

Week 3 • Aggregation and grouping

STAT163 — Data Manipulation Essentials

Oleh Omelchenko

2026-09-14

Pre-lecture quiz

Twelve enrolments

student	program	course	credits	grade
Anna	economics	Statistics	6	88
Anna	economics	Microeconomics	6	74
Anna	economics	Game Theory	4	91
Taras	economics	Statistics	6	55
Taras	economics	Microeconomics	6	58
Daryna	economics	Statistics	6	79
Daryna	economics	Game Theory	4	83
Maksym	engineering	Statistics	6	92
Maksym	engineering	Databases	5	87
Bohdan	engineering	Databases	5	71
Bohdan	engineering	Algorithms	5	95
Iryna	engineering	Algorithms	5	57

One row per enrolment. This log of twelve rows is all you are given.

How many different questions can these twelve rows answer?

Name as many as you can.

Today

- **01 Change the grain** · The grain is what one row stands for

- **02 Add a column** · Build what the log does not give you

- **03 Which rows you count once** · One question, two correct answers

- **04 An event log** · The same moves on a table where time is a column

- **05 Before your practice**

1 • Changing the grain

Three tables built from the log

THE LOG, COLOURED BY STUDENT

student	program	course	credits	grade
Anna	economics	Statistics	6	88
Anna	economics	Microeconomics	6	74
Anna	economics	Game Theory	4	91
Taras	economics	Statistics	6	55
Taras	economics	Microeconomics	6	58
Daryna	economics	Statistics	6	79
Daryna	economics	Game Theory	4	83
Maksym	engineering	Statistics	6	92
Maksym	engineering	Databases	5	87
Bohdan	engineering	Databases	5	71
Bohdan	engineering	Algorithms	5	95
Iryna	engineering	Algorithms	5	57

ONE ROW PER STUDENT • 6 ROWS

student	rows	credits
Anna	3	16
Bohdan	2	10
Daryna	2	10
Iryna	1	5
Maksym	2	11
Taras	2	12

Who is taking the most credits?

Three tables built from the log

THE LOG, COLOURED BY COURSE

student	program	course	credits	grade
Anna	economics	Statistics	6	88
Anna	economics	Microeconomics	6	74
Anna	economics	Game Theory	4	91
Taras	economics	Statistics	6	55
Taras	economics	Microeconomics	6	58
Daryna	economics	Statistics	6	79
Daryna	economics	Game Theory	4	83
Maksym	engineering	Statistics	6	92
Maksym	engineering	Databases	5	87
Bohdan	engineering	Databases	5	71
Bohdan	engineering	Algorithms	5	95
Iryna	engineering	Algorithms	5	57

ONE ROW PER COURSE • 5 ROWS

course	rows	programs
Algorithms	2	1
Databases	2	1
Game Theory	2	1
Microeconomics	2	1
Statistics	4	2

Which course is taken by both programs?

Three tables built from the log

THE LOG, COLOURED BY PROGRAM

student	program	course	credits	grade
Anna	economics	Statistics	6	88
Anna	economics	Microeconomics	6	74
Anna	economics	Game Theory	4	91
Taras	economics	Statistics	6	55
Taras	economics	Microeconomics	6	58
Daryna	economics	Statistics	6	79
Daryna	economics	Game Theory	4	83
Maksym	engineering	Statistics	6	92
Maksym	engineering	Databases	5	87
Bohdan	engineering	Databases	5	71
Bohdan	engineering	Algorithms	5	95
Iryna	engineering	Algorithms	5	57

ONE ROW PER PROGRAM • 2 ROWS

program	rows	courses
economics	7	3
engineering	5	3

How many different courses appear in each program?

One row per student

student	courses	credits	mean grade
Anna	3	16	84.3
Bohdan	2	10	83.0
Daryna	2	10	81.0
Iryna	1	5	57.0
Maksym	2	11	89.5
Taras	2	12	56.5

```
per_student = enrolments.groupby("student").agg(  
    courses=("course", "size"),  
    credits=("credits", "sum"),  
    mean_grade=("grade", "mean"),  
)
```

Each line names a new column, then says which column to read and what to do to it. `size` counts rows.

All three columns come from grouping the log once.

- `courses` adds up to 12 and `credits` to 64. Both totals are true of the log. Would the mean grades add up to anything?
- Iryna has the second-lowest mean grade here. Which other column changes how much you trust that number?

One row per program

program	enrolments	students	courses
economics	7	3	3
engineering	5	3	3

```
per_program = enrolments.groupby("program").agg(  
    enrolments=("course", "size"),  
    students=("student", "nunique"),  
    courses=("course", "nunique"),  
)
```

The log holds 12 enrolments, 6 students and 5 courses.

- Add each column down. Two of the three totals are true of the log. Which one is not?
- `students` and `courses` each count how many different values appear, and only one of them adds up. What is different about the other?

One row per program

program	enrolments	students	courses
economics	7	3	3
engineering	5	3	3

```
per_program = enrolments.groupby("program").agg(  
    enrolments=("course", "size"),  
    students=("student", "nunique"),  
    courses=("course", "nunique"),  
)
```

The log holds 12 enrolments, 6 students and 5 courses.

- Add each column down. Two of the three totals are true of the log. Which one is not?
- `students` and `courses` each count how many different values appear, and only one of them adds up. What is different about the other?

Every student belongs to one program, so no student is counted twice and 6 is right. Statistics belongs to both, so the `courses` column counts it twice. A distinct count adds up only when nothing appears in two groups.

Choosing the grain

Decide what one row of your answer stands for, then group the log on it. That gives you a new table. The question you are asking sets the grain.

2 · Adding a column

A question the columns do not answer

Which student did best compared with the others in the same courses?

Nothing in the log says "compared with the others". The columns are student, program, course, credits and grade.

- Say in one sentence what number you would compute for each student.
- What would you have to build first, and out of which columns?

Building two new columns

```
enrolments["failed"] = enrolments["grade"] < 60

enrolments["course_mean"] = (
    enrolments.groupby("course")["grade"]
    .transform("mean")
)
```

Both lines add a column to the same twelve rows.

- One of them can be worked out from a single row on its own. The other needs every row of the group. Which is which?
- Neither line changes how many rows the table has. What has changed about what one row can tell you?

The same column at two grains

ONE ROW PER STUDENT

student	rows	fails	has_failed
Anna	3	0	False
Bohdan	2	0	False
Daryna	2	0	False
Iryna	1	1	True
Maksym	2	0	False
Taras	2	2	True

```
enrolments.groupby("course").agg(
    rows=("failed", "size"),
    fails=("failed", "sum"),
    has_failed=("failed", "max"),
)
```

ONE ROW PER COURSE

course	rows	fails	has_failed
Algorithms	2	1	True
Databases	2	0	False
Game Theory	2	0	False
Microeconomics	2	1	True
Statistics	4	1	True

pandas counts True as 1, so `sum` gives the number of failures and `max` is True when the group has at least one.

- Statistics failed 1 of its 4 enrolments and Microeconomics 1 of its 2. Both read `True`. Which column separates them?

Writing the course mean on every row

enrolments, 12 rows

course grade

Statistics	88
Microeconomics	74
Game Theory	91
Statistics	55
Microeconomics	58
Statistics	79
Game Theory	83
Statistics	92
Databases	87
Databases	71
Algorithms	95
Algorithms	57

one row per course, 5 rows

Algorithms	76.0
Databases	79.0
Game Theory	87.0
Microeconomics	66.0
Statistics	78.5

$$(88 + 55 + 79 + 92) / 4 = 78.5$$

the same 12 rows

course course_mean

Statistics	78.5
Microeconomics	66.0
Game Theory	87.0
Statistics	78.5
Microeconomics	66.0
Statistics	78.5
Game Theory	87.0
Statistics	78.5
Databases	79.0
Databases	79.0
Algorithms	76.0
Algorithms	76.0

```
enrolments["vs_course"] = (
    enrolments["grade"]
    - enrolments["course_mean"]
)
```

The twelve-row table and the five-row table hold the same five means.

- Anna scored 88 in a course whose mean is 78.5. Which of the two tables can hold the number 9.5, and what does it mean?
- Name one question the twelve-row table can answer and the five-row table cannot.

Sorting the same students two ways

SORTED BY MEAN GRADE, HIGHEST FIRST

student	mean grade	vs_course	courses
Maksym	89.5	+10.8	2
Anna	84.3	+7.2	3
Bohdan	83.0	+5.5	2
Daryna	81.0	-1.8	2
Iryna	57.0	-19.0	1
Taras	56.5	-15.8	2

```
enrolments.groupby("student").agg(
    mean_grade=("grade", "mean"),
    vs_course=("vs_course", "mean"),
    courses=("course", "size"),
).sort_values("mean_grade", ascending=False)
```

Sorting by `vs_course` instead gives the second ranking.

- Two students swap places between the two rankings. Which two, and which ranking would you defend to them?
- Both rankings are computed correctly. What does each one ignore?

Sorting the same students two ways

SORTED BY MEAN GRADE, HIGHEST FIRST

student	mean grade	vs_course	courses
Maksym	89.5	+10.8	2
Anna	84.3	+7.2	3
Bohdan	83.0	+5.5	2
Daryna	81.0	-1.8	2
Iryna	57.0	-19.0	1
Taras	56.5	-15.8	2

```
enrolments.groupby("student").agg(
    mean_grade=("grade", "mean"),
    vs_course=("vs_course", "mean"),
    courses=("course", "size"),
).sort_values("mean_grade", ascending=False)
```

Sorting by `vs_course` instead gives the second ranking.

- Two students swap places between the two rankings. Which two, and which ranking would you defend to them?
- Both rankings are computed correctly. What does each one ignore?

Taras finished 8 below the mean in Microeconomics and 23.5 below it in Statistics, so his -15.8 averages two very different gaps. Iryna took one course, so her -19.0 comes from a single grade.

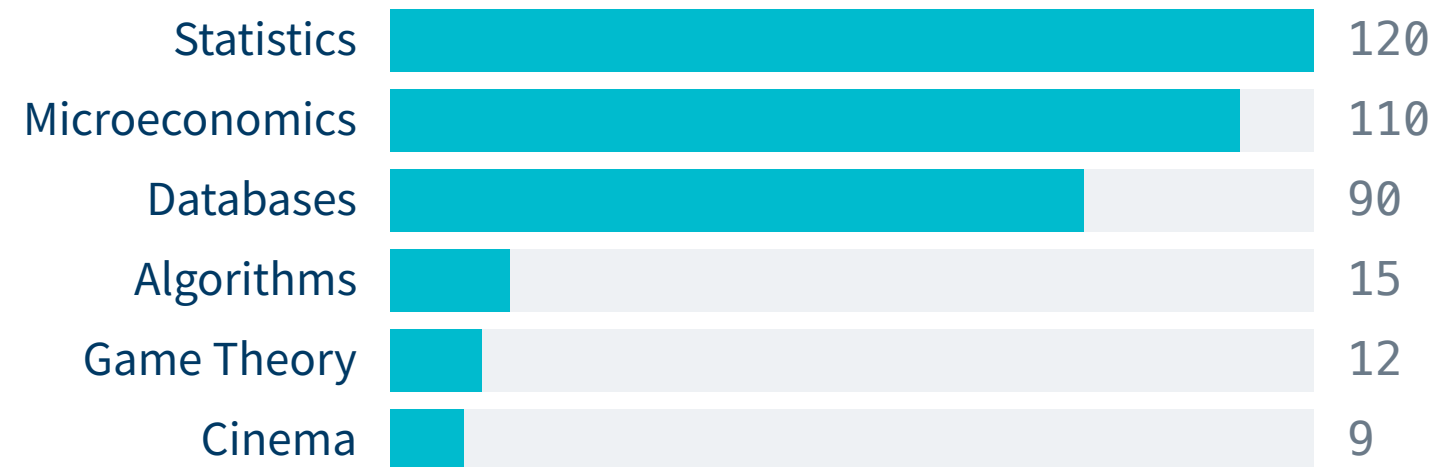
Building the column you need

Build the column the question needs: a condition, or the group's own number on every row. Then group on it, filter on it, or sort by it.

3 • Which rows you count once

How big is a course?

ENROLMENTS IN EVERY COURSE AT THE UNIVERSITY



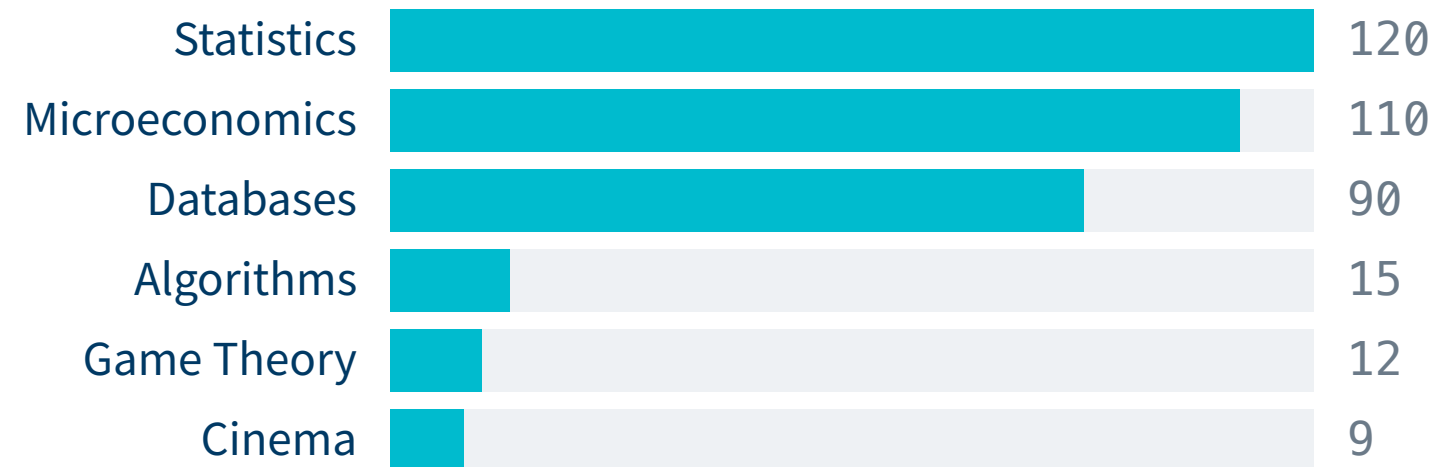
WRITE DOWN TWO NUMBERS

The enrolments log you have been reading is a small part of this university. This is every course in it: 356 enrolments.

- The university publishes the average course size. Work it out.
- Now go to every one of the 356 enrolments, write down how big that course is, and average those numbers. Write down your guess before you compute it.

How big is a course?

ENROLMENTS IN EVERY COURSE AT THE UNIVERSITY



WRITE DOWN TWO NUMBERS

The enrolments log you have been reading is a small part of this university. This is every course in it: 356 enrolments.

- The university publishes the average course size. Work it out.
- Now go to every one of the 356 enrolments, write down how big that course is, and average those numbers. Write down your guess before you compute it.

59.3 averages the six courses. **98.5** averages the 356 enrolments. Both are correct.

Two ways to average the course size

EACH COURSE COUNTED ONCE

`all_enrolments` is the university's full log, 356 rows.

```
per_course = all_enrolments.groupby("course").size()
per_course.mean()
```

59.3

FROM THE 6 COURSE ROWS

EACH ENROLMENT COUNTED ONCE

```
all_enrolments["course_size"] = (
    all_enrolments.groupby("course")["student"]
    .transform("size")
)
all_enrolments["course_size"].mean()
```

98.5

FROM THE 356 ENROLMENT ROWS

- Why is the student number so much bigger?
- Which one should the university publish, and which one describes the class a student actually sits in?

Two ways to average the course size

EACH COURSE COUNTED ONCE

`all_enrolments` is the university's full log, 356 rows.

```
per_course = all_enrolments.groupby("course").size()
per_course.mean()
```

59.3

FROM THE 6 COURSE ROWS

EACH ENROLMENT COUNTED ONCE

```
all_enrolments["course_size"] = (
    all_enrolments.groupby("course")["student"]
    .transform("size")
)
all_enrolments["course_size"].mean()
```

98.5

FROM THE 356 ENROLMENT ROWS

- Why is the student number so much bigger?
- Which one should the university publish, and which one describes the class a student actually sits in?

Averaging courses gives each course one vote. Averaging students gives Statistics 120 votes and Cinema 9.

Group size beside every mean

**Two averages of the same rows can both be correct.
Print the group size, and the reader can tell which
one you computed.**

4 • An event log

A web shop's log

time	visitor	event	item
2026-03-01 09:12	v1	view	lamp
2026-03-01 09:14	v1	addtocart	lamp
2026-03-01 09:21	v1	transaction	lamp
2026-03-01 11:03	v2	view	lamp
2026-03-02 09:47	v2	view	mug
2026-03-02 10:05	v2	view	lamp
2026-03-02 14:22	v3	view	lamp
2026-03-02 14:25	v3	addtocart	lamp
2026-03-03 08:31	v1	view	mug
2026-03-03 19:58	v4	view	desk
2026-03-03 20:02	v4	addtocart	desk
2026-03-03 20:43	v4	transaction	desk
2026-03-03 21:03	v2	transaction	lamp

A web shop, three days. One row each time a visitor did one thing to one item. An **addtocart** row is a visitor putting an item in the cart, the list of things they mean to buy. A **transaction** row is the purchase itself.

How many different questions can this log answer?

Name as many as you can. What is one row here?

One row per day

day	rows	visitors	purchases
2026-03-01	4	2	1
2026-03-02	4	2	0
2026-03-03	5	3	2

```
events["time"] = pd.to_datetime(events["time"])

events["day"] = events["time"].dt.date
events["bought"] = events["event"] == "transaction"

events.groupby("day").agg(
    rows=("event", "size"),
    visitors=("visitor", "nunique"),
    purchases=("bought", "sum"),
)
```

No column in the log holds a day. This one was built from the time.

- Add the `visitors` column down. It comes to 7, and the log has 4 visitors. Which visitors are counted more than once?
- Group by the hour instead and you get a different table. Which of the two is right?

What the summary throws away

ONE ROW PER VISITOR

visitor	views	carts	purchases
v1	2	1	1
v2	3	0	1
v3	1	1	0
v4	1	1	1

v2 and v4 each bought once, and the purchases column says so the same way.

- v4 bought the desk forty-five minutes after seeing it. v2 bought the lamp two days after seeing it. Neither gap is in the table. What would you have to keep to work them out?
- What would you have to keep to answer *did the cart come before the purchase*?

Which item is doing best?

ONE ROW PER ITEM

item	views	purchases	purchases ÷ views
desk	1	1	100%
lamp	4	2	50%
mug	2	0	0%

- Which of these three rates would you trust least, and what makes it weak?
- The lamp was viewed four times by three different visitors. Which column would you need to tell those two numbers apart?

Build the table for the question

Do visitors who use the cart buy more often than visitors who do not?

Nothing in the log says *used the cart* or *bought*.

- Which column would you build first, and from which rows?
- What does one row of your answer stand for?

5 • Before your practice

The practice log

The same kind of log, on bigger scale

YOU WILL

- Build one row per visitor, one row per month and one row per item, and say what each table can answer.
- Build the columns the log does not give you, and use them to filter and to sort.
- Write a short note about what the shop's visitors do, with the count of rows beside every number in it.

What to remember

- **Grain first.** Decide what one row stands for, then group on it.
- **size** counts rows. **nunique** counts different values.
- **Sums and row counts add up. Means never do.** A distinct count adds up only when nothing appears in two groups.
- **transform** writes the group's number on every row, to compare, filter or sort.
- **Two grains, two correct answers.** Say which rows you counted once.
- **Print the group size beside every mean.**

Sources

McKinney, *Python for Data Analysis*, 3rd ed. — Ch. 10, data aggregation and group operations.

Kimball and Ross, *The Data Warehouse Toolkit*, 3rd ed. — which measures can be added across groups, and which cannot.

Questions