

MATHEMATICAL INSTITUTE

INTERNSHIP REPORT

STATISTICAL SCIENCE FOR THE LIFE AND BEHAVIOURAL SCIENCES

Visualization of Demographic Phenomena at Statistics Netherlands

Author:
W.S. van Loon

Internal supervisors:
Dr. S.M.M. te Riele
Drs. E. de Jonge
Statistics Netherlands

External Supervisor:
Prof. Dr. H. Putter
Leiden University

June 2016



**Universiteit
Leiden**
The Netherlands

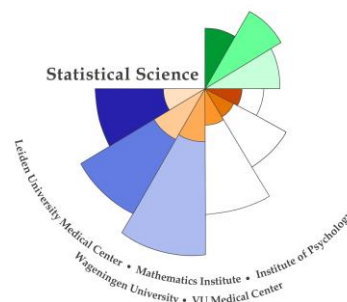


Table of contents

An Introduction to Statistics Netherlands	3
Internship Activities	4
Generic Skills Acquired or Strengthened	10
A Specific Example: Heatmappr	12
Conclusion	16
References	17
Appendix A: Heatmappr Tutorial	18

An Introduction to Statistics Netherlands

Statistics Netherlands (CBS) is a Dutch governmental institution which is tasked with collecting and publishing reliable statistics about, and for the benefit of, Dutch society. The statistics published by CBS cover a wide variety of fields including the Dutch population, the economy, social security, crime, healthcare, and the environment, and are used for policy-making at both a national and European level. Many statistics published by CBS are publicly available through the online database StatLine, and CBS can provide custom research and consultancy services on request. CBS has two major offices in the Netherlands, one in Den Haag and one in Heerlen, as well as a minor office on Bonaire.

In addition to providing useful information for policy-makers, CBS aims to effectively communicate its findings to the general public through regular news items on its website, and through social media such as Twitter. CBS news items are regularly picked up by media outlets and often receive considerable public attention (for example, the items surrounding the '17-millionth citizen'). One way of effectively communicating CBS findings is through data visualization. Most CBS news items contain at least one graph, usually in the form of line or bar charts although other kinds of graphs are used occasionally. Within CBS there is an interest in diversifying the kind of visualizations that can be produced.

I conducted my internship at the CBS department of demography in Den Haag. The demography department covers topics such as births, deaths, migration, marriage, and divorce. The demography team was interested in exploring new data visualization techniques for demographic phenomena. In particular, a heatmap of divorce in the United States was the inspiration for my internship project. The objective of my internship was to create a similar visualization for the Dutch situation, preferably in the form of some generic tool. I also had the possibility to try other visualization techniques, or suggest other demographic phenomena to visualize. During my internship I explored, presented and discussed several visualization techniques, and applied them to the demographic phenomena of divorce and migration. Furthermore, I developed a generic analysis tool for creating heatmaps.

Internship Activities

I conducted my internship at the CBS department of demography for a duration of 3 months, working 4 days a week. During this time, I was tasked with the exploration and development of different data visualizations for demographic phenomena. The initial focus was on visualizing divorce, but later other demographic phenomena were also considered. My internship can roughly be divided in two stages. The first stage of my internship was largely exploratory and involved me familiarizing myself with the available data and generating various concept versions of possible visualizations. About halfway through my internship I presented several of these concept visualizations to an audience of my colleagues. After deliberation with my supervisors, taking into account the feedback I received after my presentation, the decision was made to focus on one particular visualization technique and to develop a generic analysis tool which would allow my colleagues to generate these visualizations themselves from their own data. The development and testing of this tool covered the second stage of my internship.

Data exploration and software limitations

I spend the first week of my internship familiarizing myself with the available data. As the initial focus was on visualizing divorce, I started with a data set containing the marital status history of people in the Netherlands. Personal characteristics had to be imported through matching cases with information from other data sets. These data sets concern at least all current citizens of the Netherlands, often include previous citizens as well, and may contain multiple rows per citizen. These data sets can therefore be very large: the file containing marital status history has well over 40 million rows. As I was limited to working on a 32-bit virtual machine with 2 gigabytes of available RAM, a significant amount of pre-processing and some creativity was required to get the data into R. In the end I settled for an approach whereby I would do all of the matching and pre-processing in SPSS, aggregate the file, and then load the aggregated file into R. While this meant that I often had to generate a new aggregate whenever I wanted to visualize something different, it was the most effective solution under the given limitations.

There were also limitations regarding the use of R. CBS has a central installation of R and RStudio, including a local copy of the CRAN repository. However, whenever I wanted to use a package which was not in the repository, or when a package required additional software, this was problematic. Additionally, I was limited to using Internet Explorer as a browser. While these limitations were not such a problem for most of the exploratory phase, they did become significant when developing the generic application.

Visualization techniques

During the first stage of my internship I explored several different visualization techniques. These include Sankey diagrams, heatmaps, perspective plots, survival plots and a migration pyramid. Below, I will show some examples which I also presented to my colleagues. Note that they were meant primarily as proofs of concept, i.e., as examples of the kind of visualizations that could be considered based on the available data. They were not meant to convey the result of some extensive analysis.

The Sankey diagram shown in Figure 1 was one of the first visualizations I created. The advantage of this diagram is that it can visualize the full relationship history of a group of people in a single picture. The primary disadvantage is that because of the repeating nature of marriage and divorce, this diagram is quite atypical for a Sankey diagram. This means making a reasonable looking plot like Figure 1 in R is actually fairly difficult, and it would be hard to make a generic application that can generate such diagrams for different data sets. Also, when proportions get really small, the proportionality of the paths is hard to maintain: either a minimum path width (which is the case in Figure 1) or some transformation is needed to prevent paths from getting invisibly thin.

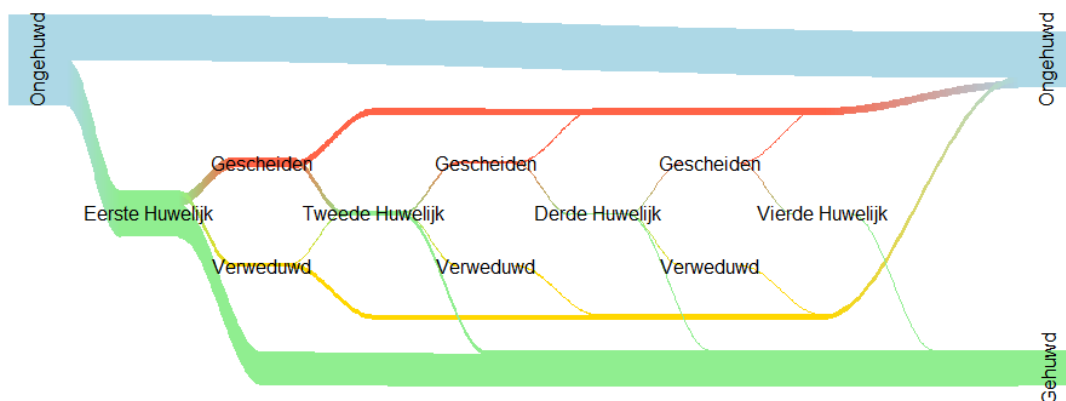


Figure 1. A Sankey diagram based on a retrospective random sample of 5700 Dutch citizens. All citizens were unmarried at birth. Currently some are unmarried (top endpoint) and some are married (bottom endpoint). Some citizens are unmarried because they never got married, others married but are now divorced. Every possible route from left to right is a path that can be taken from birth to the current marital status. The thickness of the paths is proportional to the number of people who took that path, although there is a minimum path thickness.

The heatmap shown in Figure 2 is a replication of a heatmap found in an American blog post (Cohen, 2015), which was the primary inspiration for my internship project. Although initially generated using a different package, the version shown here was created using the generic application I developed, which is discussed further in the *A Specific Example* section.



Figure 2. A heatmap of the proportion ($\times 1000$) of married women in their first straight marriage who divorced in 2015, by marriage duration and age at marriage. Divorce proportions are highest for marriages with a duration between 1 and 10 years, where the woman was young at the time of marriage. Cells with less than 100 marriages are white.

An alternative to heatmaps are perspective plots. In these plots the height of the surface indicates the value of the corresponding cell in the table. While perspective plots can be visually appealing they can also be quite hard to understand, and when shown from one particular angle (as in Figure 3), some of the information is obscured. Perspective plots work better as interactive visualizations than static images.

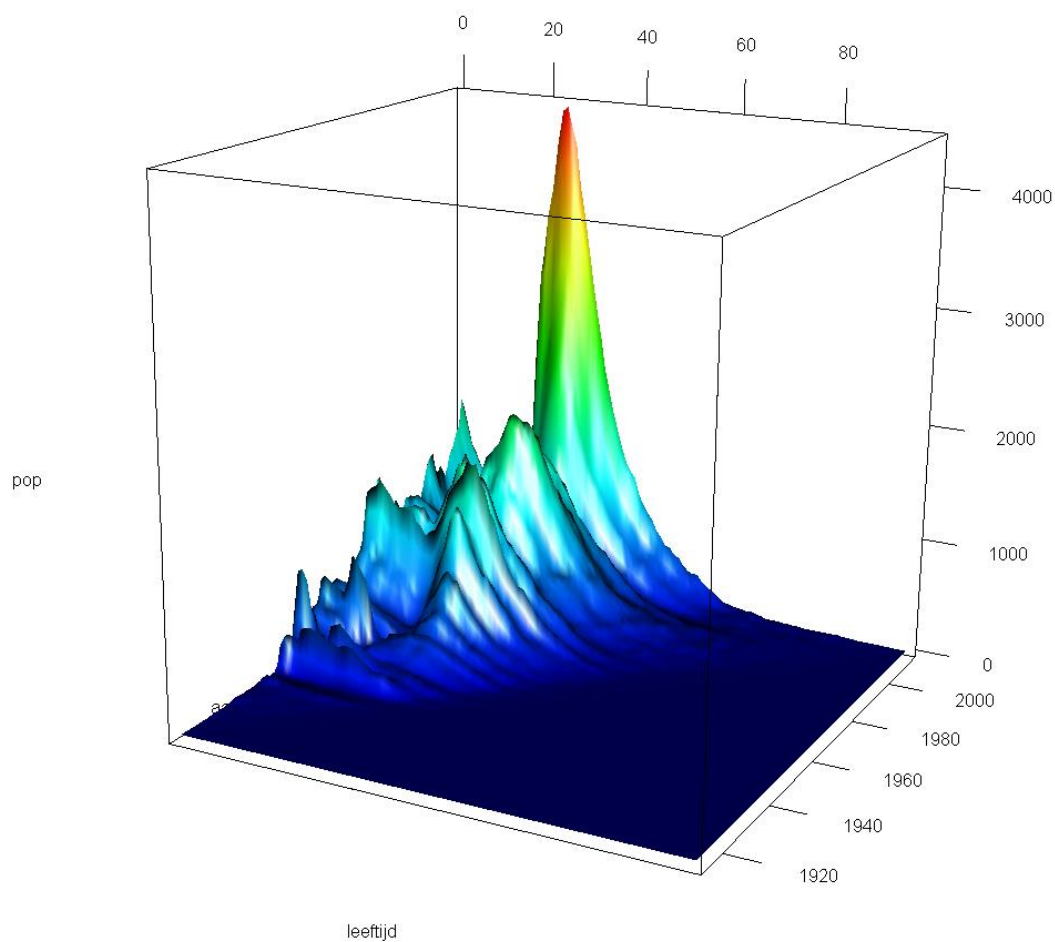


Figure 3. A perspective plot of the current population of first-generation immigrants in the Netherlands, by year of arrival and age at arrival. The immigrant population primarily consists of people who came to the Netherlands at an early working age (20-25) or as young children. Immigration in the Netherlands has followed a wave-like pattern over time, with waves often following specific events (mass migration surrounding the independence of Suriname, refugees from the Yugoslav wars, etc.). Such immigration waves going as far back as the 1950's can still be observed as peaks in the distribution of current first-generation immigrants.

Divorces (as well as emigrations) can be considered occurrences of events in some risk population. Therefore, one way of visualizing these events would be through survival curves, an example of which can be observed in Figure 4.

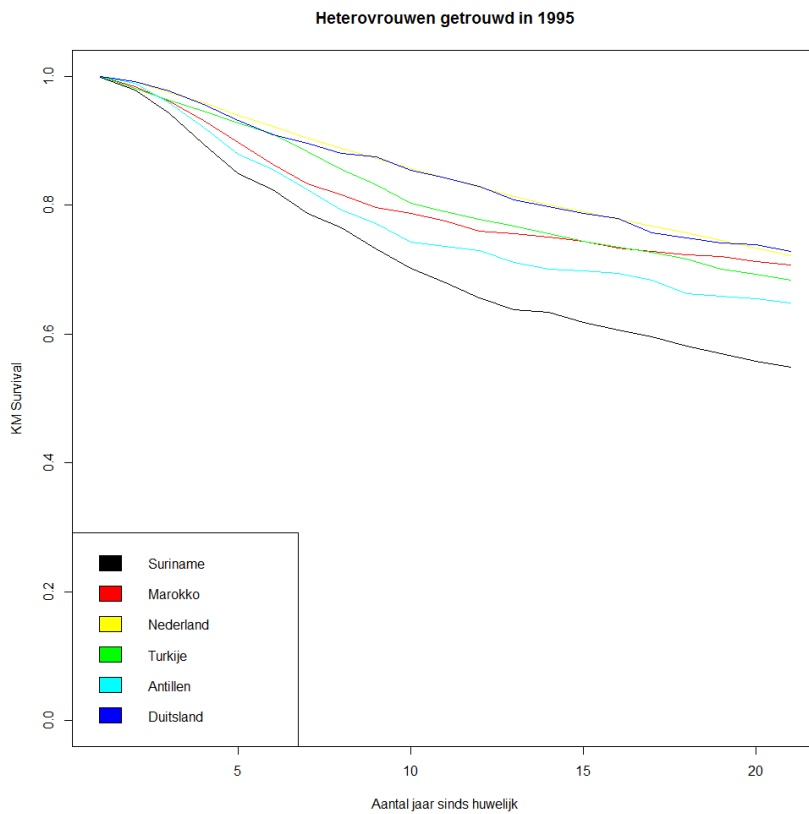


Figure 4. Kaplan-Meier survival curves over the course of 20 years for women in their first straight marriage, separated by country of birth. The event is divorce, observations for which the marriage ended due to the death of either partner are treated as censored.

Figure 5 shows another visualization, meant to look familiar to users of the ‘population pyramid’ published on the CBS website.

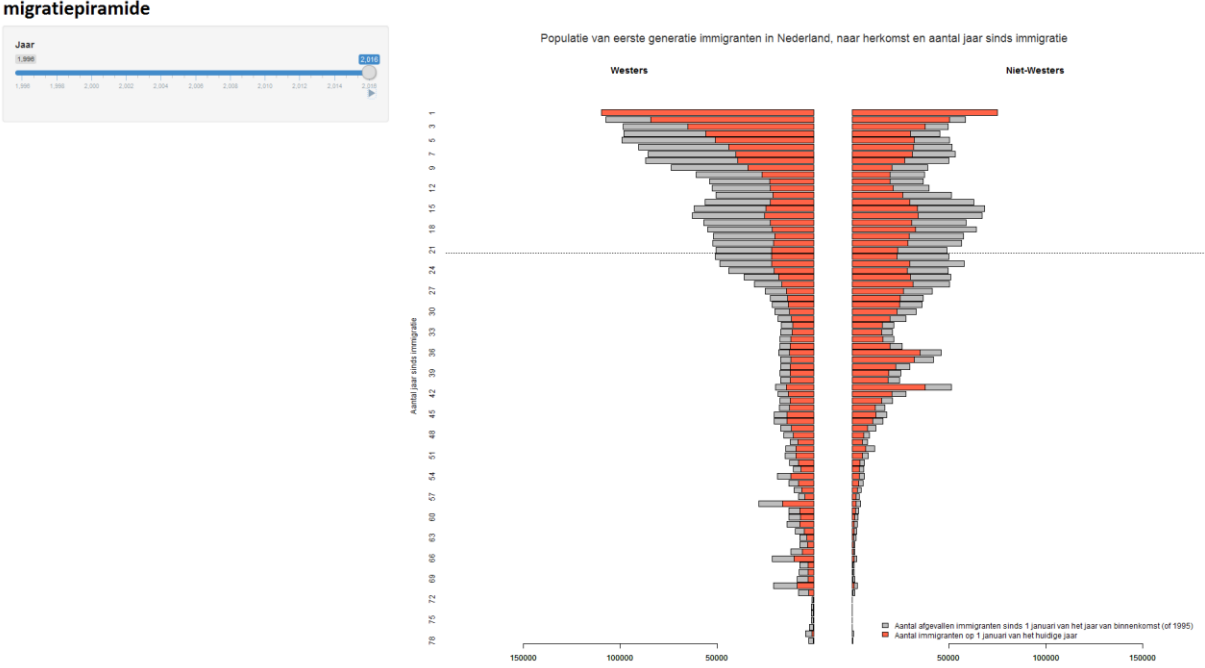


Figure 5. Two histograms. Left: Immigrants with a Western background. Right: Immigrants with a non-Western background. The vertical axis, from top to bottom, denotes the number of years since arrival. For each year since arrival, the grey bar represents the total amount of immigrants of each type which arrived that many years ago. The red bar indicates the amount of immigrants that are currently still in the Netherlands. Below the dotted line the grey bar indicates the number of immigrants present in 1995, as immigration data before that year was unavailable. In the original Shiny application, the histogram moves from top to bottom and one can see for each year since 1995, the number of immigrants that arrived that year as well as the immigrants that have remained of previous years.

About halfway through my internship I presented these and other visualizations to my colleagues. Of these visualizations the Sankey diagram and heatmaps received the most attention. After further deliberation with my supervisors the decision was made to focus on developing a generic analysis tool to allow CBS employees to make heatmaps of their own data. The development of this tool spanned the second part of my internship, and is described in the *A Specific Example* section.

Other activities

My internship consisted of a project mostly separate from other CBS proceedings. As such, I spend the entirety of my internship working independently on this project, although I did have regular meetings with several of my colleagues. I gave a presentation on the various visualizations I created, and gave a workshop on how to create heatmaps using the application I developed. I also attended several team meetings, and helped another intern with his survival analysis.

Generic Skills Acquired or Strengthened

The reason I applied for this internship position at CBS was two-fold. First of all, data visualization is a topic that is not widely discussed in the statistical science master. Graphs are of course included in every course, and some data analysis methods are also visualization methods (e.g. principal component analysis, multidimensional scaling/unfolding), but topics such as what makes a good visualization, or how to create interactive visualizations did not get as much attention as I would have liked. Because of this, I was interested in doing an internship in which I could expand my knowledge and experience in data visualization beyond that which was discussed in the statistical science program. Secondly, I really enjoy programming and sought to broaden my programming skills.

During my internship I worked with both R and SPSS. I was forced to use SPSS initially due to the memory restrictions on the system I was using. As I generally use R as my statistical analysis program, I had to significantly refresh and expand my knowledge of SPSS syntax programming for combining and processing the datasets into a format that could be loaded into R. I also experimented with a package which allowed me to work with large datasets in R, but in the end decided that the approach of processing the data in SPSS first was quicker.

After processing the data in SPSS and loading it into R, I could experiment with different visualization techniques. To this end I experimented with various plotting libraries. While a heatmap was suggested in the internship project description, I was also allowed to suggest and experiment with other visualization methods. Coming up with different visualization techniques forced me to really think about the kind of data I was working with, which aspects of the data would be interesting to visualize, and who the audience of the visualization would be. I had regular contact with my supervisors in which we discussed the merits of each visualization type. Some visualizations are effective but not very 'spectacular', e.g. survival curves. Other visualizations are visually appealing, but may be too complicated for the target audience or publication, e.g. perspective plots. The target audience was also a topic of discussion, as the heatmap was perhaps too complicated for the regular CBS press-releases, but suitable for in-house analysis and more extended publications. Eventually we made the decision to create a generic application for creating heatmaps which could be used as an analysis tool, but also allowed for easy exportation of the image in a variety of formats.

Most technical skills I acquired are with respect to programming Shiny applications. Shiny is a framework for R which allows statisticians with little to no knowledge of web development to make interactive (web)applications. While Shiny was briefly mentioned in the R programming course given in the Statistical Science program, it was not otherwise part of the curriculum. As such, I had to learn Shiny more or less from the basics. While making a basic functioning Shiny application is not difficult, more advanced applications can quickly get considerably complex.

One of the largest differences between general statistical programming in R, and programming in Shiny, is the notion of reactivity. When performing a regular statistical analysis, programming is generally strictly linear: a script is written and executed from top to bottom. However, when developing an interactive application, values in the environment are changed constantly by the user, and certain elements of the application need to react to this change in the correct fashion, including some elements of the user interface which should be dynamically shown or hidden.

Apart from the technical aspects of development, I also had to continuously monitor the wishes and requirements of my colleagues. Most of my colleagues used SPSS and Excel for their analysis. As such, the application had to be understandable without any R experience. Furthermore, one colleague would take over to maintain the project after I left, meaning the application had to be left behind in a structured and documented state. These were things I had to learn during the course of my internship, as I had no experience with developing software for use and maintenance by others.

My internship project as a whole was quite reminiscent of statistical consulting. I was presented with a 'problem' and independently worked on solving this problem, with regular meetings regarding progress and intermediate findings. At the end of my internship I provided CBS with a product which I believe is an adequate solution.

I also gave two presentations during my internship. One was a standard 'lecture' type presentation, but the other was a practical for which I had to create my own set of exercises, which is not something I had done before. However, I managed to create a comprehensive tutorial explaining the application I developed in detail, and my workshop was well received.

In summary, I acquired and strengthened various skills during my internship. I strengthened my knowledge of SPSS syntax and R programming. I learned how to develop interactive shiny applications, and deploy them in a professional setting. I expanded my knowledge about the various aspects of making a good visualization, improved my consultation and communication skills, and gave my first practical workshop.

A Specific Example: Heatmappr

In the second half of my internship I developed a Shiny application for creating heatmaps. Deliberation with my colleagues lead me to formulate several requirements for the application. First of all, the application should be usable by people who have no knowledge of R. Secondly, the application should accept datasets created in SPSS or Excel. Finally, the application should allow exportation of the heat map in several image formats commonly used within CBS, namely PNG, JPG, and SVG.

The initial heat maps I created were constructed using the *d3heatmap* library, which creates interactive heat maps as HTML widgets. However, editing and exportation of these widgets as static images proved challenging. Therefore, I made the decision to switch plotting libraries and re-wrote the application from the ground up to create static heatmaps using the *gplots* library. A screenshot of the resulting application can be observed below.

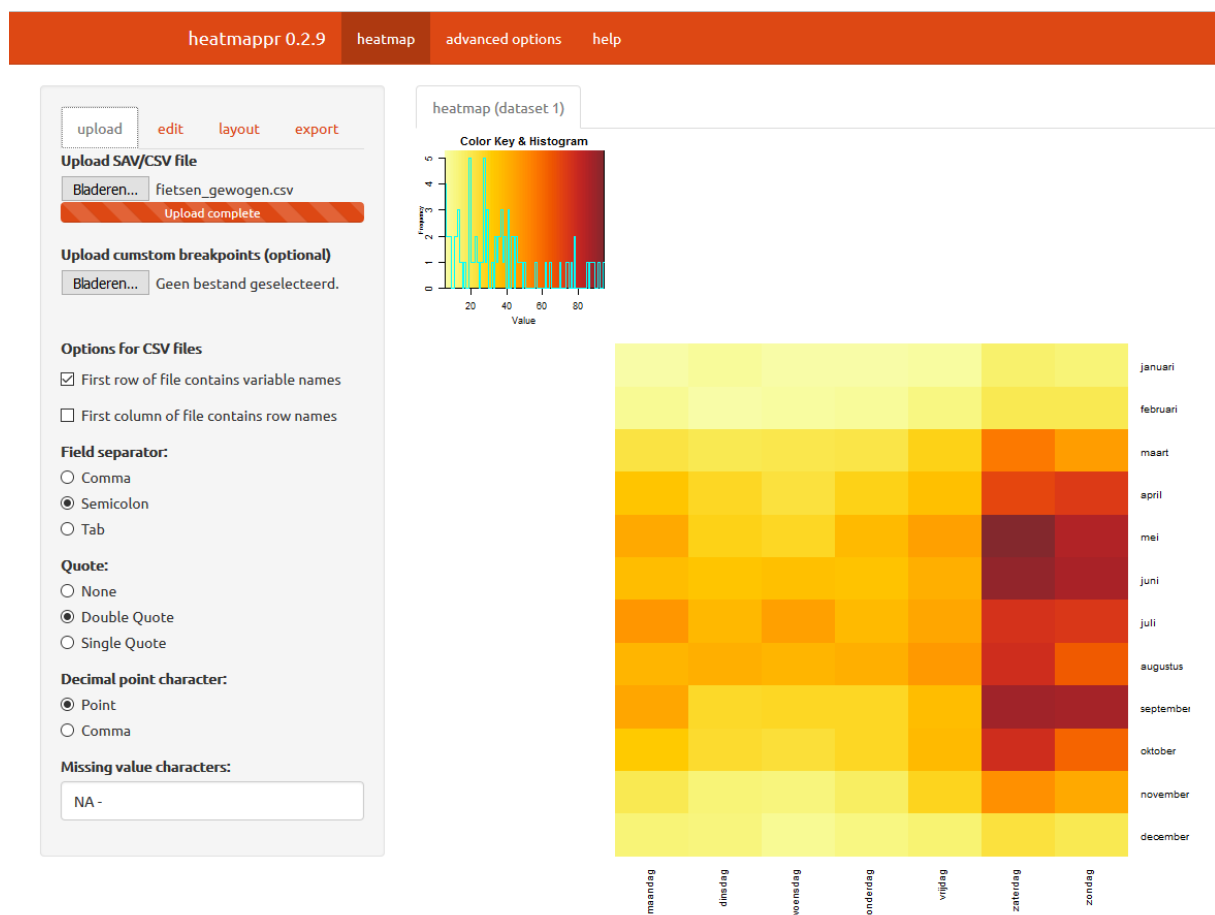


Figure 6. A screenshot of Heatmappr in use with the *upload* tab selected.

When a user first starts Heatmappr, he or she is presented with the *upload* tab. Here the user can upload a file in SAV or CSV format. For CSV files, a variety of options such as separation character and decimal character is available. When the data is uploaded successfully, a first rendition of the heatmap

will be shown in the plotting window to the right. If the data is not uploaded successfully, the user is presented with a user-friendly error message.

Under the *edit* tab, the user can edit various aspects of the heatmap. These aspects include various transformations of the heatmap values, the ability to select only a subset of the rows and/or columns, the ability to restrict the range of values, hierarchical clustering, and various options related to the color scale. Additionally, if the data is in a 'risk-population + number of events' format, this menu includes the option to switch between a heatmap of the risk population, the number of events, and the quotient of the two.

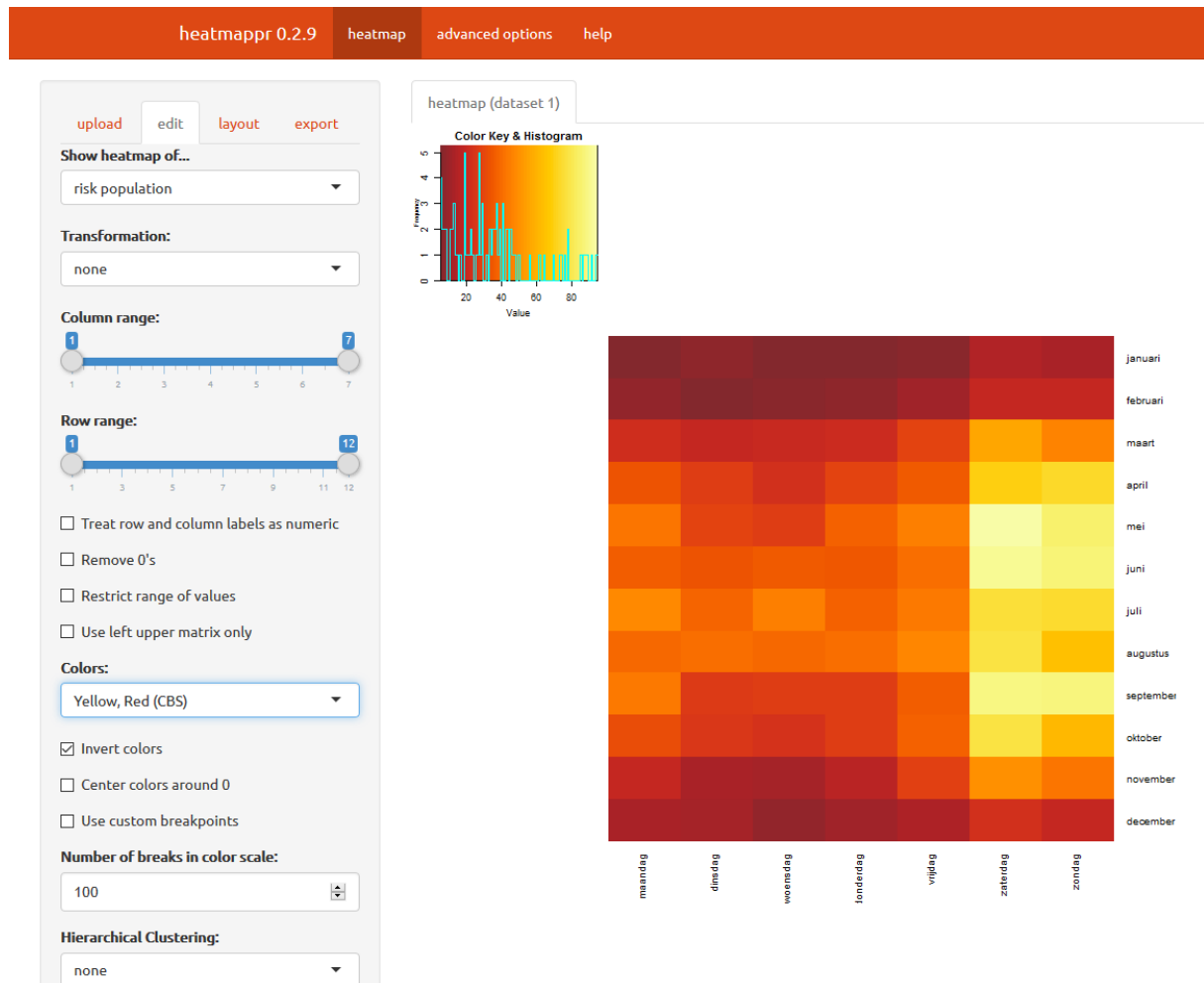


Figure 7. A screenshot of Heatmappr with the *edit* tab selected.

Under the *layout* tab, the user can add a title and axis labels to the heatmap if the defaults do not suffice. The same can be done for the color key, or it can be removed all together. There is also the option to show the values in addition to the colors in the heatmap, and various tweaking options related to colors, size of labels, etc.

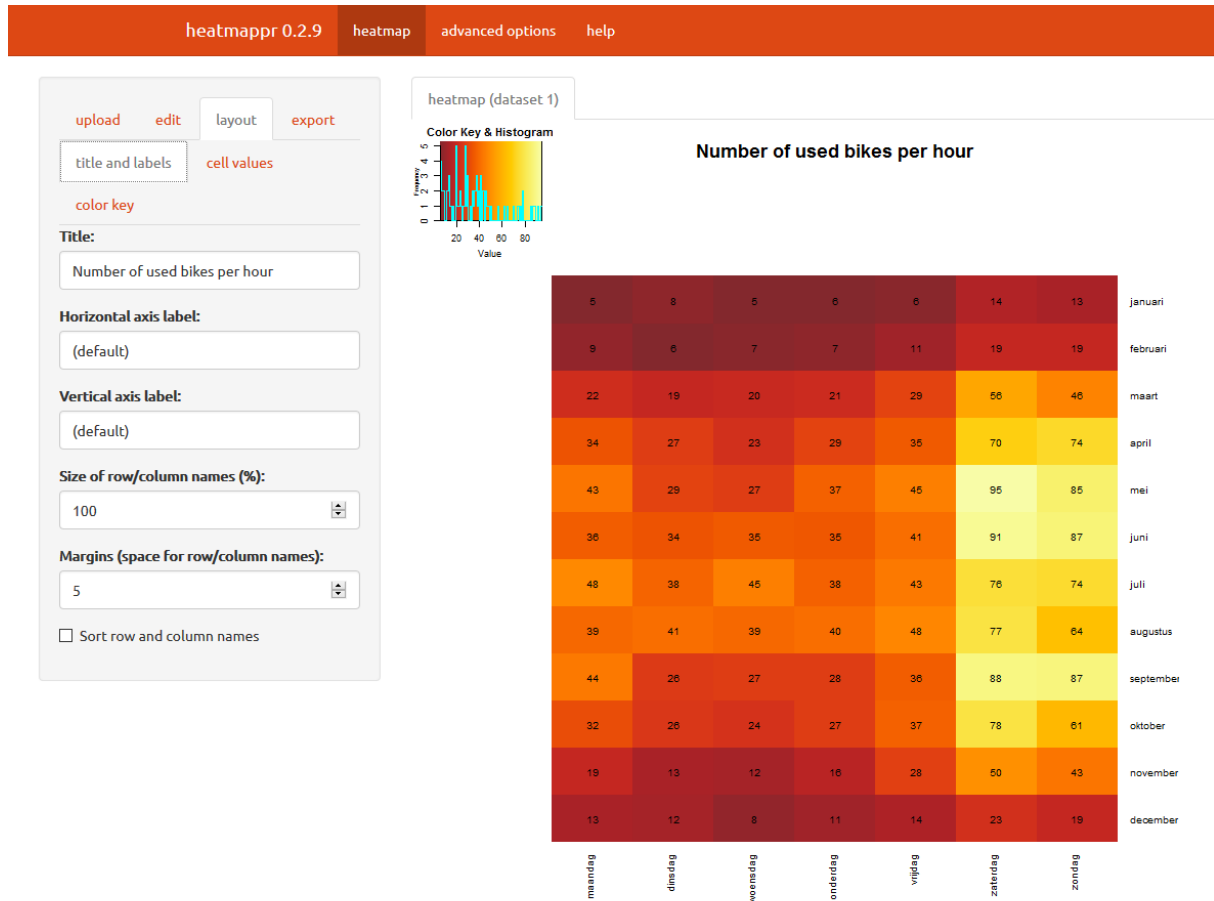


Figure 8. A screenshot of Heatmappr with the *layout* tab selected.

Under the *export* tab, the user can choose the image format and size and download the heatmap. The *advanced options* menu allows for changing hierarchical clustering options (i.e. distance measure, linkage method). There is an extensive *help* section which includes detailed instructions regarding the supported data formats and explanations for all the different options. It also includes an update history. Furthermore, a comprehensive tutorial (in Dutch, see the appendix) is available, which takes the user through all of the options in the analysis of an example dataset. Various other example datasets are available as well, each with a short description of their contents.

On the last day of my internship I gave a workshop in creating heatmaps with Heatmappr. The workshop consisted of a short presentation and a practical whereby the participants worked through the tutorial I wrote. This workshop was a major test case for both the application and the tutorial/documentation, as the app had previously only been used by a very small group of people. Fortunately, I got a lot of positive feedback and got the impression that both the application and the

documentation were understandable to my colleagues. In fact, use of the Heatmappr application has already resulted in two heatmaps appearing in a CBS news item (“Minder Huwelijken, Meer Partnerschappen”, 2016). I was also asked to return to CBS at a later time to present the application to people from other sectors to generate interest in visualizing other phenomena using the same application.

Conclusion

During the 3 months I spend at the CBS department of demography, I had the opportunity to experiment with various visualization techniques and to learn and familiarize myself with Shiny. I acquired and strengthened various skills, both technical and interpersonal.

Technical skills I acquired relate to both the development of interactive shiny applications and their deployment in a professional setting, whereas technical skills I strengthened relate more to data visualization and SPSS syntax and R programming in general. On an interpersonal level I improved my consultation and communication skills, and gave my first practical workshop.

At the end of my internship I delivered a product which is usable by my colleagues to create the kind of visualizations posed at the beginning of the internship project, and which has already lead to two of such visualizations appearing in a CBS news item.

In summary, I spend an enjoyable and informative 3 months at the CBS department of demography, during which time I had the opportunity to acquire and strengthen various skills. At the end of my internship I delivered a product which is now being used by the demography department to create new visualizations. Furthermore, I had the opportunity to experience working in a professional setting for the first time, with a pleasant team of colleagues and good supervision.

References

Cohen, P.N. (2015, July 22). How we really can study divorce using just five questions and a giant sample. Retrieved from: <https://familyinequality.wordpress.com/2015/07/22/how-we-really-can-study-divorce/>

Minder huwelijken, meer partnerschappen. (2016, June 10). Retrieved from: <https://www.cbs.nl/nl-nl/nieuws/2016/23/minder-huwelijken-meer-partnerschappen>

Appendix A: Heatmappr Tutorial

De data

Deze tutorial betreft het analyseren van een dataset over 'bike sharing' in Washington DC. Bike sharing is een vorm van fietsverhuur waarbij fietsen flexibel kunnen worden opgehaald en teruggebracht bij elke standplaats in de stad. Het bedrijf dat bike sharing aanbiedt in Washington, Capital Bikeshare, houdt iedere dag per uur bij hoeveel fietsen er in gebruik zijn. Het bestand *fietsen.csv* bevat een gedeelte van deze data, voor de periode 2011-2012. De eerste kolom geeft de maand van het jaar aan, de tweede kolom de dag van de week. De derde kolom geeft het aantal fietsen wat op dat moment in gebruik was door 'casual users' (mensen zonder abonnement). Het bestand *fietsen_gewogen.csv* bevat dezelfde data maar dan vermenigvuldigd met een wegingsfactor die o.a. corrigeert voor het feit dat niet alle maanden evenveel dagen hebben.

Bron: <https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset> (bewerkt)

Deel 1: Data uploaden

- a. Upload de data door bij **Upload SAV/CSV file** het bestand *fietsen_gewogen.csv* te selecteren.
- b. Heatmappr geeft aan dat er iets niet klopt aan de data. Haal het vinkje weg bij **First column of file contains row names**.
- c. Heatmappr sorteert de rijen en kolommen automatisch op alfabetische volgorde. Aangezien het hier gaat om maanden en dagen van de week is het logischer ze met de oorspronkelijke volgorde weer te geven. Ga hiervoor naar **layout** → **titles and labels** en haal het vinkje weg bij **Sort row and column names**.
- d. Het kan handig zijn om niet alleen de kleuren, maar ook de getallen weer te geven in de heatmap. Ga hiervoor naar **layout** → **cell values** en selecteer bij **Show values** de optie *original values*.
- e. Je kunt de getallen afronden op gehelen door bij **Nr. of decimals** 0 in te vullen.
- f. Je kunt de getallen groter maken door de **Value size (%)** te vergroten naar bijvoorbeeld 200.
- g. Je kunt de kleur van de getallen veranderen door bij **Value color** de naam van een kleur in te typen (in het Engels). Alle door R ondersteunde kleuren zijn beschikbaar. Dit zijn reguliere kleuren als *blue* of *white*, maar ook kleuren als *hotpink*, *mediumspringgreen* en *tomato* behoren tot de mogelijkheden.
- h. Elk getal in de heatmap is het gemiddeld aantal fietsen dat per uur in gebruik is. Bijvoorbeeld: op een maandag in januari zijn er gemiddeld 5 fietsen per uur in gebruik, terwijl er op een maandag in juli gemiddeld 48 fietsen per uur in gebruik zijn. Het betreft alleen fietsen van 'casual' gebruikers: mensen die een abonnement hebben en de fiets gebruiken voor hun woon-werkverkeer worden niet meegerekend.

Deel 2: De heatmap aanpassen

- a. Deze dataset bevat slechts 1 variabele om weer te geven. Als je data een 'risico populatie' en 'events' bevat kun je de optie **edit** → **Show heatmap of...** gebruiken om de andere heatmaps te bekijken.
- b. Bij **edit** → **Transformation** kan je een datatransformatie kiezen. Transformaties kunnen de heatmap verduidelijken, maar ook de interpretatie veranderen. De *root*, *log* en *log1p* transformaties kunnen uitkomst bieden als de data scheef verdeeld is, of als er extreme waarden in de data voorkomen die het kleurcontrast domineren. Als je de getransformeerde waarden wilt zien in plaats van de oorspronkelijke waarden ga je naar **layout** → **cell values** en selecteer je bij **Show values** de optie *transformed values*.
- c. De *scale columns* transformatie schaaft de data zo dat elke kolom een gemiddelde heeft van 0 en een standaarddeviatie van 1. Als je er zeker van wilt zijn dat de kleurschaal ook 0 als gemiddelde heeft zet je een vinkje bij **Center colors around 0**. Met de *scale columns* transformatie kun je bijvoorbeeld zien dat voor elke dag van de week er in de winter een beneden gemiddeld aantal fietsen wordt gebruikt, en in de zomer een bovengemiddeld aantal.
- d. De *scale rows* transformatie schaaft de data zo dat elke rij een gemiddelde heeft van 0 en een standaarddeviatie van 1. Zo kun je bijvoorbeeld zien dat voor elke maand van het jaar de meeste fietsen worden gebruikt in het weekend.
- e. De *scale values* transformatie schaaft de data zo dat *alle waarden in de heatmap samen* een gemiddelde hebben van 0 en een standaarddeviatie van 1. Zo kun je snel zien welke waarden in de heatmap boven gemiddeld zijn en welke onder gemiddeld.
- f. De *rank within row* transformatie geeft binnen elke rij de volgorde van de waarden aan. Zo zie je bijvoorbeeld dat zaterdag en zondag voor elke maand de drukste dagen zijn, en dat daarna maandag en vrijdag drukke dagen zijn.
- g. De *rank within column* transformatie doet hetzelfde voor de kolommen. De *rank* transformatie geeft de volgorde van alle waarden in de hele heatmap.
- h. Met de **Column range** en **Row range** sliders kun je selecteren welke kolommen en rijen je wilt weergeven. Als je een kleinere range hebt gekozen kun je ook de hele range verplaatsen door op het midden van het balkje te klikken en de muis heen en weer te bewegen.
- i. Met de optie **Treat row and column labels as numeric** kun je ervoor kiezen om de namen van de rijen en kolommen als numeriek te behandelen. Dit is voor deze data niet van toepassing omdat de namen van de rijen en kolommen geen getallen zijn.
- j. Met de optie **Remove 0's** kun je cellen met een waarde van 0 leeg maken, maar in deze data bevinden zich geen cellen met een waarde van 0.
- k. Met de optie **Restrict range of values** kun je ervoor kiezen om cellen met (ruwe) waarden onder of boven een bepaalde drempel niet weer te geven. Je kunt bijvoorbeeld cellen met minder dan 10 fietsen per uur verwijderen door het linker bolletje in de slider naar 10 te slepen.
- l. Met de optie **Use left upper matrix only** wordt het rechts-onder gedeelte van de heatmap leeg gemaakt. Dit kan bijvoorbeeld handig zijn als er zich in dit gebied weinig waarden bevinden. Houd er rekening mee dat als het aantal rijen van de heatmap niet gelijk is aan het aantal kolommen dit vreemde resultaten kan opleveren omdat er geen echte diagonaal is.

Deel 3: Kleuren

- a. In het keuzemenu **edit** → **Colors** kun je een kleurpalet selecteren. Je kunt kiezen uit een aantal CBS-huisstijl kleurpaletten, of wit. Een divergent kleurpalet zoals *Red, White, Blue* is vooral geschikt als je zowel positieve als negatieve waarden in je data hebt, of als je de data hebt geschaald. Als je alleen maar positieve waarden hebt kun je meestal beter kiezen voor een kleurpalet van 1 of 2 kleuren. Door wit als kleur te kiezen kan je simpelweg de tabel bekijken zonder heatmap op de achtergrond.
- b. Met de optie **Invert colors** kun je het kleurpalet omdraaien.
- c. Met de optie **Center colors around 0** kun je het kleurpalet centreren rond 0.
- d. Het kleurpalet wordt over de waarden verspreid in een aantal stappen. Standaard gebeurt dit in 100 stappen waardoor de kleurschaal een vrijwel continu verloop heeft. Je kunt ook kiezen voor meer of minder stappen door de **Number of breaks in color scale** aan te passen.
- e. Het verspreiden van de kleuren over de data gaat automatisch. Je kunt er echter ook voor kiezen zelf de breekpunten in de kleurschaal aan te geven. Hiervoor ga je naar **upload** → **Upload custom breakpoints (optional)** en selecteer je een .csv bestand met de breekpunten in een enkele kolom (voor details zie **help** → **uploading data**). Selecteer in de map *heatmappr/examples* het bestand *breakpoints_example.csv* of *breakpoints_unequal.csv* en zet bij **edit** → **Use custom breakpoints** een vinkje om de breekpunten te gebruiken. Het gebruiken van vaste breekpunten kan handig zijn bij het vergelijken van verschillende datasets: door steeds dezelfde breekpunten te gebruiken blijft de interpretatie van de kleurschaal constant.

Deel 4: Clusteren

- a. Naast het sorteren van de rijen en kolommen op alfabetische volgorde kun je er ook voor kiezen de rijen en kolommen te sorteren met behulp van een hiërarchisch clusteralgoritme. Dit kan handig zijn als je op zoek bent naar clusters in je data, of als er geen 'natuurlijke' ordening is voor de rijen of kolommen. Hiervoor kun je de optie **edit** → **Hierarchical Clustering** gebruiken. Clusteren zorgt ervoor dat rijen en/of kolommen die op elkaar lijken bij elkaar komen te staan. De kleinst mogelijke cluster bestaat uit 2 rijen of kolommen. Kleinere clusters zijn onderdeel van grotere clusters en de grootste cluster bestaat uit alle rijen of kolommen.
- b. Kies bij **Transformation** de optie *none*, selecteer de volledige **Column range** en **Row range**, en haal eventuele vinkjes bij **Restrict range of values** en **Use left upper matrix only** weg. Selecteer dan bij **Hierarchical Clustering** de optie *rows*. Heatmappr clusterd nu de rijen van de heatmap. Het dendrogram aan de linkerkant laat de hiërarchische structuur van de clusters zien. Je kunt zien dat o.a. december, februari en januari (winter) een cluster vormen, maar ook juli en augustus (zomer) en mei, juni en september (voor- en na-zomer).
- c. Om de kolommen te clusteren selecteer je bij **Hierarchical Clustering** de optie *columns*. Hier zie je vooral het verschil tussen het weekend en de doordeweekse dagen terug. Door de optie *both (independent)* te selecteren kun je beide clusteringen tegelijk weergeven. De optie *both (identical)* is bedoeld voor heatmaps waarbij de rijen en kolommen dezelfde dingen voorstellen. Onder het menu **advanced options** kun je eventueel de clustering methode aanpassen.

Deel 5: Layout

- a. Onder het menu **layout** → **titles and labels** kun je een titel en labels voor de assen toevoegen.
- b. Het kan gebeuren dat de as-labels en de namen van de rijen en kolommen elkaar in de weg zitten. Dit kan worden opgelost door de **Size of row/column names** te verkleinen of door de **Margins** te vergroten.
- c. Onder **layout** → **color key** kun je de kleurschaal/histogram linksboven een titel en as-labels geven. Je kunt ook de **Key size** aanpassen of de hele kleurschaal verwijderen door het vinkje weg te halen bij **Show color key**. (NB: als je de margins vergroot en de color key weghaalt kan het zijn dat heatmappr de volgende melding geeft: *figure margins too large*. Dit betekent dat heatmappr niet genoeg ruimte heeft op de pagina om de heatmap te laten zien. In dit geval kan het helpen om het venster te vergroten of uit te zoomen op de pagina).

Deel 6: Exporteren

- a. Als je tevreden bent met je heatmap kun je hem opslaan. Onder **export** → **Export image as** kun je het bestandstype kiezen waarin je de afbeelding op wilt slaan. De formaten *.png* en *.jpg* zijn formaten voor rasterafbeeldingen, *.svg* is een formaat voor vectorafbeeldingen. Het voordeel van vectorafbeeldingen is dat ze ook bij inzoomen altijd scherp blijven. Voor *.png* en *.jpg* afbeeldingen is een formaat van 1000 bij 1000 pixels in ieder geval geschikt voor het bekijken van de afbeelding op je beeldscherm. Houd er rekening mee dat voor *.svg* afbeeldingen het formaat wordt gegeven in inches i.p.v. pixels: kies in dit geval dus niet een formaat van 1000 bij 1000, maar bijvoorbeeld 10 bij 10.
- b. Als je het bestandstype hebt geselecteerd en de hoogte en breedte van de afbeelding hebt gekozen klik je op **Download**. Kies bij de daaropvolgende pop-up voor **Opslaan als** om de afbeelding een naam te geven en op te slaan.

Overige data

In de map heatmappr/examples staan nog meer voorbeelden van data sets.