Get Started: 3 Ways to Load CSV files into Colab

A Apte Follow

Nov 27, 2018 · 4 min read

Data science is nothing without data. Yes, that's obvious. What is not so obvious is the series of steps involved in getting the data into a format which allows you to explore the data. You may be in possession of a dataset in CSV format (short for *comma-separated values*) but no idea what to do next. This post will help you get started in data science by allowing you to load your CSV file into Colab.

Colab (short for Colaboratory) is a free platform from Google that allows users to code in Python. Colab is essentially the Google Suite version of a Jupyter Notebook. Some of the advantages of Colab over Jupyter include an easier installation of packages and sharing of documents. Yet, when loading files like CSV files, it requires some extra coding. I will show you three ways to load a CSV file into Colab and insert it into a Pandas dataframe.

(Note: there are Python packages that carry common datasets in them. I will not discuss loading those datasets in this article.)

To start, log into your Google Account and go to Google Drive. Click on the **New** button on the left and select **Colaboratory** if it is installed (if not click on **Connect more apps**, search for Colaboratory and install it). From there, import Pandas as shown below (Colab has it installed already).

```
import pandas as pd
```

1) From Github (Files < 25MB)

The easiest way to upload a CSV file is from your GitHub repository. Click on the dataset in your repository, then click on **View Raw**. Copy the link to the raw dataset and store it as a string variable called url in Colab as shown below (a cleaner method but it's not necessary). The last step is to load the url into Pandas read_csv to get the dataframe.

```
url = 'copied_raw_GH_link'

df1 = pd.read_csv(url)

# Dataset is now stored in a Pandas Dataframe
```

2) From a local drive

To upload from your local drive, start with the following code:

```
from google.colab import files
uploaded = files.upload()
```

It will prompt you to select a file. Click on "**Choose Files**" then select and upload the file. Wait for the file to be 100% uploaded. You should see the name of the file once Colab has uploaded it.

Finally, type in the following code to import it into a dataframe (make sure the filename matches the name of the uploaded file).

```
import io

df2 = pd.read_csv(io.BytesIO(uploaded['Filename.csv']))

# Dataset is now stored in a Pandas Dataframe
```

3) From Google Drive via PyDrive

This is the most complicated of the three methods. I'll show it for those that have uploaded CSV files into their Google Drive for workflow control. First, type in the following code:

```
# Code to read csv file into Colaboratory:

!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials

# Authenticate and create the PyDrive client.
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

When prompted, click on the link to get authentication to allow Google to access your Drive. You should see a screen with "**Google Cloud SDK wants to access your Google Account**" at the top. After you allow permission, copy the given verification code and paste it in the box in Colab.

Once you have completed verification, go to the CSV file in Google Drive, right-click on it and select "**Get shareable link**". The link will be copied into your clipboard. Paste this link into a string variable in Colab.

```
link = 'https://drive.google.com/open?
id=1DPZZQ43w8brRhbEMolgLqOWKbZbE-IQu' # The shareable link
```

What you want is the id portion *after the equal sign*. To get that portion, type in the following code:

```
fluff, id = link.split('=')

print (id) # Verify that you have everything after '='
```

Finally, type in the following code to get this file into a dataframe

```
downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile('Filename.csv')
```
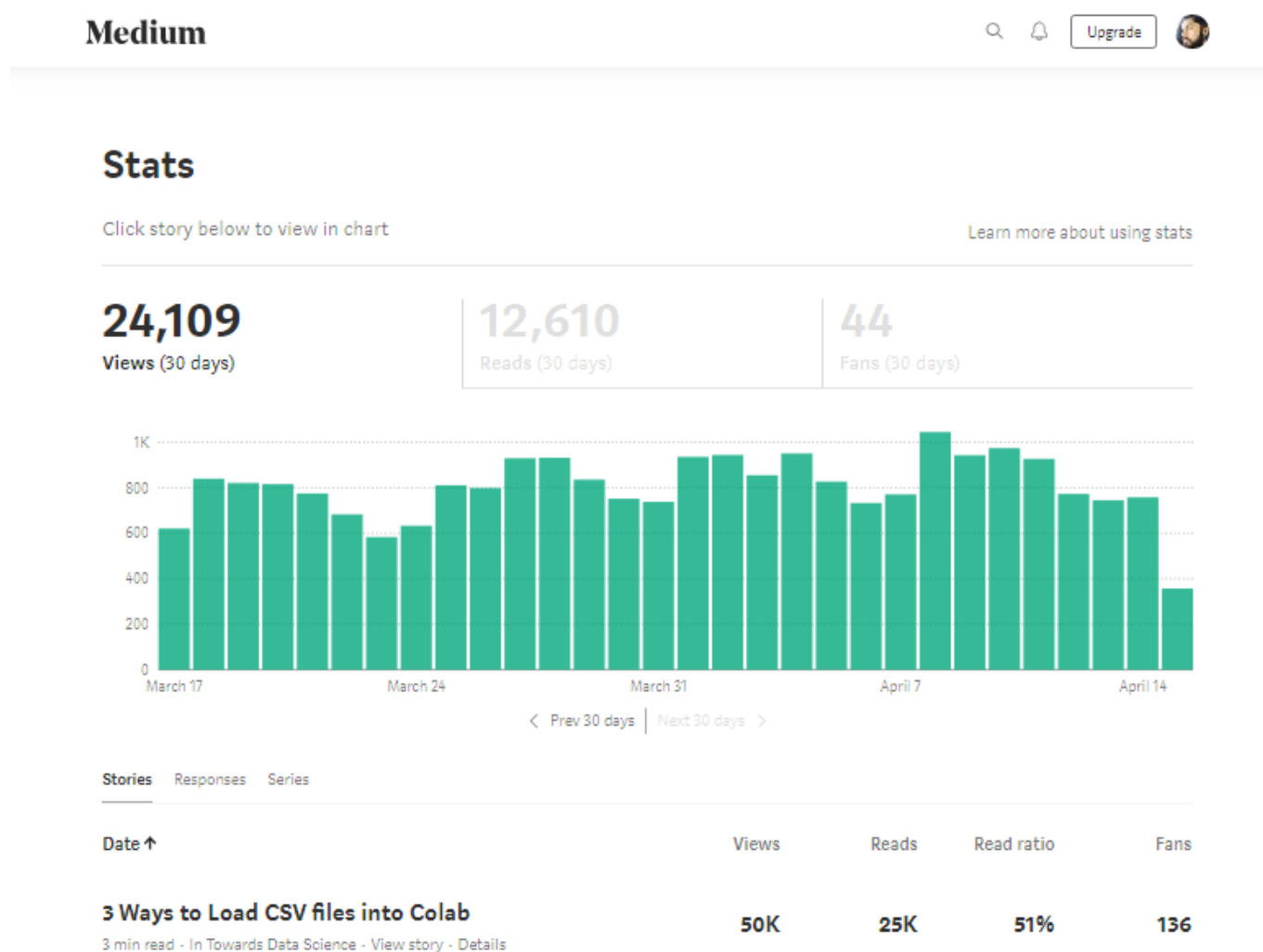
```
df3 = pd.read_csv('Filename.csv')

# Dataset is now stored in a Pandas Dataframe
```

Final Thoughts

These are three approaches to uploading CSV files into Colab. Each has its benefits depending on the size of the file and how one wants to organize the workflow. Once the data is in a nicer format like a Pandas Dataframe, you are ready to go to work.

Bonus Method — My Drive

Thank you so much for your support. In honor of this article reaching 50k Views and 25k Reads, I'm offering a bonus method for getting CSV files into Colab. This one is quite simple and clean. In your Google Drive ("**My Drive**"), create a folder called **data** in the location of your choosing. This is where you will upload your data.

From a Colab notebook, type the following:

```
from google.colab import drive
drive.mount('/content/drive')
```

Just like with the third method, the commands will bring you to a Google Authentication step. You should see a screen with **Google Drive File Stream wants to access your Google Account.** After you allow permission, copy the given verification code and paste it in the box in Colab.

In the notebook, click on the charcoal **>** on the top left of the notebook and click on **Files**. Locate the **data** folder you created earlier and find your data. Right-click on your data and select **Copy Path**. Store this copied path into a variable and you are ready to go.

```
path = "copied path"
df_bonus = pd.read_csv(path)

# Dataset is now stored in a Pandas Dataframe
```

What is great about this method is that you can access a dataset from a separate dataset folder you created in your own Google Drive without the extra steps involved in the third method.

Data Science   Python Pandas   File Management   Colab   Csv