

Assessing Data Pipeline Quality & Sanity with *Data Angiograms*

Vishnu Rao

Cost Based Data engineer & Senior Data guy at @ cueZen.com

Chief **Inspiration** Officer @ byteSpire.io

“I have little confidence in our data pipelines...”

- *Very common feeling in data teams*

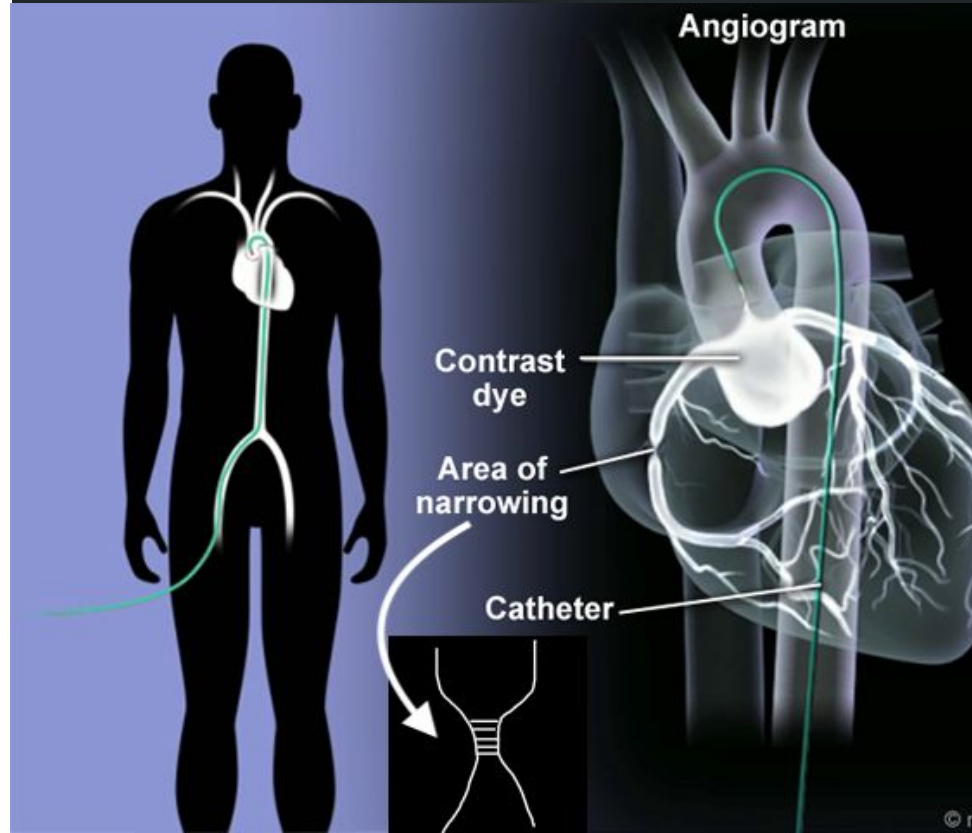
Data quality matters!

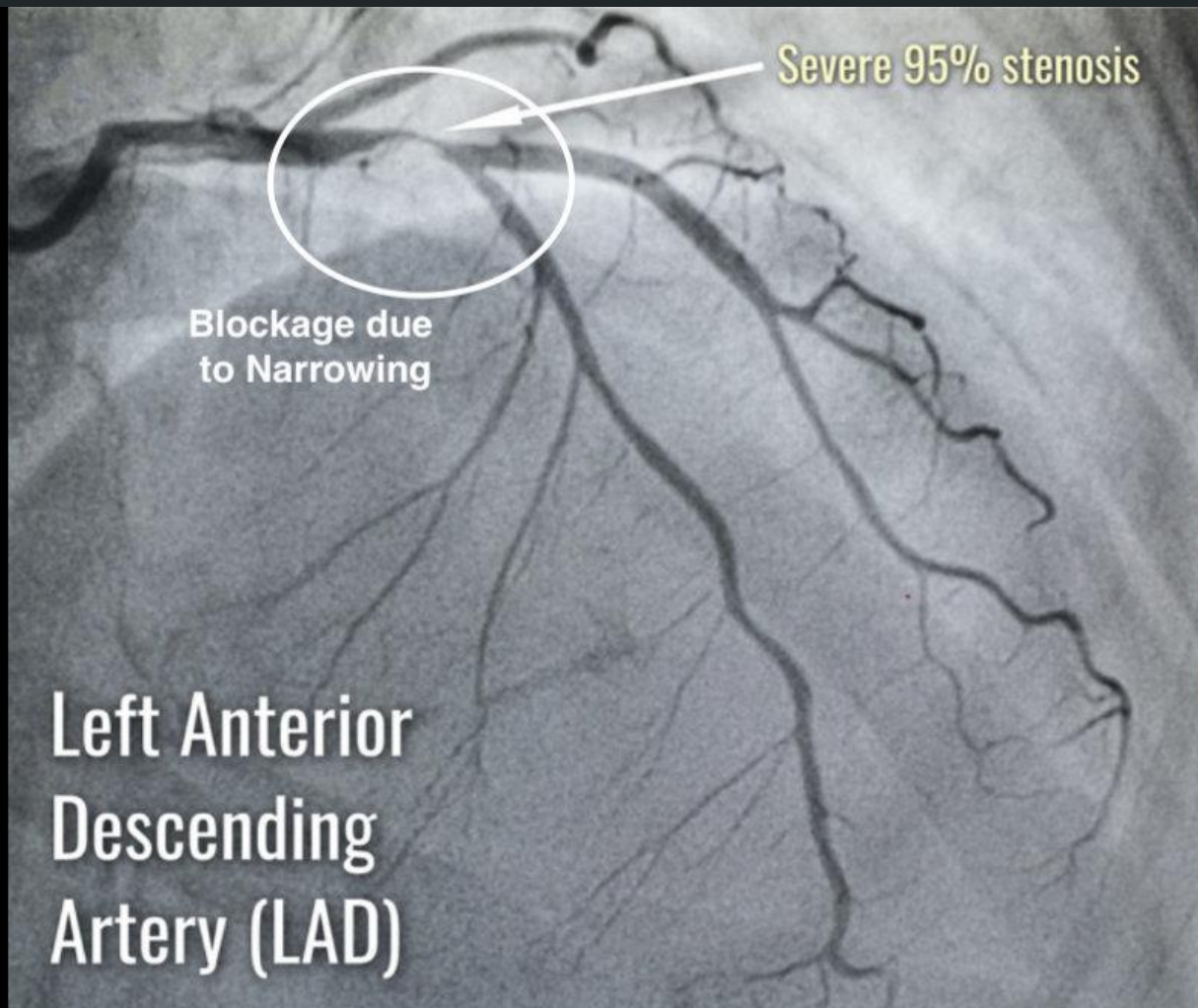
3. Uber's \$45M Driver Payment Miscalculation

In 2017, reports emerged that Uber had been miscalculating its commission and costing New York drivers a percentage of their rightful earnings. Instead of calculating its commission based on its net fare, minus sales tax and other fees, Uber took their cut based on the gross fare. This meant that for two and a half years, the company took 2.6% more from drivers than its own terms and conditions allowed.

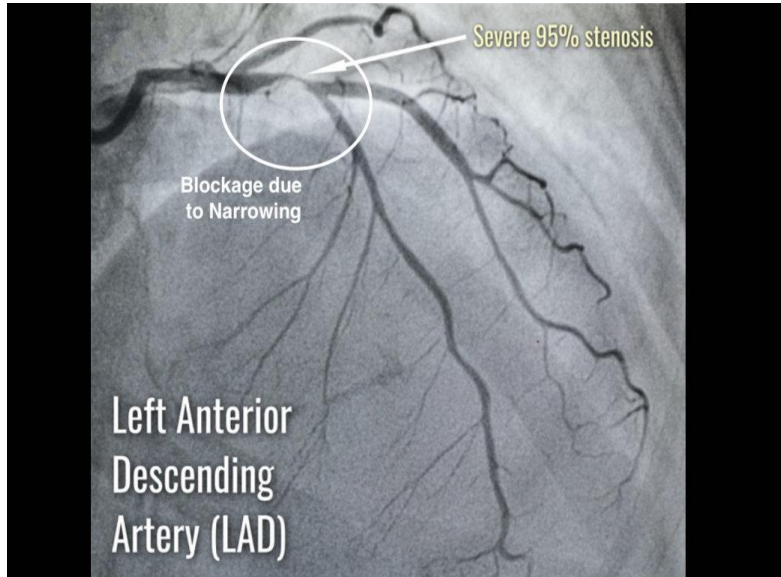
reference: <https://www.montecarlodata.com/blog-bad-data-quality-examples/>

Angiogram





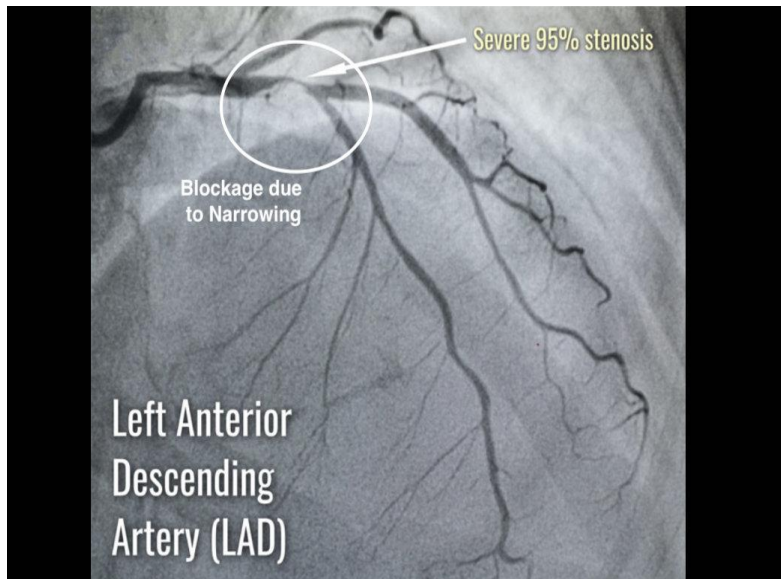
The Parallels =



=

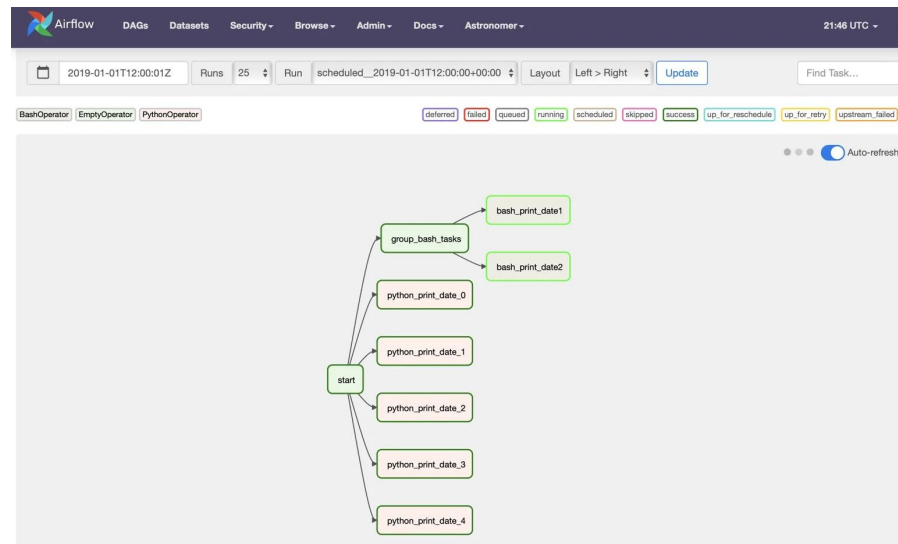
Arteries

The Parallels =



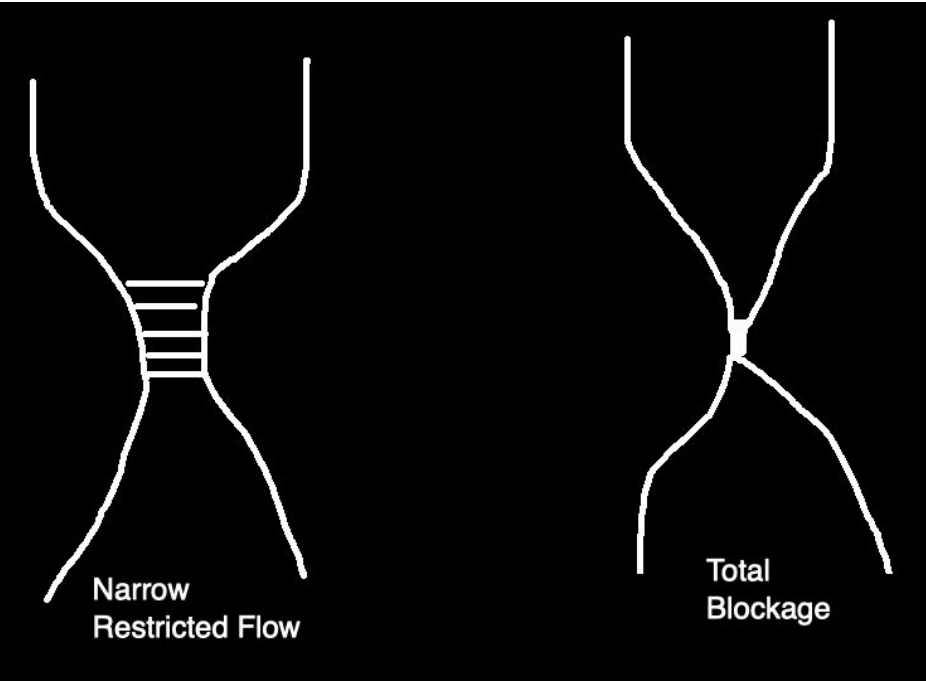
Arteries

=

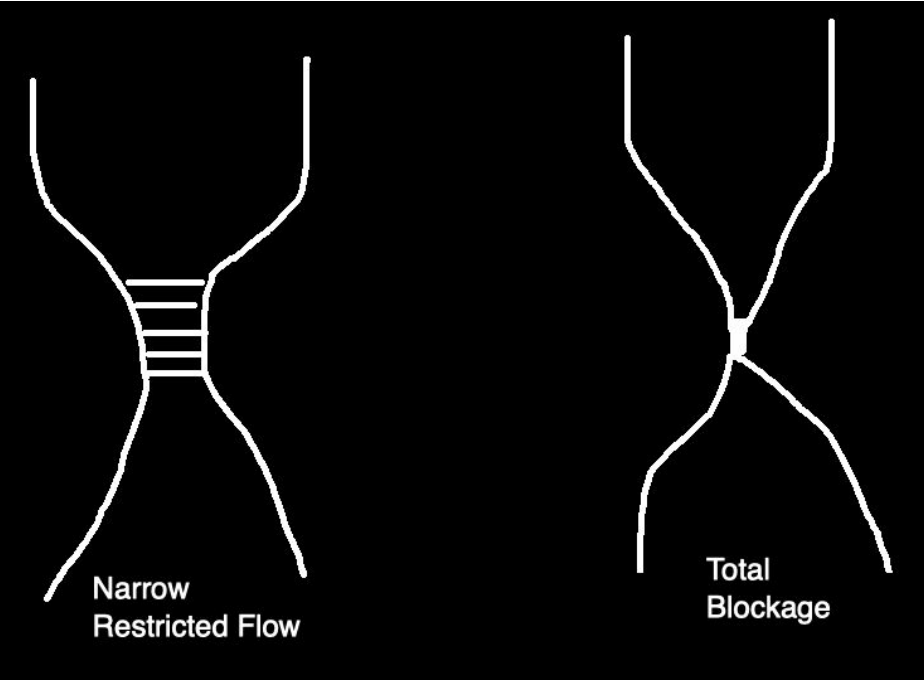


**Data Pipeline
ex: Airflow Dag**

The Parallels ... =



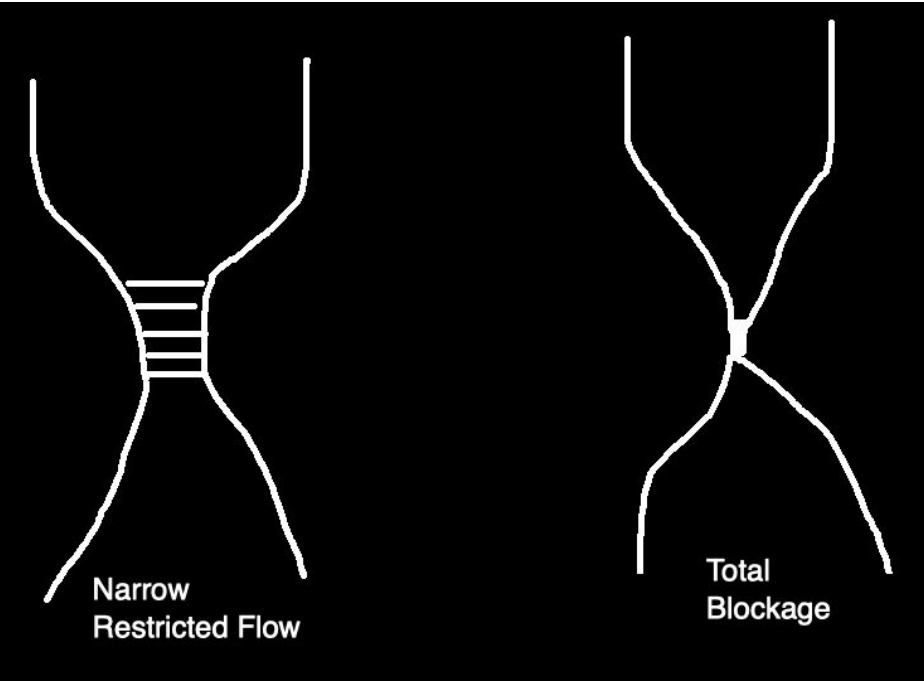
The Parallels ... =



=

**Pipeline
Succeeded
But data
consistency,
accuracy,
completeness
MAY be
affected**

The Parallels ... =



=

Pipeline Succeeded
But data consistency, accuracy, completeness
MAY be affected

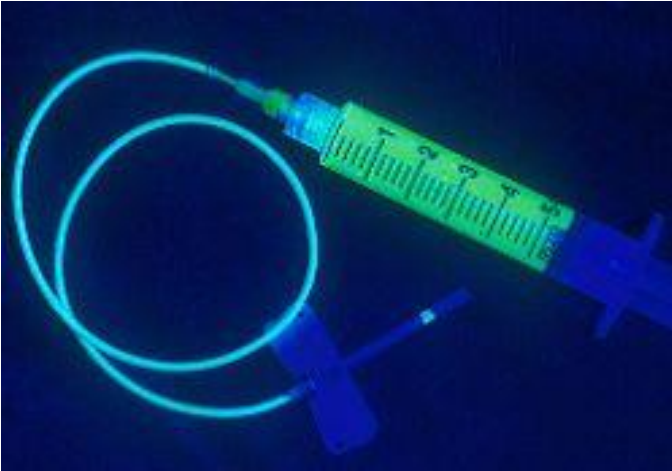
Pipeline job failed.
Data flow interrupted

Data Angiogram (n):

A curated Test dataset, which we inject into a Data Pipeline & compare actual Vs expected results of each job/component in the pipeline.



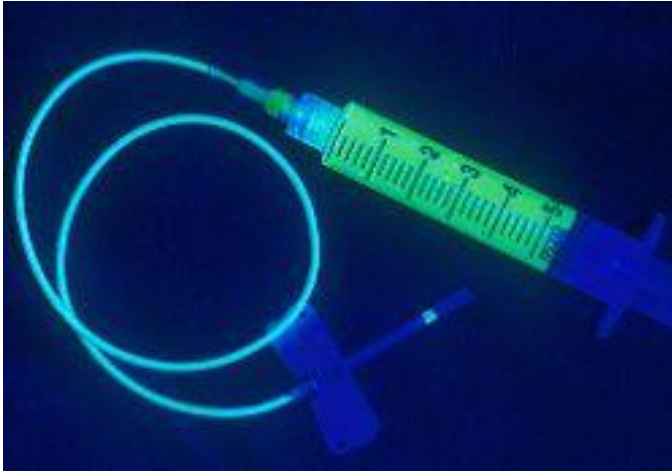
The Parallels ... =



=

**The Dye added
to the blood**

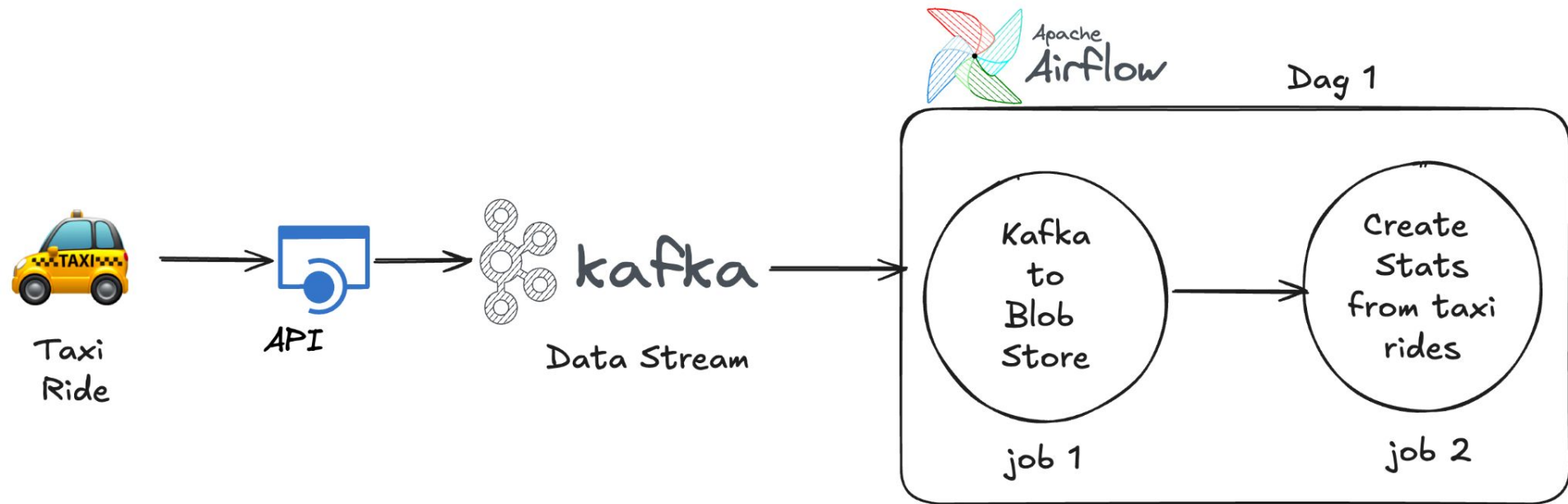
The Parallels ... =



**The Dye added
to the blood**

**= A specific Test data set created /
curated to represent a specific
test case.**

Let's consider - Taxi Data Pipeline



Let's consider - Taxi Data Pipeline ...

```
{  
  "taxi_id": "t1",  
  "pax_id": "vishnu",  
  "start_time": "2024-09-05 19:00:00",  
  "end_time": "2024-09-05 19:20:00",  
  "pickup_zipcode": "10001",  
  "dropoff_zipcode": "08405",  
  "tax_amount": 1.3,  
  "fare_amount": 15.2,  
  "currency": "USD",  
  "car_model": "Toyota leaf"  
}
```



Date	Car model	Total Revenue	currency
2024-09-05	Toyota leaf	1000.2	USD

Building the Data Angiogram

1. Setting up the pipeline:

Building the Data Angiogram

1. Setting up the pipeline:
 - a. **Environment:**

Building the Data Angiogram

1. Setting up the pipeline:

a. **Environment:**

- i. Github / Gitlab Pipeline - i.e. your Continuous Integration environment
- ii. Run at cadence - say daily

Building the Data Angiogram

1. Setting up the pipeline:

a. Environment:

- i. Github / Gitlab Pipeline - i.e. your Continuous Integration environment
- ii. Run at cadence - say daily

b. **Tools:**

Building the Data Angiogram

1. Setting up the pipeline:

a. Environment:

- i. Github / Gitlab Pipeline - i.e. your Continuous Integration environment
- ii. Run at cadence - say daily

b. Tools:

i. Docker:

1. Infra - Kafka, Postgres, MySQL, Airflow/Prefect** Image, minikube **
2. Pipeline Job related - Job specific image

Building the Data Angiogram

1. Setting up the pipeline:

a. Environment:

- i. Github / Gitlab Pipeline - i.e. your Continuous Integration environment
- ii. Run at cadence - say daily

b. Tools:

i. Docker:

1. Infra - Kafka, Postgres, MySQL, Airflow/Prefect** Image, minikube **
2. Pipeline Job related - Job specific image

ii. **Test suite:** Pytest (feel free to use junit / testng !)

Building the Data Angiogram

1. Setting up the pipeline:

a. Environment:

- i. Github / Gitlab Pipeline - i.e. your Continuous Integration environment
- ii. Run at cadence - say daily

b. Tools:

i. Docker:

1. Infra - Kafka, Postgres, MySQL, Airflow/Prefect** Image, minikube **
2. Pipeline Job related - Job specific image

ii. Test suite: Pytest (feel free to use junit / testng !)

iii. **Slack** - Notify the team of Success / Failure of the angiogram.

1. Run before your daily Standup

Building the Data Angiogram...

2. Preparing the test data set for the scenario(s)

Building the Data Angiogram...

2. Preparing the test data set for the scenario(s)

a. **Generate Test Data**

- i. Do it manually or have script to generate data or ask chatpgt
 1. Ex: generate 12 taxi rides such that
 - a. There are 3 unique passenger ids tom, jerry & batman
 - b. There are 4 different card models in the data (Toyota, Benz, Ford, Jeep)
 - c. There is 1 Jeep taxi rides whose start_time is very old say 1970
 - d. There is 1 Jeep taxi ride where passenger_id is null
 - e. There is 1 Jeep taxi ride where fare/tax amount is negative.
 - f. There are 2 valid taxi rides for Jeep
 - g. The rest of the taxi rides are for toyota, Benz, Ford

Building the Data Angiogram...

2. Preparing the test data set for the scenario(s)

a. Generate Test Data

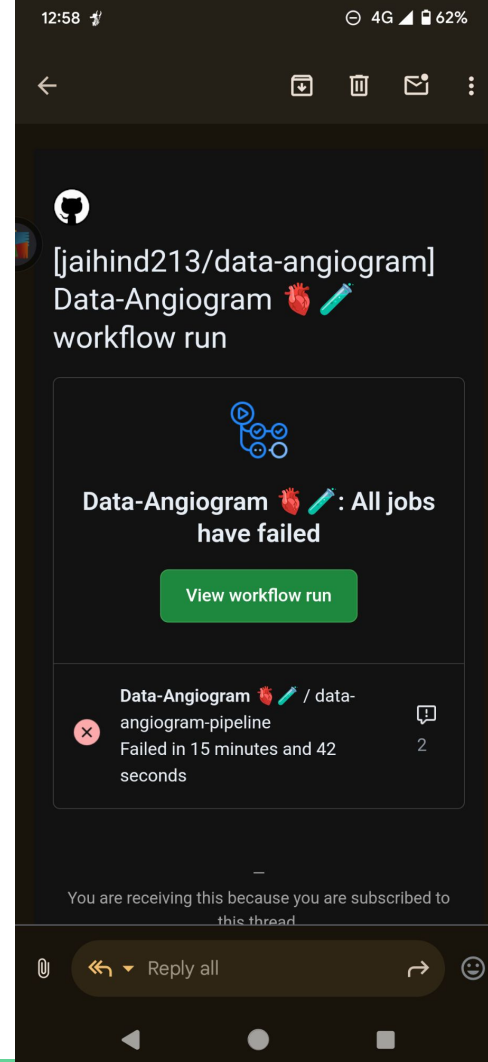
- i. Do it manually or have script to generate data or ask chatpgt
 1. Ex: generate 12 taxi rides such that
 - a. There are 3 unique passenger ids tom, jerry & batman
 - b. There are 4 different card models in the data (Toyota, Benz, Ford, Jeep)
 - c. There is 1 Jeep taxi rides whose start_time is very old say 1970
 - d. There is 1 Jeep taxi ride where passenger_id is null
 - e. There is 1 Jeep taxi ride where fare/tax amount is negative.
 - f. There are 2 valid taxi rides for Jeep
 - g. The rest of the taxi rides are for toyota, Benz, Ford

b. **Validate Test Data !**

- i. Write test case to read test data & validate that the above conditions are met.
- ii. This gives you **confidence** in the angiogram even before we perform it !


Demo

<https://github.com/jaihind213/data-angiogram/actions>




12:58 4G 62%



[jaihind213/data-angiogram]
Data-Angiogram 📄
workflow run


Data-Angiogram 📄: All jobs
have failed

[View workflow run](#)

 Data-Angiogram 📄 / data-angiogram-pipeline
Failed in 15 minutes and 42 seconds 2

You are receiving this because you are subscribed to this thread

 Reply all 

Demo

[https://github.com/jaihind213/
data-angiogram/actions](https://github.com/jaihind213/data-angiogram/actions)

Focus on:

1. Data consistency
2. Data accuracy
3. Data completeness

Refer to

`verify_angiogram.py /
angiogram.yaml`

Summary

1. This is simple way to test the **sanity** & **quality** of your data pipeline.
2. It's **cost-effective** !
3. Its Extensible & Flexible: One can continue to add more Test datasets for various scenarios.
4. It can catch breaking changes say data-type changes ! (Ex: Int becomes String?)
5. **Reminder:** The idea is focus on quality not quantity - hence stress testing is not a motivation here.
6. **Reminder:** The angiogram datasets is important ! You can test it any environment, github/gitlab is a replica of your other environments.

Our Angiogam

@ Cuezen.com

Summary

Jobs

✖ harvestor

Run details

🕒 Usage

📄 Workflow file

harvestor

failed 3 hours ago in 8m 11s


- > ✔ Set up job
- > ✔ Generate GitHub token
- > ✔ Run actions/checkout@v4
- > ✔ Run actions/checkout@v4
- > ✔ Run actions/checkout@v3
- > ✔ Run actions/setup-python@v5
- > ✔ Set up JDK 17
- > ✔ 🌱 let Pythons into garden
- > ✔ 🔒 Login to docker registry
- > ✔ 📦 Dockers 🚧
- > ✔ 🚦 Health Check Cuebox Api
- > ✔ 🚦 Health Check Cuebox Frontend
- > ✔ 🚦 Health Check Nsapi
- > ✔ 🌱 Plant Seeds
- > ✔ 📶 Reached GRPC
- > ✔ 📶 Reached Kafka
- > ✖ 🏰 Reached Tower
- 🔄 🗄 Run MarkerHub
- 🔄 🐹 Run Gopher
- 🔄 📄 Check Stats db
- 🔄 🌱 Run for next day Gopher
- 🔄 🌱 Verify Harvest
- 🔄 🔥 Tear down
- > ✔ 🗣 Slack

Latest Failures 2024 @ Cuezen

- Data Timezone issues
- Data **not** being ignored
- Api missing data fields

harvestor ▾

+ Add a bookmark


 **harvestor-Run** APP 1:26 PM Wednesday, August 14th ▾

Harvesting Seeds - Integration test CE-291 has passed

✓ Harvesting Seeds - Integration test CE-291 passed in [\[redacted\]efs/heads/main](#) on 1e74494

[View Run](#) . this is schedule run

Thursday, August 15th ▾


 **harvestor-Run** APP 1:19 PM

Harvesting Seeds - Integration test CE-291 has failed

✗ Harvesting Seeds - Integration test CE-291 failed in [\[redacted\]efs/heads/main](#) on 1e74494

[View Run](#) . this is schedule run

Yesterday ▾

 **harvestor-Run** APP 1:19 PM

Harvesting Seeds - Integration test CE-291 has failed

✗ Harvesting Seeds - Integration test CE-291 failed in [\[redacted\]efs/heads/main](#) on 1e74494

[View Run](#) . this is schedule run

Advanced Angiogram Technique

Perform a Data Angiogram in **Production** Environment.

Ex:

Inject a special taxi ride (virtual taxi ride / fake ride)

- The taxi_id , passenger_id, car model will have special values/code words which don't clash with production dataset.
 - Ex:
 - Taxi_id = Star_Ship_Enterprise_NCC1701
 - Passenger_id = Optimus_Prime
 - Car_model = Startrek_Federation_star_ship_constitution_class_x901910
- Send a taxi ride through the production pipeline....
- check if car_model you specified shows up in your datasets/reports etc, It should!

“Quality data
means Quality
decision making”

- *Quote from a data
engineering leader at Cisco*

You can reach me at

[Linkedin](#)

[X](#)

[My Medium Blog](#)

<https://byteSpire.io>

Thank you
