



## User Guide

---

AMRapp Version 1.0

## Table of Contents

1 Introduction .....	3
2 Set Up .....	5
3 AMRapp functions .....	9
4 AMRapp interactive dashboard .....	12
5 AMRapp visualisations .....	19

# 1 Introduction

AMRapp is a flexible tool with implemented laboratory specimen data processing producing informative and interactive trend plots and data tables for use in further analysis.

This user manual follows use-case scenarios and provides information in two parts:

- **Section 3:** refer to this section to run the AMRapp data processing and plot generation as a script within RStudio. This section describes the data processing and plot functionalities.
- **Section 4:** refer to this section to run AMRapp as an interactive R-Shiny dashboard. This section describes the AMR trend plotting options and investigative tool functionality for the uploaded data.

**Section 5** Shows the graphs that can be plotted with either the stepwise functions (Section3) or the Interactive dashboard (Section 4).

## Prerequisites

R and RStudio integrated development environment (IDE) (versions  $\geq 3.6.1$ ) are required. If not yet installed on your computer, download the newest versions from the following links:

- **RStudio (free version):** <https://www.rstudio.com/products/rstudio/download/>
- **R:** <https://cran.r-project.org>

Additionally, AMRapp requires a laboratory susceptibility data file in the correct template format as input files. A template file ("Amrapp\_template.csv") and example data set ("Amrapp\_template\_testdata.csv") are available as part of the AMRapp installation. Details of the field requirements are listed in **Table 1**.

## Availability

Code and Documentation can be accessed at: <https://github.com/RG-abx/AMRapp>

## Important notes

1. Be sure to follow steps **A.** and **B.** and **C.** in Section 2 (Set-Up) before continuing.

**Table 1.** Variable list, description and data type expected for use with AMRapp

<b>Template Field</b>	<b>Description</b>	<b>Data Type</b>
<b>id</b>	Unique specimen and antimicrobial susceptibility test identifier. Note, conceptually these should be distinct from each other	Character
<b>pt_id</b>	Unique patient identifier. This should be an encoded variable that identifies unique patients within the dataset. For instance, a patient NHS number would work here, or a patient episode identifier. Multiple repeat pt_id can be included, as a patient can have multiple susceptibility tests per specimen and multiple specimens.	Character
<b>genus</b>	The genus of the organism identified within the specimen	Character
<b>species</b>	The species name of the organism identified within the specimen	Character
<b>date</b>	The date of the specimen and susceptibility test in the format DD/MM/YYYY	Date
<b>antimicrobial</b>	Antimicrobial code or antimicrobial name for the susceptibility test	Character
<b>result</b>	The sensitivity result code for the susceptibility test. S=Susceptible, R=Resistant, N=Not tested, I=intermediate. There should be no blanks	Character

## 2 Set Up

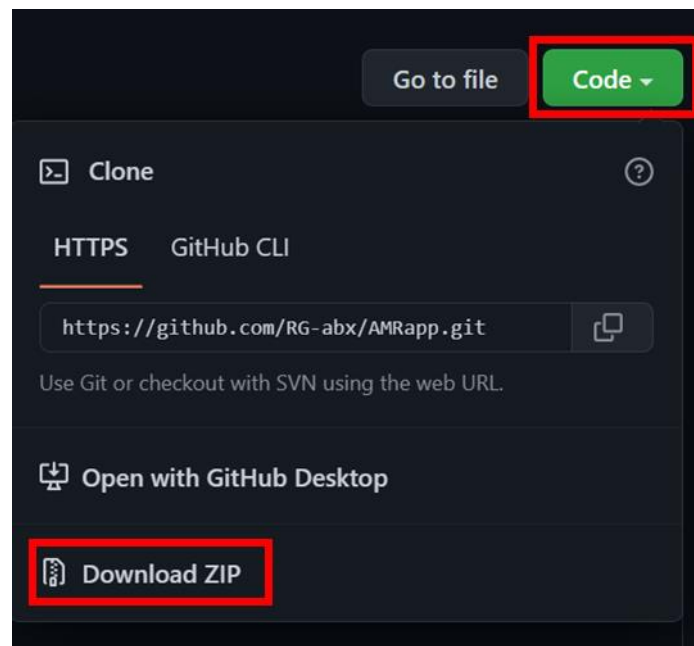
### A. Test Files

For testing the AMRapp functionality (and to make this user manual easier to follow), the following file has been provided:

- Amrapp\_template\_testdata.csv

Please follow the steps below to download the testing files and set your working directory:

1. **Create a new folder** on your computer, naming it AMRapp
2. **Download the AMRapp repository** from GitHub, by clicking on the green “Code” button on the upper-right corner and selecting the “Download ZIP” option. **(Figure 1)**



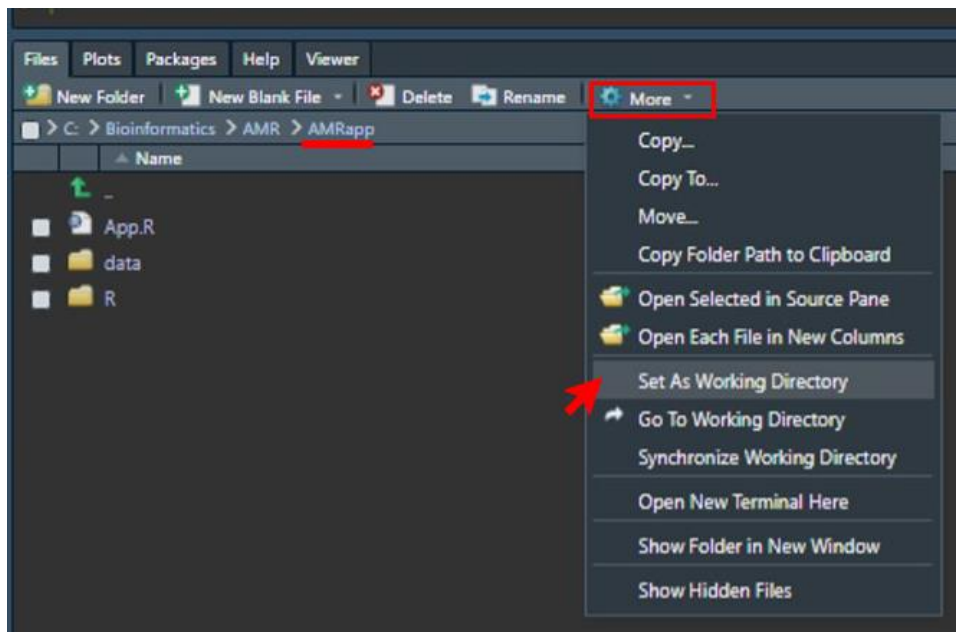
**Figure 1.** GitHub screenshot, showing the required buttons to click.

3. **Unzip** the downloaded repository files
4. Open RStudio and set you working directory, either by
  - running the following command:

```
setwd("~/Users/...yourPath.../AMRapp")
```

To run commands on the console of RStudio, paste them in and press enter on your keyboard.

- or manually, by searching the folder inside the bottom-right window in RStudio (inside the "Files" panel), clicking on the gear button and selecting "Set as working directory" (**Figure 2**)



**Figure 2.** Screenshot of RStudio, showing how to manually set the working directory.

**Please note** that steps 2. and 3. can be skipped if you wish to analyse your own pre-formatted data files.

## B. Installing AMRapp and dependencies

To start your analyses, you will need to install and load the AMRapp from GitHub.

This can be done following two simple steps, on RStudio:

### 2 Install the AMRapp package using the following command:

```
□ devtools::install_github("RG-abx/AMRapp/AMRapp")
```

### 3 Load the AMRapp package with the following command:

```
□ library(AMRapp)
```

## C. Reading the laboratory data AMR file

After successfully loading the AMRapp package, the formatted laboratory data AMR file needs to be read and loaded inside the working directory. Be aware that **this step is common for every implemented method and is needed to run the rest of the package functions.**

Using the provided laboratory data AMR file, proceed running the following command:

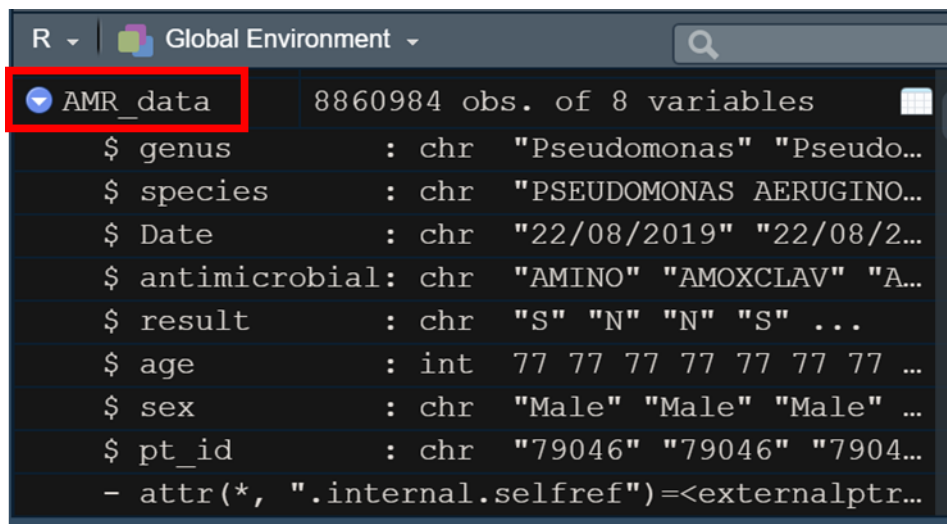
```
□ AMR_data <-  
  data_import("data/Amrapp_template_testdata.csv")
```

**Important** – this step might take time to complete (approximately 1 minute). This will vary depending on the size of any locally supplied data. It is strongly recommended to leave RStudio open and running, to properly process the data, until the red button (found at the top-right corner of the console) disappears. **(Figure 3)**



**Figure 3.** Screenshot of the console in RStudio. Highlighted in red can be found the “STOP” button, which will disappear once the AMR\_data object gets loaded inside the working environment.

After the CSV file gets loaded into the environment (**Figure 4**), you can proceed following the steps to apply the desired methods described in **Section 3** (running AMRapp functions) or **Section 4** (running the AMRapp interactive R Shiny application).



**Figure 4.** Screenshot of the loaded AMR\_data object inside the RStudio working environment.

**Important note** – if you intend to use your own data set to review the AMRapp functions, you will need to run the `data_import()` function, mentioned at the beginning of this section:

```
□ AMR_data <-  
  data_import("yourAMRtemplateFile.csv")
```



### 3 AMRapp functions

## 1 STEPWISE Processing and Plotting

Functions included in this section will apply process the data and return the plots to the user in a multistep process.

**Important note** – be sure to have run the `data_import()` function before running a method (see section C, page 4). **All the included examples refer have been applied to the testing dataset.**

### 1.1 Data grouping functions

To aggregate the data into monthly time series and to generate the 95% confidence intervals, you will need to follow the steps, which are listed below.

**Note** – parameters highlighted in bold are already set to default but can be customized. In case you are happy with these, you do not need to specify them inside the function.

- **Reformat dates and aggregate monthly time series totals for each susceptibility test result** using the `data_prep()` function

**Example:**

```
new_AMRseries <- data_prep(data=AMR_data)
```

- **Calculate binomial confidence intervals and percentages** using the `bin_CI()` function.

**Example:**

```
citable <- bin_CI(data=new_AMRseries,  
                  conf=0.05)
```



- **Percentage tested and percentage resistance plot** with no confidence intervals using the `pcentRandT_plot()` function.

**Example:**

```
ResandTeslinegraph_noCI <- pcentRandT_plot(data=citable,
                                           filter1="genus",
                                           filter2="antimicrobial")
```

- **Percentage tested and percentage resistance plot** with confidence intervals using the `pcentRandTci_plot()` function.

**Example:**

```
ResandTeslinegraph_CI <- pcentRandTci_plot(data=citable,
                                           filter1="genus",
                                           filter2="antimicrobial")
```

- **Stacked susceptibility result incidence plot** using the `stack_plot_function()` function.

**Example:**

```
stackgraph <- stack_plot_function(data=testfile,
                                  filter1="Organism",
                                  filter2="Antimicrobial")
```

**TIP:** If running these outputs using the functions as described in this part, you can define the “genus” and “antimicrobial” filters in the console as follows:

```
filter1 <- "Pseudomonas"
filter2 <- "CTZ"
```

These can then be updated to fit another genus or antimicrobial and the functions re-run.

To help you select, a full list of the antimicrobials and genera within the uploaded dataset can be obtained using the following commands, and then viewed.

```
abxlist <- as.list(x = unique(AMR_data$antimicrobial))
genus_list <- as.list(x = unique(AMR_data$genus))
view(abxlist)
view(genus_list)
```

## 4 AMRapp interactive dashboard

### 1 Initial Set-up

This section describes the extended functionality of the AMRapp, which allows the user to perform interactive AMR trend using a user-friendly R Shiny application.

#### 1.1. Installing Required Libraries

Before running interactive version of AMRapp, the user is required to manually install and load the required libraries on RStudio. Ensure that R Studio is open before progressing to the next step.

In the R Studio console, if not already done so, set the working directory to the location of the downloaded AMRapp using the following commands:

```
setwd("~/Users/...yourPath.../AMRapp")
```

#### 1.2. Running AMRapp R Shiny Application

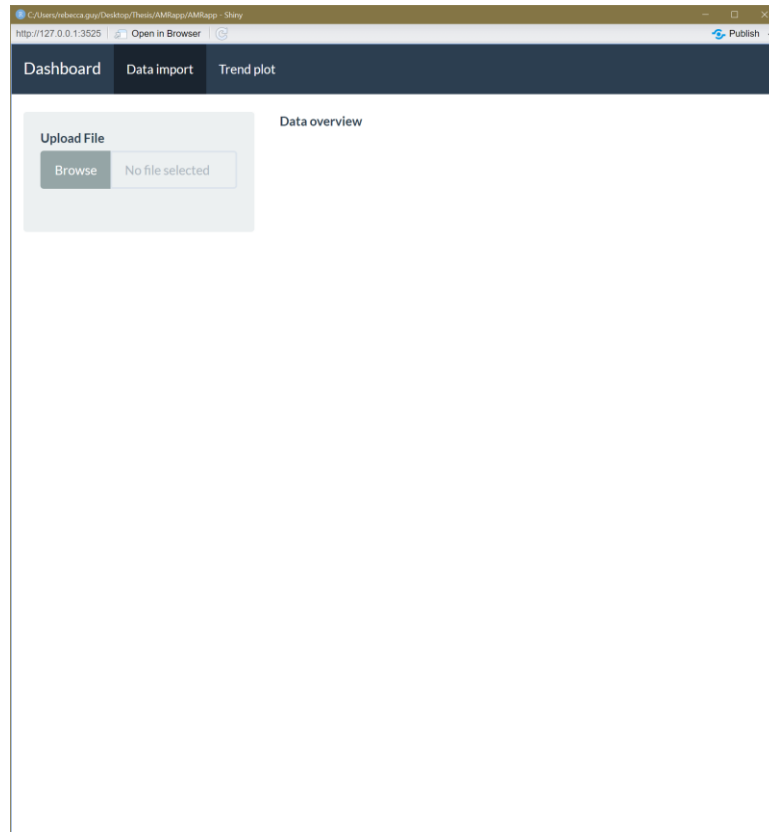
After the AMRapp package has been installed you will be able to run the interactive AMRapp tool by calling the following functions in the R Studio console.

```
library("AMRapp")
```

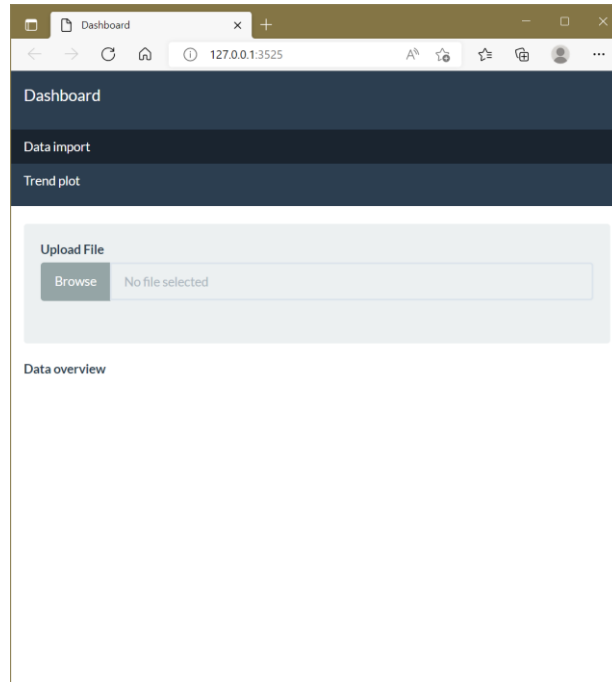
```
source("R/AMRapp.r", echo = TRUE)
```

After a few seconds, the following window will pop up on your screen: **(Figure 5a)**

**Note.** The dashboard can be opened in a browser by clicking on the 'Open in Browser' tag at the top of the window. The dashboard content will also adjust to the size of the window **(Figure 5b)**.



**Figure 5a.** Screenshot of AMRapp R Shiny application. The Data import screen.



**Figure 5b.** Screenshot of AMRapp R Shiny application opened in a browser (Microsoft Edge). Content has rearranged to factor in the small window size.

## 2 AMRapp R Shiny application

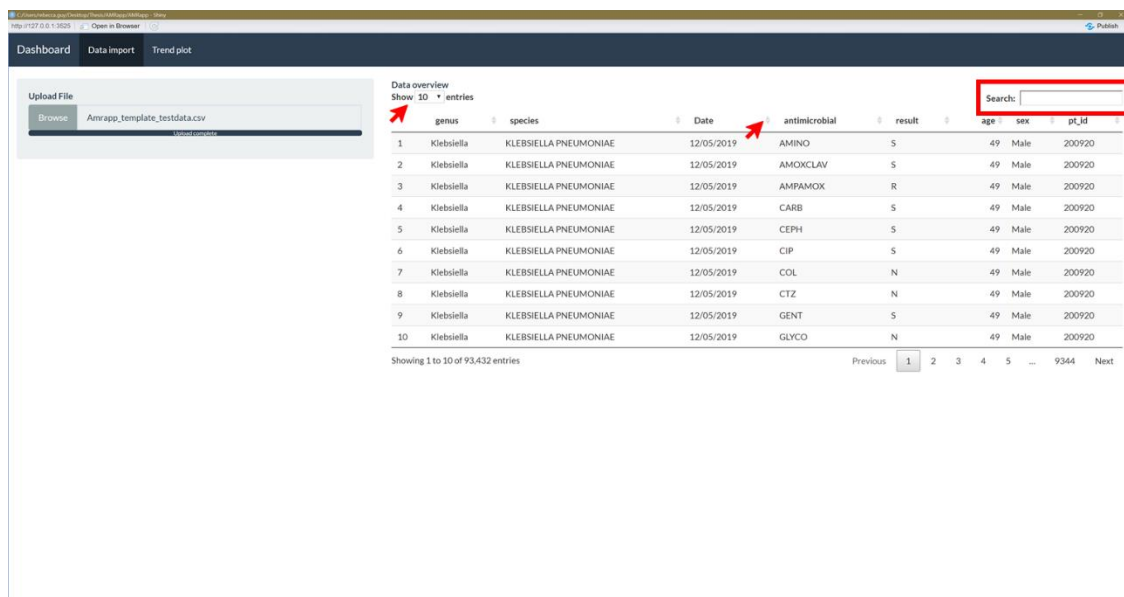
This section describes all the functionalities included in the AMRapp R-Shiny dashboard.

### 2.1. Data import panel

The dashboard opens on the Data import panel (**Figure 5**).

To upload a file (in the preformatted template; or the example dataset), click on the Browse button, in a pop out window navigate to the location of the file chosen to upload and click OK.

Once the file is uploaded the blue bar in the 'Upload file' panel will indicate it has completed, and a summary table of the data will appear on the right (**Figure 6**).



The screenshot shows the AMRapp R Shiny application dashboard. The 'Data import' panel is active, displaying an 'Upload File' section with a 'Browse' button and a file named 'Amrapp\_template\_testdata.csv'. To the right, the 'Data overview' section shows a table of 10 entries. The table has columns for 'genus', 'species', 'Date', 'antimicrobial', 'result', 'age', 'sex', and 'pt\_id'. A search bar is located at the top right of the table. Red arrows highlight the 'Browse' button and the 'antimicrobial' column header.

	genus	species	Date	antimicrobial	result	age	sex	pt_id
1	Klebsiella	KLEBSIELLA PNEUMONIAE	12/05/2019	AMINO	S	49	Male	200920
2	Klebsiella	KLEBSIELLA PNEUMONIAE	12/05/2019	AMOXCLAV	S	49	Male	200920
3	Klebsiella	KLEBSIELLA PNEUMONIAE	12/05/2019	AMPAMOX	R	49	Male	200920
4	Klebsiella	KLEBSIELLA PNEUMONIAE	12/05/2019	CARB	S	49	Male	200920
5	Klebsiella	KLEBSIELLA PNEUMONIAE	12/05/2019	CEPH	S	49	Male	200920
6	Klebsiella	KLEBSIELLA PNEUMONIAE	12/05/2019	CIP	S	49	Male	200920
7	Klebsiella	KLEBSIELLA PNEUMONIAE	12/05/2019	COL	N	49	Male	200920
8	Klebsiella	KLEBSIELLA PNEUMONIAE	12/05/2019	CTZ	N	49	Male	200920
9	Klebsiella	KLEBSIELLA PNEUMONIAE	12/05/2019	GENT	S	49	Male	200920
10	Klebsiella	KLEBSIELLA PNEUMONIAE	12/05/2019	GLYCO	N	49	Male	200920

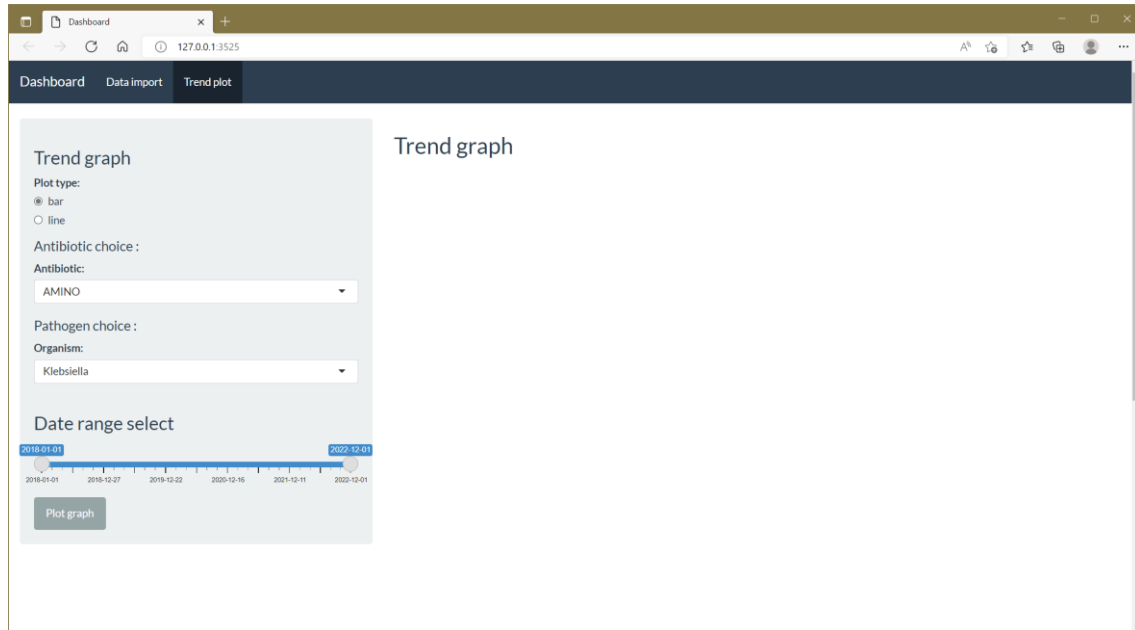
Showing 1 to 10 of 93,432 entries

**Figure 6.** Screenshot of uploaded data. Showing selected file and table of the data.

The data in the table can be searched, filtered and expanded using the inbuilt toggles (highlighted in red in **Figure 6**).

## 2.2. Trend plot panel

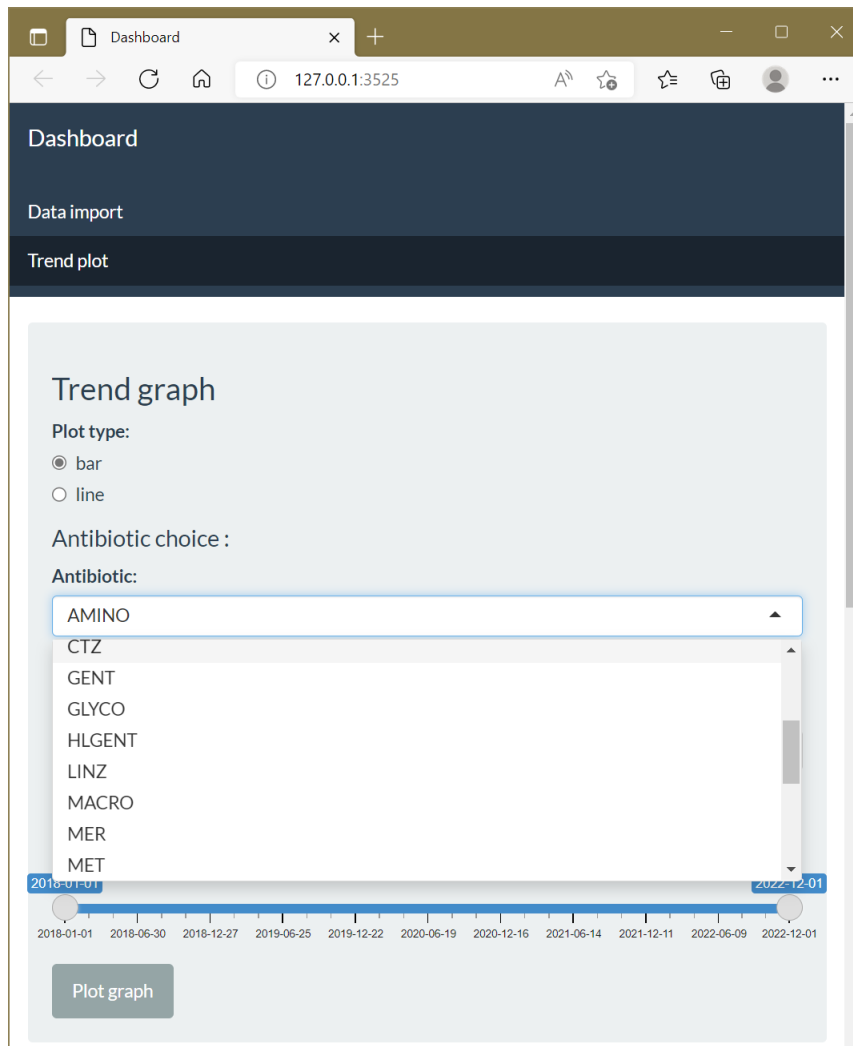
Once the data is imported, navigate to the Trend plot panel by clicking on 'Trend plot' in the navy banner at the top of the window. This will open up the menu for selecting the graph, organism, antimicrobial and time period (**Figure 7**).



**Figure 7.** Screenshot of the Trend plot panel

The drop-down menu options and date range are generated from the uploaded data, ensuring that only viable options are selectable. However, the system does not check whether the selected organism and antimicrobial are sensible options.

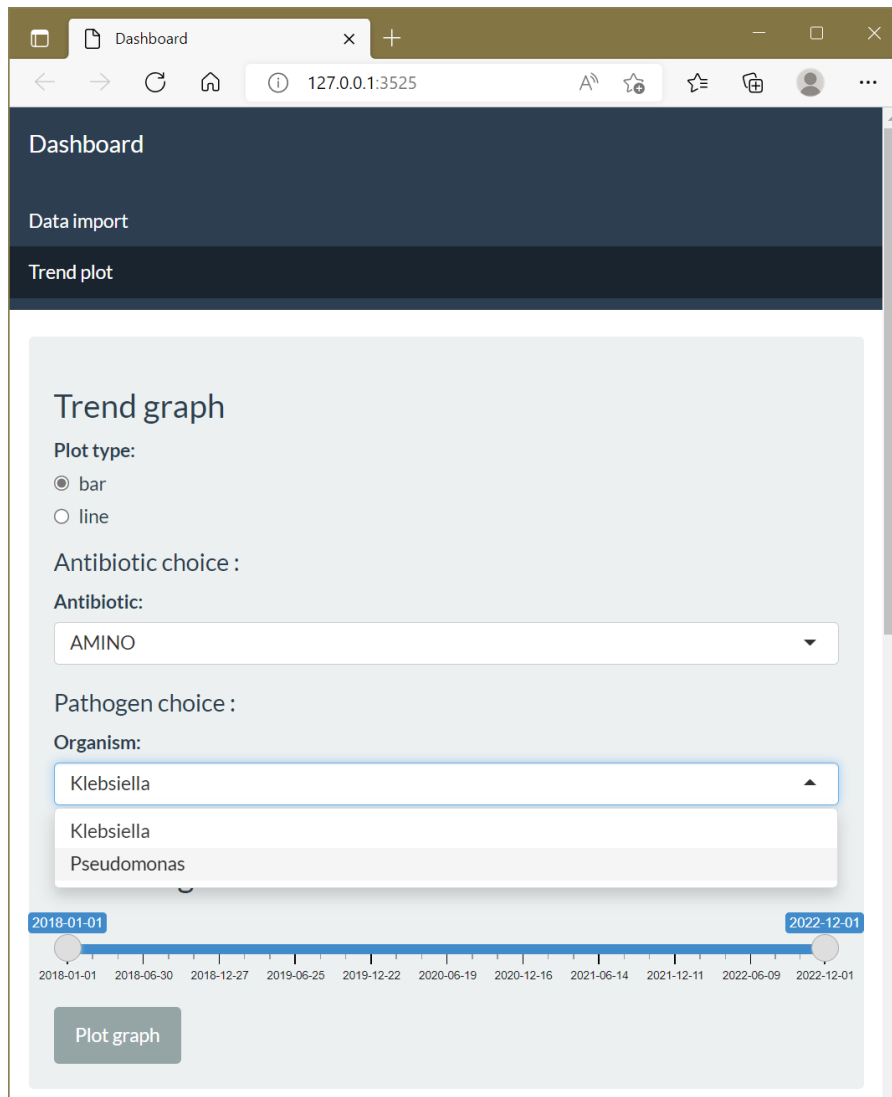
Use the plot type radio button choice to select whether an incidence **bar** plot should be generated or a percentage resistance **line** graph. The drop-down features are described over the next few pages (**Figures 8 to 10**).



**Figure 8.** Screenshot of the 'Antibiotic Choice' selector.

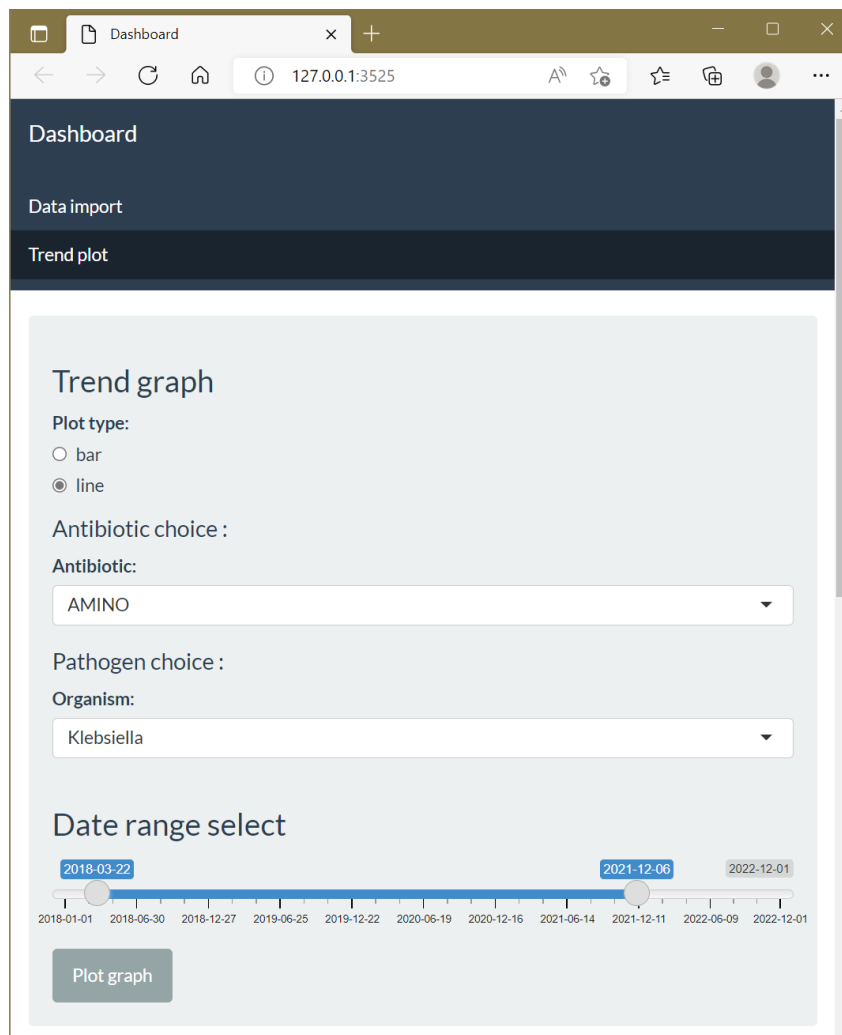
The Antimicrobial choice selector is populated with whatever has been imported into the application in the antimicrobial template field. This can be antimicrobial code (as in the figure) or the full name of the antimicrobial. This will work for antibiotic or antifungal depending on the data uploaded.





**Figure 9.** Screenshot of the 'Pathogen Choice' selector.

The Pathogen choice selector is populated with whatever has been imported into the application in the genus template field. This can be the organism code or the full name. This will work for all organism types but is primarily designed to work with bacterial or fungal pathogens.



**Figure 10.** Screenshot of the AMRapp Trend graph panel with date range selector amended.

To amend the date range ahead of visualisation the date range selector can be refined by dragging the toggle for either end to cover the desired range. The date range is based on the specimen date and operates in monthly intervals.

Note. the date range can also be narrowed once the graph has been plotted by zooming in (or back out) in the graph.

To plot the graph, click on the 'plot graph' button. This will display the graph in the panel to the right (or below if using a smaller window). In addition, a summary table of the plotted data will appear below the graph.

To change the graph just select new options and click on 'plot button' again.

Save options appear above the graphs and at the base of the tables.

## 5 AMRapp visualisations

This section describes the graphs generated by the Shiny application or the AMRapp stepwise functions.

Graphs can all be exported as interactive HTML files or a static image file. Examples of each are included within the examples folder downloaded in Section 2 or online at <https://github.com/RG-abx/AMRapp/tree/main/examples>.

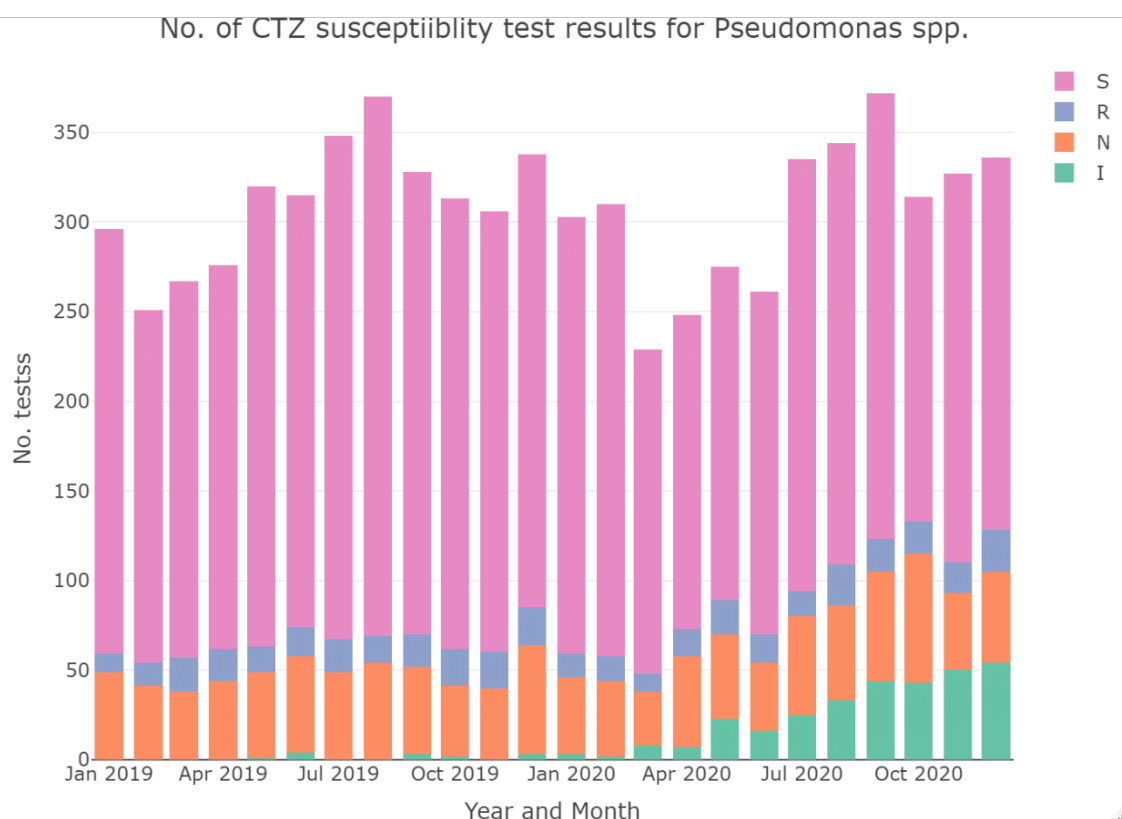
**Note.** In all plots, zoom functionality is enabled by clicking in the plot, dragging and selecting the region of interest.

**Note.** All plots have hover-over summary data displayed for each time point.

### 5.1. Stacked incidence plots (the 'bar' plot type)

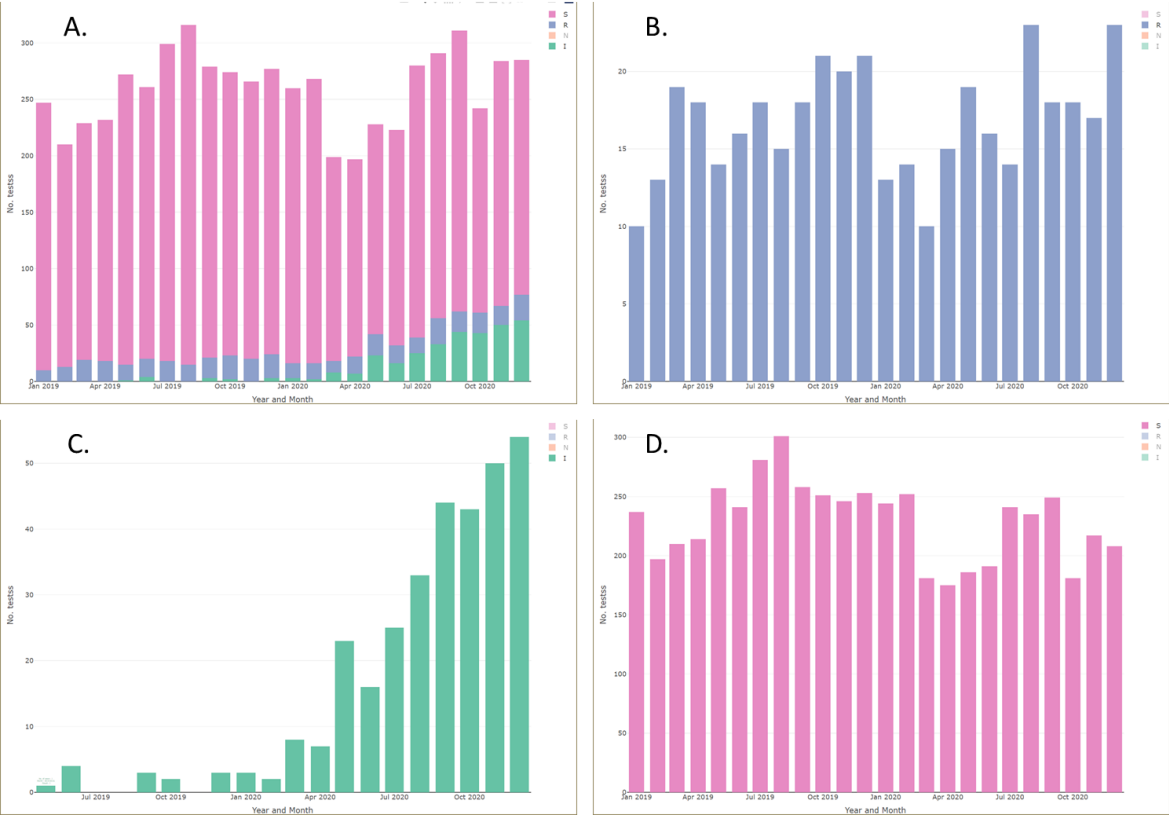
The incidence plot is created by using the `stack_plot_function()` or by selecting the 'bar' in the plot type on the AMRapp dashboard (**Figure 11**).

Plot includes the title with the antimicrobial and the organism selected.



**Figure 11.** Screenshot of the AMRapp stacked bar plot

To filter the graph to display each result category (S, susceptible; I, intermediate; R, resistant; N, Not tested) click on the relevant result code in the legend in the top right corner (**Figure 12**).

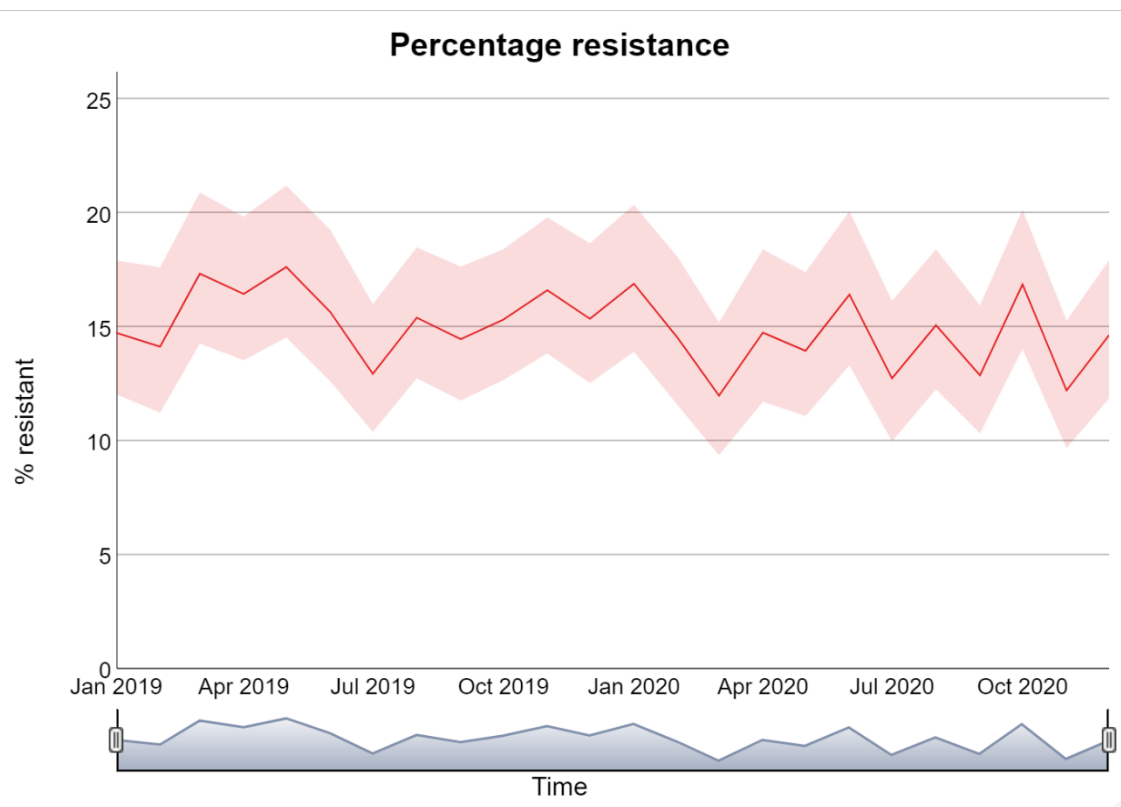


**Figure 12.** Screenshot of the filtered bar plot by result type.

## 5.2. Percentage resistant with confidence intervals (the 'line' plot type)

The percentage resistant plot is created by using the `pcentRci_plot()` or by selecting the 'line' in the plot type on the AMRapp dashboard (**Figure 13**).

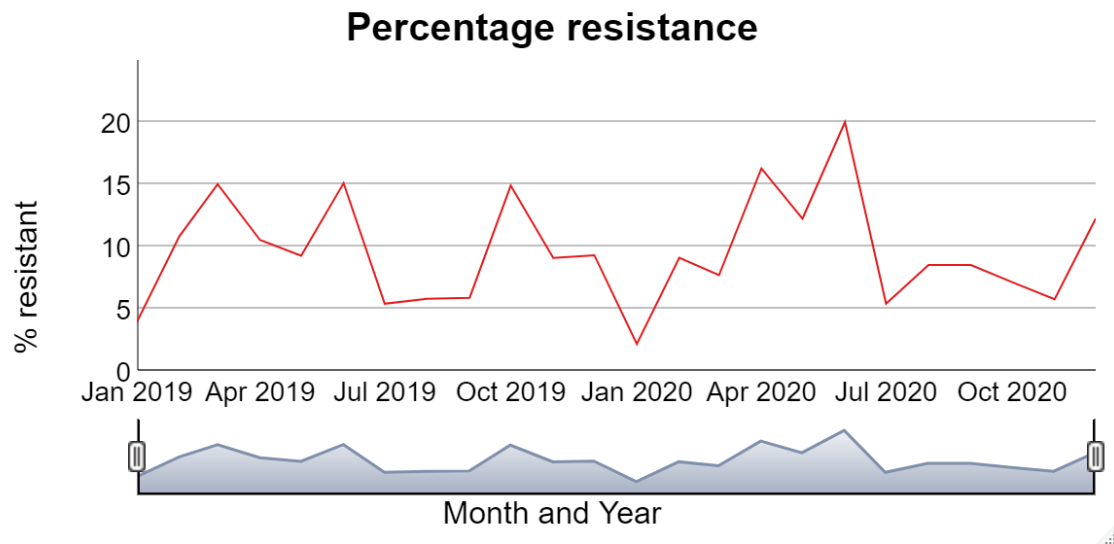
A time widget is displayed at the bottom of the graph as an additional method to view the graph for particular time periods. Move the counter from either end to narrow the field.



**Figure 13.** Percentage resistance line plot, with confidence intervals

### 5.3. Percentage resistant line plot without confidence intervals

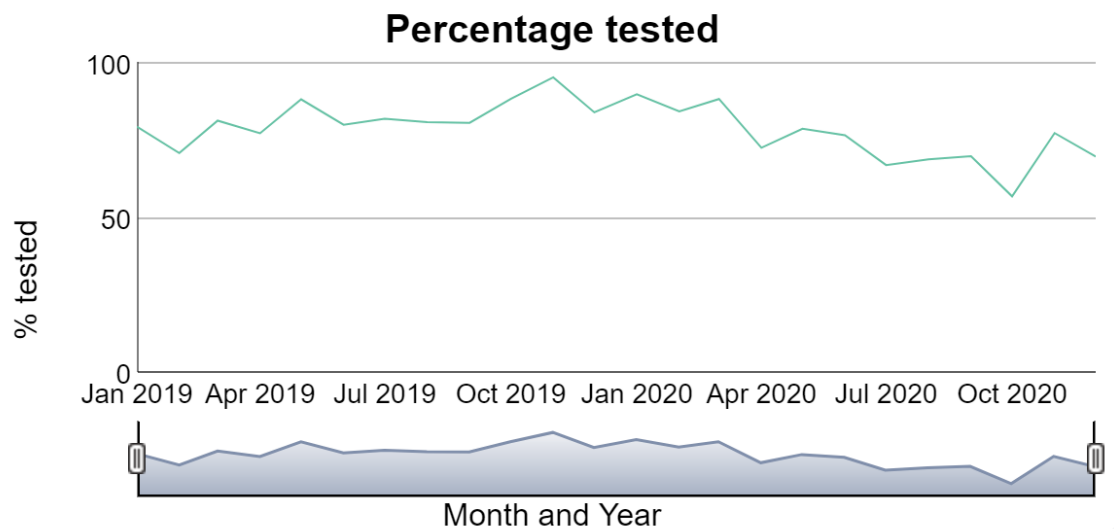
The percentage resistant plot is created by using the `pcentR_plot()` (Figure 14).



**Figure 14.** Percentage resistance line plot, without confidence intervals

### 5.4. Percentage tested line plot without confidence intervals

The percentage resistant plot is created by using the `pcentT_plot()` (Figure 15).



**Figure 15.** Percentage tested line plot, without confidence intervals

## 5.5. Percentage tested line plot with confidence intervals

The percentage resistant plot is created by using the `pcentT_ci_plot()` (Figure 16).

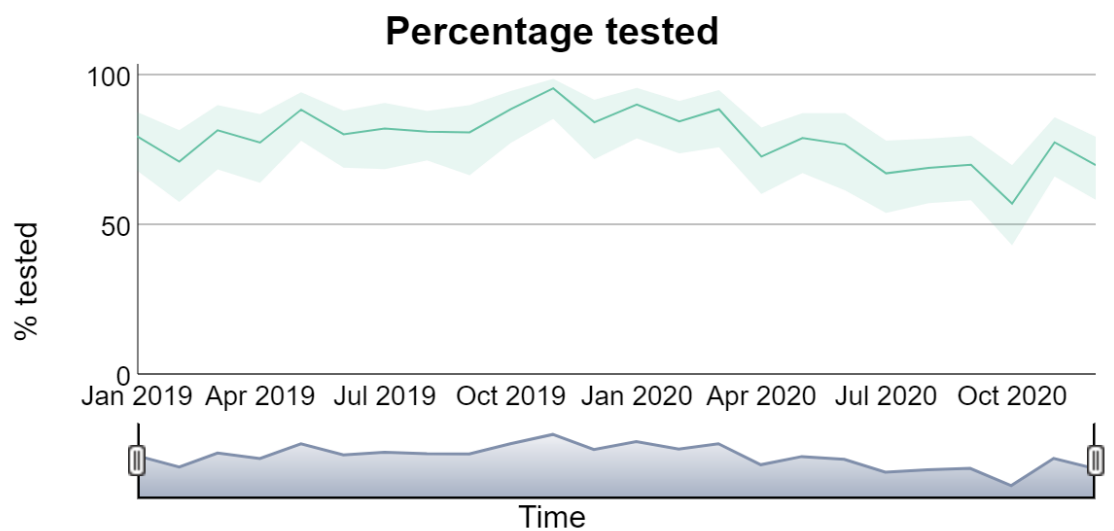


Figure 16. Percentage tested line plot, with confidence intervals

## 5.6. Percentage tested and percentage resistant line plot without confidence intervals

The percentage resistant plot is created by using the `pcentRandT_plot()` (Figure 17).

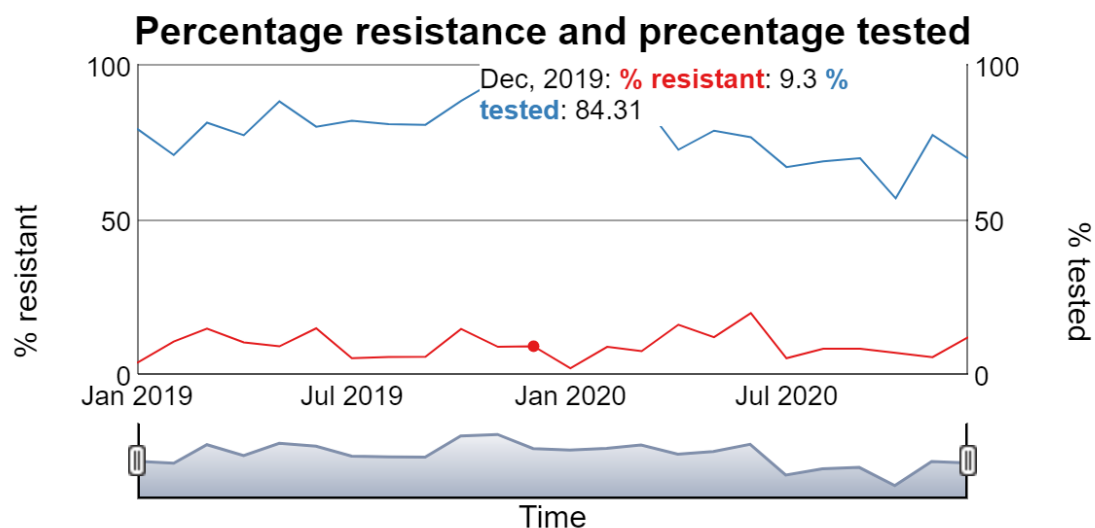
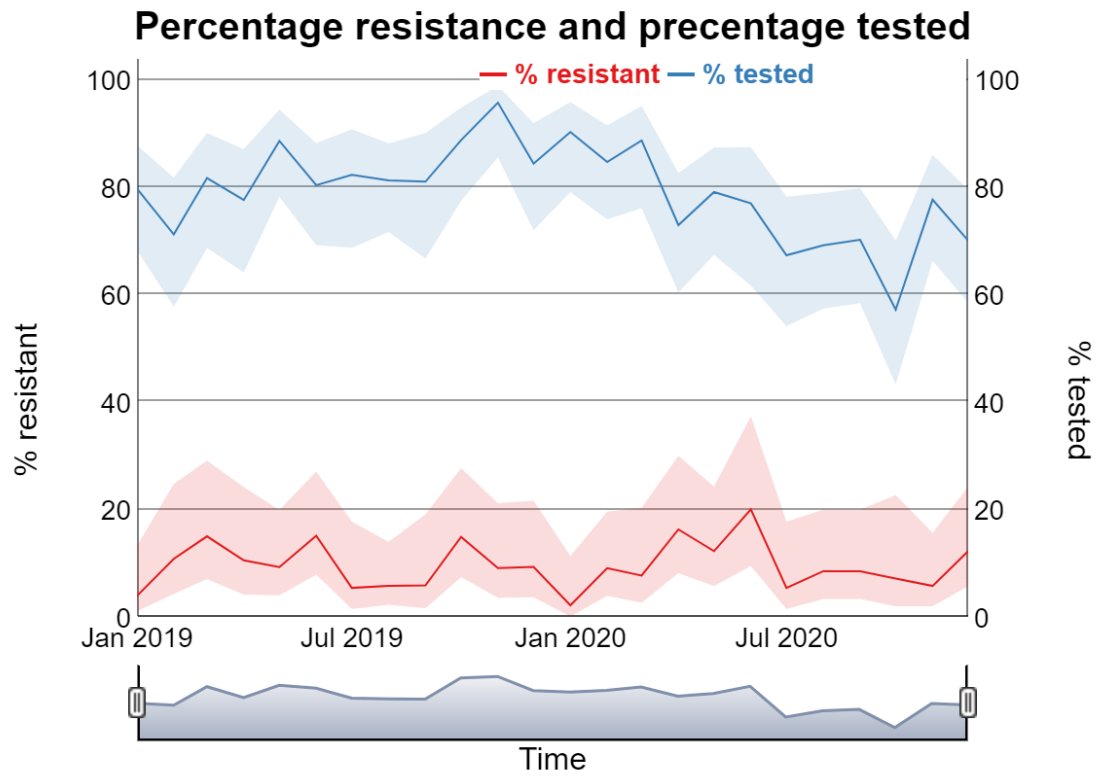


Figure 17. Percentage tested and percentage resistant line plot, without confidence intervals. Hover-over information displayed.

## 5.7. Percentage tested and percentage resistant line plot with confidence intervals

The percentage resistant plot is created by using the `pcentRandTci_plot()` (Figure 18).



**Figure 18.** Percentage tested and percentage resistant line plot, with confidence intervals.