

Google Colaboratory で Matplotlib を動かす準備

2023年度2Q 5c/6c (IL2) 木曜日

担当：地引

TA：増井

はじめに

これから行なうこと

- 少し前までは、教育用計算機システムにインストールされている gnuplot を用いてデータの整理/解析を行ないました。
- 最近では、情報通信環境を取り巻く状況を考慮し、ここではクラウド環境上で提供されるサービスを利用することになっています。
 - 個人のPCや教育用計算機システム上の gnuplot を自己裁量で利用してもよいです。
- 今回利用するサービスの構成：
 - Google が提供するプログラミング言語 Python の開発環境である **Google Colaboratory** を利用し、**その上で** gnuplot に相当する **Matplotlib** を使います。
 - とは言え、Python による本格的なプログラミング演習をするわけではありません。まずは、Colab 上で提供される Matplotlib を使うだけです。

Google Colaboratory を利用するための準備

- Google Colaboratory はクラウド上のサービスで、Google アカウントを用いてログインした後に利用します。
 - つまり、事前に Google アカウントを取得しておく必要があります。
- 既に Google アカウントを取得しており、それを利用してほしいという人は、それを使って下さい。
- Google アカウントを取得していない or アカウントを切り分けて使いたいという人は、下記のリンクをクリックし、アカウントを作成して下さい。

[Googleアカウントの作成 \(https://www.google.com/accounts/NewAccount\)](https://www.google.com/accounts/NewAccount)

Google Colaboratory の基本操作

Google Colaboratory へのログイン (1)

- Google Colaboratory は、下記の URL からログインできます。
 - <https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja>
 - 右端上に表示される **ログイン** をクリックし、各自の Google アカウントを用いてログインして下さい。



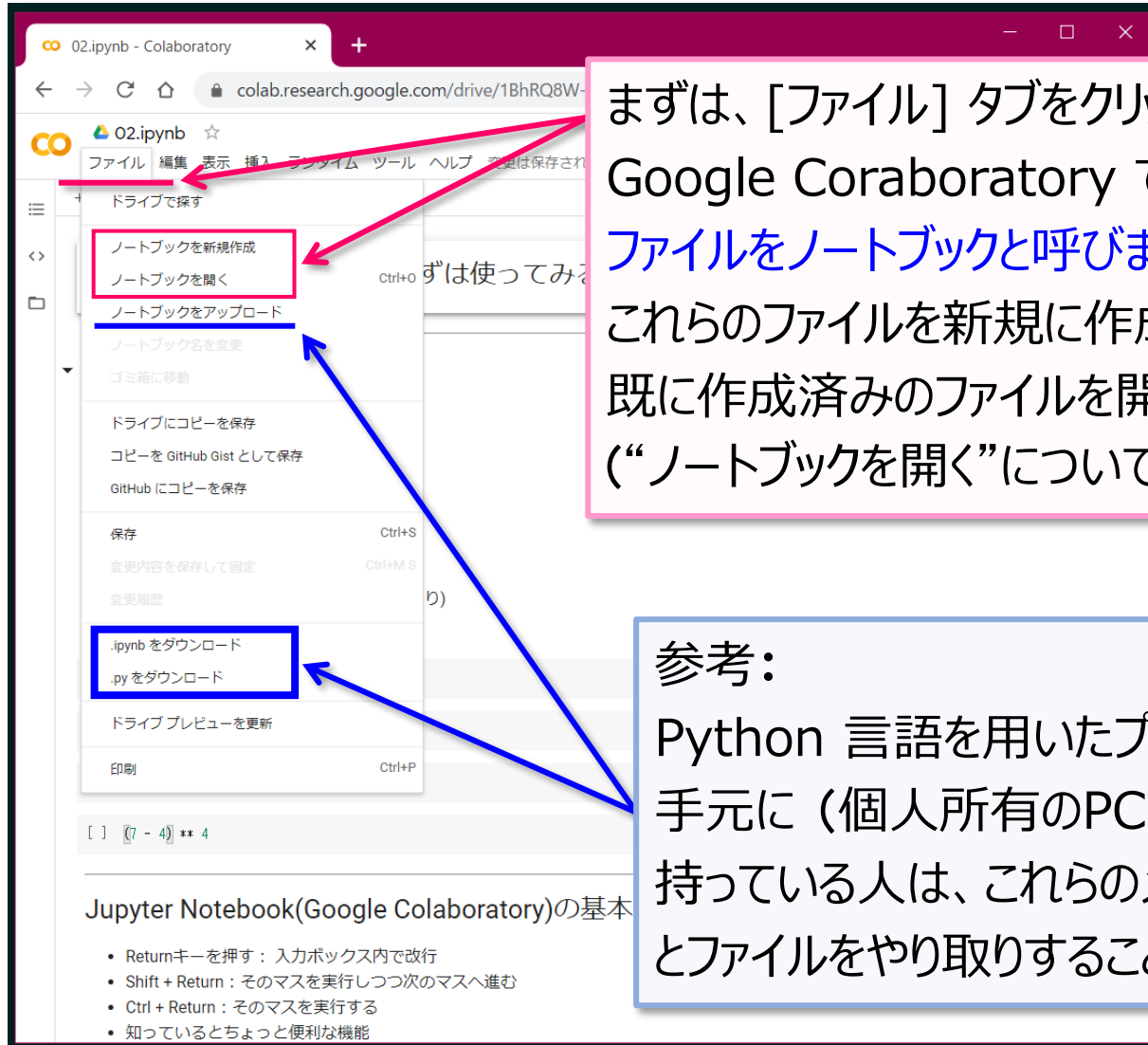
Google Colaboratory へのログイン (2)

ログイン後は、以下のようなウィンドウが表示されます (変化に気付きにくい…)。



確認

Matplotlib ファイルの作成



The screenshot shows the Google Colaboratory interface. The 'File' menu is open, and several options are highlighted with red and blue boxes and arrows. A red box highlights 'ノートブックを新規作成' (Create new notebook) and 'ノートブックを開く' (Open notebook). A blue box highlights '.ipynb をダウンロード' (Download .ipynb) and '.py をダウンロード' (Download .py). A red arrow points from the text box to the 'File' menu, and another red arrow points from the text box to the 'ノートブックを新規作成' option. A blue arrow points from the text box to the '.ipynb をダウンロード' option.

まずは、[ファイル] タブをクリックします。

Google Coraboratory では、Python や Matplotlib のプログラムファイルをノートブックと呼びます。

これらのファイルを新規に作成する場合は“ノートブックを新規作成”を、既に作成済みのファイルを開く場合は“ノートブックを開く”を選択します。（“ノートブックを開く”については、後ほど補足します）。

参考：

Python 言語を用いたプログラミングに興味があり、手元に（個人所有のPC上に）Anaconda などの開発環境を持っている人は、これらのメニューで、手元の Jupyter Notebook とファイルをやり取りすることができます。

注意:

以下では、**便宜上**、Google Colaboratory 内で作成する Python および **Matplotlib のファイル**を**総称して**、「**プログラム ファイル**」と呼ぶことにします。

とはいえ、情報リテラシでは、Python によるプログラミング スキルの習得が目標ではありません。

3Q/4Q のコンピュータ サイエンスでは、Google Colab 上でより本格的な Python プログラムを作成する場合があります。

プログラムの記述・実行

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The interface includes a top menu bar with options like "ファイル", "編集", "表示", "挿入", "ランタイム", "ツール", and "ヘルプ". A sidebar on the left shows a table of contents with "目次" and "セクション". The main area contains a code cell with the text "3+5" and a play button icon. Below the code cell, the result "8" is displayed. Several callout boxes provide instructions: a yellow box at the top points to the "ファイル" menu; a pink box on the right explains the execution steps; a green box on the left points to the play button; a blue box at the bottom points to the result; and a yellow box at the bottom right provides additional methods for adding code.

ファイル名を変更したい場合は、[ファイル] タブ以下にメニューがあります。

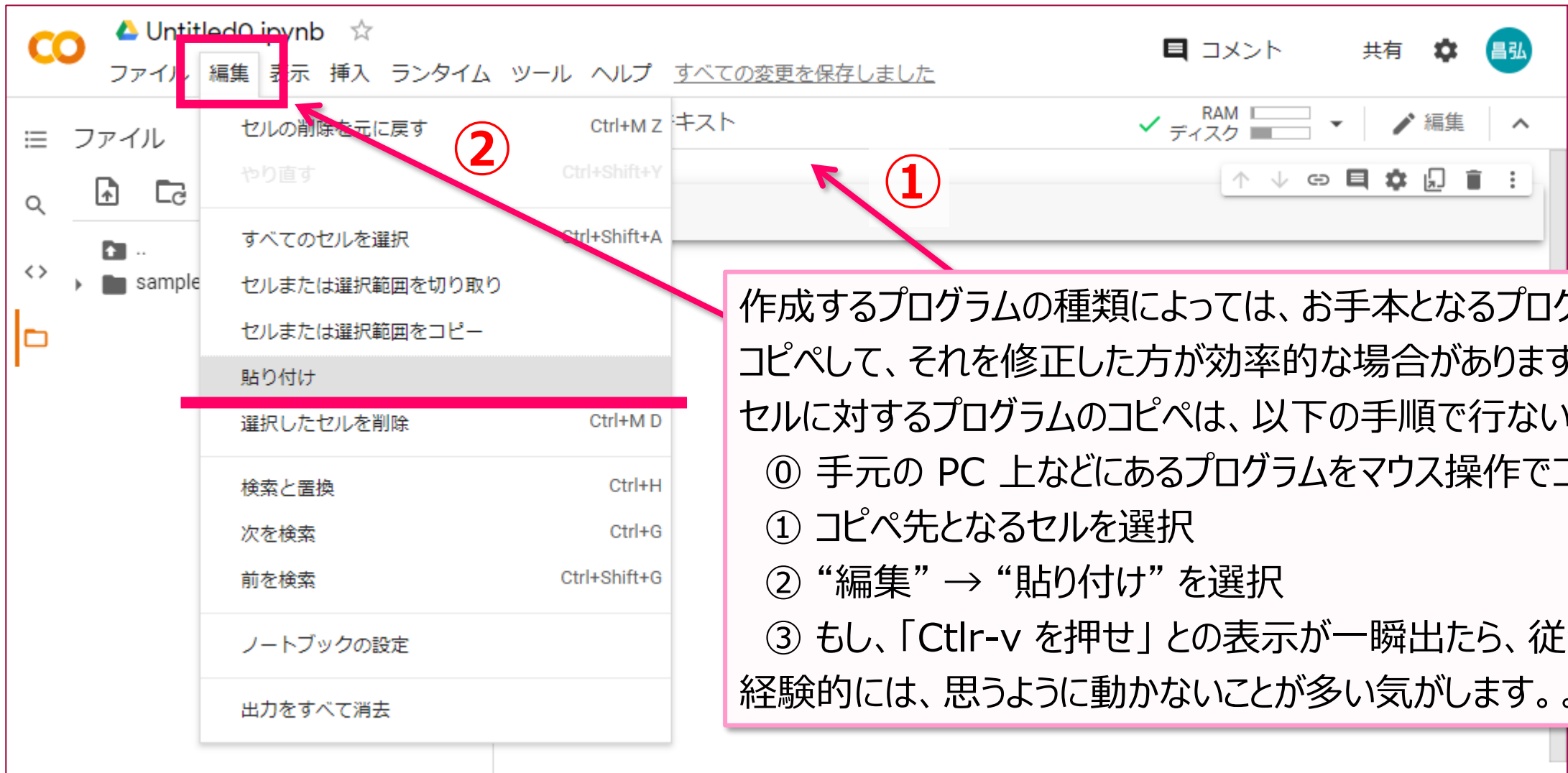
プログラムの記述と実行は、次の手順で行ないます。

- ① “+ コード” をクリックすると、プログラムを記述するセルが用意されます。
- ② このボタンを押すことで、プログラムが実行されます。
- ③ コード セルの下に結果が表示されます。

プログラムをさらに記述したい場合は、下記のどちらの方法でも可能です。

- 既存のコードセルに追記する。
- 再度 “+ ファイル” を実行し、新たに用意したコード セルに記述する。

プログラムのコピー&ペースト

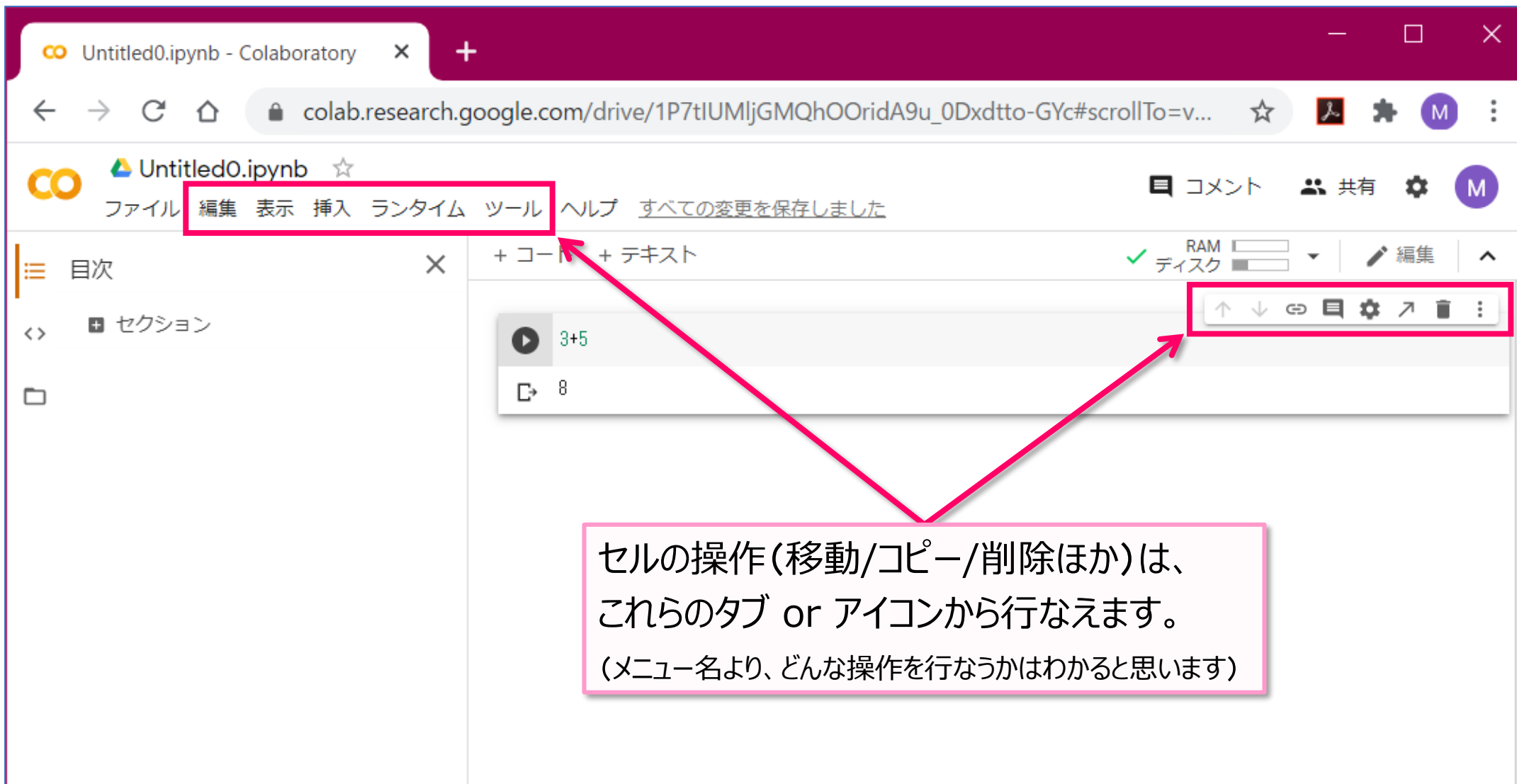


The screenshot shows the JupyterLab interface with the 'Edit' menu open. The 'Edit' menu is highlighted with a red box and a red arrow labeled '2'. The 'Paste' option is highlighted with a red box and a red arrow labeled '1'. The 'Edit' menu items are: セルの削除を元に戻す (Ctrl+M Z), やり直す (Ctrl+Shift+Y), すべてのセルを選択 (Ctrl+Shift+A), セルまたは選択範囲を切り取り, セルまたは選択範囲をコピー, 貼り付け (highlighted), 選択したセルを削除 (Ctrl+M D), 検索と置換 (Ctrl+H), 次を検索 (Ctrl+G), 前を検索 (Ctrl+Shift+G), ノートブックの設定, 出力をすべて消去.

作成するプログラムの種類によっては、お手本となるプログラムをコピーして、それを修正した方が効率的な場合があります。セルに対するプログラムのコピーは、以下の手順で行ないます。

- ① 手元の PC 上などにあるプログラムをマウス操作でコピー
- ① コピー先となるセルを選択
- ② “編集” → “貼り付け” を選択
- ③ もし、「Ctrl-v を押せ」との表示が一瞬出たら、従う。経験的には、思うように動かないことが多い気がします。。

セルの操作



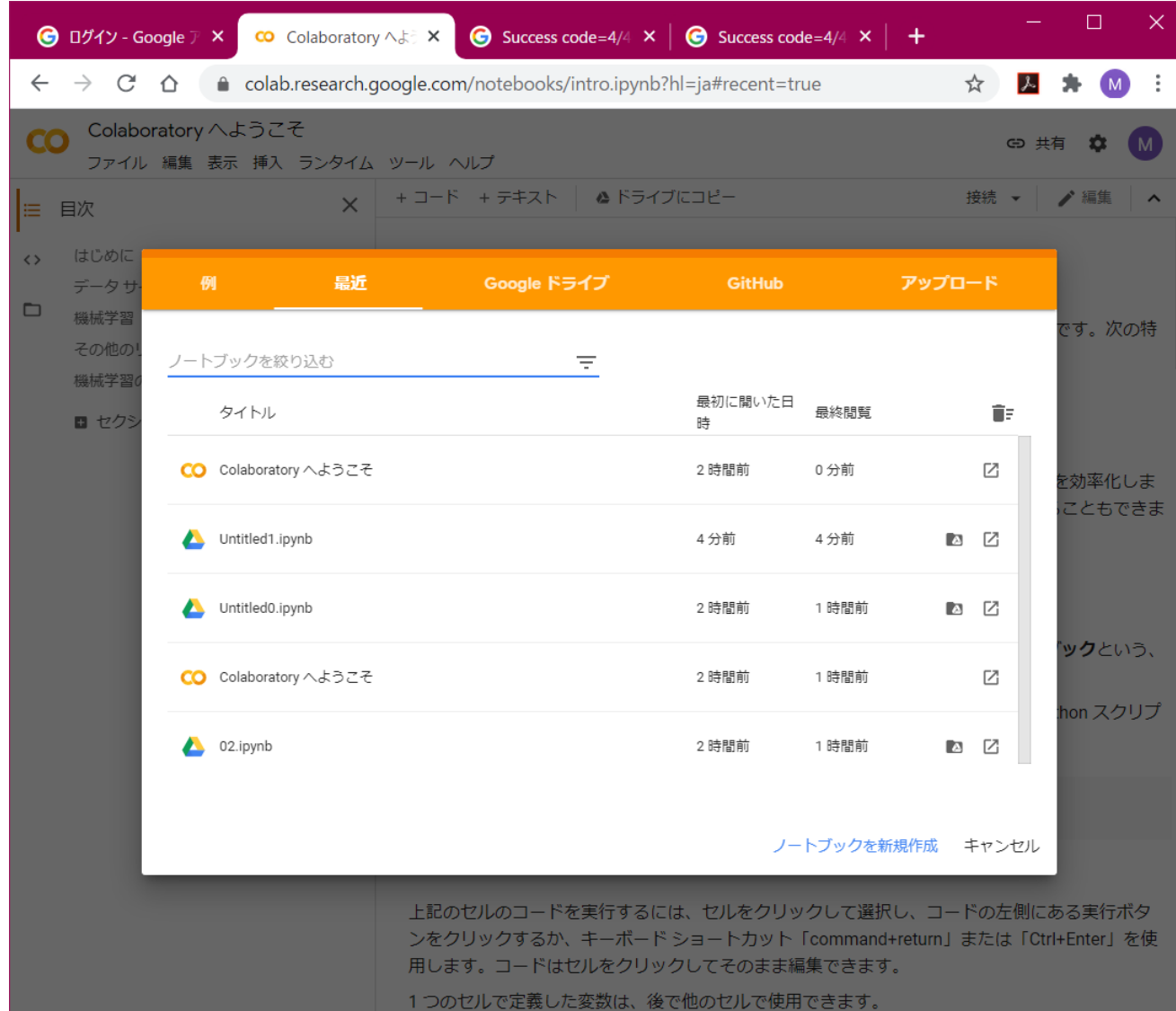
The screenshot shows the Google Colaboratory interface. At the top, there is a browser window with the address bar showing 'colab.research.google.com'. Below the browser, the Colaboratory header includes 'Untitled0.ipynb' and a menu with options like 'ファイル', '編集', '表示', '挿入', 'ランタイム', 'ツール', and 'ヘルプ'. The main workspace contains a code cell with the text '3+5' and its output '8'. A red box highlights the menu options '編集', '表示', '挿入', and 'ランタイム'. Another red box highlights a toolbar with icons for up, down, link, comment, settings, refresh, and delete. Two red arrows point from the text box below to these two red boxes.

セルの操作(移動/コピー/削除ほか)は、これらのタブ or アイコンから行なえます。(メニュー名より、どんな操作を行なうかはわかると思います)

ノートブックを開く

Google Colaboratory や Jupyter Notebook では、プログラム ファイルをノートブックと呼びます。

ノートブックを開く（1）



Google Colaboratory にログインする or [ファイル] → [ノートブックを開く] を選択すると、左のようなファイル一覧が表示されます。

しかし、これは最近開いたファイルの一覧であり、自分が開きたいファイルはこの中にもないかも知れません(履歴を消すなど)。

そのような場合は、これから説明する方法でファイルを探して下さい。

ノートブックを開く（2）

Colaboratory へようこそ

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

目次

- はじめに
- データサイエンス
- 機械学習
- その他のリソース
- 機械学習の例
- セクション

Colaboratory とは

Colaboratory (略称: Colab) は、ブラウザから Python を記述、実行できるサービスです。次の特長を備えています。

- 環境構築が不要
- GPU への無料アクセス
- 簡単

Colab は、詳しくは...

はじめ

ご覧になっているこのドキュメントは静的なウェブページではなく、Colab ノートブックという、コードを記述して実行できるインタラクティブな環境です。

たとえば次のコードセルには、値を計算して変数に保存し、結果を出力する短い Python スクリプトが記述されています。

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

86400

上記のセルのコードを実行するには、セルをクリックして選択し、コードの左側にある実行ボタンをクリックするか、キーボードショートカット「command+return」または「Ctrl+Enter」を使用します。コードはセルをクリックしてそのまま編集できます。

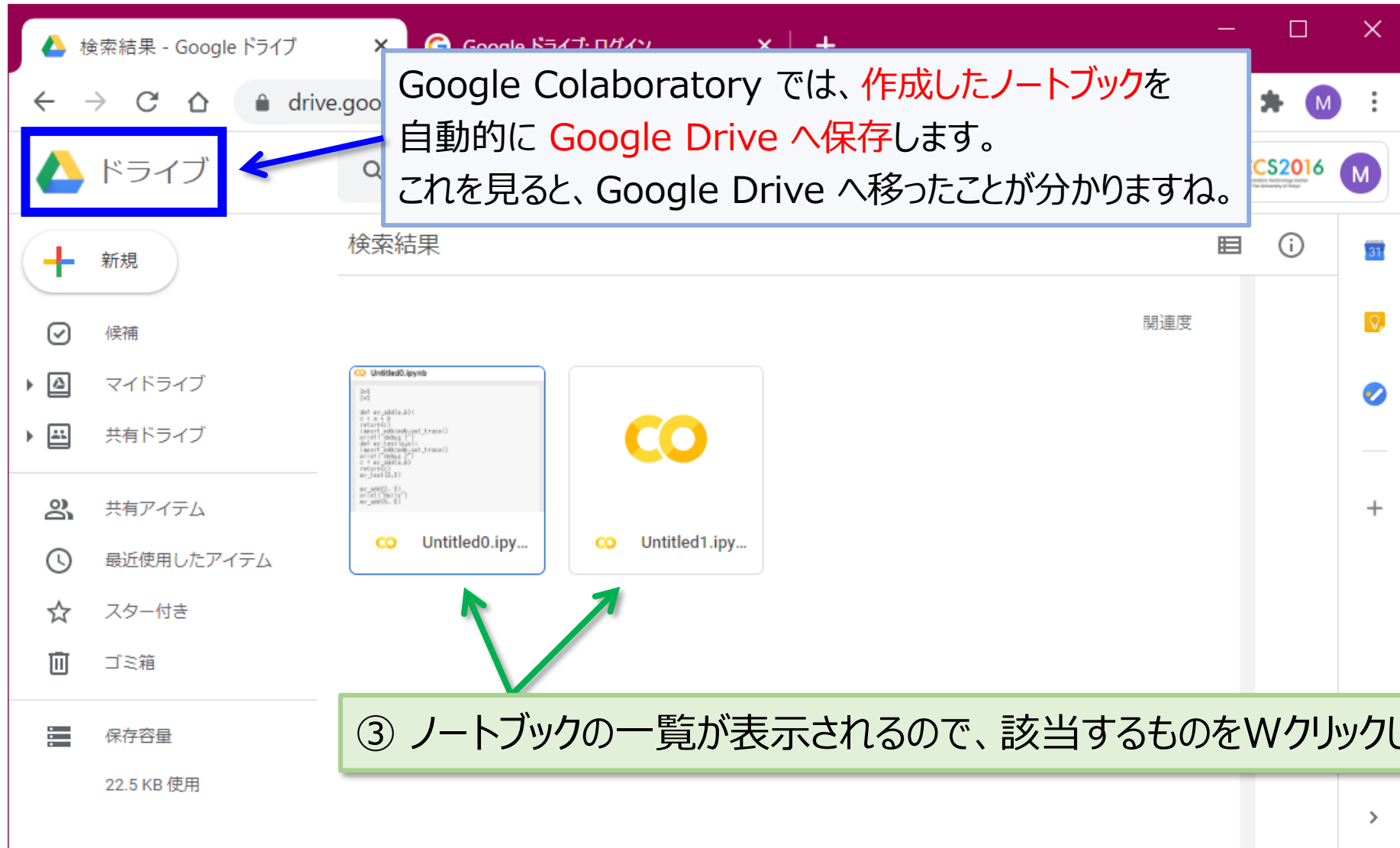
1つのセルで定義した変数は、後で他のセルで使用できます。

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
seconds_in_a_week
```

604800

- ① まずは、先ほどの一覧をキャンセルし、適当なノートブックを開きます。
- ② 次に左上にある ∞ アイコンをクリックします。

ノートブックを開く (3)



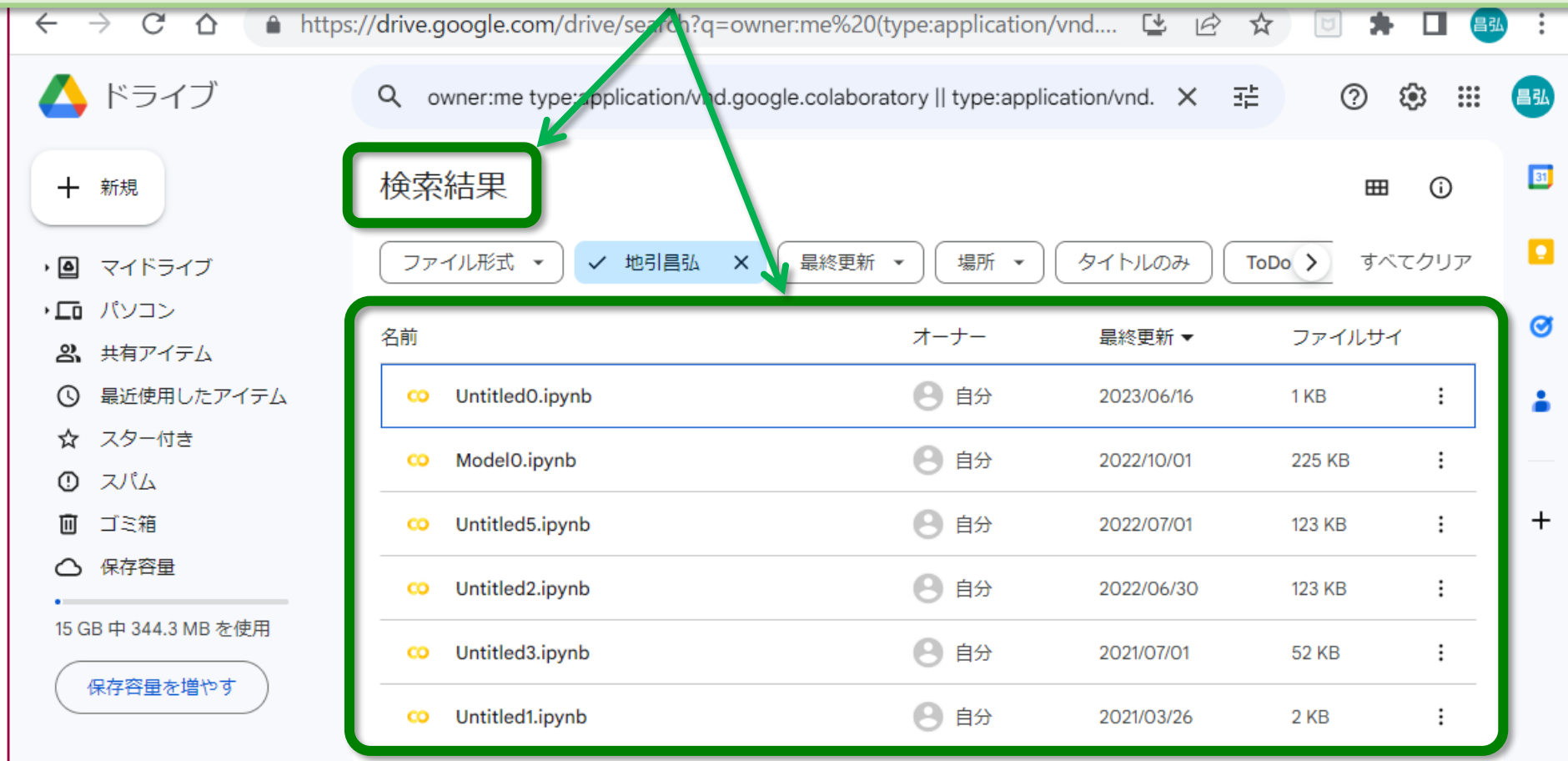
The screenshot shows the Google Colaboratory interface. At the top, there are browser tabs for '検索結果 - Google ドライブ' and 'Google ドライブ: ログイン'. The address bar shows 'drive.google.com'. A blue box highlights the 'ドライブ' (Drive) icon in the top left, with a blue arrow pointing to it. A text box above the main content area contains the following text: 'Google Colaboratory では、作成したノートブックを自動的に Google Drive へ保存します。これを見ると、Google Drive へ移ったことが分かりますね。' (In Google Colaboratory, created notebooks are automatically saved to Google Drive. Looking at this, you can see that they have moved to Google Drive.) Below this, the main content area shows a list of notebook thumbnails. The first thumbnail, 'Untitled0.ipynb', is highlighted with a blue box and contains a code snippet:

```
def ex_addr(a,b):  
    c = a + b  
    return c  
print(ex_addr(1,2))  
def ex_addr(a,b):  
    print('Debug 1')  
    def ex_addr(a,b):  
        print('Debug 2')  
        c = a + b  
        return c  
    ex_addr(1,2)  
    print('Debug 3')  
    ex_addr(1,2)
```

 The second thumbnail, 'Untitled1.ipynb', is also highlighted with a blue box and shows the Google Colaboratory logo. A green box at the bottom of the screenshot contains the text: '③ ノートブックの一覧が表示されるので、該当するものをWクリックします。' (③ Since the list of notebooks is displayed, double-click the appropriate one.)

重要!! (1)

とは言え、ここに一覧されるノートブックは、Google Drive 内にある全ノートブックの検索結果です。当然、ノートブックが増えて来ると、ズラズラと一覧され、お目当てのノートブックを探すのも大変です。



The screenshot shows the Google Drive search results page. The search bar contains the query "owner:me type:application/vnd.google.colaboratory || type:application/vnd.". The search results are displayed in a table with columns for name, owner, last updated, and file size. The results are sorted by "最終更新" (Last updated) in descending order. The first result is "Untitled0.ipynb" (1 KB, updated 2023/06/16). Other results include "Model0.ipynb" (225 KB, updated 2022/10/01), "Untitled5.ipynb" (123 KB, updated 2022/07/01), "Untitled2.ipynb" (123 KB, updated 2022/06/30), "Untitled3.ipynb" (52 KB, updated 2021/07/01), and "Untitled1.ipynb" (2 KB, updated 2021/03/26). A green box highlights the search results table, and a green arrow points from the text above to the search bar and the "検索結果" (Search results) label.

名前	オーナー	最終更新	ファイルサイ
Untitled0.ipynb	自分	2023/06/16	1 KB
Model0.ipynb	自分	2022/10/01	225 KB
Untitled5.ipynb	自分	2022/07/01	123 KB
Untitled2.ipynb	自分	2022/06/30	123 KB
Untitled3.ipynb	自分	2021/07/01	52 KB
Untitled1.ipynb	自分	2021/03/26	2 KB

重要!! (2)

保存先は Google Drive なので、別途 Google Drive にログインして、“Colab Notebooks” フォルダ以下にサブ フォルダを作成し、**適宜ノートブックを分類して整理**しましょう。
Colab からノートブックを開く時は、前スライドにある検索結果一覧ではなく、**左側に表示されるファイル システムを辿って**お目当てのノートブックを指定できるようになりましょう (**重要なスキルです**)。
Colab のメニューは、Google Drive をブラウザで開いた時と同じなので、操作も難しくないはずです。



ノートブックを開く（４） … 続き

検索結果 - Google ドライブ

Google ドライブ: ログイン

drive.google.com/drive/search?q=owner:me%20(type:application/vnd.google.colab...

Untitled0.ipynb Google Colaboratory で開く

新規

候補

マイドライブ

共有ドライブ

共有アイテム

最近使用したアイテム

スター付き

ゴミ箱

保存容量

22.5 KB 使用

```
3+5
2+5

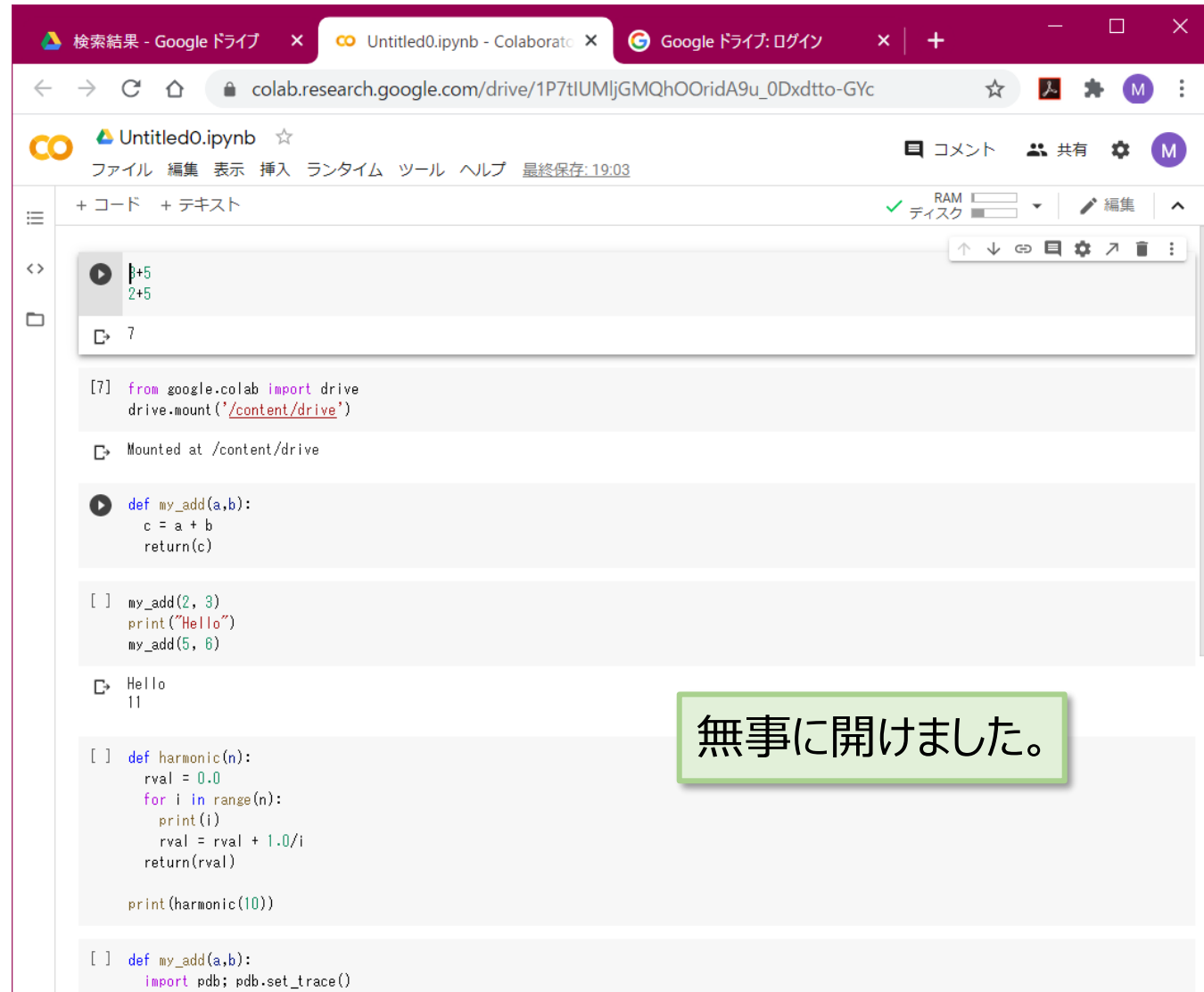
def my_add(a,b):
    c = a + b
    return(c)
import pdb;pdb.set_trace()
print("debug 1")
def my_test(a,b):
    import pdb;pdb.set_trace()
    print("debug 2")
    c = my_add(a,b)
    return(c)
my_test(2,3)

my_add(2, 3)
print("Hello")
my_add(5, 6)
```

この画面が表示されない場合もあります。

④ “Google Colaboratory” で開くを選択(クリック)します。

ノートブックを開く (5)



The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The interface includes a browser window with the URL `colab.research.google.com/drive/1P7tIUmljGMQhOOridA9u_0Dxdtto-GYc`. The notebook content is as follows:

```
+ コード + テキスト
▶ 1+5
  2+5
↳ 7

[7] from google.colab import drive
     drive.mount('/content/drive')
↳ Mounted at /content/drive

▶ def my_add(a,b):
    c = a + b
    return(c)

[ ] my_add(2, 3)
     print("Hello")
     my_add(5, 6)
↳ Hello
  11

[ ] def harmonic(n):
     rval = 0.0
     for i in range(n):
         print(i)
         rval = rval + 1.0/i
     return(rval)

     print(harmonic(10))

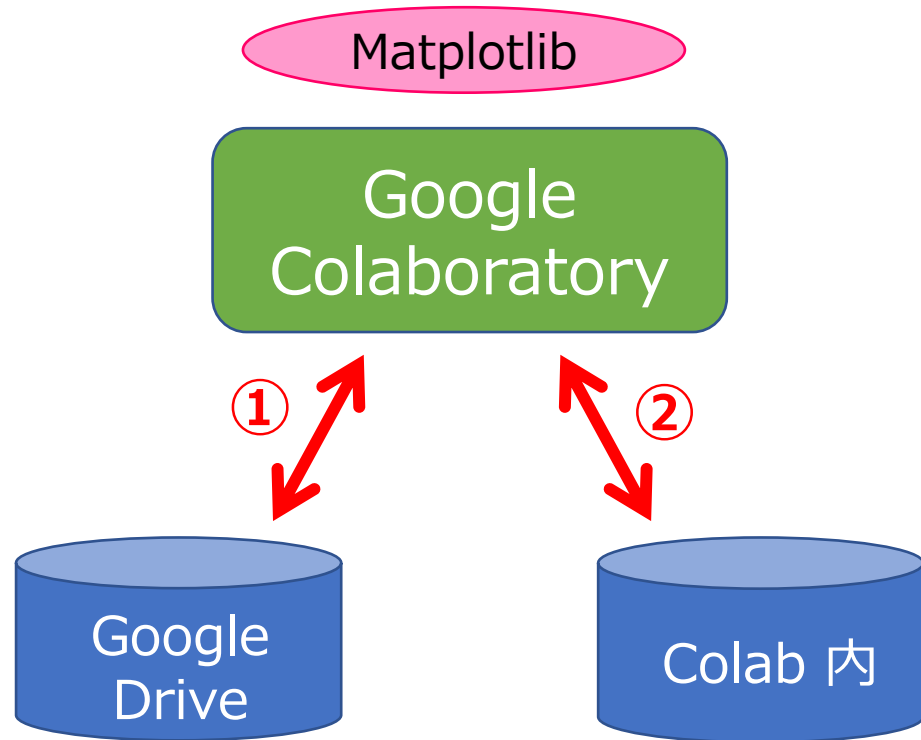
[ ] def my_add(a,b):
     import pdb; pdb.set_trace()
```

A green callout box with the text "無事に開けました。" (Successfully opened.) is overlaid on the right side of the notebook interface.

データ ファイルの扱い

Google Colaboratory のファイル管理イメージ

Google Colaboratory では、扱うファイルを下記①,②のどちらかに保存します。プログラム ファイルは、特に準備をしなくても①に保存されますが、データ用のファイルを保存する場合は、どちらに保存しても事前の準備や保存後の扱いが少々面倒です。



①にあるデータ ファイル(プログラム ファイルではない)を Colab からアクセスするには、Google Drive をマウント (一つのファイル システムとして接続)する必要があります。現状では、マウント処理を簡便に行なえない、およびパス名が複雑となるため、今回は②へ保存します(②はマウントが不要で、より直感的に扱えます)。

但し、Colab 内に保存したファイルは、一定時間後に削除されてしまうことに注意して下さい(後で補足します)。

テーマ 1 において習得するスキルの一つは、

「Matplotlib ファイルの実行場所(カレント フォルダ)とデータ ファイルの階層的な位置関係を正しく把握して、プログラム内に表記できること」です。

データ ファイルのアップロード(1)

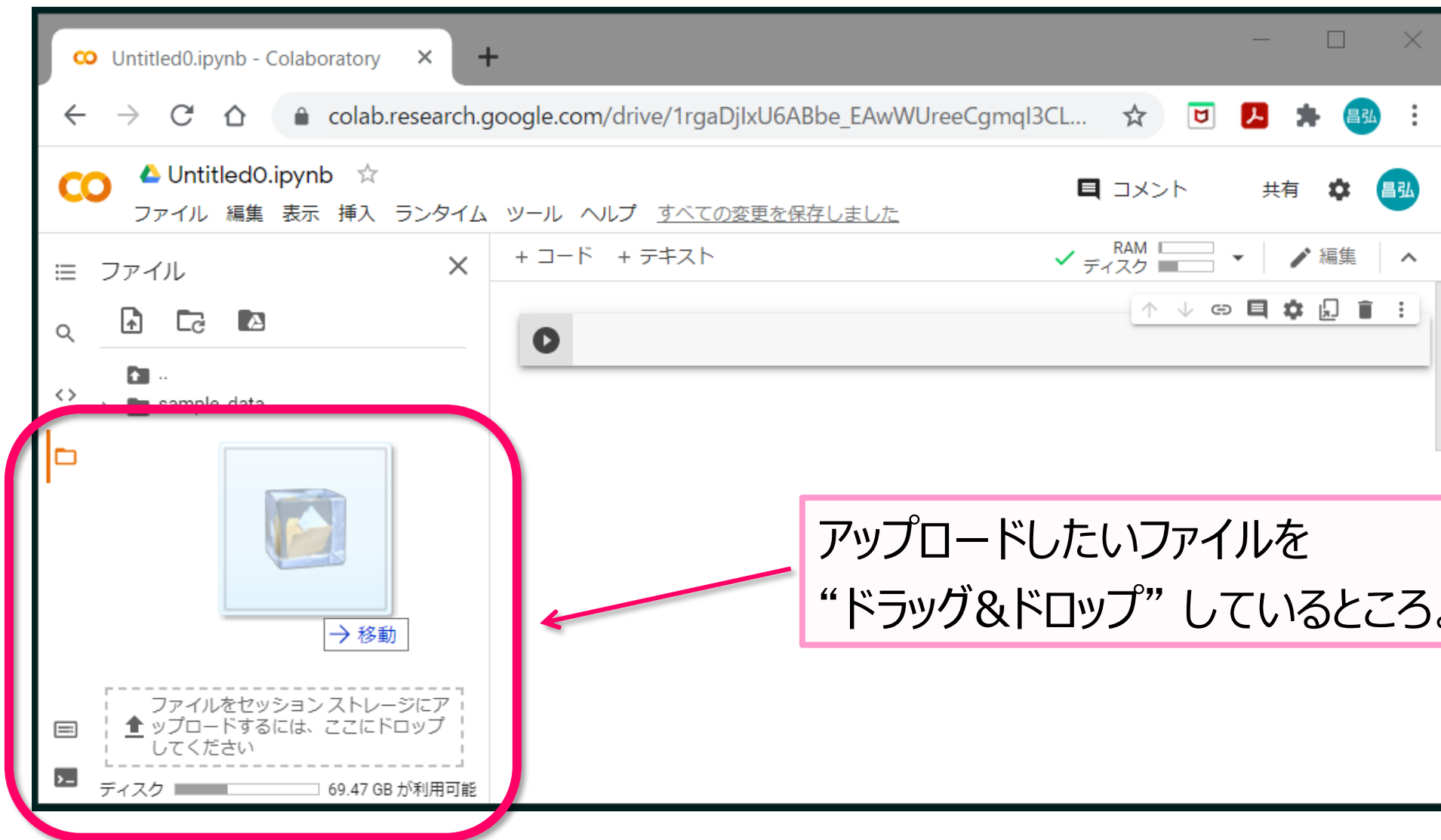
- テーマ 1 の解析で利用するデータを Colab にアップロードする手順
 - データ(theme1.zip)は、情報リテラシ第二 5c/6c ページの下記にあります。
[“データの処理と加工”で扱う解析用データ \(for matplotlib\)](#)
このリンクをクリックし、**まずは手元の情報機器へダウンロード**して下さい。
 - その後、次スライド以降に従い、Colab へアップロードします。
- **注意 !!**
 - Safari では、5c/6c ページより theme1.zip をダウンロードした段階で .zip が自動展開され、Colab へ効率的にアップロードできないことがあります。
 - これを回避する簡潔な方法を以下に紹介します。
 - ≫ a. Chrome を使う。
 - ≫ b. Safari を使う場合は、
Safari の環境設定を開いて「一般」タブを選択し、
「ダウンロード後、“安全な”ファイルを開く」のチェックを外す。

データ ファイルのアップロード(2)



- ① 一番左にあるフォルダの形をしたアイコン(ホバー表示は“ファイル”)をクリックします。
- ② この領域に、ファイル システムの階層構造が表示されます。
(手元の)ファイルを、ここへドラッグ&ドロップすると、Colab にアップロードされます。

データ ファイルのアップロード(3)



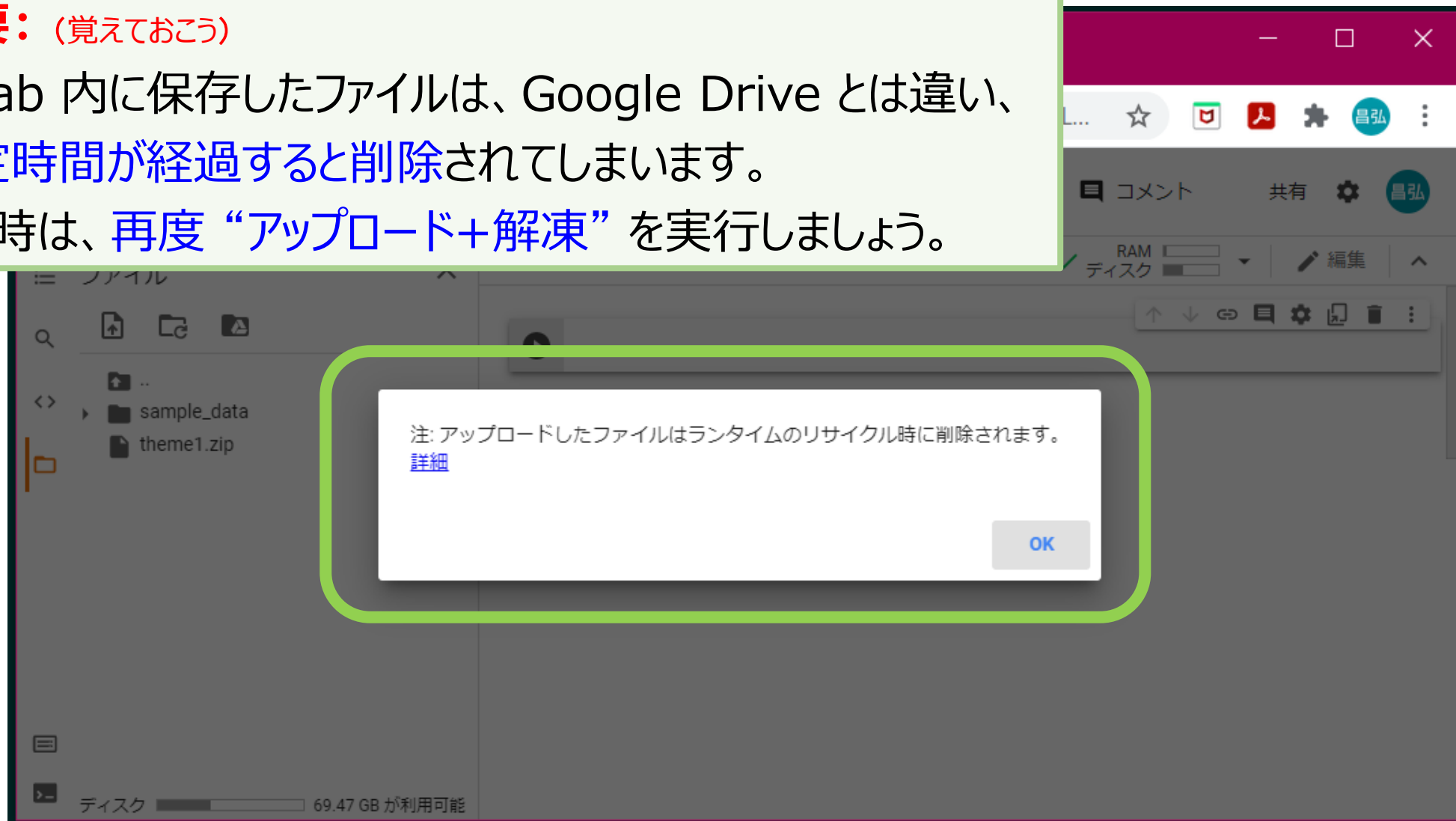
アップロードしたいファイルを
“ドラッグ&ドロップ”しているところ。

データ ファイルのアップロード(4)

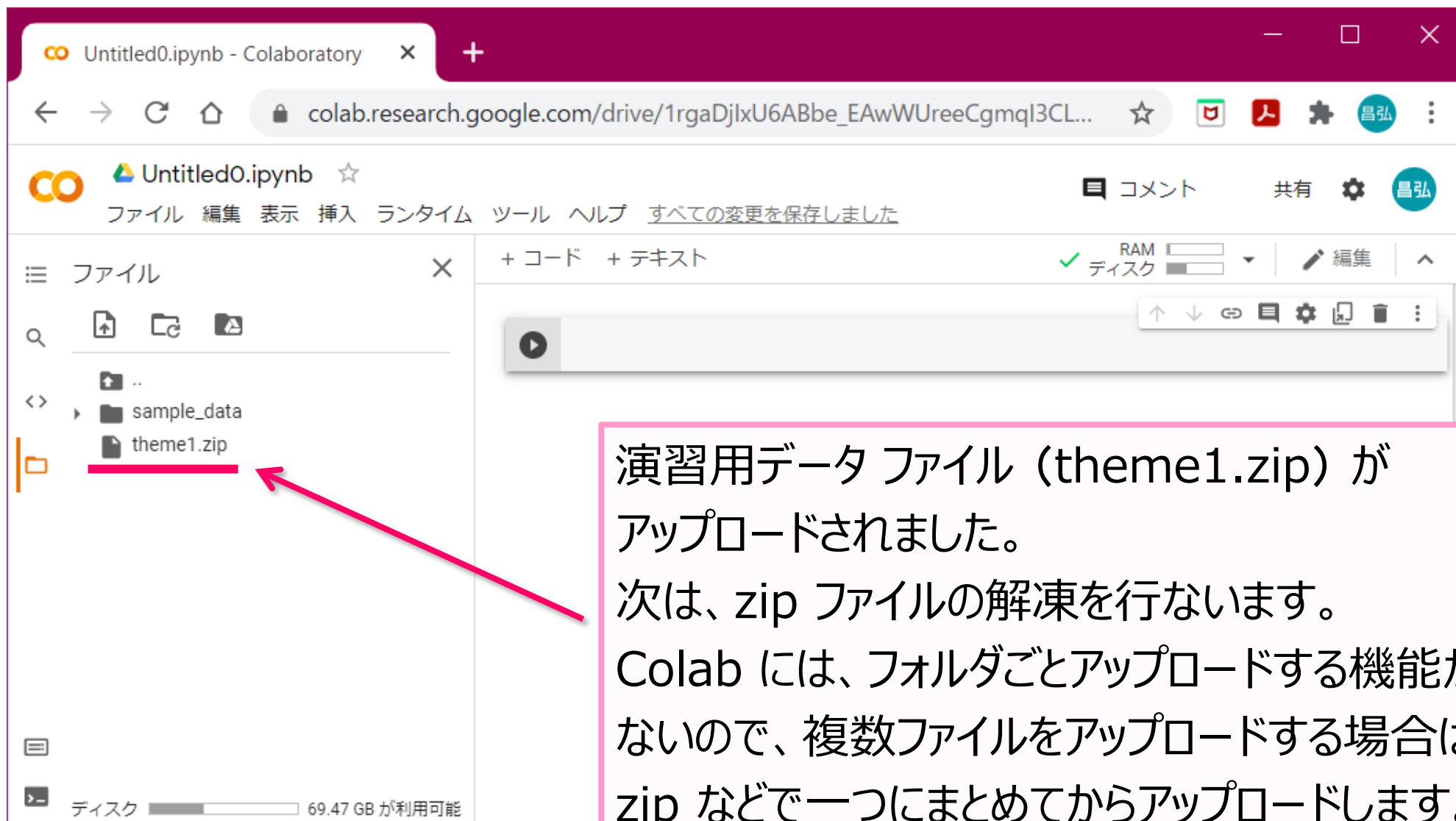
重要: (覚えておこう)

Colab 内に保存したファイルは、Google Drive とは違い、
一定時間が経過すると削除されてしまいます。

その時は、再度“アップロード+解凍”を実行しましょう。



データ ファイルのアップロード(5)

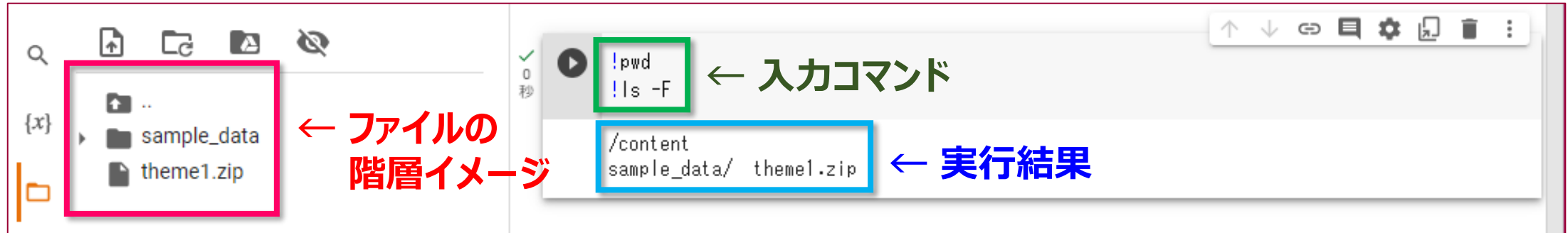


演習用データ ファイル (theme1.zip) がアップロードされました。

次は、zip ファイルの解凍を行ないます。

Colab には、フォルダごとアップロードする機能がないので、複数ファイルをアップロードする場合は、zip などで一つにまとめてからアップロードします。

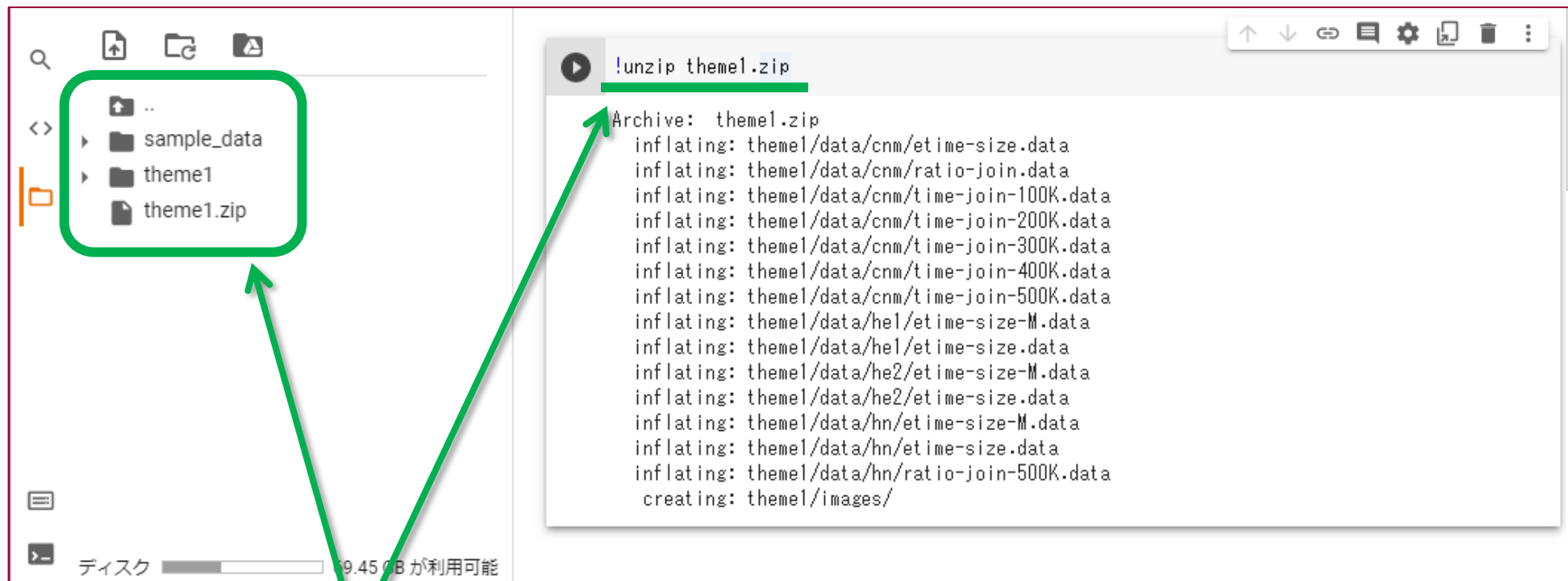
データ ファイルの解凍(1)



Colab では、コードの部分にプログラムではなくコマンドを書くことができます。
コマンドは、先頭に **!** を付けます。

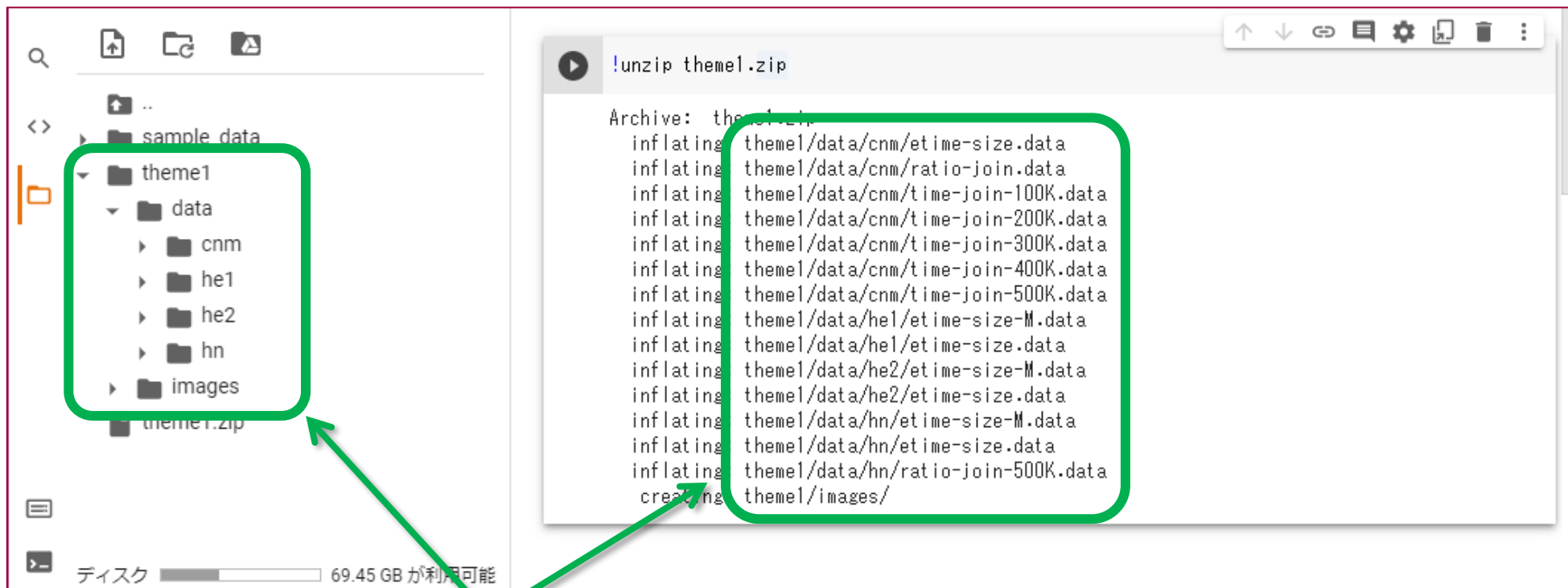
上の例では、ファイル システムの階層構造において、自分がどこにいるかを
表示する(つまり、カレント フォルダを表示する) `pwd` コマンドと、
フォルダ内のファイルを一覧する `ls` コマンド (`-F` オプション) を実行しています。
その結果として、カレント フォルダはルート フォルダ “/” の下にある
`content` フォルダであり、そこには `sample_data` フォルダと `theme1.zip`
ファイルがあることが示されています(後で補足します)。

データ ファイルの解凍(2)



カレント フォルダと演習用データ ファイルの位置関係を確認できたので、
unzip コマンドで解凍します (カレント フォルダ内なのでファイル名だけでOK)。
左側の階層図には、少し遅れて反映されます。

データ ファイルの解凍(3)

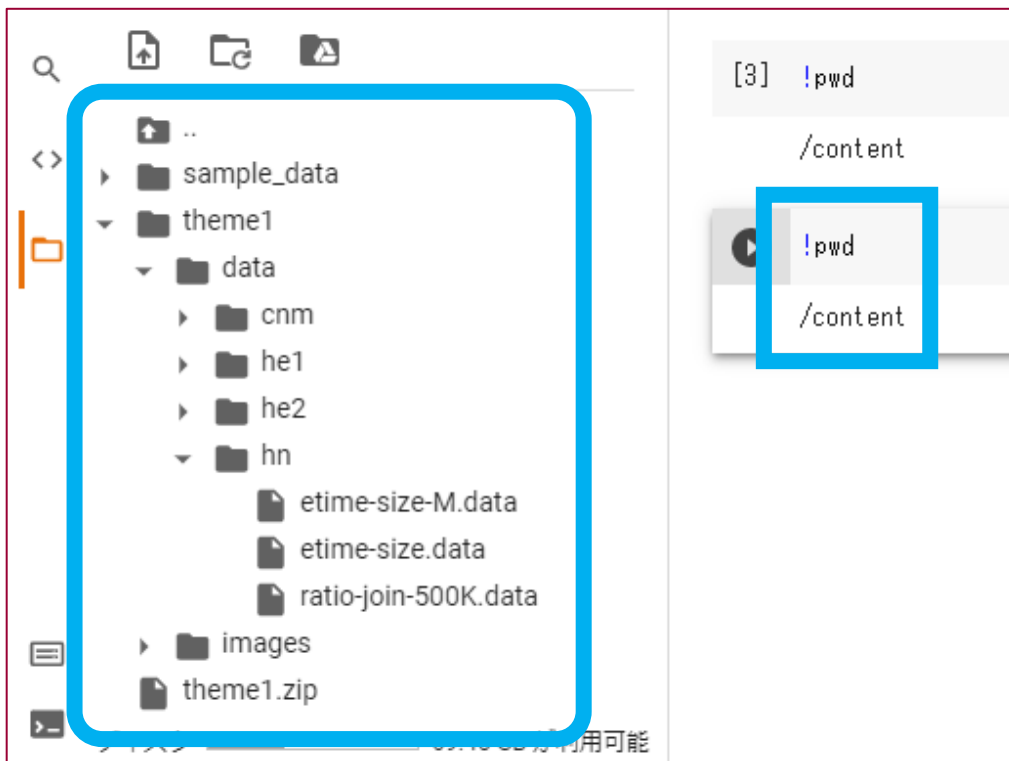


以後は、theme1 フォルダにある data フォルダ以下にある各データ ファイルを対象に、Matplotlib で解析を行ないます。

Matplotlib ファイルからは、データ ファイルのパス名を指定する必要があります。

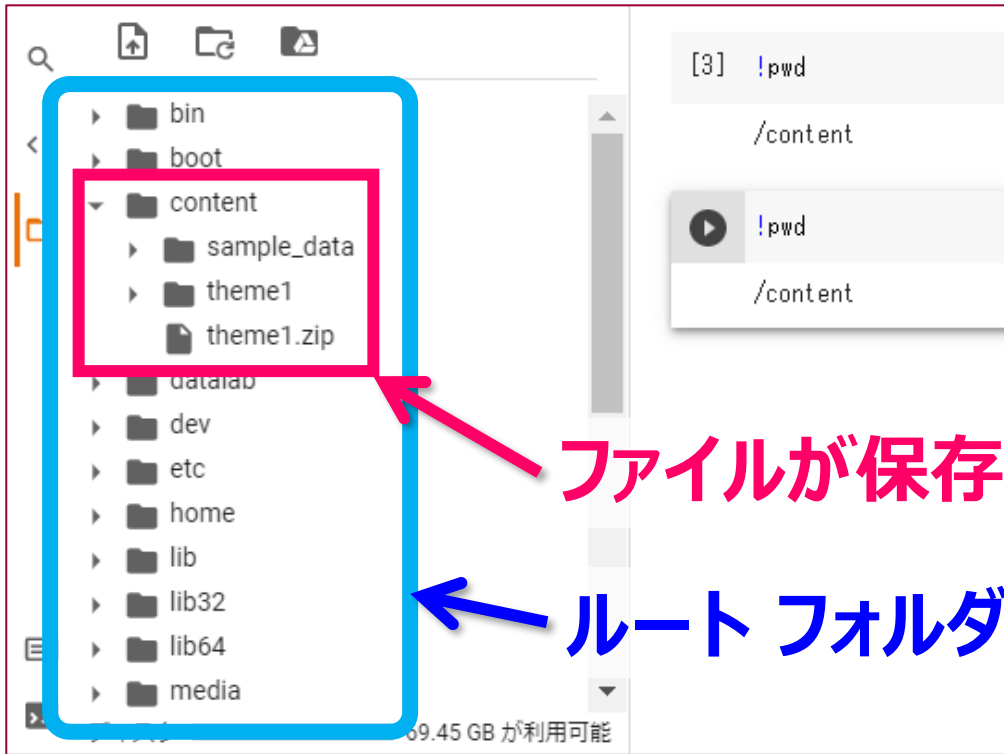
上の図を見て、データ ファイル群の全体構造をよく確認しておきましょう。

要注意(1)



左側に見える階層構造のビューですが、階層構造を展開しても、ノートブックの
カレント フォルダは変わりません。
ノートブックのカレント フォルダを変えたい
場合は、セルで !cd コマンドを実行します。
エクスプローラや Finder とは、違うので
注意して下さい。

要注意(2)

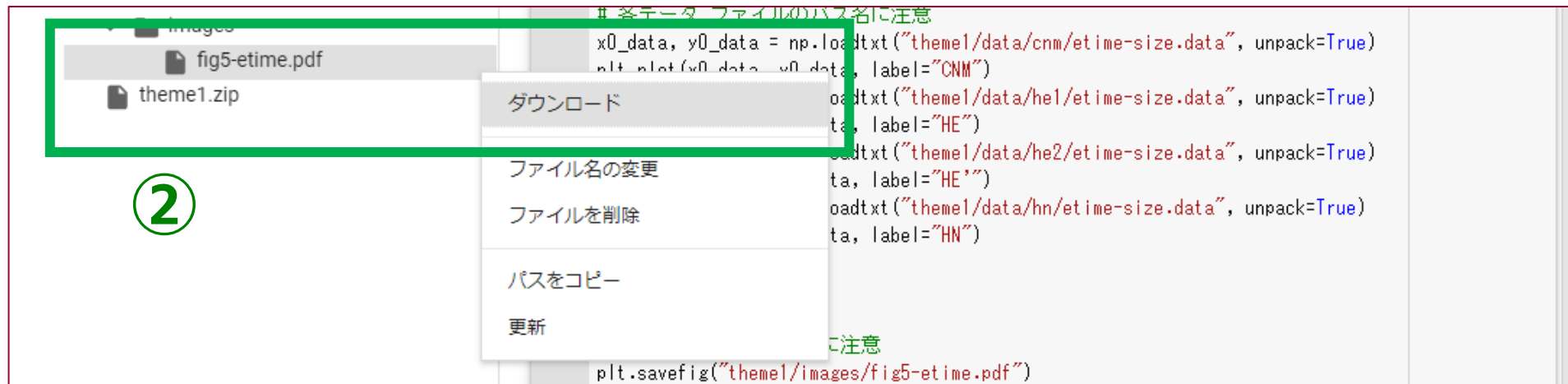
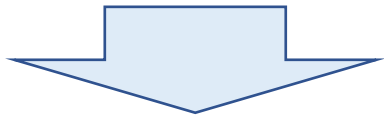


“矢印が書かれたフォルダ” のアイコンをクリックし、左下図のルート フォルダ “/” が表示されてしまった場合は、Colab に再ログインすると元に戻ります。Colab 内では、content フォルダ以下にファイルが保存されます。全体の位置関係を、確認しておきましょう。

ファイルが保存される content フォルダ

ルートフォルダ (名称は “/”)

データ ファイルのダウンロード



困った時の対応

エラー メッセージを確認しよう (1)

```
import numpy as np
import matplotlib.pyplot as plt

x0_data, y0_data = np.loadtxt("theme1/data/cnm/time-join-500K.data", unpack=True)
plt.plot(x0_data, y0_data, label="#Nodes = 500K", width=10000)

plt.legend()
plt.show()
```

例えば、上のようなプログラムを作成したとします。各コードの意味は、講義が進むにつれ説明をしますが、取り敢えず上のプログラムでは、`plt.bar()` 関数で棒グラフを表示させようとした、と考えて下さい。

しかし、間違えて線グラフを表示させる `plt.plot()` 関数を使ってしまいました。実際に動かしてみましよう。

エラー メッセージを確認しよう (2)

```
import numpy as np
import matplotlib.pyplot as plt

x0_data, y0_data = np.loadtxt("theme1/data/cnm/time-join-500K.data",
plt.plot(x0_data, y0_data, label="#Nodes = 500K", width=10000)

plt.legend()
plt.show()
```

作成した Matplotlib プログラムに誤りがある場合、colab は誤りの部分とその理由を教えてください。この例では、線グラフに存在しない“width”を設定している部分が誤りだと表示されています。これより、plot.plot() を使ったことが誤りだと気付きます。

```
-----
AttributeError                                Traceback (most recent call first)
<ipython-input-29-01029a0a6ebf> in <module>()
      3
      4 x0_data, y0_data = np.loadtxt("theme1/data/cnm/time-join-500K.data", unpack=True)
----> 5 plt.plot(x0_data, y0_data, label="#Nodes = 500K", width=10000)
      6
      7 plt.legend()

-----
          9 frames
/usr/local/lib/python3.7/dist-packages/matplotlib/artist.py in _update_property(self, k, v)
    1000         if not callable(func):
    1001             raise AttributeError('{!r} object has no property {!r}'
-> 1002                                     .format(type(self).__name__, k))
    1003         return func(v)
    1004
```

```
AttributeError: 'Line2D' object has no property 'width'
```