M4: Final Submission

Nicole Cherches
Matr. Nr.: 01506832

Alexander Gelb
Matr. Nr.: 01268620

Benjamin Neckam
Matr. Nr.: 01301917

Axinya Tokareva
Matr. Nr.: 01368965

Index Terms: Visualization—Visualization techniques; GAIA—ESA

1 Motivation

On 19th December 2013 the ”European Space Agency”, ESA, launched the ”Gaia”-satellite to gather more information about our galaxy and create a three dimensional map out of it. Since then data like velocities, positions, errors, parallax and many more were collected of about 2 million stars. All together each star has more than 50 features in the data set which means the amount of entries is extremely high.
But working with such a huge data set is not easy and most of the time not necessary because scientists only want to focus on a specific area of the galaxy. Therefore they only use subsets of the data. Now the idea was, to take the whole data set and don't look into detail but see the ”big picture” and try to find patterns, correlations, outliers or other interesting things.
This could be useful for scientists who are not sure which part of the data they want to explore and for example can focus only on the anomalies or other fascinating sections in the data set.

2 Related Work

What we first did, was searching for similar problems and finding out, what other visualizations looked like in this area. Most of the space visualizations we found online, were 3d simulations of the galaxy. Examples of them are the ESA Star Mapper [1] or 100000 stars [2]. They both are 3d interactive simulations of the stellar neighbourhood. We also found a 2d-visualization from the ESA, which visualizes the stars from the Gaia data set [3], also in 3d. But we couldn't find anything that has to do with correlations or patterns. All the existing visualizations are showing the stars themselves with their exact location. So we started prototyping and in the end we even used some of our first basic ideas in our final system. After that, we focused on finding existing visualization systems. In the VIS course we already learned about Tableau [4], which allows the user to import Data and create visualization plots based on it. And we also found out about a software named Glue [5], which helps to create plots and analyze relationships between different data. So we basically tried to create a mixture of both tools and started designing our own system.

Figure 1 is a picture, that shows how we imagined the visualization system at the beginning. As we will see later in the end product, our Lo-Fi Prototype looks similar to our visualization system. We kept the sidebar, where the user has basic information about the data and can choose options on how the data will be presented. Also the diagrams and plots appear all on one screen so the user can see how they correlate. What we didn't do, is creating different plots next to each other. We mixed different plots together into one, so it is easier to see the patterns. Also our first idea was also creating one plot in 3d, but realized later, that we don't need it for our system. However, instead of focusing on the meaning of the data, we focused on exploring the data itself, because we didn't know anything about it.

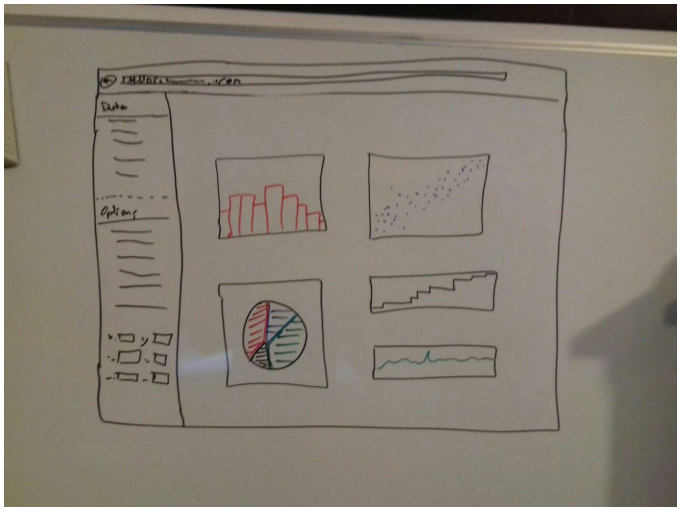

Figure 1: Lo-Fi Prototype of our user interface

3 Approach

When we first thought about our visualization design, we tried to think about our user. The system should help to analyze ”the big picture” of the Gaia data set and not only details. So, we thought about plots that could help us to accomplish that.

Anyway, the best idea was to implement a scatterplot matrix: It fits into our project well, because we can see a whole grid filled with Scatterplots, showing all correlations between selected data values. The user has to choose a set of columns from the data set. The scatterplot matrix contains histograms in the diagonal, so the user can find patterns even better.

4 Implementation

The whole program was written in JavaScript combined with CSS and mainly designed for Google Chrome. For the front-end we decided to use the toolkit ”Bootstrap” [6] which provides different templates so we didn't have to spend too much time for designing the user-interface.
To visualize the data the JavaScript library ”D3” [7] was used which offers the ability to let the developer interact with the data.

We encountered many problems which challenged us to implement things properly. One of the problems were filtering the data, so that we don't have any undefined values, which would cause problems and errors. We figured out that we cannot just delete all the entries, which contain a NULL or any other non valid value, because this would reduce the data set to almost zero entries. Therefore, we went through the documentation of the data set and tried to figure out which features of the stars can be excluded and which not. Unfortunately we could not filter out all unnecessary values at the beginning , we had to do a ”pre-filtering” to remove all non-integer values and afterwards almost every function had to do their own filtering to adapt the data to their specific task.
To minimize the data set, we got the hint to have a look on ”principal

component analysis" which can help to reduce the dimensions. At the beginning we had the idea to implement a plugin [8] for D3 which processes the input data first and then work with it. But after implementing successfully we recognized that the produced result was not what we had in mind. It was built for data sets with less dimension than we have and therefore the performance was tremendously slow. So we focused on "DimStiller" [9] and tried to work with it. Basically it almost fulfilled our requirements but unfortunately it doesn't offer an API to connect it to our application. Due to time issues, we weren't able to create a pipeline, where the input file is send to DimStiller, which processes it and then sent back to our application, where the user can work with a trimmed file. Although it is not realised yet, we were curios how the result of DimStiller would look like and played around with it a little bit and decided to present the outcome here.

DimStiller's principal component analysis consists of five so called steps, which happen consecutively. These are "Cull:Variance", "Data:Normalize", "Collect:Pearson", "Reduce:PCA" and "View:SPLOM". The first interesting and a little bit disappointing result appeared at "Collect:Pearson", which shows the correlation between the dimensions. It reduced the dimension from 52 to 28 dimension. When we look on figure 2, we see that most of the dimensions do not correlate, although we expected more correlation, since the amount of data is enormous.

At the "Reduce:PCA" step, one can choose interactively the final number of dimensions via the user interface, which is shown in figure 3. The problem here is, that most dimensions are combined and we are not sure, how much and if there is an information loss. Due to this fusion the old names of the columns are discarded and get new names like "S4.D1", like we in figure 4 on the x-axis. Furthermore we can see that the picture concerning the correlations didn't change, there are no or just slight correlations among the new dimensions.

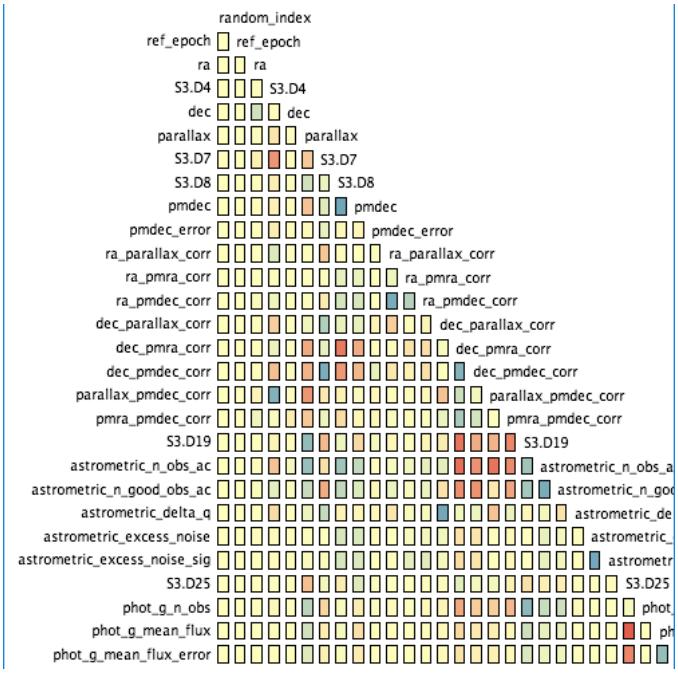

Figure 2: Pearson coefficient, where blue means positive correlation, yellow no correlation and red negative correlation

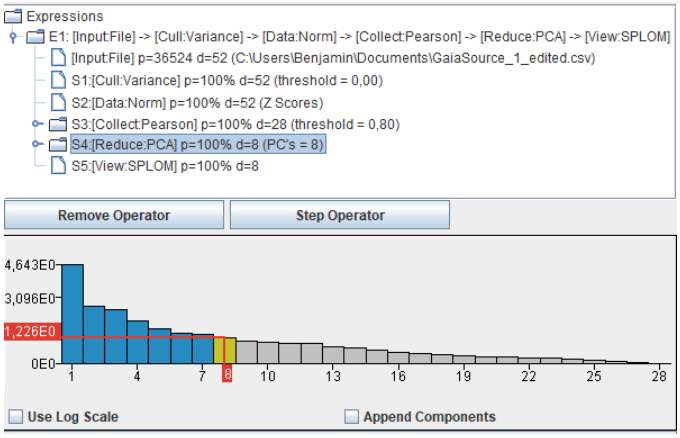

Figure 3: PCA step

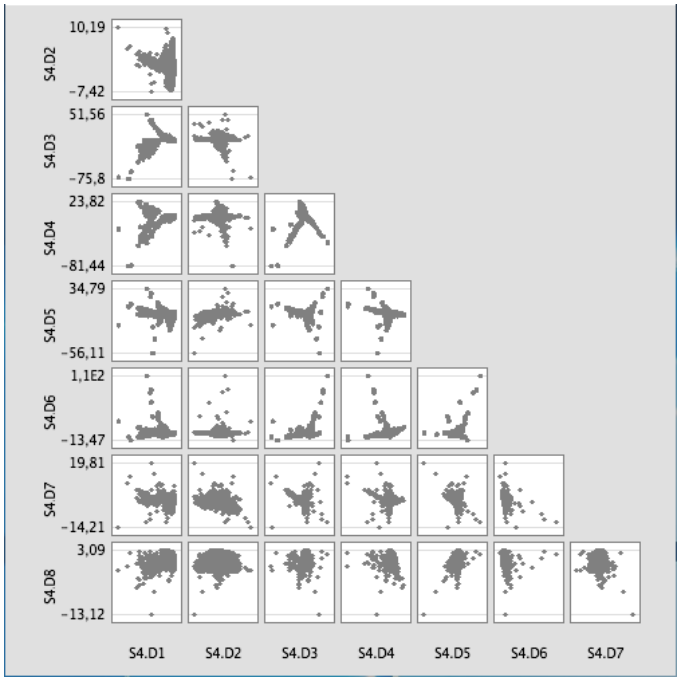

Figure 4: Scatterplot matrix with reduced dimension to 8

As correlations are a major part in this project, we needed a way to compute correlations between columns in our big data set. So we used the formula for the correlation coefficient. [10]

$$\rho_{xy} = \frac{\sum_{i=1}^{n}((x_i-\bar{x}_n)(y_i-\bar{y}_n)}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x}_n)^2\sum_{i=1}^{n}(y_i-\bar{y}_n)^2}}$$

This function is used for the scatterplot matrix. The hard part about this, was that the formula is long and nested, so it was hard to compute it in D3. Also it was difficult to debug this function because of the size of the data set. Another challenge, was the right filtering of our data. For computing the correlation, we had to filter out empty entries or NaN values without modifying the data set too much.

Also the picture of the visualization as a whole has changed. Earlier in the options, it was possible to select the type of the plot separately, but in the end it was decided, that presenting all graphics

in one representation was the more correct solution, since it is more convenient and obvious for our potential user. It was quite difficult to find the correct bin size for histograms for this huge data set. We decided to use the following formula, which is an alternative to Rice Rule:

$$n^{\frac{1}{3}}$$

Another problem was the scale. We needed to add axes, so that they fit into histograms and scatterplots but are independent from cells with correlation values. As a result we decided to leave the x- and y-axis for the scatter plot and just the x-axis for the historgrams. The y-axis values of the histograms are presented by a tooltip. Furthermore the values of the correlations are rounded so it is easier to read, but if one wants the more precise result, it is possible to get these by a tooltip too. Tooltips are also used to state the names of the correlated columns. The only big problem in this part was, that it took a long time to understand how to make this presentation understandable and to implement it.

5 Results

The first thing the user will probably look on, is the information about the Gaia data set. The name, number of rows and columns is displayed on the top left in the sidebar.

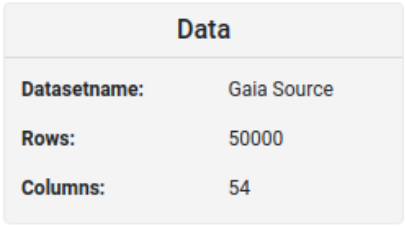

Figure 5: data set overview

This sidebar contains also the Options: Here the user can choose the Columns, he wants to analyse. He can select from all the columns in the Gaia data set. In this scenario our user wants to see the correlations between the columns ”ra”, ”dec”, ”astrometric_n_obs_al” and ”l”.

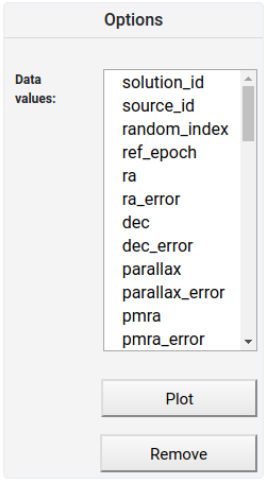

Figure 6: Options

Then he presses ”plot” and the Scatterplot Matrix is displayed on the site.

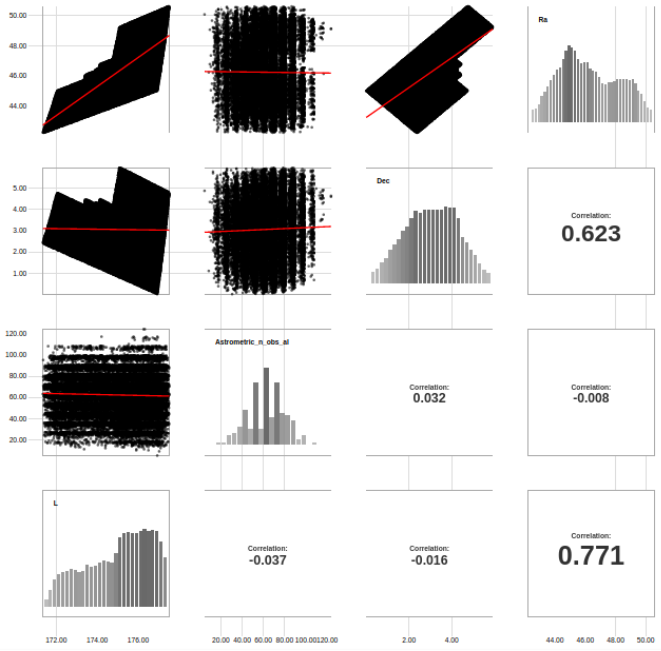

Figure 7: Scatterplotmatrix

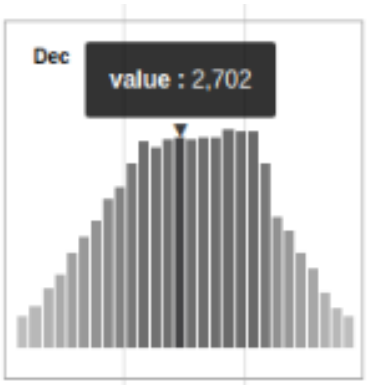

Figure 8: Cell with Tooltip function

As said before, now he can see all the columns at a glance in one big view. The correlations between all the columns are computed and displayed as Scatterplots with regression lines in the upper half of the Matrix, the values in the lower half. In the diagonal he can see histograms showing the distribution in the columns, also they have the column names in the title. When hovering over the bars, he can read the exact value of the bar. Hovering over the correlation value, displays the two column names, which where analysed. To clear the view, he can press ”remove” and select new columns to plot. Or he just plots a new matrix underneath the old one.
Also we used brushing to highlight the same points in different scatterplots (figure 9).

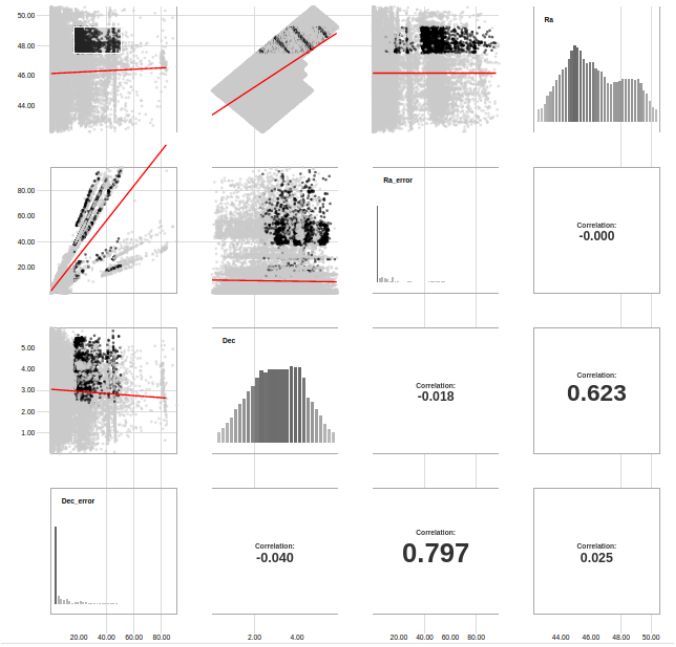

Figure 9: Brushing

5.1 Performance

The number of the chosen attributes is the dimension of the scatterplot matrix, while the dataset size is the number of rows of the csv file. The file has always the complete 58 columns, of which three columns are filtered, because they do not have numerical values. As expected, the performance decreases with the size of the dataset and the attributes. For this evaluation we used just datasets with 1000, 5000, 10000, 50000 and 100000 rows, because the performance is too bad with the original 2 billion rows file.

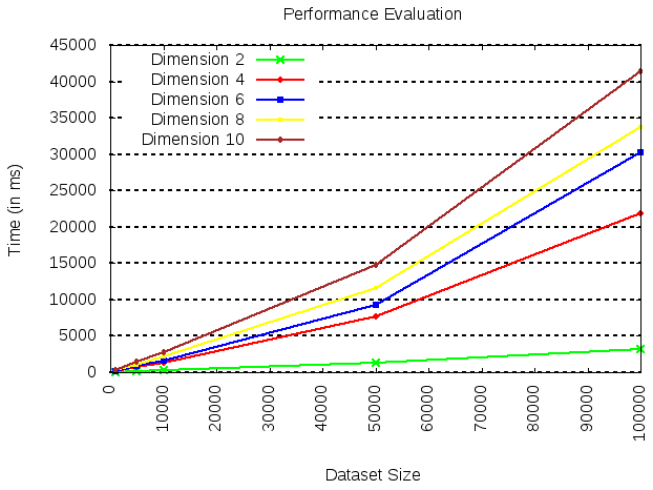

Figure 10: Performance Evaluation

6 Discussion

The main strength of our visualization project is, that the user has a fast overview, where he can see all the patterns and has ”the big picture” of the data set. With the Scatterplot Matrix, he already can see how a set of columns correlates with each other, but that’s not it: There are also histograms in the diagonal of the Matrix, which help understanding the data better. Also we implemented a regression line for the scatter plots. We are using tooltip for displaying further information about plot objects and a filter, so the user can manually choose what he wants to see and ignore the rest. The visual strengths of our project are the simple design, because we only use few colours (”Less is more!”). The weakness of the design is the weak performance. The size of the data set is a big problem, so it takes a lot of time to display the plots with so many points with Javascript and D3. Also we had a time management problem and couldn’t achieve all that we wanted to do in this project (e.g. the PCA).

6.1 Lessons learned

6.1.1 Nicole Cherches

There were a lot of things, I learned from the Gaia project. First of all, I would like to talk about the technical aspects: As this was my first visualization task, it was a whole new experience for me to work with d3 and Javascript. But the most important part is, that I learned how to work with a huge data set and how to get the most out of it. We first tried to learn everything about it, also we realized that Astronomy is too complicated for us to understand. After our talk with Mr. Moeller, we then tried another approach: We did a data exploration task, which is all about getting to know a huge data set, we don’t know much about. Another important part, I learned from this project, was working in a group. I learned a lot about coordination, planning and distributing tasks. This is also a good lesson, because collaboration with others is a necessary skill in computer science.

6.1.2 Alexander Gelb

For me the most interesting part of this project was, to learn how to visualize a big data set with the d3 library. It was very interesting to work with such a huge set with different information about the stars in our galaxy, altough we are not familiar with the values inside the gaia data. I was very excited about working with Joao Alves and Irati Larreina, who are both operating in the domain of astronomy. I learned a lot from Irati about this domain and together with my project team and Irati we produced a nice little app which shows hopefully useful information for her future work.
The most difficult part for me was to fix the performance problem. For this I still was not able to solve this problem, but maybe D3 is not the best option to handle such a big amount of data.

6.1.3 Benjamin Neckam

There were two things I will take out of this project.
The first one refers to the implementation and the chosen programming language. Basically I was not very used to JavaScript, to be more precise, I had no experience with it, but I was really impressed by ”D3”. The opportunities you have to visualize data are brilliant and allow the programmer to choose over a wide variety of plot types. The only disadvantage that really falls into weight was the performance. If the data set is bigger than, let’s say, 100.000 entries it takes some time to calculate and plot the data and in our case the file contains more than ten times more entries. But all in all it was very interesting and in my opinion very important for my future to work with JavaScript and ”D3” and get deeper into this part of front end programming.
The second part is about the collaboration between the project group of the class and research group of another university. Until now all the projects at the university were from the teachers of the faculty itself but in this case the task was from the ”Universittssternwarte Wien”. This means, we had to arrange meetings with the project leader and a student which also is a member of the research group and talk about the task in detail and what the aim of this project is. It was fun and interesting to meet all these people but I recognized that,

even if there are not too many people involved it can easily happen, that confusion arises. Like in our case, we started to focus on the wrong details and parts and had to ”restart” our project. Therefore, I would say communication is the most important part in a project, especially if it is a collaboration of two or more groups, I guess this is the biggest lesson I learned from the project.

6.1.4 Axinya Tokareva

I would like to take out a few positive points. First of all, it was a good experience with JavaScript in general, and with the D3-library. Surprisingly, the D3 was quite convenient and easy to understand for data processing and visualization, but its performance is not a good side.
Secondly, I worked with such a huge data set for the first time. It was interesting to understand how to work with it and what to focus on.
Third, I think working in a group is one of the important lessons I learned. I comprehended how important it is to plan your time and distribute the work to the group members in time.

7 Separation of Tasks

Nicole Cherches: Report, Corrrelation analysis
Alexander Gelb: Scatterplots, Regression line
Benjamin Neckam: Report, Principal Component Analysis
Axinya Tokareva: Histograms, insert correlations in scattermatrix, Tooltips, Scales

References

[1] Jan Willem Tulp, Jos de Bruijne, Karen OFlaherty, and Claudia Mignone. A visualisation based on data from the european space agency's hipparcos star mapper. http://sci.esa.int/star_mapper. Accessed: 17.01.2018.

[2] Google Data Arts Team. Interactive visualization of the stellar neighborhood. http://stars.chromeexperiments.com/. Accessed: 17.01.2018.

[3] Andr Moitinho de Almeida, Hlder Savietto, Carlos Barata, Alberto Krone-Martins, Mrcia Barros, Antnio Falco, and Tiago Fernandes. Interactive visual exploration environment for the gaia archive. https://gea.esac.esa.int/visualization/index.html. Accessed: 17.01.2018.

[4] Tableau Software. Interactive data visualization product. https://www.tableau.com/de-de. Accessed: 17.01.2018.

[5] Chris Beaumont, Thomas Robitaille, and Michelle Borkin. Python library to explore relationships within and among related datasets. http://www.glueviz.org/en/stable/. Accessed: 17.01.2018.

[6] Twitter Inc. Open source toolkit for developing with html, css and js. https://getbootstrap.com/. Accessed: 18.01.2018.

[7] Mike Bostock. Javascript library for visualizing data with html, svg, and css. https://d3js.org/. Accessed: 18.01.2018.

[8] Yosuke Onoue. Principal component analysis plugin for d3. https://github.com/likr/d3-pca. Accessed: 01.12.2017.

[9] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Mller. Dimstiller: Workflows for dimensional analysis and reduction. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pages 3–10, Oct 2010.

[10] Hendrik Schmidt. Empirische kovarianz; empirischer korrelationskoeffizient. http://www.mathematik.uni-ulm.de/stochastik/lehre/ss03/wirtschaftsstatistik/skript9/node21.html. Accessed: 21.01.2018.