

# データサイエンスへの誘い

第2回: 現代社会におけるデータサイエンスの活用事例

瓜生真也 (デザイン型AI教育研究センター・助教)

# 今日の目標

## Rの基本操作を学ぶ

# 社会の中での データサイエンスの活用

# さまざまな企業での事例

データサイエンスの手法を活用し、新たな価値創造を生み出す

## Netflix: ストリーミング配信サービス

視聴履歴や評価に基づく**推薦システム**の開発

## American Express: アメリカ合衆国起源のクレジットカード発行元

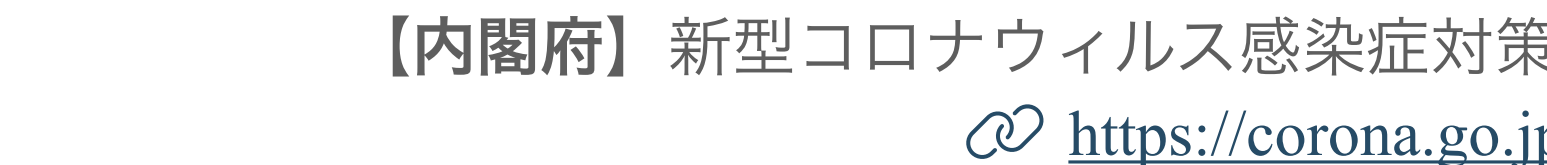
クレジットカードの利用状況から不正利用を検出（**異常検出**）、  
顧客の信用スコアの予測（**リスク評価**）

## Airbnb: 世界中の民間の宿泊施設と利用者をつなげるプラットフォーム

価格設定の最適化、供給と需要のバランスを整える

アメリカ合衆国の企業を中心に枚挙に暇がない…

# 新型コロナウイルス感染症



→小さな交差点に焦点を当てた独自の分析手法により、見落とされていた交差点の危険度を評価する基準となった

データに対してデータサイエンスで使われる分析手法・可視化表現を用い、これまでにない視点からの取材活動を行う調査報道の一手法。



# 【おさらい】 データサイエンスとプログラミング

現代ではきっても切り離せない関係

データサイエンスで行う作業の大半を担う存在

データ操作・可視化・分析・レポートまで

プログラムのパッケージ化… 誰かが書いたコードを使い回す（危険もある）

おすすめのエディタ… **RStudio** by Posit

**Visual Studio Code** by Microsoft

## データサイエンスでよく使われるプログラミング言語



どちらの言語も誰もが無料で使える

プログラミング

コピーから月面着陸までを可能にする

# かなり端折ったプログラミングの流れ



このあたりの詳しい内容は「情報科学入門」で！



# 言葉の整理

プログラム(program) コンピュータが特定の作業を行うための一連の命令（の集まり）

命令（プログラム）をコンピュータに与えることで機能する

一つ一つの命令は非常に単純な処理を実行する

多くの場合、複数の命令を組み合わせて、より複雑で実用的なプログラムを作成する→アプリケーション、ソフトウェア

## プログラミング

プログラムを作成・編集する過程や活動

デバッグ（エラーの検出と修正）を含む

## プログラミング言語

プログラムを実装するために使用される言語

多様な言語… C、Python、R、Swift

選択する言語は目的の要件によって異なる

## プログラマー

プログラムを開発・編集・デバッグする人々

プログラムの内容に応じて職種がわかる… アプリケーションエンジニア、ゲーム開発者、データサイエンティストなど

---

## スクリプト(script)ないしコード(code)

コードはプログラムを構成するプログラミング言語で記述された一連の命令や文を指す

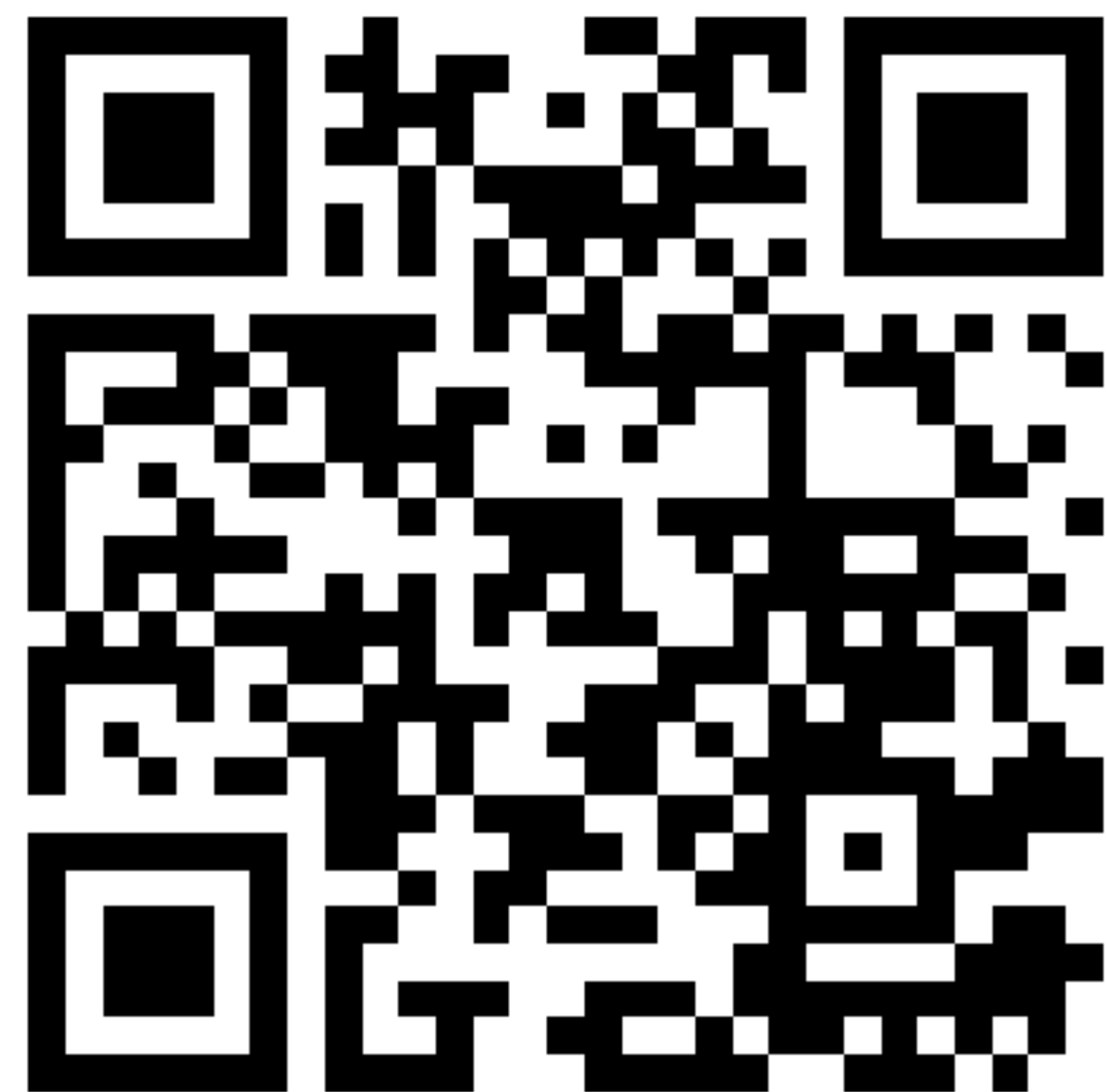
スクリプトはコードの断片

# 退屈なことはプログラムに任せよう

例えば…ウェブページのURLをもとにQRコードを作りたい  
そういうウェブサービス・ツールはいくつか存在する

対象のURLが1つならいいかも 対象のURLをコピーして貼り付け、生成 😊

URLが1000種類あったら？ 1000回もやってられない 😓



## プログラミングでできることの例

データ分析、機械学習・深層学習、ウェブスクレイピング、アプリケーション開発  
プログラミング言語ごとに得意・不得意、向き不向きがある

## プログラムとAIの力で時間短縮

戦略1: プログラミングを学ぶ

まずはここからはじめよう

戦略2: AIの力を借りる

後半で解説

# Jupyter

# Project Jupyter

<https://jupyter.org/>

科学的研究における再現性とインタラクティブ性を高めるためのプロジェクト

## Jupyter Notebook

コードと実行結果、またそれらを補足する文章（画像等を含む）からなるファイルを作成するためのツール  
ノートブックファイルはJSON形式でメタデータ等を記録した .ipynb として扱われる→テキストファイル

Python, R, Juliaを始め、40以上の言語に対応

ウェブアプリケーションとしてウェブブラウザ上で動作する

Google Colaboratory (Colab)… Googleが提供するノートブック環境

Binder

Kaggle Notebooks

## JupyterLab

Jupyter Notebookのための優れた実行環境

タブ機能による複数ファイルの同時編集やファイル操作が柔軟に可能

→Jupyter Notebook単体で使うよりも操作性に優れる



# Jupyter Notebook

The screenshot shows a Jupyter Notebook interface in a web browser. The address bar indicates the URL is `localhost:8888/lab/tree/week2/qrcode.ipynb`. The notebook has a single tab titled `qrcode.ipynb`. The kernel is `Python 3 (ipykernel)`. The notebook content is in Japanese and includes a title cell, two text cells explaining the process, a code cell for installing libraries, and three code cells for generating a QR code.

**## ウェブサイトのURLからQRコードを作成する**

まずはプログラムの実行に必要なPythonライブラリをインストールします。QRコードの作成には、`qrcode`と`pillow`の2つのライブラリを使用します。pipコマンドでライブラリをインストールしましょう。

Jupyternotebookでは、コードセルの行頭に `!` を付けた内容をコマンドとして処理します。すなわち、次のコードセルの内容はターミナルで `pip install qrcode[pil]` `pillow` を実行したときと同じです。

なお、次のセルはコードセルではなくテキストセル扱いになっているので、実際にこのノートブック上でインストールを行う際はセルの種類を `Raw` から `Code` に変換する必要があります。

```
! pip install qrcode[pil] pillow
```

### 1つのURLからQRコードの生成

pipコマンドにより2つのライブラリのインストールが終わったらPythonでのQRコード生成を行います。さきほどインストールしたライブラリを `import` によって読み込みましょう。こうすることで `qrcode` や `Image` ライブラリの機能が利用可能となります。

```
[1]: # QRコードの生成に必要なライブラリを読み込む
import qrcode
from PIL import Image
```

```
[2]: # QRコードにエンコードするテキスト
data = "https://github.com/uribo/exeai"
```

```
[3]: # QRコードの設定
```

At the bottom of the interface, the status bar shows `Simple` mode, `0` lines, `2` cells, `Python 3 (ipykernel) | Idle`, `Mode: Edit`, `Ln 1, Col 26`, `qrcode.ipynb`, and `0` notifications.

ここまでのまとめ



# ここまでのまとめ

ビッグデータとデータサイエンスの手法を組み合わせ、  
社会で新たな価値を提供する企業・団体が増えている

他企業との差別化、データサイエンティストの組織的な育成も進む

意思決定を促すツールとしてのデータサイエンス

プログラムはコンピュータ上で「何か」を行うために利用する術

プログラムの実行には機械語が求められるが、「プログラミング言語」は  
人間が判読可能な形で記述し、翻訳によってコンピュータに命令を送る形式をとる

プログラミング言語に指示を記録しておくことで、繰り返しの処理や条件が変わった時の対処がしやすくなる

Jupyter Notebookの利用により、コードの内容と実行結果、  
さらに補足文章を加えることができる

ノートブックで実行するコードの記述は「コードセル」で行う→Python, R, Juliaを始め、40以上の言語に対応

Jupyter NotebookはColabやJupyterLabなどでも利用される

Rのいろは




AIによるプログラミング支援


# 分からないことをChatGPTに聞いてみよう

## グラフの作成方法

Model: Default (GPT-3.5)

 Rのggplot2パッケージを使ってワッフルチャートを作成したいです。コードを示してください



 ggplot2パッケージを使用してワッフルチャートを作成するためには、先にビジュアル化するデータセットを準備する必要があります。ワッフルチャートは、単純なカウントデータをグリッド状の正方形で表現する方法で、通常、割合やパーセンテージなどを示すために使用されます。

以下は、ggplot2パッケージを使用してワッフルチャートを作成するための基本的なコードです。

R


Copy code

```
# 必要なライブラリを読み込む
library(ggplot2)
library(waffle)


# ワッフルチャート用のデータを作成する
```

Regenerate response

## パッケージのインストール方法

 macを使っています。rstanパッケージがインストールできずに困っています。Rのバージョンは4.2.1を使っています。



 macOS上でrstanパッケージをインストールするには、まずRtoolsとXcodeのインストールが必要です。また、Rバージョン4.2.x以降を使用する場合は、C++14が必要です。これを確認するには、ターミナルから以下のコマンドを実行してください。

bash

Copy code

```
echo 'int main(){}' | g++ -std=c++14 -xc++ -
```

もし以下のようなエラーが表示された場合は、C++14がサポートされていない可能性があります。

csharp

Copy code

```
error: invalid argument allowed with 'C++'
```

Regenerate response

# 【推奨】 GitHubの活用

## GitHub by Microsoft

プログラミングのソースコード置き場、開発者のためのプラットフォーム  
バージョン管理システムGitを用いたコード共有  
学生であれば有料プランを無料で利用可能（教育プログラム、要申請）

## GitHub Copilot

コメント、関数名等からコードの入力を自動補完するAIツール  
VSCodeの機能拡張としても利用可能  
↑ 上記の教育プログラムに加入で無料

# 【推奨】GitHubの活用



The screenshot shows an RStudio interface. The top toolbar includes icons for running, refreshing, and window management. The console shows the prompt 'tidyverse.R > ...'. The script editor contains the following R code:

```
1 # 入力の際に表示されるグレーの文字は
2 # GitHub Copilotによる入力補完です
3 # このファイルはGitHub Copilotによって自動生成されました
4 library(tidyverse)
5 df <- read_csv("https://raw.githubusercontent.com/rstudio/c
6
7 ggplot(df, aes(x = reorder(country, -lifeExp), y = lifeExp))
  geom_col() +
  coord_flip() +
  labs(x = NULL, y = "Life Expectancy (years)", title = "Li
```

The bottom status bar shows 'main\*+', a refresh icon, a close icon, '0' errors, '0' warnings, '1' info messages, 'スペース: 4' (spaces: 4), 'UTF-8' encoding, 'LF' line endings, 'R' language, and icons for help, search, and notifications.