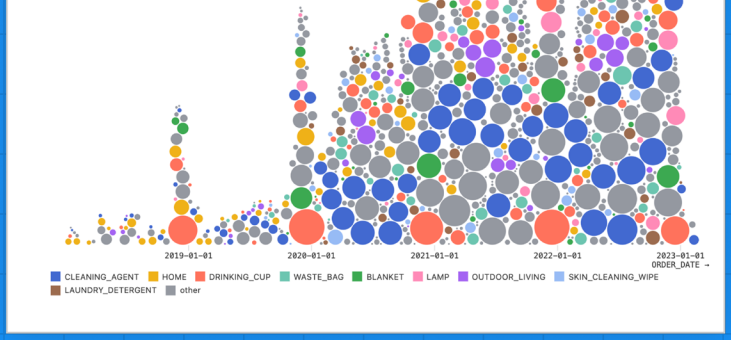
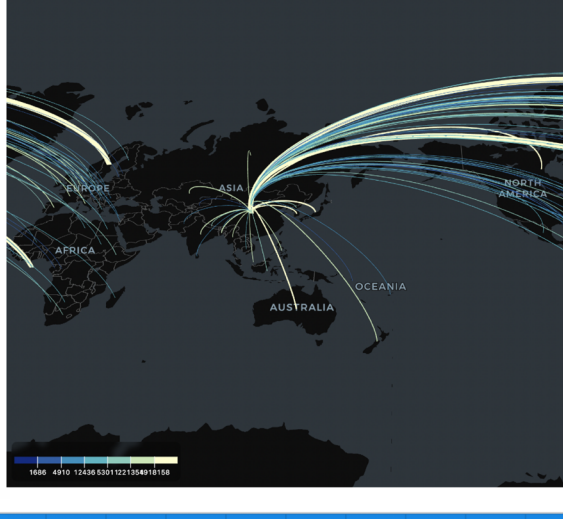
Observable

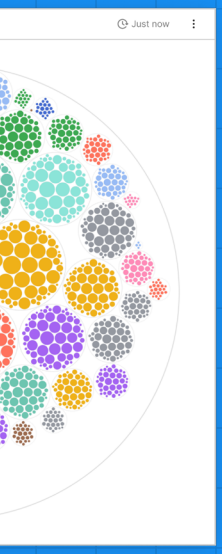
Analyzing Data 101

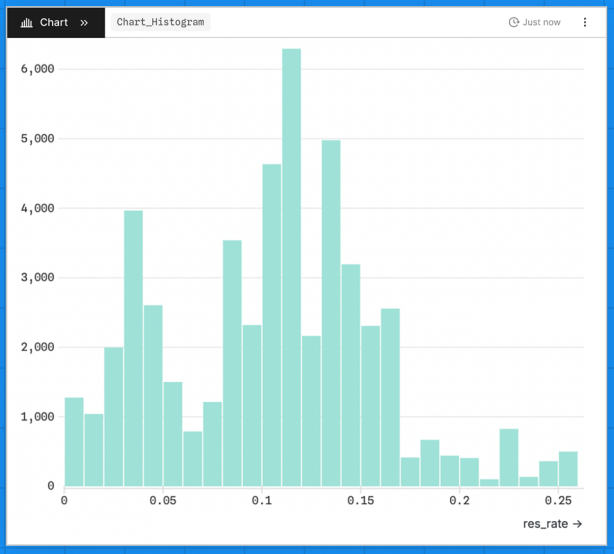
Turn information into insights and make better decisions with data.

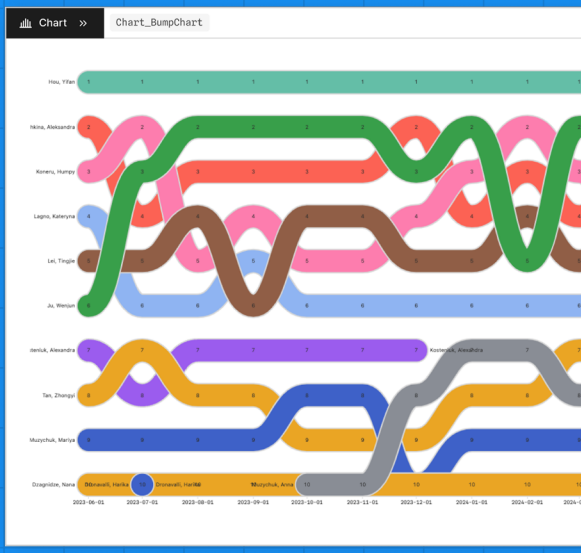
Table of contents







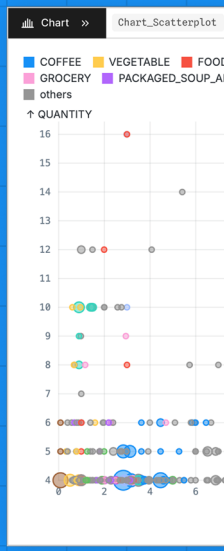


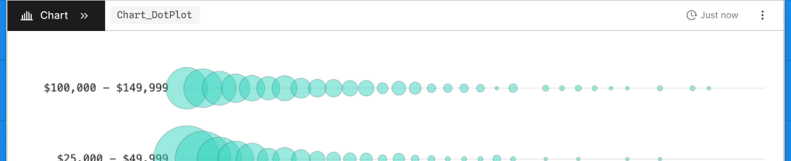


Introduction

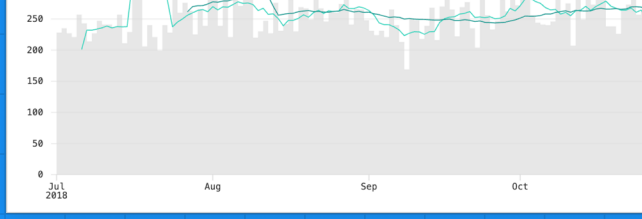
Businesses of all sizes are trying to integrate data into their decision-making process. But, turning data into actionable insights can be difficult to operationalize.

This guide introduces how to effectively analyze data to answer questions about your business, so your company can make stronger and more data-informed decisions. Let's dig in!





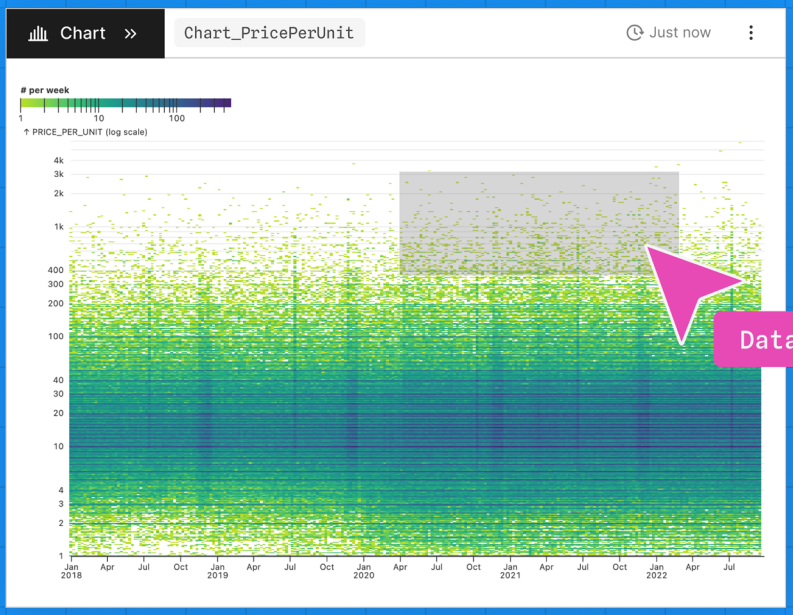
9.99	1	NV	B071JTJ5BR	PHYSICAL_MOVIE
9.99	1	NV	B0BRDFBZCQ	ABIS_BOOK
3.02	2	NV	B0BRDFBZCQ	ABIS_BOOK
51.2	1	NV	B00KYCA4QY	PHYSICAL_TV_SERIES
11.39	1	NV	B07YP2VH4B	CAN_OPENER
6.47	2	NV	B07T3NGD8B	ART_MEDIA_PAPER
7.99	1	NV	B0788YMYCV	PHYSICAL_MOVIE
25.98	1	NV	B09QLP8B45	TOILET_PAPER



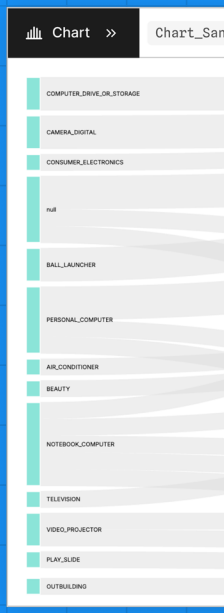
Here's SQL for the total monthly revenue for each Home & Living subcategory

Run

```
SELECT
    CATEGORY,
    DATE_TRUNC('month', ORDER_DATE)
AS month,
    ROUND(SUM(PRICE_PER_UNIT *
QUANTITY), 2) AS monthly_revenue
FROM JOIN_4
GROUP BY CATEGORY, month
ORDER BY month, monthly_revenue DESC
```



Data analyst



Unpacking the data analysis process

When working on data analysis projects, data practitioners take a number of steps to uncover insights and communicate them to others at their organization. The data analysis process typically spans the following stages: data collection, data cleaning, data wrangling, exploratory data analysis, data modeling and visualization, and interpretation and communication.

Data collection

Data can come from many places: spreadsheets, databases, CRM systems, survey tools, or dashboards. The source depends on the kind of data you're working with and established company tools and practices. For instance, sales data may come from a CRM, survey results from a spreadsheet, and website traffic from a database.

Before beginning analysis, data practitioners need to ensure that they have access to all the data they need to answer their questions, and verify that the data is correct. In some instances, you may discover partial or incomplete data. This impacts which questions you can answer, the methods you choose, and how to represent uncertainty in the resulting analysis. This first step is often overlooked, and it leads to many problems down the road. Know your data!

Make sure you know the data source and provenance, which is the data's complete history from when it was first collected to the current form you're working with. Is it raw data, coming straight from a database? If so, which database, exactly? How was it exported (if you're not accessing the database directly)? When was it last updated or exported — in other words, how fresh is the data?

Once you confirm the source, find out if the data has been processed in some way. Are you looking at a subset of the data or all of it? Has it been filtered? Have values been aggregated, manipulated, or manually added to rows? Has somebody already cleaned it up (and if so, what exactly did they do)?

Before moving on to data cleaning, it's a good idea to ensure you aren't making assumptions about the data without confirming if those assumptions are accurate.

What is data analysis?

Data analysis is the process of inspecting, cleaning, transforming, and interpreting data to discover useful insights and inform decision-making.

While data analysis is most commonly associated with business intelligence use cases, it has an important role to play across industries, roles, and business types.

Data cleaning

With the data in hand and a clear understanding of its history, you'll still likely need to do some data cleaning to fix or remove incorrect, incomplete, inconsistent, or duplicated data.

Data cleaning (also called preprocessing) can be done manually in spreadsheets for smaller datasets, but often data analysts use Python, SQL, or R for automated and reproducible cleaning workflows, powered by code.

Issues that are addressed during the data cleaning process include:

Missing values

Missing values are often overlooked and unexplored during data analysis, with many tools quietly applying listwise deletion by default. In listwise deletion, an entire record (row) is excluded if values for any fields within that row are missing. A "listwise deletion by default" approach is concerning because missing values, especially if they are prevalent and non-random, can lead to biased model outputs.

Observable Plot gallery: Line with missing data

There are two different kinds of missing values to look out for. The more obvious kind of missing values are individual values within rows that are empty, null, or encoded as missing. These should be easy to find if you know how missing values are encoded. Perhaps they're obvious like empty fields or NULLs, but in some datasets, missing values might be represented by a specific (usually large) number like 9999 (or -1 for values that can otherwise only be positive).

Missing rows are more difficult to detect. If missing values inside of existing rows are *known* unknowns, these are the unknown or invisible unknowns. One way to look for them is by counting values by category, and comparing those counts to see if certain groups are very under- or over-represented. That can be an indication that you are missing important records for a specific group, and may indicate imbalanced survey data (also called class imbalance). With temporal data, check for values per time period, keeping an eye out for days, weeks, or months with few or no records.

Carefully exploring patterns of missingness helps data analysts to decide on an appropriate strategy to handle missing values. If missing values are infrequent and random, listwise deletion may be a suitable approach. If not, imputation — the process of substituting missing values with reasonable values using methods like hot deck, regression, or multiple imputation — might be a better option.

Duplicate values

Duplicate entries occur when the same record appears more than once. In small datasets, they can be spotted by simply looking through your data when sorted by a specific field. Excel, for example, has a "Duplicate Values" option to highlight duplicate rows over selected fields.

In larger datasets, analysts often use existing tools to check for duplicate records. For example, base R has a built-in duplicated() function, the pandas library in Python has a similar duplicated() method, and in SQL you can group by multiple fields the count occurrences; any with a count greater than one indicate duplication across those fields.

Incomplete time periods

One of the biggest issues in data analysis is dealing with incomplete time periods at the end of a dataset. For example, funnel performance metrics for the current month may appear much lower than previous months simply because the current month is incomplete. This is particularly challenging because recent data is often the most important, and analysts may not know which level of granularity, such as days, weeks, or months, that users will focus on. While it's generally best to exclude partial periods to prevent misinterpretation, doing so is often impractical in real-world analysis.

Inconsistent formatting

Inconsistent formatting in datasets, such as varying date formats or differing labels like “USA” versus “United States,” can create confusion and hinder accurate analysis. To ensure consistency and reliability, you should standardize these elements so that all data uses uniform units and formats.

Parsing errors

Parsing errors happen when data is incorrectly converted or interpreted during the processing stage. Typically, parsing errors happen when fields are read in as the incorrect type. For instance, a product code may be read in as a number, when it is actually categorical data.

Depending on your data and how familiar you are with it and its context, a data dictionary can be helpful. It tells you what the different columns contain and what the values in each column mean. This isn’t always straightforward, because some datasets encode special values using certain numbers. 9999 might mean a missing value, or all 99xx values might mean different things. Knowing what each column contains, and what the values in it mean, are important for data checking, analysis, and modeling.

The FIPS identifier is a five digit code combining the two digit state code (“06” for California) and three digit county code (“037” for Los Angeles). In the table below, the FIPS code is numeric, with the leading zero removed:

County “	FIPS Code #	Estimated Population #
Los Angeles	6037	9,757,179
San Diego	6073	3,298,799
Orange	6059	3,170,435
Riverside	6065	2,529,933
San Bernardino	6071	2,214,281
Santa Clara	6085	1,926,325
Alameda	6001	1,649,060
Sacramento	6067	1,611,231
Contra Costa	6013	1,172,607

A data analyst may need to recast the FIPS code from a number to the correct 5-character string

County “	FIPS Code #	Estimated Population #
Los Angeles	06037	9,757,179
San Diego	06073	3,298,799
Orange	06059	3,170,435
Riverside	06065	2,529,933
San Bernardino	06071	2,214,281
Santa Clara	06085	1,926,325
Alameda	06001	1,649,060
Sacramento	06067	1,611,231
Contra Costa	06013	1,172,607

Data wrangling

Data wrangling is the process of transforming raw data into a useful format for downstream visualization and analysis. For example, data wrangling may involve joining data tables, deriving new columns, or pivoting data from wide to long format.

Below, we describe several of the most common data wrangling steps that show up frequently in data analysis workflows.

Filtering records

Sometimes, you're interested in analyzing only a specific subset of your data. Filtering data returns records that match one or more criteria for more focused data exploration and analysis.

Filter criteria might be based on numerical values (e.g., for an analysis of home sales, including only houses sold for over $500k), text strings (e.g., include only homes where the region field matches "Southern California"), date limits (include only sales since January 1, 2022), boolean filters, and more. Often, more than one criteria are applied to create a more restrictive subset based on multiple variables.

Joining tables

Business data is often stored across multiple tables that are related to each other in a relational database. Partitioning data into different tables by topic (for example, with sales data in one table, and customer demographics in another) limits repetition for easier and more efficient data management and processing. This means that the business question you're trying to answer might require data that is stored across multiple tables.

In that case, you'll want to join the relevant tables based on matching keys. There are different types of joins including full, left, and inner joins. The type of join you choose is important because it determines which rows from each table are included in the final output. For example:

- **Full join:** A full join returns all rows from both tables, whether or not they have a match for the join condition.
- **Left join:** A left keeps all records from the left table, but only joins records from the right table if they have a match in the left table. If a record from the left table does not have a match in the right table, the column values from the right table will be null.
- **Inner join:** A more restrictive join type that only returns rows if there is a match in both tables.

ORDERS

order_id	order_date	order_total	customer_id
01	2022-01-06	31.47	AX4
04	2022-02-15	80.56	BF2
05	2022-03-18	52.60	YP0
09	2022-03-24	14.99	MP1

CUSTOMERS

id	State	Age
CK5	California	18-24 yr
AX4	Idaho	25-34 yr
BF2	Utah	18-24 yr
UH8	Oregon	35-44 yr
MP1	California	55-64 yr

An INNER JOIN will only return records if there is a match in both tables. An inner join on customer ID (customer_id in the ORDERS table, and id in the CUSTOMERS table) will return the table below.

INNER JOIN

order_id	order_date	order_total	customer_id	State	Age
01	2022-01-06	31.47	AX4	Idaho	25-34 yr
04	2022-02-15	80.56	BF2	Utah	18-24 yr
09	2022-03-24	14.99	MP1	California	55-64 yr

Pivoting data

There are different ways to organize tabular data. It might be in tidy format, where each variable occupies a single column, or, it could be in a wide format, where values for a single variable are spread across multiple columns.

Depending on downstream tools used, and analyses planned, you may need to reshape your data to get it into a compatible format. For example, Observable Plot (our open source JavaScript library for data visualization) expects tidy data for most chart types.

You can pivot data to rotate rows to columns, or vice-versa.

Wide format → Long format

Year	Basic Tier	Pro Tier	Enterprise Tier
2021	1,281	357	27
2022	2,430	464	30
2023	4,895	490	45

Year	Tier	Customers
2021	Basic	1,281
2022	Basic	2,430
2023	Basic	4,895
2021	Pro	357
2022	Pro	464
2023	Pro	490
2021	Enterprise	27
2022	Enterprise	30
2023	Enterprise	45

Deriving new values

Deriving new values from existing data is an important data wrangling technique because it streamlines downstream analyses and improves interpretation.

There are a few different reasons why you'd want to derive new values based on existing data. For example, you may need to combine information about orders with a product price list in order to calculate total revenue across orders. Standardization, normalization, and unit conversions are common operations analysts use to improve uniformity and comparability across variables and data sources.

It's also common to derive values to represent data at a different resolution than what's in the raw data. For example, you may want to pull the year component from hourly records, or decide to aggregate continuous age data into ordinal bins (e.g., 0-18 years, 19-24 years, etc.).

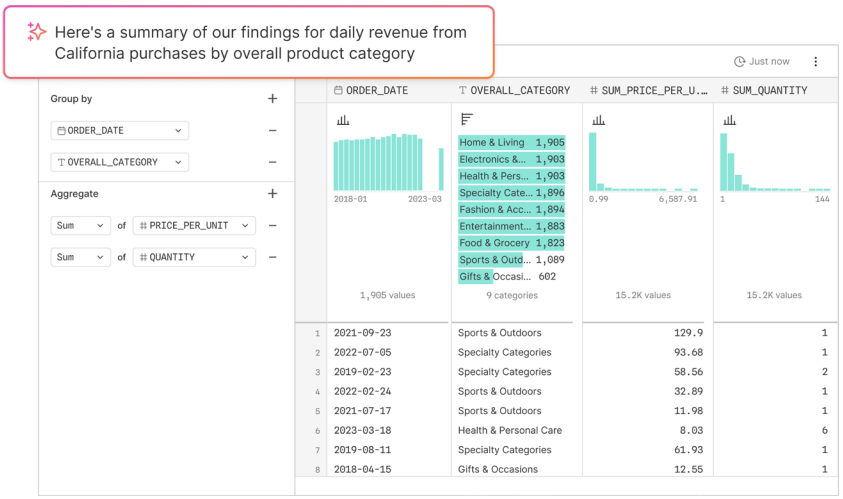
Converting variable types

An important part of data wrangling is to check data types, and when they're incorrect, recasting variables to the appropriate type before using them in analysis and modeling.

One common example is when numbers are used to represent categories, like product or ZIP codes. These are typically parsed as numbers by default in most software. For example, the ZIP code 93514 might be stored as the numeric value 93,514, which is incorrect because the code is only a categorical identifier and not a quantitative measure or count. In that case, the ZIP code variable should be recast as a string or factor (not a number) so it is treated as a categorical variable in downstream analyses.

Using AI for data wrangling

AI helps analysts speed through time-consuming data wrangling by translating natural language prompts into a series of data manipulations. A tool might do this using UI options that filter, sort, select, or derive values in sequence to complete the task. Or, the prompt might generate a SQL query that returns data in a more usable format. The latter is often called text-to-SQL or natural language to SQL ("NL2SQL" for short).



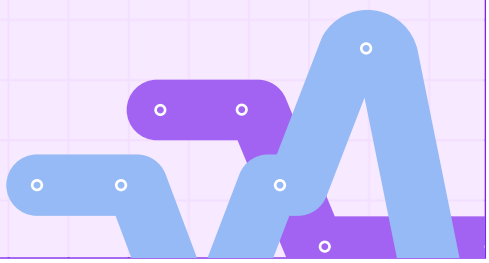
Examples of AI prompts to help with initial data profiling:

- Show me all electronics orders shipped to California over the past year
- Find the number of monthly clothing orders by customer income level
- Find the top 10 food products shipped to Florida, by total revenue

Exploratory data analysis

Exploratory data analysis (often referred to as EDA) is used to dig deeper into the data, looking at the basic structure and patterns in the dataset.

The first step of any exploratory data analysis is data profiling. Data profiling is used to gain a high-level familiarity with the structure, content, and quality of data in a database or data warehouse. In other words, it helps give a quick early answer to the question: "What's in the data, and what can I do with it?" It is an important step in data exploration because it helps data analysts understand the data and uncover data quality issues, so they can do any necessary cleaning and decide on appropriate analysis methods.



The first thing to look for are values that stick out when just browsing through the data. Are there any values that seem wrong, is there text in numeric columns, or anything that just looks off? Look for values that are unusually large, or negative values where there shouldn't be any.

Next, sort by the different columns and look at the maximum and minimum values. Do they all seem reasonable? It is often difficult to set strict upper and lower limits for numerical values, so doing this by hand can be helpful. Some datasets also use specific numbers as codes for missing values, which makes this more challenging as well.

Extreme and unusual values can be data errors, but they can also be a first hint of outliers to look for when you move from data exploration to the actual data analysis.

A large part of data profiling is finding summary statistics. These often include measures of central tendency (like mean, median, and mode), data spread (variance or standard deviation), and extrema (minimum and maximum values) for individual variables. Table-level information like table dimensions and variable types can also be explored as part of data profiling.

Using AI for data profiling

As one-off calculations, the summary statistics listed above are trivial, but they can be time-consuming to find manually when working with a large database and many fields. AI allows analysts to automate data profiling to get a surface-level look at their data in a fraction of the time.

Here are some examples of AI prompts to help with initial data profiling:

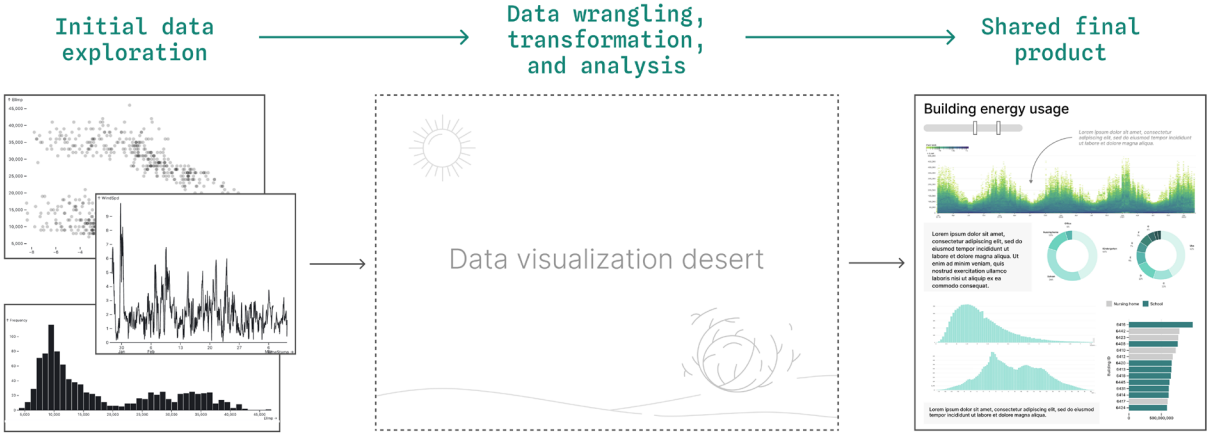
- Summarize this data including table dimensions, column names, data types, and summary statistics for each variable
- What number and proportion of values are missing for each column?

Visual data analysis

Visual data analysis, which brings data visualization and analysis workflows together, helps analysts uncover patterns and anomalies that inspire new questions and exploration paths.

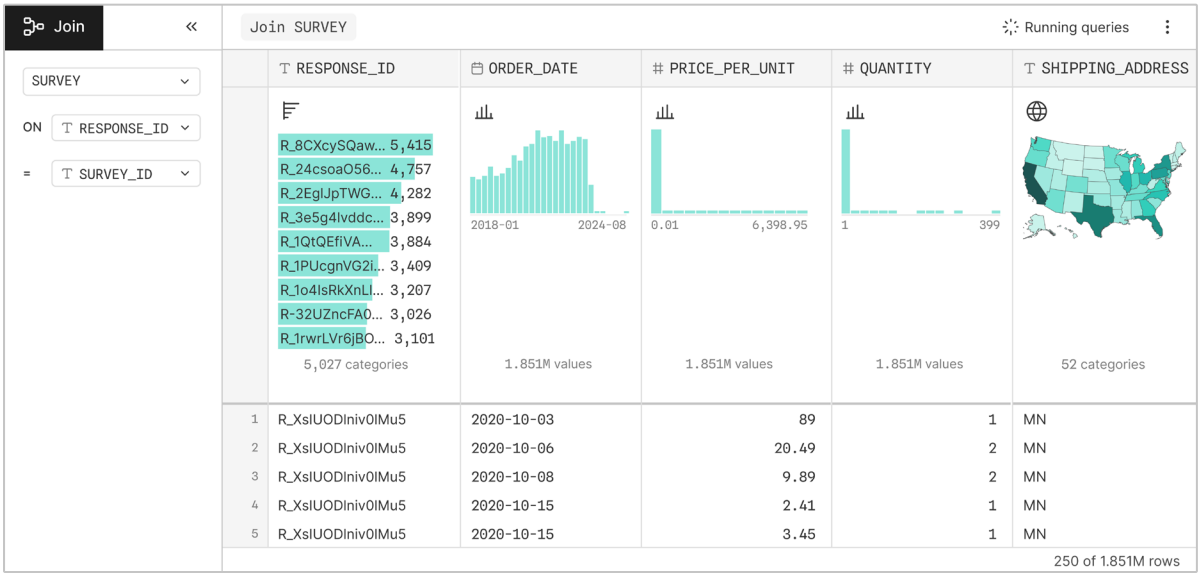
Manually building visualizations to explore the output of each data processing step can be prohibitively time consuming. As a result, visualization falls by the wayside when analysts are heads-down in data cleaning, wrangling, modeling, troubleshooting, and iterating.

We call this the data visualization desert: where data is being processed, but no visual outputs are being created. Without an eye on visualizations throughout the process, an analyst may miss an important pattern or insight, or find they spent hours working only to arrive at an unusable or uninteresting chart.



One of the benefits of visual data analysis is that it allows analysts to more easily discover trends and outliers through the use of charts, tables, and graphics. This is why tools that provide visual summaries of the data at each step in your analysis by default can be particularly helpful.

For example, in Observable Canvases, all data processing (whether performed in SQL, returned by AI, or done using UI options for common operations like joins and aggregation) are accompanied by a summary table containing small charts in each column header that provide a quick view of variable distributions. These concise data visualizations make it easy to explore the shape and content of the data.



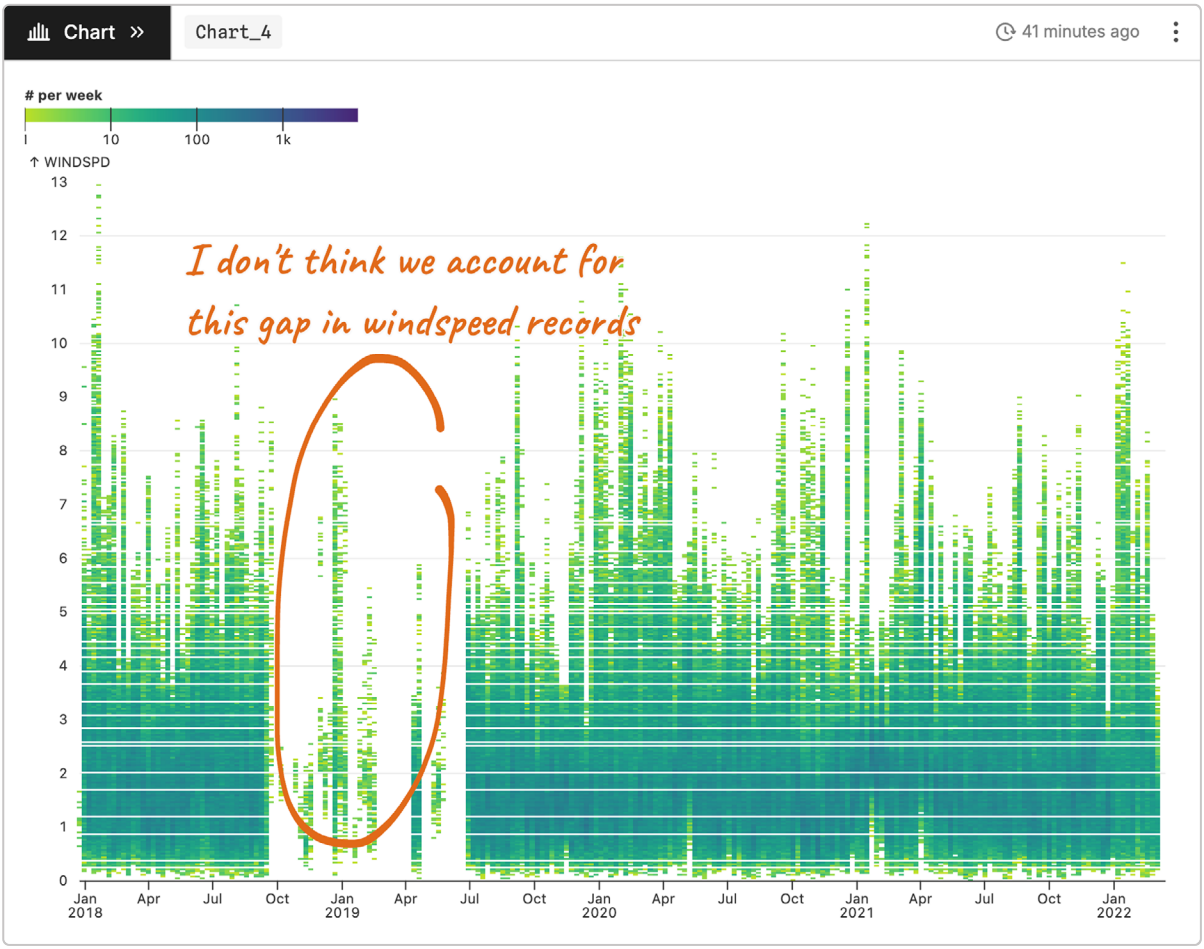
From there, it just takes a few clicks to make larger, multivariate charts that let you dig deeper into the patterns you've discovered, which could yield unexpected and valuable insights for your business.

Benefits of visual data analysis

Visual data analysis can help improve accessibility and interpretability for cross-functional collaborators. Visualizations allows many people — regardless of technical knowledge or skills — to review and interpret analyses and give high quality feedback.

When collaborators visually follow how data is transformed throughout analysis, they can more confidently interpret what they're viewing at any point. For example, they can see how data has been filtered and aggregated upstream, to know exactly what data is included in a larger downstream chart.

As a data analyst, you can also proactively identify mistakes and anomalies when you use visual data analysis. For instance, sometimes unit conversions can get fumbled. Whole groups inadvertently disappear with the smallest typo (was it `== “blueWhale”`, or `== “bluewhale”`?). And, since your team can visually track changes to the data at each step, it’s easier to determine exactly where in the analysis things went wrong. These mistakes won’t show up as error messages in the console, since code doesn’t know (or care) if what you’re asking it to do with the data is actually correct.



Data visualization is a critical part of data exploration because it helps analysts to uncover patterns, anomalies, and relationships between variables that can highlight new questions and inform subsequent analysis.

Using AI for visual data analysis

Creating visualizations for exploratory purposes can be tiresome and frustrating, especially when it involves big data and tedious preprocessing to get it in a compatible shape with your chart type. AI can automatically generate charts, helping data analysts get off-and-running with quick data displays that avoid a blank slate.

Here are example prompts to create an exploratory data visualization:

- Draft a line chart of monthly revenue over time, with a different line color for each customer income level
- Create a histogram of product prices for all items in the Electronics & Media category
- Make a stacked bar chart of top products by quantity purchased, with fill color based on customer education level

Data modeling and data visualization

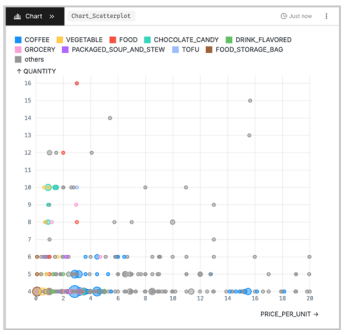
Once data analysts have decided on the path they want to explore further, they model the data to generate deeper insights or forward-looking predictions. This could include predictive modeling to forecast future outcomes based on historical values, on historical data, descriptive modeling to identify patterns or clusters in the data, or causal modeling which attempts to describe why something happened.

This stage also may include data visualization. As in exploratory data analysis, visualizations help analysts uncover trends and valuable insights that allow stakeholders to make more informed decisions.

Data visualizations, such as charts and dashboards, help communicate insights to stakeholders in a number of ways. Data visualizations are critical for communicating findings as they can:

- Increase interpretability by making the data analysis pipeline, including complex datasets, easier to understand.
- Accelerate the discovery of insights by visually demonstrating outcomes and issues.
- Improve decision-making by creating shared meaning between team members and stakeholders.

Effective data visualization turns complex information into clear, actionable insights. Well-designed charts highlight outliers, reveal trends, and allow easy comparison between groups. Choosing the right chart type matters: use bar charts for category comparisons, histograms for distributions, and scatterplots to identify relationships.



When creating data visualizations, it's also important to keep accessibility and clarity in mind. Charts should have clear titles, labeled axes, readable legends, and utilize color schemes that are accessible by everyone (including people with color blindness). Adding interactive features, like linked brushing, let users explore data in more depth.

Strong visualizations ensure your analysis is memorable and easy to understand.

Interpretation and communication

One of the most important parts of the data analysis process is the interpretation and communication of outcomes or insights to stakeholders and decision makers. While data teams will often package their findings in a report or dashboard that is shared with stakeholders, there are many benefits to bringing stakeholders into the data analysis process more thoroughly through collaborative analytics.

Data can be communicated or presented through dashboards, reports, one-off charts, and more. When creating your visualizations, it's important to consider how the stakeholder will consume the visualization, and what requirements they might have as a result. For instance, if your end-user will primarily see a chart on a mobile device, you'll want to make sure the chart renders clearly in smaller spaces and loads quickly.

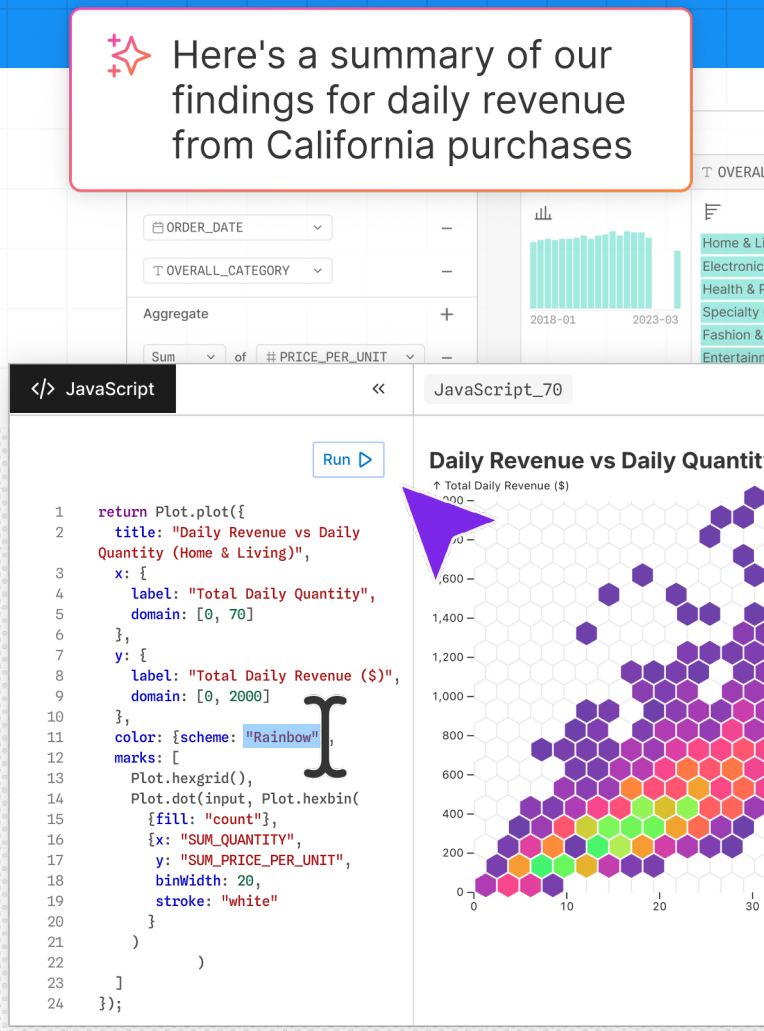
Conclusion

Data analysis allows you to turn raw data into useful insights.. Whether you're working in Excel, Python, or interactive platforms like Observable, the goal is to ask the right questions, analyze the evidence, and communicate findings clearly. It's not always a linear process, as new inputs and data can inform your analysis. Great data analysis is a team effort, which is why tools that support cross-functional collaboration can help you deliver stronger results.

Data analysis, without the paralysis

With Observable, you can explore data, collaborate with stakeholders, build with AI, and quickly create charts you can only dream of in other tools.

Contact sales →



A checklist for navigating and exploring data

- [ ] Where does the data come from?
- [ ] When was it exported (if not directly accessed from a database)?
- [ ] Has it been filtered?
- [ ] Has it been cleaned?
- [ ] Have computed fields been added?
- [ ] Do you have the data dictionary with column definitions and special values?
- [ ] Have you checked for extreme values and outliers?
- [ ] Have you checked for duplicate values?
- [ ] Have you checked for missing values within rows?
- [ ] Have you checked for missing records?
- [ ] Do you know what incomplete time periods there are at the beginning and end of your dataset?