Tien Nguyen

STATISTICAL REPORT FOR WELLBEING DATA OF COMPANIES
Wellness application ViaEsca

Bachelor’s thesis

Information Technology

Bachelor of Engineering

2021

South-Eastern Finland
University of Applied Sciences

South-Eastern Finland
University of Applied Sciences

Author (authors)	Degree title	Time
Hoang Thuy Tien Nguyen	Bachelor of Engineering	May 2021

Thesis title	
Statistical report for wellbeing data of companies Wellness application ViaEsca	57 pages 0 pages of appendices

Commissioned by

Kisko Labs

Supervisor

Timo Mynttinen

Abstract

ViaEsca was a hybrid (web and mobile) application that supported wellness coaching to help find balance in eating for its users. The objective of this thesis was to conduct statistical reports for companies that purchased ViaEsca service and present them only for ViaEsca’s administrators. The goal was to provide a measure for ViaEsca to convince the companies of the benefits that the software brought to them as a wellness solution for the employees.

In preparation for the practical part, a theoretical background study was composed for (i) the meaning of data, (ii) digital data and information security, (iii) quantitative data vs. qualitative data, and (iv) theory and methods for conducting a statistical report, which included data collection, organization, summarization, presentation, analysis, and interpretation.

Only aggregate users’ data in the field related to wellbeing will be processed for statistics. A quantitative report and a qualitative report were generated for each company. Quantitative data were gathered by metrics in the application, while qualitative data were gathered by polls and surveys. Charts and tables were used to present the statistics for both reports.

Simple but effective conclusions and information could be drawn from the achieved statistics. As a result, the implemented feature could be used by ViaEsca as a means to report statistics back to the companies or to analyze and make changes to improve their service to be suitable and beneficial for a wider range of users. Possible development of this thesis would be conducting a more in-depth analysis from the statistics to find possible correlations between data key points.

Keywords

statistics, data report, data analysis, wellbeing

CONTENTS

1 INTRODUCTION

In the last three decades, we have witnessed the boom and rise of the digital era, and in today’s time, owning a mobile phone has become the norm for every person. The first mobile phone was invented long ago, but it was not until 2007, when the first iPhone debuted, that the mobile phone industry was revolutionized and the term “smartphone” first appeared among the mainstream public. Since then, there have been tough and close competitions happening in the industry. It is also predicted that there will be approximately 3.8 billion smartphone users in the world by 2021 (Statista 2019). The spread of smartphones has raised demands for many new mobile software applications, many of which have the mission to improve people’s food consumption and promote people’s wellbeing in general. (Maringer et al. 2018.) According to Franco et al. (2016), staying healthy by eating right has been getting more and more attention from the public and these applications provide easier and more accessible methods for everyone to keep track of their diet plan. It is highly convenient to have a software application that plans the meals throughout the week as well as shows users the recipes. Other common functions of these kinds of digital solutions include helping their users lose weight, track calories, and record what was eaten. This is considered an efficient and more accurate method of collecting dietary data from users that can be beneficial for nutritionists to improve the understanding of food consumption behaviors and diet trends. (Maringer et al. 2018.)

Nowadays, many companies have started to invest in solutions that can improve their employees’ health and wellbeing, which means that it is time for those wellness applications to pick up the pace and seize the opportunities. The application that I commissioned for this thesis is ViaEsca, a digital solution that supports wellness coaching to help find balance in eating for users. The practical aim of this thesis is to implement a new feature on ViaEsca to provide an approach to convince the companies of the benefits that the software brings to them as a wellness solution for their employees. The implementation part consists of a company data report feature for ViaEsca’s administrators to monitor statistics of the aggregate data of said employees. The statistics are generated from the total data of a group of users so that no information or conclusion on a personal level can be drawn out of the report. The report results will be utilized by ViaEsca and the practical assumed benefit is that it will further support this kind of investment in the employees from the companies' standpoint.

In order to achieve the goal of this thesis, the theoretical background part will cover the basic information and general definition of a digital solution, data, and different statistical analysis methods as well as other relevant data terms for the implementation part. The structure of the thesis is described as follows:

- **Chapter 1** (*Introduction*) is the introduction about the general background of wellness digital solutions and the objective of the thesis.
- **Chapter 2** (*Theoretical background*) is a study of concepts and definitions of relevant data terms and statistics conducting steps to find the best practice to summarize and present data for ViaEsca's company data report feature.
- **Chapter 3** (*ViaEsca*) introduces what ViaEsca is, its mission, vision as well as details about its technical stack and the tools needed for the next chapter.
- **Chapter 4** (*ViaEsca report feature implementation*) includes explaining how to handle private data and describing in detail the process of implementing the company data report feature for ViaEsca.
- **Chapter 6** (*Conclusion*) is the summary of the outcome of the thesis and possible further research or development for the topic.

Information is all around us and it is crucial to utilize it for the right purpose. This thesis complies with the General Data Protection Regulation (GDPR) and no personal information is displayed in the implemented data report feature or this thesis document.

2 THEORETICAL BACKGROUND

This chapter captures data's definition from general to detailed, their characteristics as well as meanings of specific data terms and information that will be relevant for the practical part of this thesis. Following that, details about statistical analysis steps and methods are also discussed in order to find the most suitable path for the practical part.

2.1 Data

Nowadays, it is essential to be informed correctly about everything that is going on around us, especially when the internet is such a big source that provides all kinds of information. The amount of information is always on the increase and so is the need for it. Data are units that compose the information that is known. Hence it is certain that an enormous amount of data is being handled everywhere every day.

We have all heard and used the term “data” before, especially when the context is about technology, science, or research. From time to time, “data” has become a mass noun, such as “information” or “fish”. However, linguistically speaking, “data” is the plural form of datum, which points to a singular value of a single variable. (Oxford Learner’s Dictionaries 2021.) Hence, in this thesis, I will mostly address data as a plural noun. Data exist oftentimes in the form of numbers, otherwise, they exist as words for plain facts or preferences, pictures, and even signals. However, these kinds of data are proven to be able to be translated into numbers. For example, pictures are built from a number of pixels and texts have word counts. In general, most data are converted to numbers at some stage during the process. (Hand 2008, 3.) Hence, data are measurements or observations that occur around us. Typical examples of data are the population, net worths, heights, weights, grades, prices, and so on. There are other data terms, such as data item, data unit, observation, and data set. In general, data are made up of data items that are defined as attributes of a data unit. Data units are, for instance, objects, creatures, people, businesses, or even phenomena. Meanwhile, the term observation in working with data exists when a data item is being recorded for a data unit. Hence, a collection of all observations produces a data set. (Australian Bureau of Statistics 2021.)

The relationship between “data”, “information” and “knowledge” is described below in Figure 1.

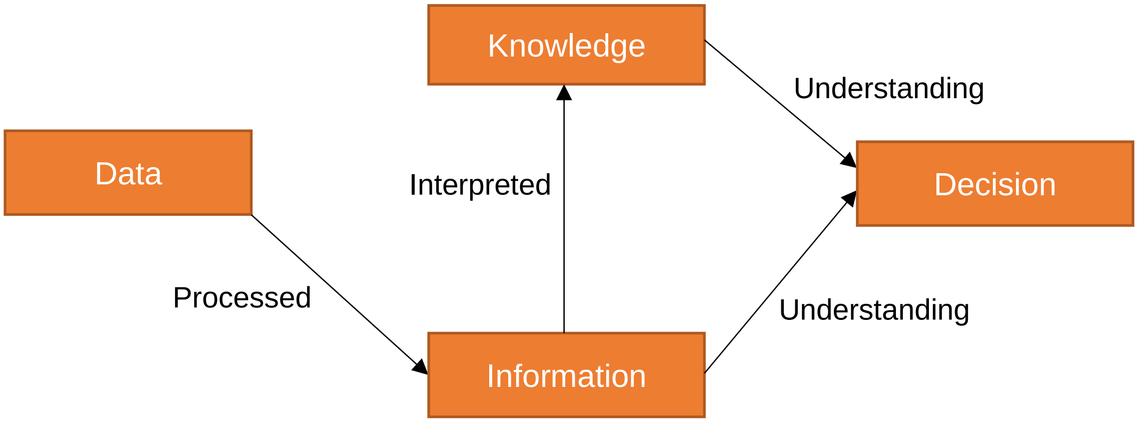


Figure 1. Relationship between data, information, and knowledge

While often addressed as information, data and information are different in meaning. Data are considered to be the source of information. Before processed, data are labeled as raw data. After raw data are processed and put into the appropriate context, they turn into useful information (Thakur 2021). Information being evaluated and organized develops into knowledge. For instance, “2021” is the data that does not imply anything when standing alone, but if it is elaborated as “2021 is my graduation year”, it becomes a piece of information, and when it is interpreted as “it is when I am no longer a student”, it becomes knowledge. The end decision is made only when an understanding is gained after extracting the information and knowledge from the preceding data.

2.1.1 Digital data and information security

Although the meaning of data is broad, nowadays, the term “data” informally implies digital information. This section’s purpose is to facilitate the implementation of the data report feature by defining principles for handling data on a digital solution. In the field of technology or computing, data, also known as digital data, are often manipulated and processed through programs to serve a purpose. In computing, data are often converted into binary form for easier processing and transmission. Computers are programmed to present and deliver data as strings of ones and zeros. That applies even to new, unstructured data including videos, images, and sounds. (Techopedia 2021.)

To protect digital data from being lost or viciously tampered with by cyber-crime, it is crucial to implement information security following the CIA triad – Confidentiality, Integrity, and Availability. The three concepts have long been treated as a “three-legged stool” instead of individual aspects when it comes to enhancing information security. Confidentiality in this context defines measures implemented to give only authorized users and programs access to the data. The access can then lead to reading or writing the data. On the other hand, integrity in this context defines measures taken in protecting the data as they should be, meaning that the data should not be accessed by users that are not authorized and tampered with by anybody, either unintentionally or intentionally. Lastly, availability in this context defines that authorized users should be able to access and modify the data when necessary, meaning that the database should always be up and ready as well as backup versions of the data are generated regularly in case of database failure. Although the three concepts of the information security triad differ in meaning, they are closely tied to one another. Thus, there are always three fundamental aspects that need to be taken into consideration when working with digital data. (Fruhlinger 2020.)

There are three states of digital data and they should be enforced with security measures following the CIA triad mentioned above. The first one is “data at rest”, which indicates that the data are stored within the database, hard drive, or network and not being accessed. For this state, it is vital to enable strong encryption to enforce confidentiality and preserve the data’s integrity. Additionally, a backup version of the data should be created to provide availability. The second state is “data in use”, which implies data that are not in the first stage, commonly when there are accesses made to the data. At this stage, it is crucial to make sure that only authorized users or processes have access to the database by strengthening authentication and authorization. The third stage is “data in transit”, which describes the data that are traveling from one end to another, commonly through the network such as emails. At this stage, it is important to establish a secure connection channel for the data to travel, which helps keep the data safe from unauthorized malicious attacks as well as preserving the data's integrity. (University of Edinburgh 2016.)

2.1.2 Quantitative vs. qualitative data

There are multiple types of data. However, data are often divided into two major branches, which are qualitative data and quantitative data. (QuestionPro 2021.) This section deepens the knowledge of recognizing data types and their attributes. It is also pre-defined that the feature implemented for the practical part of the thesis will be mainly focusing on these two types of data. Generally speaking, quantitative data are measurable, while qualitative data are not. It is straightforward to explain the difference between quantitative and qualitative data, as Figure 2 shows.

Quantitative data	Qualitative data
Associate with numbers	Associate with details
Numerical data	Non-numerical data
Can be statistically analyzed	Can be observed but not evaluated

Figure 2. Differences between quantitative data and qualitative data (QuestionPro 2021)

By definition, quantitative data are associated with numbers or counts of numbers. Quantitative data are quantifiable, which makes them suitable to execute mathematical calculations and statistical analysis. Commonly, this type of data are answers to questions such as "How much?", "How many?", etc. This type of data can be verified and evaluated with mathematical methods. Quantitative data are often numeric values that are followed by measuring parameters, such as kilograms, meters, dollars, and so on. This type of data is popular for statistical analysis. (QuestionPro 2021.)

The most common types of quantitative data examples according to QuestionPro (2021) are:

- **Counter**: count numbers of entities, for instance, the population.
- **Measurement**: a measure of any physical objects, for instance, a person's height or weight, etc.
- **Sensory calculation**: convert non-numerical data to numerical data, for instance, converting electromagnetic signals into numbers.
- **Projection of data**: predict a future outlook on data using mathematical methods, for instance, a predicted increase in sales of a company
- **Quantifications of qualitative entities**: use numbers to identify qualitative information, for instance, rating customers' satisfaction on a scale of 1 – 10.

As all concepts do, there are advantages and disadvantages to this type of data. Following QuestionPro (2021), the advantages are:

- Thanks to it being numbers, it offers more support for conducting in-depth research.
- The numerical nature of these data reduces bias manipulation over the data.
- Quantitative data are subjected to produce more accurate results.

And the disadvantages according to QuestionPro (2021) are as follows:

- Since quantitative data are not descriptive, it is more challenging to make good decisions on how to get the best results out of these data.
- The researcher's knowledge about the questions and objective of these quantitative data are the most important to keep the data disinterested and unbiased.

By definition, qualitative data are often non-numerical and are used to characterize and describe the data unit. The most common method to gather qualitative data is to observe and record since their nature does not support any kind of measurement. Some of the typical examples of qualitative data are genders, nationalities, colors, Boolean values (true or false), and so on. In statistics, qualitative data are known as categorical data since they describe attributes and properties of the data unit. For example, in a data set of cars, we can organize, or categorize, the cars based on their brands, which are qualitative data. (QuestionPro 2021.)

The advantages of qualitative data according to QuestionPro (2021) are:

- In many cases, quantitative data alone are not enough. Qualitative data provide more insights into the data unit. However, researchers need to ask the correct and narrow-enough questions when gathering qualitative data, so that the data set can be categorized for more in-depth analysis.
- Qualitative data enable providing the data set with rich data.

The disadvantages of these data according to QuestionPro (2021) are:

- They are time-consuming to collect, which may result in fewer data subjects to study.
- It is challenging to define a scope for collecting questions of qualitative data.
- They are difficult to generalize, which means it is also harder to conclude after analyzing qualitative data.

2.2 Statistical analysis

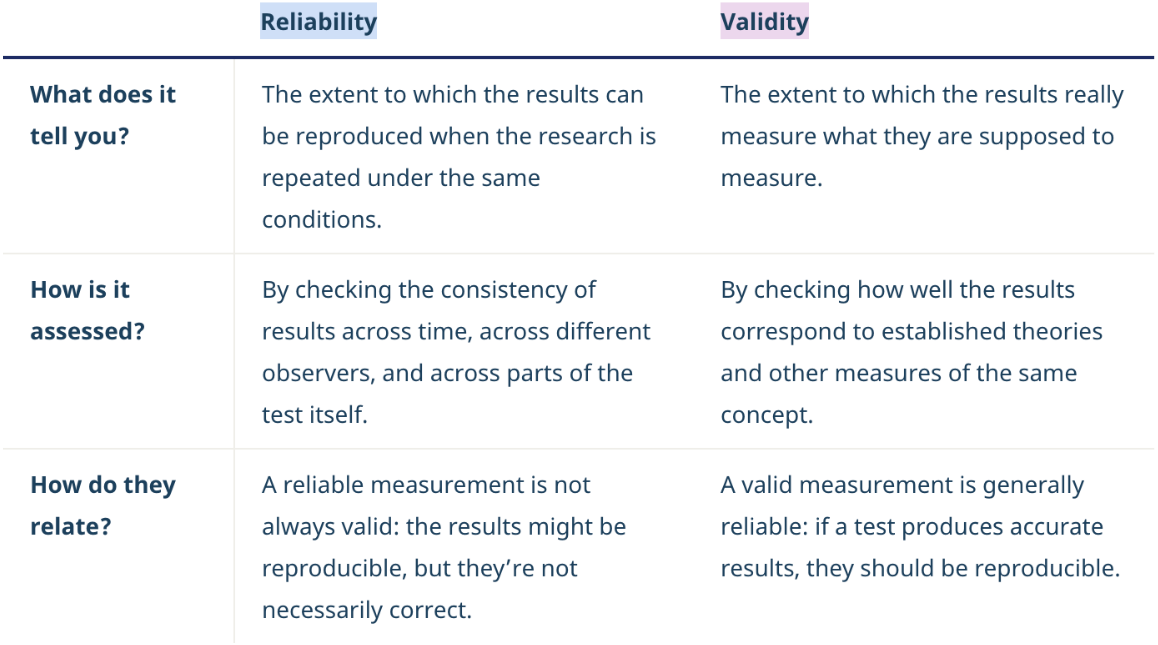
The terms “statistics” and “data” often cause a few misconceptions. Many people think that their meanings are the same. However, statistics are data that have gone through an analysis process. According to Hand (2008, 1-7), statistics are broadly utilized in almost every prospect of life. For example, most recently, statistics have been applied in tracking COVID-19 cases in every country, which helps keep track of the whereabouts of the virus and how it is spreading. Statistics simply mean *extracting meaning from data* by *producing convenient summaries of data*. Statistics assist analysts in finding the trend of data, which leads to possibly predicting the future, minimizing risk, and defining the probabilities. The best decision regarding any problem calls for the need for statistics. Statistical analysis can require various methods and tools depending on the intention of the analysis. Statistics contribute hugely to every scientific field. The word “statistics” is the plural form for “statistic”, which means a *numerical fact or summary*. Although statistics produce numerical values directly often enough, statistics can also consist of qualitative attributes. However, performing statistical analysis on such data is proven to be more sophisticated and difficult. More often, the qualitative attributes would be converted to or bear numerical values to simplify the process. The conclusion drawn by Hand (2008, 19) states that statistics are the core of every field that requires dealing with information, and it is continuing to be all the more important in our lives. Furthermore, modern statistics, which are produced by technologies, have been providing us more insights and values, especially in terms of exploring the unknown.

According to Panik (2012, 1-2), statistics contain the theory and practice of *collecting*, *organizing*, *presenting*, *analyzing,* and *interpreting* data. There are two types of statistics that can be conducted: descriptive and inductive. The term inductive statistics means that the statistics of a part of a data set can represent the results for the entire data set. The process of producing such statistics is called *sampling* and the group of subjects that are chosen to represent the whole data set must have the *margin of error* taken into account. This type of statistics is used for generalization, decision-making with uncertainty, and forecasting. For the purpose of this thesis, which is producing a means for monitoring the data produced by the application’s users, the statistics that are going to be conducted are descriptive statistics. Descriptive statistics consist of summarizing data and presenting them. With descriptive statistics, it is possible to view the overall picture of the data and they answer the question “What do the data want to transpire?” The production of descriptive statistics involves constructing tables, charts, and graphs as well as calculating the arithmetic means, medians, percentages of changes, and so on.

2.2.1 Collecting data

Collecting data is an important step because applicable statistics can only be produced with a large enough amount of high-quality, accurate data. The most common method for gathering data for statistical analysis is data aggregation. Data aggregation means gathering data from various sources and connecting them in a summarized format. (Import.io 2019.) Aggregate data, the product of data aggregation, can exist in both numerical form and non-numerical form. Since they are a combination of many data elements, they shine a light on more information and insights, such as trends, and offer comparison values, which cannot be obtained when viewing the data elements separately. (The Glossary of Education Reform 2015.) When collecting data, it is important to take into account that the collected data should cover all aspects needed and are related to or necessary for the conclusion at the end. By doing that, data collection is forced to eliminate irrelevant data points. (Hand 2008, 23.)

According to Bhandari (2020), the overall process of data collection requires three steps: understanding the purpose of the report, defining the type of data that are going to be collected, and choosing the methods to collect the data. Collecting quantitative data helps produce statistical insights while collecting qualitative data helps understand experiences. There are heaps of methods to collect quantitative data because numbers, the nature of this type of data, are what shape our logic in existence. Therefore, numerical data is all around us and the real challenge is to collect the correct material for our research. The data should be clean, unbiased, and accounted for a clear target. For quantitative data, data collection plays a major role in the research process, and analyzing the data afterward is simply pulling out meaningful information from them. The results should help us make better decisions regarding the challenge that the data represent. (QuestionPro 2021.) According to The Albert Team (2020), normally, quantitative data can be collected through experiments and observational studies, while qualitative data can be collected through surveys, interviews, and specific case studies. For conducting a survey questionnaire, it is important for the questions and answer choices to be explicit and the timing for such survey should be relevant. To assess the quality of collected data, it is most common to assess two concepts: reliability and validity. Reliability ensures consistency in data, while validity ensures accuracy for data. (Middleton 2019.)



	Reliability	**Validity**
What does it tell you?	The extent to which the results can be reproduced when the research is repeated under the same conditions.	The extent to which the results really measure what they are supposed to measure.
How is it assessed?	By checking the consistency of results across time, across different observers, and across parts of the test itself.	By checking how well the results correspond to established theories and other measures of the same concept.
How do they relate?	A reliable measurement is not always valid: the results might be reproducible, but they're not necessarily correct.	A valid measurement is generally reliable: if a test produces accurate results, they should be reproducible.

Figure 3. Understanding reliability and validity of data (Middleton 2019)

To prove a data collection method has high reliability, the data that are collected should stay mostly the same when running the same method again under the same circumstances. To prove a data collection method has high validity, the data collecting tools should function correctly and the data that are collected should bear realistic properties and characteristics. These two concepts should always be ensured simultaneously, as one is not enough to validate the other. Generally, validity is more difficult to properly determine than reliability and it is also more important in terms of data collection. Ensuring the data quality also means ensuring the quality of the conclusions drawn from processing those data. (Middleton 2019).

2.2.2 Organizing data

After data collection, it is necessary to categorize and classify data, which is also known as data organization. Logically, organized data enable easier and faster reading and processing. According to Gardner (2020), the first crucial step to data organization is cleaning the data. As the name suggests, this process requires looking for and removing unnecessary information, known as the clutters in the raw and unformatted data, even though they have gone through the same filtering procedure during the collecting stage. Then, data can be arranged by category. For instance, if the data gathered are associated with users (people), they can be categorized by gender and age; or in other cases, data can be arranged alphabetically and sometimes also by date and time. One of the most common ways to organize data is to store them in tables and with the help of technologies in this day and age, those tables are combined to become one entity that is known as the database. Following Anusha (2020), a database is *an organized collection of data* that allows easy access and management. In the database, data are placed into tables, rows, and columns with an index for easier finding of data. Each data set can be put to one table. For relational databases, the process of organizing data into tables is called data normalization. There are three forms of data normalization. In the first normal form, every data unit is placed as rows in a table, each row is independent of all the others. In the second normal form, all rows are dependent directly or indirectly on primary keys. In the third normal form, all rows must depend directly on the primary keys and repeated attributes are removed. (IBM 2020.)

Organizing data often goes with summarizing data, which is the decisive step before presenting the statistics. It is vital that the data is summarized to produce statistical values for display. According to Hand (2008, 27), data values in a collection form a “distribution” and summary statistics can show what a “typical” value of that “distribution” looks like. For a numerical data set, the most well-known and basic description of the said “typical” value is the average value, which is also popularly known as the arithmetic mean. The mean value is obtained by summing up all units in the set and dividing the sum by how many units there are. An average is often the representative for a data set unless the outliers of the set are too large for an average value to be relevant. For example, when replacing one of the data units in the set 1, 4, 7, 10, 12 with 1000, the arithmetic mean can no longer represent the data set. (Panik 2012, 24.)

According to Panik (2012, 24), the properties of the arithmetic mean are as follows:

- It always exists.
- It is unique.
- It can be manipulated by outliers.
- It is only relatively reliable.
- Each element that participated in the calculation should bear the same weight or the same relative importance.

To put it in other words, all units in the data set can relate to the arithmetic mean because they all share an equal part in producing it. An arithmetic mean is a *statistic*. Furthermore, it also offers comparison for the data units as it is considered to be a “central” value. (Hand 2008, 28.) For instance, the average age of females in Finland in 2020 is 44.7 (years), thus, if one’s age is above that value, one belongs to the older group and vice versa. (Statistics Finland, 2021.)

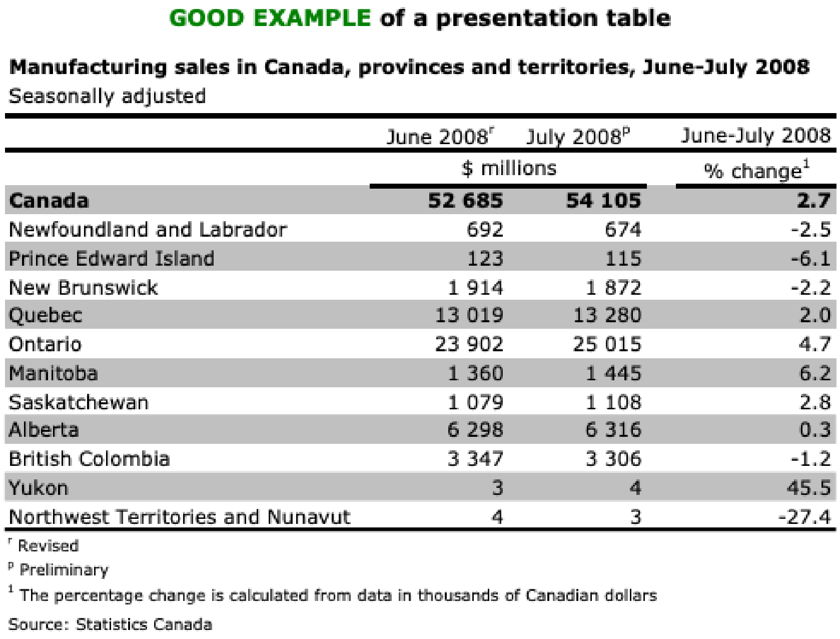
Following Hand (2008, 28-29), calculating the average value is one way of summarizing data and another way is calculating the *median* value. The difference between the two is that the arithmetic mean is calculated by adding and dividing all units, while the median is one data unit in the data set that stands in the middle and separates the others to either be bigger or smaller than that median number. To acquire the median value, the data set in question should be organized in order from smallest to biggest. Most of the time, in a diverse and large enough data set, the median value can be close to the mean value. However, there are also complex cases for finding the median value, for example, if there is a number of equal values in the data set, or the data set consists of too few data units, etc. It is expected that the median value is not exactly equal to the mean value so that the comparison between the two can provide more insights into the data set. In addition to the mean and the median, another important summary value is the *mode*, which indicates the value that occurs most frequently if there are many equal values for different data units in the data set. For example, for a data set of rooms in a hotel, there are rooms for 1 person, 2 people, and 4 people, but there are more rooms for 2 people than any of the others. In that case, the mode of the hotel room types is for 2 people.

2.2.3 Presenting data

After organizing and summarizing data, it is important to choose the correct presentation method for the results. Otherwise, the statistics would seem meaningless and ill-conducted. According to UNECE (2009), there are multiple methods of presenting data, including written reports, tables, charts, and graphs.

When presenting data in written reports, it is important to know the target audience and be fully aware of the context. It is best if the written report tells a story about the statistics. However, the story must ensure its impartiality (keep the data report unbiased) and confidentiality (ensure confidentiality of participants in producing the data for the said statistics). The language used in the written report should be clear, short, and simple. (UNECE 2009, 1-4)

Following (UNECE, 2009, 12-16), a more popular approach to displaying statistics is using tables. A good table gives the audience easy and straightforward access to the summary of data points and their statistics/summaries. The table should have metadata, for instance, a title, source, column headers, and row stubs, etc. The table should be able to provide enough context to maintain its meaning even in the case of it being copied or extracted to another place. A good table should avoid having unnecessary text, whilst the data points (rows) should be in order (either by indexing or by time), no cell should be left empty and numerical values should be placed to the right of the cell. (Figure 4.)



GOOD EXAMPLE of a presentation table

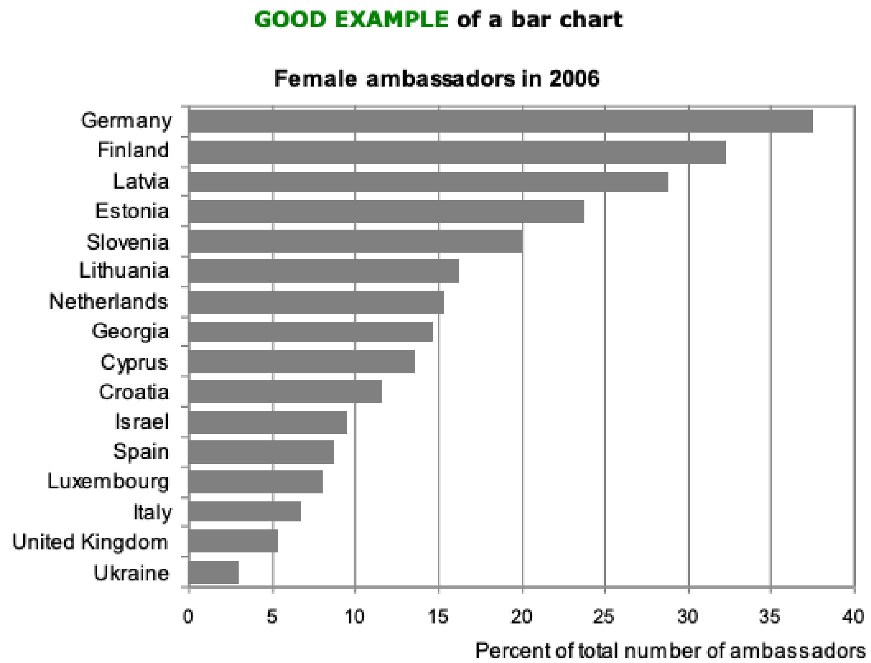
Manufacturing sales in Canada, provinces and territories, June-July 2008
Seasonally adjusted

	June 2008[r]	July 2008[p]	June-July 2008
	$ millions		% change[1]
Canada	**52 685**	**54 105**	**2.7**
Newfoundland and Labrador	692	674	-2.5
Prince Edward Island	123	115	-6.1
New Brunswick	1 914	1 872	-2.2
Quebec	13 019	13 280	2.0
Ontario	23 902	25 015	4.7
Manitoba	1 360	1 445	6.2
Saskatchewan	1 079	1 108	2.8
Alberta	6 298	6 316	0.3
British Colombia	3 347	3 306	-1.2
Yukon	3	4	45.5
Northwest Territories and Nunavut	4	3	-27.4

[r] Revised
[p] Preliminary
[1] The percentage change is calculated from data in thousands of Canadian dollars
Source: Statistics Canada

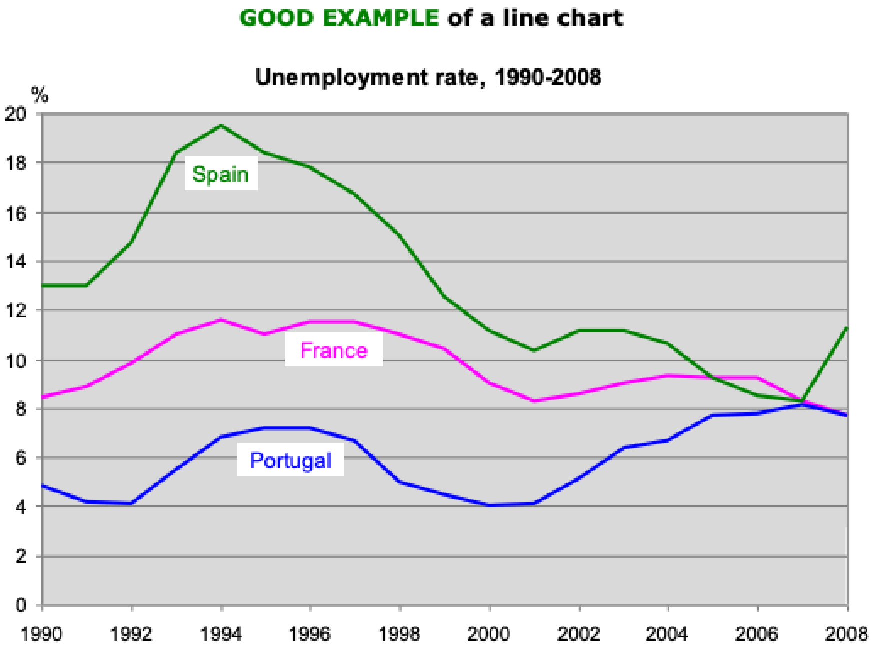
Figure 4. A good example of a presentation table (UNECE 2009)

Most of the time, statistics are presented in charts and graphs. Good use of charts and graphs may also reveal the trends of the statistics. This type of data visualization is best in showing comparison, changes, frequency distribution, and data correlation. Charts and graphs presenting statistics should be easy to understand, so it is important to know what type of chart/graph is suitable for displaying the said statistics. Bar charts (or column charts) are considered to be the easiest to utilize. They are most popular for presenting data frequency distribution. (Figure 5.) Meanwhile, line charts are used to visualize data trends, commonly over a period of time. Multiple lines chart can also produce comparison values. (Figure 6.) Another popular chart for statistics is the pie chart. Pie charts are best used for showing percentage distribution between the data points as well as showing the changes in statistics when there are two or more pie charts for different data sets. However, a pie chart should not consist of more than 6 data points to avoid a bad display of data. (Figure 7.) There is also the scatter plot chart that is used to display the data correlations between the data points. However, this type of chart is harder to interpret than the previous-mentioned charts. (UNECE 2009, 17-24)



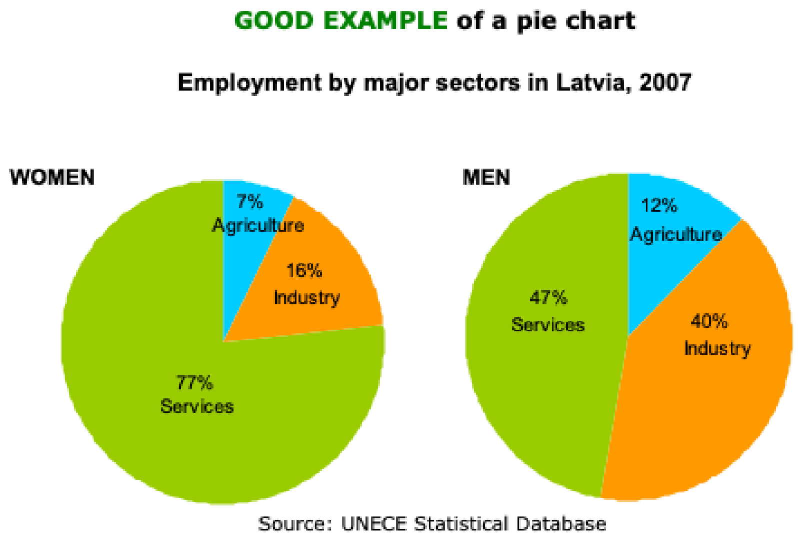
Source: UNECE Statistical Database

Figure 5. A good example of a bar chart (UNECE 2009)



Source: UNECE Statistical Database

Figure 6. A good example of a line chart (UNECE 2009)



GOOD EXAMPLE of a pie chart

Employment by major sectors in Latvia, 2007

Source: UNECE Statistical Database

Figure 7. A good example of a pie chart (UNECE 2009)

Furthermore, maps are also a powerful tool to display statistics that are related to geographical locations as they can summarize large data and grab the audience’s attention easily. There are multiple types of maps. They should only be used if they are the optimal option. Otherwise, it is recommended to use basic charts and graphs. In this digital era, there are new data visualization methods that have risen, such as animation and video. They are better at telling stories and they often allow interaction from the audience as well. (UNECE 2009, 30, 43)

2.2.4 Analyzing and Interpreting data

This section combines the two concepts, analyzing and interpreting statistics, and I will only extract relevant findings and knowledge that are appropriate for this thesis. When analyzing statistics, it is necessary to know how to handle uncertainty and probability. There is a famous quote by an anonymous person regarding this, which is “Being a statistician means never having to say you are certain”. The answer to this uncertainty issue is recognizing the trend from the statistics of aggregate data. For example, it is impossible to predict the baby’s gender when the baby has just been conceived. However, it can be said from the results of millions of births that over half of them are male. This rule is called the *law of large numbers* and it allows statisticians to conquer the uncertainty. (Hand 2008, 68.) Analyzing statistics also provides comparison values. Furthermore, statistics also often suggest relationships between different variables. Their correlation shown by statistics can be interpreted to produce a meaningful conclusion, for instance, that global warming is a consequence of the glasshouse effect. (Hand 2008, 97.)

According to Brooks (2020), performing statistical analysis requires being unbiased as it can affect the decision made in the end. Since the descriptive statistics type is basically a summary of data presented in tables and charts, it is fairly easy to visualize and understand what the statistics convey. However, this type of statistics only allows simple interpretation of the results and the conclusion that can be drawn are not in-depth.

3 VIAESCA

The first part of this chapter covers the background details for the implementation part, such as what ViaEsca is, their vision, the purpose for the implementation of the thesis as well as what kind of data are gathered on the application. The part on ViaEsca’s development environment briefly describes ViaEsca’s framework, Ruby on Rails, the concept of Active Record, and PostgreSQL. Lastly, the chapter ends with an introduction and description of the tools that assist in handling data as well as the data processing plan for the next chapter.

3.1 ViaEsca’s vision

ViaEsca in the Latin language means “the path to food”. Hence, ViaEsca is a wellness coaching mobile application that promotes changing one’s eating habits as the most efficient way to improve one’s wellbeing. ViaEsca provides coaching for anyone, with or without a weight goal, through customized meal plan programs and recipes. ViaEsca’s mission is to help its users be in better control of their food intake and achieve a more energetic and balanced feeling overall. ViaEsca was founded in 2016 and it has assisted more than 300 Finnish people on their wellness journey. (ViaEsca 2021.)

To address the need for bringing a healthy and tasty food diet to more people, ViaEsca has envisioned providing a group coaching service for companies and communities. The main reason behind this idea is to support the well-being of companies' employees, and by doing that, ViaEsca expects to be able to boost the employees' overall performance. VieEsca believes that the meaning of "wellbeing" is to find the balance in eating, exercising, and resting. Hence, a good diet can help people improve their sleep quality, feel more energetic throughout the day and recover from stress better. The primary aim of this method is to reduce sick leaves and additional costs for the companies due to employees' illness. (ViaEsca 2021.) Therefore, the goal of this thesis is to provide ViaEsca an approach to prove their effectiveness in providing group coaching to companies that purchase ViaEsca as a wellness solution for their employees by implementing a company data report feature where data are presented as statistics.

3.2 ViaEsca's features and data gathering

ViaEsca application has two separate views: one is the web view for the admin panel where ViaEsca employees, as known as administrators, can monitor the statistics of their application and manage users; and one is the mobile application view for the users. The feature that I am going to implement for this thesis will be placed on the admin panel. However, I would briefly describe ViaEsca's features on the users' mobile application for a better understanding of what kind of wellbeing data there are to gather from the users. ViaEsca mobile application has the meal plans feature, where users can view their personal recommended list of meals for the day, their recipes as well as the shopping lists with ingredients to cook the meals. The feature allows users to mark meals as eaten or add more meals throughout the day, which will log their calories consumed and the number of meals eaten. ViaEsca can also gather data from activity bracelets from which they can track users' exercise (active) calories and their sleep time as well as their burned calories. The service for connecting such devices is provided by Validic (https://validic.com/). In conclusion, according to Section 2.1.2, the five main categories are classified as quantitative data which include calories consumed, calories burnt, active calories, the number of meals eaten, and sleep time for the day.

Moreover, there are also qualitative data gathered from a questionnaire. When a user first starts using the application, they have to answer a simple poll survey that asks about their current eating habit, the nutrition from their eating rhythm, and the rate of their exercise, sleep, energy, etc. The number of questions at the time of this project is 16 questions. These are considered to be qualitative data, however, the answers have a value system embedded in them. For instance, an answer that is closest to being most healthy according to ViaEsca bears the highest value number. In this way, qualitative data can be gathered and processed as quantitative data. For comparison, a second wellness poll survey is then filled out when users finish their starter journey for 21 days. Therefore, there are two sets of qualitative data, before and after using ViaEsca's starter journey.

3.3 ViaEsca's development environment

The ViaEsca application uses Ruby on Rails for the framework and PostgreSQL for the database. Rails has been around for almost 17 years, and it is still able to stay relevant in its competition. It is now a matured framework with a stable and supportive community. Rails was released as an open-sourced project and it has stayed open-sourced until now, which contributes to Rails' undeniable success as the framework never ceases to evolve with its incredible adaptability and scalability. By technical definition, Rails is a server-side web application framework written in the Ruby language. Rails follows the MVC (Model -View – Controller) design pattern. To shortly explain the MVC design pattern, the Model is the representative for the database, which means it is the main communicator between our web app and our database. One model stands for one database table, and it is independent of the user interface. Meanwhile, the Controller is the communicator between our Model and our View, and the information input that the Controller receives can go both ways, to the Model or the View. Lastly, the View is a display of information/data, which is basically what the users see and interact with. Rails also absorbs other software engineering patterns, such as

DRY (Don’t Repeat Yourself), which keeps our codes clean and compact, and Active Record. (Rails Guide 2021.)

Active Record is an important concept to understand in order to write and execute the best possible queries for statistical calculations with Rails and PostgreSQL. According to Fowler et al. (2003, 160), in Active Record, an object is a row in a database’s table and it carries both the data value and the behavior which operates on that data. Following Rails guide (2021), Active Record is the “M” in the MVC framework. In other words, it is responsible for the logic and data of the application. The best thing about Rails with Active Record is how Rails makes Active Record work with as few manual configurations as possible. Active Record and Rails have a naming convention to identify the models and map them to their correct database tables. The Rails model class name should be in singular form with the first letter capitalized. If there are two words or more, they should follow the same rule and have no space or any type of indicator between them. On the contrary, for the database table instances of the models, their name should have an underscore between each word, no capitalized letter, and the last word must be in plural form. For instance, for a database’s table name “users”, its model class should be “User”; or in the scenario of a class called “AccessCode”, its corresponding database table is “access_codes”.

According to PostgreSQL documentation (2021) and Maayan (2019), PostgreSQL, or Postgres as a more popular alias, is an open-source object-relational database management system that supports the SQL (Structured Query Language) standard as well as many modern features: foreign keys, complex queries, etc. Thanks to it fitting well with Active Record, Postgres has become the first choice when it comes to working with Rails. Postgres is a highly extensible database. Specifically, querying with Postgres can take less time than other SQL databases due to Postgres’ support for customized ways to store data and create relations between data objects. According to Nguyen (2016), Postgres schemas facilitate pulling data from different sources and storing them in one place. The schema allows name-spacing the data, which enables easy tracking of tables and their associations while looking elegant and simple. Furthermore, Postgres offers enormous support for data reporting. Following Blitz (2018), Postgres provides one of the best services regarding the three characteristics of data – the three Vs: Variety, Volume, and Velocity. Firstly, Postgres supports a big variety of data, which includes even advanced types, such as array and jsonb (hash), that are convenient and beneficial for data collecting and reporting. Hashes (jsonb) allows storing large data sets or results, while arrays are most favorable for doing mathematical calculations. Secondly, thanks to a loving community that improves Postgres non-stop, Postgres has great scalability to adapt to sufficiently big data. However, as Postgres was initially built as a relational database, there are limitations to how much data Postgres can handle. Thirdly, Postgres can be customized and optimized to accommodate the growth speed of specific businesses. Postgres can also work coherently with many third-party systems to gain higher speed for data reporting.

3.4 Data analysis plan and tools

Before starting the implementation part, it is necessary to go through the plan for the practical implementation process to understand the meaning of the work. The choices for tools and how they assist me are explained in the latter part of this section. The company data report feature that is going to be implemented can only be viewed on the admin panel, which is a web view that is exclusive to ViaEsca administrators. For an overall picture, on the company page, there should be two tabs to switch between viewing the quantitative data and the qualitative data. The quantitative data will mainly be displayed in tables and column/bar charts and graphs, while the qualitative data will be displayed as two pie charts, before and after, for each poll question.

3.4.1 Data analysis plan

Besides all users in the company, there are six groups of users to filter the data: users that have a weight-loss goal and users that do not, users that are using the light version of the application and users that do not, and lastly users that have activity bracelets connected and users that do not. Furthermore, in order to conclude which kind of users benefit most from the application, ViaEsca has requested adding two more filters that can be applied on top of the six groups mentioned above, which are users' gender and age. The gender, male or female, and the age, which can be calculated from their date of birth, are both asked upon signing up. The age groups are 17 to 25, 24 to 35, 34 to 45, 44 to 55, 54 to 65, 64, and more. In the end, the display of the statistics allows possible data comparison between different filtered groups of users. Thus, it further supports which kind of users benefit most from using the ViaEsca service.

The main steps for the implementation part can be interpreted as follows. First, there should be a feature that provides a method for the administrators to group users to the same company. ViaEsca already has a system of access code batch to manage subscriptions to their service. Thus, it is wise to use this access code batch system to tie users to companies. Second, as mentioned in chapter 3.2, there are five main data sets that need to be handled, which are consumed calories, burnt calories, active calories, the number of meals, and sleep minutes, and all of the above are gathered per day. The type of statistics that I will apply is descriptive statistics because the goal is to provide ViaEsca's administrators a means to monitor users' statistics. The statistical results will include showing and comparing the arithmetic mean and the median values, both by using tables and charts. The goal of this thesis is to prove ViaEsca's effectiveness for users. However, an ideal situation where every employee uses the application regularly cannot be easily achieved. Thus, to avoid bad data manipulation by zeros in the data set, those will not be taken into account when doing calculations. Median values of each data set will be used as figures that provide comparative value for the averages. ViaEsca administrators will also be able to choose the date range for the gathered data that they want to observe. The averages and medians of a data set will be shown on a combo chart, on which the medians are displayed as bars and the averages are displayed as a line. In the beginning, the employees are offered to use ViaEsca for their starter journey, which lasts for 21 days. Hence, it is important for the company report to also focus on displaying the process during the said period of time. Each user would have a different start date, so a separate data report is required in order to shine more light on the users' starter journey. Similar to the report based on the chosen date range, this starter report would also show average and median values for different groups of users. However, I would have to convert all users' first 21 days to be homologous. The days will be labeled respectively "Day 1" to "Day 21" for all users.

For the wellness poll report, which is also the qualitative data report, users' answer data are divided into two states: before and after using ViaEsca's starter journey. The most common type of chart used to display the result of polls and surveys is the pie chart. Hence, for each poll question on the report, there will be a "before" pie chart and an "after" pie chart placed next to each other for a better comparison of the data values. As mentioned in Section 3.2, each question's answer option has a numeric value bound to it. Thus, I can calculate the increased percentage of the users' answers to each question so as to conclude how the starter journey has helped them on their path to improve their wellbeing.

3.4.2 Tools

First, since the feature that I am implementing also needs data that are gathered from activity bracelets and likewise, it is necessary for me to have the real data from the production database as a base example to work on. I will use Faker gem, which is popularly used to generate fake data, to seed the database. Thus, the workaround is that I will use a rake task (Rails action task) along with the Faker gem to guarantee data anonymization on the development environment implementing the feature. More about this will be discussed in Section 4.1.

For grouping data and applying mathematical calculations, I use Rails' ways of querying data with Active Record. Apart from the main utilities that Rails, Active Record, and Postgres offer as mentioned in Section 3.3, there are a few other tools, which are mostly ruby gems, that I use to assist me in presenting data for the company report feature on ViaEsca. Starting with chartkick, a gem that allows easy creation of JavaScript charts on Rails. Chartkick offers support for many basic types of charts and graphs. Not only does Chartkick have a simple installation step but Chartkick also allows me to choose the charting library that I find most suitable. (Chartkick 2021.) For this project, I choose Google Charts library for its support of combo charts. Furthermore, the library’s pie chart can show both the data value and its percentage on its slice of the whole pie.

I also use the Bullet gem to help identify “N+1 queries” in Rails, which happen when performing queries to fetch data from the model’s associations without eager loading them first. The reason it has the name “N+1 queries” is because Rails needs to perform one query for loading the model’s build, and then N queries to load its association data in each iteration. (Sarcevic 2017.) The Bullet gem will notify me when to use or when to not use eager loading, which will increase the overall performance by running fewer queries. Similar to the faker gem, the bullet gem will only be used in the development environment. (Bullet gem 2021.)

Lastly, when running a Rails server for the application, the time for the views to load, which is also the time for the queries to complete, can be seen at the end of each HTTP request on the logs of the server. However, when developing the practical task, I overall primarily use Rails’ console to test run the queries beforehand. Without the rails server running, it is almost impossible to know the exact time for the queries to complete. Hence, I use the Execution time gem to be able to monitor time and other metrics directly on the Rails console whenever running any queries. (Execution_time gem 2021.)

4 PRACTICAL IMPLEMENTATION

This chapter covers the practical work of implementing the new company data report feature for ViaEsca application on its admin web view. Before diving into the practical work, there is a private data protection issue that needs to be addressed. The quantitative data report is called the company data report, whilst the qualitative data report is called the company wellness poll report. After detailing and explaining the implementation process, there is a discussion about the results and the drawn conclusion for the results. To better demonstrating the implementation process, I will use the following syntax for this chapter:

- The model’s name is in **bold**.

- The file’s name and path are declared inside quotation marks.
- Terminal commands and other Rails keywords are in *italics*.
- Code snippets are color-coded.

Furthermore, the quantitative data report will be called the company data report, whilst the qualitative data report will be called the company wellness poll report.

4.1 Addressing the private data protection

It is not easy to work with data, needless to say, private data such as health data. Therefore, it is mandatory to pre-define a good process for how to work with retrieved content from ViaEsca’s production database. The commissioning company and I had agreed on a data anonymization process and other good practices to ensure we do not violate the GDPR (General Data Protection Regulation). GDPR is the privacy and security law enforced by the European Union (EU) to adapt to the digital age and is applicable for all organizations worldwide as long as their target data is related to EU citizens. Corporate Finance Institute (2021) defines that data anonymization is a process of removing or encoding personally identifiable information from the dataset in order to protect the confidentiality and privacy of users. The identities of the users that the data belong to will remain anonymous after the process.

First, when I pull data from the production database, there will be a rake task (a Rails task runner) using the Faker gem to execute data anonymization on my development environment. The task replaces users’ personal information (GDPR.eu, 2020, chapter 1, article 4), such as their names, emails, birthdates, biometrics measurements, and ids for third-party and so on with phony values. This serves the purpose of keeping the users’ identities completely anonymous on my local machine. The rake task is as follows:

```
namespace :anonymize_user_data do
  desc "Anonymize users in development never run on production"
  task anonymize: :environment do
    return if Rails.env.production?

    if Rails.env.development?
```

```
      require 'faker'
      puts "Anonymizing... "

      users = User.all

      users.each do |user|
        user.first_name = Faker::Name.first_name
        user.last_name = Faker::Name.last_name
        user.email = Faker::Internet.email
        user.birthdate = Faker::Date.birthday(18, 70)

        user.shopify_id = rand.to_s[2..4]
        user.polar_user_id = rand.to_s[2..4]
        user.paywhirl_id = rand.to_s[2..4]
        user.validic_id = rand.to_s[2..4]

        user.height_in_cm = rand.to_s[2..4]
        user.weight_in_kg = rand.to_s[2..4]

        user.save(validate: false)
      end
      puts "Done!"
    end
  end
end
```

Figure 8. Anonymizing user data rake task

The command to run the rake task in terminal is *rake anonymize_user_data: anonymize*. While the task is running it shows “Anonymizing” and when it is done, it shows “Done!” on the terminal.

Second, I make sure to comply with the good practice that I never keep the database on my local machine for a longer time than needed. For example, when I need more recent data and pull a new copy of the production database, I would immediately delete the older version on my local machine, and when I finish the implementation part of this thesis, I will delete the copy of the production database instantly. Furthermore, the implementation goal of this thesis is to only show results and statistics of calculated aggregate data from a pool of users. Hence, it is guaranteed that no user’s information is being processed on or presented on a personal level or specifically. Any results or conclusions drawn from the data report are speculated calculation and analysis of all employees within the scope of an organization.

According to Section 2.1.1, it is evident that the data from ViaEsca are processed following the CIA triad. In terms of confidentiality, the company data report will only be shown on the admin panel, which only ViaEsca administrators have access to. In terms of integrity, no changes are to be made to the users' data. The users' data are merely input for mathematical calculations for statistical results. There will also be no option to edit the statistics shown on the report. Lastly, in terms of availability, the queries' run-time to produce the statistical results are made to not exceed five seconds in order to avoid bottlenecks in the database, which may hinder the performance of the database as well as the application. Furthermore, five seconds is the time-out limit for the page to load.

4.2 Implementation process

There are three main steps in the plan of this project's implementation process, which are creating the company model, generating and displaying the company data report, and lastly doing a similar thing for the company wellness report. First, there should be a feature that provides a method for the administrators to group users to the same company. At this step, a link between a user and a company and a link between a company and an access code batch is created. Second, a company data report class is created to gather and handle data, doing calculations and the results will be displayed in charts and tables on the company section of the admin panel. Thirdly and lastly, a class for fetching and handling the wellness polls' answers is implemented. From this point onwards, any users' personal information, such as names, emails, etc., that is going to be displayed are anonymized.

4.2.1 Company model, controller, and views

First, a **Company** model should exist and bear a foreign key with the User model and **AccessCodeBatch** model. The company only needs a name for now. The timestamps of when a record is created and updated are automatically added by Rails when creating a migration file with *rails g migration CreateCompanies*.

```
class CreateCompanies < ActiveRecord::Migration[5.0]
  def change
    create_table :companies do |t|
      t.string :name
      t.timestamps
    end
  end
end
```

Figure 9. Migration file to create the Company model

Then, I run *rails db:migrate* to run through all migrations and a company table is created in the database. I proceed to create a “company.rb” file under the path “/app/models”. This is the Company model file. To define the relationships between **User** and **Company** or between **Company** and **AccessCodeBatch**, I generate two more migration files to add a *company_id* column to each associated table. The line to do so, for example, for the User model is `add_reference :users, :company, foreign_key: true`. The next step to define associations between models, according to Rails and Active Record, is to declare a *belongs to* and a *has one* or *has many* relationships directly on their models. The logic I have for this is a user and an access code batch have a *belongs_to* relationship with a company, and a company has a *has_many* relationship with users and access code batches. Additionally, as mentioned in the plan, there is a total of 7 groups of users and I will define them inside the company model as associations with scope. For instance, the company’s users that have a weight-loss goal are defined as follows: `has_many :weight_loss_users, -> { has_bought_starter.where(lose_weight: true) }, class_name: 'User'`

This calls for another scope name *has_bought_starter* which is defined in **User** to filter and take only users that have a *starter_start_date*, which is an indicator that the users have started their 21 days journey. The other half of the scope is self-explanatory. Hence, whenever we want to retrieve this type of users for a company, they can be called as *company.weight_loss_users*. A similar logic can be applied for the other user types: users with no weight-loss goal, users that have a Validic connection and users that do not, users that use the light version of the application, and users that do not.

Second, there are already existing forms that are used for creating and editing users and access code batches on the admin panel. I then modify those forms to have a company value and pass its parameters to the appropriate controllers' methods. For the admins to be able to create a new code batch for a new company, I use Rails' way of allowing nested attributes of a referenced model inside another model. Inside "access_code_batch.rb", I added this line: *accepts_nested_attributes_for :company*. The tricky part of this process is to make it possible to add either a new company or an existing company to a code batch. If it is a new company, a company's name parameter will be passed. Otherwise, only a "company_id" will be passed to be the reference key. Hence, I declare the parameters in the admin's "access_code_batch_controller" as below:

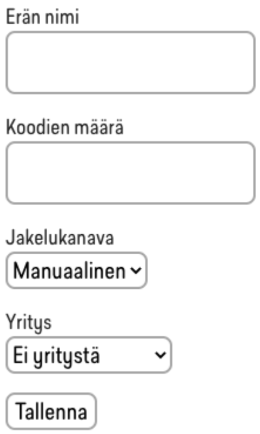
```
private def access_code_batch_params
  params.require(:access_code_batch).permit(:name, :n_codes, :distribution_channel,
:company_id)
end

private def access_code_batch_params_with_new_company
  params.require(:access_code_batch).permit(:name, :n_codes, :distribution_channel,
company_attributes: [ :id, :name, :created_at, :updated_at ])
end
```

Figure 10. Defining parameters for creating access code batch with companies

I create a drop-down select field to define whether:

- To add a company, admins can choose "Ei yritystä", which means "no company" (Figure 11.)
- To create a new company, admins can choose "Uusi yritys", which means "new company" and a company's name field will appear. (Figure 12.)
- To add an existing company, admins can choose "Nykyinen yritys", which means "existing companies" and a drop-down selection of existing companies will appear (Figure 13.)



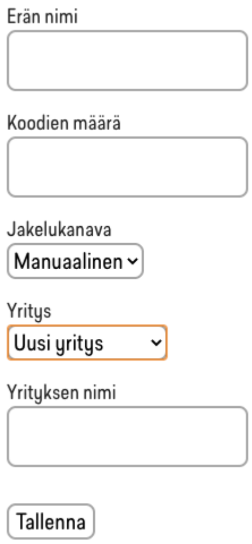
Erän nimi

Koodien määrä

Jakelukanava
Manuaalinen

Yritys
Ei yritystä

Figure 11. Creating a new access code batch without a company



Erän nimi

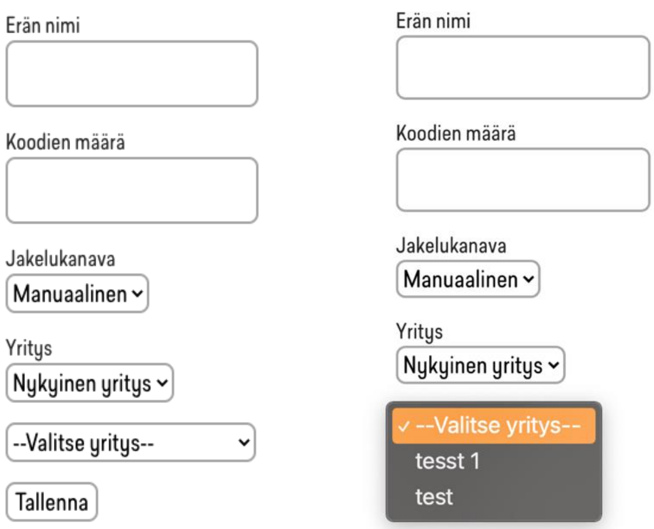
Koodien määrä

Jakelukanava
Manuaalinen

Yritys
Uusi yritys

Yrityksen nimi

Figure 12. Creating a new access code batch for a new company



Erän nimi

Koodien määrä

Jakelukanava
Manuaalinen

Yritys
Nykyinen yritys

--Valitse yritys--

Tallenna

Erän nimi

Koodien määrä

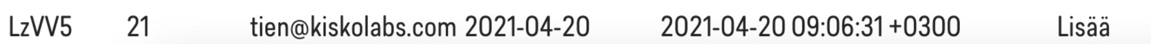
Jakelukanava
Manuaalinen

Yritys
Nykyinen yritys

✓ --Valitse yritys--
tesst 1
test

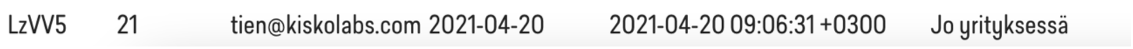
Figure 13. Creating a new access code batch for an existing company

I use JavaScript in the form of CoffeeScript to make the new company name field and the existing companies drop-down selections field toggled based on the value of the “Company” selection. The same is done also for the editing user so that the administrators can manually add or remove a user from a company. In the scenario that a company was later added to an already in-use access code batch, it would be time-consuming to manually edit each user in that code batch and add them to that company. Thus, I also add an “Add to company” link on the user line directly on the access codes view. (Figure 14.) When clicked, the user will be added to the company immediately through the *update* action of the company controller, and the status of the line will change to “Already in company”. (Figure 15.)



LzVV5 21 tien@kiskolabs.com 2021-04-20 2021-04-20 09:06:31 +0300 Lisää

Figure 14. Before adding user to the code batch’s company



LzVV5 21 tien@kiskolabs.com 2021-04-20 2021-04-20 09:06:31 +0300 Jo yrityksessä

Figure 15. After adding a user to the code batch’s company

Third, I add a “company_controller.rb” file under “/app/controllers/admin” and declare the necessary resources (controllers) and routes inside the “routes.rb” file as follows: `resources :companies, only: [:index, :update, :new, :create]`

For the basic controller's actions that Rails support, I choose to use *index*, *update*, *new,* and *create*. I use the *index* action to list all companies with a link to each company's report embedded in their names. On the *index* view, there is also a button to add a new company, which will render to the *new* view and use the *create* action to create a new company entity.

4.2.2 Company data report

I create a class name "company_data_report.rb" under "/app/models" to help calculate the mean and median values for the company data report, which are quantitative data. The class should take in two objects: *users* and *date_range*. However, the date_range is set to default as *nil* because the calculating methods defined in this class are going to be used for both the date range data report and the starter journey report, which does not require a date range input. In the *initialize* method of class, I define the variables as below:

```
def initialize(users, date_range: nil)
  @users = users
  @date_range = date_range
  @all_meals = ConsumedMeal.where(user_id: users)
  @all_sleeps = SleepMeasurement.where(user_id: users)
  @today = Date.current
  @all_activities = Activity.where(user_id: users)
  @all_manual_activities = ManualActivity.where(user_id: users)
end
```

Figure 16. Initialize method of the company data report class

These variables are generated whenever this class is called for use and they can be applied to any methods in this class. The four models in action and their roles are defined as below:

- **ConsumedMeal** stores the meals that users marked as eaten. Each object in this model has a "calories" value and a **DateTime** value. From here, it is possible to calculate how many calories users have consumed and how many meals they have eaten in a day.
- **SleepMeasurement** stores the sleeping time per day that is recorded from the users' Validic activity bracelets. Thus, sleep data are unavailable for

users without Validic connections or for users that take off the tracking device when they sleep. The sleep measurements are in seconds and they should be converted to minutes after calculation.

- **Activity** stores the activities recorded from Validic bracelets, including values for burnt calories and active calories per day. These data are also only available for users that have Validic connections.
- **ManualActivity** stores the activities that users manually log, including also burnt calories and active calories per day. These data are available for both users with Validic connections and users that do not. Hence, prior to doing any calculation, I have to merge the caloric values in this model and the **Activity** model for each user based on the user's id.

At this point, the data obtained are four lists of active records that belong to the pre-defined user's group object that is passed to this company data report class. There are three statistics reports in demand for this model class: (i) An overall table report that shows statistics of all users over all time (the time they have an active subscription for the app's service), (ii) A report of statistics based on a date range input and different groups of users of choice, (iii) A starter statistics report for the starter journey (first 21 days) of different groups of users. In this model class, I write two methods for each data set: one for calculating the mean and one for calculating the median. Hence, there are around 30 main methods for organizing data for the 5 data sets, whilst many others are secondary methods that assist the main ones. There are two reasons why I did not combine all the organization methods into one. Firstly, the data sets' attributes are not similar to each other and the data unit needed cannot be obtained in the same way. For instance, there is only one logged activity for one user in a day, whereas for the consumed meals, there are usually multiple records in one day. Secondly, although there seem to be too many methods, it is easier this way to keep track of what method does what. This is from my actual hands-on experience for this project. At first, I tried to combine the methods into one generalized method that can fit all models, but it soon proved to be unnecessarily difficult and tough to troubleshoot.

Next, I write methods to group the statistics into one array by calling the main methods' names. This enables easier displaying of statistics later in the views. For example, the below method exports the overall average values:

```
def overall_averages
  [overall_average_calories_consumed, overall_average_calories_burned,
overall_average_calories_active, overall_average_meals, overall_average_sleeps]
End
```

Figure 17. Method to export overall averages value as an array

When calculating the mean value in each calculation method, there are 3 steps that I follow for every data set. (i) I group the data by users' ids, (ii) I loop through each user and group all the data belonging to that user by dates, (iii) I construct an array to store the numbers as it is effortless to do calculations with arrays. While writing values into the said array, I also skip where the data unit equals 0. For the type of report where the *date_range* value is given, the *date_range* will be applied at the first step. The method to calculate the arithmetic mean for the array is as follow:

```
def calculate_average(array)
  return 0 if array.empty?
  return (array.sum.to_f / array.length).round(2)
end
```

Figure 18. Method to calculate the average of an array

The method basically takes an array of numbers, then sums up all elements in that array (indicated by *array.sum*), converts the sum to float data type (with *to_f*), and divides it for the number of elements in the array (indicated by *array.length*). The result is then rounded up to 2 decimal digits. The methods are written to find the medians copy the same format. The difference is that I directly put all available values into one array. The method to find the median value is as follow:

```
def find_median(array)
  return nil if array.empty?
  sorted = array.compact.sort
  len = sorted.length
  (sorted[(len - 1) / 2] + sorted[len / 2]) / 2.0
end
```

Figure 19. Method to find the median value of an array

First, the *nil* (empty) values are removed (*compact*) from the array if there are any, then the array is re-ordered from lowest to highest value (*sort*). The number of elements in the array is defined by calling *length* on the array. In the scenario that the array has an odd number of elements, the two values that are accounted for the average calculation at the last line of code will be the same. Otherwise, if the array has an even number of elements, there will be two middle values and the median is defined as the average of those two numbers. The two methods to calculate average and medians are called at the end of each main method appropriately.

As for the starter report, I have to first organize the data gathered from the four models to be from day 1 to day 21 for each user. Then I store them inside hashes. Hash's syntax includes keys and values bound to that key. The key can be an integer or a string, but the value tied to that key can be any data type, even an array or a hash. The first day is equal to a user's *starter_start_date*, which is stored in the database, so the data organizing process should stop at the user's starter_start_date plus 20 days. The hash format of the organized data is simplified as follows: *{user_id => {"Day1" => data, "Day2" => data, …, "Day21" => data }, user_id_2 => {…}…}*. After that, the data can be summarized to produce statistics similar to the date range report. The only tricky part is to gain access to data stored in a hash nested inside another hash. Nonetheless, I still use hashes to store different kinds of organized data as well as data that have gone through the summarizing process.

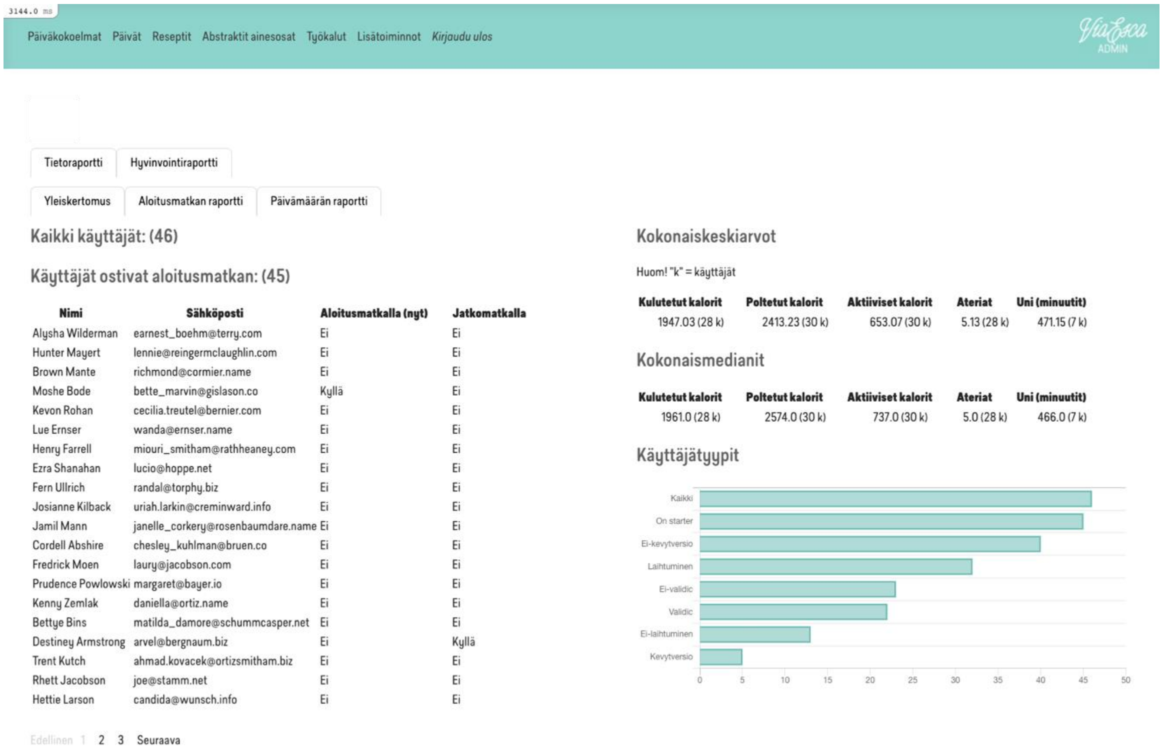
Before implementing the views for this quantitative data report, it is necessary to test run the written methods in the Rails console. With the help from the Execution time gem, I can monitor how much time my queries take to run. At the same time, the Bullet gem is also used here to help me identify where I needed to preload the associations' records.

I then declare resources and routes for the *data_reports_controller* that I am going to create for presenting the statistics. The "data_reports_controller.rb" file is created under the path "/app/controllers/admin/companies". This allows the controller to be declared nested inside the companies_controller. The routes in “routes.rb” file looks as follows:

```
resources :companies, only: [:index, :update, :new, :create] do
   resource :data_reports, only: [:show, :new, :create], controller: 'companies/data_reports' do
    member do
     get 'new_compare_data' => 'companies/data_reports#new_compare_data'
     post 'compare_data' => 'companies/data_reports#compare_data'
     get 'download_csv' => 'companies/data_reports#download_csv'
     get 'starter_report' => 'companies/data_reports#starter_report'
     post 'starter_report' => 'companies/data_reports#create_starter_report'
     get 'new_compare_starter_report' =>
'companies/data_reports#new_compare_starter_report'
     post 'compare_starter_report' => 'companies/data_reports#compare_starter_report'
    end
   end
end
```

Figure 20. Declaring needed routes for the company data report in “routes.rb”

When we click on the company name from the company list, the method *show* of the *data_reports* controller is rendered. There are tabs to allow navigating between different statistics reports. On *data_reports/show* view, there is a table that shows the users’ names, emails, and a simple yes/no value as to whether they are on any active subscription at the moment. Next to that, there are tables displaying the overall statistics for all users of all time. The numbers of users contributing to the displayed statistics are put in brackets with a letter “k” that comes after each of them which indicates the word “käyttäjät” meaning “users”. There is also a bar chart showing the number of users for all users, users that have a *starter_start_date* value, and the 6 user groups in descending order. Figure 21 is the web view of the overall report page with the company name supposedly on top of the navigation tabs, but I leave it out due to privacy reasons. Meanwhile, the users’ names and emails are shown because they have already been replaced by Faker data.



3144.0 ms

Päiväkokoelmat Päivät Reseptit Abstraktit ainesosat Työkalut Lisätoiminnot *Kirjaudu ulos*

Tietoraportti Hyvinvointiraportti

Yleiskertomus Aloitusmatkan raportti Päivämäärän raportti

Kaikki käyttäjät: (46)

Käyttäjät ostivat aloitusmatkan: (45)

Nimi	Sähköposti	Aloitusmatkalla (nyt)	Jatkomatkalla
Alysha Wilderman	earnest_boehm@terry.com	Ei	Ei
Hunter Mayert	lennie@reingermclaughlin.com	Ei	Ei
Brown Mante	richmond@cormier.name	Ei	Ei
Moshe Bode	bette_marvin@gislason.co	Kyllä	Ei
Kevon Rohan	cecilia.treutel@bernier.com	Ei	Ei
Lue Ernser	wanda@ernser.name	Ei	Ei
Henry Farrell	miouri_smitham@rathheaney.com	Ei	Ei
Ezra Shanahan	lucio@hoppe.net	Ei	Ei
Fern Ullrich	randal@torphy.biz	Ei	Ei
Josianne Kilback	uriah.larkin@creminward.info	Ei	Ei
Jamil Mann	janelle_corkery@rosenbaumdare.name	Ei	Ei
Cordell Abshire	chesley_kuhlman@bruen.co	Ei	Ei
Fredrick Moen	laury@jacobson.com	Ei	Ei
Prudence Powlowski	margaret@bayer.io	Ei	Ei
Kenny Zemlak	daniella@ortiz.name	Ei	Ei
Bettye Bins	matilda_damore@schummcasper.net	Ei	Ei
Destiney Armstrong	arvel@bergnaum.biz	Ei	Kyllä
Trent Kutch	ahmad.kovacek@ortizsmitham.biz	Ei	Ei
Rhett Jacobson	joe@stamm.net	Ei	Ei
Hettie Larson	candida@wunsch.info	Ei	Ei

Edellinen 1 2 3 Seuraava

Kokonaiskeskiarvot

Huom! "k" = käyttäjät

Kulutetut kalorit	Poltetut kalorit	Aktiiviset kalorit	Ateriat	Uni (minuutit)
1947.03 (28 k)	2413.23 (30 k)	653.07 (30 k)	5.13 (28 k)	471.15 (7 k)

Kokonaismedianit

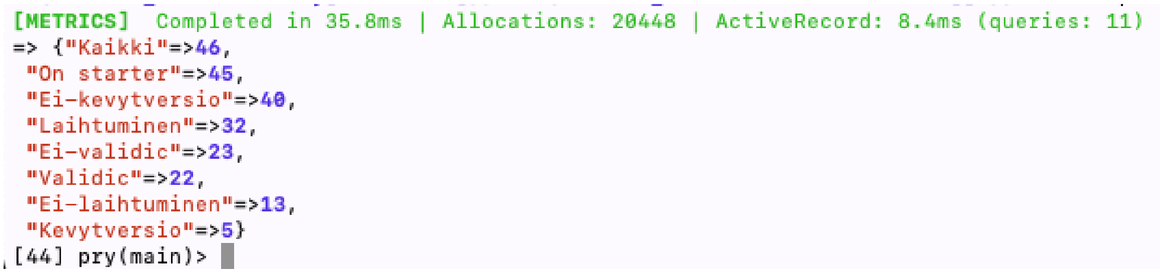
Kulutetut kalorit	Poltetut kalorit	Aktiiviset kalorit	Ateriat	Uni (minuutit)
1961.0 (28 k)	2574.0 (30 k)	737.0 (30 k)	5.0 (28 k)	466.0 (7 k)

Käyttäjätyypit

Figure 21. Overall report page of a company

One more reason to support the use of hashes to store summarized data is that they are compatible with Chartkick. Not only that they are easy to be observed in the Rails console but they can also be used to generate charts. The bar chart shown in Figure 21 is defined as follows on the view: = bar_chart @company.users_data, colors: ["#62b7ad"]

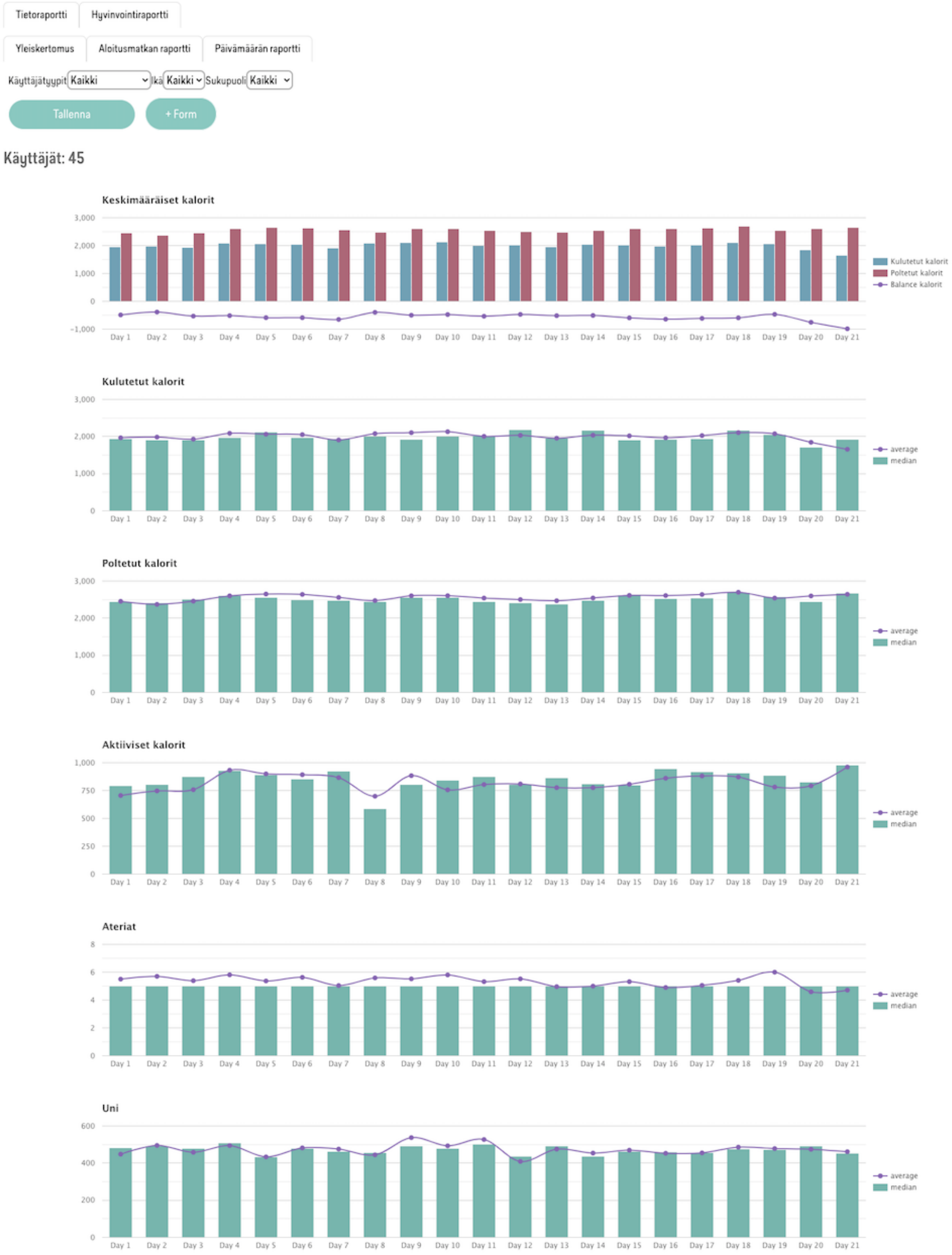
And the @company.users_data method produced a hash that looks like below:



```
[METRICS]  Completed in 35.8ms | Allocations: 20448 | ActiveRecord: 8.4ms (queries: 11)
=> {"Kaikki"=>46,
 "On starter"=>45,
 "Ei-kevytversio"=>40,
 "Laihtuminen"=>32,
 "Ei-validic"=>23,
 "Validic"=>22,
 "Ei-laihtuminen"=>13,
 "Kevytversio"=>5}
[44] pry(main)> 
```

Figure 22. The number of users for each user group

Next, the starter statistics report can be viewed when navigating to the tab “Aloitusmatkan raportti”. For this view, I define the 4 routes: one shows a form that asks for the user types and dimensions (age and gender), one adds a second form similar to the first one for easy comparison of statistics, and the other two are to process each of the forms. The form consists of three fields: user type, gender, and age. The user type and the age are processed in the data_reports_controller. However, I put the method for processing the users' age in the **CompanyDataReport** class. When pressing “Save” (“Tallenna”), the form is submitted through JavaScript. Thus, the statistics are rendered below the form without reloading the page.



Käyttäjätyypit Kaikki Ikä Kaikki Sukupuoli Kaikki

Käyttäjät: 45

Figure 23. Starter statistics by charts for all users in a company

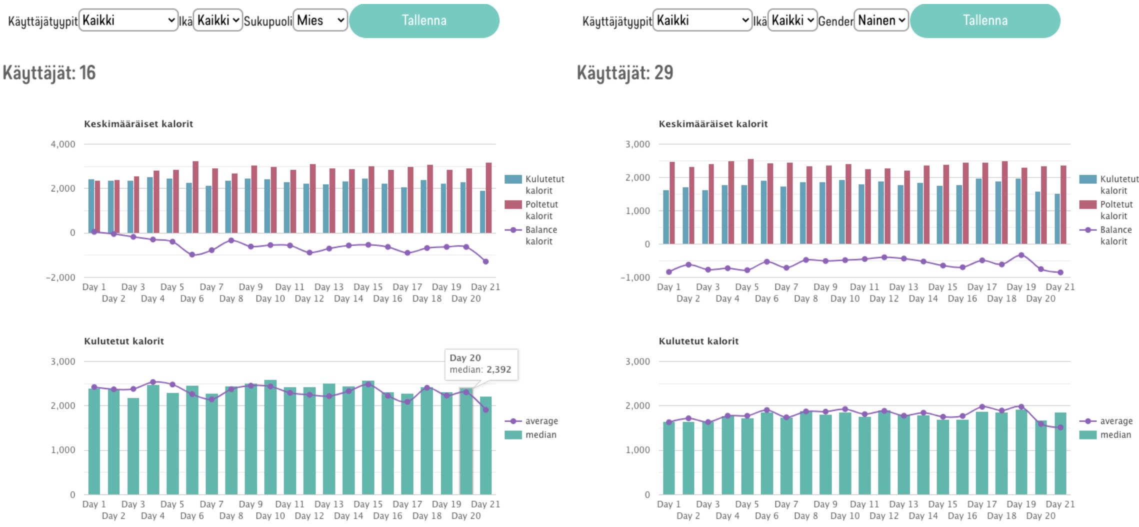
The charts shown in Figure 23 are also generated from hashes of data. The combo chart, for instance the first chart, is defined as below in the view:

```
= column_chart [{name: "average", data: @averages[0] }, {name: "median", data: @medians[0] }], adapter: 'google', library: { :series => { 0 => { type: "line"} } }, title: 'Kulutetut kalorit', colors: ["#8c62b7", "#62b7ad"], id: SecureRandom.hex(7)
```

Figure 24. Display combo chart in the report’s view using Chartkick

The first chart shows the average calorie balance, where the blue bars are average calories consumed, the red bars are average calories burnt and the purple line indicates the calorie balance. The calorie balance can be obtained by subtracting the two calorie values. The five teal combo charts (bar and line) below are statistics of the five data sets, with the bars representing the medians and the lines representing the averages. Below the charts are the tables for averages and medians, which are not displayed in Figure 25 because they have 21 rows each. Basically, the charts have delivered all information that is available on the two tables, except for the number of users that are accounted for. However, the perk of having tables is that they can be used easily to export the statistics.

When clicking the “+ Form” button, a new page will be opened with 2 forms. (Figure 25.)



Käyttäjätyypit Kaikki Ikä Kaikki Sukupuoli Mies

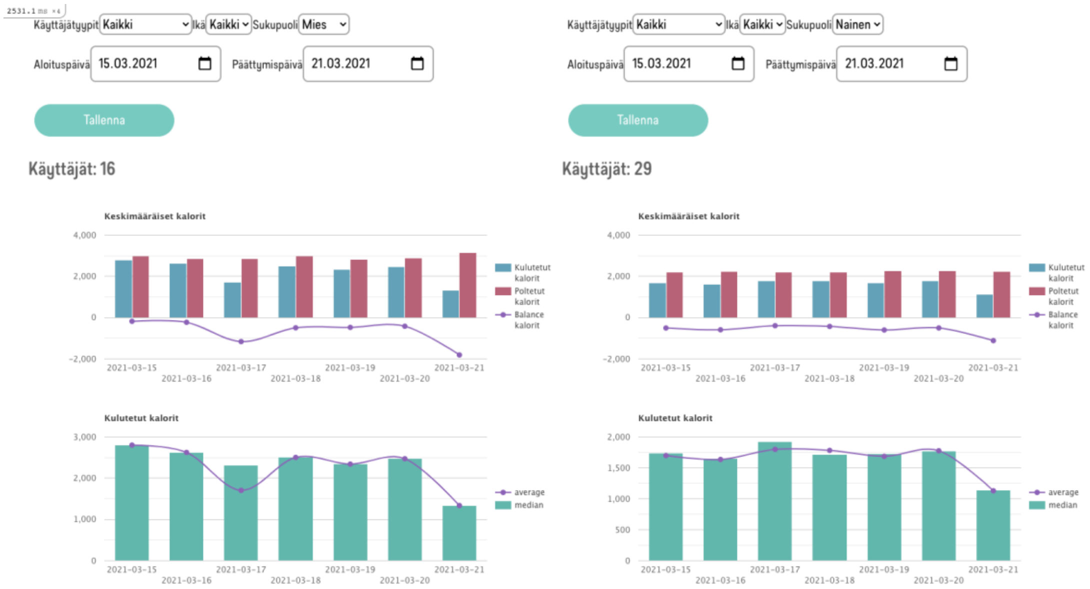
Käyttäjät: 16

Käyttäjätyypit Kaikki Ikä Kaikki Gender Nainen

Käyttäjät: 29

Figure 25. Comparison of starter statistics between male and female users in a company

The report format of 6 charts and 2 tables underneath them are used for all of the views of the stater report and the date-range-based report. However, the screenshots capture just a relevant part of the views to prove my point of demonstrating how the pages work. The statistics report by dates can be viewed when navigating to the tab “Päivämäärän raportti” and it is very similar to the starter statistics report. The only difference is that there are two more fields to the form, a start_date field, and an end_date field to set a *date_range* for the statistics. (Figure 26.)


Käyttäjätyypit Kaikki Ikä Kaikki Sukupuoli Mies

Aloituspäivä 15.03.2021 Päättymispäivä 21.03.2021

Tallenna

Käyttäjät: 16

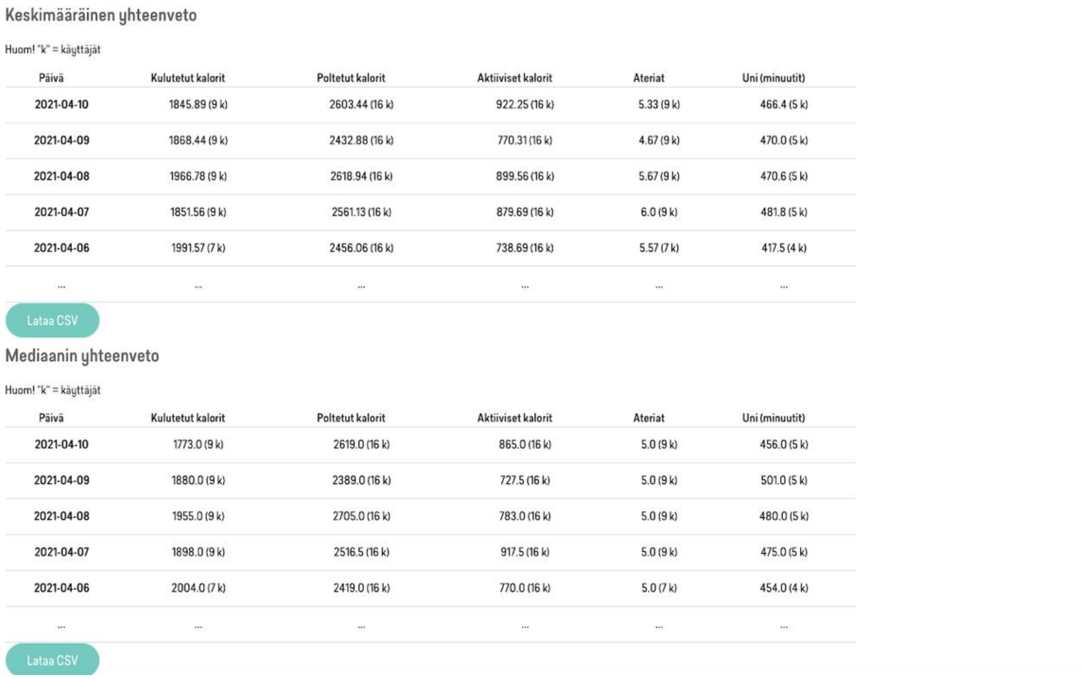
Käyttäjätyypit Kaikki Ikä Kaikki Sukupuoli Nainen

Aloituspäivä 15.03.2021 Päättymispäivä 21.03.2021

Tallenna

Käyttäjät: 29

Figure 26. Comparison of statistics between male and female users in a specific date range



Keskimääräinen yhteenveto

Huom! "k" = käyttäjät

Päivä	Kulutetut kalorit	Poltetut kalorit	Aktiiviset kalorit	Ateriat	Uni (minuutit)
2021-04-10	1845.89 (9 k)	2603.44 (16 k)	922.25 (16 k)	5.33 (9 k)	466.4 (5 k)
2021-04-09	1868.44 (9 k)	2432.88 (16 k)	770.31 (16 k)	4.67 (9 k)	470.0 (5 k)
2021-04-08	1966.78 (9 k)	2618.94 (16 k)	899.56 (16 k)	5.67 (9 k)	470.6 (5 k)
2021-04-07	1851.56 (9 k)	2561.13 (16 k)	879.69 (16 k)	6.0 (9 k)	481.8 (5 k)
2021-04-06	1991.57 (7 k)	2456.06 (16 k)	738.69 (16 k)	5.57 (7 k)	417.5 (4 k)
...	...	...	...	...	...

Mediaanin yhteenveto

Huom! "k" = käyttäjät

Päivä	Kulutetut kalorit	Poltetut kalorit	Aktiiviset kalorit	Ateriat	Uni (minuutit)
2021-04-10	1773.0 (9 k)	2619.0 (16 k)	865.0 (16 k)	5.0 (9 k)	456.0 (5 k)
2021-04-09	1880.0 (9 k)	2389.0 (16 k)	727.5 (16 k)	5.0 (9 k)	501.0 (5 k)
2021-04-08	1955.0 (9 k)	2705.0 (16 k)	783.0 (16 k)	5.0 (9 k)	480.0 (5 k)
2021-04-07	1898.0 (9 k)	2516.5 (16 k)	917.5 (16 k)	5.0 (9 k)	475.0 (5 k)
2021-04-06	2004.0 (7 k)	2419.0 (16 k)	770.0 (16 k)	5.0 (7 k)	454.0 (4 k)
...	...	...	...	...	...

Figure 27. Tables of averages and medians of the statistics report by date range

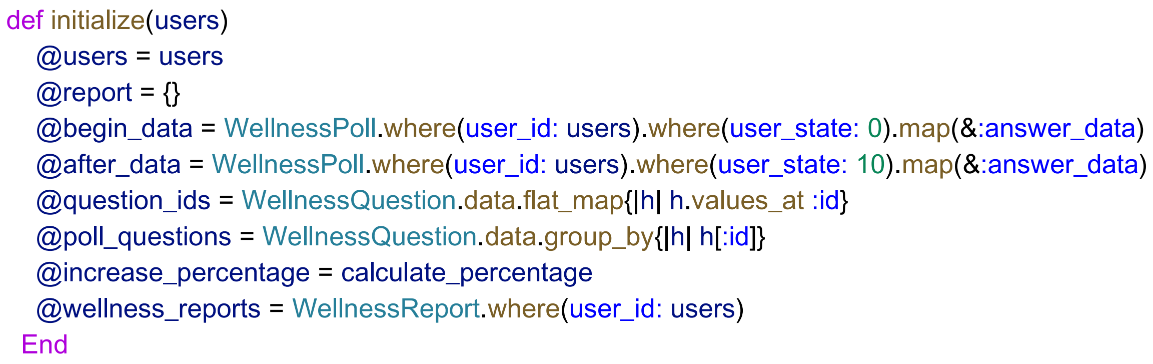
As seen from figure 27, I also add methods to the **CompanyReportClass** class to allow generating CSV files for the statistics based on date range input as an additional feature. The idea is to pass the needed parameters to the link of the "Lataa CSV" button that executes the method of generating a CSV. The must-have parameters contain the user type, gender, and age if given, the date range, and the statistics type (average or median). Those parameters are then passed to the **CompanyReportClass** to produce a CSV. The generated CSV's name is a string following the format:

```
#{@company.name}_#{params[:report_type]}_data_report_#{Date.today}.csv
```

To shortly conclude, the overall report's purpose is to present typical statistics of a company, because they are calculated for all users in the company. Meanwhile, the starter report paints a picture of users during their first 21 days. Initially, only averages were displayed as line charts. In my opinion, using combo charts to display medians and averages offer comparison value to the two data types. In case the mean and the median value are absurdly far from each other, it can be presumed that there are bad data or outliers that were overlooked, from the data collecting or organizing process. In comparison to the starter report, the report that allows a date range input also offers comparison of data for different date ranges. For example, the information on whether the users use the application more on weekdays than at weekends or vice versa can be achieved with this type of report.

4.2.3 Company wellness report

I create a class name “company_wellness_poll_report.rb” under “/app/models” to help calculate the mean and median values for the company data report, which are qualitative data. The class will take in the *users* object. In the *initialize* method of class, I define the variables as below:



```
def initialize(users)
   @users = users
   @report = {}
   @begin_data = WellnessPoll.where(user_id: users).where(user_state: 0).map(&:answer_data)
   @after_data = WellnessPoll.where(user_id: users).where(user_state: 10).map(&:answer_data)
   @question_ids = WellnessQuestion.data.flat_map{|h| h.values_at :id}
   @poll_questions = WellnessQuestion.data.group_by{|h| h[:id]}
   @increase_percentage = calculate_percentage
   @wellness_reports = WellnessReport.where(user_id: users)
 End
```

Figure 28. Initialize method of the company wellness report

Besides the *@users*, the meaning of each variable is as follows:

- *@report* is the hash that is going to store the summarized data of this class that is turning into statistics for pie charts on the views.
- *@begin_data* stores all the first polls’ answers by *@users*, which are extracted from the **WellnessPoll** model. Each poll has a “state” value. If it is 0, the poll is recorded at the start of the journey.
- *@after_data* stores all the second poll answers. These poll answers have a “state” of 10.
- *@question_ids* is an array of the ids of the poll questions defined in the **WellnessQuestion** class. This ensures that when the **WellnessQuestion** has added or removed poll questions, this **CompanyWellnessPollReport** class would also be updated correctly.

- *@poll_questions* stores the **WellnessQuestion**'s data grouped by their ids into a hash. The main attributes that **WellnessQuestion** has are *sort, question_category, question_description, answer_type, and answer_options*.
- *@increase_percentage* is the result of the *calculate_percentage* method. It is an array of numbers that are calculated by dividing the total value of the @after_data for the @begin_data for each question, in order to see whether the users' wellness/wellbeing were improved.

With the initialized variables above, the picture of summarizing statistics for the wellness poll answers is clear enough. However, it is not easy to deal with deeply nested hashes. First, the *answer_data* of every user on each poll (before and after) must be mapped out of the **WellnessPoll** records. Second, the *answer_options* inside the *answer_data* are separated and grouped by the value of each answer for each question. Those are stored in a hash, for example, if a question has five answer options, the hash looks as follows: *{"100" => 3, "75" => 2, "50" => 1, "25" => 0, "0" => 1}*. The keys of that hash are the values embedded in each answer option, while the integers followed up after each key are the number of users who gave those answers. At this stage, those answers' values are then used to calculate the increased percentage between the after-poll answers and the begin-poll answers. On the other hand, the number of users that have answered the poll can also be drawn by summing up the hash value for each key. Thirdly, before writing the statistics to the @report hash, I have to translate the answers' values to answers' description. Otherwise, it is impossible to understand what the statistics want to convey.

There are 3 scenarios: (i) none of the polls has any answer, (ii) only the begin-poll has answers, and (iii) both of the polls have answers. For the first, the @report should have nil values for both *begin* key and *after* key. Furthermore, the calculating method for increase percentage as well as the users counter method should not be called to action. For the second scenario, the *@report*'s *after* statistics is nil. The increase_percentages remains idle, while the users counter method is applied only for the *begin_data*. Lastly, for the third scenario, there should be statistics as well as the increased percentages and the user counts for both *begin* and *after* keys in *@report* hash.

When a user completes answering the second hash, he/she will get a link to view their wellness report. On that wellness report, there is a link used to share the report. The number of clicks that link received is stored for each of the **WellnessReport** objects. Thus, the number of share link clicks will also be grouped, summed, and then displayed on the wellness poll statistics report.

Next, I declare resources and routes for the *wellness_poll_answers_controller*. This controller also nested under the *companies_controller*. There are also 4 routes: 2 of them are used to display the forms and the other 2 are used to process the forms and show the statistics. The routes in “routes.rb” file at the time being looks as follows:

```
resources :companies, only: [:index, :update, :new, :create] do
      resource :data_reports, only: [:show, :new, :create], controller: 'companies/data_reports' do
        member do
          get 'new_compare_data' => 'companies/data_reports#new_compare_data'
          post 'compare_data' => 'companies/data_reports#compare_data'
          get 'download_csv' => 'companies/data_reports#download_csv'
          get 'starter_report' => 'companies/data_reports#starter_report'
          post 'starter_report' => 'companies/data_reports#create_starter_report'
          get 'new_compare_starter_report' =>
'companies/data_reports#new_compare_starter_report'
          post 'compare_starter_report' => 'companies/data_reports#compare_starter_report'
        end
      end
      resource :wellness_poll_answers, only: [:new, :create], controller:
'companies/wellness_poll_answers' do
        member do
          get 'new_compare_data' => 'companies/wellness_poll_answers#new_compare_data'
          post 'compare_data' => 'companies/wellness_poll_answers#compare_data'
        end
      end
    end
```

Figure 29. Declaring needed routes for the company wellness report in “routes.rb”

The company wellness poll report can be viewed when navigating to the tab called “Hyvinvointiraportti”. The idea is the same as implemented in the company data report. There are 3 fields on the form: user type, gender, and age.

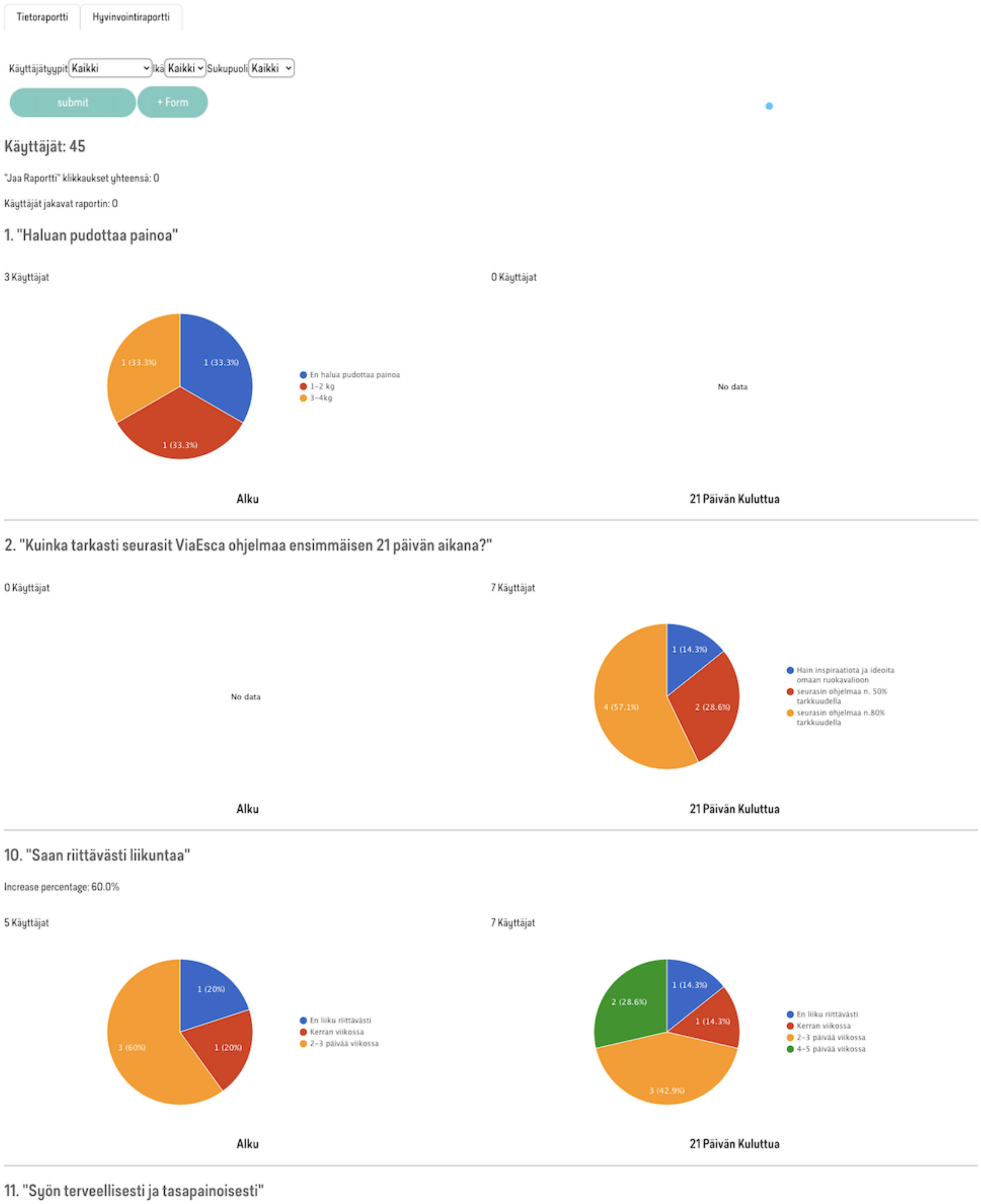
When submitting the form, the total number of total users chosen based on the form and the number of “share report” clicks are displayed at the top. Below that, the 2 pie charts, before and after, appear for each question listed. The lines that added the two pie charts into the views are:

```
= pie_chart begin_data, adapter: 'google', library: { pieSliceText: 'value-and-percentage' }, id: SecureRandom.hex(7)

= pie_chart after_data, adapter: 'google', library: { pieSliceText: 'value-and-percentage' }, id: SecureRandom.hex(7)
```

Figure 30. Display pie charts in the company wellness report’s view

Above the pie charts, the number of users answered to that poll is displayed alongside the increased percentage if there are statistics shown for both polls. The total number of total users chosen and the number of “share report” clicks are displayed at the top. (Figure 31.)



Käyttäjät: 45

"Jaa Raportti" klikkaukset yhteensä: 0

Käyttäjät jakavat raportin: 0

1. "Haluan pudottaa painoa"

3 Käyttäjat

Alku

0 Käyttäjat

No data

21 Päivän Kuluttua

2. "Kuinka tarkasti seurasit ViaEsca ohjelmaa ensimmäisen 21 päivän aikana?"

0 Käyttäjat

No data

Alku

7 Käyttäjat

1 (14.3%)
2 (28.6%)
4 (57.1%)
● Hain inspiraatiota ja ideoita omaan ruokavalioon
● seurasin ohjelmaa n. 50% tarkkuudella
● seurasin ohjelmaa n.80% tarkkuudella

21 Päivän Kuluttua

10. "Saan riittävästi liikuntaa"

Increase percentage: 60.0%

5 Käyttäjat

Alku

7 Käyttäjat

21 Päivän Kuluttua

11. "Syön terveellisesti ja tasapainoisesti"

Figure 31. The first three questions of the wellness poll statistics report for all users

Figure 31 shows the statistics for the first three questions of the wellness poll. Question 1 is only asked in the begin-poll and question 2 is only asked in the after-poll. Even so, the statistics are still correctly stored and displayed. For these two questions, there is no *increase_percentage* value. On question 3, as there are data for both polls, an *increase_percentage* value is displayed. On another note, it is also possible to compare the pie charts between different groups of users. (Figure 32).

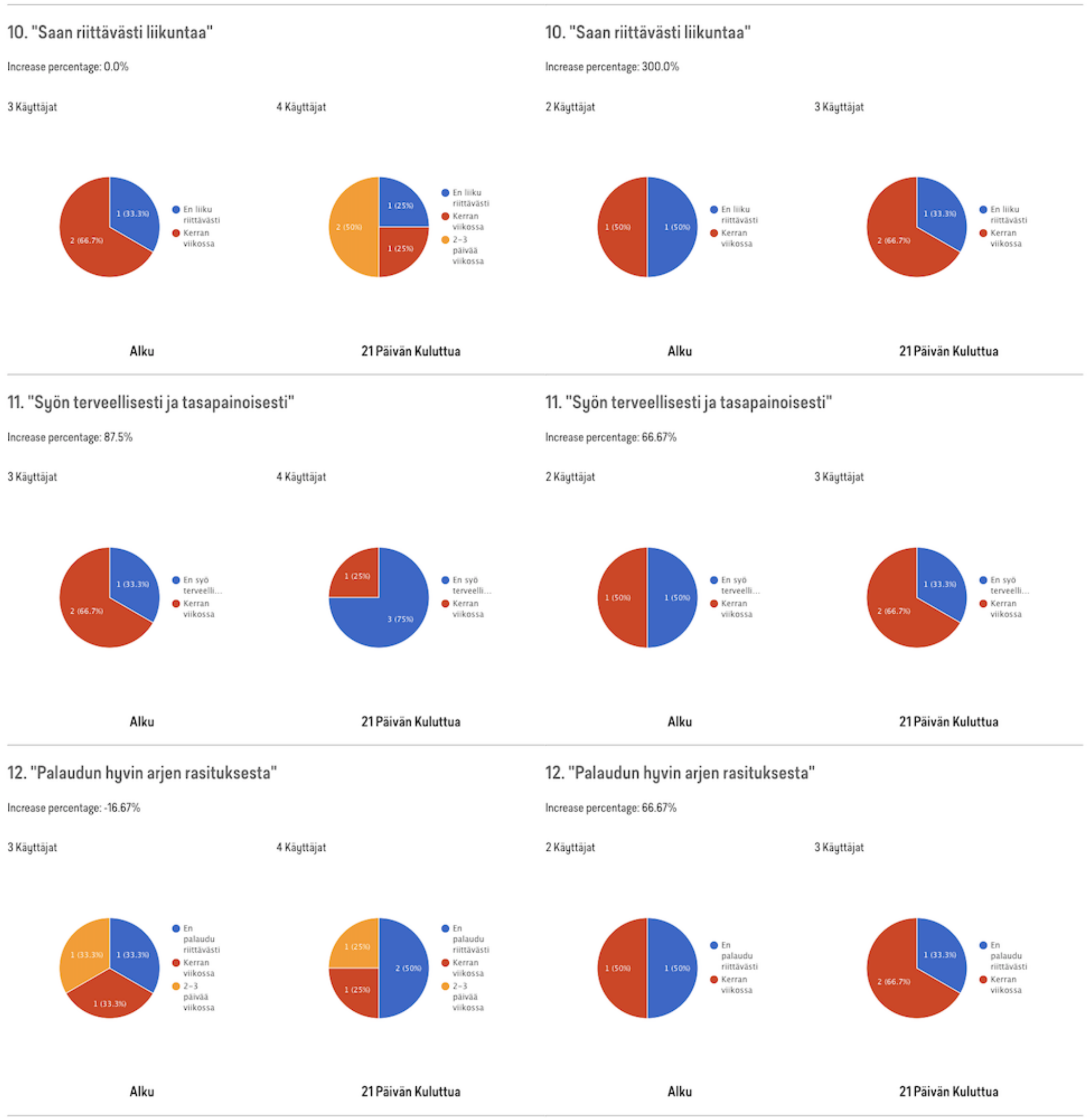


Figure 32. Comparison of statistics between male (left) and female (right) in a company

Figures 31 and 32 show only statistics for 6 questions of 16 questions in total. On a side note, the wellness polls feature is fairly new, as it was carried out from February to March of 2021. Hence, at the time of this thesis, there is not much that can be concluded from the statistics presented here. However, according to Figures 31, 32 and by observing the increased percentage, mostly all users' answers have increased for the “after” poll. Since the questions are about their wellbeing, it can be presumed that the users' wellbeing has improved thanks to the impact of ViaEsca’s starter journey. Furthermore, the *increase_percentage* values for the female users are higher than that for the male users.

5 CONCLUSION

The theoretical background of the thesis has introduced the definition of data, the two data types (quantitative and qualitative data) as well as how data is preserved and processed in this digital era of humankind. Furthermore, the thorough study on how to conduct descriptive statistics has provided adequate knowledge for me about the meanings and methods of data collection, data organization, data summarization, data presentation/visualization as well as the fundamentals of statistical analysis and interpretation. All of the above have established a solid groundwork for practicing constructing and presenting statistics.

Thanks to the theoretical background studying statistics, the outcome of the practical part successfully provides a feature for ViaEsca’s administrators to monitor the statistics of users in companies. The company data report and the company wellness polls’ answer report are descriptive statistics and they are displayed in tables and charts. The statistics are generated for each user group chosen by submitting a form. By allowing similar inputs of the user groups and dimensions for both reports’ forms, it also supports finding the link between the quantitative data and the qualitative data for different users. Moreover, it assists in examining whether good quantitative statistics also result in good qualitative statistics. As a result, ViaEsca can use this feature as a means to report statistics back to the companies or to analyze and make changes to improve their service to be suitable and beneficial for a wider range of users.

Possible further developments for this project include (i) defining and excluding the margin of error for statistics when there are more data. This can also be applied to find out the outliers of data. Therefore, a more accurate representation of data and the trend of them can be acquired. (ii) Deeper analysis for the statistics can be conducted when there are large enough data to find the relationship between different data sets, or which data attributes correlates strongly to another, for instance, the question of whether the gender and age of users have any impact on the statistics can be answered with further research and analysis.

REFERENCES

Anusha, R. 2020. A database is a collection of information that is organized. WWW document. Available at: https://www.zoomtute.com/content/blogs/28/a-database-is-a-collection-of-information-that-is-organized-dot [Accessed 2 May 2021].

Australian Bureau of Statistics. 2021. Statistical language – What are data? WWW document. Available at: https://www.abs.gov.au/websitedbs/D3310114.nsf/Home/Statistical+Language+-+what+are+data [Accessed 15 March 2021].

Bhandari, P. 2020. A step-by-step guide to data collection. WWW document. Available at: https://www.scribbr.com/methodology/data-collection/ [Accessed 1 May 2021].

Blitz, S. 2018. How to Decide if PostgreSQL is the Right Reporting Database for You. WWW document. Available at: https://www.sisense.com/blog/decide-postgresql-reporting-right/ [Accessed 30 April 2021].

Brooks, C. 2020. What Is Statistical Analysis? WWW document. Available at: https://www.businessnewsdaily.com/6000-statistical-analysis.html [Accessed 2 May 2021].

Corporate Finance Institute. 2021. What is Data Anonymization? WWW document. Available at: https://corporatefinanceinstitute.com/resources/knowledge/other/data-anonymization/ [Accessed 15 March 2021].

Fowler, M., Rice, D. 2003. Patterns of Enterprise Application Architecture. Massachusetts: Addison-Wesley Professional.

Franco, R., Fallaize, R., Lovegrove, J., Hwang, F. 2016. Popular Nutrition-Related Mobile Apps: A Feature Assessment. *JMIR Mhealth Uhealth* 4(3), 85.

Fruhlinger, J. 2020. The CIA triad: Definition, components and examples. WWW document. Available at: https://www.csoonline.com/article/3519908/the-cia-triad-definition-components-and-examples.html [Accessed 16 March 2021].

Gardner, K. 2020. 3 Ways of Effectively Organizing Data for Better Analysis and Presentation. WWW document. Available at: https://devdojo.com/kevgardner83/3-ways-of-effectively-organizing-data-for-better-analysis-and-presentation [Accessed 2 May 2021].

Gdpr.eu. 2021. General Data Protection Regulation (GDPR). WWW document. Available at: https://gdpr.eu/tag/gdpr/ [Accessed 15 March 2021].

Github. 2021. Akane/chartkick. Available at: https://github.com/ankane/chartkick [Accessed 15 March 2021].

Github. 2021. Faker-ruby/faker. WWW document. Available at: https://github.com/faker-ruby/faker [Accessed 15 March].

Github. 2021. Flyerhzm/bullet. WWW document. Available at: https://github.com/flyerhzm/bullet [Accessed 30 April 2021].

Hand, D. 2008. Statistics: A Very Short Introduction. Oxford: Oxford University Press.

IBM. 2020. Normalization. WWW document. Available at: https://www.ibm.com/docs/en/ztpf/2020?topic=database-normalization [Accessed 2 May 2021].

Maayan, D. 2019. Why You Should Learn PostgreSQL for Data Science. WWW document. Available at: https://www.dataversity.net/why-you-should-learn-postgresql-for-data-science/ [Accessed 30 April 2021].

Maringer, M., van't Veer, P., Klepacz, N. et al. 2018. User-documented food consumption data from publicly available apps: an analysis of opportunities and challenges for nutrition research. *Nutrition Journal* 17, 59.

Nguyen, H. 2016. Why You Should Use Postgres Over MySQL For Analytics Purpose. WWW document. Available at: https://www.holistics.io/blog/why-you-should-use-postgres-over-mysql-for-analytics-purpose/?utm_campaign=pg_mysql&utm_source=medium [Accessed 30 April 2021].

Oxford Learner's Dictionaries. 2021. Data. WWW document. Available at: https://www.oxfordlearnersdictionaries.com/definition/english/data?q=data [Accessed 25 April 2021].

Panik, M. 2012. Statistical Inference: A Short Course. Wiley.

Postgresql. 2021. PostgreSQL Documentation, Preface. WWW document. Available at: https://www.postgresql.org/docs/current/ [Accessed 15 February 2021].

QuestionPro. 2021. Qualitative data: Definitions, Types, Analysis and Examples. WWW document. Available at: https://www.questionpro.com/blog/qualitative-data/ [Accessed 16 March 2021].

QuestionPro. 2021. Quantitative data: Definitions, Types, Analysis and Examples. WWW document. Available at: https://www.questionpro.com/blog/quantitative-data/ [Accessed 16 March 2021].

Rails Guide. 2021. Active Record basics. Version 6.1.3.1. WWW document. Available at: https://guides.rubyonrails.org/active_record_basics.html [Accessed 30 April 2021].

Rails Guide. 2021. Getting started with Rails. Version 6.1.3. WWW document. Available at: https://guides.rubyonrails.org/getting_started.html [Accessed 15 March 2021].

Sarcevic, I. 2017. Faster Rails: Eliminating N+1 queries. WWW document. Available at: https://semaphoreci.com/blog/2017/08/09/faster-rails-eliminating-n-plus-one-queries.html [Accessed 30 April 2021].

Statista. 2019. Number of smartphone users worldwide from 2016 to 2021. WWW document. Available at: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/ [Accessed 11 February 2021].

Statistics Finland. 2021. Population. WWW document. Available at: https://www.stat.fi/tup/suoluk/suoluk_vaesto_en.html [Accessed 2 May 2021].

Stuti D. 2019. Pros and cons of using PostgreSQL for Application Development. WWW document. Available at: https://www.aalpha.net/blog/pros-and-cons-of-using-postgresql-for-application-development/ [Accessed 15 March 2021].

Techopedia. 2021. Digital Data. WWW document. Available at: https://www.techopedia.com/definition/24872/digital-data [Accessed 16 March 2021].

Thakur D. 2021. What is the difference between Data and Information? WWW document. Available at: https://ecomputernotes.com/fundamental/information-technology/what-do-you-mean-by-data-and-information [Accessed 16 March 2021].

The Albert Team. 2020. Data Collection Methods: What To Know for APR Statistics. WWW document. Available at: https://www.albert.io/blog/data-collection-methods-statistics/ [Accessed 1 May 2021].

The Glossary of Education Reform. 2015. Aggregate data. WWW document. Available at: https://www.edglossary.org/aggregate-data/ [Accessed 24 March 2021].

The University of Edinburgh. 2016. The three states of information. WWW document. Available at: https://www.ed.ac.uk/arts-humanities-soc-sci/about-us/information-security-and-governance/what-information-do-i-have-to-protect/the-three-states-of-information [Accessed 16 March 2021].

UNECE. 2009. Making Data Meaningful. Part 2: A guide to presenting statistics. PDF. Available at: https://unece.org/DAM/stats/documents/writing/MDM_Part2_English.pdf [Accessed 2 May 2021].

Viaesca. 2021. ViaEsca ryhmävalmennus. WWW document. Available at: https://viaesca.com/pages/viaesca-ryhmavalmennus [Accessed 1 March 2021].

Viaesca. 2021. ViaEscan tarina. WWW document. Available at: https://viaesca.com/pages/viaescan-tarina [Accessed 2 March 2021].