Setup

We load the catalogue dataset from CSV (semicolon-separated), and create an `outputs/` folder where we'll save derived tables (CSV) and figures (PNG).
We keep the original data intact and only add new derived columns/tables for analysis.

0) Loading and cleaning the dataset

We load the catalogue from CSV (semicolon-separated).

Two small structural adjustments are made:

1. The pandas index is not used or exported.
2. The column `kataloga nr.: A.G.` is converted to an integer type (empty values remain empty using nullable integer format).

All original metadata fields remain unchanged. Derived fields will be added separately for analysis.

```
Rows: 853
Columns: ['row_id', 'kataloga nr.: A.G.', 'darba nosaukums', 'Veids', 'gads', 'tehnik
a', 'izmēri', 'atzīmes uz darba', 'atzīmes darba otrā pusē', 'fotogrāfs', 'piezīmes']
```

	row_id	kataloga nr.: A.G.	darba nosaukums	Veids	gads	tehnika	izmēri	atzīmes uz darba	atzīmes darba otrā pusē	fotogrāfs
0	1	<NA>	Hippies. 1968.g., Rīgā. Foto: Māra Brašmane (1...		1968	sudraba želatīna kopija				
1	2	1	Andris Lielajos kapos pie krusta	foto	1968	sudraba želatīna kopija	10,7 x 7,3		Andra Grīnberga piezīme: 1968, Brašmane	Māra Brašmane
2	3	2	Andris ar ziediem matos un krellēm Lielajos kapos	foto	1968	sudraba želatīna kopija	10,7 x 7,8		Andra Grīnberga piezīme: 1968, Brašmane	Māra Brašmane
3	4	3	Andris (tumši zilā velveta žaketā ar guru stāv...	foto	60.gadi	sudraba želatīna kopija	9,7 x 13,5			Māra Brašmane
4	5	4	Andris pie Putnu dārza ar Ritas Kreicas pirmo ...	foto	1968	sudraba želatīna kopija	22,8 x 9		Trīsreiz dažādos periodos rakstīts: 1968; zem ...	Māra Brašmane
5	6	5	Andris pie Putnu dārza ar Rutas Kreices pirmo ...	foto	1968	sudraba želatīna kopija	14,4 x 9,8		Andra Grīnberga piezīme: Brašmane	Māra Brašmane
6	7	6	Hipiji Doma laukumā (Andris guļ fotogrāfijas ...	foto	1968	sudraba želatīna kopija	12,6 x 13,6		Māras Brašmanes uzlīme: Hippies in Dome Square...	Māra Brašmane
7	8	7	Andris un Andrejs Nalogins	foto	1968	sudraba želatīna kopija	12,2 x 16,9		Andra Grīnberga piezīme: 1968, Brašmane	Māra Brašmane

	row_id	kataloga nr.: A.G.	darba nosaukums	Veids	gads	tehnika	izmēri	atzīmes uz darba	atzīmes darba otrā pusē	fotogrāfs
8	9	8	Andris un hipiji Doma laukumā (xx, xx, Ruta Kr...	foto	1968	sudraba želatīna kopija	11,7 x 17,4		Andra Grīnberga piezīme: Brašmane	Māra Brašmane
9	10	9	Andris un hipiji Doma laukumā - kadrs 1 (Andri...	foto	1968	sudraba želatīna kopija	12 x 17,5		Māras Brašmanes uzlīme: Hippies in Dome Square...	Māra Brašmane

1) Making the series structure explicit

The catalogue contains two row types:

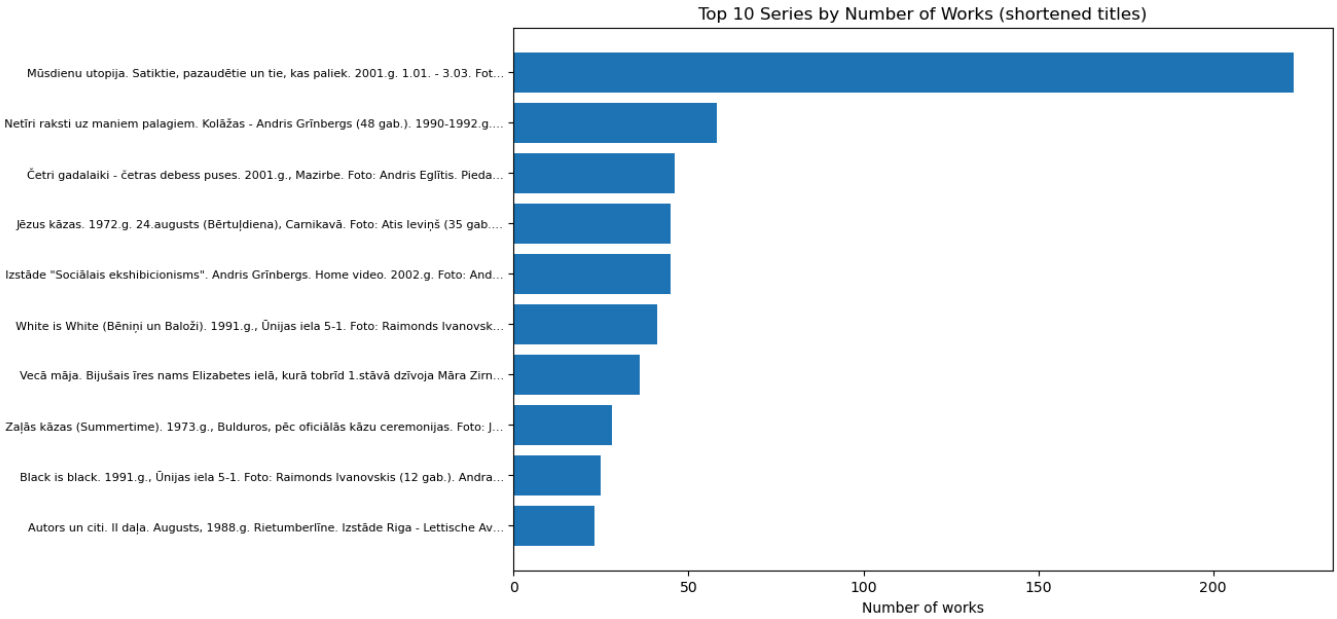
• Series header rows (e.g. “Hippies…”, “Jēzus kāzas…”) • Individual works (actual photos)

We detect header rows automatically and assign each work to the nearest preceding series.

This transforms implicit archival structure into machine-readable data, allowing us to count works per series and compare projects quantitatively.

Make the “Top 10 series” plot readable Series titles in the archive are long descriptions (often including year, location, photographers, participants). For a chart, we display a shortened version of each title, while exporting a separate CSV that preserves the full titles. This keeps the visualization legible without “simplifying away” the archival text.

```
Saved: 06_series_table.csv
Saved: 07_works_with_series.csv
Saved: 08_series_top10_lookup.csv
/var/folders/f_/dvgwc4tx01x9lqpxypf974n40000gn/T/ipykernel_29562/2898562257.py:70: Us
erWarning: This figure was using a layout engine that is incompatible with subplots_a
djust and/or tight_layout; not calling subplots_adjust.
  fig.subplots_adjust(left=0.45)
```




```
Saved: 09_works_per_series_top10_readable.png
```

2) Dates as evidence: keep uncertainty, don’t delete it

The column gads mixes different kinds of historical information: exact years (1968), decades (60.gadi), and uncertain notes (1970 (?), 1969 (2007?)). In this step we do not overwrite the source field. Instead, we add new columns that make the date logic machine-readable:

year_event_min, year_event_max — the earliest/latest plausible year for the event depicted

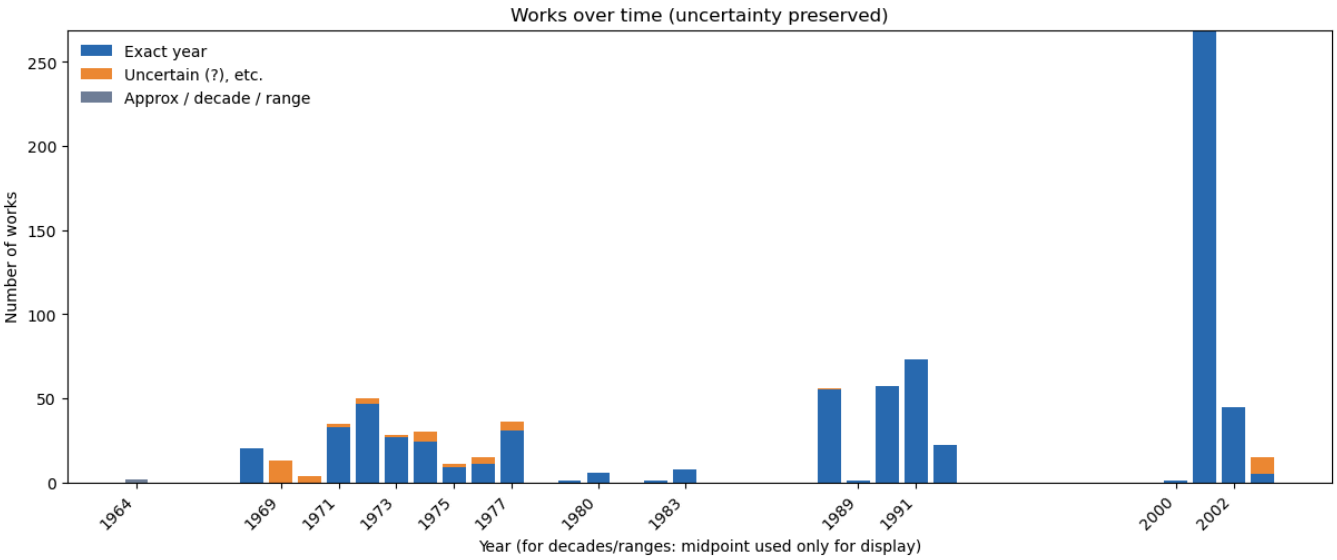
year_event_uncertain — flag for “(?)” uncertainty

year_print_year — a second year if it looks like “event year (print/scan year?)”

year_plot — a single “display year” used only for visualization (e.g., decade → midpoint)

Visualization: a timeline bar chart showing how many works fall in each year, with a separate color for approximate/uncertain dates. This lets us show ambiguity as part of the data, rather than cleaning it away.

```
Saved: 10_works_with_parsed_years.csv
```




```
Saved: 11_timeline_uncertainty.png
```

3) Photographer “lens”: who documents what, when

Catalogs often contain the same photographer written in slightly different ways (extra spaces, line breaks, different punctuation). For analysis, we keep the original fotogrāfs untouched, and add a new helper column fotografs_norm that standardizes the writing without changing the source data.

Then we can ask a simple interpretive question: How does photographic authorship distribute over time?

Visualization: a “rug / scatter timeline” where each dot is one work:

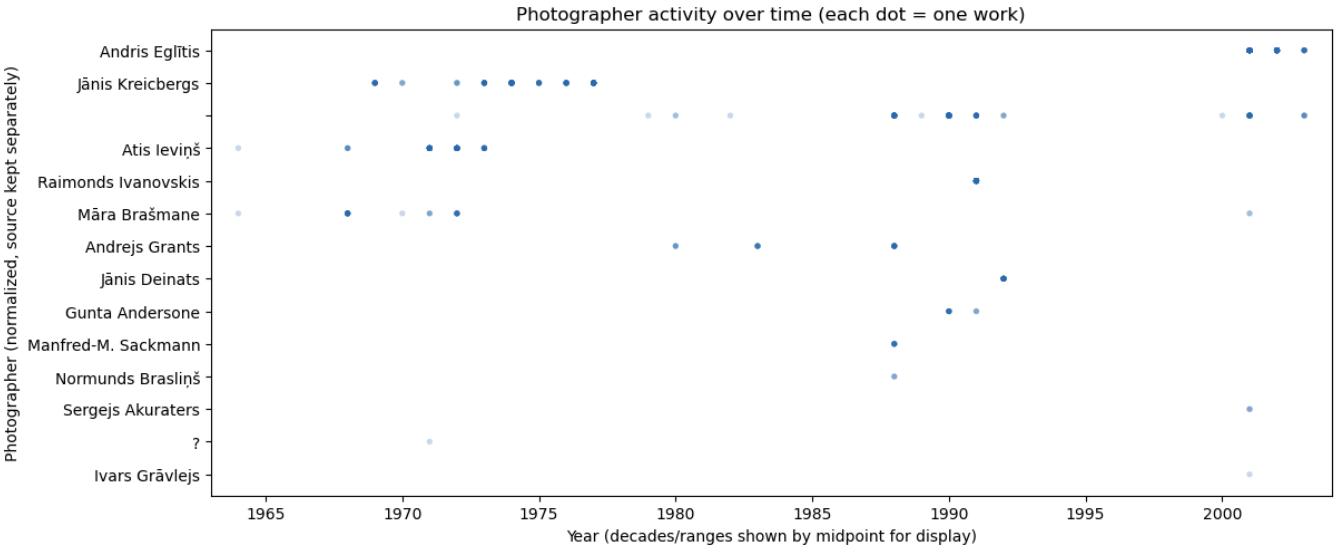
x-axis: year (using year_plot from the previous step)

y-axis: normalized photographer name

transparency: shows density when many works overlap

This gives an immediate “viewpoint map” of the archive: who is active when, and where the dataset is concentrated.

```
Saved: 12_photographer_summary.csv
```




```
Saved: 13_photographer_rug_timeline.png
```

4) Photographer “lens” — improved readability (bubble timeline)

The previous plot is good as a “quick proof”, but it becomes hard to read because many points overlap and all marks look the same.

Here is an improved version that keeps the same idea (photographer vs time), but makes the quantity visible:

We aggregate works: for each (photographer, year) we count how many works there are.

Each year becomes a bubble:

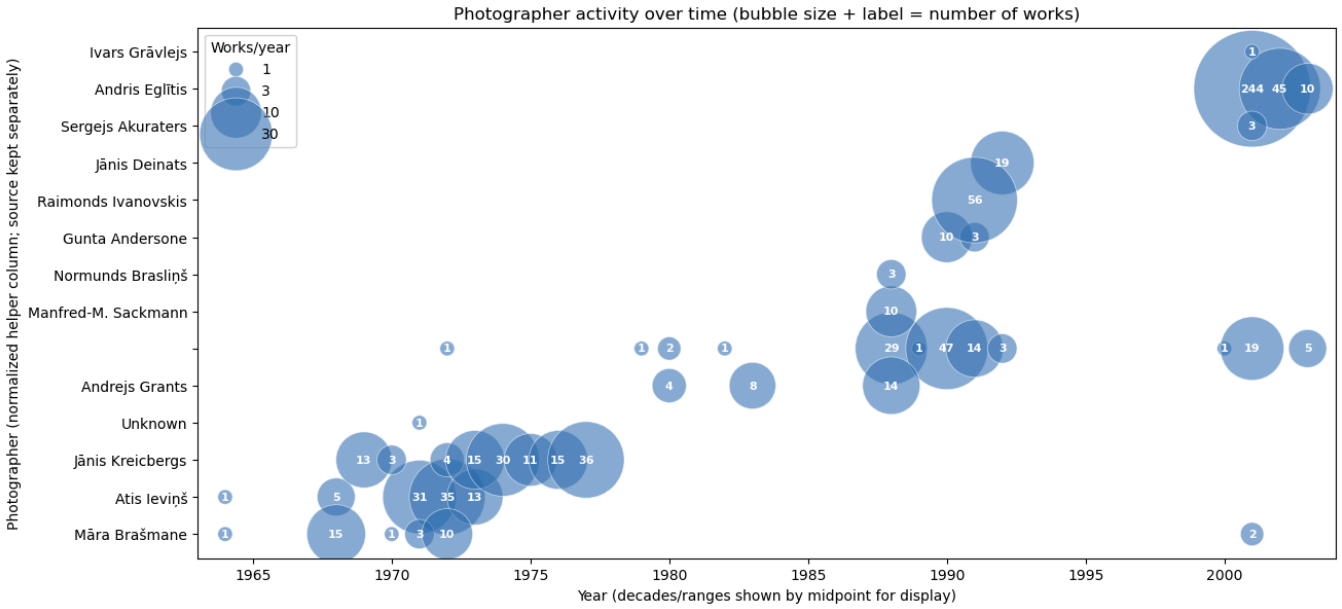
bubble size grows with the count (using a log scale so big values don’t dominate)

the count number is printed inside the bubble

Photographers are ordered by first appearance in the dataset, and the earliest appears at the bottom of the y-axis.

This reads like a compact “activity calendar”: who appears early/late, and where the archive is densest.

```
Saved: 14_photographer_year_counts.csv
```




```
Saved: 15_photographer_bubble_timeline.png
```

5) Photographer “lens” — final polish (color per photographer + better “unknown” handling)

This version keeps the bubble timeline, but improves interpretability:

Each photographer gets their own color, so you can follow one “visual voice” across time at a glance.

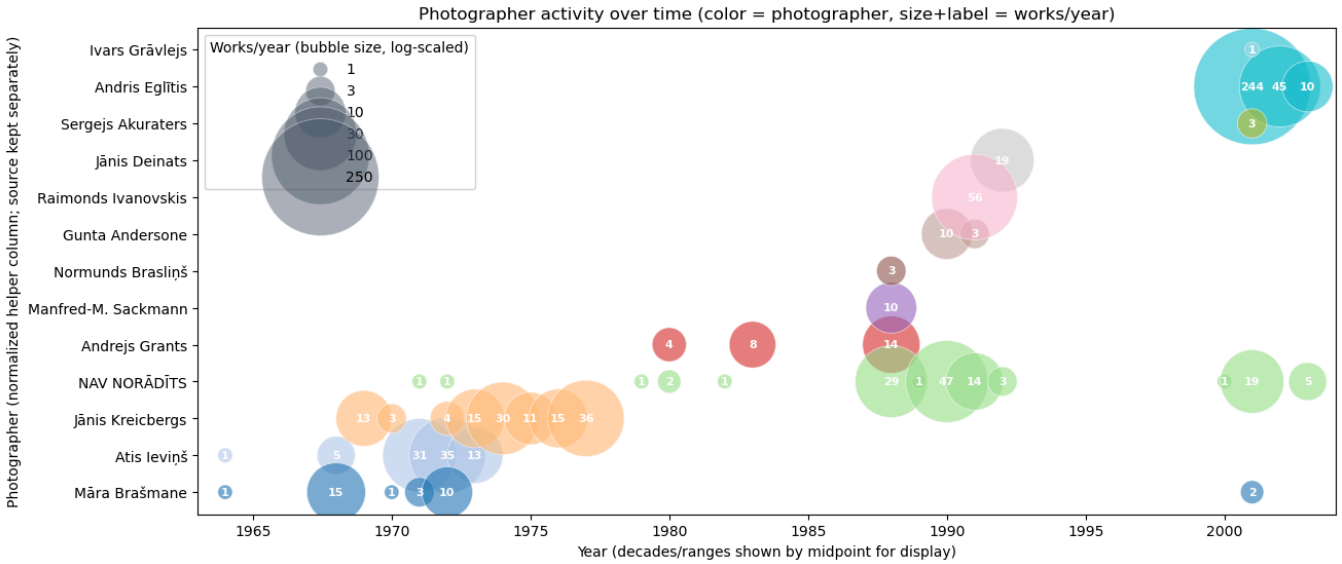
The bubble size still encodes the number of works per (photographer, year). We keep the log scaling because counts range from very small (1–3) to very large (e.g., 200+). Log keeps small values readable without letting big values dominate.

Any missing/empty photographer field is relabeled as “NAV NORĀDĪTS” (“not indicated”), so we don’t get blank rows in the y-axis.

Outputs:

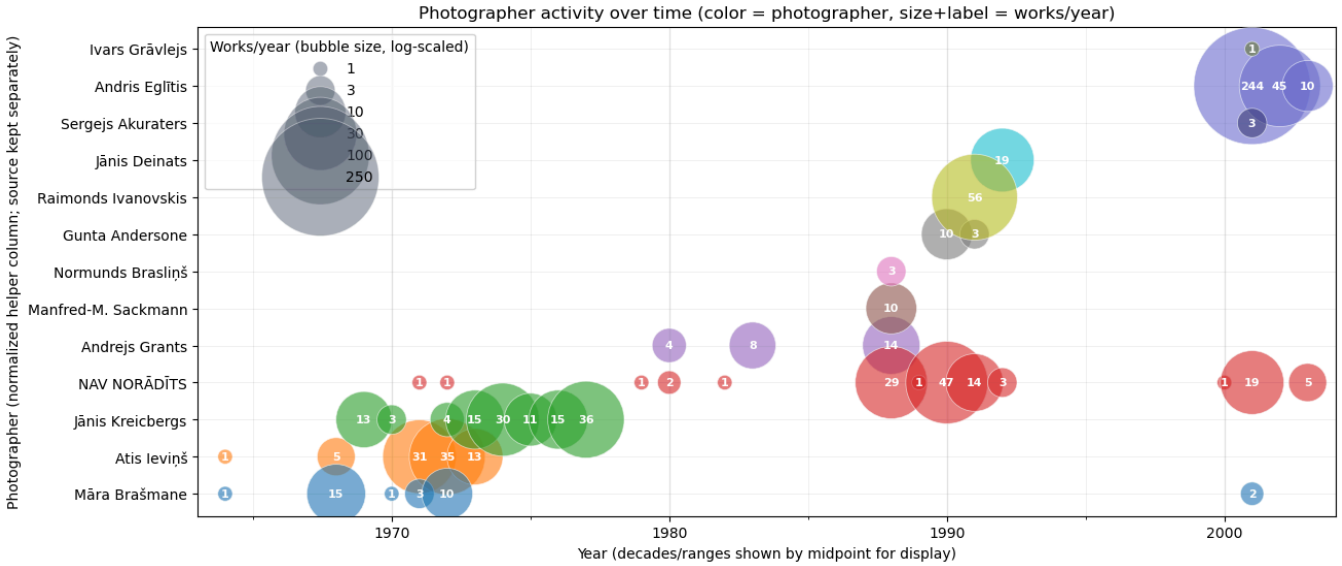
16_photographer_year_counts_named.csv (aggregated counts, with the cleaned label)

```
Saved: 16_photographer_year_counts_named.csv
/var/folders/f_/dvgwc4tx01x9lqpxypf974n40000gn/T/ipykernel_29562/4101457565.py:57: Ma
tplotlibDeprecationWarning: The get_cmap function was deprecated in Matplotlib 3.7 an
d will be removed in 3.11. Use ``matplotlib.colormaps[name]`` or ``matplotlib.colorma
ps.get_cmap()`` or ``pyplot.get_cmap()`` instead.
  cmap = cm.get_cmap("tab20", max(1, len(order)))
```



Saved: 17_photographer_bubble_timeline_colored.png

```
/var/folders/f_/dvgwc4tx01x9lqpxypf974n40000gn/T/ipykernel_29562/3879341849.py:61: MatplotlibDeprecationWarning: The get_cmap function was deprecated in Matplotlib 3.7 and will be removed in 3.11. Use ``matplotlib.colormaps[name]`` or ``matplotlib.colormaps.get_cmap()`` or ``pyplot.get_cmap()`` instead.
  cmap_local = cm.get_cmap(name)
```



Saved: 19_photographer_bubble_timeline_colored_grid.png

6) Co-occurrence network of participants (“Piedalās”) by project/series

Many series descriptions contain a short “cast list” (e.g., Piedalās: …). Here we turn those lists into a simple social proximity map:

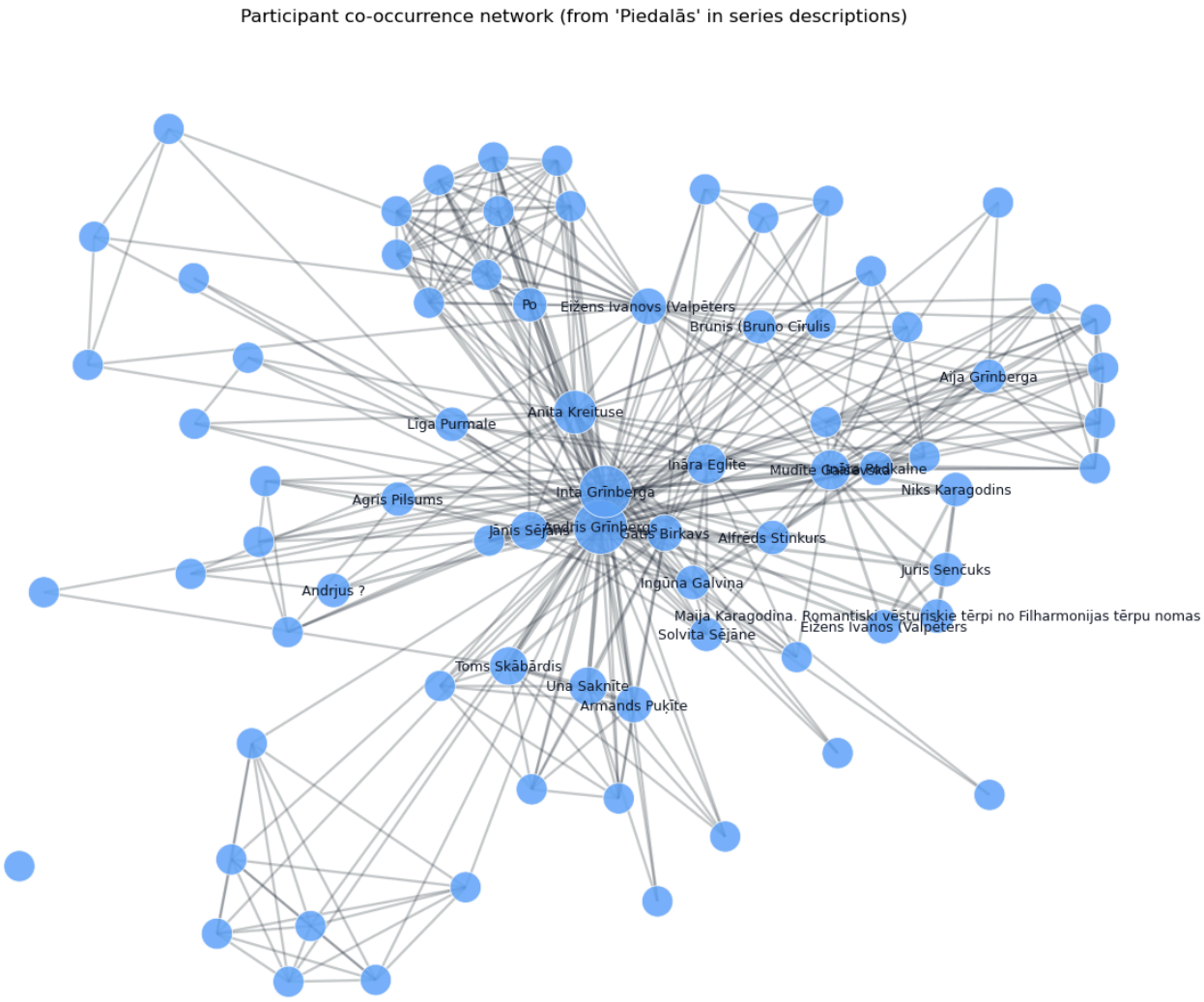
Each node = one named participant (person).

A link between two people = they are listed together in the same series/project.

Thicker links = co-occurs in more than one series (stronger repeated association).

This does not “prove” relationships, but it helps us quickly see clusters, central figures, and who appears together across the archive. We also save two CSVs (nodes + edges) so the same network can be re-used in Gephi or other tools.

```
Saved: 20_series_person_edges.csv
Saved: 21_people_cooccurrence_edges.csv
Saved: 22_people_nodes.csv
```




```
Saved: 23_people_cooccurrence_network.png
```

6b) Improved participant network (cleaner names + community colors + less overlap)

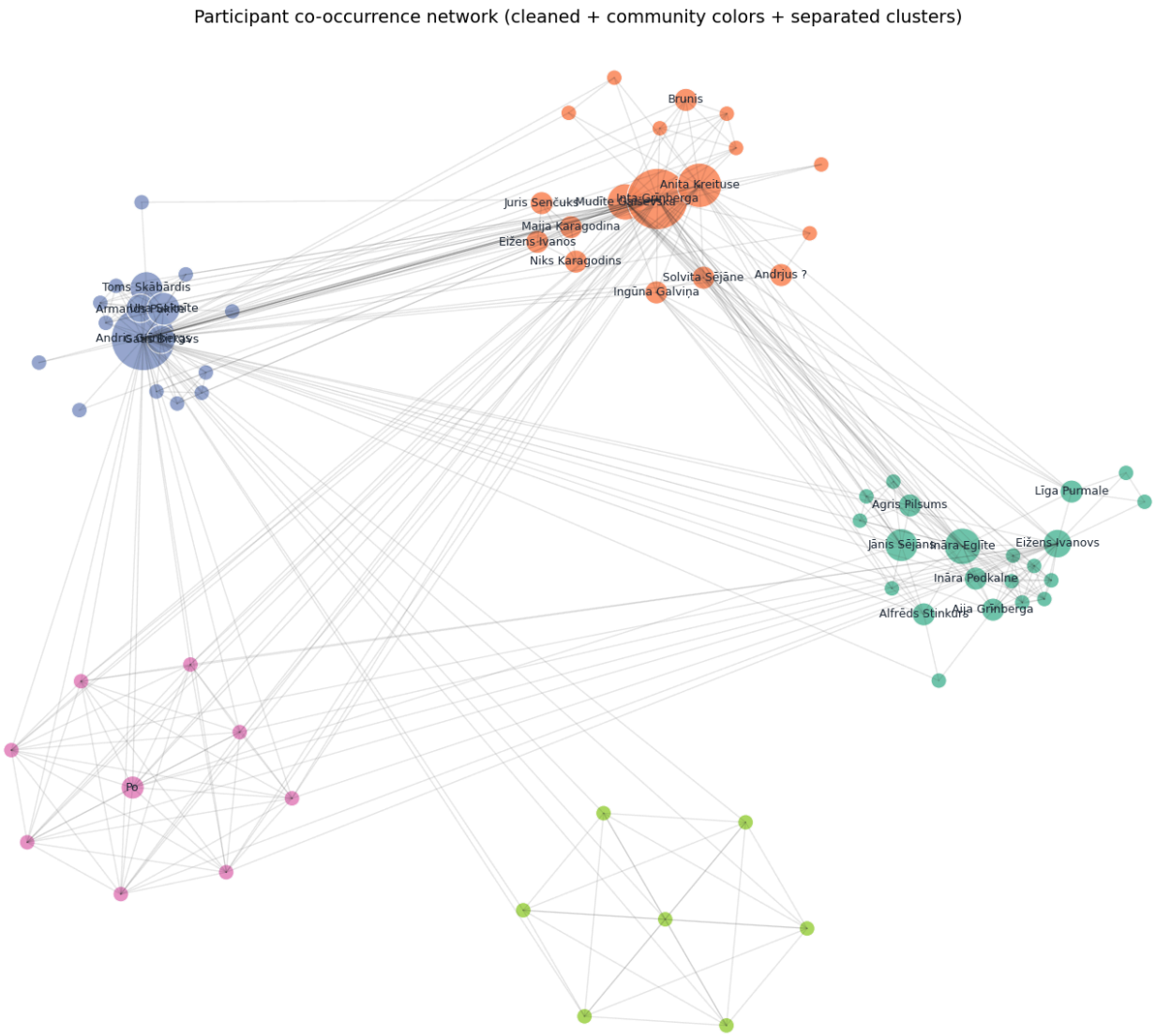
This version keeps the same basic idea (people connected when they appear together in a “Piedalās:” participant list), but improves readability in three ways:

Cleaner extraction: we cut the participant list more carefully so that long non-name phrases don’t become “people” nodes.

Community colors: we automatically detect clusters (groups of people who often appear together) and color them differently.

Less overlap: instead of one tight “hairball”, we place each cluster in its own “region” on the canvas so labels and nodes overlap less.

Output: one improved PNG image.




```
Saved: 24_people_cooccurrence_network_improved.png
```

6c) Participant network (final): label the key people in every cluster

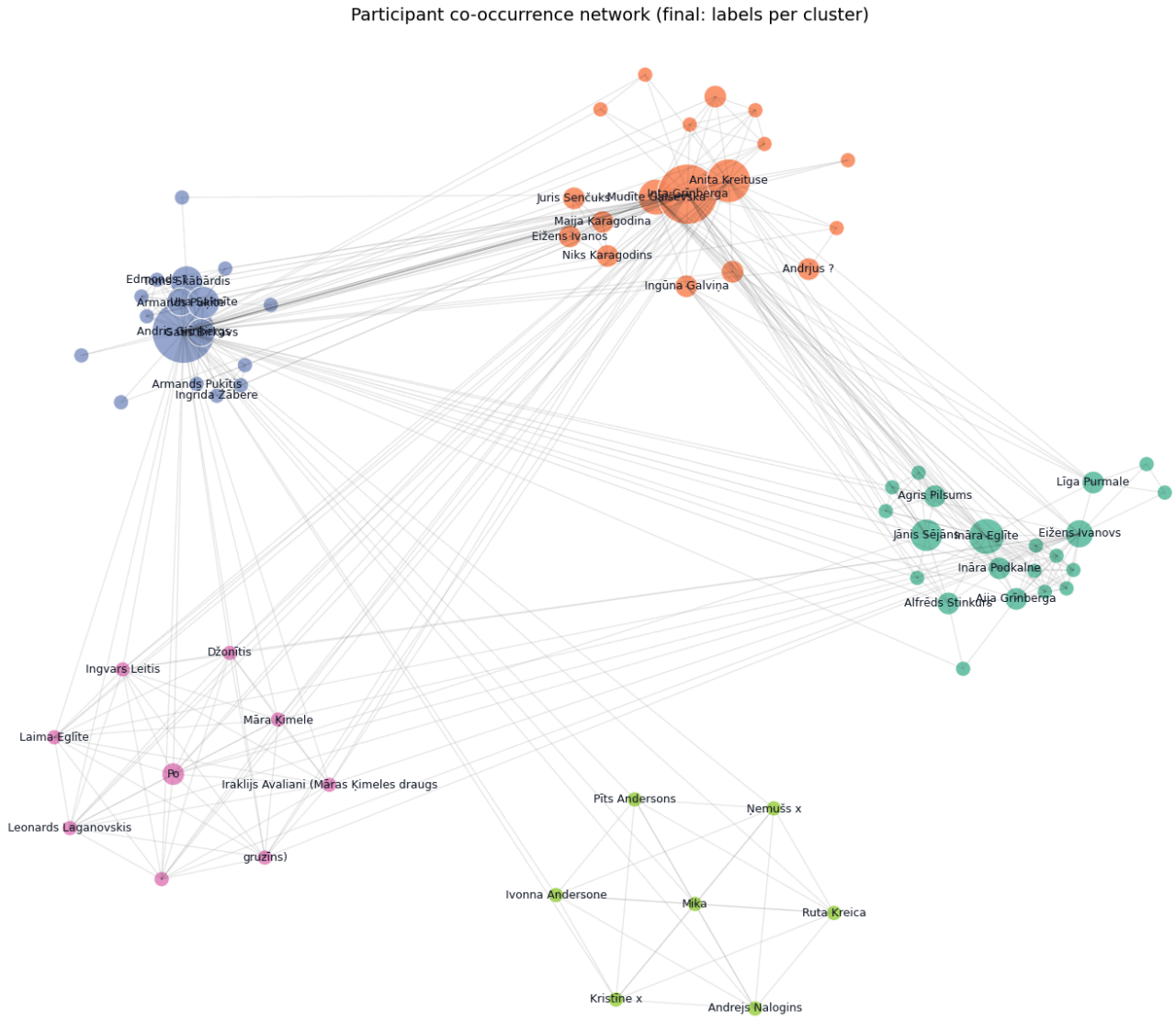
In the previous improved network, labels were shown mainly for the “global top” participants (the most frequently appearing names overall). That makes the biggest cluster readable, but smaller clusters can look anonymous.

This final version keeps the same layout + coloring, but changes the labeling rule:

For each cluster, label the top N people inside that cluster (by how many series they appear in).

Still also label any “globally important” people (so the main figures stay labeled).

Result: every colored cluster becomes interpretable, not just the largest one.



Saved: 25_people_cooccurrence_network_final.png