structuring your workspace: DS & DE/MLE perspectives

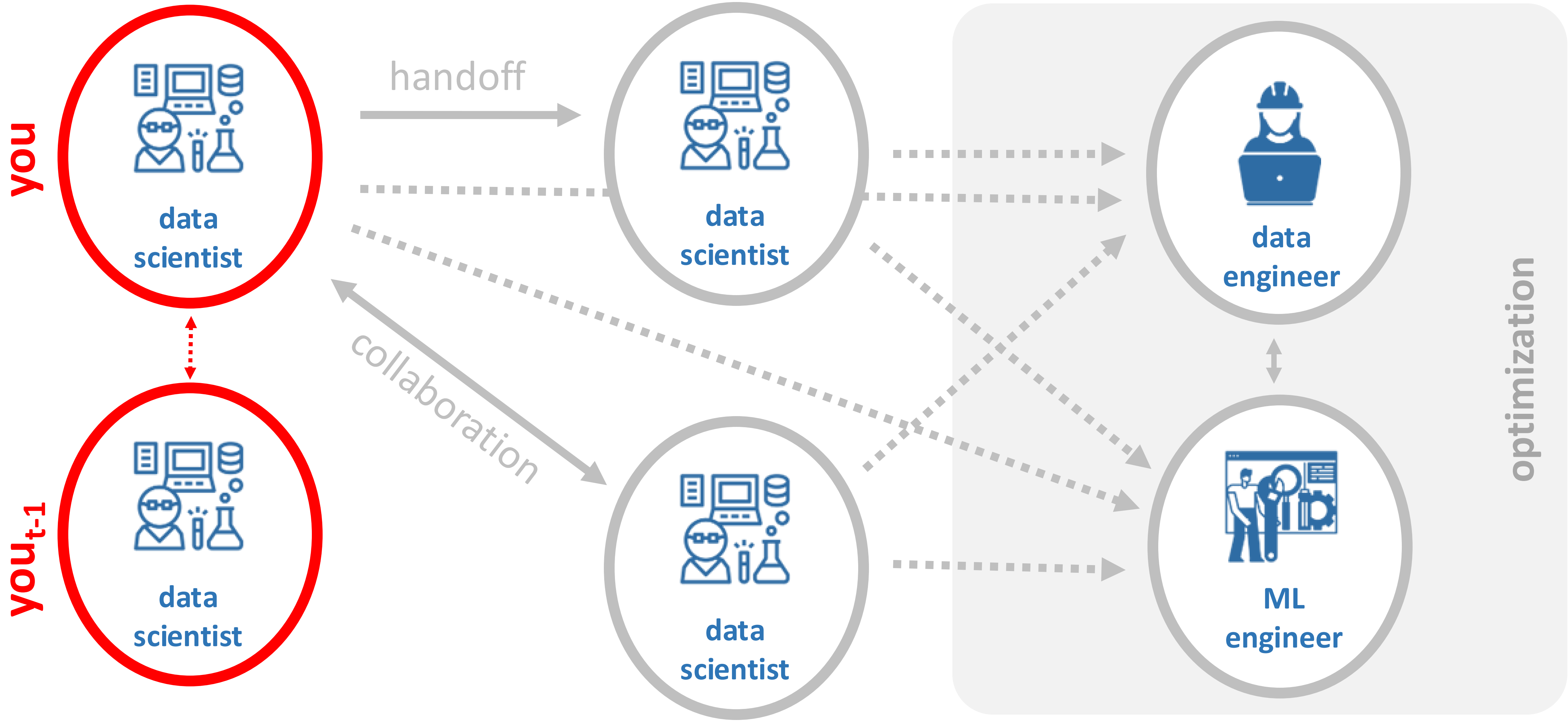
Marco Morales
marco.morales@columbia.edu

Nana Yaw Essuman
nanayawce@gmail.com

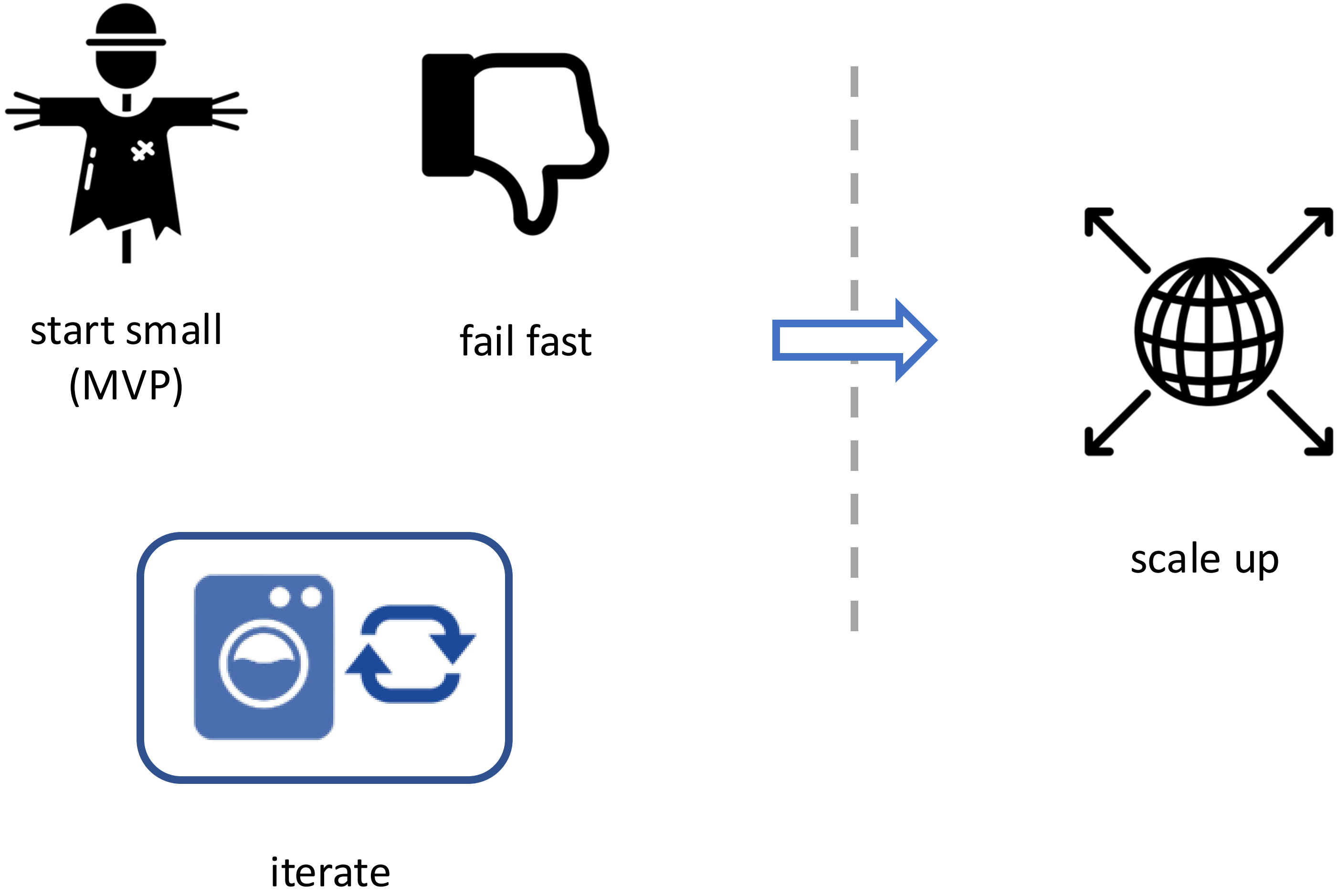
GR5069: Applied Data Science
for Social Scientists

Spring 2025
Columbia University

workflow collaboration in Data Science



recap: iteration to build Data Products



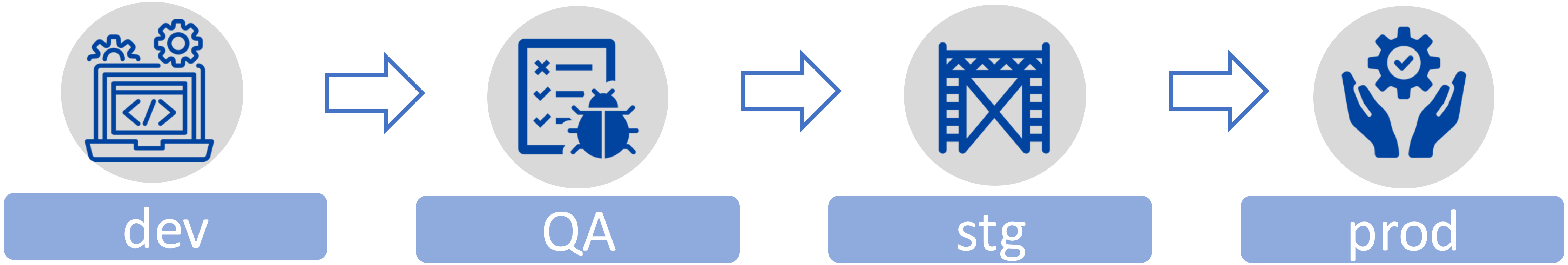
start small (MVP)

fail fast

iterate

scale up

working environments to build Data Products



dev

development

QA

quality assurance / testing

stg

staging

prod

production

operational concepts in Data Science

portability

anyone should be able to **pick up where you left off** from any machine

replicability

anyone should be able to arrive at your **same results**

scalability

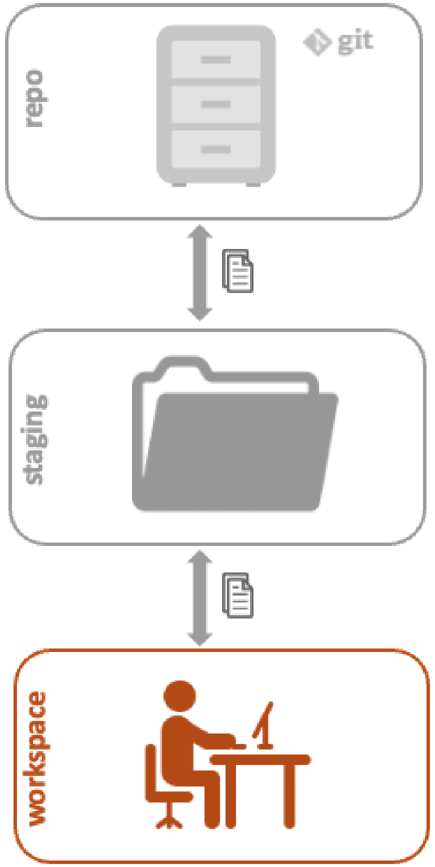
your prototype should also work for **larger data sets** and/or be on the path of **automation**

operational concepts in Data Science

	portability	replicability	scalability
what	• flexible references • structured and documented code • replicate original environment	• documentation: data, software, hardware, environments • commented code • no manual processes	• high quality code • flexible functions • modularized code
why	• seamless handoff • frictionless transitions across environments	• seamless examination, review or validation • cordial troubleshooting • harmonious optimization	• simplified review and validation • reduce time optimizing, automating and deploying

our focus today:

structuring your workspace



some **basic principles** for your workspace

1. use **scripts for everything** you do
   - **NEVER** do things **manually**
2. organize your scripts in a sequence
   - separate **activities** in sections
   - keep an early section for **definitions**
   - call **other scripts** when necessary
3. write **efficient** (aka lazy) **code**
   - turn code used multiple times into **functions**
   - **re-use** functions: make them generic enough
4. rely on **version control** (git)

some **portability** tricks

- use a **sensible folder structure** (more later)
  - create folder clusters aligned with purposes

- use relative paths in your scripts "data//external//ARCH535.csv" as opposed to "C://users//data//external//ARCH535.csv"

a **thin layer** to structure your workspace

```
workspace
|
| -- /src                  <- Code
|
| -- /data                 <- Inputs
|
| -- /reports              <- Outputs
|
| -- /references           <- Data dictionaries,
|                            explanatory materials.
|
| -- README.md
| -- TODO                 <- (opt)
| -- LabNotebook          <- (opt)
```

principle: separate function definition and application

```
workspace
|
| -- /src
|     |-- /data            <- code to read/munge raw data
|     |-- /features        <- code to transform/append data
|     |-- /models          <- code to analyze data
|     |-- /visualizations  <- code to create visualizations
|     |-- /functions       <- scripts to centralize functions
|     |-- /config          <- configuration files
|
| -- /data
|
| -- /reports
|
| -- /references
|
| -- README.md
| -- TODO
| -- LabNotebook
```

principle: separate function definition and application

- use src to organize your **code**
- use **one script per purpose**
- use **version control to "update"** your scripts
- use code to document **"manual" changes**
- call **additional scripts** as needed
- if too many functions, keep a **script with functions**

principle: input **raw data** and **its format and schema** is always immutable

```
workspace
|
| -- /src
|
| -- /data
|       |-- /raw              <- original, immutable data dump
|       |-- /external         <- data from third party sources
|       |-- /interim          <- intermediate transformed data
|       |-- /processed        <- final processed data set(s)
|
| -- /reports
|
| -- /references
|
| -- README.md
| -- TODO
| -- LabNotebook
```

principle: input **raw data** and **its format and schema** is always immutable

- **ALWAYS** keep your **raw data** as **immutable**
- keep **external data** separate and immutable
- if/when needed keep **interim data for validation**
- **processed data is ALWAYS replaceable!**
- all data should be linked to a script in src
- **document** origin of **raw & external data**

principle: outputs are disposable

```
workspace
|
| -- /src
|
| -- /data
|
| -- /reports
|      |-- /documents       <- documents synthesizing the analysis
|      |-- /figures         <- images generated by the code
|
| -- /references
|
| -- README.md
| -- TODO
| -- LabNotebook
```

principle: outputs are disposable

- use whichever document works best for your purpose: Jupyter notebooks, R Markdown

- **notebooks** can be **updated** and are **subject to change**

- use notebooks to **document deeper analysis/visualizations** in detail

principle: keep as much documentation as possible for your (future) reference and others’

```
workspace
|
| -- /src
|
| -- /data
|
| -- /reports
|
| -- /references           <- data dictionaries, explanatory materials
|
| -- README.md
| -- TODO
| -- LabNotebook
```

principle: document as much as you can about your session

```
R version 3.4.3 (2017-11-30)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: macOS High Sierra 10.13.2

Matrix products: default
BLAS: /System/Library/Frameworks/Accelerate.framework/Versions/(...)/A/libBLAS.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] bindrcpp_0.2    reshape2_1.4.3  stringr_1.2.0   lubridate_1.7.1 magrittr_1.5
 [6] dplyr_0.7.4     readxl_1.0.0    readr_1.1.1     here_0.1        tidyr_0.7.2

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.14     rprojroot_1.3-1  assertthat_0.2.0 plyr_1.8.4       cellranger_1.1.0
 [6] backports_1.1.2  stringi_1.1.6    rlang_0.1.6      tools_3.4.3      glue_1.2.0
[11] hms_0.4.0        yaml_2.1.16      rsconnect_0.8.5  compiler_3.4.3   pkgconfig_2.0.1
[16] bindr_0.1        tibble_1.3.4
```

in a nutshell…

```
workspace
|
| -- /src
|     |-- /data            <- code to read/munge raw data
|     |-- /features        <- code to transform/append data
|     |-- /models          <- code to analyze data
|     |-- /visualizations  <- code to create visualizations
|     |-- /functions       <- scripts to centralize functions
|     |-- /config          <- configuration files
|
| -- /data
|     |-- /raw             <- original, immutable data dump
|     |-- /external        <- data from third party sources
|     |-- /interim         <- intermediate transformed data
|     |-- /processed       <- final processed data set
|
| -- /reports
|     |-- /documents       <- documents synthesizing the analysis
|     |-- /figures         <- images generated by the code
|
| -- /references           <- data dictionaries, explanatory materials
|
| -- README.md            <- high-level project description
| -- TODO                 <- future improvements, bug fixes (opt)
| -- LabNotebook          <- chronological records of project (opt)
```

what actually gets pushed to GitHub

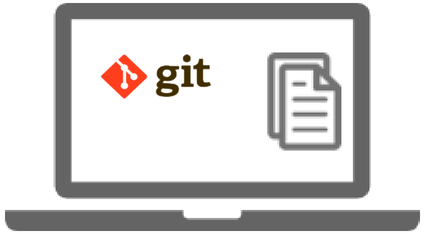
```
workspace
|
| -- /src
|     |-- /data              <- code to read/munge raw data
|     |-- /features          <- code to transform/append data
|     |-- /models            <- code to analyze data
|     |-- /visualizations    <- code to create visualizations
|     |-- /functions         <- scripts to centralize functions
|     |-- /config            <- configuration files
|
| -- README.md               <- high-level project description
```

what actually gets pushed to GitHub

- data is **NEVER** pushed to GitHub!!!!!!
- {secret keys} are **NEVER** pushed to GitHub!!!!!!
- reports could live in GitHub (depends)
- references are transferred to GitHub **wiki**
- TODO is transferred to GitHub **projects**

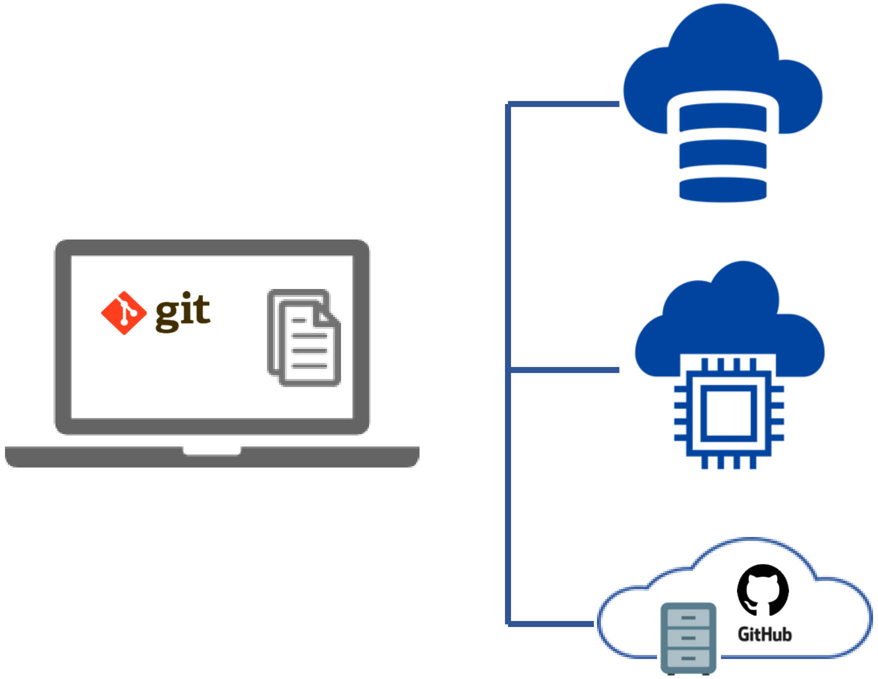
your workspace in real life

your workspace



local

also your workspace



data

computing

code

cloud-based

structuring your workspace: DS & DE/MLE perspectives

Marco Morales
marco.morales@columbia.edu

Nana Yaw Essuman
nanayawce@gmail.com

GR5069: Applied Data Science
for Social Scientists

Spring 2025
Columbia University