

Problem set 8: Data visualization and storytelling

Module 8. Total: 100 points. Format: no-code sandbox, with an optional API extension in Problem 5 using a shared class API key from your instructor.

Before you start

This problem set works with the sandbox's pre-built charts and export tools, not a coding library, and it asks you to practice the editorial judgment that separates a chart that informs from one that misleads. Choose one category table and one brand, house, or artist within it to use throughout, and confirm the exact spelling in the coverage browser before you filter on it, since exact-match filters are spelling-sensitive. You will build an honest pricing power comparison and a quarterly time series, then deliberately misbuild a version of one of them so you can explain, precisely, what went wrong.

Problem 1 (15 points): Choosing the right chart for the question

In four to five sentences, explain what a pricing power comparison across three or four brands should be charted as, and what a sold-lot count over eight to twelve quarters should be charted as, giving a reason for each choice. Then state one reason the two chart types should not be swapped, that is, why charting the brand comparison as a line and the time series as a bar chart would each serve its data poorly.

Problem 2 (20 points): Build the pricing power comparison

Write a SQL query against your chosen category table that computes pricing power (median of `usd_price_decimal` divided by `sale_estimates_high_usd_price`, over sold records with a valid, non-zero high estimate) for three or four brands or houses of your choosing, grouped so each appears as one row. Render the result as a bar chart in the sandbox and describe, in one or two sentences, what you chose for the axis and ordering and why.

```
-- your query here
```

Problem 3 (20 points): Build the quarterly time series

Write a SQL query against the same table that buckets sold lots for your chosen brand, house, or category into calendar quarters and computes a count of sold lots per quarter. Render the result as a line chart, and in two to three sentences, state exactly which of the final quarters on your chart should carry the recency caveat and why.

```
-- your query here
```

Problem 4 (25 points): Build a dishonest version and critique it

Using the same underlying query and result set from either Problem 2 or Problem 3, describe how you would deliberately produce a misleading version of that chart using one specific, named technique, either axis truncation or a cherry-picked time window. Then write a paragraph, in the voice of a chart reviewer, explaining exactly how the misleading version distorts a reader's takeaway even though every number in it is correct, and what an honest caption or footnote would need to say to correct it.

Problem 5 (20 points): The same query via the API (optional code extension)

Using the shared class API key your instructor has provided, write the endpoint and JSON request body you would send to POST /v1/tables/{name}/query to retrieve the raw fields needed to reproduce your Problem 2 chart in a Python plotting library. Note that the query endpoint returns matching rows, not a pre-computed aggregate, so describe in one to two sentences how you would paginate on offset and compute the pricing power figures yourself before plotting them.

```
POST /v1/tables/<table_name>/query
Header: X-API-Key: <class key>
Body:
{
  ...
}
```

Submission. Turn in this file with your SQL, JSON body, chart descriptions, and written answers filled in.