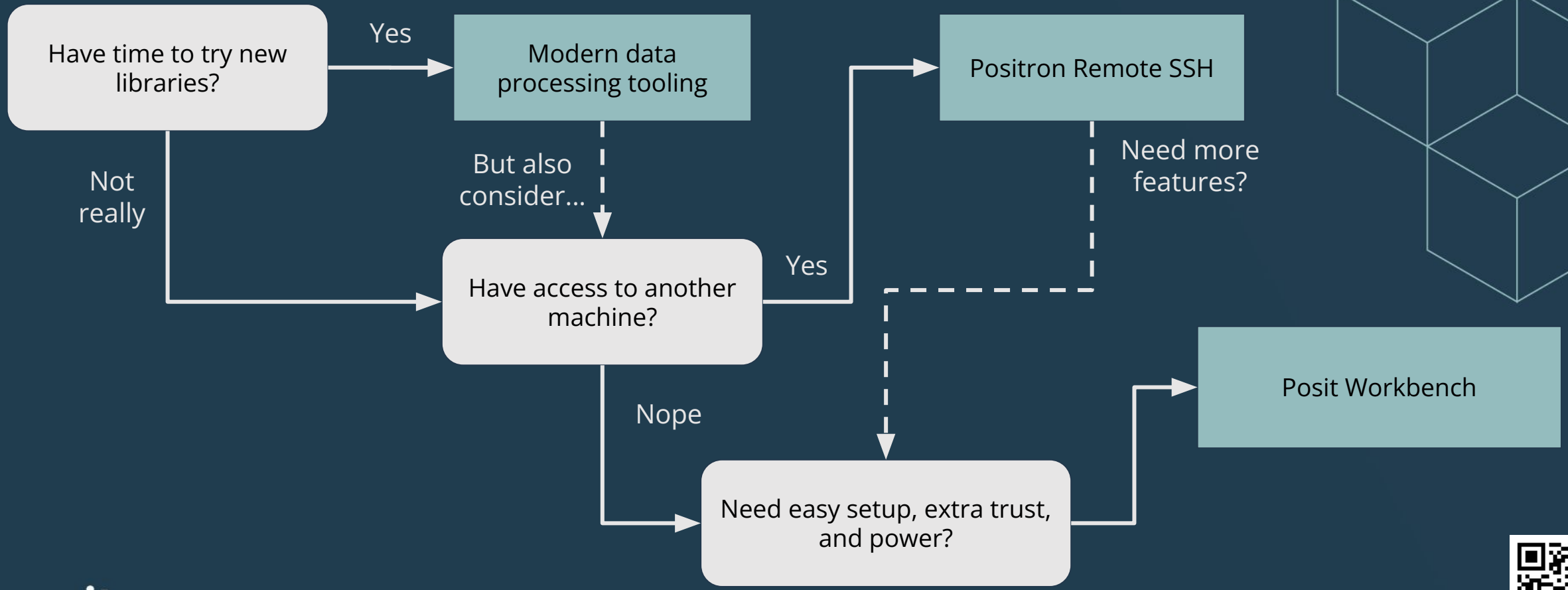
Outgrowing Your Laptop With Positron

Austin Dickey / 2025-09-17

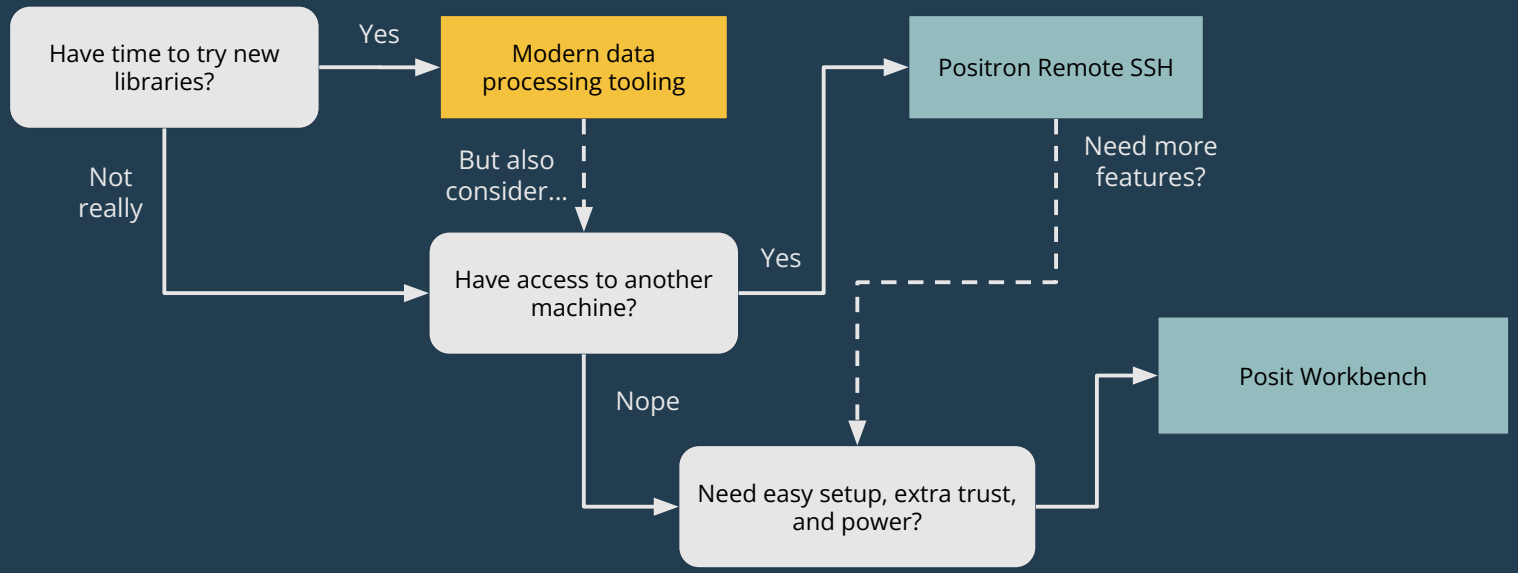
posit conf(2025)

GitHub repo with demo and slides here

What do I do when I'm struggling to analyze data in memory on my laptop?



Modern Data Processing



Who is this for?

- People who don't have access to any other machines
- Quick, local iterations
- Proofs of concept
- (Though these techniques are always useful!)

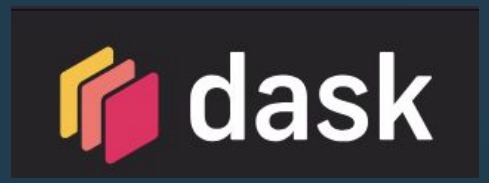
The NYC Taxi Dataset

- All trips between 2009-2019
- ~36GB on disk in parquet format
- Would be ~400GB if loaded into pandas

https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

What can I do?

- Luckily there are so many tools that help with this!
  - R and Python work with most of these



What can I do?

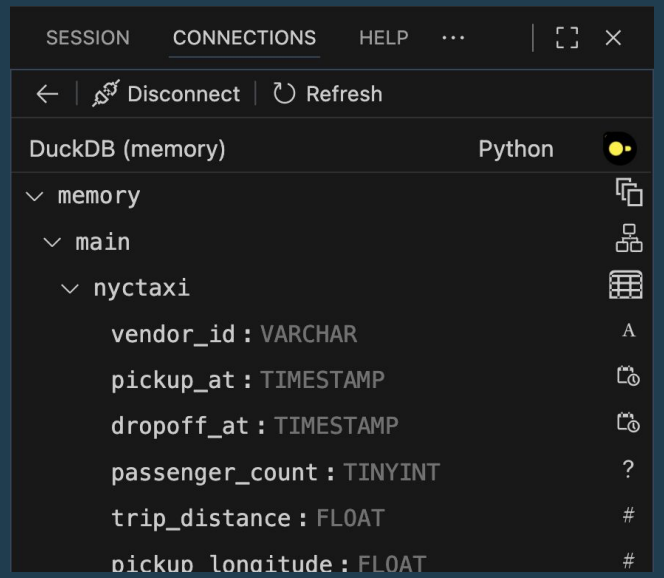
- They typically use the following strategies:
  - **Deferred/lazy evaluation**: don't query the data until the last second
  - **Predicate pushdown**: only read the data you need
  - **Split the work** among multiple processors if possible

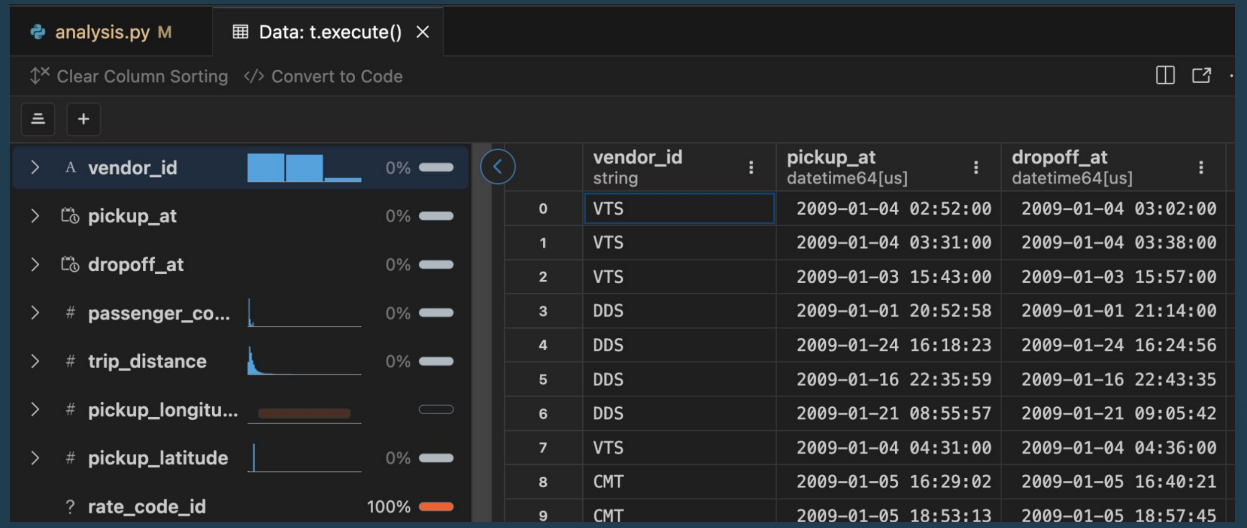
In Python, I like Ibis, which uses any of these tools.

```python
import ibis

# Set up any backend
con = ibis.duckdb.connect()
con = ibis.polars.connect()
con = ibis.pyspark.connect(session=SparkSession.builder.getOrCreate())
con = ibis.connect(f"postgres://{user}:{password}@{host}:{port}/{database}")

# Load an ibis table (this doesn't actually read the data yet)
t = con.read_parquet(
    "s3://ursa-labs-taxi-data/**",
    table_name="nyctaxi",
    union_by_name=True,
)
```

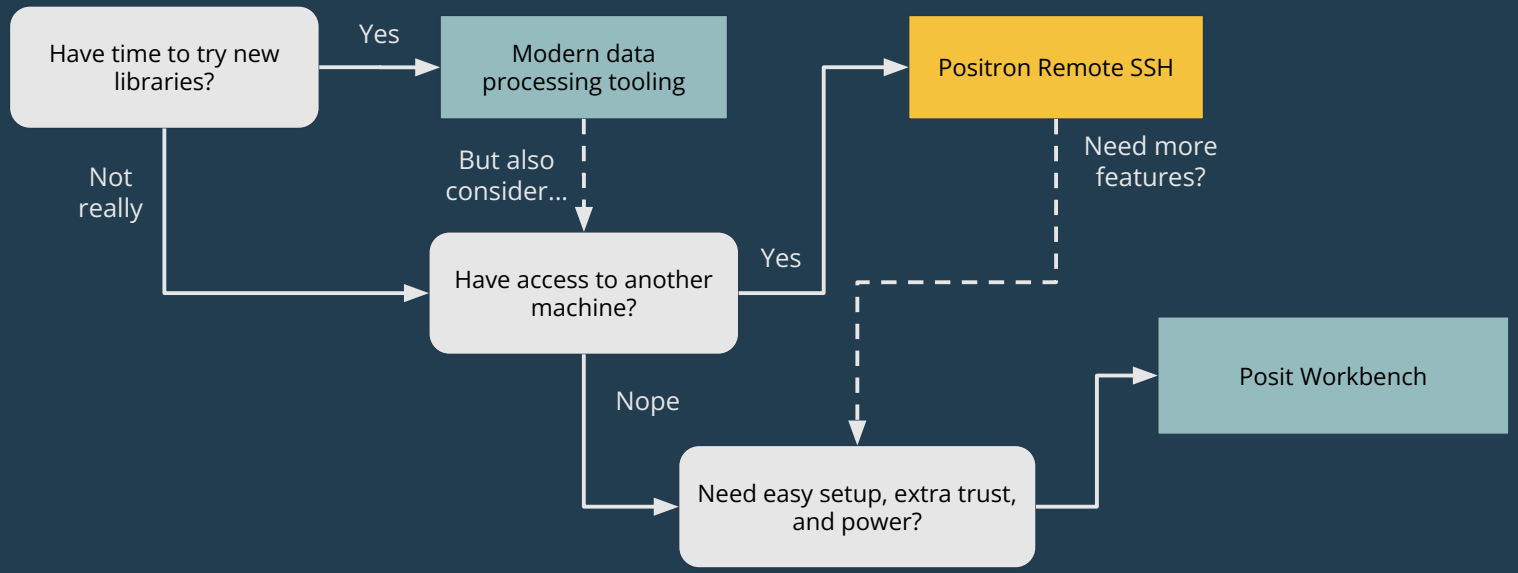
Why in Positron?

- **Help Pane** lets you learn new tools faster
- **Connections Pane** lets you understand your databases better
- **Data Explorer** lets you understand your data better





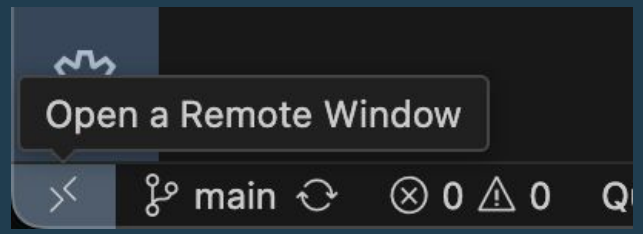
Remote SSH

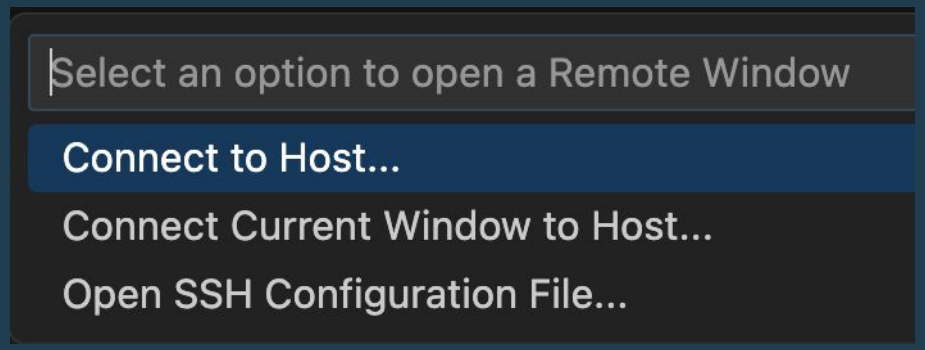

Have time to try new libraries?
Yes
Modern data processing tooling
Positron Remote SSH
Not really
But also consider...
Need more features?
Have access to another machine?
Yes
Nope
Posit Workbench
Need easy setup, extra trust, and power?

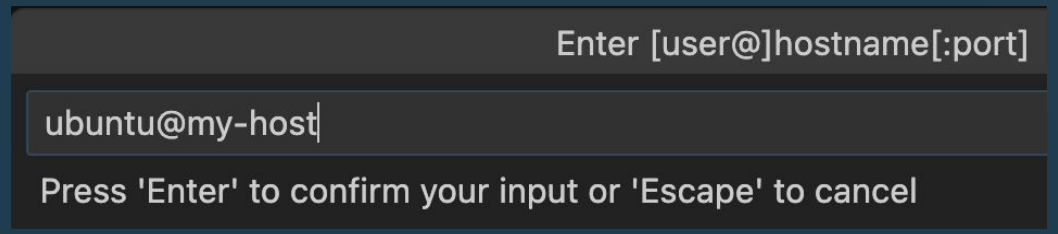
Who is this for?

- People who have access to another machine
  - Anything from a spare box to a high-performance cluster
- People who can set one up and maintain it
- Scrappy data science teams

Connecting


Open a Remote Window
main
0
0


Select an option to open a Remote Window
Connect to Host...
Connect Current Window to Host...
Open SSH Configuration File...


Enter [user@]hostname[:port]
ubuntu@my-host
Press 'Enter' to confirm your input or 'Escape' to cancel

analysis.py — ssh-demo [SSH: i-099684c376a0c179d]

New | Open | Search | Python 3.13.7 (Uv: ssh-demo) | ssh-demo

EXPLORER

SSH-DEMO [SSH: I-099684C376A0C17...

- .venv
- data
- .gitignore
- analysis.py M
- download.py
- pyproject.toml
- README.md
- uv.lock

analysis.py M

analysis.py > ...

```python
# Data from https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page
# hosted by Ursa Labs (see https://ursalabs.org/arrow-r-nightly/articles/dataset.htm

import ibis
from ibis import _
from plotnine import ggplot, aes, geom_bar, scale_x_continuous, ggtitle

# Set up any backend
con = ibis.duckdb.connect()

# Load an ibis table (this doesn't actually read the data yet)
t = con.read_parquet(
    "data/**",
    table_name="nyctaxi",
    union_by_name=True,
)

# Inspect the table structure
t
len(t.columns)
t.count().execute()

# Find the mean tip percentage by number of passengers, among trips costing more tha
tip_analysis = (
    t.filter(
        [
```

CONSOLE | TERMINAL | PROBLEMS | OUTPUT | PORTS 2 | DEBUG CONSOLE

~/ssh-demo

```
...         data=tip_analysis,
...         mapping=aes(x="passenger_count", y="mean_tip_pct", fill="log_count"),
...     )
...     + geom_bar(stat="identity")
...     + scale_x_continuous(breaks=range(15))
...     + ggtitle("Tip % by passenger count (for expensive trips)")
... )
...
100%
>>>
```

OUTLINE

TIMELINE

SESSION | CONNECTIONS | HELP | VIEWER

VARIABLES

Python 3.13.7 (Uv: ssh-demo) | filter

VALUES

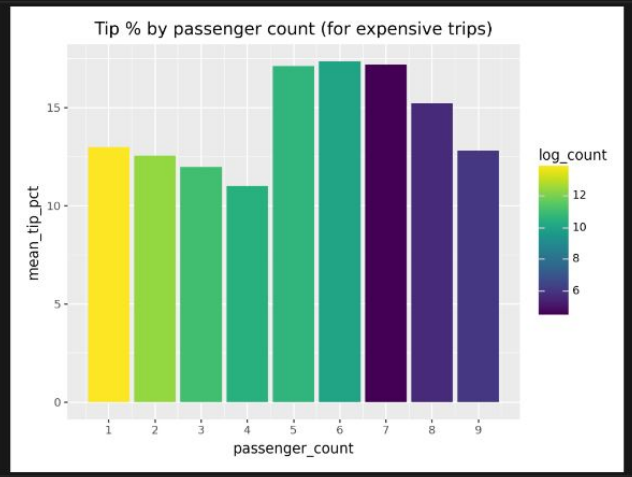
_	_	Deferred
con	<ibis.backends.duckdb.Backend objec...	Backend
t	ibis.expr.types.relations.Table	ibis.Expr
tip_analysis	ibis.expr.types.relations.Table	ibis.Expr

CLASSES

aes	<class 'plotnine.mapping.aes.aes'>	type
geom_bar	<class 'plotnine.geoms.geom_bar.geo...	Register

PLOTS

Fit | Intrinsic



SSH: i-099684c376a0c179d | main* | ./.venv/bin/python | 0 0 | 2 | Ln 49, Col 1 | Spaces: 4 | UTF-8 | LF | Python

posit conf (2025)

analysis.py — ssh-demo [SSH: i-099684c376a0c179d]

New Open Search Python 3.13.7 (Uv: ssh-demo) ssh-demo

EXPLORER

SSH-DEMO [SSH: I-099684C376A0C17...

- .venv
- data
- .gitignore
- analysis.py M
- download.py
- pyproject.toml
- README.md
- uv.lock

OUTLINE

TIMELINE

analysis.py M

analysis.py > ...

```python
# Data from https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page
# hosted by Ursa Labs (see https://ursalabs.org/arrow-r-nightly/articles/dataset.htm

import ibis
from ibis import _
from plotnine import ggplot, aes, geom_bar, scale_x_continuous, ggtitle

# Set up any backend
con = ibis.duckdb.connect()

# Load an ibis table (this doesn't actually read the data yet)
t = con.read_parquet(
    "data/**",
    table_name="nyctaxi",
    union_by_name=True,
)

# Inspect the table structure
t
len(t.columns)
t.count().execute()

# Find the mean tip percentage by number of passengers, among trips costing more tha
tip_analysis = (
    t.filter(
        [
```

CONSOLE TERMINAL PROBLEMS OUTPUT PORTS 2 DEBUG CONSOLE

~/ssh-demo

```
...         data=tip_analysis,
...         mapping=aes(x="passenger_count", y="mean_tip_pct", fill="log_count"),
...     )
...     + geom_bar(stat="identity")
...     + scale_x_continuous(breaks=range(15))
...     + ggtitle("Tip % by passenger count (for expensive trips)")
... )
...
100%
>>>
```

SESSION CONNECTIONS HELP VIEWER

VARIABLES

Python 3.13.7 (Uv: ssh-demo) filter

VALUES

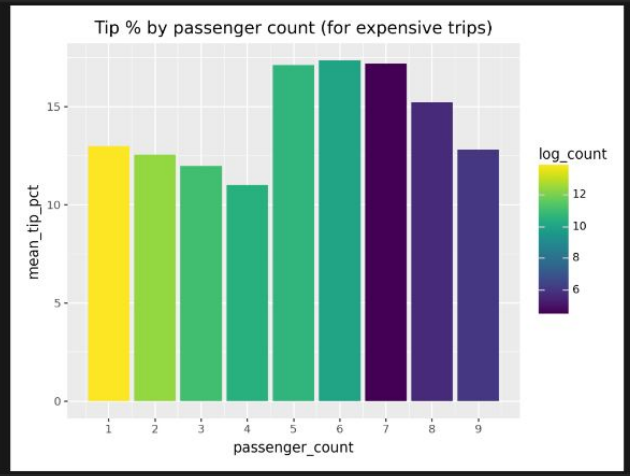
_	_	Deferred
con	<ibis.backends.duckdb.Backend objec...	Backend
t	ibis.expr.types.relations.Table	ibis.Expr
tip_analysis	ibis.expr.types.relations.Table	ibis.Expr

CLASSES

aes	<class 'plotnine.mapping.aes.aes'>	type
geom_bar	<class 'plotnine.geoms.geom_bar.geo...	Register

PLOTS

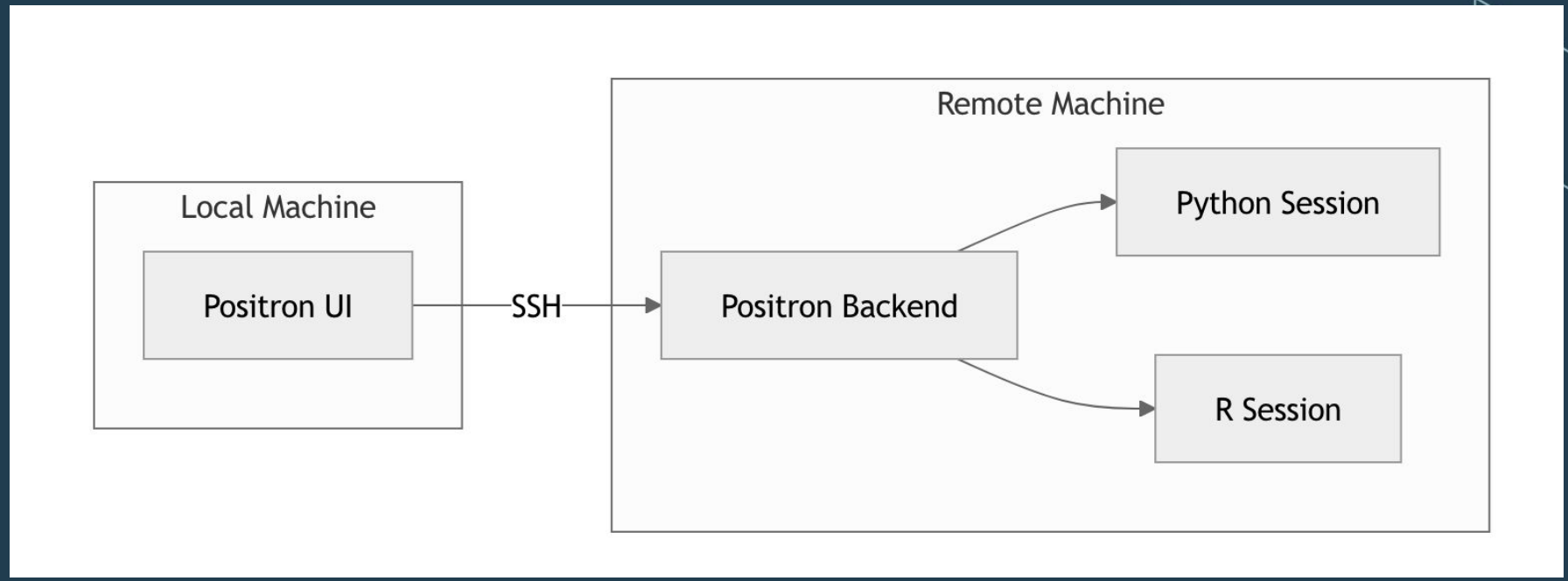
Fit Intrinsic



SSH: i-099684c376a0c179d main* ./.venv/bin/python ⊗ 0 ⚠ 0 2 Ln 49, Col 1 Spaces: 4 UTF-8 LF {} Python

posit conf (2025)

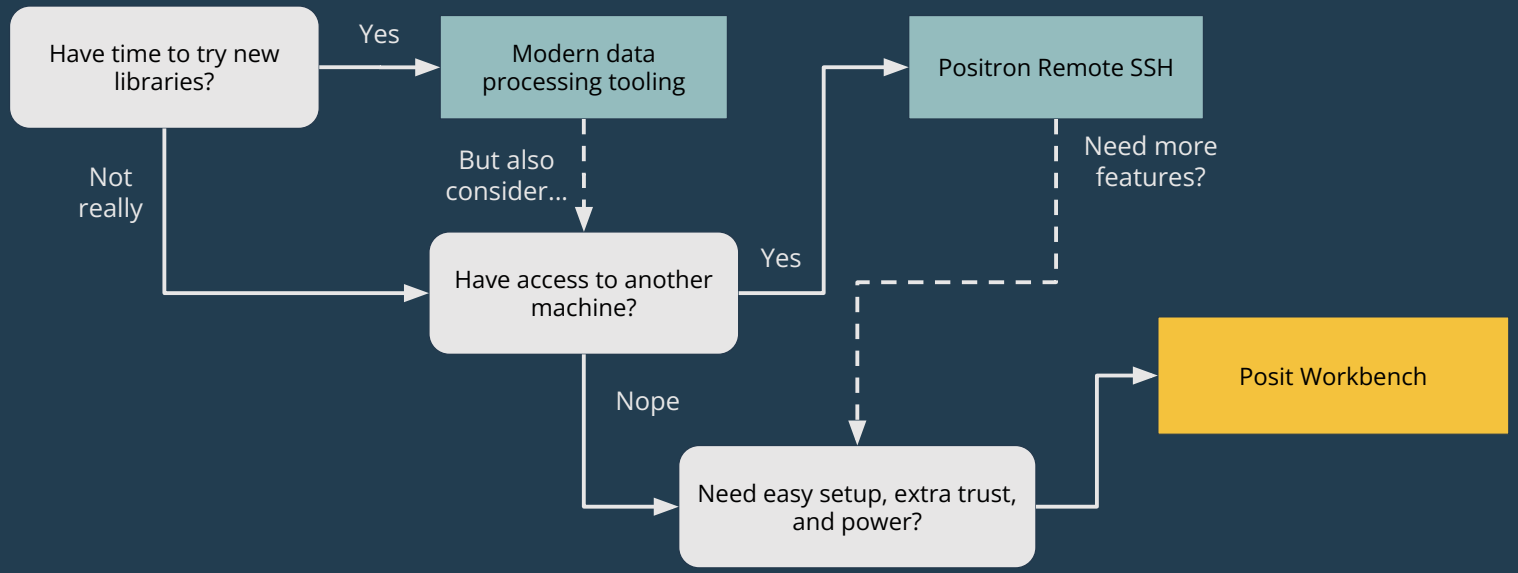
Architecture



Benefits

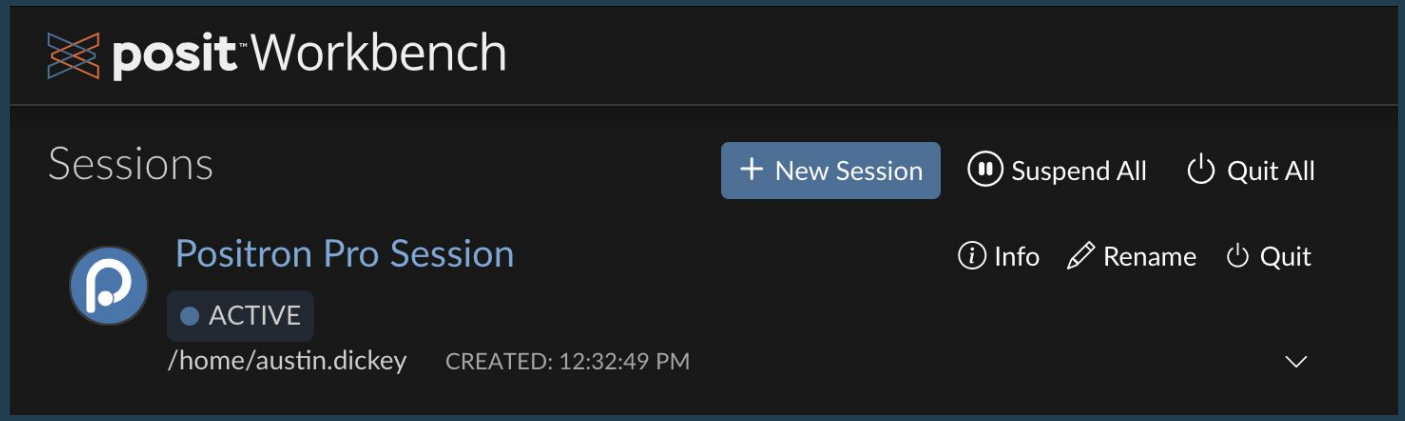
- It's free to use, including at work
- You don't have to change any code
  - Use pandas on a huge machine if you really want!
- You can disconnect and reconnect while running code

Posit Workbench



Who is this for?

- People who need a lot more power and scaling
- Industries with high compliance and trust standards
- Teams looking for standardized environments and data access



Benefits

- Like SSH and more
- Cluster management, security/auditing, consistent development environments
- Working with partners like Snowflake

Want to learn more?

- "Posit's Commercial Products" - today at 1:00pm
- "Enterprise Data Platforms" - tomorrow at 2:40pm
- I'll hang out in the Positron Lounge tomorrow morning