Week 1 - Course Introduction

QMSS 5072 Modern Data Structures

Marco Morales

January 20, 2022

Who am I

Marco Morales

Email: marco.morales@columbia.edu

Office hours: by appointment

Location: IAB 509E

Your TAs

Yingzhi Zhang
yz3988@columbia.edu

Parth Gupta
pg2677@columbia.edu


The Data in Data Science

Water, water, everywhere, nor any drop to drink.

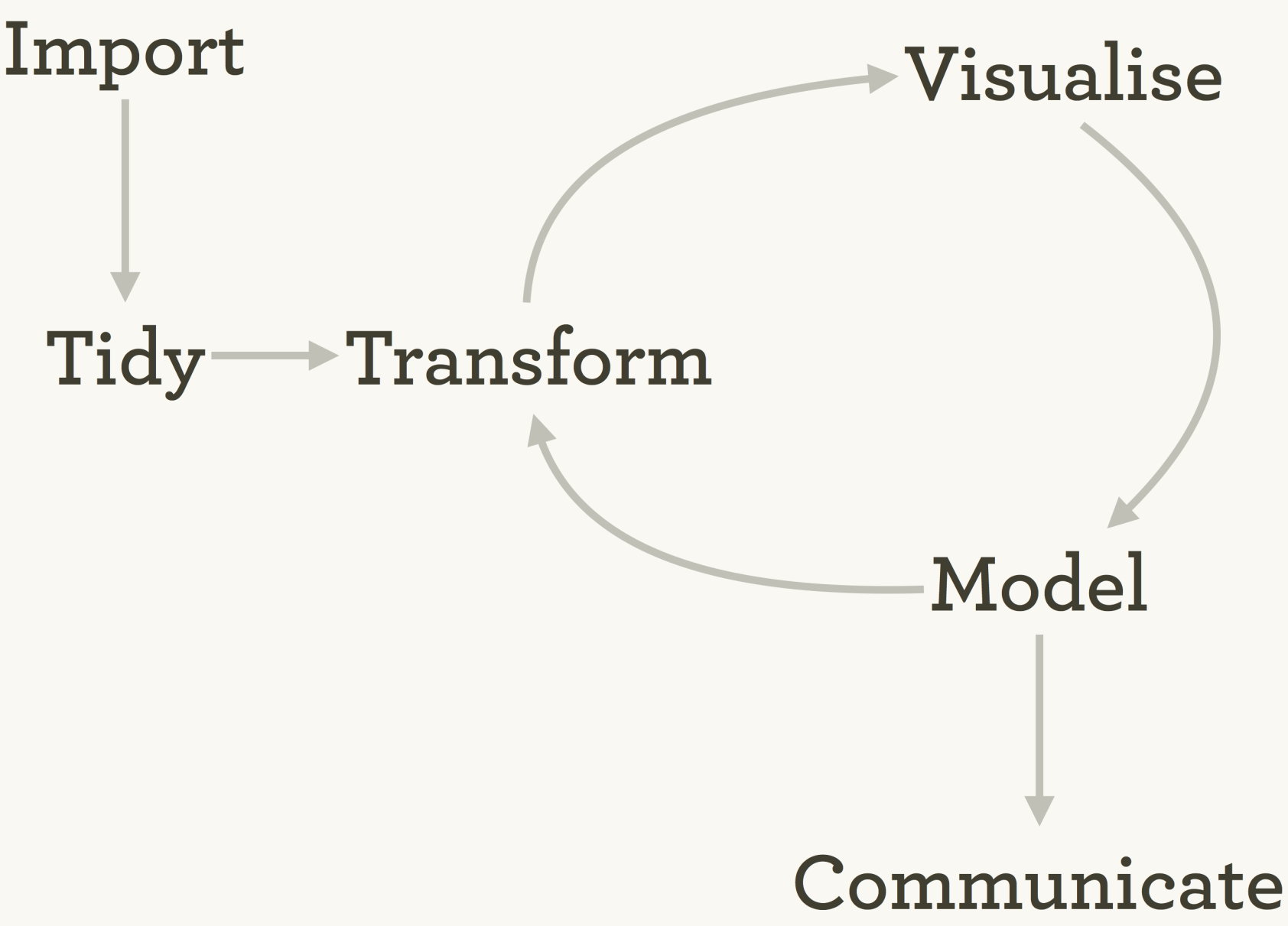
in The Rime of the Ancient Mariner, by Samuel Taylor Coleridge

Data, Data, everywhere, nor any thought to think.

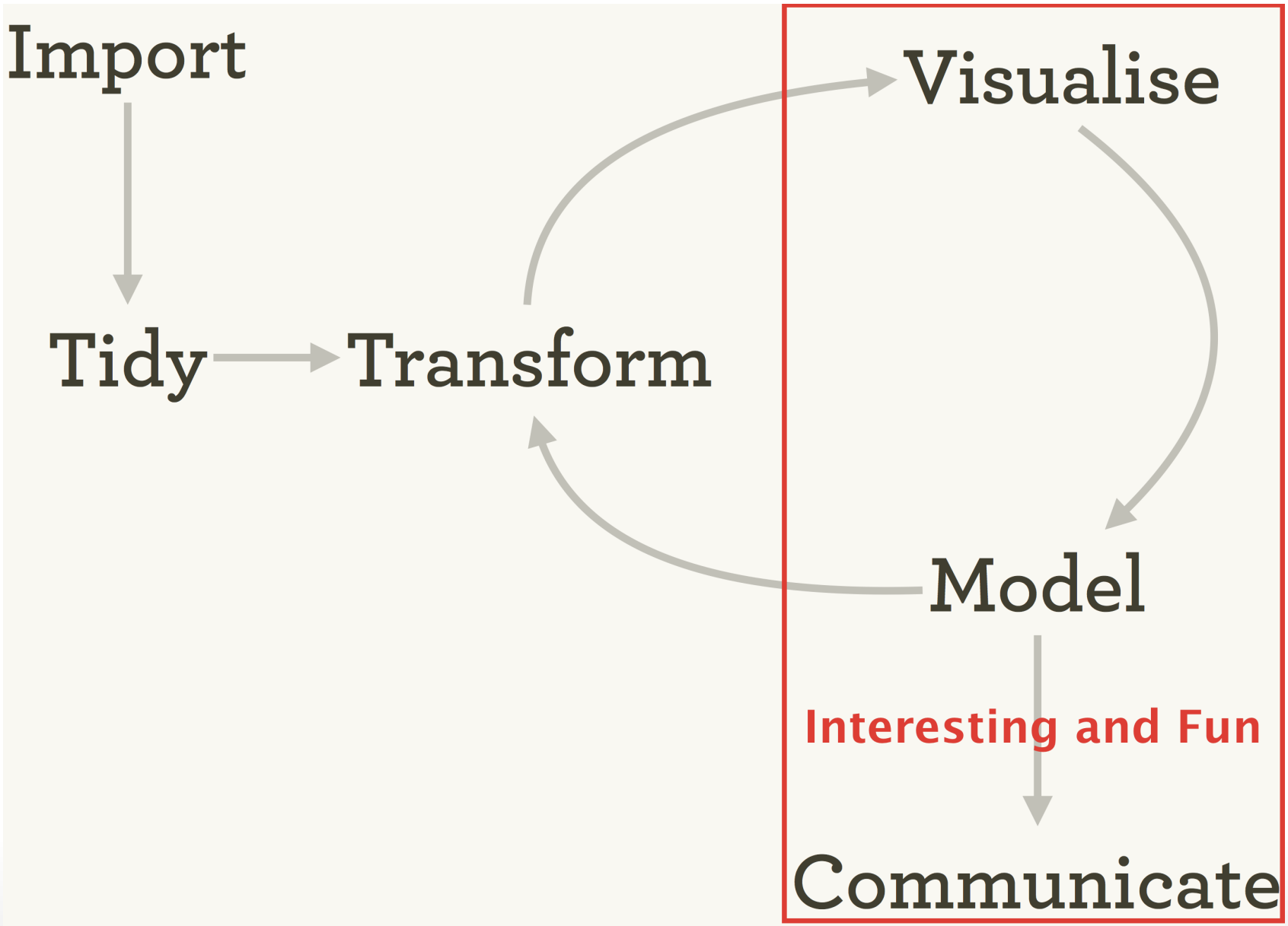
`random dude on twitter`

Data Wrangling

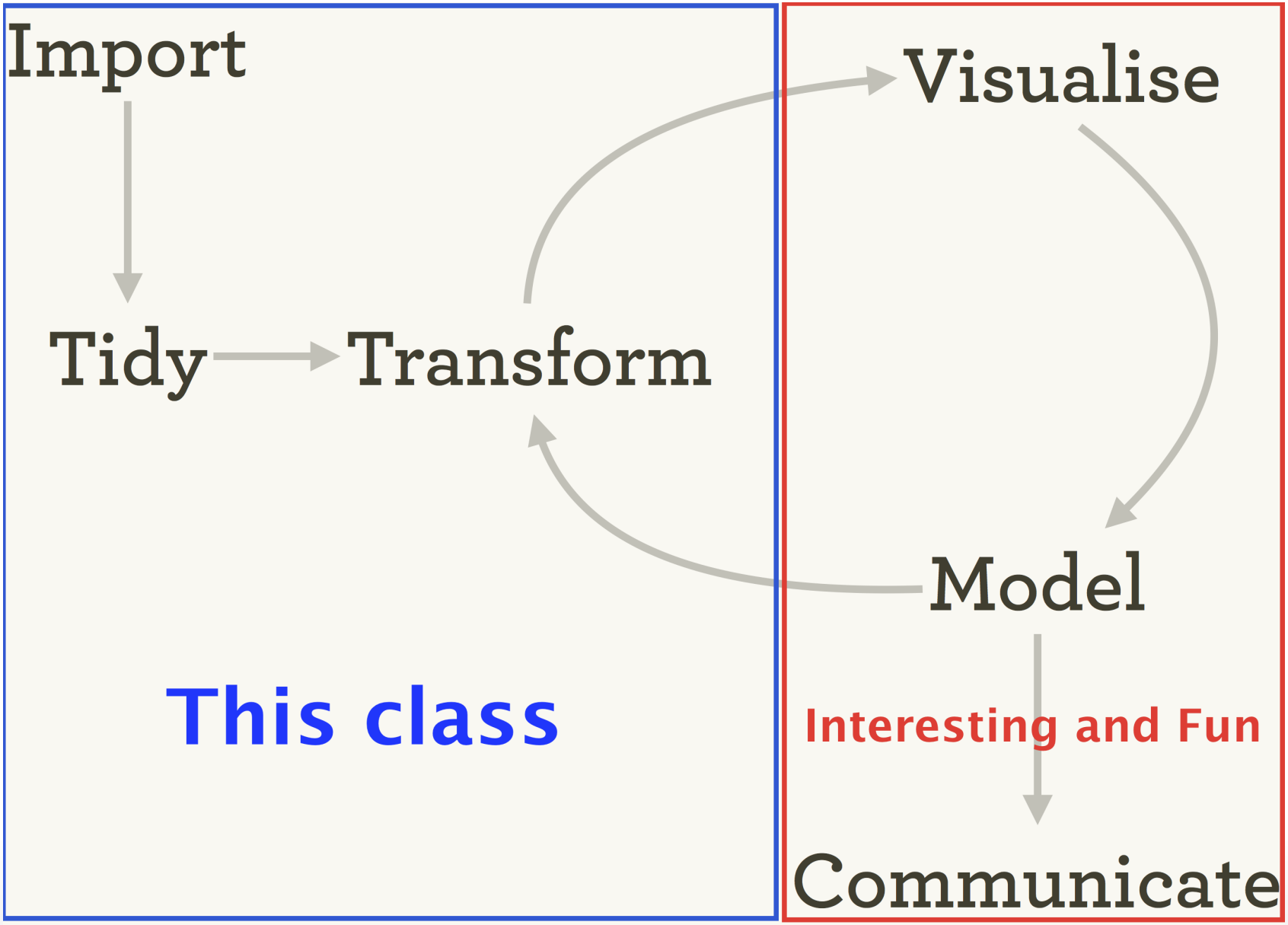
Analytical Process



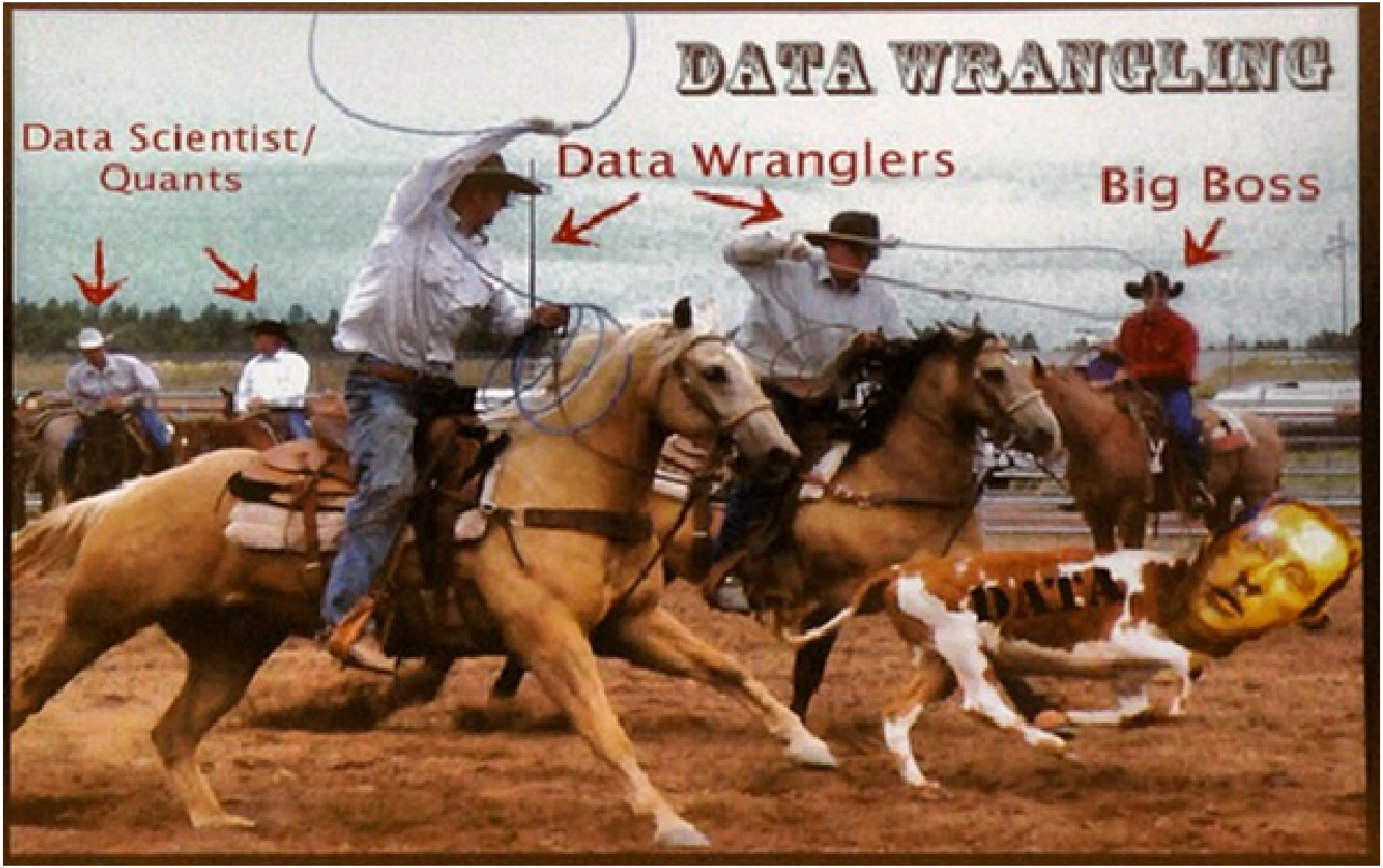
Analytical Process



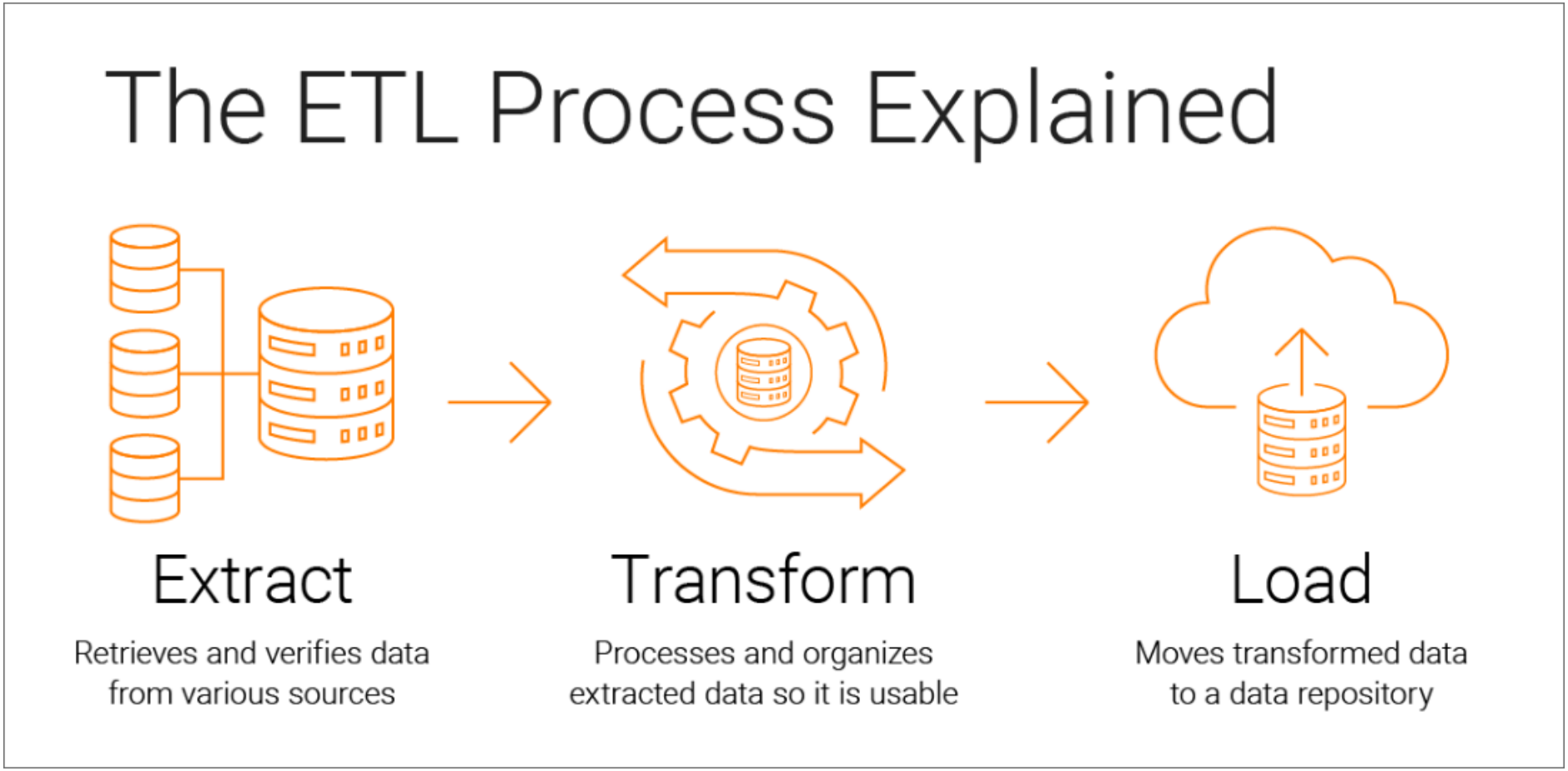
Analytical Process



Data Wrangling



ETL: the industry lingo



Course Outline

Subparts of the course

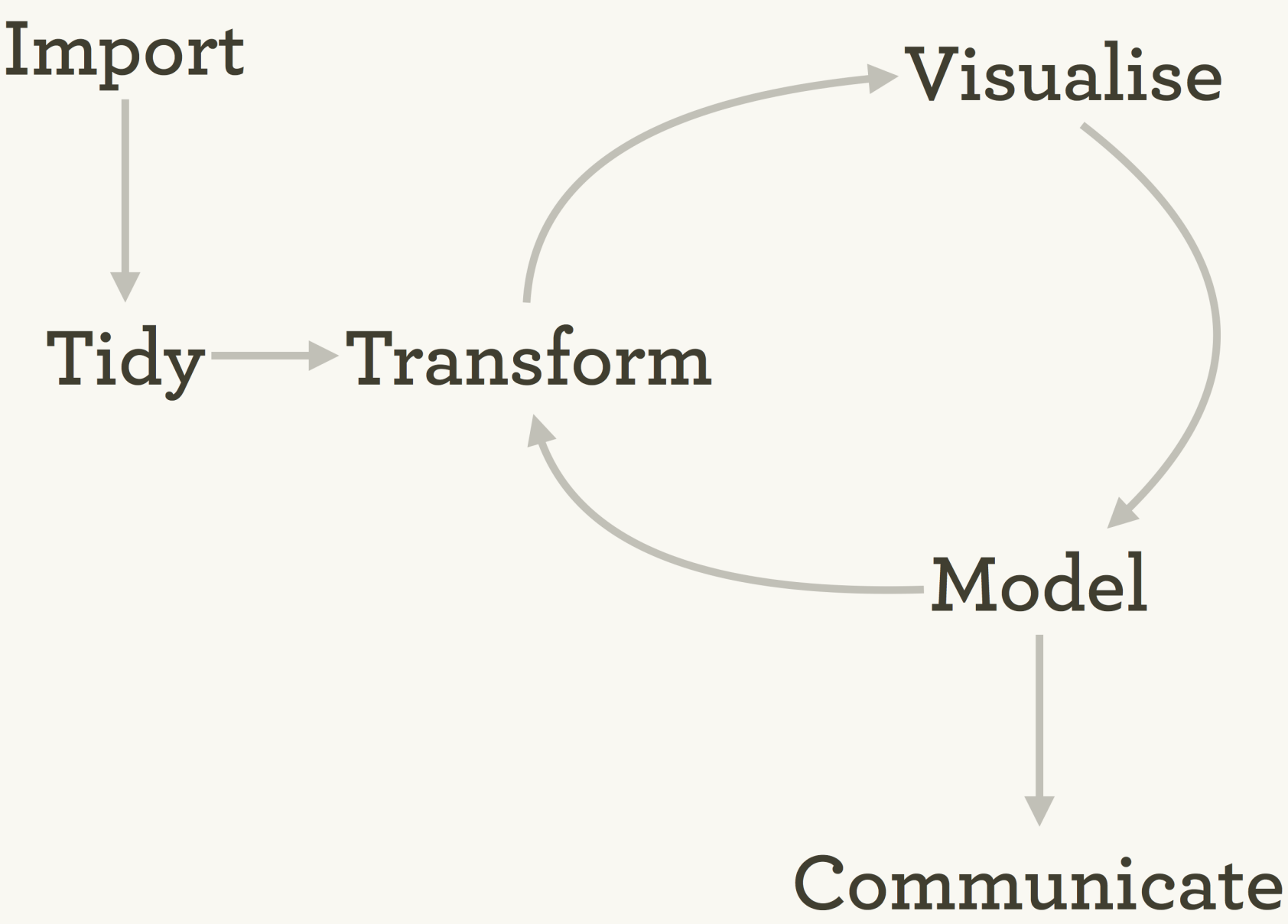
- Part 1 - Data Manipulation
- Part 2 - Getting Data In
- Part 3 - Some big data considerations

Part 1 - Data Manipulation

Git and Github. Why?

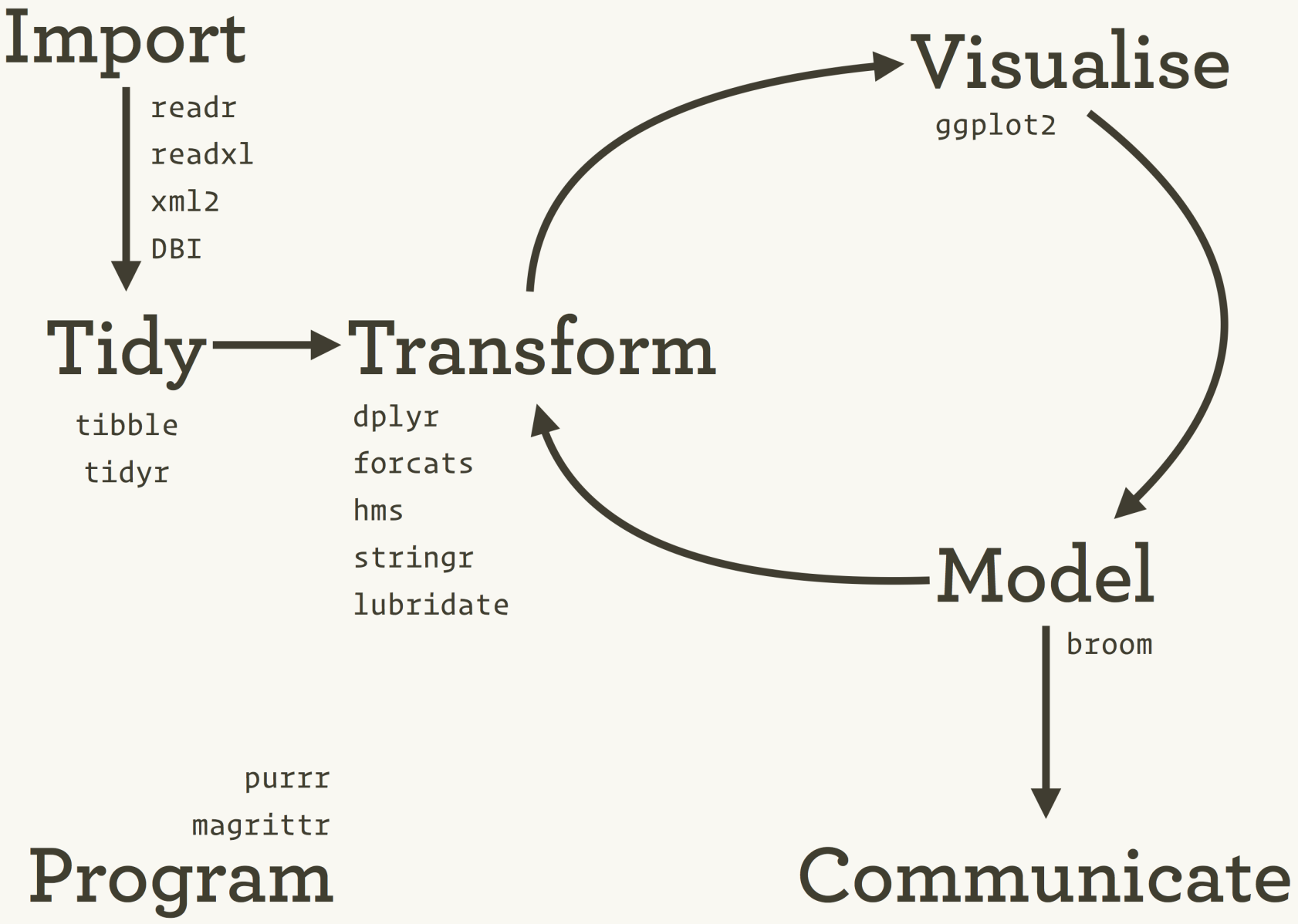
- version control is good
- cornerstone of collaboration
- enables reproducibility, portability, scalability
- industry standard
- we'll use it heavily - in class and for homeworks

Basics of the `tidyverse`



Design for humans — Hal Abelson

Basics of the `tidyverse`



Basics of the `tidyverse`

Import:
- `readr`: import different kinds of rectangular data (e.g. csv, tsc, fwf) in a fast and friendly way; `readxl` and `xml2` for special types

Tidy:
- `tidyr`: reshape the layout of dataframes into a specific type, the `tibble` – a *tidy* data frame

Basics of the `tidyverse`

Transform:
- `dplyr` provides function to manipulate and transform data frames.
- Includes select, filter, group, summarize, arrange, mutate, join etc.

Data Types:
- How to work with the different types of data such as numerics, characters (`stringr`), factors (`forcats`), and dates (`lubridate`)

What we won't cover in the `tidyverse`

- visualization (`ggplot2`)
- communication of models (`broom`)

Functions and Functional Programming

Functions and Functional Programming

- `dplyr` imports the `%>%` operator from the `magrittr` package. - `x %>% f(y)` is equivalent to `f(x, y)`.
- Easy to combine multiple operations into a readable chain of commands.

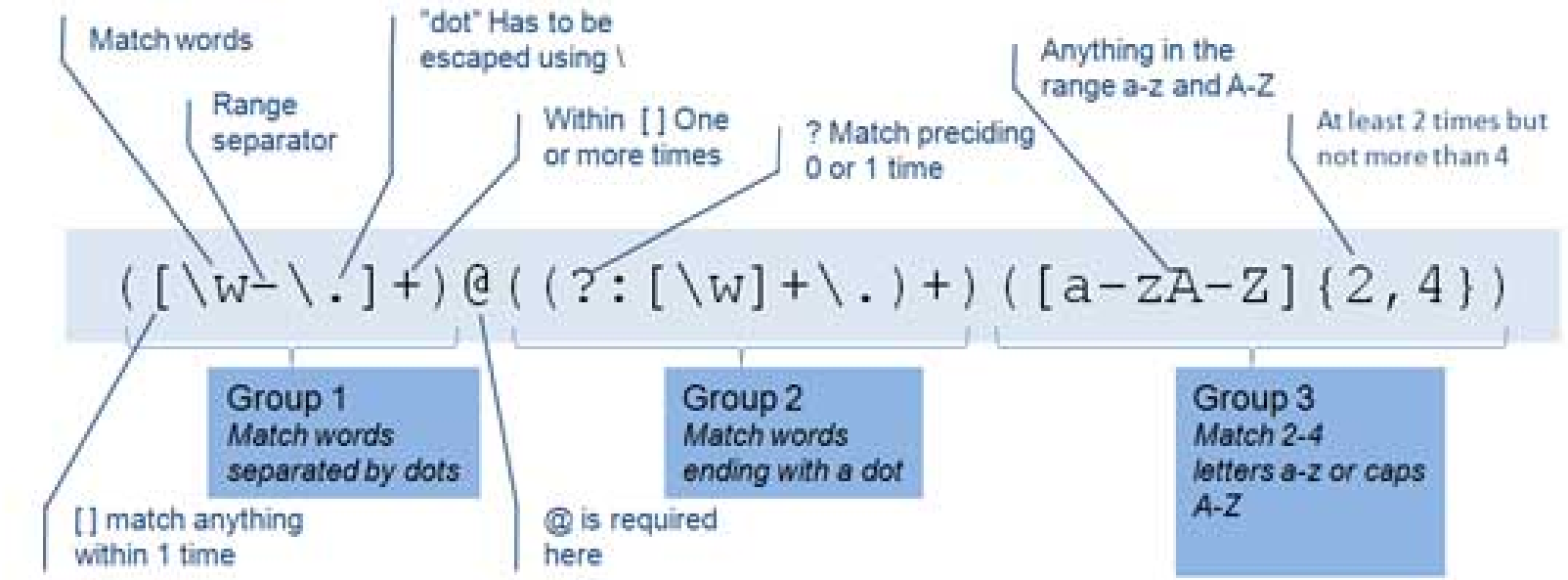
Functions and Functional Programming

Functions and Functional Programming

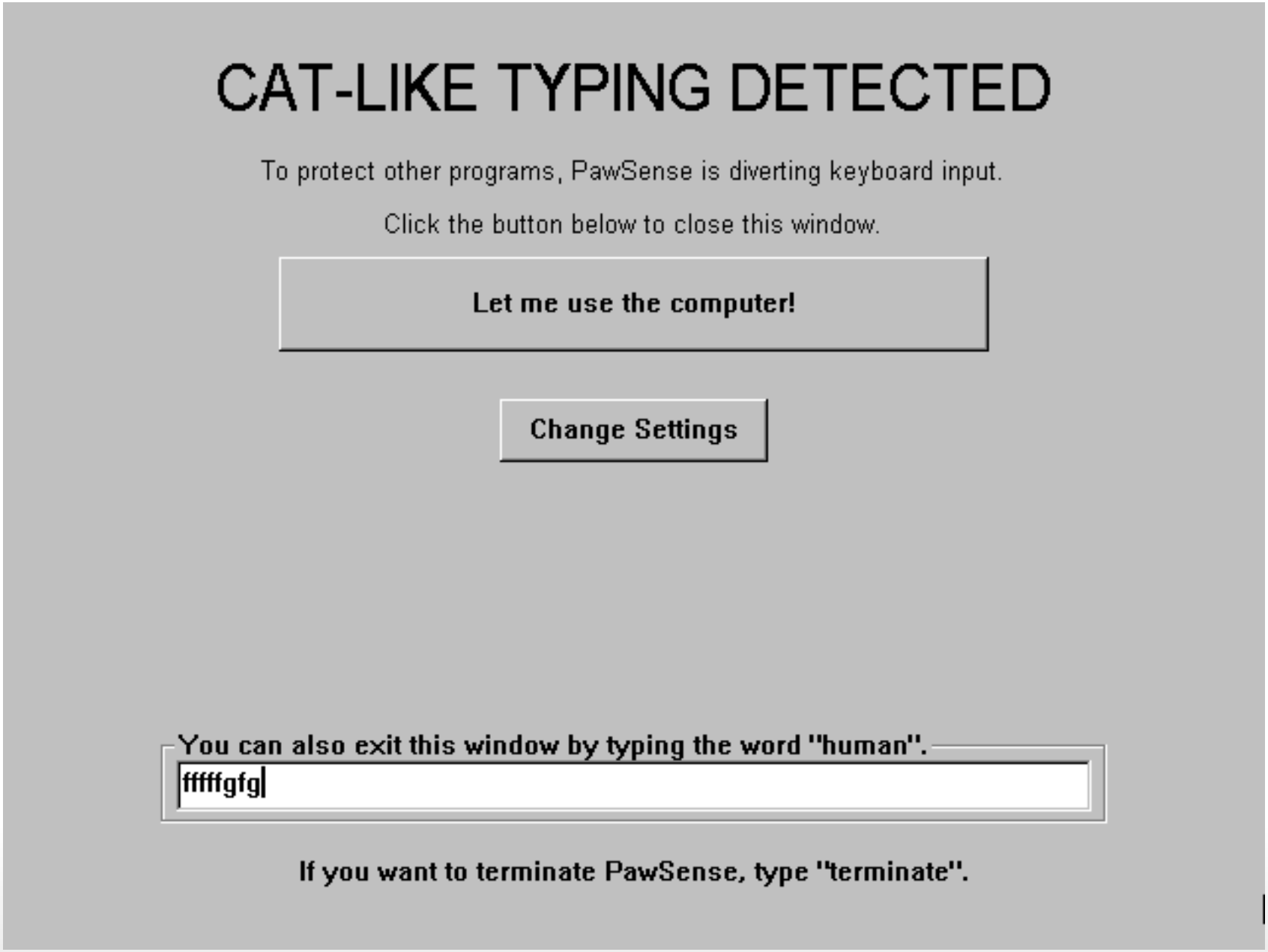
- avoid repetitive code by using functions
- functional programming:
  - Functions of functions
  - Anonymous functions
  - List of functions
  - Function operators

Free Expressions and Strings

- strings usually contain unstructured or semi-structured data
- handling and processing strings with `stringr`
- using regular expressions - a concise language for describing patterns in strings.



Regular Expressions



Part 2 - Getting Data In

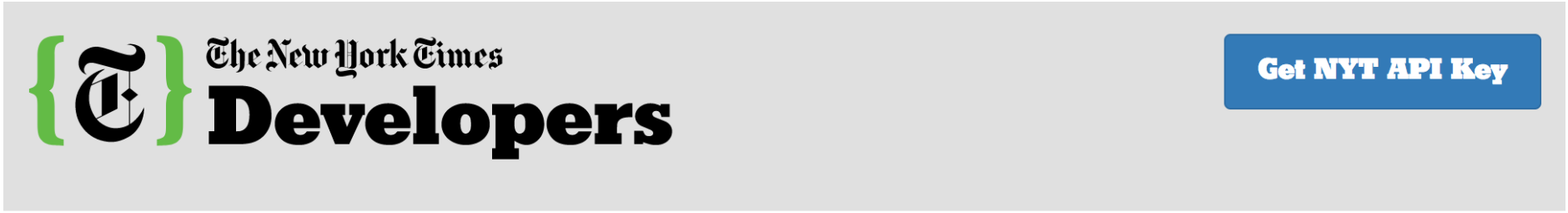
Getting Data In

Getting Data In

- importing data from a single file on disk (or stored online) is only one way of getting data
- we will explore other types of data

APIs

Using the `httr` package (a wrapper for `curl`) to access some well-known web APIs



The New York Times Developer Network

All the APIs Fit to POST

You already know that NYTimes.com is an unparalleled source of news and information. But now it's a premier source of data, too — why just read the news when you can hack it?

Handling JSON and XML

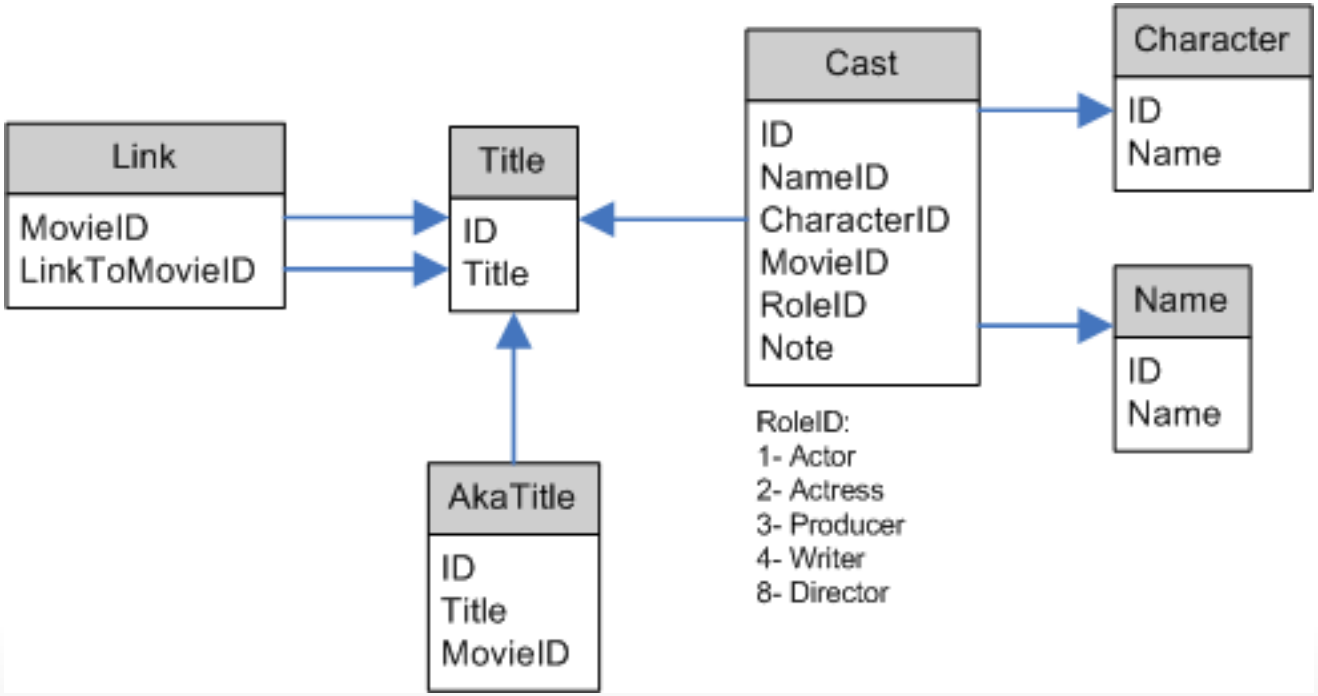
- APIs usually return data in JSON or XML format as exchange format
- We will learn how to deal with these formats in R and transform them into rectangular data formats.

Screen Scraping from HTML

- Screen scraping refers to a type of computer program that:
  - reads in a web page
  - finds some information on it
  - grabs the information
  - stores it in a data set
- But HTML is messy. Will need to select the right elements and clean it up.
- Old school way of getting information. Many websites do not allow it anymore (TOS) and/or make it difficult.

Relational Databases

- databases consisting of multiple tables of data are called **relational data** because it is the relations, not just the individual data sets, that are important.
- **SQL** allows to interact with such databases to modify, insert, remove, or request data
- we can use R to interact with SQL databases directly

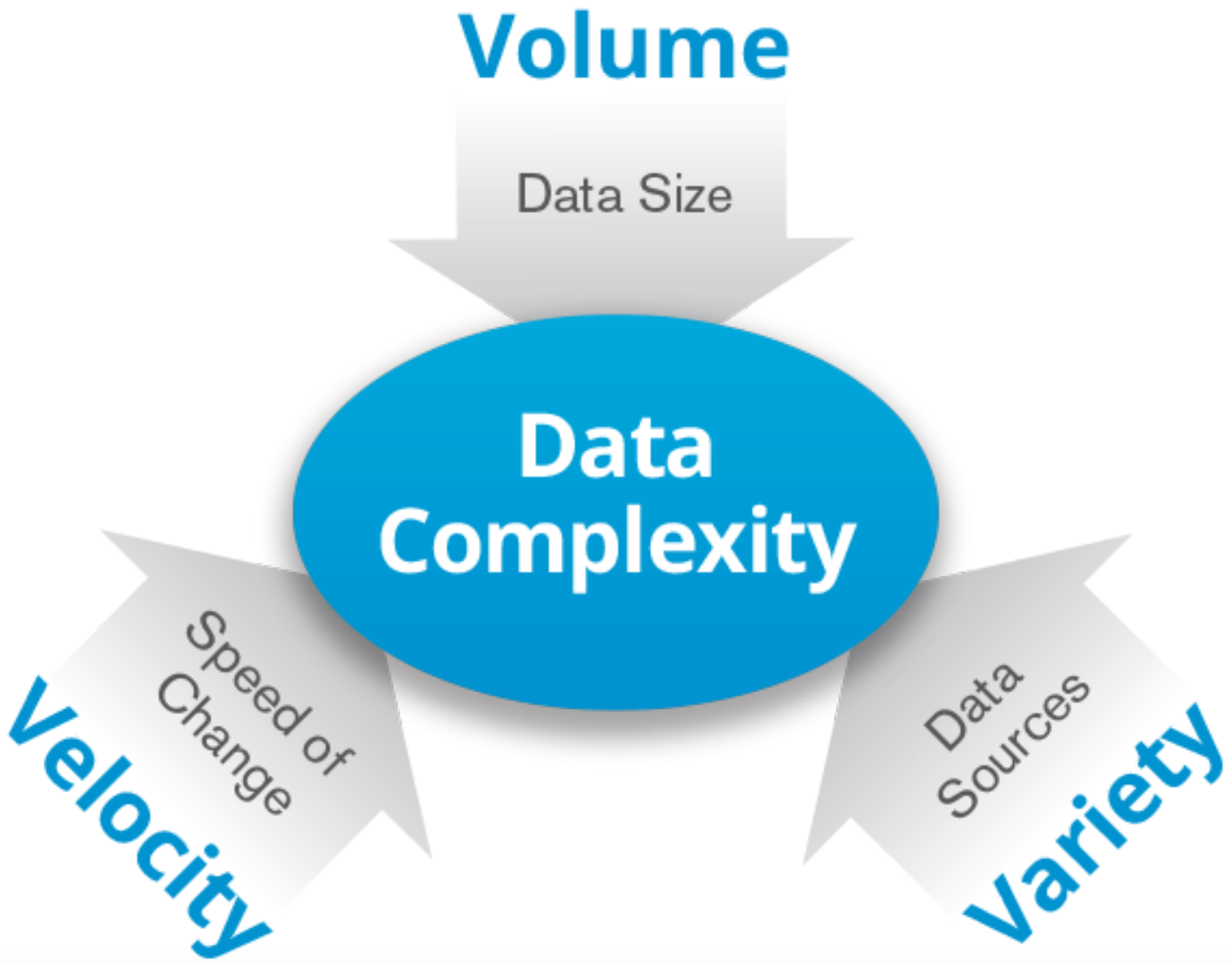


Part 3 - Other Big Data Considerations

“Big data is a term describing the storage and analysis of large and or complex data sets using a series of techniques including, but not limited to: NoSQL, MapReduce and machine learning.”

- Jonathon Ward and Adam Barker, University of St. Andrews

What is big data?



What do we need to know?

- computing power becomes a real constraint
- **efficiency** and **structure** matters (more)
- Can we use R to analyze big data?
  - bigger hardware
  - piece wise analysis
  - sampling
  - parallelization
  - higher performing programming languages like C++ or Java

Some Administrative Things -Lectures

- Thursdays 4:10-6:00 pm
- Don’t be late (it disrupts class and concentration).
- Bring a laptop (but check your social media at home).

Course materials

- **weekly curated materials** on each topic on the course's **GitHub repo** (all available online)
- weekly **RMarkdown notebooks** through **GitHub classroom** - run code + add notes in class + `push` to the repo for class participation!
- **take-home exercises** through **GitHub classroom**

Course communications

- all course-related communications through a dedicated **Slack workspace** for this course:
  - asks questions, help answer questions
- **e-mail** will be reserved for **official communications only**!! (repeat after me!)

Course dynamics - preparing for each class

Materials

GitHub  curated materials

Classroom  homework exercises

Activity

 read

 code

Submission

 questions/ comments

slack

HW repo

Classroom

canvas

} class participation

Course dynamics - live session

Materials

Classroom  **in class exercises**

Classroom  **in class exercises**

Activity

 live workshop

 live code in local machine

Submission

 questions/ comments

 R Markdown notebook

Classroom

class participation

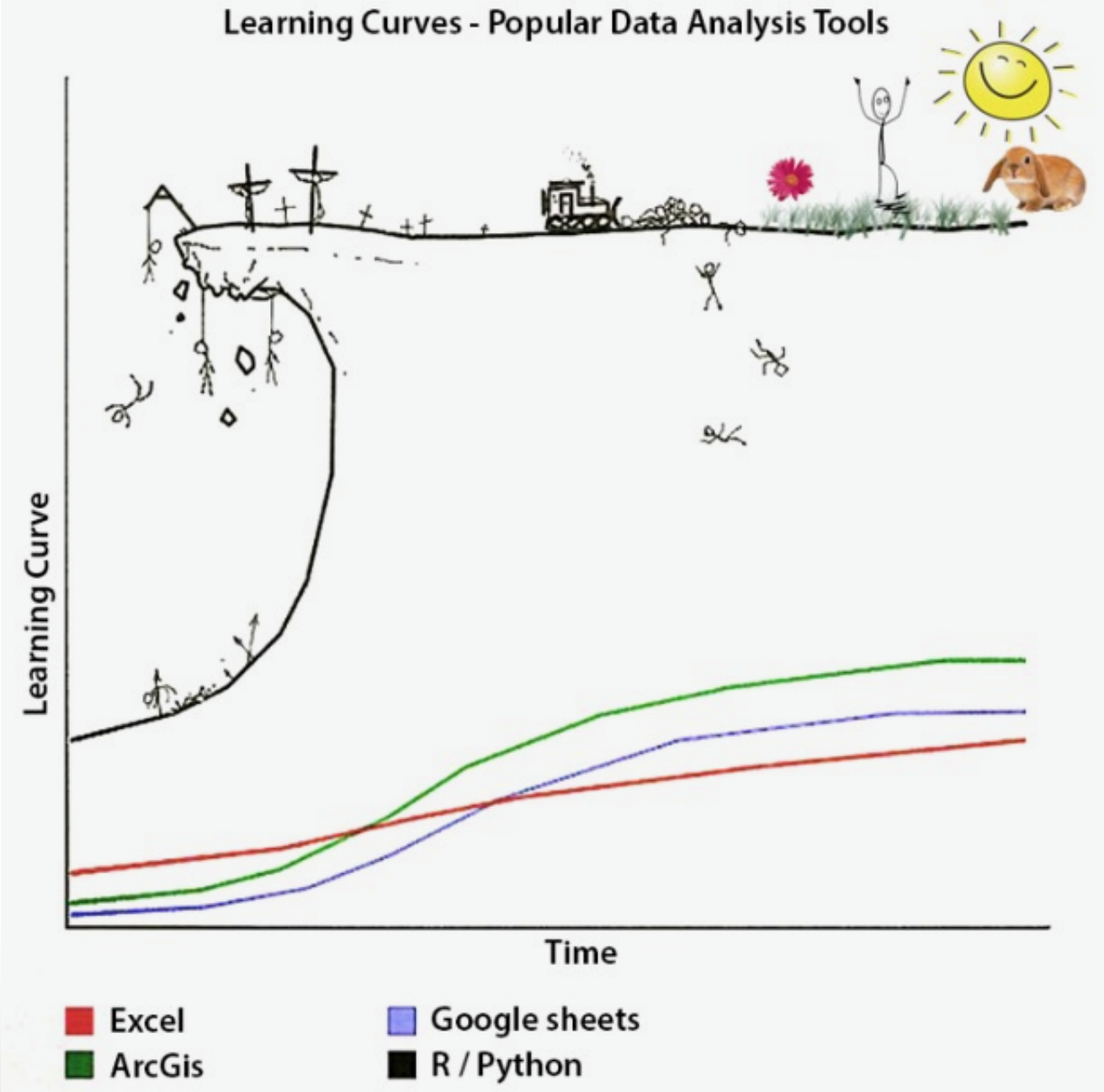
Course requirements

- **Final exam (30%)**
- **Take-home exercises (50%)**: submit through GitHub classroom
- **Participation & Attendance (20%)**: ask/answer questions on Slack + submitting your annotated `.md` notebooks each class

Is this the right class for me?

SOURCE: "The Untapped Power of Self-Service Data Analytics", Harvard Business Review

Should I take this class?



Should I take this class?