Data Science For Front-end Engineers

DataFest Africa 22 Uganda

Alecho Edwin | 2022

Who am I?

I am Alecho Edwin. I am a lawyer and Digital Solutions Architect.

I am just a **curious** chap that loves to solve problems using web-technologies usually in **JavaScript** and **Rust** at **Alaara.**

At Alaara, we are a **curious collective** of **creatives solving** interesting p**roblems** for **SMEs using** web-technologies.

Who is this talk for?

Every **lesson** should aim to meet the needs of a **specific learner**. The intended audience for this talk is the **Front-end Engineer.**

Who is a Front-end Engineer?

A Front-end Engineer **architects** and **develops websites** and **web applications** using **web technologies** (i.e., **HTML**, **CSS**, and **JavaScript**), which typically run on the Open Web Platform or act as compilation input for non-web platform environments (i.e., React Native).

The Typical Front-end Engineer

- Is skilled to some degree in **HTML**, **CSS**, **DOM**, and **JavaScript** and implements these technologies on the web platform.
- Appreciates the elusive gotchas in javaScript. E.g the **execution context** and how it relates to the **this** object.
- Works closely with **designers** to make **websites beautiful**, **functional**, and **fast**.

We are very inclusive.

If you know what a **variable**, a **function**, an **if statement** and **loops** are, have some grit and is willing to learn. Please **feel at home**, everything will make sense to you. If not now at least eventually.

Why Use JavaScript for Data Science?

- You already know **JavaScript**.
- JavaScript is a **capable** language.
- Strong **visualization ecosystem**. (Ploty, D3)
- Generally strong ecosystem. (one of the strongest user-driven eco-systems our there)
- JavaScript is **everywhere.** (server, microwave, automotive, etc.)
- JavaScript is easy to **learn**. (Apparently developed in ten days)
- JavaScript **programmers** are **easy to find**.
- JavaScript is **evolving**. (Community Driven and led eco-system, **NPM**)
- JavaScript and **JSON** (**data format** of the web)

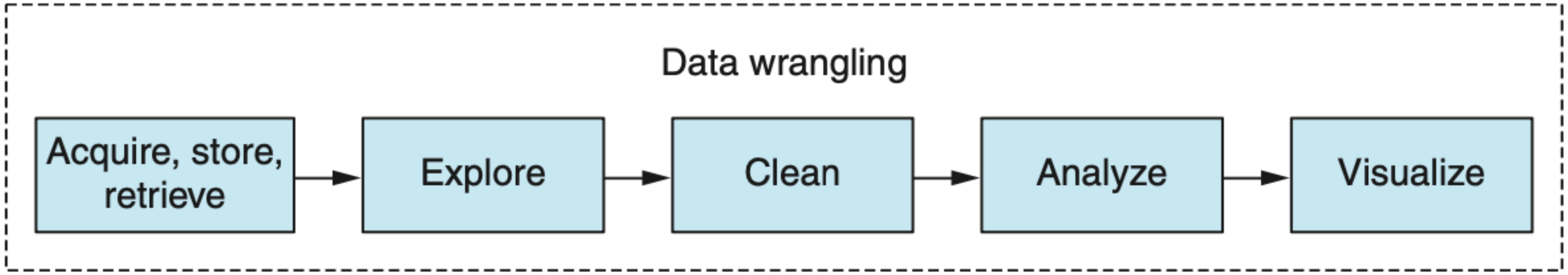
Still not convinced?

JavaScript is **the language** of the **Web**.

It's all about **communication.**

Data Management

So let us figure out what our Data processing (**Data wrangling**) pipeline looks like.



Data formats

We use to common **data formats** when processing data in JavaScript.
Comma Separated Values and **JSON** (JavaScript Object Notation)

Comma Separated Values (CSV)

Most common and widely used type of Data format. Each **row** is a line in the file, and each **column** is separated by a comma. The first line of the file is the **header row**. **favSongs.csv**

```
"Name","artist","year","rating"
"kachumbali","Quex",2019,5
"feedibacka","Malcom",2020,4
"Sirikawo Retouch","Malcom&Kloud DIpo",2021,5.7
```

JSON

CSV is good for tabular data, but a lot of data doesn’t neatly fit into **rows** and **columns**. So **JSON** is best suited. **JSON** can be used for tabular data as well. The entire table is an Array, and each **record** or **row** is an **object** with **name-value pairs.**

```
const favSongs = [
{"name": "kachumbali", "artist": “Quex”, "year": 2019, "rating": 5},{"name":
"feedibacka", "artist": “Malcom”, "year": 2020, "rating": 4},{"name": "Sirikawo",
"artist": “Malcom&Kloud DIpo”, "year": 2021, "rating": 5.7},
]
```

Exploratory Coding with Observable.

Observable is a reactive, notebook-style, JavaScript-ish programming environment for the web. Created by **Mike Bostock** (d3) in 2016. It's key feature is its **reactivity:** like a spreadsheet, cells in the document evaluate when dependencies update. Observable automatically re-runs our code **reactively** whenever something changes.

This contrasts the **linear order** of other code notebooks, such as **Jupyter** and **Wolfram**, which are **non-reactive**.

Demo Time

Let’s go the link below.

https://observablehq.com/

Access the DataSet at the link below.

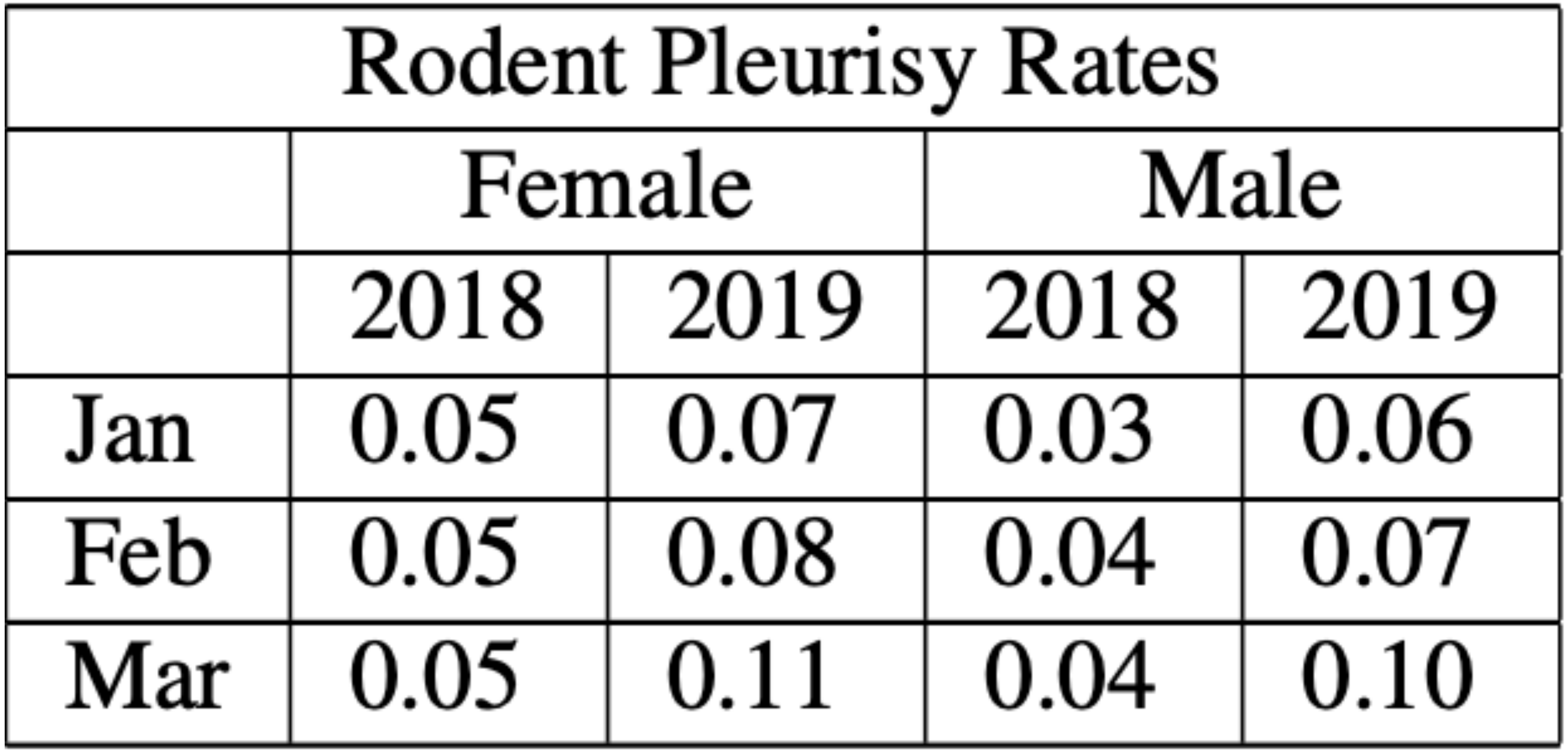
https://raw.githubusercontent.com/nshiab/simple-data-analysis/main/data/employees.csv

Cleaning data to make it tidy.

Tabular data is tidy if:

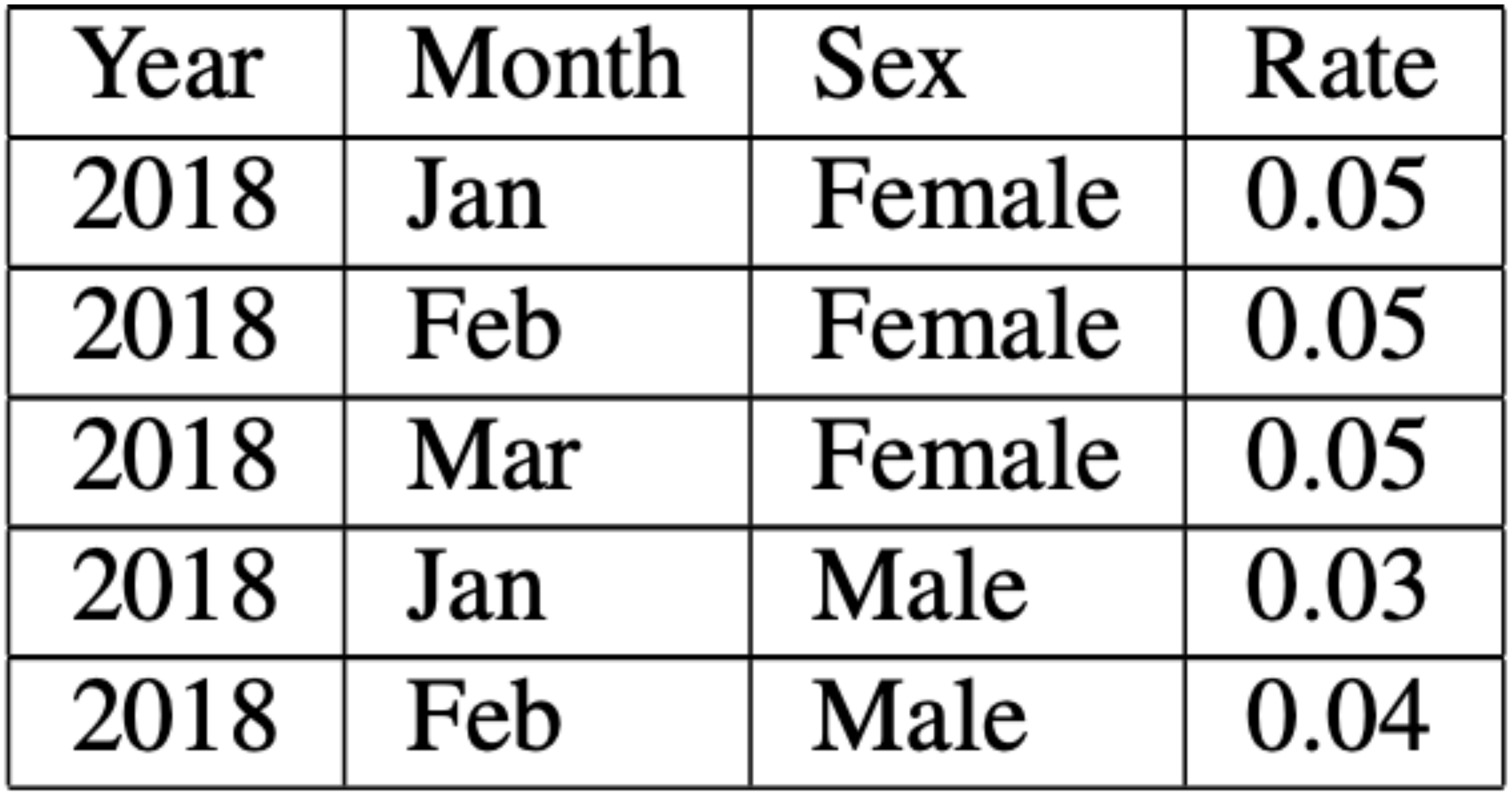
- Each **column** contains **one statistical variable** (i.e., one property that was measured or observed).
- Each different **observation** is in a different **row**.
- There is **one table** for each set of **observations**.
- If there are **multiple tables**, each t**able has a column** containing a **unique key** so that **related data can be linked.**

Untidy Data



Rodent Pleurisy Rates				
	Female		Male	
	2018	2019	2018	2019
Jan	0.05	0.07	0.03	0.06
Feb	0.05	0.08	0.04	0.07
Mar	0.05	0.11	0.04	0.10

Tidy Data



Year	Month	Sex	Rate
2018	Jan	Female	0.05
2018	Feb	Female	0.05
2018	Mar	Female	0.05
2018	Jan	Male	0.03
2018	Feb	Male	0.04

Demo Time

Let’s go the link below.

https://observablehq.com/

Access the DataSet at the link below.

https://raw.githubusercontent.com/nshiab/simple-data-analysis/main/data/employees.csv

References

https://frontendmasters.com/guides/front-end-handbook/2019/

https://medium.com/towards-data-science/javascript-for-data-analysis-2e8e7dbf63a7

Hadley Wickham. “Tidy Data”. In: Journal of Statistical Software 59.10 (2014). The defining paper on tidy data. DOI: 10.18637/jss.v059.i10

https://observablehq.com/

https://raw.githubusercontent.com/nshiab/simple-data-analysis/main/data/employees.csv

Thank-You