gdv report Simon Luder

Table of Contents

LO1: Visualization basics, chart types ... 2
Nr. 1 Histogram ... 2
Nr. 2 Line Chart & Density map ... 3
Nr. 3 Pie Chart ... 4
Nr. 4 Treemap ... 4
LO2: Visual Perception ... 5
Bertin’s seven variables ... 5
Position ... 5
Hue ... 5
Size ... 6
Shape ... 6
Value ... 6
Texture ... 6
Orientation ... 7
Gestalt Theory ... 7
LO3: Design Principles ... 8
Data Structure ... 8
Visual Structure ... 8
Tools ... 9
Matplotlob ... 9
Plotly ... 9
ggplot2 ... 10
LO4: Grammar of Graphics Tools ... 10
Theory ... 10
Praxis ... 11
LO5: Evaluation ... 13
What we did ... 13
The next step… ... 14
Conducting a Usability-test ... 15
Remote Project repositories ... 16
Useful Weblinks ... 16

LO1: Visualization basics, chart types

In HS19 I worked on the project “Warenkorbanalyse”. The task was to analyze and visualize customer data and to make suggestions for the following three fields.

- Promotion of the organic sector
- Opening hours & workplan optimization
- Store layout design

In this chapter I will reflect some of the different visualization types that were used to examine the above-mentioned sections. For more information visit our full project at: https://gitlab.fhnw.ch/projetkgruppeds2019/warenkorbanalyse_fhnw_ds.git

Nr. 1 Histogram

Category: Opening hours & workplan optimization

Question: The goal was to find out if there are many small shopping baskets where customers can also pay via a self-checkout option, which reduces the workload of the checkout staff.

Research: First, the individual products per shopping cart were added together in the data record. We want to investigate how large the individual shopping baskets are and how these sizes are distributed or in other words we want to visualize a single variable distribution of discrete values. Because histograms are generally used to visualize the distribution of data over a continuous interval, this type of diagram suits perfectly to show that information.

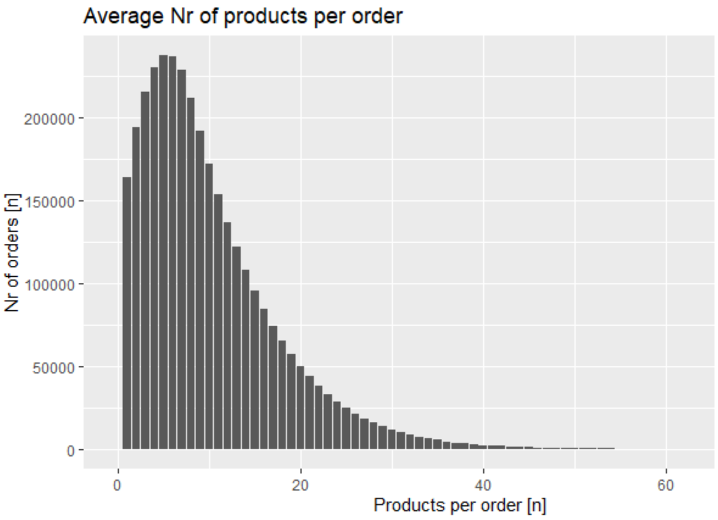


Figure 1. From: Warenkorbanalyse / Einsatzpläne.Rmd

Interpretation: In Figure 1, the x-axis describes the number of different products per order and the y-axis shows the number of orders with the corresponding size. In the histogram you can see that there are a lot of small to medium sized shopping baskets. If you now calculate the quartiles, it turns out that 75 percent of the baskets contain 14 or fewer products and 25 percent contain 5 or fewer products. These 25 percent would be suitable for a self-checkout variant.

Nr. 2 Line Chart & Density map

Category: Opening hours & workplan optimization

Question: It was assumed that more sales are generated at certain times of the day. If this is true, various thresholds can be investigated where too few products are sold and it is therefore not economically efficient to open. In addition, we had the idea that this product revenue could also be used to estimate the workload and corresponding work schedules of the employees.

Research: We multiplied the individual orders by the number of products contained in them and then summed them up per day and hour. This allowed us to show the actual number of items that went over the counter per hour/day, which was the closest approximation to the expected effort and used two different plots: A line plot and a Density map. Line plots are used to show quantitative values over an interval, in this case ratio of products per hour. The missing dimension (Order day) is visualized by using different hues. In this context the visualization can be interpreted as a time series.

Density maps visualize data trough variation in shade and coloring. When used in a tabular format like here, they are useful to analyze patterns or correlations between multiple variables.

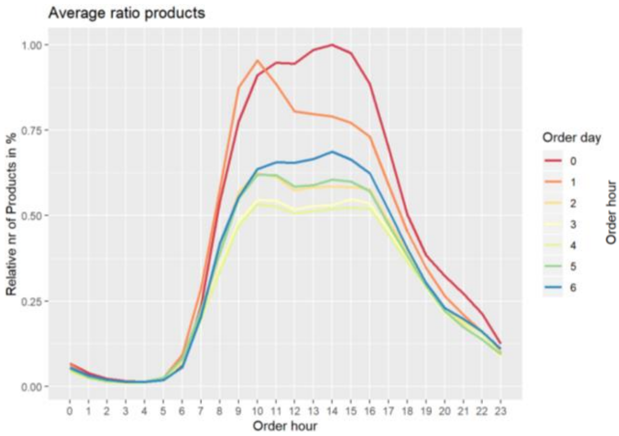


Figure 2. From: Warenkorbanalyse / Einsatzpläne.Rmd

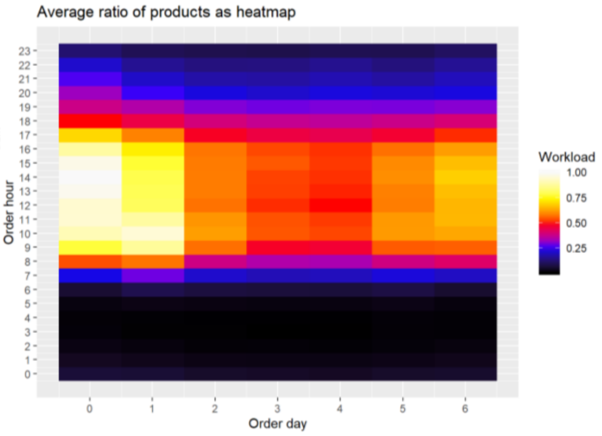


Figure 3. From: Warenkorbanalyse / Einsatzpläne.Rmd

Interpretation: All values were scaled to between 0 and 1, where 0 stands for no sales and 1 is the hour with the most sales. The line plot (Figure 2) is ideal for comparing the individual days, as it clearly shows the change in product sales between the individual hours and days. The separate days and hours can be read relatively accurately. The density map (Figure 3) on the other hand is less suitable for an exact analysis in this use case and was mainly chosen to have an alternative visualization for comparison. Since we want to create a deployment plan which orientates itself on the workload, the heatmap also gives a rough impression on how the plan might look like. One thing that could be improved are the colors chosen in the visualizations by selecting a color palette suitable for colorblind people.

Nr. 3 Pie Chart

Category: Promotion of the organic sector

Question: We wanted to know if there are already a lot of organic products on offer and how well these items sell.

Research: As a first step we examined how strongly the organic section is already represented within our product range. Subsequently, we then investigated whether this ratio is also reflected in the number of organic products sold in order to identify a potential demand. Since we wanted to compare percentages of the whole assortment and we only wanted to compare two parameters, we decided to use pie charts which are an easy and intuitive way to show simple proportions. An advantage is that this chart is easy to interpret for our target group, the store owner. If we had to compare more relationships, a percentage stacked bar chart would have been a better solution. The original vizialisations have been created by Alexandre Rau and overworked by myself.

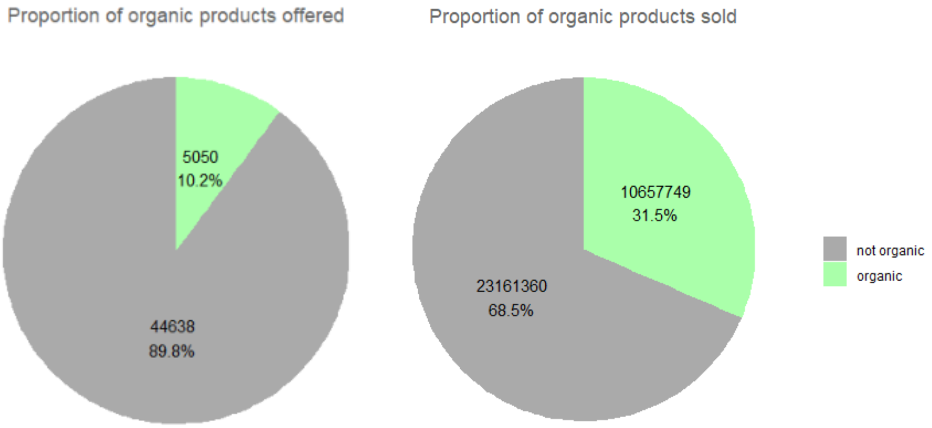


Figure 4 & 5. From: Warenkorbanalyse / Bio-Sparte-Zusammenfassung.Rmd

Interpretation: Figure 4 "Proportion of organic products offered" shows the ratio of organic and non-organic products in the offer over the entire assortment. Figure 5 "Proportion of organic products sold" shows the relationship between organic and non-organic products sold. Although only 10 percent of the products offered are organic, over 31 percent of the sold products are organic. Based on these findings it can now be assumed that there is a fundamental interest in biological products and that it makes sense to conduct further research in this sector.

Nr. 4 Treemap

Category: Store layout design

Question: We wanted to get a vague idea of how much space each department needs in our final store layout.

Research: The different products are hierarchical structured into aisles and departments. Since we have no information about the size of each product, our next best approach was to count the number of products per department. To visualize the product count per department, we had to choose between two different visualizations: A bar chart or a tree map. Because we also have some missing parameters like product volume, we are only able to get an approximated overlook and because all aisles need to be represented in our final layout, we decided that a tree map would be accurate enough and gives a better visual presentation.

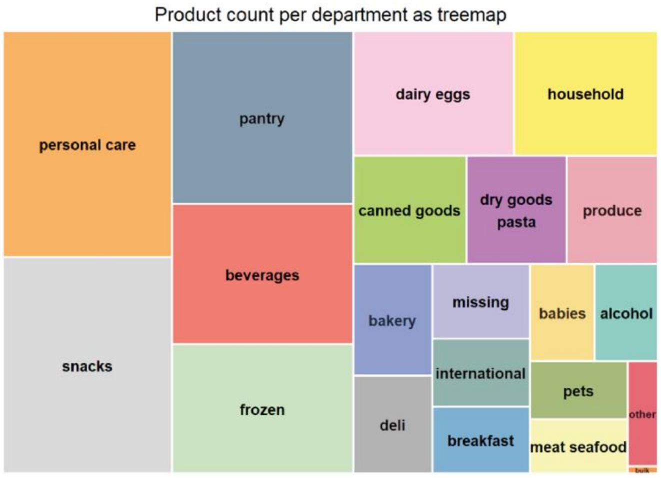


Figure 6. From: Warenkorbanalyse / StoreLayoutDesign.Rmd

Interpretation: The areas of each department corresponds to the number of different products in it. However, these should not be taken blindly, as this plot does not consider the various volumes of the different products. A good example would be the department "snacks". According to our tree map, this department has a lot of products, but it can also be assumed that these products are relatively small on average and therefore would need less space in our final layout

The color of the different departments in this plot is randomly created and has no further meaning. The department “missing” counts all items, which are in neither of the other departments.

LO2: Visual Perception

When it comes to data visualization, you probably think first about what type of graph to use. What is at least as important is the visual design. A clever visual design can support the viewers perception of the data in many ways, e.g. by drawing attention to certain parts of the graphic. It can also be used to bring in multidimensional information into a two-dimensional graphic. A systematization of these different design variables was first created by Jacques Bertin who identified seven main categories of visual variables. In this chapter I’m going to write a short summary of these variables as well as a short summary of "Gestalt theory".

Bertin’s seven variables

Position

The human eye follows the paths, lines, and curves of a design, and prefers to see a continuous flow of visual elements rather than separated objects. Positioning can therefore have significant impact in how we perceive the data. Good positioning allows clear interpretation and can also underline the hierarchy within the data. The positioning can only effectively be changed if it does not alter the data.

Hue

Usually, when representing associative data, dominant colors should be chosen, which are clearly separated from each other in the color spectrum. When generating single color Graphics, brighter colors like red are preferably used to represent heat, magnitude or intensity while colors like blue have a more neutral tone and are more generally used.

When choosing a suitable hue for your visualization, it is important to think about what it is that you want to express within your plot because specific colors can have specific meaning within your topic. A good example of this would be the USA, where the two large parties are usually represented with the same colors. The Democrats in blue and the Republicans in red. If you would choose now to swap these colors or completely change them to two new hues it might be misleading for some viewers, because this color scheme is that deeply rooted within their mind.

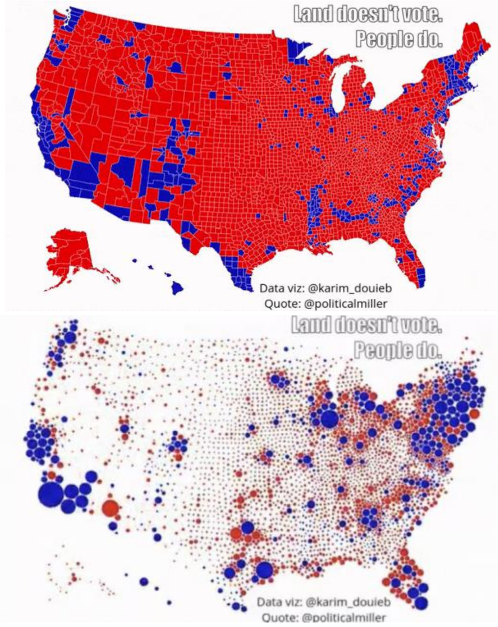


Figure 7. Source: https://www.dropbox.com/s/sfqrenfvs7epuje/election%20map.gif?dl=0

Size

Size describes how much space a symbol or label takes on a map and is often used to visualize quantitative information. Size differences are relatively easy for the viewer to interpret and can be used to draw the attention to specific points. Correct use of size can have a big impact in how we percieve data. Looking at figure 7 again the upper map looks like the republicans (red) has won in a landslide. However if we use bubble size to represent the number of people in each area the visual preception is much closer to the real voting ratio.

Shape

The human eye tends to build relationships between similar elements within a design. Shapes can therefore be used to show associative data, like different attributes on a map or to show different categories. The different shapes should be clearly distinguishable from each other and not confuse the viewer. Shape should not be used to visualize associative data, because it is very unintuitive for the viewer and cannot be interpreted without further information.

Value

Value can be used to visualize the quantity within a category and gets darker with more magnitude or importance. This concept can also be expanded with saturation. The principle there remains the same and goes from white (neutral) to full saturation.

Texture

Texture revers to the aggregate pattern of individual symbols. It can be used in a similar way like Hue, and therefore also mostly used for associative data. A commonly known example would be different street types on a roadmap.

Orientation

Orientation plays a bit part in how we experience data. Orientation can be used to indicate the importance or order in which a visualization is observed. But it can also be used to manipulate representations and thus mislead the viewer. One such example would be this graphic from C. Chan where it seems that gun deaths in Florida have declined since they enacted a new law about gun regulation. On closer inspection, however, it is noticeable that the y-axis has been rotated and therefore the number of gun deaths has, without doubt, increased.

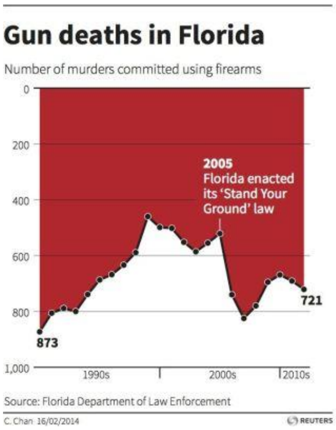


Figure 8. Source: Business Insider/Andy Kiersz/Skye Gould

In the notebook "**Astronaut dataset.html**" I made a few visualizations on this topic. The important variables for the individual visualizations are:

- Visualization 1: Position, Size, Hue
- Visualization 2: Position, Hue, Orientation
- Visualization 3: Position, Hue, Orientation

Gestalt Theory

Gestalt theory was devised in 1920 and aims to understand how humans gain important information from their environment. It consists of a set of laws which describe the natural compulsion to find order in disorder. Below is a short summary for some of them as well as a small example for data science.

- Proximity: When simple shapes are close together, we tend to see them as a group. This is especially important in data analysis, because it is a basic requirement for us to be able to interpret our data visually which is mostly easier than comparing data within a data table.
- Figure/Ground: We try to isolate shapes from backgrounds. Our brain doesn't like uncertainties. If we adapt this to data visualization, it means to clearly distinguish data and information from its background. As an example, use dark and strong colors on bright backgrounds and use bright colors on dark ones.
- Closure: We dislike seeing incomplete information. If there is a gap our brain tries to fill it with what we approximate to be there. However, this also means that with enough base information, we can approximate missing values or future development trough data.
- Common Fate: When we see multiple objects move onto the same direction, we like to think of them as more related, than when they stay in place or move away from each other. Which is nice because it allows us for example to use animated graphs to show change and relations over time.

LO3: Design Principles

Data Structure

Data comes in various forms and storing formats, whereby you can roughly distinguish between three different types. Structured data, where data is highly organized, is usually stored in clearly assigned fields and can be related to other datapoints like for example a relational database. Unstructured data, which is all data that is not organized in a pre-defined manner, like measured values from a sensor. And lastly semi-structured data which does not have the same level of structure and predictability because of the absence of clearly assigned fields, but still provides elements that allow to stricture data hierarchical. Examples of semi-structured formats are JSON, XML and .csv files. Structured data is the easiest to analyze, because its high degree of structure makes it easier to load, extract, upload and query without further transforming the data into other formats.

Unfortunately, when it comes to data analysis around 80 percent of the time invested goes into cleaning, transforming and structuring our collected pieces of information to a useful format so that it can be easily retrieved and read.

In the "Vacancy, no vacancy" project ETH Zurich built a test mock-up to research a modern apartment layout with movable walls and other elements. During the time of one year different volunteer groups are going to live in this setup while their interactions are recorded by sensors. The goal was to check that sensor data for errors and then to display and visualize it in a way that an interpretation of the data is possible without prior technical knowledge. The ETH itself already implemented an API which provided the unstructured sensor values in semi-structured form as a .csv file. We first downloaded the file and used the "pandas" library to transform it into a data frame. Since the test groups change weekly and we wanted to compare their interactions, but also the unique group constellations within each other, we decided to create a relational database, which should make it much easier to perform these tasks without much data wrangling afterwards. We also checked the data for outliers and impossible values and marked them for easier filtering afterwards. That proved to be an essential improvement because there were really a lot of wrong values, which would have been a big problem afterwards. Our final Database then had three different tables: "log_sensor" with all sensor data, "occupation_period" with the different group cycles, and "person" with all the individual participants. The transformation process therefore looks like this:

Unstructured	->	**Semi-Structured**	->	**Structured**
Raw sensor data & Volunteers list	->	.csv file	->	Relational database & Data frames

The project can be found on GitHub via the following link: https://github.com/Lukas113/Vacancy

Visual Structure

While structuring data into meaningful formats makes your live as a data scientist easier, structuring your visualizations in specific optical layouts helps the people to review and analyze your results.

Data visualizations are nowadays consumed from people of all sorts of professional backgrounds. While people with jobs like data scientists and analysts have an eye to dig out key information from even very complex visualizations, other people might the lack experience or simply don't want to invest a lot on time into understanding it. In order to create a visualization that is appealing to as many people as possible, it is recommended to follow certain design principles. I have summarized some of them here:

Visual hierarchy is the order, how the viewer proceeds information. It is important to show information in a logical, natural order which leads the viewer intuitively trough the different visualizations. As an example, in most cultures read words from top to bottom and from left to right. In visualizations people tend to follow similar patterns. The two main patterns here are the F-pattern which is often encouraged in scientific reports like our notebooks in “Vacancy, no vacancy” where you have a lot of context in written format and the Z-pattern which is often how people look over an infographic or a dashboard like the one we created in the “Wettermonitor” challenge.

You can work with size and shape to highlight key information and draw the attention on important parts. Size can also be used to display the relation of an object to another.

Colors and contrast can draw attention. Just like large elements are perceived as more important than small ones, brighter colors catch the viewers eye very easily compared to darker ones. If you use a color palette, avoid the rainbow palette to show quantitative data as it is not related to our numerical system. Also, since about 5 percent of all people, a large proportion of them men, have color vision problems. It might make sense to choose a color palette suitable for color-blind people. A way to implement a scale for colorblind people via the ggplot2 library is described here:

https://rdrr.io/cran/ggthemes/man/colorblind.html

When creating a plot, it is always important to know what the message behind it should be. Especially for scientific visualizations it is useful to focus on one piece of information per plot. Reduce unnecessary information. Focus in the main message and leave not important information’s away which could distract the viewer. This also applies to the plots dimensionality. Creating a 3-dimensional plot might look fancy, but if you don’t really need it, it’s just more difficult for the viewer to interpret.

If you plan to make interactive visualizations, e.g. for exploratory analysis and there is much rendering necessary, it might result in loading time and interrupt the flow of action therefore lowers the user experience. Because of Gestalt theory we know that people dislike missing information, in this case that the program is still running. Here it is important to keep the user informed that the system is loading, e.g. through a percentage bar or other loading symbol. This increases acceptance and ensures that the viewer is more willing to wait without perceiving it as a negative experience.

Tools

There are a lot of different programs and tools to visualize data. I personally worked mostly with different python libraries, but had also one project which was conducted in R. Here I will shortly describe the three visualization libraries I used the most.

Matplotlob

Matplotlib is one of the most popular libraries in python when it comes to two-dimensional data visualization. The library is built on top of the numpy library and works mainly with numpy arrays. It is therefore relatively fast compared to other visualization libraries and allows very easily to create simple graphs. However, it needs more time and effort if you want to visually upgrade your graphics or create animated plots. It is technically possible to create interactive visualizations with matplotlib, but it is more difficult than with other libraries. For me personally it is also less visually appealing than other libraries which is why I primary used it for explorative data analysis at the beginning of my challenges.

Plotly

Plotly.py is an interactive library for data visualization in python, built on top of the JavaScript library plotly.js. It supports over 40 different chart types for statistic, financial, geographic, animations and

3-dimensional use-cases. In my opinion the biggest benefit from visualizations with plotly is the automatically implemented interactivity. However, that can lead to performance issues when plotting large datasets. Plotly's creators also developed a library called Dash, which allows the creation of interactive web-app's in combination with plotly without the need to learn JavaScript or HTML at all. This is also how we created our dashboard in the "Wettermonitor" challenge.

ggplot2

ggplot2 is part of tidyverse package in R. It was used to create most graphics in our project "Warenkorbanalyse". It creates visually appealing graphics with very little code, which makes it ideal for data analysis and publication. However, it is slower than other libraries, which makes you wait some time if you want to plot a lot of datapoints. The library itself also does not include interactive elements.

LO4: Grammar of Graphics Tools

Theory

Grammar of graphics is a tool which allows us to precisely express the characteristics of a visualization. It does this by subdividing the graphic into different parts which then, can be described layer by layer, whereby each layer focuses on a very specific part of the whole visualization. This makes it much easier to create a meaningful graphic, especially for complex visualizations with multiple dimensionalities because it requires the creator to deal with the most important components separately.

The framework was originally proposed by Leland Wilkinson and was revised Hadley Wickham, who is also the main creator of the ggplot2 graphic package. It is therefore no surprise that the ggplot2 library is also implemented that way. The for SPSS, Vega-Lite and many others.

The basic idea behind the revised version, also called the "layered grammar of graphics" is that every visualization can be described and built of the parameters shown in the following graphic:

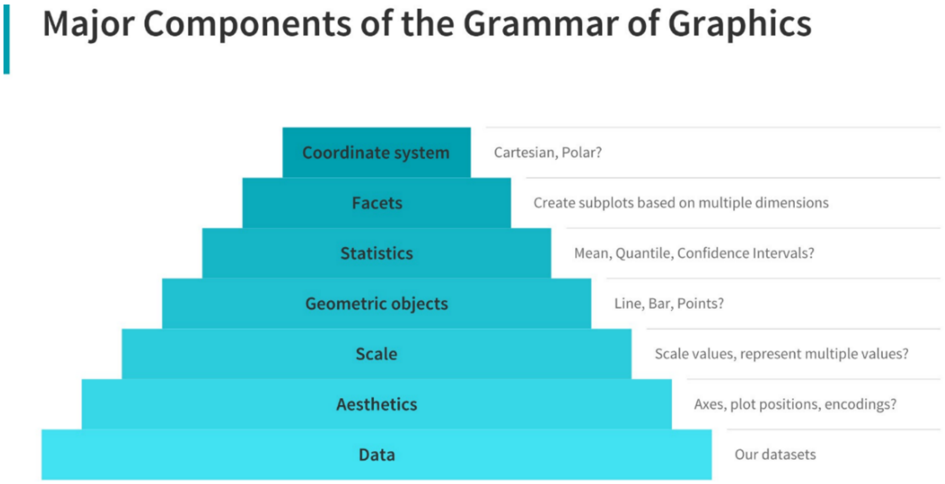

Major Components of the Grammar of Graphics
Coordinate system
Cartesian, Polar?
Facets
Create subplots based on multiple dimensions
Statistics
Mean, Quantile, Confidence Intervals?
Geometric objects
Line, Bar, Points?
Scale
Scale values, represent multiple values?
Aesthetics
Axes, plot positions, encodings?
Data
Our datasets

Figure 9. Source: A Comprehensive Guide to the Grammar of Graphics for Effective Visualization of Multi-dimensional Data

Figure 9 shows seven main layers which are ordered in a pyramid scheme. You start at "Data" and then go up one step at a time. The further up you go, the more specific should be the idea of how the final visualization should look like. We now want to take a more detailed look at the individual layers.

Data: Define the key message of your visualization and what minimal dimensionality of data is needed for visualization.

Aesthetics: Choose suitable axes based on the dimensionality. If you look at the cheatsheet of ggplot2 you will notice that the main grouping of diagrams is based on the dimensionality of the data. If more than 3 dimensions need to be visualized, you can, for example make use of “Bertin’s seven variables” described in LO2 (Color, shape, size, …).

Scale: Think about the type of data used in your visualization. Do you have nominal, ordinal, discrete or continuous data? Do you want to visualize an interval or proportions? Depending on the scale of your data some visualization work better than others or can’t even be used at all.

Geometric Objects: Depending on the scale we now want to use geoms to optimally represent our data in an intuitive way. For comparing multiple characteristics of individual samples, points might be suitable, for timelines as the name indicates line geoms are more intuitive and for counting different categorical variables, bars might be the optimal choice.

Statistics: Are additional statistics necessary for better understanding? For big or complex data, it might support the viewer to add statistical measurements like distribution metrics or confidence intervals.

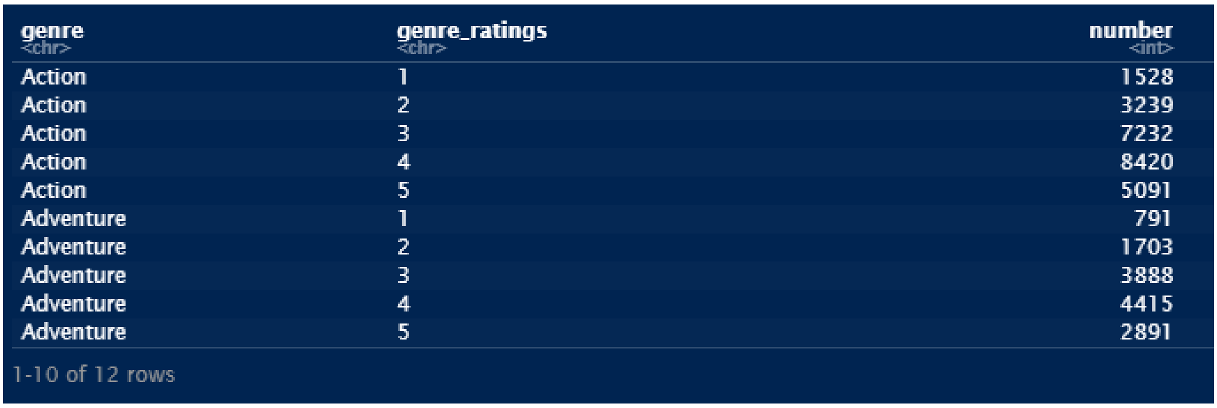
Facets: Sometimes it is necessary to split the data into multiple subplots based in specific dimensions or visualize it in multiple ways to truly understand its meaning.

Coordinate System: When visualizing data most of us are used to think in the cartesian coordinate system, however sometimes a polar coordinate system might be more suitable. The ggplot2 library itself can create cartesian, polar, and transformed cartesian visualizations, as well as maps.

Praxis

Now that we’ve learned this concept in theory, it is time to apply it with some real data.

In the MC “*Collaborative Movie Recommender*” we had to analyze data Movie ratings to create a recommender system. The different movies in the dataset can have 19 different genres and multiple users rated the movies with whole number ratings between 1 and 5. For better overview, the data has already been prefiltered and ordered so that we have a data frame with genres and the corresponding number of different ratings.



genre <chr>	genre_ratings <chr>	number <int>
Action	1	1528
Action	2	3239
Action	3	7232
Action	4	8420
Action	5	5091
Adventure	1	791
Adventure	2	1703
Adventure	3	3888
Adventure	4	4415
Adventure	5	2891

1-10 of 12 rows

Figure 10. From: RSY Minichallenge / Simon_Luder_notebook.Rmd

We now want to visualize the relative ratings per genre and compare the proportions of different ratings using the grammar of graphic framework and the ggplot2 cheat sheet.

Data: We have three dimensions we want to visualize in relation to each other. First the possible rating scale, second the count of number of rating and third are the different genres.

Aesthetics: We want to show our genres at the x-axis and the number of ratings at the y-axis the proportions between different ratings should be visualized with different colors. That means we want to choose our visualization from the "*Two Variables*" subdivision in the ggplot2 cheat sheet.

Scale: We want to show the proportion between the appearance of the 5 different ratings for each genre. We therefore want to scale the y-axis to a range between 0 and 1, where 1 corresponds to 100% of all ratings within this genre. Because we want to have a two-variable visualization with "*discrete"* values on the x-axis and "*continuous"* values on the y-axis, we can narrow the number of suitable plots down to four different geoms.

Geometric Objects: All four visualization meet the technical requirements for visualizing our data correctly. However, depending on context, some are more intuitive than others. Because we want to show the different ratings via a discrete color scale, I would say the "*geom_bar*" visualization fits best to present our message. The benefit with this geom is that it also contains the additional parameter "percentage_stacked" which does automatically relativize our absolute number of ratings.

Statistics: We don't want to implement any statistical measures, as it doesn't support the main story we want to tell with this visualization and the comparison of the mean rating per genre is already part of another visualization.

Facets: The visualization the three parameters should be intelligible enough, so that no further faceting is necessary.

Coordinate system: We want to visualize the data in the cartesian coordinate system. Alternative polar coordinates would also have been an option, which would result in having pie charts. However, this would make it difficult to compare the genres regarding the resulting large amount of pie chart subplots. Stacked Bar plots on the other side work very well for visualizing many sets of proportions.

In compliance of the defined parameters and the data from figure 10 the following visualization results:

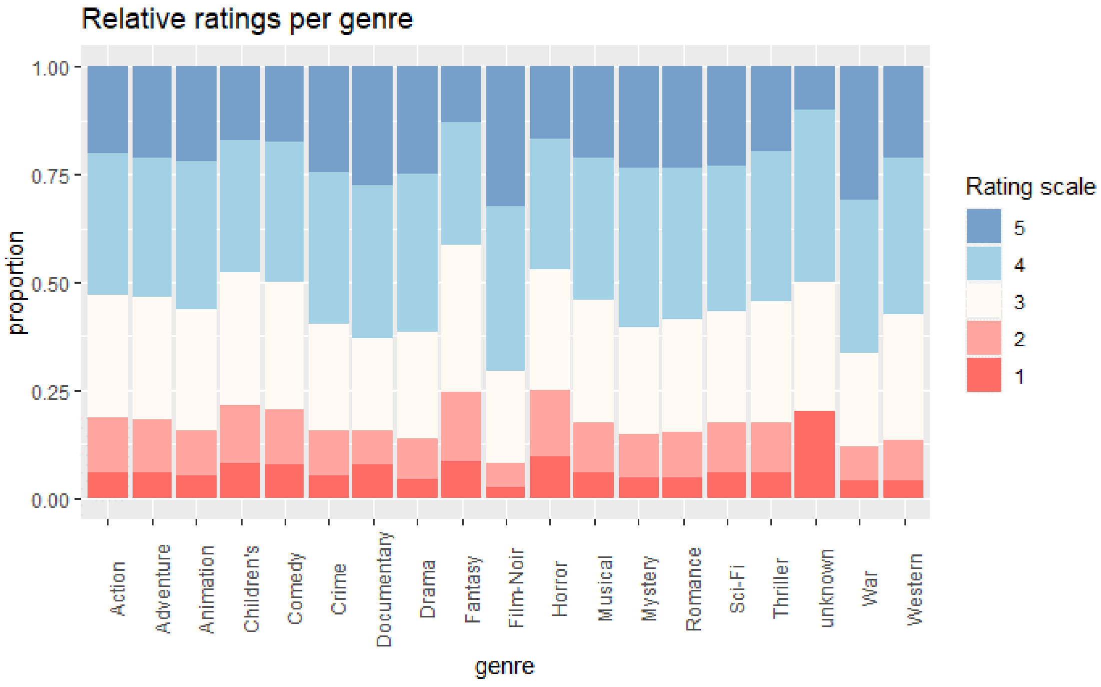


Figure 11. From: RSY Minichallenge / Simon_Luder_notebook.Rmd

The x-axis shows the different genres, the y axis shows the proportional ratings per genre, whereby the color indicates the proportion of different rating per genre. (1=bad, 5=good). With this visualization it is now possible to see the rating behavior from all users distributed by genre. The visualization does not show the total number of ratings per genre as this is already shown in a previous visualization.

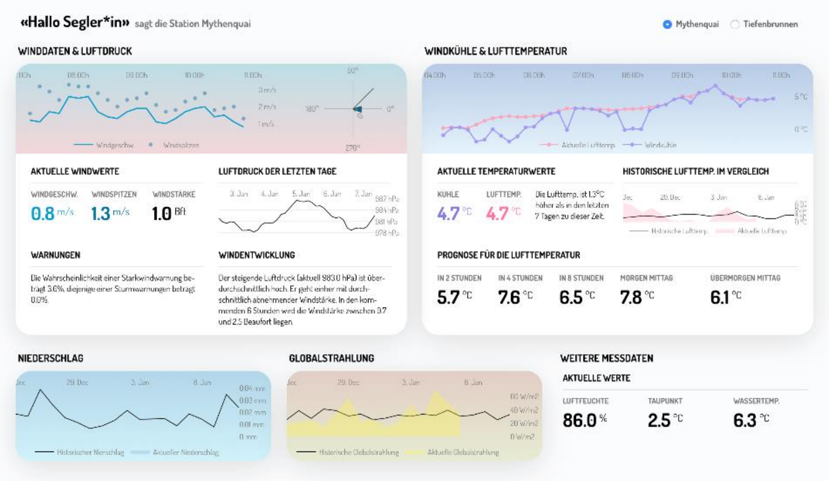
LO5: Evaluation

We have learned some fundamental concepts which can be used for creating high quality visualizations, but how do we know that these really work as intended? A successful visualization depends on the correct usage of data, an optimal visualization technique and the best possible interaction technique regarding the target group. If one of these aspects is not implemented properly people might misinterpret the visualization or are not able to gain full understanding of the presented data. This last chapter now focuses on methods for evaluation and further improvement of existing graphics.

What we did

In HS19 I had worked on the "Wettermonitor" challenge. In the beginning we had focused on creating an MVP, which included several documents like personas and a product vision to help identify our target groups (sailors) and better understand what data is important for them in order to go to their sailing trip. During this phase we were in contact with two people that are actual sailors to ensure our findings were correct and had a certain level of quality. The guideline used for creating the MVP can be found here.

This proved to be very helpful for the further progress in our challenge. In addition to finding out the important data, which is the basis for every visualization, we were also able to learn a lot about the optimal implementation and presentation. When it came to the implementation medium, we had some restrictions like that the final product needs to be on a dashboard screen via raspberry pi. However, we were free to choose the software used for creating the dashboard. There, our two main criteria were that we can create all visualizations which need be presented according to the MVP and the cost of the software which should be free. After some comparison between different software, we choose plotly/dash to be best suitable for implementing our pre-defined visualizations.



«Hallo Segler*in» sagt die Station Mythenquai

● Mythenquai ○ Tiefenbrunnen

WINDDATEN & LUFTDRUCK

AKTUELLE WINDWERTE

WINDGESCHW. 0.8 m/s
WINDSPITZEN 1.3 m/s
WINDSTÄRKE 1.0 Bft

LUFTDRUCK DER LETZTEN TAGE

WARNUNGEN

Die Wahrscheinlichkeit einer Starkwindwarnung beträgt 3.6%, diejenige einer Sturmwarnungen beträgt 0.0%.

WINDENTWICKLUNG

Der steigende Luftdruck (aktuell 983.0 hPa) ist überdurchschnittlich hoch. Er geht einher mit durchschnittlich abnehmender Windstärke. In den kommenden 6 Stunden wird die Windstärke zwischen 0.7 und 2.5 Beaufort liegen.

WINDKÜHLE & LUFTTEMPERATUR

AKTUELLE TEMPERATURWERTE

KÜHLE 4.7 °C
LUFTTEMP. 4.7 °C

Die Lufttemp. ist 1.3°C höher als in den letzten 7 Tagen zu dieser Zeit.

HISTORISCHE LUFTTEMP. IM VERGLEICH

PROGNOSE FÜR DIE LUFTTEMPERATUR

IN 2 STUNDEN 5.7 °C
IN 4 STUNDEN 7.6 °C
IN 8 STUNDEN 6.5 °C
MORGEN MITTAG 7.8 °C
ÜBERMORGEN MITTAG 6.1 °C

NIEDERSCHLAG

GLOBALSTRAHLUNG

WEITERE MESSDATEN

AKTUELLE WERTE

LUFTFEUCHTE 86.0 %
TAUPUNKT 2.5 °C
WASSERTEMP. 6.3 °C

Figure 12. From: Wettermonitor / dashboard.png

In the end, our sailing monitor was completed, and the final examination was conducted by the challenge owner M. Graber who also has experience in sailing. We did not do any further evaluation, firstly because it was not necessary according to the task definition and secondly, the semester had already ended thus we did not have enough time left. However, because our product is based on the criteria for a minimal viable product, it implies that the next step would be an evaluation via user tests for improvement of quality and experience. In the next subchapter I am going to examine how this could be implemented.

The next step…

One popular way to generate unbiased feedback from users is usability-testing. This methodology consists of three main elements. A moderator, a number of tasks which needs to be performed on our test objects and a test-participant. The moderator guides the test-participant trough the different tasks, answers questions and reports the user's feedback regarding the usability and quality of the test product, which is for e.g. in our case the comprehensibility of the visualization. The tasks are realistic activities, like reviewing, understanding or interpretation of visualizations. They should be clearly formulated and need to be the same for all participants. The participant should represent a typical user. It is therefore important to know the target group. Are they in our case also from data-science and statistical backgrounds or do we need to scale our visualizations down for more general language? For representative testing users should have been involved into the project before and the questions must be asked in a neutral context.

The goal of such a usability-test vary, but often include findings like:

- Finding out what data the user really needs to see.
- How well the readability suits the target group.
- Why and for what tasks users need the product.
- How visual representations proposed can be improved in construction.
- Ways the visualization can be manipulated or misinterpreted.

Because this process can get expensive very fast with rising numbers of test-participants and iterations, finding balance between resources and optimization plays an important part. One way of using usability-testing efficiently is by testing only a handful users at a time. Guidelines from nngroup recommend that for a total of five to eight people is already enough people for one review cycle and brings the best result in ratio to the resourced used. The information collected from these test participants is then interpreted and adjustments in the product get implemented. After that the next test-iteration starts with a similar number of test users which have not been part of the iterations before. With each iteration we have identified and removed more and more uncertainties, allowing the test users to engage more deeply into the design which again allows to recognize more usability problems. Of course, sometimes more expensive testing is required, especially when the scientific significance of a product needs to be proven or if multiple competitive designs need to be compared. However, for most use cases the return of investment (ROI) is the best with small groups.

In summary, user testing has the following advantages over other testing methods:

- It is relatively cheap.
- It can be conducted in a very short time.
- It is scalable in scope.
- Unbiased data is generated
- It is more convincing if real user data is generated instead of just more statistical measures.

Conducting a Usability-test

Finally, we created and conducted a short usability-test to review the effectivity of our "Wettermonitor" dashboard. The test was carried out with five test participants. The protocols as well as a short evaluation from the documentation can be found here: https://github.com/SimonLuder/gdv_report_github/tree/main/Usability%20test%20Wettermonitor

Remote Project repositories

Github gdv repository: https://github.com/SimonLuder/gdv_report_github

Warenkorbanalyse: https://gitlab.fhnw.ch/projetkgruppeds2019/warenkorbanalyse_fhnw_ds.git

Wettermonitor: https://github.com/fabianjordi/wettermonitor-fuer-wassersportler

RSY Minichallenge: https://github.com/roman-studer/fhnw-ds-hs2020_recommenderlab

Vacancy, no vacancy: https://github.com/Lukas113/Vacancy

Useful Weblinks

LO1

- Fundamentals of Data visualization: https://clauswilke.com/dataviz/
- Datavisualization catalogue: https://datavizcatalogue.com/

LO2

- Visualization of geographical data: https://volaya.github.io/gis-book/en/Visualization.html
- Gestalt Theory https://www.usertesting.com/blog/gestalt-principles

LO3

- RStudio cheat sheet: https://rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf
- plotly Graph library: https://plotly.com/python/
- Matplotlib library: https://matplotlib.org/

LO4

- Grammar of Graphics: https://towardsdatascience.com/a-comprehensive-guide-to-the-grammar-of-graphics-for-effective-visualization-of-multi-dimensional-1f92b4ed4149

LO5

- Iteration Zero: https://www.iterationzero.works/
- Usability testing 101: https://www.nngroup.com/articles/usability-testing-101/
- User Testing: Why & How: https://www.nngroup.com/videos/user-testing-jakob-nielsen/