

NEXT-GEN DATA SCIENCE

Redefining Data Science with Posit and Snowflake

James Blair
Senior Product Manager | Posit, PBC



© 2025 Snowflake Inc. All Rights Reserved





James Blair

**Senior Product
Manager
Posit, PBC**

AGENDA

- **Posit and Snowflake**
Native App and Snowpark Container Services
- **AI Assisted Exploratory Data Analysis**
Python EDA from Positron
- **App Development**
Creating interactive, AI enabled data applications in Snowflake
- **App Deployment**
Hosting data applications in Snowflake with RBAC
- **Conclusion**



POSIT AND SNOWFLAKE



© 2025 Snowflake Inc. All Rights Reserved

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.

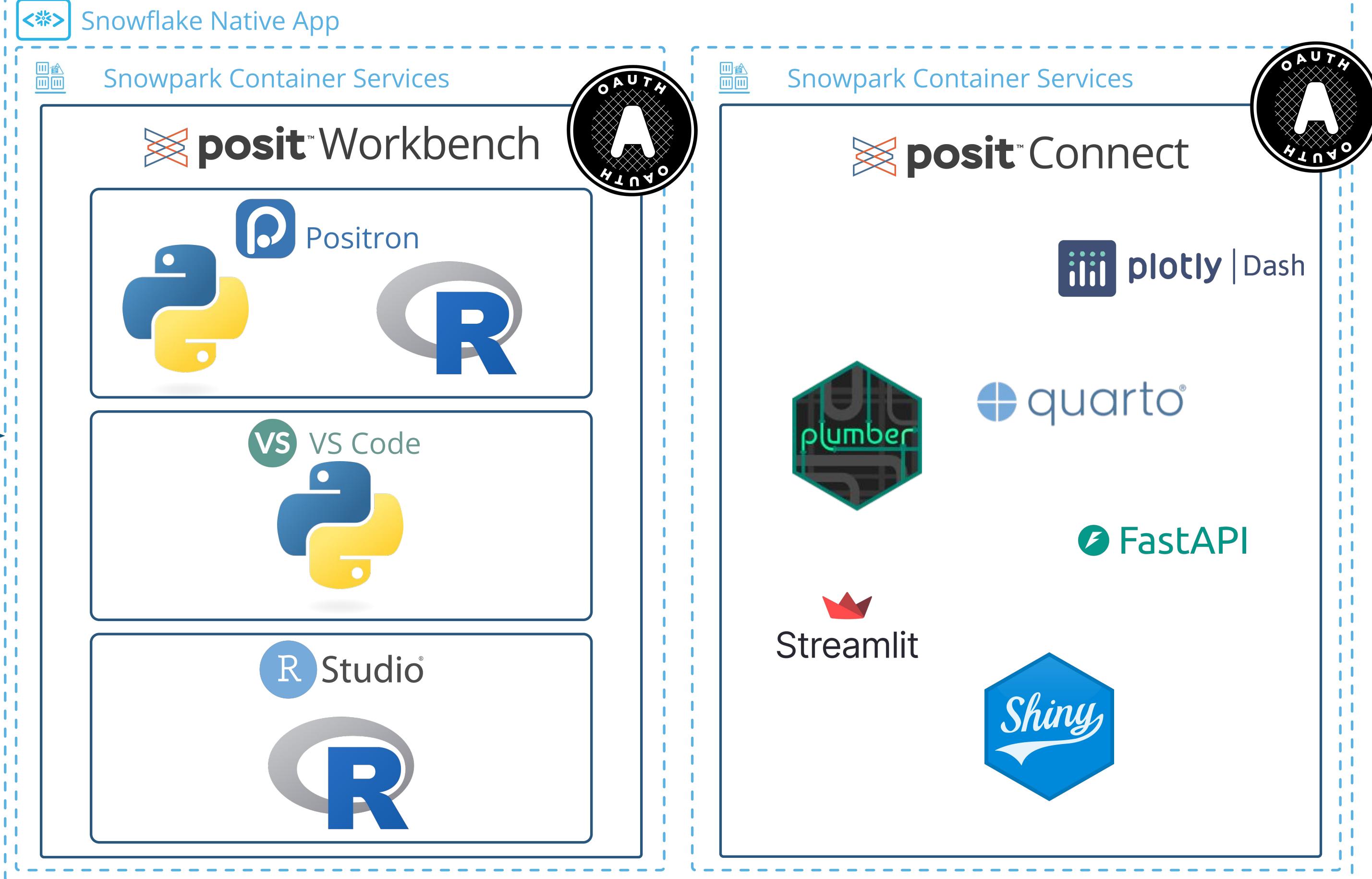
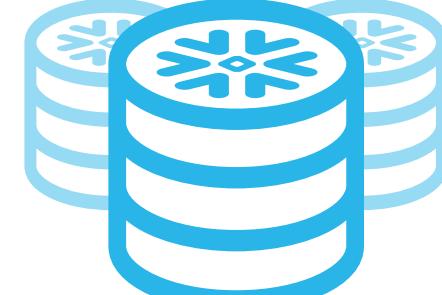
To ensure we can be a trustworthy partner and support our mission long-term, Posit **will remain independent and privately held.**







Snowflake Platform



AI ASSISTED EXPLORATORY DATA ANALYSIS





tidytuesday/data/2020/2020-04-07/readme.md

github.com/rfordatascience/tidytuesday/blob/main/data/2020/2020-04-07/readme.md

tidytuesday / data / 2020 / 2020-04-07 / readme.md

↑ Top

Files

main

Go to file

.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

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/2020/2020-04-07/tdf_winners.csv')

# Or read in with tidyTuesday package (https://github.com/dslc-io/tidyTuesday)
# PLEASE NOTE TO USE 2020 DATA YOU NEED TO USE tidyTuesday version ? from GitHub

# Either ISO-8601 date or year/week works!

# Install via pak:::pak("dslc-io/tidyTuesday")

tuesdata <- tidyTuesdayR::tt_load('2020-04-07')
tuesdata <- tidyTuesdayR::tt_load(2020, week = 15)

tdf_winners <- tuesdata$tdf_winners
```

POSIT_WORKBENCH_STAGING

app.snowflake.com/duloftf/posit_software_pbc_dev/#/apps/application/POSIT_WORKBENCH_STAGING/security/readme?isPrivate=true

Apps Posit Workbench Staging Administer Workbench DEFAULT_WH DEVELOPER :

Posit Workbench Staging
2.4 Upgraded 3 weeks ago • Active

Launch app ...

About the app Privileges App events Connections Compute Access management Cost management

Posit Workbench

Contact Listing

Installation

See the [installation instructions](#)

Administration

Posit Workbench on Snowflake is designed to be a hands-off managed service that will auto-upgrade as new features are released. Once Posit Workbench is installed and Snowflake OAuth has been configured, administrators will only need to manually add new users.

Key features:

- Users can use their preferred IDE: JupyterLab, Jupyter notebooks, RStudio, or VSCode
- Default restrictive egress not allowing any external traffic
- Operates within the Snowflake walled garden meaning all network, authentication, and authorization is controlled by your Snowflake account settings
- Integrated Snowflake OAuth which enables password-less usage of python and R ODBC drivers

Documentation

Documentation is separated into three sections:

- [Snowflake Workbench administration documentation](#) which covers installs, administration, troubleshooting, security, architecture, FAQ, and release notes



Posit Workbench x + fvloiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/s/57ea13c286bd33c286bd3/worksaces/

posit Workbench JAMES.POSIT.CO Sign Out

Sessions + New Session Projects

+ New Session Create a new session

New Session

Jupyter Notebook JupyterLab Positron Pro RStudio Pro VS Code

Session Name: Positron Pro Session

Session Credentials Edit Credentials

POSIT_SOFTWARE_PBC_DEV:DEVELOPER

Join session when ready Notify when ready Cancel Start Session

Posit Workbench 2025.08.0-daily+87.pro3, "Sandyland Bluebonnet" (34b0cdc4) © 2009 - 2025 Posit Software, PBC

12

Welcome — Positron

fvloliaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/s/0da296ae9dc1b78af5a9/

Welcome — Positron

SESSION CONNECTIONS HELP VIEWER

VARIABLES

No variables have been created.

Python 3.11.10 (Global)

Start

New Notebook New File New Console

Open

- Open File...
- Open Folder...
- New Folder...
- New Folder from Git...

Recent

snowflake-summit-2025 ~

CONSOLE TERMINAL PROBLEMS OUTPUT PORTS DEBUG CONSOLE

Python 3.11.10 (Global) started.
Python 3.11.10 (main, Apr 22 2025, 14:42:05) [GCC 11.4.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.12.3 -- An enhanced Interactive Python. Type '?' for help.
>>>

Plots

Python 3.11.10 (Global)
R 4.4.0

Posit Workbench 0 0 0 Quarto: 1.7.29 Layout: U.S. 13

The screenshot shows the Positron IDE interface. At the top, there's a browser-style header with a back/forward button, a refresh icon, and a search bar. Below it is the main application window with a title bar "Welcome — Positron". The left sidebar contains icons for file operations like new, open, and save, along with a search icon. The central area features the Positron logo and the text "an IDE for data science". Below this are sections for "Start" (with "New Notebook", "New File", and "New Console" buttons), "Open" (with options for files, folders, and Git), and "Recent" (listing a project named "snowflake-summit-2025"). The bottom half of the screen is a "CONSOLE" tab where a Python 3.11.10 session has just started, displaying standard startup messages. To the right of the console are panels for "SESSION" (showing "Python 3.11.10 (Global)" and "R 4.4.0"), "VARIABLES" (empty), and "PLOTS" (empty). The bottom navigation bar includes links for "Posit Workbench", "Layout: U.S.", and a page number "13".

exploration.qmd — snowflake

fvloiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/s/0da296ae9dc1b78af5a9/?folder=/home/JAMES.POSIT.CO/snowfl...

exploration.qmd — snowflake-summit-2025 — Positron

CHAT

Anthropic Claude

Filter to the most recent 6 years in stages then create the same plot grid

exploration.qmd:28-48

Positron Assistant

> Used 1 reference

Here's the code to filter to the most recent 6 years in the stages data and create the same plot grid:

```
# Get the stages data with elapsed and
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

# Fetch the data from Snowflake
stages_data = stages.select(["ELAPSED", "YEAR"]).execute()

# Get the most recent 6 years
unique_years = stages_data["YEAR"].unique()
unique_years.sort() # Sort years chronologically
recent_years = unique_years[-6:] # Get the last 6 years
n_years = len(recent_years)

# Calculate grid dimensions (trying to
# make it square)
n_cols = int(np.ceil(np.sqrt(n_years)))
```

CONSOLE TERMINAL PROBLEMS ... + - ×

~ /snowflake-summit-2025

```
... axes_flat[i].set_ylabel('Frequency')
...
... # Hide any unused subplots
... for j in range(i+1, len(axes_flat)):
...     axes_flat[j].set_visible(False)
...
... plt.tight_layout()
... plt.subplots_adjust(top=0.9) # Make room for the suptitle
>>>
```

SESSION CONNECTIONS HELP VIEWER

VARIABLES

Python 3.11.10 (Venv: .venv)

DATA

- > stages_data [255752 rows x 2 columns] pandas.DataFrame
- > year_data [3532 rows x 2 columns] pandas.DataFrame

VALUES

- > auth <posit.connect.external.snowflake.PositAuthenticat... PositAuthenticator
- > axes [[<Axes: title={'center': 'Year: 2014.0'}, xlabel='Ela... numpy.object (2, 3)
- > axes_flat [<Axes: title={'center': 'Year: 2014.0'}, xlabel='Elapse... numpy.object (6)

PLOTS

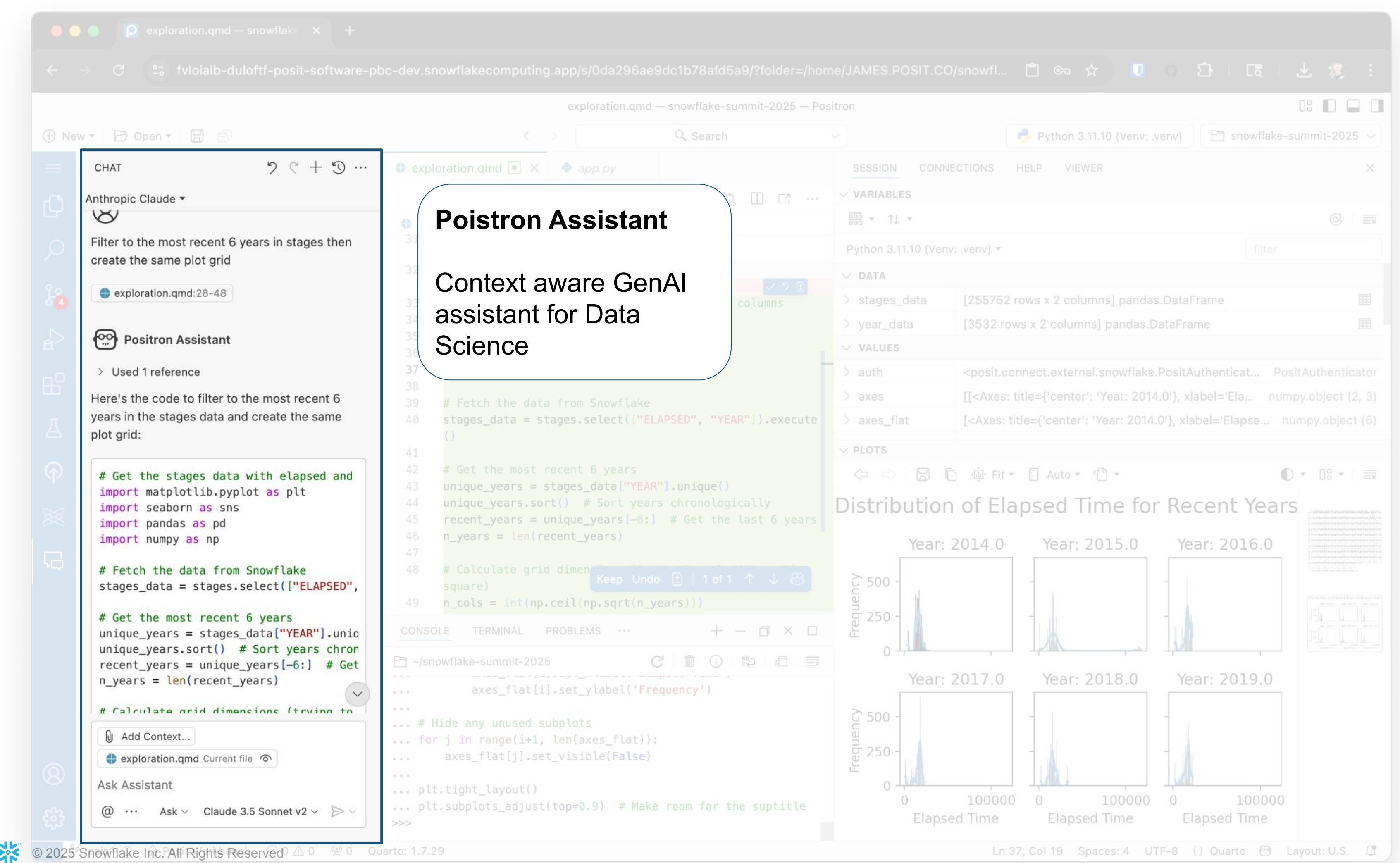
Distribution of Elapsed Time for Recent Years

Year: 2014.0 Year: 2015.0 Year: 2016.0

Year: 2017.0 Year: 2018.0 Year: 2019.0

Elapsed Time

Ln 37, Col 19 Spaces: 4 UTF-8 {} Quarto Layout: U.S.



exploration.qmd — snowflake

fvloiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/s/0da296ae9dc1b78af5a9/?folder=/home/JAMES.POSIT.CO/snowfl...

exploration.qmd — snowflake-summit-2025 — Positron

CHAT

Anthropic Claude

Filter to the most recent 6 years in stages then create the same plot grid

exploration.qmd:28-48

Positron Assistant

Used 1 reference

Here's the code to filter to the most recent 6 years in the stages data and create the same plot grid:

```
# Get the stages data with elapsed and
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

# Fetch the data from Snowflake
stages_data = stages.select(["ELAPSED", "YEAR"]).execute()

# Get the most recent 6 years
unique_years = stages_data["YEAR"].unique()
unique_years.sort() # Sort years chronologically
recent_years = unique_years[-6:] # Get the last 6 years
n_years = len(recent_years)

# Calculate grid dimensions (trying to
# make it square)
n_cols = int(np.ceil(np.sqrt(n_years)))
```

CONSOLE TERMINAL PROBLEMS ...

Add Context... exploration.qmd Current file

Ask Assistant

@ ... Ask Claude 3.5 Sonnet v2 > >

Code Editor

Full featured code editor

SESSION CONNECTIONS HELP VIEWER

VARIABLES

Python 3.11.10 (Venv: .venv)

filter

DATA

> stages_data [255752 rows x 2 columns] pandas.DataFrame

> year_data [3532 rows x 2 columns] pandas.DataFrame

VALUES

> auth <posit.connect.external.snowflake.PositAuthenticat... PositAuthenticator

> axes [[<Axes: title={'center': 'Year: 2014.0'}, xlabel='Ela... numpy.object (2, 3)

> axes_flat [<Axes: title={'center': 'Year: 2014.0'}, xlabel='Elapse... numpy.object (6)

PLOTS

Distribution of Elapsed Time for Recent Years

Year: 2014.0 Year: 2015.0 Year: 2016.0

Year: 2017.0 Year: 2018.0 Year: 2019.0

Frequency

Elapsed Time

Ln 37, Col 19 Spaces: 4 UTF-8 {} Quarto Layout: U.S.

exploration.qmd — snowflake

fvloiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/s/0da296ae9dc1b78af5a9/?folder=/home/JAMES.POSIT.CO/snowfl...

exploration.qmd — snowflake-summit-2025 — Positron

CHAT

Anthropic Claude

Filter to the most recent 6 years in stages then create the same plot grid

exploration.qmd:28-48

Positron Assistant

Used 1 reference

Here's the code to filter to the most recent 6 years in the stages data and create the same plot grid:

```
# Get the stages data with elapsed and
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

# Fetch the data from Snowflake
stages_data = stages.select(["ELAPSED",

# Get the most recent 6 years
unique_years = stages_data["YEAR"].unique()
unique_years.sort() # Sort years chronologically
recent_years = unique_years[-6:] # Get the last 6 years
n_years = len(recent_years)

# Calculate grid dimensions (trying to
# make it square)
n_cols = int(np.ceil(np.sqrt(n_years)))
```

CONSOLE TERMINAL PROBLEMS ... + - □ ×

~ /snowflake-summit-2025

```
... axes_flat[i].set_ylabel('Frequency')
...
... # Hide any unused subplots
... for j in range(i+1, len(axes_flat)):
...     axes_flat[j].set_visible(False)
...
... plt.tight_layout()
... plt.subplots_adjust(top=0.9) # Make room for the suptitle
>>>
```

SESSION CONNECTIONS HELP VIEWER

VARIABLES

Python 3.11.10 (Venv: .venv)

DATA

> stages_data [255752 rows x 2 columns] pandas.DataFrame

> year_data [3532 rows x 2 columns] pandas.DataFrame

VALUES

> auth <posit.connect.external.snowflake.PositAuthenticat... PositAuthenticator

> axes [[<Axes: title={'center': 'Year: 2014.0'}, xlabel='Ela... numpy.object (2, 3)

> axes_flat [<Axes: title={'center': 'Year: 2014.0'}, xlabel='Elapse... numpy.object (6)

PLOTS

Distribution of Elapsed Time for Recent Years

Year: 2014.0 Year: 2015.0 Year: 2016.0

Year: 2017.0 Year: 2018.0 Year: 2019.0

Elapsed Time

Ln 37, Col 19 Spaces: 4 UTF-8 { } Quarto Layout: U.S.

exploration.qmd — snowflake

fvloiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/s/0da296ae9dc1b78af5a9/?folder=/home/JAMES.POSIT.CO/snowfl...

exploration.qmd — snowflake-summit-2025 — Positron

CHAT

Anthropic Claude

Filter to the most recent 6 years in stages then create the same plot grid

exploration.qmd:28-48

Positron Assistant

Used 1 reference

Here's the code to filter to the most recent 6 years in the stages data and create the same plot grid:

```
# Get the stages data with elapsed and
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

# Fetch the data from Snowflake
stages_data = stages.select(["ELAPSED", "YEAR"])

# Get the most recent 6 years
unique_years = stages_data["YEAR"].unique()
unique_years.sort() # Sort years chronologically
recent_years = unique_years[-6:] # Get the last 6 years
n_years = len(recent_years)

# Calculate grid dimensions (trying to
# make it square)
n_cols = int(np.ceil(np.sqrt(n_years)))

CONSOLE TERMINAL PROBLEMS ...
... axes_flat[i].set_ylabel('Frequency')
...
... # Hide any unused subplots
... for j in range(i+1, len(axes_flat)):
...     axes_flat[j].set_visible(False)
...
... plt.tight_layout()
... plt.subplots_adjust(top=0.9) # Make room for the suptitle
>>>
```

SESSION CONNECTIONS HELP VIEWER

VARIABLES

Python 3.11.10 (Venv)

DATA

- > stages_data
- > year_data
- VALUES
- > auth
- > axes
- > axes_flat

PLOTS

Distribution of Elapsed Time for Recent Years

Year: 2014.0 Year: 2015.0 Year: 2016.0

Year: 2017.0 Year: 2018.0 Year: 2019.0

Elapsed Time

Frequency

PositAuthenticator

numpy.object (2, 3)

numpy.object (6)

Plots

Plots render directly within the IDE



exploration.qmd — snowflake

fvloiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/s/0da296ae9dc1b78af5a9/?folder=/home/JAMES.POSIT.CO/snowfl...

exploration.qmd — snowflake-summit-2025 — Positron

CHAT

Anthropic Claude

Filter to the most recent 6 years in stages then create the same plot grid

exploration.qmd:28-48

Positron Assistant

> Used 1 reference

Here's the code to filter to the most recent 6 years in the stages data and create the same plot grid:

```
# Get the stages data with elapsed and
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

# Fetch the data from Snowflake
stages_data = stages.select(["ELAPSED", "YEAR"]).execute()

# Get the most recent 6 years
unique_years = stages_data["YEAR"].unique()
unique_years.sort() # Sort years chronologically
recent_years = unique_years[-6:] # Get the last 6 years
n_years = len(recent_years)

# Calculate grid dimensions (trying to
# make it square)
n_cols = int(np.ceil(np.sqrt(n_years)))
```

CONSOLE TERMINAL PROBLEMS ... + - □ ×

~ /snowflake-summit-2025

```
... axes_flat[i].set_ylabel('Frequency')
...
... # Hide any unused subplots
... for j in range(i+1, len(axes_flat)):
...     axes_flat[j].set_visible(False)
...
... plt.tight_layout()
... plt.subplots_adjust(top=0.9) # Make room for the suptitle
>>>
```

SESSION CONNECTIONS HELP VIEWER

VARIABLES

Python 3.11.10 (Venv: .venv)

DATA

> stages_data	[255752 rows x 2 columns] pandas.DataFrame
> year_data	[3532 rows x 2 columns] pandas.DataFrame

VALUES

> auth	<posit.connect.external.snowflake.PositAuthenticat... PositAuthenticator
> axes	[[<Axes: title={'center': 'Year: 2014.0'}, xlabel='Ela... numpy.object (2, 3)
> axes_flat	[<Axes: title={'center': 'Year: 2014.0'}, xlabel='Elapse... numpy.object (6)

PLOTS

Distribution of Elapsed Time for Recent Years

Year: 2014.0 Year: 2015.0 Year: 2016.0

Environment

See environment details. The Viewer pane provides direct access to running applications.

exploration.qmd — snowflake

fvloiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/s/0da296ae9dc1b78af5a9/?folder=/home/JAMES.POSIT.CO/snowfl...

exploration.qmd — snowflake-summit-2025 — Positron

CHAT

Anthropic Claude

Filter to the most recent 6 years in stages then create the same plot grid

exploration.qmd:28-48

Positron Assistant

Used 1 reference

Here's the code to filter to the most recent 6 years in the stages data and create the same plot grid:

```
# Get the stages data with elapsed and
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

# Fetch the data from Snowflake
stages_data = stages.select(["ELAPSED",

# Get the most recent 6 years
unique_years = stages_data["YEAR"].unique()
unique_years.sort() # Sort years chron
recent_years = unique_years[-6:] # Get
n_years = len(recent_years)

# Calculate grid dimensions (truncating to
n_cols = int(np.floor(np.sqrt(n_years)))
n_rows = int(np.ceil(np.sqrt(n_years)))
```

CONSOLE TERMINAL PROBLEMS ...

exploration.qmd Current file

Add Context...

Ask Assistant

@ ... Ask Claude 3.5 Sonnet v2 > >

SESSION CONNECTIONS HELP VIEWER

VARIABLES

Python 3.11.10 (Venv: .venv)

filter

Distribution of Elapsed Time for Recent Years

Year: 2014.0 Year: 2015.0 Year: 2016.0

Year: 2017.0 Year: 2018.0 Year: 2019.0

Elapsed Time

Frequency

Ln 37, Col 19 Spaces: 4 UTF-8 {} Quarto Layout: U.S.

20

APP DEVELOPMENT



© 2025 Snowflake Inc. All Rights Reserved

Data: stages.csv — snowflake

fvloiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/s/0da296ae9dc1b78af5a9/?folder=/home/JAMES.POSIT.CO/snowflake-s...

Data: stages.csv — snowflake-summit-2025 — Positron

Python 3.11.10 (Venv: .venv)

File New Open Data: stages.csv app.py

Search

Clear Column Sorting Open as Plain Text File

edition year start_date stage_results_id rank time rider age

	edition BIGINT	year BIGINT	start_date DATE	stage_results_id VARCHAR	rank VARCHAR	time VARCHAR	rider VARCHAR	age VARC
0	1	1903	1903-07-01	stage-1	1	63913	Garin Maurice	32
1	1	1903	1903-07-01	stage-1	2	55	Pagie Émile	32
2	1	1903	1903-07-01	stage-1	3	2099	Georget Léon	23
3	1	1903	1903-07-01	stage-1	4	3768	Augereau Fernand	20
4	1	1903	1903-07-01	stage-1	5	3893	Fischer Jean	36
5	1	1903	1903-07-01	stage-1	6	6173	Kerff Marcel	37
6	1	1903	1903-07-01	stage-1	7	6535	Catteau Aloïs	25
7	1	1903	1903-07-01	stage-1	8	6587	Pivin Ernest	33
8	1	1903	1903-07-01	stage-1	9	7695	Habets Léon	NA
9	1	1903	1903-07-01	stage-1	10	7706	Baugendre François	22
10	1	1903	1903-07-01	stage-1	11	10669	Muller Rodolfo	26
11	1	1903	1903-07-01	stage-1	12	10669	Gauban Jean	28
12	1	1903	1903-07-01	stage-1	13	10669	Jaeck Anton	21
13	1	1903	1903-07-01	stage-1	14	10669	Pothier Lucien	20
14	1	1903	1903-07-01	stage-1	15	10669	Lequatre Marcel	20
15	1	1903	1903-07-01	stage-1	16	10669	Pasquier Gustave	25
16	1	1903	1903-07-01	stage-1	17	10976	Laeser Charles	23
17	1	1903	1903-07-01	stage-1	18	11202	Lootens Julien	26
18	1	1903	1903-07-01	stage-1	19	12221	Fischer Josef	38
19	1	1903	1903-07-01	stage-1	20	17956	Salais René	28
20	1	1903	1903-07-01	stage-1	21	17956	Borot Georges	NA
21	1	1903	1903-07-01	stage-1	22	18356	Lechartier Isidore	NA

255,752 rows 13 columns

CONSOLE TERMINAL PROBLEMS OUTPUT PORTS DEBUG CONSOLE

main* Posit Workbench 0 △ 0 0 Quarto: 1.7.29

Layout: U.S.

The screenshot shows the Posit Workbench interface. The main area displays the code for 'app.py' which imports various libraries like pandas, plotly, and matplotlib, and sets up a Shiny application. A context menu is open over the code editor, showing options for running the app or debugging it. The top bar shows the session name as 'app.py — snowflake-summit-2025 — Positron'. The left sidebar contains icons for file operations, search, and other tools. The bottom navigation bar includes tabs for CONSOLE, TERMINAL, PROBLEMS, OUTPUT, PORTS, and DEBUG CONSOLE, along with a status bar at the bottom.

app.py — snowflake-summit-2025 — Positron

SESSION CONNECTIONS HELP VIEWER

https://fvloiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/s/0da296ae9dc1b78af5a9/?folder=/home/JAMES.POSIT.CO/snowflake-summit-2025

Python 3.11.10 (Venv: .venv) snowflake-summit-2025

app.py M exploration.qmd U

dashboard > app.py > server > snowflake_claude

```

1 import pandas as pd
2 import plotly.express as px
3 import plotly.graph_objects as go
4 from pathlib import Path
5 import os
6 from shiny import App, ui, reactive
7 from shinywidgets import output_widget, render_wi
8 import matplotlib.pyplot as plt
9 import matplotlib
10 import chatlas
11 import querychat
12 import re
13 from scipy import stats
14 import numpy as np
15 from sqlalchemy import create_engine
16 from querychat.datasource import SQLAlchemySource
17 from snowflake.sqlalchemy import URL
18 from posit.connect import Client
19 from posit.connect.external.snowflake import Posi
20 import snowflake.connector
21
22
23 matplotlib.use("Agg") # Use Agg backend for matpl
24

```

CONSOLE TERMINAL ... + v - x □ bash Shiny

tatefulset-0:~/snowflake-summit-2025\$ /home/JAMES.POSIT.CO/snowflake-summit-2025/.venv/bin/python -m shiny run --port 41341 --reload --autoreload-po ...
b/bootstrap-5.3.1/fonts/HI_SiYsKILxRpg3hIP6sJ7fM7PqlPevW.woff2 HTTP/1.1" 200 OK
INFO: 185.229.59.113:0 - "GET /lib/bootstrap-5.3.1/fonts/HI_QiYsKILxRpg3hIP6sJ7fM7Pql0NvUlMI.woff2 HTTP/1.1" 200 OK

Search

Tour de France Analysis

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

Filter to only the most recent decade

```

-- Get the most recent year in :
WITH max_year AS (
    SELECT MAX(year) as latest_
    FROM stages
)
-- Select all records from the
SELECT *
FROM stages
WHERE year > (SELECT latest_

```

I've filtered the dashboard to show only stages from the most recent decade in the dataset (2010-2019).

Enter a message...

Overview

Most Stage Wins

Rider	Wins
Cavendish Mark	20
Kittel Marcel	14
Sagan Peter	12
Greipel André	11
Froome Chris	8

Most Stages Completed

Rider	Completed
Erviti Imanol	197
Moinard Amaël	189
Rolland Pierre	189
Burghardt Marcus	189
Chavanel Sylvain	184

Stage Time Distribution

Rider	Stage Time (min)
Cavendish Mark	20
Kittel Marcel	14
Sagan Peter	12
Greipel André	11
Froome Chris	8

SESSION CONNECTIONS HELP VIEWER

Python 3.11.10 (Venv: .venv) snowflake-summit-2025

Quarto: 1.7.29

Not Committed Yet Ln 108, Col 50 Spaces: 4 UTF-8 LF {} Python Layout: U.S.

24

app.py — snowflake-summit-2025 — Positron

SESSION CONNECTIONS HELP VIEWER

https://fvloiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/s/0da296ae9dc1b78af5a9/?folder=/home/JAMES.POSIT.CO/snowflake-summit-2025

Tour de France Analysis

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

Filter to only the most recent decade

```
-- Get the most recent year in the dataset
WITH max_year AS (
    SELECT MAX(year) as latest_year
    FROM stages
)
-- Select all records from the latest decade
SELECT *
FROM stages
WHERE year > (SELECT latest_year - 10)
```

I've filtered the dashboard to show only stages from the most recent decade in the dataset (2010-2019).

Enter a message...

SESSION CONNECTIONS HELP VIEWER

Python 3.11.10 (Venv: .venv) snowflake-summit-2025

Overview

Most Stage Wins

Rider	Wins
Cavendish Mark	20
Kittel Marcel	14
Sagan Peter	12
Greipel André	11
Froome Chris	8

Most Stages Completed

Rider	Stages
Erviti Imanol	197
Moinard Amaël	189
Rolland Pierre	189
Burghardt Marcus	189
Chavanel Sylvain	184

Stage Time Distribution

CONSOLE TERMINAL ...

```
tatefulset-0:~/snowflake-summit-2025$ /home/JAMES.POSIT.CO/snowflake-summit-2025/.venv/bin/python -m shiny run --port 41341 --reload --autoreload-port 41341
...
b/bootstrap-5.3.1/fonts/HI_SiYsKILxRpg3hIP6sJ7fM7PqlPevW.woff2 HTTP/1.1" 200 OK
INFO: 185.229.59.113:0 - "GET /lib/bootstrap-5.3.1/fonts/HI_QiYsKILxRpg3hIP6sJ7fM7Pql0NvUlMI.woff2 HTTP/1.1" 200 OK
Quarto: 1.7.29
```

© 2025 Snowflake Inc. All Rights Reserved

Not Committed Yet Ln 108, Col 50 Spaces: 4 UTF-8 LF Python Layout: U.S.

25

Query - 01bcc353-010d-86b2-0005-004303e5eb86

app.snowflake.com/duloftf/posit_software_pbc_dev/#/compute/history/queries/01bcc353-010d-86b2-0005-004303e5eb86/detail

Query - 01bcc353-010d-86b2-0005-004303e5eb86

JAMES@POSIT.CO

Query Details [Query Profile](#) [Query Telemetry](#)

Status	Duration	Driver Status
Success	111ms	Supported
Start Time	Query ID	Client Driver
6/2/2025, 8:47:16 AM	01bcc353-010d-86b2-0005-004303e5...	PythonConnector 3.15.0 ⓘ
End Time	Query Tag	Session ID
6/2/2025, 8:47:16 AM	—	1407662718423674 ⓘ
Warehouse Size	Attributed Compute Credits ⓘ	Parameterized query hash
—	—	2e5c0297dd5b8598ba5114491a8056b3 ⓘ

SQL Text

```
WITH max_year AS (
    SELECT MAX(year) as latest_year
    FROM stages
)
-- Select all records from the last 10 years
SELECT *
FROM stages
WHERE year > (SELECT latest_year - 10 FROM max_year)
```

JB James Blair DEVELOPER

posit-dev/querychat

github.com/posit-dev/querychat

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.

Restaurant tipping

You can use this sidebar to filter and sort the data based on the columns available in the tips table. Here are some examples of the kinds of questions you can ask me:

1. Filtering: "Show only Male smokers who had Dinner on Saturday."
2. Sorting: "Show all data sorted by total_bill in descending order."
3. Answer questions about the data: "How do tip sizes compare between lunch and dinner?"

Tips on Weekends with Tip Size >\$1

```
SELECT * FROM tips WHERE day IN ('Sat', 'Sun') AND tip > 1
```

Total tippers	Average tips	Average bill
160	15.9%	\$21.14

Tips data

total_bill	tip	sex	smoker	day
20.65	3.25	Male	No	Sat

Total bill vs tip

No packages published

Contributors 4

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

Languages

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

app.py — snowflake-summit-2025 — Positron

SESSION CONNECTIONS HELP VIEWER

https://fvloiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/s/0da296ae9dc1b78af5a9/?folder=/home/JAMES.POSIT.CO/snowflake-summit-20...

Python 3.11.10 (Venv: .venv) snowflake-summit-2025

app.py M exploration.qmd U

dashboard > app.py > server > snowflake_claude

```

86 def server(input, output, session):
87
88     snowflake_connection = querychat.datasource.SQLAlchemySource(
89         create_engine("snowflake://not@used/db", creator=get_connection),
90         "stages",
91     )
92
93
94     # create chatlas connection
95     def snowflake_claude(system_prompt: str) -> chatlas.Chat:
96         return chatlas.ChatSnowflake(
97             model="claude-3-5-sonnet",
98             connection_name = "workbench",
99             system_prompt=system_prompt,
100            account="duloftf-posit-software-pbc-dev"
101
102
103         )
104
105
106
107
108
109
110
111     querychat_config = querychat.init(
112         # Snowflake connection
113         snowflake_connection,
114         greeting=greeting,
115         data_description=data_desc,
116         create_chat_callback=snowflake_claude,
117     )
118

```

CONSOLE TERMINAL PROBLEMS OUTPUT PORTS ...

snowflake-summit-2025 JAMES.POSIT.CO@statefulset-0:~/snowflake-summit-2025\$ /home/JAMES.POSIT.CO/snowflake-summit-2025/.venv/bin/python -m shiny run --port 41341 --reload --autoreload-port 35205 /home/JAMES.POSIT.CO/snowflake-summit-2025/dashboard/app.py

INFO: 185.229.59.113:0 - "GET /lib/bootstrap-5.3.1/fonts/HI_SiYsKILxRpg3hIP6sJ7fM7PqlPevW.woff2 HTTP/1.1" 200 OK

INFO: 185.229.59.113:0 - "GET /lib/bootstrap-5.3.1/fonts/HI_QiYsKILxRpg3hIP6sJ7fM7Pql0NvULMI.woff2 HTTP/1.1" 200 OK

SEARCH

Tour de France Analysis

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

Filter to only the most recent decade

```
-- Get the most recent year in the dataset
WITH max_year AS (
    SELECT MAX(year) as latest_year
    FROM stages
)
-- Select all records from the latest year
SELECT *
FROM stages
WHERE year > (SELECT latest_year)
```

I've filtered the dashboard to show only stages from the most recent decade in the dataset (2010-2019).

Enter a message...

Overview

Most Stage Wins

Rider	Wins
Mark Cavendish	20
Alberto Contador	14
Chris Froome	12
Tom Boonen	11
Andrea Belotti	8
Others	189
Total	197

Most Stages Completed

Rider	Completed Stages
Mark Cavendish	20
Alberto Contador	14
Chris Froome	12
Tom Boonen	11
Andrea Belotti	8
Others	189
Total	197

Stage Time Distribution



Snowpark
Container
Services

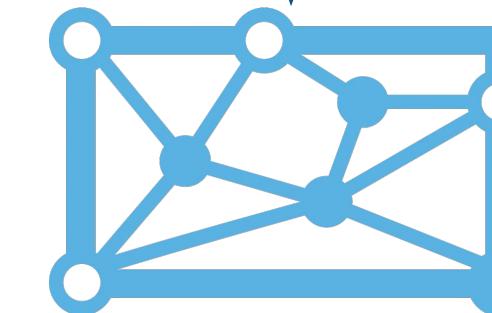


Shiny App

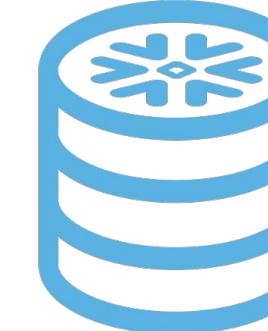


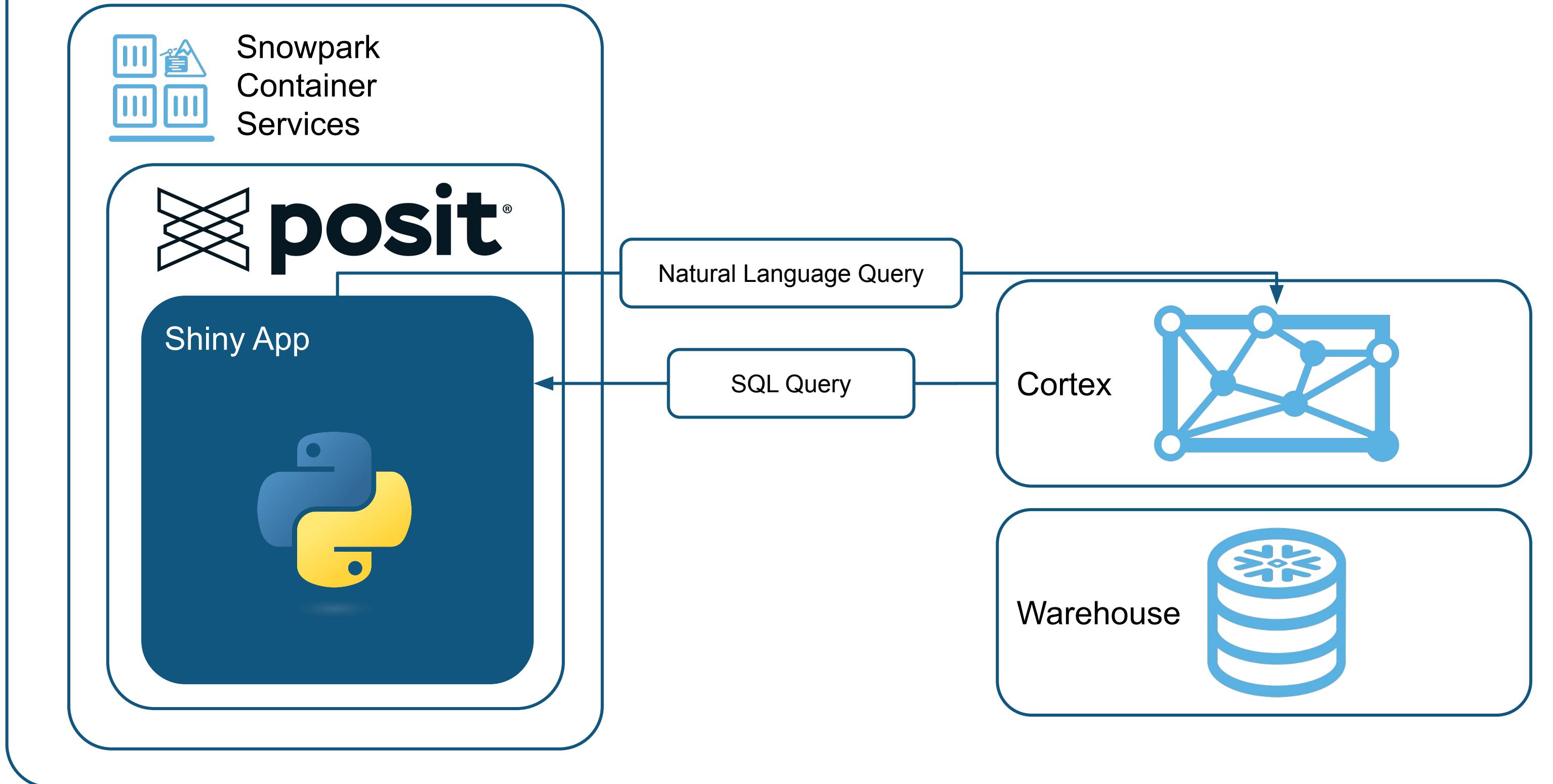
Natural Language Query

Cortex



Warehouse







Snowpark
Container
Services



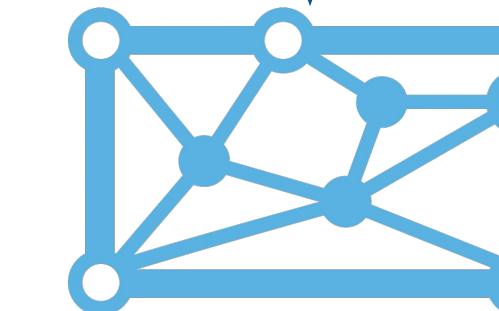
Shiny App



Natural Language Query

SQL Query

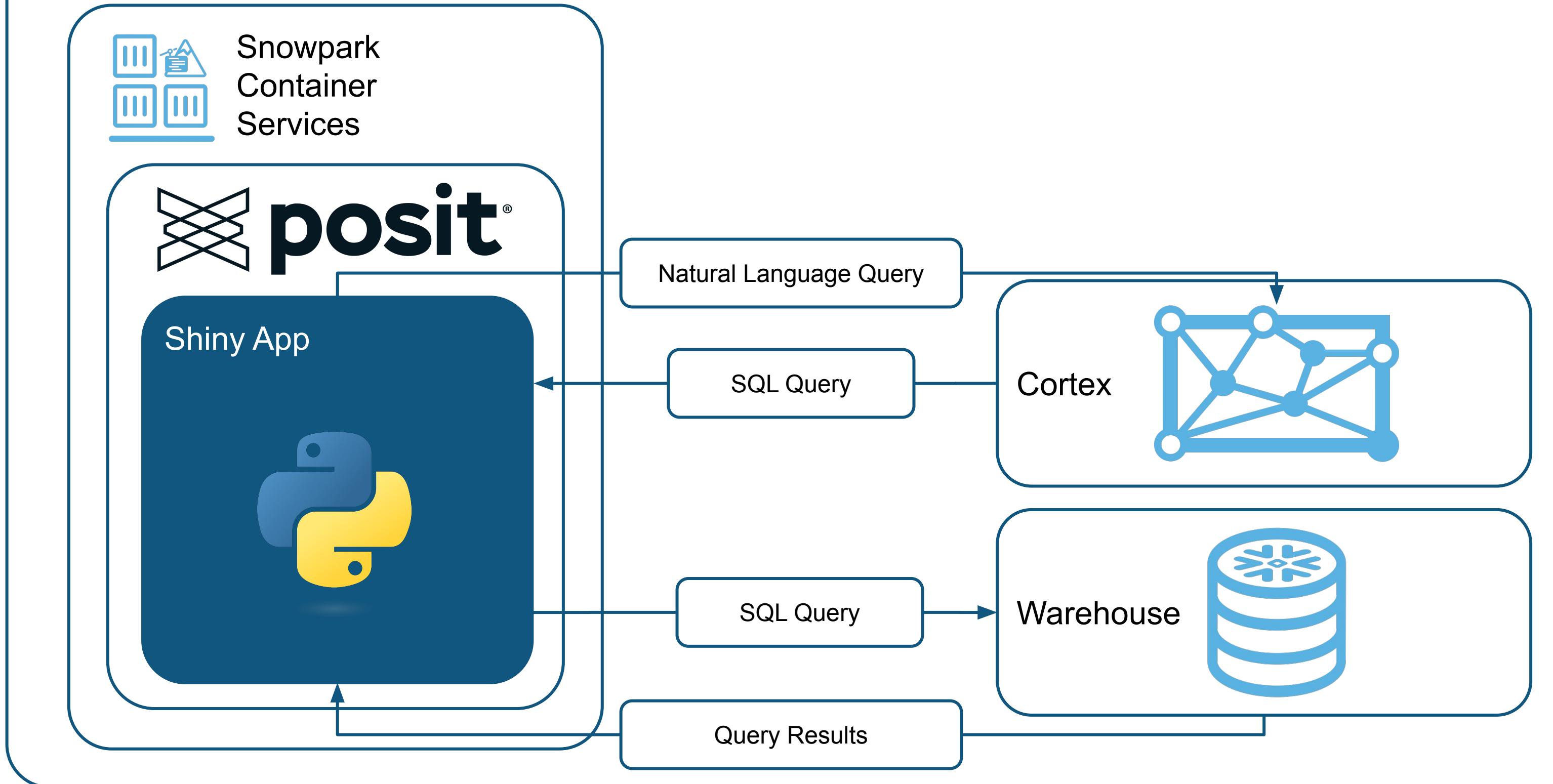
Cortex



SQL Query

Warehouse





APP DEPLOYMENT



TdF Dashboard-0C5S.toml — +

fvloiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/s/0da296ae9dc1bee13960a/?folder=/home/JAMES.POSIT.CO/snowflake-sum... ☁ ☆

TdF Dashboard-0C5S.toml — snowflake-summit-2025 — Positron

+ New | Open | 📂 | 📄 | Search | Python 3.11.10 (Venv: .venv) | snowflake-summit-2025

SESSION CONNECTIONS HELP VIEWER X

☰ POSIT PUBLISHER: HOME DEPLOYMENT TdF Dashboard + ⚙ ...

PTD Demo dashboard/app.py Deploy Your Project

Not Yet Deployed Is this already deployed to a Connect server? You can update that previous deployment.

PROJECT FILES .posit app.R app.py data data_description.md greeting.md requirements.txt

SECRETS

PYTHON PACKAGES R PACKAGES CREDENTIALS HELP AND FEEDBACK

TdF Dashboard-0C5S.toml exploration.qmd

Variables

No variables have been created.

Plots

CONSOLE TERMINAL PROBLEMS OUTPUT ... bash + ⚙ ... - ⚙ X ⚙

```
1 # Configuration file generated by Posit Publisher.
2 # Please review and modify as needed. See the documentation for more options:
3 # https://github.com/posit-dev/publisher/blob/main/docs/configuration.md
4 '$schema' = 'https://cdn.posit.co/publisher/schemas/posit-publishing-schema-v3.json'
5 type = 'python-shiny'
6 entrypoint = 'app.py'
7 validate = true
8 files =
9   '/app.py',
10  '/requirements.txt',
11  './.posit/publish/TdF Dashboard-0C5S.toml',
12  './.posit/publish/deployments/deployment-G9TR.toml',
13  '/data_description.md',
14  '/greeting.md'
15 ]
16 title = 'TdF Dashboard'
17
18 [python]
19
```

```
snowflake-summit-2025 JAMES.POSIT.CO@statefulset-0:~/snowflak
snowflake-summit-2025 JAMES.POSIT.CO@statefulset-0:~/snowflak
o snowflake-summit-2025 JAMES.POSIT.CO@statefulset-0:~/snowflak
snowflake-summit-2025 JAMES.POSIT.CO@statefulset-0:~/snowflake-summit-2
snowflake-summit-2025 JAMES.POSIT.CO@statefulset-0:~/snowflake-summit-2025$
snowflake-summit-2025 JAMES.POSIT.CO@statefulset-0:~/snowflake-summit-2025$
```

main* 🌐 Posit Workbench ☁ 0 △ 0 ⚙ 0 Quarto: 1.7.29 Ln 1, Col 51 Spaces: 2 UTF-8 LF Plain Text ☁ Layout: U.S. 34

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?

Show only riders who have ever won a stage



```
WITH stage_winners AS (
  SELECT DISTINCT rider
  FROM stages
  WHERE rank = 1
)
SELECT *
FROM stages
WHERE rider IN (SELECT rider FR
```

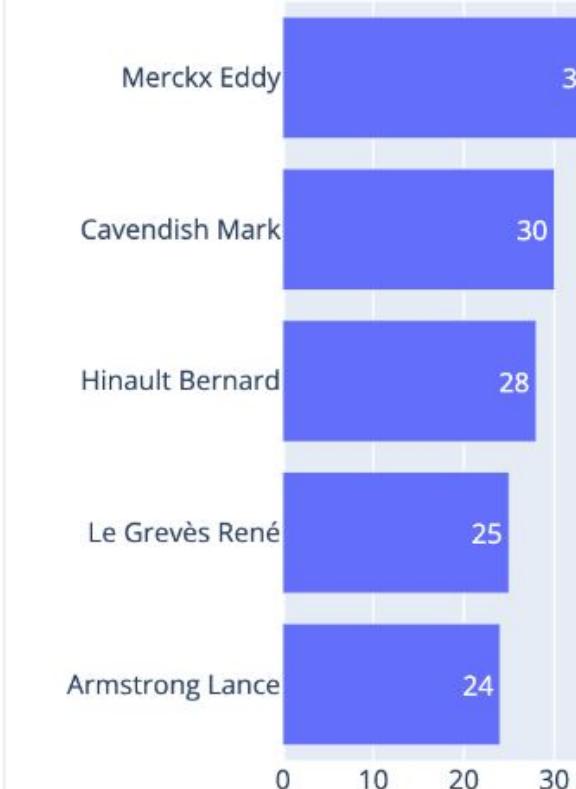
I've filtered the dashboard to show only riders who have won at least one stage in the Tour de France (those who achieved a rank of 1 in any stage).

Enter a message...

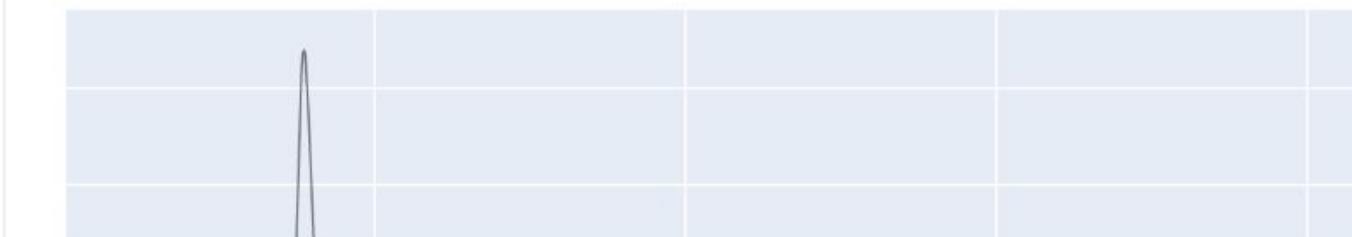


Overview

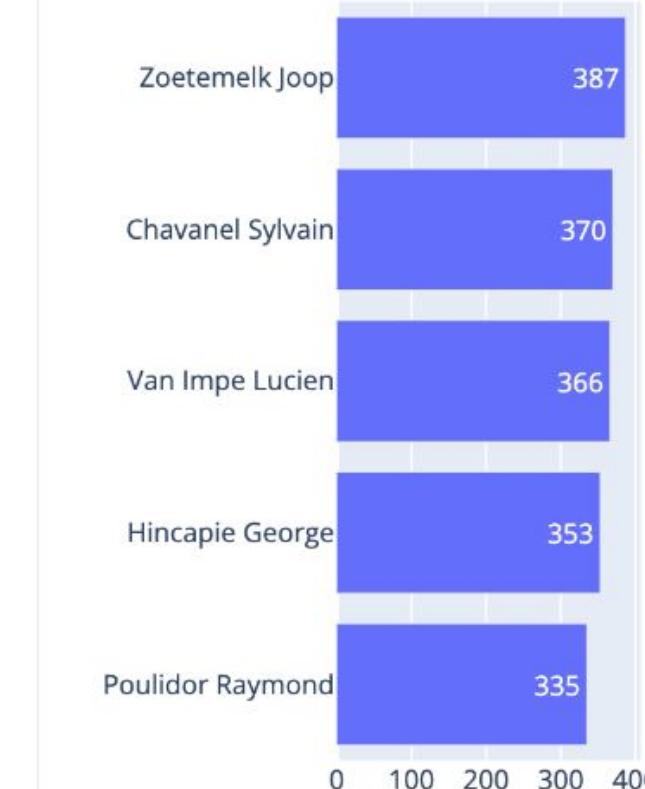
Most Stage Wins



Stage Time Distribution



Most Stages Completed



SHARING

 All users - login required

 Specific users or groups


Who can view or change this application



NICK.PELIKAN@PO...

NICK.PELIKAN@PO...

[Remove All Groups and Users](#)

INTEGRATIONS

Snowflake

[Log out](#)

CONTENT URL

Path

[Customize](#)

URL

<https://fguoiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app/content/a3f58475-8388-499e-819d-ccc75e364941>
[Copy](#)




doped-cyclist-filter - Snowflake

app.snowflake.com/duloftf/posit_software_pbc_dev/w1unx7OR5oIQ#query

Lending Club Air Quality View 2025-05-09 4:33pm 2025-05-28 9:55am doped-cyclist-filter + ▾

DEVELOPER · DEFAULT_WH (Small) Share

DEMOS.PUBLIC Settings

Open in Workspaces Code Versions

```
1 CREATE OR REPLACE ROW ACCESS POLICY doping_filter
2     AS (rider varchar) RETURNS BOOLEAN ->
3         CASE
4             WHEN current_user() IN ('DEMO_SNOWFLAKE_USER@POSIT.CO') THEN rider NOT IN (
5                 -- This list was generated by asking Claude for a list of known doping cyclists
6                 'Armstrong Lance',
7                 'Landis Floyd',
8                 'Hamilton Tyler',
9                 'Ullrich Jan',
10                'Pantani Marco',
11                'Contador Alberto',
12                'Valverde Alejandro',
13                'Vinokourov Alexandre',
14                'Basso Ivan',
15                'Rasmussen Michael',
16                'Schleck Frank',
17                'Kreuziger Roman',
18                'Rebellin Davide',
19                'Di Luca Danilo',
20                'Ricco Riccardo',
21                'Petacchi Alessandro',
22                'Scarponi Michele',
23                'Sanchez Samuel',
24                'Cobo Juan Jose',
```

Results Chart

Query Details

Query duration 164ms

Rows

Ask Copilot

Query ID 01bcaddr-010a-85c5-0...

JB

Re-run your query. Query results are cached for 24 hours.

Content / Tour De France Data Analysis

Tour de France Analysis

How many stages did Lance Armstrong win?

Let me query for stages won by Lance Armstrong.

```
SELECT COUNT(*) as stage_wins
FROM stages
WHERE rider = 'Armstrong Lance'
AND rank = 1
```

stage_wins
0

According to the data, Lance Armstrong has 0 stage wins recorded in this dataset. This is because Armstrong's Tour de France victories and stage wins from 1999-2005 were officially stripped from the record books due to his doping admission. The official Tour de France records have been amended to remove his results, which is reflected in this dataset.

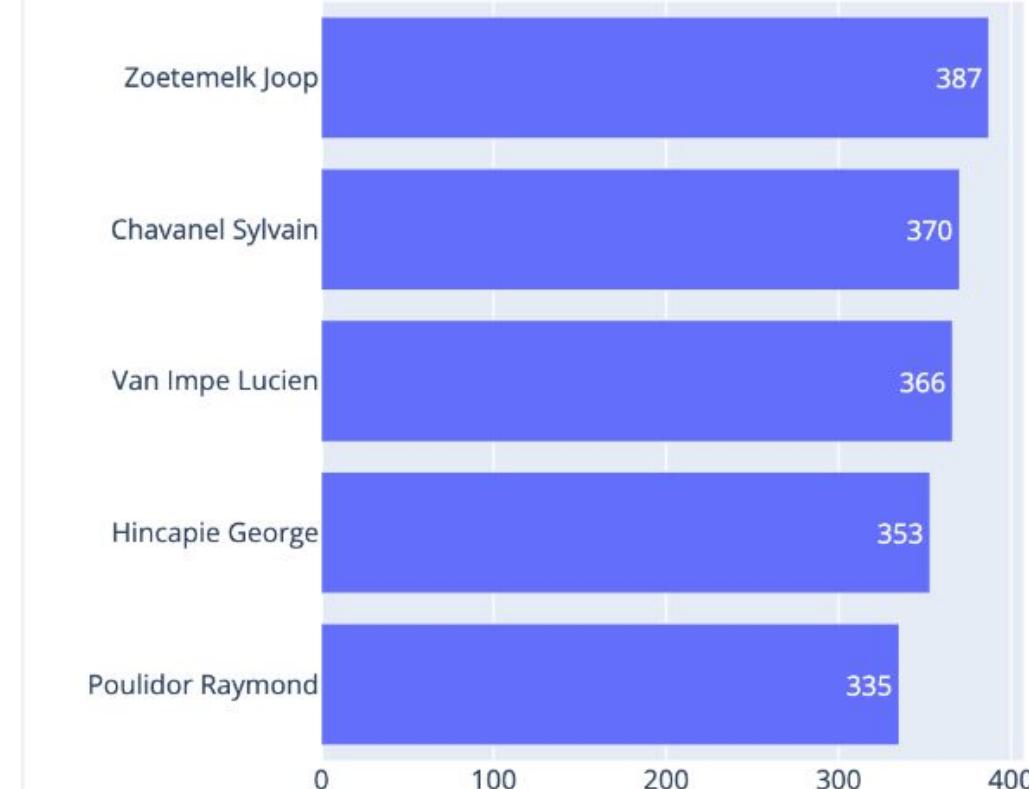
Enter a message...

Overview

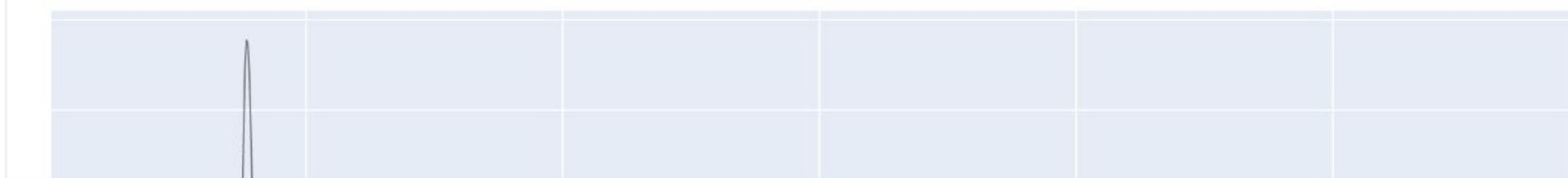
Most Stage Wins



Most Stages Completed



Stage Time Distribution



CONCLUSION

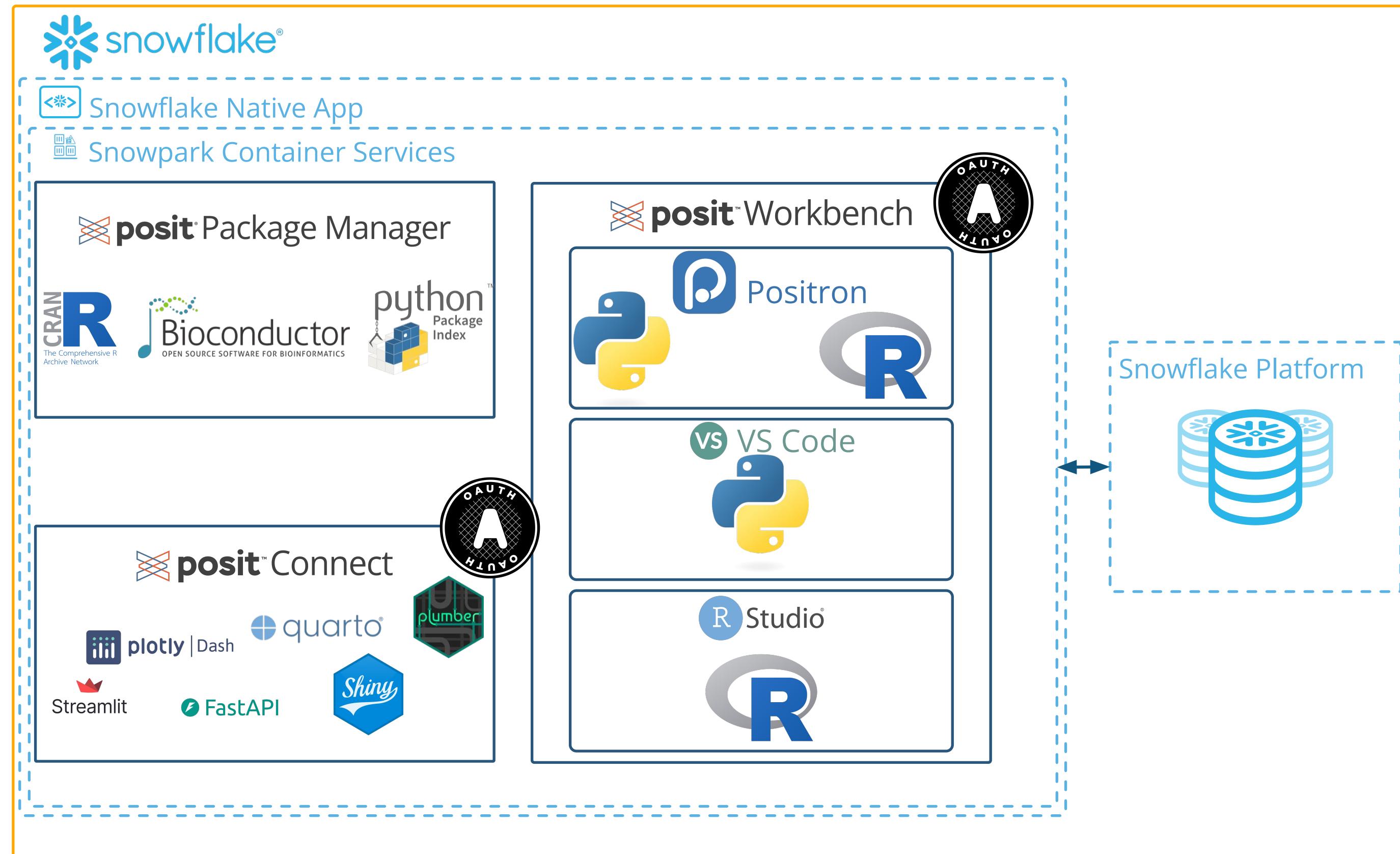


In Review

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



Next-Gen Data Science



Next-Gen Data Science

The screenshot shows a web browser window titled "Posit Team" with the URL "j6foiaib-duloftf-posit-software-pbc-dev.snowflakecomputing.app". The page displays the "posit Team" logo and navigation links for "User Docs" and "Admin Docs". Below this, there are three service status cards:

- posit Workbench**: Status: Service running. Click Manage to administer Workbench. Version: 2024.09.1. Manage button.
- posit Connect**: Status: Service running. Click Manage to administer Connect. Version: 2025.04.0. Manage button.
- posit Package Manager**: Status: Service running. Click Manage to administer Package Manager. Version: 2024.08.0. Manage button.



Learn More

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.

The screenshot shows a browser window with a Python script in the left panel and a dashboard analysis in the right panel. The Python script is titled 'app.py' and imports various libraries like pandas, plotly, and snowflake.connector. It includes code for connecting to a database and running a query. The dashboard analysis is titled 'Tour de France Analysis' and lists questions like 'What was the average rider age for each year?' and 'Which rider has won the most stages?'. Below the dashboard, there's a Shiny app interface with a query editor containing a SQL SELECT statement and a message about filtering riders who won at least one stage.

Early Access



Visit Us

Attend Keynote Access

FLAKE EXPERIENCE

Closed Wall



Booth 1903

THANK YOU



© 2025 Snowflake Inc. All Rights Reserved