

# Introduction to Digital Society (Methods 1) (6 ECTS)

## Course File

dr. Joanna Strycharz  
j.strcharz@uva.nl

Ognjan Denkovski  
o.denkovski@uva.nl

College of Communication  
University of Amsterdam

Academic Year 2021/22,  
Semester 2, block 1

# Contents

1	Short description of the course (as in the course catalogue 2021-22)	2
2	Exit qualifications	4
3	Testable objectives	5
4	Planning of testing and teaching	7
5	Literature	8
6	Specific course timetable	9
7	Testing	13
8	Lecturers' team, including division of responsibilities	15
9	Calculation of students' study load (in hours)	16
10	Calculation of lecturers' teaching load (in hours)	17

# Chapter 1

## Short description of the course (as in the course catalogue 2021-22)

### 1.1 Objectives

Upon completion of the course, students should:

1. Have a basic understanding of the use of programming languages for computational communication science and the practices behind open science.
2. Have a basic understanding of the most common data types in programming.
3. Have a basic understanding of simple control structures (loops, conditions and functions).
4. Have basic experience with exploring and visualizing different data types and executing basic statistical analyses using programming languages.
5. Have basic experience with handling errors in programming languages and debugging with the help of online resources.
6. Have basic experience with retrieving data using APIs and other sources of data.
7. Be able to independently seek, find and apply solutions for the desired operation in programming languages.

### 1.2 Contents

Recent developments in digital technologies have fundamentally changed our society and the ways in which we communicate, creating significant need for basic programming capacities and familiarity with the methods which allow for the analysis of these trends. This course provides a comprehensive introduction to the computational methods and practices that allow researchers and practitioners to study different phenomena in the digital society, while making use of the digital data generated by individuals through different daily activities.

The course starts with the assumption that students have no previous knowledge of computational methods. Students receive a comprehensive introduction to Python for data science, including best practices for data wrangling, generating basic descriptive and

inferential statistics, as well as data visualization in Python. Students are also introduced to the basic principles of APIs and version control with Git. After taking this course, students will have gained an understanding of various key concepts and techniques for data science with common programming languages for data analysis and will have hands-on experience with applying these techniques to a wide range of tasks.

### 1.3 Recommended prior knowledge

The course starts with the assumption that students have no previous knowledge of computational methods. However, it is important to note that the course discusses basic statistical methods relevant for digital data. The following book offers a comprehensive overview of and introduction to statistical methods used in communication science:

Hayes, A. F. (2020). *Statistical methods for communication science*. Routledge.

### 1.4 Teaching method and contact hours

The course is taught as one weekly lecture focused on explaining the key concepts in computational methods and one weekly tutorial session with hands-on exercises in which students directly apply the acquired skills.

### 1.5 Study materials

Students in the course follow tutorials (online and in class) to learn how to perform analyses of digital data using Python. The following book will be used:

Van Atteveltdt, W., Trilling, D., Arcilla, C. (in press). *Computational Analysis of Communication*. Hoboken, NJ: Wiley.

Additional literature and hands-on tutorials (programming scripts) will be provided. In some weeks, students will be provided with microlectures which will explore certain aspects of the week's material in greater depth.

### 1.6 Assessment

The course will use the following assessment methods:

1. Weekly knowledge questions covering the content for the week (20% of final grade)
2. Weekly assignments completed before the tutorials in which students practice the material of the week (30% of final grade)
3. Take-home exam in which students complete a data analysis challenge applying knowledge from the course (50% of final grade)

# Chapter 2

## Exit qualifications

The course contributes to the following exit qualifications of the Minor Communication in the Digital Society:

- Bekwaamheden en vermogens
- Probleemoplossend vermogen
- Academische houding

The exit qualifications are elaborated in the following four specifications:

- 1.b.3 Regarding specific topics, the graduate is able to understand, position and critically discuss not too complex scientific publications within the field. Is able to recognise and acknowledge the importance of contribution of other fields of study. Is able to adopt a position regarding scientific debate within the field.
- 4.b.2 Is able to assess research within the discipline on its practical usability.
- 4.b.3 Is able to recognise the relevance of other disciplines (interdisciplinarity).
- 4.b.9 Is able to deal with the field of study in a creative manner.
- 5.b.5 Is able to relative independently revise and expend knowledge through study.

Dutch or English?, only the one's for the elective or in general for Bachelor

# Chapter 3

## Testable objectives

The table of the following page provides an overview of the exit qualifications and related testable objectives of the course.

	Exit qualifications & Testable objectives	1.b.3	4.b.2	4.b.3	4.b.9	5.b.5
		Regarding specific topics, the graduate is able to understand, position and critically discuss not too complex scientific publications within the field. Is able to recognise and acknowledge the importance of contribution of other fields of study. Is able to adopt a position regarding scientific debate within the field.	Is able to assess research within the discipline on its practical usability.	Is able to recognise the relevance of other disciplines (interdisciplinarity).	Is able to deal with the field of study in a creative manner.	Is able to relative independently revise and expand knowledge through study.
A	Have a basic understanding of the use of programming languages for computational communication science and the practices of open science	X	X	X		
B	Have a basic understanding of the most common objects and data types in programming				X	
C	Have a basic understanding of simple control structures (loops, conditions and functions)				X	
D	Have basic experience with exploring and visualizing different data types and executing basic statistical analyses using programming languages				X	
E	Have basic experience with handling errors in programming languages and debugging with the help of online resources					X
F	Have basic experience with retrieving data using API's and other sources of data				X	
G	Are able to independently seek, find and apply solutions for the desired operation in programming languages				X	X

# Chapter 4

## Planning of testing and teaching

The course consists of fourteen meetings, two per week. Each week, in the first meeting, the instructor will give short lectures on the key aspects of the week, will demonstrate them through live-coding and host QA sessions with the students. In preparation for these meetings, students will read assigned literature. Their understanding of the key aspects of the weekly content will be tested through multiple choice questions.

The second meetings each week are practicum-meetings. Before each practicum-meeting the students follow tutorials composed of videos and analysis/coding challenges (on GitHub) created specifically for this course. In the challenges, the students will apply the techniques central to the week of the course using datasets provided in the course. The students upload the results of the tutorials to GitHub, allowing the lecturer to assess the progress and the main questions/issues that students experienced. In the practicum-meeting, both the lecturer and the students present the results in class, discuss and solve the main issues/questions.

To complete the course, next to active participation, the students have to successfully complete three summative graded assignments: multiple-choice questions regarding the content of the week, weekly challenges completed before the practicum-meetings and a final take-home exam (see Chapter 7 for details).



# Chapter 5

## Literature

The schedule in chapter 6 gives an overview of the topics covered each week, the obligatory literature that has to be studied each week, and other tasks the students have to complete in preparation of the class.

In the course students read several chapters from the book *Computational Analysis of Communication* (van Atteveldt, Trilling, & Carlos, in press). The specific chapters are mentioned in the schedule. Some basic chapters that explain how to install the software we are going to use have to be read before the course starts.

Next to the book chapters, specific literature and materials on the topic of each week will be studied. Additionally, some weeks include optional readings that are not obligatory material, but are included as reference to provide the interested students with deeper knowledge. The overview of all required and additional literature is presented below as well as in the schedule.

- Lazer et al. (2020)
- van Atteveldt and Peng (2018)
- Hilbert et al. (2019)
- Van Atteveldt, Strycharz, Trilling, and Welbers (2019)
- Freelon (2018)

maybe  
we  
should  
add  
here  
all  
the  
links  
to  
other  
ma-  
teri-  
als?

# Chapter 6

## Specific course timetable

The chapters mentioned in the schedule relate to van Atteveldt et al. (in press).

### **Before the course starts: Prepare your computer.**

#### **✓ INSTALL PYTHON AND OTHER NECESSARY TOOLS**

Make sure that you have a working Python environment installed on your computer. You cannot start the course if you have not done so. Installation instructions will be provided on Canvas.

### **Week 1: What is Computational Communication Science, and why Python?**

#### **Lecture**

We discuss what computational communication science is. We talk about challenges and opportunities as well as the implications of such developments for communication science in particular. We also pay attention to the tools used in CCS, in particular to the use of Python. Finally, we get an introduction to the tools used in class.

Mandatory readings (in advance):

- Computational Analysis of Communication, Chapter 1.2
- Computational Analysis of Communication, Chapters 3.2-3.2.4
- Lazer et al. (2020)

## Tutorial

### Week 2: Control flow tools, version control and open science

#### Lecture

We discuss the notion of open science and its relevance to computational communication science. We also focus on key aspects of reproducible computational research workflows and get introduced to the relevant tools.

In the second part of the lecture, we focus on programming simple control structures such as loops and conditions.

Mandatory readings (in advance):

- Computational Analysis of Communication, Chapter 3-3.3
- Van Atteveldt et al. (2019)
- Github workshop <https://github.com/vanatteveldt/github-workshop>

## Tutorial

### Week 3: Data types, basic pre-processing and handling errors

#### Lecture

We discuss the most common data types used in computational communication science. We focus on advantages and disadvantages of different data structures and discuss how to choose the most appropriate structure for different types of data. Finally, we get introduced to writing your own functions in Python.

Mandatory readings (in advance):

- Computational Analysis of Communication, Chapter 3.4
- Computational Analysis of Communication, Chapter 4-4.3
- Python error and exceptions run through: <https://docs.python.org/3.3/tutorial/errors.html>

## Tutorial

### Week 4: Data exploration and wrangling

We get an introduction to data exploration and wrangling steps. We will learn how to do data wrangling with Pandas: subsetting datasets, converting between wide and long

formats (melting and pivoting), aggregating data, joining datasets, and so on.

Mandatory readings (in advance):

- Computational Analysis of Communication, Chapter 7

## **Tutorial**

### **Week 5: Visualising and analysing data**

We discuss best practices for visualising different types of data and get introduced to some of the most common packages used for data visualisation in Python. We also get an introduction to conducting basic statistical analyses in Python (note that students are expected to have understanding of most common descriptive and inferential statistical methods). Finally, the lecture includes a QA and a catch-up on content from the first four weeks of the course.

Mandatory readings (in advance):

- Computational Analysis of Communication, Chapter 8.1–8.2

## **Tutorial**

### **Week 6: APIs and working with your own data**

We discuss the conceptual principles of Application Programming Interfaces (APIs), their relevance as data access services and their continued relevance as frameworks for accessing complex, state-of-the-art (machine learning) models.

In the second part of the class, we provide a practical introduction to the Twitter API and discuss research ideas for Twitter/other API data based on the weekly literature. Students develop a simple research idea for their own custom Twitter dataset.

Mandatory readings (in advance):

- Computational Analysis of Communication, Chapter 13.1
- Computational Analysis of Communication, Chapter 13.4
- Computational Analysis of Communication, Chapter 15.3

## **Tutorial**

### **Week 7: CCS — looking ahead**

In this final week, we discuss emerging trends that have come to dominate working with computational approaches. We discuss the implications of the move towards a post-API age for researchers based in computational approaches. Here, we provide a brief demonstration of web scrapping as an ad-hoc solution to the decreasing availability of

'free' data and discuss the relevance of ethical considerations while collecting custom data online.

In the second part of the class, we discuss the increasing relevance of cloud infrastructure approaches and database management as basic tools for robust business and academic approaches for managing large data sets and optimizing insights.

Mandatory readings (in advance):

- Computational Analysis of Communication, Chapter 13.2-13.3
- Computational Analysis of Communication, Chapter 17
- Freelon (2018) \_\_\_\_\_

## Tutorial

here  
or  
week  
ear-  
lier?

# Chapter 7

## Testing

An overview of the testing is given in Table ??.

### Grading

The final grade of this course will be composed of the grade of grades for weekly knowledge questions (20%), weekly assignments (30%) and a final take-home exam (50%).

#### Weekly knowledge questions (20%)

The weekly knowledge questions will be included at the beginning of tutorials and will test students knowledge of the constructs introduced in the literature and the lecture.

#### Weekly assignments (30%)

two options, see word doc

#### Take-home exam (50%)

Grading criteria are communicated to the students together with the assignment, but in general are: \_\_\_\_\_

For programming-related tasks in the exam:

- correctness, efficiency, and style of the code
- correctness, completeness, and usefulness of analyses applied.

For conceptual and planning-related tasks:

- feasibility
- level of specificity
- explanation and argumentation why a specific approach is chosen
- creativity.

check  
with  
learn-  
ing  
goals

## **Grading and 2<sup>nd</sup> try**

Students have to get a pass (5.5 or higher) for.... If the grade of one of these is lower, an improved version can be handed in within one week after the grade is communicated to the student. If the improved version still is graded lower than 5.5, the course cannot be completed. Improved versions of the final individual project cannot be graded higher than 6.0.

## Chapter 8

**Lecturers' team, including division  
of responsibilities**



# Chapter 9

## Calculation of students' study load (in hours)

- Elective total: 6 ECTS =168 hours
- Reading:
  - 13 articles, average 20 pages: 260 pages. 6 pages per hour, thus 43 hours for the literature
  - Reading and doing tutorials: 14 hours for reading tutorials to acquire skills.
  - Reading total: 57 hours.
- Presence:  
16\*1,75 hours: 28 hours.
- Mid-term take-home exam, including preparation: 14 hours
- Final individual project, including data collection, analysis, write up: 70 hours

Total: 169 hours

# Chapter 10

## Calculation of lecturers' teaching load (in hours)

### 10.1 Lecturers

- Presence: 14 hours ( $= 7 * 2$  hours)
- Preparation of course (including necessary notebooks): 26 hours
- Preparation of weekly lectures and knowledge questions,  $7 * 3$  hours: 21 hours
- Preparation of weekly challenges,  $7 * 2$  hours: 14 hours
- Preparation of final exam (incl. necessary datasets): 10 hours
- Feedback and grading final exam:  $50 * 25$  minutes: 23 hours
- Administration, e-mails, individual appointments: 7 hours

Total: 115 hours

### 10.2 Tutorial lecturer

- Presence: 14 hours ( $= 7 * 2$  hours)
- Preparation of weekly tutorial meetings,  $7 * 2$  hours: 14 hours
- Feedback and grading weekly challenges,  $7 * 4$  hours (10 minutes per student): 42 hours

Total: 56 hours

# Literature

- Freelon, D. (2018). Computational research in the post-api age. *Political Communication*, 35(4), 665–668.
- Hilbert, M., Barnett, G., Blumenstock, J., Contractor, N., Diesner, J., Frey, S., . . . others (2019). Computational communication science: A methodological catalyzer for a maturing discipline.
- Lazer, D. M., Pentland, A., Watts, D. J., Aral, S., Athey, S., Contractor, N., . . . others (2020). Computational social science: Obstacles and opportunities. *Science*, 369(6507), 1060–1062.
- van Atteveldt, W., & Peng, T.-Q. (2018). When communication meets computation: Opportunities, challenges, and pitfalls in computational communication science. *Communication Methods and Measures*, 12(2-3), 81–92.
- Van Atteveldt, W., Strycharz, J., Trilling, D., & Welbers, K. (2019). Computational communication science— toward open computational communication science: A practical road map for reusable data and code. *International Journal of Communication*, 13, 20.
- van Atteveldt, W., Trilling, D., & Carlos, A. (in press). *Computational analysis of communication: a practical introduction to the analysis of texts, networks, and images with code examples in python and r*. John Wiley Sons.