# Data communication and society with tool for social research and data analysis

## Prof. Matteo Tarantino

***COURSE AIMS AND INTENDED LEARNING OUTCOMES***

The course will provide students with a solid understanding of the role of data in contemporary societies. This understanding will be twofold: on the one hand, the course will provide students with knowledge of the fundamental concepts and frameworks concerning the way contemporary societies are being transformed by data. In particular, the course will zoom on how data and their communication impact sustainability-related processes. On the other hand, the course empowers studies with practical data-related and software-related skills focused on the Python programming language, providing students with hands-on explorations of the topics covered in the theoretical part. No prior knowledge of coding is required to access the course. The course is fully delivered in English.

 EXPECTED LEARNING OUTCOMES.

Knowledge and understanding

At the end of the course, students will be able to:

– Understand the socio-technical roots of the current abundance of data in society.

– Understand all the core concepts and techniques

– Understand challenges and opportunities provided by data and their communication towards a more sustainable society.

– Understand the nature and implications of recommender systems underpinning systems such as Amazon, Spotify and Netflix.

– Understand the real implications of datafication for privacy and security, as well as the countermeasures.

– Understand the relationships between data and space, and the challenges of “smartifying” cities.

 Ability to apply knowledge and understanding

The course does not aim to form fully-fledged coders; rather, in its applied part, it aims to provide the core skills allowing a communication professional must possess to deal competently with IT.

At the end of the teaching, students will be able to:

– Develop opportune data communication strategies

– Identify key stakeholders and their relationship in data-related processes.

– Assess the typology of data more adequate for specific needs.

– Assess the optimal method of data acquisition for each specific need.

– Assess the data quality and clean data when necessary

– Read software code and understand the core logic of code.

– Understand how to choose and use databases.

– Apply the core techniques of data visualization.

***COURSE CONTENT***

Each week features two classes: a Concepts class (C) and an Applied class (A). The former explores frameworks, theories and implications of data-related phenomena.

The latter provides a hands-on approach to the same topics, offering students the chance to understand and prototype tools using the Python language, proceeding from Python basics to data management and data visualization.

The applied class will combine pre-recorded content training on the basics, and live coding to foster their skills.

* T: Data & Society: Conceptual foundations. Datafication. Data/Information/Knowledge/Wisdom. Typologies of data. Algorithms. IoT, Blockchain.
* A: Understanding Code: Foundations of Software Logic. Building blocks of software.
* T: How to estimate the value of data? Data collection, data management.
* A: Ingesting & manipulating datasets with Python & Pandas.
* T: The challenges of data collection and protection: Access, Databases & APIs.
* A: Access to data.
* T: The challenges of Data Quality and Interoperability -1.
* A: Fundamentals of data tidiness: Data cleaning 101.
* T: The challenges of Data Quality and Interoperability -2
* A: Introduction to automated data collection from social media.
* T: Case study #1: Environmental data in China.
* A: Basics of data security: Hashing, Encryption, Redundancy
* T: How Algorithms Shape Consumption: Understanding recommender systems.
* A: Data Visualization 1: Fundamental and Core Techniques
* T: The Future of Data for Sustainability (blockchain, edge computing, satellite imagery…).
* A. Data visualization 1: Fundamentals of Pyplot, Core techniques.
* T: Data & Space: What lies beyond “Smart” Cities?
* A: Data visualization 2: Advanced Techniques
* T: Conclusions: Bringing the Social back into data: The notion of Data Cultures.

***READING LIST***

*Mandatory Readings*

R. Kitchin. The data revolution: Big data, open data, data infrastructures and their consequences. Sage, Thousand Oaks (2014).

R. Kitchin-M. Dodge. Code/space: Software and everyday life. MIT Press, Cambridge (2014).

G. Adomavicius-A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge & Data Engineering 6 (2005): 734-749.

H.Wickham, Tidy Data. Journal of Statistical Software, 59(10), 1 - 23 (2015).

N. Marres-E. Weltevrede, Scraping the Social? Issues in live social research. Journal of Cultural Economy, 6(3), pp. 313-335. Goldsmiths Research Online. ISSN 1753-0350 (2016).

M. Tarantino, Uncertainty in the Air: Communicating Urban Air Pollution, in Z. Krajina & Stevenson (eds.) The Routledge Companion to Urban Media and Communication. Routledge, New York (2019).

M. Tarantino, Navigating a datascape: challenges in automating environmental data disclosure in China. Journal of Environmental Planning and Management, 2020, 63.1: 67-86 (2020).

*Suggested Readings*

Schedl et al. Current challenges and visions in music recommender systems research. International Journal of Multimedia Information Retrieval (2018) 7:95–116 <https://doi.org/10.1007/s13735-018-0154-2>

UN Economic Commission for Europe, Guidelines on evidence-based policies and decision-making for sustainable housing and urban development. Online all’indirizzo: <https://unece.org/DAM/hlm/documents/Publications/2020_Guidelines_on_evidence-based_policies.pdf>

W. McKinney. 2017. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd. ed.). O'Reilly Media, Inc. (chapters 1, 2, 5-9)

Al Sweigart. 2015. Automate the Boring Stuff with Python: Practical Programming for Total Beginners (1st. ed.). No Starch Press, USA. (chapters 1-13, 16, 17, A1, A2)

R.C. Martin. 2008. *Clean Code: A Handbook of Agile Software Craftsmanship (1st. ed.)*. Prentice Hall PTR, USA. (chapters 1 - 4)

***TEACHING METHOD***

The theoretical part of the course will be taught through frontal lectures, involving also guest lecturers. The applied component will rely on a learning-by-doing approach, requiring students to bring along their own computer to fully experience the applied part. Recordings of both theoretical and applied classes will always be provided on the Blackboard platform. Students unable to join group project works will be required to agree on individual assignments with the instructors.

***ASSESSMENT METHOD AND CRITERIA***

The final grade will be determined by the average of three written assessments (each evaluated 0-30): theoretical proficiency (50%), practical proficiency (30%), and group project work (20%). All assignments must be submitted and discussed in English.

The theoretical proficiency assessment evaluates the student’s proficiency in the theoretical notions explained in the theoretical lessons. The practical proficiency assessment evaluates the student’s proficiency in the practical skills necessary to apply the notions explained in the applied lessons.

The project work will be used to evaluate the student’s capability of designing and realising a project related to a set of notions from T and A lessons (above). Instructions about the project work will be provided at the start of class.

***NOTES AND PREREQUISITES***

The course requires a B2 level (upper-intermediate) competency of the English language. Students must have a basic understanding of foundational theories of communication systems. No prior knowledge of coding is required, but students with experience in this sense will benefit in the initial phase of the applied class. For the applied part, it is recommended students use on either a Windows, MacOS or Linux personal computer, and not other means such as mobile phones or tablets. *A foundational tutorial on the Python language will be provided before the course via pre-recorded video lectures,* along with tutorials about setting up the necessary software.