

# Data Science Basics in R

Day 2: Data management and version control

**Thank you!**

**See you tomorrow.**

*Please come with a fully charged laptop.*

# Goals for today

- Understand the foundations of data management
- Identify some useful global health and health diplomacy datasets
- Load and clean your first dataset in R
  - explore a new dataset while learning about data structures
- Learn best practices for documentation
- Get familiar with Github (by writing poems about unicorns)

# What is data management?

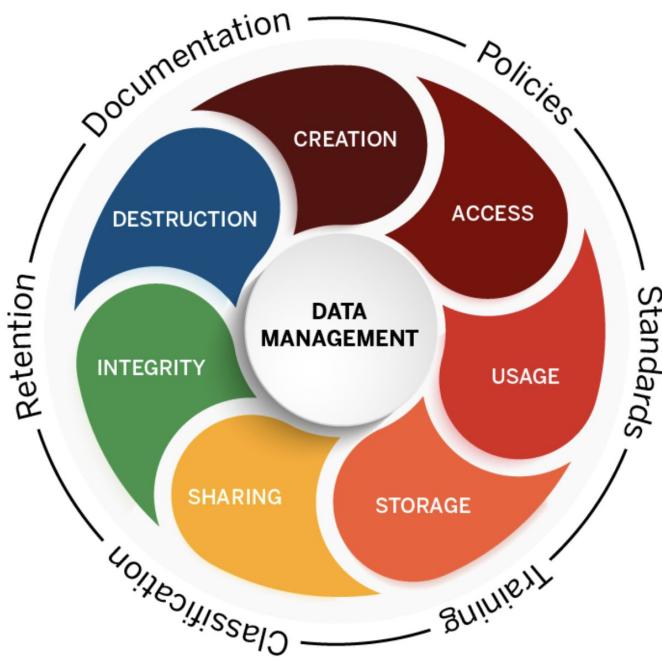
# Goals for today

- Understand the foundations of data management
- Identify some useful global health and health diplomacy datasets
- Load and clean your first dataset in R
  - explore a new dataset while learning about data structures
- Learn best practices for documentation
- Get familiar with Github (by writing poems about unicorn)

# Data management

Data management is the practice of creating, maintaining, documenting, and securing data.

It includes data cleaning, among many other tasks.



# Data access

- Where do your data “live”?
  - How can you access them?
  - Who else can access them?
  - For how long will they be available?
- What format are they in?
- Where is the documentation?
- How, and with whom, can you share data, code, and/or results?
- Identifying other discrepancies...

# Data management

Data cleaning tasks depending on what types of data you're working with and where you got them!

- Identifying and addressing missing values
- Fixing “weird” characters
- Combining multiple datasets
- Deduplicating lists
- Identifying other discrepancies...

# Real-life examples data entry and coding

## Global approaches to tackling antimicrobial resistance: a comprehensive analysis of water, sanitation and hygiene policies

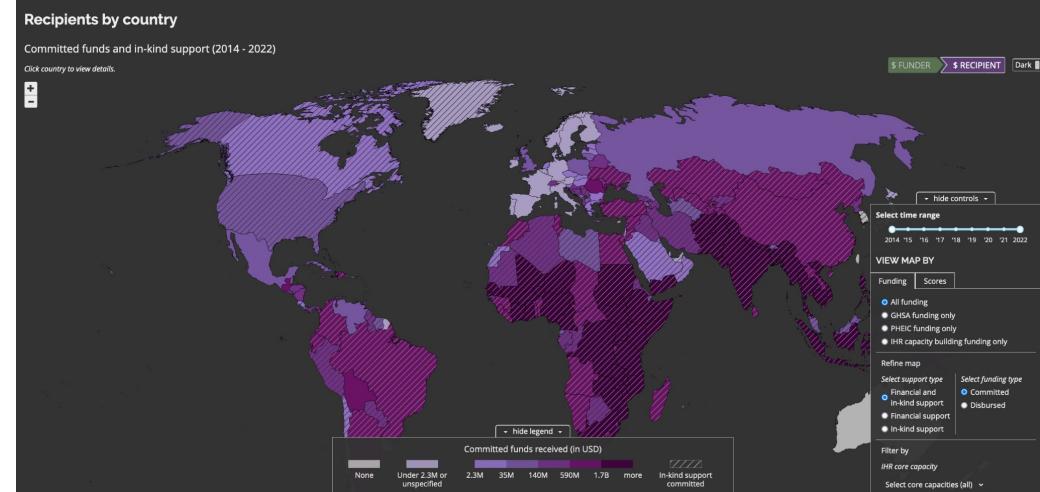
Ciara M Weets  , Rebecca Katz

A	B	C	D	E	F	G	H	
1	Country	Topic	Status justification	Color	Level	Government	Voting	Subtopic
2	Argentina	Sewerage	There is policy that requires a competent agency within the government to regulate sewerage systems for the consumption of human waste, however, in Argentina, sanitation services are regulated at the provincial level.	Option 1	Subnational	Federal	Presidential	Sewerage
3	Australia	Water quality standards	There is policy that mandates a competent environmental agency within the governing body to control and set standards for water quality, including drinking-water quality, environmental water quality, and/or recreational water quality. This happens at the provincial level, rather than the national level in Australia.	Option 1	Subnational	Federal	Parliamentary	Water Quality Standards
4	Australia	Water quality monitoring	There is policy that grants a competent environmental agency within the governing body with a mandate to monitor residues in water sources to prevent contamination that could cover contamination with antimicrobials. This happens at the provincial level, rather than the national level in Australia.	Option 1	Subnational	Federal	Parliamentary	Water Quality Monitoring
5	Australia	Pollutant disposal in water sources	There is policy that mandates a competent environmental agency within the governing body to regulate the disposal of pollutants in freshwater sources. This happens at the provincial level, rather than the national level in Australia.	Option 1	Subnational	Federal	Parliamentary	Effluent Wastewater Disposal
6	Australia	Effluent wastewater disposal	There is policy that mandates a competent environmental agency within the governing body to regulate the disposal of effluent wastewater, however, in Australia, this regulation takes place at the subnational level.	Option 1	Subnational	Federal	Parliamentary	Pollutant Disposal in water sources
7	Australia	Sewerage	There is policy that requires a competent agency within the government to regulate sewerage systems for the consumption of human waste, however, in Australia, sanitation services are regulated at the provincial level.	Option 1	Subnational	Federal	Parliamentary	Medical Waste Disposal
8	Australia	Medical waste disposal	There is policy that requires the government to regulate the disposal of medical waste from healthcare facilities, however, in Australia, this regulation takes place at the subnational level.	Option 1	Subnational	Federal	Parliamentary	Total

# Real-life examples data aggregation



Data source	Description
BWC Working Papers	Occasionally, Member States of the BWC submit national papers describing capacity building efforts for health security. Data relevant from these papers are included in GHS Tracking.
CEPI Progress Reports	The Coalition for Epidemic Preparedness Innovations (CEPI) is a partnership created in 2017 to bring together global stakeholders to develop vaccines with the goal of stopping future pandemics. During the COVID-19 pandemic that began in 2019, it operated as a facilitator in the COVAX Marketplace. CEPI announces funding calls for vaccine candidate development and coordinates partnerships to develop specific vaccines as needed. CEPI releases annual progress reports that include actual and planned donor funding amounts by year. More information about CEPI is available at <a href="https://cepi.net/">https://cepi.net/</a> .
Ebola Recovery Tracking Initiative	The Ebola Recovery Tracking Initiative tracks official development assistance towards Ebola recovery efforts in Guinea, Liberia, Sierra Leone, and the Mano River Union. The initiative is a partnership between the governments of Guinea, Liberia and Sierra Leone, the United Nations Office of the Secretary-General's Special Adviser on Community Based Medicine and Lessons from Haiti, and the United Nations Development Programme (UNDP). The Ebola Recovery Tracking Initiative is available online at: <a href="https://ebolarecovery.org/">https://ebolarecovery.org/</a> .



[tracking.GHScosting.org](https://tracking.GHScosting.org)

# Real-life examples *reformatting & cleaning*

## Methods

- Approximately 54% of PLM patients contribute free-text to the platform through user bios, forum conversations, or as annotations associated with structured data.
  - Posts contributed by PTSD patients were pre-processed to remove HTML and stopwords, and to map emojis to text-based descriptions.
  - Latent Dirichlet allocation (LDA), a form of probabilistic topic modeling, was performed to identify topics discussed among patients.
  - Model parameters were selected on the basis of perplexity as measured on a 10% holdout group.

patientslikeme®

## The Patient Voice Includes Emojis:

## A case study in the use of probabilistic topic modeling to characterize patient conversations in an online community of PTSD patients

Eaneff, S

**Fig 3A. "Sleep" Topic**

### Top 3 emojis



## Sample post

"Nightmares kicked my ass last night. Got woken up by a nightmare and stayed up for a long time feeling really freaked out."

tonight hope  
staying good early awake  
feeling need morning  
feeling asleep work till tomorrow  
scheduling bed before bedtime  
nights today hoping body  
apnea hour sleeping trouble  
days time staying  
nightmares bad sleep  
dreams fell back stay  
bad back yesterday late  
times bit time woke  
rest long time taking  
couple hours sleeping  
ready meds hard  
make exhausted wake  
headache  
**sleep**

### **Fig 3B. "School/Work" Topics**

### Top 3 emoji



## Sample post

"This semester, I'm taking four classes for the first time in a long time. I've quit my factory job, which was too physically demanding, and decided to focus on finishing school."

learn taking today break  
position math training volunteered  
teaching hard classes make hope  
years part working study learning  
end years back team starting balance  
great day physics week school year career  
sing head program year interv  
students teacher fresh people  
full kids days things team project  
work worked started fun  
class good class home  
job time college university  
student hours starting phys lot education  
homework meeting community school  
team project

Explore some useful  
sources of health  
diplomacy data

# Goals for today

- Understand the foundations of data management
- Identify some useful global health and health diplomacy datasets
- Load and clean your first dataset in R
  - explore a new dataset while learning about data structures
- Learn best practices for documentation
- Get familiar with Github (by writing poems about unicorn)

# WHO Global Health Observatory

<https://www.who.int/data/gho>



The banner features a dark purple background with abstract white and light purple geometric patterns. In the center, the text "THE GLOBAL HEALTH OBSERVATORY" is written in a bold, white, sans-serif font. Below it, the tagline "Explore a world of health data" is displayed in a slightly smaller white font. At the bottom of the banner, there are two white rectangular buttons with rounded corners, each containing the text "Indicators >" and "Countries >" respectively, in a small white font.

Health Topics ▾ Countries ▾ Newsroom ▾ Emergencies ▾ Data ▾ About WHO ▾

GHO Home Indicators Countries Data API ▾ Map Gallery Publications Data Search

THE GLOBAL HEALTH OBSERVATORY

Explore a world of health data

Indicators > Countries >

# OECD Health Data Repository

<https://data.oecd.org/health.htm>

Screenshot of the OECD Health Data Repository website.

The page features a navigation bar at the top with links to OECD.org, Data, Publications, More sites, News, and Job vacancies. It also includes a search bar for "Search for OECD data" and language links for "Français".

The main content area has a header "OECD Data" with a "Health" dropdown menu. Below this, there's a sidebar titled "Topic aspects" listing "Health care use", "Health equipment", "Health resources", "Health risks", and "Health status".

The main content area displays three bar charts under the "Health care use" section:

- Doctors' consultations**: A bar chart showing data for 2021. The Y-axis ranges from 0 to 6. The chart shows values approximately 1.5, 3.0, 3.0, 5.5, and 6.5 across five categories.
- Length of hospital stay**: A bar chart showing data for 2021. The Y-axis ranges from 0 to 6. The chart shows values approximately 4.5, 5.5, 6.0, 6.5, and 6.5 across five categories.
- Health at a Glance**: A thumbnail image of the publication cover, labeled as a "PUBLICATION (2021)".

At the bottom, there's a section titled "INDICATOR GROUP" with a link to "Health care use".

# OCHA Humanitarian Data Exchange

<https://data.humdata.org/>

The screenshot shows the homepage of the OCHA Humanitarian Data Exchange (HDX) at <https://data.humdata.org/>. The page has a dark header bar with the OCHA Services logo, a search bar, and navigation links for DATA, LOCATIONS, ORGANISATIONS, and DATAVIZ. A red "ADD DATA" button is also in the header. The main content area features a teal background with the title "The Humanitarian Data Exchange" and a subtitle "Find, share and use humanitarian data all in one place". Below this is a "LEARN MORE" button. To the right, there are two main sections: "FIND DATA" and "ADD DATA". The "FIND DATA" section displays statistics: 20,758 DATASETS, 254 LOCATIONS, and 1,890 SOURCES. The "ADD DATA" section includes options to "UPLOAD FILE" or "ADD METADATA". At the bottom, a red banner states "Learn how the HDX team supports responsible data sharing."

OCHA Services ▾

HDX Search Datasets

DATA | LOCATIONS | ORGANISATIONS | DATAVIZ ▾ ADD DATA

# The Humanitarian Data Exchange

Find, share and use humanitarian data all in one place

LEARN MORE

FIND DATA

Search Datasets

20,758 DATASETS | 254 LOCATIONS | 1,890 SOURCES

ADD DATA

UPLOAD FILE ADD METADATA

Learn how the HDX team supports responsible data sharing.

# Demographic and Health Surveys

<https://dhsprogram.com/>



SEARCH

LOGIN

Select Language ▾



COUNTRIES

DATA

PUBLICATIONS

METHODOLOGY

RESEARCH

TOPICS

The Demographic and Health Surveys (DHS) Program has collected, analyzed, and disseminated accurate and representative data on population, health, HIV, and nutrition through more than 400 surveys in over 90 countries.

Another and daughters in Ethiopia work with coffee beans after their house has received Indoor Residual Spraying (IRS) to reduce malaria transmission. Photo Credit: AIRS Ethiopia PMI

# UN SDG Global Database

<https://unstats.un.org/sdgs/dataportal>

United Nations | Department of Economic and Social Affairs  
Statistics • SDG Indicators Database

Home SDG Indicators Data SDG Reports HLG-PCCB IAEG-SDG's Events Resources

Select indicators and country or area

SDG Global Database gives you access to data on more than 210 SDG indicators for countries across the globe  
by indicator, country, region or time period

Global SDG Indicators Data Platform

SUSTAINABLE DEVELOPMENT GOALS

# UNDP Data Futures Platform

[data.undp.org](http://data.undp.org)



Data Futures  
Exchange

FOCUS AREAS

REGIONS & COUNTRIES

RESOURCES

ABOUT US

ACCESS ALL DATA



## DATA FUTURES EXCHANGE

Data innovation for decision intelligence

GREENHOUSE GAS (GHG) EMISSIONS, MTCO<sub>2</sub>E



0 50 100 250 500 1k 2.5k 5k

# IPUMS

[www.idhsdata.org/idhs/index.shtml](http://www.idhsdata.org/idhs/index.shtml)

The screenshot shows the homepage of the IPUMS DHS website. At the top left is the IPUMS DHS logo. To its right is the text "DEMOGRAPHIC AND HEALTH SURVEYS". Below that is a navigation bar with links for "HOME", "SELECT DATA", "MY DATA", and "SUPPORT". A horizontal banner below the navigation bar displays six small photographs related to demographic and health surveys.

**IPUMS DHS**

ABOUT  
THE DHS PROGRAM: HOME ↗  
REGISTER  
DONATE TO IPUMS ↗

**DATA**

BROWSE AND SELECT DATA  
ANALYZE DATA ONLINE  
DOWNLOAD OR REVISE MY DATA

**SUPPLEMENTAL DATA**

GEOGRAPHY & GIS  
CONTEXTUAL VARIABLES

**DOCUMENTATION**

USER NOTES  
SAMPLE DESCRIPTIONS  
SAMPLE UNIVERSES  
QUESTIONNAIRES  
REVISION HISTORY

**SUPPORT**

FAQ

**HEALTH-RELATED MICRODATA FOR LOW- AND MIDDLE-INCOME COUNTRIES**

IPUMS-DHS facilitates analysis of Demographic and Health Surveys, administered in low- and middle-income countries since the 1980s. IPUMS-DHS contains thousands of consistently coded variables on the health and well-being of women, children, births, men, and on all members of randomly selected households, for 32 African countries and 9 Asian countries. Users can determine variable availability at a glance and create data files with just the variables and samples they need.

**45 COUNTRIES – 180 SAMPLES – OVER 15,000 VARIABLES – 27 MILLION PERSON RECORDS**

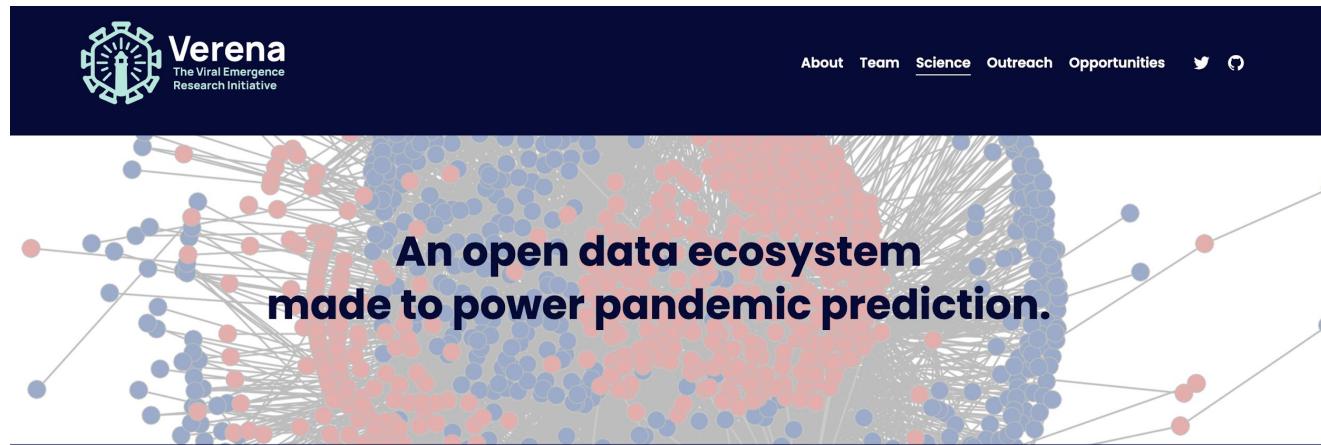
**USE IT FOR GOOD – NEVER FOR EVIL**

— CREATE AN EXTRACT —    **ONLINE TOOL FOR ANALYSIS**    — CREATE AN ACCOUNT —

Get Data    Analyze Data Online    Register

# Verena

<https://www.viralemergence.org/data>



# IDEA

<https://ghssidea.org/>



INTERNATIONAL DISEASE  
**IDEA**  
AND EVENTS ANALYSIS

---

[About](#)   [Citations](#)   [Contact](#)

---

The International Disease and Events Analysis (IDEA) platform is a suite of integrated research tools, including global health security visualization dashboards, decision support tools, and data libraries developed by the Georgetown University Center for Global Health Science and Security.

See below for a description of each tool, links to the sites, and access to a quick download of the data from each.

## Let's chat

What other datasets have you used for work or school?

# Course datasets

For this course, we'll primarily use global-scale datasets related to vaccine-preventable diseases.

- **policy data:** national policies related to vaccination
- **coverage data:** % of relevant populations vaccinated
- **incidence data:** rates of disease incidence

# Course datasets

*Some additional context*

Work with Ciara to add in a single slide

# Loading and cleaning datasets in R

# Goals for today

- Understand the foundations of data management
- Identify some useful global health and health diplomacy datasets
- Load and clean your first dataset in R
  - explore a new dataset while learning about data structures
  - do some basic data cleaning
- Learn best practices for documentation
- Get familiar with Github (by writing poems about unicorn)

# Raw data files

We're going to talk about structured, flat files today, but there are many types of data that R can work with!

- **Excel:** proprietary file format (Microsoft), but common (.xls files)
- **CSV:** comma-separated value files
- **TSV:** tab-separated value files
- **RData:** data saved from within your R environment
- **SPSS or SAS:** data from other statistical programming tools
- **Shapefiles:** geospatial or mapping data

# Functions to load data

- **Excel:** `read_excel("countries.xls")`
- **CSV:** `read.csv("countries.csv")`
- **TSV:** `read.delim("countries.tsv")`
- **RData:** `load("countries.RData")`
- **SPSS or SAS:** use the 'foreign' package
- **Shapefiles:** use the 'sf' or other geospatial data packages

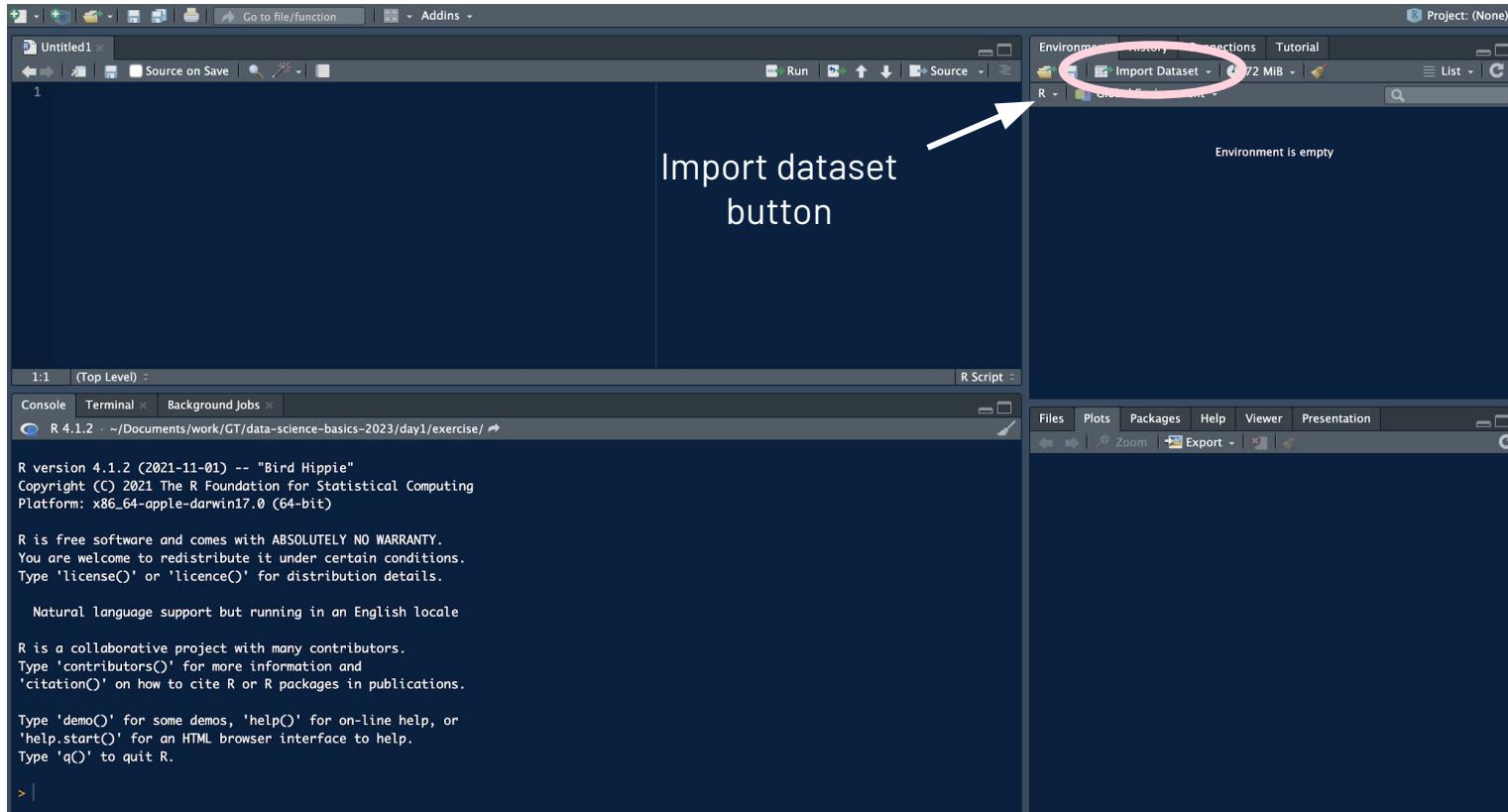
# A note on Excel files and libraries in R

To use the function `read_excel()`, we need to have installed the package `readxl` into R. You (usually) only need to install it once, and then you will need to re-load it every new R session

- **Installing:** `install.packages("readxl")`
- **Loading:** `library("readxl")`

If you are working with RStudio and using the workflow in the next slides, you will automatically be prompted to do these things if they are required

# Loading data with RStudio



# Loading data with RStudio

Several default options of data types, including Excel.  
Here, I'll select the first option, using base R

R version 4.1.2 (2021-11-01) -- "Bird Hippie"  
Copyright (C) 2021 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.

# Loading data with RStudio

The screenshot shows the RStudio interface with a file selection dialog open. The dialog lists files in the 'course-datasets' folder. A pink oval highlights the 'day2.tsv' file, which is a TSV Document. An arrow points from the text below to this highlighted file.

Select the dataset you want to import,  
you might need to click around to find  
it on your computer

Name	Size	Kind	Date Added
day2.tsv	28 KB	TSV Document	Jan 19, 2024 at 3:25
Data dictionary.xlsx	508 KB	Microsoft Excel Workbook (.xlsx)	Nov 6, 2023 at 4:07

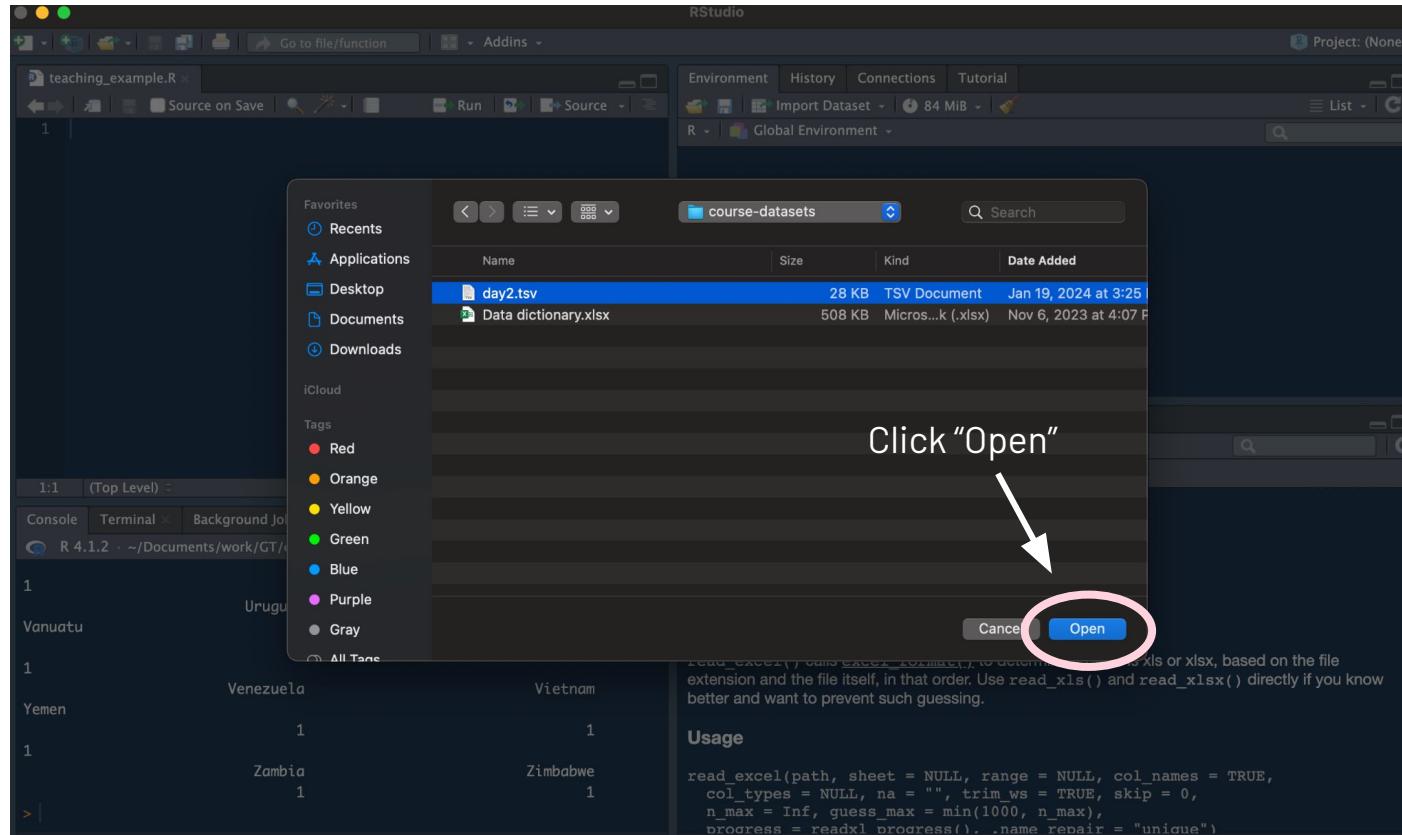
Cancel Open

read\_excel() calls guess\_format() to determine if path is xls or xlsx, based on the file extension and the file itself, in that order. Use read\_xls() and read\_xlsx() directly if you know better and want to prevent such guessing.

**Usage**

```
read_excel(path, sheet = NULL, range = NULL, col_names = TRUE,
          col_types = NULL, na = "", trim_ws = TRUE, skip = 0,
          n_max = Inf, guess_max = min(1000, n_max),
          progress = readxl::progress(verbose = TRUE, name_repair = "unique"))
```

# Loading data with RStudio



# Loading data with RStudio

The screenshot shows the RStudio interface with the "Import Dataset" dialog open. The dialog has the following settings:

- Name: day2
- Encoding: Automatic
- Heading: Yes (radio button selected)
- Row names: Automatic
- Separator: Tab
- Decimal: Period
- Quote: Double ("")
- Comment: None
- na.strings: NA
- Strings as factors

The "Input File" section displays the first few lines of the data as a character vector:

```
"iso_3166" ~> "country" ~> "who_region" ~> "income_group"  
"AFG" ~> "Afghanistan" ~> "Eastern Mediterranean Region" ~>  
"AGO" ~> "Angola" ~> "African Region" ~> "Lower middle income"  
"ALB" ~> "Albania" ~> "European Region" ~> "Upper middle income"  
"AND" ~> "Andorra" ~> "European Region" ~> "High income" ~>  
"ARE" ~> "United Arab Emirates (UAE)" ~> "Eastern Mediterranean Region"  
"ARG" ~> "Argentina" ~> "Region of the Americas" ~> "Upper middle income"  
"ARM" ~> "Armenia" ~> "European Region" ~> "Upper middle income"  
"ATG" ~> "Antigua and Barbuda" ~> "Region of the Americas" ~>  
"AUS" ~> "Australia" ~> "Western Pacific Region" ~> "High income"  
"AUT" ~> "Austria" ~> "European Region" ~> "High income" ~>  
"AZE" ~> "Azerbaijan" ~> "European Region" ~> "Upper middle income"  
"BDI" ~> "Burundi" ~> "African Region" ~> "Low income" ~> "Sub-Saharan Africa"  
"BEL" ~> "Belgium" ~> "European Region" ~> "High income" ~>
```

The "Data Frame" section shows a preview of the imported data:

V1	V2	V3
iso_3166	country	who_region
AFG	Afghanistan	Eastern Mediterranean Region
AGO	Angola	African Region
ALB	Albania	European Region
AND	Andorra	European Region
ARE	United Arab Emirates (UAE)	Eastern Mediterranean Region
ARG	Argentina	Region of the Americas
ARM	Armenia	European Region
ATG	Antigua and Barbuda	Region of the Americas
AUS	Australia	Western Pacific Region
AUT	Austria	European Region
AZE	Azerbaijan	European Region
BDI	Burundi	African Region

At the bottom of the dialog, there are "Import" and "Cancel" buttons.

In the RStudio console area, the following code is visible:

```
read_excel(path, sheet = NULL, range = NULL, col_names = TRUE,  
          col_types = NULL, na = "", trim_ws = TRUE, skip = 0,  
          n_max = Inf, n_rows_max = min(1000, n_max),  
          progress = readxl::progress(), .name_repair = "unique")
```

# Loading data with RStudio

The screenshot shows the RStudio interface with the "Import Dataset" dialog box open. The dialog box has the following settings:

- Name: day2
- Encoding: Automatic
- Heading: No
- Row names: Automatic
- Separator: Tab
- Decimal: Period
- Quote: Double ("")
- Comment: None
- na.strings: NA
- Strings as factors

The "Data Frame" section displays the first few rows of the dataset:

	V2	V3
1	iso_3166	country
	AFG	Afghanistan
AGO	Angola	African Region
ALB	Albania	European Region
AND	Andorra	European Region
ARE	United Arab Emirates (UAE)	Eastern Mediterranean Region
ARG	Argentina	Region of the Americas
ARM	Armenia	European Region
ATG	Antigua and Barbuda	Region of the Americas
AUS	Australia	Western Pacific Region
AUT	Austria	High income
AZE	Azerbaijan	European Region
BDI	Burundi	Upper middle income
BEL	Belgium	European Region

A red circle highlights the first three rows of the data frame.

In the bottom right corner of the dialog box, there is a note: "This is xls or xlsx, based on the file extension and read\_xlsx() directly if you know".

At the bottom of the dialog box are two buttons: "Import" and "Cancel".

The RStudio environment shows a code editor with "teaching\_example.R" and a terminal window. The terminal window contains the following R code:

```
read_excel(path, sheet = NULL, range = NULL, col_names = TRUE,
          col_types = NULL, na = "", trim_ws = TRUE, skip = 0,
          n_max = Inf, guess_max = min(1000, n_max),
          progress = readxl::progress(), .name_repair = "unique")
```

# Loading data with RStudio

Tell R to use the first row as the heading

The screenshot shows the RStudio interface with a dark theme. A modal dialog titled "Import Dataset" is open in the center. On the left of the dialog, there are several input fields: "Name" (set to "day2"), "Encoding" (set to "Automatic"), "Heading" (radio button selected for "Yes"), "Row names" (set to "Automatic"), "Separator" (set to a blank space), "Decimal" (set to "Period"), "Quote" (set to "Double ('')"), "Comment" (set to "None"), and "na.strings" (set to "NA"). Below these fields is a checkbox for "Strings as factors". To the right of the fields is a large text area showing the contents of the "Input File". At the bottom of the dialog are two buttons: "Import" and "Cancel". In the background, the RStudio environment shows a code editor with a script named "teaching\_example.R" containing some R code, and a data frame preview window showing the first few rows of the dataset.

Input File

```
"iso_3166" -> "country" -> "who_region" -> "income_group"
"AFG" -> "Afghanistan" -> "Eastern Mediterranean Region" -> "Lower middle income"
"AGO" -> "Angola" -> "African Region" -> "Lower middle income"
"ALB" -> "Albania" -> "European Region" -> "Upper middle income"
"AND" -> "Andorra" -> "European Region" -> "High income"
"ARE" -> "United Arab Emirates (UAE)" -> "Eastern Mediterranean Region" -> "High income"
"ARG" -> "Argentina" -> "Region of the Americas" -> "Upper middle income"
"ARM" -> "Armenia" -> "European Region" -> "Upper middle income"
"ATG" -> "Antigua and Barbuda" -> "Region of the Americas" -> "High income"
"AUS" -> "Australia" -> "Western Pacific Region" -> "High income"
"AUT" -> "Austria" -> "European Region" -> "High income"
"AZE" -> "Azerbaijan" -> "European Region" -> "Upper middle income"
"BDI" -> "Burundi" -> "African Region" -> "Low income"
"BEL" -> "Belgium" -> "European Region" -> "High income"
```

Data Frame

iso_3166	country	who_region
AFG	Afghanistan	Eastern Mediterranean Region
AGO	Angola	African Region
ALB	Albania	European Region
AND	Andorra	European Region
ARE	United Arab Emirates (UAE)	Eastern Mediterranean Region
ARG	Argentina	Region of the Americas
ARM	Armenia	European Region
ATG	Antigua and Barbuda	Region of the Americas
AUS	Australia	Western Pacific Region
AUT	Austria	European Region
AZE	Azerbaijan	European Region
BDI	Burundi	African Region
BEL	Belgium	European Region

Import Cancel

read\_excel(path, sheet = NULL, range = NULL, col\_names = TRUE,  
col\_types = NULL, na = "", trim\_ws = TRUE, skip = 0,  
n\_max = Inf, guess\_max = min(1000, n\_max),  
progress = readxl::progress(), .name\_repair = "unique")

# Loading data with RStudio

The screenshot shows the RStudio interface with the "Import Dataset" dialog box open. The dialog box has two main sections: "Input File" and "Data Frame".

**Input File:**

```
"iso_3166" ~> "country" ~> "who_region" ~> "income_group"
"AFG" ~> "Afghanistan" ~> "Eastern Mediterranean Region" ~> "Lower middle income"
"AGO" ~> "Angola" ~> "African Region" ~> "Lower middle income"
"ALB" ~> "Albania" ~> "European Region" ~> "Upper middle income"
"AND" ~> "Andorra" ~> "European Region" ~> "High income"
"ARE" ~> "United Arab Emirates (UAE)" ~> "Eastern Mediterranean Region" ~> "High income"
"ARG" ~> "Argentina" ~> "Region of the Americas" ~> "Upper middle income"
"ARM" ~> "Armenia" ~> "European Region" ~> "Upper middle income"
"ATG" ~> "Antigua and Barbuda" ~> "Region of the Americas" ~> "High income"
"AUS" ~> "Australia" ~> "Western Pacific Region" ~> "High income"
"AUT" ~> "Austria" ~> "European Region" ~> "High income"
"AZE" ~> "Azerbaijan" ~> "European Region" ~> "Upper middle income"
"BDI" ~> "Burundi" ~> "African Region" ~> "Low income"
"BEL" ~> "Belgium" ~> "European Region" ~> "High income"
```

**Data Frame:**

iso_3166	country	who_region
AFG	Afghanistan	Eastern Mediterranean Region
AGO	Angola	African Region
ALB	Albania	European Region
AND	Andorra	European Region
ARE	United Arab Emirates (UAE)	Eastern Mediterranean Region
ARG	Argentina	Region of the Americas
ARM	Armenia	European Region
ATG	Antigua and Barbuda	Region of the Americas
AUS	Australia	Western Pacific Region
AUT	Austria	European Region
AZE	Azerbaijan	European Region
BDI	Burundi	African Region
BEL	Belgium	European Region

At the bottom of the dialog box, there are "Import" and "Cancel" buttons. The "Import" button is circled in red.

**Console Output:**

```
teaching_example.R
Source on Save
Run
Source
Import Dataset
20 MB
List
C
Project: (None)
1
1:1 (Top Level)
Console Terminal Background Jobs
R 4.1.2 ~/Documents/work/GT/data...
Uruguay
Vanuatu
Venezuela
Yemen
Zambia
Zimbabwe
read_excel(path, sheet = "", range = NULL, col_names = TRUE,
          col_types = NULL, na = "", trim_ws = TRUE, skip = 0,
          n_max = Inf, guess_max = min(1000, n_max),
          progress = readxl$progress(), .name_repair = "unique")
```

Click "Import"

# Loading data with RStudio

A screenshot of the RStudio interface demonstrating the loading of a dataset. The left pane shows a data frame named 'day2' with 194 observations and 7 variables, containing columns like 'country', 'who\_region', 'income\_group', and 'wo'. A pink circle highlights this data frame. The right pane shows the 'Environment' tab with 'day2' listed. A large arrow points from the text 'There's the data!' to the 'day2' entry in the Environment pane. The bottom pane shows the R console with the command used to load the data: `day2 <- read.delim("~/Documents/work/GT/data-science-basics-2024/course-datasets/day2.tsv")`.

There's the data!

country	who_region	income_group	wo
AFG	Eastern Mediterranean Region	Low income	Sou
AGO	African Region	Lower middle income	Sub
ALB	European Region	Upper middle income	Eur
AND	European Region	High income	Eur
ARE	Eastern Mediterranean Region	High income	Mic
ARG	Region of the Americas	Upper middle income	Lat
ARM	European Region	Upper middle income	Eur
ATG	Region of the Americas	High income	Lat
AUS	Western Pacific Region	High income	Eas
AUT	European Region	High income	Eur
AZE	European Region	Upper middle income	Eur
BDI	African Region	Low income	Sub

```
R 4.1.2 · ~/Documents/work/GT/data-science-basics-2024/ →  
> day2 <- read.delim("~/Documents/work/GT/data-science-basics-2024/course-datasets/day2.tsv")  
> View(day2)  
>  
> |
```

# Loading data with RStudio

The screenshot shows the RStudio interface with the following components:

- Left Panel (Code Editor):** Displays the R script `teaching_example.R`. A pink oval highlights the command `View(day2)`, which has been executed.
- Top Bar:** Shows the RStudio logo, file menu, and tabs like "Addins".
- Environment Tab:** Shows the global environment with a data frame named `day2` containing 194 observations and 7 variables.
- Data View:** Shows the contents of the `day2` data frame, which includes columns for iso\_3166, country, who\_region, income\_group, and others.
- Console Tab:** Shows the R code used to load the data: `R 4.1.2 · ~/Documents/work/GT/data-science-basics-2024/ → > day2 <- read.delim("~/Documents/work/GT/data-science-basics-2024/course-datasets/day2.tsv")`.
- Bottom Bar:** Includes tabs for Files, Plots, Packages, Help, Viewer, and Presentation, along with navigation icons.

A pink arrow points from the highlighted code in the console to the explanatory text on the right.

R Studio wrote and  
logged some code for us  
here, which loaded the  
data

# Loading data with RStudio

The screenshot shows the RStudio interface with the following components:

- Data View (left pane):** Displays a table titled "day2" with 194 observations and 7 variables. The columns are: iso\_3166, country, who\_region, income\_group, and wo. The data includes entries for Afghanistan, Angola, Albania, Andorra, United Arab Emirates (UAE), Argentina, Armenia, Antigua and Barbuda, Australia, Austria, Azerbaijan, and Burundi.
- Console (bottom-left pane):** Shows the R code used to load the data:

```
> day2 <- read.delim("./Documents/work/GT/data-science-basics-2024/course-datasets/day2.tsv")
> View(day2)
> |
```

A red oval highlights the first two lines of code, and a white arrow points from this highlighted area to the explanatory text below.
- Environment (top-right pane):** Shows the global environment with "day2" selected, containing 194 observations and 7 variables.
- Plots, Packages, Help, Viewer, Presentation (bottom-right pane):** Standard RStudio navigation tabs.

**The “View” command printed our data above**

# Loading data with RStudio

The screenshot shows the RStudio interface. On the left, the 'Environment' tab displays a list of loaded datasets, with 'day2' highlighted and circled in red. The 'Global Environment' tab shows '194 obs. of 7 variables'. On the right, a large text box contains the message: 'We can see a list of all of our loaded datasets here'. At the bottom, the 'Console' tab shows the R code used to load the dataset:

```
R 4.1.2 · ~/Documents/work/GT/data-science-basics-2024/ →
> day2 <- read.delim("~/Documents/work/GT/data-science-basics-2024/course-datasets/day2.tsv")
> View(day2)
>
> |
```

We can see a list of all of our loaded datasets here

# Workshop dataset

The screenshot shows a GitHub repository interface for the 'data-science-basics-2024' repository. The 'course-datasets' folder contains several files, including 'Data dictionary.xlsx', 'day1', 'day2' (which is selected), 'day3', 'extras', '.gitignore', and 'README.md'. The 'day2.tsv' file is displayed as a table with 195 rows and 11 columns. The columns are labeled: iso\_3166, country, who\_region, income\_group, world\_bank\_region, total\_population, mea, Vacc, Vacci, Vacci, Vacci, Ther, and Vacc. The data includes entries for countries like Afghanistan, Angola, Albania, Andorra, United Arab Emirates, Argentina, Armenia, Antigua and Barbuda, Australia, Austria, and Azerbaijan, along with their respective region, income group, and population statistics.

1	iso_3166	country	who_region	income_group	world_bank_region	total_population	mea	Vacc	Vacci	Vacci	Vacci	Ther	Vacc
2	AFG	Afghanistan	Eastern Mediterranean Region	Low income	South Asia	40099462							
3	AGO	Angola	African Region	Lower middle income	Sub-Saharan Africa	34503774							
4	ALB	Albania	European Region	Upper middle income	Europe & Central Asia	2811666							
5	AND	Andorra	European Region	High income	Europe & Central Asia	79034							
6	ARE	United Arab Emirates (UAE)	Eastern Mediterranean Region	High income	Middle East & North Africa	9365145							
7	ARG	Argentina	Region of the Americas	Upper middle income	Latin America & Caribbean	45808747							
8	ARM	Armenia	European Region	Upper middle income	Europe & Central Asia	2790974							
9	ATG	Antigua and Barbuda	Region of the Americas	High income	Latin America & Caribbean	93219							
10	AUS	Australia	Western Pacific Region	High income	East Asia & Pacific	25685412							
11	AUT	Austria	European Region	High income	Europe & Central Asia	8955797							
12	AZE	Azerbaijan	European Region	Upper middle income	Europe & Central Asia	10137750							

## Workshop dataset

[bit.ly/health\\_diplomacy\\_day2\\_data](https://bit.ly/health_diplomacy_day2_data)

# Loading data with RStudio

- Go to [bit.ly/health\\_diplomacy\\_day2\\_data](https://bit.ly/health_diplomacy_day2_data)
- Copy URL
- Read data into R
- `read.delim("https://raw.githubusercontent.com/seaneff/data-science-basics-2023/main/reference-dataset/countries.tsv")`

# Data structures in R

# Goals for today

- Understand the foundations of data management
- Identify some useful global health and health diplomacy datasets
- Load and clean your first dataset in R
  - explore a new dataset while learning about data structures
- Learn best practices for documentation
- Get familiar with Github (by writing poems about unicorns)

# Data Structures in R

Yesterday we discussed vectors, today, we're going to talk about other types of data structures in R. There are a few different types, but the ones we'll talk about this week are:

- **vectors**: group of data elements of a certain type (1D)
- **matrix**: group of data elements of a certain type (2D)
- **data frame**: group of data elements with different types

# Data frame

R teaching\_example.R x dayz x

Filter

	iso_3166	country	who_region	income_group	wo
1	AFG	Afghanistan	Eastern Mediterranean Region	Low income	Sou
2	AGO	Angola	African Region	Lower middle income	Sub
3	ALB	Albania	European Region	Upper middle income	Eur
4	AND	Andorra	European Region	High income	Eur
5	ARE	United Arab Emirates (UAE)	Eastern Mediterranean Region	High income	Mic
6	ARG	Argentina	Region of the Americas	Upper middle income	Lat
7	ARM	Armenia	European Region	Upper middle income	Eur
8	ATG	Antigua and Barbuda	Region of the Americas	High income	Lat
9	AUS	Australia	Western Pacific Region	High income	East
10	AUT	Austria	European Region	High income	Eur
11	AZE	Azerbaijan	European Region	Upper middle income	Eur
12	BDI	Burundi	African Region	Low income	Sub

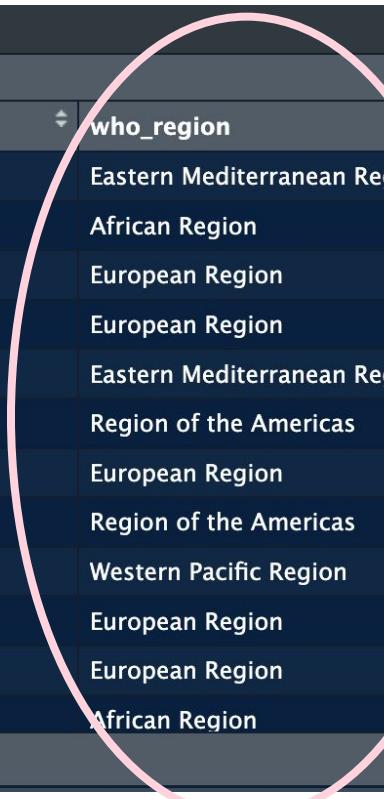
Showing 1 to 11 of 194 entries, 7 total columns

# Vectors

R teaching\_example.R x day2 x Filter

iso_3166	country	who_region	income_group	wo
1 AFG	Afghanistan	Eastern Mediterranean Region	Low income	Sou
2 AGO	Angola	African Region	Lower middle income	Sub
3 ALB	Albania	European Region	Upper middle income	Eur
4 AND	Andorra	European Region	High income	Eur
5 ARE	United Arab Emirates (UAE)	Eastern Mediterranean Region	High income	Mid
6 ARG	Argentina	Region of the Americas	Upper middle income	Lat
7 ARM	Armenia	European Region	Upper middle income	Eur
8 ATG	Antigua and Barbuda	Region of the Americas	High income	Lat
9 AUS	Australia	Western Pacific Region	High income	East
10 AUT	Austria	European Region	High income	Eur
11 AZE	Azerbaijan	European Region	Upper middle income	Eur
12 BDI	Burundi	African Region	Low income	Sub

Showing 1 to 11 of 194 entries, 7 total columns



## Now you try!

- Can you find a logical vector?
- A character vector?
- A numeric vector?

# Exploring your dataset `head()` function

The screenshot shows the RStudio interface with the following details:

- Top Bar:** Contains tabs for "teaching.example.R", "Source on Save", "Run", "Source", and "Environment".
- Global Environment:** Shows a dataset named "day2" with 194 observations and 8 variables.
- Console:** Displays the command `head(day2)` and its output, which includes columns like iso\_3166, country, who\_region, income\_group, world\_bank\_region, who\_member\_state, total\_population, and measles\_policy.
- Bottom Navigation:** Includes tabs for "Files", "Plots", "Packages", "Help", "Viewer", and "Presentation".

# Exploring your dataset

## View() function

The screenshot shows the RStudio interface with the following components:

- Code Editor:** Shows the command `> View(day2)` highlighted with a pink oval.
- Data View:** A data grid titled "day2" showing 13 rows of data. The columns are: iso\_3166, country, who\_region, income\_group, world\_bank\_region, and who\_r. The data includes entries for countries like Afghanistan, Angola, and Australia, categorized by region and income group.
- Environment View:** Shows the "day2" dataset in the global environment with 194 observations and 8 variables.
- Console:** Shows the command `> View(day2)` again, indicating the function was run from the console.

# Exploring your dataset

## Dimensions and field names

```
1 dim(day2)
```

```
[1] 194 8
```

```
1 names(day2)
```

```
[1] "iso_3166"           "country"          "who_region"  
[4] "income_group"       "world_bank_region" "who_member_state"  
[7] "total_population"   "measles_policy"
```

# Exploring your dataset

Use `table()` to check for missing values

```
1 table(is.na(day2$who_region), useNA = "ifany")
```

FALSE

194

```
1 table(is.na(day2$measles_policy), useNA = "ifany")
```

FALSE TRUE

191 3

# Accessing subsets of data

Using \$ to call vectors

```
1 day2$country
```

```
[1] "Afghanistan"  
[2] "Angola"  
[3] "Albania"  
[4] "Andorra"  
[5] "United Arab Emirates (UAE)"  
[6] "Argentina"  
[7] "Armenia"  
[8] "Antigua and Barbuda"  
[9] "Australia"  
[10] "Austria"  
[11] "Azerbaijan"  
[12] "Burundi"  
[13] "Belgium"  
[14] "Benin"  
[15] "Burkina Faso"  
[16] "Bangladesh"  
[17] "Bulgaria"  
[18] "Bahrain"  
[19] "Bahamas"
```

# Accessing subsets of data

## counts of categories with `table()`

```
1 table(day2$who_region)
```

African Region	Eastern Mediterranean Region
47	21
European Region	Region of the Americas
53	35
South-East Asia Region	Western Pacific Region
11	27

```
1 table(day2$measles_policy)
```

No data

1  
There is a childhood vaccination mandate for at least one vaccination in this country, but vaccination against this disease is not mandated.

18

Vaccination for Measles is not required

91

Vaccination for Measles is required

81

## Now you try!

- Take five minutes to explore this dataset
- How many rows are there?
- How many columns are there?
- Can you use “\$” to select a vector?

# Documenting your data, code, and results

# Goals for today

- Understand the foundations of data management
- Identify some useful global health and health diplomacy datasets
- Load and clean your first dataset in R
  - explore a new dataset while learning about data structures
- Learn best practices for documentation
- Get familiar with Github (by writing poems about unicorns)

# Documentation

- Throughout your analysis, documentation might be the single most important, and also the single most under-appreciated step of the process.
- Documentation is helpful for :
  - **current you:** writing out simple, clear, documentation helps you think clearly
  - **future you:** you'll inevitably forget why you did certain things, even if you can see your old code
  - **others:** hopefully other people will also use your work, and they need to know about it

# Documentation

Different materials, and different types of analysis benefit from different types of documentation.

- **Datasets:** Data dictionaries, rules to generate datasets
- **Code:** Comments in code, README files
- **Analysis/Results:** Methods documentation, in a word document, a google document, or in github

# **Documenting datasets:**

## *Data dictionaries*

The most common way to document a dataset is a data dictionary. At a minimum, data dictionaries:

- Define the columns (“fields”) of a dataset
- Specify what a row corresponds to (e.g., one country)
- Give context about what the data are, including where they come from and how often they are updated

## **Documenting datasets:**

### *Data dictionaries*

Data dictionaries come in lots of formats and flavors. Work with the type that makes the most sense for who you are trying to communicate with.

# Documenting datasets:

## *Example data dictionary: USGS Landsat Sata*

### Acquisition Date

- *Field Definition:* The year, month and day that the scene was acquired.

Format:

YYYY/MM/DD

### Acquisition Quality

- *Field Definition:* Acquisition Quality is a value, expressed as a single digit number, based on (1) errors encountered during archive processing; and/or (2) visible artifacts in the data when manually inspected.

Values:

9 = Excellent (no quality issues or errors detected)

7–8 = Good (minor quality issues and/or errors detected)

### Bias Parameter File Name OLI

- *Field Definition:* The Bias Parameter File used for processing OLI.

# Documenting datasets:

## *Example data dictionary: US Bureau of Labor Statistics*

2022 ATUS Data Dictionary: Public ATUS Interview Data			
Name	Description		File
TEABSRSN	Edited: what was the main reason you were absent from your job last week?		Respondent File
	<b>Edited Universe:</b> TELFS = 2		
	<b>Valid Entries:</b>	1	On layoff (temporary or indefinite)
		2	Slack work/business conditions
		3	Waiting for a new job to begin
		4	Vacation/personal days
		5	Own illness/injury/medical problems
		6	Childcare problems
		7	Other family/personal obligation
		8	Maternity/paternity leave
		9	Labor dispute
		10	Weather affected job
		11	School/training
		12	Civic/military duty
		13	Does not work in the business
		14	Other

# Documenting datasets:

## Example data dictionary: WHO Case Reporting Form

A	B	C	D	E	F
1 Tx section	Variable name	Short label EN	Short label FR	Description	Data type Format
2 Reporting	report_date	Date (DD/MM/YYYY)*: ____ / ____ / ____	Date*: ____ / ____ / ____	Date of reporting	Date DD/MM/YYYY
3 Reporting	report_country	reporting country	Pays*:	Country/territory/national boundary within which the case currently/usually resides. If tracer	String Free text
4 Reporting	report_test_reason	Why the case was tested for COVID-19		Why the case was tested for COVID-19	Coded Coded variables
5 Reporting	report_test_reason_other	Other reason the case was tested for COVID-19		Any other reason the case was tested for COVID-19	String Free text
6 Patient information	patinfo_ID	country case ID*: _____	Numéro d'identification unique*: _____	Unique case identification number (used in country)	String Free text
7 Patient information	patinfo_ageonset	or Age: _____	ou Age*: _____	Age in units on the date of illness onset.	Numeric ###
8 Patient information	patinfo_ageonsetunit	unit of age	Unité de l'âge	Years or months or days	Coded Coded variables
9 Patient information	patinfo_sex	Sex at birth: Male Female	Sexe à la naissance*: Homme Femme	Biological sex. That is the biological differential characteristics (chromosomes, hormonal profile)	Coded Coded variables
10 Patient information	patinfo_idadmin0	where the case was diagnosed, admin level 0 (country)		Administrative level 0: Country where the case was diagnosed.	String Free text
11 Patient information	patinfo_idadmin1	identified Admin Level 1 (province):	Niveau admin 1*(préfecture):	Administrative level 1: First sub-national boundary (e.g. province, state, territory prefecture)	String Free text
12 Patient information	patinfo_resadmin0	place of residence admin level 0		Administrative level 0: Country within which the case's currently/usually resides.	String Free text
13 Clinical status	Lab_date1	Date of first laboratory confirmation	Date prélevement1	Date of first laboratory confirmation	Date DD/MM/YYYY
14 Clinical status	patcourse_dateonset	Date of onset of first symptoms: ____ / ____ / ____	Date de début des premiers symptômes: ____ / ____ / ____	Date of first appearance of the signs or symptoms of the illness/disease.	Date DD/MM/YYYY
15 Clinical status	patcourse_asymp	Patient asymptomatic at time of specimen collection	Patient asymptomatique	Is the case asymptomatic?	Coded Coded variables
16 Clinical status	Comcond_present	Does the patient have any underlying conditions?		Does the patient have any underlying conditions?	Coded Coded variables
17 Clinical status	Comcond_preg	Pregnancy	Grossesse	Is the patient pregnant?	Coded Coded variables
18 Clinical status	Comcond_pregt	Trimester of pregnancy	Trimestre de grossesse		Coded Coded variables
19 Clinical status	Comcond_partum	Post-partum (<6 weeks)	Post-partum (<6 semaines)	Is the patient in the post partum period defined as less than 6 weeks after delivery date	Coded Coded variables
20 Clinical status	Comcond_immu	Immunodeficiency including HIV	Immunodéficience, y compris le VIH	Has the patient an acquired immunodeficiency (HIV) or is the patient treated with drugs that	Coded Coded variables
21 Clinical status	Comcond_cardi	Cardiovascular disease including hypertension		any cardiovascular disease	Coded Coded variables
22 Clinical status	Comcond_diabetes	Diabetes			Coded Coded variables
23 Clinical status	Comcond_liver	Liver disease		any liver diseases	Coded Coded variables
24 Clinical status	Comcond_renal	Renal disease		any renal diseases	Coded Coded variables
25 Clinical status	Comcond_neuro	Chronic neurological or neuromuscular disease			Coded Coded variables
26 Clinical status	Comcond_malig	Malignancy	Malignité		Coded Coded variables
27 Clinical status	Comcond_lung	Chronic lung disease	Maladie aiguë ou chronique associée:		Coded Coded variables
28 Clinical status	Comcond_other	Other, specify	Autre spécifier	Describe other underlying conditions and comorbidity	String Free text
29 Clinical status	patcourse_admit	admission to hospital?:	Hospitalisation* ?:	Was the case hospitalized, admitted to a hospital or other health facility as an inpatient?	Coded Coded variables
30 Clinical status	patcourse_presHCF	For this episode, date first admitted in hospital: Pour cet épisode, quelle est la date à laquelle le cas a été admis dans une établissement de santé.	Date the case was first admitted to any health facility.		Date DD/MM/YYYY

WHO Data Dictionary for Case-Based Reporting form. Available online:  
<https://www.who.int/publications/m/item/data-dictionary-for-case-based-reporting-form>

# Documenting datasets:

## *Example data dictionary template*

<p>[optional: add logo here]</p> <p>This data dictionary is intended to document the dataset(s) accessible at [URL]</p> <p>The data dictionary was last updated on [date]</p> <p>If you have any questions, please reach out to [name] at [email or other contact information]</p>				
<b>Data Dictionary - Tables/Datasets</b>				
Table	Description	Resolution(s)	Update frequency	References or other notes
example: measles_cases	example: table with information on country-level data on measles caseload, per year	example: one row per IHR member state per year	example: one-time (not updated)	example: teaching dataset

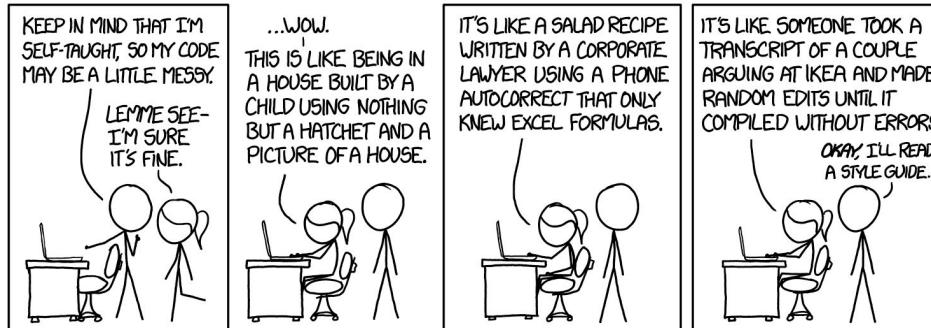
Data Science Basics in R Data Dictionary Template. Steph Eaneff.

Available online: <https://github.com/seaneff/data-science-basics-2024/blob/main/extras/Data%20dictionary%20template.xlsx>

# Documenting code:

## comments

- As we start writing code, it's helpful to leave "notes" for yourself and others reminding yourself what you did
- Comments can also help break code into sections



Comic by XKCD

# Documenting code: comments

```
## #####
## Setup #####
#####

## Load libraries
library(dplyr) ## reshape, reformat, recode data: https://dplyr.tidyverse.org/reference/recode.html
library(ggplot2) ## for plotting: https://ggplot2.tidyverse.org/
library(scales) ## for commas on axes of plots
library(treemap) ## for treemap visual

## #####
## Read in data #####
#####

line_items <- read_excel("calculator-tool/jee3_costing_worksheet.xlsx",
                         sheet = "Line items (JEE 3)")

unit_costs <- read_excel("calculator-tool/jee3_costing_worksheet.xlsx",
                         sheet = "Unit costs")

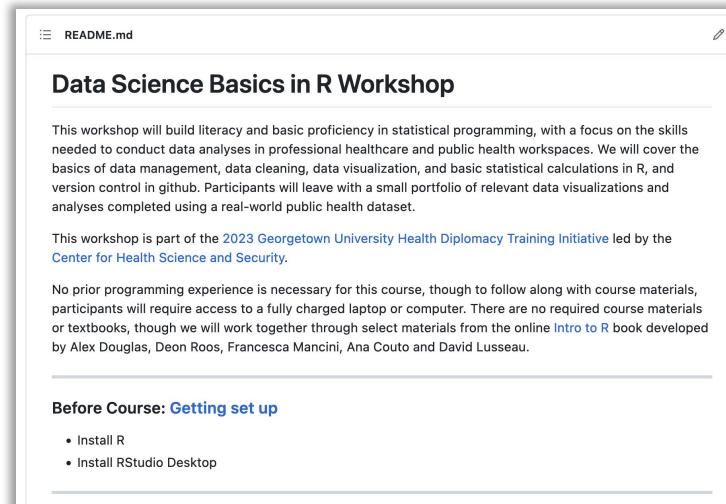
multipliers <- read_excel("calculator-tool/jee3_costing_worksheet.xlsx",
                         sheet = "Multipliers")
```

# Documenting code:

## README files

- README files summarize the contents of a specific folder
- Common in github, rendered using Markdown

```
README.md
1 # Data Science Basics in R Workshop
2
3 This workshop will build literacy and basic proficiency in statistical
4 programming, with a focus on the skills needed to conduct data
5 analyses in professional healthcare and public health workspaces. We
6 will cover the basics of data management, data cleaning, data
7 visualization, and basic statistical calculations in R, and version
control in github. Participants will leave with a small portfolio of
relevant data visualizations and analyses completed using a real-world
public health dataset.
8
9 This workshop is part of the [2023 Georgetown University Health
10 Diplomacy Training Initiative](https://ghss.georgetown.edu/health-diplomacy-training-initiative/) led by the [Center for Health
11 Science and Security](https://ghss.georgetown.edu/).
12
13 No prior programming experience is necessary for this course, though
14 to follow along with course materials, participants will require
15 access to a fully charged laptop or computer. There are no required
16 course materials or textbooks, though we will work together through
17 select materials from the online [Intro to R](https://intro2r.com/)
18 book developed by Alex Douglas, Deon Roos, Francesca Mancini, Ana
19 Couto and David Lusseau.
```



# Reproducible results

We won't cover it in detail in this workshop, but R is a great way to ensure that your code, and your results, are well documented and reproducible

- Use a single version of a **clean dataset**
- Document and version your **data cleaning decisions**
- Store your **code, results, & documentation** in one spot

# Reproducible results

## Code versioning in github

The screenshot shows a GitHub user profile for `seaneff`. The main navigation bar includes `Pull requests`, `Issues`, `Codespaces`, `Marketplace`, and `Explore`. A modal window titled "Join GitHub Global Campus!" is open, describing the program's purpose of providing practical industry knowledge through access to tools, events, learning resources, and a student community. It features several cards: "Follow your Expert", "Breaking into tech: internship edition with Helen Huang", "Level up your code with TwilioQuest", "Learning by teaching for your community - Cassidy Williams", "Popular offers you have not claimed", "Curated Experiences with popular offers", "Virtual event kit", "Connect your local Expert", "View projects at our gallery", "Learn more about an event", "Watch a Campus TV episode", "Web meet 202", and "Web Meet". A large green button at the bottom right of the modal says "Join Global Campus". Below the modal, the "For you" feed shows a message: "Welcome to the new feed! We're updating the cards and ranking all the time, so check back regularly. At first, you might need to follow some people or star some repositories to get started." The sidebar on the left lists "Top Repositories" and "Recent activity". The main content area also includes sections for "Your teams" and "Explore repositories".

**Latest changes**

- 22 minutes ago GitHub Actions: All Actions will run on Node16 instead of Node12 by default
- 1 hour ago GitHub Actions: You can now disable repo level self-hosted runners in an Enterprise and Organization
- 20 hours ago Dependabot version updates now supports pnpm
- Yesterday GitHub Advanced Security trial now available on GitHub Enterprise Cloud  
View changelog →

**Explore repositories**

`zulip / zulip`

Zulip server and web application. Open-source team chat that helps teams stay productive and focused.

18k Python

`Ebazhanov / linkedin-skill-assessments-quizzes`

Full reference of LinkedIn answers 2023 for skill assessments (aws-lambda, rest-api, javascript, react, git, html, jquery, mongodb, java, Go, python, machine-learning, power-point) linkedin excel ...

25.2k Python

# Github basics

# Goals for today

- Understand the foundations of data management
- Identify some useful global health and health diplomacy datasets
- Load and clean your first dataset in R
  - explore a new dataset while learning about data structures
- Learn best practices for documentation
- Get familiar with Github (by writing poems about unicorns)

# **Reminder from yesterday**

*What is github?*

- Have you ever saved a bunch of versions of a paper on your computer with different file names at different dates or times of day?
- Backups are useful to save progress, understand what we've done before, and look into problems/bugs
- Github is a tool do help do this with code

# Github, poetry, and unicorns

A brief adventure on youtube

<https://www.youtube.com/watch?v=BCQHnInPusY>



1.1: Introduction - Git and GitHub for Poets



The Coding Train  
1.66M subscribers

Join

Subscribe

12K

Share

Thanks

...

# Github organization

## *README files*

- README files make it easier for your collaborators to find and understand work within your github repository
- In general, readme files should contain:
  - What the project/code does
  - Why the project/code is useful
  - Where people can get help or find more info

Read more on [github](#)

# Github organization

## creating a README file

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*



seaneff

Repository name \*



teaching example

Your new repository will be created as teaching-example.

The repository name can only contain ASCII letters, digits, and the characters ., -, and \_.

Great repository names are short and memorable. Need inspiration? How about [shiny-octo-carnival](#) ?

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

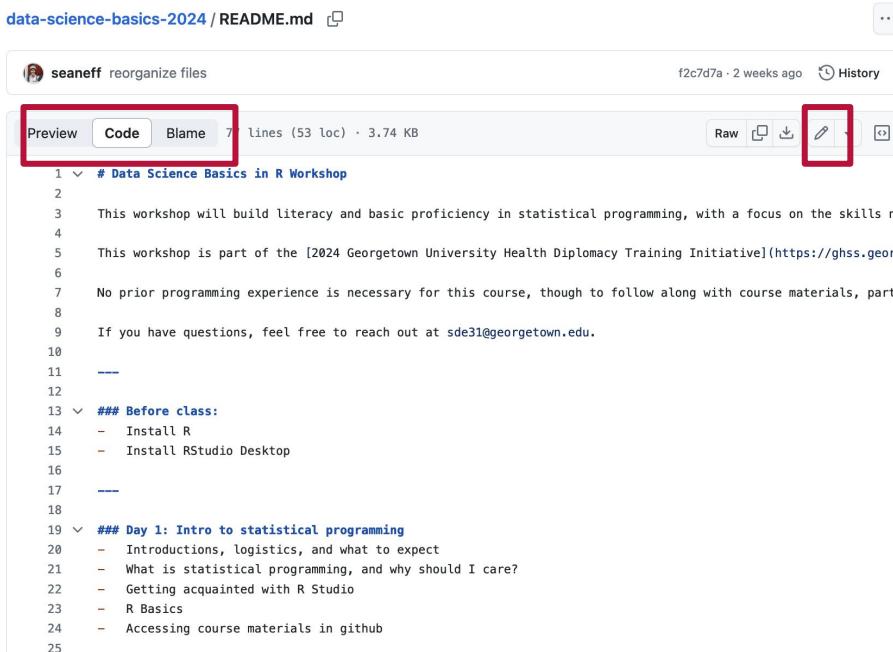
- When you create a new github repository, you can click a box to add a README file
- If you forget to do this, you can always simply add a new file to your repo and title it README.md, github will recognize this

Read more on [github](#)

# Github organization

## Editing a README file

data-science-basics-2024 / README.md 



seaneff reorganize files f2c7d7a · 2 weeks ago History

Preview Code Blame 71 lines (53 loc) · 3.74 KB

Raw    

```
1 # Data Science Basics in R Workshop
2
3 This workshop will build literacy and basic proficiency in statistical programming, with a focus on the skills ne
4
5 This workshop is part of the [2024 Georgetown University Health Diplomacy Training Initiative](https://ghss.georg
6
7 No prior programming experience is necessary for this course, though to follow along with course materials, parti
8
9 If you have questions, feel free to reach out at sde31@georgetown.edu.
10
11 ---
12
13 ## Before class:
14 - Install R
15 - Install RStudio Desktop
16
17 ---
18
19 ## Day 1: Intro to statistical programming
20 - Introductions, logistics, and what to expect
21 - What is statistical programming, and why should I care?
22 - Getting acquainted with R Studio
23 - R Basics
24 - Accessing course materials in github
25
```

README

## Data Science Basics in R Workshop

This workshop will build literacy and basic proficiency in statistical programming, with a focus on the skills needed to conduct data analyses in professional healthcare and public health workspaces. We will cover the basics of data management, data cleaning, data visualization, and basic statistical calculations in R, and version control in github. Participants will leave with a small portfolio of relevant data visualizations and analyses completed using a real-world public health dataset.

This workshop is part of the [2024 Georgetown University Health Diplomacy Training Initiative](#) led by the [Center for Health Science and Security](#).

No prior programming experience is necessary for this course, though to follow along with course materials, participants will require access to a fully charged laptop or computer. There are no required course materials or textbooks, though we will work together through select materials from the online [Intro to R](#) book developed by Alex Douglas, Deon Roos, Francesca Mancini, Ana Couto and David Lusseau.

If you have questions, feel free to reach out at [sde31@georgetown.edu](mailto:sde31@georgetown.edu).

### Before class:

- Install R
- Install RStudio Desktop

### Day 1: Intro to statistical programming

- Introductions, logistics, and what to expect

Course [readme file](#)  
Main course [github page](#)

# Github organization

## *Organizing files in github*

- There are lots of different ways to organize files in github, and there's no one *best* solution. You should think about who is using the repo, and how, when you decide how to structure it.
- One common approach is to use different folders within one repo:
  - **data/** for datasets that open/sharable and small enough for git
  - **code/** for your analysis code, which may be split up further
  - **results/** for key outputs
  - **figures/** for figures, especially if you're publishing

# Your turn!

Create a github account at <https://github.com/>

# Recap for today

What we talked about

- Understand the foundations of data management
- Identify some useful global health and health diplomacy datasets
- Load and clean your first dataset in R
  - explore a new dataset while learning about data structures
- Learn best practices for documentation
- Get familiar with Github (by writing poems about unicorns)

# Plan for tomorrow

## Exploratory data analysis

- What is exploratory data analysis (EDA)?
- Calculate basic descriptive statistics in R
- Explore different strategies for data visualization
- Build your first data visualizations in R

**Thank you!**

**See you tomorrow.**

*Please come with a fully charged laptop.*