

How to open a Parquet file on Mac

A from-scratch guide — no prior knowledge of Python or the terminal required

Goal. After completing this guide you will open a .parquet file, check its columns, and see the first records. You don't need to know Python or SQL beforehand.

First, four concepts

Parquet: a data file format — similar to CSV or Excel, but better suited to large datasets.

Terminal: the window built into macOS where you run commands. You don't need to install it.

Python: the program needed to run the code in this guide. We will install it.

Library: an add-on for Python. In this guide we use PyArrow, Polars, and DuckDB.

Important. PyArrow, Polars, and DuckDB are three alternative ways of opening the same file. You don't need to use all of them. For large data and trend analysis, DuckDB is recommended.

What you need

Computer: a Mac running macOS with an account that allows installing applications.

Internet: needed while installing Python and the libraries.

File: an unzipped file with the .parquet extension.

Time: about 10–15 minutes for the initial setup.

STEP 1 | Download and install Python

1. Open a web browser.
2. Go to the official website: <https://www.python.org/downloads/macos/>
3. In the latest stable release section, click macOS installer. A file ending in .pkg will download.
4. Open the downloaded .pkg file and go through the installer: Continue, Agree, Install.
5. The system may ask for your Mac account password or Touch ID. This confirms permission to install the program — it is not the password for the ZIP file.

6. When finished, open Finder → Applications → Python 3.14 and double-click the file Install Certificates.command.

7. When the opened window shows update complete, close it.

Do not install Terminal. Terminal is already part of macOS. You are only installing Python and its add-ons.

Apple Silicon and Intel. The current universal2 installer from python.org works on supported Macs with both Apple silicon and Intel processors.

STEP 2 | Open Terminal and check Python

1. Press Command (⌘) and spacebar at the same time to open Spotlight.

2. Type Terminal.

3. Launch the Terminal app.

4. In the Terminal window, on the line usually ending with % or \$, type exactly:

```
python3 --version
```

Watch the formatting. It should be python3 with no space, then one space, then two dashes before the word version: python3 --version.

If the command doesn't work, close Terminal, reopen it, and type python3 --version again.

The correct result looks similar to:

```
Python 3.14.7
```

STEP 3 | Install the libraries

Stay in Terminal. On the line ending with % or \$, paste:

```
python3 -m pip install pyarrow pandas duckdb
```

Press Enter and wait. The installation may display many lines of text.

Correct completion. At the end, the message Successfully installed will appear, followed by a normal Terminal line again.

If a question appears. Type y and press Enter. Yellow WARNING messages are usually just warnings, not installation failures.

STEP 4 | Prepare the Parquet file

1. Download the ZIP archive provided by the organizer. The archive contains two files: `datasprint_sample_data.zip`, approximately 18 GB in size, and `datasprint_full_data.zip`, approximately 87 GB in size. If you don't have enough space on your laptop to download `datasprint_full_data.zip`, download and use the smaller file instead. Using the smaller file will not affect the evaluation or quality of the solution — it was added for participants' convenience.
2. Open Finder → Downloads and double-click the ZIP archive. If asked for a password, enter the password provided by the organizer.
3. Find the extracted file ending with the `.parquet` extension.
4. Move this file directly into the Downloads folder.
5. For the purposes of this guide, rename it to:

```
data.parquet
```

Check the extension. The full name must be `data.parquet`, not `data.parquet.zip` or `data.parquet.txt`.

STEP 5 | Enter Python

In Terminal, type:

```
python3
```

After a moment you'll see the prompt:

```
>>>
```

This is a key distinction. On a normal Terminal line ending with % or \$ you type commands, e.g. `python3` or installing libraries. At `>>>` you type Python code, e.g. `import pandas as pd`.

STEP 6 | Set the file location

At the >>> prompt, paste the following two lines one after another. Press Enter after each:

```
from pathlib import Path
file_path = str(Path.home() / "Downloads" / "data.parquet")
```

Check what location Python has set:

```
print(file_path)
```

A path similar to this should appear:

```
/Users/YourName/Downloads/data.parquet
```

If FileNotFoundError appears. Python couldn't find the file. Check that data.parquet is really located directly in the Downloads folder and that the name matches exactly.

STEP 7 | Choose how to open the file

The options below are independent. You can use just one or try all three.

Option A — PyArrow: checking the technical structure

At the >>> prompt, type in order:

```
import os
import pyarrow as pa
import pyarrow.parquet as pq
parquet_path = os.path.expanduser(file_path)
pf = pq.ParquetFile(parquet_path)
print("Shape:", (pf.metadata.num_rows, pf.metadata.num_columns))
batch = next(pf.iter_batches(batch_size=10))
table = pa.Table.from_batches([batch])
display(table.to_pandas())
```

The first command shows the number of rows and columns in the table. The last one shows up to the first 10 records. PyArrow lets you read Parquet in batches, so you don't need to load the whole file into RAM.

Option B — Polars: data as a table

At the >>> prompt, type in order:

```
import os
import polars as pl
parquet_path = os.path.expanduser(file_path)
```

```
lf = pl.scan_parquet(parquet_path)
print(lf.limit(20).collect())
print(lf.collect_schema())
```

limit(10) shows the first 10 records. collect_schema shows the column names and data types.

Large file. Avoid pandas, since it may load the entire dataset into RAM. For very large data, use Polars lazy first, or DuckDB.

Option C — DuckDB: recommended for large datasets

At the >>> prompt, type in order:

```
import duckdb
duckdb.sql(f"SELECT * FROM '{file_path}' LIMIT 10").show()
```

LIMIT 10 means: show only 10 records. DuckDB doesn't need to display or pass an entire million rows to Pandas.

To count all records without displaying them, type:

```
duckdb.sql(f"SELECT COUNT(*) AS record_count FROM '{file_path}'").show()
```

To see the column names and their types, type:

```
duckdb.sql(f"DESCRIBE SELECT * FROM '{file_path}'").show()
```

Optional — DuckDB without running Python

This option corresponds to the CLI section of the Visa guide. It requires Homebrew. If you don't have it, you can simply stick with the Python DuckDB option described above.

First exit Python with the exit() command. Then check in Terminal:

```
brew --version
```

If a Homebrew version number appears, type:

```
brew install duckdb
```

After installation you can preview 10 records directly from Terminal:

```
duckdb -c "SELECT * FROM '$HOME/Downloads/data.parquet' LIMIT 10;"
```

If you see command not found: brew. Homebrew is not installed. This option is optional — go back to the Python-based DuckDB option.

STEP 8 | Finish

If you still see the >>> prompt, exit Python and return to the normal Terminal line by typing:

```
exit()
```

STEP 9 | How to analyze a larger dataset

With a million records, don't try to view every row. Instead, ask the dataset specific questions: how many records are there, what are the most common categories, how do values change over time, or where do unusual spikes occur.

Most common values in a selected column

```
duckdb.sql(f"""
SELECT category, COUNT(*) AS count
FROM '{file_path}'
GROUP BY category
ORDER BY count DESC
LIMIT 20
""").show()
```

Replace the word category with the actual column name from your file.

Monthly trend

```
trend = duckdb.sql(f"""
SELECT
    date_trunc('month', CAST(date AS DATE)) AS month,
    COUNT(*) AS record_count,
    SUM(amount) AS total_amount
FROM '{file_path}'
GROUP BY 1
ORDER BY 1
""").df()
print(trend)
```

The names date and amount are examples. Before using them, replace them with names that exist in your dataset.

Rule for large data. Don't use `print(df.to_string())` or `SELECT *` without `LIMIT`. First check the columns, then filter and group the data, and only move a small result to Pandas, Excel, or a chart.

STEP 10 | Common problems

“command not found: import”: Python code was pasted at the normal Terminal line. Type `python3`, wait for `>>>`, and only then paste the import.

“No module named...”: A library is missing. Type `exit()`, then in Terminal run `python3 -m pip install pyarrow pandas duckdb` again.

“FileNotFoundError”: The file name or its location doesn't match `file_path`. Check the Downloads folder and the name `data.parquet`.

Terminal shows a lot of error text: Read the last line. Usually the problem is the wrong file location, a missing library, or code typed in the wrong place.

Terminal stopped responding: Don't try to print the entire dataset. Close the window, restart Terminal, and use DuckDB with `LIMIT`.

Quick checklist

- 1: Python responds to `python3 --version`.
- 2: The libraries were installed without a red error.
- 3: The file `data.parquet` is located directly in the Downloads folder.
- 4: The `import...` code is typed only at the `>>>` prompt.
- 5: For a large dataset, `LIMIT` or aggregation is used instead of displaying everything.

Technical sources

Official Python documentation for macOS: docs.python.org/3/using/mac.html

Official installers for macOS: python.org/downloads/macos/