

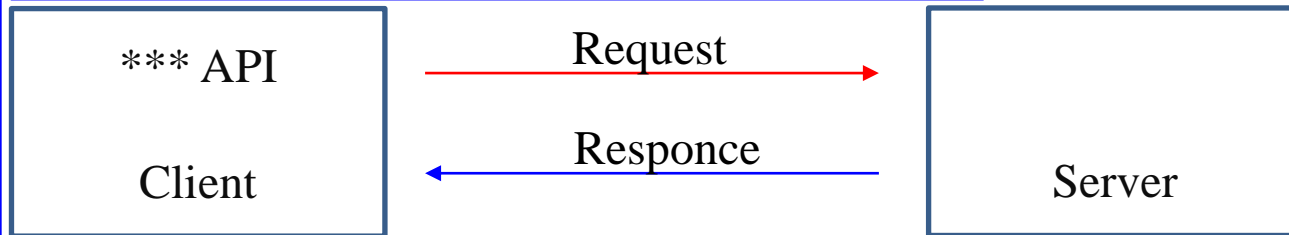
API・スクレイピングによるデータ収集・可視化・分析

データの
取得方法

- ①データベースから取得
- ②APIを用いて取得
- ③Webスクレイピングを用いて取得

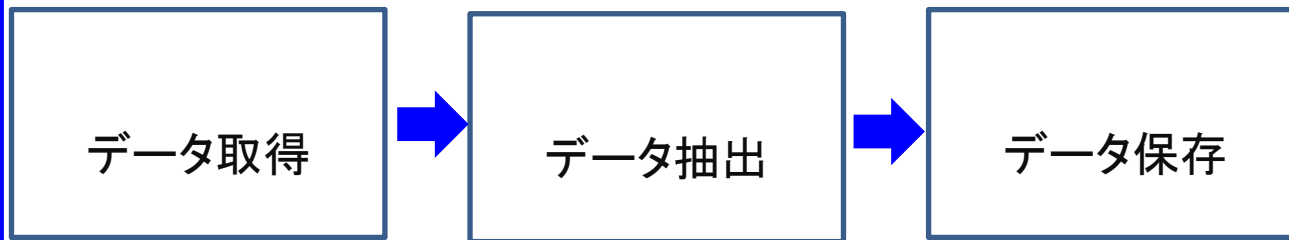
REST API

②API: Application Programming Interface



APIを用いて情報やデータの取得が可能

③Web scraping



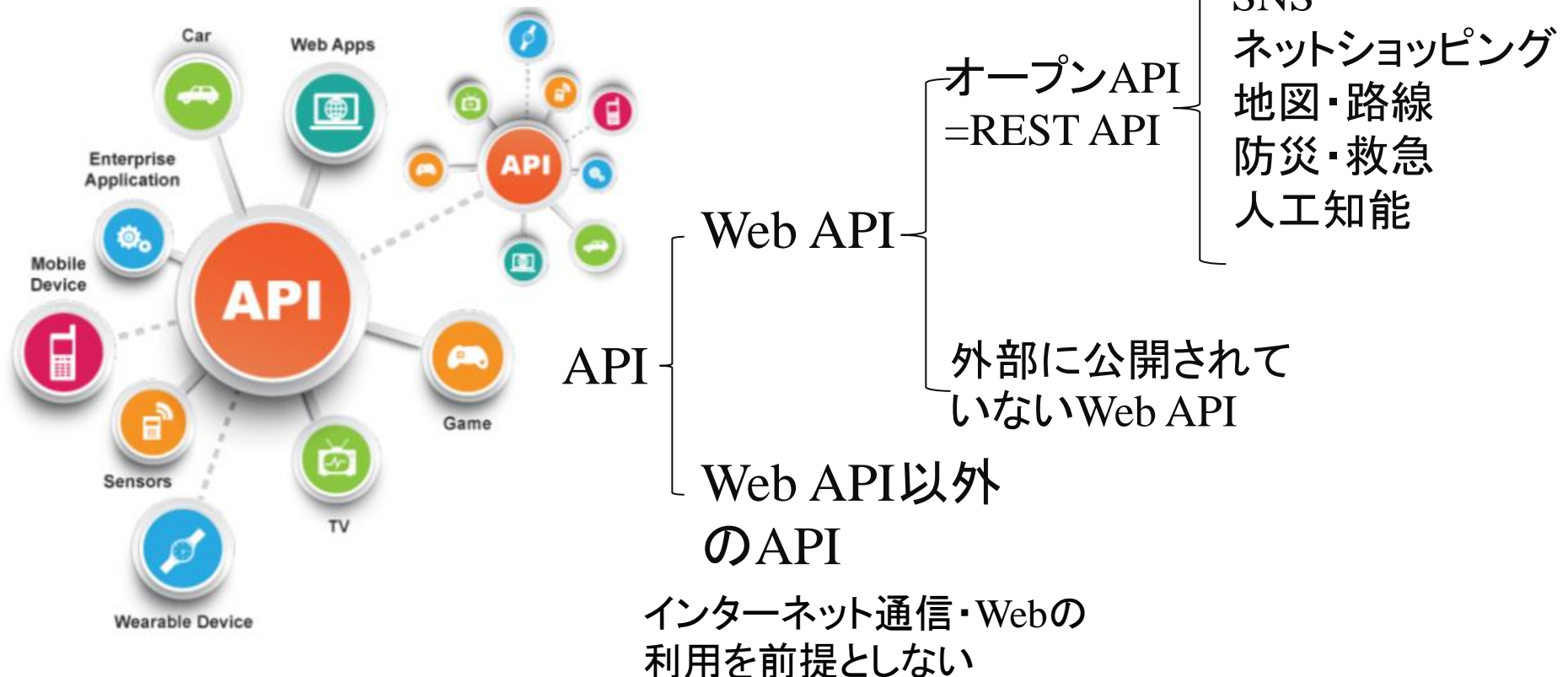
Webスクレイピングを用いて情報やデータの取得が可能

APIによるデータ収集と利活用

・API: Application Programming Interface = 様々な機器やアプリケーションをつなぐ

他者が提供する情報を収集したり、提供している機能を利用する方法

様々な種類のAPI 特にWeb APIが注目



Source: https://www.soumu.go.jp/ict_skill/pdf/ict_skill_1_5.pdf

人工知能に関するAPI事例

- ・人工知能に関するAPIでは、画像認識、音声認識を行ったり、文章の分類が可能
- ・外資系企業 (Google, AWS, Microsoft, IBM)、日本企業 (東芝, NTTドコモ, リクルート) が提供する人工知能のAPI



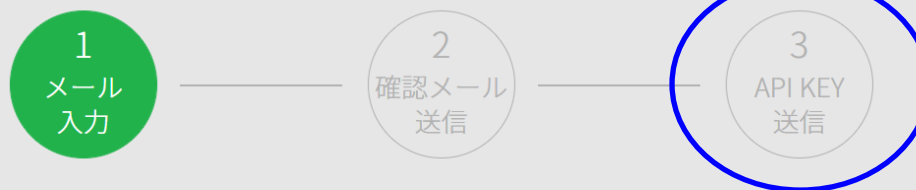
The screenshot shows the A3RT API product page. The header includes the A3RT logo, navigation links for ABOUT, PRODUCT, NEWS, and LIBRARY, social media icons for Facebook and Twitter, and a RECRUIT logo. A user login status 'Internal user >>' is visible. The main content area features two API descriptions:

- Listing API**: An icon of a document with a green bar chart. The text describes an API for generating lists based on user action logs, used for online recommendations and targeted emails. It mentions that data is provided for model training and that the API can generate and download lists. The version 'version1' is noted.
- Image Influence API**: An icon showing a line graph with two data series labeled A and B. The text explains that this API uses a model to predict scores (0-9) for images, such as meat photos, based on user preferences. It includes a link to 'MORE'.

Source: <https://a3rt.recruit-tech.co.jp/product/>

[ABOUT](#)[PRODUCT](#)[NEWS](#)[LIBRARY](#)[Internal user >>](#)[TOP](#) > [PRODUCT](#) > [Listing API](#) > API発行

Listing API API発行



API を利用するにあたって必要な API KEY を発行します。

[利用規約](#)、[プライバシーポリシー](#)を確認・ご承諾の上、下記のフォームにメールアドレスを記入後、送信ボタンを押して下さい。

OpenWeatherMapAPI事例

世界の天気情報を公開しているオンラインサービスを利用する

The screenshot shows the 'Sign In To Your Account' page of OpenWeatherMap. The navigation bar at the top includes links for Guide, API, Pricing, Maps, Our Initiatives, Partners, Blog, Marketplace, and Sign In (highlighted with a yellow box). The sign-in form contains fields for email and password, a 'Remember me' checkbox, and a 'Submit' button. Below the form, there are links for 'Not registered? Create an Account.' (highlighted with a blue box and an arrow) and 'Lost your password? Click here to recover.' To the right of the form, three numbered steps are listed: ① アカウント作成, ② APIキー取得, and ③ API利用.

Sign In To Your Account

Enter email

Password

☐ Remember me

Submit

Not registered? [Create an Account.](#)

Lost your password? [Click here to recover.](#)

① アカウント作成
② APIキー取得
③ API利用

Source: <https://openweathermap.org/>

②APIキー取得

API key:

- Your API key is **0c2adddbc1f320c0d13388f6a289e5d3**
- Within the next couple of hours, it will be activated and ready to use
- You can later create more API keys on your [account page](#)
- Please, always use your API key in each API call

Endpoint:

- Please, use the endpoint api.openweathermap.org for your API calls
- Example of API call:

[api.openweathermap.org/data/2.5/weather?
q=London,uk&APPID=0c2adddbc1f320c0d13388f6a289e5d3](https://api.openweathermap.org/data/2.5/weather?q=London,uk&APPID=0c2adddbc1f320c0d13388f6a289e5d3)

Useful links:

- API documentation <https://openweathermap.org/api>
- Details of your plan <https://openweathermap.org/price>
- Please, note that [16-days daily forecast](#) and [History API](#) are not available for Free subscribers

New Products Services **API keys** Billing plans Payments Block logs My orders My profile Ask a question

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

| Key | Name | Create key |
|----------------------------------|---------|---|
| 0c2adddbc1f320c0d13388f6a289e5d3 | Default | <input type="text" value="API key name"/> <button>Generate</button> |

③API利用

| Weather in your city | | | | |
|---|--|---|--|---|
| Guide API Pricing Maps Our Initiatives Partners Blog Marketplace sumico | | | | |
| Current weather and forecasts collection | | | | |
| Free | Startup 40 USD / month | Developer 180 USD / month | Professional 470 USD / month | Enterprise 2,000 USD / month |
| Get API key | Subscribe | Subscribe | Subscribe | Subscribe |
| 60 calls/minute 1,000,000 calls/month | 600 calls/minute 10,000,000 calls/month | 3,000 calls/minute 100,000,000 calls/month | 30,000 calls/minute 1,000,000,000 calls/month | 200,000 calls/minute 5,000,000,000 calls/month |
| Current Weather | Current Weather | Current Weather | Current Weather | Current Weather |
| Minute Forecast 1 hour* | Minute Forecast 1 hour** | Minute Forecast 1 hour | Minute Forecast 1 hour | Minute forecast 1 hour |
| Hourly Forecast 2 days* | Hourly Forecast 2 days** | Hourly Forecast 4 days | Hourly Forecast 4 days | Hourly Forecast 4 days |
| Daily Forecast 7 days* | Daily Forecast 16 days | Daily Forecast 16 days | Daily Forecast 16 days | Daily Forecast 16 days |
| National Weather Alerts* | National Weather Alerts** | National Weather Alerts | National Weather Alerts | National Weather Alerts |

③API利用

Weather in your city

Guide

API

Pricing

Maps

Our Initiatives

Partners

Blog

Marketplace

sumico ▼

Weather API

[Home](#) / [Weather API](#)

Please, [sign up](#) to use our fast and easy-to-work weather APIs for free. In case your requirements go beyond our freemium account conditions, you may check the entire list of our [subscription plans](#). You can read the [How to Start](#) guide and enjoy using our powerful weather APIs right now.

Current & Forecast weather data collection

Current Weather Data

API doc

Subscribe

- Access current weather data for any location including over 200,000 cities
- We collect and process weather data from different sources such as global and local weather models, satellites, radars and a vast network of weather

Hourly Forecast 4 days

API doc

Subscribe

- Hourly forecast is available for 4 days
- Forecast weather data for 96 timestamps
- JSON and XML formats
- Included in the Developer, Professional and Enterprise subscription plans

One Call API

API doc

Subscribe

- Make one API call and get current, forecast and historical weather data

5 Day / 3 Hour Forecast

API doc

Subscribe

- 5 day forecast for any location or city
- 5 day forecast with a 3-hour step
- JSON and XML formats
- Included in both free and paid subscriptions

5日間3時間ごとの天気

③API利用

APIを使って現在から5日間3時間ごとの天気の詳細を取得

Call 5 day / 3 hour forecast data

By city name

都市名を指定して5日間3時間ごとの天気の詳細を取得

You can search weather forecast for 5 days with data every 3 hours by city name. All weather data can be obtained in JSON and XML formats.

API call

```
api.openweathermap.org/data/2.5/forecast?q={city name}&appid={API key}
```



```
api.openweathermap.org/data/2.5/forecast?q={city name}, {state code}&appid={API key}
```



```
api.openweathermap.org/data/2.5/forecast?q={city name}, {state code}, {country code}&appid={API key}
```



OpenWeatherMapAPIを呼び出し+都市名を指定して5日間3時間ごとの天気を取得



WeatherAPI.ipynb ☆ Colaboratory: APIによるデータ収集・可視化・分析する開発環境

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ 最終編集: 2021年10月27日

+ コード + テキスト

Pythonでライブラリやモジュールを使用するにはimportを使用

requestsは、HTTP向けのPythonのライブラリ

from モジュール名 import クラス名 (もしくは関数名や変数名)

5日間 (3時間ごと) の天気を取得する: 東京

```
url = "http://api.openweathermap.org/data/2.5/forecast?q={city}&appid={key}&lang=ja&units=metric"
url = url.format(city="Tokyo,JP", key="0c2adddbc1f320c0d13388f6a289e5d3")
```

書き方: city="都市名" key="APIキー"

```
jsondata = requests.get(url).json()
pprint(jsondata)
```

pprint モジュールは Python データ構造をきれいに印刷する機能

requests.get('URL')
で GET リクエスト
できる

JSONデータで出力

キー (key) と値 (value) のペア

```
{'city': {'coord': {'lat': 35.6895, 'lon': 139.6917},
'country': 'JP',
'id': 1850144,
'name': '東京都',
'population': 12445327,
'sunrise': 1635281885,
'sunset': 1635321116,
'timezone': 32400},
'cnt': 40,
'cod': '200',
'list': [{'clouds': {'all': 75},
'dt': 1635325200,
'dt_txt': '2021-10-27 09:00:00',
```

Pythonで使えるWebスクレイピングのライブラリ

モジュール・パッケージ・ライブラリの関係

科学技術計算でよく使われるNumpyやグラフ描画に使われるMatplotlib、データ解析を支援する機能を提供するPandasなど

↓ライブラリ

(標準ライブラリ・外部ライブラリ)

↓パッケージ

パッケージは同系列のモジュールを集めたもの

モジュールは関数やクラスなどを1つのファイルにまとめたもの

↓モジュール

↓クラス、関数

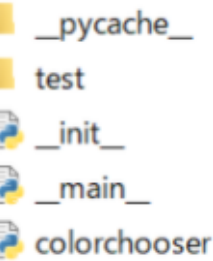
クラスはデータと機能を組み合わせる方法を提供します。関数は様々な処理が機能として1つにまとまっているもの

Pythonにおけるモジュール・パッケージ・ライブラリについて



sample
sample.py

モジュールとは、Python のコードを記述した、拡張子が「.py」のファイルのことです。ファイル名がモジュール名になります。



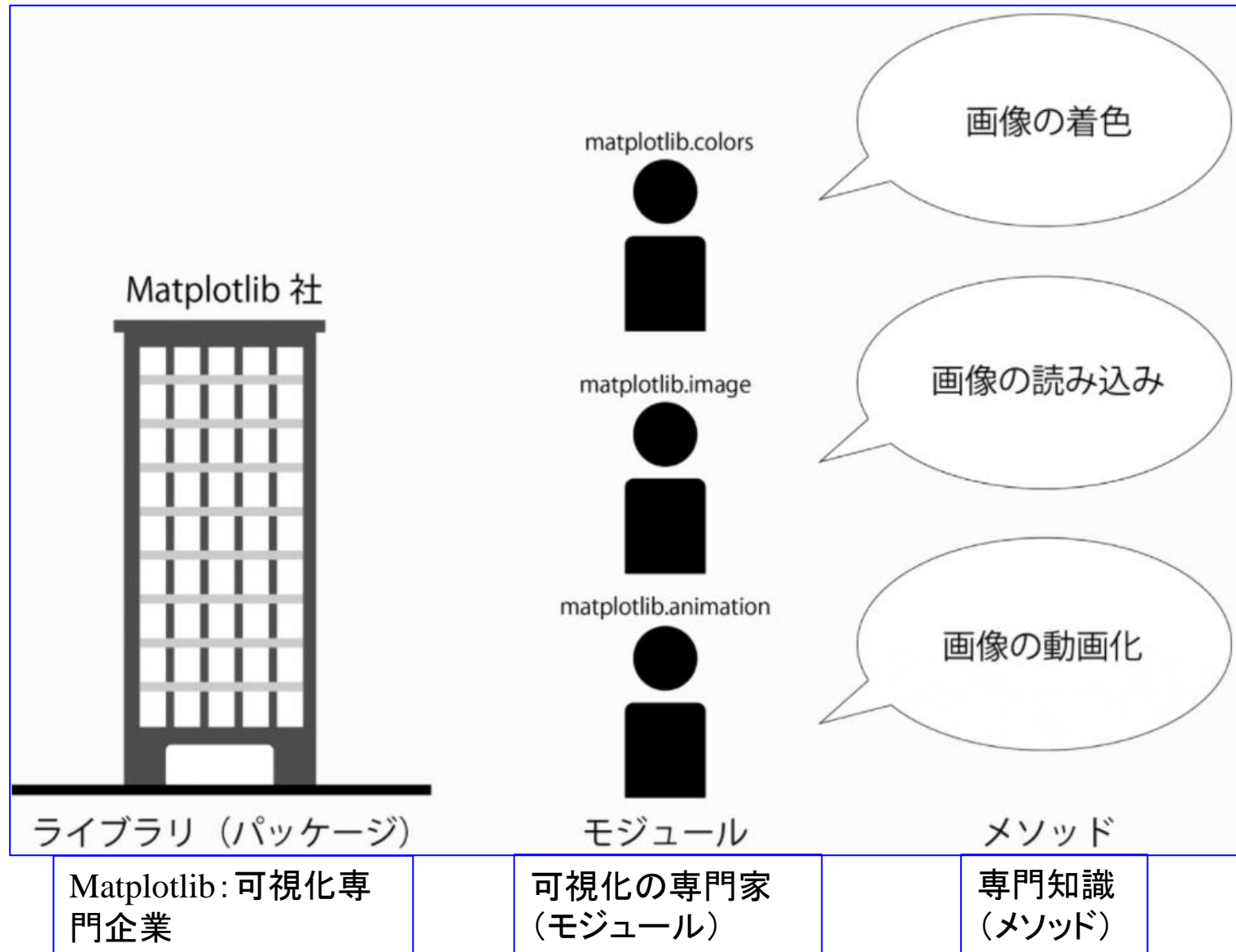
複数のモジュールをディレクトリ(フォルダ)に入れて整理し、ひとまとまりにしたものが**パッケージ**です。

__init__.py の役割: 階層のモジュールを検索するためのマーカーとして利用

Library:
NumPy
Scikit-learn
Matplotlib
Seaborn

ライブラリとはいくつかのパッケージをまとめたものとなります。標準ライブラリはPythonに標準で用意されたライブラリのこと、外部ライブラリは標準で用意されておらずインストールが必要なライブラリのことです。

Pythonにおけるモジュール・パッケージ・ライブラリについて



Source: <https://sorabatake.jp/13287/>

OpenWeatherMapAPIを呼び出し+都市名を指定して5日間3時間ごとの天気を取得



WeatherAPI.ipynb ☆ Colaboratory: APIによるデータ収集・可視化・分析する開発環境

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ 最終編集: 2021年10月27日

+ コード + テキスト

Pythonでライブラリやモジュールを使用するにはimportを使用

requestsは、HTTP向けのPythonのライブラリ

from モジュール名 import クラス名 (もしくは関数名や変数名)

5日間 (3時間ごと) の天気を取得する: 東京

```
url = "http://api.openweathermap.org/data/2.5/forecast?q={city}&appid={key}&lang=ja&units=metric"
url = url.format(city="Tokyo,JP", key="0c2adddbc1f320c0d13388f6a289e5d3")
```

書き方: city="都市名" key="APIキー"

```
jsondata = requests.get(url).json()
pprint(jsondata)
```

pprint モジュールは Python データ構造をきれいに印刷する機能

JSONデータで出力

requests.get('URL')
で GET リクエスト
できる

```
{'city': {'coord': {'lat': 35.6895, 'lon': 139.6917},
  'country': 'JP',
  'id': 1850144,
  'name': '東京都',
  'population': 12445327,
  'sunrise': 1635281885,
  'sunset': 1635321116,
  'timezone': 32400},
 'cnt': 40,
 'cod': '200',
 'list': [{'clouds': {'all': 75},
  'dt': 1635325200,
  'dt_txt': '2021-10-27 09:00:00',
```

グラフの元になる「何時・何度」の表データを作成

```
[5] import requests
import json
from pprint import pprint
from datetime import datetime, timedelta, timezone
import pandas as pd
```

Pandasは、Pythonでデータ分析を効率的に行うためのライブラリ

5日間（3時間ごと）の天気を取得する：東京

```
url = "http://api.openweathermap.org/data/2.5/forecast?q={city}&appid={key}&lang=ja&units=metric"
url = url.format(city="Tokyo,JP", key="0c2adddbc1f320c0d13388f6a289e5d3")
```

```
jsondata = requests.get(url).json()
df = pd.DataFrame(columns=["気温"])
tz = timezone(timedelta(hours=+9), 'JST')
for dat in jsondata["list"]:
    jst = str(datetime.fromtimestamp(dat["dt"], tz))[:-9]
    temp = dat["main"]["temp"]
    df.loc[jst] = temp

pprint(df)
```

pandasで空のDataFrameを作成し、
項目を気温に設定

ここに「何時(JST)」をインデックスにして、
「何度(気温)」のデータを追加していけば
「何時・何度」のDataFrameが作成される

| | 気温 |
|------------------|------|
| 2022-02-07 12:00 | 6.17 |
| 2022-02-07 15:00 | 6.44 |
| 2022-02-07 18:00 | 6.64 |
| 2022-02-07 21:00 | 6.00 |
| 2022-02-08 00:00 | 4.20 |
| 2022-02-08 03:00 | 3.46 |
| 2022-02-08 06:00 | 3.33 |

「何時・何度」DataFrameをmatplotlibでグラフ化

```
import requests
import json
from pprint import pprint
from datetime import datetime, timedelta, timezone
import pandas as pd
import matplotlib.pyplot as plt
import japanize_matplotlib
```

Pandasは、Pythonでデータ分析を効率的に行うためのライブラリ

5日間 (3時間ごと) の天気を取得する: 東京

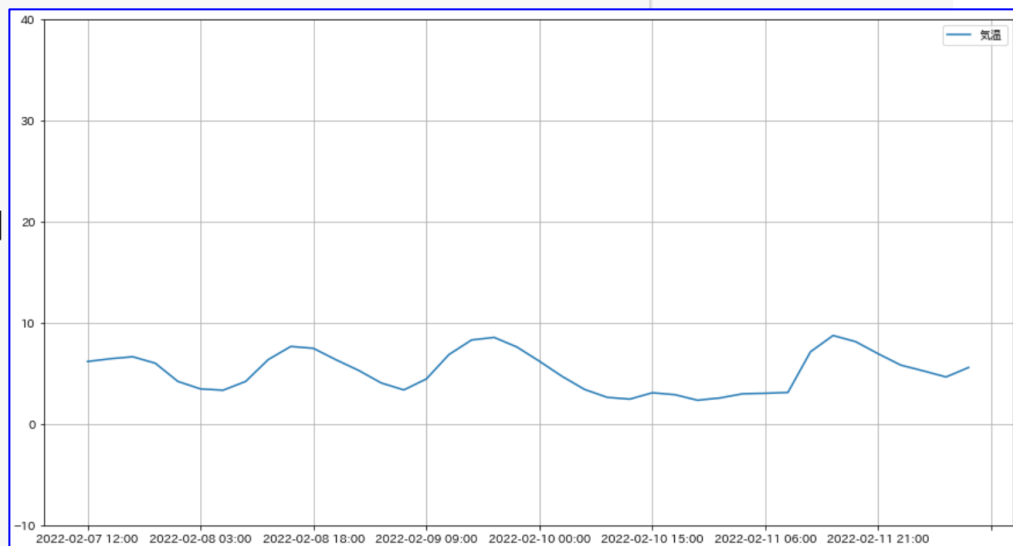
```
url = "http://api.openweathermap.org/data/2.5/forecast?q={city}&appid={key}&lang=ja&units=metric"
url = url.format(city="Tokyo,JP", key="0c2adddbc1f320c0d13388f6a289e5d3")
```

```
jsondata = requests.get(url).json()
df = pd.DataFrame(columns=["気温"])
tz = timezone(timedelta(hours=+9), 'JST')
for dat in jsondata["list"]:
    jst = str(datetime.fromtimestamp(dat["dt"], tz))
    temp = dat["main"]["temp"]
    df.loc[jst] = temp
```

```
df.plot(figsize=(15,8))
plt.ylim(-10,40)
plt.grid()
plt.show()
```

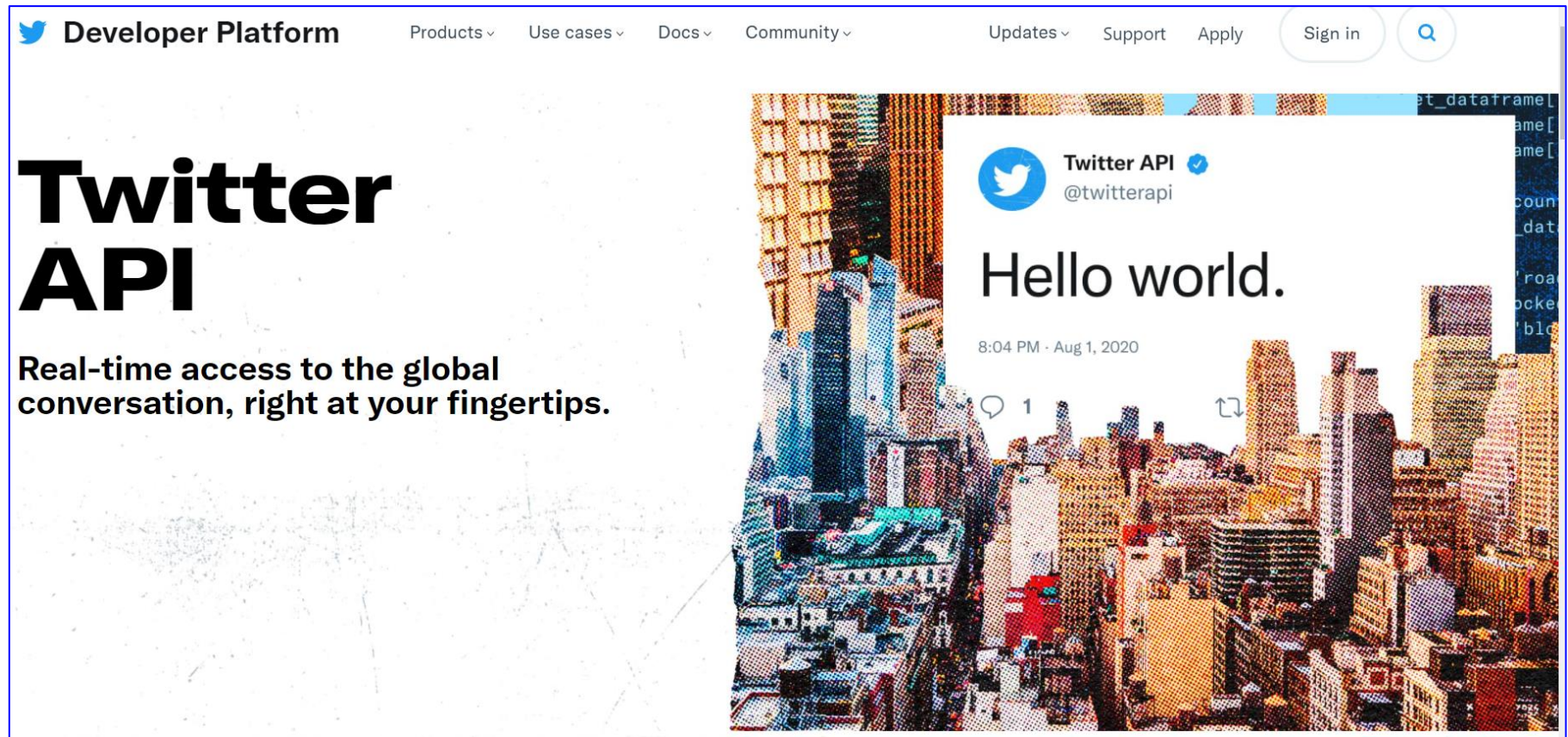
グラフの画面
サイズを調整

気温のスケールを調整:
最小値-10, 最大値40



SNSに関するAPI事例

- ・SNSのAPIを利用すると、公式サービス以外からの情報投稿や情報の取りまとめが可能
- ・TwitterのAPIを使うと、自サイトや自作アプリからTwitterにおけるツイート(Twitter上の発言)の検索や投稿が可能



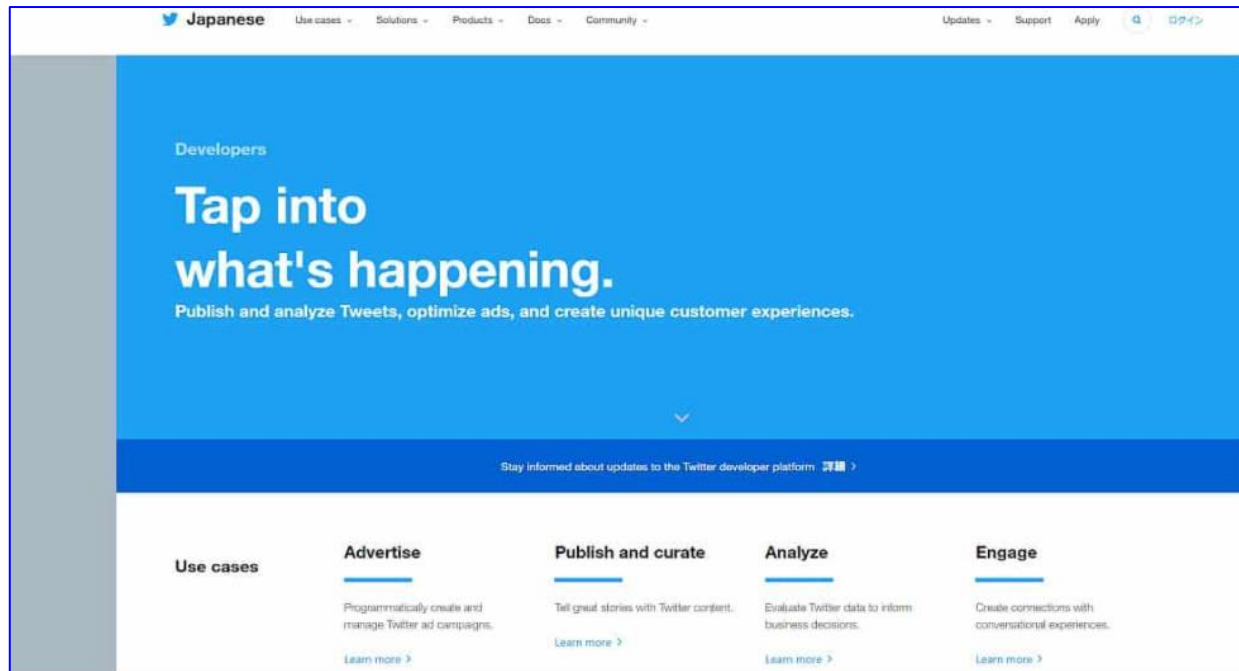
Source: <https://developer.twitter.com/en/products/twitter-api>

Twitter APIの申請・APIキー発行方法

公式: <https://developer.twitter.com/>

1. Twitterアカウント作成し、Twitter APIを登録申請
2. Twitter社から Twitter API の使用用途についての確認メール受信
3. Twitter APIの使用用途を記載してメールを返信
4. Approveされたメール受信

Twitter APIの申請・APIキー発行方法については以下のサイトがわかりやすい
https://auto-worker.com/blog/?p=3157#toc_id_1

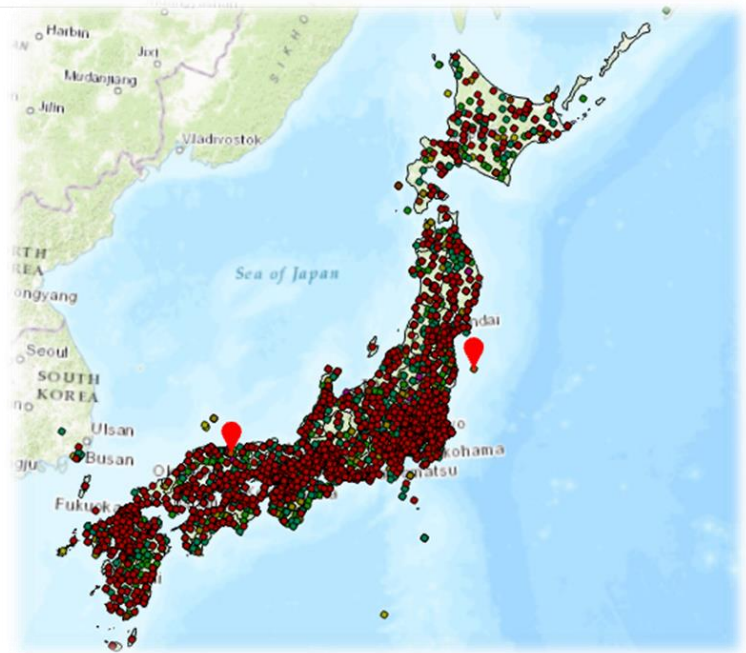


CONSUMER_KEY
CONSUMER_SECRET
ACCESS_TOKEN
ACCESS_TOKEN_SECRET

Jupyter notebookでTwitter APIを呼び出し＋ツイート取得

APIによるデータ収集・可視化・分析する開発環境

```
1. import tweepy
2. import datetime
3. import pandas as pd
4.
5. ## 認証処理
6. consumer_key = "<取得したkey>"
7. consumer_secret = "<取得したkey>"
8. access_token_key = "<取得したkey>"
9. access_token_secret = "<取得したkey>"
10.
11. auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
12. auth.set_access_token(access_token_key, access_token_secret)
13. api = tweepy.API(auth)
14.
15. ## 取得対象のスクリーンネーム
16. screen_name =
17.
18. ## ツイート取得
19. tweet_data = []
20. for tweet in tweepy.Cursor(api.user_timeline, screen_name = screen_name, exclude_replies = False).items():
21.
22. tweet_data.append([tweet.id, tweet.created_at+datetime.timedelta(hours=9), tweet.text.replace('¥n', ''), tweet.fav
23.
24. columns_name = ["ツイートID", "時間", "text", "fav", "RT"]
25. df = pd.DataFrame(tweet_data, columns=columns_name)
```



```
{ "_id" : ObjectId("57c7d38609c5d02488c0230b"), "contributors" : null, "truncated" : false,
"text" : "本文", "is_quote_status" : false, "in_reply_to_status_id" : null, "id" :
NumberLong("771242376308195330"), "favorite_count" : 0, "entities" : { "symbols" : [ ],
"user_mentions" : [ { "id" : NumberLong("4756147100"), "indices" : [ 3, 18 ], "id_str" :
"4756147100", "screen_name" : "TrainOperation",
"name" : "ユーザー名" } ]...
```

Google colaboryでTwitter APIを呼び出し1週間分のツイートを検索、ドライブ上にCSV保存、google map上に可視化

```
!pip install tweepy
import tweepy
import pandas as pd
import re

consumer_api_key = "XXX"
consumer_api_secret_key = "XXX"
access_token = "XXX"
access_token_secret = "XXX"

auth = tweepy.OAuthHandler(consumer_api_key, consumer_api_secret_key)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth_handler=auth, wait_on_rate_limit=True)

keyword = "ランチ"
geocode = "35.6467139,139.707889,1km"

tweets = tweepy.Cursor(api.search, q=keyword, tweet_mode="extended", geocode=geocode).items()
tweets_list = [tweet._json for tweet in tweets]
df = pd.json_normalize(tweets_list)
df2 = df[df["place.place_type"]=="poi"] # ランドマークが指定されているもの
df2["place.bounding_box.coordinates.2"] = list(df2["place.bounding_box.coordinates"].map(lambda x: x[0][0]))
df2 = df2[["place.bounding_box.coordinates.2", "place.full_name"]]
filename=keyword+"_"+geocode+"_1w"
df2.to_csv('/content/drive/My Drive/blog/{_}tweetdata.csv'.format(filename), index=False)
```

Source: <https://developer.twitter.com/en/products/twitter-api>

取得した経緯度付きのツイートをgoogle map上に可視化

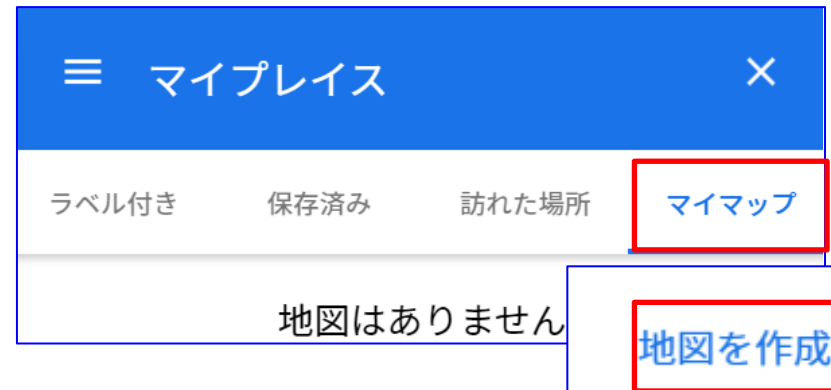
①Google Map を開く

②検索ボックス隣の「≡」アイコンをクリックして、メニューを表示

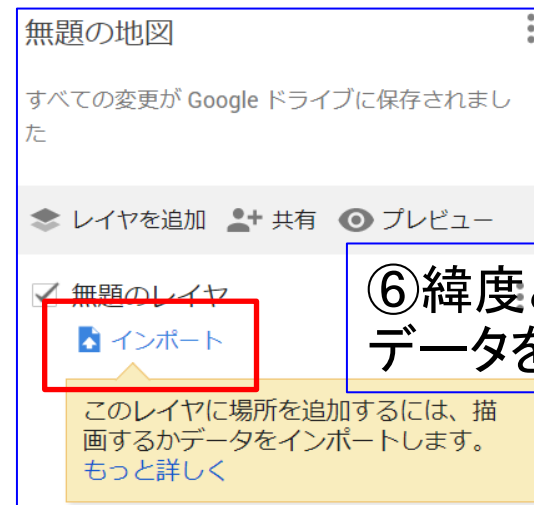
③表示されたメニューの中から「マイプレイス」を選択



④タブを「マイマップ」に切替え、画面下の「地図を作成」をクリック



⑤「無題の地図」と「無題のレイヤ」が表示されるので、「インポート」を選択



⑥緯度と経度を持ったデータをインポート

⑦GoogleDriveからファイルをアップロード

⑧緯度と経度に対応する「列」の名称を選択

⑩データが読み込まれると、位置情報に応じた場所にピンが打たれます。見づらい場合ピンのマークや色の変更が可能



目印を配置する列の選択

地図上に目印を配置する場所を示す列（住所や緯度/経度のペアなど）をファイルから指定してください。すべての列がインポートされます。

- ☒ latitude (latitude) ⓘ
- ☒ longitude (longitude) ⓘ
- ☐ number ⓘ

続行

戻る

キャンセル

⑨個別のマーカーのタイトルとなる「列」を選択

マーカーのタイトルとして使用する列

目印のタイトルとして使用するデータの列（場所やユーザーの名前など）を選択してください。

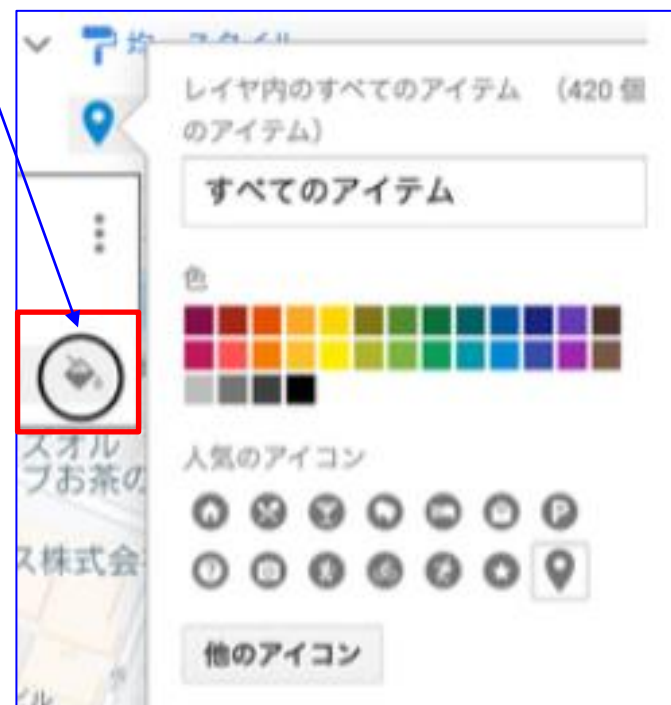
- ☐ latitude ⓘ
- ☐ longitude ⓘ
- ☒ number ⓘ

完了

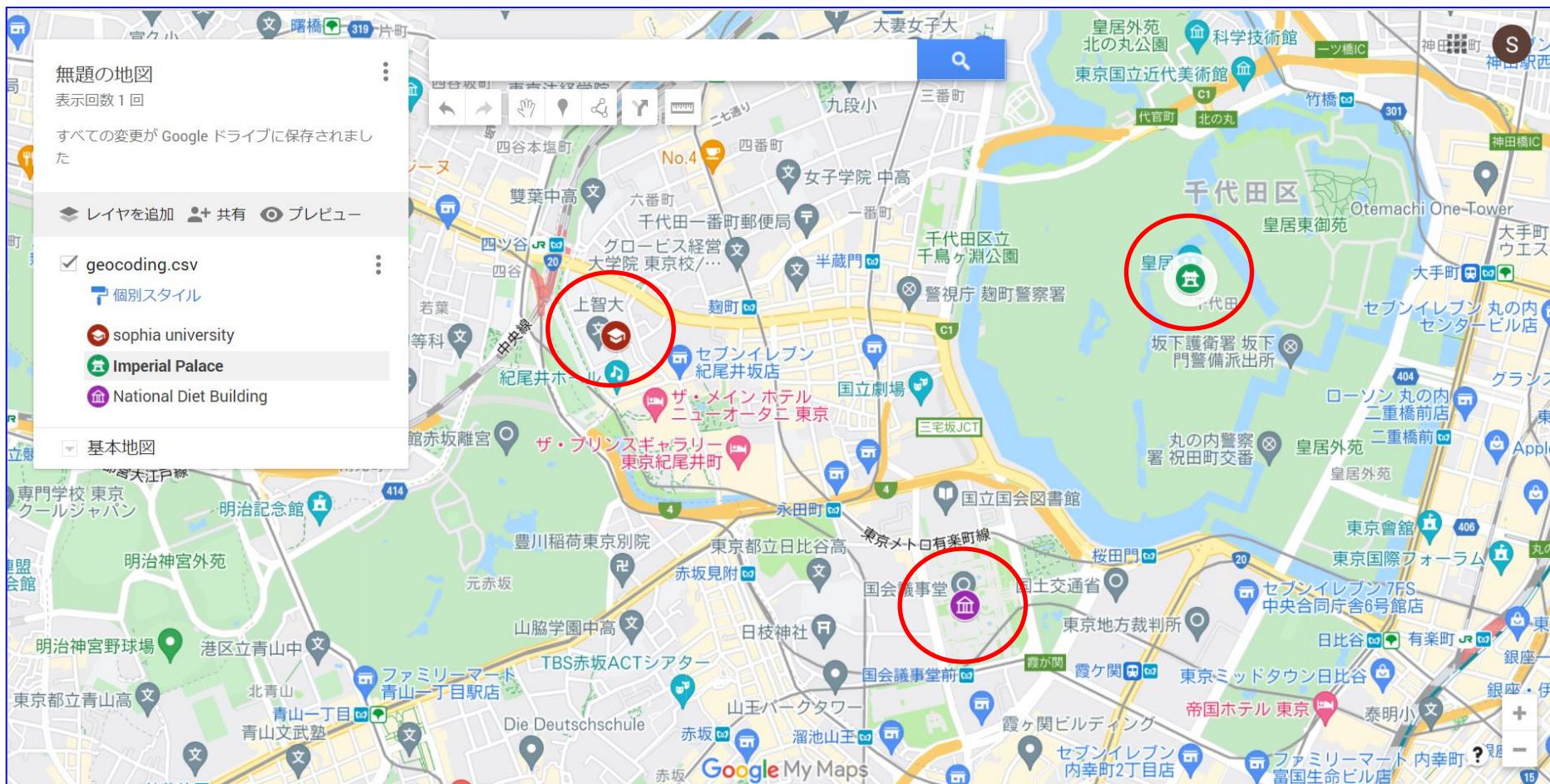
戻る

キャンセル

⑪全てのアイテムの端にあるペンキのアイコンを押し、色とアイコンの変更が可能



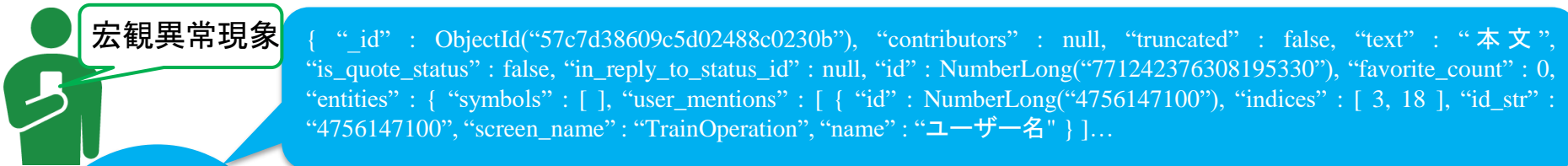
| | A | B | C | D |
|---|----|------------------------|----------|----------|
| 1 | ID | name | lon | lat |
| 2 | 1 | sophia university | 139.7326 | 35.68356 |
| 3 | 2 | Imperial Palace | 139.7528 | 35.68518 |
| 4 | 3 | National Diet Building | 139.7449 | 35.67589 |



地震予知におけるCGMの可能性の検証

Consumer Generated Media

掲示板やクチコミサイトなど一般ユーザーが参加してコンテンツができていくメディア



2016/10/10 ~ 2016/12/10

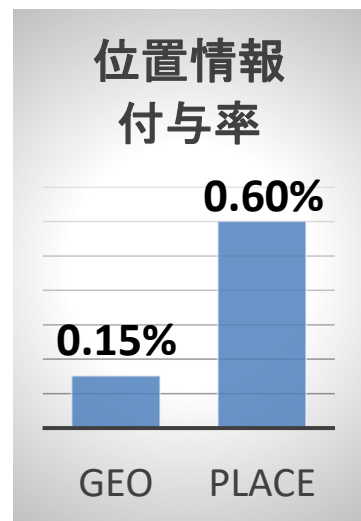
| 名称 | 内容 |
|-------------|-----------------|
| Coordinates | ツイートがされた緯度経度 |
| Geo | ツイートがされた場所の地理情報 |
| Place | Placeオブジェクト |
| Text | 本文 |
| Source | ツイートをを行ったクライアント |
| Created_at | ツイートされた日時 |

1. coordinates: ツイート投稿者の経度・緯度
2. geo: ツイートの緯度・経度
3. place: ツイート本文に含まれる位置情報

《Placeオブジェクトの構造》

| 名称 | 内容 |
|--------------|----------------|
| attributes | 住所などの追加情報 |
| Bounding_box | GeoJSON形式の座標情報 |
| country | 国名 |
| Country_code | 国名コード |
| Full_name | 場所の名称 |
| Name | 場所の短縮名称 |
| Place_type | 場所のタイプ |

| COL名 | 未分類 | 気象 | 地面 | 電波 | 動物 | 動物 2 | 人体 |
|------------|---------|-----------|--------|---------|-----------|---------|-----------|
| 検索ワード | 揺れ | 赤い夕陽 | 水量 | 電磁波障害 | 犬 | 鳥の群れ | 足が痛む |
| | 地鳴り | 発光現象 | 地鳴り | 電波 | イヌ | 大量発生 | 頭痛 |
| | 宏観異常 | 地震雲 | 地割れ | 電波ノイズ | 猫 | 大群 | 血の気 |
| | | 地震 | | | ネコ | | 吐き気 |
| | | 白い虹 | | | カラス | | 耳圧 |
| | | 熱水 | | | 烏 | | 調子が悪い |
| | | | | | 鯨 | | 体調が悪い |
| | | | | | ドジョウ | | 体調 |
| | | | | | ナマズ | | 気分が悪い |
| | | | | | モグラ | | |
| geo付与率 | 0.06% | 0.74% | 0.10% | 0.07% | 0.04% | 0.02% | 0.03% |
| place付与率 | 0.63% | 1.03% | 0.71% | 0.51% | 0.27% | 0.33% | 0.74% |
| 取得件数 | 512,159 | 2,062,325 | 45,225 | 622,852 | 8,791,356 | 155,818 | 2,773,466 |
| geo:null | 511,848 | 2,047,054 | 45,178 | 622,439 | 8,788,085 | 155,785 | 2,772,537 |
| geo有 | 311 | 15,271 | 47 | 413 | 3,271 | 33 | 929 |
| place:null | 508,922 | 2,041,035 | 44,904 | 619,699 | 8,767,840 | 155,307 | 2,753,035 |
| place有 | 3,237 | 21,290 | 321 | 3,153 | 23,516 | 511 | 20,431 |



| 回帰統計 | | | | | | | | |
|--------|----------|----------|----------|----------|-------------|----------|----------|----------|
| 重相関 R | 0.688164 | | | | | | | |
| 重決定 R2 | 0.47357 | | | | | | | |
| 補正 R2 | 0.282141 | | | | | | | |
| 標準誤差 | 0.437526 | | | | | | | |
| 観測数 | 16 | | | | | | | |
| 分散分析表 | | | | | | | | |
| | 自由度 | 変動 | 分散 | 有意 F | 観測された分散比 | | | |
| 回帰 | 4 | 1.894279 | 0.47357 | 0.105739 | 2.473864632 | | | |
| 残差 | 11 | 2.105721 | 0.191429 | | | | | |
| 合計 | 15 | 4 | | | | | | |
| | 係数 | 標準誤差 | t | P-値 | 下限 95% | 上限 95% | 下限 95.0% | 上限 95.0% |
| 切片 | -0.01132 | 0.593794 | -0.01907 | 0.985125 | -1.31826 | 1.295607 | -1.31826 | 1.295607 |
| 地鳴り | 0.041376 | 0.023766 | 1.740991 | 0.109543 | -0.01093 | 0.093684 | -0.01093 | 0.093684 |
| 受信 | 0.11997 | 0.139397 | 0.860639 | 0.407805 | -0.18684 | 0.426781 | -0.18684 | 0.426781 |
| カラス | 0.031469 | 0.011634 | 2.704838 | 0.020479 | 0.005862 | 0.057076 | 0.005862 | 0.057076 |
| 頭痛 | -0.00041 | 0.003893 | -0.10645 | 0.917144 | -0.00898 | 0.008154 | -0.00898 | 0.008154 |

地震震度3以上発生の有無

ロジスティック回帰分析



“カラス”に関するツイートが統計的有意 ($p < 0.05$)