

Troubleshooting R:

Figuring out what to do when you aren't sure what to do

June 10, 2021

Compared to other data analysis tools, R and RStudio are more powerful and precise – it's a blessing and a curse



Once you encounter a problem in R or RStudio, the path forward isn't always obvious

Sometimes the best way to learn how things work is to make them not work.

Let's break some code, see what happens, and try to fix it!



```
# code from earlier that works
library(readxl)

raw_stpov19 <- read_excel("data/usd19.xls",
                          sheet = "USSD19", skip = 2)

# what happens if we didn't know about "skip"???
icky_stpov19 <- read_excel("data/usd19.xls",
                           sheet = "USSD19")

head(icky_stpov19)
```

What went wrong?

```
> icky_stpov19 <- read_excel("data/usd19.xls",
+                             sheet = "USSD19")
New names:
* `` -> ...2
* `` -> ...3
* `` -> ...4
* `` -> ...5
* `` -> ...6
* ...
>
> head(icky_stpov19)
# A tibble: 6 x 7
  `Table with column head...` ...2 ...3 ...4 ...5 ...6 ...7
  <chr> <chr> <chr> <chr> <chr> <chr> <chr>
1 2019 Poverty Estimates ... NA NA NA NA NA NA
2 Postal FIPS distr... Name Estima... Estim... Estim...
3 AL 01 00190 Alabaster... 34669 6710 649
4 AL 01 00005 Albertvil... 22051 4131 957
5 AL 01 00030 Alexander... 16920 2575 703
6 AL 01 00060 Andalusia... 8842 1465 406
```

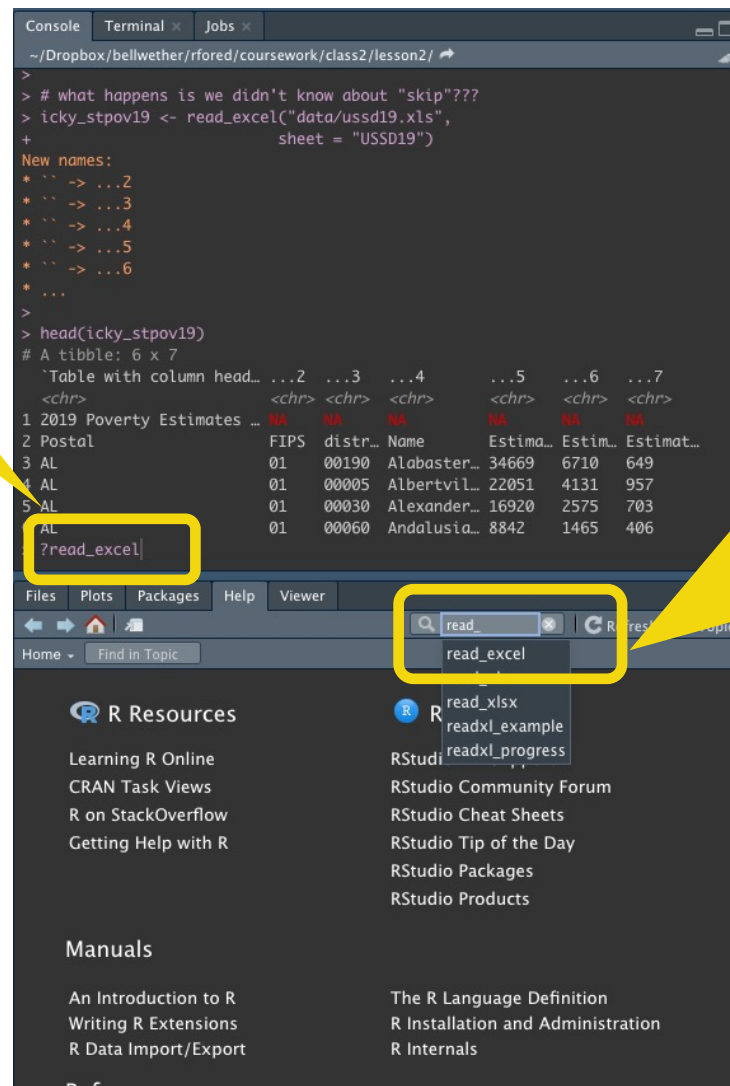
Five ways to solve an R-related issue



1. Help documentation in RStudio
2. Google
3. StackOverflow
4. #rstats Twitter
5. Ask a friend

RStudio provides several ways to access documentation related to any function or package

From the console:
Type a `?` and the function name as one word, then hit return.



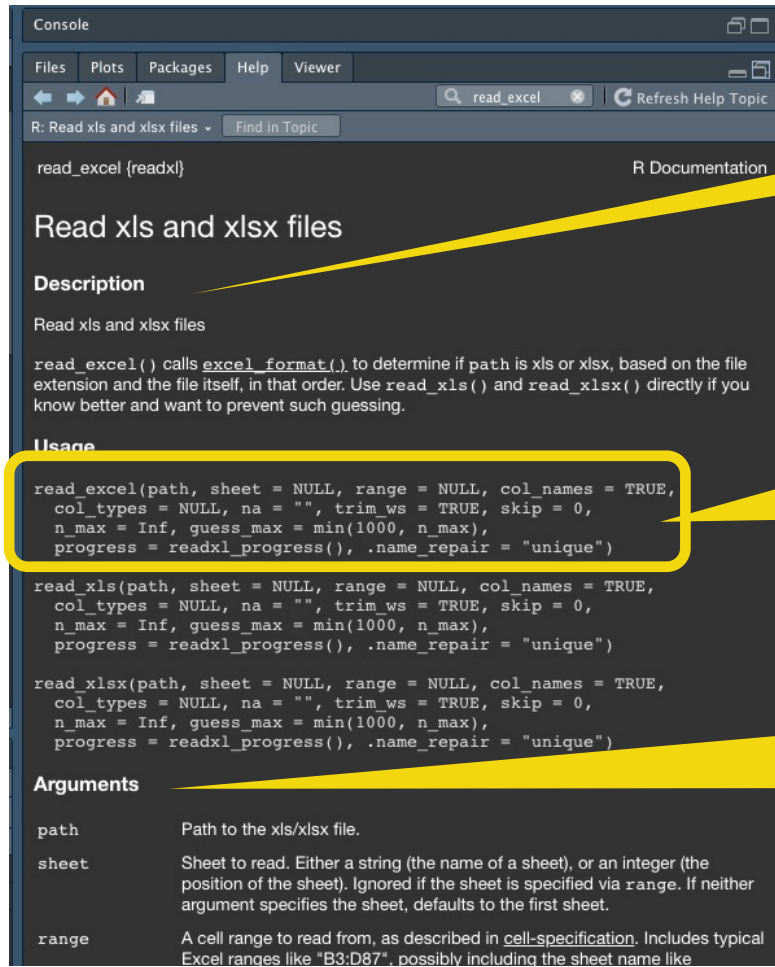
The screenshot shows the RStudio interface with the console and help tabs active. In the console, the following code is entered:

```
> # what happens is we didn't know about "skip"???  
> icky_stpov19 <- read_excel("data/usssd19.xls",  
+                             sheet = "USSD19")  
New names:  
* "" -> ...2  
* "" -> ...3  
* "" -> ...4  
* "" -> ...5  
* "" -> ...6  
* ...  
>  
> head(icky_stpov19)  
# A tibble: 6 x 7  
  `chr`    ...2    ...3    ...4    ...5    ...6    ...7  
  <chr>    <chr>    <chr>    <chr>    <chr>    <chr>    <chr>  
1 2019 Poverty Estimates ... NA    NA    NA    NA    NA    NA  
2 Postal FIPS distr... Name Estima... Estim... Estim...  
3 AL      01      00190 Alabaster... 34669    6710    649  
4 AL      01      00005 Albertvil... 22051    4131    957  
5 AL      01      00030 Alexander... 16920    2575    703  
6 AL      01      00060 Andalusia... 8842     1465    406
```

In the console, the text `?read_excel` is entered and highlighted with a yellow box. In the help tab, the search bar contains `read_excel` and is also highlighted with a yellow box. The search results show a list of suggestions including `read_excel`, `read_xlsx`, `readxl_example`, and `readxl_progress`.

From the “Help” tab:
Start typing the function name in this field and RStudio will start to offer suggestions – either type it all out and hit return, or select the function from the suggestions and hit return or click on it.

The help documentation in R can be very helpful once you understand what it's trying to show you



The screenshot shows the R help documentation for the `read_excel()` function. The interface includes a top navigation bar with tabs for Files, Plots, Packages, Help, and Viewer. The Help tab is active, showing the search results for `read_excel`. The main content area displays the function name, a brief description, and the full usage syntax. Two yellow callout boxes are present: one pointing to the 'Usage' section and another pointing to the 'Arguments' section.

Console

Files Plots Packages Help Viewer

R: Read xls and xlsx files Find in Topic

`read_excel {readxl}` R Documentation

Read xls and xlsx files

Description

Read xls and xlsx files

`read_excel()` calls `excel_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(), .name_repair = "unique")

read_xls(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")

read_xlsx(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")
```

Arguments

<code>path</code>	Path to the xls/xlsx file.
<code>sheet</code>	Sheet to read. Either a string (the name of a sheet), or an integer (the position of the sheet). Ignored if the sheet is specified via <code>range</code> . If neither argument specifies the sheet, defaults to the first sheet.
<code>range</code>	A cell range to read from, as described in cell-specification . Includes typical Excel ranges like <code>"B3:D87"</code> , possibly including the sheet name like

Description:

What is this function supposed to do?

Usage:

What is R expecting as inputs – also called “Arguments” in this function?
What default values will the function use?

Arguments:

What inputs can the function accept?
What options can you tinker with?

When reading R documentation, always scroll down to the “Examples” section – it includes code you can run

Arguments, continued:

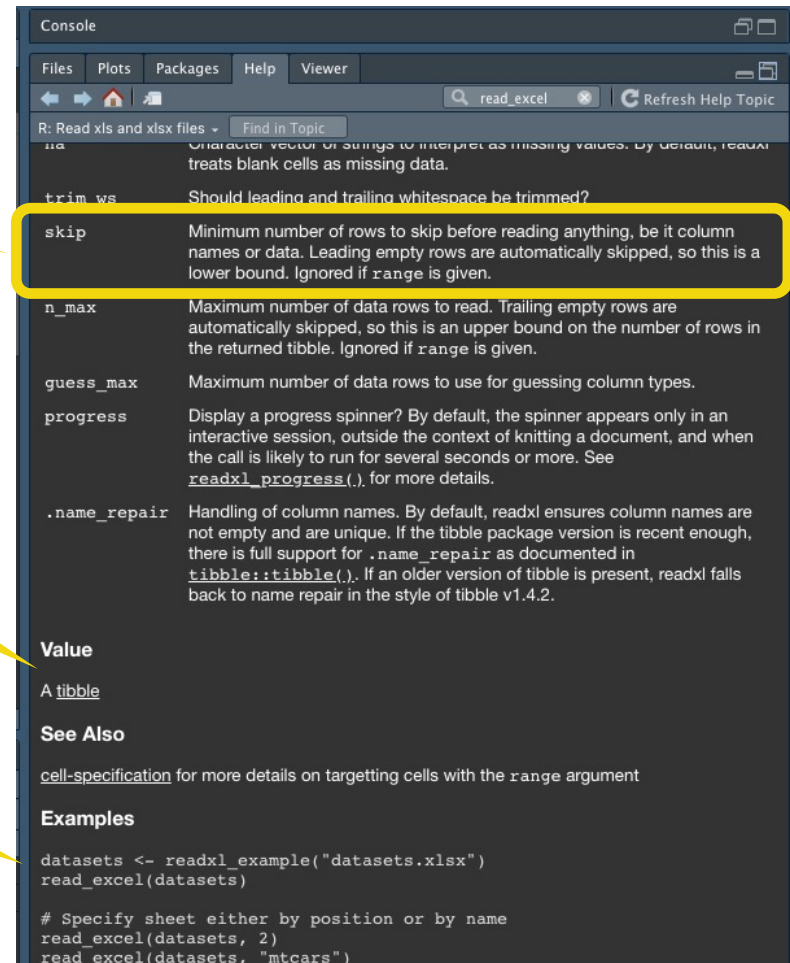
What does the ``skip`` value do in the function ``read_excel()``?

Value:

After you run this function, what will R return to you?

Examples:

Ok, all that info above is great, but how do I *actually* use this function?



The screenshot shows the R documentation for the `read_excel()` function. Three yellow callout boxes highlight specific parts of the documentation:

- The first callout points to the `skip` argument description: "Minimum number of rows to skip before reading anything, be it column names or data. Leading empty rows are automatically skipped, so this is a lower bound. Ignored if `range` is given."
- The second callout points to the `Value` section: "A `tibble`".
- The third callout points to the `Examples` section, which contains the following code:

```
datasets <- readxl_example("datasets.xlsx")
read_excel(datasets)

# Specify sheet either by position or by name
read_excel(datasets, 2)
read_excel(datasets, "mtcars")
```


CHALLENGE

1. Find the help documentation for ``summarise`` in RStudio
2. Navigate to the “Examples” section
3. Copy the first two commands, paste them in your console, and hit return

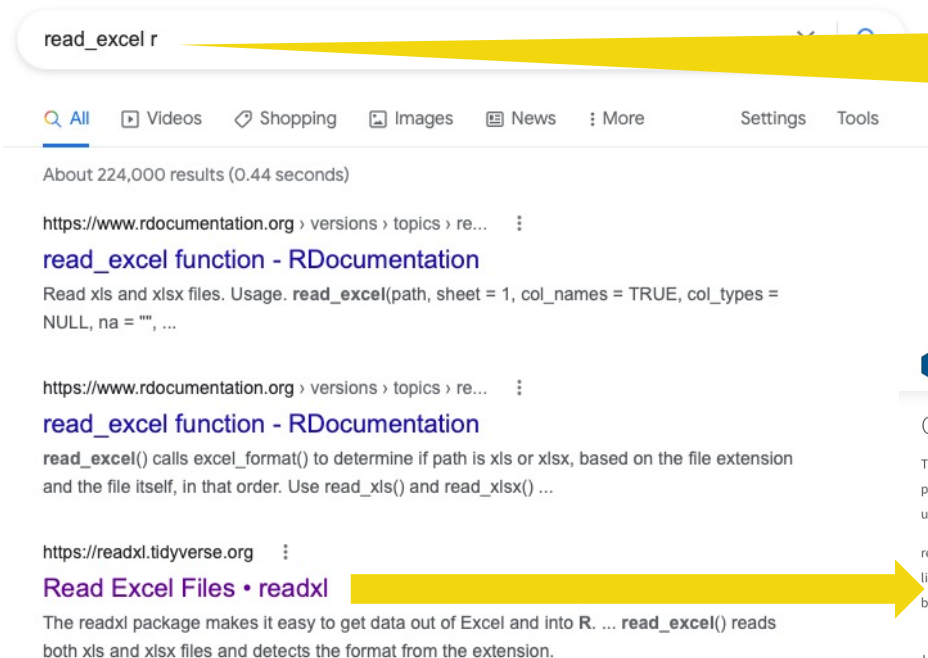


SOLUTION

```
> # A summary applied to ungrouped tbl returns a single row
> mtcars %>%
+   summarise(mean = mean(displacement), n = n())
  mean    n
1 230.7219 32
>
> # Usually, you'll want to group first
> mtcars %>%
+   group_by(cyl) %>%
+   summarise(mean = mean(displacement), n = n())
# A tibble: 3 x 3
   cyl  mean    n
  <dbl> <dbl> <int>
1     4  105.     11
2     6  183.      7
3     8  353.     14
```

Sometimes, you'll have to go beyond R documentation to solve your problems – start with Google!

Typing in a function name or error message and the letter “r” can yield really helpful results



Protip: All of the `tidyverse` packages have really nice documentation provided by RStudio – it's more user-friendly than the basic R documentation.

A screenshot of the readxl package documentation page. The page has a header with the readxl logo, the text "part of the tidyverse", and the version "1.3.1.9000". There are navigation links for "Reference", "Articles", "News", and a GitHub icon. The main content is divided into two columns. The left column has sections for "Overview" and "Installation". The "Overview" section describes the package's purpose and its dependencies. The "Installation" section provides instructions on how to install the package from CRAN. The right column has a "Links" section with various links to resources related to the package, including CRAN, source code, bug reports, and the license.

readxl part of the tidyverse 1.3.1.9000

Reference Articles News GitHub

Overview

The readxl package makes it easy to get data out of Excel and into R. Compared to many of the existing packages (e.g. gdata, xlsx, xlsReadWrite) readxl has no external dependencies, so it's easy to install and use on all operating systems. It is designed to work with *tabular* data.

readxl supports both the legacy .xls format and the modern xml-based .xlsx format. The `libxls` C library is used to support .xls, which abstracts away many of the complexities of the underlying binary format. To parse .xlsx, we use the `RapidXML` C++ library.

Installation

The easiest way to install the latest released version from CRAN is to install the whole tidyverse.

```
install.packages("tidyverse")
```

NOTE: you will still need to load readxl explicitly, because it is not a core tidyverse package loaded via `library(tidyverse)`.

Alternatively, install just readxl from CRAN:

```
install.packages("readxl")
```

Links

- Download from CRAN at <https://cloud.r-project.org/package=readxl>
- Browse source code at <https://github.com/tidyverse/readxl/>
- Report a bug at <https://github.com/tidyverse/readxl/issues>
- Learn more at <http://r4ds.had.co.nz/data-import.html>
- License [Full license](#)
- GPL-3
- Community [Contributing guide](#)

Stack Overflow is a Q&A site for programmers – if you have a specific question related to R, this is a great resource

- If you're running into an error message or have a specific problem with a function, there's a good chance someone else has had the same problem and asked a question about it on Stack Overflow.
- Many of your R-related Google searches may direct you to a Stack Overflow page.
- If you're really stuck, submit a question – people on this site love to solve problems.

The screenshot shows the Stack Overflow homepage with a search bar at the top containing the text 'read_excel r'. Below the search bar, there's a navigation menu with links for 'Home', 'PUBLIC', 'Questions', 'Tags', 'Users', 'FIND A JOB', 'Jobs', 'Companies', and 'TEAMS'. The 'Questions' section is highlighted. Below this, there's a section for 'Stack Overflow for Teams' with a 'Create a free Team' button. The main content area displays 'Search Results' for 'read_excel r', showing 374 results. Three results are visible, each with a question title, a brief description, and a '1 answer' badge. The first result is 'Q: R read_excel readxl sometimes incorrectly converts numbers to dates', the second is 'Q: Open cell range in read_excel R', and the third is 'Q: R read_excel: libxls error: Unable to parse file'.

stackoverflow About Products For Teams

People who code: we want your input. [Take the Survey](#)

Home
PUBLIC
Questions
Tags
Users
FIND A JOB
Jobs
Companies
TEAMS

Stack Overflow for Teams – Collaborate and share knowledge with a private group.

[Create a free Team](#)
What is Teams?

Search Results [Advanced Search Tips](#) [Ask Question](#)

Results for read_excel r

374 results Relevance Newest More ▾

0 votes
1 answer
Q: R read_excel readxl sometimes incorrectly converts numbers to dates
When I use read_excel to import data from Excel to R, some numeric columns are automatically converted to dates. # e.g. 5600 to 1915-05-01 Is there a way to turn of this feature? ... Other than using "col_types" argument in read_excel. ...
r excel readxl
asked Apr 13 '18 by [LeGeniusII](#)

1 vote
1 answer
Q: Open cell range in read_excel R
Is there a way to not specify a cell range in the read_excel function in R? ... Something like: A <- read_excel("excell/book.xlsx", sheet="A", range="C3:") ...
r excel
asked Jan 27 '19 by [nov](#)

0 votes
1 answer
Q: R read_excel: libxls error: Unable to parse file
I'm trying to read xls file into R using readxl::read_excel, but it's giving the following errors: Error: filepath: //... .. My current alternative is to convert this file to "xlsx" using Excel and then read in with readxl::read_excel, but I would like to import the "xls" directly. How to fix it? ...
r import xls
asked Mar 10 '20 by [LeGeniusII](#)

#rstats Twitter is a very nice corner of social media – don't be afraid to ask questions with the #rstats hashtag



Felipe PJ
@fjvost

looking for some help with [#ggplot2](#) in [#rstats](#)! I have this dataframe/tibble with an ordered factor but when I pull it into ggplot it automatically changes the x axis into alphabetical order. 🤔 I've tried everything!!! Any suggestions?

3:02 PM · Jun 8, 2021 · TweetDeck

4 Retweets 8 Likes



Samantha L Wilson, PhD @SamanthaLWilson · 23h

Replying to @fjvost and @rstatstweet

This might be helpful -



How do you specifically order ggplot2 x axis inst...
I'm trying to make a heatmap using ggplot2 using the geom_tiles function here is my code below: ...
[stackoverflow.com](#)

1 1 1



Felipe PJ @fjvost · 22h

this was exactly it!! Thank you. Will now keep a closer eye to the actual levels of factors :P



Dayani C
@forestjackfruit

UseRs please help! Does plotly conflict with dplyr? I'm getting real tired of typing dplyr:: before each function because rstudio first suggests the plotly options for summarise, filter etc. Is there a way to get it to suggest the dplyr functions first?

[#RStats](#) [#tidyverse](#) [#dplyr](#)

10:48 AM · Jun 7, 2021 · Twitter Web App

3 Retweets 4 Likes



Lluís Revilla Sancho @Lluis_Revilla · Jun 7

Replying to @forestjackfruit

Change the order of the library calls to library("plotly")
library("dplyr")
Last package loaded gets precedence.

1 3



Dayani C @forestjackfruit · Jun 7

Oh wow. I had tried loading dplyr a second time and it didn't work, but changing the original order of library calls helped. Thank you!

1 1



Lluís Revilla Sancho @Lluis_Revilla · Jun 7

Loading again on the same R session won't work. You need to restart the R session and load them on the right order.
Glad it helped.

Try not to spend *too* much time on one problem – you have friends who are more than happy to help you troubleshoot



To summarise, next time you encounter an issue in R...



Five ways to troubleshoot problems in R:

1. Help documentation in RStudio
2. Google
3. StackOverflow
4. #rstats Twitter
5. Ask a friend