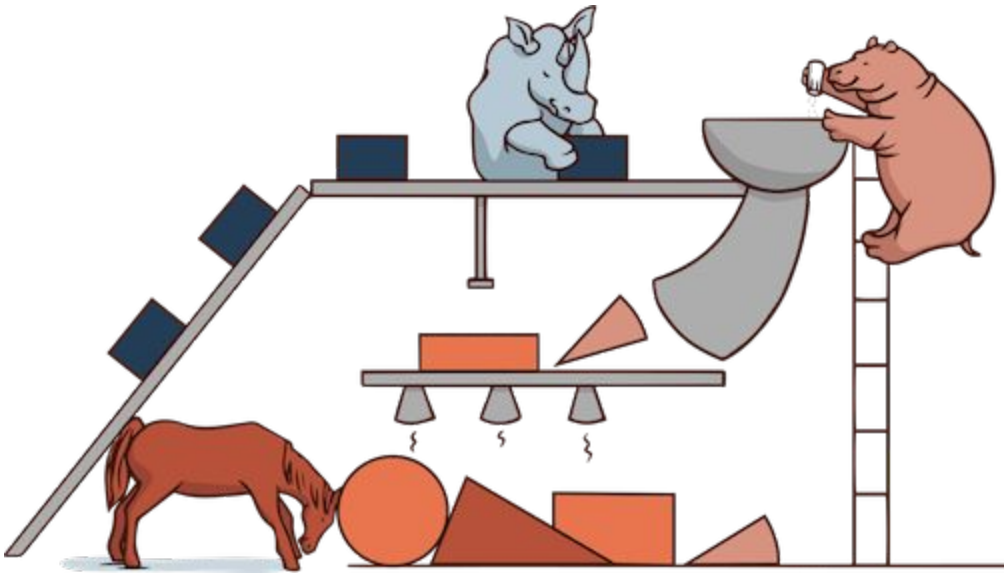
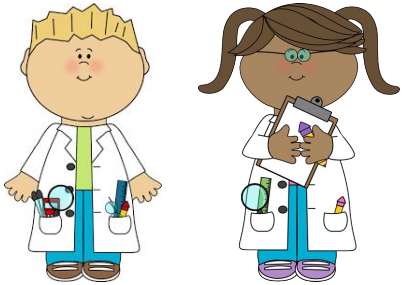
Pachyderm and the Data Science Workflow



Thanks to www.pachyderm.io and the pachyderm community for the code and illustrative examples


A B C D E F



Data

Analysis

Results

New Analysis

New Data

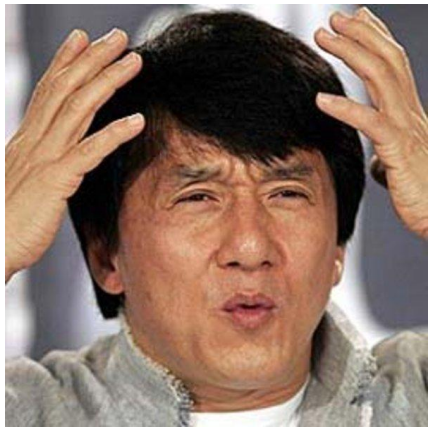


Image source: bfilipek.com an
mycutegraphics.com

Need snapshots of all changes

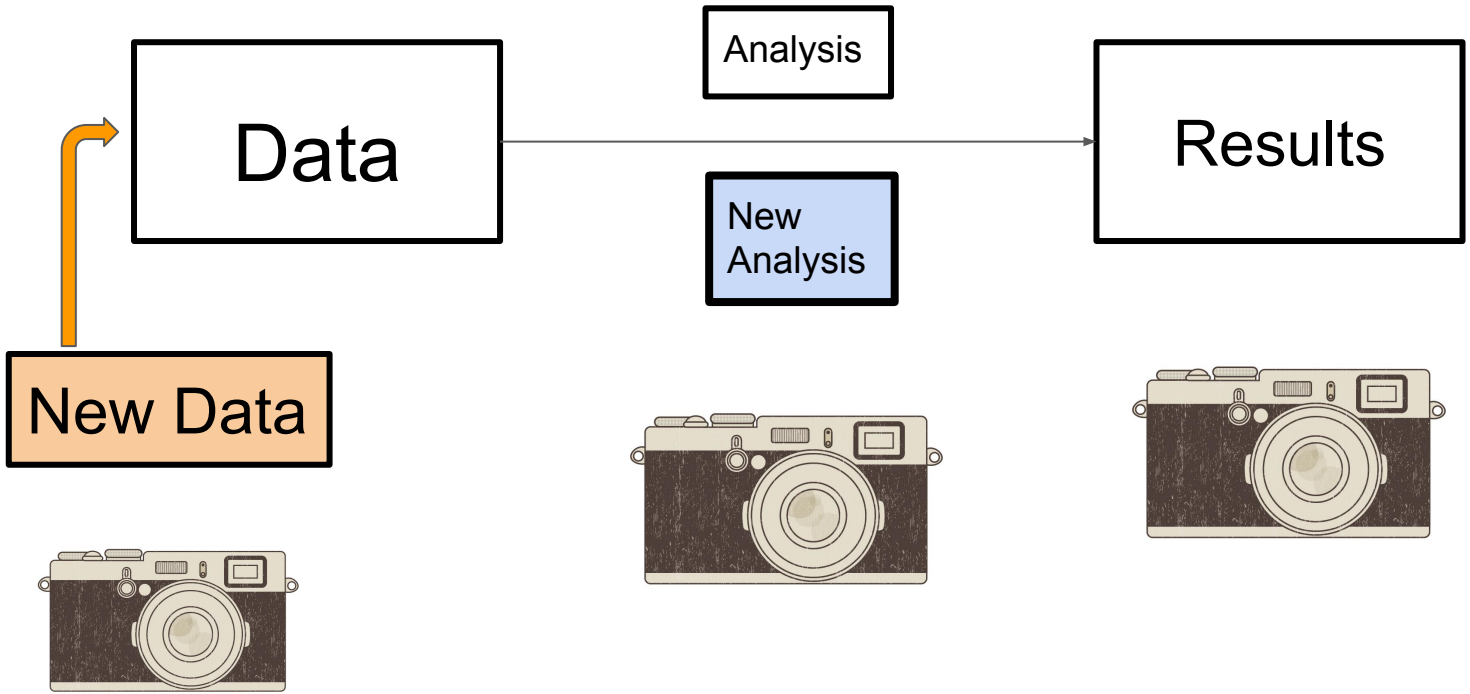


Image source: www.clipartkid.com

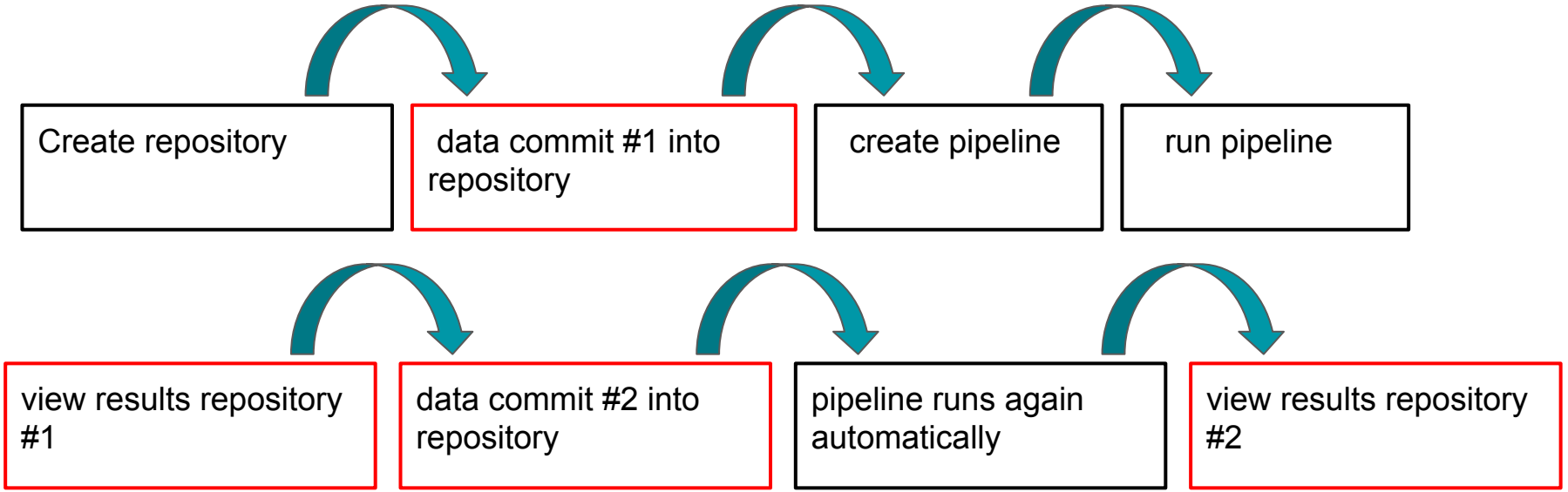
The pitch: Pachyderm is Git for Data Science

Analogous Commands:

create-repo
commit
list-repo
get-file
list-job

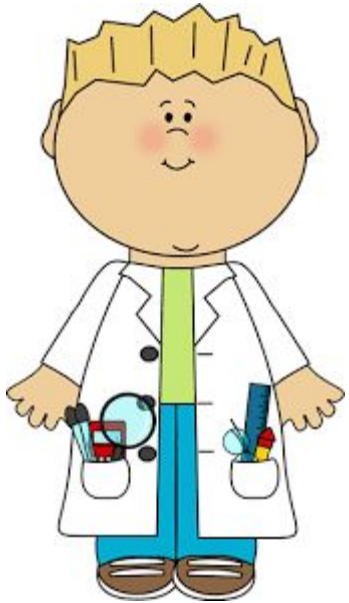
Image source: pachyderm.io

A High-Level Overview



Our data commits and analysis results are versioned. We can fork/track every step.

Classic Pachyderm Example



We own a fruit stand business and want to analyse fruit sales

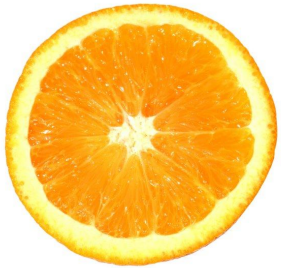


Image source: mycutegraphics.com
Examples and code from pachyderm.io

start kubernetes and deploy pachyderm

```
minikube start

pachctl deploy local
```

minikube

step 1: create repo to store data

```
$ pachctl create-repo data


# See the repo we just created
$ pachctl list-repo
NAME                CREATED             SIZE
data                15 seconds ago      0 B
```

step 2: commit data to repo

```
$ pachctl put-file data master sales -c -f
https://raw.githubusercontent.com/pachyderm/pachyderm/v1.3.2/doc
/examples/fruit_stand/set1.txt
```

```
$ pachctl list-repo
NAME                CREATED             SIZE
data                4 minutes ago       1.707 KiB
```

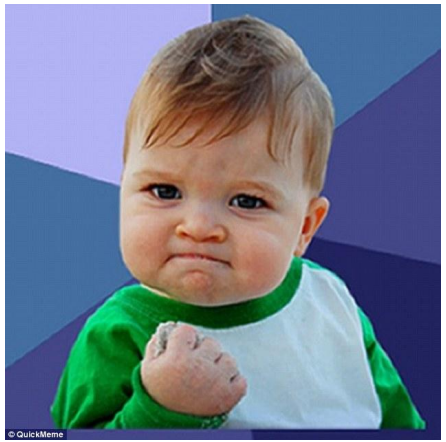


Image source: imgflip.com

```
We can view the commit we just created
pachctl list-commit data
BRANCH              REPO/ID         PARENT              STARTED
FINISHED            SIZE
master              data/master/0   <none>              6 minutes ago       6
minutes ago         1.707KB

# We can also view the contents of the file that we just added
$ pachctl get-file data master sales
orange          4
banana          2
banana          9
orange          9
...
```

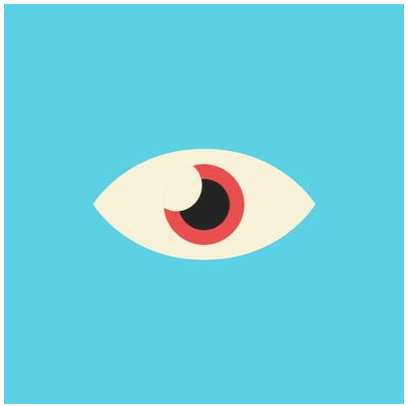


Image source: giphy.com

step 3: create pipeline

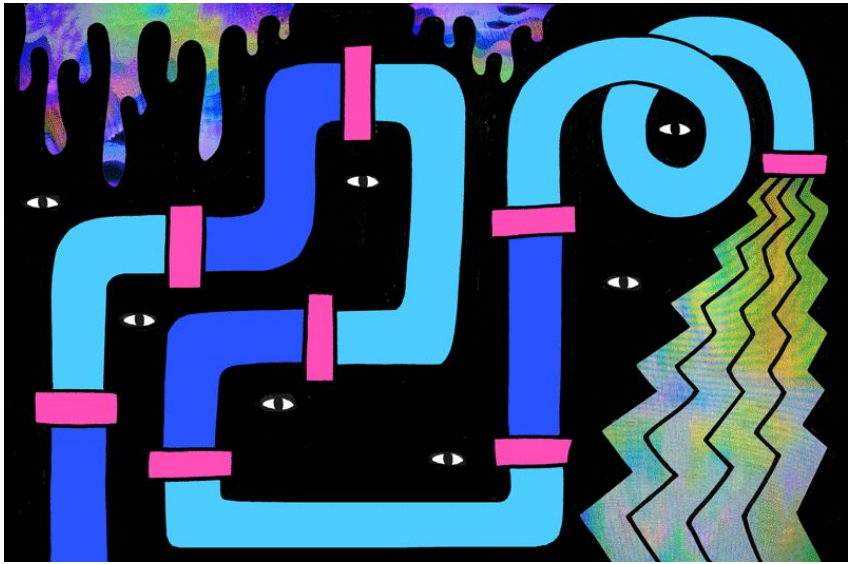
```
{
 "pipeline": {
   "name": "edges"
 },
 "transform": {
  "cmd": [ "R", "/fruitstand.R" ],
  "image": "jeannefukumaru/fruit-stand"
 },
 "parallelism_spec": {
   "strategy": "CONSTANT",
   "constant": "1"
 },
 "inputs": [
   {
     "repo": {
       "name": "sales"
```

- Name our pipeline
- myscript.R placed in an R Docker image.
- Reference myscript.R in our pipeline which is written as a JSON manifest
- In this case, we **filter** by fruit, then **sum** each group

Pipelines are language agnostic

Run and View Results

```
$ pachctl create-pipeline -f
```



```
# view the job
$ pachctl list-job
       ID                                OUTPUT
90c74896fd227f319c3c19459aa7a22b   sum/e4060e15948c4b7b89947a02eace5dca/0
67c30d70ba9d2179aa133255f5dc81db   filter/d737e9b7cfae40d4aa8a8871cdb9f783/0
```

Unique ID for reference. Allows for tracking and access

Image source: giphy.com

Results are a repo too

```
$ pachctl list-repo
NAME                CREATED             SIZE
sum                 2 minutes ago       12 B
filter              2 minutes ago       200 B
data                19 minutes ago      874 B
```

add more data as it streams in

```
$ pachctl put-file data master sales -c -f set2.txt
```

```
$ pachctl list-commit data
BRANCH              REPO/ID                 STARTED            FINISHED
master              data/master/0           19 minutes ago     19 minutes ago
master              data/master/1           2 minutes ago      2 minutes ago
```

Pipeline runs automatically

```
$ pachctl get-file sum e4060e15948c4b7b89947a02eace5dca/1 apple
324
```

Image source: giphy.com

Summary

- Snapshots of Data at all steps of the workflow

- Inspect with the flush-commit command

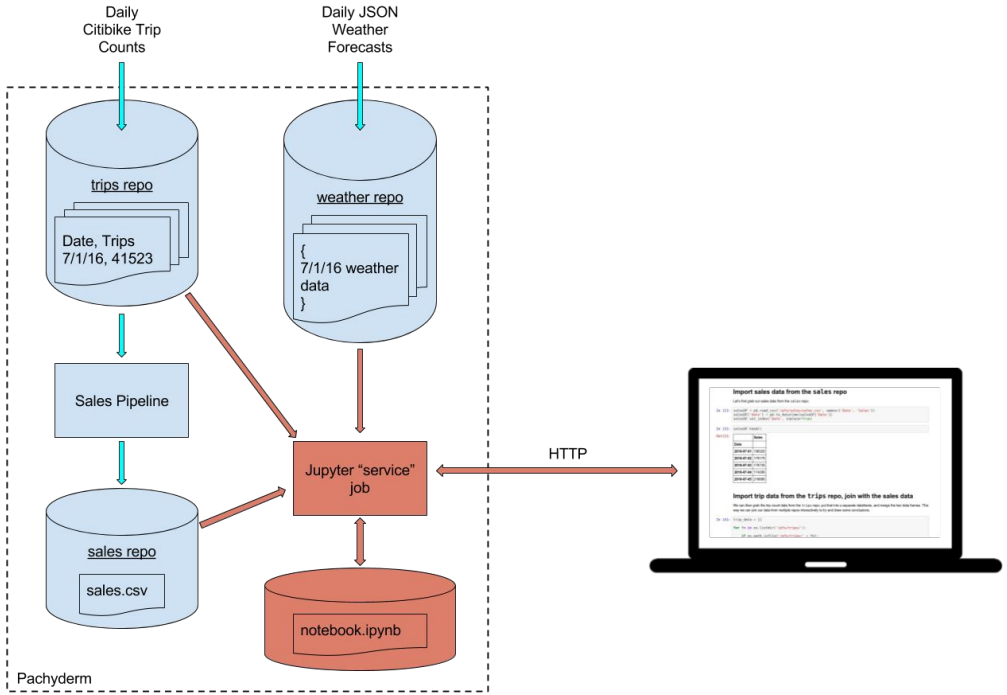
Image source: www.pixabay.com

But wait there’s more

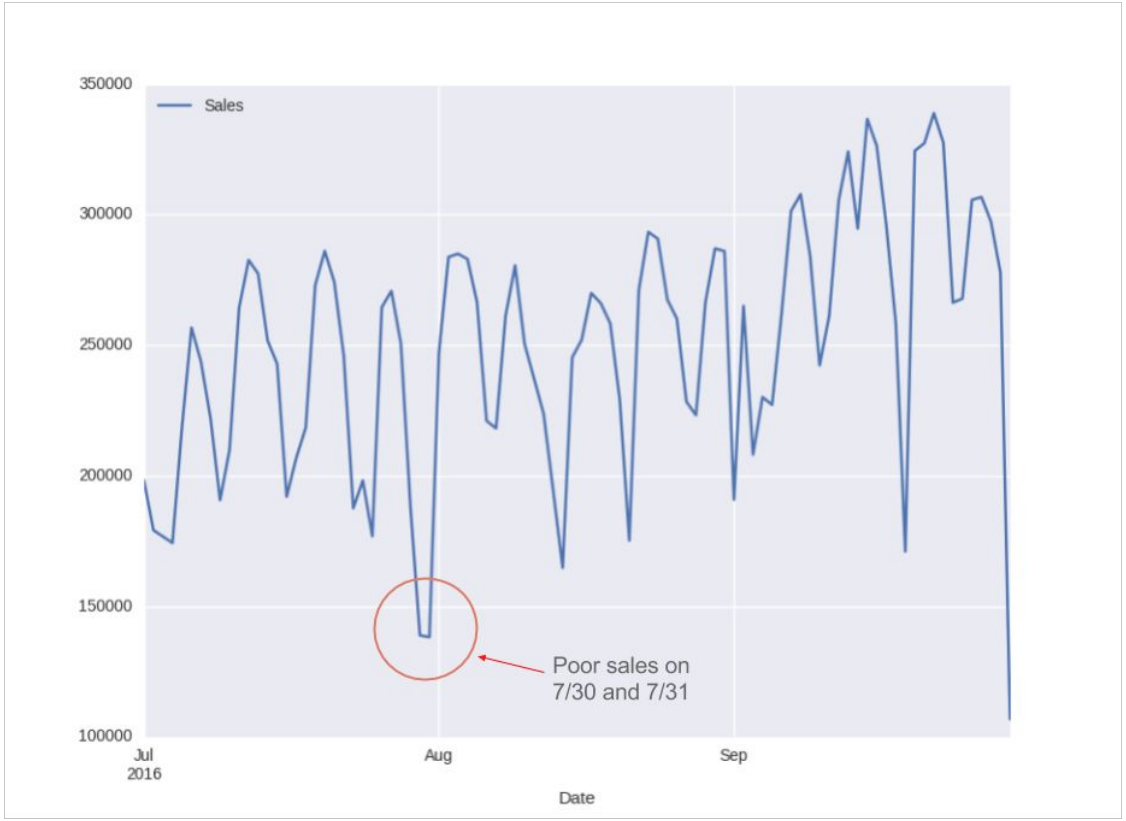
Image source: www.giphy.com

hook up to jupyter notebook to explore interactively

For full code see:
https://github.com/pachyderm/pachyderm/tree/master/doc/examples/jupyter_notebook



Why a drop in sales?



flush-commit to the rescue

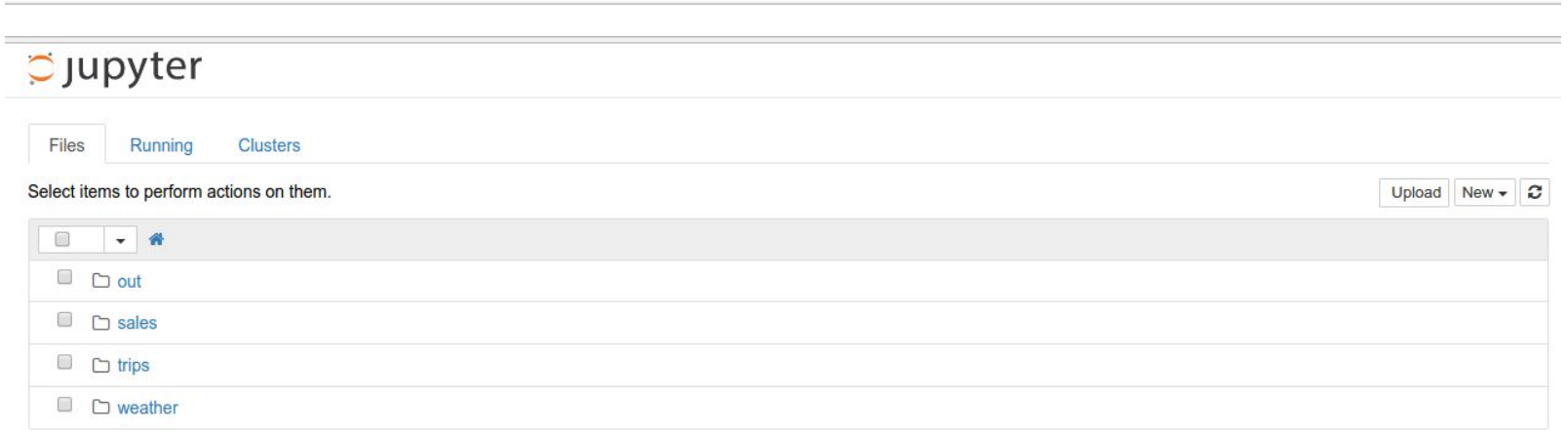
```
# get commit id

pachctl flush-commit trips/master/30

# put commit id into our jupyter json
```

```
 "transform": {
   "image": "pachyderm/pachyderm_jupyter",
   "cmd": [ "sh" ],
   "stdin": [
         "/opt/conda/bin/jupyter notebook"
{
    "commit": {
      "repo": {
          "name": "sales"
      },
      "id": "<output-commitid>/0"
```

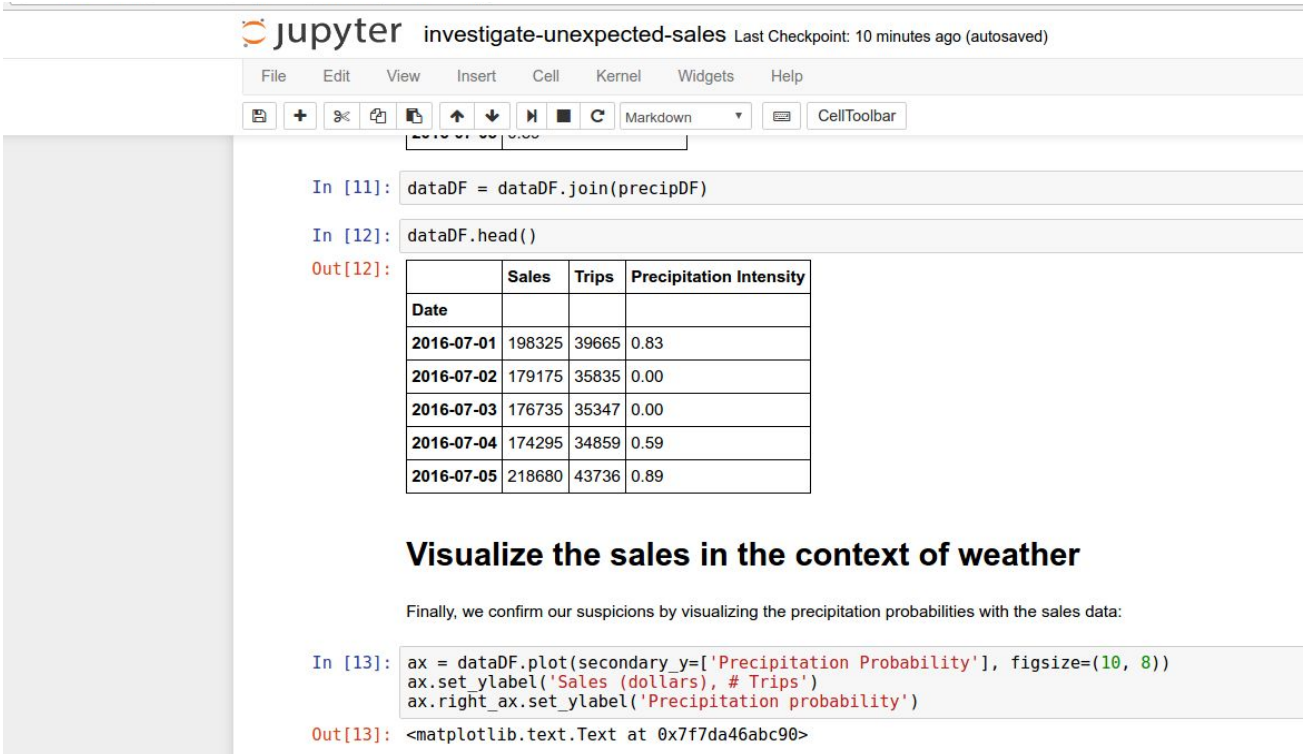
Run the container like a job

```
pachctl create-job -f jupyter.json
```

Access Jupyter at `http://localhost:8888` in a browser

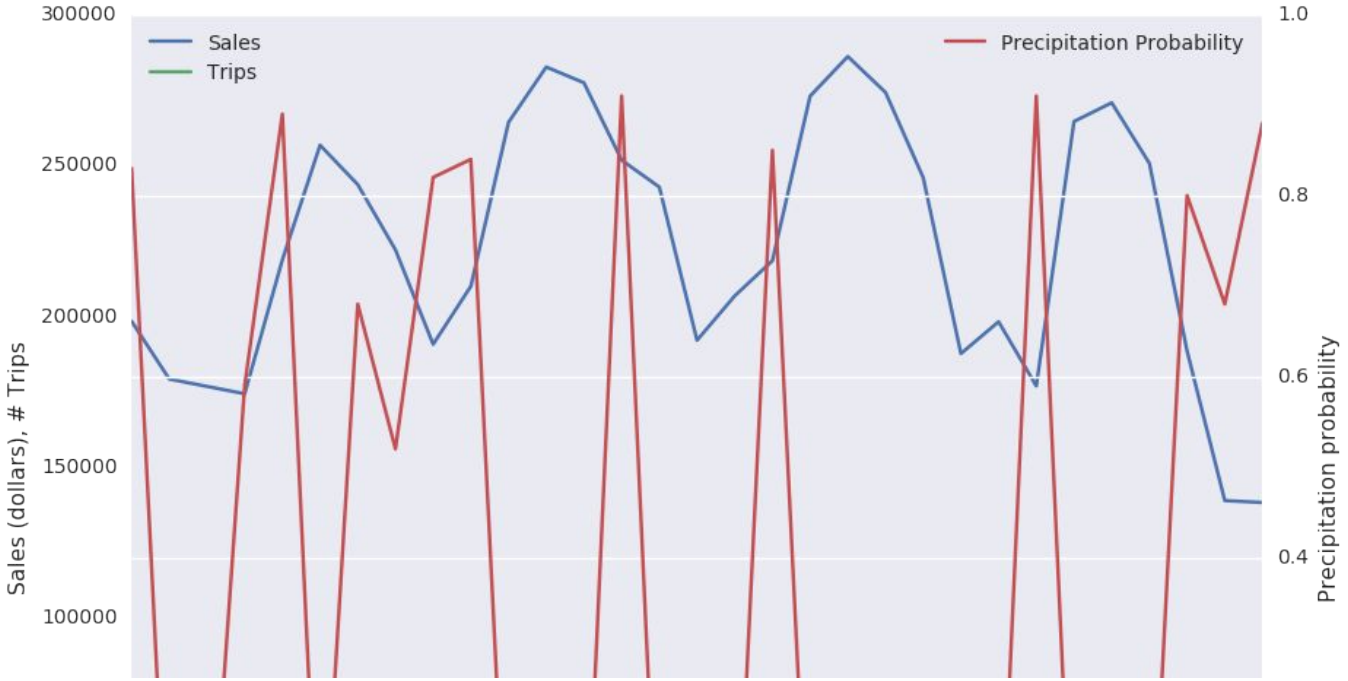


Notebook to investigate unexpected sales



	Sales	Trips	Precipitation Intensity
Date			
2016-07-01	198325	39665	0.83
2016-07-02	179175	35835	0.00
2016-07-03	176735	35347	0.00
2016-07-04	174295	34859	0.59
2016-07-05	218680	43736	0.89

High precipitation when sales dip - mystery solved!

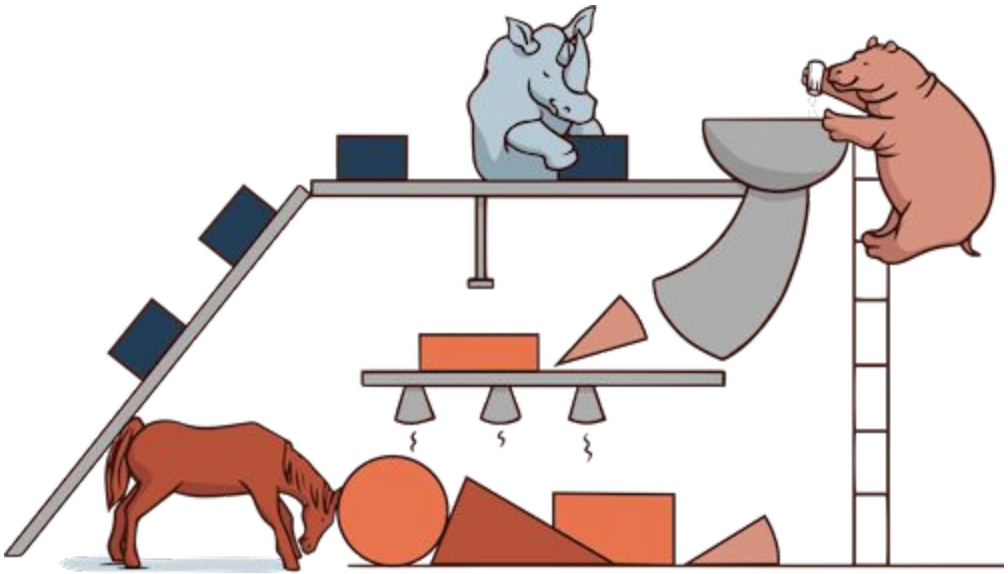


Summary

- Created reproducible workflow

- Accessed data snapshot to investigate dip in sales

Pachyderm and the Data Science Workflow



Thanks to www.pachyderm.io and the pachyderm community for the code and illustrative examples