CHAPTER 9

Connecting R to GoogleSheets

OBJECTIVE

The primary objective of this chapter is to show you how to create a Google Sheets data analysis report from an R script. Your analysis performed in R will become accessible to a wider audience in an efficient manner, if your script can automatically generate the report in Google Sheets. Google Sheets is free, runs through a browser, and is popular among professionals. R will connect to Google Sheets through a package called `GoogleSheets4`, which is discussed extensively in this chapter.

Contents

9.1 The Googlesheets4 Package

The very first thing to do before you can connect R to Google Sheets, is to install the `googlesheets4` package. This can be done from the RStudio console prompt as follows:

```
> install.packages("googlesheets4")
```

In general, this installation will run with no problem. Sometimes it may be necessary to update some old packages before the installation can be completed.

After installing the `googlesheets4` package, load it into the R environment so that the functions it offers become available for use. This will be accomplished as follows:

```
> library("googlesheets4")
```
or
```
> library(googlesheets4)
```

In this chapter, I will review the most common tasks that analysts perform on their Google spreadsheet data, and will show you how you can automate them with R. If you need to do something special, then you can always obtain more information from the `googlesheets4` documentation, which is accessible with the link https://googlesheets4.tidyverse.org/.

9.2 Read a Sheet into an R data frame

When you open a Google Sheets file also known as a Google spreadsheet containing one or several sheets (or tabs or pages), it will have a web address at the top that is structured as follows:

```
https://docs.google.com/spreadsheets/d/spreadsheetId/edit#gid=sheetId
```

where `spreadsheetId` is a unique spreadsheet identifier, and `sheetId` the unique sheet identifier. To illustrate the use of the most important functions in the `googlesheets4` package, I will use a Google spreadsheet named "us-population-statistics" located at the following web address:

```
https://docs.google.com/spreadsheets/d/
        18xuD8vY46XBaVErfCqKP7Zu6NWWzV174YWwx8INlFgk/edit
        #gid=1790756571
```

It follows that the `spreadsheetId` of the "us-population-statistics" spreadsheet is `18xuD8vY46XBaVErfCqKP7Zu6NWWzV174YWwx8INlFgk` and its first sheet named "Annual Population" has a `sheetId` of `1790756571`. I have made this spreadsheet available to everyone. For convenience, you can open it on your browser with the shorter link `https://bit.ly/USAdemo2020`.

Reading your Google spreadsheet data into an R data frame is straightforward. Consider the following line of code:

```
uspop.df<-read_sheet("18xuD8vY46XBaVErfCqKP7Zu6NWWzV174YWwx8INlFgk")
```

This command will read all data in the first sheet of the Google spreadsheet, whose Id was supplied to function `read_sheet()`. There are a few comments I like to make about this function:

- `read_sheet()` and `range_read()` are 2 functions that can be used interchangeably, as they play the exact same role. Both are available from the `googlesheets4` package, which must be installed and loaded before these functions can be used.

- Remember that all Google spreadsheets are tied to a specific gmail account with which a gmail email address is associated. Consequently, the first time you attempt to use the `read_sheet()` and `range_read()` functions, you will be asked a few questions to authorize the use of your gmail account and register your email address.

  After registering your email address, your user credentials will be stored locally in the form of something called a *token*. Next time `googlesheets4` requires authentication, you may be prompted to select which email address you want to use so that the associated token can be used to au-

thenticate you. For example, if authentication is required for the first time in an R session while reading your Google spreadsheet, you will be prompted to select the email address you want to use as shown in Figure 9.1. If you want to register another gmail account, you must enter 0 at the selection prompt, which initiates a new account registration process.

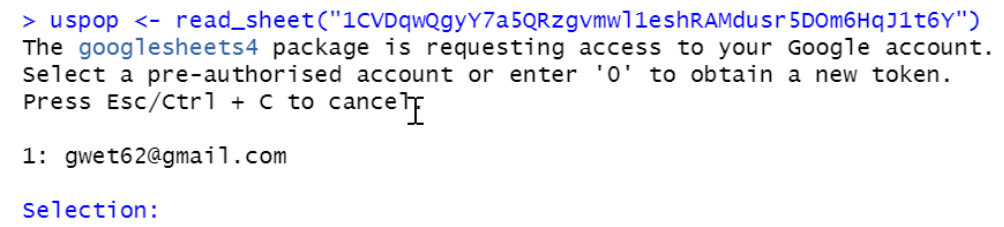


Figure 9.1: `googlesheets4` requesting access to Google account

- The spreadsheet Id is a very long string of characters as you can see. Therefore, you are expected to copy and paste it to wherever it is needed, rather than typing it one character at a time. However, you can also use the file name "us-population-statistics" to retrieve your spreadsheet data. But, you will need another R package named `googledrive`. You must first install and load this package before the `read_sheet()` function can be used. You will then be able to read your data as follows:

```
drive_get(path="us-population-statistics") %>%
     read_sheet() -> uspop.def
```

This is another and more convenient way of reading Google Sheets data into an R data frame named `uspop.def`, using the `drive_get()` function. Note that I can use the pipe symbol (`%>%`) here because `googlesheets4` is a package that is part of `tidyverse`.

> *Unlike the regular PC directory, which cannot contain 2 files with the same name, Google Drive accommodates 2 spreadsheets with the exact same name. However, 2 spreadsheets with the same name will have different spreadsheet Ids.*

Consider the following R command:

`lst.meta <- drive_get(path="us-population-statistics")`.

This command could possibly produce 2 spreadsheet id values that you can display with the command `lst.meta$id`. If `lst.meta$id` lists 2 id values, then you can access them with the 2 commands `lst.meta$id[[1]]` or `lst.meta$id[[2]]`.

For more information on the `read_sheet()` function, visit the webpage https://googlesheets4.tidyverse.org/reference/range_read.html.

9.3 Read a Specific Sheet into a Data Frame

Each time you call the `read_sheet()` function with a spreadsheet Identifier as argument, you will read the entire first sheet into an R data frame. As previously indicated, the "us-population-statistics" spreadsheet contains 3 sheets named *Annual Population*, *Monthly Population*, and *Labor Force*[1]. How can you read the first 4 columns of the *Labor Force* sheet? It is the question I am going to answer in this section.

I assume here that you want to refer to the Google spreadsheet by its name "us-population-statistics" and not by its long address. This requires the use of the `googledrive` package. Here are the 3 options for doing this:

- *Identify Sheets by their Numbers or Names.* The `read_sheet()` function is used here with the 2 arguments `sheet=` and `range=` as shown in the five-line R script shown below. First, `sheet=3` identifies the sheet number of interest, then `range="A:D"` describes the first 4 columns `A` through `D`. The newly-created file is named `lf2021.df`.

[1]Note that these sheet names are case sensitive. If you replace an uppercase character with a lowercase one, R will never locate your sheet.

```
library("googlesheets4")
library("googledrive")
drive_get(path="us-population-statistics") %>%
    read_sheet(sheet=3, range="A:D") -> lf2021.df
print(lf2021.def)
```

Note that sheets can be identified either by their numbers, or by their names. Therefore, `read_sheet(sheet=3, range="A:D")` could be replaced by `read_sheet(sheet="Labor Force", range="A:D")`. Remember that sheet names are case-sensitive.

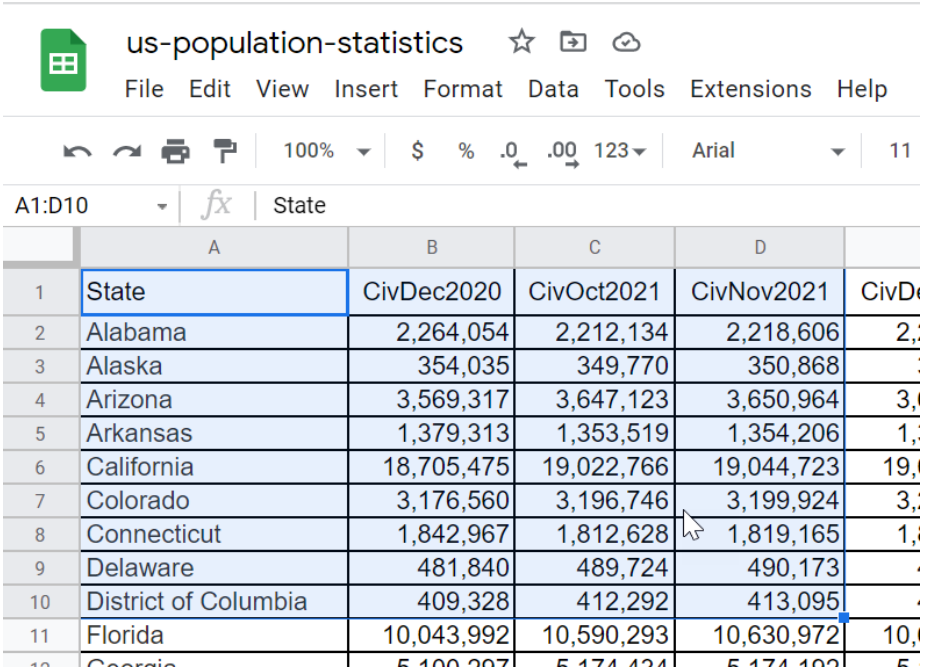
- *Specifying the Sheet Name in the Range Argument.* Using the `sheet=` argument with the `read_sheet()` is optional. In fact, you can specify both the sheet name and range of cells with the `"range="` argument as follows:

```
library("googlesheets4")
library("googledrive")
drive_get(path="us-population-statistics") %>%
    read_sheet(range = "Labor Force!A:D") -> lf2021.df
print(lf2021.def)
```

- *Reading a Narrow Range of Cells.* Instead of reading columns of data in their entirety as in the previous example, you can select a specific range of cells as follows:

```
library("googlesheets4")
library("googledrive")
drive_get(path="us-population-statistics") %>%
  read_sheet(range = "Labor Force!A1:D10") -> lf9states.def
print(lf9states.def)
```

This R script will extract only the highlighted portion of the "Labor Force" sheet shown in Figure 9.2 and defined by the range of values `A1:D10`. The printout of the resulting `lf9states.df` dataset is shown in Table 9.1 .



	A	B	C	D	
1	State	CivDec2020	CivOct2021	CivNov2021	CivD
2	Alabama	2,264,054	2,212,134	2,218,606	2,
3	Alaska	354,035	349,770	350,868	
4	Arizona	3,569,317	3,647,123	3,650,964	3,
5	Arkansas	1,379,313	1,353,519	1,354,206	1,
6	California	18,705,475	19,022,766	19,044,723	19,
7	Colorado	3,176,560	3,196,746	3,199,924	3,
8	Connecticut	1,842,967	1,812,628	1,819,165	1,
9	Delaware	481,840	489,724	490,173	
10	District of Columbia	409,328	412,292	413,095	
11	Florida	10,043,992	10,590,293	10,630,972	10,

Figure 9.2: The `"A1:D10` Range of the "Labor Force" Sheet in the "us-population-statistics" Google Spreadsheet

Table 9.1 : The `lfstates.df` R data frame

```
> print(lf9states.def)
# A tibble: 9 x 4
  State                CivDec2020 CivOct2021 CivNov2021
  <chr>                     <dbl>      <dbl>      <dbl>
1 Alabama                 2264054    2212134    2218606
2 Alaska                   354035     349770     350868
3 Arizona                 3569317    3647123    3650964
4 Arkansas                1379313    1353519    1354206
5 California             18705475   19022766   19044723
6 Colorado                3176560    3196746    3199924
7 Connecticut             1842967    1812628    1819165
8 Delaware                 481840     489724     490173
9 District of Columbia     409328     412292     413095
```