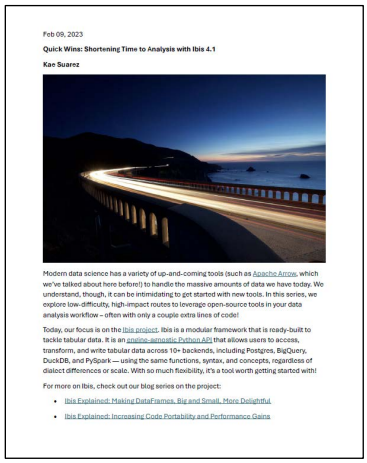
VOLTRON DATA

carahsoft



Feb 09, 2023

Quick Wins: Shortening Time to Analysis with Ibis 4.1

Kae Suarez

Modern data science has a variety of up-and-coming tools (such as Apache Arrow, which we've talked about here before!) to handle the massive amounts of data we have today. We understand, though, it can be intimidating to get started with new tools. In this series, we explore low-difficulty, high-impact routes to leverage open-source tools in your data analysis workflow – often with only a couple extra lines of code!

Today, our focus is on the Ibis project. Ibis is a modular framework that is ready-built to tackle tabular data. It is an engine-agnostic Python API that allows users to access, transform, and write tabular data across 10+ backends, including Postgres, BigQuery, DuckDB, and PySpark — using the same functions, syntax, and concepts, regardless of dialect differences or scale. With so much flexibility, it's a tool worth getting started with!

For more on Ibis, check out our blog series on the project:

- Ibis Explained: Making DataFrames, Big and Small, More Delightful
- Ibis Explained: Increasing Code Portability and Performance Gains

Quick Wins: Shortening Time to Analysis with Ibis 4.1

Thank you for downloading this Voltron Data resource. Carahsoft is the distributor for Voltron Data's AI and ML solutions available via NASA SEWP V, ITES-SW2, NASPO ValuePoint, and many more contract vehicles.

To learn how to take the next step toward acquiring Voltron Data's solutions, please check out the following resources and information:

For Voltron Data overview:
carah.io/Voltron-Data

For upcoming events:
carah.io/Voltron-Events

For additional resources:
carah.io/Voltron-Resources

For additional Artificial Intelligence:
carah.io/ai-solutions

To set up a meeting:
VoltronData@carahsoft.com

To purchase, check out the contract vehicles available for procurement:
carah.io/procurement

For more information, contact Carahsoft or our reseller partners:
VoltronData@carahsoft.com | 888-606-2770

Feb 09, 2023

Quick Wins: Shortening Time to Analysis with Ibis 4.1

Kae Suarez

Modern data science has a variety of up-and-coming tools (such as Apache Arrow, which we've talked about here before!) to handle the massive amounts of data we have today. We understand, though, it can be intimidating to get started with new tools. In this series, we explore low-difficulty, high-impact routes to leverage open-source tools in your data analysis workflow – often with only a couple extra lines of code!

Today, our focus is on the Ibis project. Ibis is a modular framework that is ready-built to tackle tabular data. It is an engine-agnostic Python API that allows users to access, transform, and write tabular data across 10+ backends, including Postgres, BigQuery, DuckDB, and PySpark — using the same functions, syntax, and concepts, regardless of dialect differences or scale. With so much flexibility, it's a tool worth getting started with!

For more on Ibis, check out our blog series on the project:

- Ibis Explained: Making DataFrames, Big and Small, More Delightful
- Ibis Explained: Increasing Code Portability and Performance Gains

Previously, we read files with Arrow, both in R and Python. Today, we'll do that with Ibis in Python, but with a special twist: file intake just got way better with the just-released Ibis 4.1!

2022 File I/O

Let's look into the not-so-distant past: about a month ago. There are various ways to handle data, and they all diverge in the interface — but perhaps in no way quite so annoying as file ingestion. Every single interface, ultimately, takes in a file, but even simple CSVs require some different decision-making or syntax to get things into your database. Let's see how this plays out, across DataFusion, DuckDB, and Polars.

DataFusion:

```python
ctx = datafusion.SessionContext()

ctx.register_csv("mil", "ratings.csv")

#alt, but yields Dataframe instead of database connection:

ctx.read_csv("ratings.csv")
```

DuckDB:

```python
con = duckdb.connect()

con.execute("CREATE VIEW tester AS SELECT * FROM read_csv_auto('ratings.csv')")
```

Polars:

```python
polars.read_csv("ratings.csv")
```

All powerful engines, and all easy ways to stumble – but what if we can use these with less stumbling?

Well, Ibis already enables developers in every other step of analysis...why not CSV ingestion?

2023 File I/O

This brings us straight to Ibis 4.1. One notable addition in this version is the pair of functions read_csv and read_parquet. These functions do what they say in the name – they help the user read in CSV or Parquet to their selected back-end. The supported back-ends are DataFusion, DuckDB, and Polars. Due to the robustness of the underlying back-ends, these reading functions can be called to immediately yield a proper reading for any well-formed file. For more complex needs, like skipping rows at the start of a CSV, access to the underlying arguments is still provided, since there are differences between syntax and functionality in back-ends that aren't easily addressed (...yet).

Let's walk through this new process with the same backends and see how Ibis can make lives easier:

DataFusion:

```python
ibis.set_backend("datafusion")

ibis.read_csv("ratings.csv")
```

DuckDB:

```python
ibis.set_backend("duckdb")

ibis.read_csv("ratings.csv")
```

Polars:

```python
ibis.set_backend("polars")

ibis.read_csv("ratings.csv")
```

This takes away those tripping hazards, at least for simple CSVs that we just want to read in and start working with.

Conclusion

Data isn't always in a nice database – in fact, CSV and Parquet are common enough that being able to read them quickly and easily is vital to quickly explore new and existing data alike. This feature is incredibly helpful for tackling the massive datasets we face today. While the pandas library also has the ability to accomplish tasks like this with one line of code, it still needs to work in memory, which guarantees that there will be datasets it cannot process. This is where Ibis is at its best, growing easier to use with each release and still offering the power and scalability of its underlying backends.

Of course, this isn't all that was added in the Ibis 4.1 release. There's plenty more to see! (If you want some motivation: do you like dplyr selectors?) Refer to Ibis Project's GitHub to check out the latest version, and the release notes for Ibis 4.1 to get up to date on everything in the new release. If you're already working with the project and want to go further, learn how a Voltron Data Enterprise Support subscription can help accelerate your success with Ibis.

Photo by: jms