as well as the digestion before the race. In the end, the race lasted only 56 minutes. The execution of the full script took about 21 minutes.
Note that the meals eaten before the race are different in both cases, but this did not affect the duration of the simulation.

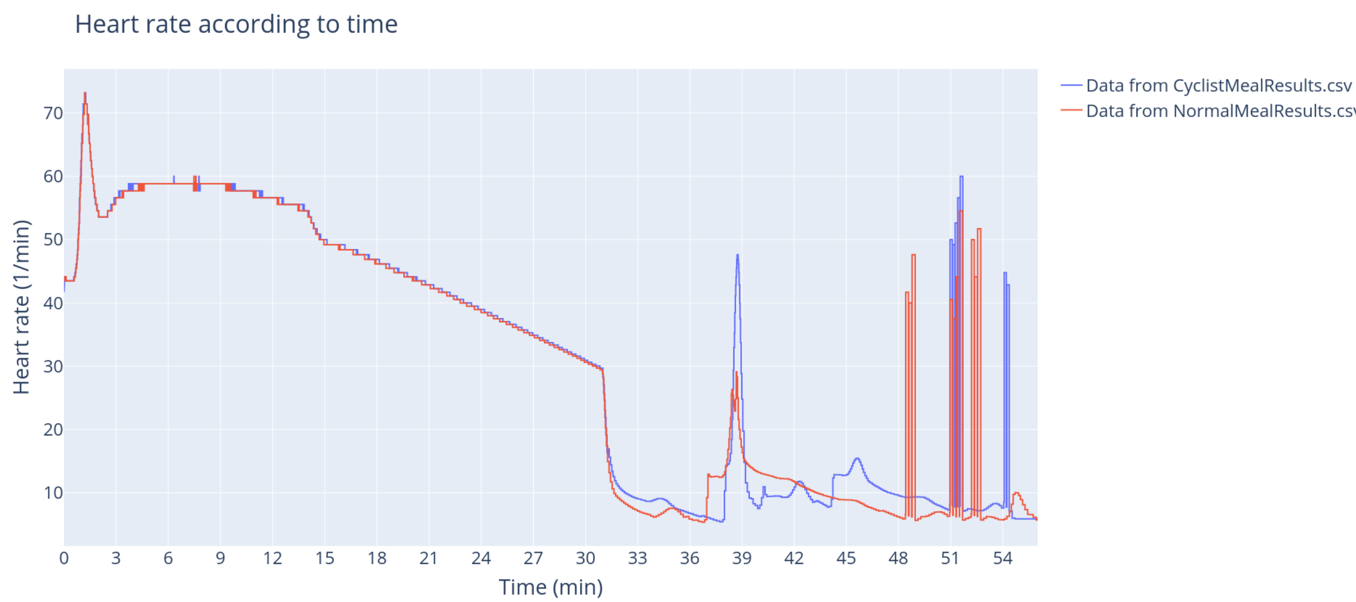

Figure 72: Output data with Pulse model

Figure **72** shows that heart rates in both scenarios are almost identical up to the 31st minute.

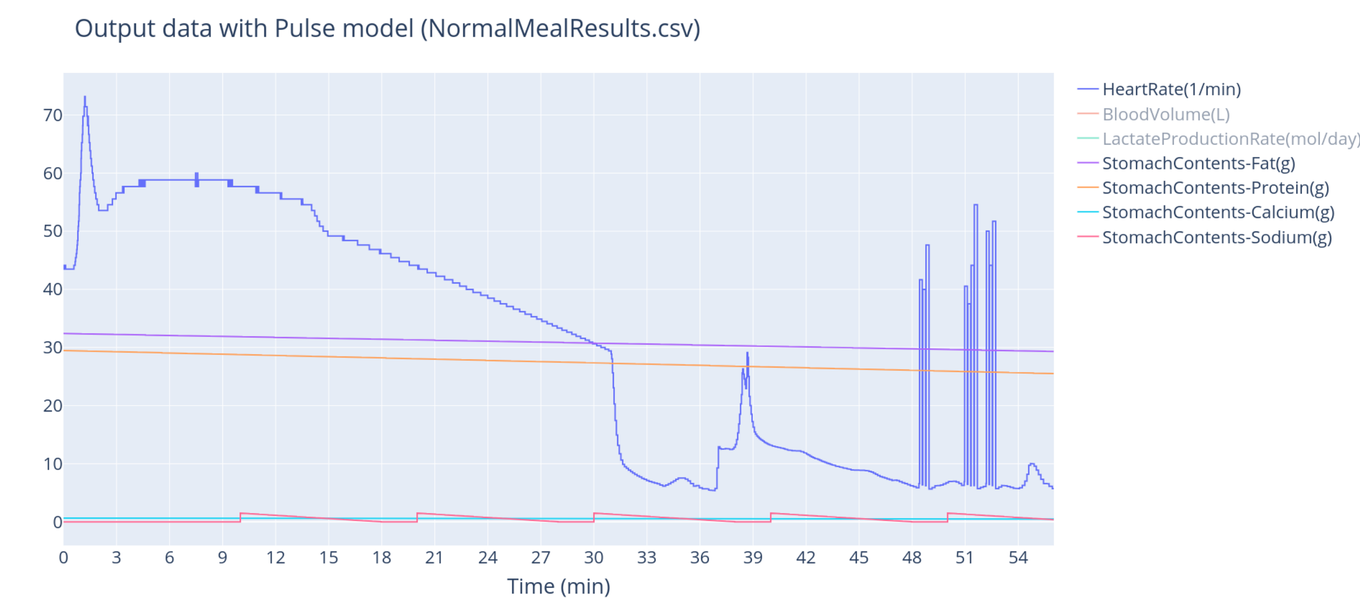

Figure 73: Output data with Pulse model

On the graph **73** we see how the contents of the cyclist's stomach change during the **NormalMeal** scenario. The amount of fat in the stomach is initialized to 32.4 g and decreases to 29.32 g at the end of the simulation. If you compare it with a meal adapted to this effort, you can see that during the course of the exercise there are 27 grams less fat in the cyclist's stomach.
The cyclist starts the exercise with 29.48 g of protein in the stomach and finishes the exercise with 25.5 g

of protein. It is noticeable that the cyclist has 13 grams less protein in his stomach, compared to a sporty meal.
The amount of calcium at the beginning of the exercise is 0. 63 g and reaches 0. 48 at the end of the simulation. Here again, we find that the stomach has 7 grams less calcium than if the cyclist had eaten a suitable meal.
Finally, the amount of sodium in the stomach is exactly the same as observed in the **TestMediumIntensity** scenario (=**CyclistMeal**) (See Figure **67**). There are small spikes of 1. 5 grams every 10 minutes, which correspond to the times when I set an energy drink consumption. Note that the high amount of sodium is still consumed entirely before the next 10-minute interval.

Output data	MAE	MSE
StomachContents-Fat(g)	27.0	729.0
StomachContents-Protein(g)	13.0	169.0
StomachContents-Calcium(g)	0.0690080175230794	0.004771042278000322
StomachContents-Sodium(g)	0.0	0.0

Table 17: Errors between CyclistMealResults.csv and NormalMealResults.csv

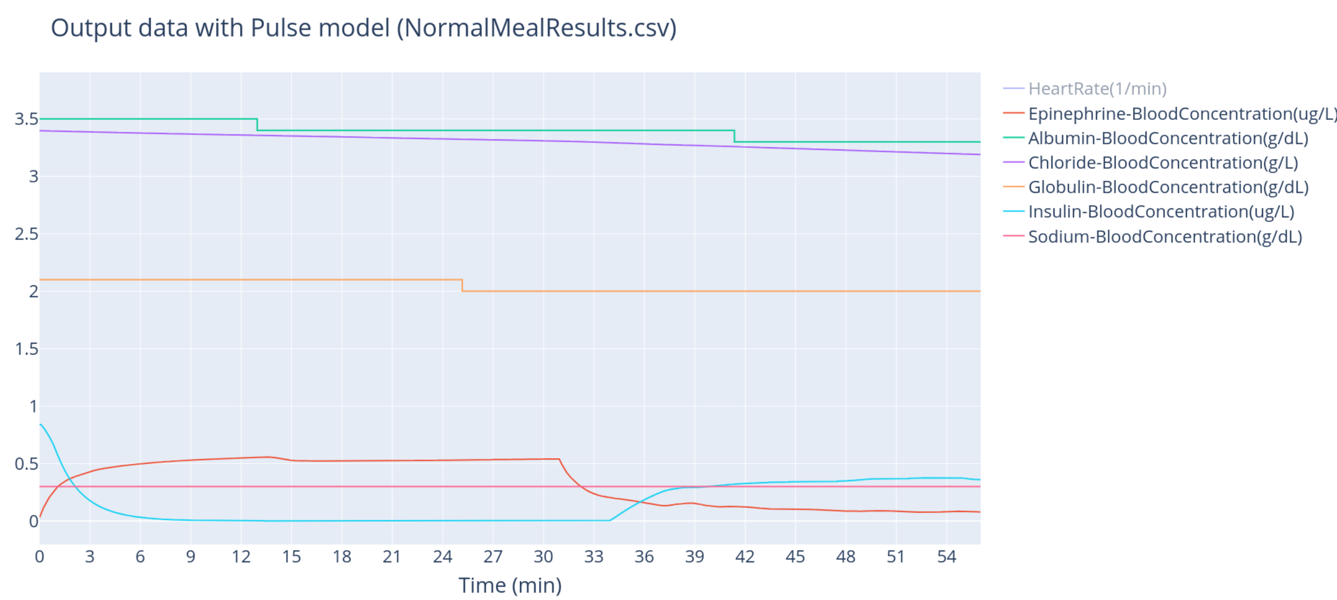

Figure 74: Output data with Pulse model

The graph **74** describes some of the nutrients and substances present in the cyclist's blood during the **NormalMeal** scenario. Observations show that the variations and amounts of these data are almost similar to those described for the **TestMediumIntensity** scenario (=**CyclistMeal**) (See Figure **68**). This may be due to the fact that heart rate curves are very similar.
Table **18** shows that the observation is correct.

Output data	MAE	MSE
Epinephrine-BloodConcentration(ug/L)	0.0023102219490854546	1.6664624246495683e-05
Albumin-BloodConcentration(g/dL)	0.0005975941478628871	5.975941478628878e-05
Chloride-BloodConcentration(g/L)	0.000989533769426274	2.5285913086954713e-06
Globulin-BloodConcentration(g/dL)	5.4759623110941854e-05	5.475962311094191e-06
Insulin-BloodConcentration(ug/L)	0.009279494306784835	0.00033580996029927316
Sodium-BloodConcentration(g/dL)	0.0	0.0

Table 18: Errors between CyclistMealResults.csv and NormalMealResults.csv

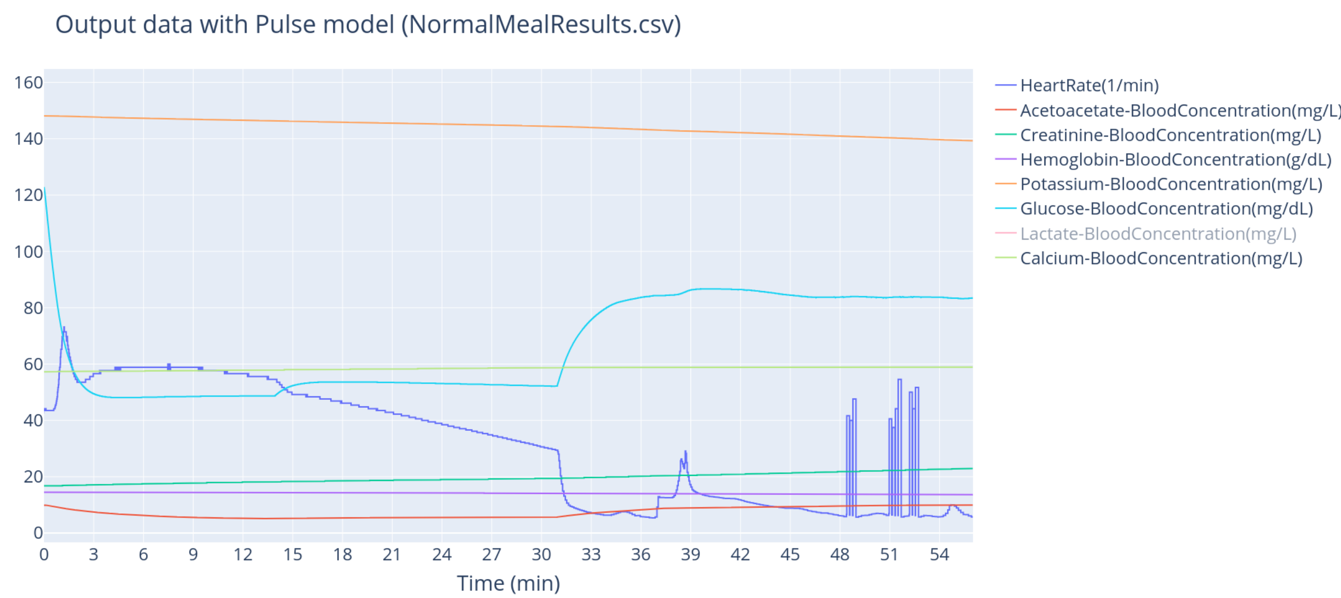

Figure 75: Output data with Pulse model

The **75** graph describes a part nutrients and substances in the cyclist's blood during the **NormalMeal** scenario.
Observations show that the variations and amounts of these data are almost similar to those described for the **TestMediumIntensity** scenario (=**CyclistMeal**) (See graph **70**).
Table **19** allows us to make the same observation.

Output data	MAE	MSE
Acetoacetate-BloodConcentration(mg/L)	0.06866456754778075	0.019270220431291552
Creatinine-BloodConcentration(mg/L)	0.08793562173004696	0.009866731743320218
Hemoglobin-BloodConcentration(g/dL)	0.0045278470539917934	6.452409721023524e-05
Potassium-BloodConcentration(mg/L)	0.04535239603111773	0.0052214657722594905
Glucose-BloodConcentration(mg/dL)	0.33163618182575705	0.35801936824060854
Calcium-BloodConcentration(mg/L)	0.029042837500818784	0.002904283750081916
Lactate-BloodConcentration(mg/L)	0.12459480854964369	0.06651008588927834

Table 19: Errors between CyclistMealResults.csv and NormalMealResults.csv

Finally, the difference in food ingested before exercise has very little influence on the vital output data obtained. However, it would be interesting to check that we get the same results by using the **ConsumeMeal** condition, when available.

6.3.5 Medium capacity : only water vs cyclist meal

This part simulates a scenario in which the cyclist eats a cyclist's meal but consumes only water during the exercise. The aim is to compare the results obtained with the classic **CyclistMeal** simulation, i.e. a suitable pre-race meal and an energy drink during exercise.

The JSON file corresponding to the **OnlyWater** scenario is stored in the data folder on GitHub.

The simulation of **OnlyWater.json** was limited to 2 hours 56 minutes, an additional time would have caused an error due to the cyclist's poor health. In this period are included 2 hours corresponding to the intake of the meal as well as the digestion before the race. In the end, the race lasted only 56 minutes. The execution of the full script took about 21 minutes.

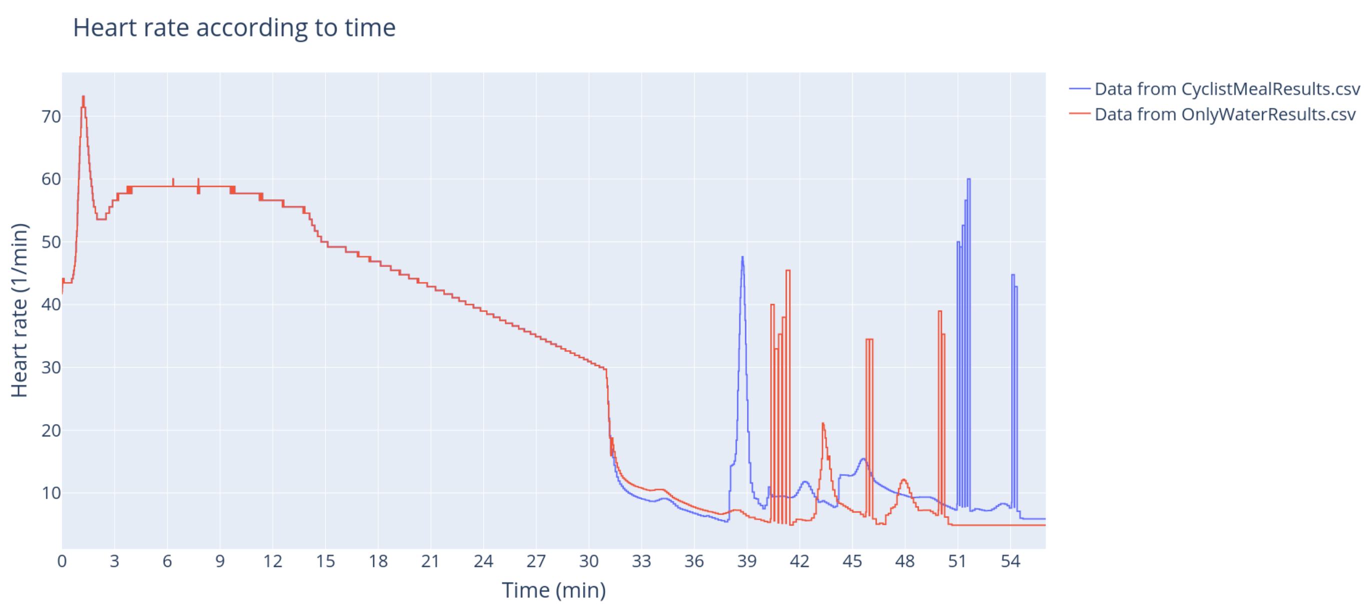

Figure 76: Output data with Pulse model

Figure **76** shows that heart rates in both scenarios **CyclistMeal** and **OnlyWater** are almost identical up to the 31st minute.

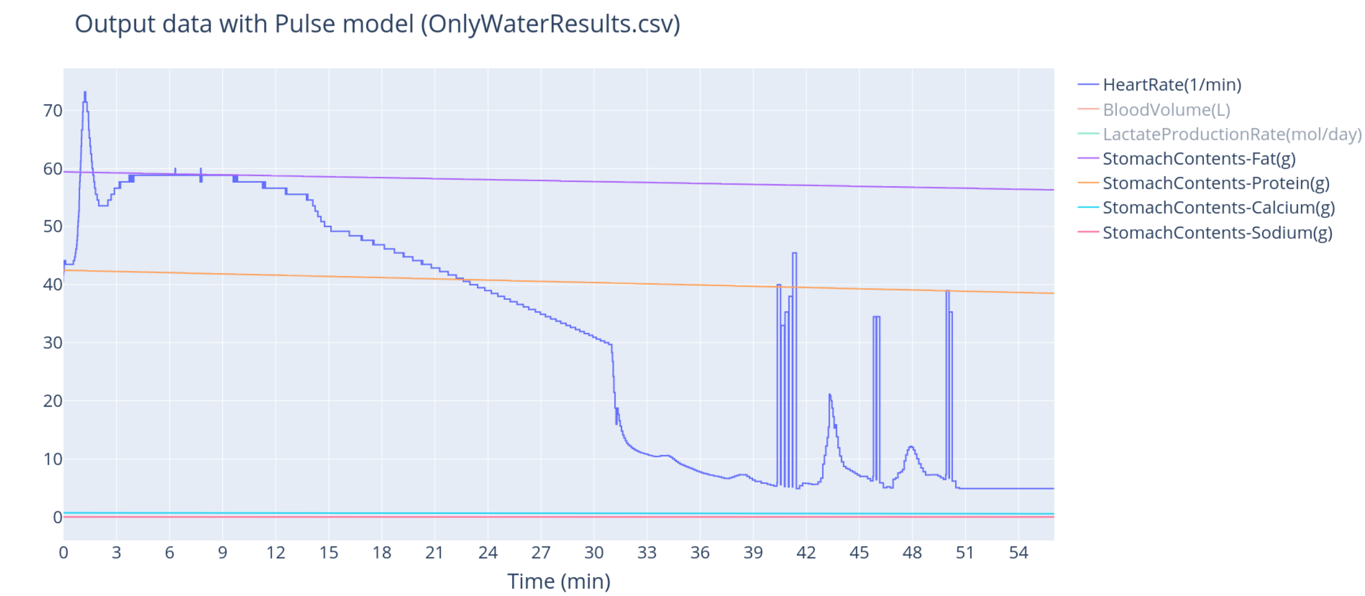

Figure 77: Output data with Pulse model

On the graph **77** we observe how the contents of the cyclist's stomach change during the **OnlyWater** scenario. During the simulation, the amount of nutrients in the stomach decreases in a linear fashion.
With the exception of sodium, changes and amounts of nutrients are similar to those seen on the **66** graph, which corresponds to the **TestMediumIntensity** scenario (=**CyclistMeal**).
The amount of sodium in the stomach was 0 g during the entire simulation, despite eating the meal described in the table **11** 2 hours earlier. The entire amount of sodium had to be digested before the start of the exercise, and since the cyclist only consumes water during the run, he cannot recharge his sodium gauge.

Output data	MAE	MSE
StomachContents-Fat(g)	0.0	0.0
StomachContents-Protein(g)	0.0	0.0
StomachContents-Calcium(g)	0.0	0.0
StomachContents-Sodium(g)	0.5285867255531019	0.5335821656240514

Table 20: Errors between CyclistMealResults.csv and OnlyWaterResults.csv

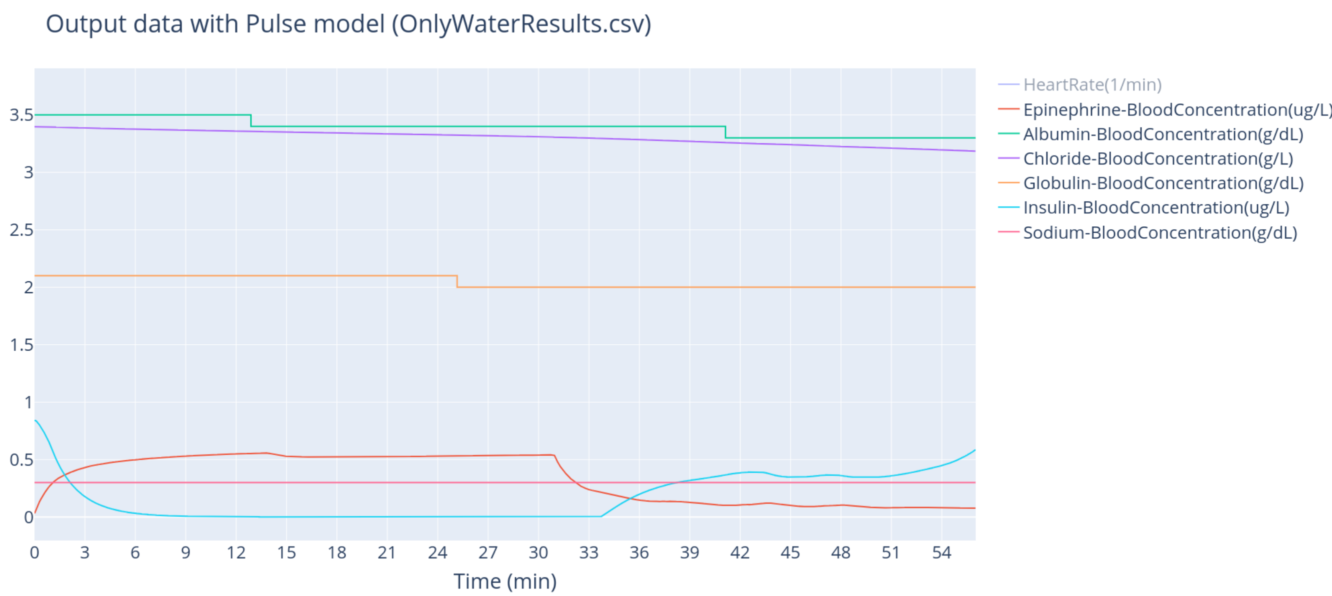

Figure 78: Output data with Pulse model

The **78** graph describes a part nutrients and substances present in the cyclist's blood during the **OnlyWater** scenario.
Observations show that the variations and amounts of these data are almost similar to those described for the **TestMediumIntensity** scenario (=**CyclistMeal**) (See Figure **68**).
We see that **21** confirms this observation.

Output data	MAE	MSE
Epinephrine-BloodConcentration(ug/L)	0.003032215336265751	4.0449560831882005e-05
Albumin-BloodConcentration(g/dL)	1.5475545661787917e-05	1.547554566178793e-06
Chloride-BloodConcentration(g/L)	0.00033410810263858053	3.302211812603045e-07
Globulin-BloodConcentration(g/dL)	3.511758438636489e-05	3.5117584386364913e-06
Insulin-BloodConcentration(ug/L)	0.019853916801085666	0.0017659641503032612
Sodium-BloodConcentration(g/dL)	0.0	0.0

Table 21: Errors between CyclistMealResults.csv and OnlyWaterResults.csv

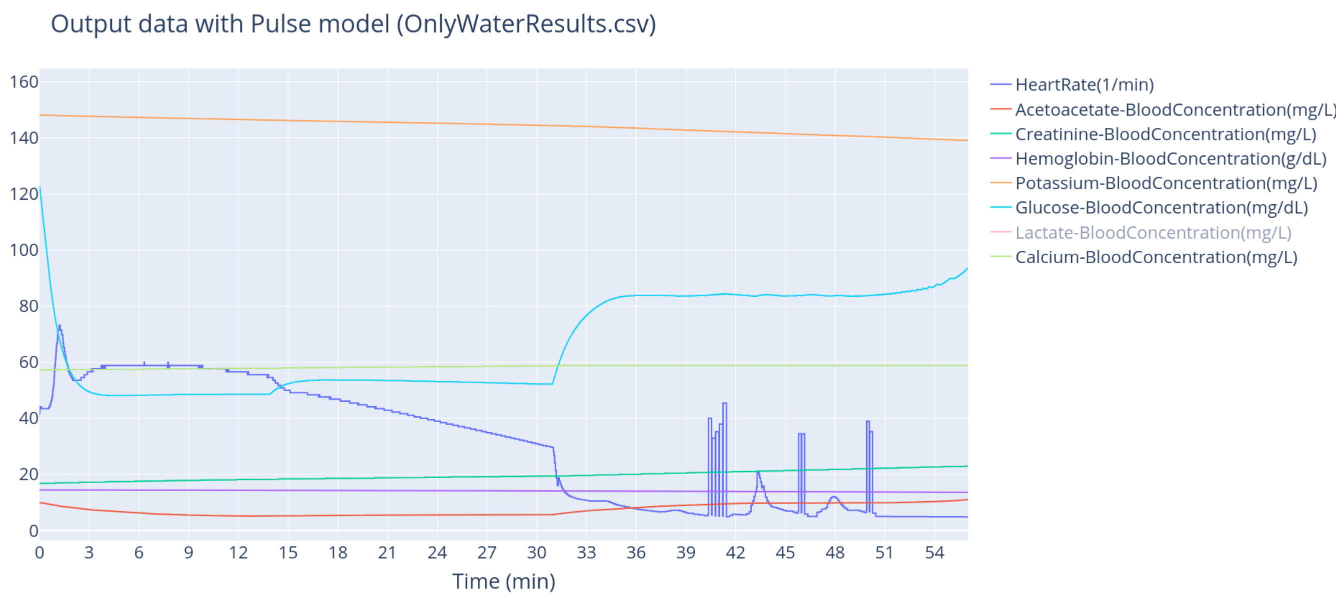

Figure 79: Output data with Pulse model

The **79** graph describes a part nutrients and substances in the cyclist's blood during the **OnlyWater** scenario.
Observations show that the variations and amounts of these data are almost similar to those described for the **TestMediumIntensity** scenario (=**CyclistMeal**) (See graph **70**). Just at the end of the simulation, we notice an increase in the glucose concentration, which reaches 94 milligrams per deciliter of blood. However, as described in the subsection **6.3.1**, the cyclist's health is considered to be poor from about the 32nd minute onwards, as the heart rate drops below 16 beats per minute.
Table **22** confirms this trend.

Output data	**MAE**	**MSE**
Acetoacetate-BloodConcentration(mg/L)	0.15163664013999417	0.08596295956120877
Creatinine-BloodConcentration(mg/L)	0.03652526382829286	0.003652526382829286
Hemoglobin-BloodConcentration(g/dL)	0.0015137464510407257	1.5137464510407212e-05
Potassium-BloodConcentration(mg/L)	0.014425291803317709	0.0006644931461189087
Glucose-BloodConcentration(mg/dL)	0.5673763593183622	1.6507109822805008
Calcium-BloodConcentration(mg/L)	0.0023951382978090463	0.00023951382978090724
Lactate-BloodConcentration(mg/L)	0.26479789532578984	0.257286839238841

Table 22: Errors between CyclistMealResults.csv and OnlyWaterResults.csv

Finally, whether or not nutrients are consumed during exercise has very little influence on the vital output data obtained. In fact, with the exception of sodium, nutrients were present in the same amounts in both scenarios.

6.3.6 Medium capacity : nutrients without water vs cyclist meal

The next step is to simulate a scenario in which the cyclist eats a cyclist's meal but consumes only nutrients during exercise, without water. The aim is to compare the results obtained with the classic **CyclistMeal** simulation, i.e. a suitable pre-race meal and an energy drink during the race.

The JSON file corresponding to the **NutrientsWithoutWater** scenario is stored in the data folder on GitHub.

The simulation of **NutrientsWithoutWater.json** was limited to 3 hours 11 minutes, an additional time would have resulted in an error due to the cyclist's poor health. In this period are included 2 hours corresponding to the intake of the meal as well as the digestion before the race. In the end, the race lasted only 1 hour 11 minutes. The execution of the full script took about 22 minutes.

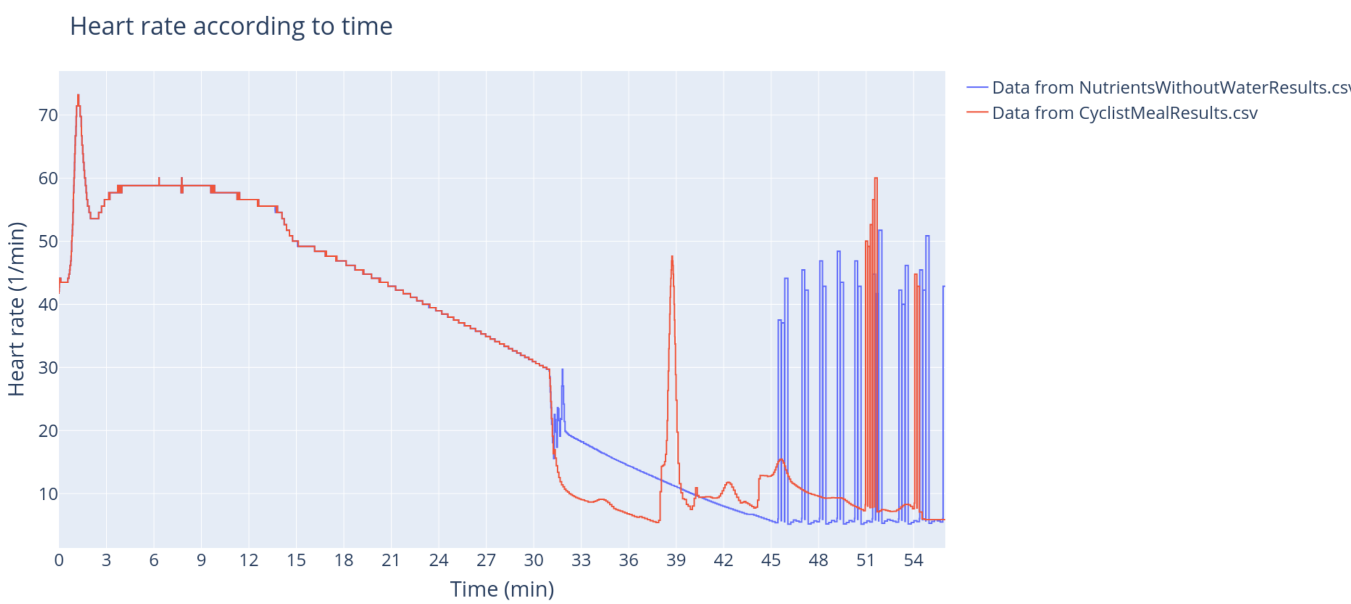

Figure 80: Output data with Pulse model

Figure **80** shows an exact similarity in heart rate between the scenario **CyclistMeal** and **NutrientsWithoutWater** up to 31st minute.

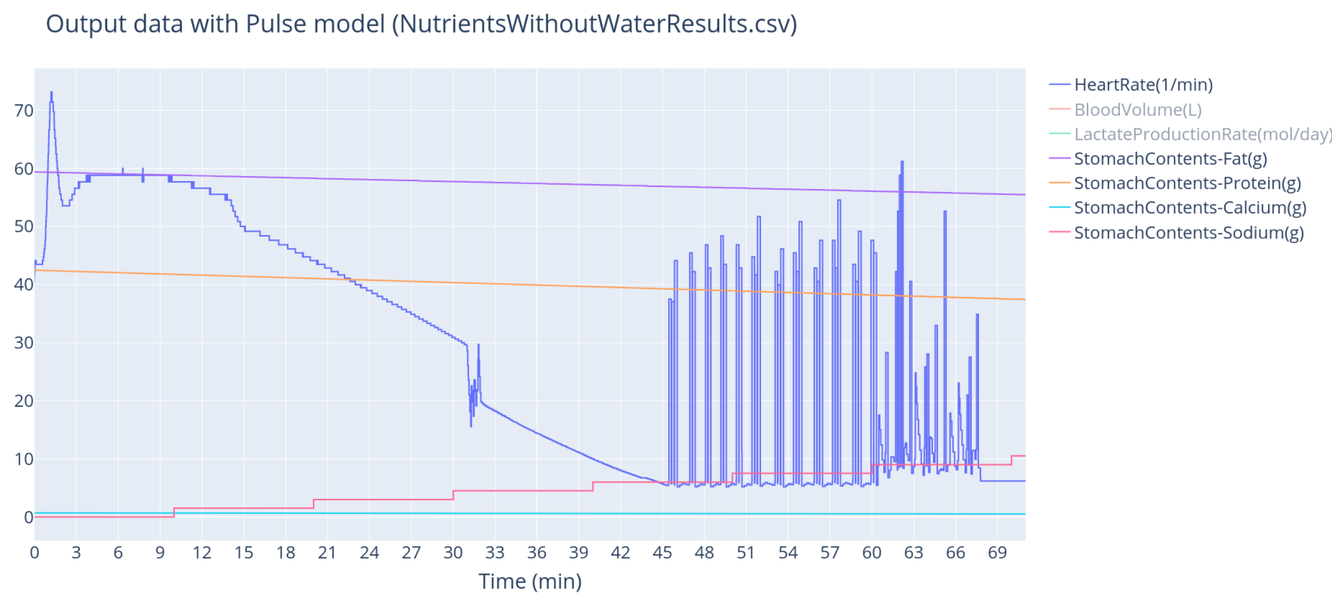

Figure 81: Output data with Pulse model

On the graph **81** we observe how the contents of the cyclist's stomach change during the simulation. With the exception of sodium, changes and amounts of nutrients are similar to those seen on the **66** graph, which corresponds to the **TestMediumIntensity** scenario (=**CyclistMeal**). The initial amount of sodium in the stomach is 0 g. This quantity increases by 1.5 g every 10 minutes, reaching 10.5 g at the end of the simulation. These peaks correspond to the time when the cyclist consumes 15 g of carbohydrate and 1500 mg of sodium during exercise.
Table **23** confirms this observation.

Output data	**MAE**	**MSE**
StomachContents-Fat(g)	0.0	0.0
StomachContents-Protein(g)	0.0	0.0
StomachContents-Calcium(g)	0.0	0.0
StomachContents-Sodium(g)	2.9535449713404796	13.896570017320707

Table 23: Errors between CyclistMealResults.csv and NutrientsWithoutWaterResults.csv

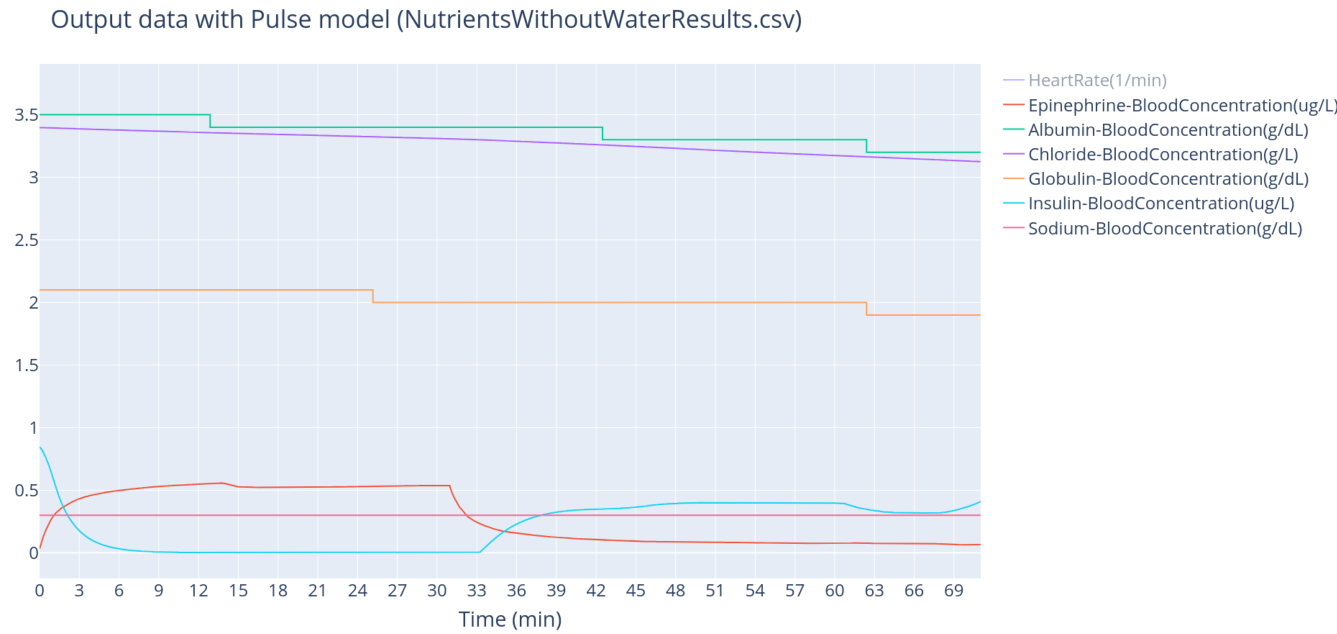

Figure 82: Output data with Pulse model

The graph **82** describes a portion of the nutrients and substances present in the cyclist's blood during the **NutrientsWithoutWater** scenario. The blood level of epinephrine at the start of exercise is 0.034 micrograms per liter. This amount increases rapidly during the first three minutes and then stabilizes at about 0. 54 micrograms per liter of blood until the 31st minute. There is then a sharp decrease in the concentration to 0.065 micrograms per liter of blood at the end of the simulation. It is noted that the fall of epinephrine occurs at a time when the heart rate also decreases sharply.

The initial amount of albumin in the cyclist's blood is 3.5 grams per deciliter, which is the standard according to my research. There is then a decrease of 0.1 grams per deciliter at the 13th, 43rd and 62nd minute of exercise, then a stabilization until the end. A concentration of 3.2 grams per deciliter of blood is a little below the ideal amount.

The chloride concentration in the cyclist's blood at the beginning of the exercise is 3.398 grams per liter. This quantity decreases linearly until the end of the simulation, where it reaches 3,125 grams per liter.

The cyclist's blood initially contains 2.1 grams of globulin per deciliter. This amount decreases to 2.0 grams per deciliter at the 25th minute and remains at 1.9 grams per deciliter from the 62nd minute.

It is interesting to note that the level of insulin in the cyclist's blood has almost a reversed trend relative to the amount of epinephrine. In fact, the blood initially contains 0.842 micrograms per liter, then it decreases rapidly during the first 9 minutes to 0.007 micrograms per liter. This concentration remains stable until the 33rd minute, then rises sharply and reaches 0.41 at the end of the simulation.

Finally, the sodium concentration in the cyclist's blood remains constant and equal to 0.3 grams per deciliter throughout the race.

Output data	MAE	MSE
Epinephrine-BloodConcentration(ug/L)	0.0031404645937371657	4.20332020458969e-05
Albumin-BloodConcentration(g/dL)	0.0024403745082050175	0.00024403745082050195
Chloride-BloodConcentration(g/L)	0.0023185962489658167	1.3456153255519115e-05
Globulin-BloodConcentration(g/dL)	3.2141517912944135e-05	3.2141517912944157e-06
Insulin-BloodConcentration(ug/L)	0.021366633533126598	0.00134316812989935
Sodium-BloodConcentration(g/dL)	0.0	0.0

Table 24: Errors between CyclistMealResults.csv and NutrientsWithoutWaterResults.csv

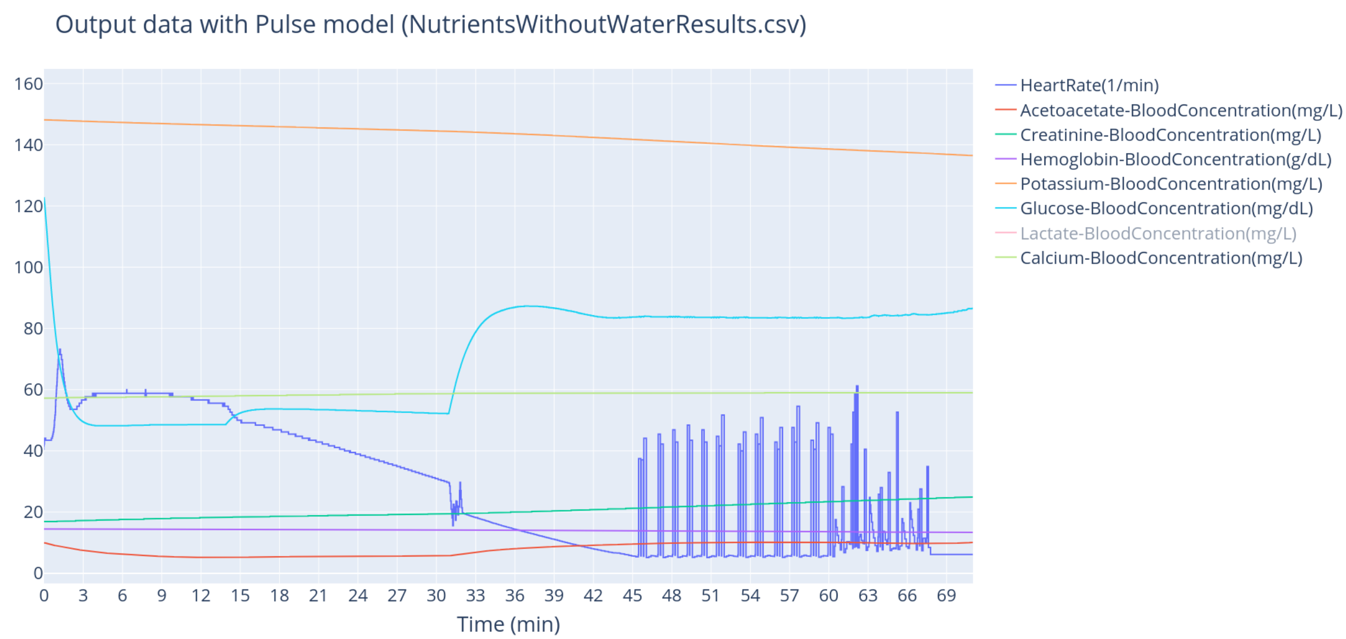

Figure 83: Output data with Pulse model

Figure **83** shows that the changes in substances in the cyclist's blood are approximately similar to those observed during the **TestMediumIntensity** scenario (=**CyclistMeal**).
Table **25** confirms this observation.

Output data	MAE	MSE
Acetoacetate-BloodConcentration(mg/L)	0.14357727356598235	0.0659146583594731
Creatinine-BloodConcentration(mg/L)	0.09424607308028832	0.018506966971614264
Hemoglobin-BloodConcentration(g/dL)	0.011244412435196138	0.00032185563696750653
Potassium-BloodConcentration(mg/L)	0.10455986952924581	0.027401823138321617
Glucose-BloodConcentration(mg/dL)	0.60788717136786	1.3212700066068677
Calcium-BloodConcentration(mg/L)	0.03940550096127002	0.004264584213753118
Lactate-BloodConcentration(mg/L)	0.18904569452463296	0.10188111209651982

Table 25: Errors between CyclistMealResults.csv and NutrientsWithoutWaterResults.csv

6.4 Conclusion.

Finally, through the various comparisons of scenarios, it was found that the link between the stomach and the blood is not yet modeled. Although there was a difference in sodium concentration in stomach between the different scenarios, the blood sodium concentration was still the same in the different cases studied.

These observations confirm the fact that nutrient intake is not yet reflected in the models, and this helps to understand why the terminal displays an early alert about cyclist fatigue. This also explains why the simulations lasted the same time before error : the energy reserves do not take nutrients as parameters to regenerate. This means that it is currently impossible to model correctly the vital data of a cyclist during a race.

References

[1] van Erp, T., Sanders, D., Lamberts, R-P. (2021, February). Maintaining Power Output with Accumulating Levels of Work Done Is a Key Determinant for Success in Professional Cycling. *American College of Sports Medicine.*

[2] Puissance critique et W'balance (2020) https://blog.nolio.io/puissance-critique-wbal/

[3] Comparison of W'balance algorithms (2018) https://medium.com/critical-powers/comparison-of-wbalance-algorithms-8838173e2c15

[4] TeamGantt (2021, April 2). *Free Gantt Chart Excel Template.* Retrieved from https://www.teamgantt.com/free-gantt-chart-excel-template

[5] What is Pulse? (2017-2021). *Pulse Physiology Engine.* https://pulse.kitware.com/index.html

[6] Common Data Model and Software Framework. (2017-2021). *Pulse Physiology Engine.* https://pulse.kitware.com/_c_d_m.html

[7] CDM Tables. (2017-2021). *Pulse Physiology Engine.* https://pulse.kitware.com/_c_d_m_tables.html

[8] Scenario Files. (2017-2021). *Pulse Physiology Engine.* https://pulse.kitware.com/_scenario_file.html

[9] Cardiovascular Methodology. (2017-2021). *Pulse Physiology Engine.* https://pulse.kitware.com/_cardiovascular_methodology.html

[10] Circuit Methodology. (2017-2021). *Pulse Physiology Engine.* https://pulse.kitware.com/_circuit_methodology.html

[11] Energy Methodology. (2017-2021). *Pulse Physiology Engine.* https://pulse.kitware.com/_energy_methodology.html

[12] Environment Methodology. (2017-2021). *Pulse Physiology Engine.* https://pulse.kitware.com/_environment_methodology.html

[13] Patient Methodology. (2017-2021). *Pulse Physiology Engine.* https://pulse.kitware.com/_patient_methodology.html

[14] Qui sommes-nous. *Groupama-FDJ Equipe Cycliste.* https://www.equipecycliste-groupama-fdj.fr/qui-sommes-nous/

[15] Baudoin, P. (2011). VO2max : un indicateur de performance. *Sport Passion - Conseils et entraînement du sportif.* https://www.sport-passion.fr/conseils/VO2max.php

[16] Shahoud, J-S., Sanvictores, T., Aeddula, N-R. (2020, September 6). Physiology, Arterial Pressure Regulation. *NCBI.* https://www.ncbi.nlm.nih.gov/books/NBK538509/#:~:text=Systolic%20pressure%20refers%20to%20the,heart%20muscle%20relaxation%20between%20beating.

[17] Carbohydrate. *Wikipedia.* https://en.wikipedia.org/wiki/Carbohydrate

[18] Fat. *Wikipedia.* https://en.wikipedia.org/wiki/Fat

[19] Protein. *Wikipedia.* https://en.wikipedia.org/wiki/Protein

[20] Calcium. *Wikipedia.* https://en.wikipedia.org/wiki/Calcium

[21] Clothing insulation. *Wikipedia.* https://en.wikipedia.org/wiki/Clothing_insulation

[22] Mean radiant temperature. *Wikipedia.* https://en.wikipedia.org/wiki/Mean_radiant_temperature

[23] Salifou, D-S. (2020, October 19). Quelle est la bonne diététique du cycliste à adopter ? *Le Blog Nutrititon* https://leblognutrition.fr/dietetique-cycliste/#:~:text=Ravitaillement%20en%20glucides%2C%20en%20eau%2C%20en%20solution%20%C3%A9lectrolyte%2C,200%20%C3%A0%20350%20calories%20de%20glucides%20par%20heure

[24] Wind speed. *Wikipedia.* https://en.wikipedia.org/wiki/Wind_speed

[25] MAE et RMSE - Quelle métrique est la meilleure? *Ichi.Pro.* https://ichi.pro/fr/mae-et-rmse-quelle-metrique-est-la-meilleure-252933908062525

[26] Dictionnaire médical de l'Académie de Médecine. (2021). *Académie de Médecine.* http://dictionnaire.academie-medecine.fr/index.php?q=bradypn%C3%A9e

[27] Anatomy and Physiology : Cardiac Output. (2021). *Anatomy and Physiology.* http://www.anaphy.com/cardiac-output/#:~:text=In%20a%20healthy%20adult%20with%2072%20bpm%2C%20cardiac,which%20heart%20pumps%20blood%20and%20its%20maximum%20capacity.

[28] Transpulmonary pressure. (2020). *Wikipedia.* https://fr.xcv.wiki/wiki/Transpulmonary_pressure#:~:text=La%20pression%20transpulmonaire%20est%20la%20diff%C3%A9rence%20entre%20la,P%20tp%20%3D%20P%20alv%20-%20P%20ip.

[29] The Alveoli in Your Lungs. (2018, November 26). *Healthline.* https://www.healthline.com/health/alveoli-function

[30] Vinik, A-I., Erbas, T. (2013, November 11). Diabetic autonomic neuropathy. *Handbook of Clinical Neurology, Volume 117.* 279-294. https://www.sciencedirect.com/science/article/pii/B9780444534910000225

[31] Total Metabolic Rate. (2009, July 23). *Wisdom That Heals : A Blog about Health, Nutrition and Natural Healing.* https://wisdomthatheals.wordpress.com/tag/total-metabolic-rate/

[32] Partial pressure. *Wikipedia.* https://en.wikipedia.org/wiki/Partial_pressure

[33] Anatomie, structure et maladies touchant l'aorte. *Urgence Aorte.* https://www.urgence-aorte.fr/presentation

[34] Vena cava. *Encyclopaedia Britannica.* https://www.britannica.com/science/vena-cava

[35] Aubineau, N. Sodium. *La diététique - Nicolas Aubineau.* https://www.nicolas-aubineau.com/sodium/

[36] Aubineau, N. Alimentation en course. *La diététique - Nicolas Aubineau.* https://www.nicolas-aubineau.com/alimentation-en-course/

[37] Aubineau, N. Boisson énergétique. *La diététique - Nicolas Aubineau.* https://www.nicolas-aubineau.com/boisson-energetique/

[38] Aubineau, N. Dîner de la veille et petit-déjeuner. *La diététique - Nicolas Aubineau.* https://www.nicolas-aubineau.com/diner-petit-dejeuner-avant-course/

[39] Aubineau, N. Boisson de récupération. *La diététique - Nicolas Aubineau.* https://www.nicolas-aubineau.com/boisson-de-recuperation/

[40] What is Sweat Rate – and why does it matter ? (2020, March 6). *Polar.* https://www.polar.com/blog/sweat-rate/

[41] Epinephrine. *The Free Dictionary.* https://medical-dictionary.thefreedictionary.com/epinephrine

[42] Graisse. *Ooreka Santé.* https://alimentation.ooreka.fr/comprendre/graisse

[43] Protéines. *Ooreka Santé.* https://alimentation.ooreka.fr/comprendre/proteines

[44] Calcium. *Ooreka Santé.* https://alimentation.ooreka.fr/astuce/voir/751011/calcium

[45] Sodium. *Ooreka Santé.* https://alimentation.ooreka.fr/astuce/voir/480979/sodium

[46] Skin temperature. *Wikipedia.* https://en.wikipedia.org/wiki/Skin_temperature

[47] Acide acétoacétique : définition, explications. *Aqua Portail.* https://www.aquaportail.com/definition-10066-acide-acetoacetique.html

[48] Analyse de la créatinine. (2015, August). *Passeport Santé.* https://www.passeportsante.net/fr/Maux/analyses-medicales/Fiche.aspx?doc=analyse-creatinine-sang

[49] Buitekant, E. (2019, April 8). Hémoglobine : quel est le taux normal ? *Le Journal des Femmes Santé.* https://sante.journaldesfemmes.fr/fiches-anatomie-et-examens/2511005-hemoglobine-taux-normes-bas-eleve/

[50] Analyse du taux de potassium. (2015, August). *Passeport Santé.* https://www.passeportsante.net/fr/Maux/analyses-medicales/Fiche.aspx?doc=analyse-potassium-sang

[51] Glucose. *Ooreka Santé.* https://diabete.ooreka.fr/comprendre/glucose

[52] Lactate. *Futura Sciences.* https://www.futura-sciences.com/sante/definitions/biologie-lactate-772/

[53] Albumine : causes, symptômes, traitement, diagnostic. *Albumine Info.* http://albumine.info/

[54] Chloride : foods, functions, how much do you need and more. (2021, January 11). *Eufic.* https://www.eufic.org/en/vitamins-and-minerals/article/chloride-foods-functions-how-much-do-you-need-more

[55] Pillou, J-F. (2013, December 6). Globuline - Définition. *Le Journal des Femmes Santé.* https://sante-medecine.journaldesfemmes.fr/faq/19321-globuline-definition

[56] Analyse de l'insuline. (2015, August). *Passeport Santé.* https://www.passeportsante.net/fr/Maux/analyses-medicales/Fiche.aspx?doc=analyse-insuline-sang

[57] Tissue Methodology. (2017-2021). *Pulse Physiology Engine.* https://pulse.kitware.com/_tissue_methodology.html

[58] Endocrine Methodology. (2017-2021). *Pulse Physiology Engine.* https://pulse.kitware.com/_endocrine_methodology.html

[59] Gastrointestinal Methodology. (2017-2021). *Pulse Physiology Engine.* https://pulse.kitware.com/_gastrointestinal_methodology.html

[60] Métabolisme. *Futura Sciences.* https://www.futura-sciences.com/sante/definitions/biologie-metabolisme-781/

[61] Quelle est exactement la différence entre l'advection et la convection ? (2021, February 7). *Tiantan.* https://tiantanstemcell.com/fr/quelle-est-exactement-la-difference-entre-ladvection-et-la-convection/

[62] Diffusion de la matière. *Wikipedia.* https://fr.wikipedia.org/wiki/Diffusion_de_la_mati%C3%A8re

[63] Métabolisme aérobie. *Wikipedia.* https://fr.wikipedia.org/wiki/M%C3%A9tabolisme_a%C3%A9robie

[64] Réactions catalysées. *Mettler Toledo.* https://www.mt.com/fr/fr/home/applications/L1_AutoChem_Applications/L2_ReactionAnalysis/Catalytic-Reactions.html

[65] Homéostasie. *Ooreka Santé.* https://bien-etre.ooreka.fr/astuce/voir/484615/homeostasie

[66] Blanc, J-R. (2020, April 19). Puissance : pourquoi les watts sont importants. *CyclingHeroes.* https://fr.cyclingheroes.com/fr/blog/puissance-watts

[67] Cemosis Website. *Cemosis.* http://www.cemosis.fr/

Appendices

Project (during the year 2021) :

Link of the code of the part : Quantify the level of performance for each activity : `https://github.com/master-csmi/project-csmi-2021-groupamafdj/blob/develop/data/Objectif_1_propre.ipynb`

Link of the code of the part : Find correlations between our calculated performance level and the cyclist feelings : `https://github.com/master-csmi/project-csmi-2021-groupamafdj/blob/develop/data/Objectif_2_projet_1_final.ipynb`

Internship (summer 2021) :

Link of the improvement of the code of work done level : `https://github.com/master-csmi/project-csmi-2021-groupamafdj/blob/develop/data/Stage_V1.ipynb`

Link of the code of the part : Quantify performance indicators and measure correlations : `https://github.com/master-csmi/project-csmi-2021-groupamafdj/blob/develop/data/Stage_ete_2021_partie1_Celine.ipynb`

Link of the code of the Critical power model,W' and Wbal : `https://github.com/master-csmi/project-csmi-2021-groupamafdj/blob/develop/data/Stage_V2_puissance_critique.ipynb`

Code to plot heart rate and exercise intensity of a Pulse simulation and real data & calculation of MAE and MSE errors : `https://github.com/master-csmi/project-csmi-2021-groupamafdj/blob/develop/data/plot_hr_intensity.py`

Code to plot graphs associated with Pulse output data : `https://github.com/master-csmi/project-csmi-2021-groupamafdj/blob/develop/data/plot_output.py`

Code for calculating MAE and MSE errors between Pulse output data of two CSV files : `https://github.com/master-csmi/project-csmi-2021-groupamafdj/blob/develop/data/error.py`

Code for superimposing the heart rate graphs of two CSV files generated by Pulse : `https://github.com/master-csmi/project-csmi-2021-groupamafdj/blob/develop/data/heart_rate_compare.py`