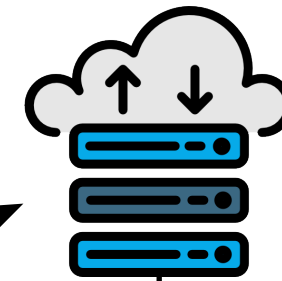Pull Data → Explore → Wrangle → Model → Explain → Deploy → Collect
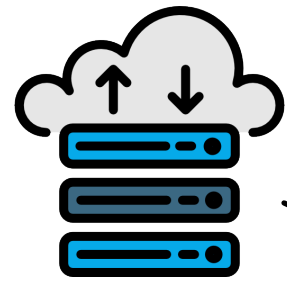
Software & Packages

VCS, e.g. Github

Data Scientist Workstation
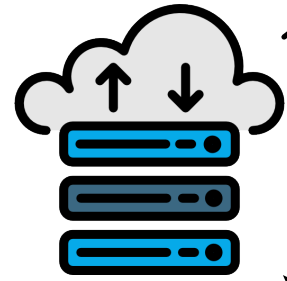
apt, pip

| Pull Data | → | Explore | → | Wrangle |

wget

| Model | → | Explain | → | Deploy |
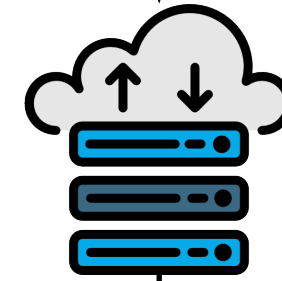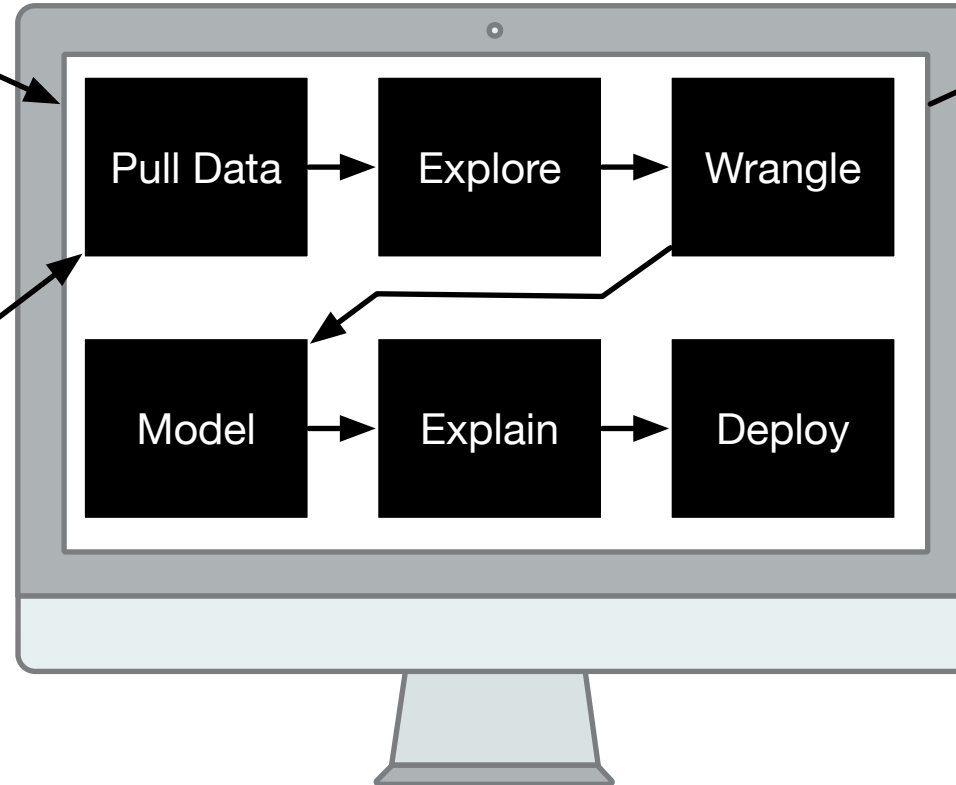
Download

CI / CD

Data
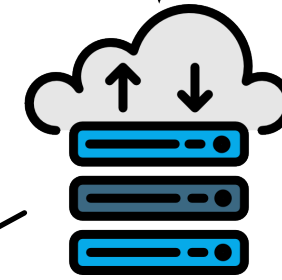
Capture Telemetry

Production Systems

api
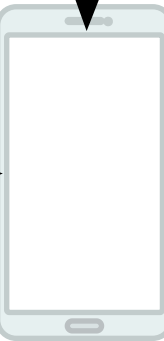
User

Notes:
* How many teams work
* Requires dev skills
* Package hell
* Poor Reproducibility
* Bound to desktop tools
* Limited compute

DataOps

DevOps
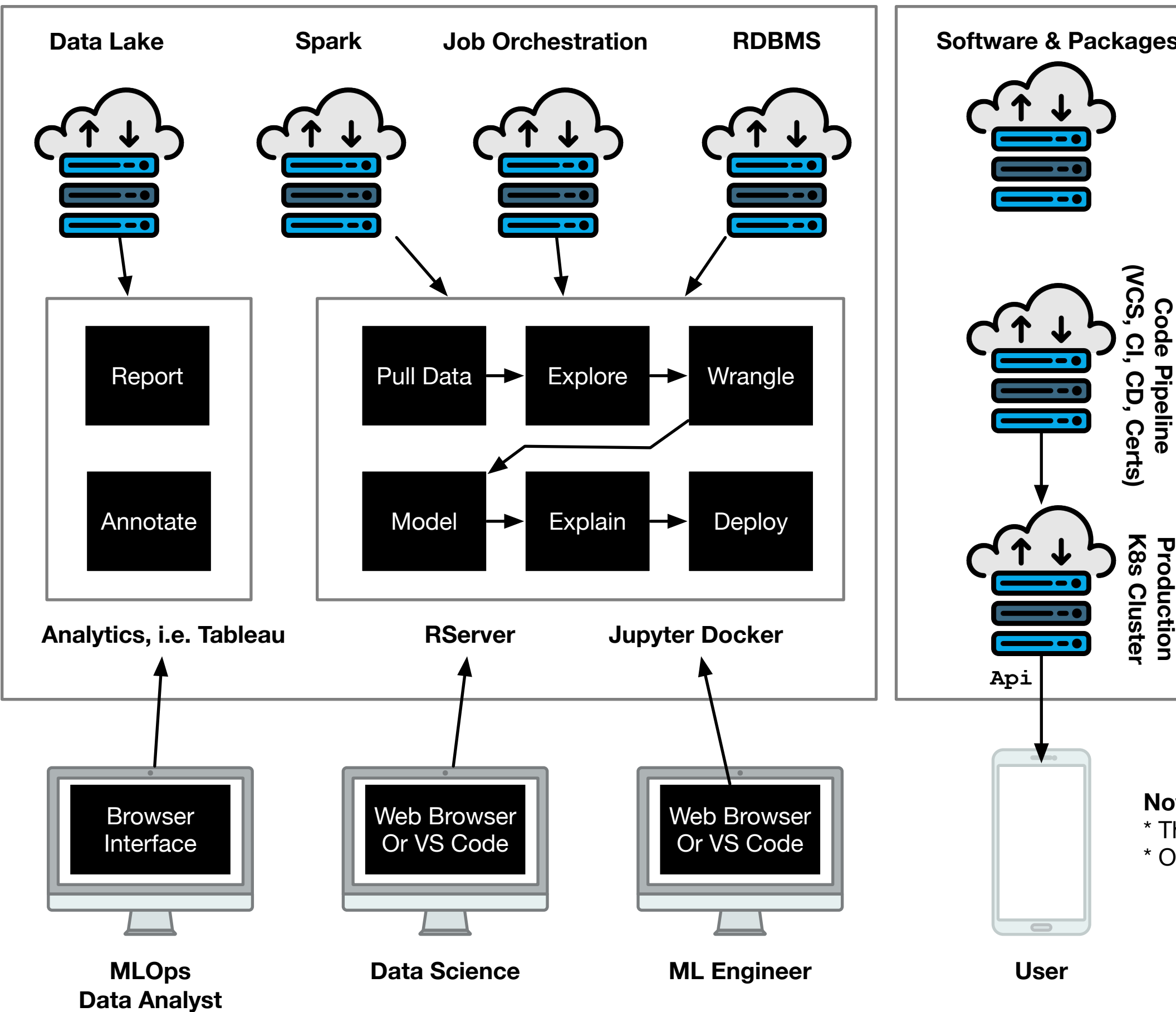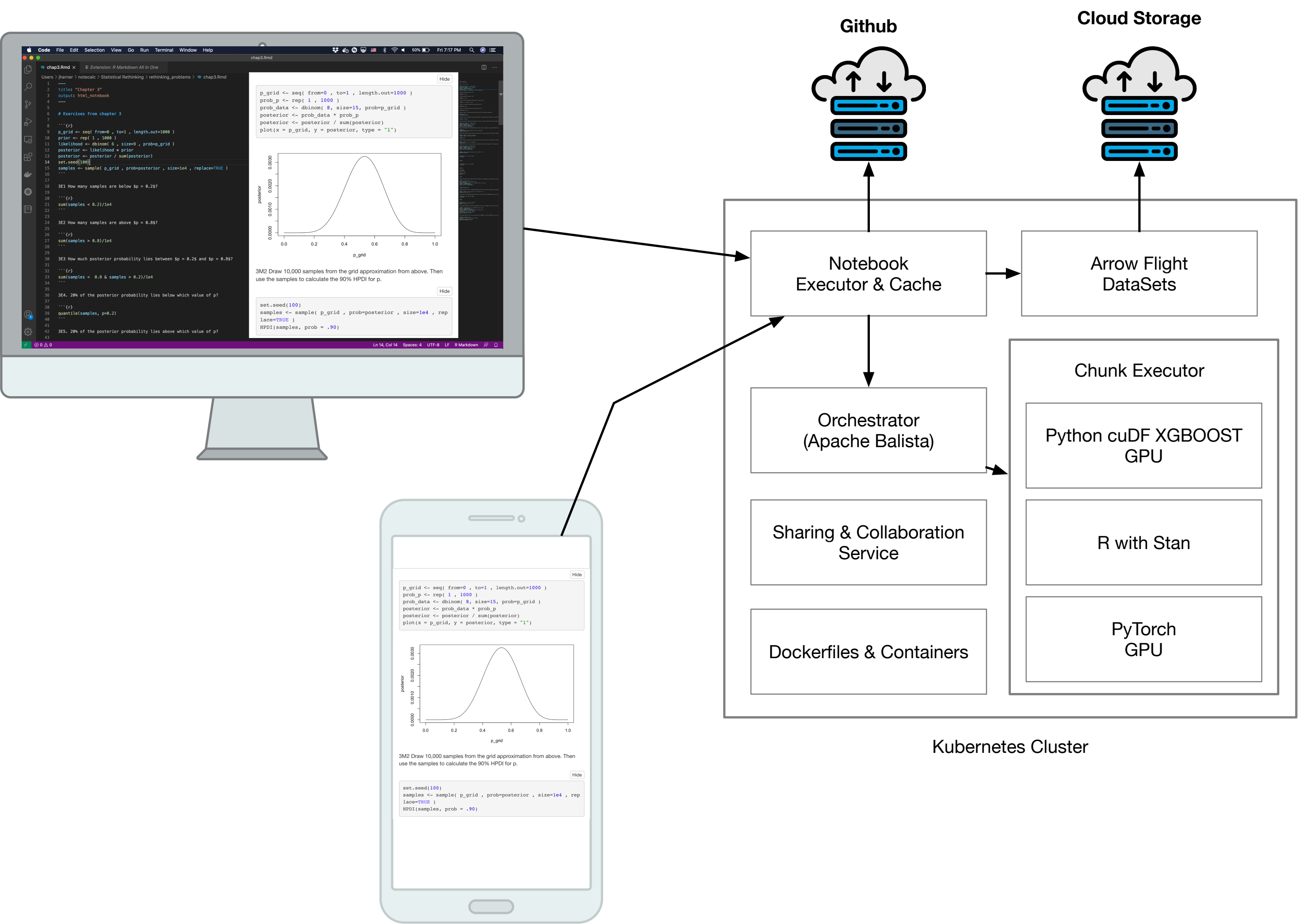
Data Lake   Spark   Job Orchestration   RDBMS

Software & Packages

| Report | | Pull Data → Explore → Wrangle |
| Annotate | | Model → Explain → Deploy |

Analytics, i.e. Tableau   **RServer**   **Jupyter Docker**

Code Pipeline (VCS, CI, CD, Certs)

Production K8s Cluster

`Api`

Browser Interface

Web Browser Or VS Code

Web Browser Or VS Code

MLOps Data Analyst   **Data Science**   **ML Engineer**   **User**

Notes:
* This is great but hard
* Orchestration isn't great

Github

Cloud Storage

Notebook
Executor & Cache

Arrow Flight
DataSets

Orchestrator
(Apache Balista)

Chunk Executor

Python cuDF XGBOOST
GPU

R with Stan

PyTorch
GPU

Sharing & Collaboration
Service

Dockerfiles & Containers

Kubernetes Cluster

```
prob_p <- rep( 1 , 1000 )
prob_data <- dbinom( 8, size=15, prob=p_grid )
posterior <- prob_data * prob_p
posterior <- posterior / sum(posterior)
plot(x = p_grid, y = posterior, type = "l")
```


3M2 Draw 10,000 samples from the grid approximation from above. Then use the samples to calculate the 90% HPDI for p.

Hide

```
set.seed(100)
samples <- sample( p_grid , prob=posterior , size=1e4 , replace=TRUE )
HPDI(samples, prob = .90)
```
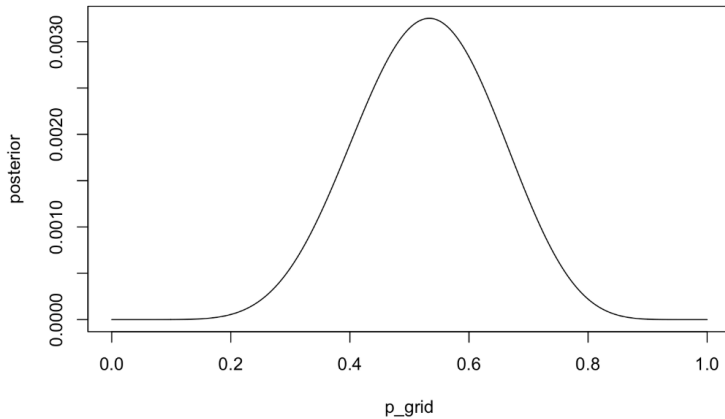
```
     |0.9      0.9|
 0.3343343 0.7217217
```

3M3 Construct a posterior predictive check for this model and data. This means simulate the distribution of samples, averaging over the posterior uncertainty in p. What is the probability of observing 8 water in 15 tosses?

Hide

```
dummy_w <- rbinom( 1e5 , size=15 , prob=samples )
simplehist( dummy_w , xlab="dummy water count" )
```

* Notebook has a unique endpoint
* Monoco Editor (from VS Code)
* Native Application
* Manage a set of Notebooks
* Interactive controls for parameters
* Edit mode
* Share with collaborators
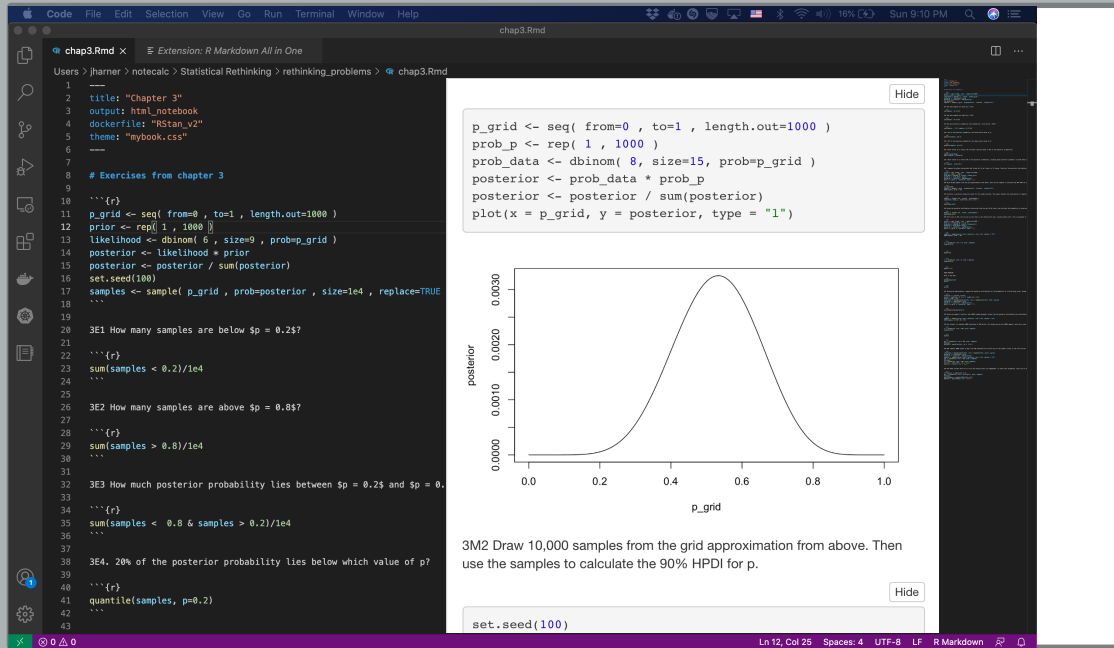
* VS Code Plugin
* Language Servers
- syntax high-lighting
- autocomplete
* CSS Themes
* Specify Docker File
* Code blocks run on server

```
chap3.Rmd
Extension: R Markdown All in One
```

Users > jharner > notecalc > Statistical Rethinking > rethinking_problems > chap3.Rmd

```
---
title: "Chapter 3"
output: html_notebook
dockerfile: "RStan_v2"
theme: "mybook.css"
---

# Exercises from chapter 3

```{r}
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prior <- rep( 1 , 1000 )
likelihood <- dbinom( 6 , size=9 , prob=p_grid )
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)
set.seed(100)
samples <- sample( p_grid , prob=posterior , size=1e4 , replace=TRUE
```

3E1 How many samples are below $p = 0.2$?

```{r}
sum(samples < 0.2)/1e4
```

3E2 How many samples are above $p = 0.8$?

```{r}
sum(samples > 0.8)/1e4
```

3E3 How much posterior probability lies between $p = 0.2$ and $p = 0.

```{r}
sum(samples <  0.8 & samples > 0.2)/1e4
```

3E4. 20% of the posterior probability lies below which value of p?

```{r}
quantile(samples, p=0.2)
```
```

Hide

```
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prob_p <- rep( 1 , 1000 )
prob_data <- dbinom( 8, size=15, prob=p_grid )
posterior <- prob_data * prob_p
posterior <- posterior / sum(posterior)
plot(x = p_grid, y = posterior, type = "l")
```

3M2 Draw 10,000 samples from the grid approximation from above. Then use the samples to calculate the 90% HPDI for p.

Hide

```
set.seed(100)
```