Data Manipulation Essentials

2026/2027, Autumn Trimester (T1)

Course code	STAT163
Course faculty	Oleh Omelchenko — o_omelchenko@kse.org.ua
Course assistants	Ivan Lukianets · Artem Avramenko
Department	Computer Science Department
Study programmes	Economics and Big Data (mandatory) · Software Engineering (in-plan elective)
ECTS credits	4 (120 academic hours total)
Class hours	48 academic hours (16 lecture + 32 practice)
Course dates	2026-09-01 → 2026-11-01
Course language	English
Course format	Offline, with materials to work through before each lecture
Form of study	Full-time

> **Status.** This syllabus is settled in structure. Some details are still being confirmed — they are marked **TBC** below. Anything that affects your grade will be fixed in writing before the work it applies to begins.

Overview

Prerequisites

This course assumes participants have completed CS201 Programming Basics or have equivalent programming experience. You should be comfortable with variables, functions, loops and conditionals in at least one programming language.

Fundamental skills:

- Basic programming logic (variables, functions, loops, conditionals)
- Ability to work with files and folders on a computer
- Basic understanding of tabular data (rows and columns)

Nice to have:

- Some Python experience
- Familiarity with Jupyter notebooks or similar interactive environments
- Basic statistical concepts (mean, median, distribution)

Background and course rationale

Every later data course assumes that you can already work with tables: read what a row and a column mean, combine two tables, and summarise data without losing track of what the summary leaves out. Almost no course teaches this directly.

A wrong number rarely comes from wrong arithmetic. It comes from averaging at the wrong level of detail, joining on a key that is not unique, or trusting a column that mixes text with numbers.

This course teaches these skills with Python and pandas. The same operations exist, under the same names, in spreadsheets, in SQL and in R, so what you learn here transfers.

Course aims

- Build mental models of data manipulation that survive a change of tool
- Establish systematic habits for profiling and quality assessment
- Give hands-on fluency with Python and pandas
- Instil discipline around missing values, join validation and aggregation correctness
- Produce work that another person can re-run and get the same answer from

Learning outcomes

By the end of the course you will be able to:

1. **Profile an unfamiliar dataset** — apply a systematic method (shape, grain, types, missingness, distributions, anomalies) and report what the data can and cannot answer
2. **Manipulate data programmatically** — filter, sort, derive, reshape, aggregate and join with pandas, in code that reads clearly
3. **Handle missing data on purpose** — recognise how missing values propagate, choose a strategy, and state what the choice cost
4. **Validate data operations** — check joins for fan-out, catch average-of-averages, confirm an operation preserved the grain you intended
5. **Work with multiple formats** — read, write and convert CSV, JSON, Excel and Parquet, including nested JSON
6. **Process text** — clean and extract with string methods and regular expressions
7. **Build reproducible pipelines** — chain transformations into documented work that another person can re-run
8. **Name operations tool-agnostically** — recognise the same pattern across pandas, SQL, spreadsheets and R

What this course feeds

The operations taught here recur in every later course that touches data — visualisation, business intelligence, econometrics, machine learning and database work all assume them.

Methods of teaching and assessment

- Live lectures, used for discussion and depth rather than first delivery
- Materials to work through before each lecture: a pre-lecture notebook, reading, video
- Two practice sessions per week, hands-on
- Pre-lecture quizzes
- Three individual assignments
- Mid-term and final exams
- Sampled oral discussion of submitted assignment work

Course faculty

Oleh Omelchenko, M.Sc. in Telecommunications

Office hours: TBC

Professional experience. Eight years in data analytics and analytics engineering, mostly on web and mobile products. Previously e-commerce and startups. Since 2020, Senior Product Analyst at MacPaw, a Ukrainian software company building applications for macOS and iOS.

Research interests. Visualisation literacy, and differences in how people comprehend data graphics.

Teaching approach. I go beyond a fixed topic list where it is useful, and the choice of extra topics is driven by what you are likely to meet in a working environment rather than by what is conventional in a course of this

name. There is a set of technical problems you are expected to learn to solve, but I would rather you left with a coherent picture of the area. That needs traffic in both directions, so ask questions during lectures.

Course assistants. Ivan Lukianets and Artem Avramenko. Practice sessions are led by the course faculty and the assistants.

Course structure

The course runs for eight weeks. Each week has one live lecture and two practice sessions.

Before the lecture. Each week comes with materials to work through before the lecture: a pre-lecture notebook, reading and video. Work through them first. The lecture assumes you have met their ideas. From Week 2 a short quiz at the start of the lecture checks that you have.

The lecture. Discussion, worked examples and the reasoning behind the operations. This is where “why” is answered.

The practice sessions. Two consecutive slots, hands-on, one notebook across the pair. Handed out through GitHub; submitted by pasting your repository URL on Moodle. This is where “how” is answered.

Assignments. Three mini-projects, each combining several weeks of material into a realistic multi-step problem. Marked on technical implementation (70%) and on documentation and explanation (30%). Communicating how you prepared data is part of the work, not an extra.

Exams. A mid-term in Week 5 and a final in Week 8. Both are written in class, on paper, closed book. Neither asks you to write code.

Hours

Four ECTS credits means about 120 hours of your time. Class time is 48 academic hours. The rest is reading, watching, assignments and exam preparation.

Make-up policy

Quizzes. Taken at the start of the lecture, in person. No make-ups.

Practices. Due by the end of the day after your practice session. Deadlines run per practice group. A practice whose work did not reach us is not accepted: a repository that holds the notebook as you received it, or work on GitHub with no URL submitted on Moodle.

Assignments. Two deadlines apply:

When you submit	What you get
On time, or within 3 hours	100% of what you earned
3 hours to 7 days late	50% of what you earned
More than 7 days late	0 points, not accepted

Extensions. Ask at least two days before the deadline. Requests made later will not be considered.

Grading policy

General grading scheme

Component	Detail	Points
Practices	7 weeks × 3 pts	21
Pre-lecture quizzes	5 weeks × 2 pts	10
Assignments	3 assignments (8 + 8 + 8)	24
Mid-term exam	Week 5	20
Final exam	Week 8	35
Total		**110**

If your total exceeds 100, your grade is 100.

65 of the 110 points are closed book. That is deliberate, and it is what makes the open policy on take-home work possible. See *AI tools usage policy* below.

Practices — 21 points

Hands-on notebooks, one per week for Weeks 1 to 7. You work in your own private copy of the week's starter repository on GitHub (instructions on Moodle) and submit by the end of the day after your practice session.

Graded 0 to 3 on how much of the notebook you completed with working answers.

Pre-lecture quizzes — 10 points

Short quizzes at the start of the lecture, through Moodle, in Weeks 2, 3, 4, 6 and 7. Five to seven questions, ten minutes. They check that you understood the ideas in that week's pre-lecture materials. You take them in the lecture, in person. No make-ups.

Graded 0 to 2 on the share of correct answers. The rule is no AI.

Treat the ten quiz points as a bonus. You can pass the course without them. They reward coming to the lectures and working through the materials.

Assignments — 24 points

Three mini-projects. Released and due as shown in the schedule.

Marked on implementation (70%) and on documentation and explanation (30%). The disclosure statement and the reflection note described below are part of the documentation mark.

Mid-term exam — 20 points

In class, on paper, in the Week 5 lecture slot. Covers Weeks 1–3. Closed book, no AI.

You read code and explain it: what a line does, what it will produce, what is wrong with an approach. The code comes from the pre-lecture notebooks, so working through them is exam preparation.

The mid-term is a checkpoint. It tells you where you actually stand while there is still time to act on it, and it shows you the final's format.

Final exam — 35 points

In Week 8, in class, on paper, 80 minutes. Covers Weeks 1–7. Closed book, no AI. You may bring one A4 sheet of your own notes. Same format as the mid-term.

Every question is on something you did in the practices and the assignments, or saw in the lectures. `resample`, `cumsum`, `shift`, `rank` and method chaining are not on the final.

A short talk about your assignment

After an assignment, I may invite about one student in ten to talk through their own submission. Five to ten minutes: why this join, why this grain, why this number.

- Whether a round of talks happens after a given assignment is my decision.

- Who is invited is not: the selection is random. Being invited is not an accusation.
- You get short notice.
- Talks are graded against a rubric, published before the first round.
- The result confirms or adjusts your mark on that assignment.
- Speak Ukrainian or English. I grade your reasoning, not your fluency.

Academic integrity warning

Academic integrity is highly valued by the KSE community and its vast majority of students. We have a zero-tolerance policy towards academic plagiarism, self-plagiarism, fabrication, falsification, cheating, deception, bribery and other types of violations. Due to the highly competitive nature of the programmes, all students must be treated equally. Even a single case of violation of academic integrity is a serious misdemeanour that may lead to unjust redistribution of grades and, consequently, overall rankings, possible grants, fee reductions, and other merit-based awards. Therefore, penalties may vary from receiving zero points or a negative grade for an assignment to expulsion of the student from KSE, depending on the severity of the case and might include additional consequences such as deprivation of scholarships, financial assistance etc. All the rules and procedures regarding academic integrity are stated in the KSE Code of Academic Integrity. Before enrolling in this course, students must be aware of and abide by the KSE Code of Academic Integrity.

AI tools usage policy

Use AI to learn. Prove you learned without it.

The rule

Where	Rule
Practices and assignments	AI is allowed. Use any tool, as much as you like
Quizzes, mid-term, final	No AI. Closed book

AI assists; you author. A submission authored end to end by AI is not your work and is not accepted. Every practice and assignment carries a block at the top stating the recommended AI mode for that task. On practices it is: try each task yourself first; in the classroom, ask the teacher or TA when stuck; remotely, turn to AI.

Three conditions on take-home work

1. Say what you used. Write one line in your notebook. Name the tool and the step.

> Used Claude to debug the groupby in Task 3.2.
>
> Used ChatGPT to explain why the merge produced more rows than I started with.

That is a disclosure. This is not:

> Used AI.

2. You own what you submit. An AI error is your error. “The model wrote it” is not a defence, and it does not reduce a penalty.

3. Expect to explain your work aloud. If you cannot say why your code does what it does, the exams will show it.

Why the rule is shaped this way

You will use AI in your studies and in your work. A course that bans it from homework bans it from where most learning happens.

At the same time a grade has to mean something. So the verification moved to where it can be controlled: 65 of the 110 points are closed book. Because those points exist, the other 45 do not need policing.

Using AI as a shortcut in take-home work predictably fails in the exams. That is the intended structure. It replaces surveillance with alignment.

No AI detector is used in this course, at any point, for any purpose. Detectors are unreliable, and no decision here rests on one.

This differs from your previous courses

Course	Its rule
CS201 Programming Basics	AI prohibited for graded work
CS301 Introduction to Databases	AI for search only, prompts attached
STAT163	**Open on take-home work, closed on exams**

The difference is deliberate, not an oversight. Different courses assess different things. Follow the rule of the course you are in.

How to use AI well

- Use it to **understand**, not to bypass. Ask it to explain an error, critique your approach, or generate practice cases. Do not ask it to produce a deliverable you then submit unread
- Verify everything. Code that runs is not code that is correct. Analysis that sounds plausible is not analysis that is right
- Ask it *why*. “Why does this NaN propagate here?” teaches you more than “fix this”
- Notice when not to use it. Judging when a tool is the wrong tool is part of the skill

---

How I use AI

The same disclosure applies in the other direction.

Where	What I do
Slides and quiz questions	Drafted with Claude. I review, edit and iterate on every one
Practices and assignments	Checked semi-automatically against an answer key. I review the output
Exams	Read and graded by me, start to finish. No automation

Every grade is my own judgement, and I am accountable for it. Your submissions are not fed into consumer AI tools.

Course plan

Learning materials

Primary text

- *Python for Data Analysis*, 3rd edition — Wes McKinney (O’Reilly)

Supplementary

- *Practical Python Data Wrangling and Data Quality* — Susan E. McGregor (O'Reilly) — profiling and data quality
- *Pandas for Everyone*, 2nd edition — Daniel Y. Chen (O'Reilly) — a gentler alternative to McKinney

Reference

- *Pandas Cookbook*, 3rd edition (O'Reilly)
- pandas documentation — pandas.pydata.org

Course materials

- Weekly pre-lecture notebooks, reading lists, videos and slides on Moodle
- Practice notebooks and datasets through GitHub

The installation chapters in both books assume a Miniconda toolchain. This course uses uv. Skip those chapters and follow the course setup instructions instead.

Tools and software

- Python 3.10 or later
- uv for environments and packages
- Positron or the Jupyter web interface for notebooks — the two environments we can help you debug in. Anything else is your choice
- pandas, NumPy
- Git and GitHub

Schedule

Times and rooms are on the KSE timetable and can change during the trimester. This table gives the week structure, which does not.

Week	Topic	Reading	Graded
1	**The foundation under data analysis** — what one row means, what one column holds, N/O/Q/T, grain	PDA Ch. 1, 5.1–5.3 · PPDW Ch. 1, 3	Practice 3
2	**DataFrames and profiling** — indexing, filtering, sorting, derived columns, type conversion, duplicates, a systematic profiling method	PDA Ch. 5.2 · PPDW Ch. 6	Quiz 2 · Practice 3
3	**Aggregation and grouping** — split-apply-combine, aggregation traps, average-of-averages, non-additive measures	PDA Ch. 10	Quiz 2 · Practice 3
4	**Text and missing values** — string methods, regular expressions, the rows a rule does not fit, NaN behaviour and strategy	PDA Ch. 7.1, 7.4 · PPDW Ch. 7	Quiz 2 · Practice 3 · **A1 released**
5	**Tidy data and reshaping** — wide and long, time as grain, resampling, categorical types. **Week 5 is the mid-term.** The topic is carried by the materials and the two practice sessions	PDA Ch. 7.5, 8.3, 11.2, 11.6	**Mid-term (20)** · Practice 3 · **A1 due (8)**
6	**Data formats** — CSV, JSON, Excel, Parquet; nested JSON; reading from an API	PDA Ch. 6 · PPDW Ch. 4, 5	Quiz 2 · Practice 3 · **A2 released and due (8)**
7	**Joining tables** — inner, left, right and outer joins, rows lost or added in a join, join validation, keys and identity	PDA Ch. 7, 8.1, 8.2 · PPDW Ch. 6–8	Quiz 2 · Practice 3 · **A3 released and due (8)**
8	**Synthesis and review** — where these skills go next, exam preparation	PDA Ch. 13 · PPDW Ch. 9	**Final exam (35)**, after the practice sessions

The Reading column lists book chapters. Each week's pre-lecture notebook and video are posted on Moodle.

Practice sessions run in fixed weekly slots per group. Check the timetable for your group.

Still to be confirmed

Item	Status
Office hours	TBC
Assignment-talk rubric	Published before the first round of talks

Nothing in this list changes a rule after the work it governs has started.