World Health Organization
REGIONAL OFFICE FOR THE Eastern Mediterranean

منظمة الصحة العالمية
المكتب الإقليمي لشرق المتوسط

REGIONAL TRAINING ON CAPACITY DEVELOPMENT OF DATA ANALYTICS AND DISSEMINATION USING "R" SOFTWARE

AMMAN, JORDAN, 3 - 7 DECEMBER, 2023

Day #2

HEALTH FOR ALL BY ALL
a call for solidarity and action

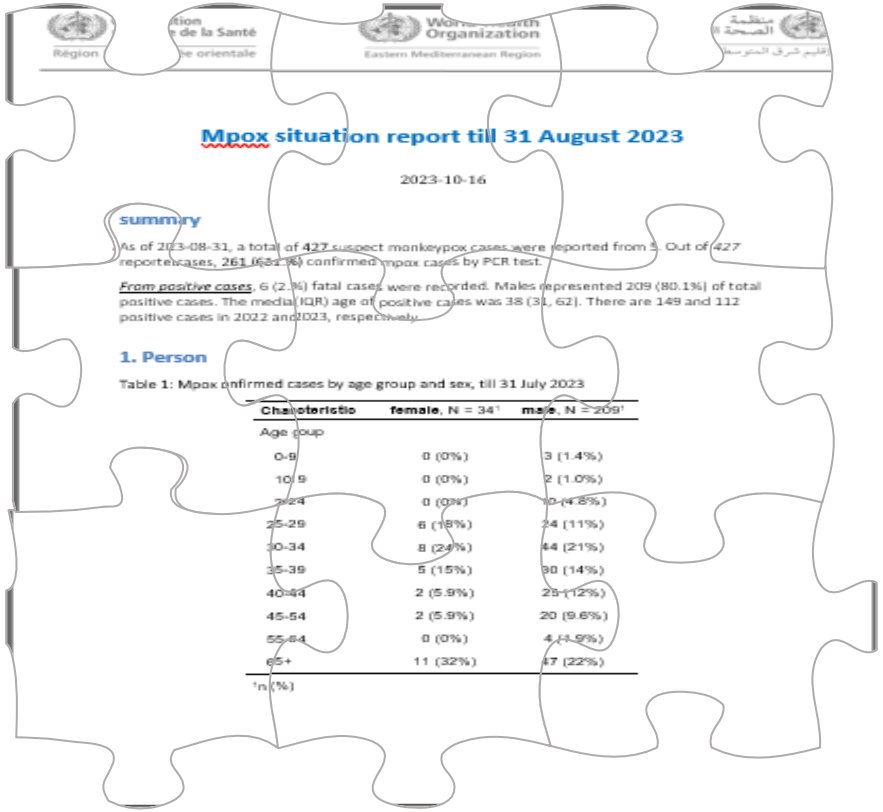
World Health Organization
REGIONAL OFFICE FOR THE Eastern Mediterranean

Outline

- Wrap-up
- Import data to R
- Exploring imported data
- Data processing – part I
  - mutate
  - Select
  - Filter
  - Working with variable names
    - clean_names
    - rename
  - recode
  - De-duplication
- Q&A

Desired output at end of the training

Desired output at end of the training



Région de la Santé orientale

World Health Organization

Eastern Mediterranean Region

منظمة الصحة العالمية

إقليم شرق المتوسط

Mpox situation report till 31 August 2023

2023-10-16

summary

As of 2023-08-31, a total of 427 suspect monkeypox cases were reported from 5. Out of 427 reported cases, 261 (61.1%) confirmed mpox cases by PCR test.

From positive cases, 6 (2.3%) fatal cases were recorded. Males represented 209 (80.1%) of total positive cases. The media (IQR) age of positive cases was 38 (31, 62). There are 149 and 112 positive cases in 2022 and 2023, respectively.

1. Person

Table 1: Mpox confirmed cases by age group and sex, till 31 July 2023

Characteristic	female, N = 34[1]	male, N = 209[1]
Age group		
0-9	0 (0%)	3 (1.4%)
10-19	0 (0%)	2 (1.0%)
20-24	0 (0%)	10 (4.8%)
25-29	6 (18%)	24 (11%)
30-34	8 (24%)	44 (21%)
35-39	5 (15%)	30 (14%)
40-44	2 (5.9%)	25 (12%)
45-54	2 (5.9%)	20 (9.6%)
55-64	0 (0%)	4 (1.9%)
65+	11 (32%)	47 (22%)

[1]n (%)

HEALTH FOR ALL BY ALL

a call for solidarity and action

World Health Organization

REGIONAL OFFICE FOR THE Eastern Mediterranean

Data importation to R

Session 2 Agenda

- 9:00 – 9: 30 (30 min): **Wrap-up**
- 9:30 – 9:50 (20 min): **Presentation "Data importation to R"**
- 9:50 – 10:20 (30 min): **Demonstration**
- 10:20 – 10:40 (20 min): **Stretching / coffee break**
- 10:40 – 12:30 (1.8 hr): **Practice/Exercise**
- 12:30 – 13:00 (30 min): **Quick debrief/ Q&A**
- 13:00 – 14:00 (60 min): **Lunch**
- 14:00 – 14:20 (20 min): presentation "Data processing – part I"
- 14:20 – 14:50 (30 min): Demonstration
- 14:50 – 15:10 (20 min): Stretching / coffee break
- 15:10 – 16:30 (80 min): Practice/Exercise
- 16:30 – 17:00 (30 min): Quick debrief/ Q&A

Importing data to R

- Different file formats can be imported to R:
  - Spreadsheets (*most common*): **.csv**, **.xls**, **.xlsx**
  - **SPSS dataset: .sav**
  - **Stata dataset: .dta**
  - **SAS dataset: .sas7sdat**
  - **SQL databases**
  - **JSON files**

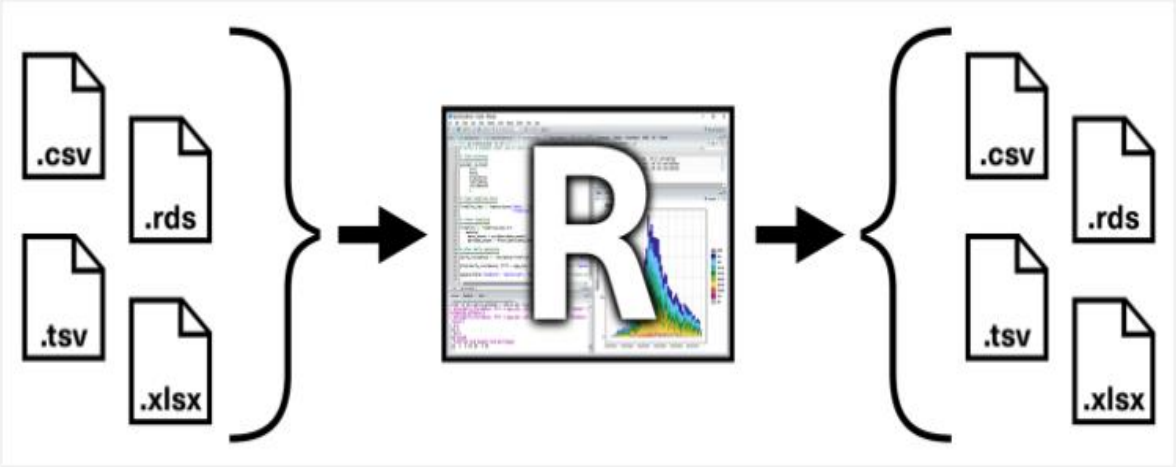
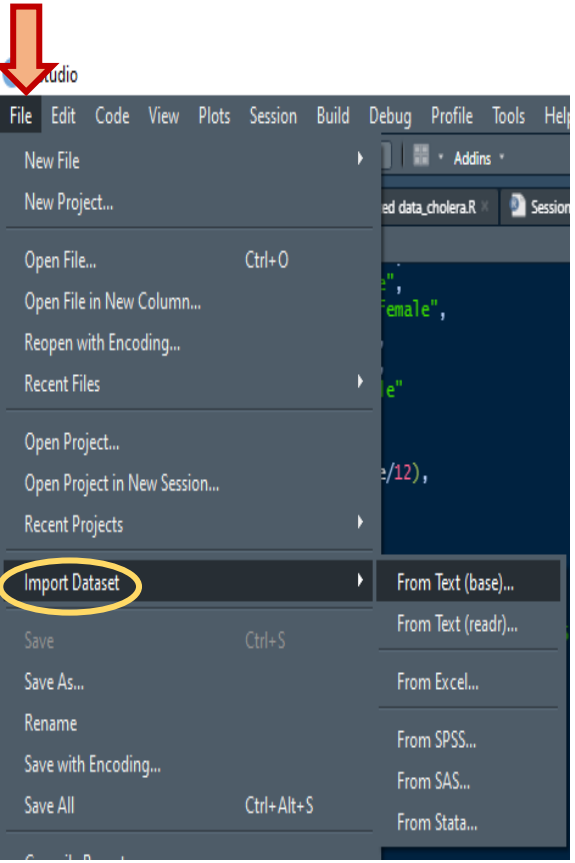


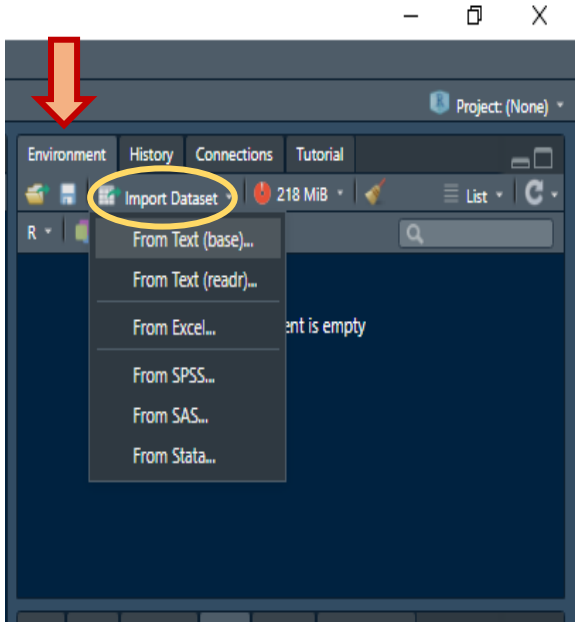
Image source: Epi R handbook

How to import data to R?

What is the file format?!!

R built-in options





Installed packages

Many packages are available to support importing different datasets to R. We will use some of:

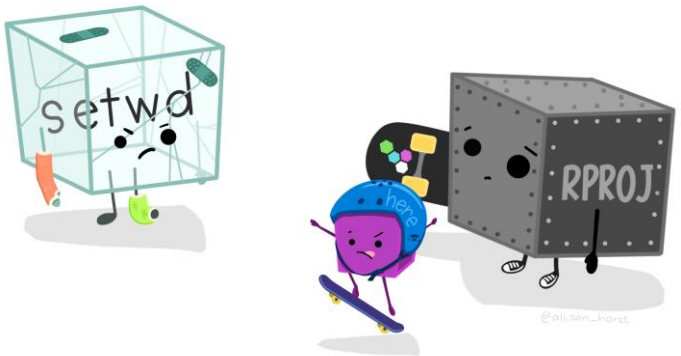
- Specific for .csv
  - readr::read_csv()
- Many file formats
  - rio::import()
  - data.table::fread

Where is that file?!!

HEALTH FOR ALL BY ALL
a call for solidarity and action

World Health Organization
REGIONAL OFFICE FOR THE Eastern Mediterranean

How to import data to R? Cont.



Where is that file?!!

Absolute path

- A specific path for the file that is unique to the user's computer
- Against the concept of reproducible analysis

```
"C:/Users/abdelgawadb/Desktop/Regional R training workshop/Data/Cholera case study/cholera_20231102.csv"
```

Relative path

- Fixing the file path to the root of R project
- Best for reproducibility
- here::here() is used

```
import(here("Data", "Cholera case study", "cholera_20231102.csv"))
```

Image source:Stats-illustrations

Data checking

Initial checking of the imported dataset (data frame):

- Ensure that data is correctly imported
- Explore for any missing, duplicates, inconsistencies

```
> dim(cholera)
[1] 108  53
```

Dimensions of the data frame (rows, columns)

```
> nrow(cholera)
[1] 108
> ncol(cholera)
[1] 53
```

Separately, number of rows/observations
Number of columns/variables

```
> str(cholera)
'data.frame':   108 obs. of  53 variables:
 $ recordx_id   : chr  "cholera-001" "cholera-003" "cholera-005" "cholera-007" ...
 $ hospital     : chr  "Place 1" "Place 2" "Place 1" "Place 2" ...
 $ sex          : chr  "female" "male" "female" "female" ...
 $ age          : int  13 27 7 20 7 9 15 49 6 6 ...
 $ Age.unit     : chr  "year" "year" "year" "month" ...
 $ adm.date     : chr  "2023-07-31" "2023-07-07" "2023-07-17" "2023-07-24" ...
 $ temp         : num  NA NA 36.4 36.9 37.1 36.8 36.2 NA 37 NA ...
 $ cons         : chr  "" "" "aware" "aware" ...
 $ cap_fill     : chr  "" "no" "no" "yes" ...
 $ pulse        : chr  "" "no" "yes" "no" ...
 $ wt.kg.       : num  NA 54 14.9 10.1 12 13.5 14.4 60 10 15 ...
 $ ht..cm.      : int  162 167 93 156 109 147 121 105 163 142 ...
 $ muac         : logi  NA NA NA NA NA NA ...
```

Display the structure of the data frame and its variables

Data checking Cont.

- Checking numeric and date variables

```
> summary(cholera$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   4.00    7.00    9.00   16.59   24.00   56.00
> summary(cholera$onset_date)
        Min.      1st Qu.       Median         Mean      3rd Qu.         Max.         NA's
"2021-05-06" "2022-02-11" "2022-12-12" "2022-09-12" "2023-07-02" "2023-07-29"          "4"
```

- Checking any variable

```
> table(cholera$f_diag, useNA = "always")

 confirmed not a case   probable    suspect       <NA>
       24          4         74          6          0
```

- *Other useful functions:*
  - skimr::skim()
  - class()
  - head()/tail()
  - duplicated()
  - is.na()
  - Other simple mathematical functions

Data checking Cont.

The concept of tidy data

"TIDY DATA is a standard way of mapping the meaning of a dataset to its structure."

–HADLEY WICKHAM

In tidy data:

- each variable forms a column
- each observation forms a row
- each cell is a single measurement

each column a variable

id	name	color
1	floof	gray
2	max	black
3	cat	orange
4	donut	gray
5	merlin	black
6	panda	calico

each row an observation

Wickham, H. (2014). Tidy Data. Journal of Statistical Software *59* (10). DOI: 10.18637/jss.v059.i10

Image source:Stats-illustrations

Data checking Cont.

The concept of tidy data

Hospital	Male	Female
Hosp. A	5	30
Hosp. B	10	40
Hosp. C	30	5
Hosp. D	42	18

Hospital	Gender	Value
Hosp. A	Male	5
Hosp. A	Female	30
Hosp. B	Male	10
Hosp. B	Female	40
Hosp. C	Male	30
Hosp. C	Female	5
Hosp. D	Male	42
Hosp. D	Female	18

Demonstration

Exercise: Importing and Exploring Cholera Data

- Open your training R project “Regional_R_training”
- Create a new R script “cholera.R” and save it to the scripts folder
- Structure the script with sections (e.g about the script, load packages,….)
- **Load** the required packages
- **Import** cholera case study “cholera_20231102.csv” dataset located in the “Data” folder, use one of the following:
  - import + here functions
  - fread function

Make sure the relevant packages are installed/loaded!

Do not forget to assign the imported data to an object “cholera0”

- **Explore** the imported dataset:
  - What are the number of observations and variables?
  - What is the class() of the imported data?
  - Summary() of the imported data
  - What is the class of “admission date” and “outcome date” variables?

Exercise cont…

- Open your training R project
- Create a new R script "cholera.R" and save to scripts folder
- Import cholera case study "cholera_20231102.csv" dataset
- **Explore** the imported dataset:
  - How many duplicates are in the imported dataset?
  - What is the latest symptom onset date?
  - What is the range and the mean of reported cases' age? [🚩 *how would you read it?!!*]
  - What is the distribution of sex among reported cases? [**HINT:** table()]

+ Bonus!!

- What are the names of the reported countries? [**HINT:** unique()]
- Cross-tabulate sex "sex" with final diagnosis "f_diag" [**HINT:** table()]

Data processing – part I

Session 2 Agenda

- 9:00 – 9: 30 (30 min): Wrap-up
- 9:30 – 9:50 (20 min): Presentation “Data importation to R”
- 9:50 – 10:20 (30 min): Demonstration
- 10:20 – 10:40 (20 min): Stretching / coffee break
- 10:40 – 12:30 (1.8 hr): Practice/Exercise
- 12:30 – 13:00 (30 min): Quick debrief/ Q&A
- 13:00 – 14:00 (60 min): Lunch
- 14:00 – 14:20 (20 min): **presentation “Data processing – part I”**
- 14:20 – 14:50 (30 min): **Demonstration**
- 14:50 – 15:10 (20 min): **Stretching / coffee break**
- 15:10 – 16:30 (80 min): **Practice/Exercise**
- 16:30 – 17:00 (30 min): **Quick debrief/ Q&A**

Data processing

- Data steps to prepare the dataset for analysis:

1. <u>On dataset level:</u> **<u>Proposed function</u>**

- Drop/keep columns → select()
- De-duplicate → distinct(), duplicated()
- Filter specific observations → filter()

2. <u>On variable level:</u>

- Variable Name → clean_names(), rename()
- Manage inconsistencies → recode()
- Change class of variable → as.Date(), as.numeric(),…
- Create a new variable → mutate(), case_when(), if_else()
- Modify same variable → mutate(), case_when(), if_else()

Commonly used function

Package::function	Utility
%>%	Arithmetic operators
dplyr::filter()	Subset **rows**
dplyr::select()	Subset **columns**
dplyr::mutate()	Create & transform columns
lubridate::mdy(), dmy(), ymd()	Tell R how to understand names
janitor::clean_names	Standardize names
dplyr::rename()	Manual renaming
dplyr::case_when()	Complex logical re-coding of values
ifelse()	Simple logical re-coding of values
dplyr::recode()	Re-code values in a column
as.Date(), as.numeric()	Convert the class of a column

Source: Epi R handbook

Demonstration

Exercise: Data Wrangling in R: Cholera Case Study Processing

- Open your training R project "Regional_R_training"
- Open the R script "cholera.R"
- Add a new section "data processing 1" and do the following data steps: (*create a new object "cholera"*)

  1. Standardize the column names [**HINT:** clean_names()]

  ✓*It is clear now why column naming is important!*

  2. Rename "f_diag" to a more meaningful name "case_cat"
  3. Fix inconsistencies in "sex" entries to male & female only [**HINT:** recode()]
  4. Create a new column "age_cat" in years:
     - ***First:*** create new column "age_yr" to standardize all age units to be in years [**HINT:** mutate() & case_when()]
     - ***Second:*** now, create the "age_cat" column [**HINT:** cut() or epikit::age_categories()]
  5. Drop all unnecessary columns (e.g. blank ones)

  + Bonus!!

  6. Create a body mass index "bmi" column as weight(kg)/height(m^2)


Q&A