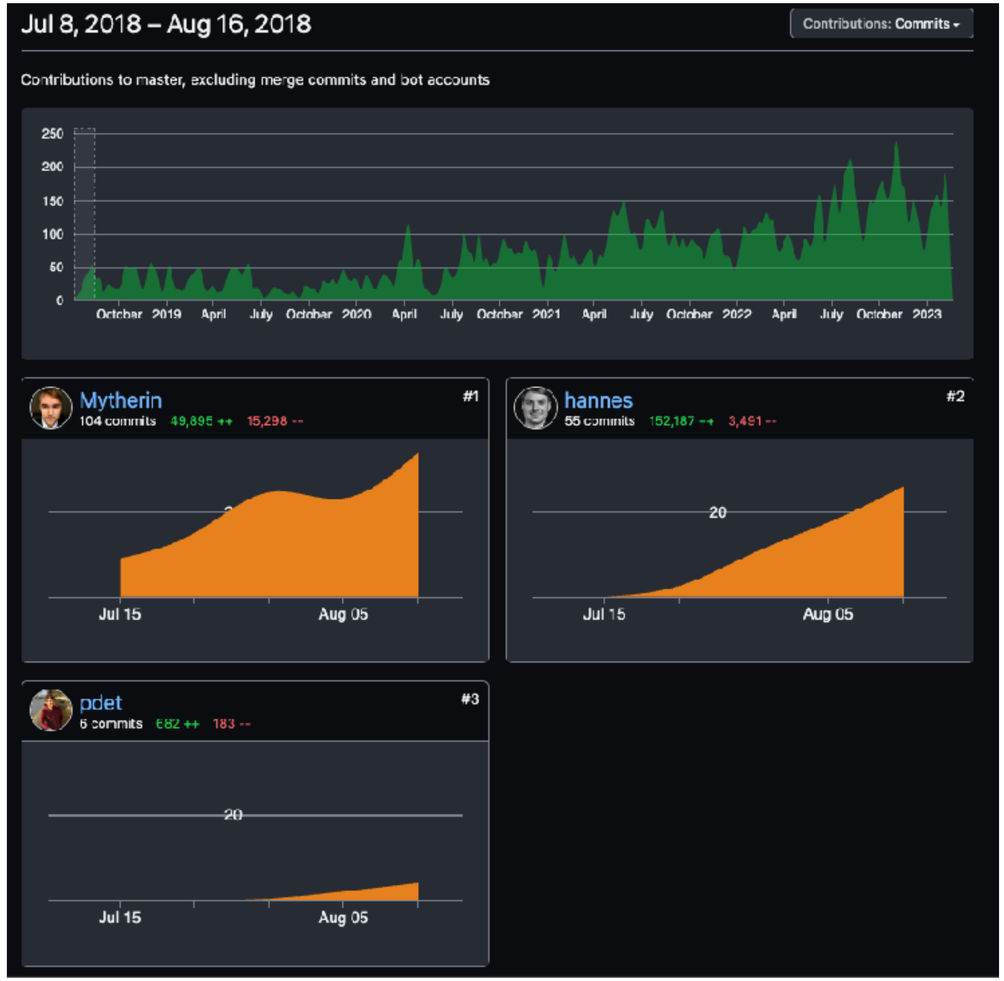
Data Analysis With DuckDB

Pedro Holanda
pedro@duckdblabs.com
pedroholanda.com

DuckDB Labs

Who am I?

- Ph.D. in Database Architectures @ CWI-Amsterdam
- COO of DuckDB Labs.
- Early collaborator of DuckDB;
  - Indexes/Zone-maps
  - Arrow Integration
  - **Big chunk of the Python API**
  - Tons More…

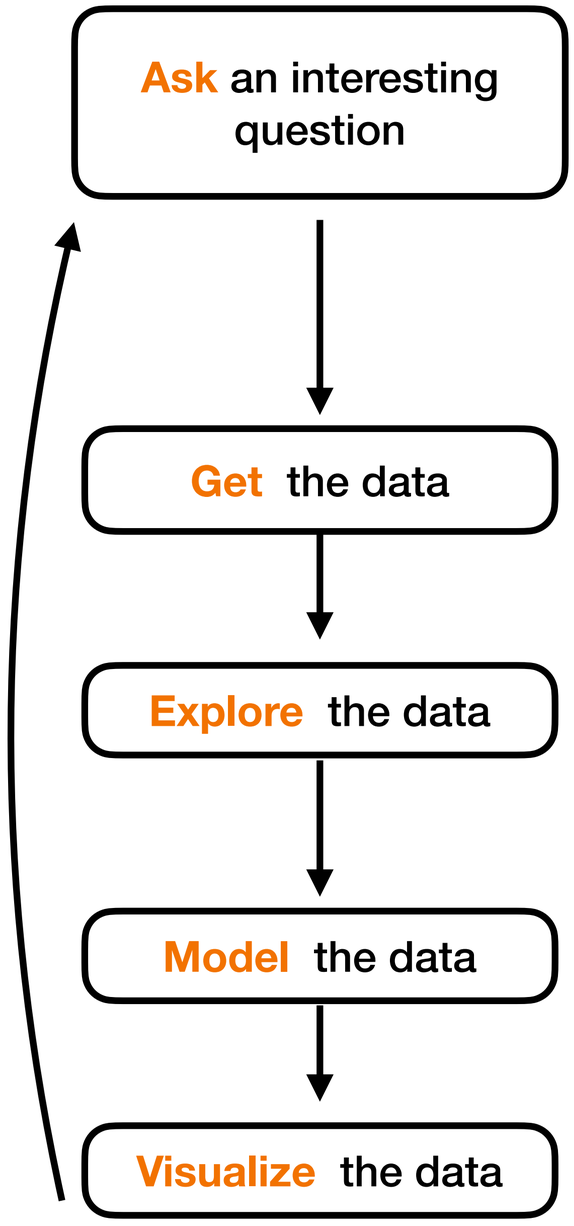


Motivation

The Problem

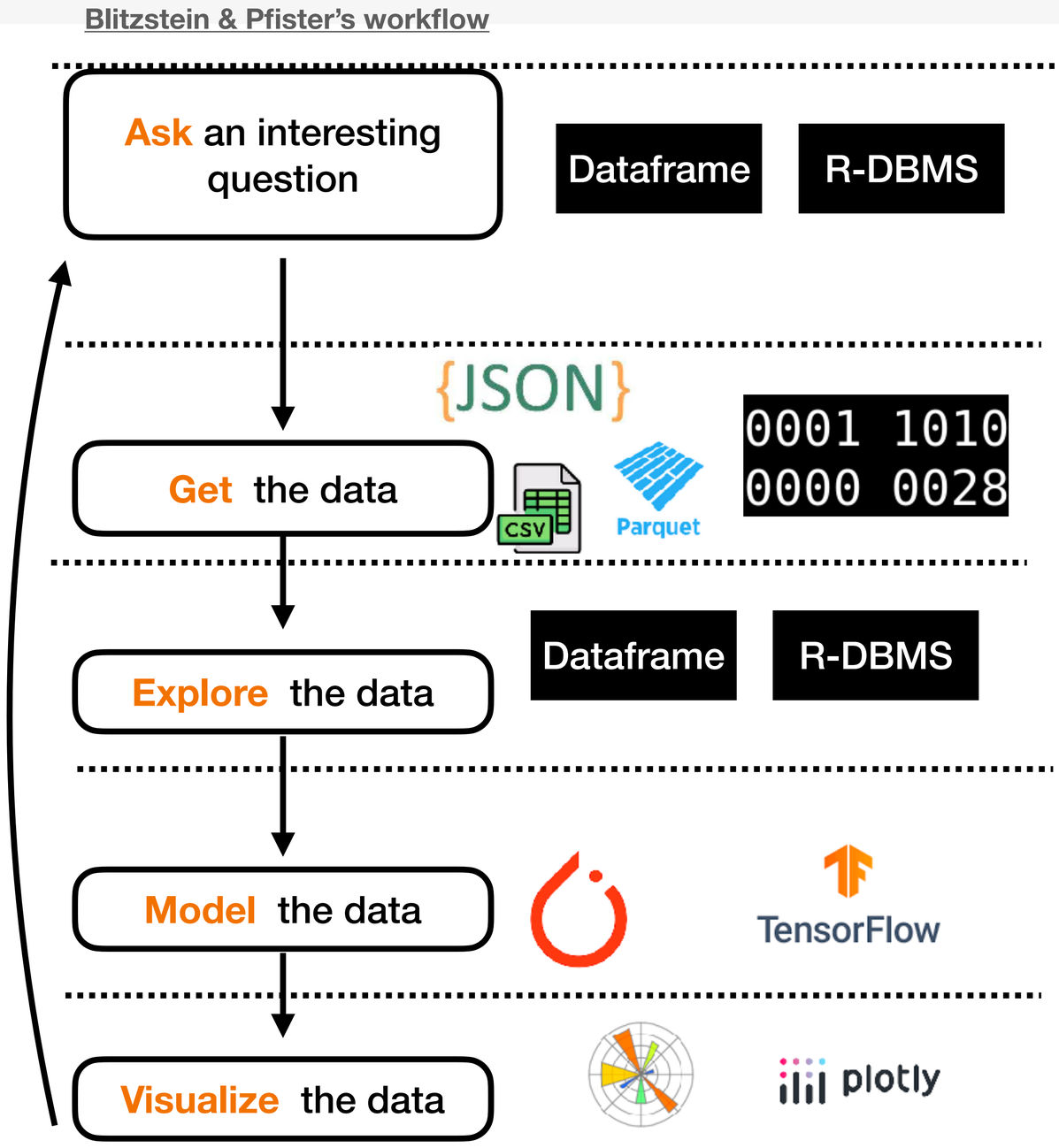
- **Data Science**
  - **Exploratory**
  - **Interactive**
  - **Trial and Error**
  - **Hypotesis Driven**

Blitzstein & Pfister's workflow



The Problem

- **Data Science**
  - **Exploratory**
  - **Interactive**
  - **Trial and Error**
  - **Hypotesis Driven**



Dataframes vs RDBMS

- Important part of a Data Science Workflow
- Must:
  - **Scan** different file formats.
    - CSV, JSON, Parquet, Binaries (Other Systems).
  - **Integrate** with Ecosystem Tools.
    - Plot Libraries, ML Libraries.
  - **Be efficient** analytical execution engines
    - Beyond Memory Execution
    - Complex Query Optimization
  - Support **SQL** and **Relational API**.

DataFrames

- Dataframe Libraries
- Integrate with Python Ecosystem
  - Numpy/PyArrow
- Easy to use.
- Relational API.
- Fast Data Transfer
- Integrated Scanners with schema detection to multiple file formats.
- Limited Analytical Query Support:
  - SQL
  - Query Optimization
  - Beyond Memory Execution
  - Lack of storage (Deal with cumbersome file paths)
  - Limited Parallelism

What is DuckDB?

DuckDB: The SQLite for Analytics

- **Simple installation**

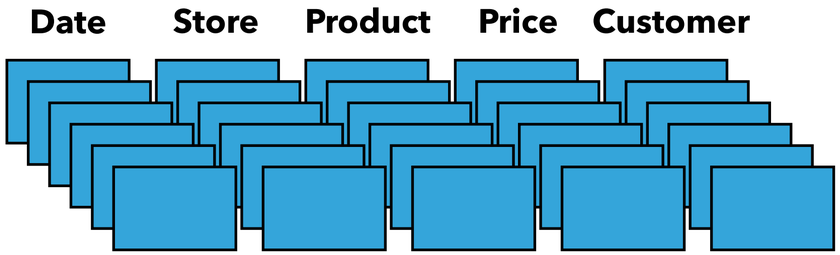
```
$ pip install duckdb
```

- **Embedded:** no server management
- **Fast analytical processing**
- **Fast transfer between R/Python and RDBMS**
- **Rich SQL Dialect**
- **Single File Format**

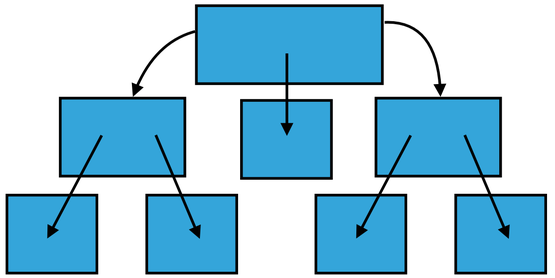
- **DuckDB** is currently in **pre-release (V0.7)**
  - Check duckdb.org for more details.

Main Characteristics

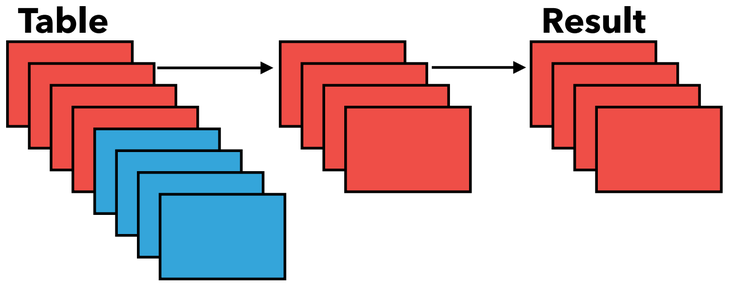
Column-Store



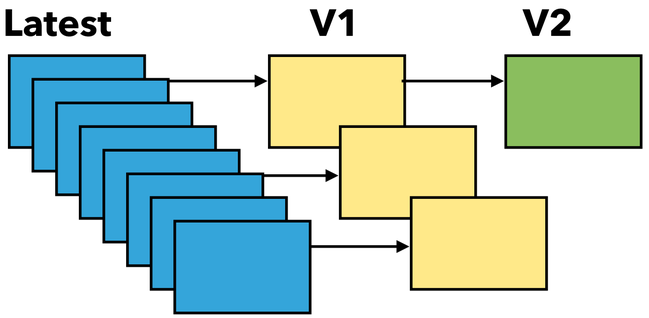
ART Index



Vectorized Processing



MVCC



Single-File Storage

database.db

Parser

Main Characteristics

- **Compression**
- **End-to-end Query Optimization**
- **Automatic Parallelism**
- **Beyond Memory Execution**

DuckDB In the Python Land

APIs

- **Python DB API 2.0 Compliant**

```
import duckdb
con = duckdb.connect("duck.db")
con.execute("SELECT j+1 FROM integers WHERE i=2")
```

- **Relational API**

```
import duckdb
con = duckdb.connect("duck.db")
# Table operator returns a table scan
rel = con.table("integers")
# We can inspect intermediates
rel.show()
# We can chain multiple operators
rel.filter("i=2").project("j+1").show()
```

Integrations

- **Integrations**
  - **NumPy**
  - **PyArrow**
  - **Pandas**
  - **Polars**
  - **Pytorch**
  - **Tensorflow***
  - **SQL Alchemy**
  - **IBIS (Default Backend)**

In-Out Integrations Examples

PyArrow

```python
import duckdb
import pyarrow as pa

my_arrow_table = pa.Table.from_pydict({'i':[1,2,3,4],
                                       'j':["one", "two", "three", "four"]})

# query the Apache Arrow Table "my_arrow_table" and return as an Arrow Table
results = duckdb.sql("SELECT * FROM my_arrow_table").arrow()
```

Pandas

```python
import duckdb
import pandas

# Create a Pandas dataframe
my_df = pandas.DataFrame.from_dict({'a': [42]})

# create the table "my_table" from the DataFrame "my_df"
# Note: duckdb.sql connects to the default in-memory database connection
duckdb.sql("CREATE TABLE my_table AS SELECT * FROM my_df").df()
```

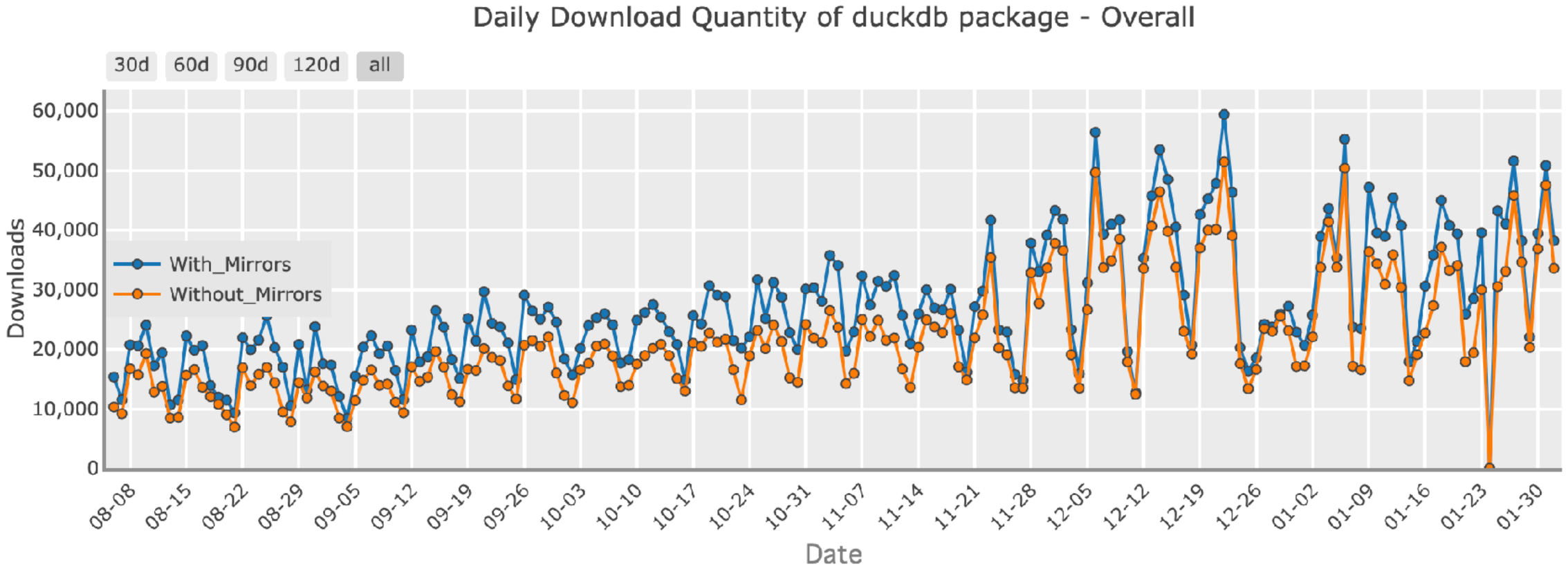
Python UDFs*

```python
import duckdb
import pandas as pd
def plus_one(x):
    table = pa.lib.Table.from_arrays([x], names=['c0'])
    import pandas as pd
    df = pd.DataFrame(x.to_pandas())
    df['c0'] = df['c0'] + 1
    return pa.lib.Table.from_pandas(df)

con = duckdb.connect()
con.create_function('plus_one', plus_one, [BIGINT], BIGINT, type='arrow')
assert [(6,)] == con.sql('select plus_one(5)').fetchall()
```

Usage

Downloads last day: 33,594
Downloads last week: 251,770
Downloads last month: 898,816



Hands-On

Integrations

- **Explore the NYC Taxi Dataset**
  - **Preview Columns/Types/Data**
  - **Run and Plot Queries like**
    - **Does the AVG tip value increase the more passengers we have?**
      - **What about only Long Trips?**
      - **Rainy Days?**
  - **Perform Data Cleaning**
  - **Linear Regression with SQL**
  - **ML with Python UDFs***

https://github.com/pdet/data_analysis_course