

```
Ввод [81]: import plotly
import plotly.graph_objects as go
from plotly.offline import download_plotlyjs, init_notebook_mode, iplot
import plotly.offline as pyo
from pytrends.request import TrendReq
import pytrends
import pandas as pd
import configparser
import logging
import sys
from logging import Formatter
from logging.handlers import RotatingFileHandler
import mysql.connector
from selenium_scrapper import selenium_scrapper
import mysql.connector
from selenium_scrapper import selenium_scrapper
import matplotlib.pyplot as plt
import time
import geopandas as gpd
import os
```

```
Ввод [ ]: # Пришлось закомментировать код с парсингом яндекса тк
# яндекс стал банить при использовании селениума и невозможно ничего сделать о
# Тем не менее в подтверждение можно изучить сохранившиеся при ранней работе с
```

```
Ввод [2]: config = configparser.RawConfigParser()
config.read('conf.ini', encoding='utf-8')

try:
    logger = logging.getLogger('logger')
    logger.setLevel(config.get('LOGGING', 'level'))
    handler = RotatingFileHandler(config.get('LOGGING', 'filename'), maxBytes=
    formatter = Formatter(fmt='[%asctime)s] %(levelname)s: %(message)s', date
    handler.setFormatter(formatter)
    logger.addHandler(handler)

    # получаем данные из файла настроек
    yandex_data = config.get('FOLDERS', 'yandex_data') # каталог для исходящи
    google_data = config.get('FOLDERS', 'google_data') # каталог для архивног
    yandex_login = config.get('TOKENS', 'yandex_log')
    yandex_password = config.get('TOKENS', 'yandex_pass')
    selenium_executor = config.get('FOLDERS', 'selenium_executor') # каталог
    create_db_sql = config.get('SQL', 'create_db') # путь до большого скрипта
    unis = config.get('SQL', 'unis')
    cnx = mysql.connector.connect(
        host=config.get('DB', 'host'),
        user=config.get('DB', 'user'),
        password=config.get('DB', 'sql_pass'),
        database="webscrapper"
    ) # конфиг для подключения к БД
except Exception as e:
    print('can not get params from ini, exit: {ex}'.format(ex=str(e)))
```

```
Ввод [82]: yandex_harvard = pd.read_csv(f"{yandex_data}{os.sep}harvard.csv")  
yandex_synergy = pd.read_csv(f"{yandex_data}{os.sep}университет синергия.csv")
```

Ввод [83]: yandex_synergy

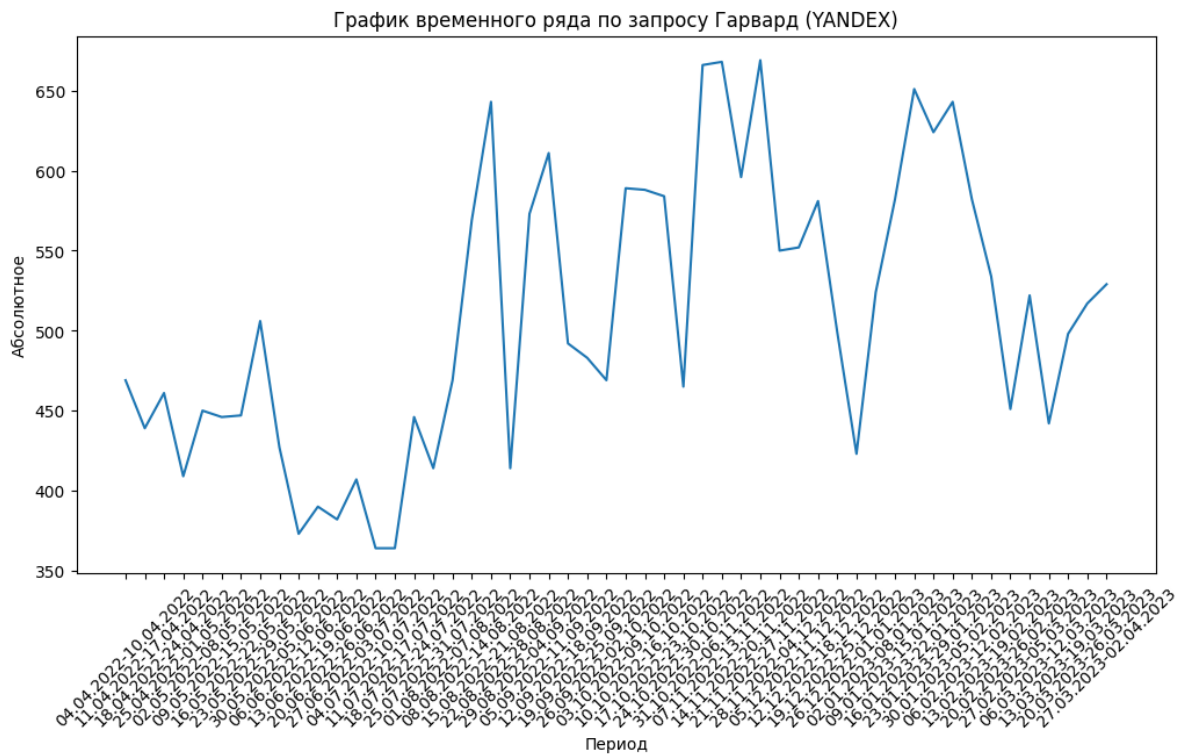
Out[83]:

	Период	Абсолютное	Относительное
0	11.04.2022-17.04.2022	11752.0	0,000004093262
1	18.04.2022-24.04.2022	11007.0	0,000003876341
2	25.04.2022-01.05.2022	9328.0	0,000003463586
3	02.05.2022-08.05.2022	7131.0	0,000002803944
4	09.05.2022-15.05.2022	8705.0	0,000003175733
5	16.05.2022-22.05.2022	10950.0	0,000003798015
6	23.05.2022-29.05.2022	12654.0	0,000004564023
7	30.05.2022-05.06.2022	12856.0	0,000004969486
8	06.06.2022-12.06.2022	14084.0	0,000005599495
9	13.06.2022-19.06.2022	16265.0	0,000006468331
10	20.06.2022-26.06.2022	20317.0	0,000007880064
11	27.06.2022-03.07.2022	26402.0	0,000010901169
12	04.07.2022-10.07.2022	23996.0	0,000010124737
13	11.07.2022-17.07.2022	20472.0	0,000008504137
14	18.07.2022-24.07.2022	18378.0	0,000007654844
15	25.07.2022-31.07.2022	18800.0	0,000007817999
16	01.08.2022-07.08.2022	14994.0	0,000006243179
17	08.08.2022-14.08.2022	15534.0	0,000006422591
18	15.08.2022-21.08.2022	15527.0	0,000006275721
19	22.08.2022-28.08.2022	13971.0	0,000005620802
20	29.08.2022-04.09.2022	17050.0	0,000006636685
21	05.09.2022-11.09.2022	15288.0	0,000005551176
22	12.09.2022-18.09.2022	13721.0	0,000004923219
23	19.09.2022-25.09.2022	17798.0	0,000006288445
24	26.09.2022-02.10.2022	28889.0	0,000010428942
25	03.10.2022-09.10.2022	22684.0	0,000008160075
26	10.10.2022-16.10.2022	31102.0	0,000011073773
27	17.10.2022-23.10.2022	20838.0	0,000007036340
28	24.10.2022-30.10.2022	13078.0	0,000004537661
29	31.10.2022-06.11.2022	11101.0	0,000003936646
30	07.11.2022-13.11.2022	12149.0	0,000004144269
31	14.11.2022-20.11.2022	11951.0	0,000003898830
32	21.11.2022-27.11.2022	11197.0	0,000003672043

	Период	Абсолютное	Относительное
33	28.11.2022-04.12.2022	10559.0	0,000003479465
34	05.12.2022-11.12.2022	10269.0	0,000003390013
35	12.12.2022-18.12.2022	9179.0	0,000003058515
36	19.12.2022-25.12.2022	8467.0	0,000002862296
37	26.12.2022-01.01.2023	6521.0	0,000002653440
38	02.01.2023-08.01.2023	5490.0	0,000002206385
39	09.01.2023-15.01.2023	11152.0	0,000003845212
40	16.01.2023-22.01.2023	12646.0	0,000004261770
41	23.01.2023-29.01.2023	12949.0	0,000004365392
42	30.01.2023-05.02.2023	12252.0	0,000004153168
43	06.02.2023-12.02.2023	12078.0	0,000004094096
44	13.02.2023-19.02.2023	11098.0	0,000003774715
45	20.02.2023-26.02.2023	9038.0	0,000003246987
46	27.02.2023-05.03.2023	11338.0	0,000003855451
47	06.03.2023-12.03.2023	13783.0	0,000004920085
48	13.03.2023-19.03.2023	11727.0	0,000004021056
49	20.03.2023-26.03.2023	11836.0	0,000004201193
50	27.03.2023-02.04.2023	11552.0	0,000004094096

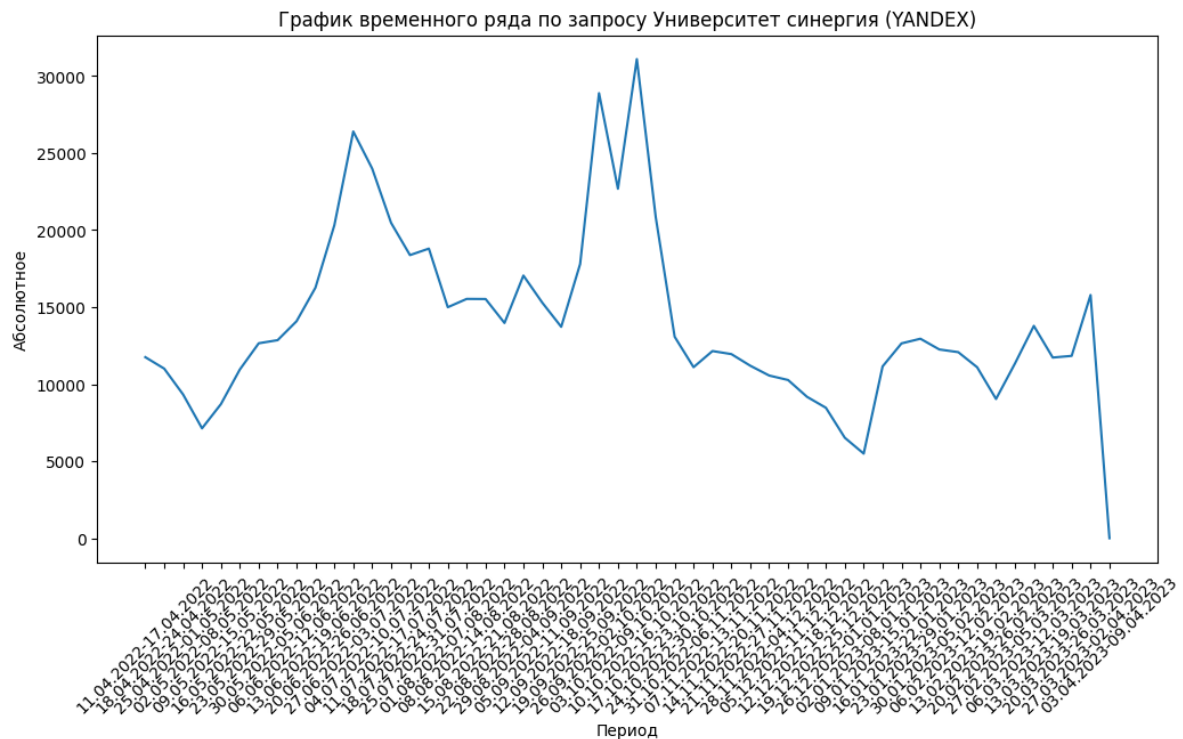
```
Ввод [87]: # график
plt.figure(figsize=(12, 6))
plt.plot(yandex_harvard['Период'], yandex_harvard['Абсолютное'])
plt.xticks(rotation=45)
# добавить заголовок и метки осей
plt.title('График временного ряда по запросу Гарвард (YANDEX)')
plt.xlabel('Период')
plt.ylabel('Абсолютное')

# сохранить график на компьютере
plt.show()
```



```
Ввод [88]: # график
plt.figure(figsize=(12, 6))
plt.plot(yandex_synergy['Период'], yandex_synergy['Абсолютное'])
plt.xticks(rotation=45)
# добавить заголовок и метки осей
plt.title('График временного ряда по запросу Университет синергия (YANDEX)')
plt.xlabel('Период')
plt.ylabel('Абсолютное')

# сохранить график на компьютере
plt.show()
```



```
Ввод [3]: cursor = cnx.cursor()

# Execute a query
query = """
SELECT * FROM
    webscrapper.universities

"""
cursor.execute(query)
```

```
Ввод [4]: sql_data = list(cursor.fetchall())
```

```
Ввод [5]: sql_data[1][0]
```

```
Out[5]: 'Университет боккони'
```

```
Ввод [12]: kw_list = []
```

```
Ввод [13]: for x in (sql_data):  
            kw_list.append(x[1])  
            print(x[0])  
            selenium_scrapper(selenium_executor,x[0],yandex_login, yandex_password, ya
```

Гарвардский университет
Университет боккони
Лондонская школа экономики и политических наук
Парижский университет
массачусетский технологический институт
Университет Стэнфорд
Венский университет
МГУ
МФТИ
МИФИ
ВШЭ
Университет Эрасмус

```

Ввод [21]: kw_group = list(zip(*[iter(kw_list)]*1))
kw_grplist = [list(x) for x in kw_group]
trendshow = TrendReq hl='en-US', tz=360
dict = {}
i = 0
for kw in kw_grplist:
    trendshow.build_payload(kw, timeframe = 'today 12-m')
    dict[i] = trendshow.interest_over_time()
    i += 1

trendframe = pd.concat(dict, axis=1)
trendframe.columns = trendframe.columns.droplevel(0)
trendframe = trendframe.drop('isPartial', axis = 1)
trendframe

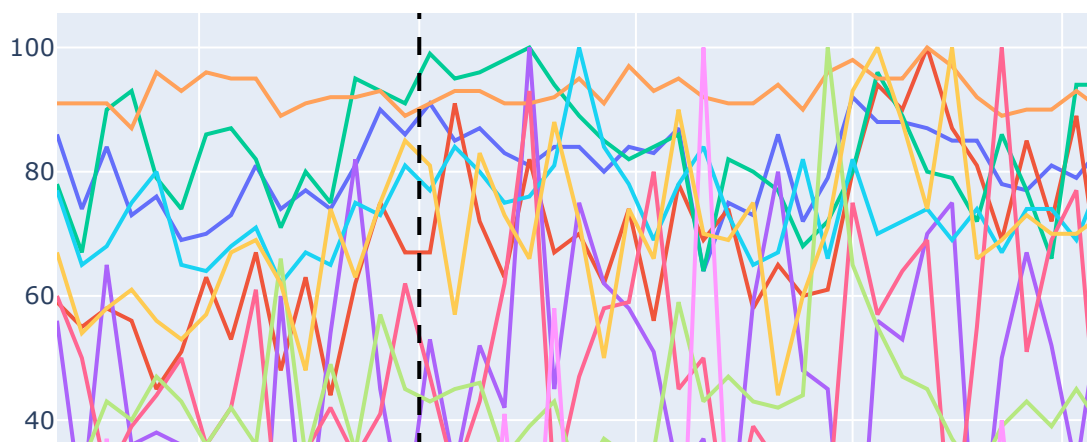
init_notebook_mode(connected=True)

trace = [go.Scatter(
    x = trendframe.index,
    y = trendframe[col], name=col) for col in trendframe.columns]

data = trace
layout = go.Layout(title='Популярность поисковых запросов к университетам 2022
fig = go.Figure(data=data, layout=layout)
fig.add_vline(x='2022-09-01', line_dash='dash', line_color='black')

```

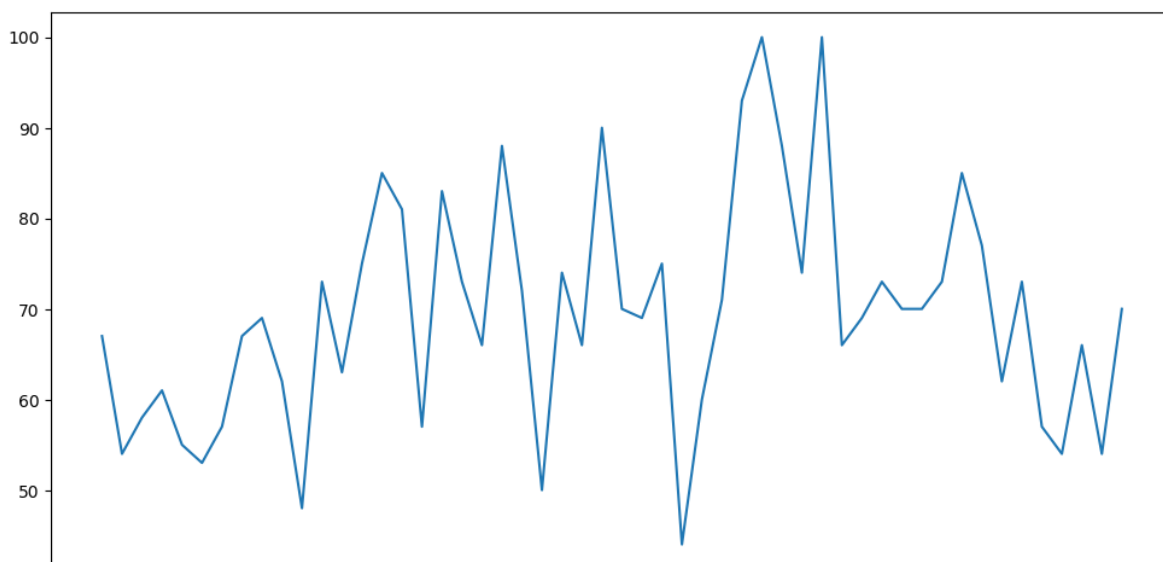
Популярность поисковых запросов к университетам 2022-09-




```
Ввод [56]: for univ in (sql_data):  
    print(x[1])  
    print("Дата начала приемной кампании "+x[3])  
    pytrends = TrendReq()  
    pytrends.build_payload(kw_list=[x[1]], timeframe = 'today 12-m')  
    interest_over_time_df = pytrends.interest_over_time()  
    data = pytrends.interest_over_time()  
    plt.figure(figsize=(12, 6))  
    plt.plot(interest_over_time_df.index, interest_over_time_df[x[1]])  
    plt.xticks(rotation=45)  
  
    plt.show()  
    # Сним чтобы гугл не забанил  
    time.sleep(3)
```

Erasmus university Rotterdam

Дата начала приемной кампании 2022-09-01



%%html

ТК объект карты в рамках одной ячейки не дает
вырисовать много отдельных карт то хардкодом
нарисуем их все

```

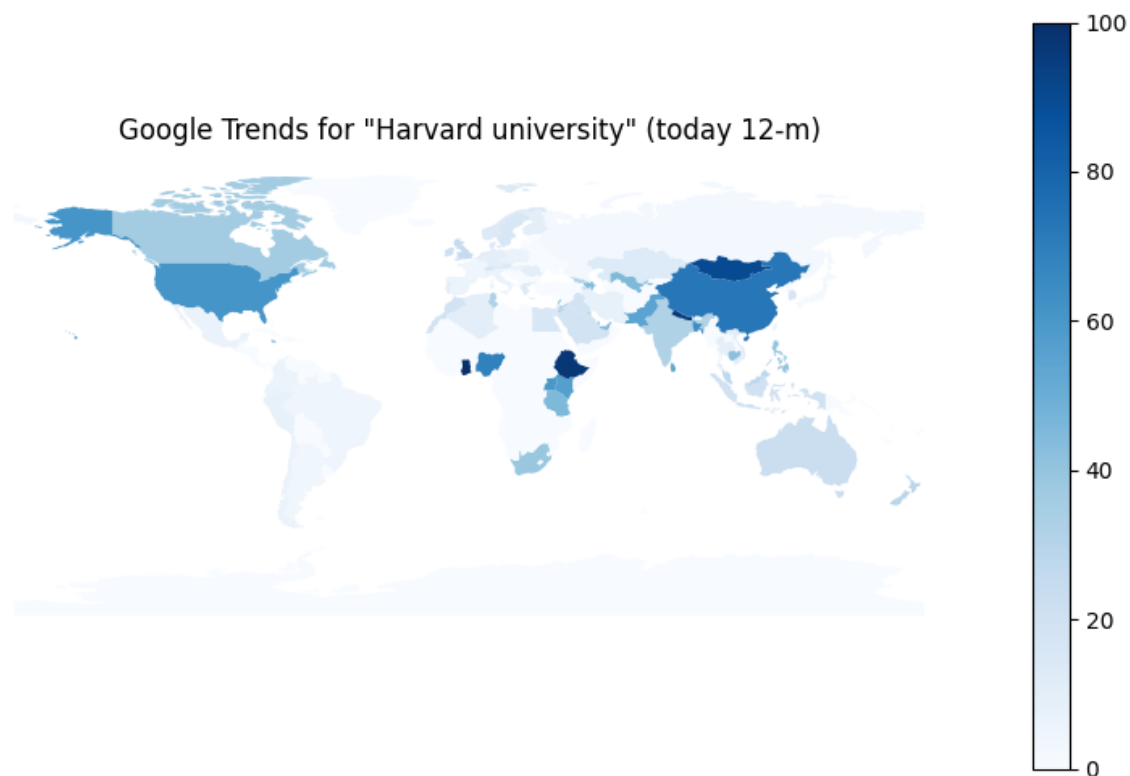
Ввод [67]: timeframe = 'today 12-m'
keyword = sql_data[0][1]
pytrends = TrendReq hl='en-US', tz=360)
pytrends.build_payload([keyword], cat=0, timeframe=timeframe, geo='')
data = pytrends.interest_by_region()
interest_over_time_df = pytrends.interest_over_time()
data = data.rename(index={"United States": "United States of America"})
# Меняем название
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world = world.merge(data, how='left', left_on='name', right_index=True)
world = world.fillna(0)
world.plot(column=keyword, cmap='Blues', legend=True, figsize=(10, 6))
plt.title(f'Google Trends for "{keyword}" ({timeframe})')
plt.axis('off')
fig = plt.gcf()

```

Harvard university

C:\Users\krezn\AppData\Local\Temp\ipykernel_21476\1704941736.py:10: FutureWarning:

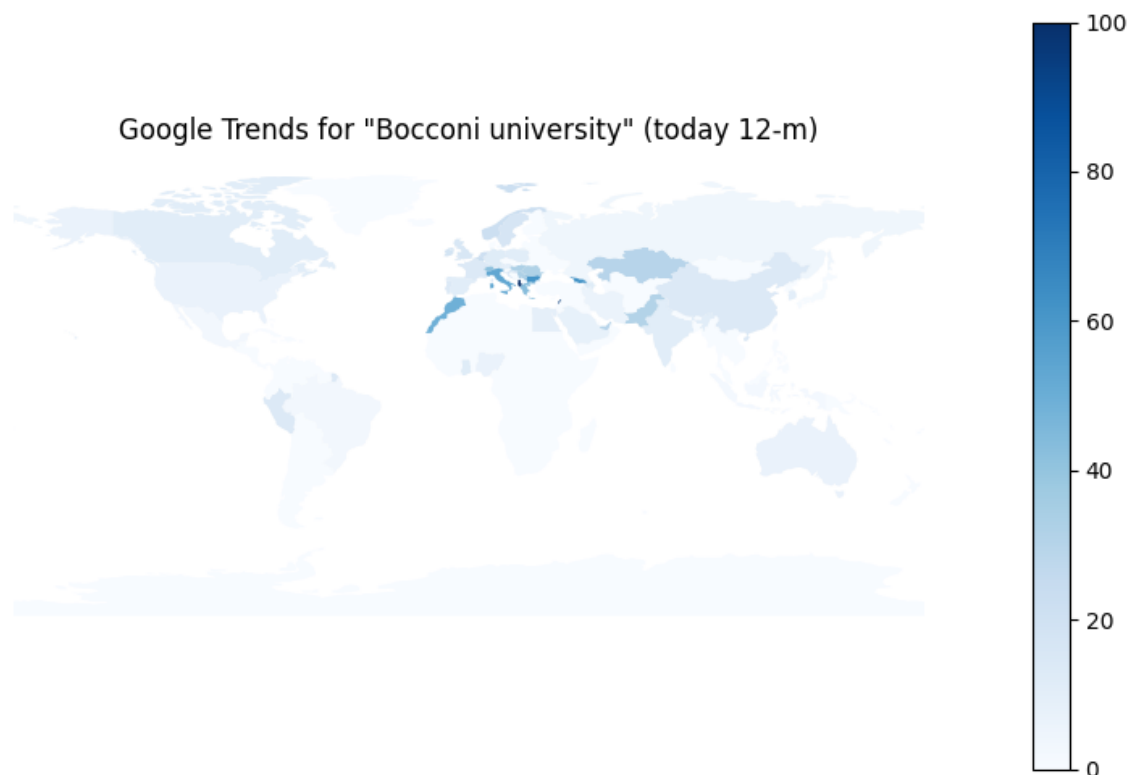
The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>. (<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.)



```
Ввод [68]: timeframe = 'today 12-m'
keyword = sql_data[1][1]
pytrends = TrendReq hl='en-US', tz=360)
pytrends.build_payload([keyword], cat=0, timeframe=timeframe, geo='')
data = pytrends.interest_by_region()
interest_over_time_df = pytrends.interest_over_time()
data = data.rename(index={"United States": "United States of America"})
# Меняем название
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world = world.merge(data, how='left', left_on='name', right_index=True)
world = world.fillna(0)
world.plot(column=keyword, cmap='Blues', legend=True, figsize=(10, 6))
plt.title(f'Google Trends for "{keyword}" ({timeframe})')
plt.axis('off')
fig = plt.gcf()
```

C:\Users\krezn\AppData\Local\Temp\ipykernel_21476\1404146409.py:9: FutureWarning:

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>. (<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.)



```

Ввод [69]: timeframe = 'today 12-m'
keyword = sql_data[2][1]
pytrends = TrendReq hl='en-US', tz=360)
pytrends.build_payload([keyword], cat=0, timeframe=timeframe, geo='')
data = pytrends.interest_by_region()
interest_over_time_df = pytrends.interest_over_time()
data = data.rename(index={"United States": "United States of America"})
# Меняем название
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world = world.merge(data, how='left', left_on='name', right_index=True)
world = world.fillna(0)
world.plot(column=keyword, cmap='Blues', legend=True, figsize=(10, 6))
plt.title(f'Google Trends for "{keyword}" ({timeframe})')
plt.axis('off')
fig = plt.gcf()

```

C:\Users\krezn\AppData\Local\Temp\ipykernel_21476\2311554800.py:9: FutureWarning:

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>. (<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.)



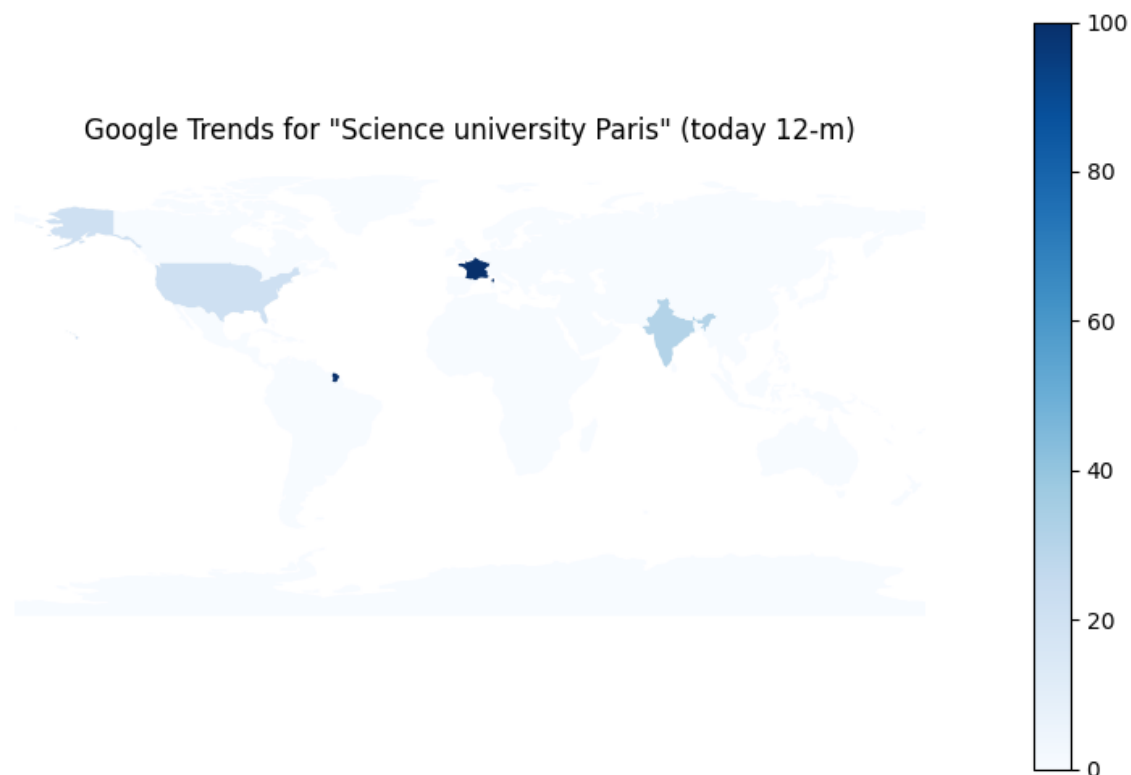
```

Ввод [70]: timeframe = 'today 12-m'
keyword = sql_data[3][1]
pytrends = TrendReq hl='en-US', tz=360)
pytrends.build_payload([keyword], cat=0, timeframe=timeframe, geo='')
data = pytrends.interest_by_region()
interest_over_time_df = pytrends.interest_over_time()
data = data.rename(index={"United States": "United States of America"})
# Меняем название
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world = world.merge(data, how='left', left_on='name', right_index=True)
world = world.fillna(0)
world.plot(column=keyword, cmap='Blues', legend=True, figsize=(10, 6))
plt.title(f'Google Trends for "{keyword}" ({timeframe})')
plt.axis('off')
fig = plt.gcf()

```

C:\Users\krezn\AppData\Local\Temp\ipykernel_21476\3032639717.py:9: FutureWarning:

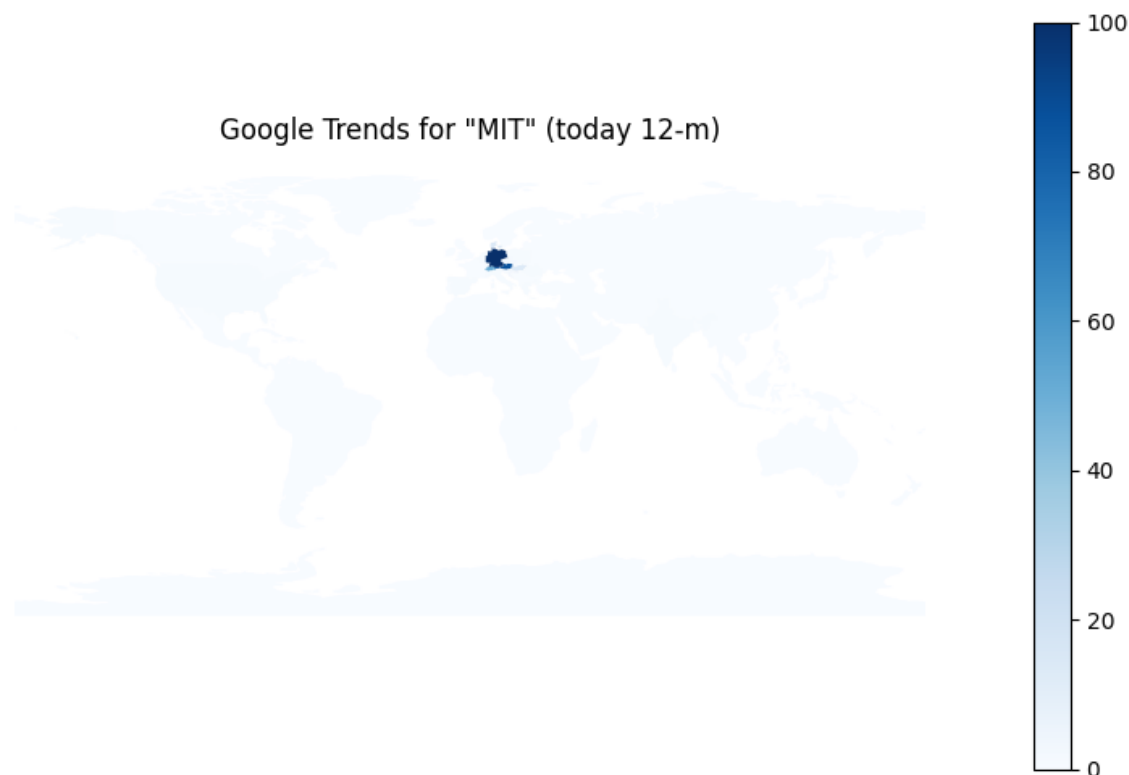
The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>. (<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.)



```
Ввод [71]: timeframe = 'today 12-m'
keyword = sql_data[4][1]
pytrends = TrendReq(hl='en-US', tz=360)
pytrends.build_payload([keyword], cat=0, timeframe=timeframe, geo='')
data = pytrends.interest_by_region()
interest_over_time_df = pytrends.interest_over_time()
data = data.rename(index={"United States": "United States of America"})
# Меняем название
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world = world.merge(data, how='left', left_on='name', right_index=True)
world = world.fillna(0)
world.plot(column=keyword, cmap='Blues', legend=True, figsize=(10, 6))
plt.title(f'Google Trends for "{keyword}" ({timeframe})')
plt.axis('off')
fig = plt.gcf()
```

C:\Users\krezn\AppData\Local\Temp\ipykernel_21476\3689273683.py:9: FutureWarning:

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>. (<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.)



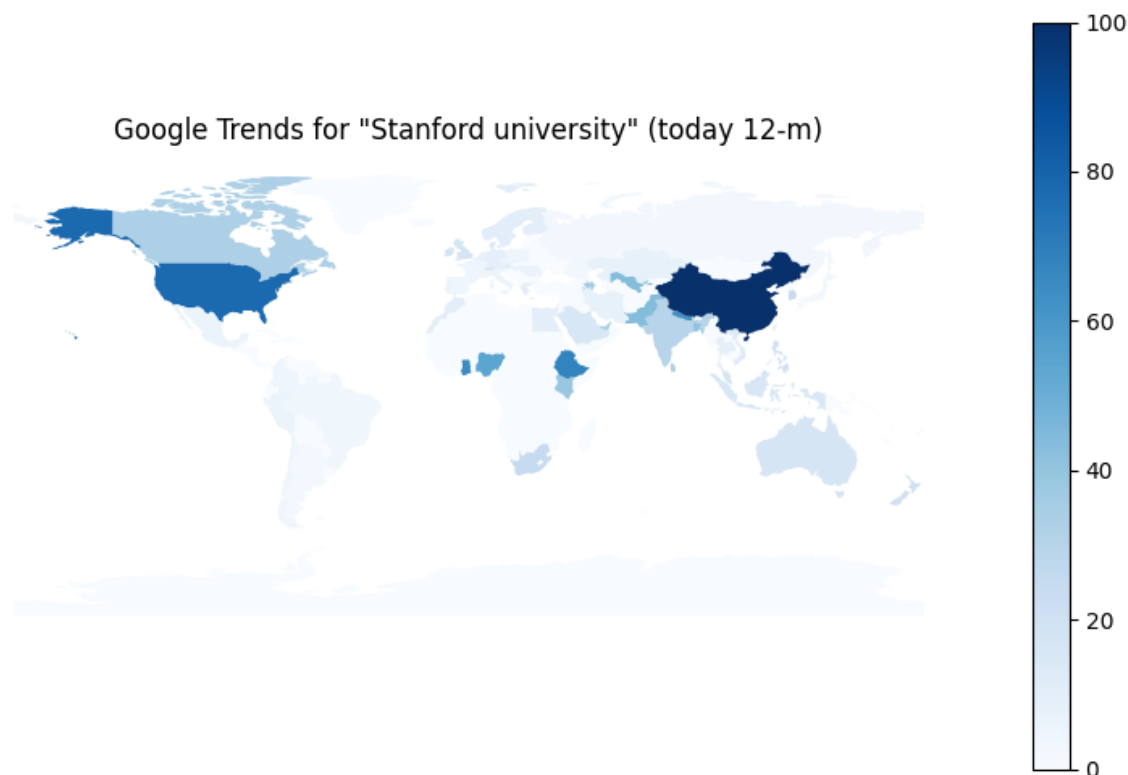
```

Ввод [72]: timeframe = 'today 12-m'
keyword = sql_data[5][1]
pytrends = TrendReq hl='en-US', tz=360)
pytrends.build_payload([keyword], cat=0, timeframe=timeframe, geo='')
data = pytrends.interest_by_region()
interest_over_time_df = pytrends.interest_over_time()
data = data.rename(index={"United States": "United States of America"})
# Меняем название
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world = world.merge(data, how='left', left_on='name', right_index=True)
world = world.fillna(0)
world.plot(column=keyword, cmap='Blues', legend=True, figsize=(10, 6))
plt.title(f'Google Trends for "{keyword}" ({timeframe})')
plt.axis('off')
fig = plt.gcf()

```

C:\Users\krezn\AppData\Local\Temp\ipykernel_21476\977618603.py:9: FutureWarning:

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>. (<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.)



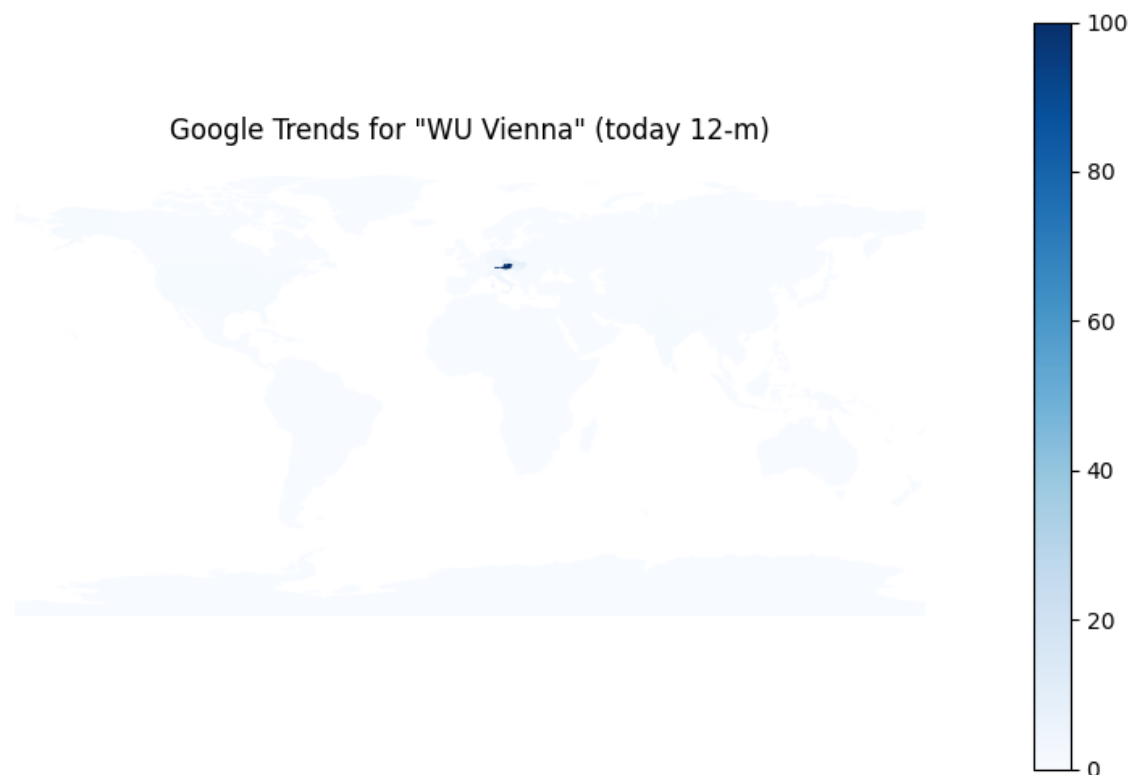
```

Ввод [73]: timeframe = 'today 12-m'
keyword = sql_data[6][1]
pytrends = TrendReq(hl='en-US', tz=360)
pytrends.build_payload([keyword], cat=0, timeframe=timeframe, geo='')
data = pytrends.interest_by_region()
interest_over_time_df = pytrends.interest_over_time()
data = data.rename(index={"United States": "United States of America"})
# Меняем название
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world = world.merge(data, how='left', left_on='name', right_index=True)
world = world.fillna(0)
world.plot(column=keyword, cmap='Blues', legend=True, figsize=(10, 6))
plt.title(f'Google Trends for "{keyword}" ({timeframe})')
plt.axis('off')
fig = plt.gcf()

```

C:\Users\krezn\AppData\Local\Temp\ipykernel_21476\241470280.py:9: FutureWarning:

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>. (<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.)



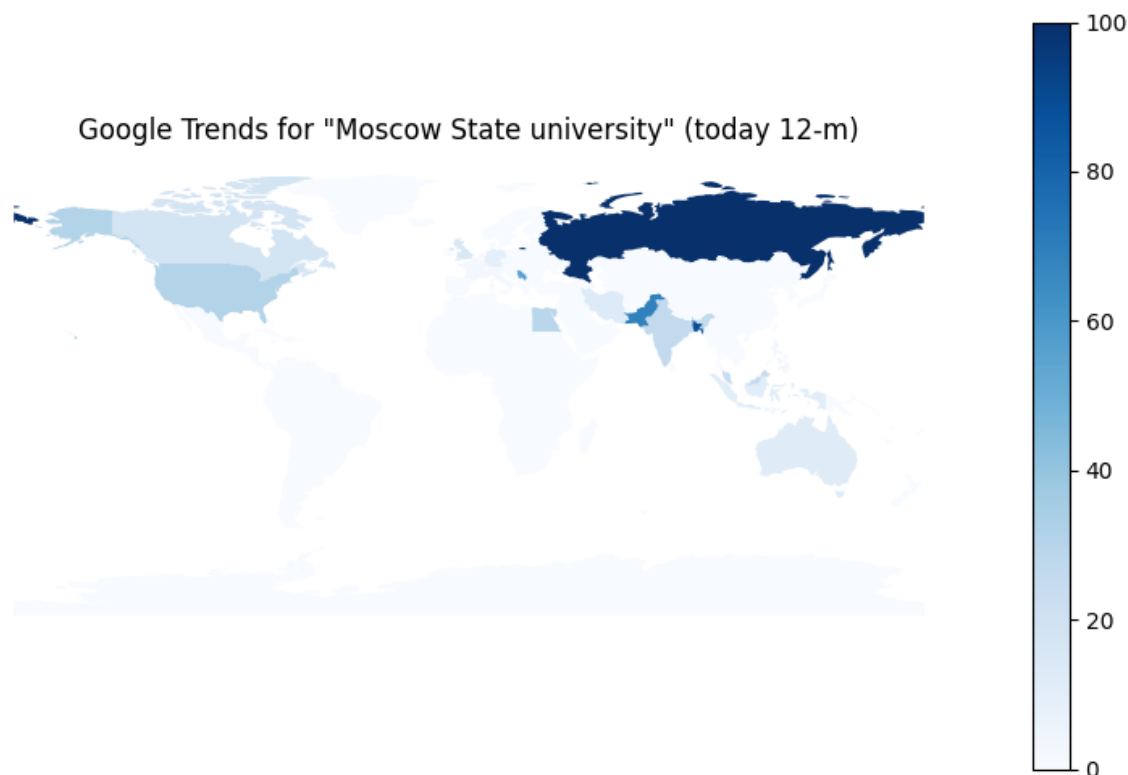

```

Ввод [74]: timeframe = 'today 12-m'
keyword = sql_data[7][1]
pytrends = TrendReq hl='en-US', tz=360)
pytrends.build_payload([keyword], cat=0, timeframe=timeframe, geo='')
data = pytrends.interest_by_region()
interest_over_time_df = pytrends.interest_over_time()
data = data.rename(index={"United States": "United States of America"})
# Меняем название
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world = world.merge(data, how='left', left_on='name', right_index=True)
world = world.fillna(0)
world.plot(column=keyword, cmap='Blues', legend=True, figsize=(10, 6))
plt.title(f'Google Trends for "{keyword}" ({timeframe})')
plt.axis('off')
fig = plt.gcf()

```

C:\Users\krezn\AppData\Local\Temp\ipykernel_21476\1029640309.py:9: FutureWarning:

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>. (<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.)



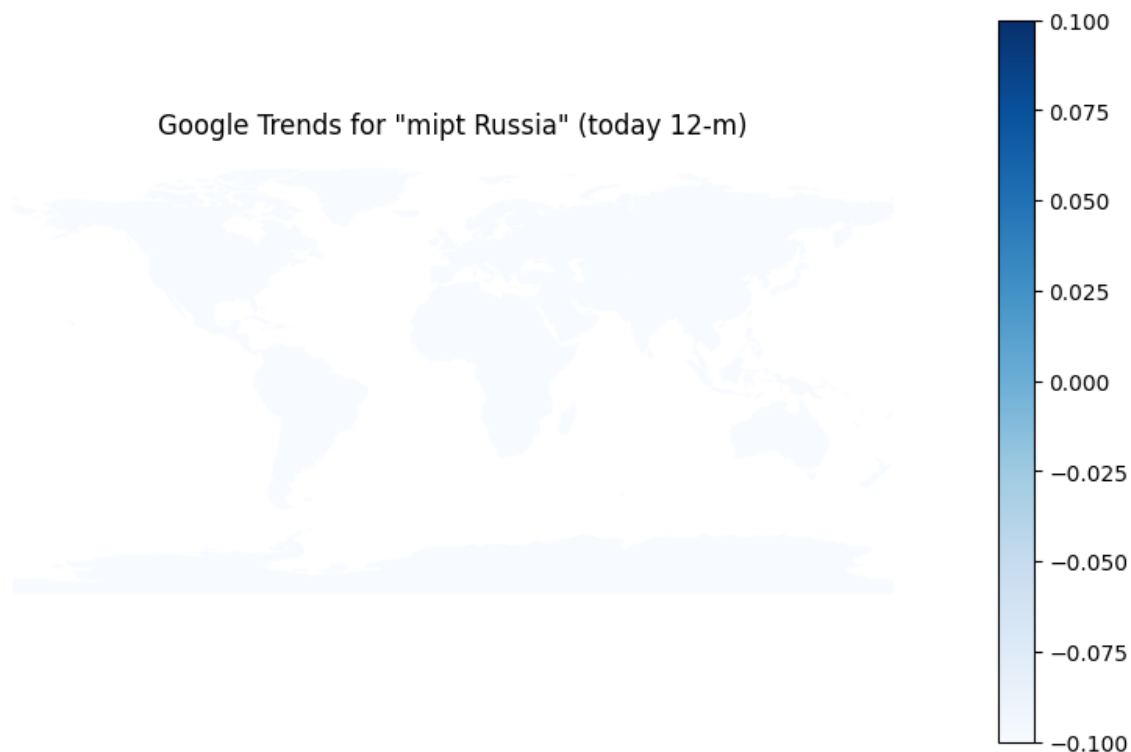
```

Ввод [76]: timeframe = 'today 12-m'
keyword = sql_data[8][1]
pytrends = TrendReq(hl='en-US', tz=360)
pytrends.build_payload([keyword], cat=0, timeframe=timeframe, geo='')
data = pytrends.interest_by_region()
interest_over_time_df = pytrends.interest_over_time()
data = data.rename(index={"United States": "United States of America"})
# Меняем название
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world = world.merge(data, how='left', left_on='name', right_index=True)
world = world.fillna(0)
world.plot(column=keyword, cmap='Blues', legend=True, figsize=(10, 6))
plt.title(f'Google Trends for "{keyword}" ({timeframe})')
plt.axis('off')
fig = plt.gcf()

```

C:\Users\krezn\AppData\Local\Temp\ipykernel_21476\1999017801.py:9: FutureWarning:

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>. (<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.)



```

Ввод [77]: timeframe = 'today 12-m'
keyword = sql_data[9][1]
pytrends = TrendReq hl='en-US', tz=360)
pytrends.build_payload([keyword], cat=0, timeframe=timeframe, geo='')
data = pytrends.interest_by_region()
interest_over_time_df = pytrends.interest_over_time()
data = data.rename(index={"United States": "United States of America"})
# Меняем название
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world = world.merge(data, how='left', left_on='name', right_index=True)
world = world.fillna(0)
world.plot(column=keyword, cmap='Blues', legend=True, figsize=(10, 6))
plt.title(f'Google Trends for "{keyword}" ({timeframe})')
plt.axis('off')
fig = plt.gcf()

```

C:\Users\krezn\AppData\Local\Temp\ipykernel_21476\3122429321.py:9: FutureWarning:

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>. (<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.)

Google Trends for "National Research Nuclear University Russia" (today 12-m)



```

Ввод [78]: timeframe = 'today 12-m'
keyword = sql_data[10][1]
pytrends = TrendReq hl='en-US', tz=360)
pytrends.build_payload([keyword], cat=0, timeframe=timeframe, geo='')
data = pytrends.interest_by_region()
interest_over_time_df = pytrends.interest_over_time()
data = data.rename(index={"United States": "United States of America"})
# Меняем название
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world = world.merge(data, how='left', left_on='name', right_index=True)
world = world.fillna(0)
world.plot(column=keyword, cmap='Blues', legend=True, figsize=(10, 6))
plt.title(f'Google Trends for "{keyword}" ({timeframe})')
plt.axis('off')
fig = plt.gcf()

```

C:\Users\krezn\AppData\Local\Temp\ipykernel_21476\2601618980.py:9: FutureWarning:

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>. (<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.)

Google Trends for "Higher School of Economics Russia" (today 12-m)



```

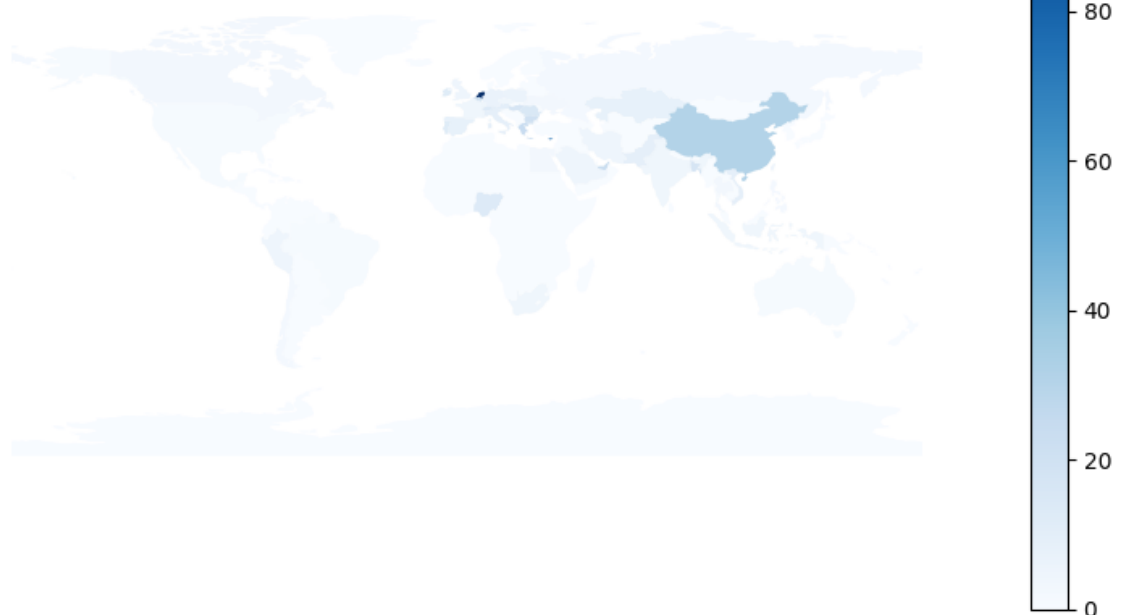
Ввод [79]: timeframe = 'today 12-m'
keyword = sql_data[11][1]
pytrends = TrendReq(hl='en-US', tz=360)
pytrends.build_payload([keyword], cat=0, timeframe=timeframe, geo='')
data = pytrends.interest_by_region()
interest_over_time_df = pytrends.interest_over_time()
data = data.rename(index={"United States": "United States of America"})
# Меняем название
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world = world.merge(data, how='left', left_on='name', right_index=True)
world = world.fillna(0)
world.plot(column=keyword, cmap='Blues', legend=True, figsize=(10, 6))
plt.title(f'Google Trends for "{keyword}" ({timeframe})')
plt.axis('off')
fig = plt.gcf()

```

C:\Users\krezn\AppData\Local\Temp\ipykernel_21476\2050079194.py:9: FutureWarning:

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>. (<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.)

Google Trends for "Erasmus university Rotterdam" (today 12-m)



```

Ввод [ ]: # закрываем коннект к БД
cursor.close()
cnx.close()

```

Ввод [90]:

```
# В целом подводя итог можно сказать что интерес к университетам стабильно дер.  
# и незначительно подрастает во время приемных кампаний  
# Так же можно сказать что Наиболее популярны в мире американские университеты  
# судя по поисковым запросам  
# Для того чтобы быстро проверить другие гипотезы и теории в удобном интерфейсе  
# Созданное мной стримлит приложение app.ru благодаря нему можно быстро изучит  
# даже по нескольким странам и по миру и по любым запросам (не только универси  
  
# для этого прочтите ридми и запустите приложение командой из ридми
```

Ввод []: