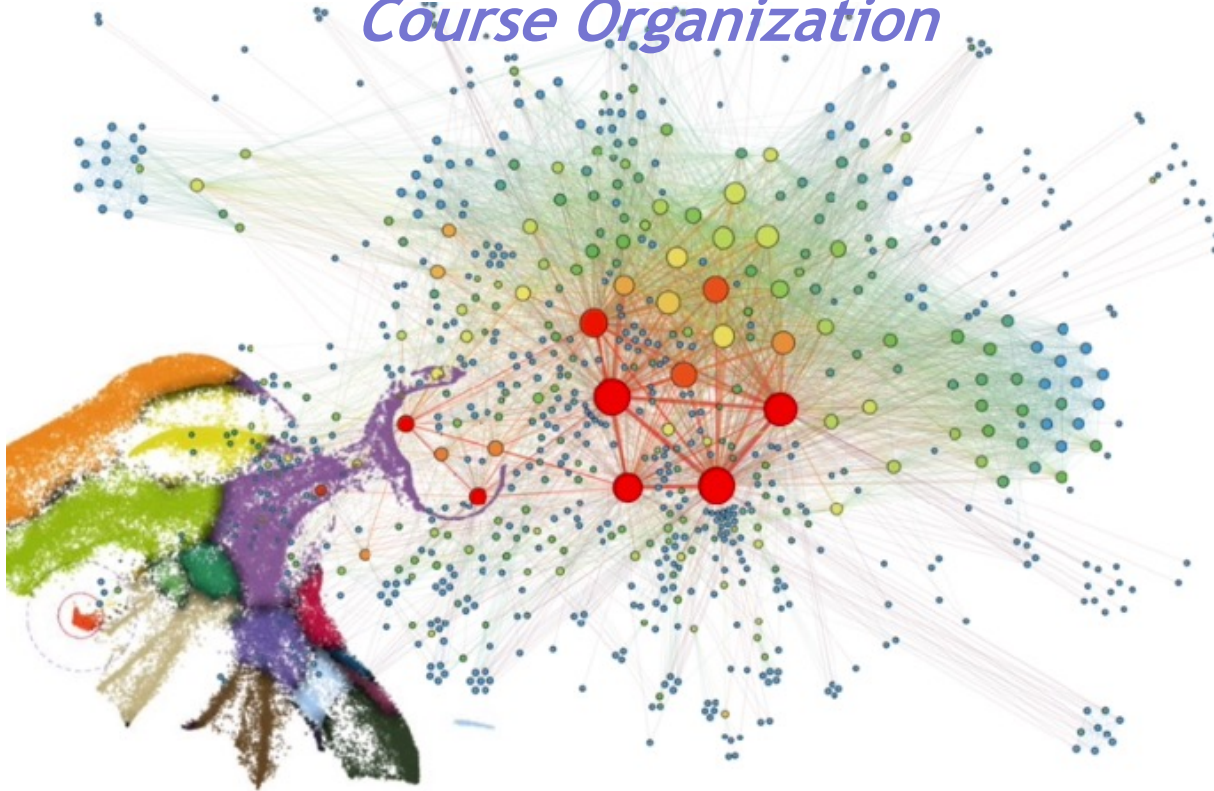


# (High-dimensional and Relational) Data Visualization

## *Course Organization*



**dr. Tamara Mchedlidze**

**prof. dr. Alexandru (Alex) Telea**

Department of Information and Computing Sciences  
Utrecht University, the Netherlands

# Introduction - Instructors

## Tamara Mchedldize (Mtsentlintze)

- PhD in graph drawing (NTUA Athens, 2012)
- PostDoc in network visualization (KIT Karlsruhe, 2012-2020)
- assistant professor (UU Utrecht, since 2020)
- research: graph drawing (theory, practice, ML), applications of datavis in humanities
- check: <https://evm.science.uu.nl/>



<https://www.uu.nl/staff/Mtsentlintze1>

## Alex Telea

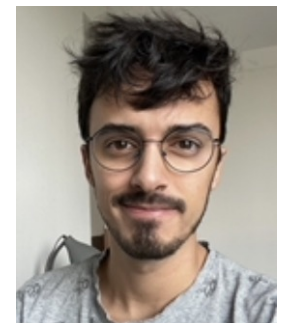
- PhD in scientific visualization (TU Eindhoven, 2000)
- assistant professor in visualization (TU Eindhoven, 2000-2007)
- professor in multiscale visual analytics (RUG, 2007-2019)
- professor in visual data analytics (UU, since 2019)
- research: visual analytics, shape analytics, machine learning



<http://www.staff.science.uu.nl/~telea001>

## Alister Machado dos Reis (tutorials)

- Bachelor in Computer Science (UFRGS, Brasil, 2017)
- Master's in Data Science (INP, Grenoble, 2017)
- Software Engineer (Google Brasil, 2018-2021)
- PhD student (UU Utrecht, since 2021)
- research: interactions between Visualization and Reinforcement Learning



<https://www.uu.nl/staff/AMachadodosReis>

## For *whom* is this course

- (under)graduate students in the MSc phase
- interested to learn in depth algorithms and methods for data visualization

## What you need to know *before*

- a programming language (C, C++, Python, Java, C#)
- data/software management (platforms, tools, scripting, etc)
- background in math (linear algebra, calculus, optimization, statistics)
- knowledge of algorithms and computational complexity
- fundamentals of visualization and/or graphics

# Course Structure

## Lectures

- 7 lecture weeks
- communicate and illustrate theory, techniques, methodology
- present examples
- interactive setting: slides, questions to\from audience (!)

## Tutorials

- develop a tool for relational, high-dimensional data visualization
- work in groups (1..3 students); **groups of 2** are encouraged
- assignment is online (assignment -> Overview)
- work on the assignment as the course progresses (25% of the grade, more later)

## Feedback

- global, during lectures
- per-group, from lecturers, upon submission of progress reports

# Course Structure

## Grading (more later)

- **25% Process**
- **25% Final project presentation**
- **50% Final project deliverable**

# Course Outline

**Module 1: Introduction**

**Module 2: Tree visualizations**

**Module 3: Visualization of general graphs**

**Module 4: Visualization of directed graphs**

**Module 5: Visualization of clustered/multilayer networks**

**Module 6: Low dimensional data visualization**

**Module 7: High dimensional data visualization**

**Note:** A module is not 1-to-1 to one lecture

# Course Schedule

Week	Date	Content	Lecturer
6	February 8	Introduction to Network Visualization	Tamara
7	February 13	Tutorial 1: Read and draw a graph	Alister
	February 15	Tree Visualizations	Tamara
8	February 20	Tutorial 2: Extract and draw a tree	Alister
	February 22	Visualization of general graphs	Tamara
9	February 27	Tutorial 3: Implement force-directed	Alister
	March 1	Visualization of hierarchical graphs	Tamara
10	March 6	Tutorial: Implement hierarchical layout	Alister
	March 8	Lecture-free week	
11	March 13	Tutorial: additional task	Alister
	March 15	Visualization of multilevel networks	Tamara
12	March 20	Tutorial: multilevel and bundling	Alister
	March 22	High-dimensional data visualization: basics	Alex
13	March 27	Tutorial	Alister
	March 29	High-dimensional data visualization: advanced	Alex
14	April 3	Final Presentations	Students
	April 5	Final Presentations	Students

**Note:** MSTeams -> General -> Files

# Assessment

## Practical assignment

*Design and implement a system for the visualization for relational and high-dimensional data*

### 1. Progress report updates (25% of the grade)

- describes implementation choices, parameters of the algorithms, presents output visualizations, qualitative and quantitative evaluation of the visualizations
- submit progress of the report every week - same updated document
- use github for collaborative work both for code and progress report
- submit progress report every week (on Monday) to MSTeams (Tutorials -> Files)
- we grade: consistency, quality, and completeness

### 2. Final project deliverable (50% of the grade)

- PDF of the final report (completeness of the assignments and explanations, quality of writing)
- GitHub repository with code (not graded separately, but must work to pass course)
- Readme + built instruction

### 3. Final presentation (25% of the grade)

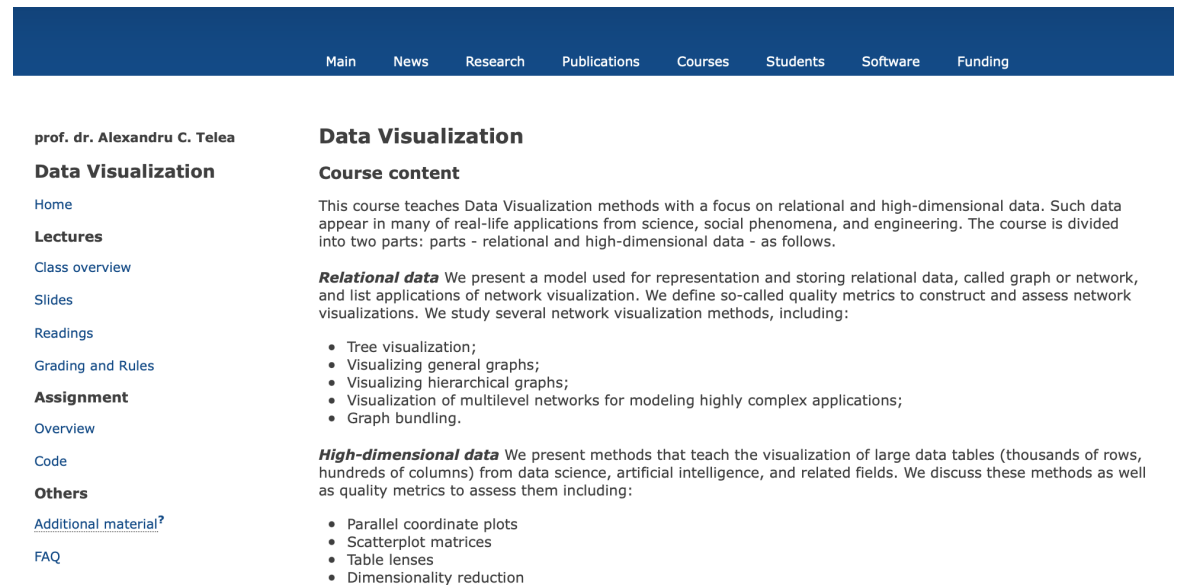
- present orally your system
- clarity and structure of the presentation, quality of the slides, engagement with the audience, addressing questions



# Communication Channels

## Course webpage

- slides (as the course progresses)
- recommended books/articles
- datasets
- sample code fragments
- assignment description
- grading rules
- FAQ



The screenshot shows a dark blue navigation bar at the top with links for Main, News, Research, Publications, Courses, Students, Software, and Funding. Below the navigation bar, the page is divided into two columns. The left column contains a profile for prof. dr. Alexandru C. Telea and a list of navigation links: Home, Lectures, Class overview, Slides, Readings, Grading and Rules, Assignment, Overview, Code, and Others. The right column features the title 'Data Visualization' and 'Course content'. The 'Course content' section includes a paragraph about the course's focus on relational and high-dimensional data, followed by two sub-sections: 'Relational data' and 'High-dimensional data', each with a list of topics to be covered.

prof. dr. Alexandru C. Telea

**Data Visualization**

[Home](#)

**Lectures**

[Class overview](#)

[Slides](#)

[Readings](#)

[Grading and Rules](#)

**Assignment**

[Overview](#)

[Code](#)

**Others**

[Additional material?](#)

[FAQ](#)

**Data Visualization**

**Course content**

This course teaches Data Visualization methods with a focus on relational and high-dimensional data. Such data appear in many of real-life applications from science, social phenomena, and engineering. The course is divided into two parts: parts - relational and high-dimensional data - as follows.

**Relational data** We present a model used for representation and storing relational data, called graph or network, and list applications of network visualization. We define so-called quality metrics to construct and assess network visualizations. We study several network visualization methods, including:

- Tree visualization;
- Visualizing general graphs;
- Visualizing hierarchical graphs;
- Visualization of multilevel networks for modeling highly complex applications;
- Graph bundling.

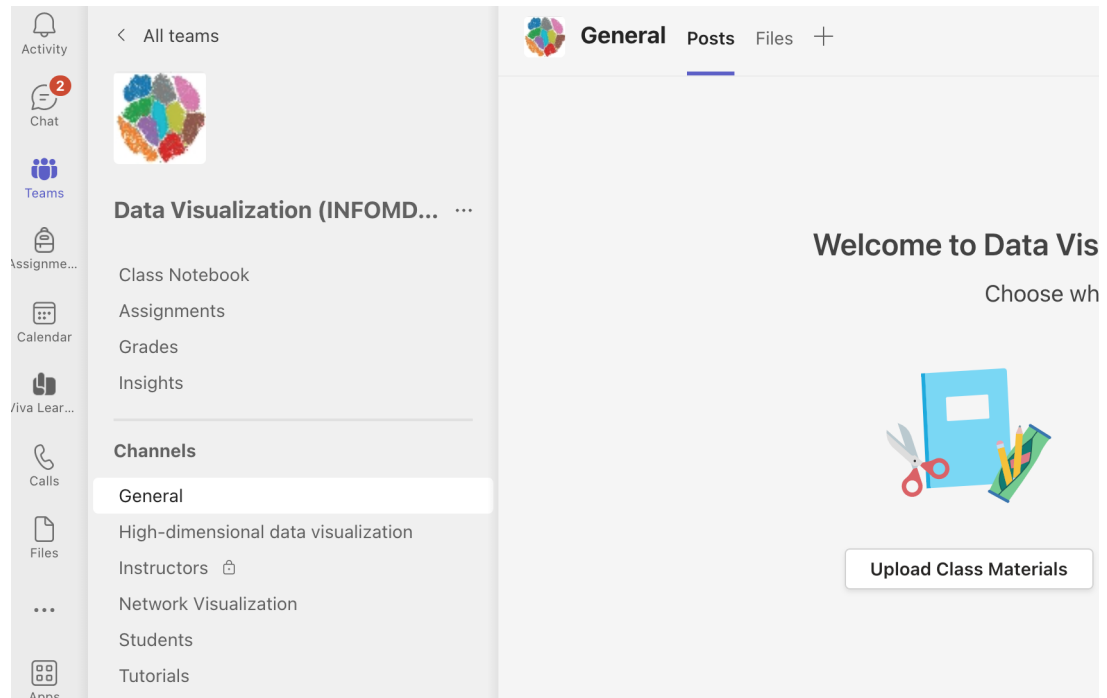
**High-dimensional data** We present methods that teach the visualization of large data tables (thousands of rows, hundreds of columns) from data science, artificial intelligence, and related fields. We discuss these methods as well as quality metrics to assess them including:

- Parallel coordinate plots
- Scatterplot matrices
- Table lenses
- Dimensionality reduction

# Communication Channels

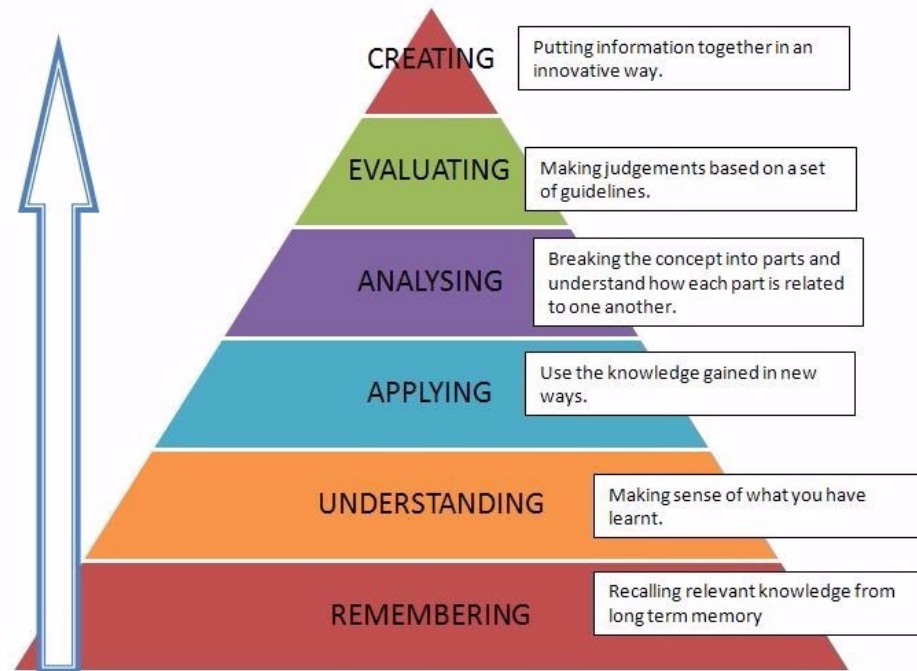
## MSTeams

- link in blackboard – please request access
- General : announcements and questions about organizational matters
- High-dim data vis : questions about high-dim vis topics
- Net Vis : questions about network visualization topics
- Students : communication among the students, can help you to find project partner
- Tutorials : questions about assignments



# Learning Goals

- Have an overview of the state-of-the art visualization methods for relational and high-dimensional data (Remember)
- Be able to explain the functionality of these methods (Understand)
- Implement the methods in your programming language of choice (Apply)
- Parametrise the methods for a range of data sets (Apply)
- Assess the results of the constructed visualizations (Analyse, Evaluate)
- Present and motivate all taken choices (Analyse, Create)



Bloom's Taxonomy for Learning Outcomes

# Additional Remarks

## Competences assumed you have (and develop further)

### Coding

- you need to be able to **program** (well)!!!
- lecturer is **not assumed** to debug your source code

### Communication

- describe a problem/question **correctly**, **compactly**, and **technically** (“it does not work, what to do?” is not a good example 😊)

### Reporting

- your final report should be at the level of an **industry white paper / tech report / scientific publication**
- well structured, complete, coherent, with clear math notations, references
- **illustrate** all you do by graphs, charts, snapshots

# Additional Remarks

## Be present!

- lectures give **detailed explanation** of the methods you need to implement
- tutorial sessions give opportunity to **ask questions** and **anticipate challenges**

## Be proactive!

- ask **questions** during lectures!
- provide assignment progress **reports**
- do read all information on course **website**

## Watch the time!

- make sure you progress **continuously** with the assignment
- report blockers ASAP!

## Last but not least: Have fun!