Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Cybernetics

# Analysis of *Sleep as Android* Application Users' Sleep

# Analýza spánkových vzorů uživatelů aplikace *Sleep as Android*

*Diploma Thesis*

Study programme: Biomedical Engineering and Informatics
Field of study: Biomedical Engineering

Thesis advisors: Ing. Jiří Anýž, Ing. Eduard Bakštein, Ph.D.

**Miroslav Domankuš**

Prague 2018

# MASTER'S THESIS ASSIGNMENT

## I. Personal and study details

Student's name: **Domankuš Miroslav** Personal ID number: **426112**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Biomedical Engineering and Informatics**

Branch of study: **Biomedical Engineering**

## II. Master's thesis details

Master's thesis title in English:

**Analysis of Sleep as Android Application Users' Sleep**

Master's thesis title in Czech:

**Analýza spánkových vzorů uživatelů aplikace Sleep as Android**

Guidelines:

The Sleep as Android application database of more than 10 million sleep recordings is unique source for sleep patterns analyses. The database size allows for a study of sleep patterns on population level in relationship to various influences. An important problem in the sleep patterns analysis is the internal structure in the sleep recordings caused by users' subpopulation with various social background, type of job, everyday customs or chronotypes. The goal of this diploma thesis is to identify these groups of users by clustering of users' timeseries.
Instruction:
1. Examine the approaches to clustering of timeseries data based on review of scientific literature.
2. Explore the data and perform steps necessary to the analysis - data cleaning and preprocessing.
3. Based on the review choose an appropriate data representation for the timeseries clustering.
4. Perform the timeseries clustering and present the results.
5. Develop an appropriate procedure for validation of the observed behaviour of the users.

Bibliography / sources:

[1] Montag, C., Duke, É., Markowetz, A. (2016). Toward psychoinformatics: Computer science meets psychology. Computational and Mathematical Methods in Medicine, 2016, 1-10.
[2] Walch, O. J., Cochran, A., & Forger, D. B. (2016). A global quantification of "normal" sleep schedules using smartphone data. Science Advances, 2(5), e1501705-1501705.

Name and workplace of master's thesis supervisor:

**Ing. Jiří Anýž, National Institute of Mental Health, Klecany, Czech Republic**

Name and workplace of second master's thesis supervisor or consultant:

**Ing. Eduard Bakštein, Ph.D., Analys. and Interpr.of Biomed. Data, FEE**

Date of master's thesis assignment: **09.01.2018** Deadline for master's thesis submission: _____

Assignment valid until: **30.09.2019**

| Ing. Jiří Anýž | doc. Ing. Tomáš Svoboda, Ph.D. | prof. Ing. Pavel Ripka, CSc. |
|---|---|---|
| Supervisor's signature | Head of department's signature | Dean's signature |

## III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

. _____ _____

Date of assignment receipt | Student's signature

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 24.05.2018

Miroslav Domankuš

# Acknowledgements

First, I would like to thank the developers from the *Urbandroid* team for providing the data set for analysis. Furthemore, I would like to thank Ing. Jiří Anýž for his invaluable comments and guidance in this thesis. I would also like to acknowledge professor Bülent Yılmaz from the Abdullah Gül University, who has my profound gratitude for his help and excellent advice during my Erasmus exchange.

# Abstract

The large data set of sleep recordings of the *Sleep as Android* application's users offers an excellent opportunity to study sleep patterns in a large population of users all over the world. We have analyzed the influence of various factors, for example alcohol or caffeine, on sleep. Important nation-wide events, such as presidential elections, have been observed to have significant influence on sleep parameters of users. Several findings from sleep science literature have been confirmed on this data set, which shows that collecting sleep scheduling data with this sleep tracking application is valid. Various clustering approaches and data representations have been used to find meaningful subgroups based on sleep patterns of users. Two clusters have been found to be present in the data based on clustering sleep duration time series, which correspond to the two chronotypes – evening and morning types.

# Keywords

sleep, clustering, time series, data mining, R, classification

# Abstrakt

Data set spánkových záznamů uživatelů aplikace *Sleep as Android* umožňuje studovat charakteristiky spánku ve velkém měřítku v populaci uživatelů této aplikace. V této práci byl analyzován vliv několika faktorů (alkohol, kofein a další) na spánek. Byl pozorován vliv významných událostí, jako například prezidentské volby v USA, na spánek uživatelů. Data set *Sleep as Android* umožnil ověřit některé poznatky z vědecké literatury, co taktéž potvrzuje validitu sběru spánkových dat s použitím této aplikace. Pro nalezení podskupin v časových řadách délek spánku uživatelů byly použity různé metody shlukování a několik reprezentací časových řad. V datech se vyskytují dva shluky, které odpovídají dvěma chronotypům – ranní a večerní typy.

# Klíčová slova

spánek, shlukování, časové řady, dobývání znalostí, R, klasifikace

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| AD | Average Distance |
| ADM | Average Distance Between Means |
| APN | Average Proportion of Non-overlap |
| AUC | Area Under the Receiver Operating Characteristic Curve |
| BCa | Bias-Corrected and accelerated bootstrap confidence interval |
| BMI | Body Mass Index |
| CART | Classification and Regression Trees |
| CI | Confidence Interval |
| CRAN | Comprehensive R Archive Network |
| DB | Davies-Bouldin |
| DCT | Discrete Cosine Transform |
| DFT | Discrete Fourier Transform |
| DP | Density Peaks |
| DTW | Dynamic Time Warping |
| DWT | Discrete Wavelet Transform |
| EDA | Exploratory Data Analysis |
| EEG | Electroencephalography |
| EM | Expectation Maximization |
| EMG | Electromyography |
| EOG | Electrooculography |
| ET | Evening Type |
| FOM | Figure of Merit |
| GAM | Generalized Additive Model |
| GMM | Gaussian Mixture Model |
| kNN | k Nearest Neighbors |
| L2 | Euclidean Distance |
| LCB | Lower Confidence Bound |
| LDA | Linear Discriminant Analysis |
| MEQ | Morningness-Eveningness Questionnaire |
| MT | Morning Type |
| NPV | Negative Predictive Value |
| NT | Neither Type |
| PAA | Piecewise Aggregate Approximation |
| PAM | Partitioning Around Medoids |
| PPV | Positive Predictive Value |
| REM | Rapid Eye Movement |
| SBD | Shape-Based Distance |
| SP | Seasonal Profile |
| TADPole | Time-series Anytime Density Peaks |
| UCB | Upper Confidence Bound |
| UTC | Coordinated Universal Time |

# 1 Introduction

The goal of this thesis is to apply statistical methods and methods of machine learning to a large data set of sleep recordings for the purpose of quantification of sleep on a large scale. The large data set of sleep recordings provided by the Urbandroid team offers numerous possibilities to study sleep parameters, including parameters reflecting sleep quality and sleep patterns in a large population of users all over the world. This database of sleep recordings suffers from the usual quirks of large databases, such as incorrect entries or missing data. To uncover the underlying structure of the data set we use a number of visualizations, statistical and machine learning methods. Several interesting questions about sleep and various factors that can affect sleep parameters are addressed. Validity of several findings about sleep supported by previous research are tested on this data set, what also serves as validation for this sleep data collection method. We address the question of finding meaningful user subgroups based on sleep patterns of users throughout the year. Methods of time series clustering are used for this purpose.

First, a short introduction to sleep science is presented. Subsequently, after describing the analyzed data set and methods that were used in analysis, we move on to the application of these methods to the data set.

## 1.1 Sleep

In this section, a short review of sleep and factors that can affect sleep is presented. Several findings presented in this section are examined and their validity is tested in the *Sleep as Android* data set in section 5. Some of the other sections also rely on the findings presented in this section.

Behavioral criteria defining sleep include suppressed cognitive functions, decreased motor activity and elevated arousal thresholds [10]. While it is clear that sleep is necessary for the human brain to function properly, the main reasons why sleep occurs and why it has developed in humans are not well agreed on. A number of theories have been proposed. It has been proposed, for example, that sleep may serve the purpose of saving energy [12], which seems unlikely due to the relatively small amount of saved energy and the large cost of losing consciousness and being vulnerable to possible threats from the outside world. Another theory trying to explain why sleep should occur is the removal of neurotransmitters from interstitial fluid during sleep [13]. It was also shown that REM sleep (see section 1.1.1) contributes to detaching emotional experiences from memories of the past [22].

Sleep is not a phenomenon unique to humans and it has been observed in most studied animals [10]. It is well known that sleep deficiency can lead to various health problems and in extreme cases of sleep deprivation even to death [11].

### 1.1.1 Sleep stages

Sleep occurs in different stages, which are all characterized by behavioral changes and changes observable in several biological signals when measured during sleep. Sleep stages can be accurately determined by the use of polysomnography – measurement of several biological signals during sleep. Stages that occur during sleep are: REM (rapid eye movement), N1, N2, N3

- non-REM (NREM) sleep. Each of the sleep stages is characterized by different manifestations in polysomnographic measurements and effects on physiological functions. Manifestations of these stages in a polysomnogram in the most basic set-up, where only three signals are measured (EEG - electroencephalography, EOG - electrooculography, EMG - electromyography), are shown in Fig. 1.1.

Electroencephalography measures changes in electric potential on the scalp resulting from the brain's electrical activity. The movement of eyes is recorded by electrooculography by the means of measuring the electrical activity near the eye. Finally, electromyography records electrical activity of muscles and can be used to evaluate movement and tension of the measured muscles.

The state of wakefulness is characterized by alpha and beta waves in the EEG with signal energy at theta and delta frequencies being low. Eye movements and muscle tension is present, as can be seen from the electrooculographic and electromyographic measurements. The first sleep stage that occurs during normal sleep is the N1 stage in which theta waves can be seen to be present in the EEG and slow eye movements with lowered tension of the muscles is observed. [7]

In the next sleep stage N2, eye movements are no longer present. The muscle tension remains somewhat lowered and so called K-complexes and spindles (see Fig. 1.1) can appear in the EEG signal recording. The next stage N3 is somewhat similar to N2 stage, but waves at even lower frequency (delta) are present in the EEG recording. Finally, in the rapid eye movement (REM) sleep stage, muscle tension is lowered even beyond the point of NREM sleep and as the name suggests, bursts of quick eye movements can be observed during this stage. [7]

This cycle of changing sleep stages repeats several times during the night and for normal duration of sleep occurs on average around five times in one night, while rapid eye movement sleep stages tend to get longer throughout the night [7].

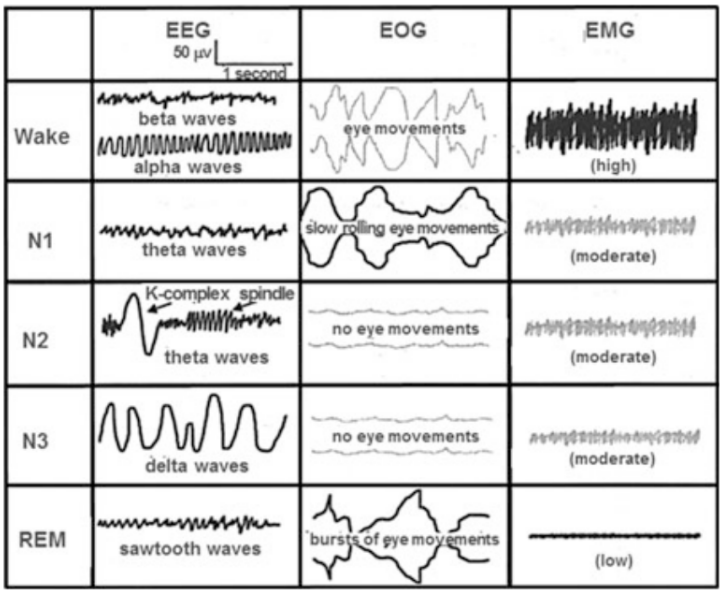



Fig. 1.1: Characteristics of signals in sleep stages measured during polysomnography. Taken from: [7]

### 1.1.2 Factors affecting sleep

**Age** The recommended duration of normal sleep based on observed sleep durations varies with age and is the highest for newborns and young children and gets lower with increasing age, with seniors being recommended the shortest sleep periods. Need for sleep varies in different age groups, with average adult between 20-50 years needing 7.5–8.5 hours of sleep [7].

Elderly people report shorter sleeping periods (around 6–7 hours) in the night, compared to their sleep durations when younger [7]. Studies also show that elderly people have more regular

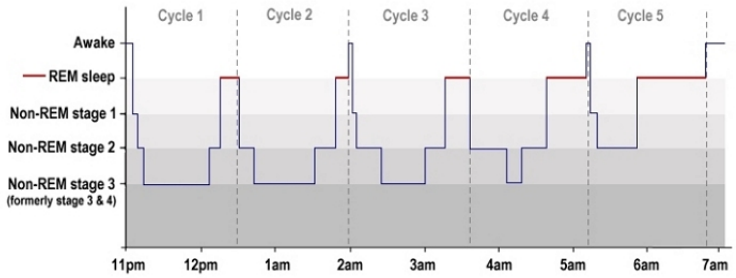



Fig. 1.2: Sleep stage sequence through the night. Taken from: [48]

sleep scheduling than young people [3] and tend to go to sleep earlier and wake up earlier [6], thus showing that the chronotype (see paragraph *Chronotype* in this section) of a person can also change with age.

In a study that analyzed sleep data obtained by the use of a smartphone application, age had the most dominant influence on determining the sleep timing (midsleep – the time in the middle between bedtime and wake time). The study also reported earlier occurrence of sleep (bedtime) in the age group of 18–19 years compared to the age group of 20–24 years [2].

**Chronotypes** In some individuals, the circadian rhythm is phase shifted, resulting in either highly delayed or advanced sleep period compared to other people. Chronotypes are usually determined in studies by the use of questionnaires (e.g. Morningness-Eveningness Questionnaire (MEQ)).

There are two chronotypes depending on the direction of the phase shift of sleep cycle. People with highly delayed sleep periods are called evening types (ETs, commonly called owls), whereas people with advanced sleep periods are called morning types (MTs, commonly called larks). Around forty percent of population belongs into one of these categories, i.e. forty percent of population displays behavior that could be attributed to one of the chronotypes. The rest of the population does not belong to either of these chronotypes and is called neither type (NT) [7].

It is known that evening types can be more likely to build up a "sleep debt" during weekdays, since they tend to schedule their sleep later, but cannot schedule a later wake time, because of work or other responsibilities during work. This results in reduced sleep duration during weekdays and accumulation of sleep debt during week. Evening types thus tend to sleep longer on weekends to compensate for their sleep debt accumulated throughout the week[7].

It was shown that morningness and eveningness depend on factors such as age, sex, exposure to light, altitude and latitude of residence and even photoperiod (time between sunrise and sunset) at birth [6]. People born in autumn or winter (during the period of year with shorter photoperiod) are more likely to be morning types, whereas people born in spring and summer (during the period of year with longer photoperiod) are observed to be evening types more often, while this difference was found to be more pronounced in males. [6].

Chronotype of a person can change significantly throughout their lifespan, with morningness decreasing from early childhood into early adolescence. For example in [4] eveningness reached its peak at approximately 16 years of age for girls and 17 years of age for boys. After this age, eveningness decreases throughout lifetime with the elderly having a significantly more pronounced morningness [3] than other age groups.

**Gender** Gender differences in sleep had been reported in literature. Women schedule more sleep in nearly every age group. The difference between genders was most pronounced in the age group 30–60 years in [2]. Chronotype frequency had been also found to differ among the two genders. Larks (morning types) were found to be more common among females, whereas owls

(evening types) were found to be more typical among males [6]. It has also been observed that women tend to schedule sleep earlier with a longer sleeping period than men [7].

**Circadian rhythm** A study comparing the expected effect of sun (using a mathematical model of the ascending arousal system coupled to a model of the circadian clock) on sleep scheduling to actual sleep scheduling data suggests that the effect of sunset on sleep in the evening is weaker than expected. It has been proposed that these effects are ignored in real life due to social pressure and use of artificial lighting. Influence of sunrise and sunset on wake times and bedtimes was shown to be particularly strong in some subgroups, mainly women, older people and people that reported outdoor lighting as typical for them. [2].

These findings are in contrast with a study examining sleep patterns in three preindustrial societies and shows that sleep patterns in these groups are not that different from those of "modern" humans with access to artificial lighting [14]. This study, however, found the temperature to be a major predictor of sleep scheduling, with increase in sleep duration by about one hour in the winter. The authors argue that this influence of temperature can be suppressed in modern society as a result of indoor temperature being relatively constant during the year in households. It must be noted, however, that the small sample of people living in these societies may not be a representative group of people without lighting and social pressures, and that the results may not provide an accurate picture of sleep scheduling in the whole population without artificial lighting or social pressures.

**Alcohol** Alcohol is commonly known for its relaxing effect on the body. Research suggests that ingestion of alcohol before bedtime affects sleep in unfavorable ways. It has been confirmed that it causes a reduction in sleep latency, but delays the onset of the first REM period and decreases the overall percentage of REM sleep [15]. Because of its muscle-relaxing effect, alcohol may also cause more intensive snoring during the night.

**Caffeine** Caffeine is known to increase sleep latency, decrease total sleep duration and reduce the percentage of N3 and N4 sleep [17]. These effects are more pronounced for larger doses of caffeine.

**Other** Summary of different variables that can also have an effect on parameters of sleep is shown in Fig. 1.4.

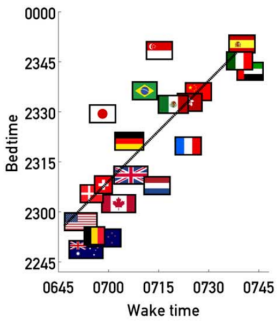



Fig. 1.3: Mean wake time and bedtime by country. Taken from: [2]

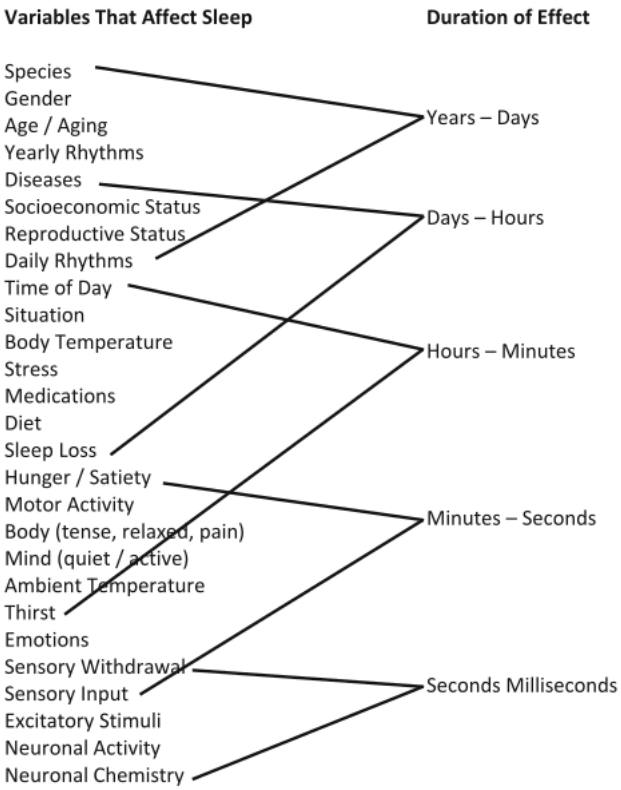



Fig. 1.4: Factors affecting sleep. Taken from: [7]

A country to country difference in sleep timing has been observed, where country was found to be more important in determining bedtimes than wake times [2]. Bedtimes and wake times for various countries are shown in Fig. 1.3.

Another important variable in determining sleep duration and intensity of sleep is the time spent awake since the last sleep period before sleeping. The longer the awake time after last period of sleep is, the longer the subsequent sleep duration will be (this is also called homeostatic sleep drive) [7]. Periods in which an individual did not get enough sleep, cause a heightened urge to sleep even during times of day normally not usual for the individual and subsequent prolonged sleep. This phenomenon is called sleep rebound.

# 2 Data

In this chapter, the data set of sleep recordings used in this thesis is introduced.

Sleep patterns are usually studied in a controlled environment using polysomnography and questionnaires on sleeping habits. However, with now ubiquitous smartphone devices and increasing popularity of sleep tracking applications recording data, it has become possible to analyze sleep patterns of a very large population of their users. In this thesis, data from one such application are analyzed.

Data analyzed in this thesis were provided by the Sleep as Android application's developers from the Urbandroid team. The data are anonymized and contain information about the application's users and their sleep parameters.

## 2.1 *Sleep as Android* application

*Sleep as Android* is an application available for the *Android* operating system which allows users to track their sleeping habits. For this purpose, user's movement during sleep is measured either via built-in smartphone sensors or using external wearable devices. In the case when sensors in the user's smartphone are used, the phone should be placed on the bed close to the user and activity of the user during sleep is tracked using either the phone's accelerometer or using the "sonar" function. Users can also opt to record sound during the night and thus keep track of their snoring or sleep talking.

The application tries to automatically detect the user's sleep stages, namely Light sleep, Deep sleep and REM, and tries to wake the user in the Light sleep period which is supposed to help the user wake up more easily. The users have to start the recording after lying down and stop the recording after waking up themselves. The recording can be set to start after a certain period which can be set by user.

The application also provides other functions such as the lullaby function in which the user can listen to a chosen sound track before falling asleep which is supposed to help them relax while falling asleep and reduce their sleep latency (the time it takes to fall asleep after lying down).

Statistics of various parameters of sleep are provided by the application for users to keep track of their sleeping habits. After a user chooses the desired duration of sleep for each day, the application notifies them to go to sleep at the ideal time calculated from the time of the alarm set for the following day. The application also calculates "sleep debt" as the difference between desired and actual sleep duration.

After a sufficient number of sleep recordings is accumulated, advice on modifying the sleep duration and bed time to improve sleep quality and the ratio of deep sleep is given based on regression models. Users can also set goals they want to accomplish, such as increasing the average sleep duration by a chosen percentage in a chosen number of days or improving sleep scheduling regularity.

## 2.2 Recorded parameters

The basic list and explanation of the available parameters in the original data set is included in this section. A more detailed look on the recorded values and their distributions is given in section 4. The original data set contains 15 905 401 recordings of these 24 parameters:

**Recorded parameters**

- *id* – unique identification number of sleep recording
- *userId* – unique anonymized identification number of user
- *timeZone* – time zone set on the user's device
- *from, to* – time of the start and the end of recording (all of the times are encoded as a number of milliseconds from 01.01.1970 00:00 UTC)
- *commentTags* – contains optional sleep tags or comments entered by the users. Some of the tags are generated automatically. See section 5.3 for more on *commentTags*.
- *avgNoiseLevel* – average noise during recording
- *noOfCycles* – total number of detected sleep cycles
- *snoringTime* – time spent snoring in seconds
- *subjectiveRating* – user's subjective rating of sleep quality. Contains values between 0 and 5.
- *deepSleepRatio* – ratio of deep sleep duration to total sleep duration
- *geoLatitude, geoLongitude* – fields containing the user's location
- *netSleepLength* – total sleep duration in minutes
- *gender* – contains values "MALE", "FEMALE" or "" (not known)
- *height, weight* – height in centimeters, weight in kilograms
- *birthdate* – date of birth
- *alarmTime* – time of set alarm
- *netSleepAdjustment* – duration of intervals in which the user did not sleep during recording (includes duration of paused recording and the standard fall asleep period, if it is set by user)
- *civilSunrise, civilSunset* – times of sunrise and sunset
- *device* – names of users' smartphone devices are recorded
- *sleepStart* – time of the first detection of sleep by the software

Apart from these values, actigraph and noise recordings were available for each sleep recording. A data set of events generated by the software was also available. These data sets were, however, not used for the purpose of this thesis.

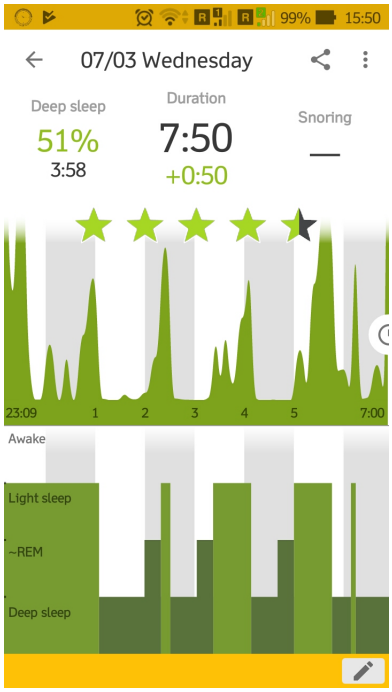



Fig. 2.1: Screenshot from the *Sleep as Android* application. The recorded actigraph and calculated sleep cycles are shown.

*Sleep as Android*'s output after a night of sleep is shown in Fig. 2.1. In the upper part, parameters such as total sleep duration, difference in sleep duration and the desired duration set by user, ratio of deep sleep and snoring time are shown. Users can rate their perceived quality of sleep using the stars shown in the figure. The application also shows the recorded actigraph and sleep stages detected from the actigraph. Users can add comments to the sleep recording using the button in the right bottom corner.

## 2.3 Remarks on data quality

One of the advantages of this approach to sleep data collection is that a very large data set of sleep recordings is obtained easily. It is therefore possible to analyze sleep patterns of the population of users worldwide and on a large scale. Another advantage of this kind of data collection methods is that the collected sleep recordings come from "real world" (in situ) conditions. This, at the same time, may be considered a disadvantage. Since these data do not originate from a controlled environment, the researcher does not have control over the conditions of the recording. Incorrect use or other factors such as different sensors being used among various smartphone or wearable devices may be another cause for inaccuracies.

It must be noted that as the sleep parameters were measured only by the use of actigraphy, the accuracy of these calculated parameters may be low compared to polysomnographic experiments. The actigraphic measurements were not validated against polysomnographic recordings and studies show that detection of sleep stages by actigraphy by some health monitoring devices can yield poor results [42].

The method by which the sleep recordings are obtained may also be a source of inaccuracy in the measurements. For example, actigraphic recordings obtained from smartphone sensors with the smartphone placed on bed may be inaccurate in certain situations, such as when more than one person sleeps in the same bed, the person sleeps with a pet, etc. We will, therefore, abstain from using the parameters which rely on the actigraphic measurement in analysis and focus more on the parameters related to sleep scheduling and information about users.

Missing data is also one of the problems of the data set. Some parameters, mainly the ones relating to users' demographics, contain large proportions of missing data.

Big emphasis must therefore be given on the data preprocessing step in order for the analysis to be possible and to avoid false conclusions.

## 2.4 Previous studies

A short summary of previous studies working with similar data sets to uncover sleep patterns in population is presented.

A study working with perhaps the most similar data set obtained from a smartphone application is *A global quantification of "normal" sleep schedules using smartphone data* [2]. In this study, data from a smartphone application called ENTRAIN are used. The application collects data in a form of a questionnaire, where users can record their normal sleep times, typical lighting, time zone, subjective jet-lag experiences etc. The study analyzed reported sleep patterns of 5 450 users in total and used multivariate regression models to model sleep parameters wake time, bedtime, sleep duration and midsleep based on predictors gender, age, typical lighting, sunrise, sunset, travel frequency and country. Focus is given mainly on characterizing sleep and variability of sleep parameters among different countries, age groups and genders. One of the main conclusions of this study is related to "social jet-lag" which can be thought of as a disruption of circadian rhythm caused by social pressures or habits prevalent in society. By comparison with a model of circadian rhythm, it is concluded that social pressures cause people to delay their bedtime and thus shorten their total sleep duration. Moreover, the study shows that mobile technologies are a viable source of collecting sleep data.

Another study *Harnessing the Web for Population-Scale Physiological Sensing: A Case Study of Sleep and Performance* [20] analyses data obtained from wearable devices (Microsoft Band) to relate sleep parameters to cognitive performance, measured through interactions with a search engine. The study analyses a data set of 3 million nights from 31 thousand users recorded with wearable sensors. The study uses the variability of sleep duration by gender and in different age groups to validate its data collection method. The study demonstrated that performance varies throughout the day and is related to chronotype and prior sleep, in close agreement with small-scale laboratory-based studies [20].

# 3 Methods

Methods that are going to be used to analyze the data set in the later chapters are presented in this chapter.

## 3.1 Data mining

For the purpose of uncovering the structure of the data and relationships between parameters, methods of data mining are used. There are plenty of definitions of data mining with minor differences between them, but in general it can be said that data mining is a process which discovers knowledge from data using statistical and machine learning tools. Two primary goals of data mining are prediction and description. The predictive approach to data mining focuses on creating models of the data and applying these models to the prediction of new data observations. The second approach is descriptive data mining which tries to come up with new information based on the available data [1]. Most of this thesis is concerned with description of the data set, but some predictive tasks are also addressed.

## 3.2 R

Most of the statistical and machine learning algorithms used in this thesis have been programmed in the *R* programming language [40] with the use of packages from the *CRAN* archive (see References section).

Because the original data set is a relatively large file (almost 4GB), some computations on this data set could not be performed with the data in memory. The *ff* package [18] in *R* was used to work with the original data set. This package introduces new data structures, which allow working with data that are stored on disk, but behave as if they were stored in RAM in *R*.

## 3.3 Hypothesis testing

Procedures that were used for hypothesis testing are described in this section. Hypothesis tests for equality of means of two samples are used in this thesis and presented in this section.

One of the most commonly used procedures to test whether the means of two distributions are equal is the Student's $t$-test. The $t$ statistic for two samples $x_1$ and $x_2$ of size $n_1$ and $n_2$ respectively is calculated as [43]

$$t = \frac{\overline{x_1} - \overline{x_2}}{s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{3.1}$$

where $\overline{x_1}$ and $\overline{x_2}$ are the sample means of the two samples and $s$ can be obtained from the equation

$$s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2} \tag{3.2}$$

where $s_1$ and $s_2$ are estimates of variance in the two groups. The $t$-test assumes normal distribution and equal variance in the samples. However, the $t$-test is very robust to the normality assumption for large samples even for highly skewed distributions (see [44]). The size of the sample does not, however, play a big role when the equal variance assumption is violated.

Instead of the Student's $t$-test, we therefore use the Welch's $t$-test also known as unequal variance $t$-test. As the name suggests, this test does not rely on the assumption of equal variances and works well even if the variances are equal [45], it is therefore preferred to the Student's $t$-test. The $t$ statistic for the unequal variance $t$-test is calculated as

$$t = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{3.3}$$

When the assumptions for the previous tests are violated, non-parametric tests can be used. One such test, which will be used later on in this thesis, is the non-parametric bootstrap hypothesis test which does not make any assumptions about the underlying distribution of the tested parameter. Bootstrapping in statistics refers to methods which use resampling techniques with replacement. To test equality of means using the bootstrap, first the samples are centered to their combined mean $\overline{x}$ (sample mean calculated from samples $x_1$ and $x_2$ combined)

$$\tilde{x_1} = x_1 - \overline{x_1} + \overline{x} \tag{3.4}$$

$$\tilde{x_2} = x_2 - \overline{x_2} + \overline{x} \tag{3.5}$$

We obtain $B$ bootstrap data sets $x^*_{1b}$ and $x^*_{2b}$ (for $b = 1, 2, \ldots, B$) of the same size as $x_1$ and $x_2$ respectively by sampling $\tilde{x_1}$ and $\tilde{x_2}$ with replacement. This means that the bootstrap samples are sampled under the null hypothesis of equal means. For each bootstrap data set, the $t$ statistic for unequal variances is evaluated

$$t(b) = \frac{\overline{x^*_{1b}} - \overline{x^*_{2b}}}{\sqrt{\frac{s^{*2}_{1b}}{n_1} + \frac{s^{*2}_{2b}}{n_2}}} \qquad b = 1, 2, \ldots, B \tag{3.6}$$

The distribution of $t(b)$ approximates the distribution of the parameter if the means are equal. The resulting $p$-value is thus calculated as

$$p_{boot} = \frac{\#\{t(b) \geq t_{obs}\}}{B} \tag{3.7}$$

where $\#\{\cdot\}$ denotes the number of cases where the condition is satisfied and $t_{obs}$ is the observed $t$ statistic calculated from the original samples.

**Multiple hypothesis testing and data dredging** The relatively high number of recorded parameters and a large number of comment tags used by users to tag their sleep give rise to a huge set of subgroups and hypothesis tests of difference of sleep parameters in these subgroups that could be performed. However, if such a number of hypothesis tests was performed, some percentage of these tests (depending on the confidence level used) would reject the null hypothesis as a result of pure chance. The practice of testing a large number of hypotheses and choosing the ones which rejected the null hypothesis is called data dredging and can lead to false conclusions and measures should be taken to avoid this practice. In this thesis, a lower cutoff $p$-value ($\alpha = 0.001$) is therefore generally used for hypothesis testing and corrections of confidence levels for multiple testing are also used. The Bonferroni correction is used which corrects the $\alpha$ value to $\alpha/m$ for $m$ tested hypotheses. Testing hypotheses suggested by the data is being avoided in this thesis and all hypotheses are formed prior to

testing.

## 3.4 Classification

### 3.4.1 Introduction

Classification is an approach where observations are classified into two or more classes based on values of other parameters (features) as accurately as possible. A classifier takes features as input and outputs the class based on the values of the features. To be able to differentiate between classes, the classifier must find a decision boundary based on which it will give its decisions. The process of finding such a decision boundary is called training of a classifier. For the purpose of classifier training, a subset of the data set is chosen – the training set. The classifier tries to find a decision boundary which separates the classes well and its goal is to achieve the maximum accuracy of classification. The classifier must be able to generalize and give reasonable classification results for new data. There are numerous methods by which the ability of the classifier to generalize can be assessed:

1. *Holdout method.* In the most basic case, data set is divided into two smaller data sets – the training and testing data set. To avoid too optimistic estimates of accuracy on the testing data set, a validation data set, on which the classifier's output is evaluated when optimizing the parameters of a classifier, can be included. The classifier with the best result on the validating data set is then tested on the testing set. This results in more realistic estimates of classification accuracy on new data, since the testing set was not involved in the process of choosing the classifier's parameters.
2. *k-fold cross validation.* The data set is randomly divided into $k$ distinct samples. In every one of $k$ iterations, one sample is used as a testing set and the other ones are used for training. A mean accuracy can then be calculated. In the extreme case when $k$ is equal to the number of observations in the data set, this method is known as *leave-one-out*. This method, however, has large computational requirements.
3. *Bootstrap.* The bootstrap is a resampling technique which generates a certain number of sets of the same size as the original set by resampling the original data set with replacement.

Since the number of observations in this thesis is generally very high, we use hold-out methods to estimate performance of classifiers.

There are numerous available classification algorithms and none of them is the most suitable for all applications (this is known as the *no free lunch theorem* in machine learning). Therefore, different approaches must be tried and compared for every classification problem.

### 3.4.2 Feature selection

In some cases, the algorithms used for classification can benefit from reducing the number of features and classifier performance can be increased when some of the features are left out. This is because some features can be irrelevant or can contain noisy information. Also, if the number of features is high, the *curse of dimensionality* can occur. There are three main approaches to feature selection:

1. *Filtering methods.* Metods which select features based on a measure calculated from the data i.e. features are selected prior to classification
2. *Wrapper methods.* Methods which use the output of a classification algorithm to determine the subset of features that leads to the best performance of the classifier. Different subgroups of features are selected and the output of the classifier is examined. Since the number of

possible subsets is generally too high, greedy algorithms are usually used, what can lead to non-optimal solutions.

3. *Embedded methods.* Methods where feature selection is part of the classifying algorithm

**Relief** In this thesis, the *Relief* algorithm for feature selection was used. The algorithm evaluates a function for each feature in order to determine the quality of the feature, without directly optimizing a classification model. It is therefore a *filtering* method. *Relief* is scalable for data sets containing large number of samples and data sets with high dimensionality, while also being unaffected by noise and feature interaction. The algorithm, however, cannot help with removing redundant (highly correlated) features [1].

*Relief* algorithm randomly chooses $m$ observations from the training data set, where $m$ is a user-defined parameter. Based on this subset a quality score $W$ for $i$-th feature $A_i$ is calculated as

$$W_{new}(A_i) = W_{old}(A_i) - ((X[A_i] - H[A_i])^2 + (X[A_i] - M[A_i])^2)/m \tag{3.8}$$

where $X$ is an observation from the randomly chosen subset, $H$ is the nearest hit (nearest belonging to the the same class) and $M$ is the nearest miss (nearest observations belonging to a different class). $W$ is initialized as zero and updated for each observation from the $m$ chosen observations.

Based on the calculated values of $W$, features which have $W$ over a selected threshold can be selected.

### 3.4.3 Measures of classifier performance

Measures of classifier performance that were used for the purpose of this thesis are defined in this section using the numbers in the contingency table Tab. 3.1.

| Pred \ Ref | 0 | 1 |
|---|---|---|
| 0 | A | B |
| 1 | C | D |

Tab. 3.1: Contingency table. Pred – Predicted class, Ref – Reference class

*Accuracy* of classification is the ratio of the number of observations classified correctly to the number of all observations.

$$Accuracy = \frac{A + D}{A + B + C + D} \tag{3.9}$$

*Sensitivity* reflects the accuracy of classifying the positive (1) class and is calculated as

$$Sensitivity = \frac{D}{B + D} \tag{3.10}$$

*Specificity* reflects the accuracy of classifying the negative (0) class and is calculated as

$$Specificity = \frac{A}{A + C} \tag{3.11}$$

*Positive predictive value (PPV)* and *negative predictive value (NPV)* are calculated as

$$PPV = \frac{D}{D + C} \tag{3.12}$$

$$NPV = \frac{A}{A + B} \tag{3.13}$$

*Balanced accuracy* is the average of sensitivity and specificity

$$BalancedAccuracy = (Specificity + Sensitivity)/2 \tag{3.14}$$

The area under the curve ($AUC$) is calculated as the are under the ROC (receiver operating characteristic) curve. ROC is created by plotting the false positive rates (C in 3.1) against the true positive rates (D in 3.1) for various decision thresholds. AUC is used in this thesis for comparison of classifiers.

### 3.4.4 Classifier algorithms

***k*-nearest neighbors (*kNN*)** *kNN* is one of the most basic classifier algorithms which does not need training. The classifier instead memorizes all of the training data and calculates the nearest $k$ nearest neighbors (using Euclidean distance) to the input data. The class of the input observation is then determined by a majority vote of the classes of the nearest neighbors.

**Logistic regression** Logistic regression models the probability that an observation belongs to a certain class. In classification problems with two classes, the classes are encoded as 0 and 1. Rather than modeling a linear relationship, logistic regression models the *logistic function*. Logistic function for variables $x_1$, ..., $x_p$ and regression coefficients $\beta_0$, ..., $\beta_p$ has the form:

$$p(x_1, \ldots, x_p) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}} \tag{3.15}$$

The logistic function takes on values between 0 and 1 for all values of $x$. The regression coefficients $\beta_i$ are determined using maximum likelihood. [35]

**Naive Bayes** Naive Bayes classifier tries to estimate the probability that observation $x_1, ..., x_p$ belongs to the class $c$: $P(C = c|X_1 = x_1, \ldots, X_p = x_p)$ [26] using the Bayes theorem:

$$P(C = c|X_1 = x_1, \ldots, X_p = x_p) = \frac{P(X_1 = x_1, \ldots, X_p = x_p|C = c)P(C = c)}{P(X_1 = x_1, \ldots, X_p = x_p)} \tag{3.16}$$

The probability $P(C = c)$ can be estimated from the training data set as the ratio of observations belonging to class $c$ to the number of all observations. The classifier is called "naive", because it assumes that the variables $X_1, ..., X_p$ are conditionally independent of each other and therefore the probability can be calculated as

$$P(C = c|X_1 = x_1, \ldots, X_p = x_p) = \frac{\prod_{i=1}^{p} P(X_i = x_i|C = c)P(C = c)}{P(X_1 = x_1, \ldots, X_p = x_p)} \tag{3.17}$$

If the feature variables are categorical, the probability $P(X_i = x_i|C = c)$ can be determined simply as the ratio of observations with the value $x_i$ in the class $c$ to the number of observations in class $c$. To avoid problems with zero probability when no such observation is available, parameter $\alpha$ can be used (Laplacian smoothing). Parameter $\alpha$ is added to the numerator and $\alpha m_i$, where $m_i$ is the number of distinct values of variable $x_i$, is added to the denominator.

To generalize this classifier to numeric variables, the variable can be either discretized or a probability distribution is assumed for the data. Most commonly the Gaussian distribution is used and its parameters mean and variance are estimated from the training set as the mean and variance of the variable in the class.

**Linear discriminant analysis (LDA)** Similarly to the Naive Bayes classifier, LDA tries to estimate the probability that observation $x_1, ..., x_p$ belongs to the class $c$: $P(C = c|X_1 = x_1, \ldots, X_p = x_p)$ using the Bayes theorem. Linear discriminant analysis does not rely on the assumption of independence of variables $X_1, ..., X_p$. The conditional probability $P(X_1 = x_1, \ldots, X_p = x_p|C = c)$ is instead assumed to have multivariate Gaussian distribution. Parameters of the Gaussian distribution and prior probabilities $P(C = c)$ need to be estimated. Observations are then classified to the class for which the estimated probability is the highest.

Linear discriminant analysis is more stable than logistic regression in some cases and can classify observations to more than two classes without any extensions. [35]

**Decision trees** Decision tree is a method of classifying data by dividing the sample space into regions based on the values of the input variables by the use of decision rules. A typical univariate tree consists of nodes at which one of the features is tested. Outgoing branches must cover all of the possible outcomes of the tested feature. The first node is called the root node and the terminal nodes of the tree, which assign the observations to a class, are called leaf nodes.

There are numbers of algorithms for creating decision trees which vary in how the splits are generated, the method used for handling of missing values, pruning, etc. In this thesis the CART (Classification and Regression Trees) algorithm is used.

CART uses the Gini index to evaluate the quality of a split. The Gini index for a set $S$ is calculated as [1]

$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2 \tag{3.18}$$

where $c$ is the number of classes and $p_i$ is the fraction of the observations which belong to the $i$-th class in the set. It is a measure of partition purity in such a way that partitions favoring one class will result in lower values of the index. The quality of the split is calculated as a weighted sum of Gini indexes of the resulting $k$ subsets

$$Gini_{split} = \sum_{i=1}^{k} \frac{|S_i|}{|S|} Gini(S_i) \tag{3.19}$$

Split with minimum value of $Gini_{split}$ is chosen.

One of the main advantages of CART is its robustness to outliers [1].

#### 3.4.4.1 Ensemble learning

Ensemble methods in which not one classifier, but a combined decision of more classifiers determines the resulting class, may in some cases greatly increase the performance of a classifier. There are various approaches to generating the individual classifiers which will be combined to an ensemble and can be divided into these categories [1]:

1. each classifier is trained using a different classification algorithm
2. each classifier is trained using the same classification algorithm with different parameters
3. classifiers are trained using different input representations (e.g. subsets of input features)
4. classifiers are trained using different subsets of input data

**Bagging** Bagging (or bootstrap aggregating) trains classifiers using different subsets of input data. The training subsets are created using bootstrap sampling where sets of the same size as the original set are obtained by the use of random sampling with replacement. The classifiers' combined decision is determined by voting.

Bagging can lead to improved classification performance when unstable classifiers with small changes in the training set resulting in large changes in performance are used [1].

**Random Forest** Random forest also uses bootstrap samples for training its classifiers. The classifiers are decision trees, but only a random subset of features is considered when generating each split.

**Boosting** Boosting uses only one set for training its classifiers. Classifiers (weak learners) are added and the observations in the training set are weighted iteratively so that the newly added classifiers are focused more on the previously misclassified observations.

This approach to ensemble learning can be very effective, but can also be sensitive to outliers [1].

## 3.5 Clustering

### 3.5.1 Introduction

Clustering is an unsupervised data mining method, where no target variable is specified and the algorithm tries to find subgroups based on all of the present variables. Clustering tries to find subgroups or clusters in the data, such that observations present in one cluster should be as similar as possible, while observations from different clusters should be as dissimilar as possible. The number of clusters is determined by the algorithm itself, or is an input to the algorithm calculating the clusters. While there are plenty objective measures of clustering quality, the clustering should contain subgroups which help subjective interpretation of data and improve understanding of the data problem at hand.

There are lots of clustering algorithms which use different approaches to the clustering problem and are more suitable for certain types of data or finding clusters of different shapes. Clustering is said to be crisp, when each observation belongs to just one cluster, while in fuzzy or soft clustering, observation can be in more clusters at once with a probability of assignment to each cluster.

In the following sections, first, the problem of clustering static data is addressed. Some of the commonly used methods are summarized in the next section, as these algorithms can be in their modified form, used for clustering time series. The next section summarizes the most common methods for clustering data with static observations.. Then the problem of clustering time series is presented.

### 3.5.2 Static data clustering

1. partitioning
    (a) crisp
    (b) fuzzy
2. hierarchical
    (a) agglomerative
    (b) divisive
3. density-based
4. model-based
    (a) statistical
    (b) neural networks

**1. partitioning** The most common partitioning algorithm, which is also one of the most well-known clustering algorithms, is the $k$-means algorithm. $k$-means is an iterative algorithm

which is suitable for finding spherically shaped clusters. $k$-means creates a crisp partitioning of the data, meaning that every observation belongs to exactly one cluster. The number of cluster $k$ has to be specified prior to the calculation. There are, however, number of methods for choosing the most suitable number of clusters. The $k$-means algorithm can be summarized as follows:

1. choose $k$ random observations which are called centroids
2. calculate Euclidean distance of every observation from the centroids
3. assign every observation to the closest centroid
4. calculate means in every cluster and set them as the new centroids
5. repeat 2.-4. until the assignment of observations does not change

It can be shown that $k$-means algorithm minimizes the average inter-cluster distance of all clusters ($CAD$):

$$CAD = \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} \|x_{ki} - x_{kj}\|^2 \tag{3.20}$$

where $K$ is the number of clusters and $n_k$ is the number of observations belonging to the cluster $k$.

A similar partitioning method, which is more robust, is called $k$-medoids. The most common algorithm to compute $k$-medoids is a greedy algorithm called PAM (partitioning around medoids). The algorithm can be summarized as follows:

1. choose $k$ random observations which are called medoids
2. for each medoid and for every other observation, swap the medoid and the observation and calculate the cost function
3. if the cost increased, undo previous step
4. repeat 2.-3. while the cost decreases

The advantage of $k$-medoids is that the resulting medoids are observations from the original data set, as opposed to centroids in $k$-means, which are means of observations in the clusters, which in some cases may lead to better interpretability of clusters.

**2. hierarchical** Hierarchical clustering methods can be either agglomerative or divisive. In the agglomerative approach, first, every observation is considered to be one cluster. The most similar clusters are then merged until only one cluster remains. The advantage of hierarchical methods lies in that the number of clusters does not need to be specified prior to the calculation. Results of hierarchical clustering can be plotted in the form of dendrogram which is a binary tree where the height of each node is proportional to the distance of the merged daughter nodes. The dendrogram can be "cut" at any point and clustering with various numbers of clusters can be examined.

Distance measures that can be used to calculate distances between two observations are mentioned in section 3.5.2.1. To be able to cluster observations using the hierarchical approach, distance between two clusters must also be calculated. Distance measures that can be used are

- single linkage – the distance between two clusters is calculated as the distance of two observations from different clusters which are closest to each other. This distance measure usually results in a larger number of smaller clusters compared to complete link.
- complete linkage – the distance between two clusters is calculated as the distance of two observations from different clusters which are farthest from each other. Complete link generally results in larger clusters than single link.
- average linkage – the distance is calculated as the mean of distances of observations from the two clusters. This method has the advantage that it is robust to noise compared to the two previous ones.

**3. density-based** Density-based clustering methods rely on the assumption that clusters are regions with high density of observations separated from each other by regions with low density of observations. One of the commonly used density-based clustering algorithms is called DBSCAN. DBSCAN has two input parameters $\varepsilon$ (size of neighborhood) and $m$ (minimum points). The algorithm then finds such a clustering that for each point in a cluster there are at least $m$ points in its $\varepsilon$ neighborhood. Experiments show that $m = 4$ is a good choice for minimum points and clusters with $m > 4$ do not significantly differ [1].

The algorithm classifies each observation as a core point (contains at least $m$ observations in $\varepsilon$ neighborhood), border point (is in the neighborhood of a core point, but is not a core point itself) and noise point (is neither core point nor border point).

DBSCAN has the advantage of being able to work with noisy data. It can also find clusters of any shape and the number of clusters does not have to be specified prior to clustering. The algorithm may not, however, be suitable in situations where the clusters are not well separated by a region with low density.

**4. model-based** Model-based clustering approaches divide data to subgroups by creating a model of the data. An example of a statistical model-based clustering method is the use of Gaussian mixture models (GMM).

GMM assumes that the data in each cluster (the number of clusters $k$ has to be specified prior to clustering) come from a multivariate Gaussian distribution. The parameters of the distributions are then estimated so that the data has maximum likelihood of being generated by the model using the iterative expectation maximization algorithm (EM). For each observation, GMM calculates probabilities of assignment to all of the $k$ clusters, it is thus a "soft" clustering algorithm [26].

Data can also be clustered using artificial neural networks (Kohonen networks also called self-organizing maps [52]). This approach was, however, not used for the purpose of this thesis.

#### 3.5.2.1 Distance measures for static data

Distance between two objects $A$ and $B$ is a function $d$ which fulfills these criteria:

$$d(A, B) = d(B, A) \tag{3.21}$$

$$d(A, B) = 0 \Leftrightarrow A = B \tag{3.22}$$

$$d(A, B) \geq 0 \tag{3.23}$$

for every $A$, $B$, $C$ [29]. If the distance function also fulfills the criterion

$$d(A, C) \leq d(A, B) + d(B, C) \tag{3.24}$$

it is called metric.

One of the commonly used distances for numerical variables is the Minkowski distance. Minkowski distance between two vectors $x$ and $y$ which contain $p$ values (or features in the context of clustering) is defined by

$$d(x, y) = \sqrt[k]{\sum_{i=1}^{p} |x_i - y_i|^k} \tag{3.25}$$

where $k > 0$ [28]. For $k = 1$, the distance function is called Manhattan distance and for $k = 2$, the function is called Euclidean distance.

For categorical variables, the simplest measure of distance is the proportion of values which differ among the two vectors. For example for two categorical vectors $x$ and $y$ consisting of $p$

values, the distance would be

$$d(x, y) = 1 - \frac{m}{p} \tag{3.26}$$

where $m$ is the number of values which agree in the two vectors.

### 3.5.3 Clustering of time series data

Time series clustering algorithms are procedures which try to find groups of similar time series. There are various approaches which can cluster time series of one variable or more variables (multivariate time series) and which can be used for time series with the same length or for time series with varying lengths. A lot of these algorithms are similar to the ones used for static data clustering, but can use different approaches to accommodate for time series.

One approach is to try to modify algorithms that are used for static data and transform them in such a way that they can be used for time series. This can be accomplished by replacing distance measures used for static data with appropriate distance measures for time series. The quality of clustering then lies in finding the best distance measure for given time series. A review of commonly used distance measures for time series is in section 3.5.3.1.

The other approach is to convert time series data to static data by either extracting features from the data or creating a model of the time series and then using parameters of this model.

Thus, there are three main approaches that fall into these categories, based on what is the input to the clustering algorithm [9]:

1. raw data
2. features
3. model parameters

These approaches are also summarized in Fig. 3.1.

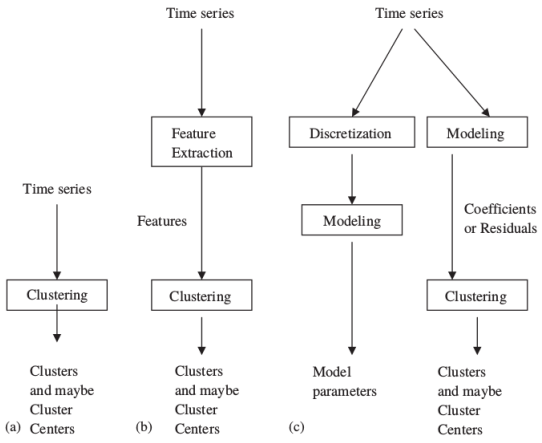



Fig. 3.1: Three time series clustering approaches: (a) raw-data-based, (b) feature-based, (c) model-based. Taken from: [9]

#### 3.5.3.1 Distance measures for time series

As the most basic distance measure, Minkowski distance measures mentioned in 3.5.2.1 can be used.

Other distance measures, which are applicable to time series, are the Pearson correlation coefficient and other related distance measures. Pearson correlation coefficient $r$ of two $p$-dimensional vectors $x$ and $y$ is defined as

$$r_{x,y} = \frac{\sum_{k=1}^{p}(x_k - \bar{x})(y_k - \bar{y})}{S_x S_y} \tag{3.27}$$

where $S_x$ and $S_y$ are the sample standard deviations.

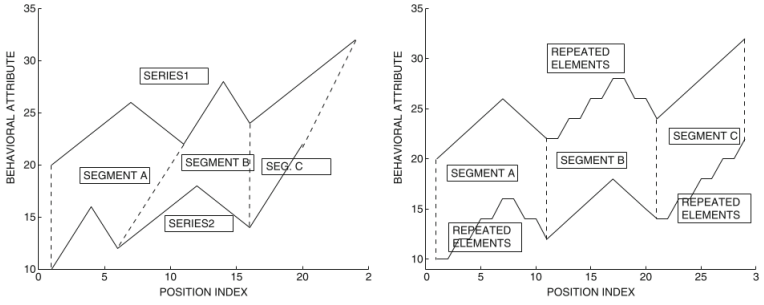
Another distance measure based on correlation, which is used by authors of [47] in fuzzy $c$-means time series clustering, is

$$d_{cc}^1 = \left(\frac{1 - cc}{1 + cc}\right)^\beta \tag{3.28}$$

and

$$d_{cc}^2 = 2(1 - cc) \tag{3.29}$$

*Dynamic time warping* (DTW) is a distance measure for time series which creates a mapping between two time series so that the distance (such as Minkowski distance) between them is minimized. DTW can be used also for time series of different length, since many-to-one mapping is allowed [26]. Illustration of dynamic time warping is in Fig. 3.2.




(a) Original series (b) Warped series

Fig. 3.2: Illustration of dynamic time warping. Taken from: [26]

For two time series of length $m$ and $n$, an $m \times n$ matrix of distances between points is calculated. A warping path with the minimum distance which starts and finishes at the opposite corners of the matrix and also satisfies conditions of monotonicity and continuity is searched for using dynamic programming. To prevent warpings where a small part of the time series maps on a large part of the other time series, global constraints for the warping path are used (restricted zones are introduced in the distance matrix). Lower bounds for DTW are introduced to further speed up the computation (for example *LB_keogh* [27]).

#### 3.5.3.2 Raw time series clustering algorithms

Hierarchical clustering methods for static data can be applied to time series, when an appropriate distance for time series is chosen. Partitional clustering also works similarly for time series with an appropriate distance measure for time series.

The basic principles of two algorithms, which were developed to cluster time series and are used in this thesis, are explained in the next paragraphs.

**TADPole** TADPole (Time-series Anytime Density Peaks) is a clustering algorithm introduced in [31] based on the density peaks (DP) algorithm [32] and using DTW distance measure. DP is a density-based algorithm and assumes that cluster centers are points with higher local density surrounded by points with lower local density and their distance from another points with higher local density is relatively high.

Input parameters to TADPole are $d_c$ (cutoff distance) and $k$ (number of clusters). The TADPole algorithm uses upper bound (Euclidean distance) and lower bound (Keogh [27]) on DTW distance. The algorithm uses these bounds to calculate points which have a lot of neighbors (distance lower than $d_c$). TADPole then tries to prune as many DTW distance calculations as possible and finds the centroids in the highest density regions [33].

**k-Shape** $k$-Shape is a partitional algorithm for clustering time series with a distance measure called shape-based distance (SBD) [34]. SBD is a distance measure based on cross-correlation and for two time series $x$ and $y$ is calculated as

$$SBD(x, y) = 1 - \max_w \left( \frac{CC_w(x, y)}{\sqrt{R_0(x, x)R_0(y, y)}} \right) \tag{3.30}$$

where $CC_w$ is the cross-correlation computed for shift $w$ and $R_0(x, x)$, $R_0(y, y)$ are the autocorrelations of the two series at zero shift. This distance measure has values between 0 and 2. The algorithm works similarly to $k$-means, the centroids are, however, not calculated as a simple mean. The centroid is calculated as the time series that minimizes the sum of squared SBD distances from other observations in the same cluster.

#### 3.5.3.3 Time series representations

Representation of time series $x$ with length $n$ is a model of the time series such that it approximates $x$ and its dimensionality $p$ is reduced ($p < n$) [49]. Calculating time series representations serves the purpose of reducing dimensionality to avoid the curse of dimensionality, reducing computational requirements and handling of noise. The following time series representations were chosen for the purposes of this thesis:

- *Seasonal profile (SP)* – the series are divided into sub-series of chosen length and mean of all the sub-series is calculated
- *Generalized additive models (GAM)* – regression coefficients of the model are taken as the representation
- *Discrete Fourier transform (DFT)* – a chosen number of coefficients are taken as the representation
- *Discrete wavelet transform (DWT)* – a chosen number of coefficients are taken as the representation
- *Discrete cosine transform (DCT)* – a chosen number of coefficients are taken as the representation
- *Piecewise aggregate approximation (PAA)* – divides the series into sub-series of chosen length. For each sub-series, mean is calculated and dimensionality is thus reduced

### 3.5.4 Clustering validation

In clustering as an unsupervised method, there is usually no response variable to assess the quality and correctness of the resulting clustering. Therefore, different methods from the ones used in classification are used to analyze the clusters and assess their validity. Two possible approaches, which were used to analyze clusters in this thesis, are *internal* and *stability measures.*

#### 3.5.4.1 Internal measures

Internal measures are clustering quality measures that reflect properties of the resulting clusters such as compactness of clusters or separation between them and are calculated from the parameters of the observations in the resulting clusters.

Internal measures used in this thesis are *Davies-Bouldin index, silhouette width* and *Dunn index.*

The Davies-Bouldin ($DB$) index is defined as [25]

$$DB = \frac{1}{N} \sum_{i=1}^{N} \max_{j, i \neq j} \left\{ \frac{S_i + S_j}{M_{ij}} \right\} \tag{3.31}$$

where $S_i$ and $S_j$ are the dispersions (standard deviations) of cluster $i$ and $j$ respectively and $M_{ij}$ is the distance between centroids of the two clusters (vectors which are chosen as characteristic of the clusters).

"The Dunn Index is the ratio of the smallest distance between observations not in the same cluster to the largest intra-cluster distance." [23] The value of this index should be maximized.

The silhouette width is calculated as the average value of silhouette ($s$) of each observation. The silhouette value of $i$-th observation is calculated as [24]

$$s(i) = \frac{b(i) - a(i)}{max\{b(i), a(i)\}} \tag{3.32}$$

where $a(i)$ is the average distance of $i$-th observation to other observations in the same cluster and $b(i)$ is the average distance of $i$-th observation to observations in the nearest cluster (to which $i$ does not belong). The silhouette width ($Sil$) is then calculated as the average of $s(i)$ of all observations.

#### 3.5.4.2 Stability measures

Stability measures are based on the comparison of clusterings where one of the variables was removed compared to the clustering on the full data set. Four stability measures were used in this thesis and were calculated using the *clValid R* package [23].

*Average proportion of non-overlap* ($APN$) is calculated as

$$APN = \frac{1}{MN} \sum_{i=1}^{N} \sum_{l=1}^{M} \left( 1 - \frac{\left| C^{i,l} \cap C^{i,o} \right|}{|C^{i,o}|} \right) \tag{3.33}$$

where $N$ is the number of observations, $M$ is the number of variables, $C^{i,l}$ is the cluster containing the $i$-th observation resulting from clustering with the $l$-th variable removed and $C^{i,o}$ is the cluster resulting from clustering using all of the variables. This measure takes on values between zero and one with values close to zero denoting more stable clustering.

*Average distance (AD)* is calculated as

$$AD = \frac{1}{MN} \sum_{i=1}^{N} \sum_{l=1}^{M} \left[ \frac{1}{|C^{i,l}|\,|C^{i,o}|} \left( \sum_{i \in C^{i,o}, j \in C^{i,l}} dist(i,j) \right) \right] \tag{3.34}$$

and has the meaning of average distance between clusters of observations in the same cluster resulting from clustering with one of the variables removed and clustering with all of the variables. Small values of this measure denote stable clustering.

*Average distance between means (ADM)* is calculated as

$$ADM = \frac{1}{MN} \sum_{i=1}^{N} \sum_{l=1}^{M} dist\left( \bar{x}_{C^{i,l}}, \bar{x}_{C^{i,o}} \right) \tag{3.35}$$

where $\bar{x}_{C^{i,l}}$ is the average of observations in cluster $C^{i,l}$. This measure computes the average distance between averages of observations in clustering with all of the variables and with one of the variables removed. $ADM$ should, again, be minimized to obtain a stable clustering.

*Figure of merit (FOM)* is calculated as ($K$ is the number of clusters)

$$FOM = \frac{1}{M} \sum_{l=1}^{M} \sqrt{\frac{1}{N} \sum_{k=1}^{K} \sum_{i \in C_k(l)} dist(x_{i,l}, \bar{x}_{C_k(l)})} \tag{3.36}$$

and calculates the average distance of the observations in the left-out column $x_{i,l}$ from the mean of the cluster resulting from clustering without the $l$-th column $C_k(l)$.

# 4 Exploratory Data Analysis

Exploratory data analysis was performed on a subset of sleep recordings of the original data set which was created by excluding incorrect values and outliers (see Preprocessing). Subsequently, a subset of users satisfying certain conditions was chosen and a set of summarizing features was extracted for every user, which allowed to explore variations of sleep parameters between users rather than between recordings.

## 4.1 Recordings

The original data set from the *Sleep as Android* application consist of 15 905 401 sleep recordings. For each sleep recording 24 parameters were saved and the actigraph was recorded. Noise levels were recorded for some of the recordings.

### 4.1.1 Preprocessing

The basic preprocessing step was removing the incorrect entries such as negative values of sleep duration and other parameters. In the next step, extreme outliers were removed from the data set by visual inspection of histograms. The fields *id*, *netSleepAdjustment* and *device* were deleted from the data set, since they were not used in any of the following analyses.

New features were calculated from the existing ones, such as $BMI$, *timeToSleep* (sleep latency – time that it takes to fall asleep, calculated as $sleepStart - from$ and saved in hours), $alarmWakeDiff$ (difference between the time of set alarm and actual wake time, calculated as $alarmTime - to$ and saved in hours), *midsleep* (calculated as the time in the middle between bedtime and wake time).

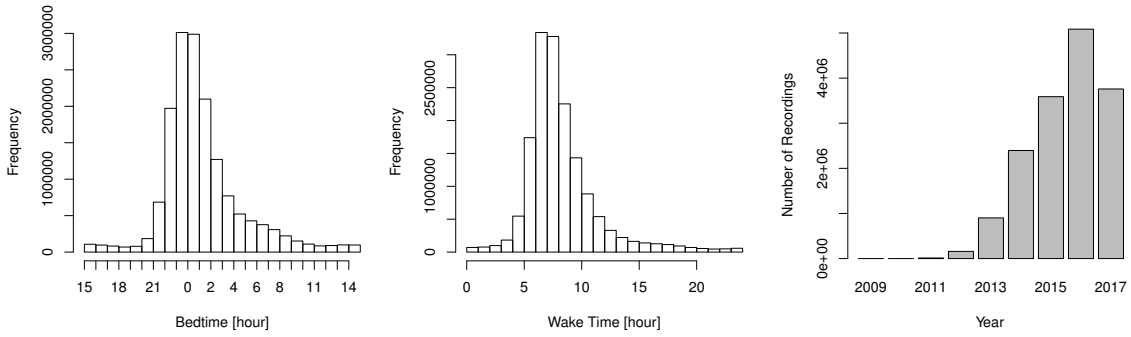
### 4.1.2 from, to

These fields contain the start and end of recording (time in milliseconds since 01/01/1970 00:00:00 UTC). Recordings were collected between 2009 and September 2017. The first two years were not used for analysis, since they contain only a very small portion of the recordings (Fig. 4.1c).

These original fields were not used for analysis, but were transformed to the fields *bedtime* and *waketime* which contain a real number between 0 and 24 representing the time in hours in the user's local time zone. Fields *year* and *yday* (number of day in the year) were calculated to keep the information about date.

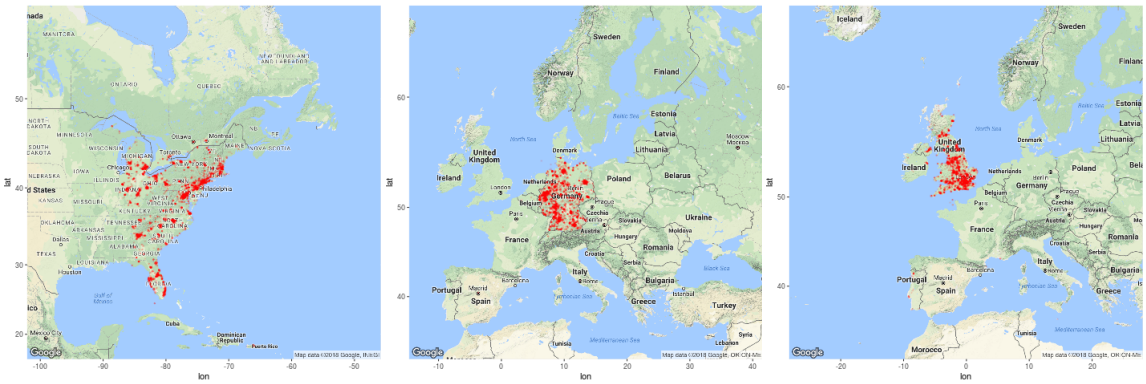
In some of the used algorithms, a modified *bedtime* field was used, which contains values between 15 and 39 and was created by adding the number 24 to every value of bedtime which was less than 15, was used for simplicity. This change will be pointed out again in sections where it was used.

Histograms of *bedtime* and *waketime* parameters can be seen in Fig. 4.1.




(a) (b) (c)

Fig. 4.1: Histograms of (a) bedtime and (b) wake time, (c) number of recordings by year




(a) (b) (c)

Fig. 4.2: Locations of users in three time zones with the most recordings (a) America/New_York, (b) Europe/Berlin, (c) Europe/London

### 4.1.3 timeZone

This field contains the time zone which is set on the user's device in the format Continent/City (e.g. Europe/Prague). Numbers of recordings present in seven largest time zones can be seen in Tab. 4.1. Locations of users in the four time zones that contain the most recordings can be seen in figure 4.2 (these plots were created using the fields *geoLatitude* and *geoLongitude* with the *R* package *ggmap* [19]).

| Time zone | Recordings | Percentage |
|---|---|---|
| America/New_York | 1492334 | 9.4% |
| Europe/Berlin | 1438375 | 9.0% |
| Europe/London | 1141170 | 7.2% |
| America/Chicago | 1064308 | 6.7% |
| America/Los_Angeles | 938371 | 5.9% |
| Europe/Amsterdam | 797973 | 5.0% |
| Asia/Tokyo | 723015 | 4.5% |

Tab. 4.1: Number of recordings in the seven time zones with the most recordings

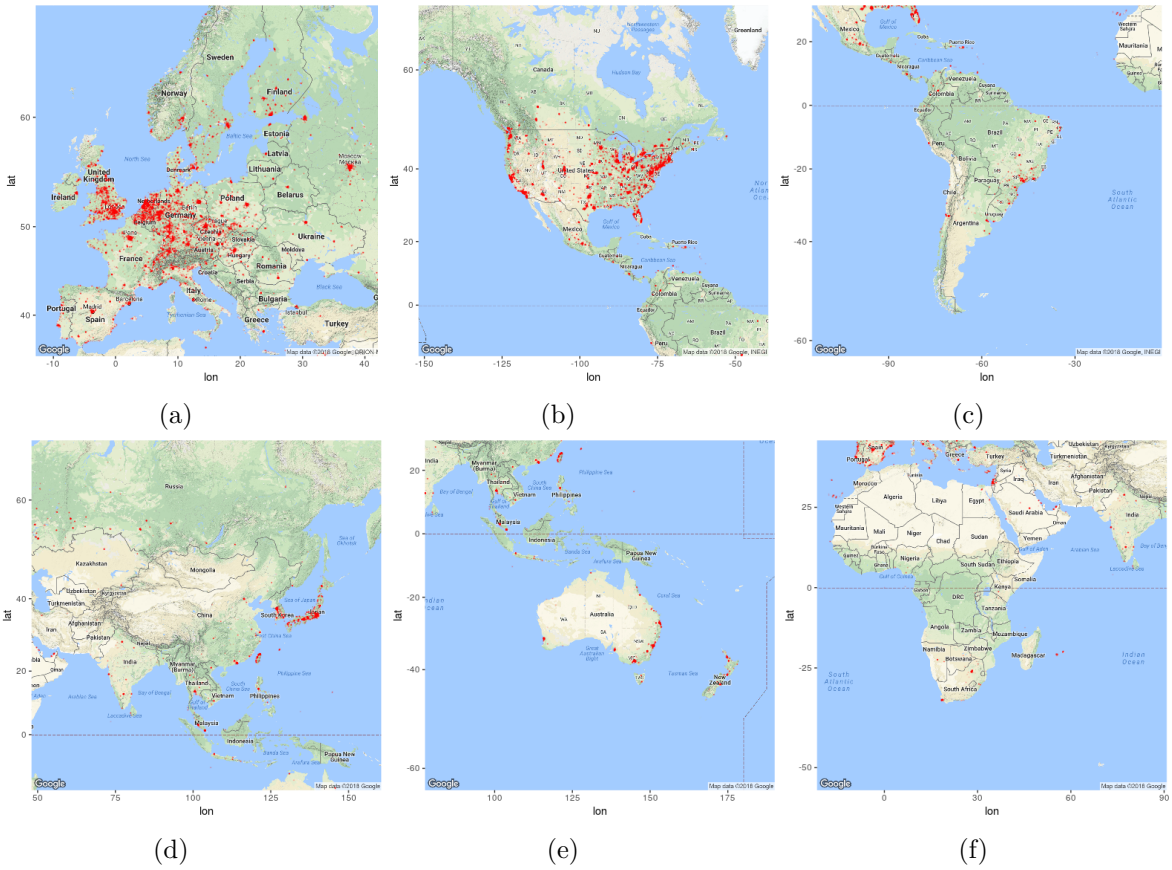



Fig. 4.3: Locations of users in continents (a) Europe, (b) North America, (c) South America, (d) Asia, (e) Australia, (f) Africa

### 4.1.4 geoLatitude, geoLongitude

Information about the user's location in the form of latitude and longitude are recorded in the fields *geoLatitude* and *geoLongitude*. Entries of users' locations are present approximately in 56% of recordings. Numbers of recordings on continents which contain the most recordings are present in Tab. 4.2. Locations are plotted in figure 4.3. These figures were created using the *R* package *ggmap* [19].

| Location | Recordings | Percentage |
|---|---|---|
| Europe | 7947854 | 50% |
| Americas | 5307041 | 33% |
| Asia | 1683958 | 11% |
| Australia | 530056 | 3% |
| Africa | 136607 | 1% |

Tab. 4.2: Number of recordings by continents

### 4.1.5 commentTags

The field *commentTags* contains comments that users can use to label their sleep recordings with various additional information. One sleep recording can contain multiple comment tags. While some of the comment tags are automatically generated by the software, users can also add their own comment tags. Some examples of the comment tags generated automatically by software include:

- #watch – is generated when other wearable devices (smart watches) are used for sleep tracking
- #newmoon – is generated automatically in recordings from nights when the moon is in the new moon phase
- #fullmoon – is generated automatically in recordings from nights when the moon is in the full moon phase
- #home, #geo0, #geo1, #geo2 – the application logs users' most used locations and labels the recordings accordingly by these tags
- #lullaby – is generated when the user uses the lullaby function to play a lullaby before falling asleep

Users can choose from some standard tags (see Fig. 4.4), or can include their own tags. Examples of some standard user tags, from which users can choose include:

- #sport
- #food
- #stress
- #work
- #med (medication)
- #gooddream, #baddream
- #caffeine
- #alcohol

A table of the most common tags of recordings is given in Tab. 4.3. In the figure 4.4, the screen on which user can add comment tags to the recording is shown. Users can choose from standard tags by clicking on the pictures or add their own comment.

| Tag | Recordings |
|---|---|
| – | 6082214 |
| #home | 3858457 |
| #watch;#home | 976793 |
| #watch | 503293 |
| #geo1 | 344232 |
| #sonar;#home | 256260 |
| #newmoon;#home | 191403 |
| #fullmoon;#home | 186111 |
| #cloud | 185538 |
| #sonar | 168544 |
| #newmoon | 144270 |
| #fullmoon | 143091 |
| #geo0 | 103840 |
| #lullaby;#home | 80204 |
| #watch;#geo1 | 78376 |
| #geo2 | 77191 |
| #lullaby | 76770 |
| #alcohol | 49700 |

Tab. 4.3: Comment tags sorted by number of recordings in which they appear

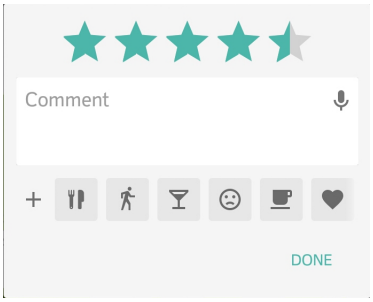
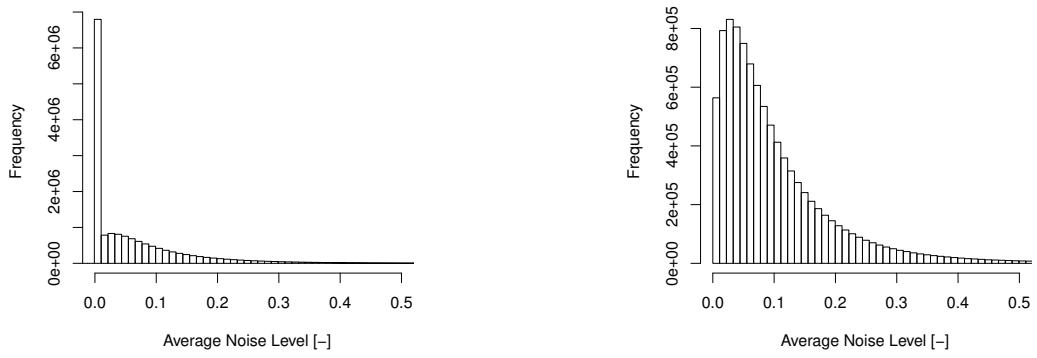



Fig. 4.4: Adding comment tags in the *Sleep as Android* application




(a) (b)

Fig. 4.5: Histograms of *avgNoiseLevel* (a) with zero values, (b) without zero values

### 4.1.6 avgNoiseLevel

This field contains the average level of noise picked up by the smartphone's microphone. In a large number of recordings (6 247 550) average noise level has a zero value. These are the recordings where noise recording was not used. It can be seen from histograms shown in figures 4.5a and 4.5b of *avgNoiseLevel* and *avgNoiseLevel* without the zero values that this parameter has a right skewed distribution. Summary of parameters of the distribution of *avgNoiseLevel* without zero values is given in Tab. 4.4.

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|
| 0.00000 | 0.03620 | 0.07347 | 0.10633 | 0.13664 | 1.00000 |

Tab. 4.4: Summary statistics of *avgNoiseLevel* variable

### 4.1.7 noOfCycles

Number of sleep cycles detected during the sleep recording by the software is contained in the field *noOfCycles*. *noOfCycles* is usually a number between 1 and 10, but the dotplot in Fig. 4.6b shows that it takes on different values in the oldest and newest recordings. The zero values that are present in the older recordings could be missing values or could mean that a change had been made in the detection algorithm. The values higher than 10 in newer recordings could also mean a change in the detection algorithm, or simply the change of maximum possible detected cycles and the parameter should therefore be used with caution.

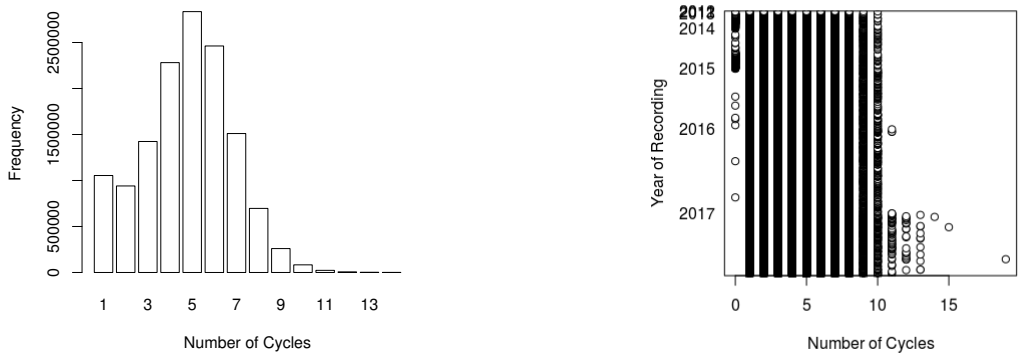
A bar plot for *noOfCycles* is shown in Fig. 4.6a. Summary of the variable is in Tab. 4.5.

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. | NA's |
|---|---|---|---|---|---|---|
| 0.0 | 3.0 | 5.0 | 4.8 | 6.0 | 21.0 | 2282735 |

Tab. 4.5: Summary statistics of variable *noOfCycles*

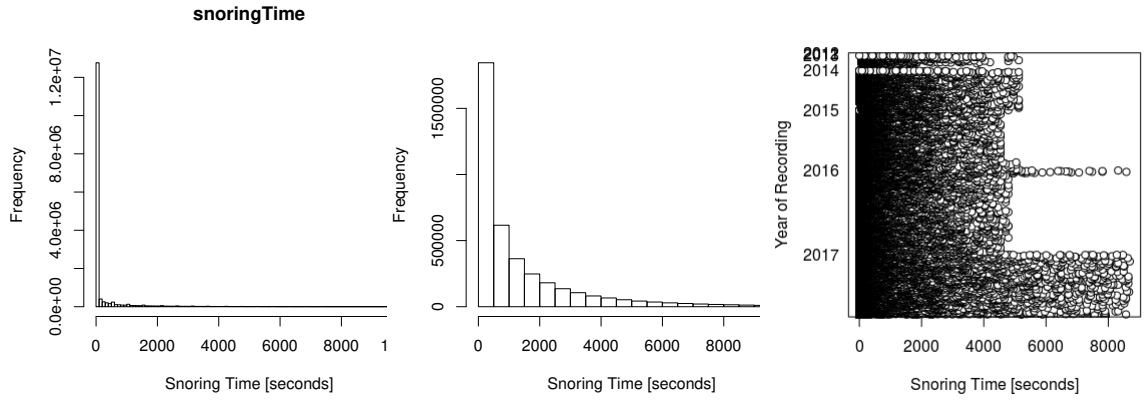
### 4.1.8 snoringTime

Total time of snoring in seconds is stored in the field *snoringTime*. Total of 3 461 511 values are zero. This may mean that the user either did not snore or that sound recording was not used. Histograms of *snoringTime* with and without zero values are in figures 4.7a and 4.7b.




(a) (b)

Fig. 4.6: (a) histogram and (b) dotplot of *noOfCycles*




(a) (b) (c)

Fig. 4.7: *snoringTime*: (a) histogram, (b) histogram without zero values, (c) dotplot

As with some of the previously mentioned variables the problem of abrupt changes in range is also present in *snoringTime* (see Fig. 4.7c).

### 4.1.9 netSleepLength

Total sleep duration is recorded in the field *netSleepLength*. *netSleepLength* is recorded in minutes and excludes the intervals in which the recording was paused by user.

Histogram of *netSleepLength* can be seen in Fig. 4.8. The histogram contains two peaks: one above 400 minutes (6 hours and 40 minutes) which corresponds to normal sleep recordings in the night. *Sleep as Android* can also be used to record sleep during ”naps” with a default length of 30 minutes. Most of the recordings near the lower peak at around 30 minutes are probably recordings of these naps.

### 4.1.10 subjectiveRating

The field *subjectiveRating* contains a number from 0 to 5 which reflects the user's subjective rating of perceived sleep quality. The subjective rating of a sleep recording can be inserted to the application in a form of stars (see Fig. 2.1), where the maximum is five stars, but even a

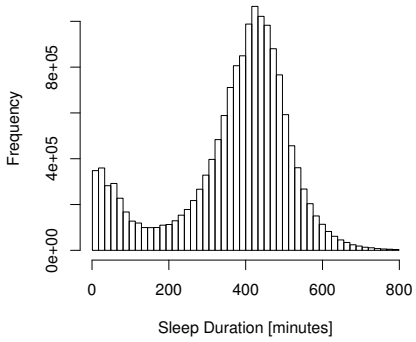
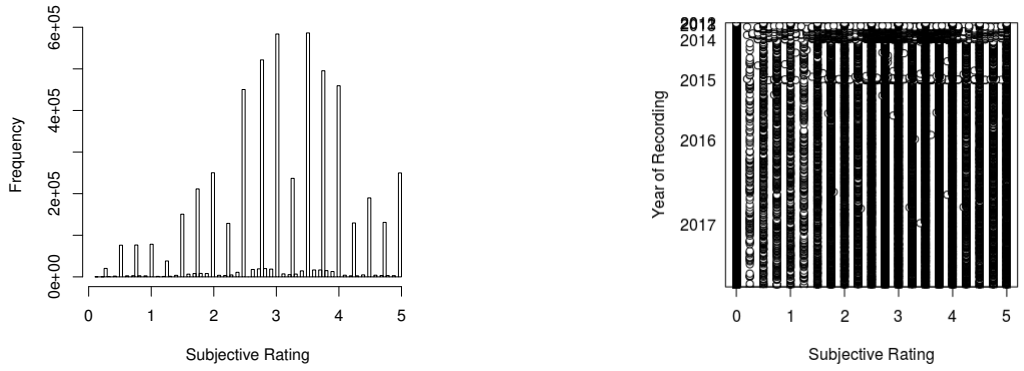



Fig. 4.8: Histogram of sleep duration




(a) (b)

Fig. 4.9: *subjectiveRating*: (a) barplot and (b) dotplot

non-integer value can be chosen. Subjective rating is present in 5 337 454 recordings.

Histogram of *subjectiveRating* is shown in Fig. 4.9a. The dot plot shown in 4.9b reveals that values which the parameter can take on were modified early on and that some values different from the standard ones are occasionally present.

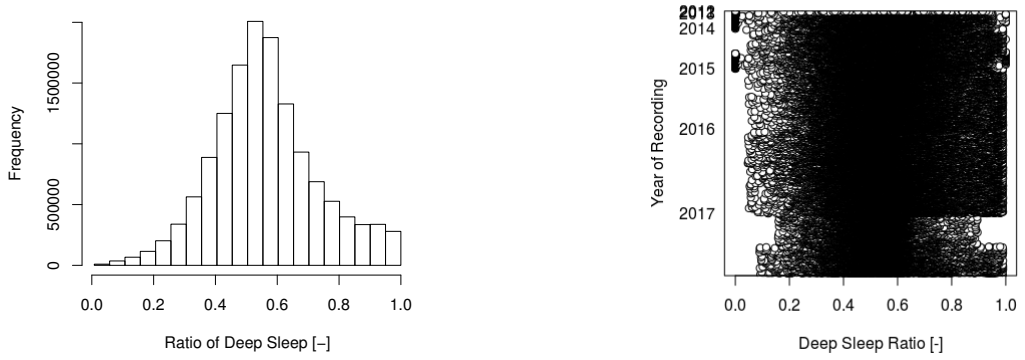
### 4.1.11 deepSleepRatio

The ratio of deep sleep duration to total sleep duration is recorded in the field *deepSleepRatio* and it can be seen from the histogram in Fig. 4.10a that the most common values are around 0.5 to 0.6. The distribution is slightly skewed to the right.

In earlier recordings, zero values are present, as can be seen from the dot plot in Fig. 4.10b. It can also be seen that the skewness tends to get smaller in newer recordings. Also, abrupt change in range can be seen in newer recordings.

### 4.1.12 gender

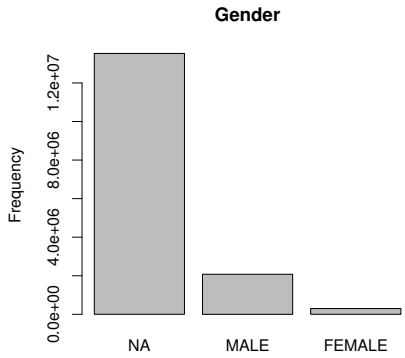
Information about the user's gender is present in some of the recordings, although as can be seen in Fig. 4.11 and Tab. 4.6, in most of the recordings (85%) this field is miss-




(a) (b)

Fig. 4.10: *deepSleepRatio*: (a) histogram and (b) dotplot

ing. The number of recordings labeled as female is particularly low (only 2% of the recordings).




| Gender | Recordings | Percentage |
|---|---|---|
| MALE | 2076760 | 13% |
| FEMALE | 298616 | 2% |
| NA | 13530025 | 85% |

Tab. 4.6: Number of recordings by *gender*

Fig. 4.11: Barplot of *gender*

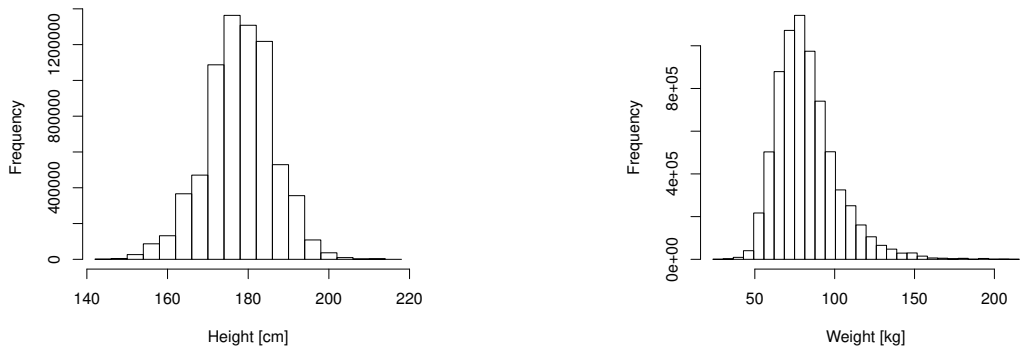
### 4.1.13 height, weight

Height and weight data are present in approx. 44% (7 116 965 for height and 7 143 797 for weight) of the recordings. Histograms of these parameters are shown in figures 4.12a and 4.12b.

$BMI$ was calculated for the recordings in which both height and weight were present (7 055 381), as the ratio of weight in kilograms to the square of height in meters. Histogram of the resulting parameter $BMI$ is shown in Fig. 4.13a. Placement of users to $BMI$ groups is shown in Fig. 4.13b.

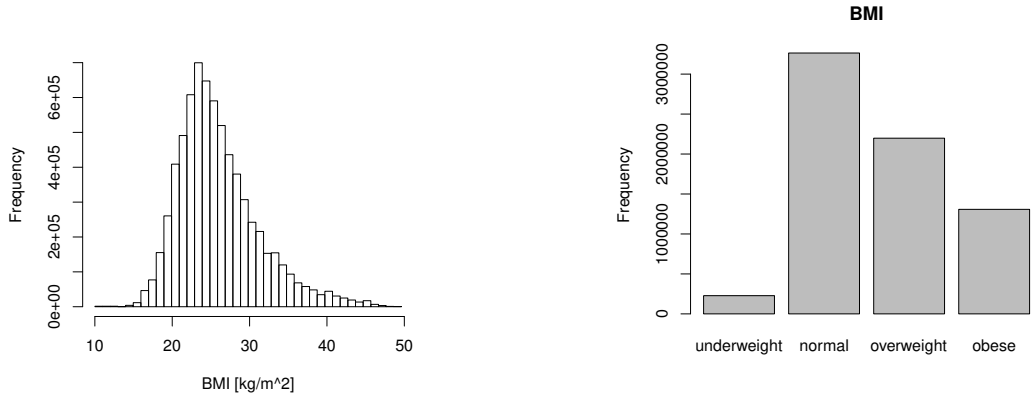
### 4.1.14 birthdate

Information about users' birth date is present in some of the users' recordings. This entry is, however, present only in approximately 14% of the recordings. In most of the analyses age rather than birth date was used. The histogram of age is shown in Fig. 4.14.




(a) (b)

Fig. 4.12: Histograms of (a) height and (b) weight




(a) (b)

Fig. 4.13: $BMI$: (a) histogram, (b) barplot of $BMI$ classes

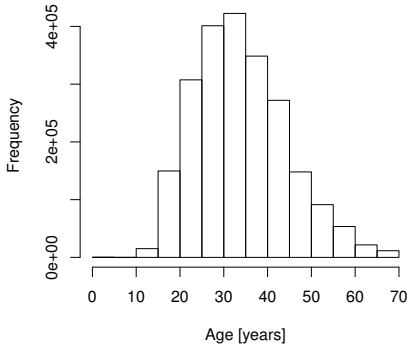



Fig. 4.14: Histogram of age

## 4.2 Users

There are 88879 users in total with 13632 users having more than 365 recordings. The following table 4.7 displays numbers of users that have available sleep recordings on more than 80% of days in one year.

| Year | Users |
|---|---|
| 2014 | 2631 |
| 2015 | 3952 |
| 2016 | 5806 |
| 2017 | 1407 |

Tab. 4.7: Number of users with more than 80% of recordings in a year

For an exploratory analysis of users a subset of recordings which satisfy following conditions was chosen:

```
BMI  - 15 to 40
age  - between 15 and 60 years
gender  - male
year - 2016
bedtime - after 6 P.M.
wake time - before 1 P.M.
subjective rating - is not a missing value
```

Subsequently, users that have more than 20 recordings available in this subset were chosen. For every user eleven features were calculated. After this procedure 1 446 users without missing values remained. The extracted features are:

- *BMI* – mean BMI of the user
- *age* – mean age of the user
- *mid_wday* – average midsleep on weekdays
- *mid_diff* – difference of average midsleep on weekends and weekdays
- *length_wend* – average duration of sleep during weekends
- *length_diff* – difference of average sleep duration on weekends and weekdays
- *rating_wday* – average subjective sleep quality rating on weekdays
- *rating_wend* – average subjective sleep quality rating on weekends
- *snoring* – average time spent snoring in seconds
- *DSratio_wday* – average deep sleep ratio on weekdays
- *DSratio_wend* – average deep sleep ratio on weekends

The eleven features calculated for each user are displayed in a matrix in Fig. 4.15. On the diagonal, a histogram is plotted. Above the diagonal Pearson correlation coefficients between the two parameters are shown. Below the diagonal scatter plots of the two parameters with linear trends are shown.

While most of the correlation coefficients in Fig. 4.15 show negligible correlations (using the rule of thumb from [21]), low correlations are present between $BMI$ and $age$ (positive), $mid\_wday$ and $mid\_diff$ (negative), $mid\_wday$ and $length\_wend$ (negative), $snoring$ and $BMI$ and also $age$ (positive). Moderate positive correlations are present between $length\_wend$ and $length\_diff$, $rating\_wday$ and $rating\_wend$. A very high positive correlation is present between $DSratio\_wday$ and $DSratio\_wend$.

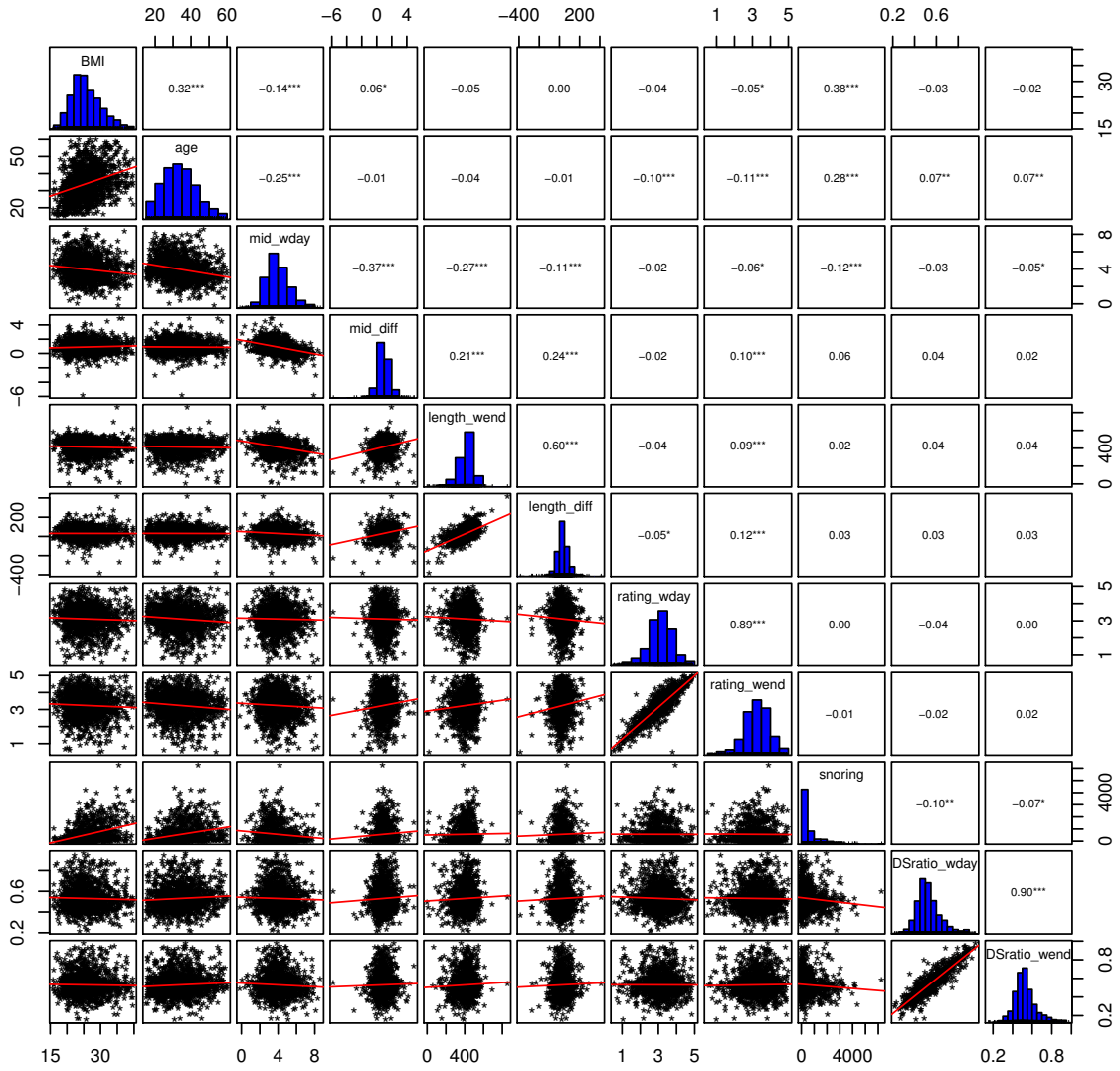



Fig. 4.15: Grid plot of extracted features. Histogram is plotted on the diagonal. Above the diagonal, Pearson correlation coefficients are shown. Scatter plots and linear trends are shown below the diagonal

# 5 Analysis of the *Sleep as Android* Data Set

This chapter contains the results of applying the methods presented in chapter 3 to the *Sleep as Android* data set.

## 5.1 Chronotype analysis

In section 1.1 (Chronotypes), it was mentioned that eveningness and morningness changes throughout lifespan. To analyze chronotypes a subset of users satisfying these conditions were chosen:

```
age - between 15 and 70
gender - male
number of available recordings - more than 20
```

Total of 6366 users were chosen by these conditions. Variables $length\_diff$ (difference between mean sleep duration on weekends and weekdays, see section 4.2) or $mid\_diff$ (difference between mean midsleep on weekends and weekdays) could provide some measure of a person's chronotype since evening types tend to accumulate sleep debt during the week and sleep longer on weekends. It can be seen in Fig. 4.15 that no linear correlation with age is present. We chose the variable $length\_diff$ to analyze chronotypes.

The users were divided into six age groups (15-20, 20-30, ..., 60-70 years). The difference in chronotypes should be most pronounced between the youngest age group, where eveningness has its peak and the oldest age group, where morningness is more common. We therefore have a hypothesis that $length\_diff$ should be higher in the youngest age group compared to the oldest age group. Since the numbers of users in the two groups were not particularly high and the older group does not seem to come from a normal distribution (visual comparison to normal distribution using qq plot), we used the bootstrap hypothesis test with 10 000 bootstrap samples to test the equality of means. The results of the test is in Tab. 5.1. The resulting $p$-value of an unequal variance $t$-test is also presented for comparison.

We therefore reject the null hypothesis at the significance level $\alpha = 0.001$ and conclude that the mean difference in sleep duration between weekends and weekdays is higher by at least 13.97 minutes (LCB – CI 95%) compared to the older group. As can be seen in Fig. 5.1, the other age groups do not seem to differ from each other.

| Parameter | $N_1$ | $N_2$ | $Mean_1$ | $Mean_2$ | Diff. CI (95%) | $p$-value | $p_{boot}$ | Sig. |
|---|---|---|---|---|---|---|---|---|
| $length\_diff$ | 508 | 100 | 35.61 | 12.22 | 13.97 (LCB) | 0.00003 | $< 0.0001$ | ✓ |

Tab. 5.1: Results of hypothesis testing for $length\_diff$

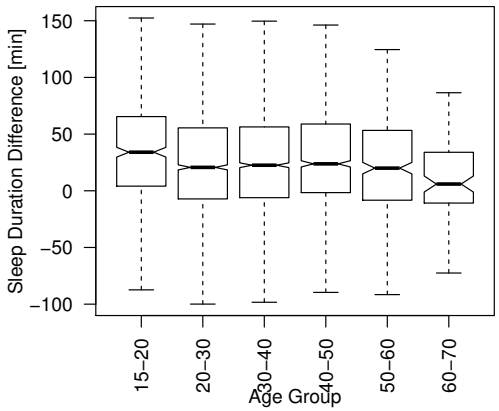



Fig. 5.1: Distribution of $length\_diff$ in age groups

## 5.2 Analysis of Effects of Important Events on Sleep Scheduling

Two events which could possibly alter sleeping habits of the population were chosen. Namely the vote in the United Kingdom to leave the European Union (Brexit vote) and presidential elections in the United States of America which were won by Donald Trump.

We hypothesize that these large scale events could alter the population's sleeping habits and lower the sleep duration, as people are waiting for the results or debating about the outcomes of these votes.

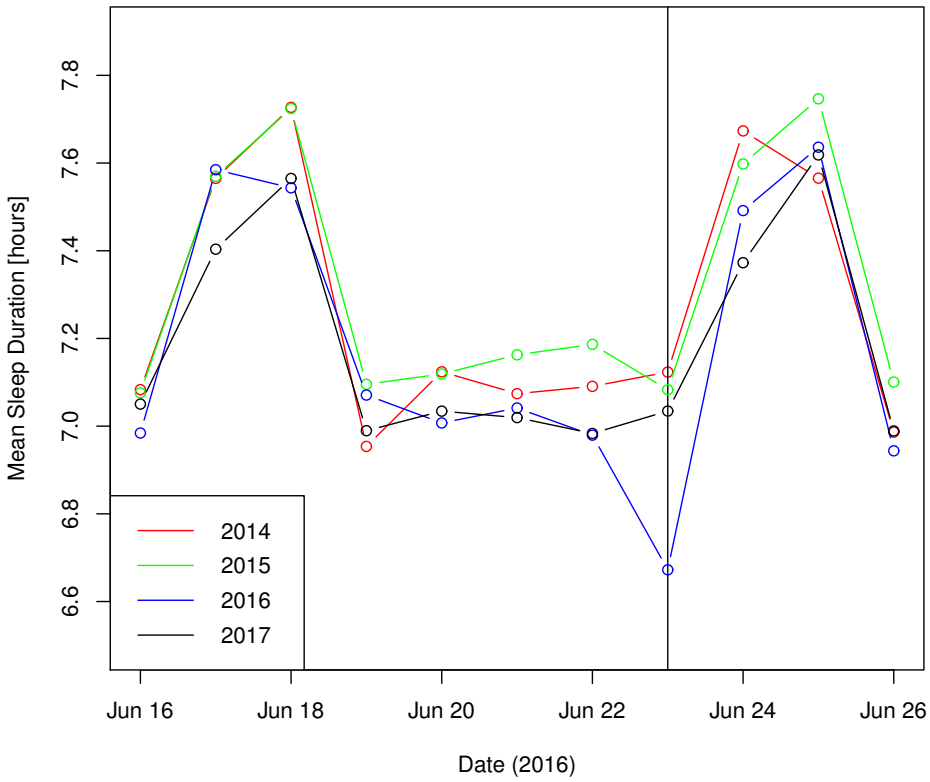



Fig. 5.2: Mean sleep lengths in Great Britain. The black vertical line denotes the night of the Brexit vote.

### 5.2.1 Brexit

Recordings from the United Kingdom with sleep duration longer than four hours and shorter than twelve hours were chosen. Mean sleep durations of users from the United Kingdom in the nights around the Brexit vote from years 2014 through 2017 are plotted in Fig. 5.2 (the sleep duration sequences in each year were aligned to the closest dates such that the days of the week are the same). A shorter mean sleep duration in comparison to other years in year 2016 in which the vote took place is evident. To test whether the differences are statistically significant, bootstrap hypothesis tests for difference in means were calculated pairwise among the years. Confidence intervals (95%) for difference in means were calculated using bootstrap confidence intervals (BCa). Ten thousand bootstrap samples were used in the calculation. Number of recordings for each year are in Tab. 5.2. Resulting $p$-values were corrected for multiple comparisons using the Bonferroni correction. The results in Tab. 5.2 show that the mean sleep duration in 2016 is

shorter than in the other years with a $p$-value less than 0.001. No difference is observed between the other years (Tab. 5.3).

| Year | No. of recordings | Mean sleep length [minutes] | Observed diff. | CI (UCB) | $p$-value |
|---|---|---|---|---|---|
| 2014 | 329 | 427.4 | -27.1 | -17.9 | $< 0.001$ |
| 2015 | 445 | 425.0 | -24.6 | -16.7 | $< 0.001$ |
| 2016 | 453 | 400.3 | – | – | – |
| 2017 | 493 | 422.1 | -21.7 | -13.4 | $< 0.001$ |

Tab. 5.2: Results of sleep duration tests in the UK for the night of the Brexit vote. The table contains the observed difference between mean sleep length in year 2016 and other years. The upper bounded confidence interval for the difference is presented (CI (UCB))

| Years | Observed diff. | CI | $p$-value |
|---|---|---|---|
| 2015-2014 | -2.4 | (-13.32, 7.836) | 0.62 |
| 2017-2014 | -5.3 | (-15.66, 5.004) | 0.31 |
| 2015-2017 | 2.9 | (-6.098, 12.21) | 0.51 |

Tab. 5.3: Results of pairwise comparisons of sleep duration in the UK in years 2014, 2015, 2017

### 5.2.2 Presidential elections in the USA

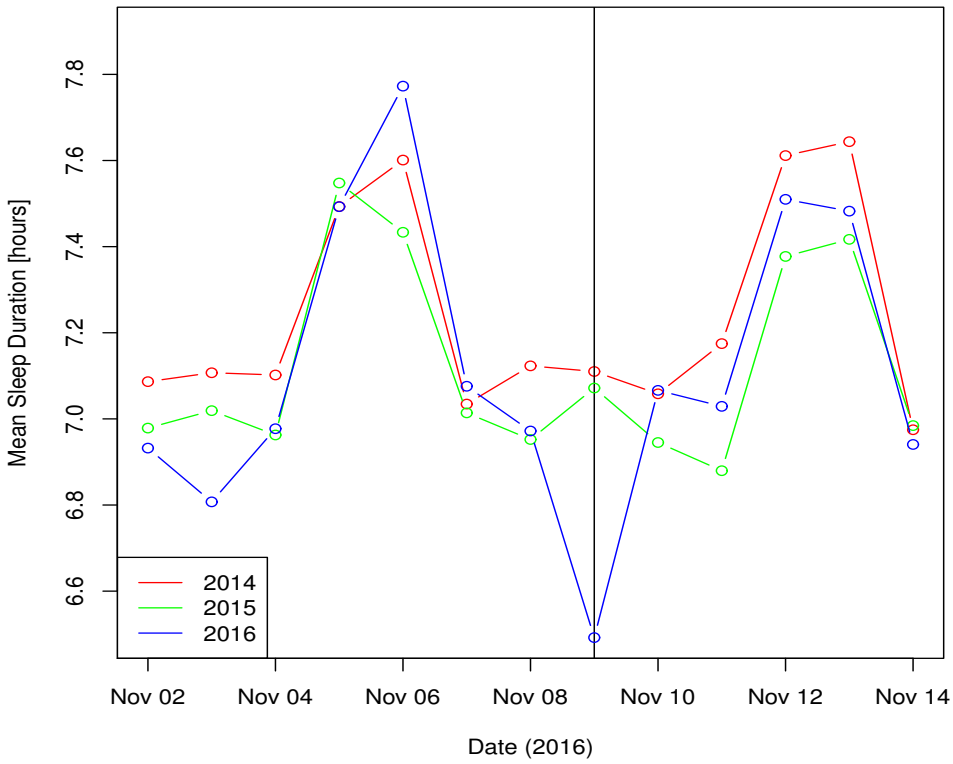



Fig. 5.3: Mean sleep durations in the United States of America. The black vertical line denotes the night of the presidential elections.

The same procedure was repeated for the presidential elections in the United States of America. Fig. 5.3 shows the mean sleep durations of users from the United States (the sleep

duration sequences in each year were aligned to the closest dates such that the days of the week are the same). In this case, however, data from year 2017 were not available for this time of the year. To test the hypothesis of mean sleep duration being shorter in 2016, the same method as the one used for Brexit was used. The results show that the null hypothesis can be rejected with a $p$-value less than 0.001. No difference was observed between the other two years (Tab. 5.5)

| Year | No. of recordings | Mean sleep length [minutes] | Observed diff. | CI (UCB) | $p$-value |
|---|---|---|---|---|---|
| 2014 | 685 | 426.6 | -37.1 | -30.8 | $< 0.001$ |
| 2015 | 878 | 424.3 | -34.8 | -28.8 | $< 0.001$ |
| 2016 | 1095 | 389.5 | – | – | – |

Tab. 5.4: Results of sleep duration tests in the US for the night of the presidential elections. The table contains the observed difference between mean sleep durations in year 2016 and other years. The upper bounded confidence interval for the difference is presented (CI (UCB))

| Years | Observed diff. | CI | $p$-value |
|---|---|---|---|
| 2014-2015 | 2.3 | (-5.414, 10.12) | 0.57 |

Tab. 5.5: Results of comparisons of sleep duration in the US in years 2014 and 2015

## 5.3 Analysis of comment tags

Variabilities in sleep parameters among groups of sleep recordings containing different tags are examined in this section. For each pair of tags chosen for analysis, a set of hypotheses on how these tags could affect various parameters of sleep was formed. These hypotheses were then tested using the Welch's $t$-test and by bootstrapping the distribution of the parameter under null hypotheses. For each two groups that were compared, confidence levels were corrected for multiple comparisons. For the purpose of comparisons, a subgroup of users satisfying these conditions was chosen:

```
sleep duration  - 4 to 14 hours
```

to exclude recordings of naps, outliers and recordings which were not stopped by the user. All of the recordings which had the tested parameter available were subsequently chosen for each test. The modified version of *bedtime*, which was mentioned in section 4, was used in this section.

Parameters in chosen groups were tested for difference in means. Since the number of observations in these groups was generally very high (tens of thousands to millions), an unequal variance $t$ test should be sufficient. However, $p$-values of bootstrap hypothesis tests with 1 000 bootstrap samples are presented for comparison.

Results presented in the tables include values $N_1$, $N_2$ – number of observations in the first and second group, $Mean_1$, $Mean_2$ – sample mean in the two groups, Diff. CI (95%) – 95% confidence interval for the difference in means, $p$-value of the unequal variance $t$-test, $p_{boot}$ – $p$-value of bootstrap hypothesis test with 1 000 bootstrap samples.

All of the presented confidence intervals of difference in means between the groups were calculated for 95% confidence. The intervals are presented either as upper bounded (UCB - upper confidence bounded), lower bounded (LCB - lower confidence bounded) or with both lower and upper bounds, depending on the null hypothesis. Bonferroni correction for multiple comparisons was used to correct the threshold $p$-value $\alpha$ for hypothesis testing.

### 5.3.1 How does alcohol affect sleep?

Recordings tagged with the #alcohol tag were compared to recordings tagged with the #home tag. It was found that users using the alcohol tag have mean sleep latency lower by at least 3.2 minutes (UCB). Sleep latency being lower does confirm previous findings summarized in [15], but the value of the difference is quite low. The alcohol group also has later bedtime, midsleep and wake times, with the biggest difference observed in wake times - at least 20.9 minutes (LCB). Mean sleep duration of recordings labeled with alcohol was found to be higher by at least 9.6 minutes (LCB).

It was hypothesized that snoring could be more common in the alcohol group (based on the findings in [16]). We can see that the alcohol group has higher mean snoring time by at least 28 seconds. The overall negative effects alcohol has on sleep quality could reflect in a lower subjective rating. The alcohol group has mean subjective rating lower by at least 0.17 (UCB).

Total of seven parameters were compared, the corrected $\alpha$ is therefore $\alpha = 0.001/7 \approx 0,00014$. The results are summarized in Tab. 5.6.

| Parameter | $N_1$ | $N_2$ | $\text{Mean}_1$ | $\text{Mean}_2$ | Diff. CI (95%) | $p$-value | $p_{boot}$ | Sig |
|---|---|---|---|---|---|---|---|---|
| timeToSleep [h] | 130217 | 5550070 | 0.41 | 0.47 | -0.053 (UCB) | $< 0.00001$ | $< 0.001$ | ✓ |
| midsleep [h] | 161738 | 5619291 | 4.46 | 4.17 | 0.284 (LCB) | $< 0.00001$ | $< 0.001$ | ✓ |
| bedtime [h] | 161738 | 5619291 | 24.75 | 24.51 | 0.231 (LCB) | $< 0.00001$ | $< 0.001$ | ✓ |
| waketime [h] | 161738 | 5619291 | 8.16 | 7.81 | 0.348 (LCB) | $< 0.00001$ | $< 0.001$ | ✓ |
| netSleepLength [min] | 161738 | 5619291 | 434.89 | 424.93 | 9.617 (LCB) | $< 0.00001$ | $< 0.001$ | ✓ |
| snoringTime [s] | 73524 | 3042614 | 691.73 | 655.82 | 28.59 (LCB) | $< 0.00001$ | $< 0.001$ | ✓ |
| subjectiveRating [-] | 114098 | 2250504 | 2.96 | 3.14 | -0.17 (UCB) | $< 0.00001$ | $< 0.001$ | ✓ |

Tab. 5.6: Results of comparisons of recordings with #alcohol (group 1) and #home (group 2) tags

### 5.3.2 How does caffeine affect sleep?

Recordings tagged with the #caffeine tag were again compared to the #home tagged recordings. Based on several studies [17] it was hypothesized that the caffeine group should have a harder time falling asleep and the sleep latency should therefore be increased. However, the null hypothesis could not be rejected. Differences in means of sleep duration, bedtimes and wake times were observed, but the absolute values are quite low and their practical significance is doubtful. Mean snoring time was observed to be lower in the caffeine group by approximately 1 to 1.5 minute. Subjective rating was found to be slightly lower in the caffeine group by at least -0.05 (UCB).

Total of seven parameters were compared, the corrected $\alpha$ is therefore $\alpha = 0.001/7 \approx 0,00014$. The results are summarized in Tab. 5.7.

| Parameter | $N_1$ | $N_2$ | $\text{Mean}_1$ | $\text{Mean}_2$ | Diff. CI (95%) | $p$-value | $p_{boot}$ | Sig |
|---|---|---|---|---|---|---|---|---|
| timeToSleep [h] | 41112 | 5550070 | 0.39 | 0.47 | -0.10 (LCB) | 1.00 | 1.00 | × |
| midsleep [h] | 49229 | 5619291 | 4.25 | 4.17 | (0.066,0.105) | $< 0.00001$ | $< 0.001$ | ✓ |
| bedtime [h] | 49229 | 5619291 | 24.56 | 24.51 | (0.034,0.075) | $< 0.00001$ | $< 0.001$ | ✓ |
| waketime [h] | 49229 | 5619291 | 7.92 | 7.81 | (0.092,0.134) | $< 0.00001$ | $< 0.001$ | ✓ |
| netSleepLength [min] | 49229 | 5619291 | 429.80 | 424.93 | (4.13,5.62) | $< 0.00001$ | $< 0.001$ | ✓ |
| snoringTime [s] | 23003 | 3042614 | 571.63 | 655.82 | (-98.83,-69.54) | $< 0.00001$ | $< 0.001$ | ✓ |
| subjectiveRating [-] | 34057 | 2250504 | 3.08 | 3.14 | -0.050 (UCB) | $< 0.00001$ | $< 0.001$ | ✓ |

Tab. 5.7: Results of comparisons of recordings with #caffeine (group 1) and #home (group 2) tags

### 5.3.3 Do lullabies improve sleep?

Lullaby tagged recordings were compared to the home tagged recordings. Reduced sleep latency in the lullaby tagged recordings was not observed in the data set. Although research suggest that music-aided relaxation can lead to heightened sleep quality [50], subjective quality rating was not found to be improved in the lullaby group in our data set. Snoring appears to be less common when using a lullaby. Snoring time appears to be lower by at least 39 seconds (UCB) when using a lullaby.

Total of three parameters were compared, the corrected $\alpha$ is therefore $\alpha = 0.001/3 \approx 0,00033$. The results are summarized in Tab. 5.8.

| Parameter | $N_1$ | $N_2$ | $\text{Mean}_1$ | $\text{Mean}_2$ | Diff. CI (95%) | $p$-value | $p_{boot}$ | Sig |
|---|---|---|---|---|---|---|---|---|
| timeToSleep [h] | 109413 | 5550070 | 0.56 | 0.47 | 0.10 (UCB) | 1 | 1.00 | × |
| snoringTime [s] | 58434 | 3042614 | 607.73 | 655.82 | -39.22 (UCB) | $< 0.00001$ | $< 0.001$ | ✓ |
| subjectiveRating [-] | 44233 | 2250504 | 3.15 | 3.14 | 0.005 (LCB) | 0.004 | 0.01 | × |

Tab. 5.8: Results of comparisons of recordings with #lullaby (group 1) and #home (group 2) tags

### 5.3.4 Does the moon phase affect sleep?

To see the effect the moon phase has on sleep parameters, #newmoon and #fullmoon tagged recordings were compared. Difference between means of sleep latency, midsleep, wake time and sleep duration in the two groups were determined to be statistically significant. The differences are, however, so small that we do not consider them to be practically significant.

Total of seven parameters were compared, the corrected $\alpha$ is therefore $\alpha = 0.001/7 \approx 0,00014$. Results are displayed in Tab. 5.9.

| Parameter | $N_1$ | $N_2$ | $\text{Mean}_1$ | $\text{Mean}_2$ | Diff. CI (95%) | $p$-value | $p_{boot}$ | Sig |
|---|---|---|---|---|---|---|---|---|
| timeToSleep [h] | 447122 | 439172 | 0.49 | 0.46 | (0.020,0.045) | $< 0.00001$ | $< 0.001$ | ✓ |
| midsleep [h] | 458255 | 450612 | 4.18 | 4.20 | (-0.035,-0.018) | $< 0.00001$ | $< 0.001$ | ✓ |
| bedtime [h] | 458255 | 450612 | 24.50 | 24.53 | (-0.033,-0.015) | $< 0.00001$ | $< 0.001$ | ✓ |
| waketime [h] | 458255 | 450612 | 7.83 | 7.85 | (-0.037,-0.019) | $< 0.00001$ | $< 0.001$ | ✓ |
| netSleepLength [min] | 458255 | 450612 | 427.13 | 426.77 | (0.01,0.70) | 0.04 | 0.04 | × |
| snoringTime [s] | 230596 | 226834 | 642.80 | 655.21 | (-19.61,-5.20) | 0.0007 | 0.002 | × |
| subjectiveRating [-] | 176535 | 173160 | 3.14 | 3.14 | (-0.005,0.008) | 0.68 | 0.69 | × |

Tab. 5.9: Results of comparisons of recordings with #newmoon (group 1) and #fullmoon (group 2) tags

### 5.3.5 How does sickness affect sleep?

Tags #sick and #home were compared. Mean subjective sleep quality rating was observed to be lower by at least 0.37 (UCB) in the sick group. Mean wake time was found to be later by at least 24 minutes (LCB) for sick tagged recordings. Mean sleep duration is at least 29.9 minutes (LCB) longer for #sick tagged recordings. Mean bedtime is occurring a few minutes earlier in the sick group. Sick tagged recordings have a higher mean snoring time by at least 71 seconds (LCB).

Total of seven parameters were compared, the corrected $\alpha$ is therefore $\alpha = 0.001/7 \approx 0,00014$. The results are summarized in Tab. 5.10.

| Parameter | $N_1$ | $N_2$ | $\text{Mean}_1$ | $\text{Mean}_2$ | Diff. CI (95%) | $p$-value | $p_{boot}$ | Sig |
|---|---|---|---|---|---|---|---|---|
| timeToSleep [h] | 32690 | 5550070 | 0.52 | 0.47 | (0.009,0.078) | 0.014 | 0.01 | × |
| midsleep [h] | 41201 | 5619291 | 4.32 | 4.17 | (0.13,0.17) | $< 0.00001$ | $< 0.001$ | ✓ |
| bedtime [h] | 41201 | 5619291 | 24.38 | 24.51 | (-0.15,-0.10) | $< 0.00001$ | $< 0.001$ | ✓ |
| waketime [h] | 41201 | 5619291 | 8.22 | 7.81 | 0.40 (LCB) | $< 0.00001$ | $< 0.001$ | ✓ |
| netSleepLength [min] | 41201 | 5619291 | 455.65 | 424.93 | 29.94 (LCB) | $< 0.00001$ | $< 0.001$ | ✓ |
| snoringTime [s] | 19045 | 3042614 | 741.64 | 655.82 | 71.02 (LCB) | $< 0.00001$ | $< 0.001$ | ✓ |
| subjectiveRating [-] | 31354 | 2250504 | 2.76 | 3.14 | -0.37 (UCB) | $< 0.00001$ | $< 0.001$ | ✓ |

Tab. 5.10: Results of comparisons of recordings with #sick (group 1) and #home (group 2) tags

### 5.3.6 How do dreams affect sleep?

Recordings tagged #gooddream and #baddream were compared. Subjective rating is higher for good dreams by almost 1 point (LCB). Mean sleep duration was observed to be longer by 4 to 6 minutes (CI 95%) for nights with good dreams. Wake time was also found to be later by 6 to 12 minutes (CI 95%).

Total of five parameters were compared, the corrected $\alpha$ is therefore $\alpha = 0.001/5 \approx 0,0002$. The results are summarized in Tab. 5.11.

| Parameter | $N_1$ | $N_2$ | $\text{Mean}_1$ | $\text{Mean}_2$ | Diff. CI (95%) | $p$-value | $p_{boot}$ | Sig |
|---|---|---|---|---|---|---|---|---|
| waketime | 40670 | 40007 | 8.21 | 8.01 | (0.16,0.22) | $< 0.00001$ | $< 0.001$ | ✓ |
| bedtime | 40670 | 40007 | 24.54 | 24.40 | (0.11,0.17) | $< 0.00001$ | $< 0.001$ | ✓ |
| netSleepLength | 40670 | 40007 | 445.57 | 440.38 | (4.01,6.37) | $< 0.00001$ | $< 0.001$ | ✓ |
| snoringTime | 21497 | 21570 | 543.95 | 592.45 | (-70.18,-26.83) | 0.00001 | $< 0.001$ | ✓ |
| subjectiveRating | 33631 | 30236 | 3.67 | 2.67 | 0.995 (LCB) | $< 0.00001$ | $< 0.001$ | ✓ |

Tab. 5.11: Results of comparisons of recordings with #gooddream (group 1) and #baddream (group 2) tags

## 5.4 Classification

In this section we try to determine how well the user is going to rate their sleep quality (*subjectiveRating*) based on the recorded sleep parameters. Classification, rather than regression was chosen, because the values of *subjectiveRating* take only on few discrete values and since the rating is subjective, a classification into fewer classes is more suitable. Classification into two classes is considered, where the two classes refer to recordings in which users rated their sleep as bad or good.

In addition to classifying, we are also interested in which of the features are the most important ones to determine sleep quality. Feature selection and models in which some measure of variable importance can be calculated are used for this purpose.

Variables used as inputs to the classifier were selected using the Relief algorithm. The following 9 variables were selected: *age*, *timeToSleep*, *alarmWakeDiff*, *netSleepLength*, *BMI*, *bedtime*, *snoringTime*, *waketime*, *midsleep*. Since the variable *midsleep* is calculated from values of *bedtime* and *waketime*, it was removed and thus 8 input variables remained. The sleep recordings were ranked according to subjective rating as bad (0) for values of *subjectiveRating* 0-2 and good(1) for values 4-5.

Recordings from the years 2015 and 2016 with non-missing values of the selected features and response (*subjectiveRating*) were selected and divided into training set (80%) and validation set (20%). Parameters of classifiers were optimized on the validation set. Data from the year 2017 were used as a testing set. The classifier algorithms that were used are

- *CART* (Classification and Regression Trees) – the implementation in *R*'s *rpart* package [36] was used
- *Bagging* – the bagging algorithm using classification trees (CART) as single classifiers, the implementation in *R*'s *adabag* package [37] was used
- *kNN* (*k* nearest neighbors) – the implementation in *R*'s *class* package [38] was used
- *Naive Bayes* – the implementation in *R*'s *e1071* package [39] was used
- *Boosting* – the AdaBoost algorithm using classification trees (CART) as single classifiers, the implementation in *R*'s *adabag* package [37] was used
- *LDA* – the implementation in *R*'s *MASS* package [38] was used
- *Logistic regression* (Log. reg.) – the implementation in *R*'s *stats* package [40] was used
- *Random forest* – the implementation in *R*'s *randomForest* package [41] was used

Performances of these classifier algorithms on the test set are in Tab. 5.12.

| Classifier | Accuracy | Sensitivity | Specificity | PPV | NPV | Balanced acc. | AUC |
|---|---|---|---|---|---|---|---|
| CART | 0.6307 | 0.8089 | 0.3752 | 0.6500 | 0.5778 | 0.5920 | 0.6159 |
| Bagging | 0.6300 | 0.7987 | 0.3881 | 0.6519 | 0.5734 | 0.5934 | 0.6178 |
| kNN (3) | 0.6507 | 0.7185 | 0.5536 | 0.6978 | 0.5782 | 0.6360 | 0.6360 |
| Naive Bayes | 0.6340 | 0.8089 | 0.3831 | 0.6529 | 0.5829 | 0.5960 | 0.6486 |
| Boosting | 0.6323 | 0.7719 | 0.4322 | 0.6610 | 0.5691 | 0.6020 | 0.6595 |
| LDA | 0.6380 | 0.8654 | 0.3118 | 0.6434 | 0.6176 | 0.5886 | 0.6672 |
| Log. reg. | 0.6385 | 0.8622 | 0.3175 | 0.6444 | 0.6164 | 0.5899 | 0.6676 |
| Random Forest | 0.6817 | 0.7767 | 0.5454 | 0.7102 | 0.6299 | 0.6610 | 0.7164 |

Tab. 5.12: Performances of classifier algorithms on the test set (sorted by AUC)

It can be seen that the performance gained by bagging and boosting to increase the performance of a single CART classifier is relatively low and that even some of the simple linear classifiers have better performance. Most of the classifiers exhibit a relatively low specificity (accuracy of classification of sleep recordings labeled as "bad"). Random forest has the best performance amongst the classifiers in terms of the AUC and accuracy.

| | Variable importance ranking | | | | | |
|---|---|---|---|---|---|---|
| Variable | Log. reg. | CART | Bag | Boost | Random Forest | Avg |
| netSleepLength | 1 | 1 | 1 | 3 | 3 | 1.8 |
| BMI | 7 | 2 | 3 | 1 | 1 | 2.8 |
| age | 5 | 3 | 2 | 2 | 2 | 2.8 |
| alarmWakeDiff | 2 | 5 | 5 | 4 | 4 | 4.0 |
| waketime | 4 | 4 | 4 | 5 | 5 | 4.4 |
| bedtime | 6 | 7 | 6 | 8 | 6 | 6.6 |
| snoringTime | 3 | 8 | 7 | 7 | 8 | 6.6 |
| timeToSleep | 8 | 6 | 8 | 6 | 7 | 7.0 |

Tab. 5.13: Ranking of variables by importance

Measures of importance of variables were calculated for classifiers for which such a calculation is possible. In logistic regression, the absolute value of the $t$-statistic for each model parameter was used to measure variable importance. For random forest classifier, the mean decrease in Gini index was used. In CART decision tree, the reduction in the mean squared error attributed to each variable at each split is calculated and the sum is used to measure variable importance. In bagged trees, the same methodology for single tree is applied to all trees.

Since several different importance measures were used amongst the classifiers, only the ranking of the variables by importance was compared between the classifiers. Ranking of variables by importance in each of the classifiers is in Tab. 5.13. The variables are sorted by their average ranking.

Sleep duration was determined to be the most important variable in determining the subjective rating of sleep quality. Age along with BMI were determined to the next most important variables. Sleep latency was found to be the least important variable for determining rating.

## 5.5 Clustering

The goal of this section is to find meaningful subgroups of users based on their sleep parameters, which could lead to new insights and aid in interpretation of the data. Since there can be more data points for each recorded parameter of each user, the data of one user have a form of multivariate time series rather than static data. Therefore, the problem of clustering time series is addressed. A short review of clustering algorithms and approaches used in this chapter is given in section 3.5. The described clustering algorithms are applied to the *Sleep as Android* data set.

### 5.5.1 Preprocessing

**Time series** For the purpose of clustering time series, time series of one year of the parameter *netSleepLength* (sleep duration) were chosen. The year 2016 was chosen since it contains the most recordings. Users with more than 250 sleep recordings with sleep durations between 4 to 14 hours in 2016 were selected. Only users who have less than 5 subsequent missing recordings were chosen.

The missing values were approximated by linear approximation separately for weekday and weekend recordings. Missing values at the start and end of the recordings were imputed with median value (also separately for weekdays and weekends).

The time series were first clustered without smoothing, subsequently also time series smoothed using Local Polynomial Regression Fitting (function *loess* in R) with $\alpha = 0.1$ separately for weekdays and weekends were used (see Fig. 5.4 for comparison of the smoothed and original time series).

Each of the time series was normalized to have mean zero and standard deviation of 1 prior to clustering.

**Features** The subset of users that was chosen and features that were extracted is the same as in section 4.2 (without the limitation on BMI). However, since the correlation of the parameters which contain deep sleep ratios on weekend and weekdays is very high, only one parameter of these was kept ($DSratio\_wday$). Therefore, total of 10 features were used for clustering. All of the features were scaled to have mean zero and standard deviation of 1 before clustering.

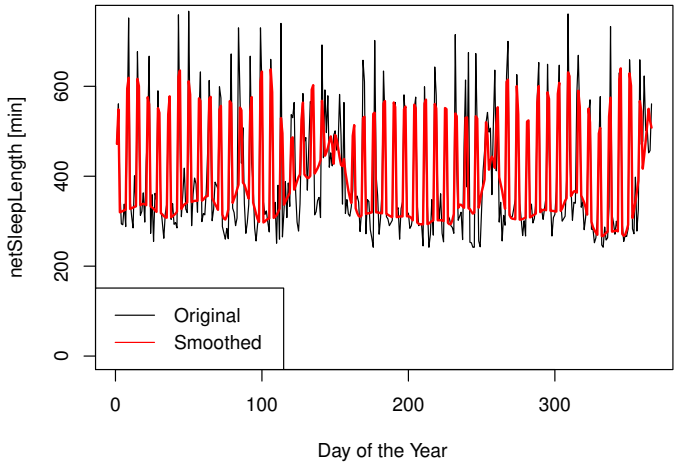



Fig. 5.4: Comparison of smoothed and original sleep duration series of one user

### 5.5.2 Clustering extracted features

First, the simplest case where features were extracted from the time series and clustering is performed on these features as static data is realized.

| Algorithm | Index \ Clusters | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| | DB | 2.38 | 2.12 | 2.02 | 1.90 | 1.68 | 1.76 | 1.82 | 1.69 |
| $k$-means | Dunn | 0.05 | 0.06 | 0.02 | 0.02 | 0.02 | 0.05 | 0.03 | 0.05 |
| | Sil | 0.13 | 0.11 | 0.12 | 0.10 | 0.10 | 0.10 | 0.09 | 0.09 |
| | DB | 2.85 | 2.19 | 2.18 | 2.09 | 1.91 | 2.00 | 1.86 | 1.90 |
| PAM | Dunn | 0.05 | 0.05 | 0.05 | 0.05 | 0.04 | 0.01 | 0.01 | 0.01 |
| | Silhouette | 0.10 | 0.11 | 0.10 | 0.08 | 0.08 | 0.08 | 0.08 | 0.07 |
| | DB | 3.05 | 3.53 | 3.84 | 3.49 | 2.89 | 2.64 | 2.97 | 4.40 |
| GAM | Dunn | 0.05 | 0.05 | 0.04 | 0.04 | 0.04 | 0.04 | 0.05 | 0.04 |
| | Silhouette | 0.09 | 0.01 | 0.01 | −0.01 | −0.00 | −0.01 | 0.03 | 0.01 |
| | DB | 1.70 | 1.01 | 1.04 | 1.05 | 1.05 | 1.05 | 1.05 | 1.10 |
| H. | Dunn | 0.12 | 0.13 | 0.09 | 0.09 | 0.10 | 0.08 | 0.09 | 0.09 |
| | Silhouette | 0.22 | 0.24 | 0.13 | 0.16 | 0.16 | 0.13 | 0.14 | 0.13 |

Tab. 5.14: Feature clustering results. Green fields denote the best value for the algorithm, red fields denote the best value overall.

Various clustering algorithms with the number of clusters ranging from 2 to 9 were used on the data and were compared using internal and stability measures. At the first run, two clusters were determined to be the best number. However, one of the clusters contained only 3 observations which were dismissed as outliers and the process was repeated. The performance of various algorithms as measured by internal measures are in Tab. 5.14. The best algorithms and number of clusters according to stability measures are in Tab. 5.15.

When we look at the three clusters created using hierarchical clustering, which were determined to be the best using internal measures, one of the clusters consists only of four observations. We therefore examine two clusters created by hierarchical clustering, which is also determined to be

the best number of clusters by two of the stability indexes and has the second best values of internal measures. The number of observations in the second of the clusters is still relatively low (34) compared to the other cluster (1818). Next, we examine the distributions of parameters in the two clusters which are displayed in Fig. 5.5.

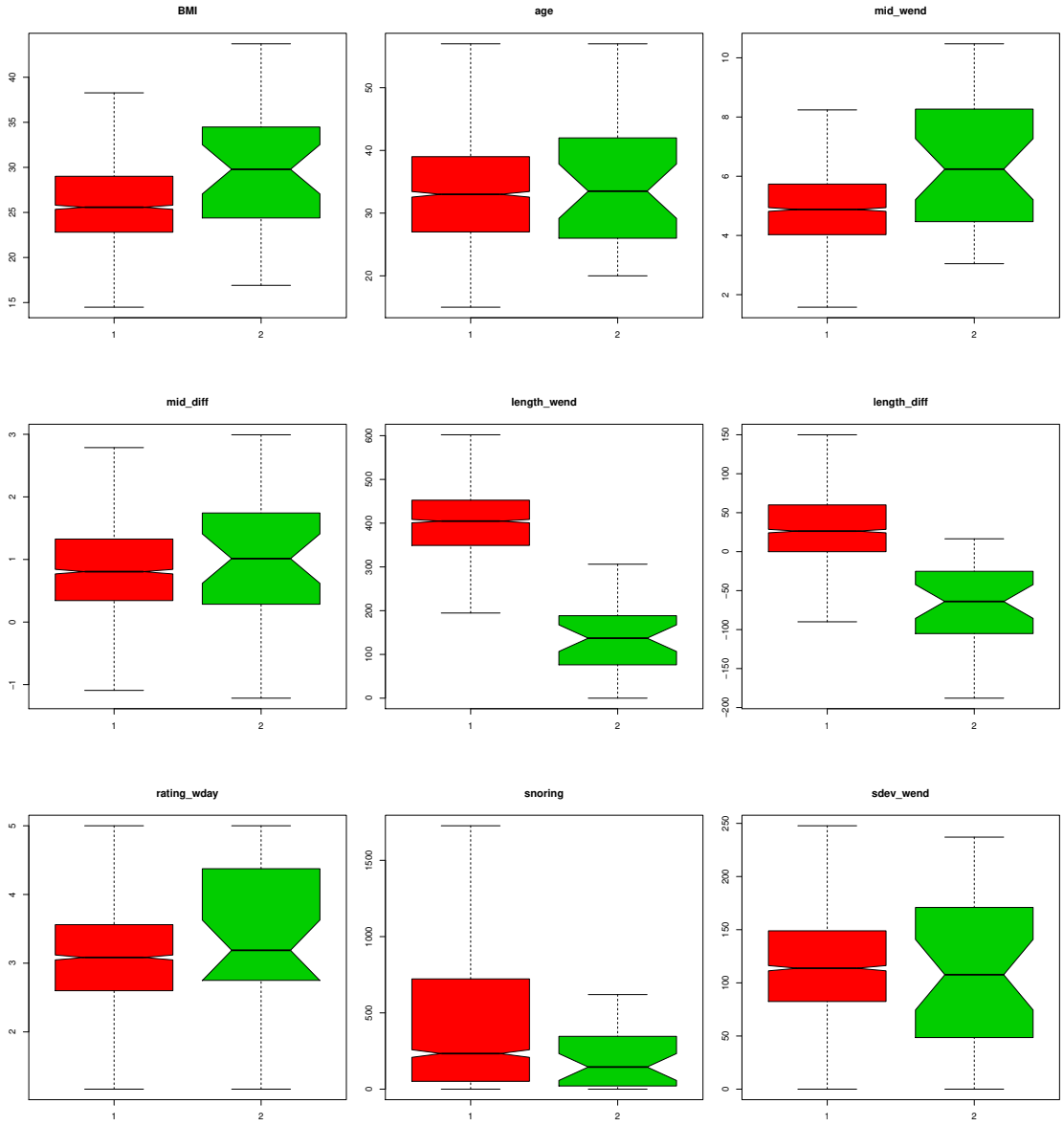



Fig. 5.5: Distributions of parameters in clusters

| Measure | Score | Method | Clusters |
|---|---|---|---|
| APN | 0.0016 | hierarchical | 2 |
| AD | 3.5548 | kmeans | 9 |
| ADM | 0.0210 | hierarchical | 2 |
| FOM | 0.9507 | kmeans | 9 |

Tab. 5.15: Best results for stability measures of feature clustering

### 5.5.3 Clustering time series representations

Next, various representations of time series were calculated and clustered. The calculation of time series representations serves mainly for dimension reduction and the resulting representations can be then clustered with the use of algorithms commonly used for static data. The representations were calculated using the *TSrepr* package [30] and subsequently clustered using the PAM algorithm. Clustering was calculated for numbers of clusters ranging from 2 to 9 and for each clustering three internal clustering validity measures were calculated. Total of six representations of time series were calculated. The representations that were used are described in 3.5.3.3.

The resulting values of three internal clustering validity measures for 2 to 9 clusters and various representations are shown in Tab. 5.16. Results for 2 clusters with the *SP* and *GAM* representations are very alike and the centroids of the clusters are similar to the ones obtained by hierarchical clustering in Fig. 5.6. Other representations do not yield very good results for any number of clusters with the resulting clusters' centroids being very much alike.

| Repr. | Index \ Clusters | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| SP(7) | DB | 1.05 | 1.14 | 1.64 | 1.69 | 1.28 | 1.90 | 1.37 | 1.53 |
| | Dunn | 0.03 | 0.03 | 0.01 | 0.01 | 0.03 | 0.01 | 0.03 | 0.03 |
| | Silhouette | 0.36 | 0.25 | 0.17 | 0.15 | 0.14 | 0.13 | 0.12 | 0.11 |
| GAM(7) | DB | 0.93 | 1.06 | 1.42 | 1.37 | 1.65 | 1.16 | 1.79 | 1.60 |
| | Dunn | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Silhouette | 0.40 | 0.26 | 0.23 | 0.18 | 0.17 | 0.15 | 0.14 | 0.15 |
| DFT(21) | DB | 4.63 | 4.05 | 4.16 | 3.85 | 3.90 | 3.97 | 4.03 | 3.94 |
| | Dunn | 0.13 | 0.16 | 0.15 | 0.15 | 0.13 | 0.15 | 0.15 | 0.15 |
| | Silhouette | 0.04 | 0.03 | 0.03 | 0.03 | 0.03 | 0.02 | 0.02 | 0.02 |
| DWT(haar) | DB | 3.76 | 3.36 | 3.12 | 3.65 | 3.42 | 2.61 | 3.36 | 3.39 |
| | Dunn | 0.14 | 0.15 | 0.03 | 0.16 | 0.16 | 0.16 | 0.16 | 0.16 |
| | Silhouette | 0.06 | 0.04 | 0.04 | 0.03 | 0.03 | 0.03 | 0.02 | 0.02 |
| DCT(21) | DB | 3.26 | 3.90 | 3.89 | 3.77 | 3.18 | 3.16 | 3.31 | 2.47 |
| | Dunn | 0.14 | 0.13 | 0.09 | 0.09 | 0.14 | 0.14 | 0.13 | 0.13 |
| | Silhouette | 0.07 | 0.04 | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 |
| PAA(7) | DB | 5.27 | 4.84 | 5.30 | 5.42 | 4.24 | 4.28 | 4.45 | 4.08 |
| | Dunn | 0.25 | 0.25 | 0.25 | 0.10 | 0.11 | 0.11 | 0.11 | 0.11 |
| | Silhouette | 0.03 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |

Tab. 5.16: Reults of clustering time series representations. Green color denotes the best value of the index for the representation. SP(7) - mean seasonal profile for 7 days. DFT(21) - discrete Fourier transform (first 21 coefficients). DWT(haar) - wavelet transform, haar wavelet, DCT(21) - cosine transform (first 21 coefficients), PAA(7) - piecewise aggregate approximation (mean of 7 days)

### 5.5.4 Raw data clustering

First, the unsmoothed time series were clustered using various approaches. Results can be seen in Tab. 5.17. The best clustering as determined by the three internal validity indexes is hierarchical clustering with two clusters. The resulting centroids from this clustering are shown in Fig. 5.6. The numbers of observations in the two clusters are : $N_1 = 1435$, $N_2 = 189$.

| Algorithm | Index \ Clusters | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| PAM (L2) | DB | 2.78 | 1.87 | 1.93 | 2.38 | 2.38 | 2.00 | 2.19 | 2.44 |
| | Dunn | 0.51 | 0.64 | 0.69 | 0.49 | 0.53 | 0.68 | 0.49 | 0.48 |
| | Silhouette | 0.02 | 0.04 | 0.04 | 0.01 | 0.01 | 0.03 | 0.00 | 0.01 |
| PAM (DTW) | DB | 2.26 | 2.01 | 2.05 | 2.19 | 2.08 | 2.04 | 2.06 | 2.11 |
| | Dunn | 0.55 | 0.64 | 0.66 | 0.63 | 0.02 | 0.66 | 0.02 | 0.60 |
| | Silhouette | 0.01 | 0.01 | 0.00 | 0.00 | −0.00 | −0.00 | −0.00 | −0.01 |
| H. (L2, avg) | DB | 1.23 | 1.74 | 1.80 | 1.84 | 1.94 | 1.96 | 1.99 | 1.99 |
| | Dunn | 0.72 | 0.72 | 0.72 | 0.72 | 0.71 | 0.71 | 0.71 | 0.71 |
| | Silhouette | 0.04 | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 |
| TADPole | DB | 12.24 | 12.58 | 13.69 | 14.18 | 15.94 | 17.24 | 16.62 | 17.09 |
| | Dunn | | | | | | | | |
| | Silhouette | | | | | | | | |
| k-shape | DB | 2.01 | 2.31 | 2.90 | 3.98 | 6.17 | 3.69 | 3.40 | 3.56 |
| | Dunn | 0.57 | 0.57 | 0.53 | 0.49 | 0.32 | 0.43 | 0.48 | 0.47 |
| | Silhouette | 0.04 | 0.04 | 0.04 | 0.02 | −0.00 | 0.01 | 0.02 | 0.01 |

Tab. 5.17: Green value denotes the best value of index for each algorithm. Red value denotes the best value of index overall. H. – Hierarchical clustering

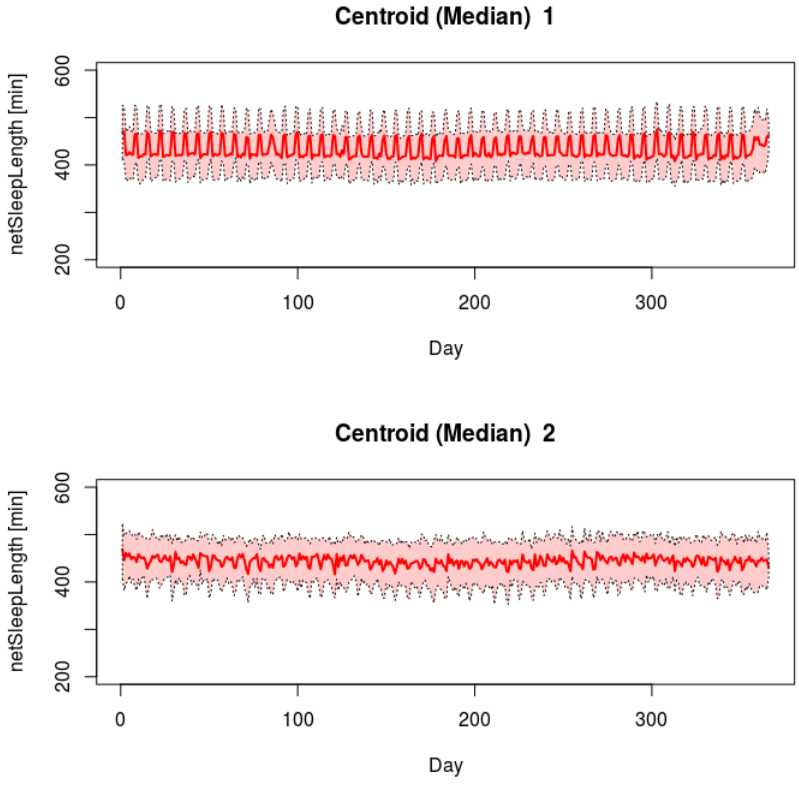



Fig. 5.6: Centroids for hierarchical clustering with two clusters, *L2* distance and average linkage. The median is displayed in red, the black dotted lines are the 1st and the 3rd quartiles. $N_1 = 1435$, $N_2 = 189$

The clusterings with more than two clusters, which were determined by some internal validity indexes as the best, were also examined. The additional clusters are, however, usually very similar and do not provide any interesting grouping (see Fig. 5.7 for centroids of PAM (L2) clustering with four clusters).

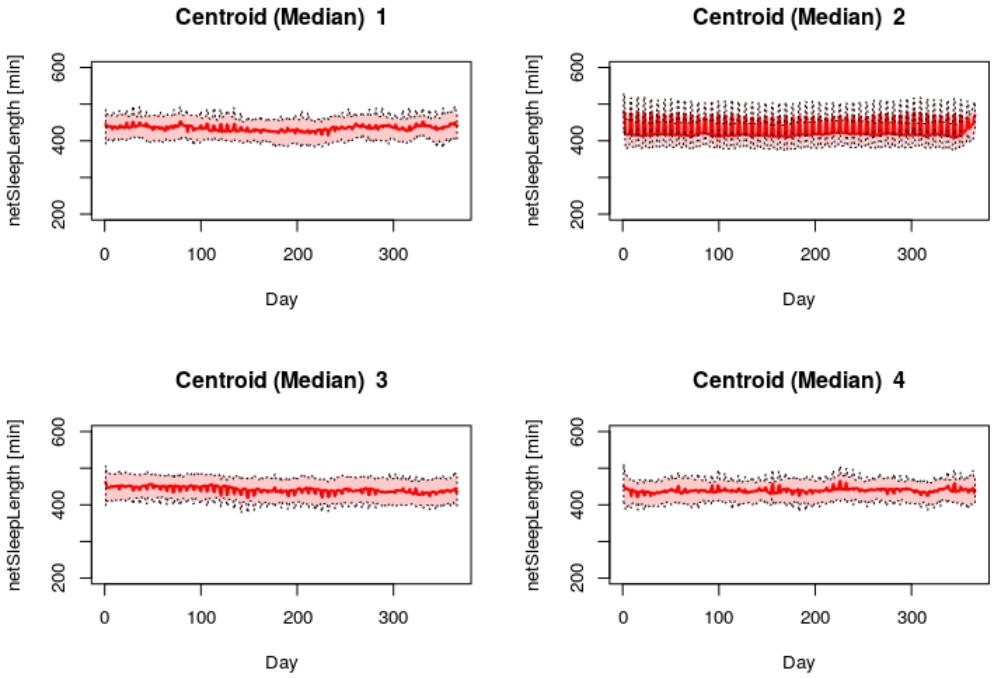



Fig. 5.7: Centroids for PAM (L2) clustering. The median is displayed in red, the black dotted lines are the 1st and the 3rd quartiles. $N_1 = 142$, $N_2 = 1105$, $N_3 = 237$, $N_4 = 140$

Results of clustering the smoothed time series are in Tab. 5.18. Hierarchical clustering with two clusters was again determined to be the best by two of the three internal validity indexes. The resulting centroids of hierarchical clustering with two clusters are in Fig. 5.8. Numbers of observations in the two clusters are: $N_1 = 1373$, $N_2 = 251$.

| Algorithm | Index \ Clusters | 2.00 | 3.00 | 4.00 | 5.00 | 6.00 | 7.00 | 8.00 | 9.00 |
|---|---|---|---|---|---|---|---|---|---|
| PAM (L2) | DB | 1.35 | 1.70 | 1.82 | 1.89 | 2.66 | 2.66 | 2.69 | 2.42 |
| | Dunn | 0.48 | 0.46 | 0.49 | 0.42 | 0.28 | 0.28 | 0.28 | 0.35 |
| | Silhouette | 0.13 | 0.10 | 0.11 | 0.08 | 0.02 | 0.02 | 0.02 | 0.03 |
| PAM (DTW) | DB | 2.00 | 2.28 | 2.16 | 2.10 | 2.15 | 2.14 | 2.34 | 2.01 |
| | Dunn | 0.39 | 0.37 | 0.44 | 0.44 | 0.40 | 0.41 | 0.40 | 0.40 |
| | Silhouette | 0.07 | 0.02 | 0.02 | 0.02 | 0.01 | 0.00 | 0.00 | 0.00 |
| H. (L2, avg) | DB | 1.22 | 1.67 | 1.81 | 1.87 | 1.86 | 1.86 | 1.83 | 1.91 |
| | Dunn | 0.51 | 0.49 | 0.49 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 |
| | Silhouette | 0.12 | 0.11 | 0.09 | 0.09 | 0.09 | 0.08 | 0.08 | 0.08 |
| TADPole | DB | 9.39 | 13.11 | 13.61 | 14.74 | 14.35 | 13.49 | 14.64 | 13.96 |
| | Dunn | | | | | | | | |
| | Silhouette | | | | | | | | |
| k-shape | DB | 1.43 | 1.66 | 4.00 | 2.81 | 3.31 | 2.86 | 2.98 | 3.60 |
| | Dunn | 0.25 | 0.26 | 0.13 | 0.18 | 0.16 | 0.15 | 0.16 | 0.12 |
| | Silhouette | 0.16 | 0.13 | 0.02 | 0.05 | 0.03 | 0.03 | 0.03 | 0.01 |

Tab. 5.18: Green value denotes the best value of index for each algorithm. Red value denotes the best value of index overall. H. – Hierarchical clustering

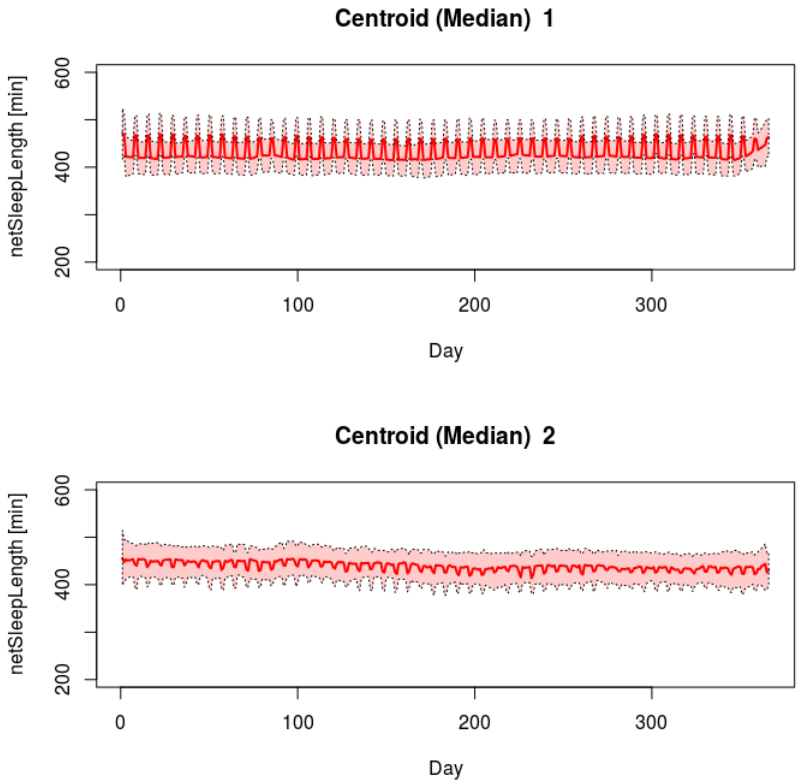
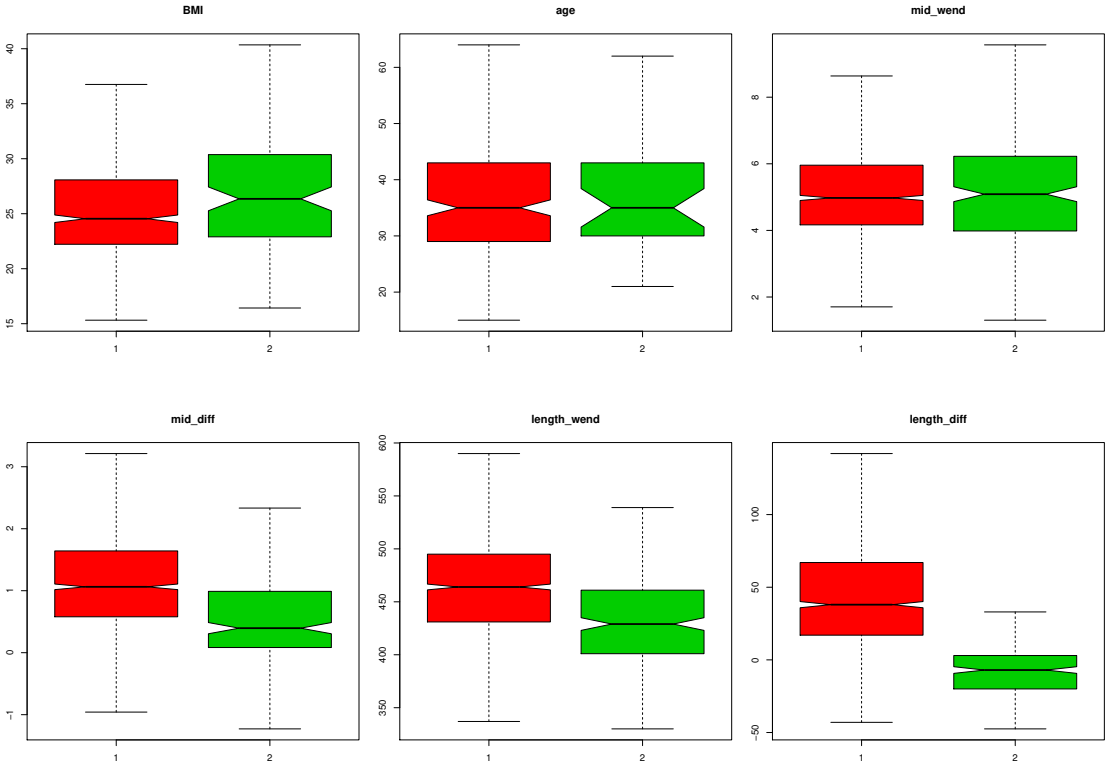



Fig. 5.8: Centroids for hierarchical clustering with two clusters, *L2* distance and average linkage. The median is displayed in red, the black dotted lines are the 1st and the 3rd quartiles. $N_1 = 1373$, $N_2 = 251$



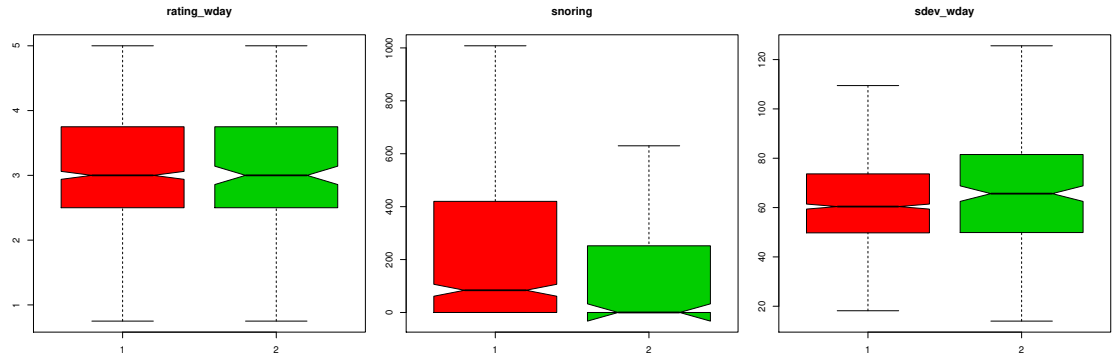



Fig. 5.9: Distributions of parameters in clusters obtained by hierarchical clustering

Resulting centroids for the $k$-shape clustering which was determined to be the best in terms of the silhouette width is in Fig. 5.10.

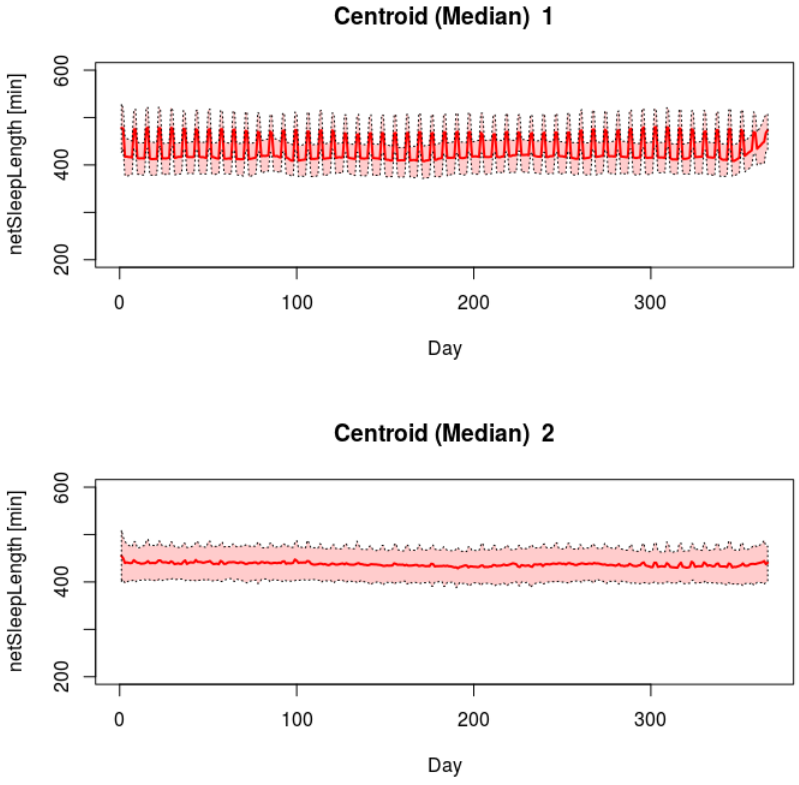



Fig. 5.10: Centroids for $k$-shape clustering. The median is displayed in red, the black dotted lines are the 1st and the 3rd quartiles. $N_1 = 943$, $N_2 = 681$

We also examined the clusterings with more than two clusters, however, some of the clusters tend to be very similar, see Fig. 5.11 , where centroids are displayed for four clusters resulting from PAM with *L2* distance. Three of these centroids seem to be very similar.

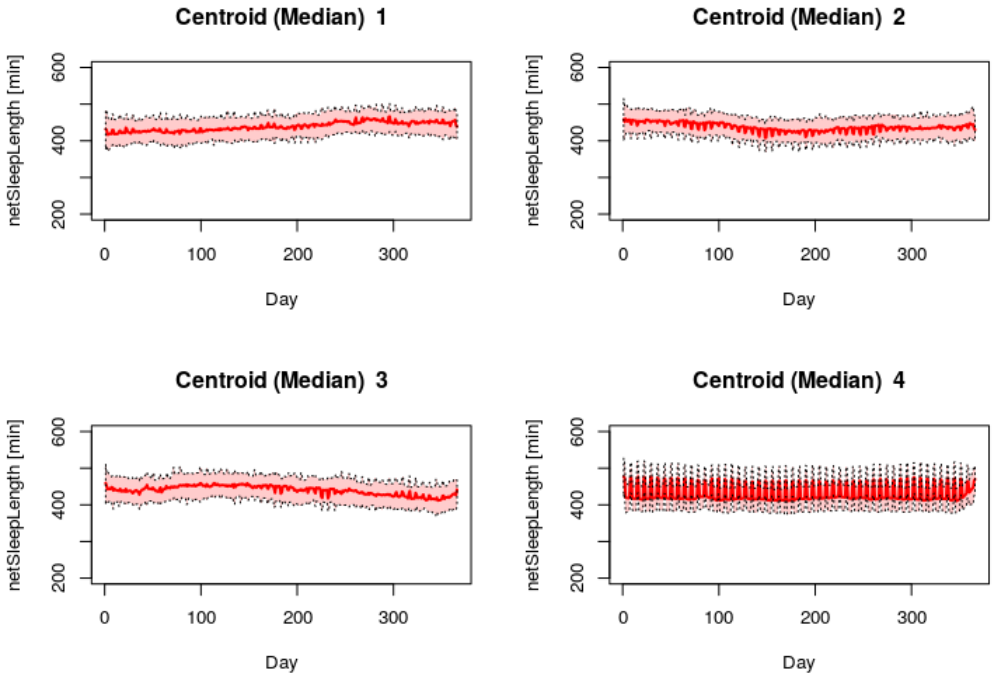



Fig. 5.11: Centroids for PAM (L2) clustering. The median is displayed in red, the black dotted lines are the 1st and the 3rd quartiles. $N_1 = 173$, $N_2 = 211$, $N_3 = 131$, $N_4 = 1109$

### 5.5.5 Clustering conclusion

It seems that two is the most appropriate number of clusters for the studied data. Similar clusters were obtained using different clustering methods, although the numbers of time series in the clusters tend to differ. From the results of clustering time series representations we can see that the time series are clustered by the users' sleeping habits during week rather than changes of sleeping habits during the year, since clustering of seasonal profiles of one week yields similar results to other methods. On the other hand, the piecewise aggregate approximation of seven days, which effectively cancels the influence of sleeping habits during weekdays, does not yield any interpretable results.

Smoothing the time series led to more robust results with almost all of the validity indexes indicating two to be the best number of clusters for every clustering algorithm. Subjectively, the hierarchical clustering with Euclidean distance and with average linkage used on the smoothed time series and even on the original time series seems to be the best choice for the data. $k$-Shape has also good results, but the resulting cluster centroids subjectively seem to be more similar to each other than the ones obtained by hierarchical clustering.

One of the resulting clusters includes users who tend to sleep longer on weekends. These users are evening types and sleep longer on weekends due to the sleep debt accumulated during the week.

The other group contains users whose sleep durations are comparable on weekdays and weekends or their sleeping schedule is "inverted" compared to the users in the first cluster. It also seems that the BMI of these users tends to be higher and their snoring time lower (see Fig. 5.9 – the "notches" on the boxplots signify 95% confidence interval for the median). The users in the second cluster also seem to have more irregular sleep during weekdays.

# 6 Conclusion

In the first part of data analysis we have tested several hypotheses about sleep parameters. We have carried out $t$-test for unequal variances and bootstrap hypothesis tests and we observed that the resulting $p$ values tend to be very similar. Therefore, we conclude that the $t$-test for unequal variances is sufficient for the purposes of this thesis, where the number of observations in the tested samples was generally very high, ranging from tens of thousands to millions. Several findings related to sleep science were confirmed in this thesis with the use of the data from the *Sleep as Android* data set.

For example, we have observed elderly people to have significantly lower eveningness than the youngest age group. Next, we observed that alcohol reduces sleep latency, which confirms previous findings. The difference in means is, however, rather low – based on our results alcohol reduces the time needed to fall asleep only by 3.2 minutes. Alcohol reduced sleep quality rating by at least 0.17 points (all of the numbers presented are based on 95% confidence intervals for the difference in means from section 5.3). The recordings with alcohol label were observed to have later bedtimes, midsleep and wake times, which was expected since alcohol is usually consumed as a part of social activities and celebrations when people tend to go to sleep at later times. Users were found to snore more after ingestion of alcohol by at least 28.59 seconds. We must note that although our findings about alcohol confirm the findings from scientific literature, we have no information about the time when alcohol was ingested and in what amounts do users ingest alcoholic beverages. These factors may greatly affect how alcohol influences sleep.

We have also observed a small drop in sleep quality rating for users using caffeine, but only by 0.05 points. We could, however, not confirm that caffeine causes prolonged sleep latency, which was observed previously in scientific literature. As with alcohol, we do not have information about the time of ingestion or the doses in which caffeine was used.

On the other hand, we could not confirm that lullabies decrease sleep latency or subjective rating. We did, however, observe that users using lullabies tend to snore less by at least 39.22 seconds. To our knowledge there is no study that relates lullabies to shortened snoring periods. This finding should therefore be studied more thoroughly. There are, however, studies that relate listening to music before sleeping to improved sleep quality [50]. In the Sleep as Android application there are several sound tracks that users can choose from to use as a lullaby, including binaural beats, but we do not possess the information about which one of them was used by the user or for how long it was being played prior to falling asleep. These factors may have an effect on the influence of lullabies on sleep quality, since the previously carried out studies identified only certain styles of music, such as jazz or classical music, to aid the sleeping process.

We did not observe moon phase to affect sleep duration. We have found that users who tag their recordings with the sick tag have a longer sleep duration by at least 29.9 minutes, have higher snoring time by at least 71 seconds and rate their sleep quality lower by at least 0.37 points. Sleep duration has been found to be the most important variable in determining subjective sleep quality rating. The sick users, however, tend to rate their sleep lower even when their sleep duration is prolonged. Medication could also contribute to altering sleep parameters. Furthermore, dreams have been found to affect sleep quality ratings in such a way that sleep recordings with good dreams have had higher ratings than bad dreams by at least 1 point.

Another of our findings is that important nation-wide events have impact on users' sleep scheduling. We compared sleep durations in the United Kingdom during the Brexit vote to sleep durations of other years. We did the same for presidential elections in the United States of America. We have found that both of these events impacted users' sleep durations and caused them to sleep less, which is in accordance with our initial hypothesis.

Moreover, we addressed the clustering problem and tried various approaches to finding subgroups based on users' sleep patterns. We selected time series of sleep durations of length one year for the purpose of clustering. We have evaluated the clustering results by the use of internal clustering validity measures and stability measures. First, we have tried extracting summarizing features for each user's time series and subsequently clustering them. This approach is very simple, but tended to result in small clusters which could be dismissed as outliers. We also tried clustering various time series representations as well as the original and smoothed time series. Based on clustering the time series representations, we have arrived to the conclusion that users tend to get clustered based on their sleep scheduling habits during the week rather than by slower monthly or yearly trends. Clustering the smoothed time series proved to be the most robust approach since almost all of the resulting internal measures of clusters suggested the same number of clusters to be the best option. We examined resulting clusterings with various number of clusters, but we have found that for number of clusters higher than two, some of the clusters tend to be very similar. Therefore we conclude that there are two clusters present among the users. These clusters correspond to the two chronotypes – evening and morning types. One of the groups has a relatively large difference in their sleep scheduling between weekdays and weekends in terms of sleep duration and midsleep and corresponds to the evening types, i.e. owls. This group has also been observed to have lower median of BMI and higher snoring time. The other group corresponds to morning types and differs from the other group mostly by a lower difference of sleep duration between weekdays and weekends.

The *Sleep as Android* data set offers many more opportunities for sleep parameters to be studied. For example, country-to-country differences in sleep parameters or influence of daylight saving time on sleep could be analyzed. Another interesting question that was not addressed in this thesis is the effect of sun on sleep parameters. Further intriguing question is whether the smart alarm does really help users to wake up more easily. Recordings could be classified by subjective sleep quality rating for each user individually. Including tags which were found to alter sleep quality rating as features could help improving the classifier performance. Studying the issue of sleep deprivation affecting users' cognitive performance would also be very interesting, but that question is not possible to address with the current data from the application. Since the information about users' gender is present in a relatively low number of recordings and the number of recordings labeled as female is particularly low, we have not addressed sleep differences among genders.

Usage of sleep monitoring applications or wearable devices could have an immense social impact and has potential to improve health of their users by advising and motivating them to have better sleep habits. It could also lead to improved users' safety by detecting sleep deprivation and advising users not to attempt any activities that could be dangerous when their cognitive performance is lowered.

From our findings we conclude that this method of collecting sleep scheduling data by smartphone application is valid and because of its ease of use, it can be used to study sleep scheduling in the large population of its users. However, to make use of parameters such as deep sleep ratio and number of sleep cycles, the sleep stage detection method should be validated against polysomnography. Research would also benefit from more detailed information about users and their recorded parameters.

# 7 References


[1] M. Kantardzic (2011), *Data mining: concepts, models, methods, and algorithms*, 2nd ed., John Wiley & Sons, Inc., Hoboken, New Jersey, ISBN 978-1-118-02912-1

[2] O. J. Walch, A. Cochran, D. B. Forger (2016), *A global quantification of "normal" sleep schedules using smartphone data*, Science Advances 06 May 2016, Vol. 2, no. 5, e1501705

[3] C.J. Kramer et al. (1999), *Age differences in sleep-wake behavior under natural conditions* Personality and Individual Differences 27 853-860

[4] Randler, C. et al. (2017), *From Lark to Owl: developmental changes in morningness-eveningness from new-borns to early adulthood.* Scientific Reports 7:45874

[5] T. Roenneberg, A. Wirz-Justice and M. Merrow (2003), *Life between Clocks: Daily Temporal Patterns of Human Chronotypes.* Journal of Biological Rhythms, Vol. 18 No. 1

[6] A. Adan et al. (2012), *Circadian typology: A comprehensive review*, Chronobiology International, 2012 Nov;29(9):1153-75

[7] W. H. Moorcroft (2013), *Understanding Sleep and Dreaming.* Springer, ISBN 978-1-4614-6467-9

[8] Michael A. Grandner et al. (2015), *The Relationship between Sleep Duration and Body Mass Index Depends on Age*, Obesity (Silver Spring). 23(12): 2491–2498

[9] T. W. Liao (2005), *Clustering of time series data – a survey*, Pattern Recognition 38 1857 – 1874

[10] R. Allada, J. M. Siegel (2008), *Unearthing the Phylogenetic Roots of Sleep.* Current Biology 18, R670–R679

[11] J. Schenkein (2006), *Self-management of Fatal Familial Insomnia. Part 2: Case Report*, Medscape General Medicine, 8(3): 66.

[12] M. H. Schmidt (2014), *The energy allocation function of sleep: A unifying theory of sleep, torpor, and continuous wakefulness.* Neuroscience and Biobehavioral Reviews 47 122–15

[13] R. S. Cantor (2015), *The evolutionary origin of the need to sleep: an inevitable consequence of synaptic neurotransmission?,* Frontiers in Synaptic Neuroscience, Volume 7, Article 15

[14] Yetish et al. (2015), *Natural Sleep and Its Seasonal Variations in Three Pre-industrial Societies*, Current Biology 25, 2862–2868

[15] Ebrahim et al. (2013), *Alcohol and Sleep I: Effects on Normal Sleep*, Alcoholism: Clinical and Experimental Research, Vol. 37, No. 4,

[16] F. G. Issa, C. E. Sullivan (1982), *Alcohol, snoring and sleep apnoea*, Journal of Neurology, Neurosurgery, and Psychiatry 45:353-359

[17] T. Roehrs, T. Roth (2008), *Caffeine: Sleep and daytime sleepiness*, Sleep Medicine Reviews 12, 153–162

[18] D. Adler, C. Gläser, O. Nenadic, J. Oehlschlägel and W. Zucchini (2014). *ff: memory-efficient storage of large data on disk and fast access functions*. R package version 2.2-13. `https://CRAN.R-project.org/package=ff`

[19] D. Kahle and H. Wickham (2013). ggmap: Spatial Visualization with ggplot2. The R Journal, 5(1), 144-161. `http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf`

[20] T. Althoff, E. Horvitz, R. W. White, J. Zeitzer (2017), *Harnessing the Web for Population-Scale Physiological Sensing: A Case Study of Sleep and Performance*, Proceedings of the 26th International Conference on World Wide Web, Pages 113-122, Perth, Australia — April 03 - 07

[21] D. E. Hinkle, W. Wiersma, S. G. Jurs (2003), *Applied Statistics for the Behavioral Sciences*. 5th ed. Boston: Houghton Mifflin

[22] E. Helm et al. (2011),*REM Sleep Depotentiates Amygdala Activity to Previous Emotional Experiences*, Current Biology 21, 2029–2032

[23] G. Brock et al (2008), *clValid: An R Package for Cluster Validation*. Journal of Statistical Software, 25(4), 1-22. `http://www.jstatsoft.org/v25/i04/`

[24] P. J. Rousseeuw (1987), *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*. Journal of Computational and Applied Mathematics 20 53-65

[25] D. L. Davies, D. W. Bouldin (1979), *A Cluster Separation Measure*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. pami-1, No. 2,

[26] C. C. Aggarwal (2015), *Data Mining: The Textbook*, Springer, ISBN 978-3-319-14142-8

[27] E. Keogh, C. A. Ratanamahatana (2004), *Exact indexing of dynamic time warping*, Knowledge and Information Systems, DOI 10.1007/s10115-004-0154-9

[28] B. Mirkin (2012), *Clustering: A Data Recovery Approach*, Chapman and Hall, ISBN 978-1439838419

[29] D. T. Larose (2005), *Discovering Knowledge in Data: An Introduction to Data Mining*, John Wiley & Sons, Inc., Hoboken, New Jersey, ISBN 0-471-66657-2

[30] P. Laurinec (2018), *TSrepr: Time Series Representations.* R package version 1.0.0. `https://CRAN.R-project.org/package=TSrepr`

[31] N. Begum et al. (2015), *Accelerating Dynamic Time Warping Clustering with a Novel Admissible Pruning Strategy*, Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining

[32] A. Rodriguez, A. Laio (2014), *Clustering by fast search and find of density peaks*, Science, Vol. 344, Issue 6191, pp. 1492-1496

[33] A. Sarda-Espinosa (2018), *dtwclust: Time Series Clustering Along with Optimizations for the Dynamic Time Warping Distance*. R package version 5.3.1. `https://CRAN.R-project.org/package=dtwclust`

[34] J. Paparrizos, L. Gravano (2016), *k-Shape: Efficient and Accurate Clustering of Time Series*, SIGMOD Record, (Vol. 45, No. 1)

[35] G. James, D. Witten, T. Hastie, R. Tibshirani (2013), *An Introduction to Statistical Learning with Applications in R*, Springer Science+Business Media New York, ISBN 978-1-4614-7138-7

[36] T. Therneau and B. Atkinson (2018). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-13. `https://CRAN.R-project.org/package=rpart`

[37] Alfaro, E., Gamez, M. Garcia, N.(2013). *adabag: An R Package for Classification with Boosting and Bagging*. Journal of Statistical Software, 54(2), 1-35. `http://www.jstatsoft.org/v54/i02/`

[38] Venables, W. N. and Ripley, B. D. (2002), *Modern Applied Statistics with S. Fourth Edition*. Springer, New York. ISBN 0-387-95457-0

[39] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel and F. Leisch (2017). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071)*, TU Wien. R package version 1.6-8. `https://CRAN.R-project.org/package=e1071`

[40] R Core Team (2018). *R: A language and environment for statistical computing. R Foundation for Statistical Computing*, Vienna, Austria. `https://www.R-project.org/`

[41] A. Liaw and M. Wiener (2002). *Classification and Regression by randomForest*. R News 2(3), 18–22.

[42] J. Mantua, N. Gravel and R. M. C. Spencer (2016), *Reliability of Sleep Measures from Four Personal Health Monitoring Devices Compared to Research-Based Actigraphy and Polysomnography*, Sensors, 16, 646

[43] J. Zvárová (1998), *Biomedicínská statistika I.: Základy statistiky pro biomedicínské obory*, Karolinum, Praha, ISBN 978-80-7184-786-1

[44] T. Lumley et al. (2002), *The Importance of the Normality Assumption in Large Public Health Data Sets*, Annual Review of Public Health 2002. 23:151–69

[45] G. D. Ruxton (2006), *The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test*, Behavioral Ecology, Volume 17, Issue 4, Pages 688–690

[46] E. Bradley (1994), *An Introduction to the Bootstrap*, Chapman & Hall/CRC, ISBN 0-412-04231-2

[47] X. Golay et al. (1998), *A new correlation-based fuzzy logic clustering algorithm for fMRI*, Magnetic Resonance in Medicine, 1998 Aug;40(2):249-60.

[48] `https://www.howsleepworks.com/images/hypnogram.jpg`

[49] P. Esling and C. Agon (2012), *Time-series data mining*, ACM Computing Surveys 45 (1). ACM: 1–34.

[50] G. de Niet et al. (2009), *Music-assisted relaxation to improve sleep quality: meta-analysis.* Journal of Advanced Nursing, 2009 Jul;65(7):1356-64

[51] J. A. Horne, O. Ostberg (1976), *A self-assessment questionnaire to determine morningness-eveningness in human circadian rhythms*, International Journal of Chronobiology 1976;4(2):97-110

[52] T. Kohonen (2013), *Essentials of the self-organizing map,* Neural Networks 37 52–65