


Search Engine Land

- SEO
- SEM
- Local
- Retail
- Google
- Bing
- Social
- Resources
- Live
- More
- Events
- 

Search

MIT

in

MIT

MIT

f

MIT

t

MIT

MIT

MIT

SUBSCRIBE

Get the best search news each day. Enter your business email here.

SUBSCRIBE

SEO

SEM

Local

Retail

Google

Bing

Social

Resources

More

Newsletters

Home

Search

SUBMIT

Search Engine Land

Analytics & Conversion

Learn how to chart and track Google Trends in Data Studio using Python

By using the codes in this article, you can connect Google Spreadsheets and Jupyter Notebook to import data into Google Data Studio and easily share the analysis with your team.

Hülya Çoban on February 12, 2020 at 3:27 pm



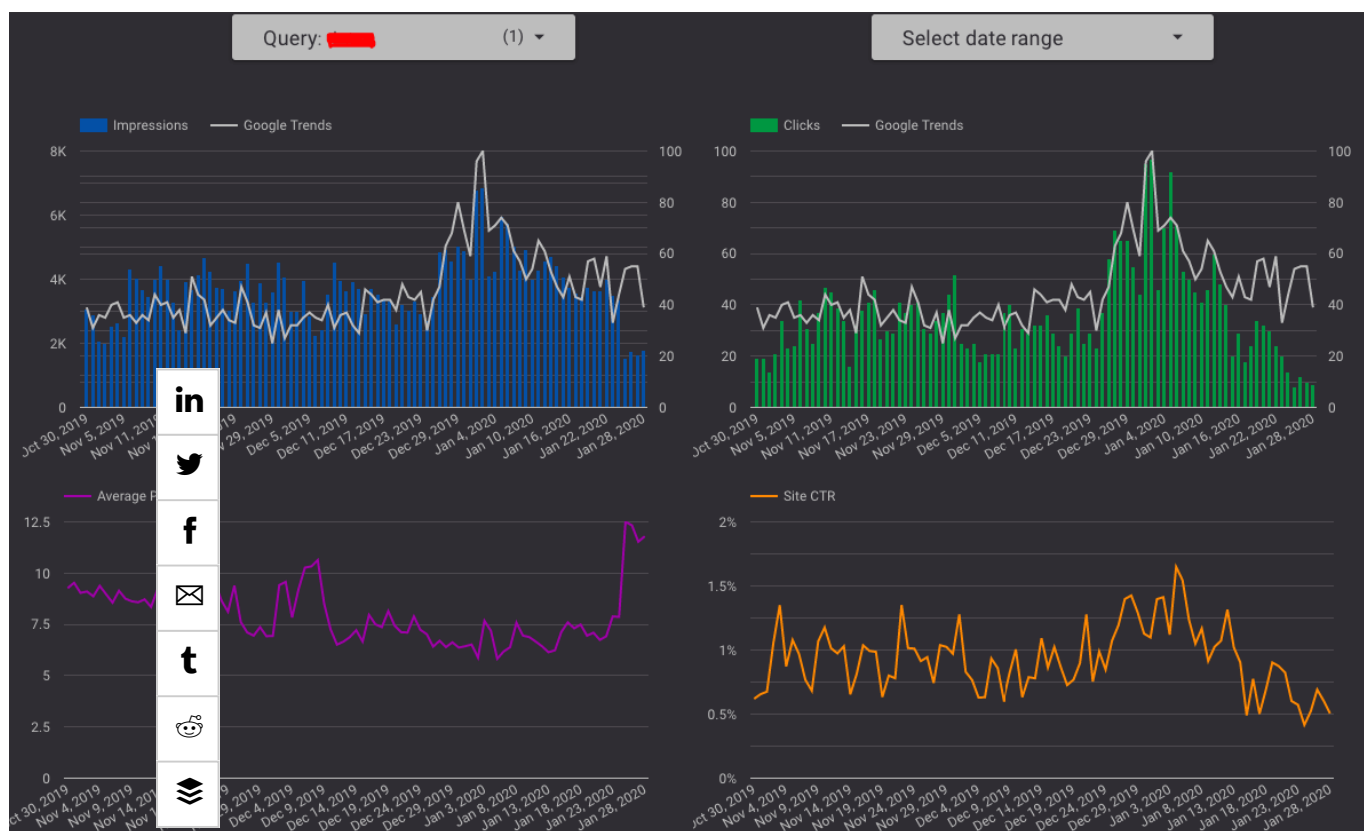
Google Trends is a free and incredibly useful tool that provides search interests, popular keywords and hot topics in a lot of languages for different platforms such as web search, Youtube or Google Shopping. Regardless of the marketing channel, it can be a very helpful tool to get valuable insights and make meaningful choices for the next steps of your project.

Basically, it gives the data on the relative popularity of a keyword from 2004 to the present, which is really cool! (*Relative popularity means the ratio of your search term interest to the interests of all keywords searched on Google.*)

Everything is great so far, but analyzing Google Trends data at scale is mostly not practical. Many of us don't use it much because it seems like a tedious job to search for

keywords on the website and get data points one by one. So how can we use Google Trends in a more effective way?

In this article, my aim is to show you the [pytrends](#) library in Python and what benefits you can get from it in your data analysis. I will also explain the connection between Google Spreadsheets and Jupyter Notebook in order to import data into Google Data Studio to share it with others easily. For example, while analyzing Search Console data on Data Studio dashboard, wouldn't it be nice to have Google Trends data on the same page? If your answer is yes, let's dig in!



3 topics I will cover in this article:

- Coding with Pytrends library and exploring its features
- Connecting Jupyter Notebook to Google Spreadsheets with gspread library
- Importing data into Google Data Studio

System requirements to use the Pytrends Library

- Python 2.7+ and Python 3.3+
- Requires Requests, lxml, Pandas libraries. If you don't know how to install libraries, check this [Python document](#). (hint: pip install pandas)

- Jupyter Notebook is an open source web application provides the environment to run your code.

Coding with Pytrends Library

First of all, you have to install the library:

```
pip install pytrends
```

Importing necessary libraries:

```
import pytrends
from pytrends.request import TrendReq
import pandas as pd
import time
import datetime
from datetime import datetime, date, time
```

Now it is time to code!

```
pytrend = TrendReq()
pytrend.build_payload(kw_list=['tea', 'coffee', 'coke', 'milk',
'water'], timeframe='today 12-m', geo = 'GB')
```

Payload function is important to specify your search. Write your keywords, decide date range, location and many other things like choosing Youtube or Shopping channel to analyze. In the code above, "today 12-m" means one year data. You can narrow your results by specifying location with 'geo'.

Let's say you have a Youtube channel and you only want to see Youtube search trends. Then your code will be like this:

```
pytrend.build_payload(kw_list=['tea', 'coffee', 'coke', 'milk',
'water'], timeframe='today 12-m', geo = 'GB', gprop= youtube)
```

Or let's assume that you have a food&drink blog and want to get trend data of your keywords in that category, not relative to all searches. Then it will be something like this:

```
pytrend.build_payload(kw_list=['tea', 'coffee', 'coke', 'milk',
'water'], timeframe='today 12-m', geo = 'GB', cat = 71)
```

In order to see all features and filters, you should check [this repository](#) on Github and also you can find all category codes in [here](#).

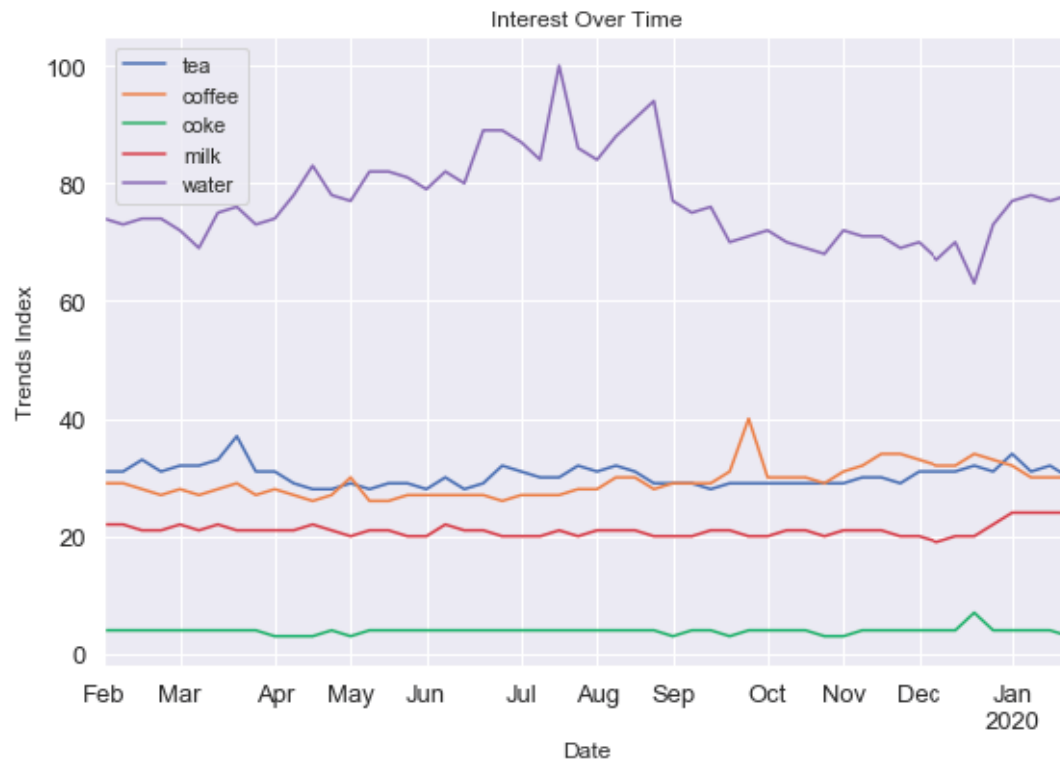
(By the way, be careful that you cannot write directly more than 5 keywords in here. It will give an error because you can compare only 5 keywords on Google Trends. I will use another code to analyze keywords more than 5.)

So, let's keep on and get trends score now.

```
#to get interest over time score, you'll need
pytrend.interest_over_time() function.
#For more functions, check this:
https://github.com/GeneralMills/pytrends
interest_over_time_df = pytrend.interest_over_time()
print(interest_over_time_df.head())
```

	tea	coffee	coke	milk	water	isPartial
date						
2019-02-03	31	29	4	22	74	False
2019-02-10	31	29	4	22	73	False
2019-02-17	33	28	4	21	74	False
2019-02-24	31	27	4	21	74	False
2019-03-03	32	28	4	22	72	False

```
in
# Let's
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style='whitegrid', palette='muted', font_codes=True)
dx = interest_over_time_df.plot(figsize = (9,6), title =
"Interest Over Time")
dx.set_xlabel('Date')
dx.set_ylabel('Trends Index')
dx.tick_params(axis='both', which='major', labelsize=13)
```



Suggested keywords

Now I will show you another cool feature of Google Trends. If you use the suggestion function, it will return with suggested keywords and their **"types."**

```
print(pytrend.suggestions(keyword='search engine land'), '\n')
print(pytrend.suggestions(keyword='amazon'), '\n')
print(pytrend.suggestions(keyword='cats'), '\n')
print(pytrend.suggestions(keyword='macbook pro'), '\n')
print(pytrend.suggestions(keyword='beer'), '\n')
print(pytrend.suggestions(keyword='ikea'), '\n')
```

```
[{'mid': '/m/0j1rt7p', 'title': 'Search Engine Land', 'type': 'Blog'}]

[{'mid': '/m/0mgkg', 'title': 'Amazon.com', 'type': 'E-commerce company'}, {'mid': '/m/0cx4p', 'title': 'Amazon Rainforest', 'type': 'Rainforest in Brazil'}, {'mid': '/g/11b6x7cx6n', 'title': 'Amazon Prime', 'type': 'Topic'}, {'mid': '/m/0gvlg', 'title': 'Amazon Prime Video', 'type': 'Television channel'}, {'mid': '/m/0sd7', 'title': 'Amazon River', 'type': 'River in South America'}]

[{'mid': '/m/01yrx', 'title': 'Cat', 'type': 'Animal'}, {'mid': '/m/01fhl', 'title': 'Cats', 'type': 'Musical by T. S. Eliot'}, {'mid': '/m/03071', 'title': 'Felidae', 'type': 'Animal'}, {'mid': '/m/03dkx', 'title': 'Geelong Football Club', 'type': 'Football club'}, {'mid': '/m/0ht80', 'title': 'Ketchup', 'type': 'Sauce'}]

[{'mid': '/m/09tzfp', 'title': 'Apple MacBook Pro', 'type': 'Laptop'}, {'mid': '/g/11bydj_78', 'title': 'MacBook Pro 13-inch', 'type': 'Laptop'}, {'mid': '/g/11g8lq9vs9', 'title': 'Apple MacBook Pro (13", 2017, Two Thunderbolt 3 Ports)', 'type': 'Laptop'}, {'mid': '/g/11bwlf16m3', 'title': 'Apple MacBook Pro (Retina, 15", Mid 2015)', 'type': 'Laptop'}, {'mid': '/g/11b6skc7__', 'title': 'Apple MacBook Pro (13", Mid 2010)', 'type': 'Laptop'}]

[{'mid': '/m/01599', 'title': 'Beer', 'type': 'Alcoholic drink'}, {'mid': '/m/068fpk', 'title': 'Drink coaster', 'type': 'Topic'}, {'mid': '/m/0r5_gvg', 'title': 'Heineken', 'type': 'Beer'}, {'mid': '/g/1215pllg', 'title': 'Craft beer', 'type': 'Topic'}, {'mid': '/m/01dws', 'title': 'Bear', 'type': 'Animal'}]

[{'mid': '/m/03tbl', 'title': 'IKEA', 'type': 'Furniture retail company'}, {'mid': '/g/11f3_5hs9d', 'title': 'IKEA', 'type': 'Topic'}, {'mid': '/g/1tscn519', 'title': 'IKEA', 'type': 'Furniture store in Breda, Netherlands'}, {'mid': '/g/1tjpppw', 'title': 'IKEA', 'type': 'Home goods store in Amsterdam, Netherlands'}, {'mid': '/g/1q62b6fp4', 'title': 'IKEA', 'type': 'Furniture store in Croatia'}]
```

Related queries

This is my favorite! Especially because it can be really helpful in Google Ads, keyword research and content creation.

Let's check "foundation" keyword in the Beauty category and get related keywords.

```
pytrend = pytrend.build_payload(kw_list=['foundation'], geo = 'US',
timeframe = 'today 3-m', cat = 44)
related_queries = pytrend.related_queries()
print(related_queries)
```



```

{'foundation': 'top':
0      best foundation      100
1      makeup foundation    62
2      powder foundation    42
3      foundation brush     24
4      maybelline foundation 23
5      maybelline           23
6      ulta foundation       21
7      sephora foundation    19
8      liquid foundation     19
9      cream foundation      19
10     drugstore foundation   18
11     mac foundation         18
12     face foundation        17
13     full coverage foundation 16
14     foundation stick       16
15     nars foundation        15
16     foundation for dry skin 15
17     foundation for oily skin 15
18     best foundation makeup 15
19     best drugstore foundation 14
20     estee lauder foundation 14
21     loreal foundation     13
22     covergirl             12
23     covergirl foundation   12
24     foundation match      12, 'rising':
0      wet n wild dewy foundation 8600
1      maybelline dream radiant liquid foundation 8400
2      milani screen queen foundation 7800
3      becca 24 hour foundation 700
4      becca cosmetics foundation 350
5      laura mercier flawless fusion foundation 200
6      tarte foundation brush 200
7      dermacol foundation 190
8      too faced dew you foundation 170
9      becca ultimate coverage foundation 160
10     concealer vs foundation 160
11     becca aqua luminous perfecting foundation 150
12     il makiage            130
13     bobbi brown intensive skin serum foundation 130
14     best foundation for older women 120
15     drugstore foundation for dry skin 110
16     il makiage foundation 100
17     physicians formula healthy foundation 100
18     some alexander airbrush foundation 100
19     best foundation at ulta 90
20     bobbi brown serum foundation 90
21     soap and glory foundation 80
22     best moisturizing foundation 70
23     stila foundation       70
24     becca foundation shades 70}}

```



You will see two parts in the output; top keywords and rising keywords. The value of top keywords shows Google Trends score from 0 to 100. However, the value of rising keywords shows how much interest in the keywords have increased in percentage.

If a website sells foundations, it would be great to follow what people are searching for lately, right? These products might be getting popular or reverse, they might have a bad reputation lately and that's why people might search for them. For instance, noticing this as soon as possible in Google Ads may prevent you from spending excessive amounts of money with no conversion.

Tracking lots of keywords

Now, I will write a group of random keywords here and get their data. You can also read keywords from a csv or excel file but make sure that its type must be a "list."


```

searches = ['detox', 'water fasting', 'benefits of fasting',
'fasting benefits',
'acidic', 'water diet', 'ozone therapy', 'colon hydrotherapy',
'water fast',
'reflexology', 'balance', 'deep tissue massage', 'cryo', 'healthy
body', 'what is detox',
'the truth about cancer', 'dieta', 'reverse diabetes', 'how to
reverse diabetes',
'water cleanse', 'can you drink water when fasting', 'water
fasting benefits', 'glycemic load', 'anti ageing', 'how to water
fast', 'ozone treatment', 'healthy mind', 'can you reverse
diabetes', 'anti aging', 'health benefits of fasting',
'hydrocolonic', 'shiatsu massage', 'seaweed wrap', 'shiatsu',
'can you get rid of diabetes', 'how to get rid of diabetes',
'healthy body healthy mind', 'colonic hydrotherapy', 'green
detox', 'what is water fasting', '21 day water fast', 'benefits
of water fasting', 'cellulite', 'ty bollinger', 'detox diet',
'detox in gram', 'anti aging treatments', 'ketogenic', 'glycemic
index', 'water fasting weight loss', 'keto diet plan', 'acidic
symptom', 'alkaline diet', 'water fasting diet', 'laser
therapy', 'anti cellulite massage', 'swedish massage', 'benefit
of fast', 'detox your body', 'colon therapy', 'reversing
diabete', 'detoxing', 'truth about cancer', 'how to remove
acidity', 'om body', '21 day water fast results', 'colon cleanse',
'fastin', 'ealth benefits', 'antiaging', 'aromatheraphy massage']

groupke, rds = list(zip(*[iter(searches)]*1))
groupkeywords = [list(x) for x in groupkeywords]

dicti = {}
i = 1
for trending in groupkeywords:
pytrend.build_payload(trending, timeframe = 'today 3-m', geo =
'GB')
dicti[i] = pytrend.interest_over_time()
i+=1

result = pd.concat(dicti, axis=1)
result.columns = result.columns.droplevel(0)
result = result.drop('isPartial', axis = 1)

```

result

Out[229]:

	detox	water fasting	benefits of fasting	fasting benefits	acidic	water diet	ozone therapy	hydrotherapy	colon	water fast	reflexology	...	benefit of fasting	detox your body	colon therapy
2019-10-31	35	30	27	31	48	25	0		99	15	50	...	0	0	0
2019-11-01	49	29	26	56	59	19	0		47	53	37	...	0	32	0
2019-11-02	36	32	27	46	45	34	35		48	46	64	...	0	32	89
2019-11-03	51	44	0	47	48	30	35		50	48	100	...	0	0	91
2019-11-04	42	81	38	64	58	28	66		45	54	70	...	0	30	0
...
2020-01-25	61	44	27	93	69	30	0		48	62	81	...	0	0	90
2020-01-26	61	56	0	33	59	71	33		0	76	66	...	0	61	0
2020-01-27	60	46	24	48	62	58	31		44	46	62	...	77	29	0
2020-01-28	53	71	75	57	74	37	0		45	64	55	...	0	30	0
2020-01-29	51	54	0	86	64	55	33		0	47	71	...	0	61	0

9 x 66 columns

Yes! I have some of them, but I need to reshape my data frame in case of merging this data with Search Console.

```
result.set_index(level=0, inplace=True)
pd.melt(result, id_vars='date', value_vars=searches)
```

Out[172]:

	date	variable	value
	2019-10-31	detox	35
1	2019-11-01	detox	30
2	2019-11-02	detox	35
3	2019-11-03	detox	52
4	2019-11-04	detox	41
...
6295	2020-01-24	aromatheraphy massage	0
6296	2020-01-25	aromatheraphy massage	0
6297	2020-01-26	aromatheraphy massage	0
6298	2020-01-27	aromatheraphy massage	0
6299	2020-01-28	aromatheraphy massage	0

6300 rows x 3 columns

```
result.to_excel('trends.xlsx')
```

Google Trends data is ready to go!

Connecting Jupyter Notebook to Google Spreadsheets with gspread library

First of all, you need to enable some APIs and create a secret client JSON file in order to authorize Google Sheets access. I will not explain this in this article, but [here is a great guide explaining how to do that step by step](#).

Then you can just use these codes below:

```
import gspread
from oauth2client.service_account import
ServiceAccountCredentials
links = ['https://spreadsheets.google.com/feeds',
'https://www.googleapis.com/auth/drive']
credentials =
ServiceAccountCredentials.from_json_keyfile_name('ENTER-YOUR-
JSON-FILE-NAME-HERE.json', links)
gc = gspread.inad.authorize(credentials)
```

Creating a spreadsheet and opening a spreadsheet:

```
sh = gc.create('My cool spreadsheet')
wks = gc.open("My cool spreadsheet").sheet1
# check out my lab documents here for more examples →
https://colab.research.google.com/notebooks/io.ipynb
```

Creating a custom formula to send data frames into sheets :

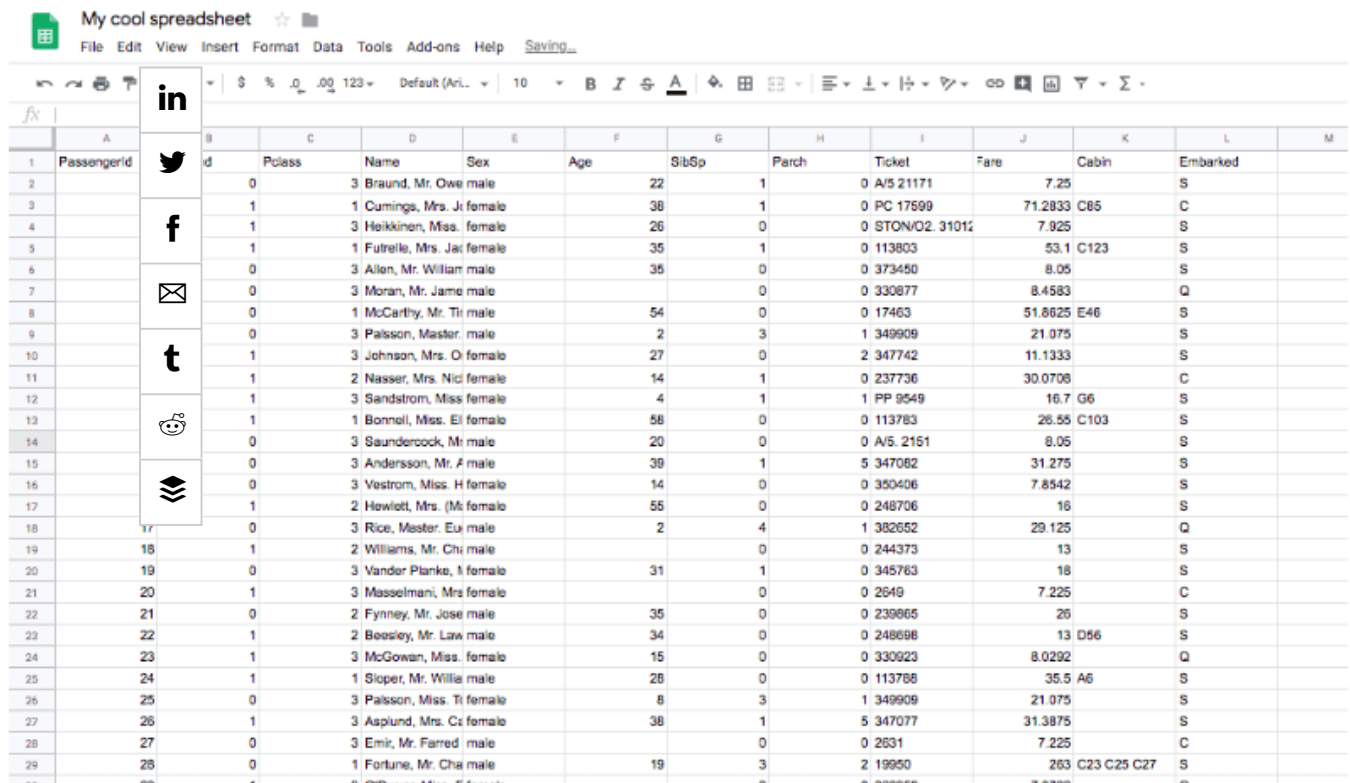
```
#https://www.danielecook.com/from-pandas-to-google-sheets/
```

```
def iter_pd(df):
for val in list(df.columns):
yield val
for row in df.values:
for val in list(row):
if pd.isna(val):
yield ""
else:
yield val
```

```
def pandas_to_sheets(pandas_df, sheet, clear = True):
# Updates all values in a workbook to match a pandas dataframe if
clear:
sheet.clear()
(row, col) = pandas_df.shape
cells = sheet.range("A1:
{}".format(gspread.utils.rowcol_to_a1(row + 1, col)))
for cell, val in zip(cells, iter_pd(df)):
cell.value = val
sheet.update_cells(cells)
```

An example to see how it works:

```
df = pd.read_csv("train.csv")
pandas_to_sheets(df, wks)
```



PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cummings, Mrs. J.	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss.	female	26	0	0	STON/O2. 31012	7.925		S
4	1	1	Futrelle, Mrs. J.	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, Mr. Tim	male	54	0	0	17463	51.6625	E46	S
8	0	3	Palsson, Master.	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. O.	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs. Nid	female	14	1	0	237736	30.0708		C
11	1	3	Sandstrom, Miss	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. El	female	58	0	0	113783	26.55	C103	S
13	0	3	Saunderscock, Mr	male	20	0	0	A/5. 2151	8.05		S
14	0	3	Andersson, Mr. A	male	39	1	5	347082	31.275		S
15	0	3	Vestrom, Miss. H	female	14	0	0	350406	7.8542		S
16	1	2	Hewlett, Mrs. (M)	female	55	0	0	248706	16		S
17	0	3	Rice, Master. Eu	male	2	4	1	382652	29.125		Q
18	1	2	Williams, Mr. Chi	male		0	0	244373	13		S
19	0	3	Vander Planke, M	female	31	1	0	345763	18		S
20	1	3	Masaelmani, Mrs	female		0	0	2649	7.225		C
21	0	2	Fynney, Mr. Jose	male	35	0	0	239665	26		S
22	1	2	Beesley, Mr. Law	male	34	0	0	248698	13	D56	S
23	1	3	McGowan, Miss.	female	15	0	0	330923	8.0292		Q
24	1	1	Sloper, Mr. Willie	male	28	0	0	113788	35.5	A6	S
25	0	3	Palsson, Miss. Ti	female	8	3	1	349909	21.075		S
26	1	3	Asplund, Mrs. C.	female	38	1	5	347077	31.3875		S
27	0	3	Emir, Mr. Farred	male		0	0	2631	7.225		C
28	0	1	Fortune, Mr. Cha	male	19	3	2	19950	263	C23 C25 C27	S

Let's continue with trends data and merge it with Search Console data.

```
sh = gc.create('GoogleTrends')
wks = gc.open("GoogleTrends").sheet1
pandas_to_sheets(result, wks)
```

```
dx = pd.read_excel('Trends.xlsx', sheet_name='Sheet1')
dz = pd.read_excel('Trends.xlsx', sheet_name = 'console') #my
console data is here, make sure where yours is
dm = pd.merge(dx, dz, on = ['Query', 'Date'])
dm
```

Out[222]:

	Date	Query	value	Impressions	Clicks	Site CTR	Average Position
0	2019-10-30	de:ox	39	1000	10	1.000000	1.000000
1	2019-10-31	de:ox	31	1000	10	1.000000	1.000000
2	2019-11-01	de:ox	36	1000	10	1.000000	1.000000
3	2019-11-02	de:ox	35	1000	10	1.000000	1.000000
4	2019-11-03	de:ox	40	1000	10	1.000000	1.000000
...
4916	2020-01-23	how to remove acidity from body	0	10	1	1.000000	1.000000
4917	2020-01-24	how to remove acidity from body	0	10	1	1.000000	1.000000
4918	2020-01-25	how to remove acidity from body	0	10	1	1.000000	1.000000
4919	2020-01-27	how to remove acidity from body	0	10	1	1.000000	1.000000
4920	2020-01-28	how to remove acidity from body	0	10	1	1.000000	1.000000

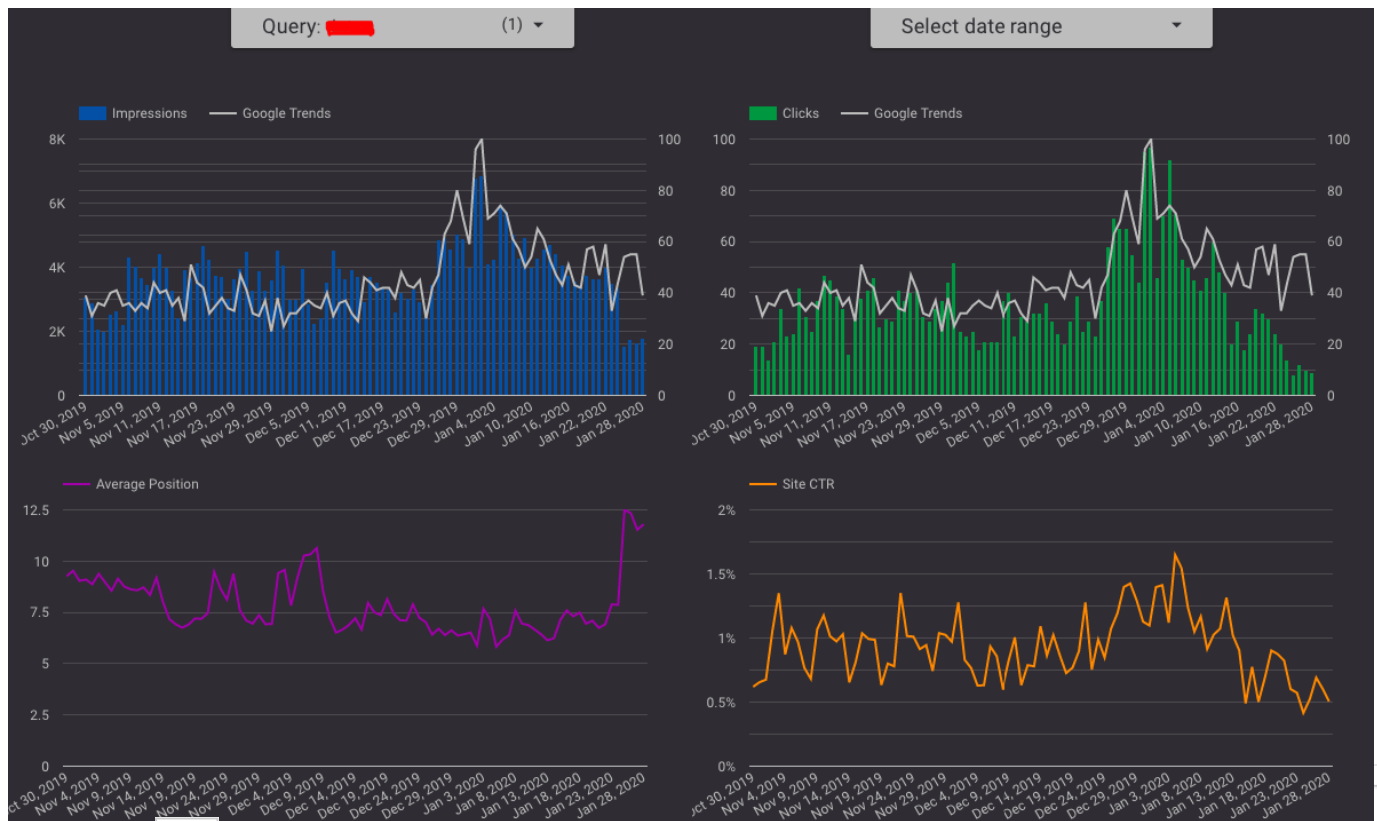
in rows x 7 columns

And let's s this one also into Google Sheets.

```
wks = g.open("GoogleTrends").sheet3
pandas_sheets(dm, wks)
```

Import data into Google Data Studio

Now you c ust connect this spreadsheet with Google Data Studio:



Tracking rising keywords

```
pytrend.build_payload(kw_list=['foundation', 'eyeliner',
'concealer', 'lipstick'], geo = 'US', timeframe = 'today 3-m',
cat = 4)
related_queries= pytrend.related_queries()
dg=related_queries.get('lipstick').get('rising')
dg
```

Out[330]:

	query	value
0	strawberry lipstick state of mind	3200
1	oryza lipstick	2300
2	ogee lipstick	350
3	louis vuitton lipstick case	300
4	doll 10 lipstick	250
5	givenchy black lipstick	200
6	mac lipstick set	200
7	mac sin lipstick	200
8	bellapierre lipstick	200
9	matte lipstick set	190
10	realher lipstick	170
11	mac kiss of stars lipstick	160
12	drake lipstick alley	160
13	gucci lipstick	150
14	jaclyn hill lipstick drama	120
15	givenchy black magic lipstick	120
16	molly ringwald lipstick	120
17	lipstick sets	120
18	ruby woo lipstick	90
19	lipstick set	90
20	mac ruby woo lipstick	90

in

twitter

f

✉

t

reddit

stack

Use *pandas* *to_sheets* again. Import these into Data Studio and visualize:

LIPSTICK RISING KEYWORDS			FOUNDATION RISING KEYWORDS		
query		Rising Percentage	query		Rising Percentage
strawberry lipstick state of mind		3.2K	maybelline dream radiant liquid foundation		15.5K
oryza lipstick		2.3K	milani screen queen foundation		11.8K
ogee lipstick		350	becca 24 hour foundation		500
louis vuitton lipstick case		300	becca cosmetics foundation		350
doll 10 lipstick		250	becca ultimate coverage foundation		250
givenchy black lipstick		200	dior airflash foundation		250
mac lipstick set		200	bobbi brown serum foundation		200
mac sin lipstick		200	stila hide and chic foundation		190
bellapierre lipstick		200	becca aqua luminous perfecting foundation		170
matte lipstick set		190	il makiage		130
realher lipstick		170	origins foundation		120
mac kiss of stars lipstick		160	becca foundation shades		120
drake lipstick alley		160	bourjois healthy mix foundation		110
gucci lipstick		150	tarte foundation brush		110
jaclyn hill lipstick drama		120	tarte amazonian clay foundation		100
givenchy black magic lipstick		120	how to apply liquid foundation		90

Wrapping up

It seems complicated at first, but just try these codes and create your own dashboards. Because at the end, you will just run the code on Jupyter Notebook and refresh the data on Google Data Studio. It will take only 10-15 seconds to update all of them, I promise!

[Here is my Github repository for all Python codes together.](#)

Happy coding!

This year's SMX Advanced will feature a brand-new SEO for Developers track with highly-technical sessions – many in live-coding format – focused on using code libraries and architecture models to develop applications that improve SEO. SMX Advanced will be held June 8-10 in Seattle. [Register today.](#)

Opinions expressed in this article are those of the guest author and not necessarily Search Engine Land. Staff authors are listed [here](#).

ABOUT THE AUTHOR



Hülya Çoban



Follow @hulyacobans



Hülya Çoban is a SEO analyst at [Keyphraseology](#) focusing on technical SEO and data-driven marketing. She believes that successful projects derive their strength from data so she always tries to read the story behind numbers. Follow her on Twitter for technical SEO, Python and data science.



We're listening.

Have something to say about this article? Share it with us on [Facebook](#), [Twitter](#) or our [LinkedIn Group](#).

ATTEND OUR EVENTS

SMX® Search Engine Land

April 13-15, 2021: [SMX Create](#)

- SEM

June 8-9, 2021: [SMX Paris](#)

- Local

June 15-16, 2021: [SMX Advanced](#)

- Retail
- Google

June 21-22, 2021: [SMX Advanced Europe](#)

- Bing

August 17, 2021: [SMX Convert](#)

- Social
- Resources

November 9-10, 2021: [SMX Next](#)

- More

December 14, 2021: [SMX Code](#)

- Events

Available On-Demand: [SMX](#)

- 

Available On-Demand: [SMX Report](#)



LEARN MORE ABOUT OUR SMX EVENTS

MARTECH®

Martech is marketing.

Discover actionable tactics that can help you overcome crucial marketing challenges. Our next conference will be held:

MarTech 2021: [March 16-17](#)

MarTech 2021: [Sept. 14-15](#)

MarTech 2020: [Watch On-Demand](#)



Engine Land

LEARN MORE ABOUT OUR MARTECH EVENTS

- Local

WHITE PAPERS

- Retail
- Google

Win With Faster Webpage Load Time

The Hidden CX Strategy: How Accessible Digital Experiences Unlock Wide-Ranging Growth

Content experience toolkit: 6 Priorities for digital decision-makers in 2021

Into the Wild – Digital Strategy for CMOs

The Six Principles of Building a Memorable Customer Experience

-

SEE MORE WHITEPAPERS

WEBINAR

Accelerate Your SEO & Content Marketing Program with 4 Key Milestones

Set Your Budget For PPC Success This Spring and Beyond

B2B Marketing Trends Shaping 2021

Get the best search news each day. Enter your business email here.

SEE MORE WEBINARS

SUBSCRIBE

RESEARCH REPORTS

Local Marketing Solutions for Multi-Location Businesses

Enterprise Digital Asset Management Platforms

Identity Resolution Platforms

Customer Data Platforms

B2B Marketing Automation Platforms

Call Analytics Platforms

SEE MORE RESEARCH

ATTEND SMX FOR ONLY \$99

\$99
April 13

Drive **awareness**,
traffic, and **sales**
with compelling
content.

in

The Agenda

Twitter

f

• SUBSCRIBE

Receive daily search news and analysis.

Enter your business email here.

SUBSCRIBE

SUBSCRIBE

Channels



- SEO
- SEM
- Local
- Retail
- Google
- Bing
- Social

Our Events

- SMX
- MarTech

Resources

- White Papers

- Research
- Webinars

About

- About Us
- Contact
- Privacy
- Marketing Opportunities
- Staff Retail
 - Google

Follow Us

-  Facebook
-  Twitter
-  LinkedIn
-  Newsletters
-  RSS
-  Youtube

Search Engine Land

Search

in

MIT

f

✉

t

👤

☰

• SUBSCRIBE

Get the best search news each day. Enter your business email here.

SUBSCRIBE