

Week 4: Text and missing values

STAT163 — Data Manipulation Essentials

Oleh Omelchenko

2026-09-21

Pre-lecture quiz

Before we start: a short form



Three questions on your latest food delivery experience

We will need this later

forms.gle/Y9VLGHNxLF4pqfu86

Today

- **01 Several values in one cell**

- **02 Regular expressions**

- **03 Checking a rule**

- **04 Missing values**

1. Several values in one cell

One row is one order

order	courier	items	minutes	tip	rating	comment
o1	Koval, Ivan	pizza, cola	28	3	5	great, thank you
o2	Bondar, Olena	burger, no onions, cola	55	0	—	arrived late, cold fries
o3	Koval, Ivan	sushi set, miso soup	31	—	4	—
o4	Shevchuk, Maria	pizza, cola	62	0	—	Late again
o5	Bondar, Olena	salad; chocolate cake	25	2	5	chocolate cake was perfect
o6	Melnyk, Andrii	burger, fries, cola	58	—	—	very late
o7	Shevchuk, Maria	pizza, extra cheese	30	4	4	will order again later
o8	Melnyk, Andrii	pasta, salad	27	0	3	—
o9	Koval, Ivan	burger, no onions	65	—	—	late by 40 minutes
o10	Bondar, Olena	sushi set, cola	33	5	5	quick, not late this time
o11	Shevchuk, Maria	pasta, tiramisu	29	2	4	fine
o12	Melnyk, Andrii	pizza, fries	35	0	4	—

DECIDE BEFORE WE CHECK

How many orders included a cola?

The quick count with contains

```
orders["items"].str.contains("cola").sum()
```

6

ORDERS COUNTED

DECIDE BEFORE WE CHECK

Is 6 right?

order	items
o1	pizza, cola
o2	burger, no onions, cola
o3	sushi set, miso soup
o4	pizza, cola
o5	salad; chocolate cake
o6	burger, fries, cola
o7	pizza, extra cheese
o8	pasta, salad
o9	burger, no onions
o10	sushi set, cola
o11	pasta, tiramisu
o12	pizza, fries

The quick count with contains

order	items
o1	pizza, cola
o2	burger, no onions, cola
o3	sushi set, miso soup
o4	pizza, cola
o5	salad; chocolate cake
o6	burger, fries, cola
o7	pizza, extra cheese
o8	pasta, salad
o9	burger, no onions
o10	sushi set, cola
o11	pasta, tiramisu
o12	pizza, fries

```
orders["items"].str.contains("cola").sum()
```

6

ORDERS COUNTED

DECIDE BEFORE WE CHECK

Is 6 right?

chocolate holds the letters cola. Five orders had a cola.

The count needs one row per item

THREE OF THE ORDERS, ONE ROW PER ORDER

order	items
o1	pizza, cola
o6	burger, fries, cola
o8	pasta, salad

→
split, then explode

ONE ROW PER ORDER AND ITEM

order	item
o1	pizza
o1	cola
o6	burger
o6	fries
o6	cola
o8	pasta
o8	salad

```
items = orders["items"].str.split(", ").explode()  
items.value_counts()
```

The grain changed: one row is now one item of one order. In a Google Form, a multiple-choice answer has the same shape: everything a person ticked is in one cell.

Item counts after the split

ALL 12 ORDERS, 25 ROWS AFTER THE SPLIT

DECIDE BEFORE WE CHECK

Every count is a whole number, nothing is empty, and Python showed no error.

Is this table right?

item	orders
cola	5
pizza	4
burger	3
fries	2
no onions	2
pasta	2
sushi set	2
extra cheese	1
miso soup	1
salad	1
salad; chocolate cake	1
tiramisu	1

Three of these are not items

item	orders
cola	5
pizza	4
burger	3
fries	2
no onions	2
pasta	2
sushi set	2
extra cheese	1
miso soup	1
salad	1
salad; chocolate cake	1
tiramisu	1

order	items
o1	pizza, cola
o2	burger, no onions, cola
o3	sushi set, miso soup
o4	pizza, cola
o5	salad; chocolate cake
o6	burger, fries, cola
o7	pizza, extra cheese
o8	pasta, salad
o9	burger, no onions
o10	sushi set, cola
o11	pasta, tiramisu
o12	pizza, fries

The comma separates two items, and it also sits inside one: **burger, no onions**. In **salad; chocolate cake** the separator is a semicolon, so that cell stays whole.

Your answers from the form

| Open the response sheet

2. Regular expressions

Which orders complain about a late delivery?

order	minutes	comment
o1	28	great, thank you
o2	55	arrived late, cold fries
o3	31	—
o4	62	Late again
o5	25	chocolate cake was perfect
o6	58	very late
o7	30	will order again later
o8	27	—
o9	65	late by 40 minutes
o10	33	quick, not late this time
o11	29	fine
o12	35	—

DECIDE BEFORE WE CHECK

There is no column for it. Which orders would you count?

Which orders complain about a late delivery?

order	minutes	comment
o1	28	great, thank you
o2	55	arrived late, cold fries
o3	31	—
o4	62	Late again
o5	25	chocolate cake was perfect
o6	58	very late
o7	30	will order again later
o8	27	—
o9	65	late by 40 minutes
o10	33	quick, not late this time
o11	29	fine
o12	35	—

```
c = orders["comment"]  
c.str.contains("late")
```

6

ORDERS COUNTED

Which orders complain about a late delivery?

order	minutes	comment
o1	28	great, thank you
o2	55	arrived late, cold fries
o3	31	—
o4	62	Late again
o5	25	chocolate cake was perfect
o6	58	very late
o7	30	will order again later
o8	27	—
o9	65	late by 40 minutes
o10	33	quick, not late this time
o11	29	fine
o12	35	—

```
c = orders["comment"]
c.str.contains("[Ll]ate")
```

7

ORDERS COUNTED

[Ll] is either of the two letters.

Which orders complain about a late delivery?

order	minutes	comment
o1	28	great, thank you
o2	55	arrived late, cold fries
o3	31	—
o4	62	Late again
o5	25	chocolate cake was perfect
o6	58	very late
o7	30	will order again later
o8	27	—
o9	65	late by 40 minutes
o10	33	quick, not late this time
o11	29	fine
o12	35	—

```
c = orders["comment"].dropna()
(
    c.str.contains("[Ll]ate")
    & ~c.str.contains("chocolate")
    & ~c.str.contains("later")
)
```

5

ORDERS COUNTED

Reordering the parts of a name

THE FOUR COURIERS IN THE ORDERS TABLE

courier
Bondar, Olena
Koval, Ivan
Melnyk, Andrii
Shevchuk, Maria

Reordering the parts of a name

THE FOUR COURIERS IN THE ORDERS TABLE

courier		wanted	
Bondar	Olena	Olena	Bondar
Koval	Ivan	Ivan	Koval
Melnyk	Andrii	Andrii	Melnyk
Shevchuk	Maria	Maria	Shevchuk

Reordering the parts of a name

THE FOUR COURIERS IN THE ORDERS TABLE

courier	wanted
Bondar, Olena	Olena Bondar
Koval, Ivan	Ivan Koval
Melnyk, Andrii	Andrii Melnyk
Shevchuk, Maria	Maria Shevchuk

`match ( \w+ ), ( \w+ )`

`replace \2 \1`

`\w+` is one or more letters or digits. Each pair of brackets is a group:
`\1` is the first group, `\2` the second.

```
orders["courier"].str.replace(r"(\w+), (\w+)", r"\2 \1", regex=True)
```

One pattern for a whole log line

A LOG FILE OF THE DELIVERY APP: SEVERAL PROGRAMS WRITE WHAT HAPPENED, ONE LINE EACH

```
2026-09-21 09:17:31 INFO [orders] order o2 accepted, 3 items
2026-09-21 09:17:32 ERROR [auth] login failed for user 42
10.0.3.17 - - [21/Sep/2026:09:17:33 +0300] "GET /api/orders/o2 HTTP/1.1" 200 512
2026-09-21 09:17:35 INFO [courier] Koval assigned to order o2
{"ts":"2026-09-21T09:17:36+03:00","level":"warn","svc":"billing","msg":"retry 2 of 3"}
21.09.2026 09:17:40 payment card 4242 4242 4242 4242 amount 389.00 UAH approved
2026-09-21 09:17:41 INFO [orders] order o2 paid
10.0.3.22 - - [21/Sep/2026:09:17:44 +0300] "POST /api/login HTTP/1.1" 401 87
2026-09-21 09:17:44 ERROR [auth] login failed for user 42
Traceback (most recent call last):
  File "routing.py", line 88, in assign
KeyError: 'courier_id'
2026-09-21 09:17:52 WARN [courier] no courier free in zone 4
21.09.2026 09:18:02 payment card 5555 5555 5555 4444 amount 1240.50 UAH declined
2026-09-21 09:18:03 ERROR [orders] order o6 payment declined
10.0.3.17 - - [21/Sep/2026:09:18:05 +0300] "GET /api/orders/o6 HTTP/1.1" 200 498
{"ts":"2026-09-21T09:18:09+03:00","level":"info","svc":"billing","msg":"refund queued"}
2026-09-21 09:18:10 INFO [courier] Bondar picked up order o4
21.09.2026 09:18:15 payment card 4000 0566 5566 5556 amount 215.00 UAH approved
2026-09-21 09:18:16 INFO [orders] order o9 accepted, 2 items
2026-09-21 09:18:31 WARN [orders] order o4 is 20 minutes behind
10.0.3.40 - - [21/Sep/2026:09:18:33 +0300] "GET /health HTTP/1.1" 200 2
```

One pattern for a whole log line

A LOG FILE OF THE DELIVERY APP: SEVERAL PROGRAMS WRITE WHAT HAPPENED, ONE LINE EACH

```
2026-09-21 09:17:31 INFO [orders] order o2 accepted, 3 items
2026-09-21 09:17:32 ERROR [auth] login failed for user 42
10.0.3.17 - - [21/Sep/2026:09:17:33 +0300] "GET /api/orders/o2 HTTP/1.1" 200 512
2026-09-21 09:17:35 INFO [courier] Koval assigned to order o2
{"ts":"2026-09-21T09:17:36+03:00","level":"warn","svc":"billing","msg":"retry 2 of 3"}
21.09.2026 09:17:40 payment card 4242 4242 4242 4242 amount 389.00 UAH approved
2026-09-21 09:17:41 INFO [orders] order o2 paid
10.0.3.22 - - [21/Sep/2026:09:17:44 +0300] "POST /api/login HTTP/1.1" 401 87
2026-09-21 09:17:44 ERROR [auth] login failed for user 42
Traceback (most recent call last):
  File "routing.py", line 88, in assign
KeyError: 'courier_id'
2026-09-21 09:17:52 WARN [courier] no courier free in zone 4
21.09.2026 09:18:02 payment card 5555 5555 5555 4444 amount 1240.50 UAH declined
2026-09-21 09:18:03 ERROR [orders] order o6 payment declined
10.0.3.17 - - [21/Sep/2026:09:18:05 +0300] "GET /api/orders/o6 HTTP/1.1" 200 498
{"ts":"2026-09-21T09:18:09+03:00","level":"info","svc":"billing","msg":"refund queued"}
2026-09-21 09:18:10 INFO [courier] Bondar picked up order o4
21.09.2026 09:18:15 payment card 4000 0566 5566 5556 amount 215.00 UAH approved
2026-09-21 09:18:16 INFO [orders] order o9 accepted, 2 items
2026-09-21 09:18:31 WARN [orders] order o4 is 20 minutes behind
10.0.3.40 - - [21/Sep/2026:09:18:33 +0300] "GET /health HTTP/1.1" 200 2
```

pattern `(\d{4}-\d{2}-\d{2}) (\d{2}:\d{2}:\d{2}) (\w+) \[(\w+)\] (.+)`

One column per group

pattern `(\d{4}-\d{2}-\d{2})` `(\d{2}:\d{2}:\d{2})` `(\w+)` `\[(\w+)\]` `(.+)`

`str.extract` ON THE FIRST SEVEN LINES OF THE LOG

line	date	time	level	service	message
1	2026-09-21	09:17:31	INFO	orders	order o2 accepted, 3 items
2	2026-09-21	09:17:32	ERROR	auth	login failed for user 42
3	—	—	—	—	—
4	2026-09-21	09:17:35	INFO	courier	Koval assigned to order o2
5	—	—	—	—	—
6	—	—	—	—	—
7	2026-09-21	09:17:41	INFO	orders	order o2 paid

10

OF 22 LINES
MATCHED

12

LINES WITH NO
MATCH

`\[` is a real bracket. `.+` is
everything that is left.

Hiding card numbers

THE THREE PAYMENT LINES OF THE SAME LOG

before	after
payment card 4242 4242 4242 4242 amount 389.00 UAH approved	payment card **** * 4242 amount 389.00 UAH approved
payment card 5555 5555 5555 4444 amount 1240.50 UAH declined	payment card **** * 4444 amount 1240.50 UAH declined
payment card 4000 0566 5566 5556 amount 215.00 UAH approved	payment card **** * 5556 amount 215.00 UAH approved

match `\d{4} \d{4} \d{4} ( \d{4} )`

replace `**** * \1`

Only the last four digits are in brackets, so only they are kept, as `\1`.

3. Checking a rule

False positives and false negatives

4 ORDERS ARE REALLY LATE: 02, 04, 06, 09

order	minutes	comment
o1	28	great, thank you
o2	55	arrived late, cold fries
o3	31	—
o4	62	Late again
o5	25	chocolate cake was perfect
o6	58	very late
o7	30	will order again later
o8	27	—
o9	65	late by 40 minutes
o10	33	quick, not late this time
o11	29	fine
o12	35	—

```
c = orders["comment"]
c.str.contains("late")
```

the rule's result	orders
counted, and right	3
counted, and wrong: a false positive	3
missed: a false negative	1

False positives and false negatives

4 ORDERS ARE REALLY LATE: 02, 04, 06, 09

order	minutes	comment
o1	28	great, thank you
o2	55	arrived late, cold fries
o3	31	—
o4	62	Late again
o5	25	chocolate cake was perfect
o6	58	very late
o7	30	will order again later
o8	27	—
o9	65	late by 40 minutes
o10	33	quick, not late this time
o11	29	fine
o12	35	—

```
c = orders["comment"]
c.str.contains("[Ll]ate")
```

the rule's result	orders
counted, and right	4
counted, and wrong: a false positive	3
missed: a false negative	0

False positives and false negatives

4 ORDERS ARE REALLY LATE: 02, 04, 06, 09

order	minutes	comment
o1	28	great, thank you
o2	55	arrived late, cold fries
o3	31	—
o4	62	Late again
o5	25	chocolate cake was perfect
o6	58	very late
o7	30	will order again later
o8	27	—
o9	65	late by 40 minutes
o10	33	quick, not late this time
o11	29	fine
o12	35	—

```
c = orders["comment"].dropna()
(
    c.str.contains("[Ll]ate")
    & ~c.str.contains("chocolate")
    & ~c.str.contains("later")
)
```

the rule's result	orders
counted, and right	4
counted, and wrong: a false positive	1
missed: a false negative	0

Four rules compared

4 ORDERS ARE REALLY LATE. POINT AT A RULE TO SEE ITS ROWS

the rule on the comment	orders counted
<code>contains("late")</code>	6
<code>contains("[Ll]ate")</code>	7
<code>contains("late")</code> , without <code>chocolate</code> and <code>later</code>	4
<code>contains("[Ll]ate")</code> , without <code>chocolate</code> and <code>later</code>	5

DECIDE BEFORE WE CHECK

Which rule would you trust?

order	minutes	comment
o1	28	great, thank you
o2	55	arrived late, cold fries
o3	31	—
o4	62	Late again
o5	25	chocolate cake was perfect
o6	58	very late
o7	30	will order again later
o8	27	—
o9	65	late by 40 minutes
o10	33	quick, not late this time
o11	29	fine
o12	35	—

Four rules compared

4 ORDERS ARE REALLY LATE. POINT AT A RULE TO SEE ITS ROWS

the rule on the comment	orders counted	right	false positives	false negatives
contains("late")	6	3	3	1
contains("[Ll]ate")	7	4	3	0
contains("late") , without chocolate and later	4	3	1	1
contains("[Ll]ate") , without chocolate and later	5	4	1	0

DECIDE BEFORE WE CHECK

Which rule would you trust?

order	minutes	comment
o1	28	great, thank you
o2	55	arrived late, cold fries
o3	31	—
o4	62	Late again
o5	25	chocolate cake was perfect
o6	58	very late
o7	30	will order again later
o8	27	—
o9	65	late by 40 minutes
o10	33	quick, not late this time
o11	29	fine
o12	35	—

How to check a rule

- Count the rows the rule did not match, and read some of them.
- Read some of the rows it did match.
- Write down both numbers: false positives and false negatives.

Which error costs more?

the rule on the comment	false positives	false negatives
<code>contains("[Ll]ate")</code>	3	0
<code>contains("late")</code> , without <code>chocolate</code> and <code>later</code>	1	1

The late orders are needed twice:

- for a monthly report on the share of late deliveries
- to pay money back to every customer whose order was late

Which rule for each?

4. Missing values

A zero and a missing value in the tip column

order	minutes	tip	rating
o1	28	3	5
o2	55	0	—
o3	31	—	4
o4	62	0	—
o5	25	2	5
o6	58	—	—
o7	30	4	4
o8	27	0	3
o9	65	—	—
o10	33	5	5
o11	29	2	4
o12	35	0	4

mean tip	
over the 9 orders that have one	1.78
after <code>fillna(0)</code>	1.33

A zero is a customer who tipped nothing. A missing tip is a cash order: nobody recorded it.

The ratings that are missing

order	minutes	tip	rating
o1	28	3	5
o2	55	0	—
o3	31	—	4
o4	62	0	—
o5	25	2	5
o6	58	—	—
o7	30	4	4
o8	27	0	3
o9	65	—	—
o10	33	5	5
o11	29	2	4
o12	35	0	4

DECIDE BEFORE WE CHECK

The mean of the 8 ratings is **4.25**.

Is the true mean higher, lower, or about the same?

The ratings that are missing

order	minutes	tip	rating
o1	28	3	5
o2	55	0	—
o3	31	—	4
o4	62	0	—
o5	25	2	5
o6	58	—	—
o7	30	4	4
o8	27	0	3
o9	65	—	—
o10	33	5	5
o11	29	2	4
o12	35	0	4

DECIDE BEFORE WE CHECK

The mean of the 8 ratings is **4.25**.

Is the true mean higher, lower, or about the same?

All four missing ratings are on the orders that took 55 minutes or more.

Before you drop or fill

- A value is missing for a reason. Find the reason first.
- Look at which rows have the missing values. If they are one group, dropping those rows changes the answer.
- – and **n/a** are text, not missing values.

What to remember

- **Several values in one cell:** split, one row per value. The grain changes.
- **A separator can sit inside a value.** Check the pieces.
- **With a pattern you find, extract, reorder and hide text.**
- **A rule has false positives and false negatives.** Count both.
- **A zero is a value. A missing value is not.** `-` is text.
- **Before `dropna()` or `fillna()`:** which rows are missing, and why?

Bonus: which LLMs students on this course use

27 ANSWERS ON THE FORM FROM THE FIRST LECTURE, SEVERAL TOOLS PER PERSON

tool	people
Gemini	23
Claude	16
ChatGPT	11
Grok	2
DeepSeek	1

the whole answer	people
Claude, Gemini	10
ChatGPT, Gemini	4
Gemini	4
ChatGPT	3
ChatGPT, Claude, Gemini	2
four other combinations	1 each

KSE gives every student Gemini. 19 of the 27 combine two or more tools.

Which table answers which question?

Sources

McKinney, *Python for Data Analysis*, 3rd ed. — Ch. 7.1 handling missing data, Ch. 7.4 string manipulation.

McGregor, *Practical Python Data Wrangling and Data Quality* — Ch. 7, cleaning and transforming data.

Questions