数理・データサイエンス教育強化拠点コンソーシアム
関東・首都圏ブロックワークショップ
2021年6月24日

模擬講義2
「データエンジアリング基礎」

森 純一郎

数理・情報教育研究センター
東京大学

自己紹介

- 専門
  - 人工知能、データ解析
- 数理・情報教育研究センター　准教授
- 数理・データサイエンス教育強化拠点コンソーシアム 教育用データベース分科会委員
- リテラシーレベル教材の担当
  - 4-2 アルゴリズム基礎、4-3 データ構造とプログラミング、4-5 テキスト解析、4-7 データハンドリング
- 応用基礎レベル教材の担当
  - 2-3データ収集の一部、3-1人工知能の歴史、3-7言語・知識
- 数理・情報教育研究センターでの主担当科目
  - Pythonプログラミング入門（全学対象）
  - データマイニング入門（1,2年生対象）
  - データマイニング概論（3,4年生、大学院生対象）

数理・データサイエンス教育プログラム

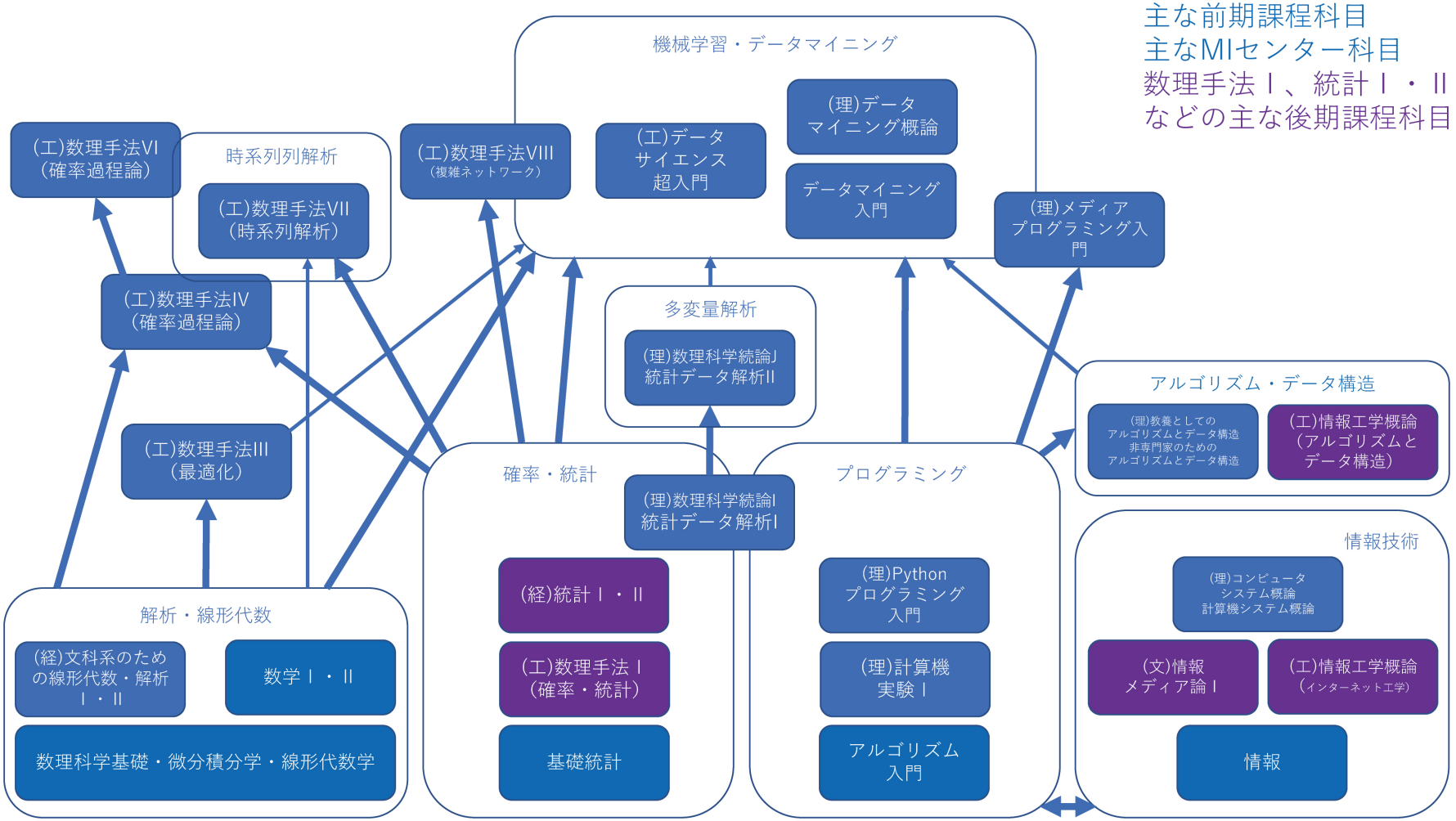
http://www.mi.u-tokyo.ac.jp/mds-oudan/index.html

- 数理・データサイエンス分野の166科目から構成される（2021年度）
  - http://www.mi.u-tokyo.ac.jp/mds-oudan/Data_Science_20210430.pdf
- 数理・情報教育研究センター教員の担当科目は以下

開講部局	授業科目	単位数	開講学期	詳細情報
工学部	数理手法IV（確率論）	2	S1S2	授業カタログ
工学部	数理手法VI（確率過程論）	2	A1A2	授業カタログ
工学部	数理手法III（最適化手法）	2	A1A2	授業カタログ
工学部	数理手法VII（時系列解析）	2	S1S2	授業カタログ
理学部	数理科学続論I（統計データ解析I）	2	S1S2	授業カタログ
理学部	数理科学続論J（統計データ解析II）	2	A1A2	授業カタログ
理学部	確率統計学基礎	2	S1S2	授業カタログ
理学部	Pythonプログラミング入門	1	S1A1S2	授業HP 授業カタログ 授業カタログ
理学部	データマイニング概論	2	A1A2	授業カタログ
理学部	コンピュータシステム概論	1	S1	授業カタログ
理学部	計算機実験I	1	S1S2	授業カタログ
理学部	計算機実験II	1	A1A2	授業カタログ
工学部	数理手法VIII	2	A1A2	授業カタログ
教養学部	特殊講義「社会科学のための統計分析」	2	S1S2	授業カタログ
経済学部	文科系のための線形代数・解析 I	2	S1	授業カタログ
経済学部	文科系のための線形代数・解析 II	2	S2	授業カタログ
理学部	教養としてのアルゴリズムとデータ構造	1	S2	授業カタログ
理学部	メディアプログラミング入門	1	S2A2	授業カタログ 授業カタログ
工学部	データサイエンス超入門	1	S1A1	授業カタログ 授業カタログ

データサイエンス履修の手引き

http://www.mi.u-tokyo.ac.jp/mds-guide/index.html



担当科目の例1

- Pythonプログラミング入門
  - 全学対象（1,2年生、大学院生は聴講）
  - S1ターム（4月~6月）とA1ターム（9月~11月）に開講、全7回
  - 300名程度の履修者
  - 6-7名の教員と複数のTAで運営
  - 反転授業
    - 学生は事前に教材を自学し授業当日は演習課題に取り組む
    - 授業では課題の解説および学生からの個別質問に対応
  - 教材は一般公開
    - https://utokyo-ipp.github.io/
    - オンラインプログラミング環境のGoogle Colaboratoryで実行可能
  - 独自の課題配布・提出・自動採点システムを導入
    - 学生はコードの動作を複数のテストケースで確認可能
    - 教員は特に誤答に対してフィードバックを行う

Pythonプログラミング入門 教材

https://utokyo-ipp.github.io/

1-1. 数値演算

数値演算について説明します。

参考

- https://docs.python.org/ja/3/tutorial/introduction.html#using-python-as-a-calculator
- https://docs.python.org/ja/3/tutorial/modules.html
- https://docs.python.org/ja/3/library/numeric.html

簡単な算術計算 ¶

CodeセルにPythonの式を入力して、プレイボタンを押すか、 Shiftを押しながらEnterを押すと、式が評価され、その結果の値がセルの下に挿入されます。

1+1 の計算をしてみましょう。次のセルに `1+1` と入力して、 Shiftを押しながらEnterを押してください。

```
[ ]:
```

このようにして、電卓の代わりにPythonを使うことができます。`+` は言うまでもなく**足し算**を表しています。

```
[1]: 7-2
[1]: 5

[2]: 7*2
[2]: 14

[3]: 7**2
[3]: 49
```

`-` は**引き算**、`*` は**掛け算**、`**` は**べき乗**を表しています。

式を適当に書き換えてから、Shiftを押しながらEnterを押すと、 書き換えた後の式が評価されて、セルの下の値はその結果で置き換わります。 たとえば、上の `2` を `100` に書き換えて、7の100乗を求めてみてください。

割り算はどうなるでしょうか。

```
[4]: 7/2
[4]: 3.5

[5]: 7//2
```

担当科目の例2

- データマイニング概論
  - 3,4年生、大学院対象
    - 1,2年生にはSセメスターに「データマイニング入門」として開講
  - Aセメスター（9月~1月）に開講、全14回
  - 200名程度の履修者
  - 基本的に1名の教員で運営
  - 座学（前半）+演習課題（後半）
    - 前半（45分）
      - スライドによる授業
    - 後半（45分）
      - Google Colaboratory (Colab)でプログラミング演習
        - Pythonを使用
      - プログラミング課題
  - 質問対応
    - zoomのチャット、メール、LMS（Classroom）でのコメント

      * 学生によって質問の障壁に差がある

データマイニング概論のシラバス

https://catalog.he.u-tokyo.ac.jp/detail?code=0590105&year=2021

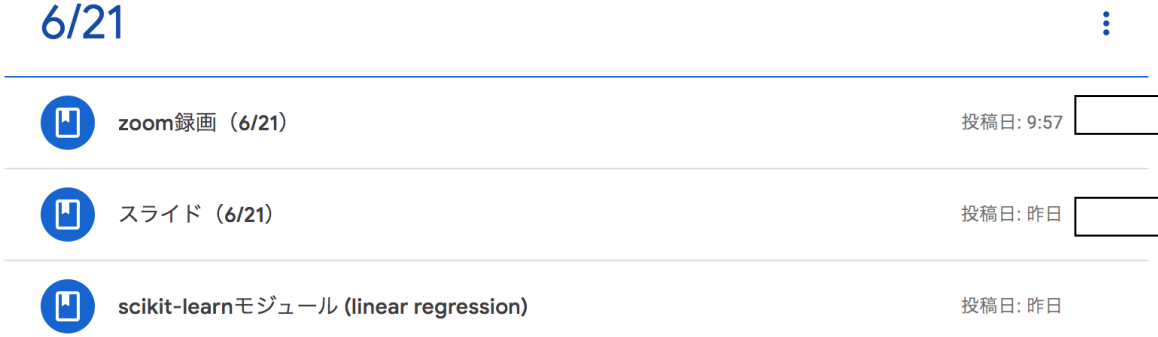
1. ガイダンス、データ分析のためのプログラミング基礎1: Pythonの基礎
2. データ分析のためのプログラミング基礎2: Numpy, Scipy, Pandas, Matplotlibなどのモジュール
3. データの記述統計・前処理: 記述統計、分布、最尤法、欠損値・外れ値の処理など
4. テキストデータ分析: tfidf、ベクトル空間モデル、形態素解析、類似度、潜在意味解析など
5. ネットワーク分析: 隣接行列, 最短距離, 中心性、コミュニティ抽出、ネットワークの数理モデルなど
6. 機械学習の基礎（教師なし学習）: k-means、階層化クラスタリング
7. 機械学習の基礎（教師なし学習）: EMアルゴリズム、主成分分析
8. 機械学習の基礎（教師あり学習）: 線形回帰、ロジスティック回帰
9. 機械学習の基礎（教師あり学習）: 過学習と正則化、モデル評価と選択
10. データ分析の実践
11. 深層学習の基礎：多層パーセプトロン、誤差逆伝搬法、最適化など
12. 畳み込みニューラルネットワークと画像処理応用
13. 再帰的ニューラルネットワークと言語処理応用

データマイニング概論の進め方（オンライン授業）

- zoomによるオンライン授業
  - zoomは毎回録画し公開。学生が後から見直せるようにする
- 教材（スライドと演習教材）はLMS経由で配布
  - LMSはGoogle Classroomを使用
  （Google Workspace（旧 G Suite）が全学で利用可能）
- 座学部分
  - 学生へはGoogleスライドを共有
    - 学生は任意のファイル形式でスライドを保存可能
  - 授業ではタブレット（iPad）のアプリケーション（GoodNotes5）でスライドのpdfを映し、ペン（Apple Pencil）で書き込みながら説明
- 演習部分
  - 演習教材（主にJupyter Notebook）はGitHubで管理
  - ColabはGitHubを参照するようにする
  - 学生へはColabのリンクを共有
  - 学生は同リンクを開き各自のDriveへ保存、演習に取り組む
  - 演習中は教員のColabを映し、実行しながら説明
  - 演習課題はColabファイルをClassroomから提出
  - Classroom経由で提出物に対するフィードバックを行う

教材配布

Colabへのリンク
（ColabはGitHub上の
ファイルを参照している）



zoom録画へのリンク

Googleスライドへのリンク

座学部分の進め方

重要な点は部分記述・空欄にしておき、書き込みながら説明

特にオンライン授業ではスライドの読み上げだけでは学生の集中力が続かない

16:13 6月22日(火)

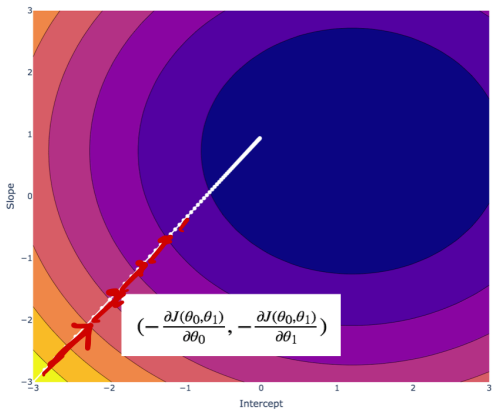
0621

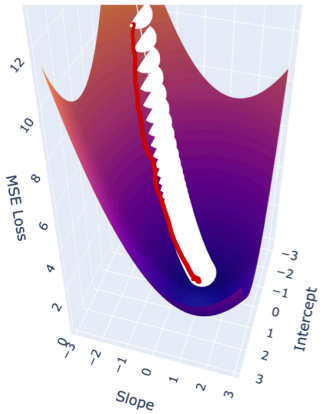
最急降下法

- 勾配ベクトルの定義より、$-\nabla J(\theta_0, \theta_1)$ の方向に進むようにパラメータを更新すれば $J(\theta_0, \theta_1)$ を減少させることができる
- この時、どれぐらい勾配ベクトルの方向へ進むようかを定めるハイパーパラメータを α（学習率）として，パラメータ θ は任意の初期値から開始して $J(\theta_0, \theta_1)$ を減少させる方向に以下の様に更新できる

$$\theta_0 := \theta_0 - \alpha \frac{\partial J(\theta_0,\theta_1)}{\partial \theta_0} = \theta_0 - \alpha \Sigma_{i=1}^{m} (h(x^{(i)}) - y^{(i)})/m$$

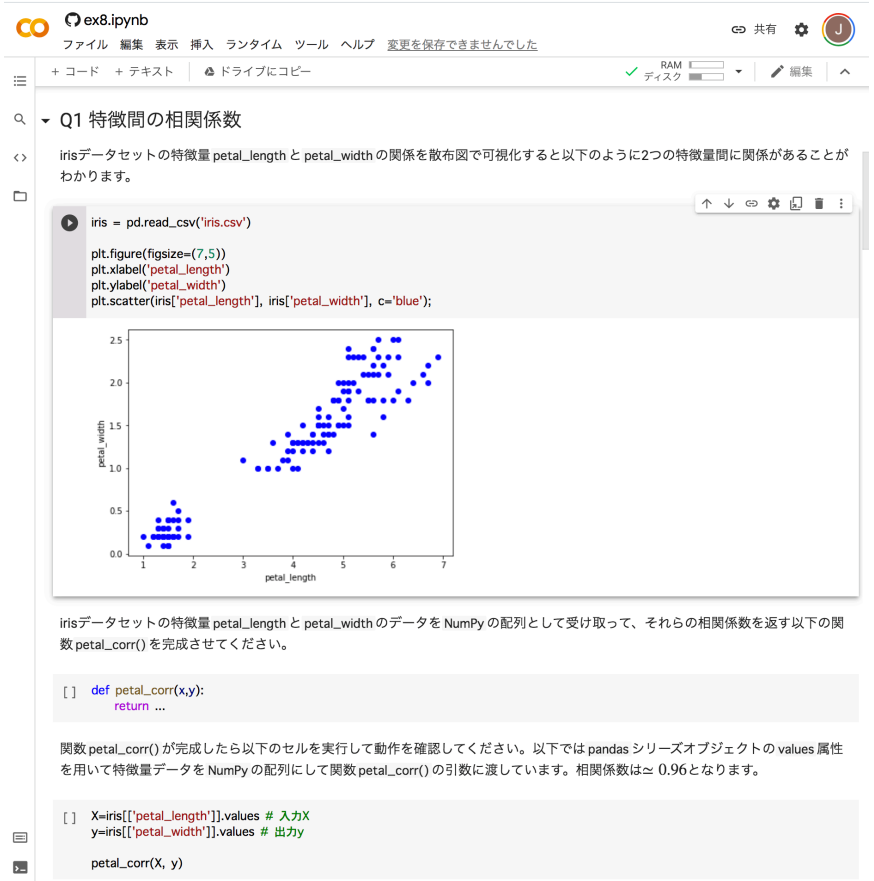
$$\theta_1 := \theta_1 - \alpha \frac{\partial J(\theta_0,\theta_1)}{\partial \theta_1} = \theta_1 - \alpha \Sigma_{i=1}^{m} ((h(x^{(i)}) - y^{(i)}) x^{(i)})/m$$





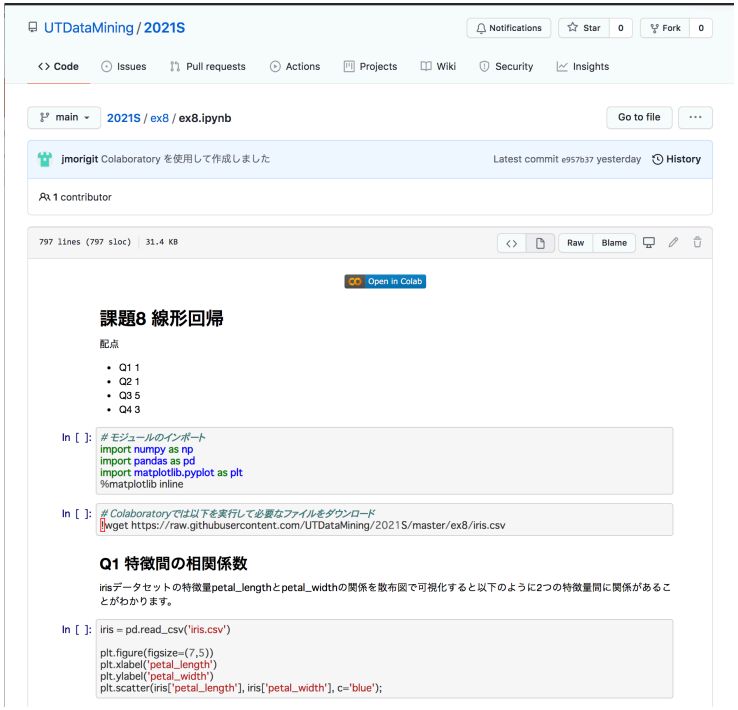
演習部分の進め方

配布するColabファイル



参照

演習教材を管理するGitHubのレポジトリ



学生はColabファイルを各自のDriveに保存して実行
教員はColabファイルを映し、実行しながら説明

演習の進め方

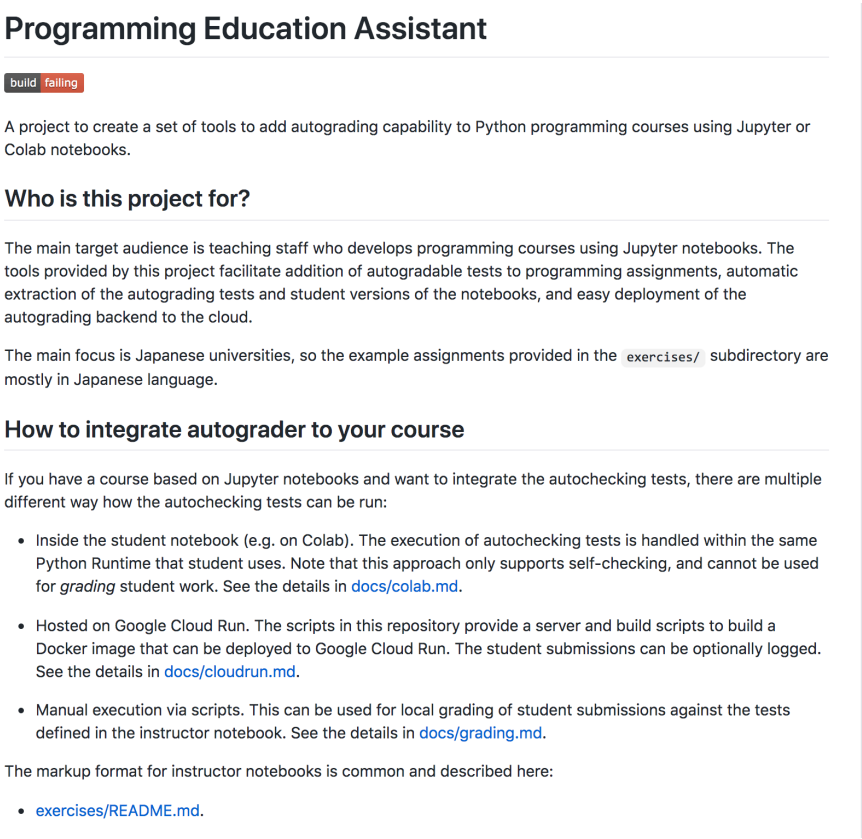
学生は作成したコードを提出前にテスト可能
（テストケースはColabファイルに埋め込まれている）

教員はテストケースをパスできなかった誤答に対する
フィードバックに注力できる

演習の進め方

同テスト機能はGoogle社との連携により開発され、
オープンソースとして以下に公開されている

https://github.com/google/prog-edu-assistant/



Programming Education Assistant

build failing

A project to create a set of tools to add autograding capability to Python programming courses using Jupyter or Colab notebooks.

Who is this project for?

The main target audience is teaching staff who develops programming courses using Jupyter notebooks. The tools provided by this project facilitate addition of autogradable tests to programming assignments, automatic extraction of the autograding tests and student versions of the notebooks, and easy deployment of the autograding backend to the cloud.

The main focus is Japanese universities, so the example assignments provided in the `exercises/` subdirectory are mostly in Japanese language.

How to integrate autograder to your course

If you have a course based on Jupyter notebooks and want to integrate the autochecking tests, there are multiple different way how the autochecking tests can be run:

- Inside the student notebook (e.g. on Colab). The execution of autochecking tests is handled within the same Python Runtime that student uses. Note that this approach only supports self-checking, and cannot be used for *grading* student work. See the details in docs/colab.md.
- Hosted on Google Cloud Run. The scripts in this repository provide a server and build scripts to build a Docker image that can be deployed to Google Cloud Run. The student submissions can be optionally logged. See the details in docs/cloudrun.md.
- Manual execution via scripts. This can be used for local grading of student submissions against the tests defined in the instructor notebook. See the details in docs/grading.md.

The markup format for instructor notebooks is common and described here:

- exercises/README.md.

演習の進め方

課題の提出

課題の配布、実行、提出、すべてブラウザ上で可能

演習の進め方

フィードバック

課題8

[redacted] 提出済み

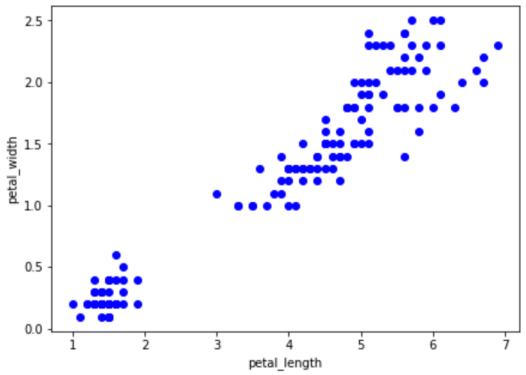
返却

＋ コード ＋ テキスト 最終保存: 6月21日 接続

Q1 特徴間の相関係数

irisデータセットの特徴量 petal_length と petal_width の関係を散布図で可視化すると以下のように2つの特徴量間に関係があることがわかります。

```
[ ]  1 iris = pd.read_csv('iris.csv')
     2
     3 plt.figure(figsize=(7,5))
     4 plt.xlabel('petal_length')
     5 plt.ylabel('petal_width')
     6 plt.scatter(iris['petal_length'], iris['petal_width'], c='blue');
```



irisデータセットの特徴量 petal_length と petal_width のデータを NumPy の配列として受け取って、それらの相関係数を返す以下の関数 petal_corr() を完成させてください。

ファイル

提出日時: 6月21日 20:06

履歴を表示

課題8.ipynb

成績

/10

限定公開のコメント

限定公開コメントを追加...

キャンセル 投稿

以降、応用基礎教材 2-3「データ収集」より抜粋

ウェブからのデータ収集における留意点

- データの信ぴょう性
  - ウェブに公開されているデータを収集する際には，元データの公開元，元データ自体の収集方法や内容，などの検証を十分に行い，データの信ぴょう性を確認する必要があります.

- バイアス
  - 収集したデータにはバイアス（偏り）が含まれる可能性があることに留意する必要があります.
    - 選択バイアス：データを集める際に観測したものと観測しなかったものの間の性質の差によって生じるバイアス
    - 情報バイアス：観測者の先入観や観測対象の過少申告や過剰反応によって生じるバイアス

- 個人情報の扱い
  - データを収集する際には，個人情報の扱いに十分に留意する必要があります．収集したデータが個人情報を含む場合は，あらかじめ利用目的を公表しておくか，または取得後速やかに利用目的を本人に知らせなければいけません.

ウェブクローラ

- 一般にウェブブラウザは以下のような流れでウェブページを取得しています.
  - ウェブブラウザのようなクライアントはウェブサーバへhttpリクエストをおくる
    - この時，http://から始まるURL (universal resource locator) を指定する
  - ウェブサーバはこのリクエストに答えてURLで指定されたコンテンツ（HTMLで記述されたファイル）を返す

- ウェブクローラは，ウェブブラウザが行うようなウェブサーバへのコンテンツのリクエストと受信をウェブのリンクを辿りながら逐次的に行うことで自動的にウェブのコンテンツを取得し蓄積するプログラムです.
  - ウェブクローラはボットやスパイダーとも呼ばれます.

- 検索エンジンではウェブクローラを用いて膨大な数のウェブページを自動で収集し索引付け（インデキシング）を行い管理しています.

ウェブスクレイピング

- ウェブページ（一般にはHTMLで記述されたファイル）から情報を抽出することをウェブスクレイピングと呼びます.
- ウェブクローラなどで自動で収集したウェブページからウェブスクレイピングにより情報の抽出を行うことで，ウェブから自動的にデータを収集することができます.
- ウェブスクレイピングでは非構造的なウェブページから情報を抽出し，データ分析に利用可能な構造的なデータに整理します.
- ウェブクローリング・スクレイピングを行う際はウェブサーバに負荷がかからないように十分に注意する必要があります．また，サイトによってはウェブクローリング・スクレイピングを禁止している（代わりにWeb APIの利用を求める）こともあるため，事前にサイトの規約をよく確認しておく必要があります.

2-3.ipynb ☆

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

コメント 共有 J

＋ コード ＋ テキスト 接続 編集

ウェブクローリング

```
1 # モジュールのインポート
2 import urllib
3 from bs4 import BeautifulSoup
```

指定したURLのウェブページ（HTMLファイル）を取得する

```
[ ] 1 url = 'https://www.u-tokyo.ac.jp/en/'
    2 response = urllib.request.urlopen(url)
    3 print(response.getcode()) # HTTPステータスコードの表示
    4 print(response.info()) # HTTPヘッダーの表示
    5 print(response.read()) # HTMLコンテンツの表示
    6 response.close()
```

```
200
Date: Wed, 23 Jun 2021 06:08:39 GMT
Server: Apache
Accept-Ranges: bytes
Access-Control-Allow-Origin: http://cdn.pr.u-tokyo.ac.jp
Access-Control-Request-Headers: *
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html


b'<!DOCTYPE html>\r\n<html lang="en">\r\n    <head>\r\n        <meta charset="UTF-8">\r\
```

ウェブスクレイピング

Beautiful Soupは、HTMLおよびXMLドキュメントを解析するためのPythonパッケージです。以下ではBeautiful Soupを使ったウェブスクレイピングの例を示します。

```
1 url = 'https://www.u-tokyo.ac.jp/en/'
2 response = urllib.request.urlopen(url)
3 soup = BeautifulSoup(response)
4 response.close()
5 print(soup) # HTMLコンテンツの表示
```

取得したウェブページ（HTMLファイル）のtitleタグに記述されたテキストを抽出

```
[ ]  1 print(soup.title.text)
```

```
The University of Tokyo
```

課題

上記で取得したウェブページから更新情報（NOTICES, PAGE UPDATES）に掲載されているテキストをすべて抽出してください。

更新情報に掲載されているテキストは以下のタグで記述されています。

`<p class="p-top-updates__list-title">テキスト</p>`

BeatifulSoupでは以下のようにタグとクラスを指定して、そのタグに記述されたテキストを抽出できます。

```
for i in soup.find('タグ', class_='クラス'):
    print(i.text)
```

```
[ ]  1 ### your code
```