

富士宮焼きそばを食べ歩きしたいのでマップを作った話















Python駿河 / Unagi.py 2021/08月

Hiroshi Sano

※なおこのスライドはJupyter + RISEで作成したのでどなたでも動作可能です

- 必要なパッケージをインストールしてください(venvやcondaで仮想環境用意)
 - Google Colabもパッケージを入れればいけると思います
 - `pip install requirements.txt`
- Plotly+MapBoxを使ったので、Mapboxアカウント作成、API利用登録=>トークンが必要になります
 - .envファイルに `MAPBOX_TOKEN=[token]` を入れてください
- 公開先: <https://github.com/hrsano645/fujinomiya-yakisoba-map>

お前誰よ

- 佐野浩士(Hiroshi Sano)@hrs_sano645 :静岡県の富士市 
- Job 
 - 佐野設計事務所    
 - 自動車系機械の3D設計事務所
ソフトウェアエンジニア他いろいろ
 - 米農家 
- Community  
 - : Python駿河, PyCon mini Shizuokaスタッフ
 - : PyCon JP 2020 チュートリアル講師
 -  : Code for ふじのくに

みなさんは富士宮焼きそば知ってますよね？



しんかわな - 投稿者が撮影, [CC 表示 3.0](#), [リンク](#)による

富士宮焼きそば

- 静岡県富士宮市で生まれた焼きそば。麺が太めで蒸し麺なのが特徴
 - キャベツ、豚の背脂を揚げた肉カスを入れて、ソースとだし粉をかけて食べるのがオーソドックス
- B級グルメの祭典、B-1グランプリ で殿堂入りなのでもはや説明いらないよね
- なお私の地元の富士市でも家庭料理でよく作ります
 - でも麺が別の地域だと手に入らない時がありますね

家で作った焼きそばの様子

- https://www.instagram.com/p/CRtD4QcpdQn/?utm_source=ig_embed&utm_campaign=loading

作り方

- 富士宮焼きそば麺を用意する
- フライパンで油を温める
- キャベツと肉カスを炒める
- 麺を入れる
- 水を入れて少し蒸し焼き
- ウスターソースと醤油を入れる(オイスターソース入れると美味しい)
- 水気を少し飛ばして皿に守る
- だし粉をかけて出来上がり

長年作った結果

はっきりいってこれ以上美味しい焼きそばの作り方が思いつかなくなる

(いわゆるマンネリ化です)

家の味に限界を感じる(伸び代がない

ではエンジニアならどうするか...

- 環境を変える: 転職したりしますよね
- 文献をあたる: 技術書を見たり

料理だと

- 環境を変える: お店で食べ歩きする
- 文献をあたる: レシピ本を見たり、動画を漁ったり

料理だと

- 環境を変える: **お店で食べ歩きする**
- 文献をあたる: レシピ本を見たり、動画を漁ったり

そうだ、食べ歩きをすればいいのでは👉

富士宮焼きそば食べ歩き計画

しかし現状(2021/8/28時点)は外に出れない状況



**不要不急の
外出は
控えてください**

外出ができる前に下準備をしよう

ということで、今日のテーマです

まず情報源

- 富士宮焼きそば学会

見てみるとこんな感じ

さて、これをどう調理するか。。。

ということで実演していきます

In [14]:

```
# 必要なパッケージ
import json
import os
import random
import time
import urllib.parse

import pandas
import plotly.express as px
import requests
from dotenv import load_dotenv
```

In [15]:

```
# トークン読み込み  
# .envにMAPBOX_TOKENを入れてください。  
load_dotenv()  
mapbox_token = os.environ["MAPBOX_TOKEN"]
```

```
# 焼きそばのリストをとってくる
# http://www.umya-yakisoba.com/contents/taberu/

yakisoba_dfs = pandas.read_html("http://www.umya-yakisoba.com/contents/taberu/")
print(len(yakisoba_dfs))

yakisoba_dfs[2]
```

Out[16]:

[illegible]

	番号	地図	店名	住所	電話番号 (0544)	営業時間(休憩時間含)	休日	受入人数	駐車場	焼き方	焼き方.1	料金
11	NaN	NaN	つかさ	上井出 276-7	54-0033	10:00-18:00	無休	36	5	○	NaN	650円
12	NaN	NaN	かいじ	上井出 141-1	54-2500	9:00-20:00	無休	18	10	○	NaN	400-600円
13	NaN	NaN	豊延	下条 1611-1	58-5454	11:00-14:0017:00-22:00	水曜日	25	10	○	NaN	350円大盛り+100円トッピング+50円しぐれ+100円
14	NaN	NaN	とおり道	源道寺 町1295	22-3039	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15	NaN	NaN	こころ	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
16	NaN	NaN	あおい	淀師 1117-6	22-3344	10:00-21:00	火曜日	24座敷有り 車いす可	10分以内 バス可	○	○	450-600円大盛り+100円もやし+50円
17	NaN	NaN	ページ	錦町8- 25	23-8101	11:30-19:00	日曜日	20	5	○	NaN	600円
18	NaN	NaN	ふじの食堂	万野腹 新田 2917	27-4305	11:00-14:0017:00-21:30	月曜日	カウンター6席 +24	8	○	NaN	950円
19	NaN	NaN	富士山に こここ長屋	山宮 517-2	58-6485	10:30-20:00	火曜日	70	30	○	NaN	500円
20	NaN	NaN	ふくじん	東町14- 3	27-7162	11:00-14:0016:30-21:00	月曜日	座敷12+15	2	○	NaN	500-600円
21	NaN	NaN	ふくいや	浅間町 8-5	26-4698	11:30-19:00	日曜日	15	5	○	NaN	450-700円大盛り+100円トッピング+50円
22	NaN	NaN	ひかり	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
23	NaN	NaN	さの	NaN	27-6981	11:00-19:00	水曜日	8	5	○	NaN	400円～
24	NaN	NaN	平石屋	上井出 261-1	54-0068	9:00-16:00	不定	70	6	○	NaN	630円
25	NaN	NaN	夢千代	東町11- 5	27-6179	11:00-24:00	不定	30座敷有り	20	○	NaN	380-600円

In [17]:

```
# 最初の二つは別の情報なので除外してます
# 全体を統合して前処理

yakisoba_all = pandas.concat(yakisoba_dfs[2:]).rename(columns={
    "店 名":"店名",
    "住 所":"住所",
    "電話番号(0544)":"電話番号",
    "料 金":"料金",
    "休 日":"休日",
    "焼き方":"焼き方:店",
    "焼き方.1":"焼き方:客"
})
# 必要ない列を除去
yakisoba_all = yakisoba_all.drop(columns=["番号", "地図"])
# 住所がNaNなもの除去
yakisoba_all = yakisoba_all.dropna(subset = ["住所"])
# 店名で重複除去
yakisoba_all = yakisoba_all.drop_duplicates(subset=["店名"])

# インデックスリセット
yakisoba_all = yakisoba_all.reset_index(drop=True)

# 住所に`静岡県富士宮市`を挿入, 電話番号の市外局番をつける
yakisoba_all["住所"] = yakisoba_all["住所"].apply(lambda x: f"静岡県富士宮市{x}")
yakisoba_all["電話番号"] = yakisoba_all["電話番号"].apply(lambda x: f"0544-{x}")
```

In [6]: # 富士宮焼きそば学会のお店一覧を作った
yakisoba_all

Out[6]:

	店名	住所	電話番号	営業時間(休憩時間含)	休日	受入人数	駐車場	焼き方:店	焼き方:客	料金
0	すぎ本(お宮横丁)	静岡県富士宮市宮町4-23	0544-24-8272	10:30-16:00	火曜日	お宮横丁	無	○	NaN	300～600円
1	お食事処岩市	静岡県富士宮市大宮町8-18	0544-27-7153	11:00-20:00	水曜日	28+座敷7間	2	○	NaN	600-780円
2	ままん	静岡県富士宮市大宮町5-17	0544-29-6288	11:00-14:0018:00-22:00	水・木曜日	26	無	○	NaN	600円
3	みっちゃん家	静岡県富士宮市宝町11-24	0544-27-5305	11:00-15:00(LO14:30)	水曜日第1・3日曜日	8+座敷2間	5	○	NaN	350-600円
4	わびすけ	静岡県富士宮市野中町806	0544-23-8578	11:30-14:0014:30-22:00(LO21:00)	水曜日	54	25	○	NaN	620-980円
...
147	大勝	静岡県富士宮市淀師136-8	0544-27-3968	10:00-20:00	月曜日	25	7	○	NaN	450～700円
148	うるおいてい	静岡県富士宮市淀師415-2	0544-24-7155	11:30-14:00	月曜日	50	15	○	○	350～800円
149	お好み食堂伊東	静岡県富士宮市淀師468-2	0544-27-6494	10:30-20:00	月曜、	50	20	○	NaN	350～600円
150	しまい	静岡県富士宮市淀師583-20	0544-22-2767	12:00-22:00	日曜日	15	8	○	NaN	350～550円
151	ひまわり	静岡県富士宮市若の宮町32	0544-26-3279	11:00-22:00	火曜日	10	3	○	NaN	400～650円

152 rows × 10 columns

地図にマッピングしてみます

- plotlyのmapbox埋め込み地図にポイント(散布図)を打つグラフで描写してみる
- そのために、住所から緯度経度を出す必要がある
- 今回は国土地理院のAPIを使いました。
 - GeoJsonという位置や道やエリアを定義するJSON形式
 - 国土地理院のサイトにこのAPIの説明がつけられてない...

In [8]:

```
def get_lon_lat(address: str) :  
    """  
        国土地理院のAPIを使って、  
        負荷防止のために、ランダムで問い合わせを行うようにしてます。  
  
        returnは(lat, lon) を返す  
  
        ref: https://memo.appri.me/programming/gsi-geocoding-api#%E3%82%B8%E3%82%AA%E3%82%B3%E3%83%BC%  
    """  
  
    # 住所文字をURLエンコード  
    encoding_address = urllib.parse.quote(address)  
  
    req_api = requests.get(f"https://msearch.gsi.go.jp/address-search/AddressSearch?q={encoding_addre  
  
    # 結果をprint  
    print(f"search: {address} -> {req_api.text}")  
  
    # GeoJsonからlon、latを得る  
    req_json = json.loads(req_api.text)  
    coordinates = req_json[0]["geometry"]["coordinates"]  
  
    # 問い合わせの時間をランダムですらす  
    time.sleep(random.randint(2, 5))  
  
    return (coordinates[1], coordinates[0])
```

In [9]:

```
# このコードは実行すると大体600秒ぐらいかかるので、LT当日は行わないように

# pandasにlon, latを入れる

# copy_yakisoba_all = yakisoba_all.copy()
# yakisoba_all["lon_lat"] = copy_yakisoba_all["住所"].map(lambda x: get_lon_lat(x))
# copy_yakisoba_all = yakisoba_all.copy()
# yakisoba_all["lon"] = copy_yakisoba_all["lon_lat"].map(lambda x: x[1])
# yakisoba_all["lat"] = copy_yakisoba_all["lon_lat"].map(lambda x: x[0])

# INFO:2021-08-26 copyを使うのがやや微妙感あるけど、警告出てたのでそうしました
```

In [10]:

```
# 上のセルの実行結果を保存しておいて
# yakisoba_all.to_csv("yakisoba_all_loclat.csv", encoding="utf-8")
# yakisoba_all.to_csv("yakisoba_all_loclat_sjis.csv", encoding="cp932")

# すでにlon、lat入りのcsvファイルで読み込みます

yakisoba_all = pandas.read_csv("yakisoba_all_loclat.csv")
```

In [11]:

```
# Plotly + scatter mapboxで地図にポイント
px.set_mapbox_access_token(mapbox_token)

fig = px.scatter_mapbox(yakisoba_all, lat="lat", lon="lon", hover_name="店名",
                        zoom=10, title="富士宮焼きそばマップ")
fig.show()
```

もうちょっときれいにしよう

- ラベルにお店情報が出せるようにする

In [12]:

```
fig = px.scatter_mapbox(yakisoba_all, lat="lat", lon="lon", hover_name="店名",  
                        hover_data=["住所", "休日", "駐車場", "料金"],  
                        zoom=9, title="富士宮焼きそばマップ")  
fig.show()
```

ここまでのまとめ

- `pandas.read_html`便利
- 富士宮焼きそばが食べられるお店は中心外以外でもあるが、車がないと行くのが難しい
- mapboxでプロットしたはいいものの、どうやってスマホとかで見ようか悩む

どうやってスマホで見るか🤔

↓

Googleマップに載せたい

Google マイマップでさっきのCSVファイルを取り込む

Googleマップにもマッピングできた

富士宮に行ったらお店をすぐに探せます😊

まとめ

富士宮焼きそばはうまいのでぜひ食べよう(県外の人向け)

まとめ

- pandasを駆使してデータを作ってオープンデータ化するのは面白い
- Plotly強い。便利。
- マップ系の連携とかもうちょっと勉強したい

もっとやりたかったこと

- 地図のシェープデータで富士宮の地域ごとのお店出現数のコロプレスマップを見たかった
 - ?
- 食べ歩きの最適なルート算出とかしてみたい
 - ?
- GeoPandasとか扱いたかった
 - 位置情報系興味あるので勉強していこうかと

もっとやりたかったこと

- 地図のシェープデータで富士宮の地域ごとのお店出現数のコロプレスマップを見たかった
 - 細かい地域のシェープデータはなさそうだった(探しきれてないかも)
- 食べ歩き最適なルート算出とかしてみたい
 - ネットワーク図というかダイクストラを解けば良さそう
 - それするならGoogleマップでやるかな...