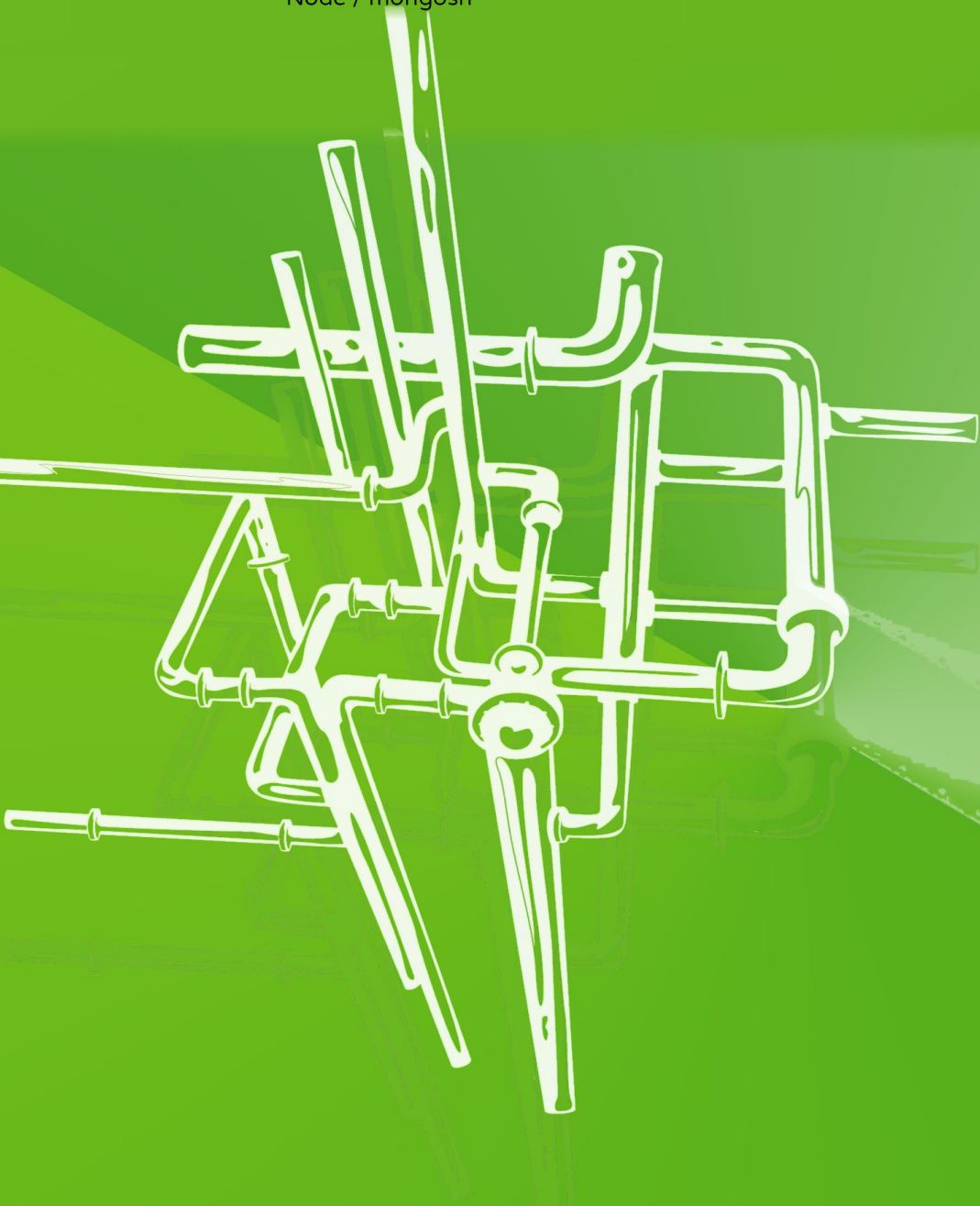


MongoDB Aggregation Pattern Pack

Vol. 1: Reporting & Dashboards

Node / mongosh



John Lynn

MongoDB Aggregation Pattern Pack — Vol. 1: Reporting & Dashboards

Copy-paste aggregation patterns for reporting and dashboards, using Atlas sample data.

John Lynn

This book is available at <https://leanpub.com/mongo-agg-patterns>

This version was published on 2026-10-02



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2026 John Lynn

Contents

About	1
How to read these patterns	3
Leading commas	3
Conventions in this pack	3
Sample data	4
On Atlas	4
On local MongoDB	5
What a sale looks like	6
How to run the examples	7
mongosh	7
Compass	7
Node scripts	7
Pattern 1 – Time-bucket totals	9
When this pattern fits	9
When it does not fit	9
Sample document (trimmed from Atlas)	9
Pipeline	10
Expected output shape	11
Watch out	12
Variations	12
Pattern 2 – Top-N per category	13
When this pattern fits	13
When it does not fit	13
Sample data	13
Pipeline	13
Expected output shape	13

CONTENTS

Watch out	13
Variations	14
Pattern 3 – Running totals / cumulative series	15
When this pattern fits	15
When it does not fit	15
Sample data	15
Pipeline	15
Expected output shape	15
Watch out	15
Variations	16
Pattern 4 – Period-over-period	17
When this pattern fits	17
When it does not fit	17
Sample data	17
Pipeline	17
Expected output shape	17
Watch out	17
Variations	18
Pattern 5 – Null-safe metrics	19
When this pattern fits	19
When it does not fit	19
Sample data	19
Pipeline	19
Expected output shape	19
Watch out	19
Variations	20
Pattern 6 – Multi-collection report via \$lookup	21
When this pattern fits	21
When it does not fit	21
Sample data	21
Pipeline	21
Expected output shape	21
Watch out	21
Variations	22
Pattern 7 – Faceted dashboard payload	23

When this pattern fits	23
When it does not fit	23
Sample data	23
Pipeline	23
Expected output shape	23
Watch out	23
Variations	24
Pattern 8 – Histogram / distribution	25
When this pattern fits	25
When it does not fit	25
Sample data	25
Pipeline	25
Expected output shape	25
Watch out	25
Variations	26
Pattern 9 – Active vs churned in a window	27
When this pattern fits	27
When it does not fit	27
Sample data	27
Pipeline	27
Expected output shape	27
Watch out	27
Variations	28
Pattern 10 – Pagination-friendly report pages	29
When this pattern fits	29
When it does not fit	29
Sample data	29
Pipeline – page and count	29
Keyset variation (preferred for deep pages)	29
Expected output shape (facet form)	29
Watch out	30
Variations	30
What's next	31

About

Welcome! If you need reporting pipelines you can run and adapt, this is a good place to start.

If you've ever stared at an empty aggregation pipeline with a deadline close, you know why this book exists. Maybe you needed sales by week, top products, or this month against last month. Maybe the dashboard needed all of it in one round trip. The MongoDB docs are great, but you also need a few tested reporting patterns that you can run as they are and change to fit your data.

That's what this is: a pattern pack, not a catalog of every aggregation operator. Ten practical recipes for everyday reporting and dashboard work. Each chapter starts from a real question, shows a pipeline you can run against Atlas sample data (we lean on `sample_supplies` a lot), and talks through when the pattern fits, when it doesn't, and the common mistakes to watch for.

What I like to do is keep the pipelines honest and editable. You'll see leading commas so stages are easy to comment out while you test. You'll see the same habits I use when I'm building something for a real report, not a slide deck. The goal is a clear answer you can ship, not clever aggregation.

If you know basic `$match` and `$group`, you're ready. If aggregation is new to you, skim the MongoDB docs first. Then come back when you want working examples to help apply what you learned. Plenty of good data people start right here.

Volume 1 covers reporting and dashboards. The examples use Node and mongosh. The same ten patterns also come in a Python companion, so you can work in the language you like best. If this book helps you, I'd love to write more. Later volumes could cover more reporting, ops diagnostics, or domain packs, all in the same hands-on format.

Examples target MongoDB 7.x / 8.x unless a chapter notes an earlier requirement (for example `$setWindowFields` on 5.0+). Version-sensitive behavior is called out where it matters. Corrections and clarifications are always welcome; Leanpub makes it easy to keep a living manuscript as the material improves.

You're in the right place. Open How to read, load the sample data, and start with Pattern 1. Then jump to the chapter that matches the report you're building.

Happy building – and thank you for reading!

How to read these patterns

Each pattern follows a consistent structure:

- Problem statement
- When it fits / when it does not
- Sample collection
- Pipeline (leading-comma style)
- Expected output shape
- Watch outs
- Short variations

Leading commas

Aggregation examples in this pack use a leading comma on every stage after the first:

```
1 db.sales.aggregate([
2   { $match: { /* ... */ } }
3   , { $group: { /* ... */ } }
4   , { $sort: { /* ... */ } }
5 ])
```

The first stage has no comma. Later stages begin with a comma. That layout makes it easier to reorder, insert, delete, or comment out stages without adjusting a trailing comma on the previous line.

While developing a pipeline, comment out later stages and advance one stage at a time. Starting from `$match` (often with `$limit`) is a practical way to inspect intermediate results.

Conventions in this pack

1. Prefer Atlas sample data so examples are reproducible.
2. Apply `$match` early when it reduces work for later stages.
3. Use the same timezone for date truncation that you use for display.

Sample data

The pipelines in this volume use the official MongoDB Atlas sample datasets. You need:

- **sample_supplies.sales** – Patterns 1–5 and 7–10
- **sample_mflix.movies** and **sample_mflix.comments** – Pattern 6 (\$lookup)

Load that data into **Atlas** or into a **local MongoDB**. The collection names and document shapes are the same either way.

Everything you need to load the samples is in this chapter: Atlas steps, local restore, Windows notes, and the Pattern 6 index. If a chapter needs a join, it says so up front.

* * *

On Atlas

1. Open your cluster in [MongoDB Atlas](#).
2. On the cluster card, open the  menu and choose **Load Sample Dataset**.
3. Wait until the load finishes, then connect with mongosh or Compass.

```
1 use sample_supplies
2 db.sales.countDocuments()
3 db.sales.findOne()
4
5 use sample_mflix
6 db.movies.countDocuments()
7 db.comments.countDocuments()
```

Expect about **5,000** sales, **about 21,000** movies, and **about 41,000** comments.

An M0 free cluster is enough for most chapter examples. Heavier aggregations (especially Pattern 6) can hit free-tier limits. If that happens, switch to local MongoDB. Pattern 6 also benefits from this index:

```
1 use sample_mflix
2 db.comments.createIndex({ movie_id: 1 })
```

* * *

On local MongoDB

1. Install and start [MongoDB Community Server](#). The default port is **27017**.
2. Download and restore the official archive:

```
1 curl -L https://atlas-education.s3.amazonaws.com/sampleddata.archive -o
   ↪ sampledata.archive
2 mongorestore --archive=sampledata.archive --host=127.0.0.1 --port=27017
```

On Windows PowerShell, use `curl.exe` and `mongorestore --archive=sampledata.archive --port=27017`. Install [Database Tools](#) if `mongorestore` is missing.

3. Verify with the same `countDocuments` checks as above.

Prefer local MongoDB when you want to rerun pipelines quickly without Atlas free-tier limits.

* * *

What a sale looks like

Each `sample_supplies.sales` document is one sale: `saleDate`, `items` (with price and quantity), `storeLocation`, `customer`, `purchaseMethod`, and `couponUsed`.

Revenue comes from line items. Use `$unwind` or `$reduce` over `items` when totaling money.

Secondary collection (when a chapter needs a join): `sample_mflix`. Chapters that require it say so explicitly.

How to run the examples

Once the sample data is loaded (see **Sample data**), you can run any chapter pipeline the same way.

mongosh

1. Connect to Atlas or to local MongoDB.
2. Select the database the chapter uses (usually `sample_supplies`; Pattern 6 uses `sample_mflix`).
3. Paste the chapter's `aggregate([...])` block and press Enter.
4. Page through the cursor until you have seen enough rows.

A tiny smoke test before you paste a full pattern:

```
1 use sample_supplies
2
3 db.sales.aggregate([
4   { $limit: 3 }
5   /* paste later stages from a pattern chapter here */
6 ])
```

When you open a pattern chapter, copy its full `db....aggregate([...])` example over this smoke test (or paste it on a fresh line). Keep the leading-comma style from **How to read these patterns** so you can comment stages in and out while you learn.

If a chapter names parameters (date ranges, limits, sort keys), set those first, then run.

Compass

In MongoDB Compass, open the same database and collection, go to the **Aggregations** tab, and rebuild the stages from the chapter (or paste them if your Compass version accepts a full pipeline). The results should match mongosh for the same data.

Node scripts

The examples in this volume are written for mongosh. The same pipeline array works in the Node driver: pass it to `collection.aggregate([...])`. Connection strings and driver setup are outside the scope of these chapters; mongosh is enough to follow along.

Pattern 1 – Time-bucket totals

This pattern produces revenue and counts by day, week, or month for a time-series chart.

A common mistake is to format dates as strings and `$group` on those strings, or to truncate timestamps in UTC while the chart axis uses another timezone. Day boundaries then disagree with what the user sees, especially around midnight and around daylight-saving changes.

The examples use the Atlas sample dataset so you can run them without preparing your own data. Load **Sample Dataset** in Atlas, then use the `sample_supplies.sales` collection. Each document is one sale. Line-item amounts are in the `items` array (price and quantity). Sale-level revenue is the sum of `price × quantity` across those items.

When this pattern fits

- Charts with a continuous time axis
- Daily, weekly, or monthly totals from `saleDate`
- Bucket sizes that `$dateTrunc` supports directly (day, week, month, and related units)

When it does not fit

- Fiscal calendars or custom business weeks – use a calendar collection or precomputed period keys
- Fine-grained histograms (for example sub-second or arbitrary numeric ranges) – use `$bucket` or `$bucketAuto`
- Charts that must show periods with no activity as zeros – this pipeline returns only buckets that contain data; fill missing periods in the application, or `$unionWith` a calendar of dates

Sample document (trimmed from Atlas)

```

1  {
2    _id: ObjectId("5bd761dcae323e45a93ccfe8"),
3    saleDate: ISODate("2015-03-23T21:06:49.506Z"),
4    items: [
5      { name: "notepad", price: NumberDecimal("35.29"), quantity: 2 },
6      { name: "pens", price: NumberDecimal("56.12"), quantity: 5 }
7      // ...
8    ],
9    storeLocation: "Denver",
10   purchaseMethod: "Online",
11   couponUsed: true,
12   customer: { gender: "M", age: 42, email: "cauho@witwuta.sv", satisfaction: 4 }
13 }

```

Pipeline

Pipelines in this pack use **leading commas**: the first stage has no comma; each later stage begins with a comma. That makes it easier to reorder, insert, delete, or comment out stages without fixing a trailing comma on the previous line.

```

1  use sample_supplies
2
3  const timezone = "America/New_York" // use the same timezone the chart displays
4  const unit = "day"                  // "day" | "week" | "month"
5
6  db.sales.aggregate([
7    // Restrict the range early. In production, an index on saleDate is usually
8    ↪ appropriate.
9    { $match: {
10      saleDate: {
11        $gte: ISODate("2015-01-01T00:00:00Z"),
12        $lt:  ISODate("2016-01-01T00:00:00Z")
13      }
14    }
15    // One document per line item, so price × quantity is straightforward
16    , { $unwind: "$items" }
17    , { $set: {
18      lineRevenue: { $multiply: ["$items.price", "$items.quantity"] }
19    }
20  }
21  // Truncate to the bucket start in the display timezone, then aggregate
22  , { $group: {
23    _id: {
24      $dateTrunc: {
25        date: "$saleDate",

```

```

26         unit: unit,
27         timezone: timezone
28         // optional for weeks: startOfWeek: "monday"
29     }
30 },
31 revenue: { $sum: "$lineRevenue" },
32 lineItems: { $sum: 1 },
33 sales: { $addToSet: "$_id" } // distinct sale ids in the bucket
34 }
35 }
36 , { $project: {
37     _id: 0,
38     bucketStart: "$_id",
39     revenue: 1,
40     lineItems: 1,
41     sales: { $size: "$sales" }
42 }
43 }
44 , { $sort: { bucketStart: 1 } }
45 ])
```

When developing a pipeline, it is useful to comment out later stages and work forward from `$match` (often with `$limit`) so you can inspect intermediate results.

```

1 db.sales.aggregate([
2   { $match: {
3     saleDate: {
4       $gte: ISODate("2015-01-01T00:00:00Z"),
5       $lt:  ISODate("2016-01-01T00:00:00Z")
6     }
7   }
8 }
9 , { $limit: 3 }
10 // , { $unwind: "$items" }
11 // , { $set: { lineRevenue: { $multiply: ["$items.price", "$items.quantity"]
12   ↪ } } }
13 // , { $group: { /* ... */ } }
14 ])
```

Expected output shape

```
1  [
2    {
3      bucketStart: ISODate("..."), // start of that local calendar day
4      revenue: NumberDecimal("..."),
5      lineItems: 12,
6      sales: 4
7    }
8    // periods with no matching sales are omitted
9  ]
```

Exact values depend on the sample data loaded in your cluster. The field names and types above are the contract for the chart.

Watch out

Timezone. If `$dateTrunc` uses UTC and the chart labels use `America/New_York`, bucket boundaries will not match the axis. Pass the same `timezone` into `$dateTrunc` that you use for display.

Revenue. Summing `items.price` without multiplying by `quantity`, or without `$unwind` / `$reduce` over `items`, understates revenue.

Distinct sales after \$unwind. `$addToSet: "$_id"` still identifies unique sales because `_id` is the sale id, but large buckets can use significant memory. If you only need a sale count, aggregate that before `$unwind`, or report line-item counts only.

Variations

- **By store:** include `storeLocation` in the `$group _id` together with the `$dateTrunc` expression.
- **Sale count only:** omit `$unwind` and `$set`; `$group` by `$dateTrunc` of `saleDate` with `sales: { $sum: 1 }`.

Apply `$match` as early as practical so later stages process fewer documents.

Pattern 2 – Top-N per category

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When this pattern fits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When it does not fit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Sample data

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Expected output shape

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Watch out

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Variations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pattern 3 — Running totals / cumulative series

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When this pattern fits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When it does not fit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Sample data

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Expected output shape

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Watch out

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Variations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pattern 4 — Period-over-period

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When this pattern fits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When it does not fit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Sample data

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Expected output shape

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Watch out

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Variations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pattern 5 – Null-safe metrics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When this pattern fits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When it does not fit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Sample data

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Expected output shape

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Watch out

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Variations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pattern 6 – Multi-collection report via \$lookup

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When this pattern fits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When it does not fit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Sample data

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Expected output shape

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Watch out

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Variations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pattern 7 — Faceted dashboard payload

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When this pattern fits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When it does not fit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Sample data

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Expected output shape

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Watch out

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Variations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pattern 8 – Histogram / distribution

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When this pattern fits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When it does not fit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Sample data

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Expected output shape

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Watch out

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Variations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pattern 9 — Active vs churned in a window

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When this pattern fits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When it does not fit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Sample data

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Expected output shape

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Watch out

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Variations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pattern 10 — Pagination-friendly report pages

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When this pattern fits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

When it does not fit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Sample data

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Pipeline — page and count

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Keyset variation (preferred for deep pages)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Expected output shape (facet form)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Watch out

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

Variations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.

What's next

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/mongo-agg-patterns>.