University of Rome “La Sapienza”

Master’s degree in data science

Academic Year 2025/2026

Course “Fundamentals of statistical learning”

Professor Pierpaolo Brutti; Teacher Assistant Valeria Avino

Students: Riccardo Pugliese, Suriano Leonardo, Mehmet Sener

Report for “The Homework”

Rome, February 2026

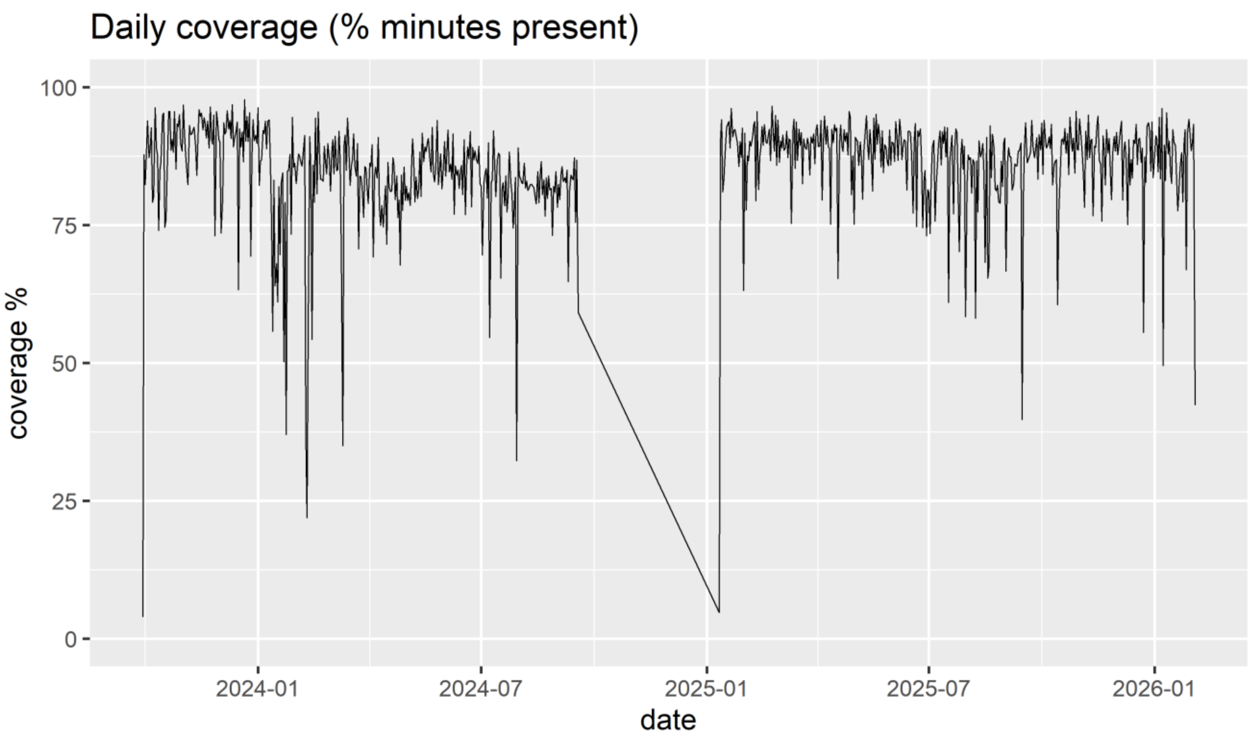
During the first year of our Master at Sapienza University, in Rome, we (Riccardo Pugliese, Leonardo Suriano and Mehmet Sener) had the pleasure of attending, in Fall 2025, the course of Fundamentals of Statistical Learning. The course was taught by Professor Pierpaolo Brutti, that during all his lessons managed to keep us engaged and interested, making his lessons enjoyable and never boring. Brutti was helped, during the whole semester, also by his teacher assistant, Valeria Avino, that managed to create with us a special bond, being close to us both in age and in academic background. She helped us during her office hours with her exercises, that were useful both to understand what we were doing in lessons with the professor but also to understand a bit better how the exams would have looked like. It was also very kind of her that after some of her office hours, she spent some time talking to us, sharing with us insight about future courses and tips about what we found to be really useful. We are grateful to her for all of that.

At the end of his course, the Professor gave us one project to do, about conformal prediction procedures aimed at functional data. It was not the first time we were doing this type of homework (during the semester he gave us two classworks, really useful for us not only to keep the pace of what we were doing during classes, but also they gave us the opportunity to apply in practice some of the theoretical knowledge we were gaining in classes), but the ones we already did were smaller, were requiring less time and were carried out in class. For these reasons the last homework the Professor gave us made us face new challenges that we tried, with effort, steadiness and a bit of pain, to face. What is written in this document is not only a presentation of the project/work we did and that we will handle to the professor and to his teacher assistant as our final homework, but also an insight on the processes that led us doing the choice we did, on the difficulties we had and on how we handled them.

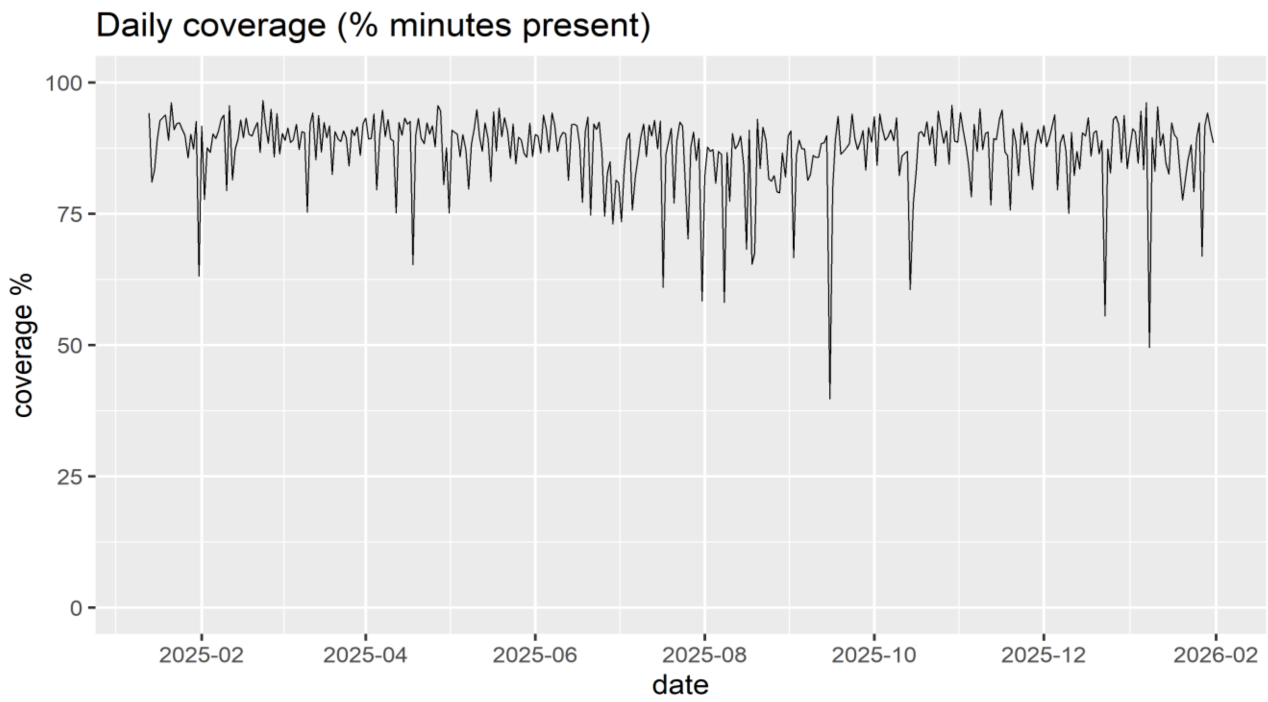
Moreover, before we start talking more in depth about the project itself, we want to notify the reader that this project has been possible thanks to our hard work but also thanks to Artificial Intelligent tools, that gave us a huge help in organizing our ideas, helping us not only in the starting phase (making us understand better what we were asked to do), but also during the code writing (the code below may be AI generated code). In addition, these tools provided us with especially good help during our difficult and/or uncertain moments, enabling us to properly solve the problems we met and move forward. Since we just wrote how much AI tools have helped us, is also worth mentioning that still most of the work has been done, as it should be, by us. All this essay has been written by us without any AI tools help, the results explanation has been done by us and the code, whenever was written with the help of AI tools, has been carefully reviewed, correct and modified (when needed) by us. Our work has not been just a copy paste from AI chatbot.

Coming back to our project, the first thing we must do is to present, both to the readers and to whoever is interested in our work, the data we have used. We will also, as explicitly required by the teacher, describe the data collection phase. Where needed, some pictures will be inserted, and great details will be provided. After the “data description” phase, we will explain how we turned our data into functional data, showing summing statistics and plots that helped us (and we hope will help the readers as well) understanding what was going on in our data. In this phase we will basically explain you how we build our own “functional” dataset. After these two phases we will present our last step, in which we picked the approach presented in section 3 of the article proposed to us by the teacher (A Conformal Prediction Approach to Explore Functional Data by Jing Lei, Alessandro Rinaldo, Larry Wasserman) in order to implement conformal predictions on functional data. Of course, the functional data we used are the ones we created from our data and that we stored in our own functional dataset. The presentation of this last phase will also include several comments on both how we handled our data and on the results we got. During the description of all the three phases we will share with you our thoughts about the data, commenting on and analyzing the graphs and the images we will show. We will share with you what we think they are suggesting us, hoping all of this will be useful to readers and will help them create their own thoughts, comments and analysis. Moreover, we would love to hear from you some of them, so feel free to share them with us (our contact informations are at the end of this paper). In all the phases we used R as programming language, because of both its statistical focus and its simplicity, but almost everything we did in R would be possible to replicate in several other programming languages. We will attach to this file also the R code we wrote, however please note that most of our comments about the code will be placed in the R code itself, so if you want a detailed explanation of our code line, it is better to directly deep dive into the scripts.

As suggested by the professor, we decided to create our own functional dataset starting from data that objects around us collect. In particular, we realized how massive the data that a smartwatch or a smartphone can collect about us are, mainly because we are using them a lot granting them every type of permission they ask for. In our case, both Riccardo and Leonardo have a smartwatch, that is constantly tracking all our health statistics, like hearth rate, how much we are sleeping, what time we go to bed and what time we wake up, how many steps we are doing every day, where we are going every minutes of our life (Riccardo's watch has gps built in, so it can do it by itself, while Leonardo's watch has to be connected, for retrieving gps information, to Leo's phone for accessing the phone gps location), how much oxygen we have in our blood and so on. We decided to focus our attention on the heart rate, and we tried to download the data that both of our watches collected. We managed to download only Riccardo's heart rate data, and this unfortunately was not up to us but was due to the different policy of smartwatch makers. Since Riccardo has a Samsung smartwatch, it was easy to get the data about his health statistics, because in Samsung health app there is a shortcut made properly for exporting your health data. About Leonardo's smartwatch, in the Xiaomi health app (he has a Xiaomi smartwatch) there is no such shortcut, and if he wanted to get his data he needed to write to them directly (something that he did), and in up to 15 working days they would have reach him back with his health data. Of course we could not wait 15 working days just to have his data, so we decided to go with Riccardo's ones. From the Samsung app it is possible to download only a big folder with all the data your watch has collected about your health, so once we downloaded the folder, we decided to clean it, in order to create a dataset only with his heart rate. We started our cleaning process by extracting from the big folder only the heart rate data that were stored in the folder we will attach to this paper. We then started to work on this data, cleaning them. We decided to keep only the most recent data, the ones from January 2025. This choice was made for two reasons. The first one is because there is a big gap in health rate data, as visible in the graph below. Daily coverage is the percentage of measurements available divided by the total possible number of measurements, day by day.



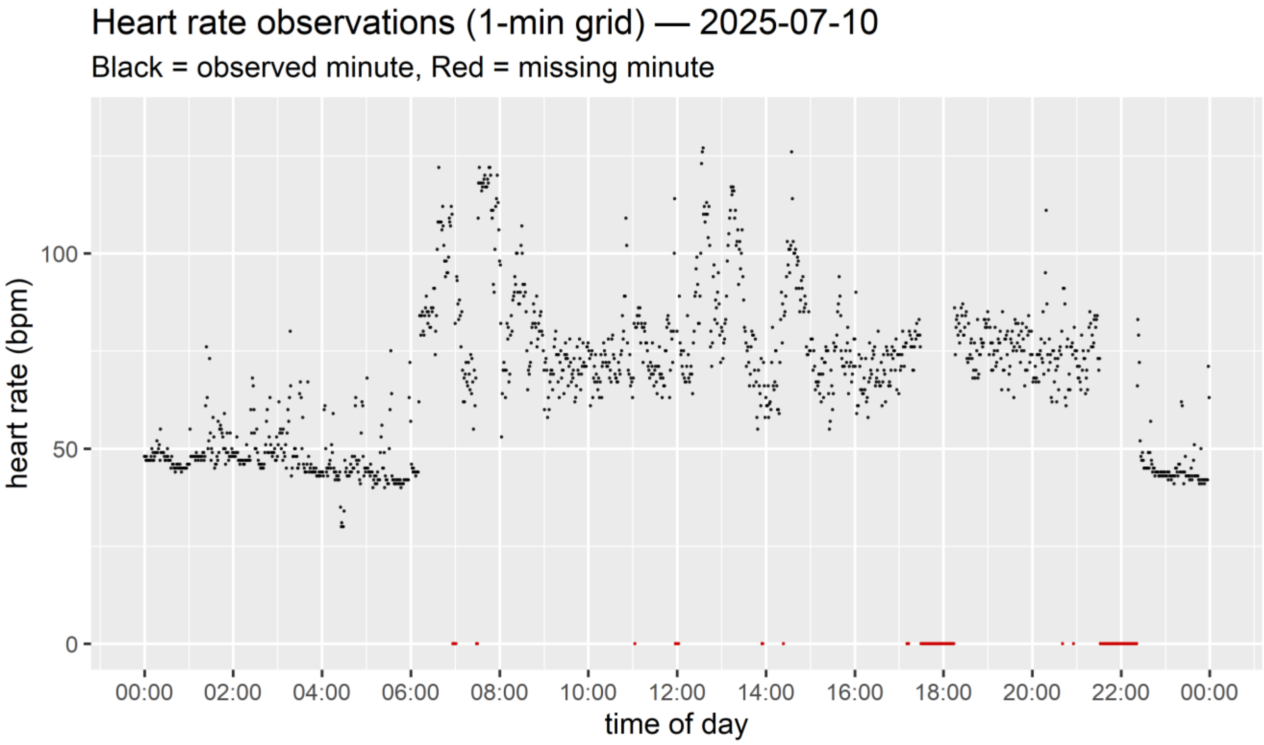
As you can see on the graph, there is a huge gap between august/September 2024 and January 2025, due to the fact that in the end of 2024'summer Riccardo's watch broke down, and He bought a new one a bit after Christmas 2024. Because of this gap, we decided to delete the data between September 2024 and December 2024. About the reason why we didn't kept even the data collected by the first watch before summer 2024, it is important to notice that the watch Riccardo stopped using in the end of summer 2024 was a 2 years older model than the one He bought on Christmas 2024, and so it is possible that Samsung updated the hearth rate sensors in the watch. Since the sensor could have been different, we decided to delete all the old watch data, keeping basically only data from January 2025 on. Moreover, the data we ended up having at the end of our cleaning process were still enough, so we didn't even need the old watch data so much. We also deleted the few data the new watch collected after January 31, 2026. At the end, we ended up having almost 13 full months of data (our first record is dated January 13, 2025, while the last is dated January 31, 2026), during which I used my watch continuously. This can clearly be seen in the graph below, in which there are no gaps in the measurement.



After this first cleaning, we decided to check how often the watch was recording the heart rate, and we found out that the watch is measuring Riccardo's heartbeat continuously, having up to 60 measurements for hours (one observation for minute). This was easy to find out, thanks to both r (we wrote a small code that counted the maximum number of observations in every one-hour period) and to the watch settings, that were set to measure heartbeat continuously.

```
> day_1440[, hour := format(minute_ts, "%H")]
> max_obs_1h <- day_1440[!is.na(heart_rate_1min), .N, by = hour][, max(N)]
> cat("Max osservations in 1 hour:", max_obs_1h, "\n")
Max osservations in 1 hour: 60
```

We were also curious to check how the heartbeats were approximately going in a random day, and we decided to plot the observation we had for a given day on a graph and most of the time we were plotting a random day, we were getting a graph somehow like the one below.



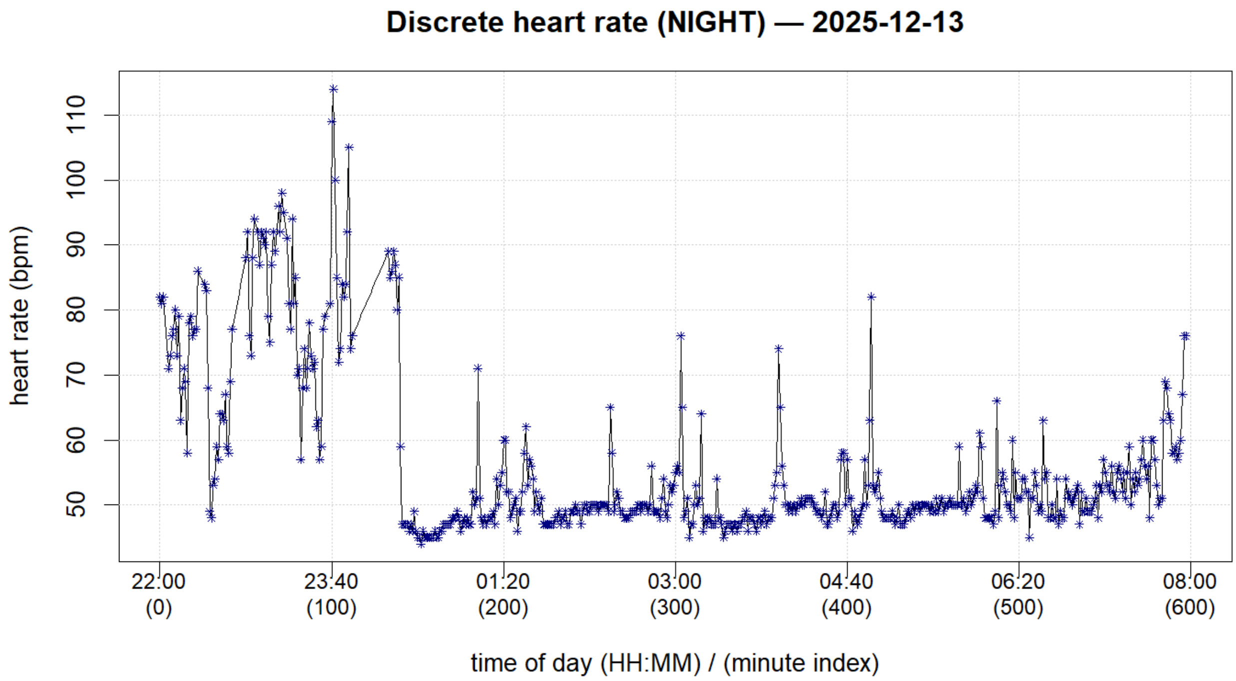
In most of them we noticed that there were periods of time with no observation at all (mainly in the morning and/or in the evening, as can we see in the graph above), and that, in general, during the day the heartbeats were higher than the ones during the night (except for Riccardo's, very rare, clubbing nights). We quickly realized that it was completely normal to have higher heartbeat during the day, because in the night most of the time Riccardo is sleeping, so his heart doesn't work as hard as it does during the day, in which he is moving and/or doing something all the time. We also realized that the interval, in the graphs, with no observations were most probably the time in which he was charging his watch, something that unfortunately for him, he has to do at least once a day and that usually happens either in morning, when he is getting ready for his daily activities, or in the evening, when he come back at home and get ready to go sleeping. In all the graphs we looked at, we almost never observed some missing measurements during the night (unless the watch died while he was sleeping, something difficult because he usually charges it if it doesn't have enough battery to survive until the next morning). Because of these "charging breaks", usually happening in the morning and in the evening, and the fact that in the night his heartbeats were significantly lower, we decided to split the 24 hours day period into two smaller periods, one being in the night and one being in the day. We also decided not to include in our day and night split the time between 8.00 and 9.30 and between 19.30 and 22.00, because usually these are the times in which he is usually charging his watch. We ended up having two different groups of data of 10 hours each: a day group with a higher average heartbeat rate with records between 9.30 and 19.30 and a night group with a much lower average heartbeat rate with records between 22.00 and 8.00.

Once we had, for each day, two different groups of data (day and night), we started studying them, and after looking at some graphs (mainly the time series of the heartbeat records) we decided to allow no gaps in our records bigger than 30 minutes for each gap, and in case the biggest gap in each series is a big gap (between 20 and 30 minutes), we wanted the second biggest gap of that series to be smaller than 20 minutes (so to be small gap). Moreover, we also decided the total missing points of one serie should not be bigger than 15% of the maximum number of observations the watch could record during the 10 hours period (so the missing values should not be more than 90 in each 10 hours period). These measures were aimed at keeping, as data, only series with a lot of real observation and really few NA values, and even if there were some NA values, they would never be too much concentrated close to each other in continuous strikes (the biggest number of continuous NA values was set to 30). These measures were especially important to be taken considering that this data will be used to create some functional data, so they need, as much as possible, to be homogenous and continuous thought time.

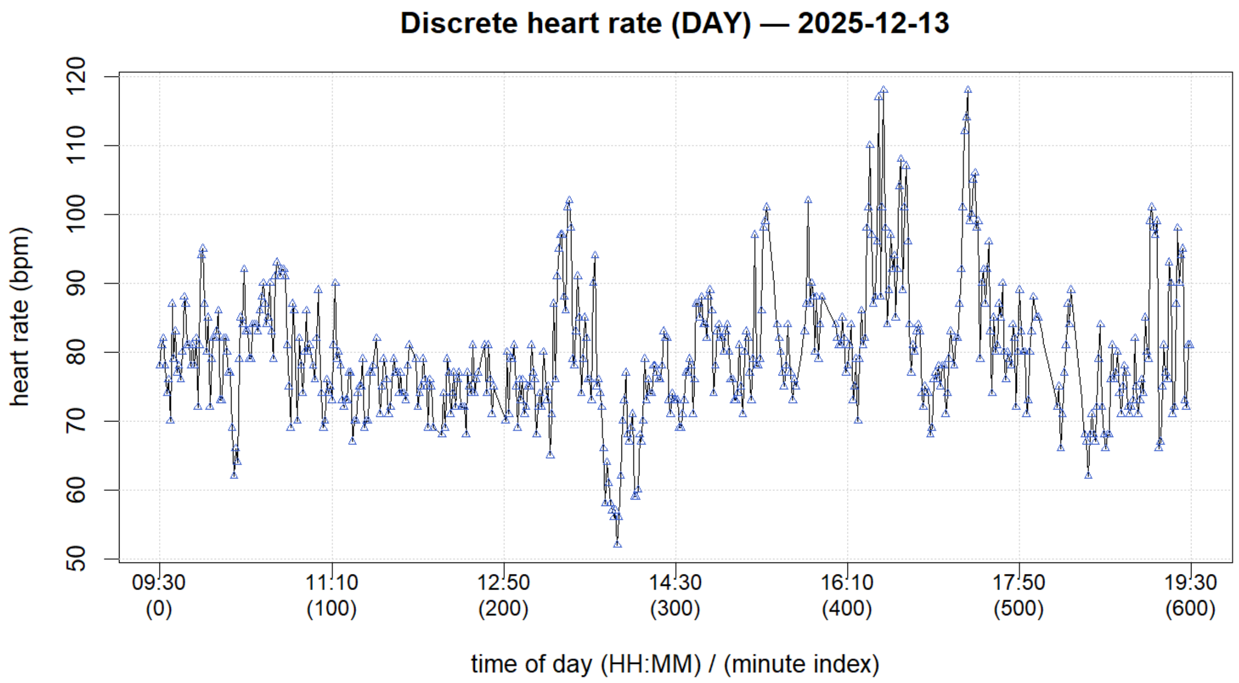
After all these procedures, we ended up having 148 day periods and 159 night periods. Each of them has at least 510 observations, and the average coverage (the number of points in the day period divided by the total number of possible points, i.e. 600) during the day is 96.34%, while the one in the night is 96.22%. These stats can be seen in the screenshot below, where we summarize some of them. As you can clearly see, these stats are not bad at all, and in general having an almost balanced set of samples (148 and 159 are very similar numbers) makes the prediction we can make based on these samples comparable.

```
=== SOMMARIO PER PERIODO ===
   period     n gap1_medio gap1_min gap1_max gap2_medio gap2_min gap2_max
   <char> <int>      <num>    <int>    <int>      <num>    <int>    <int>
1:    day   148       9.35        1       30       3.14        0       19
2:  night   159      14.20        0       30       4.40        0       20
   coverage_media coverage_min coverage_max
            <num>        <num>        <num>
1:          96.34     86.16667     99.83333
2:          96.22     87.33333    100.00000
>
```

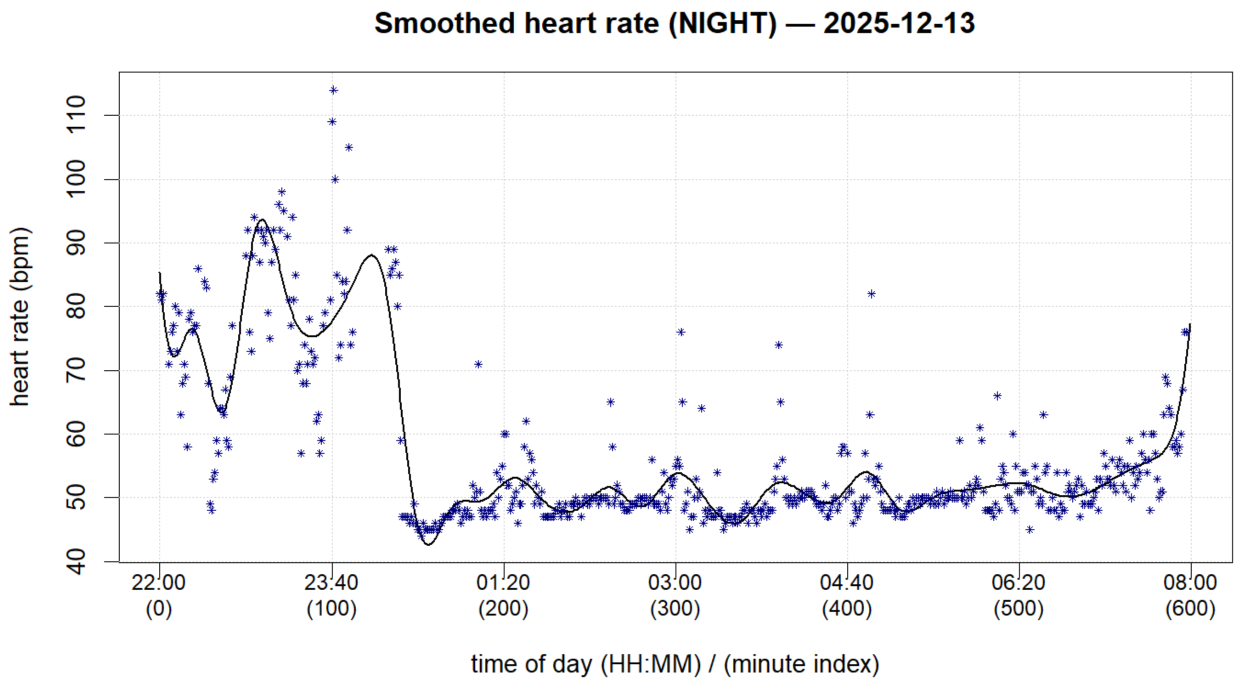
We stored this data in three different datasets, one for night, one for day and one with the combination of day and night. Once we did this, we tried to get, out of them, some functional data. To do so, we used a technique called smoothing, that enabled us to create a curve for each period from the observations we had in each period. Since in almost every period of time we had some missing values, we inputted some of them with a process called linear interpolation, that allowed us not to have anymore NA values. In the end we managed to create two functional objects that contain, inside them, the curves (one of them contain discrete functions, while the other contains continuous functions). Below there are some images of our functional data.



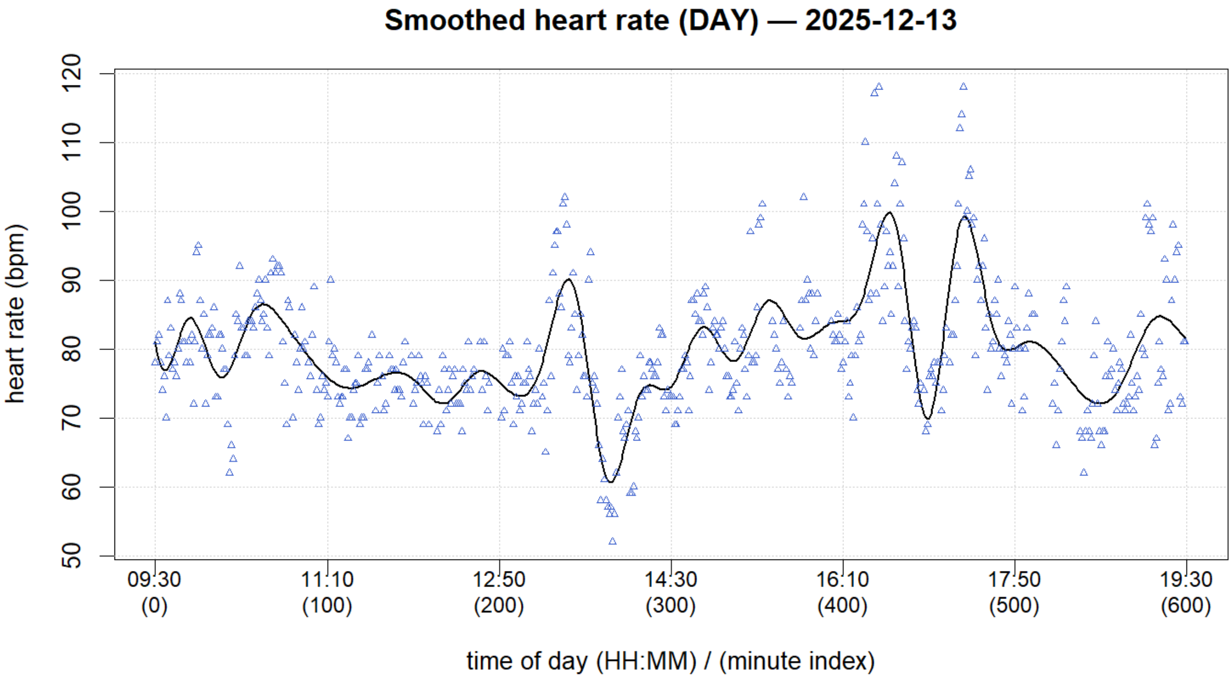
This image depicts a discrete function of a random night period. Each star is one recording. As you can see there is a really low heartrate between minute 140 and 582 (≈ 00:20–07:42), meaning that at that time Riccardo was probably sleeping. Moreover, you can see for example how, between minute 113 and 132 (≈ 23:53–00:12) there are no points, meaning that the watch was unable to record the heartbeat, most probably because it was charging. Overall, after midnight the heart rate becomes much more stable (around ~50–60 bpm).



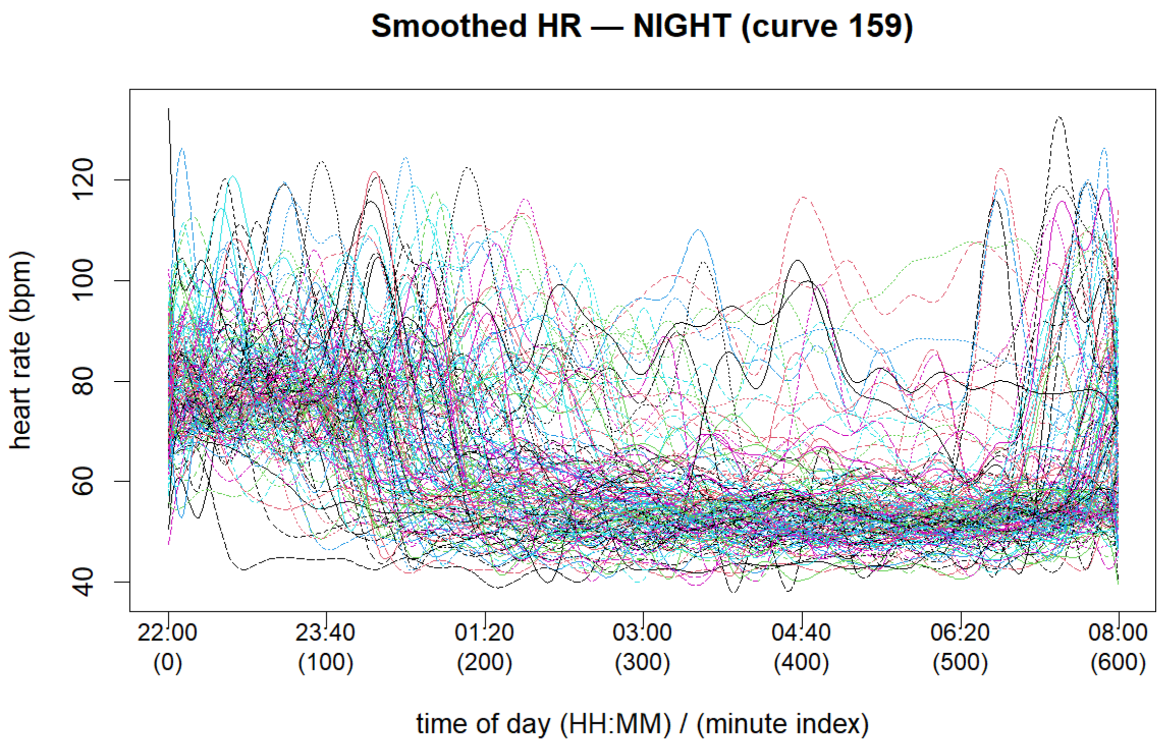
This image depicts basically the same thing as before, with the only difference that here we are looking at a discrete function in a random day period, and that recordings are shown as triangles and not as stars. Here the average heart rate is higher and more variable, with several peaks likely due to daily activities. As you can see during this specific day range probably Riccardo did not recharge his watch, so no clear gaps (charging intervals) are observable.



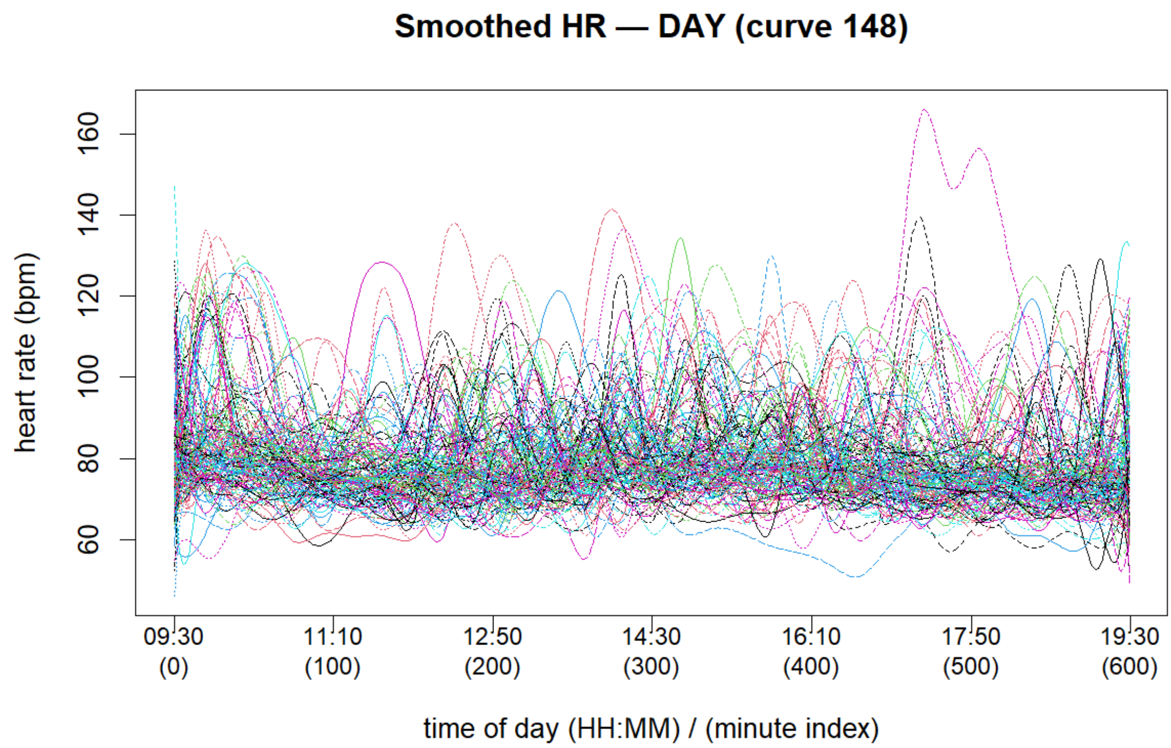
Here we have the continous version of the night period that before we represented as discrete. As you can see, we have shoothned the function a bit in order not to account for too much noise. Looking at this function it is easily visible when I went sleeping and when I woke up.



Here we have the exact same thing as the graph above but regarding the day (it is the same day that before we represented as discrete function).

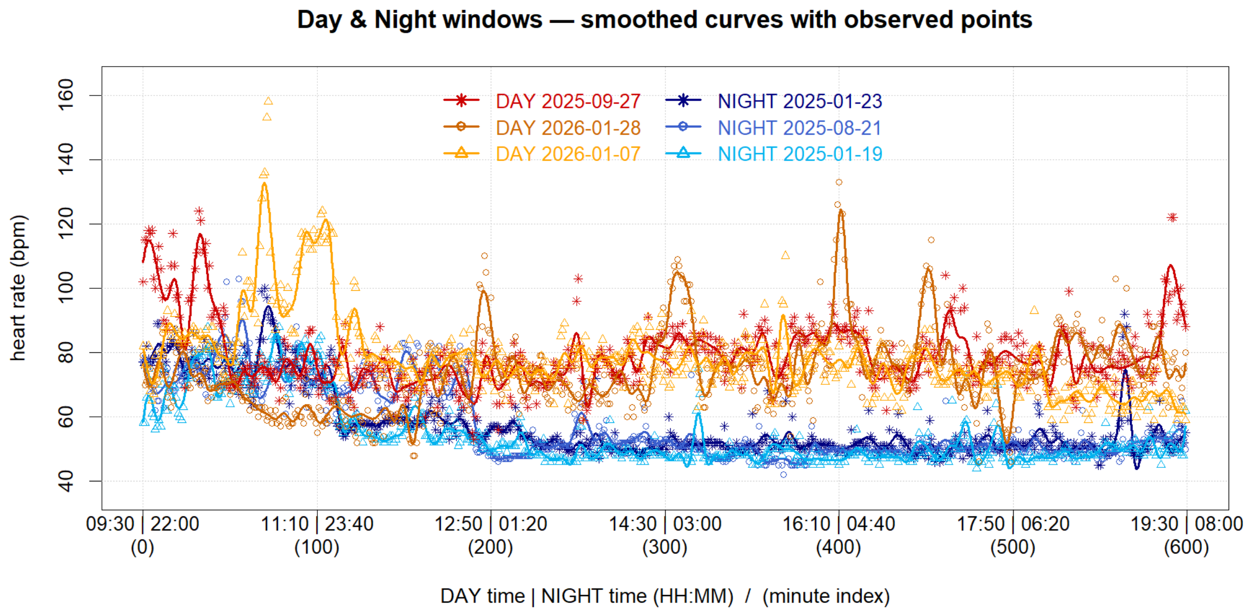


Here we have all the continous curves of all nights combined. As you can see, in the vast majority of nights, Riccardo was sleeping between minute 200 and minute 550, with some few exceptions (the lines that are above 60 between minute 200 and 550). The nights that he was not sleeping it is probably because he hang out in the night, went clubbing or worked. Please note that this data shows a lot of variability, especially if compared to the one below.



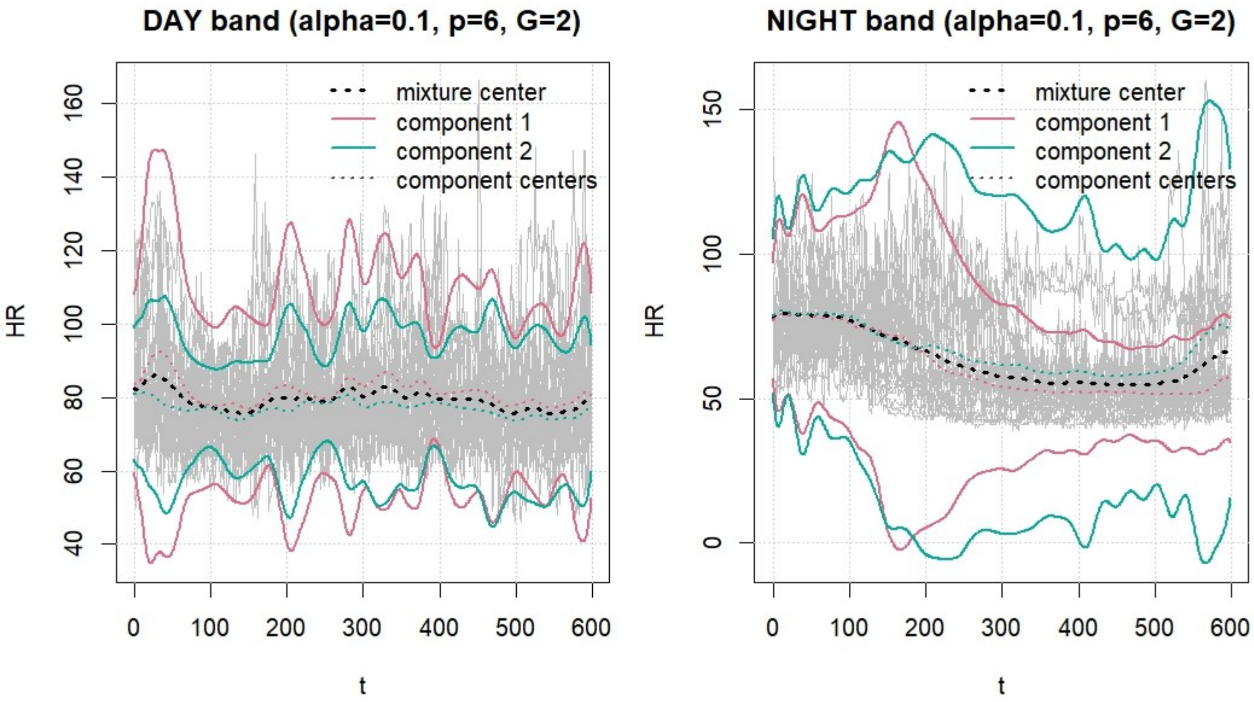
This graph is, on the other hand, basically the same thing as the graph above but for days. As you can see the heart rates are constant at basically everytime, being mainly concentrated between 65 and 100 bpm. Another interesting fact to notice is that Riccardo almost never slept in the afternoon, as you can see from the fact that there are basically no record below 60.

Last but not least is a graph considered by us as especially nice. Here we can see 6 random curves from six different dates. Three of them are from days period and three of them are from night period. In this graph we can clearly see at what time Riccardo has fallen asleep, and whether he woke up before 8.00. In the three nights in the graph, he never woke up before 8.00. We can also see when he was particularly active during the day (on February 11, for example, he was especially active in the afternoon, while on May 15 he was not active at all (except for a little hill on the right side of the graph).



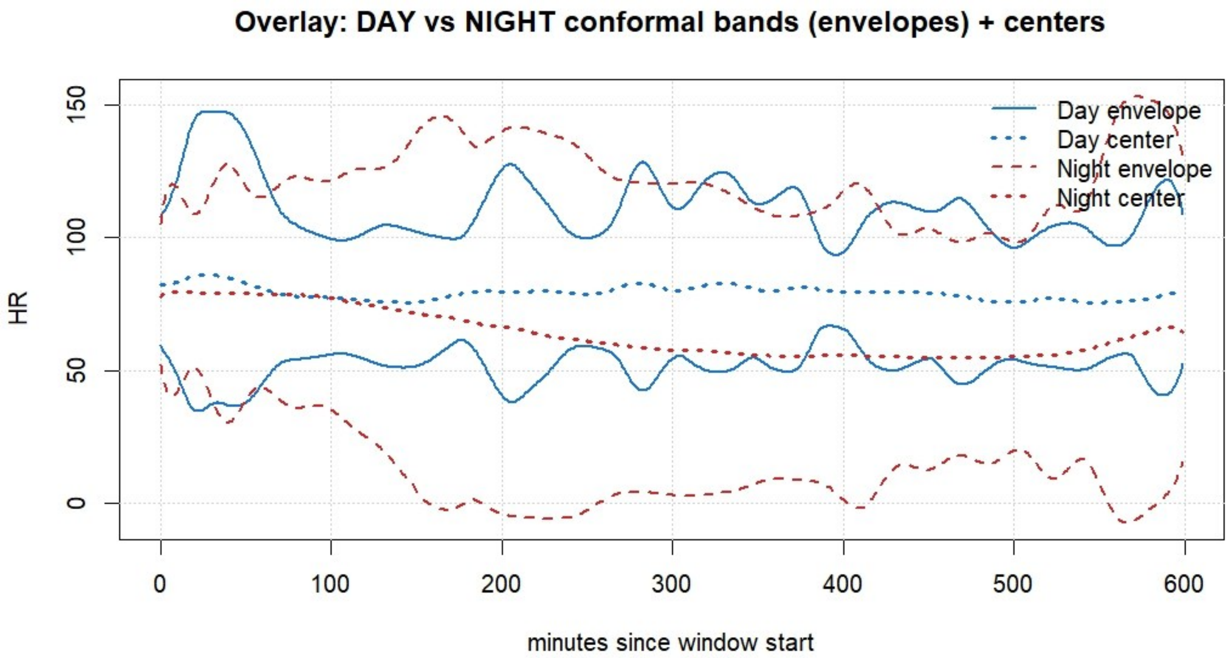
Now that we have our functional data, we decided to implement, in order to get conformal predictions for our data, the approach proposed by the authors of the article A Conformal Prediction Approach to Explore Functional Data in section 3 (Projection-based Conformal Bands). We have chosen this approach because, in our opinion, it was the best for our data, given that our main goal was to obtain a prediction band for future day/night heart-rate curves (one band for night and one band for day) with confidence one minus alpha. In the paper the projection-based strategy is presented exactly for this purpose, because projecting the curves onto a low-dimensional space makes the prediction set easier to get and allows the construction of prediction bands with a finite-sample coverage guarantee. Moreover, section 3 matches our data, because the variability of smooth functional processes can often be captured by a small number of components, and the coverage result remains valid even when the basis is estimated from the training set (as in FPCA). We did not choose section 4 because it is mainly aimed at building a conformal clustering tree based on pseudo-densities (kernel + bandwidth + distance), which is extremely useful for exploratory clustering and hierarchy visualization, but it requires additional tuning choices (e.g., bandwidth selection) and the paper itself highlights that tuning-parameter selection is an important and non-trivial issue. Since hierarchical clustering trees were not our goal, we decided to go for section 3.

In order to implement this approach, we first splitted our functional data into train and calibration set, with a division 70/30. We did it because in this way we could use the train data to get a curve (the curve around which the conformal prediction's band will be built), and then we could use the calibration set in order to make the band more narrow or wider and so including in it the desired alpha level of new curves. This is basically what was suggested in the paper (split (inductive) conformal framework). After that we did a principal component analysis on the functional data of the train set (FPCA), aimed to project each function (each data) onto a lower dimensional score vector which makes the conformal procedure tractable and allows us to model the data distribution in a finite-dimensional space. We then fitted a Gaussian Mixture Model (GMM) to the FPCA scores using two components. We used the GMM because our data show a multi-modal structure (i.e., more than one "typical" pattern), and a mixture model can of course represent this heterogeneity better than a simple Gaussian. After fitting the GMM on the training scores, we computed the conformity score for each curve in the calibration set, in order to check how much the two different groups we detected were similar and/or compact within each other. We also computed the calibration scores, useful to determine the cutoff correlation to the chosen alpha and that determines how wide the final band must be to guarantee coverage. Finally, we map the prediction set obtaining a band for each mixture component, and the final conformal band is the union (envelope) of these component-wise bands. We repeated the full procedure twice, once for the day's functional data and once for the night's functional data. To better illustrate the results, the next section presents the main plots produced by our conformal prediction pipeline, that hopefully will be useful to the reader to better visualize our work and our conformal predictions. It is important to note and remind you that the alpha we choose for our conformal predictions was 0.1, but it was possible to arbitrarily choose any other value (0.01, 0.05 etc).



In this graph we can see our bands in a very clear way. We computed, with alpha = 0.1, 6 dimensions and 2 components, the two relative bands. The idea behind computing different bands is that in the group of day/night functional data, there are some data that are similar to each other, and here we have taken the two more similar clusters for each group (two clusters for the day and two clusters for the night) and compute their own conformal prediction band. In practice, this means that if a functional data belongs to one cluster, it will most probably fall into the respective conformal prediction bands. For having the final intervals in which with an alpha = 0.1 the new curves will be we have to take the union of the two curves, doing an envelope (take the highest curve (even if disjoint) as upper bound and, even if disjoint, the lowest curve as the lower bound). We have done the envelopes both for night and day and below there is the graph of them. It is important to keep in mind that we were "authorized" and "legitimized" to do the envelope for expressing and represent the union because in our specific case they are the same thing (see pictures below) but is not every time like this. Whenever they are not the same, is more correct to take the union of them rather than the envelope. Since in our case it is the same, it is easier to represent our bands as a unique band for day and a unique band for night (instead of two bands for each of them). You can see it in the pictures below.

```
> check_union_equals_envelope(fit_day)
$any_disjoint
[1] FALSE

$frac_disjoint
[1] 0

$first_disjoint_t
[1] NA

> check_union_equals_envelope(fit_night)
$any_disjoint
[1] FALSE

$frac_disjoint
[1] 0

$first_disjoint_t
[1] NA
```



We tried first to do this two-component approach and conformal bands because, based on the results R is giving us below it is clear there are two different clusters in each of our group, being them clearly marked and separated (as you can see in the picture below, more than 75% of all curves has a value bigger than 0.95, both for day and night).

```
> # quanto è "sicura" l'assegnazione al cluster (vicino a 1 = sicura)
> summary(apply(fit_day$gmm$z, 1, max))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.7006  0.9529  0.9918  0.9549  0.9998  1.0000 
> summary(apply(fit_night$gmm$z, 1, max))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.5125  0.9783  0.9989  0.9726  1.0000  1.0000 
> 
```

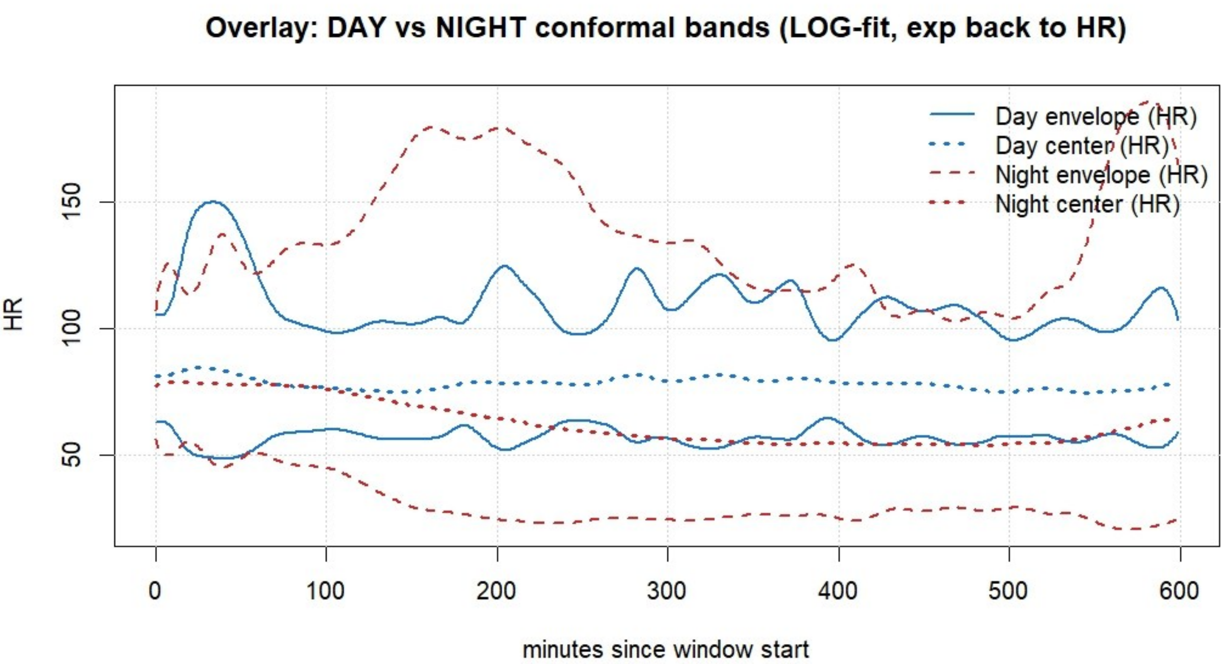
Moreover, is it important to underline that we choose two components because with two components the bic value was the smallest among 1, 2, 3 and 4 components (we tried all of them and computed the Bayesian Information Criterion, and the smallest value was found when we set component = 2, both for day and night). Below you can see the bic values we were getting.

```
[day] BIC values (model=VVV):

  G       BIC
1 1 -7209.483
2 2 -7204.887
3 3 -7266.040
4 4 -7342.736

[day] Selected G = 2 (BIC=-7204.887)
[day] Calibration inside-set fraction: 0.933 (target ≈ 0.9)
```

```
[night] BIC values (model=VVV):

  G       BIC
1 1 -7965.501
2 2 -7886.043
3 3 -7929.848
4 4 -7983.085

[night] Selected G = 2 (BIC=-7886.043)
[night] Calibration inside-set fraction: 0.938 (target ≈ 0.9)
```

Moving forward, if we study the conformal bands picture a bit more carefully, we notice couple of stuffs. One thing we noticed is that the day band is a lot narrower than the night one. This is probably due to the fact that during the day Riccardo is almost never sleeping, so there is less variability in his heartbeat records because of the simple fact that, even if that day he is moving a lot, he is not having huge jumps in his records. Matter of facts, his heartbeats during the day are most of the time between 60 and 100. Moreover, during the hours of which the records are from (09.30 – 19.30) he is almost for sure awake doing something. We have a clean 10 hours record where he is kind of constant in what he is doing. On the other hand, in the night, his heartbeat varies more, because of two main reasons. The first one is that he is not going to sleep and waking up every day at the same time. Usually, Riccardo is going sleeping between 22.30 and 00.00 (rarely at 00.30), while he is usually waking up between 7.00 (rarely at 06.30) and 08.00, but this is not a fixed rule. Because of this, we cannot say that during the night records (22.00 – 08.00) he is sleeping all the time (while we can say that during the day he is every time awake), and so we end up having a record that is not constant in what he is doing. We have a record that mixes up hours in which he may be awake (from 22.00 to 00.00 and from 06.30 to 08.00 he may be awake, while having the almost certainly that he is sleeping only between 00.30 and 06.30). Moreover, if we consider that Riccardo is also hanging out sometimes in the weekend, we cannot even be so much sure anymore that his records will find him sleeping between 00.30 and 6.30. As you can easily see, there is a huge variability in his night heartbeat data, much more than the one we have during the day. This much bigger variability of the day compared to the night is visible in the prediction bands, and actually it creates a strange situation. In the night band, as we can see from the graph, his heartbeat goes, sometimes, slightly below zero. Physically this is impossible, because the minimum speed at which his heart can run is zero, and if he had ever run at that paste (or anyways below 15bpm) probably he wouldn't even be here anymore, being most probably buried in a (he hopes nice) cemetery with some (he hopes a lot) flowers on his grave. His heart runs basically like an airplane during a flight: it can go slow but not so much, and it cannot stay still (in both situations the ending for the plane and the passenger would be as nice as turbulence… and then silence.), but you cannot for sure go at a speed smaller than zero.

So now the question you may ask is how is it possible then that the night band is below zero. The answer to this question has to be searched in the way we build the band itself. First of all, we never actually implement in our code a constraint to force the band to be positive (or in general to be in a specific range, that in our case could be 0-250, given that heartbeat cannot go below 0 and above (usually) 250, and second of all the band is basically a central curve to which we add and subtract the uncertainty (the variance) of our data. During the night, the central curve is, understandably, lower than the day one, because the heartbeat during the night is usually lower than the one during the day, but since the variance is higher, we have that the band is a lot wider. The combination of these two stuff basically created a conformal band that is going, sometimes, below zero, even if from a reality standpoint, this is impossible. In order to solve this problem, both for having a better visualization and a more real band, we decided to use a log transformation and so run our conformal prediction on a positive scale. We basically move the entire procedure to a log scale (transforming every observation using a logarithm) and then transform the results back to the original heart-rate scale. We do this with the exponential, and since the back transformation is monotone and one to one, the conformal prediction logic is still there. It is important also to note that on the original scale the conformal band is constructed by adding and subtracting an uncertainty term around a central curve, which in our case produce unrealistic negative values when the center is low and variability is high (exactly what happens in night). On the log scale, uncertainty behaves more like a relative effect rather than a purely absolute one, so the resulting band on the bpm scale becomes naturally asymmetric and stays in a plausible range. The results of this log transformation were great, and we managed to have both a better visualization and a more realistic band. Our conformal predictions are now on a positive scale and the final conformal bands we got are shown in the picture below.



As you can see from the picture, now we have a lot more realistic band. Still of course the night one is much wider than the day one but now is never going below zero. It is still going a bit too low (approximately around 10 and 25 bpm after minute 150 onword) considering that I basically never had an hearbeat lower tha 40bpm (luckly), but at least it is not going as down as the previous version was. For sure, over the previous bands, log transformation is an improvement.

We have now come to the end of our work, and before we link our mails and our contact information, we wish to sincerely thank both the professor Pierpaolo Brutti and his teacher assistant Valeria Avino. We thank the professor because he always supported us during his lessons, tying to make them as much enjoyable, interactive and interesting as he can. He managed doing it with really good results. Moreover, we will keep in our memories for a long period of time his early Friday morning lessons, that were made sweeter when, especially during strikes, he was usually bringing some sweets to the class. Coming to Valeria, we thank her as sincerely as we did to the professor for couple of reasons. The first one being her kindness. Both during and after her lessons we were asking her a lot of questions (sometimes we bothered her even in late afternoon or evening, we are sorry about that), not only related to statistics, and she was always answering us very kindly, with patience and with willingness to help. The second reason we want to thank her is because her help was never limited only to what she was getting paid for and was not limited only to Statistics. Often we asked her questions about future exams, study plans, tips and so on. She never refused to answer, and every time she did it with great details and kindness.

We are grateful to both of you, thank you so much.

Our contact information:

Riccardo Pugliese
pugliese.2264350@studenti.uniroma1.it

Leonardo Suriano
suriano.2264349@studenti.uniroma1.it

Mehmet Sener
sener.2224418@studenti.uniroma1.it