version 2022/10/08

PG01: データファイルへのアクセス

扱うデータファイルは csv 形式(comma-separated values)に限ることとする。 データをコンマで区切ったテキストファイルであり、単純な構造なので広く普及している。 表計算ソフト Excel でも扱うことができる。

1. Jupyter Notebook の場合

データはローカルに確保して作業することとする。 したがって、ウェブサイトにあるデータは（授業で扱うものは小規模なので）とりあえず、 ダウンロードして、自分のPCの内蔵ディスクや外付け記憶媒体（SSD、USB、DVD等）に保存して利用する。

データファイルがどこに保存されているか、「ファイルのパス」を確認する。 自分のPCの内蔵ディスク、またはPCにつないだ外付けの記憶媒体には、 通常、C,D,E,F,・・・のようなアルファベットで区別されるドライブ名が付けられる。 各ドライブにはフォルダーが階層的に作られていて、フォルダーを掘り下げることで 目的のファイル（例として StatData01_1.csv）に達するファイルの絶対パスが得られる。ふつうは、

例）C:\Users\Aoba Taro\Documents\2022_MathStatData\StatData01_1.csv

例）C:/Users/Aoba Taro/Documents/2022_MathStatData/StatData01_1.csv

のように記述される。

なお、区切り文字は使用環境（OSなど）によって /（スラッシュ）、\（バックスラッシュ）、￥（半角の円記号、日本語キーボードではバックスラッシュの代わりに使われる）などが使われる。

以下、Python のコードにおいて区切り文字は /（スラッシュ）を用いることを推奨する。

三種の神器

基本的な統計的データ処理を行うために3つのライブラリ（三種の神器）をあらかじめ読み込んでおく。

NumPy: 数値計算、特に行列計算

pandas: 表データ処理

Matplotlib: 描画

In [1]:

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

特に出力はないがそれでよい。

CSVファイルの読み込み (StatData01_1.csv)

では、自分のPCまたは外付けドライブに保存されているデータファイル（StatData01_1.csv）を読み出す。ここでは、ファイルのパスが

D:/2022_数理統計学/StatData/StatData01_1.csv

となっているものとする。

作業する上で、読み出したデータファイルに適当な名前をつける。 ここでは Data という名前を付けて読み出しており、今後、この名前で参照する。

In [2]:

```
Data = pd.read_csv('F:/2022_数理統計学概論/StatData/StatData01_1.csv')        # 読み出したいファイルの
```

特に出力はないがそれでよい。次のセルで Data の中身を確認する。

なお、コマンドセルの中で、# 以降の文（の一部）は Python によって実行されないコメントである。

読み込んだデータの確認

In [3]:

```
Data
```

Out[3]:

	新生児の体重(g)
0	3110
1	3100
2	3140
3	3050
4	2480
...	...
95	3170
96	2460
97	3360
98	3150
99	4180

100 rows × 1 columns

この形式を DataFrame という。データには自動的に、0番から番号がつく。 最後の表示を見ると、データは100行1列（レコード数100、カラム数1）からなることが分かる。

このようにデータの行数（レコード数）が60行を超すと、最初と最後の5行のみを表示して途中は省略される。

【注意】パスの表示について

ファイルのパスを取得するとき、ウィンドウに表示されているパスをコピペすると

D:\2022_数理統計学\StatData\StatData01_1.csv

のような形式で得られることがある。つまり、フォルダーの区切りがスラッシュ/ ではなく、バックスラッシュ \ になっている。ただし、日本語キーボードであれば、バックスラッシュ \ は半角 ¥ 記号になる。

これをそのまま、pd.read_csv() 関数に渡してもエラーが出るだろう. それは、\ が Python のコードにおいて特殊な役割（エスケープシーケンス）を持つからであり、

Data = pd.read_csv('D:\\2022_数理統計学\\StatData\\StatData01_1.csv')

バックスラッシュを（空白を入れずに）2つ連続させることによって避けられる。

日本語キーボードであれば半角 ¥ 記号（バックスラッシュの代用）を空白を入れずに2個連続させる。

2. Google Colaboratory の場合

Google Colaboratory は Jupyter Notebook の機能をクラウドで提供するサービスであり、自分のPCは端末としての機能しかなく、すべての処理がクラウドで実行されるため、ローカルにあるファイルへアクセスできない（ちょっと不便）。 Google account とともに My Drive （マイドライブ）という名の自分専用のドライブ（フォルダーのこと）が与えられていて、このドライブの中のデータファイルにアクセスすることになる。

また、Google Colab では連続利用に関して制限がある（2022年5月現在）。

【90分ルール】起動してから、PCがスリープ状態になる、またはキーボードやマウスに触れずに放置する、という状態が90分続くと自動的に初期画面に戻る。

【12時間ルール】たとえ使用中であったとしても、12時間経過すると強制終了する。

CSV ファイルのアップロード

Google Colab で使いたい CSV ファイルは My Drive （マイドライブ）にアップロードする。 アップロードはドラッグ＆ドロップ、あるいはファイルの場所を指定すればよいだけで簡単なはず。 わからなければ、ウェッブにある解説記事などを参照されたい。

なお、授業で使用する CSV ファイルは Google Classroom で配布するので、そこからダウンロードする。

大量のファイル（この授業に限らず、Google account の下に作ったすべてのファイル）が My Drive に格納される ことになるので、適当なサブドライブを作って、ファイルの保存場所が分かるようにするのがよい。たとえば、Course というサブドライブを作ってその中に、StatData01_1.csv が保存されていれば、ファイルのパスは次のようになる。

例）My Drive/Course/StatData01_1.csv

Google drive のマウント

Google Colaboratory でファイルにアクセスするためには、まず「google driveをマウント」することが必要。次の2行を実行すると、ドライブへのアクセス許可を求められるので許可する。 通常、セッションの初めに1回やれば、ネットにつながっている限り有効なはず。

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

こうして、データファイルの読み出しの準備ができた。 Jupyter Notebook と同様のコマンドで読み出すことができる。 ここでは Data という名前を付けて読み出している。

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

```
Data = pd.read_csv('/content/drive/My Drive/Course/StatData01_1.csv')
```

読み込んだデータの確認

このあとは、Jupyter Notebook と共通である。 次のセルに Data と打ち込んで実行して、DataFrame が表示されることを確かめればよい。