DATA+AI SUMMIT

Forward-looking Statement

This presentation has been prepared for informational purposes only. The information set forth herein does not purport to be complete or contain all relevant information. Statements contained herein are made as of the date of this presentation unless stated otherwise.

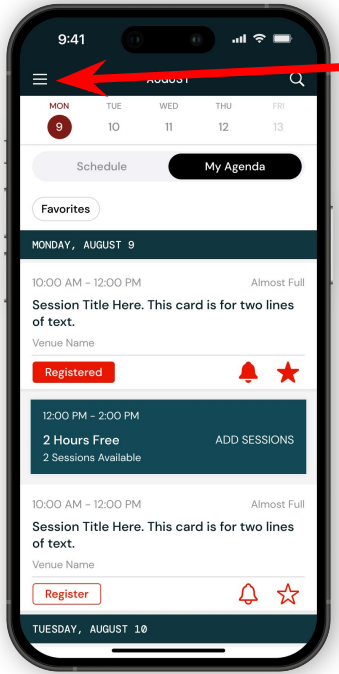
This presentation and the accompanying oral commentary may contain forward-looking statements. In some cases, forward-looking statements can be identified by terms such as “may”, “will”, “should”, “expects”, “plans”, “anticipates”, “could”, “intends”, “projects”, “believes”, “estimates”, “predicts”, or “continue”, or the negative of these words or other similar terms or expressions that concern Databricks’ expectations, strategy, plans, or intentions.

Forward-looking statements are based on information available at the time those statements are made and are inherently subject to risks and uncertainties that could cause actual results to differ materially from those expressed in or suggested by the forward-looking statements.

Forward-looking statements should not be read as a guarantee of future performance or outcomes. Except as required by law, Databricks does not undertake any obligation to publicly update or revise any forward-looking statement, whether as a result of new information, future developments or otherwise.

Complete Your Surveys

Your feedback has a direct impact on Data + AI Summit content



- You will receive a survey for each session attended
- Open the Databricks Events app and select “My Surveys” from the menu
- Surveys can also be submitted in the Attendee Portal

Next-Gen Data Science

How Posit and Databricks Are Transforming Analytics at Scale

James Blair
Date

Agenda

- **Posit and Databricks**
  Better Together
- **Next-Gen Data Science**
  Introduction to Positron
- **GenAI Powered Applications**
  Natural language dashboarding
- **App Deployment**
  Share securely
- **Conclusion**
  Only the beginning

Posit and Databricks

Built to last

Proud to be a Certified B Corporation®.

Our mission is **free and open-source software** for data science, scientific research & technical communication for the public good.

Our corporate charter requires that company decisions **balance the interests** of community, customers, employees, & shareholders.

To ensure we can be a trustworthy partner and support our mission long-term, Posit **will remain independent and privately held**.

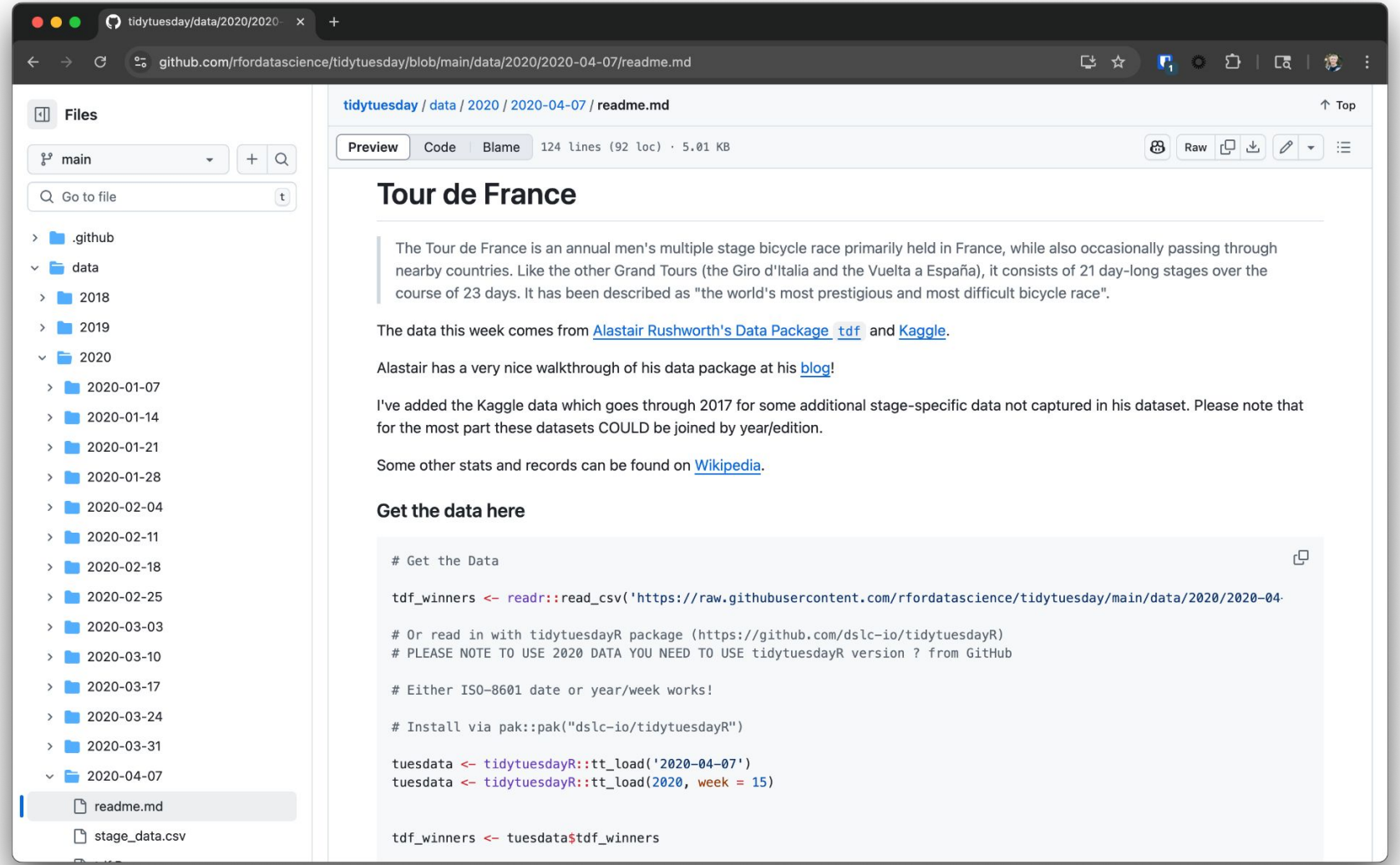
databricks DATA+AI SUMMIT

Winner

2025 Databricks Partner Awards

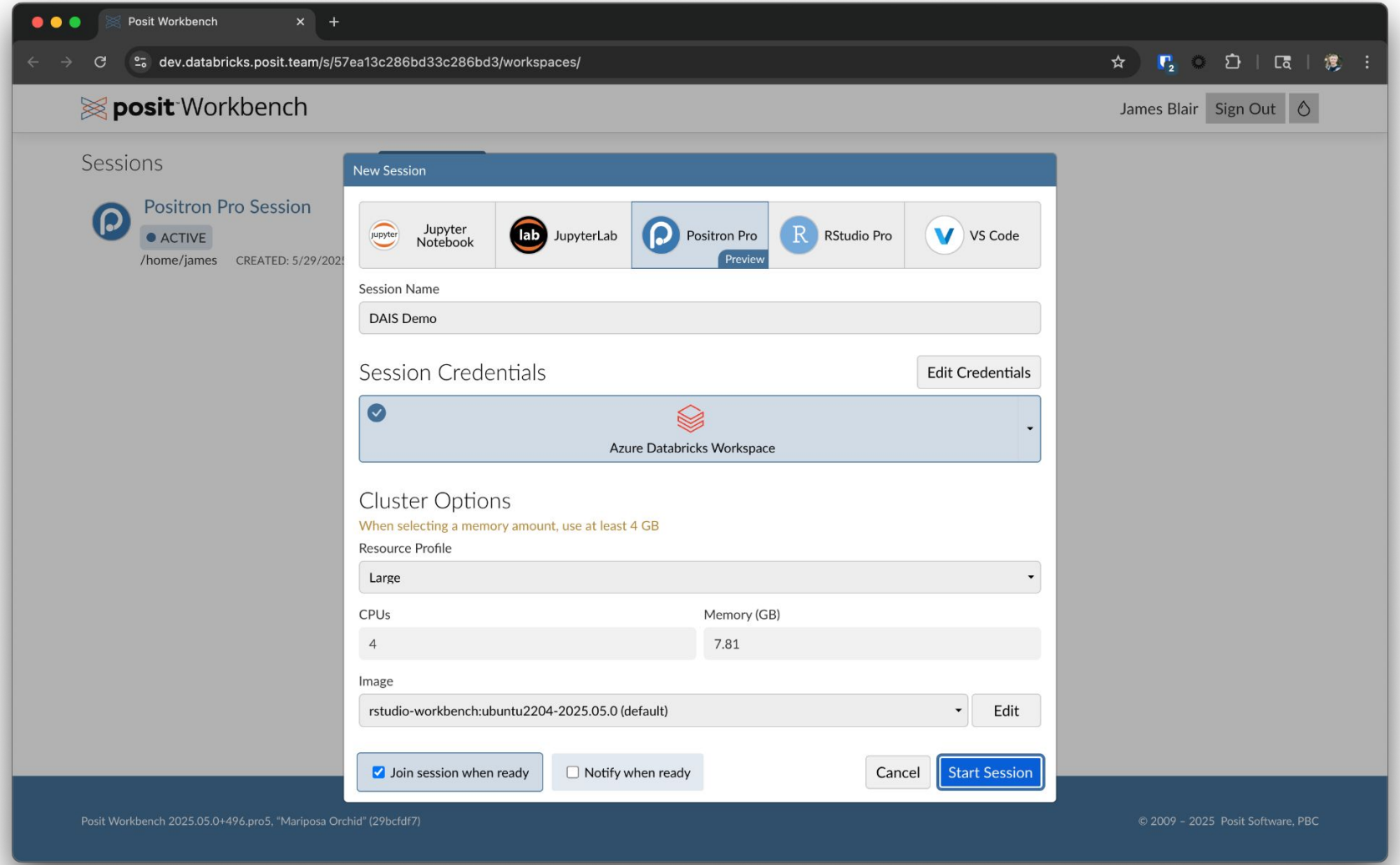
Developer Tools Partner of the Year

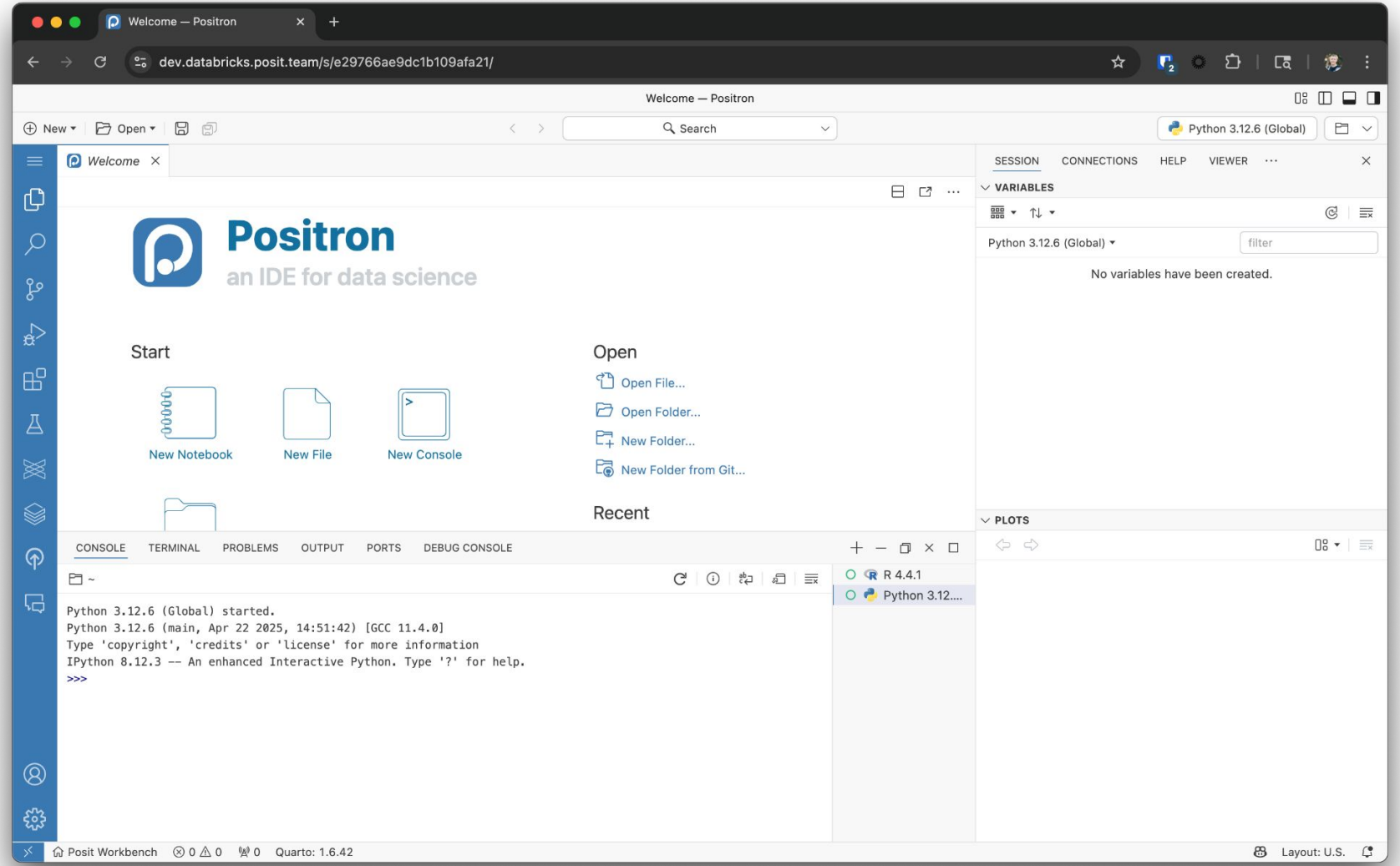

AGGIES


tidytuesday/data/2020/2020-
github.com/rfordatascience/tidytuesday/blob/main/data/2020/2020-04-07/readme.md
Files
main
Go to file
t
.github
data
2018
2019
2020
2020-01-07
2020-01-14
2020-01-21
2020-01-28
2020-02-04
2020-02-11
2020-02-18
2020-02-25
2020-03-03
2020-03-10
2020-03-17
2020-03-24
2020-03-31
2020-04-07
readme.md
stage_data.csv
tidytuesday / data / 2020 / 2020-04-07 / readme.md
Top
Preview
Code
Blame
124 lines (92 loc) · 5.01 KB
Raw
Tour de France
The Tour de France is an annual men's multiple stage bicycle race primarily held in France, while also occasionally passing through nearby countries. Like the other Grand Tours (the Giro d'Italia and the Vuelta a España), it consists of 21 day-long stages over the course of 23 days. It has been described as "the world's most prestigious and most difficult bicycle race".
The data this week comes from Alastair Rushworth's Data Package tdf and Kaggle.
Alastair has a very nice walkthrough of his data package at his blog!
I've added the Kaggle data which goes through 2017 for some additional stage-specific data not captured in his dataset. Please note that for the most part these datasets COULD be joined by year/edition.
Some other stats and records can be found on Wikipedia.
Get the data here
# Get the Data
tdf_winners <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/main/data/2020/2020-04
# Or read in with tidytuesdayR package (https://github.com/dslc-io/tidytuesdayR)
# PLEASE NOTE TO USE 2020 DATA YOU NEED TO USE tidytuesdayR version ? from GitHub
# Either ISO-8601 date or year/week works!
# Install via pak::pak("dslc-io/tidytuesdayR")
tuesdata <- tidytuesdayR::tt_load('2020-04-07')
tuesdata <- tidytuesdayR::tt_load(2020, week = 15)
tdf_winners <- tuesdata$tdf_winners

Positron

A Next-Gen Data Science IDE


Posit Workbench
dev.databricks.posit.team/s/57ea13c286bd33c286bd3/workspaces/
posit Workbench
James Blair
Sign Out
Sessions
Positron Pro Session
ACTIVE
/home/james
CREATED: 5/29/202
New Session
Jupyter Notebook
JupyterLab
Positron Pro
Preview
RStudio Pro
VS Code
Session Name
DAIS Demo
Session Credentials
Edit Credentials
Azure Databricks Workspace
Cluster Options
When selecting a memory amount, use at least 4 GB
Resource Profile
Large
CPUs
4
Memory (GB)
7.81
Image
rstudio-workbench:ubuntu2204-2025.05.0 (default)
Edit
Join session when ready
Notify when ready
Cancel
Start Session
Posit Workbench 2025.05.0+496.pro5, "Mariposa Orchid" (29bcfdf7)
© 2009 – 2025 Posit Software, PBC


Welcome — Positron
dev.databricks.posit.team/s/e29766ae9dc1b109afa21/
Welcome — Positron
New
Open
Search
Python 3.12.6 (Global)
Welcome
Positron
an IDE for data science
Start
New Notebook
New File
New Console
Open
Open File...
Open Folder...
New Folder...
New Folder from Git...
Recent
SESSION
CONNECTIONS
HELP
VIEWER
VARIABLES
Python 3.12.6 (Global)
filter
No variables have been created.
PLOTS
CONSOLE
TERMINAL
PROBLEMS
OUTPUT
PORTS
DEBUG CONSOLE
Python 3.12.6 (Global) started.
Python 3.12.6 (main, Apr 22 2025, 14:51:42) [GCC 11.4.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.12.3 -- An enhanced Interactive Python. Type '?' for help.
>>>
R 4.4.1
Python 3.12....
Posit Workbench
Quarto: 1.6.42
Layout: U.S.

eda.qmd — eda — Positron

dev.databricks.posit.team/s/e29766ae9dc1bd980d399/?folder=/home/james/data-and-ai-summit-2025/eda

eda.qmd — eda — Positron

New | Open | Search | Python 3.11.12 (Uv: .venv) | eda

CHAT

Let me check if the plot was created:

✓ Using "Get active plot"...

The histogram has been created successfully. It shows the distribution of elapsed times for the Tour de France stages. From the plot, we can observe that:

1. The distribution is heavily right-skewed
2. Most elapsed times are clustered in the lower range (close to 0)
3. There are some outliers with much longer elapsed times (extending out to around 500000)

The current scale might make it a bit hard to see the details in the main cluster of times. Would you like me to modify the plot to better visualize the distribution, perhaps by:

1. Using a log scale
2. Focusing on the main cluster of times
3. Converting the elapsed time to a more interpretable unit (like hours or minutes)

Just let me know if you'd like any of these adjustmen

analyze the relationship between elapsed time and stage distance using a scatter plot

investigate outliers in elapsed time - what stages had unusually long times?

convert elapsed time to hours and create a more readable histogram

eda.qmd Current file

eda.qmd

eda.qmd > (code cell)

```
41  ```{python}
43  ```
44
    ▷Run Cell | Run Above
45  ```{python}
46  import matplotlib.pyplot as plt
47
48  plt.figure(figsize=(10, 6))
49  plt.hist(stages.elapsed.execute(), bins=50)
50  plt.xlabel('Elapsed Time')
51  plt.ylabel('Frequency')
52  plt.title('Distribution of Stage Elapsed Times')
53  plt.show()
54  ```
55
56
```

Keep Undo | 1 of 1

CONSOLE TERMINAL PROBLEMS 2 OUTPUT PORTS

Python 3.11.12 (Uv: .venv) ~/data-and-ai-summit-2025/eda

```
      4 elapsed_times = stages.select('elapsed').execute()
>>> import matplotlib.pyplot as plt
...
... # Get the elapsed time data
... elapsed_times = stages.elapsed.execute()
...
... # Create the histogram
... plt.figure(figsize=(10, 6))
... plt.hist(elapsed_times, bins=30, edgecolor='black')
... plt.title('Distribution of Stage Elapsed Times')
... plt.xlabel('Elapsed Time')
... plt.ylabel('Frequency')
... plt.grid(True, alpha=0.3)
>>>
```

SESSION CONNECTIONS HELP VIEWER

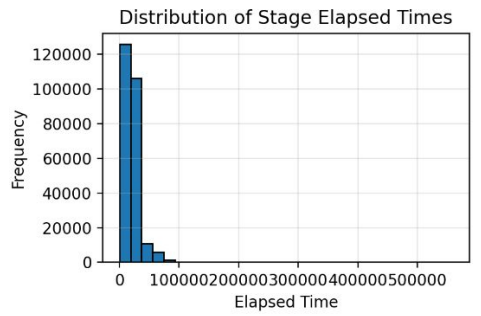
VARIABLES

Python 3.11.12 (Uv: .venv) filter

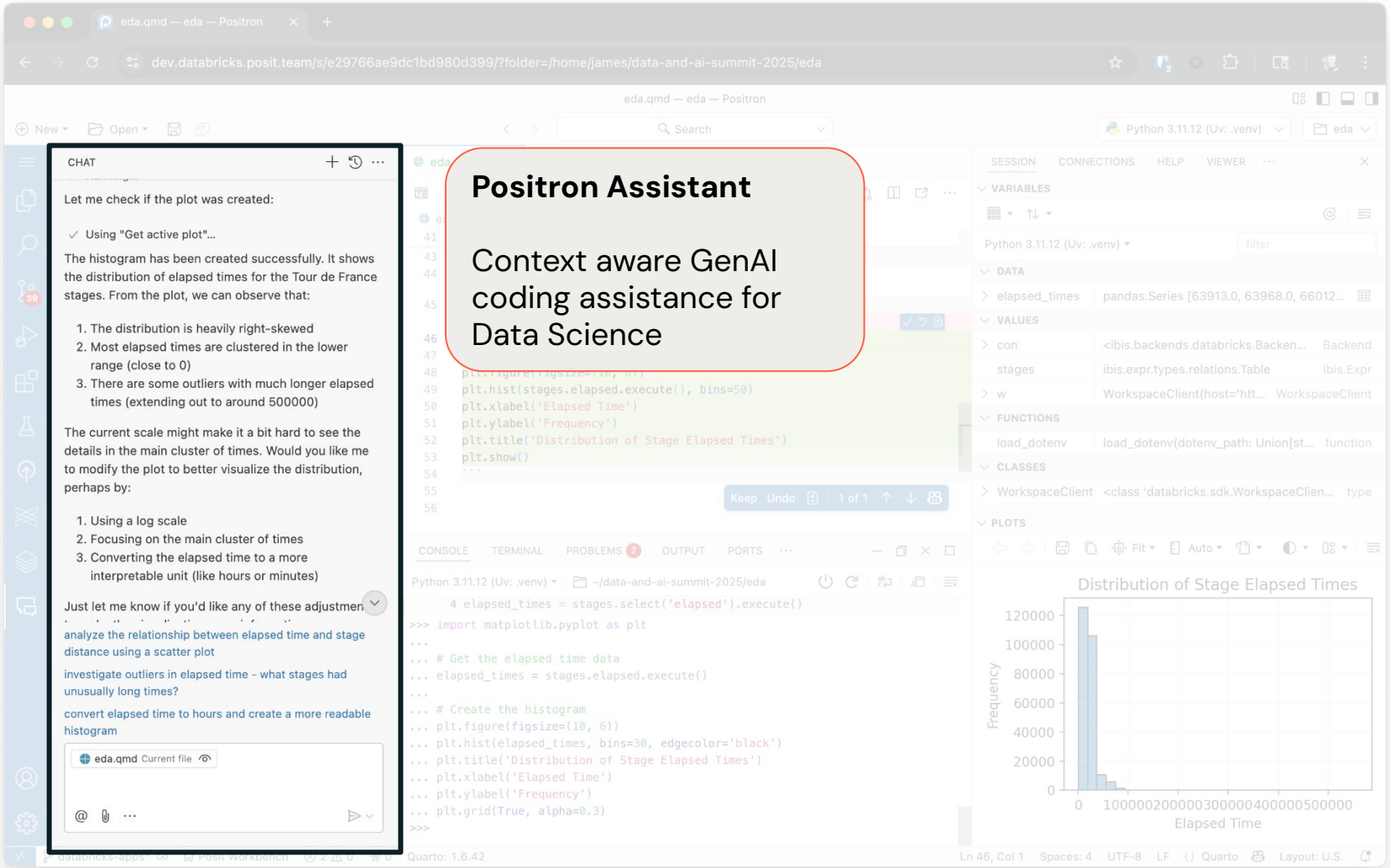
DATA		
elapsed_times	pandas.Series [63913.0, 63968.0, 66012...	
VALUES		
con	<ibis.backends.databricks.Backen...	Backend
stages	ibis.expr.types.relations.Table	ibis.Expr
w	WorkspaceClient(host='htt...	WorkspaceClient
FUNCTIONS		
load_dotenv	load_dotenv(dotenv_path: Union[st...	function
CLASSES		
WorkspaceClient	<class 'databricks.sdk.WorkspaceClien...	type

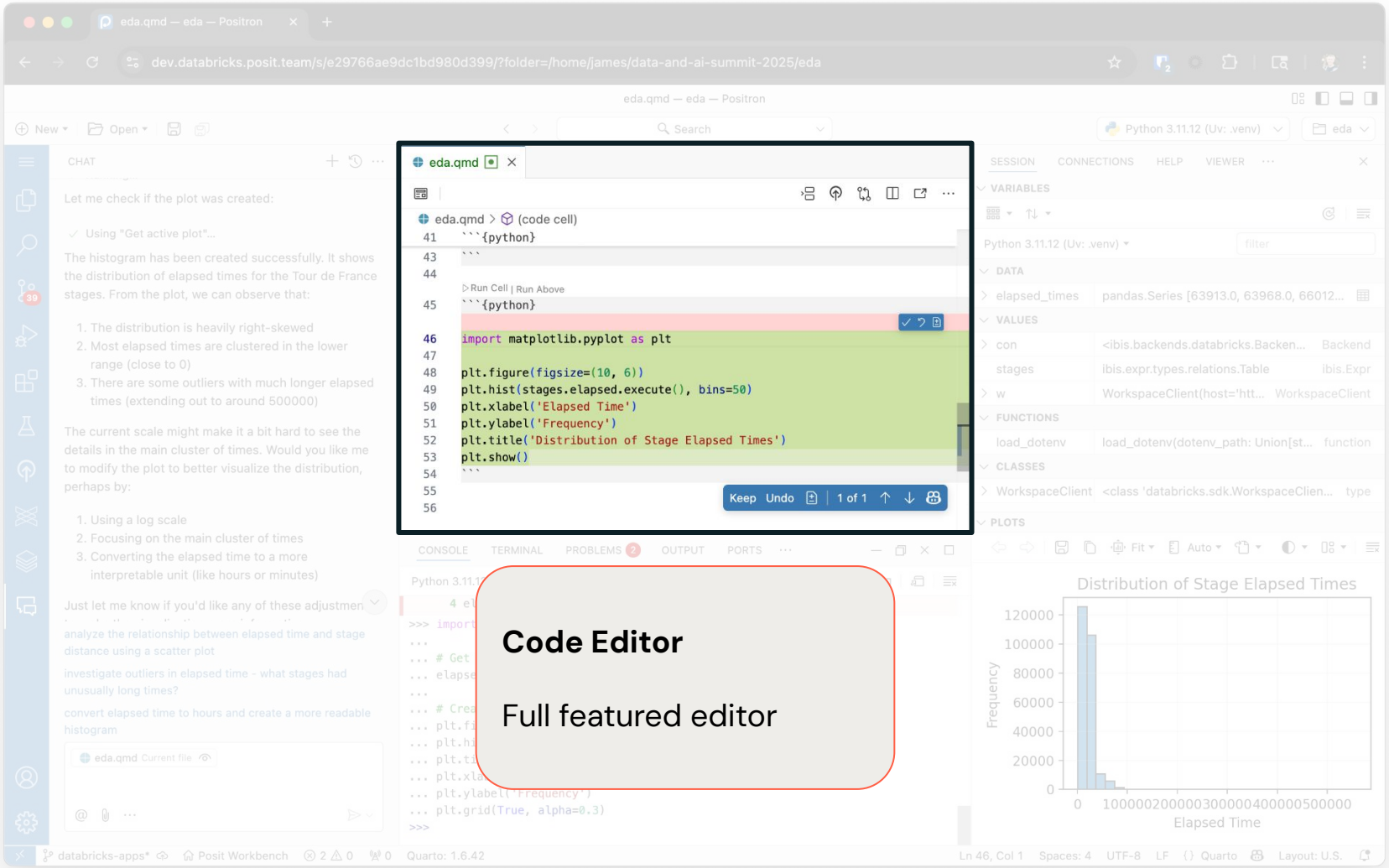
PLOTS

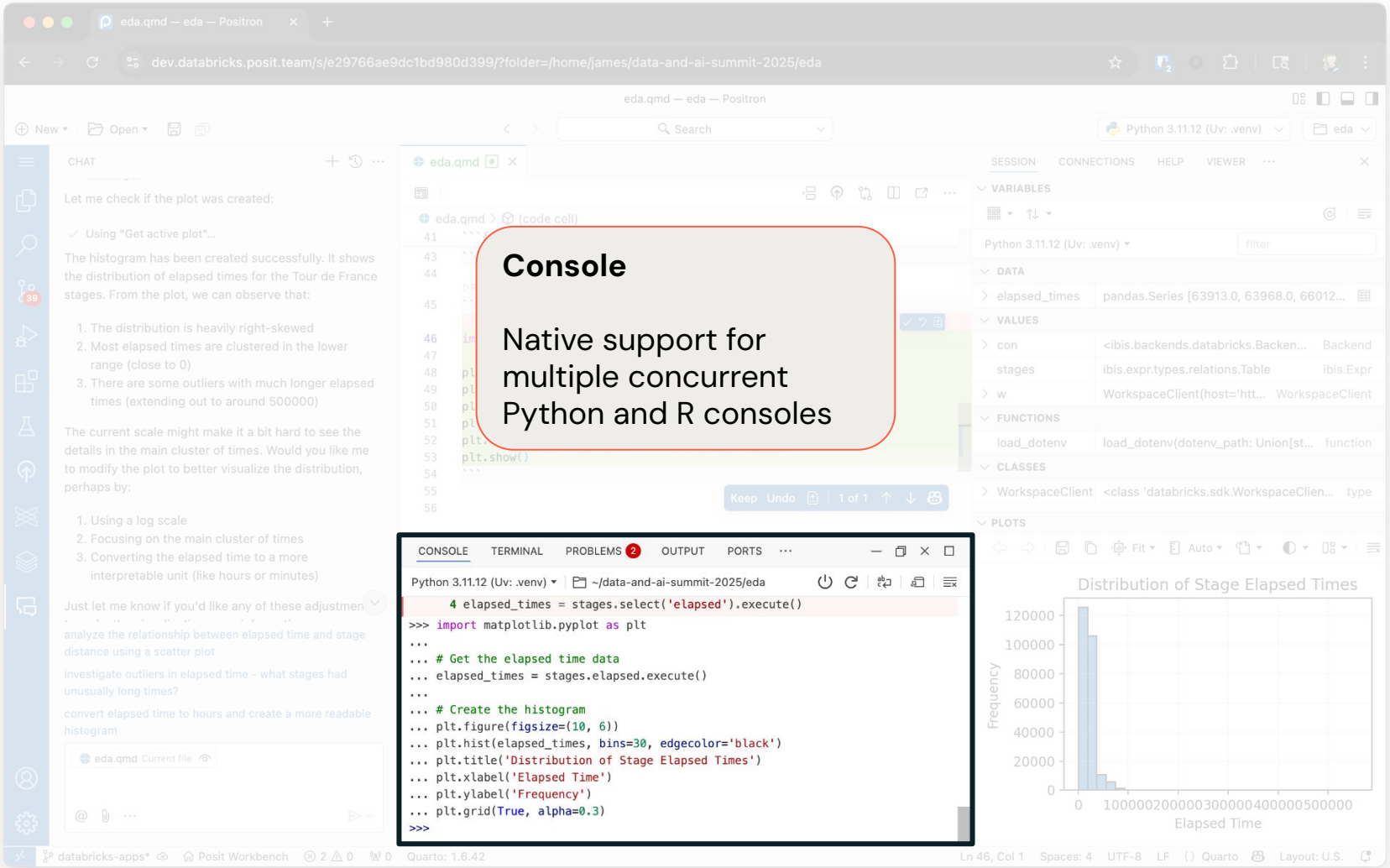
Fit | Auto



databricks-apps* | Posit Workbench | ⊗ 2 ⚠ 0 | 0 | Quarto: 1.6.42 | Ln 46, Col 1 | Spaces: 4 | UTF-8 | LF | Quarto | Layout: U.S.

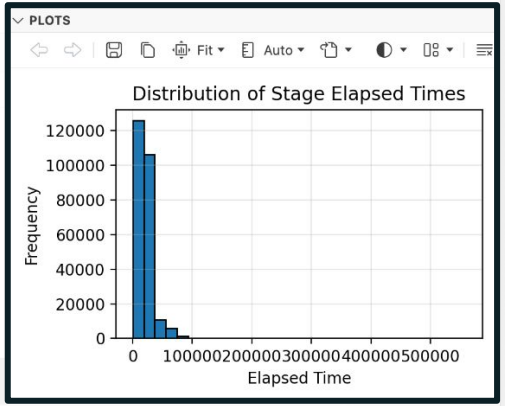

eda.qmd — eda — Positron
dev.databricks.posit.team/s/e29766ae9dc1bd980d399/?folder=/home/james/data-and-ai-summit-2025/eda
CHAT
Let me check if the plot was created:
Using "Get active plot"...
The histogram has been created successfully. It shows the distribution of elapsed times for the Tour de France stages. From the plot, we can observe that:
1. The distribution is heavily right-skewed
2. Most elapsed times are clustered in the lower range (close to 0)
3. There are some outliers with much longer elapsed times (extending out to around 500000)
The current scale might make it a bit hard to see the details in the main cluster of times. Would you like me to modify the plot to better visualize the distribution, perhaps by:
1. Using a log scale
2. Focusing on the main cluster of times
3. Converting the elapsed time to a more interpretable unit (like hours or minutes)
Just let me know if you'd like any of these adjustmen
analyze the relationship between elapsed time and stage distance using a scatter plot
investigate outliers in elapsed time - what stages had unusually long times?
convert elapsed time to hours and create a more readable histogram
eda.qmd Current file
Positron Assistant
Context aware GenAI coding assistance for Data Science
Distribution of Stage Elapsed Times
Frequency
Elapsed Time

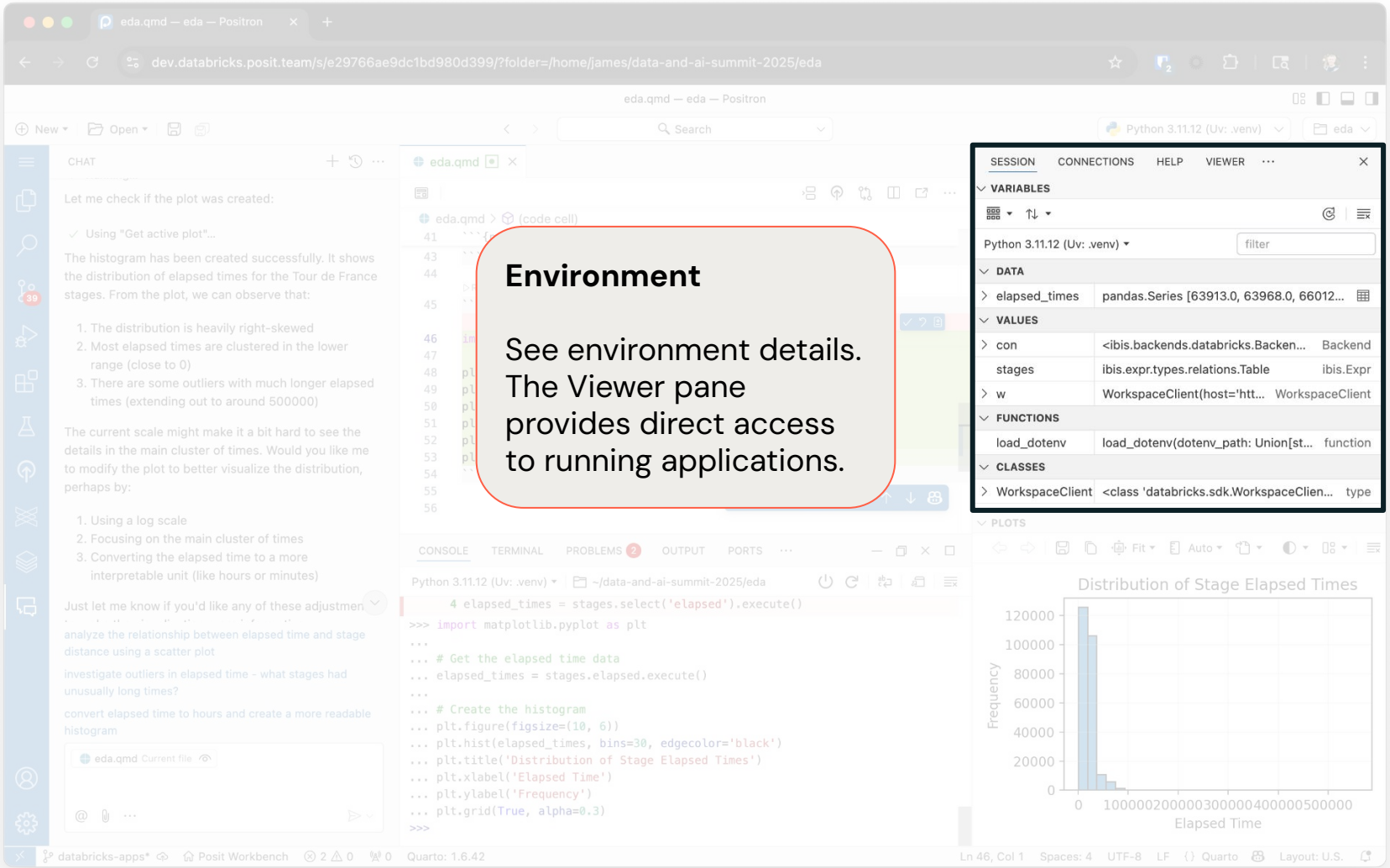

eda.qmd — eda — Positron
dev.databricks.posit.team/s/e29766ae9dc1bd980d399/?folder=/home/james/data-and-ai-summit-2025/eda
Let me check if the plot was created:
Using "Get active plot"...
The histogram has been created successfully. It shows the distribution of elapsed times for the Tour de France stages. From the plot, we can observe that:
1. The distribution is heavily right-skewed
2. Most elapsed times are clustered in the lower range (close to 0)
3. There are some outliers with much longer elapsed times (extending out to around 500000)
The current scale might make it a bit hard to see the details in the main cluster of times. Would you like me to modify the plot to better visualize the distribution, perhaps by:
1. Using a log scale
2. Focusing on the main cluster of times
3. Converting the elapsed time to a more interpretable unit (like hours or minutes)
Just let me know if you'd like any of these adjustmen
analyze the relationship between elapsed time and stage distance using a scatter plot
investigate outliers in elapsed time - what stages had unusually long times?
convert elapsed time to hours and create a more readable histogram
eda.qmd Current file
eda.qmd
eda.qmd > (code cell)
41 ```{python}
43 ```
44
Run Cell | Run Above
45 ```{python}
46 import matplotlib.pyplot as plt
47
48 plt.figure(figsize=(10, 6))
49 plt.hist(stages.elapsed.execute(), bins=50)
50 plt.xlabel('Elapsed Time')
51 plt.ylabel('Frequency')
52 plt.title('Distribution of Stage Elapsed Times')
53 plt.show()
54 ```
55
56
Keep Undo 1 of 1
CONSOLE TERMINAL PROBLEMS 2 OUTPUT PORTS
Code Editor
Full featured editor
SESSION CONNECTIONS HELP VIEWER
VARIABLES
Python 3.11.12 (Uv: .venv)
DATA
elapsed_times pandas.Series [63913.0, 63968.0, 66012...
VALUES
con <ibis.backends.databricks.Backen... Backend
stages ibis.expr.types.relations.Table ibis.Expr
w WorkspaceClient(host='htt... WorkspaceClient
FUNCTIONS
load_dotenv load_dotenv(dotenv_path: Union[st... function
CLASSES
WorkspaceClient <class 'databricks.sdk.WorkspaceClien... type
PLOTS
Distribution of Stage Elapsed Times
Frequency
Elapsed Time
0
20000
40000
60000
80000
100000
120000
0
100000200000300000400000500000
databricks-apps* Posit Workbench Quarto: 1.6.42
Ln 46, Col 1 Spaces: 4 UTF-8 LF Quarto Layout: U.S.

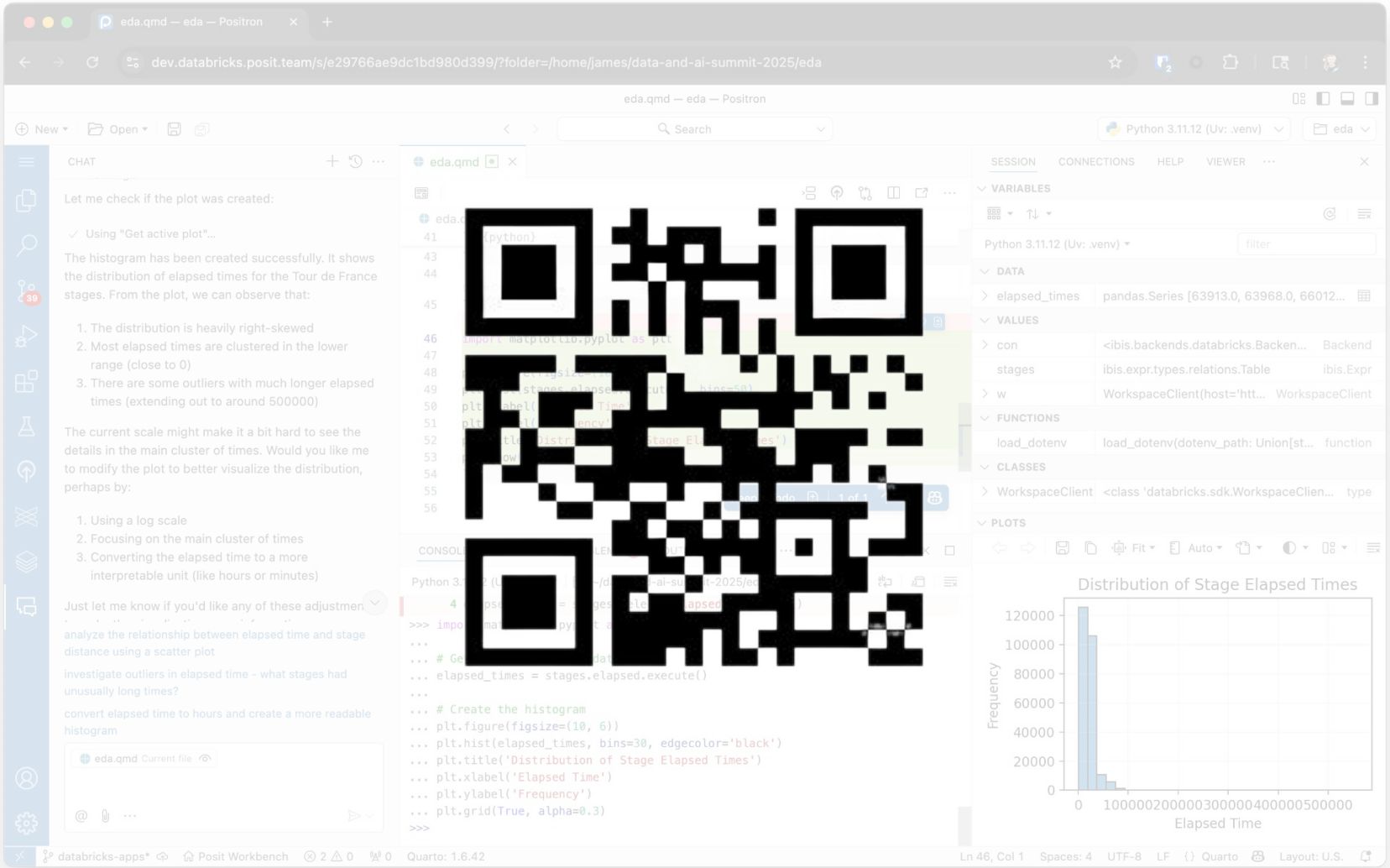

eda.qmd — eda — Positron
dev.databricks.posit.team/s/e29766ae9dc1bd980d399/?folder=/home/james/data-and-ai-summit-2025/eda
Console
Native support for multiple concurrent Python and R consoles
CONSOLE
TERMINAL
PROBLEMS 2
OUTPUT
PORTS
Python 3.11.12 (Uv: .venv)
~/data-and-ai-summit-2025/eda
4 elapsed_times = stages.select('elapsed').execute()
>>> import matplotlib.pyplot as plt
... # Get the elapsed time data
... elapsed_times = stages.elapsed.execute()
... # Create the histogram
... plt.figure(figsize=(10, 6))
... plt.hist(elapsed_times, bins=30, edgecolor='black')
... plt.title('Distribution of Stage Elapsed Times')
... plt.xlabel('Elapsed Time')
... plt.ylabel('Frequency')
... plt.grid(True, alpha=0.3)
>>>
Distribution of Stage Elapsed Times
Frequency
Elapsed Time

Plots

Plots render directly within the IDE

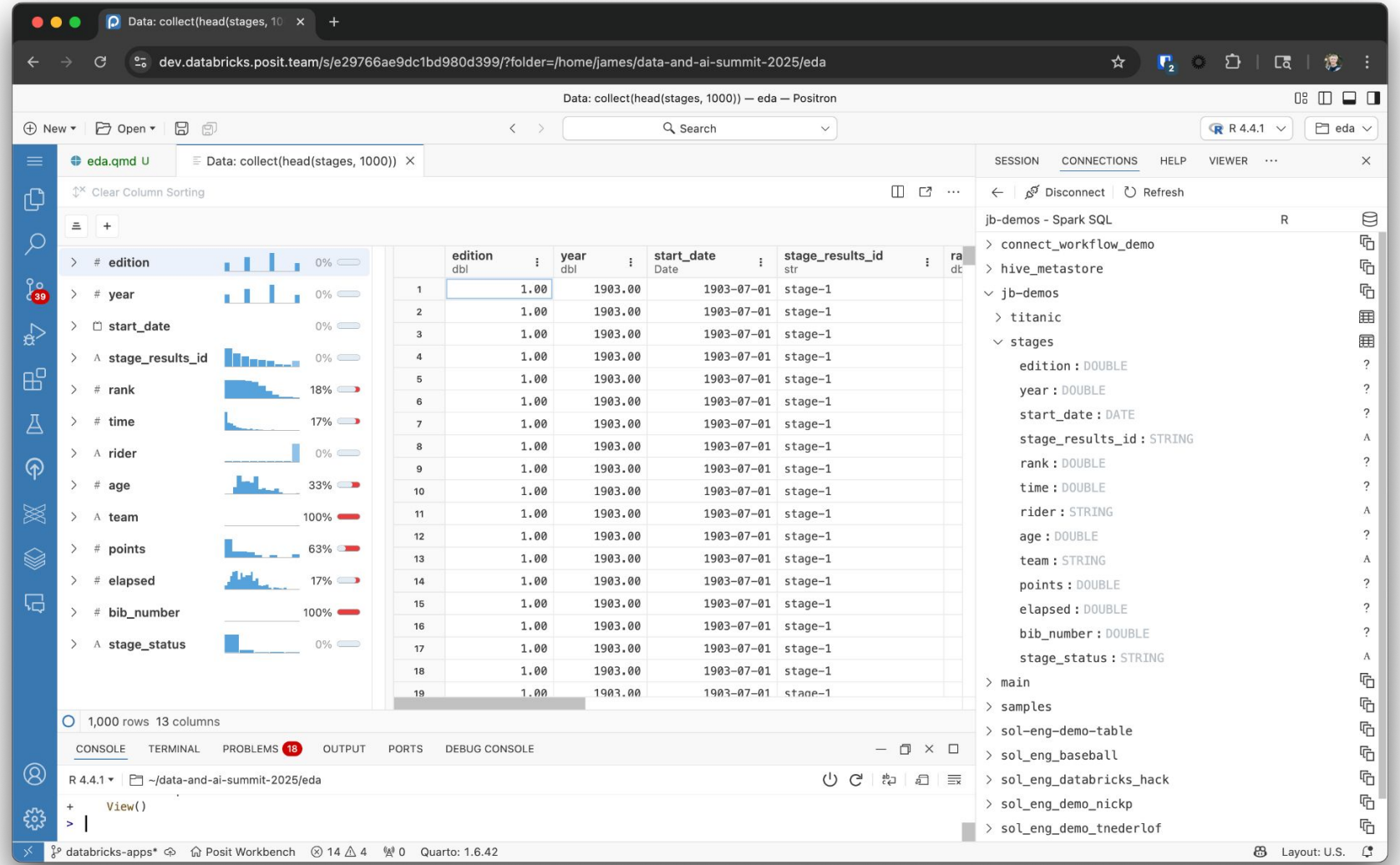


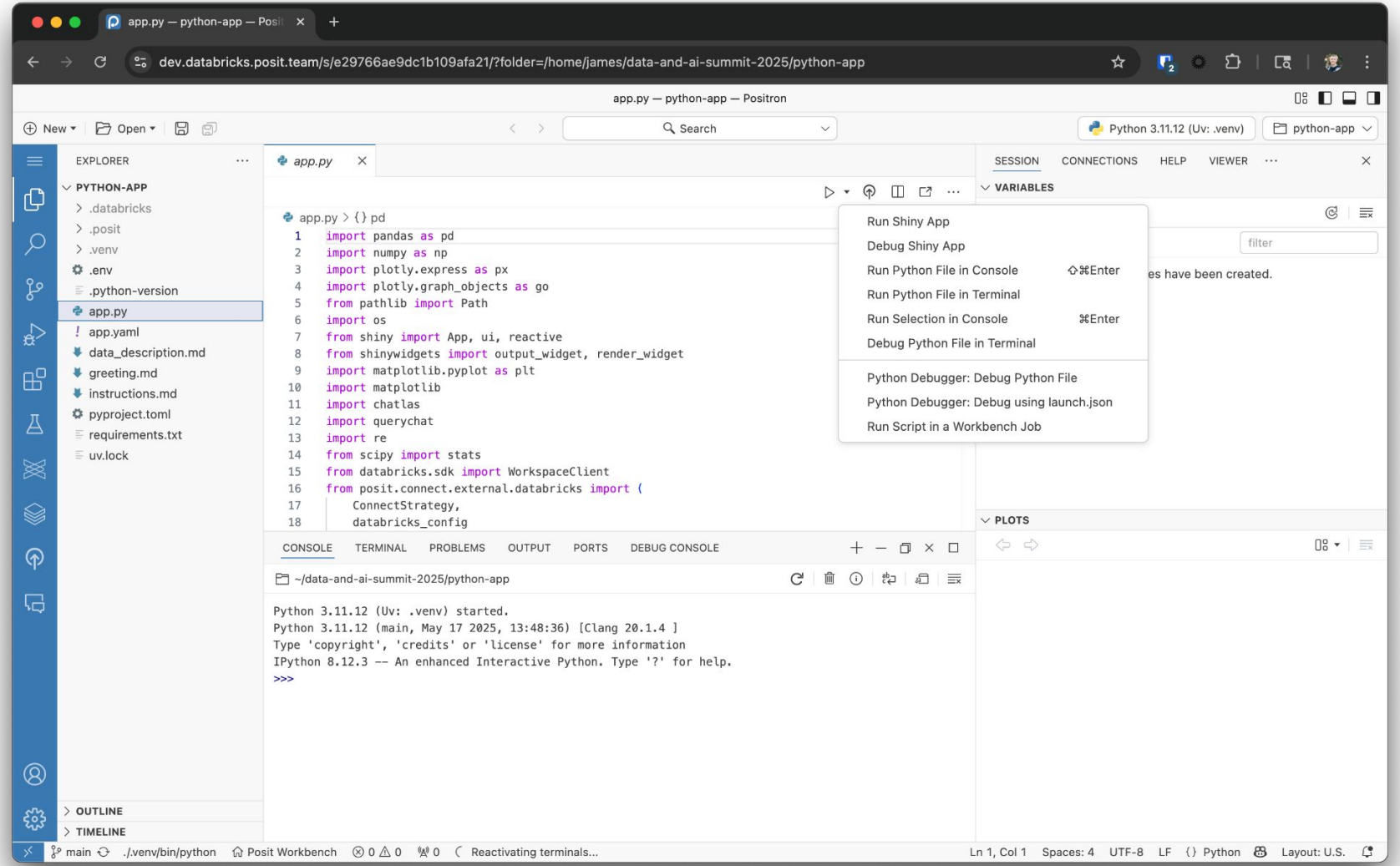

Environment
See environment details. The Viewer pane provides direct access to running applications.
SESSION
CONNECTIONS
HELP
VIEWER
VARIABLES
Python 3.11.12 (Uv: .venv)
filter
DATA
elapsed_times
pandas.Series [63913.0, 63968.0, 66012...
VALUES
con
<ibis.backends.databricks.Backen...
Backend
stages
ibis.expr.types.relations.Table
ibis.Expr
w
WorkspaceClient(host='htt...
WorkspaceClient
FUNCTIONS
load_dotenv
load_dotenv(dotenv_path: Union[st...
function
CLASSES
WorkspaceClient
<class 'databricks.sdk.WorkspaceClien...
type
PLOTS
Distribution of Stage Elapsed Times
Frequency
Elapsed Time
CONSOLE
TERMINAL
PROBLEMS
OUTPUT
PORTS

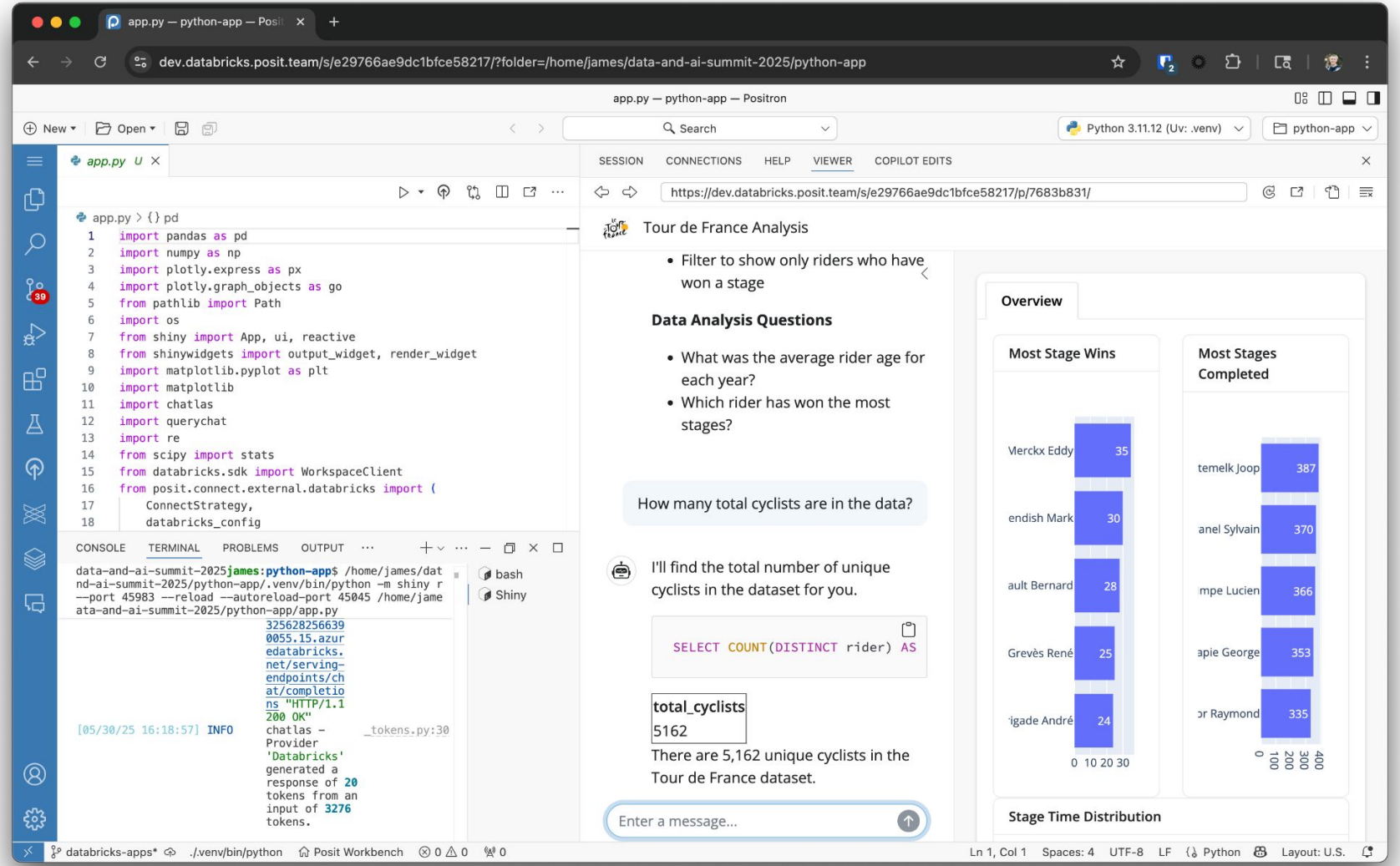

eda.qmd — eda — Positron
dev.databricks.posit.team/s/e29766ae9dc1bd980d399/?folder=/home/james/data-and-ai-summit-2025/eda
New
Open
Search
Python 3.11.12 (Uv: .venv)
eda
CHAT
Let me check if the plot was created:
Using "Get active plot"...
The histogram has been created successfully. It shows the distribution of elapsed times for the Tour de France stages. From the plot, we can observe that:
1. The distribution is heavily right-skewed
2. Most elapsed times are clustered in the lower range (close to 0)
3. There are some outliers with much longer elapsed times (extending out to around 500000)
The current scale might make it a bit hard to see the details in the main cluster of times. Would you like me to modify the plot to better visualize the distribution, perhaps by:
1. Using a log scale
2. Focusing on the main cluster of times
3. Converting the elapsed time to a more interpretable unit (like hours or minutes)
Just let me know if you'd like any of these adjustmen
analyze the relationship between elapsed time and stage distance using a scatter plot
investigate outliers in elapsed time - what stages had unusually long times?
convert elapsed time to hours and create a more readable histogram
eda.qmd Current file
eda.qmd
SESSION
CONNECTIONS
HELP
VIEWER
VARIABLES
Python 3.11.12 (Uv: .venv)
filter
DATA
elapsed_times
pandas.Series [63913.0, 63968.0, 66012...
VALUES
con
<ibis.backends.databricks.Backen...
Backend
stages
ibis.expr.types.relations.Table
ibis.Expr
w
WorkspaceClient(host='htt...
WorkspaceClient
FUNCTIONS
load_dotenv
load_dotenv(dotenv_path: Union[st...
function
CLASSES
WorkspaceClient
<class 'databricks.sdk.WorkspaceClien...
type
PLOTS
Fit
Auto
Distribution of Stage Elapsed Times
Frequency
Elapsed Time
120000
100000
80000
60000
40000
20000
0
0
100000
200000
300000
400000
500000
>>> import matplotlib.pyplot as plt
... elapsed_times = stages.elapsed.execute()
... # Create the histogram
... plt.figure(figsize=(10, 6))
... plt.hist(elapsed_times, bins=30, edgecolor='black')
... plt.title('Distribution of Stage Elapsed Times')
... plt.xlabel('Elapsed Time')
... plt.ylabel('Frequency')
... plt.grid(True, alpha=0.3)
>>>
databricks-apps*
Posit Workbench
Quarto: 1.6.42
Ln 46, Col 1
Spaces: 4
UTF-8
LF
Quarto
Layout: U.S.

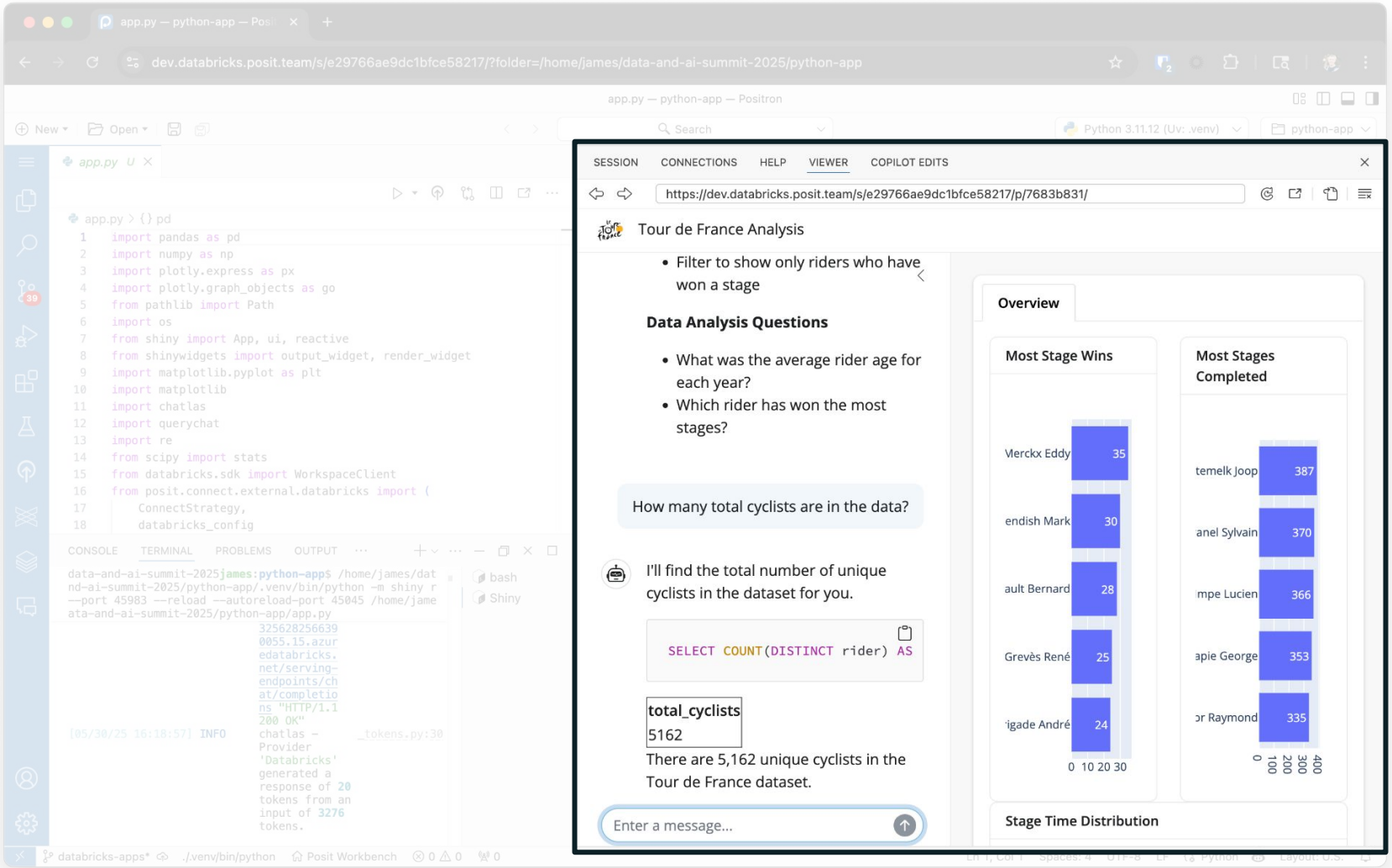
GenAI Applications

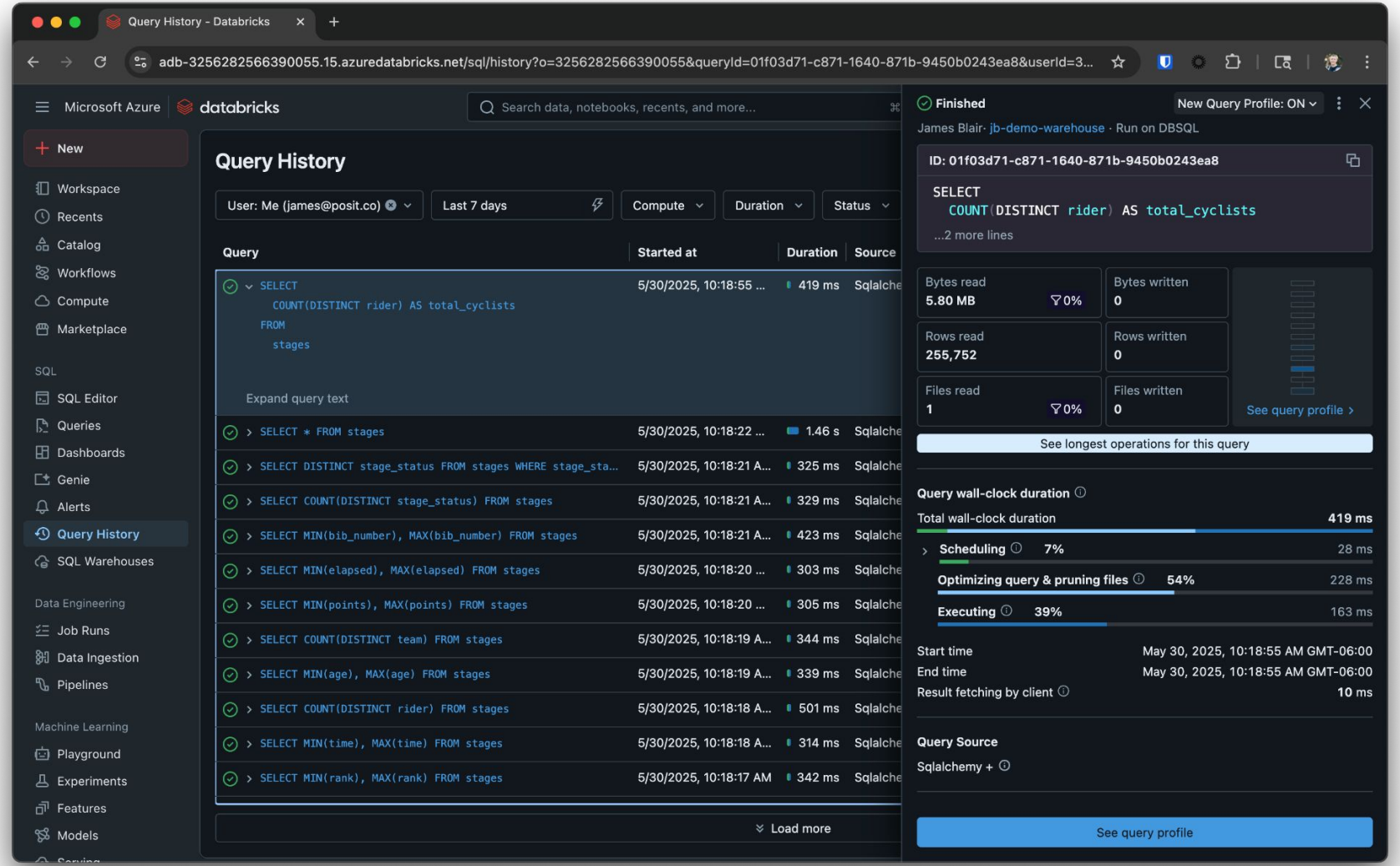
Building intelligent dashboards


Data: collect(head(stages, 10
dev.databricks.posit.team/s/e29766ae9dc1bd980d399/?folder=/home/james/data-and-ai-summit-2025/eda
Data: collect(head(stages, 1000)) — eda — Positron
New
Open
Search
R 4.4.1
eda
eda.qmd U
Data: collect(head(stages, 1000))
Clear Column Sorting
edition 0%
year 0%
start_date 0%
stage_results_id 0%
rank 18%
time 17%
rider 0%
age 33%
team 100%
points 63%
elapsed 17%
bib_number 100%
stage_status 0%
edition dbl
year dbl
start_date Date
stage_results_id str
1,000 rows 13 columns
CONSOLE
TERMINAL
PROBLEMS 18
OUTPUT
PORTS
DEBUG CONSOLE
R 4.4.1 ~/data-and-ai-summit-2025/eda
View()
SESSION
CONNECTIONS
HELP
VIEWER
Disconnect
Refresh
jb-demos - Spark SQL R
connect_workflow_demo
hive_metastore
jb-demos
titanic
stages
edition : DOUBLE
year : DOUBLE
start_date : DATE
stage_results_id : STRING
rank : DOUBLE
time : DOUBLE
rider : STRING
age : DOUBLE
team : STRING
points : DOUBLE
elapsed : DOUBLE
bib_number : DOUBLE
stage_status : STRING
main
samples
sol-eng-demo-table
sol_eng_baseball
sol_eng_databricks_hack
sol_eng_demo_nickp
sol_eng_demo_tnederlof
databricks-apps*
Posit Workbench
14 4
0
Quarto: 1.6.42
Layout: U.S.


app.py — python-app — Posit
dev.databricks.posit.team/s/e29766ae9dc1b109afa21/?folder=/home/james/data-and-ai-summit-2025/python-app
app.py — python-app — Positron
New
Open
Search
Python 3.11.12 (Uv: .venv)
python-app
EXPLORER
PYTHON-APP
.databricks
.posit
.venv
.env
.python-version
app.py
app.yaml
data_description.md
greeting.md
instructions.md
pyproject.toml
requirements.txt
uv.lock
OUTLINE
TIMELINE
app.py
app.py > {} pd
1 import pandas as pd
2 import numpy as np
3 import plotly.express as px
4 import plotly.graph_objects as go
5 from pathlib import Path
6 import os
7 from shiny import App, ui, reactive
8 from shinywidgets import output_widget, render_widget
9 import matplotlib.pyplot as plt
10 import matplotlib
11 import chatlas
12 import querychat
13 import re
14 from scipy import stats
15 from databricks.sdk import WorkspaceClient
16 from posit.connect.external.databricks import (
17 ConnectStrategy,
18 databricks_config
Run Shiny App
Debug Shiny App
Run Python File in Console ⇧⌘Enter
Run Python File in Terminal
Run Selection in Console ⌘Enter
Debug Python File in Terminal
Python Debugger: Debug Python File
Python Debugger: Debug using launch.json
Run Script in a Workbench Job
SESSION
CONNECTIONS
HELP
VIEWER
VARIABLES
filter
es have been created.
PLOTS
CONSOLE
TERMINAL
PROBLEMS
OUTPUT
PORTS
DEBUG CONSOLE
~/data-and-ai-summit-2025/python-app
Python 3.11.12 (Uv: .venv) started.
Python 3.11.12 (main, May 17 2025, 13:48:36) [Clang 20.1.4]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.12.3 -- An enhanced Interactive Python. Type '?' for help.
>>>
main
./.venv/bin/python
Posit Workbench
Reactivating terminals...
Ln 1, Col 1
Spaces: 4
UTF-8
LF
Python
Layout: U.S.


app.py — python-app — Positron
dev.databricks.posit.team/s/e29766ae9dc1bfce58217/?folder=/home/james/data-and-ai-summit-2025/python-app
app.py U
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
from pathlib import Path
import os
from shiny import App, ui, reactive
from shinywidgets import output_widget, render_widget
import matplotlib.pyplot as plt
import matplotlib
import chatlas
import querychat
import re
from scipy import stats
from databricks.sdk import WorkspaceClient
from posit.connect.external.databricks import (
ConnectStrategy,
databricks_config
CONSOLE
TERMINAL
PROBLEMS
OUTPUT
bash
Shiny
SESSION
CONNECTIONS
HELP
VIEWER
COPILOT EDITS
https://dev.databricks.posit.team/s/e29766ae9dc1bfce58217/p/7683b831/
Tour de France Analysis
Filter to show only riders who have won a stage
Data Analysis Questions
What was the average rider age for each year?
Which rider has won the most stages?
How many total cyclists are in the data?
I'll find the total number of unique cyclists in the dataset for you.
SELECT COUNT(DISTINCT rider) AS
total_cyclists
5162
There are 5,162 unique cyclists in the Tour de France dataset.
Enter a message...
Overview
Most Stage Wins
Most Stages Completed
Merckx Eddy
endish Mark
ault Bernard
Grevès René
igade André
35
30
28
25
24
0 10 20 30
temelk Joop
anel Sylvain
mpe Lucien
apie George
or Raymond
387
370
366
353
335
0 100 200 300 400
Stage Time Distribution


app.py — python-app — Positron
dev.databricks.posit.team/s/e29766ae9dc1bfce58217/?folder=/home/james/data-and-ai-summit-2025/python-app
SESSION
CONNECTIONS
HELP
VIEWER
COPILOT EDITS
https://dev.databricks.posit.team/s/e29766ae9dc1bfce58217/p/7683b831/
Tour de France Analysis
Filter to show only riders who have won a stage
Data Analysis Questions
What was the average rider age for each year?
Which rider has won the most stages?
How many total cyclists are in the data?
I'll find the total number of unique cyclists in the dataset for you.
SELECT COUNT(DISTINCT rider) AS
total_cyclists
5162
There are 5,162 unique cyclists in the Tour de France dataset.
Enter a message...
Overview
Most Stage Wins
Most Stages Completed
Stage Time Distribution


Query History - Databricks
adb-3256282566390055.15.azuredatabricks.net/sql/history?o=3256282566390055&queryId=01f03d71-c871-1640-871b-9450b0243ea8&userId=3...
Microsoft Azure
databricks
Search data, notebooks, recents, and more...
New
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
Models
Query History
User: Me (james@posit.co)
Last 7 days
Compute
Duration
Status
Query
Started at
Duration
Source
SELECT COUNT(DISTINCT rider) AS total_cyclists FROM stages
Expand query text
5/30/2025, 10:18:55 ...
419 ms
Sqlalche
SELECT * FROM stages
5/30/2025, 10:18:22 ...
1.46 s
SELECT DISTINCT stage_status FROM stages WHERE stage_sta...
5/30/2025, 10:18:21 A...
325 ms
SELECT COUNT(DISTINCT stage_status) FROM stages
5/30/2025, 10:18:21 A...
329 ms
SELECT MIN(bib_number), MAX(bib_number) FROM stages
5/30/2025, 10:18:21 A...
423 ms
SELECT MIN(elapsed), MAX(elapsed) FROM stages
5/30/2025, 10:18:20 ...
303 ms
SELECT MIN(points), MAX(points) FROM stages
5/30/2025, 10:18:20 ...
305 ms
SELECT COUNT(DISTINCT team) FROM stages
5/30/2025, 10:18:19 A...
344 ms
SELECT MIN(age), MAX(age) FROM stages
5/30/2025, 10:18:19 A...
339 ms
SELECT COUNT(DISTINCT rider) FROM stages
5/30/2025, 10:18:18 A...
501 ms
SELECT MIN(time), MAX(time) FROM stages
5/30/2025, 10:18:18 A...
314 ms
SELECT MIN(rank), MAX(rank) FROM stages
5/30/2025, 10:18:17 AM
342 ms
Load more
Finished
New Query Profile: ON
James Blair · jb-demo-warehouse · Run on DBSQL
ID: 01f03d71-c871-1640-871b-9450b0243ea8
SELECT COUNT(DISTINCT rider) AS total_cyclists
...2 more lines
Bytes read
5.80 MB
0%
Bytes written
0
Rows read
255,752
Rows written
0
Files read
1
0%
Files written
0
See query profile
See longest operations for this query
Query wall-clock duration
Total wall-clock duration
419 ms
Scheduling
7%
28 ms
Optimizing query & pruning files
54%
228 ms
Executing
39%
163 ms
Start time
May 30, 2025, 10:18:55 AM GMT-06:00
End time
May 30, 2025, 10:18:55 AM GMT-06:00
Result fetching by client
10 ms
Query Source
Sqlalchemy +
See query profile

README MIT license

querychat: Chat with your data in any language

querychat is a multilingual package that allows you to chat with your data using natural language queries. It's available for:

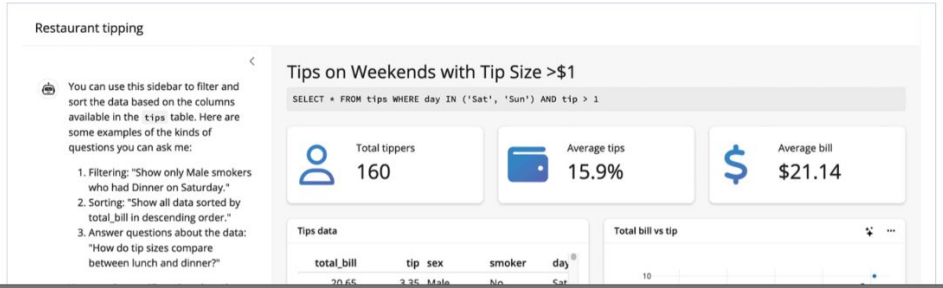
- R - Shiny
- Python - Shiny for Python

Overview

Imagine typing questions like these directly into your dashboard, and seeing the results in realtime:

- "Show only penguins that are not species Gentoo and have a bill length greater than 50mm."
- "Show only blue states with an incidence rate greater than 100 per 100,000 people."
- "What is the average mpg of cars with 6 cylinders?"

querychat is a drop-in component for Shiny that allows users to query a data frame using natural language. The results are available as a reactive data frame, so they can be easily used from Shiny outputs, reactive expressions, downloads, etc.



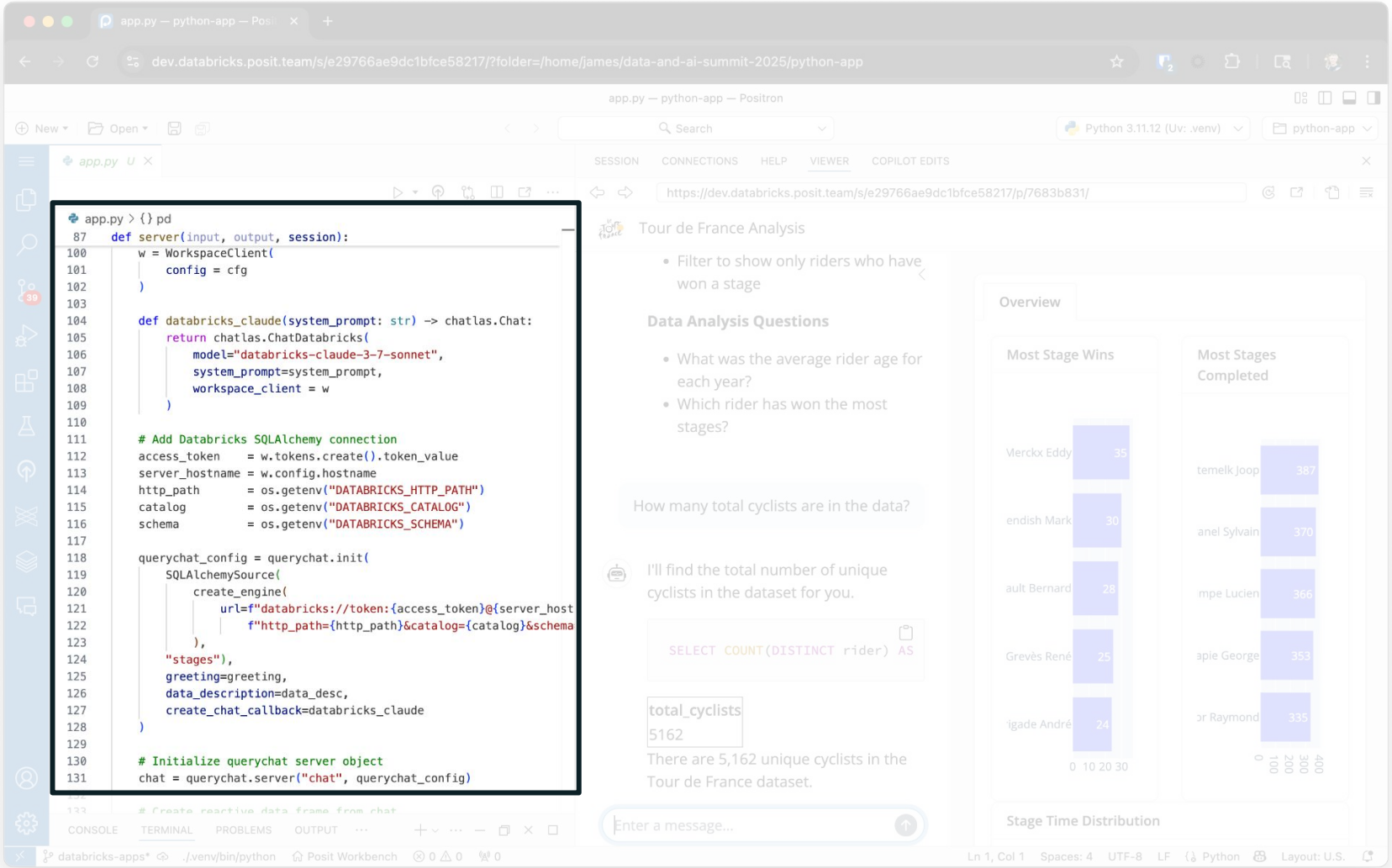
No packages published

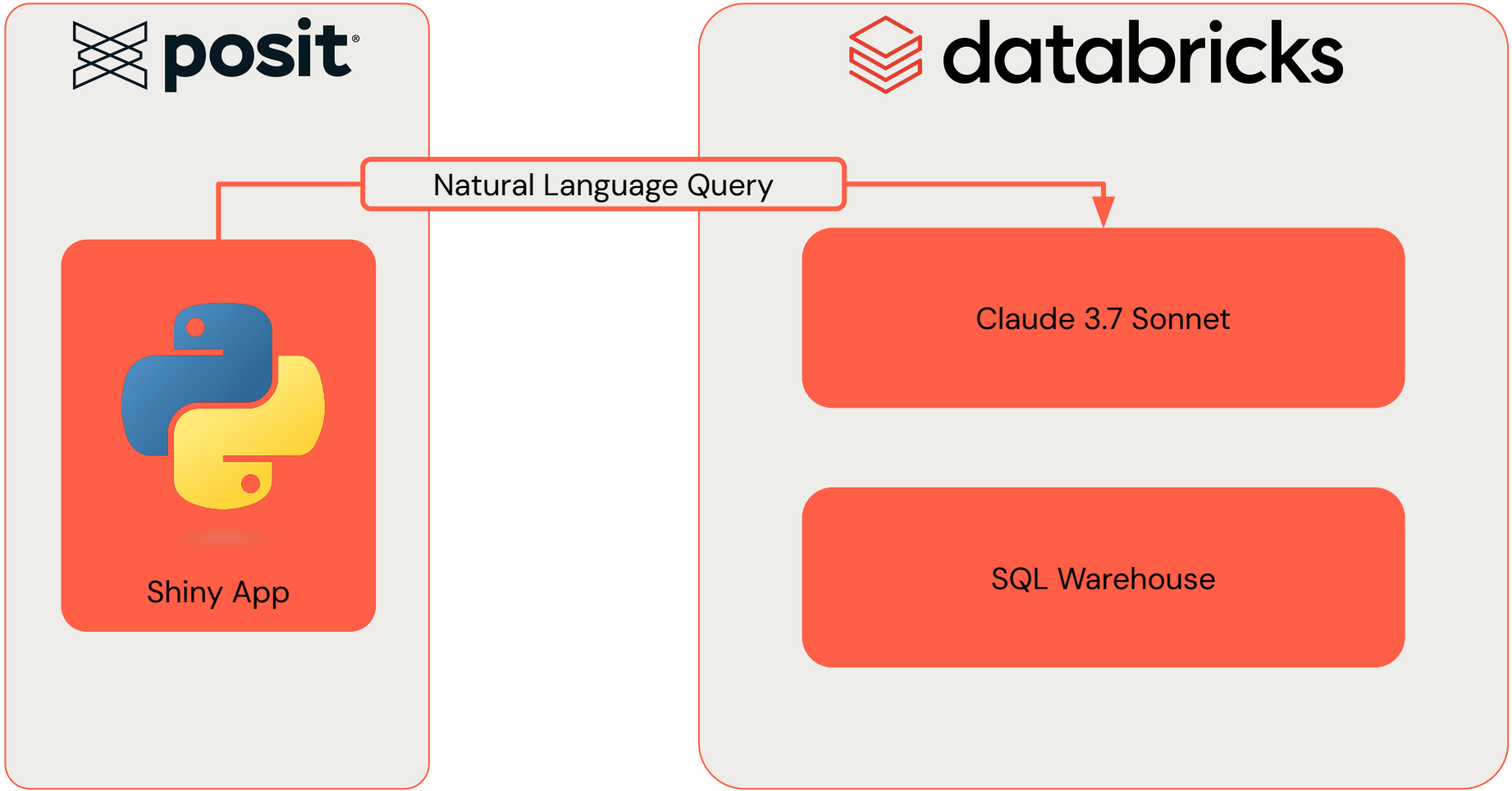
Contributors 4

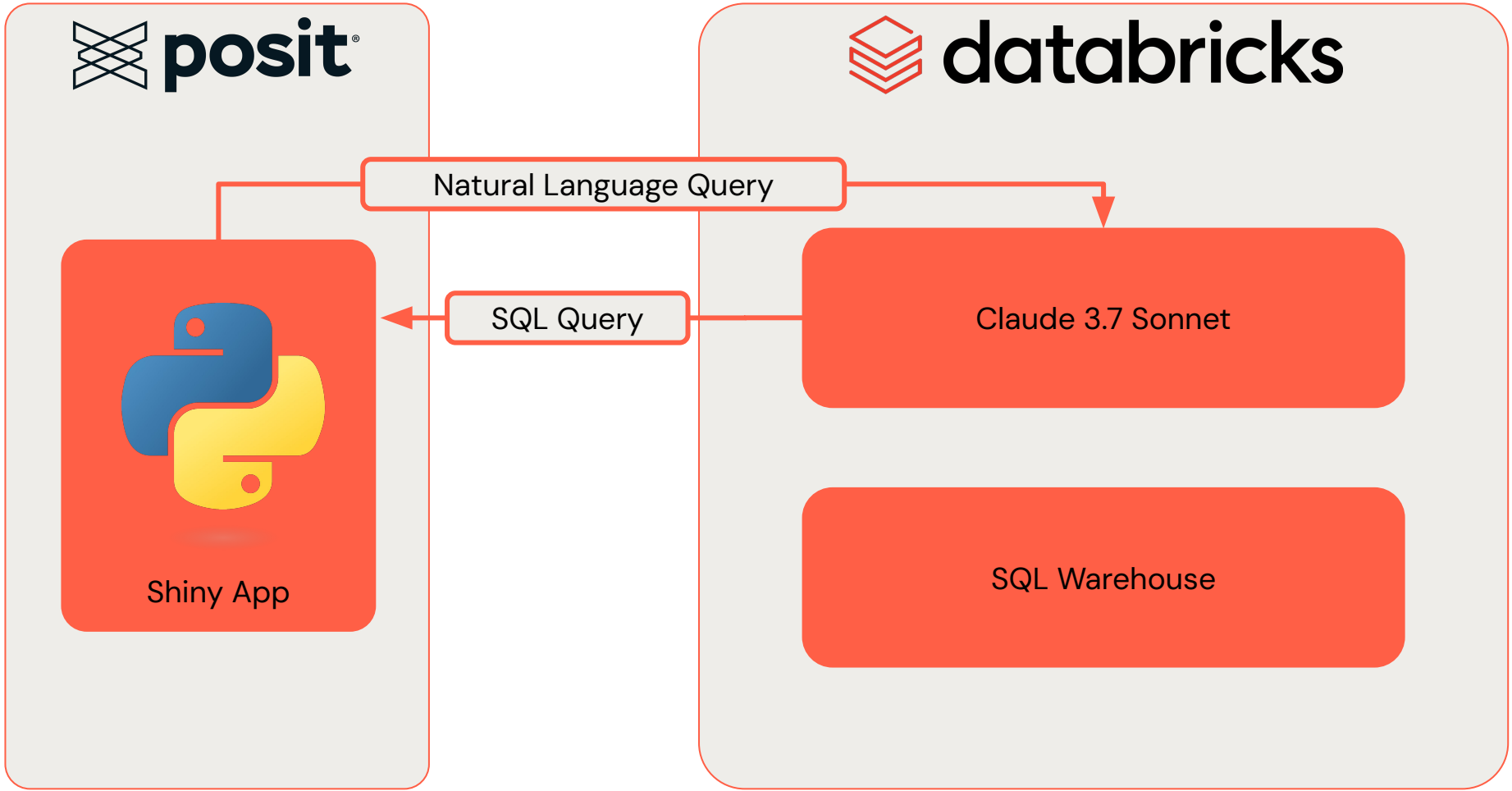
- jcheng5 Joe Cheng
- cpsievert Carson Sievert
- chendaniely Daniel Chen
- schloerke Barret Schloerke

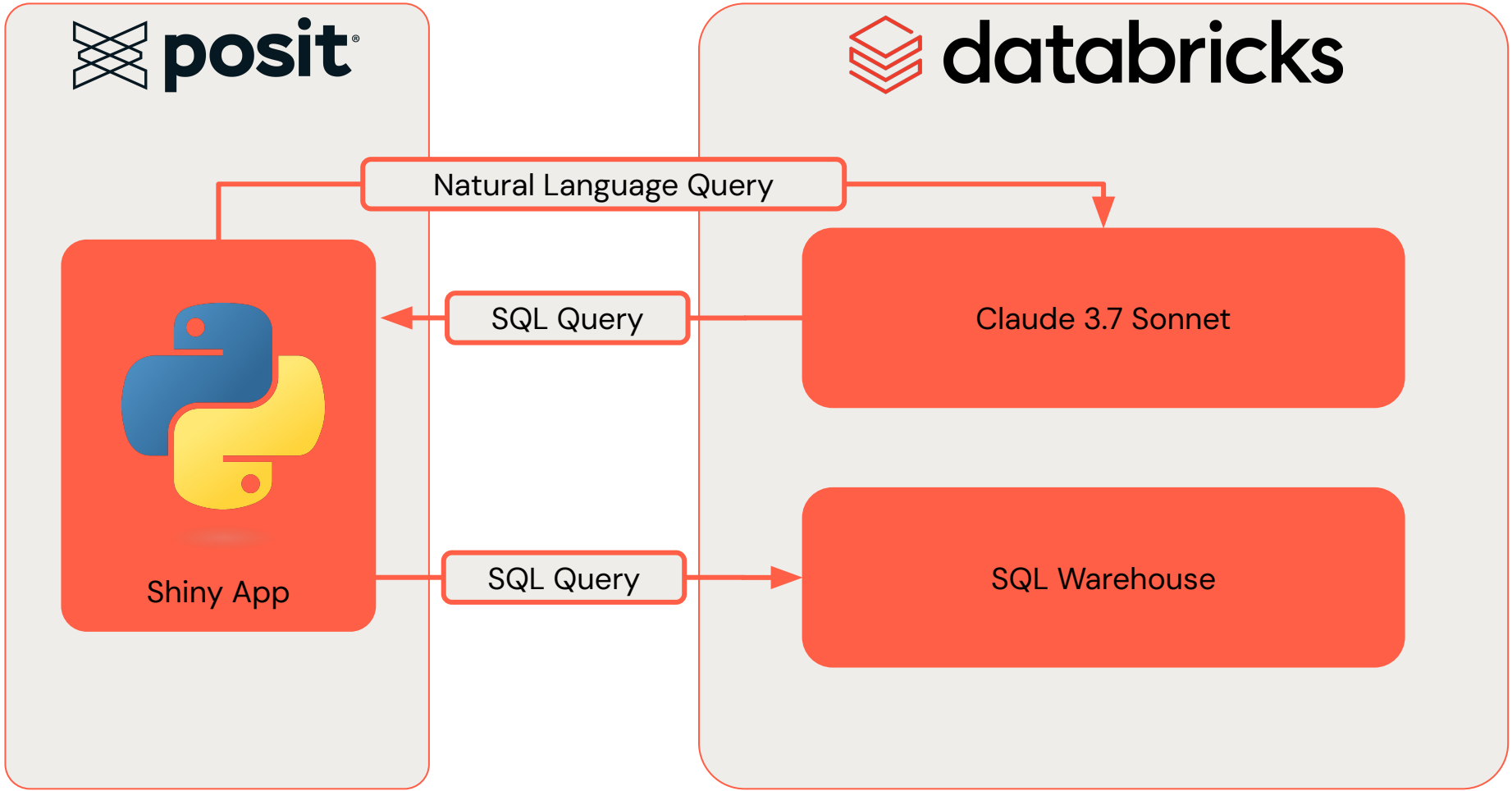
Languages

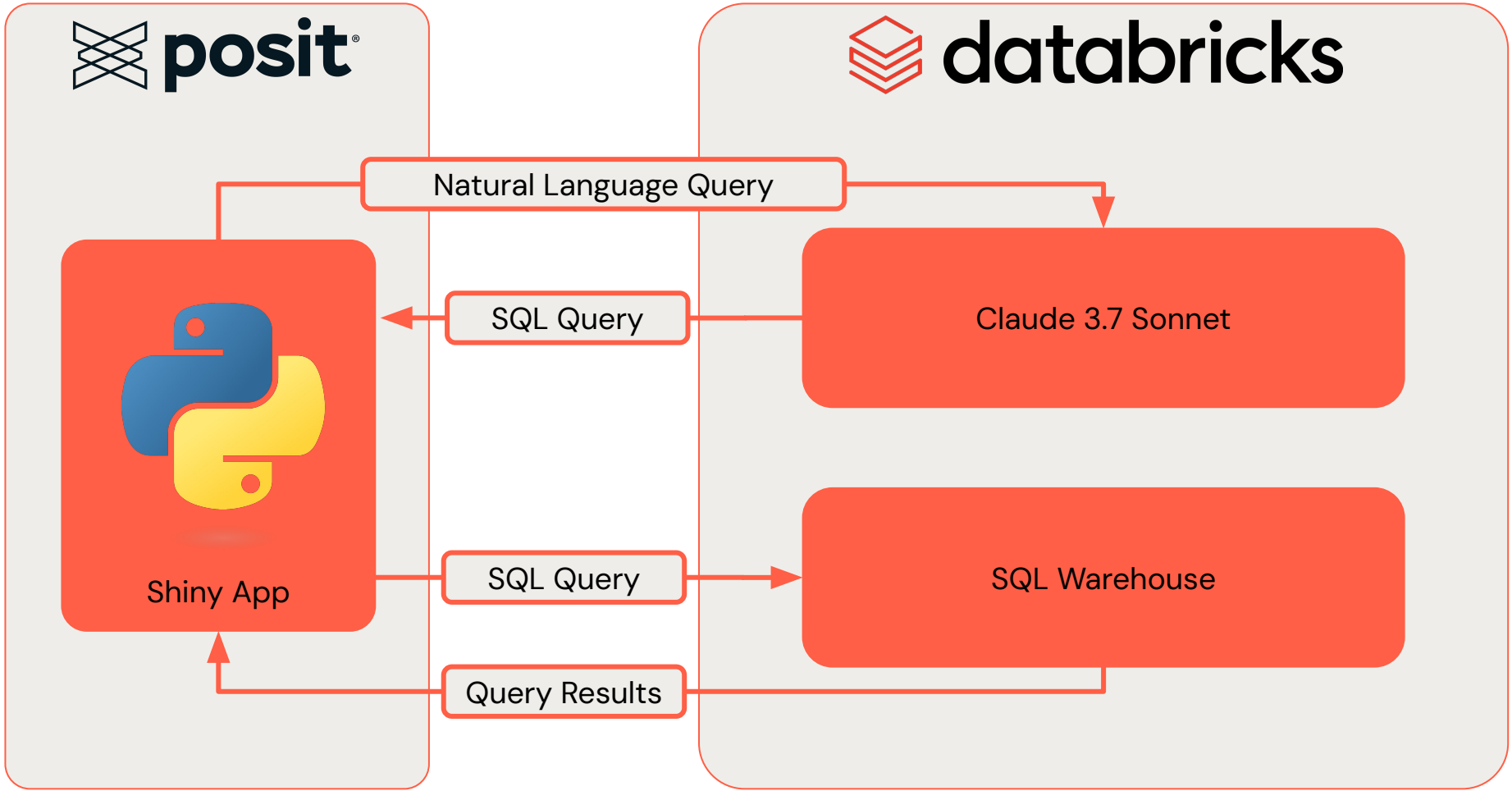
Python 46.7% R 41.2%
CSS 7.2% Makefile 4.9%


app.py — python-app — Positron
dev.databricks.posit.team/s/e29766ae9dc1bfce58217/?folder=/home/james/data-and-ai-summit-2025/python-app
app.py > {} pd
def server(input, output, session):
w = WorkspaceClient(
config = cfg
)
def databricks_claude(system_prompt: str) -> chatlas.Chat:
return chatlas.ChatDatabricks(
model="databricks-claude-3-7-sonnet",
system_prompt=system_prompt,
workspace_client = w
)
# Add Databricks SQLAlchemy connection
access_token = w.tokens.create().token_value
server_hostname = w.config.hostname
http_path = os.getenv("DATABRICKS_HTTP_PATH")
catalog = os.getenv("DATABRICKS_CATALOG")
schema = os.getenv("DATABRICKS_SCHEMA")
querychat_config = querychat.init(
SQLAlchemySource(
create_engine(
url=f"databricks://token:{access_token}@{server_host
f"http_path={http_path}&catalog={catalog}&schema
),
"stages"),
greeting=greeting,
data_description=data_desc,
create_chat_callback=databricks_claude
)
# Initialize querychat server object
chat = querychat.server("chat", querychat_config)
Tour de France Analysis
Filter to show only riders who have won a stage
Data Analysis Questions
What was the average rider age for each year?
Which rider has won the most stages?
How many total cyclists are in the data?
I'll find the total number of unique cyclists in the dataset for you.
SELECT COUNT(DISTINCT rider) AS
total_cyclists
5162
There are 5,162 unique cyclists in the Tour de France dataset.
Enter a message...
Overview
Most Stage Wins
Most Stages Completed
Stage Time Distribution


posit
databricks
Natural Language Query
Shiny App
Claude 3.7 Sonnet
SQL Warehouse

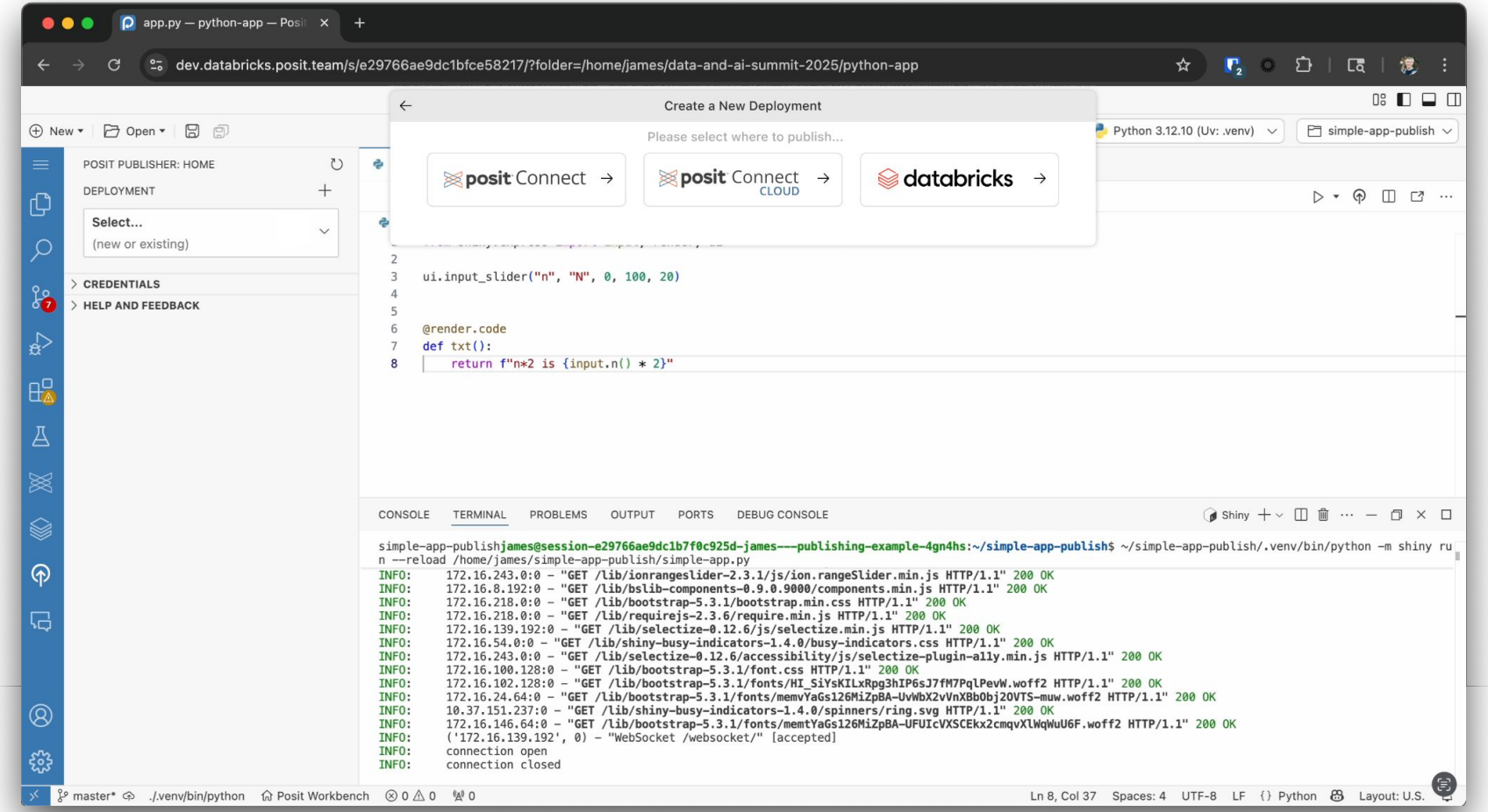

posit
databricks
Natural Language Query
SQL Query
Shiny App
Claude 3.7 Sonnet
SQL Warehouse

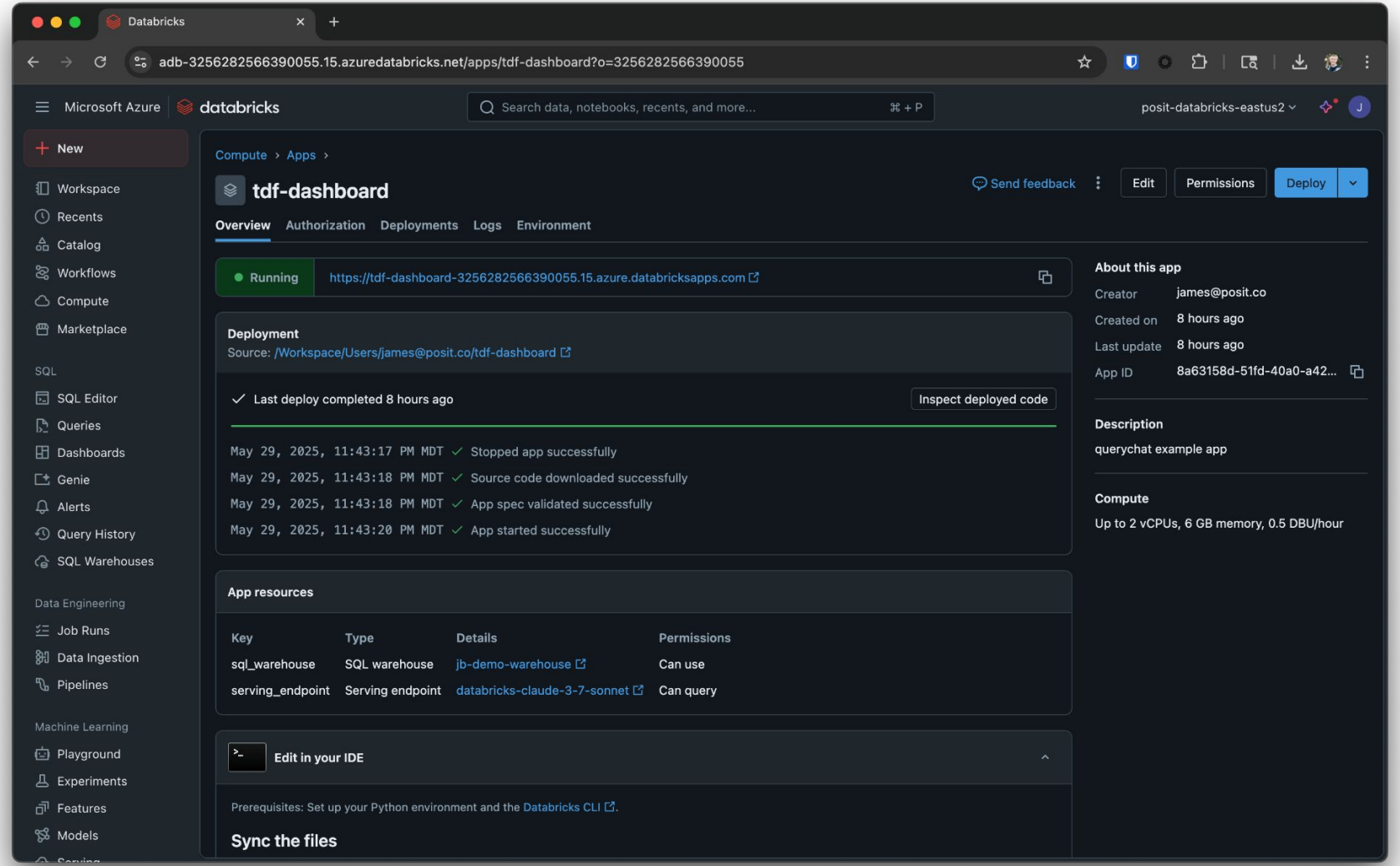

posit
databricks
Natural Language Query
SQL Query
Claude 3.7 Sonnet
Shiny App
SQL Query
SQL Warehouse


posit
databricks
Natural Language Query
SQL Query
Claude 3.7 Sonnet
Shiny App
SQL Query
SQL Warehouse
Query Results

App Deployment

Share securely


app.py — python-app — Posit
dev.databricks.posit.team/s/e29766ae9dc1bfce58217/?folder=/home/james/data-and-ai-summit-2025/python-app
Create a New Deployment
Please select where to publish...
posit Connect
posit Connect CLOUD
databricks
New
Open
Python 3.12.10 (Uv: .venv)
simple-app-publish
POSIT PUBLISHER: HOME
DEPLOYMENT
Select...
(new or existing)
CREDENTIALS
HELP AND FEEDBACK
ui.input_slider("n", "N", 0, 100, 20)
@render.code
def txt():
return f"n*2 is {input.n() * 2}"
CONSOLE
TERMINAL
PROBLEMS
OUTPUT
PORTS
DEBUG CONSOLE
Shiny
simple-app-publishjames@session-e29766ae9dc1b7f0c925d-james---publishing-example-4gn4hs:~/simple-app-publish$ ~/simple-app-publish/.venv/bin/python -m shiny run --reload /home/james/simple-app-publish/simple-app.py
INFO: 172.16.243.0:0 - "GET /lib/ionrangeslider-2.3.1/js/ion.rangeSlider.min.js HTTP/1.1" 200 OK
INFO: 172.16.8.192:0 - "GET /lib/bslib-components-0.9.0.9000/components.min.js HTTP/1.1" 200 OK
INFO: 172.16.218.0:0 - "GET /lib/bootstrap-5.3.1/bootstrap.min.css HTTP/1.1" 200 OK
INFO: 172.16.218.0:0 - "GET /lib/requirejs-2.3.6/require.min.js HTTP/1.1" 200 OK
INFO: 172.16.139.192:0 - "GET /lib/selectize-0.12.6/js/selectize.min.js HTTP/1.1" 200 OK
INFO: 172.16.54.0:0 - "GET /lib/shiny-busy-indicators-1.4.0/busy-indicators.css HTTP/1.1" 200 OK
INFO: 172.16.243.0:0 - "GET /lib/selectize-0.12.6/accessibility/js/selectize-plugin-a11y.min.js HTTP/1.1" 200 OK
INFO: 172.16.100.128:0 - "GET /lib/bootstrap-5.3.1/font.css HTTP/1.1" 200 OK
INFO: 172.16.102.128:0 - "GET /lib/bootstrap-5.3.1/fonts/HI_SiYsKILxRpg3hIP6sJ7fM7PqlPevW.woff2 HTTP/1.1" 200 OK
INFO: 172.16.24.64:0 - "GET /lib/bootstrap-5.3.1/fonts/memvYaGs126MiZpBA-UvWbX2vVnXBbObj2OVTS-muw.woff2 HTTP/1.1" 200 OK
INFO: 10.37.151.237:0 - "GET /lib/shiny-busy-indicators-1.4.0/spinners/ring.svg HTTP/1.1" 200 OK
INFO: 172.16.146.64:0 - "GET /lib/bootstrap-5.3.1/fonts/memtYaGs126MiZpBA-UFUIcVXSCEkx2cmqvXlWqWuU6F.woff2 HTTP/1.1" 200 OK
INFO: ('172.16.139.192', 0) - "WebSocket /websocket/" [accepted]
INFO: connection open
INFO: connection closed
master*
./.venv/bin/python
Posit Workbench
Ln 8, Col 37
Spaces: 4
UTF-8
LF
Python
Layout: U.S.


Databricks
adb-3256282566390055.15.azuredatabricks.net/apps/tdf-dashboard?o=3256282566390055
Microsoft Azure
databricks
Search data, notebooks, recents, and more...
⌘ + P
posit-databricks-eastus2
New
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
Models
Compute › Apps ›
tdf-dashboard
Send feedback
Edit
Permissions
Deploy
Overview
Authorization
Deployments
Logs
Environment
Running
https://tdf-dashboard-3256282566390055.15.azure.databricksapps.com
Deployment
Source: /Workspace/Users/james@posit.co/tdf-dashboard
Last deploy completed 8 hours ago
Inspect deployed code
May 29, 2025, 11:43:17 PM MDT Stopped app successfully
May 29, 2025, 11:43:18 PM MDT Source code downloaded successfully
May 29, 2025, 11:43:18 PM MDT App spec validated successfully
May 29, 2025, 11:43:20 PM MDT App started successfully
App resources
Key
Type
Details
Permissions
sql_warehouse
SQL warehouse
jb-demo-warehouse
Can use
serving_endpoint
Serving endpoint
databricks-claude-3-7-sonnet
Can query
Edit in your IDE
Prerequisites: Set up your Python environment and the Databricks CLI.
Sync the files
About this app
Creator
james@posit.co
Created on
8 hours ago
Last update
8 hours ago
App ID
8a63158d-51fd-40a0-a42...
Description
querychat example app
Compute
Up to 2 vCPUs, 6 GB memory, 0.5 DBU/hour

Databricks | Tour de France Analysis

tdf-dashboard-3256282566390055.15.azure.databricksapps.com

Tour de France Analysis

- What was the average rider age for each year?
- Which rider has won the most stages?

Filter to the most recent decade

I'll filter the dashboard to show data from the most recent decade in the dataset.

```
SELECT MAX(year) FROM stages
```

max(year)
2019.0

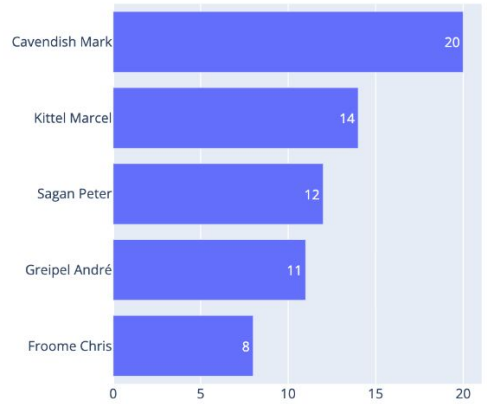
Based on the data, the most recent year in the dataset is 2019. I'll filter to show the last decade (2010-2019).

```
SELECT * FROM stages WHERE year
```

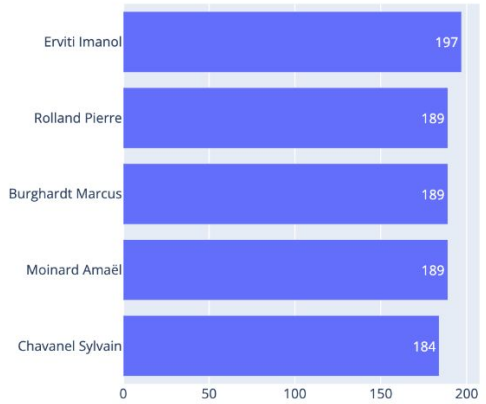
I've filtered the dashboard to show only the Tour de France data from the most recent decade (2010-2019).

Enter a message...

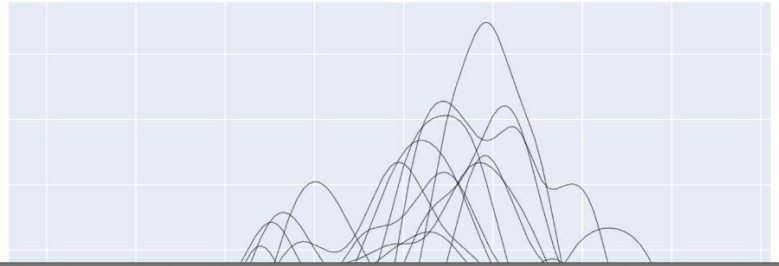
Most Stage Wins

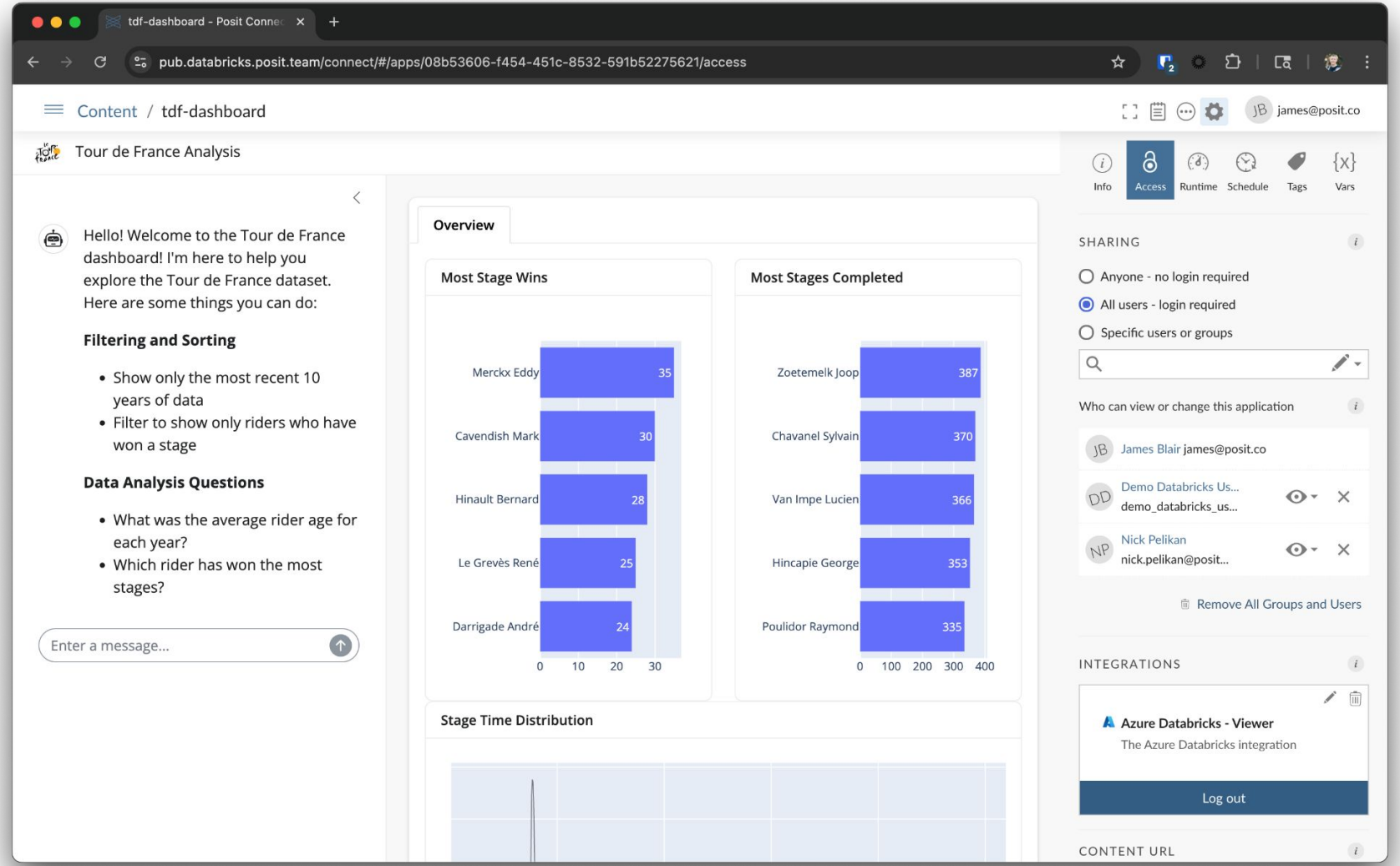


Most Stages Completed

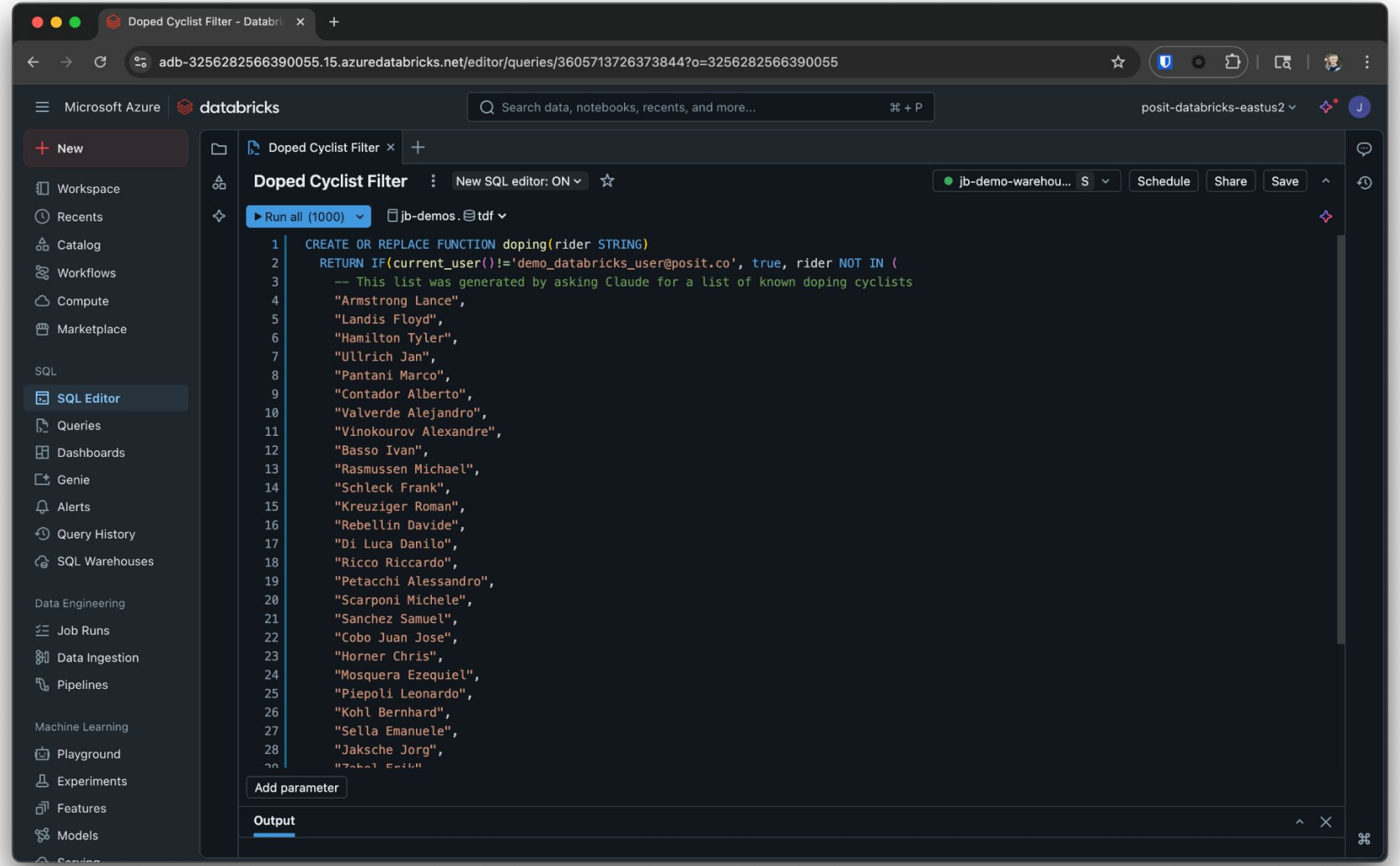


Stage Time Distribution




tdf-dashboard - Posit Connec
pub.databricks.posit.team/connect/#/apps/08b53606-f454-451c-8532-591b52275621/access
Content / tdf-dashboard
james@posit.co
Tour de France Analysis
Info
Access
Runtime
Schedule
Tags
Vars
Hello! Welcome to the Tour de France dashboard! I'm here to help you explore the Tour de France dataset. Here are some things you can do:
Filtering and Sorting
Show only the most recent 10 years of data
Filter to show only riders who have won a stage
Data Analysis Questions
What was the average rider age for each year?
Which rider has won the most stages?
Enter a message...
Overview
Most Stage Wins
Merckx Eddy 35
Cavendish Mark 30
Hinault Bernard 28
Le Grevès René 25
Darrigade André 24
0 10 20 30
Most Stages Completed
Zoetemelk Joop 387
Chavanel Sylvain 370
Van Impe Lucien 366
Hincapie George 353
Poulidor Raymond 335
0 100 200 300 400
Stage Time Distribution
SHARING
Anyone - no login required
All users - login required
Specific users or groups
Who can view or change this application
JB James Blair james@posit.co
DD Demo Databricks Us... demo_databricks_us...
NP Nick Pelikan nick.pelikan@posit...
Remove All Groups and Users
INTEGRATIONS
Azure Databricks - Viewer
The Azure Databricks integration
Log out
CONTENT URL


UNITED STATES
POSTAL SERVICE
BERRY FLOOR TREK AMD
CHAMPS-ELYSEES


Doped Cyclist Filter - Databri
adb-3256282566390055.15.azuredatabricks.net/editor/queries/3605713726373844?o=3256282566390055
Microsoft Azure
databricks
Search data, notebooks, recents, and more...
⌘ + P
posit-databricks-eastus2
New
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
Playground
Experiments
Features
Models
Doped Cyclist Filter
Doped Cyclist Filter
New SQL editor: ON
jb-demo-warehou... S
Schedule
Share
Save
Run all (1000)
jb-demos . tdf
1 CREATE OR REPLACE FUNCTION doping(rider STRING)
2 RETURN IF(current_user()!='demo_databricks_user@posit.co', true, rider NOT IN (
3 -- This list was generated by asking Claude for a list of known doping cyclists
4 "Armstrong Lance",
5 "Landis Floyd",
6 "Hamilton Tyler",
7 "Ullrich Jan",
8 "Pantani Marco",
9 "Contador Alberto",
10 "Valverde Alejandro",
11 "Vinokourov Alexandre",
12 "Basso Ivan",
13 "Rasmussen Michael",
14 "Schleck Frank",
15 "Kreuziger Roman",
16 "Rebellin Davide",
17 "Di Luca Danilo",
18 "Ricco Riccardo",
19 "Petacchi Alessandro",
20 "Scarponi Michele",
21 "Sanchez Samuel",
22 "Cobo Juan Jose",
23 "Horner Chris",
24 "Mosquera Ezequiel",
25 "Piepoli Leonardo",
26 "Kohl Bernhard",
27 "Sella Emanuele",
28 "Jaksche Jorg",
Add parameter
Output

tdf-dashboard - Posit Connec

pub.databricks.posit.team/connect/#/apps/08b53606-f454-451c-8532-591b52275621/access

Content / tdf-dashboard

james@posit.co

Tour de France Analysis

won a stage

Data Analysis Questions

- What was the average rider age for each year?
- Which rider has won the most stages?

How many stages did Lance Armstrong win?

I'll find out how many stages Lance Armstrong won in the Tour de France.

```
SELECT COUNT(*) AS stages_won
FROM stages
WHERE rider LIKE '%Armstrong%La
```

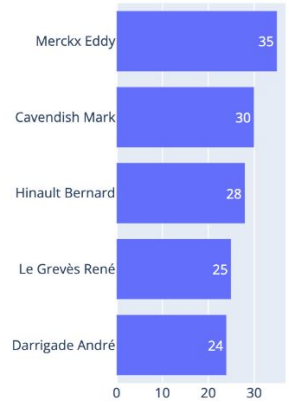
stages_won
24

Lance Armstrong won 24 stages in the Tour de France according to this dataset.

Enter a message...

Overview

Most Stage Wins



Most Stages Completed

Stage Time Distribution

Info | Access | Runtime | Schedule | Tags | Vars

SHARING

- Anyone - no login required
- All users - login required
- Specific users or groups

Who can view or change this application

JB James Blair james@posit.co

DD Demo Databricks Us... demo_databricks_us...

NP Nick Pelikan nick.pelikan@posit...

Remove All Groups and Users

INTEGRATIONS

Azure Databricks - Viewer

The Azure Databricks integration

Log out

CONTENT URL

tdf-dashboard - Posit Connec

pub.databricks.posit.team/connect/#/apps/08b53606-f454-451c-8532-591b52275621

Incognito

Content / tdf-dashboard

demo_databricks_user@posit.co

Tour de France Analysis

How many stages did Lance Armstrong win?

I'll check how many stages Lance Armstrong won in the Tour de France.

```
SELECT COUNT(*) AS stage_wins FI
```

stage_wins
0

Let me check if there might be a different way his name is recorded in the database:

```
SELECT DISTINCT rider FROM stag
```

rider
Lance Pascal

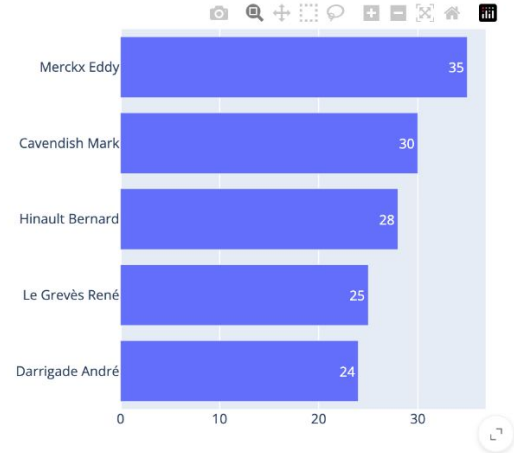
I need to check if there's another way Lance Armstrong might be listed:

```
SELECT DISTINCT rider FROM stag
```

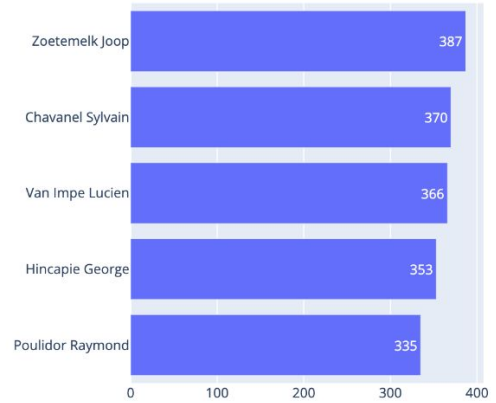
Enter a message...

Overview

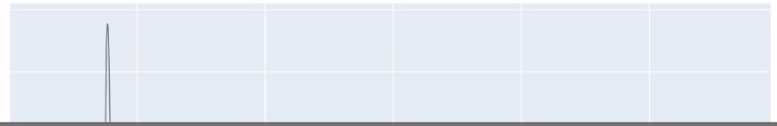
Most Stage Wins



Most Stages Completed



Stage Time Distribution



tdf-dashboard - Posit Connec

pub.databricks.posit.team/connect/#/apps/08b53606-f454-451c-8532-591b52275621

Incognito

Content / tdf-dashboard

demo_databricks_user@posit.co

Tour de France Analysis

	rider	stage_wins
Merckx Eddy		35
Cavendish Mark		30
Hinault Bernard		28
Le Grevès René		25
Darrigade André		24

(Showing only the first 5 rows out of 10.)

Based on my search of the database, Lance Armstrong doesn't appear to be listed among the riders in this dataset. This could be because:

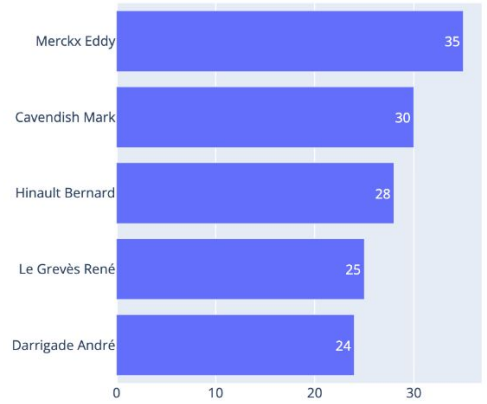
1. His name might be recorded differently than expected
2. His results may have been removed from the dataset due to his disqualification from the Tour de France (his 7 Tour de France titles from 1999-2005 were stripped due to doping violations)

According to official Tour de France records, Lance Armstrong had 22 stage wins before his disqualification, but these appear to have been removed from this particular dataset.

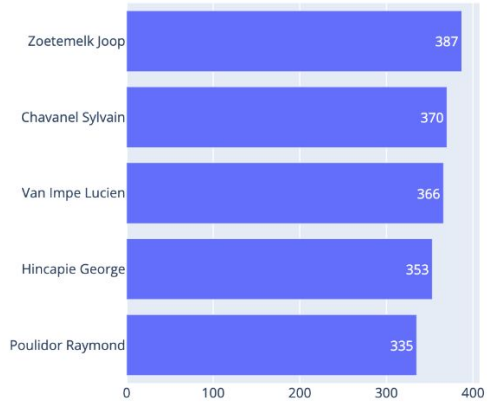
Enter a message...

Overview

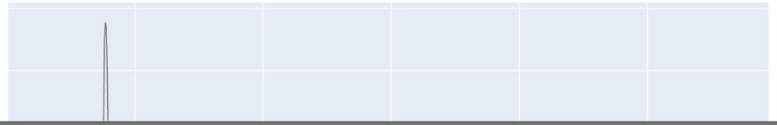
Most Stage Wins



Most Stages Completed



Stage Time Distribution



Conclusion

And a look towards the future

In Review

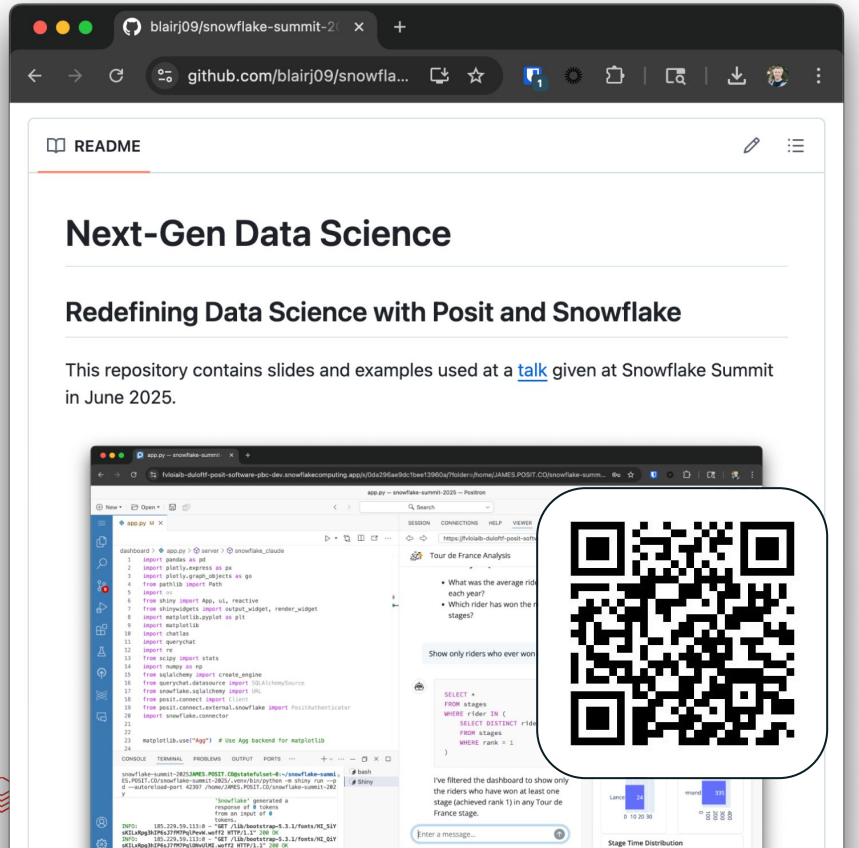
- Positron is a next-gen IDE purpose built for data science and is **best in class** for developing and deploying interactive data apps
- Databricks OAuth support in Posit Workbench **simplifies Databricks access** for Python and R developers
- `querychat` provides a simple interface for building intuitive, **GenAI powered app interfaces** that work directly with Databricks
- Publish and deploy applications to Posit Connect or Databricks Apps for **secure, scalable access** across your organization
- Unity Catalog governance and access controls can be used to **limit who has access to what data**

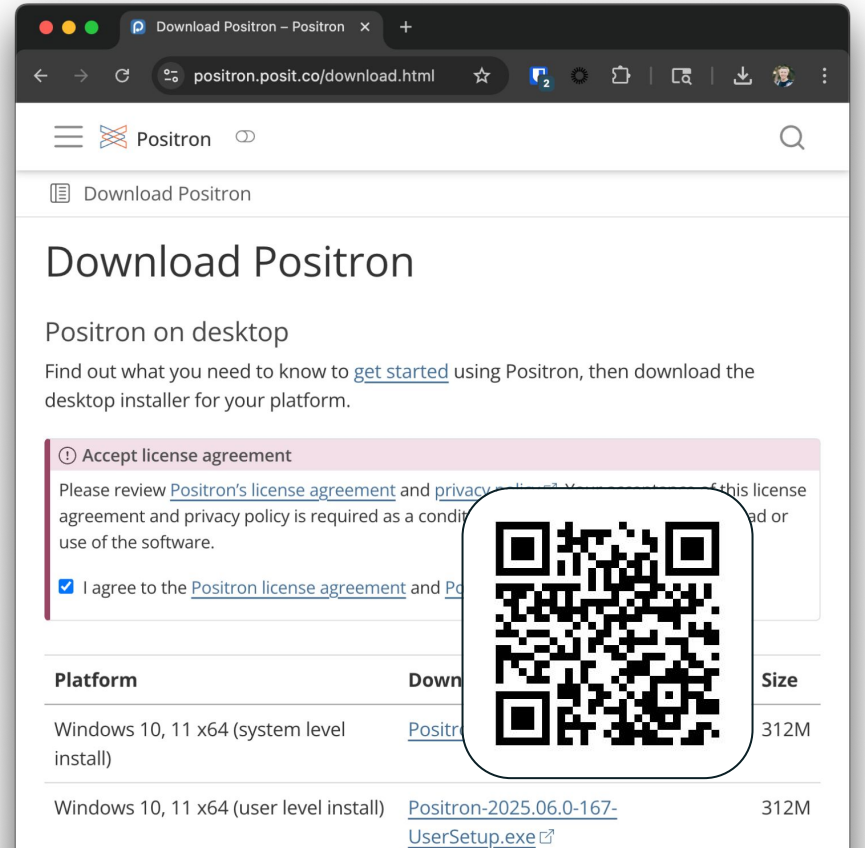
What's Next

- One-click publishing for Databricks Apps
- Posit tools available within Databricks
- Databricks-hosted LLM support for Positron Assistant
- Improved SQL support within Positron

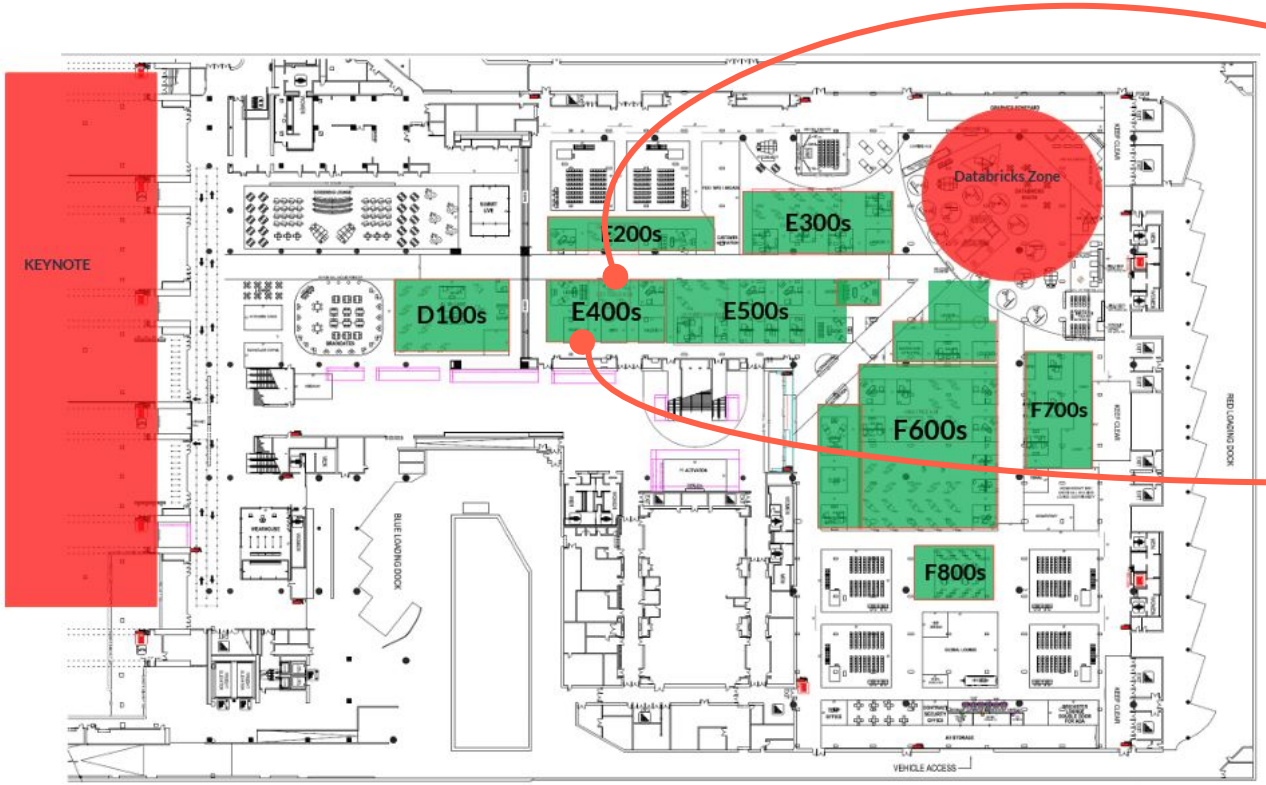
*These are not promises but rather aspirations

Learn More


blairj09/snowflake-summit-2
github.com/blairj09/snowfla...
README
Next-Gen Data Science
Redefining Data Science with Posit and Snowflake
This repository contains slides and examples used at a talk given at Snowflake Summit in June 2025.


Download Positron – Positron
positron.posit.co/download.html
Positron
Download Positron
Download Positron
Positron on desktop
Find out what you need to know to get started using Positron, then download the desktop installer for your platform.
Accept license agreement
Please review Positron's license agreement and privacy
agreement and privacy policy is required as a condit
use of the software.
I agree to the Positron license agreement and Po
Platform
Down
Size
Windows 10, 11 x64 (system level install)
Positr
312M
Windows 10, 11 x64 (user level install)
Positron-2025.06.0-167-UserSetup.exe
312M

Visit Us



Posit Booth
E405

Posit Lounge
402


databricks