

本書で用いているバージョン

欧陽恵子

```
import sys
print("Python version:{}".format(sys.version))
```

```
Python version:3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)]
```

Pythonは多くのデータサイエンスアプリケーションの共同語となっている。

Python を使う大きな利点の一つは、ターミナルやJupyter Notebookなどのツールを利用することで、直接コードに触れることができる点にある。

Pythonは汎用言語なので、複雑なグラフィカルユーザインターフェイスやWebサービスを構築することもできるし、既存のシステムに埋め込むこともできる。

```
import pandas as pd
print("pandas version:{}".format(pd.__version__))
```

```
pandas version:0.22.0
```

pandasは、データを変換したり解析したりするためのライブラリだ。

このライブラリは、RのDataFrameを模して作られた、DataFrameというデータ構造を中心に構成されている。簡単に言うとPandasのDataFrameはテーブルのようなもので、エクセルのスプレッドシートに似ている。

```
import matplotlib
print("matplotlib version:{}".format(matplotlib.__version__))
```

```
matplotlib version:2.1.2
```

matplotlibは、最も広く使われているPythonの科学技術計算向けのグラフ描画ライブラリだ。

Jupyter Notebookの内部では、`%matplotlib notebook`や`%matplotlib inline`コマンドを用いると図をブラウザ上に直接表示することができる。

```
import numpy as np
print("NumPy version:{}".format(np.__version__))
```

```
NumPy version:1.14.0
```

NumPyは、Pythonで科学技術計算をする際の基本的なツールの一つである。
多次元配列機能や、線形代数やフーリエ変換、疑似乱数生成器等の、高レベルの数学関数が用意されている。
scikit-learnでは、Numpyの配列が基本的なデータ構造となる。

```
import scipy as sp
print("SciPy version:{}".format(sp.__version__))
```

```
SciPy version:1.0.0
```

SciPyは、Pythonで科学技術計算を行うための関数を集めたものである。

高度な線形代数ルーチンや、数学関数の最適化、信号処理、特殊な数学関数、統計分布などの機能を持つ。

scikit-learnは、アルゴリズムを実装する際にSciPyの関数群を利用している。

```
import IPython
print("IPython version:{}".format(IPython.__version__))
```

```
IPython version:6.2.1
```

IPython(アイパイソン)はPythonを対話的に実行するためのシェルである。

オリジナルの Python に比較して、型推定を強化し、対話的実行のための文法を追加してあり、コード・ハイライティングおよびタブによる補完が行える。

IPython は SciPy パッケージの一部として公開されている。

```
import sklearn
print("scikit-learn version:{}".format(sklearn.__version__))
```

```
scikit-learn version:0.19.1
```

scikit-learnは、二つのPythonパッケージに依存している。NumPyとSciPyである。scikit-learnは科学技術計算向けPythonライブラリである。NumPyとSciPyの上に構築されている。