**R session**

**Data visualization for public health**

**Thursday 08/02**

**8:30 – 10:30 CET**

**Agenda**

|  |  |  |
| --- | --- | --- |
| **Introduction** | **15 min** | **8:30 – 8:45** |
| **Reminders about data visualization with {ggplot2}** | **15 min** | **8:45 – 9:45** |
| **Managing dates**   * Live demo * Practice! | **20 min** |  |
| **Making epicurves**   * Live demo * Practice! * Debriefing | **60 min** | **10:15 – 10:30** |
| **Wrap-up** | **10 min** | **10:15 – 10:30** |

Material:

* PPT
* epiweek\_cheatsheet\_DRAFT.pdf
* Scripts
  + 6\_ebola\_after\_ggplot\_session\_2.R
  + 8\_ebola\_after\_advanced\_ggplot.R

**Email sent**

Dear all,

I hope you’re doing well!

This is the second R session, the topic is “Data visualization for public health”.

Preliminary agenda:

* Intro (10 min)
* A few reminders about Data visualization with {ggplot2} (10 min)
* Managing dates (20 min)
  + live Demo
  + Practice
* Making epicurves (50 min)
  + live Demo
  + Practice
* Debrief - question/response (20 min)
* Session wrap-up (10 min)

Preliminary work (optional):

This is recommended to complete some preparation work:

* Watch the video from the module 8 ‘Data visualization for public health’ available on the applied epi website:

<https://files.appliedepi.org/media/mod8>

* Read the epiweek cheatsheet (attached in the invite)
* Explore the script “8\_ebola\_after\_advanced\_ggplot.R” that is available in the folder “intro\_course\scripts\backup”.

You can also review the training material from the Module 6 of the training: Data visualization with {ggplot2}

* Video

<https://files.appliedepi.org/media/mod6>

* Slides

<https://appliedepi.github.io/slides/intro/en/module_6-1/module_6-1.html#1>

<https://appliedepi.github.io/slides/intro/en/module_6-2/module_6-2.html#1>

**Managing dates - Live demo**

How dates are handled in R

How epi weeks can be created

You can look at the pdf on your own screen

This is a table, screen shot of a table

dates from ‘2021-12-27’ to and in 2022

red line represent the new calendar year

11- 01-2023

Apply the mutate function to create new variables

package and function that were used to create the new variables

For example, this is **day function** from the **lubridate package** and it returns the numeric day of the month.

(Extract a number from a date and print it)

Likewise **function wday**, return abbreviation abbreviation of the day of the week. Epiweekens what days it starts

Format functions from base R

Syntax

strptime

<https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/strptime>

also ?strptime

show some examples %d, %m %Y

%j jay is the julian day 1 to 365 looking at many years

%U week number Sunday start

%W is you want a monday start

Gap when there’s a gap between end of the week and the start of week 1 one new year.

To be aware of this!!!

What would be disadvantage. More than 1 year ???

**tsibble package** (time series analysis)

**yearweek** in a much better format

change start of the week

extend the end of the week if there’s a gap

Sometime it’s good to have epi-weekyear as a date

axis handle differently character than dates

**as.Date()** on a **yearweek()** will return the first day of the week

**aweek package**, alternative to **tsibble**

**lubridate floor\_week**

basically it does the same as **as.Date(yearweek)**

round down to nearest Monday

lubridate has also 2 these function

**Making an epicurve key steps - PPT**

From 18:14

Step 1 – not mandatory

Depending on what you want to see in your x-axis you may want to 2 new columns using mutate

* yearweek of the date of onset
* the Monday of that week of onset

Step 2 – not mandatory

A list of the different breakpoint

**Making an epicurve key steps - PPT**

From 18:30

We can say ggplot and

This 2 functions, the base skeleton of what we are doing

We need to specify the dataset

We map the columns in the data

It’s not clear the default of the bins width

binwidth = 1 => by 1 unit so 1 day

every bin is 1 day

## step 3 - binwidth = 7

The start of the first bin is going to be at the date of first date of onset

It may be a Monday, it may be a Friday, etc

## step 4

# 1. define the break points for the histogram bins

The at which the bins are starting

seq => sequence of number

seq.Date = > it’s expecting dates

min(combined$date\_onset, na.rm=T) => earliest none missing dates in the data.frame

max(combined$date\_onset, na.rm=T) => max in the date of onset

dynamic codes that updates when the data updates

seq.Date(

from = min(combined$date\_onset, na.rm=T),

to = max(combined$date\_onset, na.rm=T),

by ="week")

=> a sequence of dates starting on the first case onset date

wday(ymd("2014-05-06"), label = T)

strat a Tuesday but I want a Monday

floor\_date of the minimum date

? floor\_date

* takes a date-time object and rounds it down to the nearest boundary of the specified time unit.

The unit is week => round down to the nearest start. week\_start = 1 (for Monday)