訳者まえがき

本書の特徴

本書は 2016 年 11 月に出版された Osvaldo Martin 氏による *Bayesian Analysis with Python* の邦訳です。

日本ではベイズ統計分析に関する書籍がすでに数多く出版されています。ベイズの定理をわかりやすく解説する初学者向けの書籍を除き、本格的なベイズ統計分析を志向する書籍は、次の三つの特徴軸で位置づけることができるでしょう。

- 対象読者（理系向け ⇔ 文系向け）
- 説明内容（理論志向 ⇔ 実践志向）
- 計算ツール（R/Stan ⇔ Python/PyMC3）

これら三つの特徴軸に基づくと、原書は「文系向け」「実践志向」「Python/PyMC3」を特徴として位置づけることができます。

■ 文系向け

数学があまり得意ではない理系の人も含め、研究者だけでなく、大学生や大学院生、ビジネスマンなど幅広い読者を対象としています。また、これまで伝統的な頻度主義の統計学を利用してきたけれど、ベイズ統計学に転向したいと考えている人にも本書は向いています。翻訳にあたっては、多くの訳注をつけることで「文系向け」を強く意識しました。

■ 実践志向

本書では理論的な説明は少なくなっており、考え方、分析の仕方や計算方法を、コードを使って具体的に示し、「実践的に理解する」方式で説明しています。ですから、本書で使われているデータを読者が持っているデータと取り換えれば、所望のベイズ統計分析を行うことができるようになるでしょう。

■ Python/PyMC3

日本で最近出版されたベイズ統計分析の書籍の多くは、計算ツールとして R/Stan を利用しています。R は統計分析や科学技術計算に焦点を当てたコンピュータ言語であ

り、Stan は確率プログラミングのライブラリです。一方、本書では Python/PyMC3 を採用しています。Python は汎用のコンピュータ言語であり、併せて多数の便利で有用なライブラリが開発されています。PyMC3 はその一つで、確率プログラミングのライブラリです。近年、大きな注目を集めているディープラーニングの実装フレームとして TensorFlow、Chainer、Theano などがあり、これらは主に Python を通じて実行されます。Python はディープラーニング、さらに広くは機械学習をはじめとするコンピュータサイエンスの分野で評判が良く、そして、機械学習はマーケティングをはじめビジネス分野への幅広い応用が期待されています。Python の将来性はきわめて豊かであると言えます。また、Python は初学者が学習しやすいコンピュータ言語であるとの定評もあります。

Python とそのライブラリおよび PyMC3 のインストール

原著者は、Python とその科学計算ライブラリを導入するディストリビューションとして、Anaconda を推奨しています。Anaconda のインストールについて、「はじめに」に記載がありますが、ここでは、初学者向けに追記します。

訳者の計算環境（Windows 10、Linux/Ubuntu 6.04、MacBook Air: macOS X）では、Mac を除き、インストール後 Anaconda の最新版を動作させるために、システムに一つの環境変数を追加する必要がありました。Anaconda の最新版をインストールした後、もし読者の PC で本書第 1 章のコード 1.1（これは PyMC3 を使いません）がうまく動作しないなら、以下のように環境変数を設定してください。

Windows　プログラムやファイルの検索欄で「環境変数」という語を検索し、環境変数を設定するダイアログを開いてください。その後「新規」ボタンを押して

変数名：`MKL_THREADING_LAYER`
変数値：`GNU`

を入力し、OK ボタンを押して保存してください。

Linux　ターミナル（端末）を起動した後、コマンドプロンプトから次のように入力してエディタ vi を立ち上げてください（他のエディタでも構いません）。vi の使い方はインターネットで調べられます。

```
> vi .bashrc
```

すると、ターミナル起動時の実行ファイル.bashrc が開きますので、その最下行に次の 1 行を追加してください。

```
export MKL_THREADING_LAYER="GNU"
```

追加したら.bashrc を保存します。ターミナルを再起動すると Anaconda のインストールは完了です。

Mac　訳者の環境では、デフォルト設定で Anaconda をインストールした後、環境変数について何も変更しなくても動作しました。読者の環境で動作しない場合は、Linux の場合がヒントになるかもしれません。

Python の確率プログラミングのライブラリである PyMC3 は、本書の中核をなす重要なツールです。しかし、Anaconda に PyMC3 は含まれていないため、別途インストールする必要があります。その方法については「はじめに」で触れられていますが、Linux や Mac へのインストールに対応した内容ですので、Windows へのインストールを考慮して少しだけ追記します。

Windows　スタートメニューから Anaconda3 を探し、その中の Anaconda Prompt を選択するとターミナルが起動します。その中で、

```
> pip install pymc3
```

と入力・実行してください。

Linux・Mac　ターミナルを起動後、Windows の場合と同様に、コマンドプロンプトから上記のコマンドを入力・実行してください。

本書のコードについて

本書で使用しているすべてのコードとデータは、共立出版 HP にある本書のサイト

http://www.kyoritsu-pub.co.jp/bookdetail/9784320113374

からダウンロードできます。

また、原著者の Osvaldo Martin 氏の GitHub サイト

https://github.com/aloctavodia/BAP

にも、原書の全コード、データ、カラー版のグラフ、正誤表などがアップされています。

原著者の GitHub にアップされているコードは、章ごとにノートブック形式になっており、Web ブラウザで閲覧することができます。ノートブック形式は、IPython という対話型実行ツールの出力を保存したものです。コードのほかに出力されたカラーグラフも含まれていますので、ダウンロードしてそのまま実行することはできません。まずは、ノートブックファイルからコード部分をテキストエディタにコピー&ペーストし、ファイルを保存してから実行してください（インターネットを検索すると、もっと洗練された方法が見つかります）。

なお、本書で使用しているコードは、2018 年 4 月時点で原著者の GitHub にアップされていたコードに、以下の変更を加えてあります。

- グラフ出力時の色指定と文字の大きさを指定
- グラフ出力時の保存ファイル名を指定
- 関数 `sample` に `njobs=1` を追加

最後の「関数 `sample` に `njobs=1` を追加」について少しコメントしておきます。原著者の GitHub にアップされていたコードは、訳者の Linux 環境と Mac 環境では問題なく実行できましたが、Windows 環境では一部のコードが動作しませんでした。そこで、動作しないコードに含まれる関数 `sample` の引数に `njobs=1` を追加したところ、無事に動作しました。本書にはこれを追加したコードが載せてあります。`njobs` については 52 ページに説明があります。

コードの実行方法

本書のコードを実行する場合、コードを章単位でまとめて、BAP_Chapter1.py のようなファイル名で保存しておくとよいでしょう。というのも、各章の後半のコードはその章の前半のコードで書かれた変数や関数を使っているため、小さく分割したコードを単独で実行しても動作しないからです。これとともに、分析で使うデータが、あるフォルダ（以下、データフォルダと呼びます）に保存してあるものとします。例えば、読者の PC のホームディレクトリに Documents フォルダがあり、さらにその中に BAP というフォルダがあり、これが読者のデータフォルダだとしましょう。

コードを実行する方法は多様です。そのいくつかを示します。ここで紹介する方法はいずれもターミナルを起動し、そこに現れるコマンドラインから各種コマンドを入力・実行するスタイルのものです。コマンドラインとは、ターミナルの左側から、まず読者の PC のホームディレクトリ名、次に Windows なら “>”、Linux や Mac なら “$” といった記号、そして縦長の四角いカーソルが表示された、コンピュータがコマンドの入力を待っている場所を指します。

■ 対話型実行ツールによる方法

実際の分析では、コマンドを入力・実行してその結果を確認し、さらに別のコマンドを入力・実行してその結果を確認し…と、システムと対話をするように作業を進めていくことがよくあります。このように対話形式で Python を実行するツールとしては、IPython とその発展型の Jupyter QTConsole が有名です。IPython は 2001 年に最初のリリースが行われ、その後、機能を拡充させてきました。今日、IPython は Python だけでなくさまざまなコンピュータ言語に対応し、さらに Web ブラウザ上でのコード実

行など、さまざまな作業をすることができるようになり、Jupyter と呼ばれるようになりました。現在では IPython と Jupyter は互いに関連しつつ開発が進められています。原書では IPython を使ってコードを実行していますが、訳者は IPython と Jupyter の QTConsole を使ってコードの動作確認を行いました。

Windows の場合、スタートメニューの Anaconda3 の中に Python Prompt がありますので、これを選択・起動するとターミナル（端末）が開きます。一方、Linux や Mac の場合にはシステムに登録されているターミナルを起動してください。ターミナルが起動してカーソル（縦長の四角）が現れたら、その左側に現在の作業ディレクトリが表示されているはずです。コマンドラインのカーソルに続けて、次のように cd コマンドを使って作業ディレクトリを読者のデータフォルダに移動してください。

Windows	`> cd Documents¥BAP`
Linux・Mac	`$ cd Documents/BAP`

これは、現在の作業ディレクトリ（ホームディレクトリ）をその下にある Documents というフォルダの、さらにその中にある BAP というフォルダに移動させるコマンドです。Enter キーを押すと、カーソルのすぐ左側に BAP の文字列が表示されます。続いて、コマンドラインから次のように入力・実行することで、Jupyter QTConsole または IPython を起動します[*7]。

Jupyter QTConsole	`> jupyter qtconsole`
IPython	`> ipython`

これにより Jupyter QTConsole の場合は別ウィンドウで、IPython の場合はターミナル内でそれぞれの対話型実行ツールが起動します。そこに `In[1]:` というプロンプトとカーソルが表示されたら、次のように入力してコードを実行します。

```
In[1]:run BAP_Chapter1.py
```

Windows 環境では、コードによってはこの方法で実行できないことがあるかもしれません。その場合は、コード自体をコピーして実行します。`In[1]:` の後ろにコード全体をコピー&ペーストし、貼り付けられたコードの末尾で Shift キーを押しながら Enter キーを押してください。この方式により本書のすべてのコードが Windows 10 環境で動作することを確認してあります。

[*7] 以下では Jupyter QTConsole が先に記してありますが、その理由は訳者が本書のコードの動作確認を主にこれで行ったというだけのことです。本書のコードを実行するだけなら、どちらを選んでも構いません。対話型実行ツールとしては、このほかに、Web ブラウザで使う Jupyter や、統合開発環境の Spyder などがありますので、いろいろ使ってみて読者のお気に入りの実行ツールを選ぶとよいでしょう。

■ ターミナルのコマンドラインから直接実行する方法

本書のコードなど、動作することがわかっているコードファイルを一括して実行する場合には、ターミナルのコマンドラインから直接 Python を呼び出すことができます。読者が Python のコード作成に慣れているなら、この方法がシンプルで使いやすいでしょう。訳者の Linux 環境や Mac 環境では、この方法で問題なく本書のコードを実行できました。一方、Windows 環境ではこの方法でうまく実行できませんでした。読者の Windows 環境でも同様にうまく動作しないかもしれません。その場合は、先に記したように、Jupyter QTConsole または IPython を起動し、コードをファイルからコピーして貼り付けたり、キーボードから直接打ち込んだりして実行してください。

上記と同じ方法でターミナルを起動し、作業ディレクトリをデータフォルダに移動した後、コマンドラインから次のように直接 Python を呼び出して、コードを実行することができます。

```
> python BAP_Chapter1.py
```

コードの実行関連の TIPS

ここでは、訳者が本書の翻訳を通じて得たコードの実行に関連するヒントをいくつか紹介します。

■ Windows 向け

- PyMC3 は Theano の機能を使う際に g++[*8]を呼び出すことがあるため、MinGW ライブラリ（g++ などを含むフリーソフトウェアのパッケージ）をインストールしておきます。Anaconda Prompt から

  ```
  > conda install mingw
  ```

  と入力・実行してインストールしてください。MinGW ライブラリがインストールされていないと、コードによっては Theano の機能が使えず、処理が著しく低速になることがあります。

■ Mac 向け

- Mac にはデフォルトで g++ がインストールされていますが、そのままでは使えず、PyMC3 から Theano の機能を呼び出して高速処理することができません。g++ を使えるようにするためには、App Store から Xcode をダウンロードして

[*8] コンピュータ言語 C++ と互換性の高いフリーソフトウェアで、GNU プロジェクトによる開発ツール。

インストールし、次にターミナルを起動して、コマンドラインの `$` に続けて `g++ --version` と入力して Enter キーを押してください。ターミナル内に g++ のバージョン情報が表示されたら使える状態になっています。これで PyMC3 が Theano の機能を使えるようになります。

PyMC3 に関する情報源

PyMC3 のドキュメントサイトとして、次のサイトがあります。数多くのコード例が掲載されており、非常に参考になります。

http://docs.pymc.io/

PyMC3 は、本書の原著が出版された 2016 年 11 月から 2018 年 5 月までに、5 回のバージョンアップが行われています。PyMC3 のリリースノートは次のサイトにあります。

https://github.com/pymc-devs/pymc3/blob/master/RELEASE-NOTES.md

バージョンアップでどのような変更があったのかを知ることができるので、過去のコードを新しいバージョンに対応させる際に役立つでしょう。