

UPA_project

December 7, 2020

1 UPA

Jakub Zárybnický (xzaryb00), Matěj Mlejnek (xmlejn04) - třetí člen se během semestr rozhodl s námi už dále nekomunikovat, pokračovali jsme ve dvou.

Pro správné fungování notebooku je potřeba mít v Jupyteru mít povolenou/nainstalovanou [integraci s Matplotlib](#) a v prostředí Python kernelu mít nainstalované balíčky: - psycopg2 - pymongo - pandas - matplotlib

2 PRO SPRÁVNÉ VYGENEROVÁNÍ OBRÁZKŮ POUŽÍJTE JUPYTER

```
[1]: %matplotlib widget
import sys; sys.path.insert(0, '.')

import csv
from datetime import date, datetime
import json
import os
import time

from bson.json_util import dumps
from dateutil.relativedelta import relativedelta
import matplotlib.pyplot as plt
import numpy as np
from psycopg2 import extensions
import pandas as pd
import pandas.io.sql as sqlio
import requests

from db_connects import MONGO_DB_CURRENCIES, MONGO_DB_COL_CURRENCIES,
    ↳connect_to_postgres, connect_to_mongodb
from scrape import parse
```

Některé součásti řešení zde nebudeme demonstrovat v celé délce, použijeme funkce pro zpracování vstupních dat nebo pro připojení k databázím, které máme předdefinované v našich knihovních souborech. Hlavní strukturu zde ale nastíníme, počínaje stáhnutím a zpracováním vstupních souborů.

2.1 Stažení zdrojových souborů

```
[2]: scrape_dir = 'scraped/'
    if not os.path.isdir(scrape_dir):
        os.mkdir(scrape_dir)
    len(os.listdir(scrape_dir))
```

[2]: 123

```
[3]: def scrape(base_url, output_dir, start_date, end_date):
    any_downloads = False
    for ordinal in range(start_date.toordinal(), end_date.toordinal()):
        url = base_url + date.fromordinal(ordinal).strftime('%d.%m.%Y')
        path = date.fromordinal(ordinal).strftime('%Y-%m-%d') + '.txt'
        filename = os.path.join(output_dir, path)
        if os.path.isfile(filename):
            continue
        any_downloads = True
        print("Requesting %s..." % url, end='')
        try:
            request = requests.get(url)
            if not request.text:
                print(' Empty!')
                continue
            print(' OK')
            with open(filename, 'w') as handle:
                handle.write(request.text)
            time.sleep(0.2)
        except Exception as ex:
            print(' %s' % ex)
            continue
    if not any_downloads:
        print("All files already present.")

start_date = datetime.today() - relativedelta(months=4)
end_date = datetime.today()
scrape(
    base_url='https://www.cnb.cz/cs/financni-trhy/devizovy-trh/
↳kurzy-devizoveho-trhu/kurzy-devizoveho-trhu/denni_kurz.txt?date=',
    start_date=start_date,
    end_date=end_date,
    output_dir=scrape_dir,
)
print("Got %s input files" % len(os.listdir(scrape_dir)))
print()
with open(scrape_dir + '/' + os.listdir(scrape_dir)[0], 'r') as f:
    print(f.read())
```

All files already present.
Got 123 input files

```
09.10.2020 #196
země|měna|množství|kód|kurz
Austrálie|dolar|1|AUD|16,526
Brazílie|real|1|BRL|4,120
Bulharsko|lev|1|BGN|13,862
Čína|žen-min-pi|1|CNY|3,430
Dánsko|koruna|1|DKK|3,643
EMU|euro|1|EUR|27,110
Filipíny|peso|100|PHP|47,490
Hongkong|dolar|1|HKD|2,966
Chorvatsko|kuna|1|HRK|3,579
Indie|rupie|100|INR|31,449
Indonesie|rupie|1000|IDR|1,563
Island|koruna|100|ISK|16,652
Izrael|nový šekel|1|ILS|6,802
Japonsko|jen|100|JPY|21,694
Jižní Afrika|rand|1|ZAR|1,395
Kanada|dolar|1|CAD|17,443
Korejská republika|won|100|KRW|2,007
Maďarsko|forint|100|HUF|7,610
Malajsie|ringgit|1|MYR|5,554
Mexiko|peso|1|MXN|1,081
MMF|ZPČ|1|XDR|32,441
Norsko|koruna|1|NOK|2,496
Nový Zéland|dolar|1|NZD|15,212
Polsko|zlotý|1|PLN|6,065
Rumunsko|leu|1|RON|5,565
Rusko|rubl|100|RUB|29,811
Singapur|dolar|1|SGD|16,958
Švédsko|koruna|1|SEK|2,602
Švýcarsko|frank|1|CHF|25,162
Thajsko|baht|100|THB|74,009
Turecko|lira|1|TRY|2,908
USA|dolar|1|USD|22,983
Velká Británie|libra|1|GBP|29,737
```

```
[4]: for input_file in parse(scrape_dir):
      print(input_file)
      break
```

```
{'date': datetime.datetime(2020, 10, 9, 0, 0), 'currency': {'country':
'Austrálie', 'name': 'dolar', 'code': 'AUD'}, 'lotSize': '1', 'price': '16,526'}
```

Nyní máme stažené všechny textové/CSV vstupní soubory a zpracované v takovém formátu, že je

můžeme přímo vložit do MongoDB bez dalšího zpracování. Do tohoto bodu se veškeré zpracování skládalo z načtení CSV souboru a přidání data ke každému řádku tak, se dá dále zpracovávat.

```
[5]: client = connect_to_mongodb()
collection = client[MONGO_DB_CURRENCIES][MONGO_DB_COL_CURRENCIES]
collection.drop()

res = collection.insert_many(parse(scrape_dir))
print("Loaded %s records to MongoDB" % len(res.inserted_ids))

collection.find_one()
```

Loaded 4059 records to MongoDB

```
[5]: {'_id': ObjectId('5fceb0e70595952406faa5b4'),
      'date': datetime.datetime(2020, 10, 9, 0, 0),
      'currency': {'country': 'Austrálie', 'name': 'dolar', 'code': 'AUD'},
      'lotSize': '1',
      'price': '16,526'}
```

Takto vypadají všechny záznamy v MongoDB. Nyní je převedeme to PostgreSQL, konkrétně do normalizovaného formátu ve dvou tabulkách, jedna tabulka měn a jedna tabulka kurzů.

- Měna = Kód měny (string, primární klíč) x Název (string) x Země (string)
- Kurz = Den (date) x Kód měny (cizí klíč) x Normalizovaný kurz (Float)

(Float sice není ideální reprezentace pro finanční výpočty, ale pro naše účely postačuje.)

```
[6]: conn = connect_to_postgres()

conn.set_isolation_level(extensions.ISOLATION_LEVEL_AUTOCOMMIT)

cursor = conn.cursor()
cursor.execute("DROP TABLE IF EXISTS kurz")
cursor.execute("DROP TABLE IF EXISTS mena")
cursor.execute("CREATE TABLE mena (zeme varchar(100), nazev varchar(100), kod_
↳varchar(10) primary KEY)")
cursor.execute(
    "CREATE TABLE kurz (den DATE, kod varchar(10), "
    "CONSTRAINT fk_mena FOREIGN KEY(kod) REFERENCES mena(kod) ON DELETE SET_
↳NULL, "
    "normalizovany_kurz FLOAT)"
)
```

```
[7]: mena_res = collection.find({}, {"currency": 1, "_id": 0}).distinct("currency")
for mena_item in mena_res:
    cursor.execute("INSERT INTO mena VALUES ('{}', '{}', '{}')".format(
        mena_item["country"],
        mena_item["name"],
```

```

        mena_item["code"]
    ))

for item in collection.find({}, {"_id": 0}):
    cursor.execute("INSERT INTO kurz VALUES ('{}', '{}', '{}')".format(
        item["date"].strftime("%Y-%m-%d"),
        item["currency"]["code"],
        float(item["price"].replace(',', '.')) / int(item["lotSize"])
    ))

```

```

[8]: cursor.execute("SELECT * from mena")
print("%s rows" % cursor.rowcount)
for row in cursor:
    print(row)
    break
print()
cursor.execute("SELECT * from kurz")
print("%s rows" % cursor.rowcount)
for row in cursor:
    print(row)
    break

```

33 rows
('Austrálie', 'dolar', 'AUD')

4059 rows
(datetime.date(2020, 10, 9), 'AUD', 16.526)

Nyní máme všechna data ve strukturované reprezentaci v PostgreSQL a můžeme se pustit do jednotlivých úkolů.

2.2 Úkol A

První úkol, který jsme si ze zadání vybrali, je vytvoření žebříčku měn, které v daném období nejvíce posílily/oslabily.

```

[9]: cursor.execute(
    "select kod, normalizovany_kurz from kurz where den = (SELECT MIN(den) from_
    ↪kurz)"
    " ORDER BY kod ASC"
)
min_hash = dict(cursor)
cursor.execute(
    "select kod, normalizovany_kurz from kurz where den = (SELECT MAX(den) from_
    ↪kurz)"
    " GROUP BY kod, normalizovany_kurz ORDER BY kod ASC"
)
diff = {}

```

```

for item in cursor:
    diff[item[0]] = min_hash[item[0]] - item[1]
diff = {k: v for k, v in sorted(diff.items(), key=lambda x: -x[1])}

fig = plt.figure()
x = np.arange(len(diff))
plt.bar(x, height=diff.values())
plt.xticks(x, diff.keys(), rotation=-90);

```

```

Canvas(toolbar=Toolbar(toolitems=[('Home', 'Reset original view', 'home',
↳ 'home'), ('Back', 'Back to previous ...

```

```

[10]: print("Between %s and %s the best performing currency was %s which changed by
↳ %s units." % (
    start_date.date(), end_date.date(), list(diff.items())[0][0],
↳ round(list(diff.items())[0][1], 2)
))

```

Between 2020-08-07 and 2020-12-07 the best performing currency was USD which changed by 0.31 units.

2.3 Úkol 2

Druhý úkol je nalezení skupin měn s podobným chováním (skupiny měn, které obvykle současně posilují/oslabují) pomocí korelační matice.

```

[11]: sql = "SELECT * FROM kurz ORDER BY den ASC"
df = sqlio.read_sql_query(sql, conn, parse_dates="den")
df = df.pivot_table(columns='kod', index="den", values="normalizovany_kurz")
df

```

```

[11]: kod      AUD      BGN      BRL      CAD      CHF      CNY      DKK      EUR \
den
2020-08-06  15.888  13.397  4.137  16.638  24.353  3.183  3.517  26.200
2020-08-07  16.020  13.436  4.155  16.669  24.326  3.196  3.528  26.280
2020-08-10  15.924  13.388  4.112  16.637  24.269  3.196  3.516  26.185
2020-08-11  15.941  13.371  4.085  16.706  24.344  3.197  3.512  26.155
2020-08-12  15.815  13.352  4.095  16.692  24.280  3.194  3.507  26.115
...
2020-11-30  16.123  13.391  4.122  16.878  24.162  3.325  3.520  26.190
2020-12-01  16.121  13.416  4.128  16.900  24.216  3.336  3.525  26.240
2020-12-02  16.121  13.501  4.195  16.909  24.406  3.335  3.548  26.410
2020-12-03  16.150  13.509  4.193  16.846  24.394  3.322  3.549  26.420
2020-12-04  16.184  13.558  4.224  16.965  24.511  3.340  3.563  26.520

kod      GBP      HKD      ...      PLN      RON      RUB      SEK      SGD      THB \
den
...

```

2020-08-06	29.101	2.855	...	5.945	5.417	0.30138	2.540	16.138	0.71209
2020-08-07	29.080	2.870	...	5.962	5.432	0.30212	2.547	16.220	0.71388
2020-08-10	29.046	2.872	...	5.948	5.415	0.30235	2.547	16.207	0.71482
2020-08-11	29.110	2.864	...	5.942	5.409	0.30522	2.543	16.188	0.71480
2020-08-12	28.866	2.862	...	5.931	5.401	0.30258	2.548	16.153	0.71302
...
2020-11-30	29.148	2.820	...	5.858	5.374	0.28732	2.573	16.340	0.72233
2020-12-01	29.213	2.828	...	5.860	5.387	0.28906	2.569	16.348	0.72461
2020-12-02	29.177	2.823	...	5.904	5.420	0.28919	2.566	16.323	0.72402
2020-12-03	29.246	2.806	...	5.906	5.422	0.29087	2.564	16.284	0.72074
2020-12-04	29.375	2.814	...	5.924	5.443	0.29444	2.585	16.366	0.72329

kod	TRY	USD	XDR	ZAR
den				
2020-08-06	3.059	22.125	31.240	1.262
2020-08-07	3.091	22.241	31.423	1.269
2020-08-10	3.043	22.262	31.411	1.256
2020-08-11	3.062	22.194	31.223	1.268
2020-08-12	3.034	22.185	31.273	1.273
...
2020-11-30	2.811	21.861	31.158	1.422
2020-12-01	2.787	21.922	31.351	1.431
2020-12-02	2.793	21.887	31.277	1.423
2020-12-03	2.783	21.748	31.143	1.425
2020-12-04	2.803	21.814	31.341	1.436

[84 rows x 33 columns]

```
[12]: corr = df.corr()
corr
```

```
[12]: kod      AUD      BGN      BRL      CAD      CHF      CNY      DKK  \
kod
AUD    1.000000  0.830581  0.469135  0.864231  0.777882  0.803441  0.834628
BGN    0.830581  1.000000  0.218837  0.950046  0.984522  0.833479  0.999578
BRL    0.469135  0.218837  1.000000  0.209070  0.144362  0.235185  0.227533
CAD    0.864231  0.950046  0.209070  1.000000  0.938080  0.908475  0.950105
CHF    0.777882  0.984522  0.144362  0.938080  1.000000  0.801546  0.983113
CNY    0.803441  0.833479  0.235185  0.908475  0.801546  1.000000  0.829084
DKK    0.834628  0.999578  0.227533  0.950105  0.983113  0.829084  1.000000
EUR    0.831635  0.999879  0.221910  0.950551  0.984250  0.832854  0.999736
GBP    0.708502  0.822542 -0.030606  0.888014  0.810807  0.846289  0.819090
HKD    0.765298  0.947429  0.131143  0.908704  0.948238  0.763070  0.945530
HRK    0.792345  0.983191  0.235679  0.899319  0.974996  0.747082  0.983073
HUF    0.088851  0.363602 -0.010359  0.203199  0.393998 -0.024009  0.358875
IDR    0.631449  0.640320  0.141069  0.755961  0.609301  0.924078  0.631536
ILS    0.763946  0.891313  0.130787  0.932787  0.868666  0.913598  0.889497
```

INR	0.855644	0.954910	0.227446	0.927315	0.933529	0.764152	0.957285
ISK	0.547348	0.633431	0.425259	0.531451	0.572624	0.608250	0.631532
JPY	0.811039	0.938557	0.157799	0.927371	0.936178	0.893358	0.933979
KRW	0.655351	0.706522	0.156190	0.789178	0.673561	0.956989	0.699230
MXN	0.640958	0.520842	0.353461	0.650117	0.476201	0.858396	0.519456
MYR	0.872331	0.944603	0.272759	0.953356	0.913197	0.925952	0.943078
NOK	0.211986	0.097856	0.221597	0.115570	0.087050	-0.113665	0.112564
NZD	0.774855	0.615861	0.414791	0.712192	0.547241	0.890812	0.614583
PHP	0.830148	0.961457	0.156201	0.947941	0.948995	0.863858	0.959449
PLN	0.494267	0.525110	0.261377	0.398503	0.508661	0.089646	0.531850
RON	0.791260	0.988002	0.186188	0.917444	0.986031	0.750262	0.988055
RUB	0.014979	0.188869	0.201694	0.021808	0.221564	-0.258765	0.192643
SEK	0.726132	0.808955	0.206714	0.871349	0.804502	0.895076	0.805652
SGD	0.857877	0.954610	0.208351	0.970631	0.932793	0.937638	0.951751
THB	0.751915	0.785670	0.245979	0.857295	0.757112	0.957796	0.778101
TRY	-0.190327	-0.135619	0.127408	-0.289149	-0.139475	-0.530662	-0.128475
USD	0.766223	0.947312	0.131031	0.910027	0.948155	0.766350	0.945307
XDR	0.819544	0.967763	0.185194	0.954253	0.956359	0.860735	0.965821
ZAR	0.604020	0.448734	0.258776	0.600433	0.397289	0.838230	0.445788

kod	EUR	GBP	HKD	...	PLN	RON	RUB	\
kod				...				
AUD	0.831635	0.708502	0.765298	...	0.494267	0.791260	0.014979	
BGN	0.999879	0.822542	0.947429	...	0.525110	0.988002	0.188869	
BRL	0.221910	-0.030606	0.131143	...	0.261377	0.186188	0.201694	
CAD	0.950551	0.888014	0.908704	...	0.398503	0.917444	0.021808	
CHF	0.984250	0.810807	0.948238	...	0.508661	0.986031	0.221564	
CNY	0.832854	0.846289	0.763070	...	0.089646	0.750262	-0.258765	
DKK	0.999736	0.819090	0.945530	...	0.531850	0.988055	0.192643	
EUR	1.000000	0.822696	0.947589	...	0.525691	0.988196	0.189186	
GBP	0.822696	1.000000	0.781841	...	0.208899	0.785219	-0.088797	
HKD	0.947589	0.781841	1.000000	...	0.517598	0.953059	0.240995	
HRK	0.983500	0.762627	0.948440	...	0.587515	0.991832	0.305726	
HUF	0.361975	0.107237	0.395778	...	0.666207	0.448087	0.618780	
IDR	0.639195	0.780498	0.596508	...	-0.121089	0.543329	-0.395797	
ILS	0.891390	0.931901	0.816839	...	0.271677	0.839532	-0.088474	
INR	0.955621	0.784950	0.960768	...	0.599416	0.952540	0.223340	
ISK	0.632356	0.407835	0.470336	...	0.279177	0.578959	0.003473	
JPY	0.938161	0.827783	0.939188	...	0.337578	0.909586	0.051977	
KRW	0.705071	0.786263	0.611986	...	-0.085260	0.606356	-0.365944	
MXN	0.520091	0.603234	0.370441	...	-0.160234	0.402005	-0.468334	
MYR	0.944589	0.838473	0.926527	...	0.367589	0.900046	0.005417	
NOK	0.100105	0.119137	-0.027622	...	0.502029	0.140477	0.385703	
NZD	0.615800	0.698169	0.489579	...	-0.066605	0.505559	-0.395989	
PHP	0.961324	0.821121	0.974866	...	0.428047	0.940083	0.081689	
PLN	0.525691	0.208899	0.517598	...	1.000000	0.597438	0.634026	
RON	0.988196	0.785219	0.953059	...	0.597438	1.000000	0.298226	

RUB	0.189186	-0.088797	0.240995	...	0.634026	0.298226	1.000000
SEK	0.808059	0.838988	0.701015	...	0.168029	0.754698	-0.069383
SGD	0.954173	0.871904	0.929772	...	0.368640	0.911900	-0.003335
THB	0.784499	0.806603	0.748210	...	0.097642	0.708444	-0.220001
TRY	-0.133922	-0.359386	-0.011863	...	0.530263	-0.037154	0.699508
USD	0.947473	0.784027	0.999948	...	0.513344	0.952394	0.235973
XDR	0.967911	0.848624	0.979125	...	0.442604	0.950631	0.140469
ZAR	0.447845	0.600869	0.316315	...	-0.259534	0.321221	-0.597814

kod	SEK	SGD	THB	TRY	USD	XDR	ZAR
kod							
AUD	0.726132	0.857877	0.751915	-0.190327	0.766223	0.819544	0.604020
BGN	0.808955	0.954610	0.785670	-0.135619	0.947312	0.967763	0.448734
BRL	0.206714	0.208351	0.245979	0.127408	0.131031	0.185194	0.258776
CAD	0.871349	0.970631	0.857295	-0.289149	0.910027	0.954253	0.600433
CHF	0.804502	0.932793	0.757112	-0.139475	0.948155	0.956359	0.397289
CNY	0.895076	0.937638	0.957796	-0.530662	0.766350	0.860735	0.838230
DKK	0.805652	0.951751	0.778101	-0.128475	0.945307	0.965821	0.445788
EUR	0.808059	0.954173	0.784499	-0.133922	0.947473	0.967911	0.447845
GBP	0.838988	0.871904	0.806603	-0.359386	0.784027	0.848624	0.600869
HKD	0.701015	0.929772	0.748210	-0.011863	0.999948	0.979125	0.316315
HRK	0.740277	0.906554	0.711022	-0.023477	0.947585	0.946371	0.318652
HUF	0.140476	0.220050	0.105736	0.364498	0.394056	0.294847	-0.382908
IDR	0.833373	0.808858	0.946824	-0.627023	0.601572	0.710526	0.864760
ILS	0.902709	0.931803	0.874852	-0.375493	0.818528	0.892064	0.661707
INR	0.700435	0.924618	0.712858	0.002201	0.959939	0.954042	0.361886
ISK	0.535415	0.598420	0.596894	-0.220361	0.470346	0.542749	0.473438
JPY	0.805304	0.969904	0.860547	-0.248281	0.940534	0.968360	0.547164
KRW	0.880396	0.838782	0.946396	-0.662471	0.616161	0.731986	0.888199
MXN	0.791157	0.668669	0.817719	-0.706750	0.374584	0.523151	0.969243
MYR	0.818957	0.988273	0.887876	-0.238076	0.927683	0.971812	0.615295
NOK	0.173637	-0.019884	-0.150855	0.248622	-0.030956	-0.012589	-0.124620
NZD	0.789578	0.746301	0.851872	-0.582621	0.493194	0.630241	0.919875
PHP	0.763521	0.973376	0.828638	-0.174739	0.975686	0.985594	0.493182
PLN	0.168029	0.368640	0.097642	0.530263	0.513344	0.442604	-0.259534
RON	0.754698	0.911900	0.708444	-0.037154	0.952394	0.950631	0.321221
RUB	-0.069383	-0.003335	-0.220001	0.699508	0.235973	0.140469	-0.597814
SEK	1.000000	0.864012	0.883577	-0.486703	0.703722	0.797144	0.717558
SGD	0.864012	1.000000	0.910626	-0.286049	0.931288	0.975152	0.624381
THB	0.883577	0.910626	1.000000	-0.498988	0.752265	0.832746	0.788772
TRY	-0.486703	-0.286049	-0.498988	1.000000	-0.017629	-0.143524	-0.758203
USD	0.703722	0.931288	0.752265	-0.017629	1.000000	0.979865	0.321146
XDR	0.797144	0.975152	0.832746	-0.143524	0.979865	1.000000	0.472030
ZAR	0.717558	0.624381	0.788772	-0.758203	0.321146	0.472030	1.000000

[33 rows x 33 columns]

```
[13]: corr[corr != 1.0][corr > 0.98].stack()
```

```
[13]: kod  kod
      BGN  CHF    0.984522
           DKK    0.999578
           EUR    0.999879
           HRK    0.983191
           RON    0.988002
      CHF  BGN    0.984522
           DKK    0.983113
           EUR    0.984250
           RON    0.986031
      DKK  BGN    0.999578
           CHF    0.983113
           EUR    0.999736
           HRK    0.983073
           RON    0.988055
      EUR  BGN    0.999879
           CHF    0.984250
           DKK    0.999736
           HRK    0.983500
           RON    0.988196
      HKD  USD    0.999948
      HRK  BGN    0.983191
           DKK    0.983073
           EUR    0.983500
           RON    0.991832
      MYR  SGD    0.988273
      PHP  XDR    0.985594
      RON  BGN    0.988002
           CHF    0.986031
           DKK    0.988055
           EUR    0.988196
           HRK    0.991832
      SGD  MYR    0.988273
      USD  HKD    0.999948
      XDR  PHP    0.985594
dtype: float64
```

```
[14]: fig, ax = plt.subplots(figsize=(len(corr) / 3, len(corr) / 3))
      cax = ax.matshow(corr, cmap='RdYlGn')
      plt.xticks(range(len(corr.columns)), corr.columns, rotation=90);
      plt.yticks(range(len(corr.columns)), corr.columns);

      # Add the colorbar legend
      cbar = fig.colorbar(cax, ticks=[-1, 0, 1], aspect=40, shrink=.8)
```

```
Canvas(toolbar=Toolbar(toolitems=[('Home', 'Reset original view', 'home', ↵  
↵'home')), ('Back', 'Back to previous ...
```

2.4 Úkol C

Ve třetím úkolu, našem vlastním, jsme se rozhodli zjistit zda nemají jednotlivé dny v týdnu vliv na změnu kurzu.

```
[15]: import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
  
cursor.execute(  
    "SELECT date_part('dow', den::date) as dow, AVG(normalizovany_kurz) FROM ↵  
↵kurz GROUP BY dow order by dow"  
)  
days = {}  
for item in cursor:  
    day_str = ""  
    if (item[0] == 1):  
        day_str = "mon"  
    elif (item[0] == 2):  
        day_str = "tue"  
    elif (item[0] == 3):  
        day_str = "wed"  
    elif (item[0] == 4):  
        day_str = "thu"  
    elif (item[0] == 5):  
        day_str = "fri"  
    days[day_str] = item[1]  
  
fig = plt.figure()  
x = np.arange(len(days))  
plt.bar(x, height=days.values())  
plt.xticks(x, days.keys(), rotation=-90);
```

```
Canvas(toolbar=Toolbar(toolitems=[('Home', 'Reset original view', 'home', ↵  
↵'home')), ('Back', 'Back to previous ...
```

```
[16]: days
```

```
[16]: {'mon': 8.10535261497326,  
      'tue': 8.1323463315508,  
      'wed': 8.11074049431818,  
      'thu': 8.11117252693603,  
      'fri': 8.11452631074379}
```

Vidíme, že rozdíl mezi jednotlivými dny je téměř zanedbatelný, ač je znát mírný skok mezi hodnotami v pondělí a v úterý.