

ELLYPUNK

Applied

WEEKLY DIGEST / T201

14-20 September 2026



This week

14-20 September 2026

Available archived editions, in their original date order.

THURSDAY / 17 September 2026

R Language Basics: Vectors, Data Frames, and Summaries

3

[Open original online](#)

FRIDAY / 18 September 2026

Data Lineage: Trace a Number Back to Its Source

8

[Open original online](#)

SATURDAY / 19 September 2026

Pasta Night: Salt, Sauce, Save the Water

13

[Open original online](#)

SUNDAY / 20 September 2026

Sketchbook Composition: Lead the Eye Before the Details

18

[Open original online](#)

Only retained editions are included; gaps have not been filled.

Copyright Ellypunk. Contains AI-generated artwork.

ellypunk.com

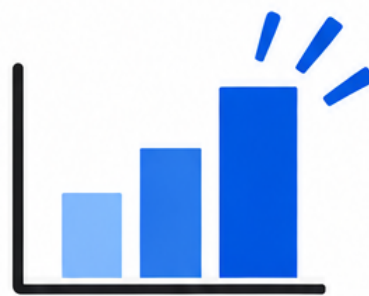
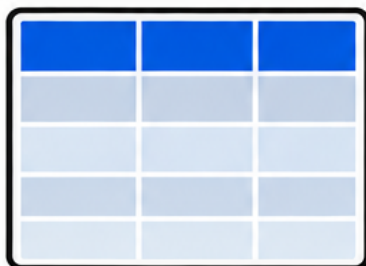
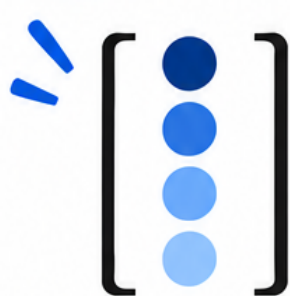
R LANGUAGE BASICS

**VECTORS, DATA FRAMES,
AND SUMMARIES**



Objective: turn named data into useful answers.

Start with a **vector**. Build a **table**. Summarise one clear question.



Small, checked steps
make analysis
reproducible.

VECTORS: ONE ORDERED SEQUENCE

PAGE 2



$$\begin{bmatrix} 12 & 18 & 15 \end{bmatrix} \rightarrow \frac{12 + 18 + 15}{3} = 15$$

`c()` combines values.

`c(12, 18, 15)`

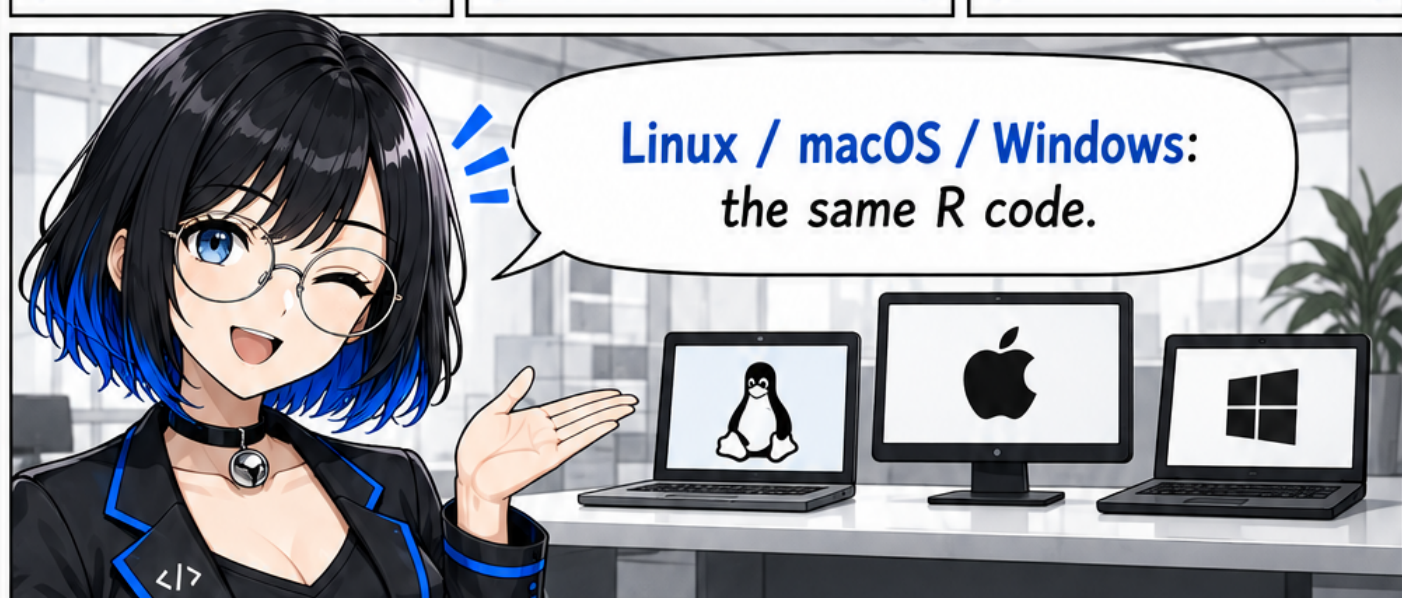
`<-` assigns a name.

`sales <- c(12, 18, 15)`

`mean()` returns the average.

`mean(sales)`

Linux / macOS / Windows:
the same R code.



DATA FRAMES: TABLES WITH NAMES

```
orders <- data.frame(
  region = c("North", "South"),
  sales = c(12, 18)
)
```

region

[1] "North"
[2] "South"

sales

[1] 12
[2] 18

data.frame()



region

North

South

sales

12

18

Columns are **variables**. Rows are **observations**.

Keep related vectors the **same length**.

</>

WORKED EXAMPLE: CHECK THE ANSWER



orders\$**sales**

order_id	sales
1	12
2	18

mean(orders\$**sales**)

```
> mean(orders$sales)
[1] 15
```

Problem: calculate average sales.



Action: select sales with \$.



Check: $(12 + 18) / 2 = 15$

$$12 + 18 = 30$$

$$30 / 2 = \underline{15}$$



Result: mean(orders\$**sales**) is **15**.

```
> mean(orders$sales)
[1] 15
```



A quick **manual check** catches many mistakes.

R BASICS RECAP

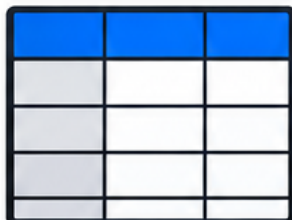
PAGE 5



1. Name objects clearly.
2. Use vectors for one sequence.
3. Use data frames for tables.
4. Inspect the result before trusting it.



`c(1, 2, 3)`



R code is the **same**
on Linux / macOS /
Windows.



Small, checked
steps make
analysis
reproducible.

Data Lineage:

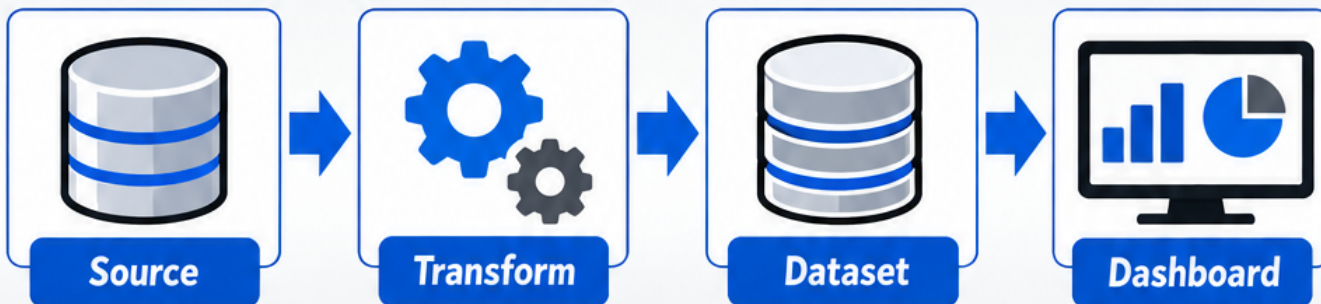
Trace a Number Back to Its Source

Objective:

explain a surprising metric by following its evidence trail.



The Evidence Trail



Elly:

A dashboard number is a **claim**. Lineage shows where it came from, what changed it, and who owns each step.



Start from the number. Ask for its upstream evidence.

Follow the trail backward

Do not guess from a chart. Compare run IDs, row counts, and definitions.

1

Name the metric and time window.



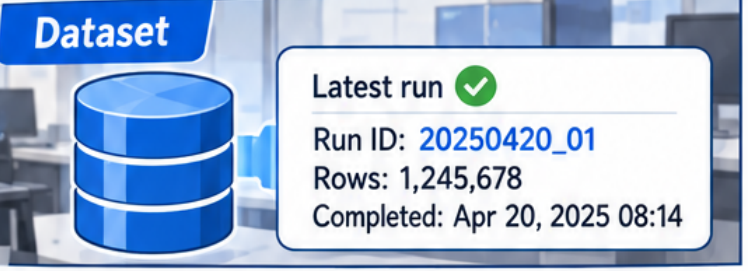
2

Find the dashboard query or semantic definition.



3

Identify the dataset and its latest successful run.



4

Follow inputs and transformations to the source.



5

Stop at the first changed or missing evidence.



Make lineage **useful,** not decorative

Capture **four facts** at every handoff:

DATA
LINEAGE
BUILDS
TRUST

1



Input:
exact dataset
and version

2



Process:
job, logic,
and run ID

3



Output:
dataset, owner,
and freshness

4



Meaning:
metric definition
and grain



Failed run

Guardrail:

record failed runs too.
A missing edge is a blind spot.

Elly:

Ownership tells
you who can
confirm the
next fact.

WORKED EXAMPLE: WHY DID REVENUE DOUBLE?

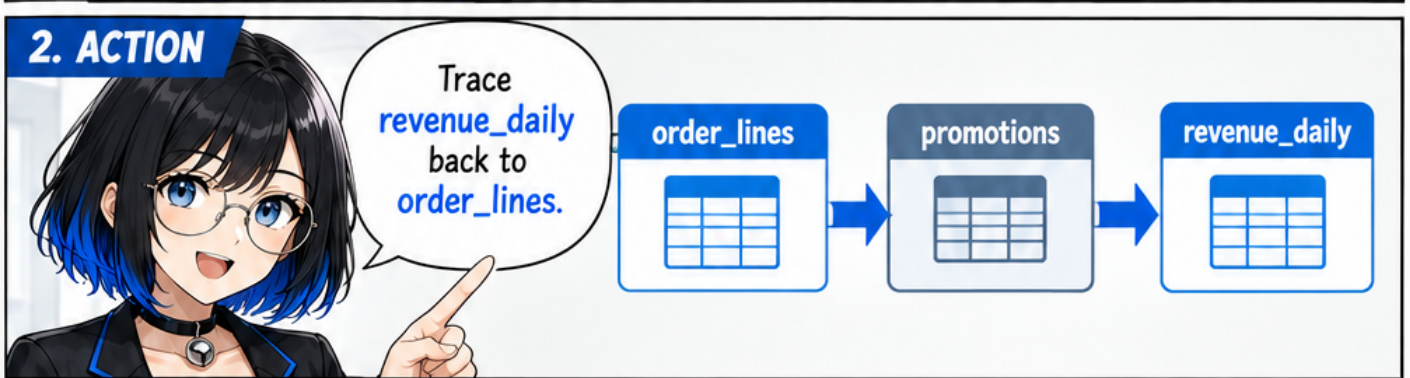
1. PROBLEM



Dashboard revenue is £240k. Yesterday it was £120k.

Revenue
£240k

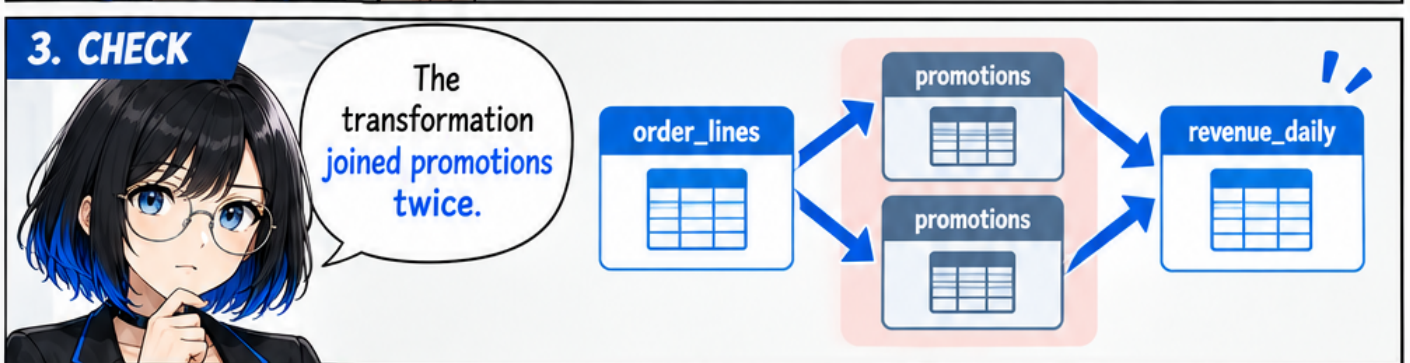
2. ACTION



Trace revenue_daily back to order_lines.

order_lines → promotions → revenue_daily

3. CHECK



The transformation joined promotions twice.

order_lines → promotions → revenue_daily

4. IMPROVEMENT



Fix the join. Re-run with a new run ID.

5. VERIFY



£120k returns. Record the changed job and owner.

Revenue
£120k

Elly: Lineage narrows the question; a check proves the fix.

PAGE 5

Lineage checklist

DATA
PEOPLE
BUILD
A BRIGHTER
TOMORROW



- ☒ Metric has a clear definition and grain
- ☒ Every output has an owner
- ☒ Inputs, transformations, and run IDs are recorded
- ☒ Failed runs and manual changes are visible
- ☒ Changes include impact and verification evidence

Source



Transform



Curated Data



Dashboard



Elly: If you cannot trace a number, do not trust it yet.

Final tip:
Treat lineage as an investigation map, not a poster.



THE SIMPLE RHYTHM



1. Boil plenty of water.



2. Salt it; skip the oil.



3. Add pasta and stir early.



4. Set a timer — then taste before it ends.



THE THREE THINGS TO KEEP

BIG POT: Room to move.



TIMER + TASTE: Aim for a little bite.



MUG: Save a cup of starchy pasta water before draining.



SAFETY:

Steam burns —
use a long spoon
and dry oven
gloves.

WORKED EXAMPLE: QUICK TOMATO SPAGHETTI

PAGE 4



Problem: Sauce looks thick; pasta is nearly ready.



Action: Move pasta into warm tomato sauce. Add a small splash of saved water. Toss for one minute.



Check: Every strand looks coated, not soupy.



Improve: Too thick? Add another splash. Too loose? Toss a little longer.



ELLY'S PASTA-NIGHT CHECKLIST

- 
- ☐ Salted boiling water
 - ☐ Stir early
 - ☐ Taste, don't guess
 - ☐ Save pasta water
 - ☐ Finish in the sauce

The secret ingredient is the water you nearly poured away. ♡

Good Food
Brighter Days
♡



FIND THE FIRST GLANCE

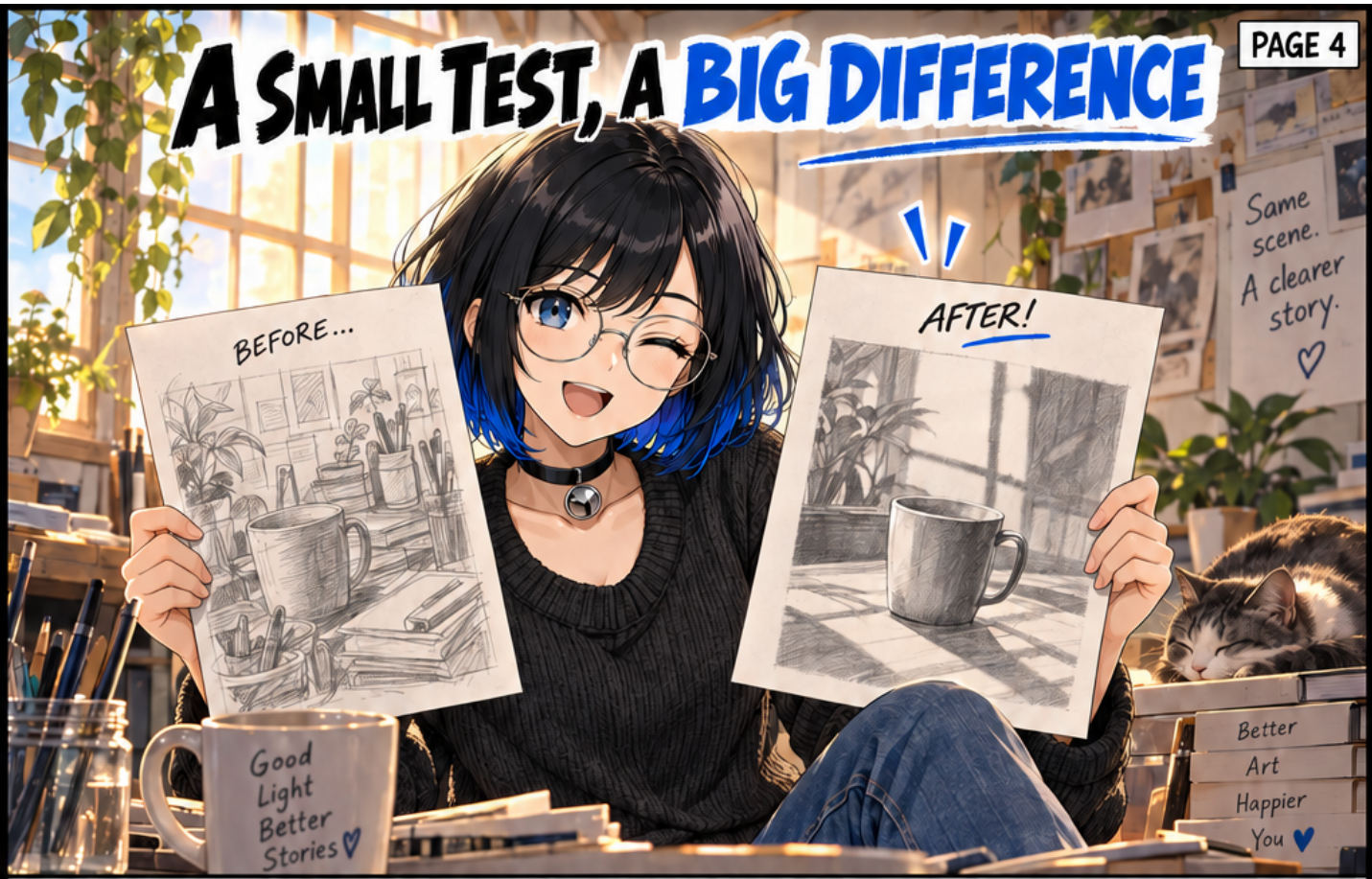


LET THE EYE WANDER - GENTLY

Lines, overlap, and even empty space can whisper: look this way.



A SMALL TEST, A BIG DIFFERENCE



MAKE ROOM FOR THE STORY

