What drives the VICTUALLING OFFICE!

Notebook for the IBM's Data Science Professional Certificate final project.

5 June 2021 **By Hein vd Westhuizen**

Introduction

Historical Background

- 1826 - A French lawyer, politician and gastronome Jean Anthelme Brillat publishes his work "Physiology of taste, or meditations on transcendental gastronomy". He coins the saying `“Dis-moi ce que tu manges, je te dirai ce que tu es - [tell me what you eat, and I will tell you what you are]”`.
- 1863 - A the German philosopher Ludwig Andreas Feurerbach publishes his essay "Spiritualism and Materialism", and states `“Der Mensch ist, was er ißt - [A man is what he eats]”`.
- 1942 - A renown English nutritionist Victor Lindlahr publishes his extremely successful book: `"You Are What You Eat"`.
- 2019 - "YouGov surveyed over 25,000 people from 24 global locations, asking them which out of 34 different national cuisines was their favorite. 84% of all surveyed ranked **Italian** food as their all-time fave, and honestly, we can't blame them. To break it down even more, 88% of Americans, 90% of Filipinos, and 85% of Japanese participants put Italian fist. Even 99% of Italians ranked their cuisine at the top." (Quote from https://soyummy.com/most-popular-cuisine/)

For almost a hundred years, mankind has been plagued by the question `"What is it that we eat?"`.

We are going to attempt to answer this question; to find out what **ingredient(s) are in our most beloved cuisine**.

Our approach will be:

1. To find the most popular places in the world we, as humans, visit like clockwork.
2. Once we know where we visit most often, we will build a picture of the type of cuisines in our most visited places.
3. Armed with the knowledge of the type of cuisines in our most visited places, we will establish the most popular cuisines in the most popular cities.

Problem Statement

What are the **MAIN INGREDIENT(s)** in the world's **MOST POPULAR CUISINES** in the world's **MOST POPULAR CITIES**?

Surely, to answer this we will have to:

1. Find the **most popular destinations** we visit.
2. Find the **most popular cities in each of the most popular destinations**.
3. Find the **most popular cuisines in the most popular cities in each of the most popular destinations**.
4. Find the **most popular ingredient(s) used in the most popular cuisines in the most popular cities in each of the most popular destinations**.

In summary then: WE NEED TO FIND THE MAIN INGREDIENT(s) IN MANKIND's FAVOURATE CUISINE

Whilst the aim of this analysis is to find the main ingredient(s) in our most favourable cuisine, it may be usefull to answer questions like why is poultry consumption increasing and beef consumption declining? Adding product pricing to the analysis clearly indicates that part of the move to poultry from beef is definitaly price-driven.

Data Definition

Finding the `'most popular destinations'` we visit.

We will scrape a Wikipedia page publishing the "most visited destinations" and wrangle the data, clean it, and then read it into a pandas dataframe so that it is in a structured format. Typically, the data should resemble a list like:

```
['France',
 'Spain',
 'United States',
 'China',
 'Italy',
 'Turkey',
 'Mexico',
 'Thailand',
 'Germany',
 'United Kingdom']
```

The Wiki-page can be found at: `'https://en.wikipedia.org/wiki/World_Tourism_rankings'`

Finding the `'most popular cities in each of the most popular cities'`.

For each country = "most visited destinations" found on the Wiki-page mentioned above, we'll use CountryInfo (A python module for returning data about countries, ISO info and states/provinces within them) to find the capital in the countries. We need the data in list format as in:

```
['France, Paris'],
 ['Spain, Madrid'],
 ['United States, Washington D.C.'],
 ['China, Beijing'],
 ['Italy, Rome'],
 ['Turkey, Ankara'],
 ['Mexico, Mexico City'],
 ['Thailand, Bangkok'],
 ['Germany, Berlin'],
 ['United Kingdom, London']
```

Finding the `'most popular cuisines in each of the most popular cities in each of the most popular cities'`

Our first objective is to obtain a categorized list of all world-cuisines from FourSquare. This can be done with the API https://api.foursquare.com/v2/venues/categories?&client_id={}&client_secret={}&v={}&limit={} and the processed data should be a dictionary of cuisines/id's e.g. `{'Afgan': '503288ae91d4c4b30a586d67'}`

Armed with the "most popular cities for the most visited destinations" and a dictionary of cuisines/id's, we extract the number of restaurants for each cuisine in each city for the most visited destinations using a Foursquare API https://api.foursquare.com/v2/venues/explore?&near={city}&categoryId={category_id}). Our objective is to generate a dataframe with columns `'City', 'Cuisine', 'Number of restaurants'`

Finding the `'ingredient(s) in the most popular cuisines in each of the most popular cities in each of the most popular cities'`

Using the acquired "restaurants by cuisine" information from FourSquare, we extract recipes for the relevant cuisine from the Kaggle dataset "Raw_recipes" (https://www.kaggle.com/.../food-com-recipes-and-user-interactions/version/2).

Answering the question `'WHAT IS/ARE THE MAIN INGREDIENT(s) IN MANKIND's FAVOURATE CUISINE?'`

Analysis of the recipes for the most popular cuisine reveals the core ingredient(s) we are after. We will assign word-frequencies to relevant words in the totallity of recipes for a given cuisine e.g.

`[('olive oil', 2791), ('tomato', 2241), ('chicken', 2058), ('cheese', 1767), ('oil', 1740), ('butter', 1733), ('garlic', 1298), ('flour', 1165), ('meat', 1091), ('egg', 1011)]` and use this frequency to pick the 'main ingredients'.

1. Setup

Import the needed libraries, setup the Foursquare API credentials.

Load & Import Libraries

In [4]:
```
import pandas as pd
import requests
from bs4 import BeautifulSoup
import matplotlib as mpl
import matplotlib.pyplot as plt
mpl.style.use(['seaborn'])
!pip install folium
import folium
import numpy as np
!pip install countryinfo
from countryinfo import CountryInfo

!pip install wordcloud
from wordcloud import WordCloud, STOPWORDS

import re
!pip install spacy
import spacy
from spacy.cli.download import download
download(model="en_core_web_sm")
nlp = spacy.load('en_core_web_sm',disable=['parser','ner'])

print('Libraries imported.')
```

```
Requirement already satisfied: folium in c:\users\conta\anaconda3\lib\site-packages (0.12.1)
Requirement already satisfied: numpy in c:\users\conta\anaconda3\lib\site-packages (from folium) (1.19.2)
Requirement already satisfied: requests in c:\users\conta\anaconda3\lib\site-packages (from folium) (2.24.0)
Requirement already satisfied: branca>=0.3.0 in c:\users\conta\anaconda3\lib\site-packages (from folium) (0.4.2)
Requirement already satisfied: jinja2>=2.9 in c:\users\conta\anaconda3\lib\site-packages (from folium) (2.11.2)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\users\conta\anaconda3\lib\site-packa
ges (from requests->folium) (1.25.11)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\conta\anaconda3\lib\site-packages (from requests->f
olium) (2020.6.20)
Requirement already satisfied: idna<3,>=2.5 in c:\users\conta\anaconda3\lib\site-packages (from requests->folium)
(2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\conta\anaconda3\lib\site-packages (from requests->fo
lium) (3.0.4)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\conta\anaconda3\lib\site-packages (from jinja2>=2.9->
folium) (1.1.1)
Requirement already satisfied: countryinfo in c:\users\conta\anaconda3\lib\site-packages (0.1.2)
Requirement already satisfied: wordcloud in c:\users\conta\anaconda3\lib\site-packages (1.8.1)
Requirement already satisfied: pillow in c:\users\conta\anaconda3\lib\site-packages (from wordcloud) (8.0.1)
Requirement already satisfied: numpy>=1.6.1 in c:\users\conta\anaconda3\lib\site-packages (from wordcloud) (1.19.
2)
Requirement already satisfied: matplotlib in c:\users\conta\anaconda3\lib\site-packages (from wordcloud) (3.3.2)
Requirement already satisfied: cycler>=0.10 in c:\users\conta\anaconda3\lib\site-packages (from matplotlib->wordc
loud) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\users\conta\anaconda3\lib\site-pack
ages (from matplotlib->wordcloud) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\conta\anaconda3\lib\site-packages (from matplotli
b->wordcloud) (2.8.1)
Requirement already satisfied: certifi>=2020.06.20 in c:\users\conta\anaconda3\lib\site-packages (from matplotlib
->wordcloud) (2020.6.20)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\conta\anaconda3\lib\site-packages (from matplotlib->
wordcloud) (1.3.0)
Requirement already satisfied: six in c:\users\conta\anaconda3\lib\site-packages (from cycler>=0.10->matplotlib->
wordcloud) (1.15.0)
Requirement already satisfied: spacy in c:\users\conta\anaconda3\lib\site-packages (3.0.6)
Requirement already satisfied: pydantic<1.8.0,>=1.7.1 in c:\users\conta\anaconda3\lib\site-packages (from spacy)
(1.7.4)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\conta\anaconda3\lib\site-packages (from spacy) (4.
50.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\conta\anaconda3\lib\site-packages (from spacy)
(2.24.0)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\conta\anaconda3\lib\site-packages (from spacy)
(3.0.5)
Requirement already satisfied: wasabi<1.1.0,>=0.8.1 in c:\users\conta\anaconda3\lib\site-packages (from spacy)
```