UNIVERZA V LJUBLJANI, FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO
TER FAKULTETA ZA MATEMATIKO IN FIZIKO

ANALIZA SISTEMA NUJNE MEDICINSKE POMOČI: ŠTUDIJA PRIMERA

DIPLOMSKO DELO NA INTERDISCIPLINARNEM UNIVERZITETNEM ŠTUDIJU

Andraž Žagar

Ljubljana, 2013

UNIVERSITY OF LJUBLJANA, FACULTY OF INFORMATION AND COMPUTER SCIENCE AND FACULTY OF MATHEMATICS AND PHYSICS

ANALYSIS OF EMERGENCY MEDICAL SERVICE SYSTEM: A CASE STUDY

INTERDISCIPLINARY UNIVERSITY STUDIES THESIS

Andraž Žagar,

Mentor: Assoc. Prof. Dr. Janez Demšar

Ljubljana, 2013

UNIVERZA V LJUBLJANI, FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO
TER FAKULTETA ZA MATEMATIKO IN FIZIKO

ANALIZA SISTEMA NUJNE MEDICINSKE POMOČI: ŠTUDIJA PRIMERA

DIPLOMSKO DELO NA INTERDISCIPLINARNEM UNIVERZITETNEM ŠTUDIJU

Andraž Žagar,

Mentor: izr. prof. dr. Janez Demšar

Ljubljana, 2013

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00050 / 2013
Datum: 14.5.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko ter Fakulteta za matematiko in fiziko izdajata naslednjo nalogo:

Kandidat: **ANDRAŽ ŽAGAR**

Naslov: **ANALIZA SISTEMA NUJNE MEDICINSKE POMOČI: ŠTUDIJA PRIMERA**
ANALYSIS OF EMERGENCY MEDICAL SERVICE SYSTEM: A CASE STUDY

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Sistem nujne medicinske pomoči na Japonskem v obstoječi obliki je bil zasnovan v šestdesetih letih, prvi centri pa so bili vzpostavljeni v letu 1977. V zadnjih letih se pojavlja vse več indikatorjev o njegovih pomanjkljivostih, ki se zrcali v nesprejemljivo dolgih časih od klica do sprejema v bolnice, pri čemer je ozko grlo predvsem iskanje proste bolnice za posamezen tip poškodbe oziroma bolezenskega stanja. Kot primer študije tovrstnih podatkov analizirajte težave, do katerih prihaja v prefekturi Nara. Za to uporabite programski paket Orange. Najprej razvijte dodatne grafične gradnike, potrebne za analizo tovrstnih podatkov. Nato jih uporabite na podatkih, ki so bili zbrani v tej prefekturi in opišite pridobljena spoznanja.

Mentor:

izr. prof. dr. Janez Demšar

Fakulteta za računalništvo in informatiko - dekan:

prof. dr. Nikolaj Zimic

Fakulteta za matematiko in fiziko - dekan:

prof. dr. Anton Ramšak

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani/-a ______________________________________,

z vpisno številko ______________________________________,

sem avtor/-ica diplomskega dela z naslovom:

____

____

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)
  ____
  in somentorstvom (naziv, ime in priimek)
  ____
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne ______________________ Podpis avtorja: ___________________________

Acknowledgements

This thesis would not have been possible without the support of many people. First and foremost, the author would like to extend his thanks to his mentor, prof. Dr. Janez Demšar, for giving valuable guidance and advice. Deepest gratitude also goes to co-mentor, prof. Dr. Noriaki Aoki, who provided great input from the medical field and guided the author to final results. Finally, an honorable mention goes to the authors’ family and friends for their understanding and support.

謝辞

今回の論文は、沢山の方々の支えなくしては不可能なものでした。まず第一に、著者はヤネス デムシャー教授の、メンター としてのとても価値あるご指導と助言にお礼申し上げます。そしてもう一人の指導者である、あおきのりあき教授の医学分野における素晴らしいご意見により、この論文を完成に導いてくださったことに、深い感謝を申し上げます。そして最後になりますが、著者がこの論文を完成させる上での、著者の家族と友人達によるご理解とご支援に感謝致します。

Table of contents

1. Povzetek

Sistem nujne medicinske pomoči na Japonskem v obstoječi obliki je bil zasnovan v šestdesetih letih, prvi centri pa so bili vzpostavljeni v letu 1977. V zadnjih letih se pojavlja vse več indikatorjev o njegovih pomanjkljivostih, ki se zrcali v nesprejemljivo dolgih časih od klica do sprejema v bolnico, pri čemer je ozko grlo predvsem iskanje proste bolnice za posamezen tip poškodbe oziroma bolezenskega stanja.

Kot primer študije tovrstnih podatkov smo analizirali težave, do katerih prihaja v prefekturi Nara. V namen analize podatkov smo v programskem orodju Orange najprej razvili gradnike za pred procesiranje podatkov, nato pa smo razvili še vrsto gradnikov za analizo ter prikaz časovno odvisnih podatkov ter podatkov, ki so temeljili na geografskih lokacijah. Nato smo izvedli analizo podatkov, pri kateri smo uporabili nove gradnike.

Raziskava je z novimi gradniki nazorno prikazala distribucijo pacientov v bolnišnicah in nam je omogočila, da smo določili glavne razloge za zavračanje pacientov v nekaterih izmed bolnišnic. Na splošno pa nam je omogočila, da smo naredili predloge glede razporejanja delovne sile po bolnišnicah ter izmeni, da bi lahko bolnišnice sprejele čim več primernih pacientov in s tem zmanjšale število zavrnjenih klicev.

Ključne besede: center za nujno medicinsko pomoč, vizualizacija podatkov, analiza podatkov

2. Abstract

The system of emergency medical services in Japan in its current form has been planned in the 60's, and the first centers started to operate in 1977. In the last years, we have observed some of its shortcomings, in particular the unacceptable times between the call and the admission to the hospital. The bottleneck is often the search for the suitable hospital for the particular type of patient.

As a case study, we have analyzed the problems that occur in the Nara prefecture. For this purpose we have created new widgets for data mining suite Orange. First, we developed preprocessing widgets. After that we focused on several geo-specific and time-specific widgets. We then used the newly developed widgets to perform data analysis.

Research with the new widgets clearly showed the distribution of patients in hospitals and enabled us to specify the main reasons for refusing the patients in some of the hospitals. Generally it enabled us to make suggestions on how to distribute work of the doctors for each hospital, so they would be able to serve as many patients as possible and by doing so reduce the number of refused calls.

Keywords: emergency medical service, data visualization, data analysis

3. Introduction

In Japan, the emergency medical services (EMS) are much different from EMS in Europe. The first emergency hospitals in Japan were designated in the 1960s, and by the year 1977 the first emergency medical centers were built. With these EMS centers, all medical facilities were reorganized into three levels. The aging of society presented a new problem, as the number of non-trauma patients steeply rose and some hospitals could not accept emergency patients [4]. This resulted in major problem with long ambulance transportation time, as the paramedics were not able to find a hospital that would accept the patient. In 2010 a branch of the Ministry of Internal Affairs and Communications recorded 727 cases of more than 10 hospitals refusing to take a patient. One specific emergency patient was also refused by more than 40 hospitals.

In this thesis we introduce several methods of data visualization and analysis in order to analyze data collected in Nara prefecture, Japan. The data describes the transportation of over 100.000 patients and the health state they were in when the EMS teams picked them up. For the means of presenting visualization and analysis ideas we will use Orange, the data mining and machine learning suite, developed by Faculty of Computer and Information Sciences, University of Ljubljana. In the first chapter we will first describe the problem that we have faced in this thesis, its origin and attempted solutions for this problem so far. This will be followed by the presentation of the source of the data and its nature. This will help us in further analysis, as understanding the data is mandatory for its research. After that, we will take a look at the data format, describing variables that were collected and a smaller set of variables that we derived from the original variables. The derived variables enhance the understanding of the data, which can be seen from multiple visualizations we created. We also created visualizations of primarily supplied variables, and compared the data with national/regional data while we tried to explain the unusual patterns in the distribution of the variable values.

In the following chapter, several new widgets will be presented, which were developed to simplify the process of extracting information from the collected data. Some widgets that existed before are also presented, however, they were modified or enhanced. First widgets presented were made to preprocess data and to prepare the data to be used in more advanced widgets. These widgets include the SQL Filter widget, which connects to the medical database and selects data filtering it with predefined filters; the Log transform widget, which performs logarithmic transformation on selected variables; and the Quick Select widget, which limits the supply of values for discrete variable by value selection via simple user interface. We will also take a look at the Pivot widget, which allows us to view aggregated data in a table-like form and enhanced version of the Sieve Diagram widget, where several new ways of optimizing the display of data were added.

The widgets introduced next provide basic visualizations of some of the data's combined variables, such as Q-Q Plot Widget, which makes a Q-Q graph of two selected values from the same discrete variable or a Q-Q graph of one value compared to all other values. In this chapter, the Compare Examples widget will also be introduced; it is used to make stacked bar charts of data, grouped by selected variables, and aggregate geographical data into a more compact form, which is used in more advanced widgets introduced in later chapters. Though these widgets are created for purpose of extracting information out of medical data, they can be used in solving any other problem that calls for visualization of data grouped by defined variables values.

After that, more advanced widgets that are specific to our problem domain will be introduced. They contain hard-coded variable names specific to our data and commands to query the medical database we use. These widgets can be split in two kinds, the geo-specific and time-specific widgets. We designed geo-specific widgets to display data with geographical representation, set by latitude and longitude of hospitals and emergency medical service centers (EMS) on map of the Nara prefecture, where all the municipalities' borders are marked to better distinguish the location of hospital or EMS. The geo-specific widgets are the Shadow Map Widget, which enables the user to find the epicenter location of patient origin; the Narpis widget, which with input of data from Compare Examples widget can display fan or bar charts on map; and the Narhodo widget, which helps determine the connections between hospitals and ERs, as it displays them on a map and connects them with lines that represent patient transfer from region to the hospital. The time specific widgets, on the other hand, focus on displaying data with help of temporal variables. This includes times of accidents, durations of transportations to patients and to hospitals, and times when patients were accepted to receiving hospitals. Widgets relying on time variables are the Time Comparison widget, the Timeline widget and the ER-FD graph. The Time Comparison widget helps the user study a distribution of a discrete variable over repeating time periods of, typically, a day or a week. The Timeline widget is similar to the Time Compare widget in the sense that it enables the user to view the data in time periods, but displays the data as events in time, which makes it easier for the user to display numeric values like time of transportation. The ER-FD graph widget is the last widget that will be presented and it is used to display connections between emergency rooms in hospitals and fire departments, based on the time the ambulance from the fire department uses on average to get the patient to certain hospital. As fire departments are closely related to regions, this graph also represents the region-hospital connections.

After introducing all the widgets, developed for use in the research done in this thesis, the data mining that was done under guidance of EMS expert will be described. We explored the data with the Orange data mining suite and used introduced widgets in different combinations on canvas schemas to get some interesting results, including some answers on why the patients are being refused by hospitals and why some ambulance transportations have prolonged patient delivery times to hospitals. Our goal was also to find any repeating pattern in data which would, in the future, help us make justified suggestions for patient transportation system changes to EMS personnel, which would ultimately lead to lower transportation times and lower chances of having multiple hospitals refuse the patient.

4. The data presentation and formation

4.1. The data source

The data used in this thesis was collected during a 2 month period, from 2010-7-1 to 2010-8-31 in Nara prefecture, Japan. The data describes all the ambulances that were dispatched from emergency response (ER) centers to sites with patients, and then delivered the patients to the hospital. It contains the health information about around 100,000 patients that emergency medical personnel collected before and during transportation, and all the calls to the hospitals they made, before the patient was finally accepted into one. The data was collected in order for the EMS specialists to be able to find the factors that affect hospital refusal rate and transportation time to accident site and to the hospital.

4.2. Variables

The data has several variables that tell us the whole cycle, from the received phone call, to the acceptance of the patient to a hospital, and the status of the patient on the scene, before he/she was transported. Table 1 represents the variables with a short description.

Table 1 – Data presentation

Variable Name	Variable Type	Variable description and values
Patient ID	String	ID of the patient used to identify the records for the same patient
Sex	Enumeration	男 (man) or 女 (woman)
Age	Integer	The age of the patient
Disease Group Name	Enumeration	呼吸器系 (respiratory system) 脳疾患 (brain disorder) 心疾患等 (cardio disease and similar) 泌尿器系 (urinary system) 消化器系 (digestive system) 精神系 (mental) 感覚系 (sensory system) 新生物 (neoplasm) 不明確 (indecisive)
Injury Name	Enumeration	ＡＣＳ疑い (ACS suspicion) ＨＥＬＬＰ症候群 (HELLP syndrome) ＴＩＡの疑い (suspicion of TIA) アトピー性皮膚炎 (atopic dermatitis) アナフィラキシー (anaphylaxis) アナフィラキシーショック (anaphylactic shock) ... Altogether there are more than 900 distinct values, so we will not cover the rest of them in this thesis. The sample was given only for the reader to better understand the data.
Severities	Enumeration	Severity of the injury 軽症 (mild injury) 中等症 (moderate injury) 重症 (serious injury) 死亡 (death)

		その他 (other)
JCS Name	Enumeration	The joint coordinate system. Possible values are: JCS 0 JCS 1 JCS 2 JCS 3 JCS 10 JCS 20 JCS 30 JCS 100 JCS 200 JCS 300 その他 (other) 調べず (not examined)
GCS_E_Name	Enumeration	Glasgow coma scale type 音声で (verbal) 自発的 (eye) 疼痛で (motor)
GCS_V_Name	Enumeration	Glasgow coma scale value 不適当 (unfit) 理解不能 (incomprehension) 見当識 (disorientation) 錯乱状態 (confusion) なし (none)
sBP	Integer	Systolic blood pressure
dBP	Integer	Diastolic blood pressure
HR_Details	Enumeration	Hart rate details 頻脈 (tachycardia) 正常 (normal) 徐脈 (bradycardia) 不整脈 (arrhythmia) その他 (other) 調べず (not examined)
HR_Times	Integer	Hart rate in beats/minute.
Resp_Details	Enumeration	Respiratory details 頻呼吸 (tachypnea) 正常 (normal) 徐呼吸 (slow breathing) 努力呼吸 (forced breathing) その他 (other) 調べず (not examined)
Resp_Times	Integer	The number of breaths per minute.
SPO2	Integer	Oxygen saturation
SPO2 wO2	Integer	Oxygen saturation measured by pulse oximerty
Medical History	Enumeration	Patients medical history
Dispatch Time	DateTime	Date and time of sending the dispatch to the scene
On Scene Time	DateTime	Date and time of dispatch arrival
Transport Started	DateTime	Date and time of departure from scene to

Time		the hospital
In Hosp Time	DateTime	Date and time of arrival to the accepting hospital
Call ID	Integer	Numerical representation of the call ID used to match the call to other data sets

Beside all the provided variables, we can generate a few more, which could be useful in the future data analysis:

Table 2 - Derived variables

Return trip time	Time (Dispatch Time – In Hosp Time)	The time from the call to the arrival to the hospital
Transportation time	Time (Transport Started Time – In Hosp Time)	The time the patient spent in ambulance
Without medical help time	Time (Dispatch Time – On Scene Time)	The amount of time in which the patient was without medical help

4.3. Basic statistics of data

To understand the data we are working with in this thesis, it is important to make a preliminary statistical analysis of it. This lets us know what kind of population we are dealing with and how the population differs from the representative population of that nation/region. This helps us discover the outliers that do not fit the pattern, which are the most important patients in our case, as those patients are in danger, if the outlying variable is transportation time from site to hospital. It also helps us gather information about regions and hospitals, which can help us understand why certain hospitals made the decision to reject a patient.

If we look at the distribution of patients by gender, as shown in Figure 1, we can notice that the gender distribution between patients is about equal. But, although there is about the same amount of patients of each gender, we can assume that the probability of needing emergency medical services is greater for males, as Japanese gender population ratio is just below 0.95 [1], which makes the viewpoint very different.

The age distribution of patients is, as expected, very different from the Japans general age distribution, but still holds some similarities. From Figure 2 we can see that the spike between ages 60 and 65 in patient distribution is actually not a major health problem, although it is the first major spike in the graph, as it matches the general population distribution. We can also see that the growing curve of patients after the age of 70 is not associated with general population and is a sign of a major increase in health problems after the age of 70, as the curves of population and patient distributions diverge.

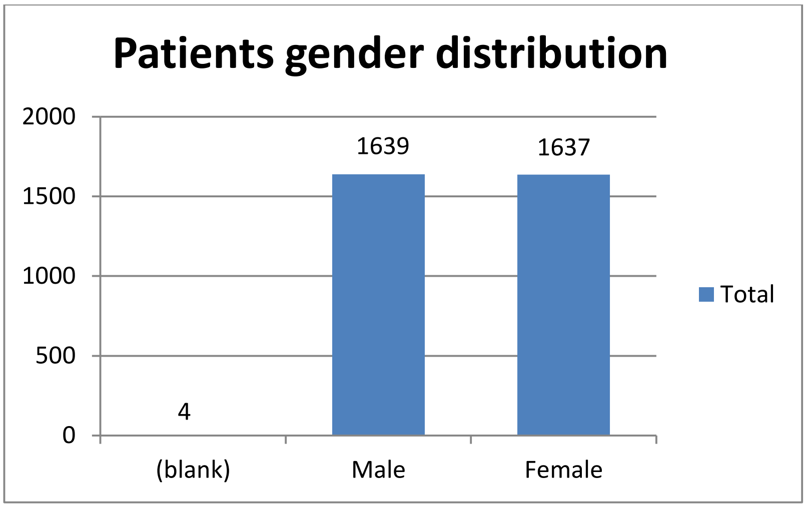


Figure 1 - Number of patents by gender

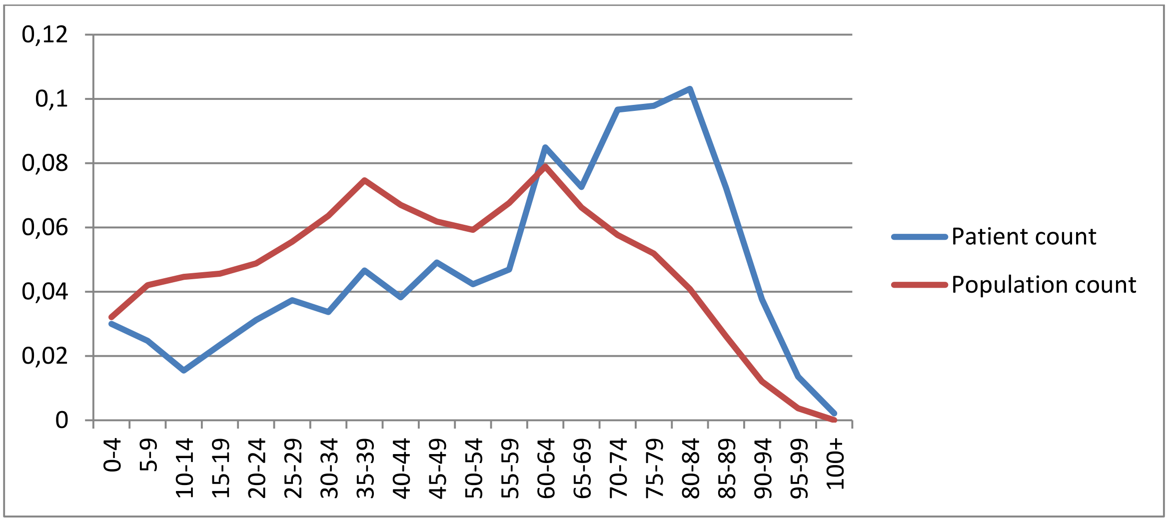


Figure 2 – Patient and population distribution graph

After introducing the basic description of population, we can perform analysis of disease groups. As we can see from Figure 3, the disease data does not look reliable. »Other« (»その他«) appears too many times, so the data is not of much use later on, as all the other groups, blank data excluding, together account for only about 15% of the data.

Now that we know the distribution of diseases, we can view the statistics for the severities of the diseases/injuries. Figure 4 shows us that most of patients were injured only lightly or moderately. This time there is only 1 patient whose severity was marked as »Other« (»その他«), which makes the integrity of the data better and the data itself usable in further research.

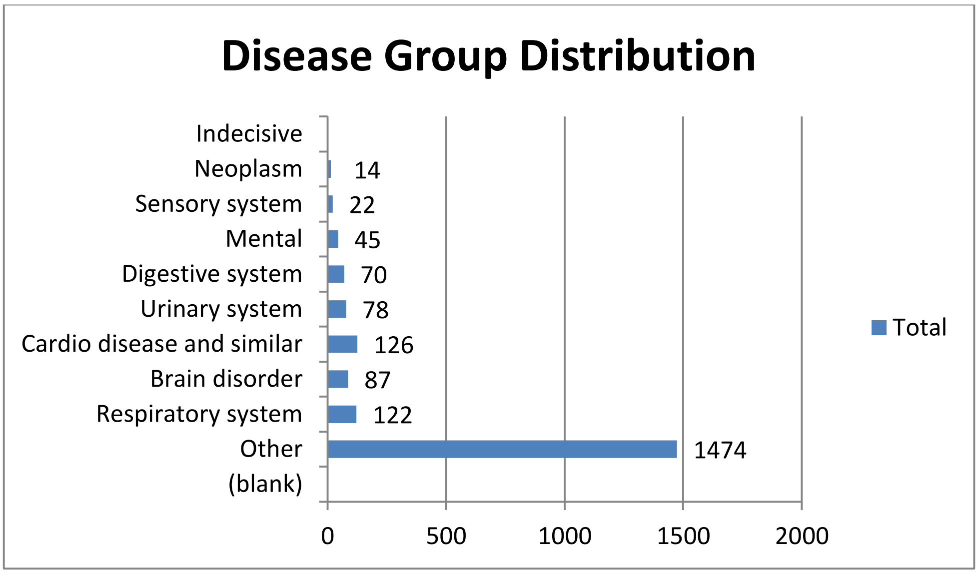


Figure 3 - Distribution of patietnts desease group

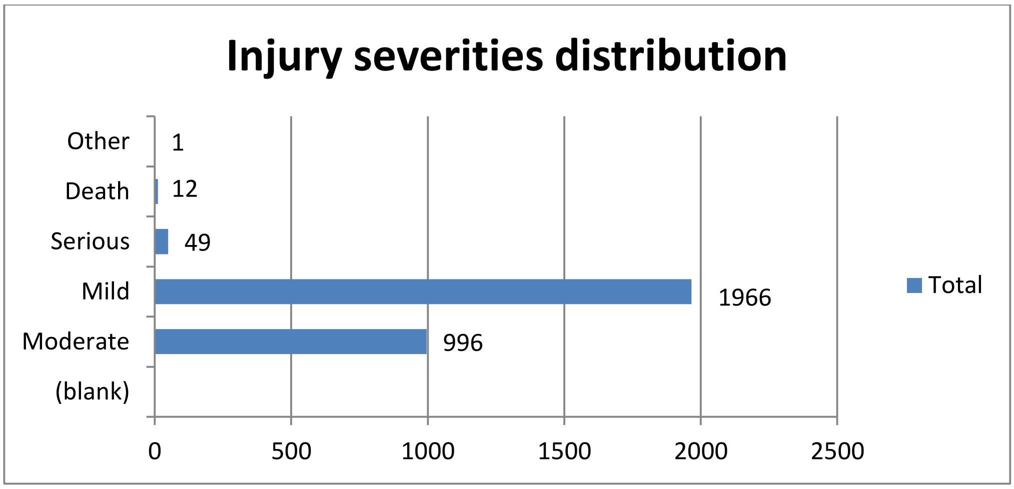


Figure 4 - Distribution of accident severities

The next interesting graph that we generated, is accident occurrence distribution. The result of analysis, shown in Figure 5, indicates that there are two peaks during the day, when the accidents are the most common. The two peaks are between 9:00 and 11:00 and again during 17:00 and 18:00. This seems to coincide with people arriving to and leaving from the workplace, as most of the companies have official working hours from 9:00 to 17:30. During the night, as expected, the number of accidents decreases.

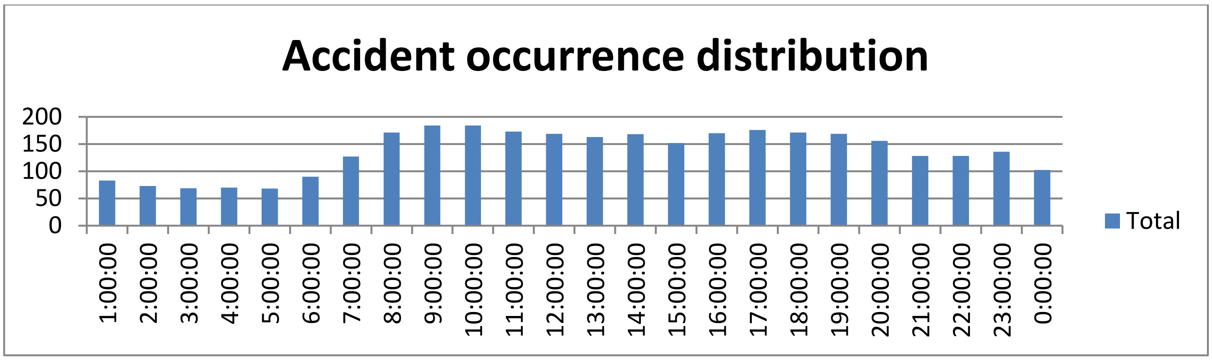


Figure 5 - Distribution of accident occurrences

Finally, we should also take a look at the distributions of calculated variables. In Figure 6 we can notice that distribution of times needed for the ambulance to travel from the Emergency Response Center to the accident site closely matches the Poisson distribution with λ of approximately 4. This enables us to create a simulation of ambulances picking up the patients with appropriate simulation software.

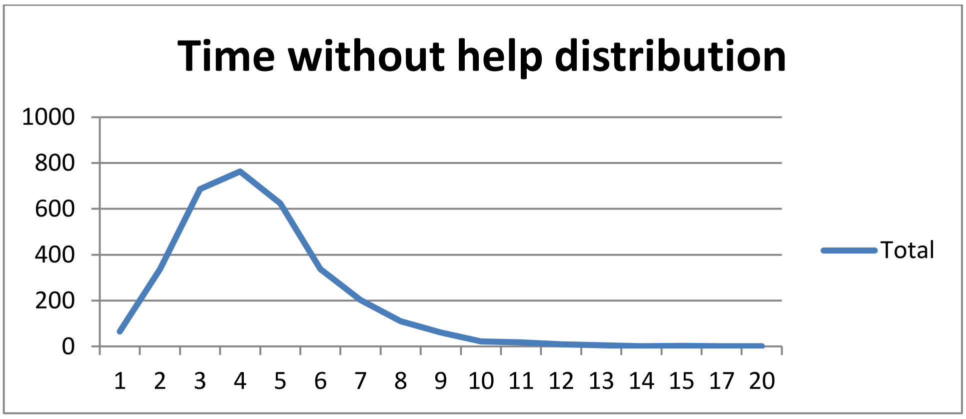


Figure 6 - Distribution of time, needed for the ambulance to arrive to the site

4.4. Basic data preprocessing

Because sometimes there are too few examples for us to work on specific cases, or the data has outlying elements, or for other reasons, we can employ data preprocessing to help us generalize or clean the data.

All the preprocessing was done using the data mining suite Orange.

Preprocessing the age variable

As age is a variable that has very few examples in some cases, it is best if we cluster it into age groups. Orange supports many forms of clustering, where the best suited for this case is the hierarchical clustering with the Ward's linkage. Hierarchical clustering joins elements into groups, so that the closer the two elements are, the bigger is the probability of them being in the same group. The Ward's linkage balances the cluster sizes and also considers the variance of data inside clusters. Feeding the EMS calls data to the Hierarchical Clustering widget and observing the age of patients gives approximately the following age groups:

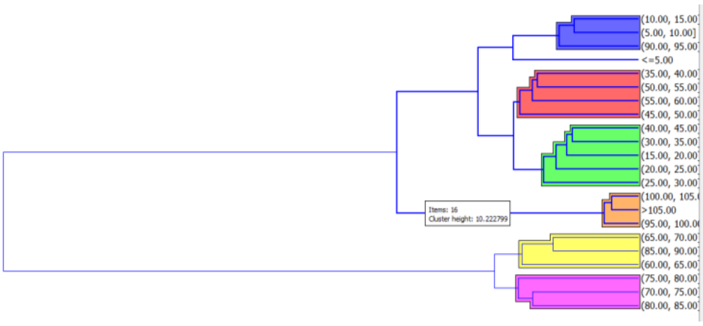


Figure 7 - Result of Hierarchical clustering widget

After cross-referencing the results with the data gained from several resources [2,3] and consulting with EMS specialists, we formed the following age groups that are assumed to appropriately describe the patient population:

- A (ages 0 - 9)
- B (ages 10 - 14)
- C (ages 15 - 39)
- D (ages 40 - 64)
- E (ages 65 - 84)
- F (ages 85+)

Preprocessing the location data

To better understand from what kind of area the patients come from, it is best to group cities into urban, suburban and rural groups. This way, the analysis will be simpler, since we will be able to relate the patients to other patients by using the types of the cities patients lived in. For the purpose of clustering cities, we use publicly available data which contains general information, such as *Population*, *Population density*, *Area*, *Number of medical centers*, and add few of more specific medical variables, like *Population within medical* area and similar. As we did not want to use duplicate variables, because they bias the distance, we remove the variables that are directly correlated. We leave the variables that are correlated but represent unrelated data. As an example, let us consider three variables, *Population*, *Area* and *Number of mesh*, which are well correlated. *Area* and *Number of mesh* actually represent the same quantity, area of the city in square meters or in count of mesh (with fixed value). If we use both, the distance (not a physical distance, but the one used in, for instance, cluster analysis) between the two cities in *Area* will have double effect on the distance between data instances, as the *Number of mesh* will have approximately the same effect as *Area*. Although the doubling effect is not desired in this case, it becomes desired when using *Population*. Although *Population* variable is correlated to *Area*, it has a different origin, as it does not represent the area, but the population count for that area. Mostly this variable will still be correlated to *Area*, but in some (important) cases, it will be able to determine which area is suburban and which rural, as suburban areas have a high value for the *Area* variable, while the value for *Population* is not as high.

At the end, we only chose the variables *Number of mesh lived on*, *Population*, *Population per livable mesh* and *Number of medical centers*. We used k-means clustering to perform the analysis of the location data, since we found out that it yields the best results. The cities are grouped in 3 categories, which are *rural*, *suburban* and *urban*. The centroids (typical examples) for each cluster are:

- Rural: Medium amount of *Number of mesh lived on*, very low *Population*, *Population per livable mesh* and *Number of medical centers*
- Suburban: Very high *Population* and *Number of mesh lived on*, low *Population per livable mesh* and high amount of *Number of medical centers*
- Urban: Medium *Population* and *Number of medical centers* but very high *Population per livable mesh* and very low *Number of mesh lived on*

The scatter plot displaying results can be seen in Figure 8, where the red dots represent suburban, green dots urban and blue dots rural cities.

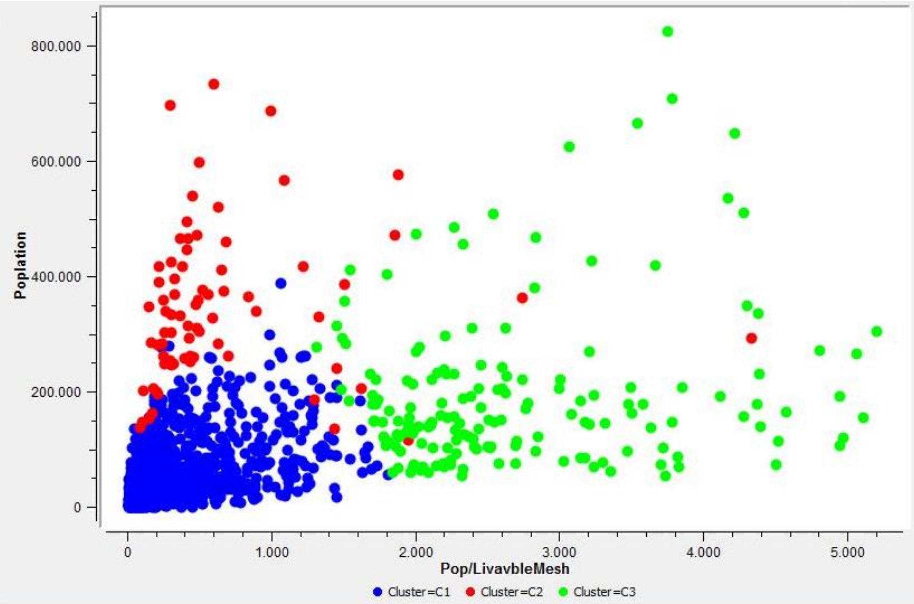


Figure 8 - Scatter plot displaying results of city clustering

Similar to the preprocessing done in the previous chapter, we have to preprocess emergency medical centers to determine in what category of medical care the patients were accepted.

5. Widget development

The standard widgets provided by Orange did not suffice for our study, so we created several new widgets.

Some widgets that are described in this section were developed by the mentor while we were working together at the CHORD-J institute in Nara, Japan; I, the author of the thesis, modified them in my later work when needed. As this thesis is as much about the development of new widgets as it is about using them to mine the data, we decided to include the descriptions of all widgets developed within the project, for the sake of completeness of the text, but with appropriately marked authorship.

For the purpose of this thesis, we divided the widgets into three categories, based on their purpose: preprocessing widgets, geo-specific widgets and time-specific widgets.

5.1. Preprocessing widgets

The first batch facilitates further preprocessing or creating simple visualizations of data. Although some of these widgets are able to help us with analysis (like Compare Examples widget), they are primarily intended for preparation of the data for other, more complex widgets. These widgets are the Quick Select widget, the SQL Filter widget, the Pivot widget, the Compare Examples widget and the Log Transform widget.

Quick Select widget

While working with data, we noticed the need to quickly filter discrete values by just selecting them instead of using a more complex standard widget. We[1] created a simpler widget. In the Quick Select widget (shown in Figure 9) the user can select desired values by clicking on them. User can select more than one value by clicking while holding the CTRL key (only clicked values will be selected), or by clicking on one entry and then holding SHIFT while clicking on the other value (all the intermediate values will be selected). The widget commits changes automatically, which enables quick review of changes in other widgets.

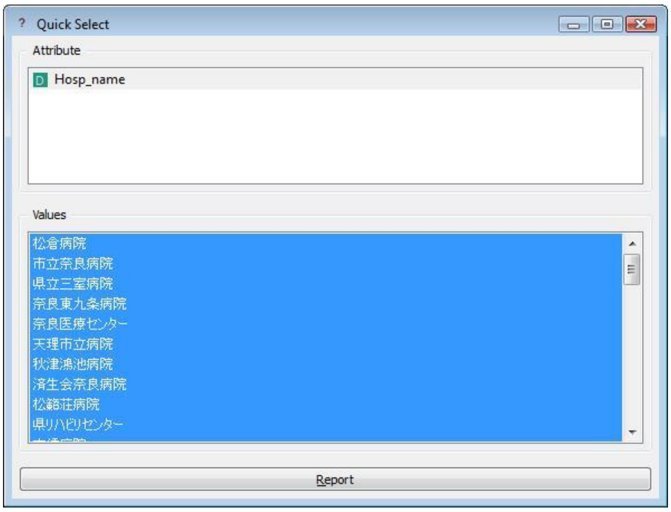


Figure 9 - Quick Select widget displaying selected hospitals

SQL Filter widget

Because all the data is stored in PostgreSQL database, it would be convenient to filter the data within the query, instead of loading all the data and then filtering it. To do this, we introduced a new widget called the SQL Filter, seen in Figure 10. This widget enables selection and filtering of data directly from our database in PostgreSQL.

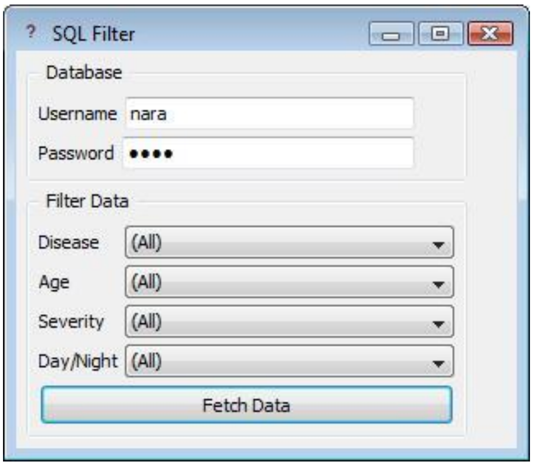


Figure 10 - SQL Filter widget

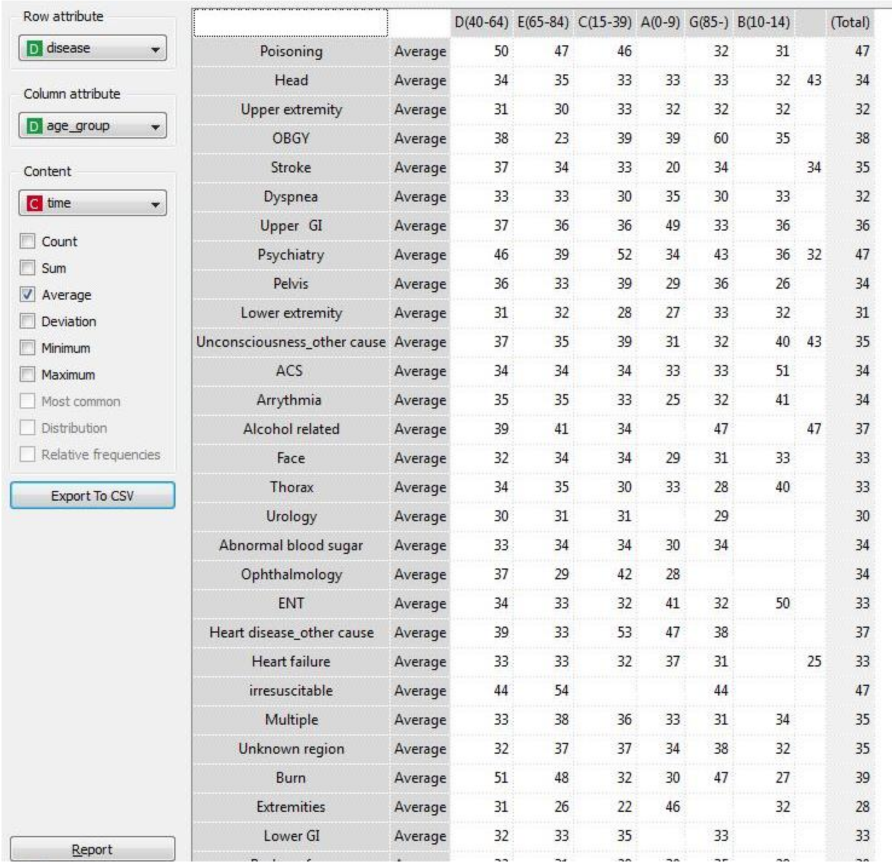
[1] This widget was developed by the mentor.

To connect to the database, the username and password of one of the users with the access to database on PostgreSQL server are needed. This filter is specialized to work only with the medical database we used in our research. User can filter by disease, age group, severity (severe/dead) and day/night calls. The possible filters are filled directly from the database upon the input of correct username and password from correlating columns. The user can send the data to the connected widget by clicking »Fetch Data« button.

The SQL Filter widget offers two types of output, first represents the data and the second represents the aggregated data, which can be used in widgets introduced later in this thesis. First, the classical output outputs selected variables from the database, containing the name of the hospital that accepted the patient, the different times needed for the transportation (when was the call made to the emergency center, when was the ambulance dispatched, ...), municipality where the patient was picked up, latitude and longitude of the accident and many others. The aggregated output contains count data of filtered values for regions and hospitals, and a combination of both. If the user wants, he/she can review the data in the Data Table widget, which comes in the default package of Orange.

Pivot widget

Another of the more beneficial preprocessing methods is data aggregation. With this method, we aggregate the data to show averages/ratios/sums for groups instead of having an entry for each specific case. To perform the aggregation of our data, we created a specialized widget for Orange called Pivot, which we extended to offer more functions. The original was created by the mentor, but later extended by the author of this thesis, in order to use the aggregated data that is computed directly on the database. This widget allows the input data to be aggregated by limited amount of general functions – count, sum, average, deviation, minimum, maximum, most common (median), distribution and relative frequencies – which the user can select by checking checkboxes on the widget.



		D(40-64)	E(65-84)	C(15-39)	A(0-9)	G(85-)	B(10-14)		(Total)
Poisoning	Average	50	47	46		32	31		47
Head	Average	34	35	33	33	33	32	43	34
Upper extremity	Average	31	30	33	32	32	32		32
OBGY	Average	38	23	39	39	60	35		38
Stroke	Average	37	34	33	20	34		34	35
Dyspnea	Average	33	33	30	35	30	33		32
Upper GI	Average	37	36	36	49	33	36		36
Psychiatry	Average	46	39	52	34	43	36	32	47
Pelvis	Average	36	33	39	29	36	26		34
Lower extremity	Average	31	32	28	27	33	32		31
Unconsciousness_other cause	Average	37	35	39	31	32	40	43	35
ACS	Average	34	34	34	33	33	51		34
Arrythmia	Average	35	35	33	25	32	41		34
Alcohol related	Average	39	41	34		47		47	37
Face	Average	32	34	34	29	31	33		33
Thorax	Average	34	35	30	33	28	40		33
Urology	Average	30	31	31		29			30
Abnormal blood sugar	Average	33	34	34	30	34			34
Ophthalmology	Average	37	29	42	28				34
ENT	Average	34	33	32	41	32	50		33
Heart disease_other cause	Average	39	33	53	47	38			37
Heart failure	Average	33	33	32	37	31		25	33
irresuscitable	Average	44	54			44			47
Multiple	Average	33	38	36	33	31	34		35
Unknown region	Average	32	37	37	34	38	32		35
Burn	Average	51	48	32	30	47	27		39
Extremities	Average	31	26	22	46		32		28
Lower GI	Average	32	33	35		33			33

Figure 11 - Pivot widget displaying data aggregations of transportation time by diseases and age group

As we can see in Figure 11, the Pivot widget offers the user to select variable representing rows and columns and the variable used for aggregation calculations. We designed the widget to support more than one aggregation, which can be seen in Figure 12.

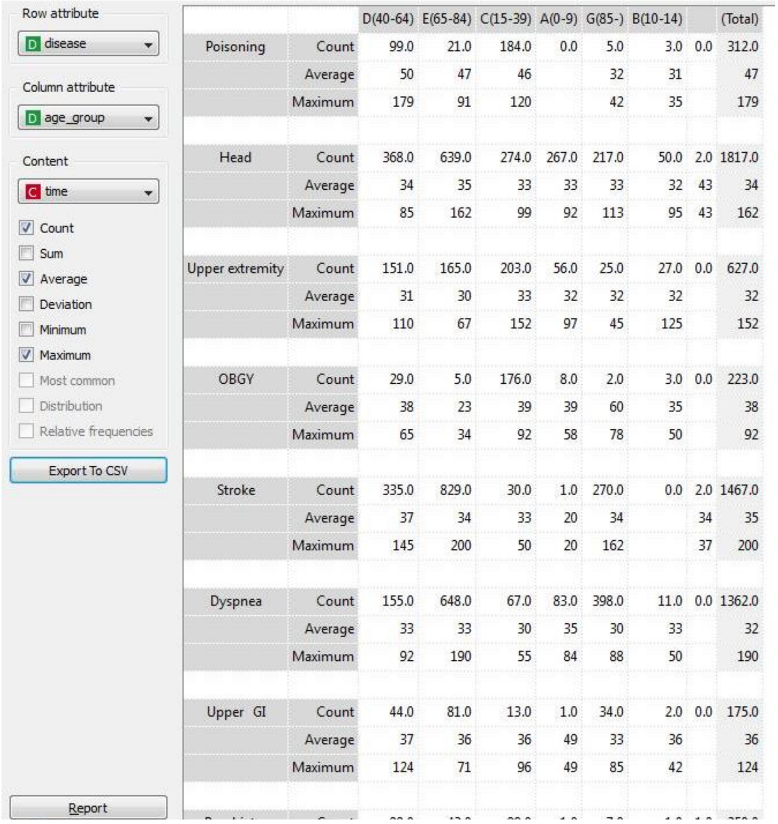


Figure 12 - Pivot widget displaying multiple aggregations

The Pivot widget is also capable of sending aggregated data to other widgets, or exporting the aggregated data to a .csv (comma separated values) file. Because there were some aggregations that appeared much more frequently, we hardcoded the aggregation directly into SQL Filter widget, so the user is able to connect to the database and use default aggregation without having to set the intermediate (Pivot) widget. This is demonstrated later, in the chapter »Narpis widget«.

Compare Examples widget

To help us visualize the aggregated data, we created the widget called Compare Examples. This widget is used to compare aggregated data side by side, to better evaluate the difference between different groups. In the example seen in Figure 13, we have compared age structure of patients, accepted to hospitals, by hospitals that accepted them. As some hospitals have accepted very few patients, we have arranged the hospitals by the number of patients they accepted, so the more representable samples are located on the left side.

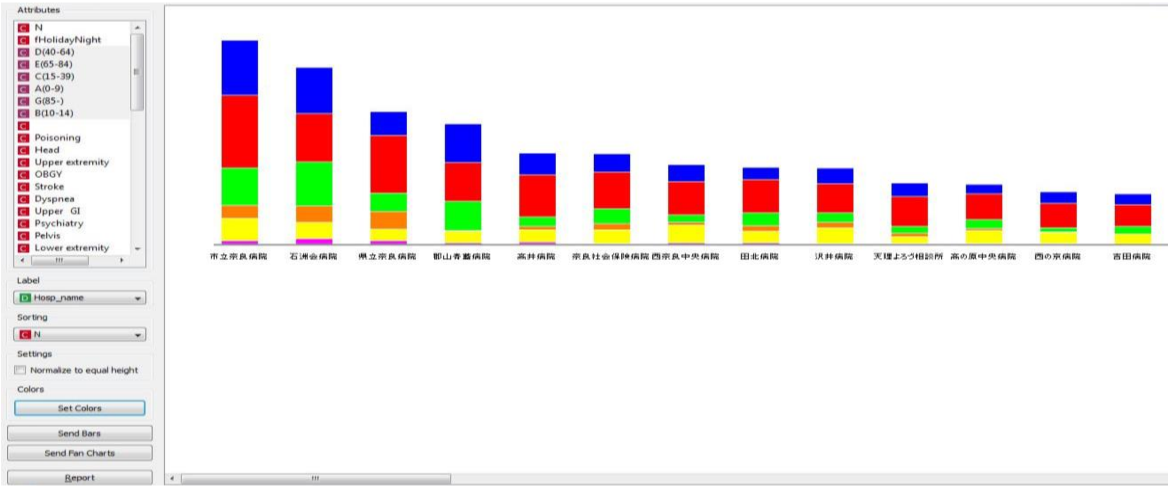


Figure 13 - Compare Examples widget comparing age structure of patients by hospitals that accepted them

We can also use »Normalize to equal height« check box to set all bars to be of equal height. This allows us to compare different distributions of groups more easily. We can take Figure 14 as a reference. It displays the same bars as Figure 13, but instead of using total amount of patients accepted in the hospital as a fact determining bar height, all the bars are of equal height. This way we can compare age structure between hospitals more carefully. What we have to be careful about in this case is the missing information of actual patients.

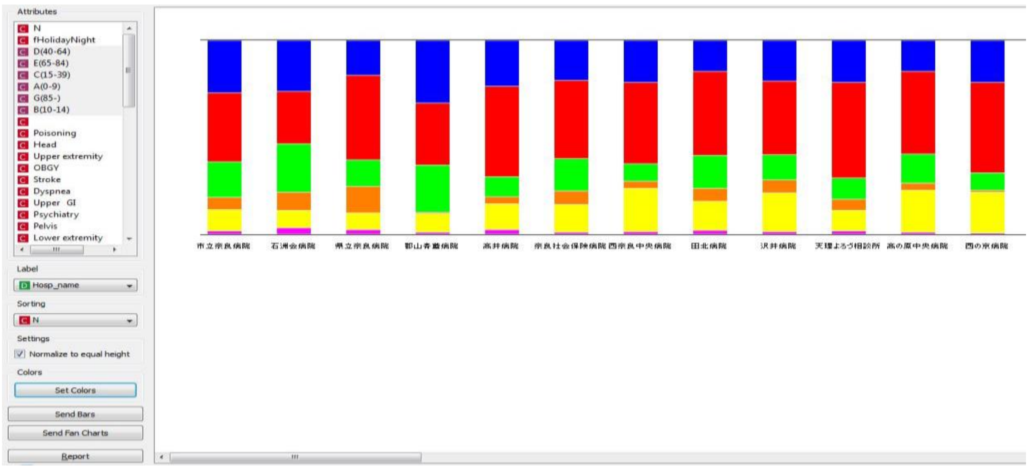


Figure 14 - Compare Examples widget with normalized heights of bars

Log Transform widget

Some of the data is scaled to have only a few big values, while the rest of the values are nicely distributed, but much smaller. This makes visualizations unreadable as the isolated maximum values scale the visualization to hide the average value. To correct that, Dr. Demšar constructed a widget that performs a log transform operation on the selected variables, making their values more acceptable. Example of the widgets use is shown in Figure 15, where the widget will perform a log transform of Pop/mesh variable without any skewness or kurtosis.

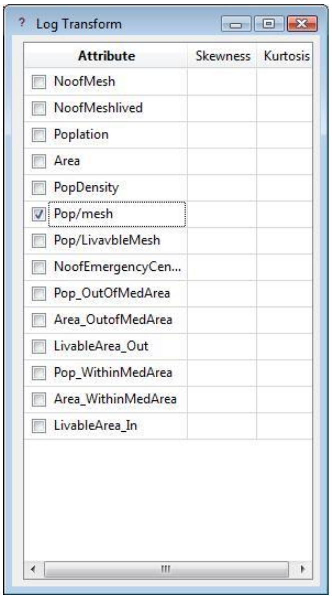


Figure 15 - Log Transform widget prepared to perform a log transform on Pop/mesh variable

Q-Q Plot widget

Q-Q graph is one of the basic visual representations of data statistics, where we compare distribution of values, divided into two groups. While having one group on x-axis and the other on y-axis, we draw points on each of the three quartiles. But, as it is hard to represent data with only 3 points, it is better, if the Q-Q graph is represented with point for every percentile. To better understand the difference in the distribution of data, we need to implement a widget with Q-Q graph generation support.

The Q-Q Plot widget receives data with at least one discrete and one numeric variable and allows the user to create Q-Q graph. To create a graph, user must first select numeric variable that represents the values needed to generate the graph. Then the user selects the compared attribute, which is the variable, whose values will be compared. Then the user selects values to display on the graph from the list. These values will be represented on x-axis. The user then selects second value, the value which will represent y-axis, as can be seen in Figure 16. Alternatively, the user can check the »Compare Value Against All« check box to display y-axis based on all other values (values that are not covered by selected values). The user also has option to change the number of displayed points or to draw lines instead of points. To improve the user experience, we provided an option to select the number of drawn points (3 equals quartile, 99 equals percentile ...), or even interpolate a line through the possible point locations. Additional added options are the grid display, to help determine the distance between points; the option to change grid density, to determine the distance between closer or farther points; and the option to change the point size to accommodate for dense or sparse point density.

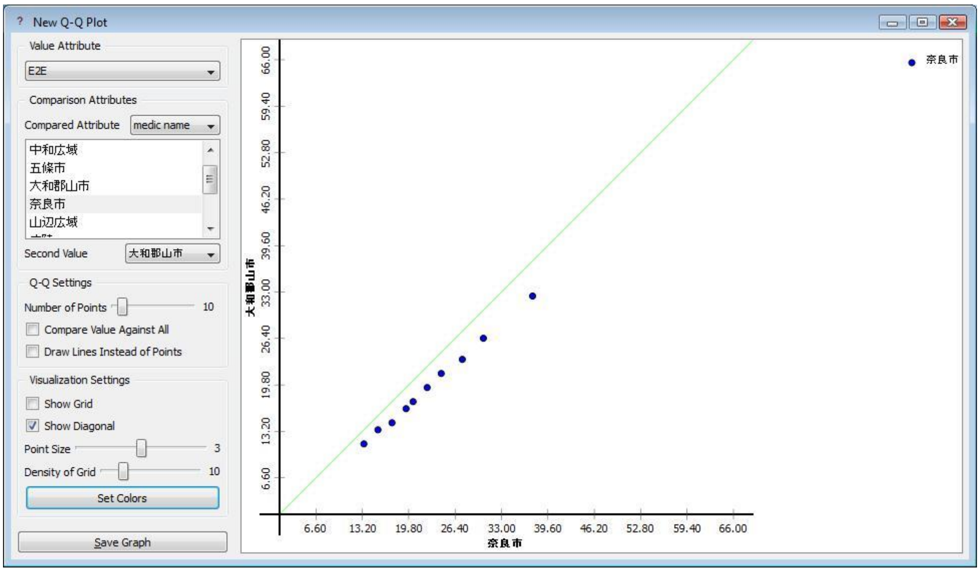


Figure 16 - Q-Q Plot widget displaying a Q-Q plot of E2E times

5.2. Geo-specific widgets

As preprocessing and simple visualizations did not satisfy our presentational needs, we created another batch of widgets, with the purpose of displaying different variable subset on the maps, called geo-specific widgets. This group contains the Shade Map, Narpis and Narhodo widgets.

Shade Map widget

As we wanted to know from where the patient population comes and how it is distributed, we created a widget called the Shade Map Widget. We use the widget to display simple marks on the locations where the patients come from, on the map of the Nara prefecture. If two marks overlap, the overlapping area gets brighter, so the brighter the area is, the more patients came from the vicinity of that area.

To simplify the loading and filtering of the data, we added the option to load data via SQL, where the filters are predefined to Disease, Age, Severity and Day/Night filters with the selection options loaded from database. For the user to use the SQL connectivity option, the database username and password have to be provided in appropriate text boxes. To show data on the map, we first have to load the map data. We do that by clicking on the »Fetch Data« button. The user will not see any data until the maps are properly loaded. Map data needs to be in the MapInfo tabular form and with file extension .mif, which is one of the standard formats used in geographical information systems. MIF format was developed by company MapInfo, and is structured in the following way:

- The head with the data about the map presented in the file
- a DATA tag, followed by tag Region
- the number of points representing that region
- the region polygon points in clockwise order, one per line
- the Pen, Brush and Center of the region is written in separate lines
- the next region tag.

In Figure 17 we can see that most patients came from the Nara municipality (the top-most cluster of patients), which is the main city of the Nara prefecture. The other municipalities, which contributed patient data for this research, are Yamatokoriyama (cluster just below Nara), Gojo (lower left cluster) and Shimoichi (cluster right of Gojo municipality).

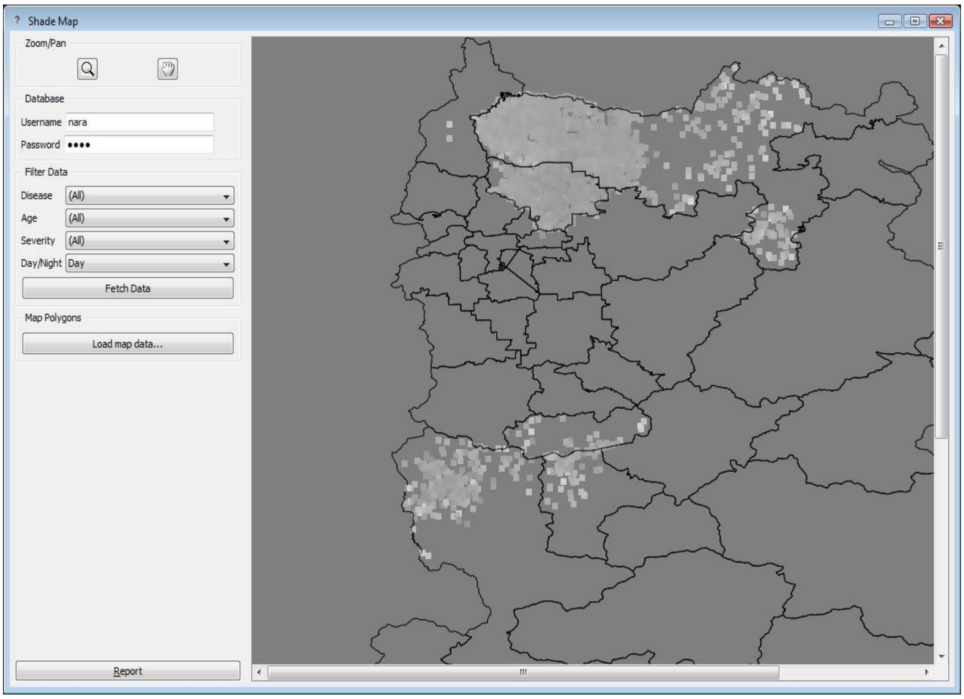


Figure 17 - Shadow Map Widget displaying patient data

We can get visualization similar to this widget by loading data to scatter plot widget, setting x-axis to display longitude, y-axis to display latitude and by setting the transparency of elements to about 5%. The results of this process are displayed in Figure 18. Although the visualization is similar, we still prefer the Shadow Map widget, as it has direct connection to database and makes loading and filtering data easier. Also, the Shadow Map widget uses maps to help users determine the exact location more easily, while within the scatter plot it is hard to determine the municipality of the patient.

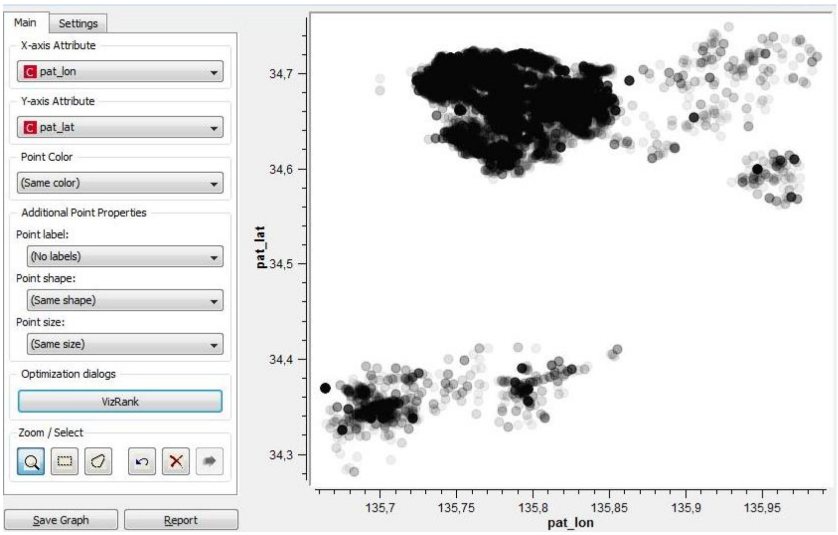


Figure 18 - Scatter plot resembling results from Shade Map widget

Narpis widget

As we were provided with coordinates of hospitals, we want to be able to display the charts and other results of data aggregation on the map. For this purpose we created a new widget called Narpis. This widget is used for visualization of aggregated data on the map, where the data has to be grouped by locations (as the graphs are shown only on one location for each entry). To do this, we altered the Compare Examples widget to be able to send chart and coordinate data to the connected widget. We added the »Send Bars« and »Send Fan Charts« buttons for sending the graph data to the connected widget. Fan chart describes a chart similar to the pie chart, but instead of using the full circle, it only uses the upper half of a circle to represent distributions, and instead of the lower half of the circle, there is a label representing hospital/ER.

The Narpis widget is able to display the charts from Compare Examples widget on the map, provided that the user has loaded the map data by clicking »Load map data...« and then selected the appropriate file with .mif extension. After the map is loaded, the data is displayed as shown in Figure 19.

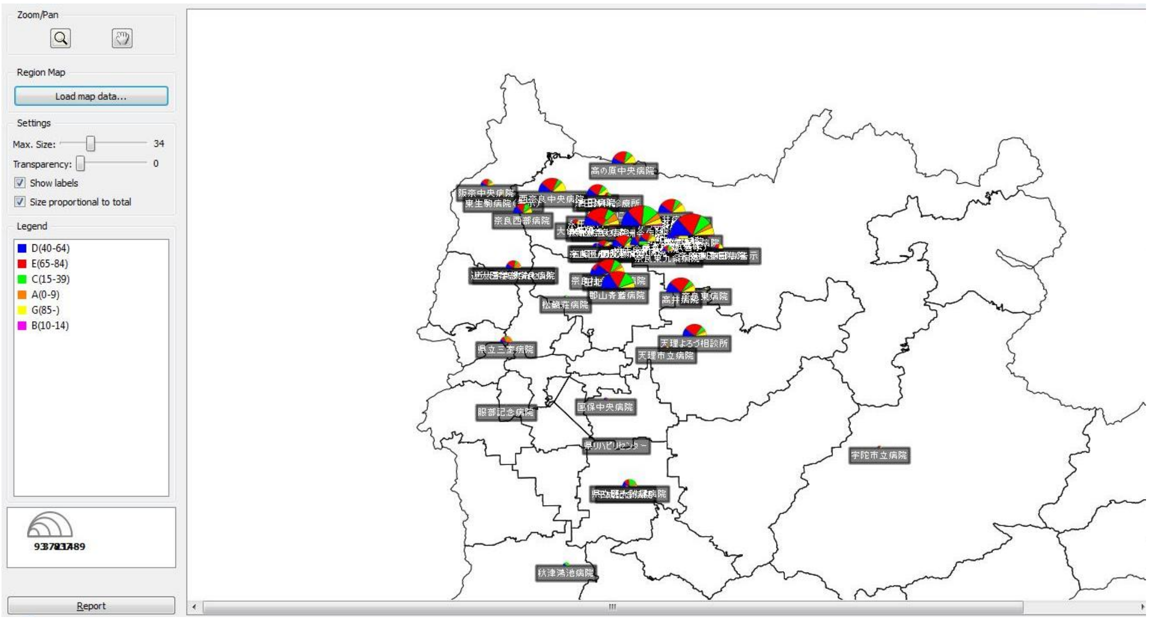


Figure 19 - Narpis widget displaying aggregated hospital data

The charts can then be resized, and their opacity can be changed to better notice smaller, hidden hospitals. As the size in this case represents the total amount of patients accepted, the more patients the hospital accepts, the bigger the graph representing the hospital will be. If resizing and changing opacity is not enough to see smaller (but also less significant) hospitals, we can also hide labels and remove the proportional sizing. The result of the latter two options is shown in Figure 20.

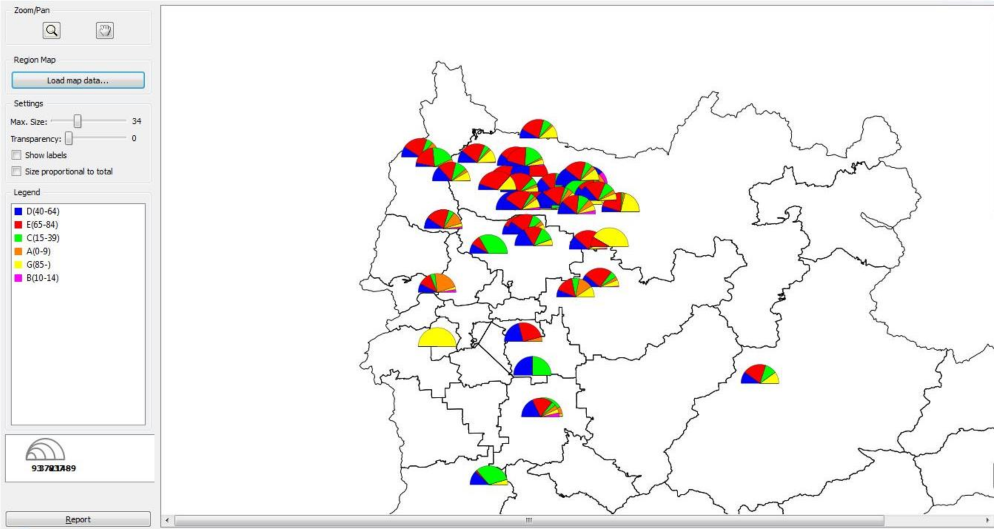


Figure 20 - Narpis displaying data without labels and with constant size

Since sometimes all this is still not enough, we added zooming. We can perform zoom operation by left-clicking on the map and then dragging the mouse pointer until displayed rectangle covers the area the user wants to zoom into. The zoom enables the user to see the graphs more distinctly, as we can note from Figure 21. We can also notice that two hospitals in the left-central part of the image share the same location. After examining data, we found out that these were two actual hospitals, operating in same building.

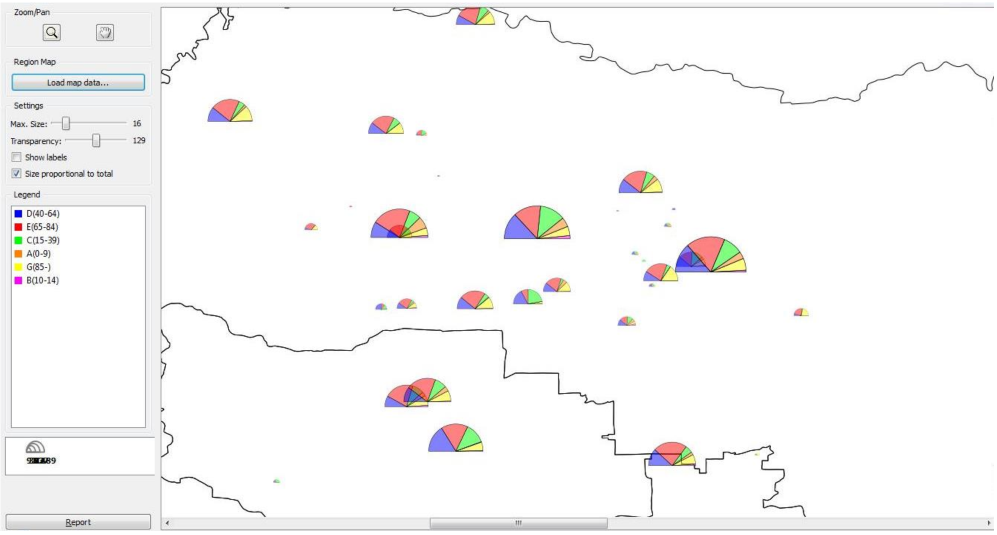


Figure 21 - Narpis widget zoomed in and using transparency

Narhodo widget

To go one step beyond the Narpis widget, we decided to add the functionality of the SQL Filter widget, described in the chapter Basic data preprocessing. Moreover, after consultation with EMS experts, it seemed that the Narpis widget is missing data on regions. The user could see the data separately, by inputting different aggregations, but then the hospital data would not be shown or distinguished from the region data. To accommodate the new ideas, we created a new widget, called Narhodo (Nara Region and HOspital Data Observer). As seen from Figure 22, the Narhodo widget requires a username and password for the PostgreSQL database like the SQL Filter widget, and has to have map loaded like Narpis. Another major difference from the Narpis widget is that Narhodo widget does not allow the user to choose values displayed in fan charts. The values always show the general acceptance rate of the hospital.

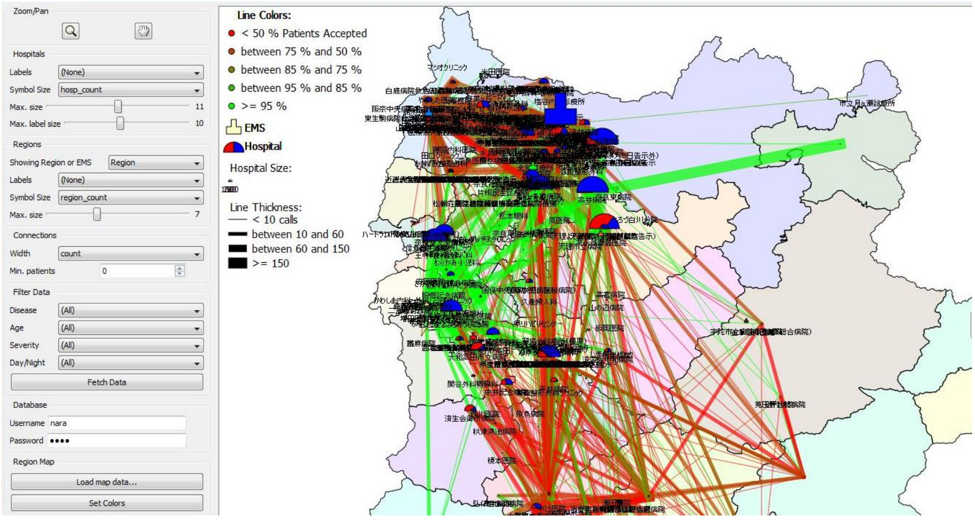


Figure 22 - Narhodo widget displaying connections between hospitals and regions

Unfortunately, without filtering the data, graph generated by the Narhodo widget is unreadable. We added transparency to connections lines and allowed line filtering by the number of patients, as lines representing only a few patients are not statistically significant and represent isolated examples. As some hospitals or even regions are not connected to any other hospital or region, we removed them, since we are not interested in their statistics. We also made the line opacity change, when the mouse indicator is hovered over the region or hospital connected to the line. This way, we enabled the user to focus on one hospital or region at a time. The result of these improvements can be seen in Figure 23.

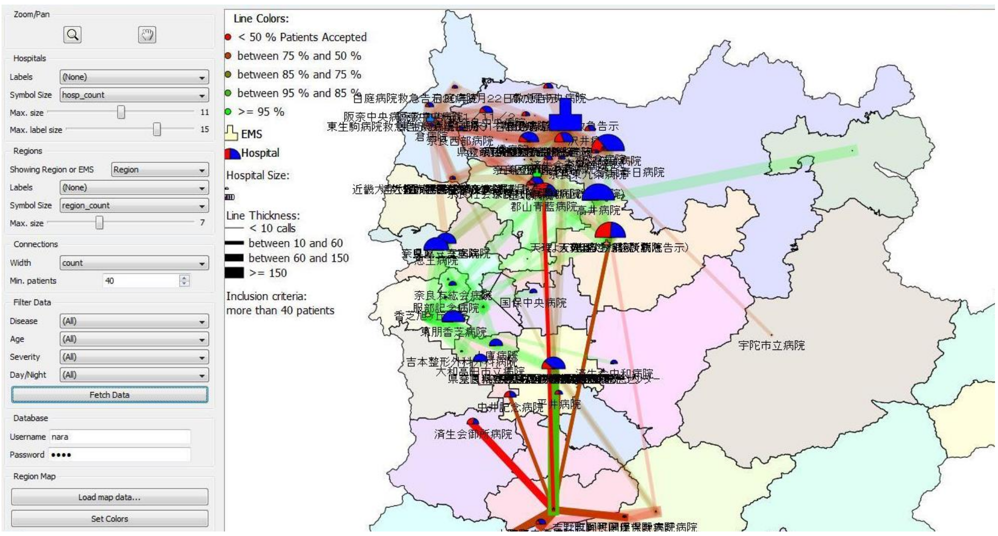


Figure 23 - Narhodo displaying filtered data and half-transparent lines

As the visibility of data is still not good enough, we can also add the features from Narpis, such as scalability of hospital and region sizes, scalability of hospital text sizes and the zoom option. With the added features we can use Narhodo to its fullest potential and view the connections between regions and hospitals, along with hospital acceptance statistics.

5.3. Time-specific widgets

As patient delivery times are still hard to compare in geographical widgets like Narpis or Narhodo, we decided to make a new set of widgets, dedicated to the data visualization through time variables. The widgets that belong in this category are the Time Compare widget, the Timeline widget and the ER-FD Graph widget.

Time Compare widget

First of the so-called time-specific widgets is the Time Compare widget, which was created to view the data aggregated by time intervals and separated into bins. This widget is used to view repetitive data with known repeat cycle and to bin the data into grouped chunks for a more robust view of the data. It also contains option to repeat the graph multiple times, since it is easier to find patterns this way.

Figure 24 shows how the widget looks like by displaying the generated graph of timed comparison of number of patients, received from two bigger pick-up regions. As we can see, the user can manually select desired variable and its values for comparison. Also, the user has the power to choose number of bins and length of cycle, which is based on the basic time value of the variable selected under Time Attribute. User can also set the number of repeats of the graph and change the color set, if he/she wants to do so. We also enabled the user to see detailed information of the specific bars by hovering the mouse pointer over them.

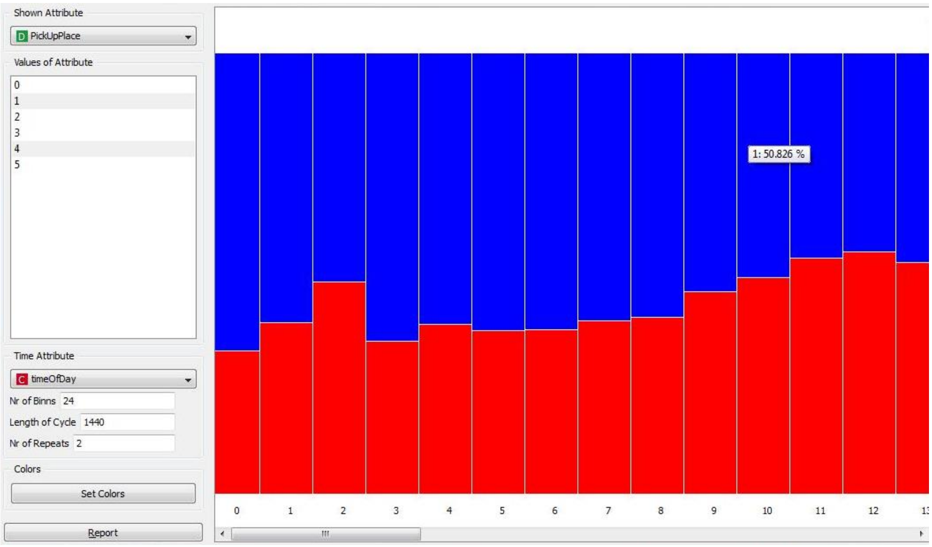


Figure 24 - Time Compare widget displaying time comparation of two more common areas of patient pick-up

Timeline widget

As it is very hard to represent numerical data like transportation times or age on Time Compare widget, the mentor created another widget, which is able to display non-aggregated data on timeline. The Timeline widget allows us to see patterns that repeat themselves on daily, weekly or monthly basis, by displaying symbols (for discrete variables) or lines (for numerical variables) on the timeline. We can normalize the numerical values, so the differences are more obvious, but then we have to be careful to check whether the results are really significant. We can also set a variable to represent the label written under the symbols/lines, but only if the data is not too dense. Another option that we added to the widget is the ability to group data by any discrete variable. In Figure 25 we can notice that the grouping is done by the PickUpPlace variable. Selection of the time scale is also available and is limited to Year, Months, Week, Day, Hour, Entire time range and a custom value, which should have value of a time span represented in ticks. This scale determines how much data will be seen on users screen at the same time – the bigger the time scale, the denser the data will be. To enable the user experience, similar to the previous widget, Time Compare, we also added a check box, allowing the users to aggregate data within the time scale.

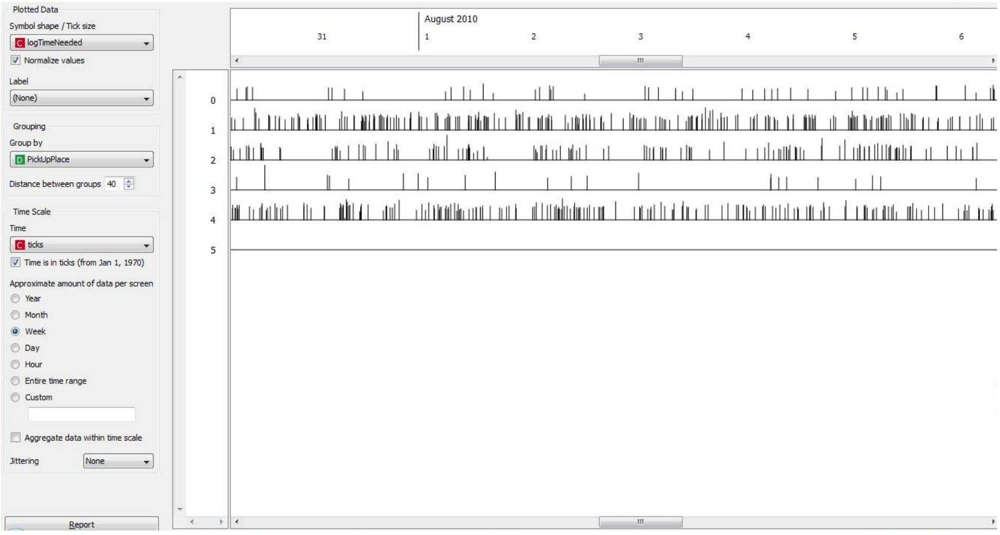


Figure 25 - Timeline widget displaying log of patient delivery time for each of the major patient pick up areas

ER-FD Graph widget

Before we conclude the widget introduction, there is one more important widget that we have to introduce. Although Narpis and Narhodo widgets visualize geographical data quite nicely, the problem with them is that they cannot visualize times except by using colors, which is not very efficient. To contrast the connection-based widgets like Narpis and Narhodo, we introduce the ER-FD Graph (Emergency Room – Fire Department Graph). The ER-FD Graph widget uses similar methods of visualization as the geographical widgets, but instead of projecting the data on the map, it uses time as a distance measure. So - the faster the hospital receives patients from a certain region, the closer they will be on the graph. As it is impossible to precisely represent more than two regions in this graph, it is suggested that the user uses this widget in combination with Quick Select widget, which was introduced earlier.

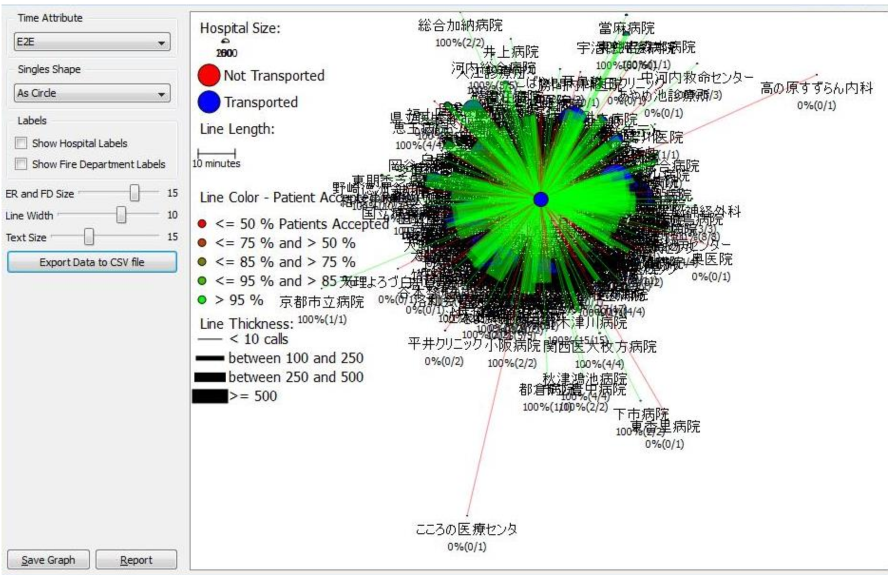


Figure 26 - ER-FD Graph widget displaying connections between hospitals and Nara region

Figure 26 shows an example of ER-FD widget use. We can notice that the graph the widget generated is very confusing, as there is no specific order of hospitals. To make the visualization clearer, we first added the option to display only labels for hospitals. This way we removed the text representing the excess statistical data and displayed the data in background color of a bounding rectangle. We also added the zoom option, which enables the user to zoom in on multiple overlapping labels, to distinct what is written on them. But this was still not enough to be able to work with the data. Our next step in visualization enhancement was to add the Singles Shape option. This option enables user to change the general shape of the graph (circle by default) when displaying the data of only one fire department/city/region. The example of using Spiral option can be seen in Figure 27. The difference between the first and the second generated graph is more than obvious. On the second graph, the user can easily notice the hospital that is connected to the region by the longest time and the one that is connected by the shortest time. Sincewe know their locations, it is easy to compare time distance and geographical distance.

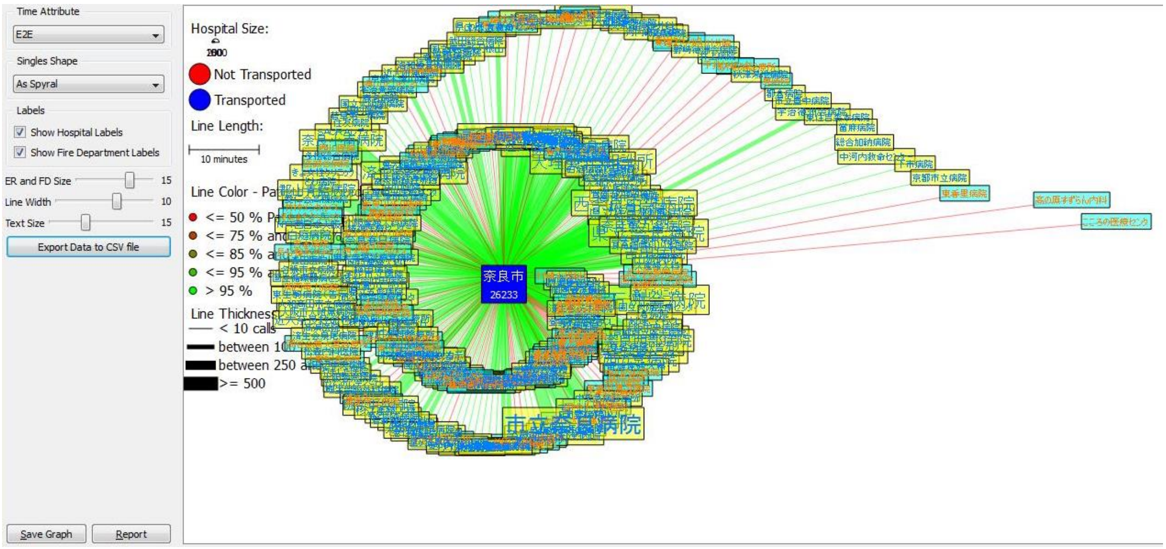


Figure 27 - ER-FD Graph widget displaying hospital labels and has the hospitals ordered in a spyral

6. Data analysis

Data analysis was based on the common widgets in Orange and the new ones described in the previous chapter .With analysis we tried to help improving the emergency medical services by answering some crucial questions, like

- why the transportation of some specific kinds of patients took much longer than average,
- which hospitals accepted more patients than others,
- what were the specific patterns with regard to hospitals that accepted specific types of patients.

The common analytic steps include:

- using the Q-Q plot for comparison of transportation times and finding any outlying fire departments or hospitals,
- using Shade Map widget to find explanations for the outlying fire departments,
- we use Compare Examples widget to compare age and disease distributions
- we use Narhodo widget to display connections between hospitals and EMS

6.1. Preliminary analysis

For the preliminary analysis, we check the hospitals and fire departments for any outliers in traveling time. The easiest method to do this is with Q-Q Plot widget, which enables us to compare fire departments and hospitals independent of how many patients they transported/received. On the other side, we have to remove hospitals and fire departments with too few patients, as the graphs provided by Q-Q graph are not appropriate for them. Files used in this analysis were generated by running SQL queries through *SQL* widget and by saving the results into a file with the *Save* widget. This was done so the scheme can be loaded later without accessing the database. The scheme can be found on attached disk in folder *"Thesis Files\Schemas\Preliminary Analysis"* in a file named *Preliminary Analysis 1 – Q-Q Plot.ows*. To learn how to load the schema to Orange, visit Orange help pages at http://orange.biolab.si/

After we create a schema, as displayed in Figure 28, or load it from file, we load an aggregated file, which contains information about the average transportation times between hospitals and regions in the *File* widget and select all but empty hospitals in the *Quick Select* widget. This removes all patients that were not accepted to any hospital. In the *Select Data* widget, we remove all connections that have less than 500 patients. This way, the Q-Q plot displays only relevant graphs, ignoring the smaller (and thus statistically unstable) hospitals.

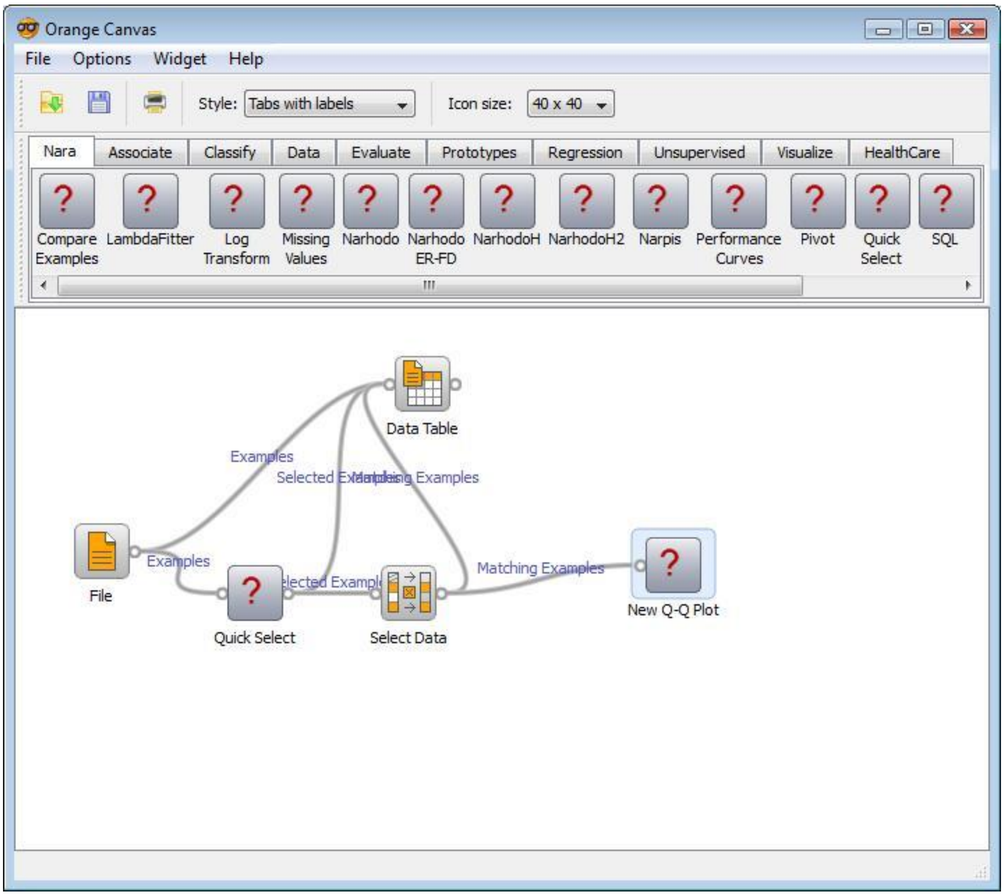


Figure 28 – Schema for use of Q-Q Plot widget

When we check all the hospitals, we can notice that there is one hospital unlike others. From the Q-Q graph, we can read that some hospitals, like 恵王病院 (Keiou hospital) have much faster average transportation times. This is shown in Figure 29, where we can read out that the time from departure and to arrival to the hospital is much smaller when compared to other hospitals.

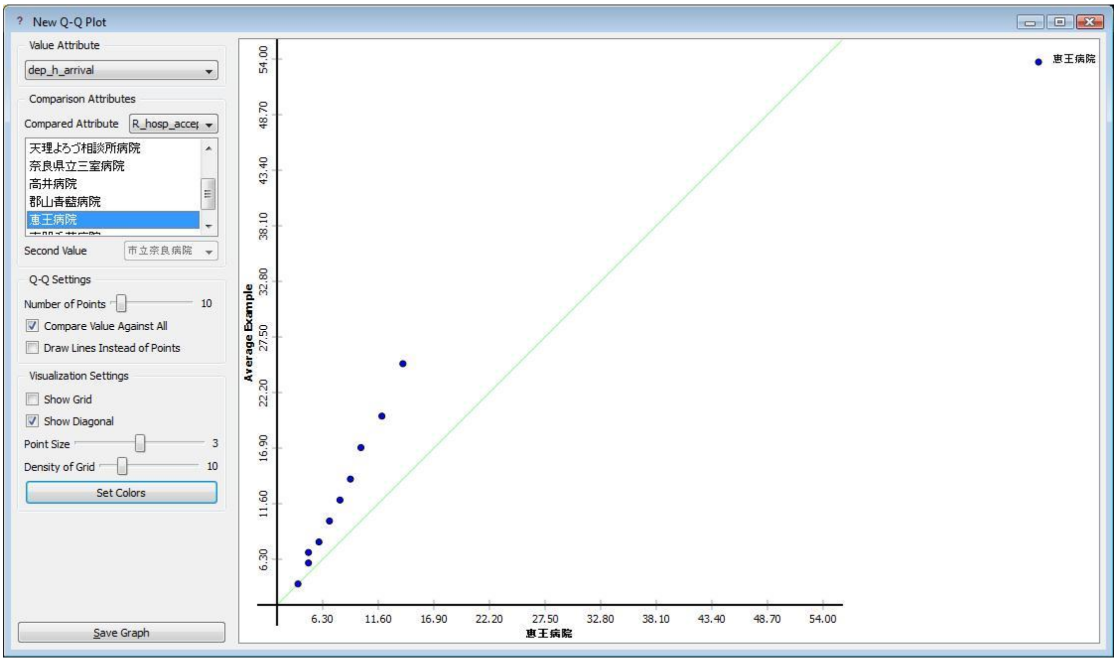


Figure 29 - Q-Q Plot displaying Keiou hospital in comparison to other hospitals

To pursue the discovery, we create a scatter plot with all hospitals, where x-axis represents the average age of the patients the hospital accepted, y-axis represents the average time the paramedics needed from the scene to the hospital and the size of the points represent the size of hospitals, measured with the number of accepted patients. We are able to create this by loading schema, saved in file *Preliminary Analysis 2 – ScatterPlot.ows*. In the Figure 30 we circled most of the bigger hospitals with an ellipse. Our further research will focus more on those hospitals, as they compose a big percentage of hospitals. We will also study the hospitals above the red line, which have very long transportation times and we try to determine the reasons behind slow patient delivery times.

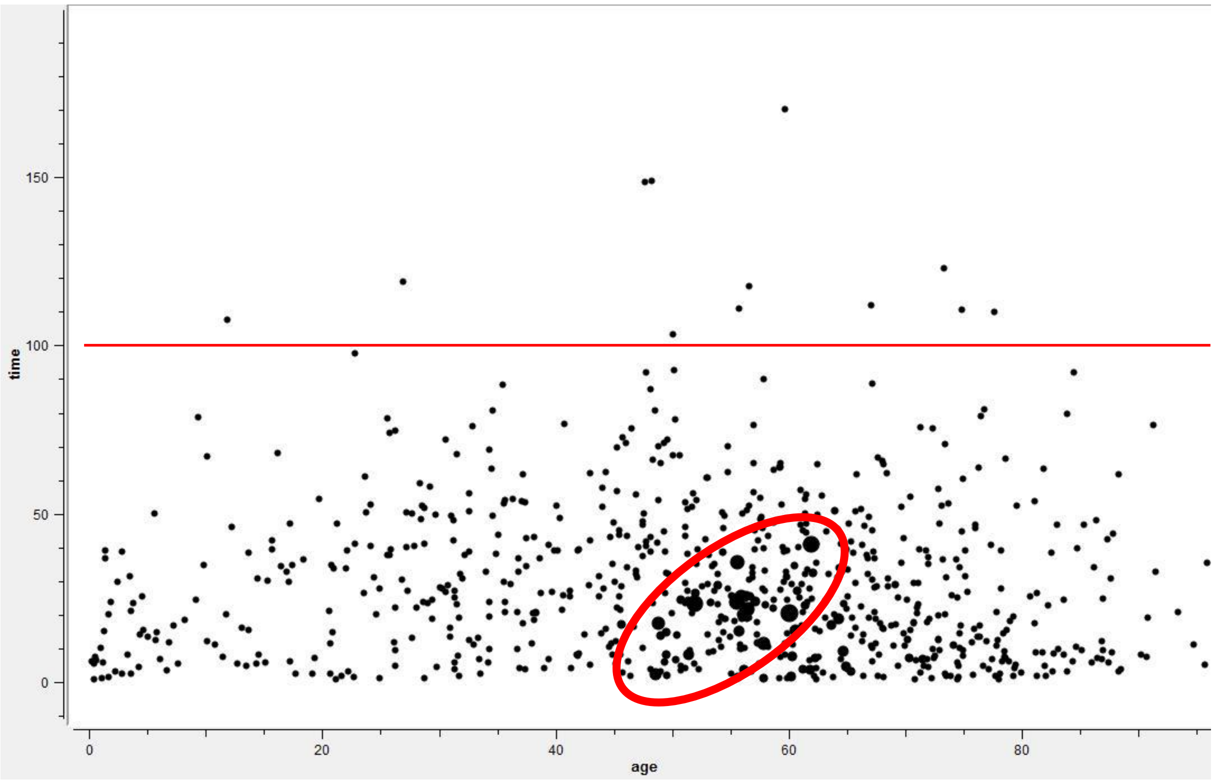


Figure 30 - Scatter plot of hospitals based on delivery time and patient age dependency

Although we analyzed all the hospitals, we will split the main part of the research into two chapters, analysis of bigger hospitals and patient slow delivery time analysis to make the research more transparent. This will also enable us to focus more on the smaller set of hospitals for this thesis.

6.2. Analysis of main hospitals

For main hospitals, we chose hospitals with at least 2500 patients accepted (within our data). We limit this by loading the file *Main Hospital Analysis 1 – ScatterPlot.ows*, which contains schema with data selector that removes all the hospitals with less than 2500 patients. If we open the *Scatter plot* widget, we can see the hospitals arranged as it is shown in Figure 31. We can notice that most of the hospitals were removed, except for the following hospitals:

- 高井病院
- 市立奈良病院
- 天理よろづ相談所病院
- 郡山青藍病院
- 恵王病院
- 県立医大附属病院
- 東朋香芝病院
- 県立三室病院
- 県立奈良病院
- 石洲会病院

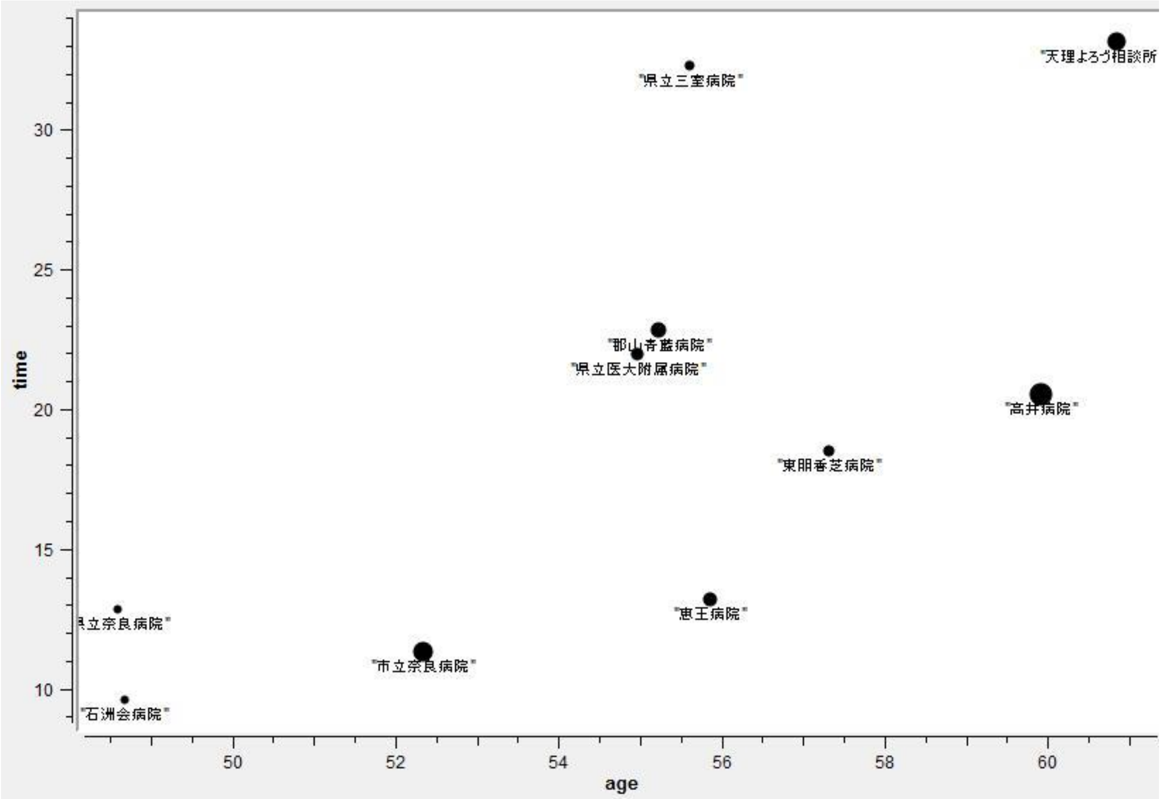


Figure 31 - Scatter plot displaying only selected main hospitals

To better understand why hospitals reject patients, we create a new file that contains aggregated information about calls to the hospitals that accepted the patients. Then we connect the file widget to one of the newly created widgets – *Compare Examples*. This way we can get to understand the type and origin of patients for each of the main hospitals. We can load the schema from file names *Main Hospital Analysis 2 – Comparation.ows*. As aggregation is a bit harder to perform on the PostgreSQL (even with the use of *SQL Filter* widget it may take some time), we created utility application that aggregates all the data by every attribute from a simple Orange .tab file. This way we can select data with the *SQL Filter* widget and aggregate it later without having to select it again. The console application is located in the *Utils* folder on the thesis disc.

After loading the scheme and double-clicking the *Compare Examples* widget, we can select all age attributes (from *A (0-9)* to *G (85-)*) to see the age distribution of patients for each hospital, as seen on Figure 32. We can notice that the hospitals are mostly the same with the exception of the way they handle the children – some hospitals accept very few children under the age of 10. Even the largest hospital, 高井病院, belongs to that group, with children under the age of 10 composing only 2.6% of all accepted patients. 郡山青藍病院, with only 0.6% of children under the age of 10 and 1.1% of children under the age of 15 for accepted patients is the most extreme example in this group of hospitals.

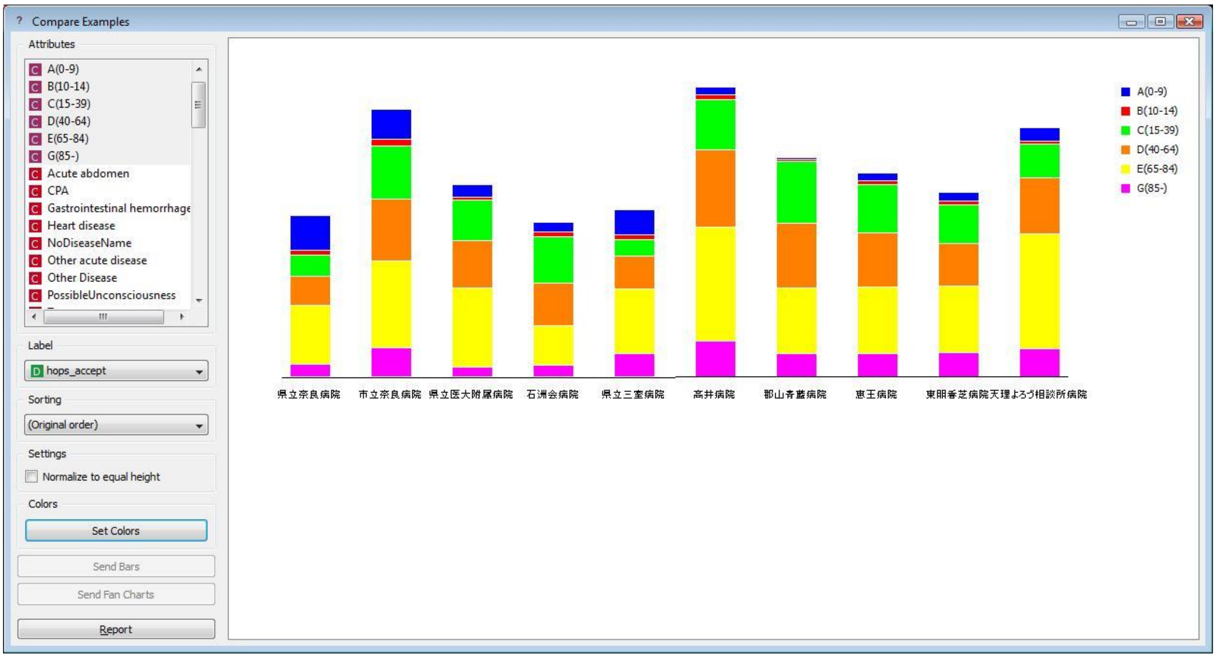


Figure 32 - Compare Examples widget displaying age groups for main hospitals

If we aggregate the data to display disease type of the patient, we can see that the disease type distribution for hospitals is much more different that the age distribution. On Figure 33, we can notice that hospital 石洲会病院 focused mostly on trauma patients (77.5%), while 県立医大附属病院 focuses mostly on heart and acute diseases (together 58.2%) and handles much less trauma patients – only 9%.恵王病院, which we mentioned in the previous chapter as the hospital with faster delivery times for patients, also focuses mainly on trauma patients (53.8%), but receives the least amount of patients with possible unconsciousness (4.6%).

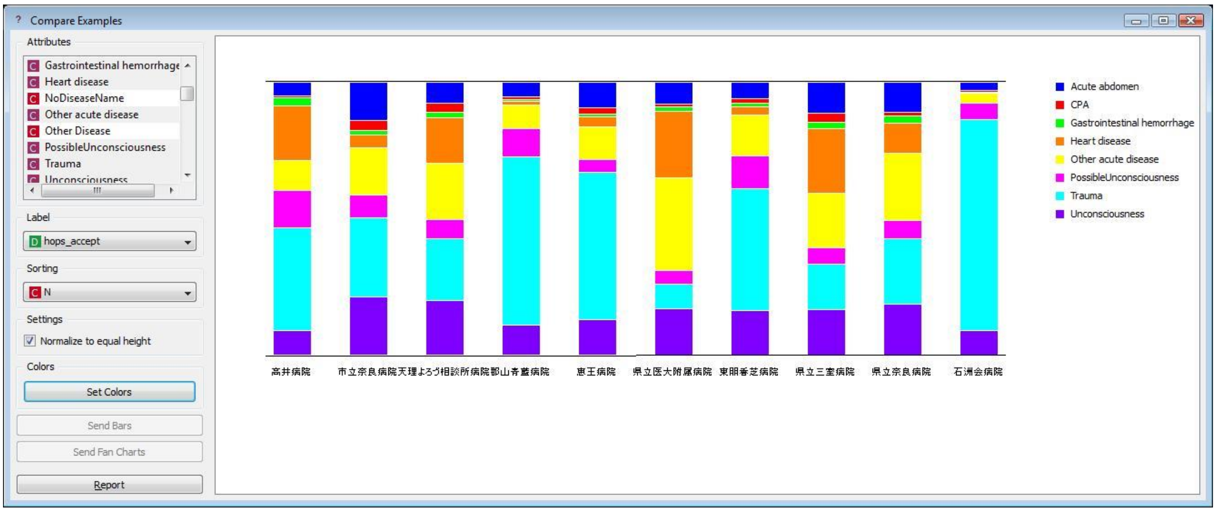


Figure 33 - Compare Examples widget displaying disease types for main hospitals

To better understand all incoming calls to these hospitals, we perform the same step with data on the refused calls. The author prepared another file called *Main Hospital Analysis File 3 - Referral.tab - aggregated.tab*, which contains aggregated data of refused patients for each hospital. Even though some patients were refused by multiple hospitals, we use the data of all refusals. If we load the scheme file *Main Hospital Analysis 2 – Refusal Comparation.ows*, we can open the *Compare Examples* widget to view refused patient distribution by age group. On the Figure 34, we can see that most of the hospitals have similar age group refusal distribution, with exception of hospital 県立三室病院, which seems to refuse much more children under the age of 10 than other hospitals. Although it also accepts above average number of children, the hospital 県立奈良病院 refuses smaller percentage of child patients. But, in 県立奈良病院 case, we can see that they generally refuse much more patients – even more than they accept.

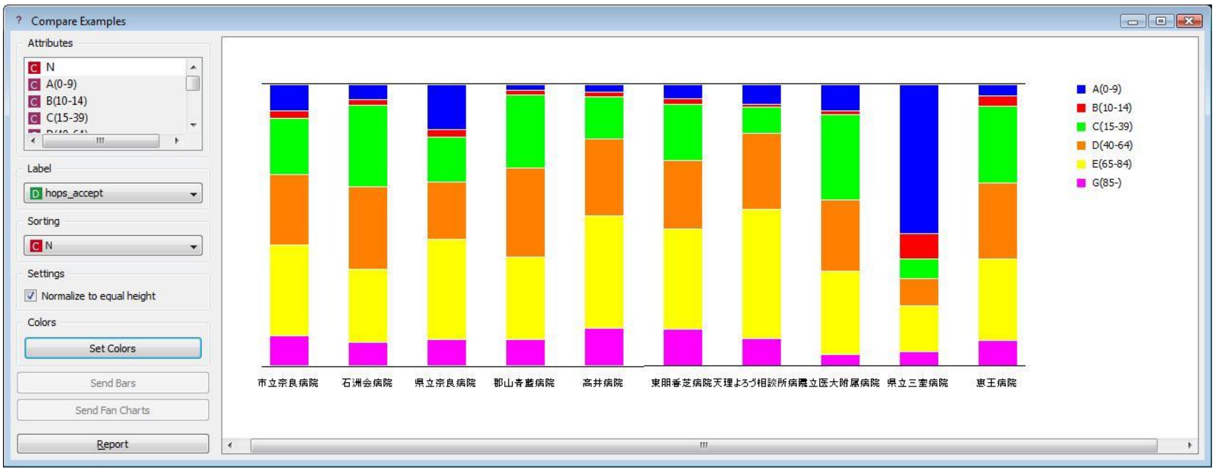


Figure 34 - Compare Examples witget displaying referral datas patient age group distribution for main hospitals

If we change the *Compare Examples* widget to display diseases, we can see that the hospitals statistics mostly match the *Compare Examples* widget in Figure 33, with the exception of hospital 県立医大附属病院, which rejects fewer patients with heart diseases than expected. In the provided schema file, the *Compare Examples* widget is further connected to the *Narpis* widget, but as this is used mostly to present results to EMS specialists, we will not discuss it in this thesis. Instead, we will go on to present the data in the more advanced *Narhodo* widget.

To better visualize and later present the analysis to the proponents of this research, we created a new schema, saved in the file *Main Hospital Analysis 3 – Narhodo.ows*. If we load this schema and open the NarhodoH2 widget, we can click *Report* and generate the same image as on Figure 35. As mentioned before, Narhodo widget displays connections between hospitals and EMS centers. From here, we can see that the 3 main hospitals are very close together in the EMS named 奈良市, but have a lot of rejected patients (more than 50%), while patients from EMS named 中和広域 seem to be without any rejected patients. This probably means that the data on refused calls was not collected for the latter EMS.

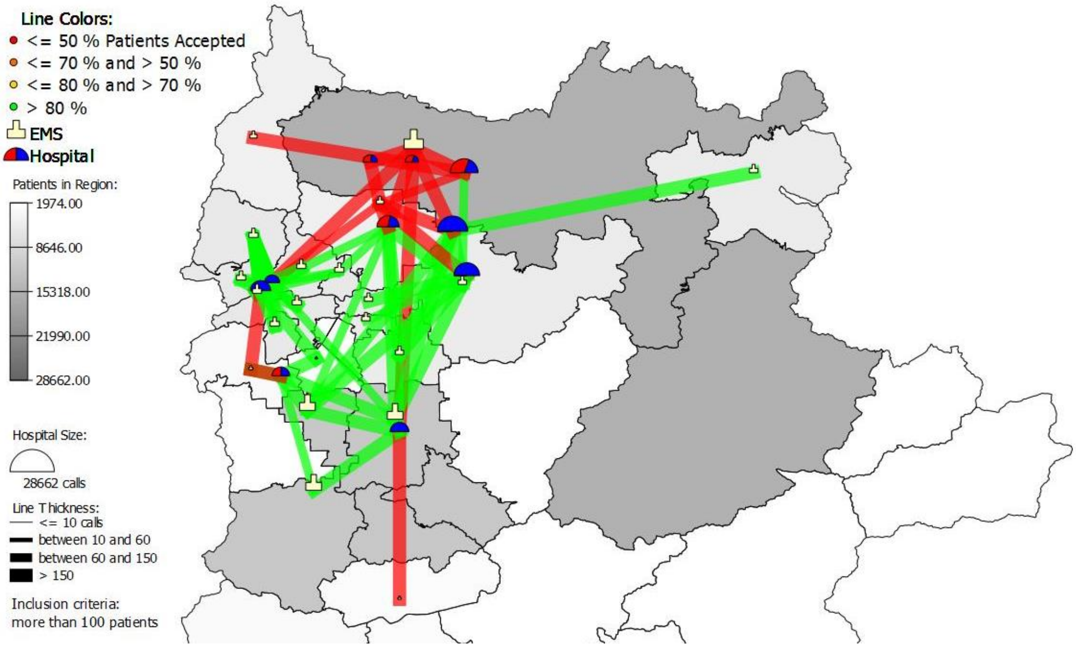


Figure 35 - Narhodo widget displaying information on main hospitals

6.3. Analysis of hospitals with slow delivery time

Now that we reviewed the main hospitals, we can continue on analyzing other significant type of hospitals – the hospitals with slow delivery time. If we open schema in file named *Slow Hospital Analysis 1 – Scatter Plot.ows*, we can see a schema similar to the one in the beginning of previous chapter. Again, we open the *Scatter Plot* widget which now only displays hospitals, which have delivery time of over 60 minutes and at least 50 accepted patients. Using this filter values yields only four hospitals, as other hospitals either have less patients or faster patient delivery times. The problematic hospitals, which we can see in Figure 36, are:

- 紀和病院
- 近畿大学医学部奈良病院救命救急センター
- 県立奈良病院救命救急センター
- 小南病院

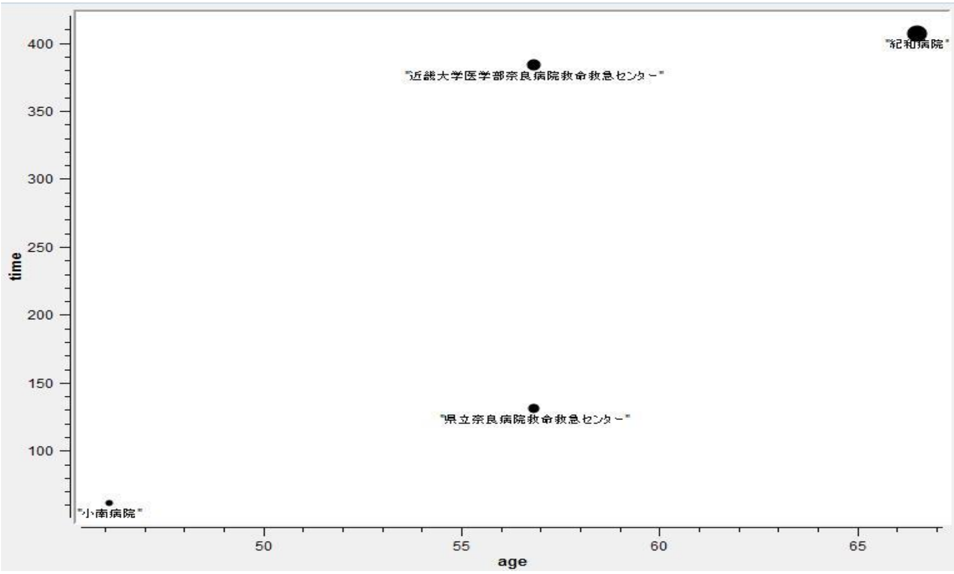


Figure 36 - Scatter Plot widget displaying data on hospitals with slow patient delivery time

As there are only few hospitals with slow delivery times, we will first focus on comparing the hospitals among themselves and analyze the patients with the slowest delivery times. Then we will compare the hospital statistics to main hospitals.

If we create a schema similar to the one from previous chapter (or load it from schema file *Slow Hospital Analysis 2 – Comparison.ows*) and open the *Compare Examples* widget as displayed on Figure 37, we can see that the patients in these hospitals are mostly of ages 65-84, except in the hospital 小南病院, which mostly accepts young adults of ages 15-39. This is also the only hospital in this study to accept mostly patients of this age. If we display distribution of diseases, we can notice that these hospitals accept much less trauma patients than main hospitals, but hospitals 近畿大学医学部奈良病院救命救急センター and 県立奈良病院救命救急センター seem to accept much more patients with CPA than any other hospital researched.

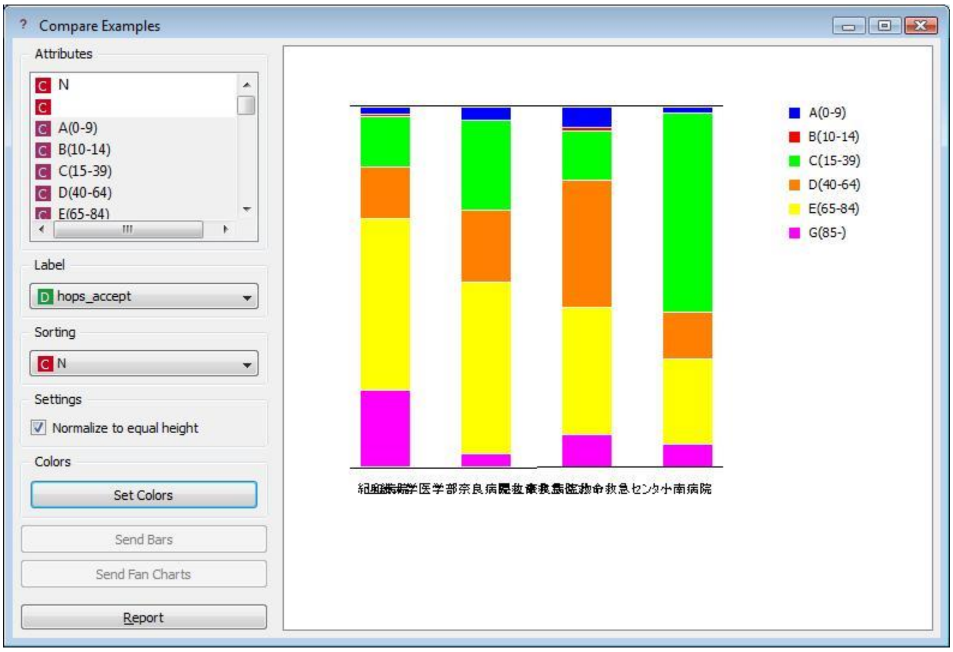


Figure 37 - Compare Examples widget comparing patient year distribution for "slow" hospitals

If we look at the *Data Table* widgets that contain referral data and acceptance data, we can notice that hospital 県立奈良病院救命救急センター has a much higher refusal rate than other hospitals – only about 42%, while all other hospitals have acceptance rate more than 65%, with 紀和病院 having acceptance rate more than 95%. *Compare Examples - Referral* widget again contains visualization of aggregated data of the patients hospitals refused to accept.

As hospitals 紀和病院 and 小南病院 did not mark any diseases in referrals, it is impossible for us to analyze the disease distribution for these two hospitals (only for referrals). Fortunately, we can still analyze the age group distribution, as shown in Figure 38. Since we can still perform disease distribution analysis on the other two hospitals, we can select all disease types to display the bar charts. We can notice that although hospital 県立奈良病院救命救急センター receives more patients with CPA, they also refuse more of these patients.

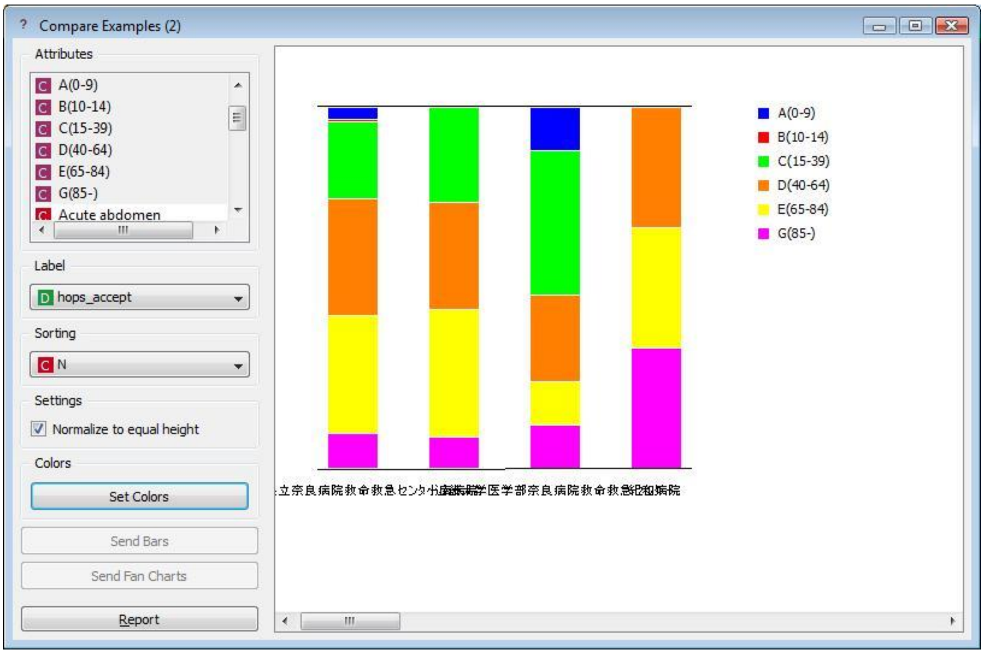


Figure 38 - Compare Examples widhet displaying age group distribution for hospitals with slow patient delivery times

6.4. Analysis of delivery times

For us to better advise the hospitals, it is crucial to know what happens with incoming patients through the day. To see how the distribution of different variables changes, we use the *Time Compare* widget. In Figure 39, we can see the age group distribution, which varies greatly with time. We can read out that there is an interesting pattern of age group distribution during the average of 24 hours. In the morning there seems to be more of the older patients, with the peak at 8:00, but towards mid-day, the number of younger patients rises until it reaches the peak at 15:00 and starts falling again towards the evening.

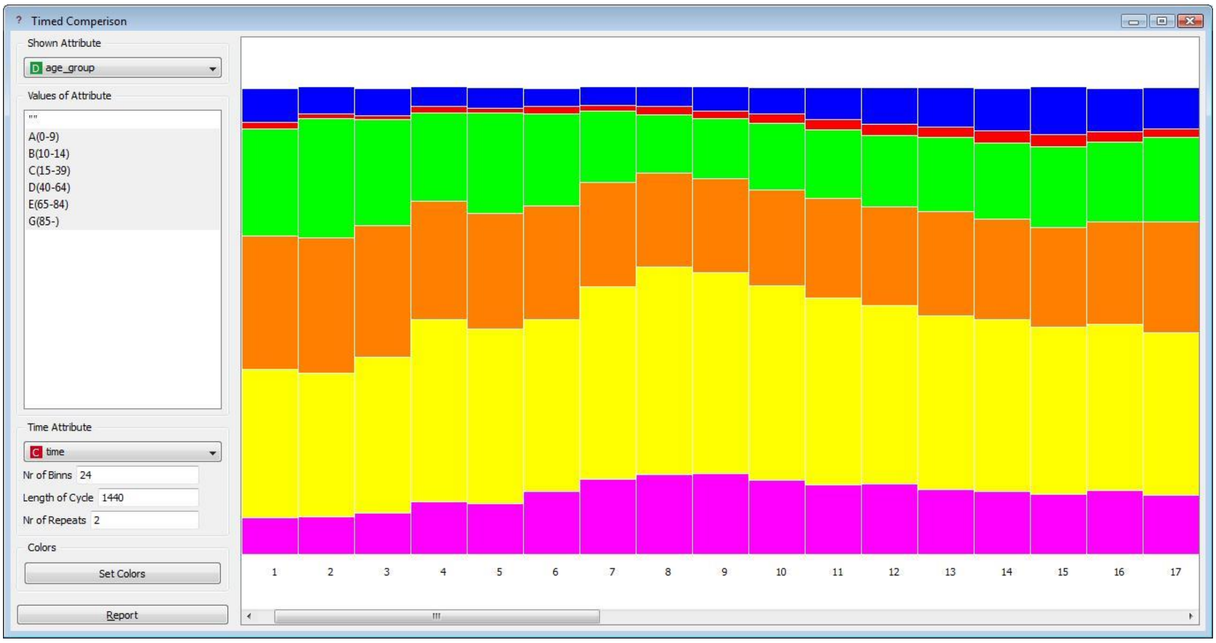


Figure 39 - Patients age group distribution throughout the day

It is also interesting to see Figure 40, which displays the disease distribution throughout the day. We can notice that the number of trauma patients decreases during the night, but the number of patients with acute abdomen diseases greatly increases.

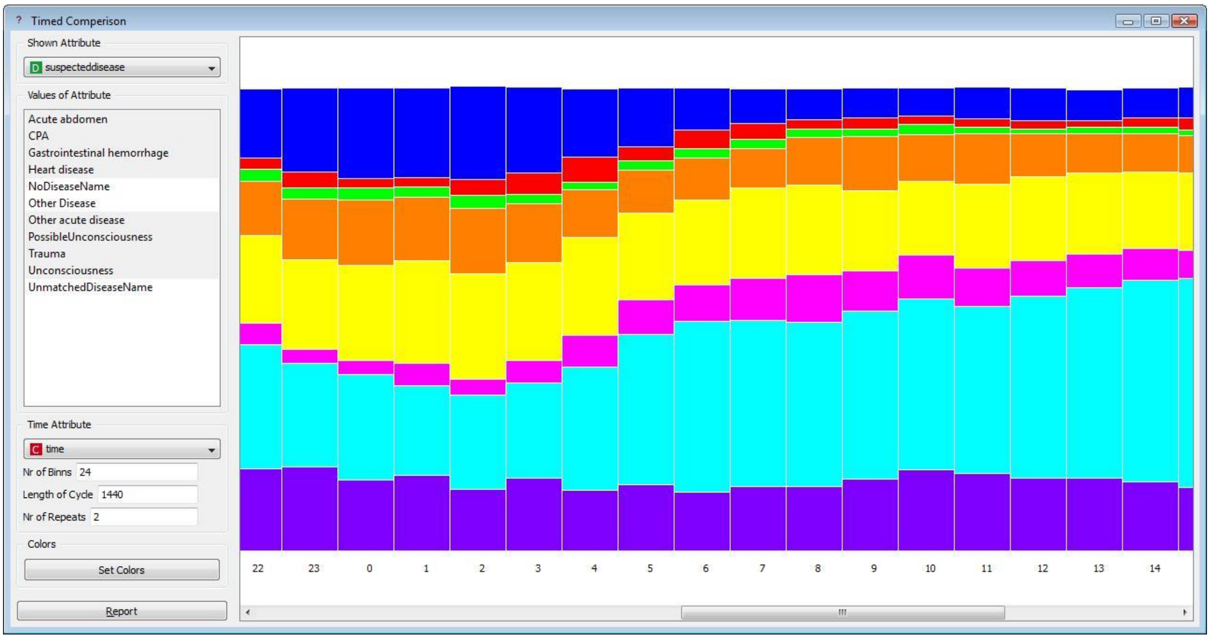


Figure 40 - Patients disease distribution throughout the day

7. Conclusion

The data analysis showed that the patients' ages and disease types vary a lot by hospitals and the time of accident. Unfortunately, the data analysis described here contains only a portion of original analysis, as there were some widgets that we did not use, because they were created only to create visualizations and to prepare the data for EMS specialists' inspection. There were also some widgets that we introduced, but they have used the non-aggregated data, which is classified and thus cannot be presented with this thesis.

With the widgets used in this thesis, we can still determine that hospitals would be able to process patients better if they optimized the doctor shifts to match the disease and age distributions. Also, if the EMS teams knew the hospitals acceptance rate, they would know which of the nearby hospitals is more likely to accept the patient and would not waste time on unnecessary transportation time.

Although the Japanese emergency medical services have evolved greatly since they first appeared in 1977, the research on lowering the patient delivery times is still a very young subject. To help with the development of this research, we first created new widgets for the Orange data mining suite and then used them to research hospitals and try to determine the reasons for slow patient delivery times. As the data by itself was not useful enough, we created additional widgets to help us with aggregation, filtering and selection directly from database. As there are still many possible visualizations that could be created with the developed widgets, the author recommends that they be used also in other researches. There is also a possibility of creating a unified system for processing patient data based on Orange, as all the necessary widgets were developed in the scope of this thesis.

8. References

[1] Statistical office of Japan (http://www.stat.go.jp/)

[2]H. Miyazaki, N. Hanada, M. I. Andoh, Y. Yamashita, T. Saito, A. Sogame, K. Goto, R. Shirahama, T. Takehara, "Periodontal disease prevalence in different age groups in Japan as assessed according to the CPITN", published online, 29 MAY 2006 DOI: 10.1111/j.1600-0528.1989.tb00591.x

[3] Toshifumi Konda, Kazunari Kamachi, Masaaki Iwaki, Yasuko Matsunaga, "Distribution of pertussis antibodies among different age groups in Japan", in publication *Vaccine*, Volume 20, Issues 13–14, 15 March 2002, Pages 1711–1717

[4] Shingo Hori, "Emergency Medicine in Japan", in publication "*The Keio Journal of Medicine*", Vol. 59 (2010) No. 4 P 131-139