Prof. Dr. Thomas Schultz
Mohammad Khatami (khatami@cs.uni-bonn.de)
Ikram Jumakulyyev (ijumakulyyev@cs.uni-bonn.de)

University of Bonn
B-IT/Computer Science II
May 22, 2020

Summer term 2020

# Visual Data Analysis

**Assignment Sheet 6**

Solution has to be uploaded by June 1, 2020, 8:00 a.m.
to https://uni-bonn.sciebo.de/s/94j4BY6iEzK3U9H with password `vda.2020`

Please bundle the results (as PDF) and scripts (*.py/*.ipynb files) in a single ZIP file. Submit each solution only once, but include names and email addresses of all team members in the PDF and each script. Name the file `vda-2020-xx-names.zip`, where `xx` is the assignment sheet number, and `names` are your last names.

If you have questions concerning the exercises, please write to our mailing list:
vl-scivis@lists.iai.uni-bonn.de.

## Exercise 1 (Graph Visualization, *15 Points*)

In this exercise you will learn how to visualize graphs in Python. We suggest the Graphviz package, which was used for our own example solution. In case you should already know and prefer an alternative Python package that allows you to perform the same tasks, you are free to use it.

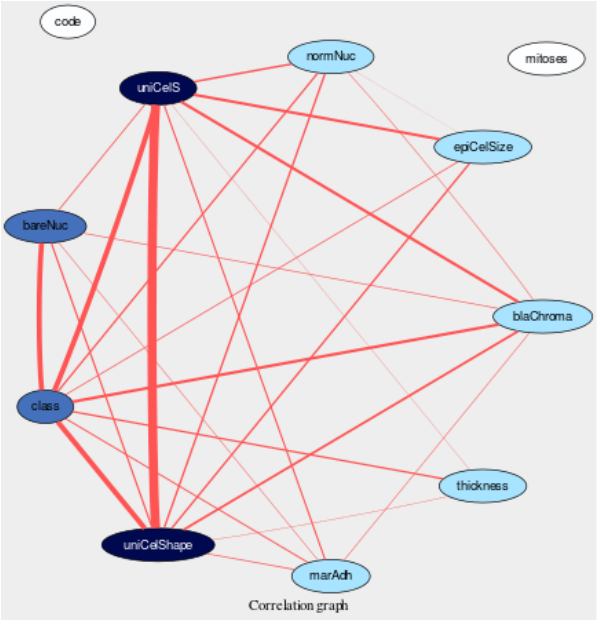


Figure 1: Variable correlation graph.

a) Use the Breast Cancer Dataset dataset `breast-cancer-wisconsin.xlsx` and fill in the missing values. Then compute the Pearson correlation between any pair of variables, and store them in a matrix. (2P)

b) Create a graph from the correlation matrix and visualize it with a force-directed layout. Represent each variable as a node in the graph. Insert an edge between two variables whenever the Pearson correlation between them exceeds the threshold $\rho > 0.6$. (4P)

c) Modify the visual attributes of edges to reflect the magnitude of the correlation. (3P)

d) Produce an alternative visualization with a circular layout. Color the nodes so that there are four set of nodes, one color for having at least one correlation more than 0.9 to other nodes, another for having at least a correlation $0.8 < \rho_{\max} <= 0.9$, one for having a correlation $0.6 < \rho_{\max} <= 0.8$ and the last for the remaining nodes. (3P)

e) Answer the following questions:
   - At the selected threshold, which nodes are disconnected from the rest of the graph and what do they indicate? (1P)
   - If two nodes A and B are strongly correlated, and node C is strongly correlated with node B, can we conclude that node C will be also strongly correlated with node A? (1P)
   - Based on the visualization, which variables would you propose to predict the class? (1P)

## Exercise 2 (Large-Scale Graph Visualization, *10 Points*)

The node-and-link diagrams that were discussed for graph visualization in the lecture do not scale well to large graphs. In this task, you will read about an alternative visualization approach that has been proposed for such cases, based on visualizing an adjacency matrix representation instead. The corresponding paper `elmqvist-zame-2008.pdf` is available from the lecture webpage.

Please answer the following questions in your own words. Remember that **we will not grant even partial credit for copy-pasted text.**

a) The ZAME visualization tool uses a specific hierarchical data structure for storing graphs at multiple scales. At the lowest level, four integers are stored per vertex, and either six or four per edge. What do these integers describe? Which two are optional in case of the edges, and what is their purpose? (3P)

b) The zoomable edge table stores edges in a particular order that makes it fast to search for an edge given its vertices. Write efficient pseudocode that returns the index within this table of an edge connecting vertices $u$ and $v$, and returns `None` if the table does not contain such an edge. (4P)

c) In the pseudocode listed in the paper's Figure 4, some modifications are highlighted in boldface, on lines starting with a bar. What is the purpose of these modifications? (2P)

d) What is the difference between geometric zoom and detail zoom in the system? (1P)

## Exercise 3 (Parallel Coordinates Plot in Plotly, *10 Points*)

In this task, you will become familiar with plotly. Plotly is a plotting library with bindings for different programming languages. It enables the creation of interactive plots based on HTML, CSS and javascript. An introduction to the parallel coordinates plot with plotly can be found here: https://plot.ly/python/parallel-coordinates-plot/

a) Read the file `Data_Cortex_Nuclear.xls` that we previously used in Exercise 1b) of Sheet 5. Extract subgroups t-CS-s and c-CS-s. Use plotly to create a parallel coordinates plot from the following 5 proteins: (pPKCG_N, pP70S6_N, pS6_N, pGSK3B_N, ARC_N). Assign different colors to the two selected classes. Annotate every axis with the correct protein name. (9P)

b) Explore the data by interacting with the parallel coordinates plot. Do you find anything suspicious about the data set? (1P)

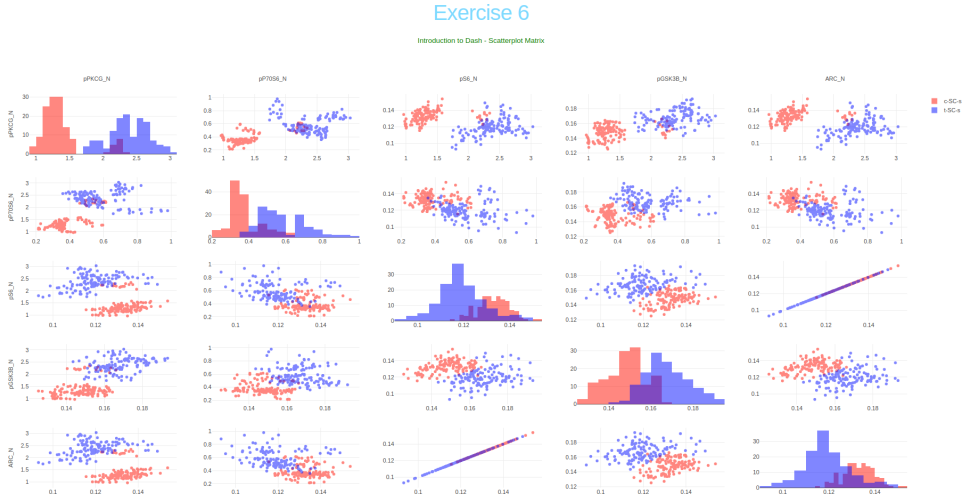


Figure 2: Scatterplot matrix created within the Dash framework

## Exercise 4 (Scatterplot Matrix in Dash, *15 Points*)

The main purpose of this task is to make you familiar with dash, a productive Python framework for building web applications. Dash is ideal for building data visualization apps with highly custom user interfaces in pure Python. An introduction and tutorial to the Dash framework can be found here: https://dash.plot.ly/.

a) Use the same dataset, subgroups, and proteins as in Exercise 1. Use the dash framework to create a $5 \times 5$ scatterplot matrix, for which the diagonal plots represent the histogram of the corresponding attribute. Assign a separate color to each class and annotate the axis. (6P)

b) In the non diagonal cells, visualize the scatterplots of corresponding pair. Assign corresponding name and text to each sample on scatterplot, so that the exact values of a point and its class are shown when moving the mouse over it. (9P)

*Note:* Dash builds on plotly, and it is possible to achieve this visualization with plotly alone. However, please use this opportunity to learn about dash, since we will use some of its advanced features on the next sheet.

# Good Luck!