How to use R in Google Colab

And use data from your Drive and BigQuery

Photo by Annie Spratt on Unsplash

Colab, or Colaboratory is an interactive notebook provided by Google (primarily) for writing and running Python through a browser. We can perform data analysis, create models, evaluate these models in Colab. The processing is done on Google-owned servers in the cloud. We only need a browser and a fairly stable internet connection.

Colab is a great alternative tool to facilitate our work, whether as a student, professional, or researcher.

Although Colab is primarily used for coding in Python, apparently we can also use it for R (#Rstats).

This post will tell you how to run R in Google Colab and how to mount Google Drive or access BigQuery in R notebook.

There are two ways to run R in Colab

- The first way is to use the rpy2 package in the Python runtime. This method allows you to execute R and Python syntax together.
- The second way is to *actually* start the notebook in the R runtime.

How to use R and Python together in Colab

1. Open your favorite browser.
2. Create a new notebook: https://colab.research.google.com/#create=true.
3. Run *rmagic* by executing this command `%load_ext rpy2.ipython`.

4. After that, every time you want to use R, add `%%R` in the beginning of each cell.

Start *rmagic* by executing this in a cell:

```
%load_ext rpy2.ipython
```

Use `%%R` to execute **cell magic**. Use this if you want all syntax in a cell to be executed in R. Note that this must be placed at the beginning of the cell.

```
%%R
x <- seq(0, 2*pi, length.out=50)
x
```

These lines will return a variable *x*, and display it on the cell output:

```
 [1] 0.0000000 0.1282283 0.2564565 0.3846848 0.5129131 0.6411414 0.7693696
 [8] 0.8975979 1.0258262 1.1540544 1.2822827 1.4105110 1.5387393 1.6669675
[15] 1.7951958 1.9234241 2.0516523 2.1798806 2.3081089 2.4363372 2.5645654
[22] 2.6927937 2.8210220 2.9492502 3.0774785 3.2057068 3.3339351 3.4621633
[29] 3.5903916 3.7186199 3.8468481 3.9750764 4.1033047 4.2315330 4.3597612
[36] 4.4879895 4.6162178 4.7444460 4.8726743 5.0009026 5.1291309 5.2573591
[43] 5.3855874 5.5138157 5.6420439 5.7702722 5.8985005 6.0267288 6.1549570
[50] 6.2831853
```

Image by Author, x <- seq(0, 2*pi, length.out=50)

Use `%R` to execute **line magic**. Use this if you want a single line in a cell to be executed in R.

Here is how you could use this line magic to copy R variable to Python:

```
x = %R x
```

```
x = %R x
x

array([0.        , 0.12822827, 0.25645654, 0.38468481, 0.51291309,
       0.64114136, 0.76936963, 0.8975979 , 1.02582617, 1.15405444,
       1.28228272, 1.41051099, 1.53873926, 1.66696753, 1.7951958 ,
       1.92342407, 2.05165235, 2.17988062, 2.30810889, 2.43633716,
       2.56456543, 2.6927937 , 2.82102197, 2.94925025, 3.07747852,
       3.20570679, 3.33393506, 3.46216333, 3.5903916 , 3.71861988,
       3.84684815, 3.97507642, 4.10330469, 4.23153296, 4.35976123,
       4.48798951, 4.61621778, 4.74444605, 4.87267432, 5.00090259,
       5.12913086, 5.25735913, 5.38558741, 5.51381568, 5.64204395,
       5.77027222, 5.89850049, 6.02672876, 6.15495704, 6.28318531])
```

Image by Author, Transfer variable between R and Python by using line magic and display x in Python

How to use R in Colab

To use the notebook directly with R:

1. Open your favorite browser.
2. Go to this URL: https://colab.research.google.com/#create=true&language=r, or this short URL https://colab.to/r

After accessing the URL, you will be taken to a new Colab notebook with the default title *Unitled.ipynb*.

At first glance, there is no difference between notebooks with Python and R runtimes. However, if we go to the "Runtime" settings, and select "Change runtime type", we will get a dialog confirming that we are already in R runtime.

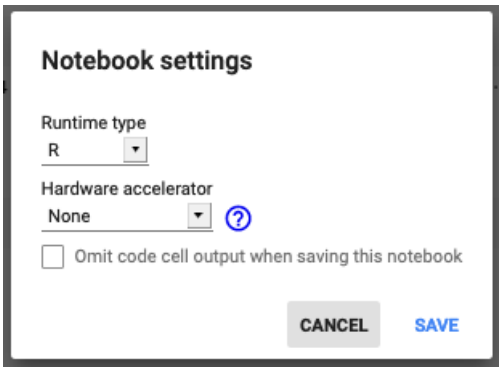

Image by Author, Runtime -> Change runtime type

You can also confirm that you are in the R runtime by trying to mount your Drive to the notebook. Doing so, you will get an unfortunate message like this:

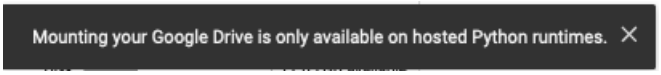

Image by Author, Uh-oh, we can't mount our Google Drive!

The message *"Mounting your Google Drive is only available on hosted Python runtimes."* clearly indicates that you are not in the Python runtime.

Congratulations, you have now successfully run R in Colab. You can check the R version by typing `R.version.string` to print out the R version.

Here, several packages that are useful for data processing and data visualization are already available. You can check it by running `print(installed.packages())`.

If you're having trouble installing packages, this post might help you:

How to Install Packages in R Google Colab

Some limitations and how to overcome some of them

towardsdatascience.com

How to mount Google Drive in Colab R runtime

This should be done fairly easily. We only need to install the "googledrive" package and perform the authentication process.

```
install.packages("googledrive")
library("googledrive")
```

After installing the package, we need to authenticate and authorize the googledrive package. You can read the package documentation here:

An Interface to Google Drive

Most functions begin with the prefix drive_. Auto-completion is your friend. Goal is to allow Drive access that feels…

googledrive.tidyverse.org


```
# authorize google drive

drive_auth(
  email = gargle::gargle_oauth_email(),
  path = NULL,
  scopes = "https://www.googleapis.com/auth/drive",
  cache = gargle::gargle_oauth_cache(),
  use_oob = gargle::gargle_oob_default(),
  token = NULL
)
```

Unfortunately the process did not go smoothly when trying to authenticate. We are instead faced with an error message like this:

```
Error: Can't get Google credentials.
Are you running googledrive in a non-interactive session? Consider:
  * `drive_deauth()` to prevent the attempt to get credentials.
  * Call `drive_auth()` directly with all necessary specifics.
  * Read more in: https://gargle.r-lib.org/articles/non-interactive-auth.html
Traceback:

1. drive_auth()
2. stop("Can't get Google credentials.\n", "Are you running googledrive in a non-interactive session? Consider:\n",
 .     "  * `drive_deauth()` to prevent the attempt to get credentials.\n",
 .     "  * Call `drive_auth()` directly with all necessary specifics.\n",
 .     "  * Read more in: https://gargle.r-lib.org/articles/non-interactive-auth.html",
 .     call. = FALSE)
```

Image by Author, Error: Can't get Google credentials.

Apparently, the error occurred because the interactive function in *httr* package could not be executed.

Here's a workaround that we can use, provided by jobdiogene's: https://gist.github.com/jobdiogenes/235620928c84e604c6e56211ccf681f0

```
# Check if is running in Colab and redefine is_interactive()
if (file.exists("/usr/local/lib/python3.6/dist-packages/google/colab/_ipython.py")) {
  install.packages("R.utils")
  library("R.utils")
  library("httr")
  my_check <- function() {return(TRUE)}
  reassignInPackage("is_interactive", pkgName = "httr", my_check)
  options(rlang_interactive=TRUE)
}
```

After running that lines, we can try to authenticate Google Drive again, and now it will work!

```
drive_auth(use_oob = TRUE, cache = TRUE)
```

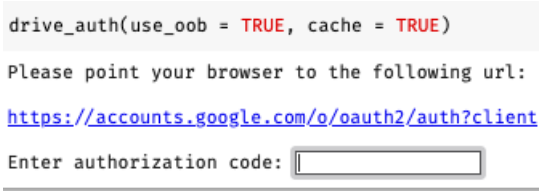

Image by Author, Interactive auth dialog

You will need to click the link and grant permission for the packages to access your Google Drive. After this you should be able to get the authorization code to be pasted in the code field.

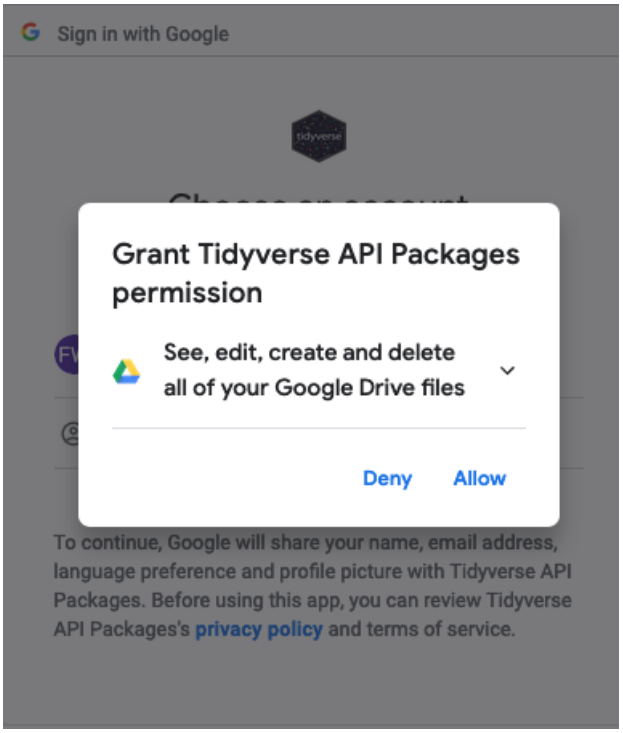

Image by Author, Grant permission

How to use BigQuery in Colab R runtime

For business people, or for researchers who are more comfortable using R, perhaps we need to retrieve data from company-owned BigQuery or publicly available datasets there.

Now that we have the workaround, authorizing BigQuery and retrieve data from there would be simple:

```
install.packages("bigrquery")
library("bigrquery")
bq_auth(use_oob = TRUE, cache = FALSE)
```

Extract data from BigQuery with custom query:

```
# Store the project id
projectid = "your-project-id"
```

```
# Set the query
sql <- "SELECT * FROM your_table"

# Run the query
project_query <- bq_project_query(projectid, sql, use_legacy_sql = FALSE)

# Download result to dataframe
df <- bq_table_download(project_query)
```

Conclusion

This is what I think I can contribute to the data community. As I mentioned earlier, Google Colab provide us an alternative for learning or working with R, besides Kaggle and RStudio Cloud. All of them are good platforms, especially when used for learning purposes; can shorten the time for initial setup (downloading and installing R, and installing packages). Even though the way to use R in Google Colab is a bit confusing, and doesn't yet have the same services as the Python runtime, in the end, it still works quite well