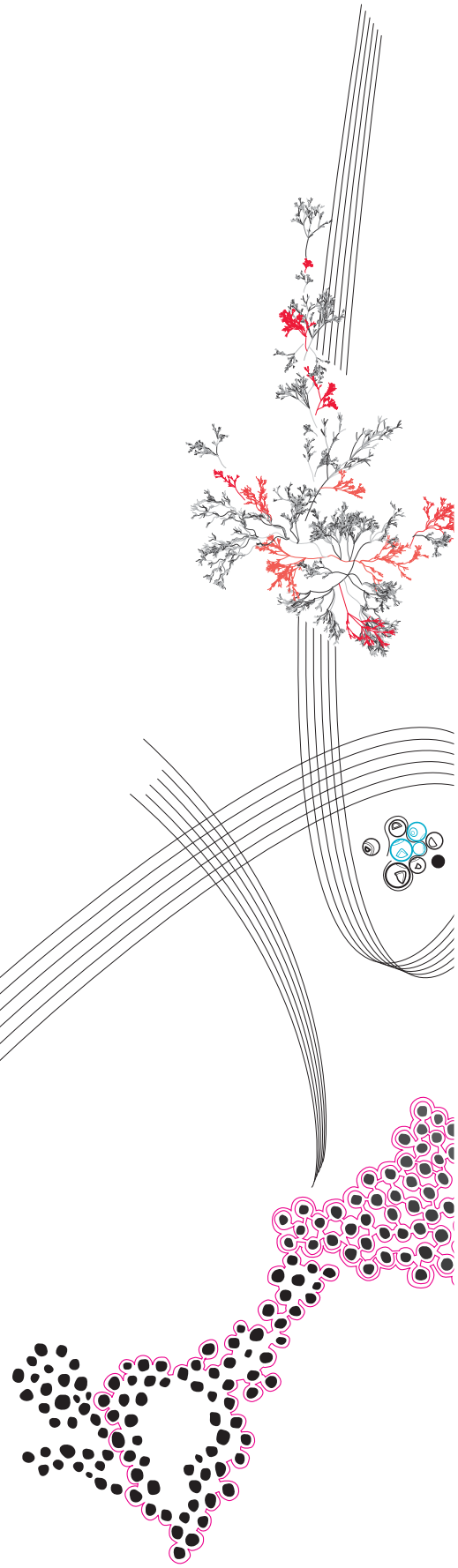


BSc Thesis Applied Mathematics

Detection of changes in movement patterns of runners

Marjolein Bolten

Supervisors: Katharina Proksch & Rupsa Basu

June, 2021

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

Preface

With this article I want to conclude my bachelor Applied Mathematics at the University of Twente. In this article I combined two interests of me, mathematics and running, which made me enjoy working on the bachelor assignment even more.

I want to thank my supervisors, Katharina Proksch and Rupsa Basu, for the weekly meetings with always useful feedback, reading my concept versions, and always willing to answer any questions. I also want to thank all the other members of the Sports, Data and Interaction team for welcoming me during the biweekly meetings and letting me use the data of the data sprint. Last but not least I want to thank all my friends who supported and distracted me during my bachelor assignment.

Contents

Detection of changes in movement patterns of runners

Marjolein Bolten *

June, 2021

Abstract

Introduction: Running is a popular sport, but has also a high injury risk. To prevent injuries, analyzing movement patterns of runners is useful and in this article different statistical methodologies to analyze movement patterns are discussed.
Method: Three statistical methodologies, confidence bands, two-sample t-test and bootstrap based testing are tested on synthetic data and time series data of the knee angle during running. Two groups are used, in a non-fatigued and fatigued state.
Results: The statistical methodology with confidence bands was not able to detect differences in time series of the knee angle while the two-sample t-test and bootstrap based test were.
Conclusion: Both the two-sample t-test and the bootstrap-based test are good tests to measure for a difference in the movement patterns of the knee between a non-fatigue and fatigued runner. The bootstrap-based test is the better one, because this test has the highest power and considers the whole trajectory instead of single points.

Keywords: Bootstrap, Fatigue, Gait cycle, Movement patterns, Statistics

1 Introduction

Practicing a sport is beneficial for one's health and physical fitness, and because running is a low accessible sport, one can easily go outside and run a bit, it is one of the most practiced sports [11]. But it is also a sport with high injury risk. The injury rate for the average recreational runner, one who is steadily training, is between 37% and 56% per year [12]. Of these injuries, the most common one is a knee injury [11].
Knee injuries often occur during the stance phase of a stride. At this part of a stride, the knee absorbs a large portion of the impact forces during the first contact with the ground [1]. The gait cycle of running consists of the stance phase and the swing phase. One cycle is called a stride, which starts when the foot of one leg touches the ground and ends when the same foot touches the ground again. The part where that foot is on the ground is called the stance phase and the part where that foot is 'swinging' forward is the swing phase. The swing phase starts at the moment of toe-off [6].

During running, the joints experience greater stress and more flexion than during walking [5]. One of the differences between the gait cycle of walking and running is that for walking the stance phase is longer than 50% of the stride, so at the beginning and end of the stance phase both legs are on the ground. This is also called the double supported phase. For running the stance phase is shorter than 50% of the stride, so at the beginning and end

*Email: m.m.bolten@student.utwente.nl

of the swing phase, both legs are not on the ground. This is called the double float phase [6]. These differences can be seen in Figure 1.

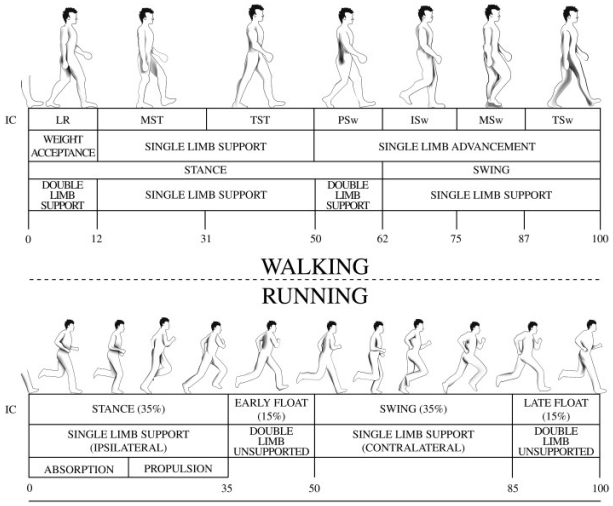


FIGURE 1: The gait cycle of human during walking and running [5].

The angle of the knee (γ) is defined as the angle between the lower leg and the extended upper leg, which varies between $\gamma = 0°$ and $\gamma = 180°$. The flexion of the knee is the movement of the ankle towards the buttocks. Extension of the knee is the opposite movement, that is, moving the ankle away from the buttocks. These terms can be seen in Figure 2.

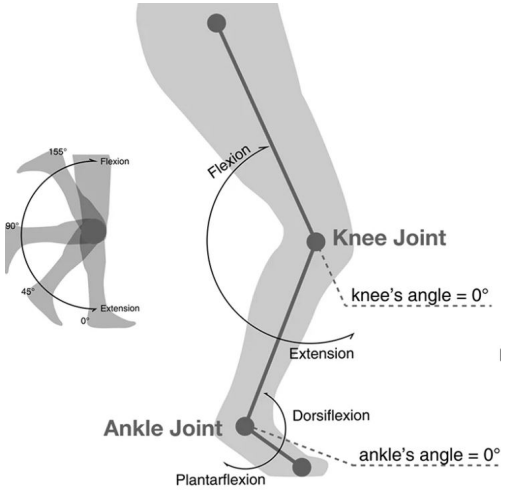


FIGURE 2: Visualisation of the knee angle, flexion, and extension of the knee [3].

The most common cause of injuries is a training error, such as training too much and taking too little rest to recover between trainings [10]. If one takes too little rest, one will start a new training too soon and will reach a fatigued state earlier during this training. The usage of monitoring systems during training has increased in recent years. Watches can monitor your heart rate, pace, estimated time to recover before the next training, and even more [2]. More recently, one started with monitoring movement patterns with the use of sensors on the body. Now things like joint angles during every stride can be monitored [9]. But these movement patterns are functions, while the parameters as for example heart rate, are one dimensional, and can be analyzed easily, this is not the same for movement

patterns. For example, looking at knee angles during a run, one gets a function of the knee angle over time for every stride, so data will consist of sequences of curves.
This information of movement patterns, in this case knee angles, during a run can be interesting to analyze, to prevent injuries in the future. The goal of this article is to establish a statistical methodology that is suitable for testing whether there is a difference in the movement patterns of the knee of a runner in a non-fatigued and fatigued state. Therefore different statistical tests will be evaluated and performed. A look will be taken at the whole trajectory of the knee angle during one stride, but also at specific points in the stride. Finally, there will be a conclusion on the difference in knee angle between the strides of a runner in a non-fatigued and fatigued state.

2 Method

2.1 Data collection

To collect data, there has been a data sprint. During the data sprint, the subject was asked to perform a fatiguing run on 103% of their eight km competition pace for as long as they could. When running on 103% of their competition pace one is certain they will reach a fatigued state. The data sprint took place indoors on a treadmill, to reduce environmental influences, in a controlled situation. During the run every three minutes the subject was asked, on an RPE (Rate of Perceived Exertion) scale of one to ten, how hard the running was. Next to the RPE, several parameters were monitored with sensors on the body and with cameras tracking these sensors. Sensors were placed on the foot, tibia and thigh. The movement of the ankle, knee and hip joints could therefore be measured in the three dimensions (X,Y,Z). One can subtract a lot of information about these measurements, like the time series of the angle of the joints, but also the velocity and acceleration of these joints. For other research the ground reaction force, the force extended by the ground on the body, was also measured via a force plate that was built in the treadmill.

2.2 The model

The following general model is used as a model for the data of the knee angle during a run:

$$Y_i(t) = \mu_i(t) + \epsilon_i(t), \quad i = 1, \ldots, N, \tag{1}$$

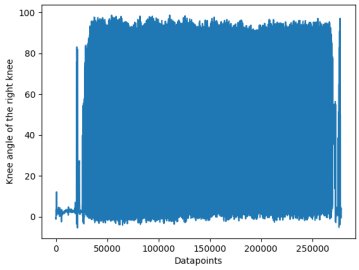
where $t \in [0\%, 100\%]$ denotes the time series of the knee within one stride and i is the number of strides. For each i, the error process $\{\epsilon_i(t) \quad | \quad t \in [0\%, 100\%]\}$ is a Gaussian process which simulates noise. This means in particular that for each fixed pair (i, t) the random variable $\epsilon_i(t)$ is normally distributed.
It is assumed that for $i \neq j$ $\{\epsilon_i(t) \quad | \quad t \in [0\%, 100\%]\}$ and $\{\epsilon_j(t) \quad | \quad t \in [0\%, 100\%]\}$ are independent, i.e. any two strides do not influence each other.

In the rest of this article the following discretized version of model (1) is used where Y_i is observed at s time points on an equidistant grid:

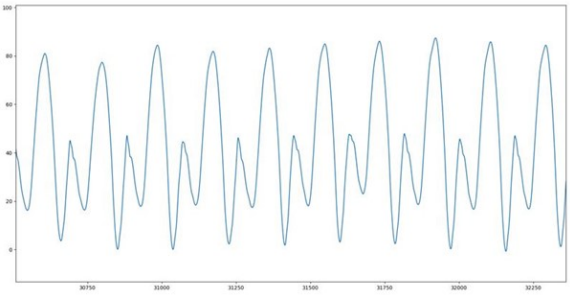
$$Y_i(t_j) = \mu_i(t_j) + \epsilon_i(t_j), \quad i = 1, \ldots, N, \quad j = 1, \ldots, s. \tag{2}$$

2.3 Data processing

The data collection started and ended with three vertical jumps for calibration. The subject was able to run for almost sixteen minutes, and this data can be seen in Figure 3, where in subfigure 3a, all the strides are plotted and in subfigure 3b, the individual strides are visible.



(A) All measurements in the z-axis of the knee angle of the right knee during the data sprint.



(B) A detailed version of the measurements in the z-axis of the knee angle of the right knee during the data sprint.

FIGURE 3: Measurements in the z-axis of the knee angle of the right knee during the data sprint.

From the three dimensions, the data of the sagittal plane (Z-axis) was used for the knee angle during the run. Only the sagittal plane was used, because sagittal motion is the predominant direction of motion. This data of the knee angle per stride was divided into two groups: the non-fatigued group and the fatigued group. For the data in the non-fatigued group the first 60 strides, when running at a constant pace, of the data were used, while for the fatigued group the last 60 strides of the data were used.
Every stride is distributed in $s = 100$ equal intervals between 0% and 100%. That is, for every one % time step in the stride, the knee angle at that moment is known. In terms of model (2), the data in the non-fatigued group can be described as:

$$X_i(t_j) = \mu_1(t_j) + \epsilon_i(t_j), \quad i = 1, \ldots, 60, \quad j = 1, \ldots, 100 \tag{3}$$

and for the fatigued group, the data can be described as:

$$Y_i(t_j) = \mu_2(t_j) + \epsilon_i(t_j), \quad i = 1, \ldots, 60, \quad j = 1, \ldots, 100 \tag{4}$$

To make conclusive statements about whether there is a significance difference between the two groups, statistical testing has been used. Three different tests were used, starting with confidence bands, performing two-sample t-test, and last a bootstrap-based test.

2.4 A statistical test

In this article the definition of a general non-randomized statistical test is as follows:

$$\phi : X \to [0, 1],$$

so that ϕ is the map from the sampling space into the set $[0, 1]$.
The test ϕ in this article will test on the difference of the means of two data sets:

$$H_0 : \mu_1 = \mu_2 \quad \text{vs} \quad H_1 : \mu_1 \neq \mu_2, \tag{5}$$

where H_0 is defined as the null hypothesis and H_1 as the alternative hypothesis.
If the outcome of test ϕ is 1, the test rejects the null hypothesis and if the outcome is 0, ϕ will accept the null hypothesis. Rejecting the null hypothesis means that with the level of the test one can say that the null hypothesis, so the means of two data sets are equal, is not true. When one is accepting the null hypothesis, this does not mean that the means are the same, it just cannot be rejected.

2.5 Confidence bands

The first statistical methodology where a look has been taken at, is the use of confidence intervals for testing on the difference of means of the two groups.
Therefore, for every integer j between 0 and s the sample mean:

$$\overline{X}(t_j) = \frac{1}{N}\sum_{i=1}^{N} X_i(t_j) \quad \text{and} \quad \overline{Y}(t_j) = \frac{1}{N}\sum_{i=1}^{N} Y_i(t_j)$$

and sample variance:

$$\hat{\sigma}_X^2(t_j) = \frac{1}{N-1}\sum_{i=1}^{N}(X_i(t_j) - \overline{X_i}(t_j))^2 \quad \text{and} \tag{6}$$

$$\hat{\sigma}_Y^2(t_j) = \frac{1}{N-1}\sum_{i=1}^{N}(Y_i(t_j) - \overline{Y_i}(t_j))^2 \tag{7}$$

are calculated. Then for every j time step, a confidence interval can be calculated, which is, for the non-fatigued group:

$$[\overline{X}(t_j) - z_{\frac{\alpha}{2}} \times \hat{\sigma}_X(t_j) \quad , \quad \overline{X}(t_j) + z_{\frac{\alpha}{2}} \times \hat{\sigma}_X(t_j)],$$

with $\hat{\sigma}_X(t_j)$ and $\hat{\sigma}_Y(t_j)$ the standard deviations of X and Y respectively. And where $z_{\frac{\alpha}{2}}$ is the $(1-\frac{\alpha}{2})-$ quantile of the standard normal distribution for a confidence level of $(1-\alpha)\times 100\%$. This means that $(1-\alpha)\times 100\%$ of the values lie within this confidence interval, $\frac{\alpha}{2}$ has been used because it is a two sided distribution so that on both sides a maximum of $\frac{\alpha}{2}\times 100\%$ of the values do not lie in the interval.
For $\alpha = 0.05$, the confidence level will be $(1-\alpha)\times 100\% = 95\%$ and the $\frac{\alpha}{2}-$ quantile is given by 1.96 [8]. The confidence interval $(I_X(t_j))$ for every $j \in [0, s]$, of the non-fatigued group, will thus be:

$$I_X(t_j) := [\overline{X}(t_j) - 1.96 \times \sigma_X(t_j) \quad , \quad \overline{X}(t_j) + 1.96 \times \sigma_X(t_j)]$$

The same has been done for the fatigued group, but with $\overline{Y}(t_j)$ and $\hat{\sigma}_Y(t_j)$ instead of $\overline{X}(t_j)$ and $\hat{\sigma}_X(t_j)$.

The statistical test for the hypotheses stated in (5) will be:

$$\phi_1 = \begin{cases} 1 & \text{if } I_X(t_j) \cap I_Y(t_j) = \emptyset \\ 0 & \text{else} \end{cases},$$

where $t_j \in [0\%, 100\%]$ is a fixed point. ϕ_1 will thus reject if the two confidence intervals of the different groups have no overlap at point t_j. In this case the test ϕ_1 will depend on the choice of t_j. That is, if $\mu_1(t_j) = \mu_2(t_j)$, ϕ_1 will not reject the test, although there could be a point $t_j \neq t_k$ such that $\mu_1(t_k) \neq \mu_2(t_k)$. Thus there should be looked at confidence intervals for the whole trajectory instead of focusing at one point.

2.5.1 Multiplicity correction for the confidence intervals

To construct the confidence bands for the whole trajectory, one needs all the confidence intervals for j between 0 and s. Now a test can be constructed where the whole trajectory is taken into account.
As we test for all $j \in [0, s]$ simultaneously it is now a multiple testing problem. There are two different kinds of multiple testing, the normal multiple test and the combined test. For the combined test, multiple tests are applied to draw one conclusion, reject or accept the null hypothesis. While for the normal multiple test, multiple tests are applied to draw multiple conclusions, for each test there is an individual conclusion [8].
The statistical methodology using confidence bands will be a combined test, one is looking for one conclusion after performing the tests. For a single test, the level $\alpha = 0.05$ means that if the test is performed 100 times, a maximum of 5% of those tests will falsely reject the null hypothesis. For the combined test one wants the level of the combined test to be $\alpha = 0.05$ so that the number of false rejections is controlled. Therefore a simple Bonferroni correction is used to the single confidence intervals described above. The following Bonferroni correction is used:

$$\alpha_i = \frac{\alpha}{s}, \quad i = 1, 2, \ldots, s. \tag{8}$$

In this way the Bonferroni correction adjusts the probability values because, with multiple statistical tests there is an increased risk of a type I error. And with this Bonferroni correction the risk of a type I error is controlled again by α.
Taking the multiplicity correction into account the confidence band is constructed with the following confidence intervals for each $j \in [0, s]$:

$$[\overline{X}(t_j) - 3.84 \times \sigma_X(t_j) \quad , \quad \overline{X}(t_j) + 3.84 \times \sigma_X(t_j)] \quad \text{and} \tag{9}$$

$$[\overline{Y}(t_j) - 3.84 \times \sigma_Y(t_j) \quad , \quad \overline{Y}(t_j) + 3.84 \times \sigma_Y(t_j)], \tag{10}$$

where 3.84 is the $z_{\frac{\alpha_i}{2}}$ quantile of the standard normal distribution [8].
When there is a part in the confidence bands where the two bands of (9) and (10) do not overlap, one can say that the $\mu_1(t)$ and $\mu_2(t)$ are significantly different [4]. The statistical test ϕ_2 testing on the hypothesis stated in (5) will thus be:

$$\phi_2 = \begin{cases} 1 & \text{if the confidence intervals (9) and (10) are disjoint for at least one } t_j \\ 0 & \text{else} \end{cases},$$

2.6 Two-sample t-test

The second methodology that has been looked at is the two-sample t-test, the two-sample t-test can be used to test whether there is a difference in the means of two data sets from a normal distribution. The following t-test is performed s times and a vector is created which shows the rejected points.
For the knee angles as time series, the data is described as in equations (3) and (4). The following hypotheses will be tested for every $j \in [0, s]$:

$$H_{0_j} : \mu_1(t_j) = \mu_2(t_j) \quad \text{vs} \quad H_{1_j} : \mu_1(t_j) \neq \mu_2(t_j). \tag{11}$$

The test statistic of the test will be:

$$T(t_j) = \frac{\overline{X}(t_j) - \overline{Y}(t_j)}{\sqrt{S^2(t_j)(\frac{1}{n} + \frac{1}{m})}} \quad \text{with}$$

$$S^2(t_j) = \frac{n-1}{n+m-2} \times \hat{\sigma}_X^2(t_j) + \frac{m-1}{n+m-2} \times \hat{\sigma}_Y^2(t_j),$$

with $\hat{\sigma}_X, \hat{\sigma}_Y$ the standard deviations of $X(t), Y(t)$ as described in equation (6) and (7), and n, m the number of data points in $X(t)$ and $Y(t)$, respectively. $\overline{X}(t_j)$ is the mean of the j^{th} time step of all the strides in the non-fatigued group and $\overline{Y}(t_j)$ the mean of the j^{th} time step of all the strides in the fatigued group.
The test statistic has a t_{n+m-2} distribution under H_0 and a test ϕ_3 for the hypotheses in (11) is given by:

$$\phi_3 = \begin{cases} 1, & \text{if } \mid T(t_j) \mid > t_{\frac{\alpha}{2}} \\ 0, & \text{else} \end{cases}$$

2.6.1 Multiplicity correction for the two-sample t-test

To construct a combined test of these j tests, the same Bonferroni correction as in equation (8) is used for those t-tests. The combined test will then test the following:

$$H_{0,\text{global}} = \cap_{j=1}^{N} H_{0_j} \quad \text{with}$$
$$H_{0_j} : \mu_1(t_j) = \mu_2(t_j) \quad \forall j \quad \text{vs} \quad H_{1_j} : \mu_1(t_j) \neq \mu_2(t_j) \quad \text{for at least one} \quad j,$$

where $H_{0,\text{global}}$ is the global null hypothesis.
The two-sample t-test thus rejects a point if the p-value of the t-test at time point j is smaller than α_i. If at least one point is rejected, this means that the global null hypothesis, will not hold [8].
Formally, the statistical test ϕ_4 can be defined as follows:

$$\phi_4 = \begin{cases} 1, & \text{if } \mid T(t_j) \mid > t_{\frac{\alpha_i}{2}} \quad \text{for at least one } t_j \\ 0, & \text{else} \end{cases}$$

2.7 A bootstrap test for whole trajectories

The methods discussed in sections 2.5 and 2.6 are based on point-wise confidence intervals and t-tests, respectively. In order to obtain simultaneous confidence bands and a combined test, a Bonferroni correction was used. The true confidence level of the Bonferroni confidence bands is typically much higher than the normal confidence level of 95% and the true level of the combined test is typically much lower than the anticipated 5%. As a consequence the Bonferroni corrected methods will not be able to detect small changes between the mean trajectories of the different groups. A test which takes the mean trajectories as a whole into account can do this.

For this third methodology it is assumed that there is a collection of random functions, where one function is one stride, satisfying

$$X_{i,j}(t) = \mu_i(t) + \epsilon_{i,j}(t), \quad i = 1, 2, ..., K, \quad j = 1, 2, ..., n_i, \quad t \in I$$

with $K = 2$ the number of groups and n_i the number of observations (strides) in group i. The following hypotheses are tested

$$H_0 : \mu_1 = \mu_2 \quad \text{vs.} \quad H_1 : \mu_1 \neq \mu_2. \tag{12}$$

The difference between the hypotheses for this test (12) and the one for the two-sample t-test (11) is that in (12) the mean functions μ_1 and μ_2 are considered as a whole, while

in (11), the points are considered individually.
It is assumed that μ_1 and μ_2 are square integrable, so:

$$\int_I |\mu_i(t)|^2 \mathrm{dt} < \infty.$$

Because this assumption holds true, it is reasonable to base a test on the Euclidean distance of the mean trajectories of the two groups. The test statistic S_N for the hypotheses described in (12) is given by:

$$S_N = \frac{n_1 n_2}{N} ||\overline{X}_{1,n_1} - \overline{X}_{2,n_2}||^2, \tag{13}$$

where $\overline{X}_{1,n_1}$ and $\overline{X}_{2,n_2}$ are the mean trajectories of the two groups, and $N = n_1 + n_2$.

The statistical test ϕ_5 is defined as:

$$\phi_5 = \begin{cases} 1, & \text{if } S_N > C_\alpha \\ 0, & \text{else} \end{cases},$$

where C_α is the (1- α)-quantile of the distribution of S_N.

Unlike the two-sample t-test, the distribution of S_N under the null hypothesis is unknown. Therefore, the critical value is estimated by using a bootstrap resampling method.
Bootstrapping is the process where from a small data set one can infer the population of a data set. One randomly selects a data point x times and adds this to a sample. From this sample, all kinds of parameters can be estimated such as the mean and standard deviation. If this process is repeated many times, say 1000 times, the estimated parameters will be close to the real parameters. Bootstrap is a good method to test whether the sample of data points is a good representation of the original data set.

The above description of the Bootstrap process works for data sets containing points and not functions. The stride during running is a function, so the process should be adjusted a little. The following bootstrap-based test for functions was proposed in [7] and is used to test for a difference in the mean trajectories. The bootstrap test for functions is performed as follows:

Step 1: Calculate the sample mean functions in each population

$$\overline{X}_{i,n_i}(t) = \frac{1}{n_i} \sum_{j=1}^{n_i} X_{i,j}(t), \quad t \in I, \quad i \in 1, 2$$

Step 2: Calculate the residual functions in each population

$$\epsilon_{i,j}(t) = X_{i,j} - \overline{X}_{i,n_i}(t), \quad t \in I, \quad j = 1, 2, ..., n_i$$

Step 3: Generate bootstrap functional pseudo-observations $X^+_{i,j}(t)$, according to

$$X^+_{i,j}(t) = \overline{X}_N(t) + \epsilon^+_{i,j}(t), \quad t \in I, \tag{14}$$

with $\overline{X}_N(t)$ the pooled mean function estimator, given as follows:

$$\overline{X}_N = \frac{1}{N} \sum_{i=1}^{K} \sum_{j=1}^{n_i} X_{i,j}(t), \quad t \in I$$

and $\epsilon^+_{i,j}(t) = \epsilon_{i,J}(t)$ where J is a random integer in the range of $1, 2, ..., n_i$.

Step 4: Calculate the test statistic S^+_N.

This test statistic is the same as S_N of (13) but calculated using the bootstrap functional pseudo observations of (14). The test statistic S^+_N can be found by applying the bootstrap process to steps one, two, and three. Then for every bootstrap sample, the test statistic S^+_N is calculated, and one makes a list with all those values to eventually find the bootstrap critical value (C^+_a). For a level α of 5%, one can find the critical value by taking the 95^{th} percentile of the list of sorted S^+_N values.
With the Euclidean norm, the distance between the two sample mean functions $\overline{X}_{1,n_1}$ and $\overline{X}_{2,n_2}$ has been used. By doing this, one makes sure to test over the whole trajectory and not only at one point. Therefore the use of the Euclidean norm is useful for the bootstrap test.

Step 5: Reject the null hypothesis H_0 if and only if

$$S_N > C^+_\alpha$$

This will mean that the test will reject if the test statistic is greater than 95% of the bootstrapped test statistics. Formally, the test ϕ_6, testing on the hypotheses stated in (12) is defined as:

$$\phi_6 = \begin{cases} 1, & \text{if } S_N > C^+_\alpha \\ 0, & \text{else} \end{cases}$$

In [7] it is shown that as the sample sizes n_1 and n_2 increase, the bootstrap critical value C^+_α converges to the 0.95 - quantile C_α of the distribution of S_N and therefore, the use of C^+_α instead of C_α is justified.

2.8 Power of the tests

The power of a test is the probability that a test rejects the null hypothesis when the alternative hypothesis is true. So the probability that the test correctly rejects the null hypothesis. The power of a test ranges between 0 and 1, where a power of 1 means that the test always rejects the null hypothesis, when it is indeed wrong.
The probability of making a type II error is defined as:

$$\beta = 1 - E_{H_1}[\phi(X)],$$

where $\phi(X)$ is the test as defined in equation (5) and the power of the test is thus defined as:

$$1 - \beta.$$

When increasing the power of the test, this also means that there will be more type I errors, because it is easier to reject the null hypothesis. So it is important to find a balance here, to find the most optimal test. A visualization of type I or II errors can be seen in Table 1.

TABLE 1: Definition of type I and II errors.

	Test rejects	Test accepts
H_0 true	Type I error	v
H_0 false	v	Type II error

In order to determine which of the tests ϕ_2, ϕ_4 and ϕ_6 has the highest power, a simulation study is performed in section 3.2.

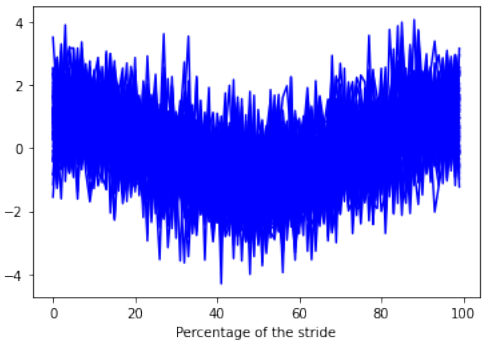
3 Results

3.1 Validity of the tests

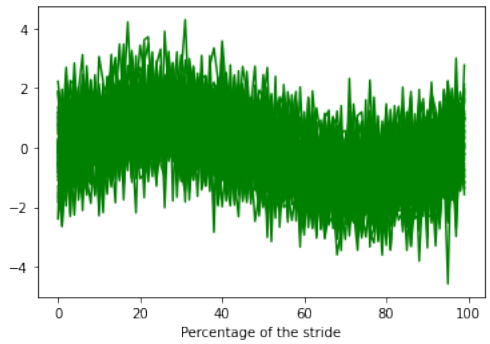
Before performing the three statistical methodologies on the data from the data sprint, first, the tests were run on simulated data. This is done to test the validity of the tests. Therefore, simulated data from a known distribution is used. The stride of a human looks like a cosine, so for the simulated data cosines with added noise ($\epsilon(t_j)$) are used. There were tests performed in two general situations, the first one with both groups distributed as a cosine with random normal noise added, so $X_1(t_j)$ versus $X_2(t_j)$. The second one with group 1, cosine, and group 2 sine with both random normal noise added, so $X_1(t_j)$ versus $Y_2(t_j)$. With $X_i(t_j)$ and $Y_i(t_j)$ defined as follows:

$$X_i(t_j) = \cos(t_j) + \epsilon(t_j),$$
$$Y_i(t_j) = \sin(t_j) + \epsilon(t_j),$$
$$\epsilon(t_j) \sim N(0,1) \quad \text{i.i.d. (independent and identically distributed)}$$

This data can be seen in Figure 4.



(A) A cosine with random normal $\sim$ (0,1) noise added



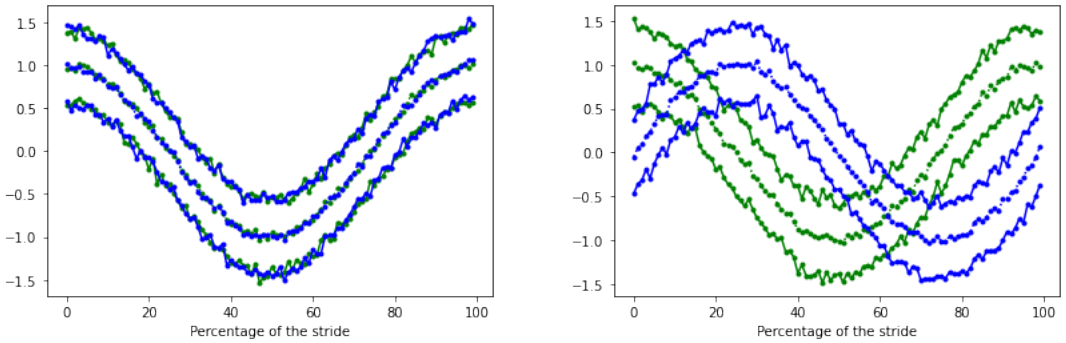
(B) A sine with random normal $\sim$ (0,1) noise added

FIGURE 4: The synthetic data

The tests are validated if the level of the test is controlled by $\alpha = 0.05$ for both scenarios. In the scenario of the two cosines, for which we know the mean trajectories are the same, this means that there is a maximum of 5% of false rejections. For the cosine and sine, we know the mean trajectories are not the same, so the tests should reject.

3.1.1 Confidence bands

The confidence bands are made of all the confidence intervals and are together with the mean trajectory plotted in Figure 5. As can be seen in subfigure 5a there is not a part in the confidence bands where the two confidence bands are not overlapping, for the data sets $((X_1(t_j) \quad \text{and} \quad X_2(t_j)))$, which means that the null hypothesis cannot be rejected. If the confidence bands are constructed one hundred times, there will not be a single time that there is no overlap between the bands, and thus the statistical methodology will reject the null hypothesis. The level of the statistical methodology is thus controlled by α.
In subfigure 5a one can see that there are multiple places where there is no overlap between the two confidence bands. Therefore, one can directly say that the mean trajectories of the two data sets $(X_1(t_j) \quad \text{and} \quad Y_2(t_j))$ are not the same and so the null hypothesis can be rejected. If the confidence bands are constructed 100 times there will always be a part where the confidence bands are not overlapping, thus it is also controlled by α.



(A) Cosines with random normal $\sim$ (0,1) noise added.

(B) Cosine and sine with random normal $\sim$ (0,1) noise added.

FIGURE 5: Confidence bands for the simulated data.

For both scenarios the outcome of the statistical methodology is as expected and the level of the statistical methodology is controlled by $\alpha = 0.05$ so the validity of the statistical methodology with confidence bands is tested. This methodology can thus be used to test the running data for differences in the time series of the knee angles.

3.1.2 Two-sample t-test

For the two-sample t-test with Bonferroni adjustment, the global null hypothesis can be rejected if at least one of the individual tests rejects. If at one point the means of the two groups are not the same, then the mean trajectory of the two groups is also not the same. For the two cosines all the single tests do not reject, so the combined test also does not reject the null hypothesis. When performing this combined test 100 times, there are some false rejections, but those are always less than five. This means that the level of the two-sample t-test is controlled by $\alpha = 0.05$. For the cosine and the sine, there is always at least one point that rejects, so the combined test always rejects, as expected.

For both scenarios the outcome of the test is as expected and the level of the test is controlled by $\alpha = 0.05$ so the validity of the two-sample t-test is tested. This methodology can thus be used to test the running data for differences in the time series of the knee angles.

3.1.3 Bootstrap

The results of the Bootstrap-based test for both scenarios on the simulated data for one example can be found in Table 2 in the first three columns. To test the validity of the bootstrap based methodology the test is performed one hundred times and the percentage of rejections in both scenarios is given in column 5 of Table 2. As can be seen for both scenarios the bootstrap test rejected if this was expected and did not reject if this was not expected. Also, the percentage of false rejections was controlled by the given level $\alpha = 0.05$.

TABLE 2: Results of the Bootstrap-based test on the simulated data.

	Test statistic S_N	Critical value	Rejected	% of rejections if performing it 100 times
$X_1(t_j), X_2(t_j)$	5.29	6.55	no	4 %
$X_1(t_j), Y_2(t_j)$	2994.25	6.53	yes	100 %

The bootstrap-based test is validated to be used as a statistical test to analyze the time series of the knee angles during a run.

3.2 Power of the tests

The power of a test will be higher if this test has a higher probability of rejecting the null hypothesis when it is actually wrong. So in the case of testing on the means of trajectories, rejecting the null hypothesis when there is actually a difference in the means of two trajectories. To estimate which statistical methodology has the highest power, a lot of tests have been performed on the following two groups:

$$
\begin{aligned}
&X_i(t_j) = \cos(t_j) + \epsilon(t_j),\\
&X_i(t_j) = \cos(t_j - \delta) + \epsilon(t_j),\\
&\epsilon(t_j) \sim N(0,1) \quad \text{i.i.d.},
\end{aligned}
$$

with $\delta \in [0, \frac{\pi}{2}]$ such that for $\delta = 0$ and $\delta = \frac{\pi}{2}$ the test is the same as in the two scenarios described above (cosine and cosine, and cosine and sine). The results of these tests can be seen in Table 3.

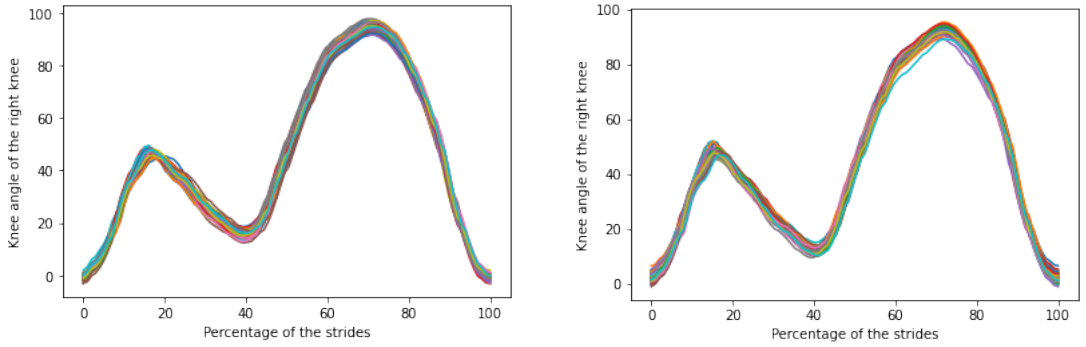
As one can see in Table 3, the Bootstrap-based test has the most rejections in cases where there is a small difference between the trajectories. Because it has the most rejections and we know that the means are indeed different, it means that it rejects the null hypothesis the most while it is indeed wrong, so this statistical methodology has the highest power.

TABLE 3: The percentage of rejections of the statistical tests for different values of δ.

δ	% rejections confidence bands	% rejections t-test	% rejections S_N
0	0	4	4
$\frac{1\pi}{200}$	0	7	8
$\frac{2\pi}{200}$	0	18	46
$\frac{3\pi}{200}$	0	34	96
$\frac{4\pi}{200}$	0	62	100
$\frac{5\pi}{200}$	0	86	100
$\frac{6\pi}{200}$	0	100	100
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\frac{3\pi}{10}$	0	100	100
$\frac{4\pi}{10}$	11	100	100
$\frac{1\pi}{2}$	96	100	100

3.3 The data

Once the data was preprocessed the knee angles were plotted as can be seen in figure 6. At a first glance, one cannot see if there is a difference between the mean trajectories of the first 60 and the last 60 strides. It can be seen that at around $15-20\%$ of a stride, the knee angle during the last 60 strides has a more pronounced peak than during the first 60 strides. This is the moment in the human gait when the knee is flexed the most in the stance phase.
Because of this difference that can be seen at first glance, next to the tests on the whole trajectory, a look has also been taken at this specific interval during the human stride.

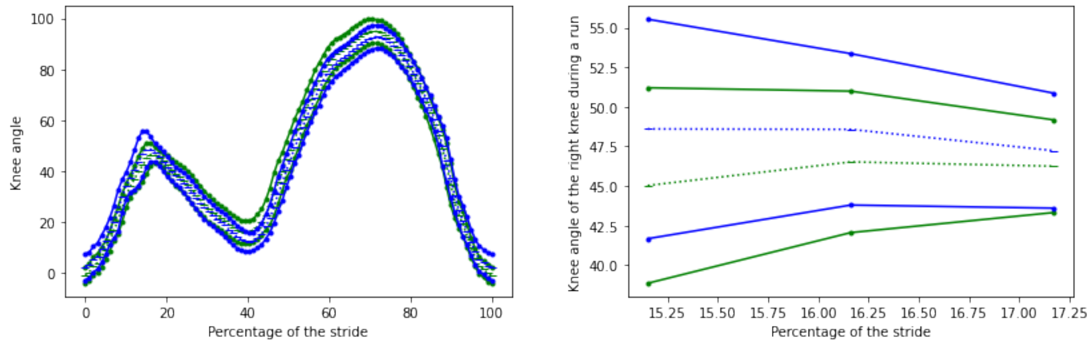


(A) The knee angle during the first 60 strides. (B) The knee angle during the last 60 strides.

FIGURE 6: Knee angles of the subject when performing the fatiguing run.

3.4 Confidence bands

The results of the statistical methodology with the confidence bands can be seen in Figure 7. One can see in subfigure 7a that there is no point during the stride that the confidence bands of the first 60 and last 60 strides do not overlap. So for this statistical methodology, the null hypothesis cannot be rejected, and one cannot say anything about a difference in means of the first 60 and last 60 strides.



(A) The confidence bands of the first 60 strides (green) and the last 60 strides (blue) during a fatiguing run.

(B) The confidence bands of the specific region of the first 60 strides (green) and the last 60 strides (blue) during a fatiguing run.

FIGURE 7: Knee angles of the right knee of the subject when performing the fatiguing run.

For the specific region as can be seen in subfigure 7b, between 15% and 17% of a stride, there is overlap between the green and blue confidence bands, so the null hypothesis is accepted, and one can not say anything about a difference in means in that region for the first 60 and last 60 strides of a fatiguing run.

3.5 Two-sample t-test

The outcome of the multiple two-sample t-test is the following vector with 1 if the t-test rejected at that point:
Rejected points $= [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$.

This vector is visualized in Figure 8, where a red dot indicate that the t-test rejects the null hypothesis at that point and a black dot that the t-test accepts the null hypothesis.

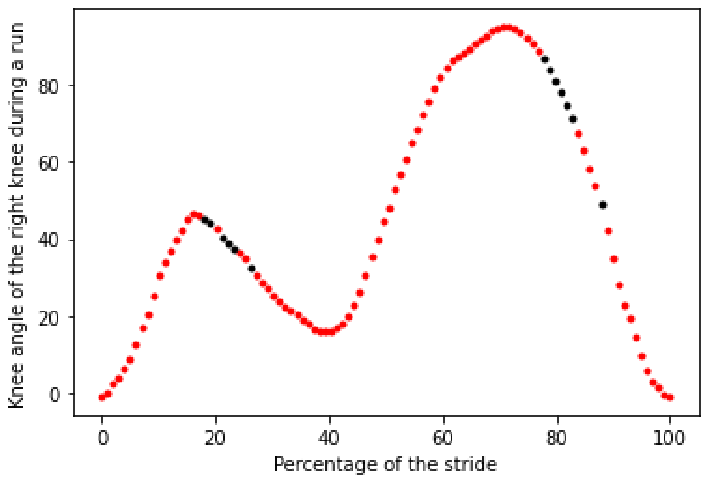


FIGURE 8: In red the points that reject and in black the points that accept the null hypothesis with the two-sample t-test.

Because there is at least one point that is rejected, the combined test also rejects, and thus the global null hypothesis, the mean trajectory of the first and last 60 strides during a fatiguing run is the same, can be rejected with a confidence level of 95%.

At the interval, $15-17\%$, of a stride, the two-sample t-test performed without Bonferroni level adjustment gives as an outcome the following vector:
Rejected points $= [1, 1, 1]$.

So, one can conclude with a level of 95% that at both, the whole trajectory as the specific interval, there is indeed a difference in means of the first 60 strides and the last 60 strides of a fatiguing run.

3.6 Bootstrap

The test statistic S_N and the critical value of the bootstrap-based test can be found in Table 4. The test statistic is greater than the critical value which means that the null hypothesis can be rejected. For the specific region, around $15\% - 17\%$ of the stride, the test statistic is also greater than the critical value.

TABLE 4: Results of the Bootstrap-based test on the data of the knee angle during a run.

	Test statistic S_N	Critical value	Rejected
Time series of the knee (whole trajectory)	24843.21	586.11	yes
Time series of the knee at $15\% - 17\%$	1082.69	111.62	yes

One can conclude with a confidence level of 95%, based on the bootstrap-based test, that there is a difference in the means of the first 60 strides and last 60 strides of a fatiguing run for the whole trajectory and for the specific region.

4 Discussion & Conclusion

In this section, the results are discussed and reflected upon and finally, recommendations will be given for future research. The aim of this study was to establish a statistical methodology that is suitable for testing whether there is a difference in the movement patterns of the knee between a runner in a non-fatigued and fatigued state. Therefore three different statistical methodologies have been compared and each test was first checked by running it with simulated data.

For the simulated data, the cosines and sines, all of the tests gave the expected results and were controlled for a level of $\alpha = 0.05$. With the test with confidence bands and the two-sample t-test, a look has been taken at specific points and at least one rejection means that the whole trajectory is not the same. On the other hand, for the bootstrap-based test there is directly tested at the means of the whole trajectories.

For the cosine and sine, all of the three tests rejected the null hypothesis, which means that the mean trajectories are the same. For the time series of the knee angles during a run, only the two-sample t-test and the bootstrap-based test rejected the null hypothesis.

Both the two-sample t-test and the bootstrap-based test are good tests to test for a difference in the movement patterns of the knee between a runner in a non-fatigued and fatigued state. The advantage of the two-sample t-test is that one can more easily see where the mean trajectories of the two groups differ the most, because the p-value of the two-sample t-test will be the lowest there.
The disadvantage of the two-sample t-test is that it is a combined test, because of all the testing on single points. And with the simple Bonferroni correction used, the power of the combined test is a lot smaller than for the single tests.

The simulated data was created with adding i.i.d. noise, this noise is what could be happen in the worst case scenario. The real data was a lot smoother than cosines with i.i.d. noise, so it could be interesting to also test the statistical methodologies with simulated data that is smoother. It can be expected that, when looking at the power of the statistical methodologies with this simulated data with smooth noise, the overall conclusion will stay the same, i.e. the bootstrap based test is best suitable. Still, it could be interesting to look how the power will differ when the data is becoming smoother.

All these tests were performed on a data set obtained in a controlled situation: indoor treadmill running. The variability of this data is therefore smaller than it would be while outdoor running. The confidence bands of a data set with higher variability will be wider than the confidence bands of a data set with a smaller variability, so the statistical methodology with confidence bands is also not suitable for uncontrolled situations like outdoor running.
For the two-sample t-test and the bootstrap based test, nothing can be said about how those statistical methodologies will behave when testing on data with more variability, so this could be interesting for further research.

Due to time limitations the validity and power of the statistical methodologies is estimated while performing a test 100 times. This is a good estimation, but for further research it can be interesting to actually calculate the power instead of estimating it.

Next to this, the methodologies were only tested on one dataset of one runner instead on a dataset of multiple runners. The dataset of the runner did not contain weird movement patterns, so the methodologies probably also work for other runners, but that's not certain. So in future research the methodologies, and especially the bootstrap-based test should be tested on samples of multiple different runners.

Overall, one can say that the bootstrap-based test is a good statistical methodology to test whether there is a difference in the knee angle of a runner in a non-fatigued or fatigued state. In this article, a look has only been taken at the movement patterns of the knee during a run, but for further research, the bootstrap-based test should be checked for other movement patterns of joints, and eventually of multiple joints together.

References

[1] John P Abt, Timothy C Sell, Yungchien Chu, Mita Lovalekar, Ray G Burdett, and Scott M Lephart. Running Kinematics and Shock Absorption Do Not Change After Brief Exhaustive Running. *Journal of Strength and Conditioning Research*, 25(6):1479–1485, June 2011.

[2] André Henriksen, Martin Haugen Mikalsen, Ashenafi Zebene Woldaregay, Miroslav Muzny, Gunnar Hartvigsen, Laila Arnesdatter Hopstock, and Sameline Grimsgaard. Using Fitness Trackers and Smartwatches to Measure Physical Activity in Research: Analysis of Consumer Wrist-Worn Wearables. *Journal of Medical Internet Research*, 20(3):e110, March 2018.

[3] Jong Hyun Kim and Byeong Hee Won. Kinematic on Ankle and Knee Joint of Post-Stroke Elderly Patients by Wearing Newly Elastic Band-Type Ankle–Foot Orthosis in Gait. *Clinical Interventions in Aging*, Volume 14:2097–2104, December 2019.

[4] Mark W. Lenhoff, Thomas J. Santner, James C. Otis, Margaret G.E. Peterson, Brian J. Williams, and Sherry I. Backus. Bootstrap prediction and confidence bands: a superior statistical method for analysis of gait data. *Gait & Posture*, 9(1):10–17, March 1999.

[5] Everett B. Lohman, Kanikkai Steni Balan Sackiriyas, and R. Wesley Swen. A comparison of the spatiotemporal parameters, kinematics, and biomechanics between shod, unshod, and minimally supported running as compared to walking. *Physical Therapy in Sport*, 12(4):151–163, November 2011.

[6] Tom F Novacheck. The biomechanics of running. *Gait & Posture*, 7(1):77–95, January 1998.

[7] Efstathios Paparoditis and Theofanis Sapatinas. Bootstrap-Based K-Sample Testing For Functional Data. *arXiv:1409.4317 [math, stat]*, September 2016. arXiv: 1409.4317.

[8] K. Proksch. Simultaneous statistical inference. *Lecture notes*, 2021.

[9] C. Strohrmann, H. Harms, C. Kappeler-Setz, and G. Troster. Monitoring Kinematic Changes With Fatigue in Running Using Body-Worn Sensors. *IEEE Transactions on Information Technology in Biomedicine*, 16(5):983–990, September 2012.

[10] University of Primorska, Faculty of Health Sciences, Izola, Slovenia, Žiga Kozinc, Nejc Sarabon, and University of Primorska, Faculty of Health Sciences, Izola, Slovenia. Common Running Overuse Injuries and Prevention. *Montenegrin Journal of Sports Science and Medicine*, 6(2):67–74, September 2017.

[11] R N van Gent, D Siem, M van Middelkoop, A G van Os, S M A Bierma-Zeinstra, B W Koes, and J. E Taunton. Incidence and determinants of lower extremity running injuries in long distance runners: a systematic review * COMMENTARY. *British Journal of Sports Medicine*, 41(8):469–480, March 2007.

[12] Willem van Mechelen. Running Injuries: A Review of the Epidemiological Literature. *Sports Medicine*, 14(5):320–335, November 1992.

A Table of symbols

TABLE 5: Table with symbols used in the article.

Symbol	Description	Page number
γ	angle of the knee	2
t_j	time point t, $j \in [0, s]$	3
s	amount of intervals between begin and end point, $s = 100$	3
N	number of strides in the group	3
i	stride i, $i \in [0, N]$	3
$X_i(t_j)$	data in group non-fatigue	4
$Y_i(t_j)$	data in group fatigue	4
$\overline{X}(t_j)$	sample mean of $X_i(t_j)$	5
$\overline{Y}(t_j)$	sample mean of $Y_i(t_j)$	5
$\hat{\sigma}_X(t_j)$	standard deviation of $X_i(t_j)$	5
$\hat{\sigma}_Y(t_j)$	standard deviation of $Y_i(t_j)$	5
α	significance level	5
α_i	Bonferroni correction of α	6
$T(t_j)$	test statistic of two-sample t-test	6
$S^2(t_j)$	pooled sample variance	6
K	number of groups	7
S_N	test statistic of bootstrap-based test	8
$\overline{X_{i,n_i}}(t_j)$	sample mean functions	8
x	amount of data points to bootstrap	8
$\epsilon_{i,j}(t)$	residual function	8
$X^+_{i,j}(t)$	bootstrap functional psuedo-observations	8
$\overline{X}_N$	pooled mean function estimator	8
J	random integer between 1 and N	9
S^+_N	test statistic of pseudo observations	9
C^+_α	bootstrap critical value	9
C_α	critical value	9
β	probability of making a type II error	9
δ	variable running between 0 and $\frac{\pi}{2}$	12

B Code

Listing 1: Code for running with simulated data.

```
import math
import numpy as np
import matplotlib.pyplot as plt
from sklearn.utils import resample
import csv
from scipy import stats
import random
from math import sqrt

s = 100 #amount of steps between -2pi and 2pi
b = 60 #amount of functions (strides) in the original (input) data #equal for all groups
B = 1000 #amount of Bootstrap iterations for the statistical testing (TEST 3)
ni = 60 #amount of Xij's in group 1 (TEST 3)
mi = 60 #amount of Xij's in group 2 (TEST 3)
k = 2 #amount of groups
N = ni + mi
SnB = []  #list of test statistics for each bootstrap

Percentage_array = []
for i in range(100):
  Percentage_array.append(i)

#Creating the input data, in the example the cosine / sine
in_array = np.linspace(-(0 * np.pi), 2 * np.pi, s)

#Creating a data set with ni functions (Xij) per group
vectorX = []
vectorY = []
c = 20  #Bootstrap samples
for o in range(ni):
    # For each new sample we take c trajectories and calculate the mean trajectory, that is the new sample
    X = []
    Y = []
    # make cosines with noise added between -2pi and 2pi
    for i in range(c):
        out_array_i_X = []
        out_array_i_Y = []
        for j in range(s):
            out_array_i_X.append(math.cos(in_array[j]) + np.random.normal(0, 1))
            out_array_i_Y.append(math.cos(in_array[j]) + np.random.normal(0, 1))
        X.append(out_array_i_X)
        Y.append(out_array_i_Y)

    # Make vectors with all points with the same input value in one vector, so s vectors
    vectorX_k = []
    vectorY_k = []
    for k in range(s):
        point_X_k = []
        point_Y_k = []
        for l in range(c):
            point_X_k.append(X[l][k])
```

```
            point_Y_k.append(Y[l][k])
        vectorX_k.append(point_X_k)
        vectorY_k.append(point_Y_k)

    # Calculating mean of every vector created above
    meanX_o = []
    bootstrapX = []
    meanY_o = []
    bootstrapY = []
    for m in range(s):
        meanX_o.append(np.mean(vectorX_k[m]))
        bootstrapX.append(meanX_o[m])
        meanY_o.append(np.mean(vectorY_k[m]))
        bootstrapY.append(meanY_o[m])
    vectorX.append(bootstrapX) #make a list with all the samples in there
    vectorY.append(bootstrapY)

plt.plot(Percentage_array, out_array_i_X, color='green')
plt.plot(Percentage_array, out_array_i_Y, color='blue')
plt.show()

#Statistical test with the use of confidence bands (TEST 1)

# Make vectors of the new trajectories with all points with the same input
    value in one vector, so s vectors
database_p = []
database_q = []
for p in range(s):
    point_p = []
    point_q = []
    for q in range(ni):
        point_p.append(vectorX[q][p])
        point_q.append(vectorY[q][p])
    database_p.append(point_p)
    database_q.append(point_q)

#Calculating mean and standard error of the samples
meanX_test1 = []
meanY_test1 = []
stdX_test1 = []
stdY_test1 = []
for mm in range(s):
    meanX_test1.append(np.mean(database_p[mm]))
    meanY_test1.append(np.mean(database_q[mm]))
    stdX_test1.append(np.std(database_p[mm]))
    stdY_test1.append(np.std(database_q[mm]))

#Calculating the confidence intervals per point
confidence_band_low_group1 = []
confidence_band_high_group1 = []
mean_trajectory_group1 = []
confidence_band_low_group2 = []
confidence_band_high_group2 = []
mean_trajectory_group2 = []
for nn in range(s):
    confidence_band_low_group1.append(meanX_test1[nn] - 1.96*stdX_test1[nn])
    mean_trajectory_group1.append(meanX_test1[nn])
    confidence_band_high_group1.append(meanX_test1[nn] + 1.96*stdX_test1[nn
        ])
    confidence_band_low_group2.append(meanY_test1[nn] - 1.96*stdY_test1[nn])
    mean_trajectory_group2.append(meanY_test1[nn])
```

```
    confidence_band_high_group2.append(meanY_test1[nn] + 1.96*stdY_test1[nn
        ])

#Plotting the cosines and the confidence bands
plt.plot(Percentage_array, confidence_band_low_group1, color='green', marker
    ".")
plt.plot(Percentage_array, mean_trajectory_group1, color='green', marker="."
    , linestyle='dotted')
plt.plot(Percentage_array, confidence_band_high_group1, color='green',
    marker=".")
plt.plot(Percentage_array, confidence_band_low_group2, color='blue', marker=
    ".")
plt.plot(Percentage_array, mean_trajectory_group2, color='blue', marker=".",
     linestyle='dotted')
plt.plot(Percentage_array, confidence_band_high_group2, color='blue', marker
    ".")
plt.xlabel("Percentage of the stride")

# Statistical test with two sample t.test on each input point of the mean
    functions
of the groups
X = database_p
Y = database_q

alpha = 0.05
alpha_bonferroni = 0.05/s

pvalues = []
rejected_pvalues = []

for t in range(s):
  ttest = stats.ttest_ind(X[t],Y[t])
  pvalues.append(ttest[1])

for p in range(s):
  if pvalues[p] < alpha_bonferroni:
    rejected_pvalues.append(1)
  else:
    rejected_pvalues.append(0)

print(rejected_pvalues)

#Bootstrap based k sample testing for functional data (TEST 3)

# Make vectors of the new trajectories with all points with the same input
    value in one vector, so s vectors
database_p = []
database_pp = []
for p in range(s):
    point_p = []
    point_pp = []
    for q in range(ni): #ni=mi
        point_p.append(vectorX[q][p])
        point_pp.append(vectorY[q][p])
    database_p.append(point_p)
    database_pp.append(point_pp)

#Calculating mean and mean trajectory (STEP 1)
meanX = []
meanY = []
for m in range(s):
```

```
    meanX.append(np.mean(database_p[m]))
    meanY.append(np.mean(database_pp[m]))

#Residual functions (STEP 2)
residualX = []
residualY = []
for r in range(ni): #ni=mi
  residuala = []
  residualb = []
  for bb in range(s):
    residuala.append(vectorX[r][bb] - meanX[bb])
    residualb.append(vectorY[r][bb] - meanY[bb])
  residualX.append(residuala)
  residualY.append(residualb)

SnB = [] #empty list to fill with the Sn^+
for bbb in range(B): #apply step 3 and 4 for every B bootstrap sample's to
    eventually analyze the distribution of the individual Sn's
  #Generating bootstrap functional pseudo-observations (STEP 3)
  pseudo_observations = []
  X_bar = 0
  for j in range(ni):
      for i in range(s):
          X_bar = X_bar + vectorX[j][i] + vectorY[j][i]

  pooled_mean_function_estimator = 1/N * X_bar

  X_plus = [] #bootstrap pseudo observations for group 1
  Y_plus = [] #bootstrap pseudo observations for group 2
  for j in range(ni):
      J = random.randint(0,ni-1)
      X_plus.append(pooled_mean_function_estimator + residualX[J])
      JJ = random.randint(0,mi-1)
      Y_plus.append(pooled_mean_function_estimator + residualY[JJ])

  # Calculating the mean of the new sample of bootstrap functional pseudo
      observations
  database_w = []
  database_ww = []
  for w in range(s):
      point_w = []
      point_ww = []
      for v in range(ni): #ni=mi
          point_w.append(X_plus[v][w])
          point_ww.append(Y_plus[v][w])
      database_w.append(point_w)
      database_ww.append(point_ww)

  #Calculating mean and mean trajectory of bootstrap functional pseudo
      observations
  meanXB = []
  meanYB = []
  for mm in range(s):
      meanXB.append(np.mean(database_w[mm]))
      meanYB.append(np.mean(database_ww[mm]))

  #Calculating Sn^+ for every bootstrap sample
  difference = []
  for i in range(len(meanXB)):
    difference.append(meanXB[i] - meanYB[i])

```

```
  L2_norm = 0
  Sn = 0
  L2_norm = np.linalg.norm(difference) #euclidean norm
  Sn = ((ni * mi)/N) * (L2_norm * L2_norm)
  SnB.append(Sn)

#Sort the SnB's
SnB_sorted = sorted(SnB)
#Estimate the critical value
Critical_value = (SnB_sorted[949] + SnB_sorted[950])/2
print('The critical value for this test is:', Critical_value)

#Calculate the 'real' Sn, for the input data
difference_real = []
for i in range(len(mean_trajectory_group1)):
  difference_real.append(mean_trajectory_group1[i] - mean_trajectory_group2[
      i])

L2_norm_real = np.linalg.norm(difference_real) #euclidean norm
Sn_real = ((ni * ni)/(N)) * (L2_norm_real * L2_norm_real)
print('The test statistic for this test is:', Sn_real)

if Sn_real > Critical_value:
  print("With a confidence level of 95% the null hypothesis (the mean
      trajectories are the same) is rejected")
else:
  print("With a confidence level of 95% we can not reject the null
      hypothesis (the mean trajectories are the same)")
```

LISTING 2: Code for running with data of the data sprint.

```
import pandas as pd
import math
import numpy as np
import matplotlib.pyplot as plt
from sklearn.utils import resample
import csv
from scipy import stats
import random
from math import sqrt
from scipy.signal import argrelmin
from scipy.signal import argrelextrema
from google.colab import drive
drive.mount('/content/gdrive', force_remount=True)

s = 100 #amount of steps between -2pi and 2pi
b = 60 #amount of functions (strides) in the original (input) data #equal
    for all groups
B = 1000 #amount of Bootstrap iterations for the statistical testing (TEST
    3)
ni = 60 #amount of Xij's in group 1 (TEST 3)
mi = 60 #amount of Xij's in group 2 (TEST 3)
k = 2 #amount of groups
N = ni + mi
SnB = []  #list of test statistics for each bootstrap

Percentage_array = np.linspace(0,100,s)

#import the data and process it
data = pd.read_csv('./gdrive/My Drive/Bacheloropdracht/
    Xsens_RP_0904_Meas_angle.csv')
```

```
# adding header to the data

#header_names = ["L5S1", "L4L3", "L1T12", "T9T8", "T1C7", "Head", "
    RightT4Shoulder", "RightShoulder", "RightElbow", "RightWrist", "
    LeftT4Shoulder", "LeftShoulder", "LeftElbow" ,"LeftWrist", "RightHip", "
    RightKneeX", "RightAnkle","RightBallFoot", "LeftHip", "LeftKnee", "
    LeftAnkle", "LeftBallFoot"]

header_names = ["L5S1", "L5S1", "L5S1", "L4L3", "L4L3", "L4L3", "L1T12", "
    L1T12", "L1T12", "T9T8", "T9T8", "T9T8", "T1C7", "T1C7", "T1C7", "Head",
    "Head", "Head", "RightT4Shoulder", "RightT4Shoulder", "RightT4Shoulder",
    "RightShoulder", "RightShoulder", "RightShoulder", "RightElbow", "
    RightElbow", "RightElbow", "RightWrist", "RightWrist", "RightWrist", "
    LeftT4Shoulder", "LeftT4Shoulder","LeftT4Shoulder", "LeftShoulder", "
    LeftShoulder", "LeftShoulder", "LeftElbow", "LeftElbow" , "LeftElbow" ,"
    LeftWrist", "LeftWrist", "LeftWrist", "RightHip", "RightHip", "RightHip",
     "RightKneeX", "RightKneeY", "RightKneeZ", "RightAnkle", "RightAnkle", "
    RightAnkle", "RightBallFoot", "RightBallFoot", "RightBallFoot", "LeftHip"
    , "LeftHip", "LeftHip", "LeftKnee", "LeftKnee", "LeftKnee", "LeftAnkle",
    "LeftAnkle", "LeftAnkle", "LeftBallFoot", "LeftBallFoot", "LeftBallFoot"]

b = []

for i in header_names:
    b.extend([i])

data.columns = b

right_knee_data=data[["RightKneeZ"]]


#first 60 steps
first_60_steps = right_knee_data[33986:450000].to_numpy().flatten()
minima = [179, 358, 538, 719, 898, 1076, 1254, 1435, 1614, 1791, 1970, 2148,
     2326, 2506, 2684, 2864, 3046, 3225, 3404, 3584, 3766, 3947, 4125, 4303,
    4479, 4657, 4834, 5012, 5190, 5366, 5541, 5718, 5896, 6074, 6252, 6430,
    6607, 6784, 6964, 7143, 7322, 7500, 7678, 7856, 8035, 8212, 8393, 8573,
    8752, 8930, 9109, 9284, 9463, 9643, 9823, 10000, 10180, 10359, 10538,
    10717, 10893] #List with all the beginpoints of a stride
minima2 = [358, 538, 719, 898, 1076, 1254, 1435, 1614, 1791, 1970, 2148,
    2326, 2506, 2684, 2864, 3046, 3225, 3404, 3584, 3766, 3947, 4125, 4303,
    4479, 4657, 4834, 5012, 5190, 5366, 5541, 5718, 5896, 6074, 6252, 6430,
    6607, 6784, 6964, 7143, 7322, 7500, 7678, 7856, 8035, 8212, 8393, 8573,
    8752, 8930, 9109, 9284, 9463, 9643, 9823, 10000, 10180, 10359, 10538,
    10717, 10893] #List with all the endpoints of a stride

counter = 0
X = []
for i in range(len(minima2)):
    counter = counter + 1
    a = minima[i]
    b = minima2[i]
    x = b - a
    plt.xlabel("Percentage of the strides")
    plt.ylabel("Knee angle of the right knee")
    steps1 = np.linspace(a,b,s)
    stepk1 = []
    for jj in steps1:
        kk = round(jj)
        stepk1.append(first_60_steps[kk])
    X.append(stepk1)
```

```
64      plt.plot(np.linspace(0,100,s),stepk1)
65
66  plt.show() #Plotting the first 60 strides against the percentage of the
        stride
67
68  #last 60 steps
69  last_60_steps = right_knee_data[257400:268400].to_numpy().flatten()
70  # print(argrelextrema(last_60_steps, np.less))
71  minima_last = [23, 200, 374, 550, 725, 901, 1077, 1254, 1431, 1607, 1781,
        1957, 2133, 2310, 2485, 2661, 2837, 3011, 3189, 3362, 3538, 3717, 3895,
        4072, 4251, 4428, 4607, 4783, 4960, 5134, 5312, 5486, 5661, 5837, 6012,
        6189, 6365, 6544, 6723, 6901, 7081, 7259, 7437, 7613, 7791, 7967, 8146,
        8324, 8500, 8677, 8853, 9029, 9206, 9383, 9561, 9735, 9912, 10090, 10267,
         10443, 10621] #List with all the beginpoints of a stride
72  minima2_last = [ 200, 374, 550, 725, 901, 1077, 1254, 1431, 1607, 1781,
        1957, 2133, 2310, 2485, 2661, 2837, 3011, 3189, 3362, 3538, 3717, 3895,
        4072, 4251, 4428, 4607, 4783, 4960, 5134, 5312, 5486, 5661, 5837, 6012,
        6189, 6365, 6544, 6723, 6901, 7081, 7259, 7437, 7613, 7791, 7967, 8146,
        8324, 8500, 8677, 8853, 9029, 9206, 9383, 9561, 9735, 9912, 10090, 10267,
         10443, 10621] List with all the endpoints of a stride
73
74  counter2 = 0
75  Y = []
76  for i in range(len(minima2_last)):
77      counter2 = counter2 + 1
78      a2 = minima_last[i]
79      b2 = minima2_last[i]
80      x2 = b2 - a2
81      plt.xlabel("Percentage of the strides")
82      plt.ylabel("Knee angle of the right knee")
83      steps = np.linspace(a2,b2,s)
84      stepk = []
85      for j in steps:
86          k = round(j)
87          stepk.append(last_60_steps[k])
88      Y.append(stepk)
89      plt.plot(np.linspace(0,100,s),stepk)
90
91  plt.show() #Plotting the last 60 strides against the percentage of the
        stride
92
93  #Statistical test with the use of confidence bands (TEST 1)
94
95  # Make vectors of the new trajectories with all points with the same input
        value in one vector, so s vectors
96  database_p = []
97  database_q = []
98  for p in range(s):
99      point_p = []
100     point_q = []
101     for q in range(ni):
102         point_p.append(X[q][p])
103         point_q.append(Y[q][p])
104     database_p.append(point_p)
105     database_q.append(point_q)
106
107 #Calculating mean and standard error of the samples
108 meanX = []
109 meanY = []
110 stdX = []
111 stdY = []
```

```
for mm in range(s):
    meanX.append(np.mean(database_p[mm]))
    meanY.append(np.mean(database_q[mm]))
    stdX.append(np.std(database_p[mm]))
    stdY.append(np.std(database_q[mm]))


#Calculating the confidence intervals per point
confidence_band_low_group1 = []
confidence_band_high_group1 = []
mean_trajectory_group1 = []
confidence_band_low_group2 = []
confidence_band_high_group2 = []
mean_trajectory_group2 = []
for nn in range(s):
    confidence_band_low_group1.append(meanX[nn] - abs(stats.norm.ppf(.05/s))
        *stdX[nn])
    mean_trajectory_group1.append(meanX[nn])
    confidence_band_high_group1.append(meanX[nn] + abs(stats.norm.ppf(.05/s)
        )*stdX[nn])
    confidence_band_low_group2.append(meanY[nn] - abs(stats.norm.ppf(.05/s))
        *stdY[nn])
    mean_trajectory_group2.append(meanY[nn])
    confidence_band_high_group2.append(meanY[nn] + abs(stats.norm.ppf(.05/s)
        )*stdY[nn])

#Plotting the cosines and the confidence bands

plt.plot(Percentage_array[15:18], confidence_band_low_group1[15:18], color='
    green', marker=".")
plt.plot(Percentage_array[15:18], mean_trajectory_group1[15:18], color='
    green', marker="_", linestyle='dotted')
plt.plot(Percentage_array[15:18], confidence_band_high_group1[15:18], color=
    'green', marker=".")
plt.plot(Percentage_array[15:18], confidence_band_low_group2[15:18], color='
    blue', marker=".")
plt.plot(Percentage_array[15:18], mean_trajectory_group2[15:18], color='blue
    ', marker="_", linestyle='dotted')
plt.plot(Percentage_array[15:18], confidence_band_high_group2[15:18], color=
    'blue', marker=".")
plt.xlabel("Percentage of the stride")
plt.ylabel("Knee angle of the right knee during a run")
plt.show()

# Statistical test with two sample t.test on each input point of the mean
    functions of the groups
x = database_p
y = database_q

alpha = 0.05
alpha_bonferroni = 0.05/s

pvalues = []
rejected_pvalues = []

for t in range(s):
  ttest = stats.ttest_ind(x[t],y[t])
  pvalues.append(ttest[1])

#print(pvalues)

```

```
for p in range(s):
  if pvalues[p] < alpha_bonferroni:
    rejected_pvalues.append(1)
  else:
    rejected_pvalues.append(0)

print(rejected_pvalues)

for i in range(len(rejected_pvalues)):
  if rejected_pvalues[i] == 1:
      plt.plot(Percentage_array[i], mean_trajectory_group1[i], color='red',
          marker=".", linewidth = "0")
  else:
      plt.plot(Percentage_array[i], mean_trajectory_group1[i], color='black'
          , marker=".", linewidth = "0")
      plt.xlabel("Percentage of the stride")
      plt.ylabel("Knee angle of the right knee during a run")

plt.show() #plot with rejected pvalues

#Bootstrap based k sample testing for functional data (TEST 3) first cell

# Make vectors of the new trajectories with all points with the same input
    value in one vector, so s vectors
database_p = []
database_pp = []
for p in range(s):
    point_p = []
    point_pp = []
    for q in range(ni): #ni=mi
        point_p.append(X[q][p])
        point_pp.append(Y[q][p])
    database_p.append(point_p)
    database_pp.append(point_pp)

#Calculating mean and mean trajectory (STEP 1)
meanX = []
meanY = []
for m in range(s):
    meanX.append(np.mean(database_p[m]))
    meanY.append(np.mean(database_pp[m]))

#Residual functions (STEP 2)
residualX = []
residualY = []
for r in range(ni): #ni=mi
  residuala = []
  residualb = []
  for bb in range(s):
    residuala.append(X[r][bb] - meanX[bb])
    residualb.append(Y[r][bb] - meanY[bb])
  residualX.append(residuala)
  residualY.append(residualb)

SnB = [] #empty list to fill with the Sn^+
for bbb in range(B): #apply step 3 and 4 for every B bootstrap sample's to
    eventually analyze the distribution of the individual Sn's
  #Generating bootstrap functional pseudo-observations (STEP 3)
  pseudo_observations = []
  X_bar = 0
  for j in range(ni):
```

```
        for i in range(s):
            X_bar = X_bar + X[j][i] + Y[j][i]

    pooled_mean_function_estimator = 1/N * X_bar

    X_plus = [] #bootstrap pseudo observations for group 1
    Y_plus = [] #bootstrap pseudo observations for group 2
    for j in range(ni):
        J = random.randint(0,ni-1)
        X_plus.append(pooled_mean_function_estimator + residualX[J])
        JJ = random.randint(0,mi-1)
        Y_plus.append(pooled_mean_function_estimator + residualY[JJ])

    # Calculating the mean of the new sample of bootstrap functional pseudo
        observations
    database_w = []
    database_ww = []
    for w in range(s):
        point_w = []
        point_ww = []
        for v in range(ni): #ni=mi
            point_w.append(X_plus[v][w])
            point_ww.append(Y_plus[v][w])
        database_w.append(point_w)
        database_ww.append(point_ww)

    #Calculating mean and mean trajectory of bootstrap functional pseudo
        observations
    meanXB = []
    meanYB = []
    for mm in range(s):
        meanXB.append(np.mean(database_w[mm]))
        meanYB.append(np.mean(database_ww[mm]))

    #Calculating Sn^+ for every bootstrap sample
    difference = []
    for i in range(len(meanXB)):
      difference.append(meanXB[i] - meanYB[i])

    L2_norm = 0
    Sn = 0
    L2_norm = np.linalg.norm(difference) #euclidean norm
    Sn = ((ni * mi)/N) * (L2_norm * L2_norm)
    SnB.append(Sn)


#Sort the SnB's
SnB_sorted = sorted(SnB)
#Estimate the critical value
Critical_value = (SnB_sorted[949] + SnB_sorted[950])/2
print('The critical value for this test is:', Critical_value)

#Calculate the 'real' Sn, for the input data
difference_real = []
for i in range(len(mean_trajectory_group1)):
    difference_real.append(mean_trajectory_group1[i] - mean_trajectory_group2[
        i])

L2_norm_real = np.linalg.norm(difference_real) #euclidean norm
Sn_real = ((ni * mi)/N) * (L2_norm_real * L2_norm_real)
print('The test statistic for this test is:', Sn_real)
```

```
277
278 if Sn_real > Critical_value:
279   print("With a confidence level of 95% the null hypothesis (the mean
        trajectories are the same) is rejected")
280 else:
281   print("With a confidence level of 95% we can not reject the null
        hypothesis (the mean trajectories are the same)")
```