

BYOD を前提とした初学者向けの Python によるデータ分析環境の構築と実践

三島 啓雄^{*1}

^{*1} 目白大学 情報教育センター

Building and implementing a data analysis environment using Python for beginners in BYOD classes

Yoshio MISHIMA ^{*1}

^{*1} Informatics and Computer Education Center, Mejiro University

At universities and high schools that adopt bring your own device (BYOD) as a method of procuring educational PCs, students inevitably use PCs with varied OSs and performance. Furthermore, students vary in their knowledge of PCs and their motivation to learn data analysis at the time of enrollment. Considering this variation, in this presentation, we examine the Python execution environment necessary to provide students with beginner-level data analysis exercises using programming techniques. In addition, we also report on how to proceed with data analysis exercises using “Pyodide” and “JupyterLite”, which we selected after consideration.

キーワード: Python, JupyterLite, Pyodide, BYOD,

1. はじめに

2023 年度当初時点で全国 47 都道府県の公立高等学校のうち 22 都道府県において保護者負担で整備する BYOD 端末の使用が採用されている⁽¹⁾。群馬県においては 2024 年度からの BYOD 端末の導入を決定する⁽²⁾など、都道府県立の高等学校における BYOD の導入高校は増加傾向にある。またやや古い情報となるが、2018 年に発表された「BYOD を活用した教育改善に関する調査研究 結果報告書（大学 ICT 推進協議会）」によると、BYOD を全学で導入している大学は 36.1%（有効回答数 490 機関中 177 機関）⁽³⁾であると報告されている。学校が指定した端末を私費購入する方式である BYAD（Bring Your Assigned Device）も含め、大学においても学生が所有する個人の PC 端末の利用を前提として教育を行う機関は増加傾向にある。

2022 年 4 月から、高等学校においても新しい学習指導要領がスタートし、共通必修科目「情報Ⅰ」が新設され、全ての生徒がプログラミングやネットワー

ク、データベースの基礎等について学習することになった。これに伴い高等学校ではプログラミングの実習が開始され、2022 年度の調査によると高等学校の情報Ⅰ科目におけるプログラミング単元で採用されたコンピュータ言語のなかで Python が 69.8%を占めていた⁽⁴⁾ことが明らかになった。

大学においては、2021 年度より、数理・データサイエンス・AI に関する知識及び技術について体系的な教育を行うものを文部科学大臣が認定する取り組みが始まり、大学・高専卒業者全員に対し、「学生の数理・データサイエンス・AI への関心を高め、適切に理解し活用する基礎的な能力を育成」する目標が示され、その基準として認定教育プログラム（リテラシーレベル）が設定された⁽⁵⁾。2023 年度における全国の大学数が 810 校、同じく高等専門学校が 58 校であるが⁽⁶⁾、2023 年 8 月時点での認定数は 382 件(校)である⁽⁷⁾。数理・データサイエンス・AI 教育強化拠点コンソーシアムによって公開されているリテラシーレベルのモデルカリキュラム対応教材⁽⁸⁾をみると、基礎的な記述統計やデ

ータの可視化の項においても、プログラミングの演習が組み込まれている。

このような状況下においては、大学および高校において、BYOD 端末での実施を前提とした、プログラミング技術を用いたデータ分析についての解析環境の整備に取り組む必要があるものと考えられる。

そこで本発表では、BYOD 環境下かつ、初学者向けであることを前提とした、プログラミング技術を用いたデータ分析環境の構築について検討した。その検討結果をもとにしたデータ分析演習の進め方についても併せて報告する。また本発表で対象とするプログラミング言語は、データサイエンスの分野で広く利用されている言語である「Python」とする。「Python」は、「R」言語と並び、データ分析を行ううえで汎用性が高いプログラミング言語であり、特に、機械学習や AI 開発の分野での利用が進んでいる。

2. 現状での課題

2.1 学生間におけるコンピュータリテラシのばらつきへの対応

スマートフォンネイティブ世代において、PC のファイルシステムの理解ができていないまま大学へ進学する学生が数多くみられることが報告されている⁽⁹⁾。執筆者は 2020 年以降より、大学初年次教育としての情報リテラシ系の実習を担当するようになったが、実習内においてもっとも多い質問が「自分のファイルが見つかりません」である。これは理工学系の学生においても見られることが指摘されている⁽¹⁰⁾。大学の初年次教育においては、データ分析についての実習は、PC についての基本的な理解を深める実習と並行して実施しなければならないケースが多い。そのため、実際のデータ分析の演習に入るまでの準備段階において、受講生が躓くことが多く、それを原因とした受講生の学習モチベーションの低下や、実習の進行の遅延が予想される。これはコンピュータリテラシが高い学生にとっても学習機会の損失につながり、モチベーションの低下になり得る。

2.2 BYOD 方式の採用に伴う学生間における PC 環境

のばらつきへの対応

BYOD を採用する大学や高等学校においては、OS や PC のスペックについての推奨基準の設定や、基準を満たす PC の推薦販売の実施などにより、学生間における所有端末の差異を平準化する取り組みが行われている。しかし特定の学部・学科における特定の OS の指定は散見されるものの、平準化を強制しない、または猶予期間を設定するケースも同様に見受けられる。そのため、ソフトウェアのインストールに必要な管理者権限が与えられていない状態で保護者や親族の PC を使用したり、アルバイトをして新しいものを購入するためのつなぎとして旧式の PC を用いたりしている学生は、とくに入学初年度において相当数存在するものと考えられる。これは金銭が絡む問題であるため、BYOD 導入における不可避の課題である。

BYOD をとりいれることにより、従来 PC 教室でのみ行われていた情報系の実習や学修が、自宅、研究室、キャンパス、野外といった様々な場所において柔軟に実施可能となることは、教育機関における最大の受益者である学生にとって多大なメリットがあると考えられる。よって前述したような課題を解決し、初年度の学修において躓かないような学習環境を設計することは重要である。

3. データ解析及びプログラミング環境整備に求められる要件

前述した状況から、大学又は高校の初年次教育において行うデータ解析及びプログラミング環境整備には、限られた時間やコマ数の範囲内で必要な演習要素を実施するために以下に示す要点を抑える必要があるものと考えられる。

要点 1：受講者のコンピュータリテラシへの依存の低減

- ・ファイルシステムを意識せずに気軽に操作できる
- ・ソフトウェアやクラウドサービスの使用方法について習熟を求めない

要点 2：受講者の PC 環境への依存の低減

- ・特定のソフトウェアや OS に依存しない

- ・ PC のスペックへの依存を極力避ける

上記の要点を抑えるためには次に示す条件を満たす学習環境が必要であると考える。

条件 1 : 安定したインターネット回線への接続が担保されていること

条件 2 : 出来得る限りブラウザベースでの解析環境整備を行うこと

これらの条件下のもと、要点を押さえた以下の 2 つの Python の実行環境を採用することとした。

- ・ **Pyodide** の対話実行環境
- ・ **JupyterLite** のノートブック環境

次章では上記 2 環境の特性を紹介する。さらに、要点 1 は押さえられていないものの、より発展的な解析環境として、ブラウザベースの Python の解析環境として広く利用されている Google Colaboratory についても併せて取り扱うこととする。

4. ブラウザベースの Python 実行環境

以下より採用した Python の実行環境の概要を示す。また当資料の閲覧者向けに、テスト可能な環境にアクセスできる URL をあわせて示すこととする。

4.1 Pyodide⁽¹¹⁾ :

Pyodide は、C 言語を使って書かれた Python 処理系である CPython が WebAssembly へ移植されたものである。Pyodide は PyPI で wheel が配布されている Python のパッケージであれば、利用可能であり、データ分析や統計解析において多用される numpy, pandas, scipy, matplotlib, scikit-learn のような Python パッケージも利用可能である。

実行環境の利用にアカウントの登録等は不要であり、公式サイトで公開されているコードを html ファイルとしてローカルフォルダに保存しブラウザで開くだけで Python が実行可能となる(Pyodide alternative-example exsample を参照のこと)。また対話実行環境

(RPEL: Read-Eval-Print Loop)環境も用意されている(Pyodide Console を参照のこと)。これについてもユーザ側は指定された URL にブラウザでアクセスするのみで Python の実行環境を利用できる(図 1)。なお、モダンなブラウザであればいずれも動作はするものの、Firefox および Chrome の利用が推奨されている。

Pyodide alternative-example :

<https://pyodide.org/en/stable/usage/quickstart.html#alternative-example>

Pyodide Console :

<https://pyodide.org/en/stable/console.html>

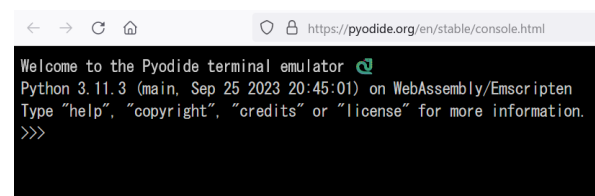


図 1 Pyodide の対話実行環境の初期状態

4.2 JupyterLite⁽¹²⁾ :

JupyterLite は、Python の学習やデータサイエンスの分野において広く利用されている JupyterLab および Jupyter Notebook と同じユーザーインターフェイス(以降 UI)が用意されている実行環境である。Python の実行を行っているコアは前述した Pyodide である。

JupyterLite の利用は、Pyodide と同様に、ソフトウェアのインストールやクラウドサービスのようなアカウント登録等は不要であり、ユーザ側は指定された URL にアクセスするだけで Jupyter ノートブック(.ipynb ファイル)を使った Python の実行環境を手に入れることができる。

コードの記述や解析結果の表示、マークダウンによるメモ書きが行われるノートブックファイルや、入力ファイルへのアクセスは、ブラウザの IndexedDB を利用して行われる。PC 上のローカルファイルへの出力処理として UI 上で「アップロード」「ダウンロード」と呼称される作業は必要であるが、実際にはデー

タが IndexedDB に登録される処理であるため、データはクラウドサービス等の外部サーバには配置されない。 .ipynb ファイルと .csv ファイル等のデータセットをファイルブラウザ部にドラッグ&ドロップし、 ipynb ファイルを JupyterLite 上で開くことで、ファイルパスを意識しないデータの読み書きが可能となる(図 2)。

Pyodide の節にて触れた NumPy, Scikit-learn, pandas, matplotlib 等のライブラリ/モジュールはインストール済みであり、seaborn や folium などの可視化/地図化ライブラリも pip を用いてインストールすることで使用可能である。公式サイトにもテスト・デモ用の実行環境が用意されている。さらに GitHub Pages 等の静的なホスティングサービス上に独自の JupyterLite のサイトを平易に構築することが可能であり、その手順は JupyterLite 公式サイトに明示されている⁽¹²⁾。執筆者の授業およびテスト用に GitHub Pages 上に構築したサイトの URL は以下のとおりである。なお、Pyodide と同様に JupyterLite についてもモダンなブラウザであれば使用可能である。

JupyterLite (執筆者のテスト環境) :

<https://pygeosham.github.io/practice/lab/index.html>

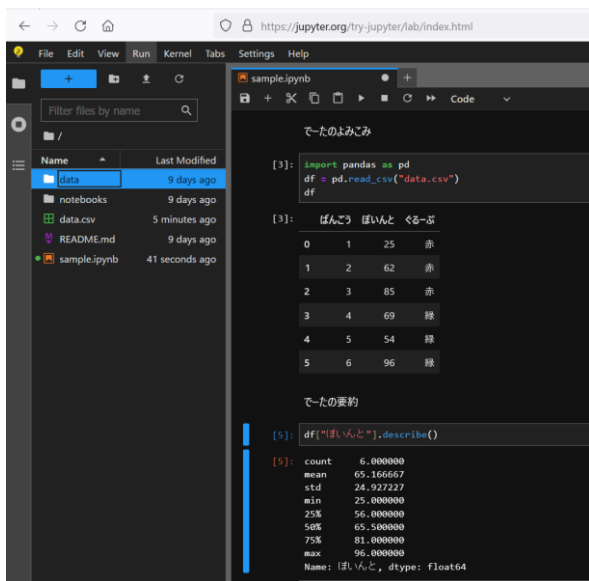


図 2 JupyterLite の画面構成(ダークモード)

図 2 補足 :

左ペイン : ファイルブラウザ

右ペイン : ノートブック(コード実行&出力部)

4.3 Google Colaboratory :

大学や高校での Python の実行環境として広くもちいられている Google Colaboratory は、データサイエンス系の実習や解析を行ううえでの各種の環境が整えられている。またライセンスにより使用時間等の制限があるものの GPU による計算資源も用意されているため、機械学習や深層学習のプラットフォームとして有用である。一方で、Google のアカウントを持ち、サインインを行うことが前提のサービスであり、ファイルの読み込みは Google Drive 等から行う必要があること、そのために Google Drive をマウントして仮想 PC のドライブとして認識させる処理が必要であること、Google Drive ドライブと Google Colaboratory 上におけるドライブのファイルの同期にやや時間差があることなど、解析を開始するまでの準備において、必要な処理の手数や知っておくべき知識が多い。それゆえ、クラウドサービスの利用経験やファイルシステムの理解がおぼつかない初学者にとっては解析を開始するまでの障壁がやや高いサービスとなっている。よって、大学における初年次教育の一環としての実習においては、PC 操作経験を積み、情報リテラシの向上が伴い、実習内容も進行した段階で、より複雑で重たい処理を実施する際に利用するのが適切なサービスと考える。

5. 実際の授業への適用

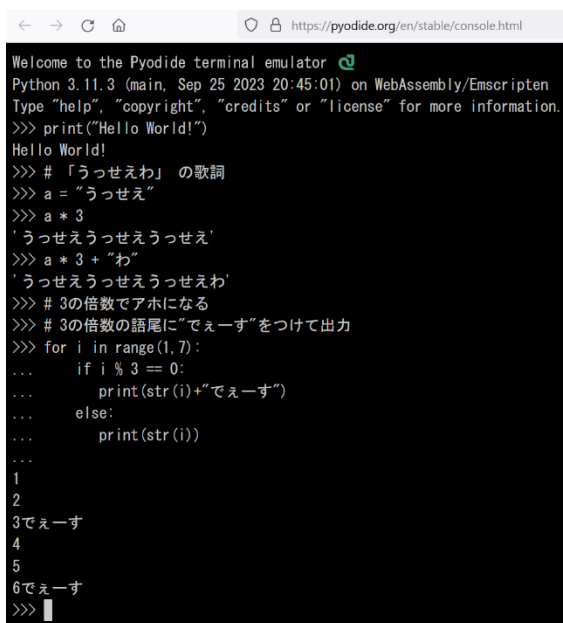
上記に示した Python の実行環境の特性をもとに、学修段階を 3 つの Phase にわけ、学修の進行状況にあわせて Python の実行環境を遷移しながらすすめるデータ分析の枠組みの素案を以下より示す。これは学修の順序を示すかたちをとっているものの、実習内容や解説すべき内容の特性に合わせて適宜実行環境を選択することができる。たとえば、学生の質問に対応し、ヒントや考え方を示す際には Pyodide のコンソールを起動しコードを記述して解説する、などがその具体例として挙げられる。

5.1 授業の進行に応じた使用環境の使い分け

5.1.1 Phase01 : Python の基礎を Pyodide の対話実

行環境で実施

Hello world ! の出力や、データ型の解説、繰り返し処理や条件分岐の解説など、基本的な Python の解説をする学習初期段階においては、受講学生のコンピュータリテラシもまだ発展途上であることが予想される。テキストやスライドを示しながらハンズオン形式で実施する実習になりがちな初期段階の学習においては、環境構築に時間を取られず、少ない手数で教員に追従して同じコードの実行ができる実行環境を用意する授業設計が必要である。Pyodide による Python の対話実行環境は、前述したように当該 URL をブラウザで開くだけで利用できる。そのため、実行コードがワンライナーであったり、多くとも数行で済むような Phase の学習段階に適していると判断する(図 3)。この場合教員側は、LMS 等において URL を示し学生にアクセスさせるだけでコードを書く演習が開始できる。



```
Welcome to the Pyodide terminal emulator
Python 3.11.3 (main, Sep 25 2023 20:45:01) on WebAssembly/Emscripten
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello World!")
Hello World!
>>> # 「うっせえわ」の歌詞
>>> a = "うっせえ"
>>> a * 3
'うっせえうっせえうっせえ'
>>> a * 3 + "わ"
'うっせえうっせえうっせえわ'
>>> # 3の倍数でアホになる
>>> # 3の倍数の語尾に"でえーす"をつけて出力
>>> for i in range(1,7):
...     if i % 3 == 0:
...         print(str(i)+"でえーす")
...     else:
...         print(str(i))
...
1
2
3でえーす
4
5
6でえーす
>>>
```

図 3 Hello World 出力,文字列のハンドリング, fizzbuzz 問題等を Pyodide の対話実行環境で実行

5.1.2 Phase02 : 図表の描画を伴うデータ解析は JupyterLite を用いて実施

Python の基本的な文法を学修した後は、データファイルの読み込み、前処理と呼ばれるデータの加工や

修正、データの要約や可視化といった処理を実施する Phase が訪れる。この段階においては、ノートブック (.ipynb) と呼ばれるドキュメントを作成し、プログラムの記述と実行、メモの作成、保存と共有などをブラウザ上で行うことができる、JupyterLite に移行するのが適切と考える(図 4)。データのハンドリング,データの前処理(整然データへの整形),記述統計,相関係数,回帰分析,推定統計といった範囲をカバーする実習には十分な解析環境といえる。

これまで、最大 40 人程度が前述した執筆者のテスト環境に同時にアクセスし、後述するヒートマップを作成する処理を実施した実績があるが、特にトラブルなく解析が実行できている。

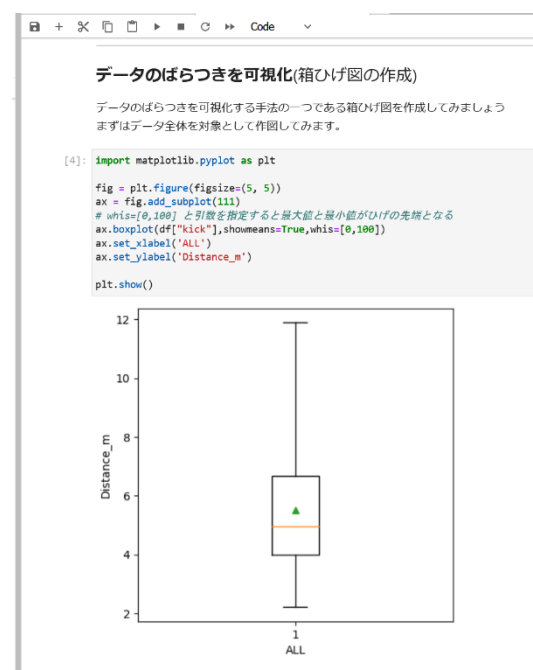


図 4 JupyterLite 上での作図例(ライトモード)

5.1.3 Phase03 : GPU を使用する処理は Google Colaboratory で実施

記述統計やデータのハンドリング、データおよび解析結果の可視化などの学修が進んだうで、機械学習や深層学習を用いたデータ解析や、3D モデルの生成などの GPU を使った処理を行わなければならない Phase においては Google Colaboratory に移行するのが適切と考える。前述したように Google Colaboratory におけるデータ解析は、ファイルの入出

力や事前準備が、著しく JupyterLite とは異なるものの、解析用のノートブックとして .ipynb ファイルが採用されており UI も似通っているため、解析段階にうつってからの JupyterLite と Google Colaboratory 間の操作方法の差異はさほど感じないことが予想される。

6. 異なる OS およびスペックにおいての JupyterLite のデータ処理テスト

異なる OS およびスペックにおいての JupyterLite の稼働試験として、執筆者が所有している PC のうち、スペックは最新のものではないが、最新の OS が搭載されている機材を選択し、JupyterLite の実行環境下において初学者向けデータ解析処理の中では比較的負荷の強い処理を実行させた。テスト機としては、Windows 機、MacOS 機、および Linux 機(Ubuntu)に加え、高等学校において採用数が多い PC 仕様として ChromeOS で稼働する Chromebook 機も対象とした。すべての端末においてブラウザは Chrome を用いた。データ処理前後の出力図面(図 5)、使用データ、処理内容およびテスト機一覧を以下に示す。

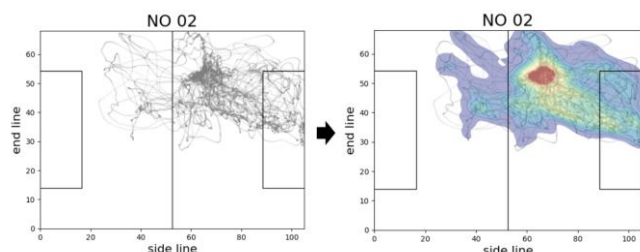


図 5 サッカー選手の移動ログデータをもとにしたヒートマップの作成

使用データ : Soccer video and player position dataset (ZXY data raw (20Hz))⁽¹³⁾ に執筆者が列名を追記して解析用に改変したもの

処理内容 : No 2 のプレイヤーの移動ログデータ(測点数:45865)をもとに seaborn の `seaborn.kdeplot()`関数を用いたカーネル密度推定を実施

テストに使用した機材一覧:

Windows 機:

製品名 :Surface Laptop3,CPU:Intel® Core™ i5-1035G7,メモリ:16GB,OS:Windows11Home

MacOS 機:

製品名:Mac mini(2018), CPU:3.6GHz Intel® Core™ i3,メモリ:8GB,OS : macOS Sonoma 14.2

Linux(Ubuntu)機:

製品名:ThinkPad Edge E130, CPU: Intel® Core™ i3-2367M,メモリ:4GB,OS:Ubuntu Desktop 22.04.3 LTS

ChromeOS 機:

製品名:Asus CX1102CK, CPU: Intel® Celeron N5100,メモリ:4GB,OS ChromeOS

基本的なスペックやデータ処理時のネットワークの環境が異なるため、処理時間に差は生じたものの、いずれのテスト機においても処理は最大 1 分程度の処理時間で実行された。このことからそれぞれ異なる OS やスペックを持つ端末が混在したデータ分析実習においても JupyterLite の有用性が確認された。

7. 改善すべき課題

実際にデータ分析を行った結果より、今後の教育現場での利用という視点から改善を試みておきたい点を 2 つあげておく。

7.1 作図キャプションのマルチバイト文字対応

matplotlib により作図した図上のキャプションにおいて日本語出力ができない問題については、通常の Python 環境下における対応(明示的なフォントの指定や `japanize-matplotlib` のインストール)が不可であったため、これについての対応策の検討は必要である。しかし、その他のデータ処理におけるマルチバイト文字の処理について、データのハンドリングと表示のいずれの点においても問題が発生しなかった。そのため、当面はキャプションや軸ラベル、タイトルについては英語表記とすることで対応は可能である。

7.2 バイナリデータの読み込みと書き込み

また、授業を実践していくうえで、CSV のようなテキスト形式によるファイルの出入力には問題が発生し

なかったものの、Microsoft Excel ブック (.xlsx) のようなバイナリ形式のファイルを複数回出力し上書きを繰り返したケースにおいて上書き処理が上手くいかずファイルが取り出せないケースがあった。これは、ファイル名を変更し新たにファイルを生成することで回避が可能であった。これについては現在調査中である。

8. おわりに

今年度実施したブラウザベースでの Python の実行環境の試行と実践により、安定したインターネット接続環境とキーボード付きのラップトップ PC またはタブレット PC の環境があれば、今回取り上げた Pyodide および JupyterLite を利用することで、Python によるデータ解析環境を提供できることが判明した。また、これらの環境下では、大学の初年度教育における基礎的なデータ解析におけるプログラミング演習についてはほぼ問題なく実行できることがわかった。この原稿執筆段階では、まだ学期の中盤であり、実習の進行途上ではあるものの、実習の進行にともない環境を使い分けることで、より高度な解析へとスムーズに移行するための目途がたった。

Pyodide ベースの Python 実行環境は、いくつかの問題や超えるべき課題は抱えているものの「電卓」がわりの気軽さで Python の実行環境があらゆる端末から利用できるメリットがある。これらの環境の利用が進むことは、プログラミングやデータ解析に対する初学者にとっての心理的障壁の低減のみならず、BYOD 端末を保有する大学生・高校生にとって、大学や高校以外での学習機会の確保にもつながる。今回の報告をきっかけとし、高校および大学における情報教育の現場での実践例を増やす働きかけを行い、事例や情報を共有できるコミュニティの形成を行っていきたい。

参 考 文 献

- (1) 文部科学省：高等学校における学習者用コンピュータの整備状況について（令和 5 年度当初）
https://www.mext.go.jp/content/20230711-mxt_shuukyo01-000020467_001.pdf (2023 年 12 月 14 日確認)
- (2) 群馬県庁：県立高等学校の BYOD 移行について
<https://www.pref.gunma.jp/page/615375.html> (2023 年 12 月 14 日確認)
- (3) 大学 ICT 推進協議会 (AXIES)：BYOD を活用した教育改善に関する調査研究結果報告書（第 1 版）
https://axies.jp/files/report/ict_survey/2016survey/byod_report_2016.pdf (2023 年 12 月 14 日確認)
- (4) 特定非営利活動法人みんなのコード：2022 年度プログラミング教育・高校「情報 I」実態調査報告書
<https://speakerdeck.com/codeforeveryone/2022nian-du-hurokuraminkujiao-yu-gao-xiao-qing-bao-i-shi-tai-diao-cha-bao-gao-shu> (2023 年 12 月 14 日確認)
- (5) 文部科学省：AI 戦略 2019 と数理・データサイエンス・AI 教育プログラム認定制度について
https://www.mext.go.jp/content/20230921-mxt_senmon01-000012801_6.pdf (2023 年 12 月 14 日確認)
- (6) 文部科学省：令和 5 年度学校基本調査（速報値）
https://www.mext.go.jp/content/20230823-mxt_chousa01-000031377_001.pdf (2023 年 12 月 14 日確認)
- (7) 文部科学省：「数理・データサイエンス・AI 教育プログラム認定制度」の概要／申請に当たってのポイント
https://www.mext.go.jp/content/20230920-mxt_senmon01-000012801_1.pdf (2023 年 12 月 14 日確認)
- (8) 数理・データサイエンス・AI 教育強化拠点コンソーシアム：リテラシーレベルモデルカリキュラム対応教材
<http://www.mi.u-tokyo.ac.jp/consortium/e-learning.html> (2023 年 12 月 14 日確認)
- (9) 川喜田多佳子，寺家尚美：スマホネイティブ世代に必要なパソコン教育について（その 2），キャリア研究センター紀要・年報（4），55-59（2018）
- (10) File not found - A generation that grew up with Google is forcing professors to rethink their lesson plans
<https://www.theverge.com/22684730/students-file-folder-directory-structure-education-gen-z> (2023 年 12 月 14 日確認)
- (11) Pyodide(公式ドキュメント)
<https://pyodide.org/en/stable/> (2023 年 12 月 14 日確認)
- (12) JupyterLite(公式ドキュメント)
<https://jupyterlite.readthedocs.io/en/stable/> (2023 年 12 月 14 日確認)

- (13) S. A. Pettersen, D. Johansen, H. Johansen, V. Berg-Johansen, V. R. Gaddam, A. Mortensen, R. Langseth, C. Griwodz, H. K. Stensland, and P. Halvorsen: “Soccer video and player position dataset”, Proceedings of the International Conference on Multimedia Systems (MMSys), Singapore (2014)
<https://datasets.simula.no/alfheim/> (2023 年 12 月 14 日 確認)