

zadanie_1_odovzдание

October 10, 2021

1 Zadanie 1

poznámka: žiadna časť kodu nebola v tomto notebooku spustená. Tento dokument bol vytvorený z viacerých notebookov, ktoré sú uložené na mojom gite

```
[1]: %matplotlib inline
from psycpg2.extras import RealDictCursor
import psycpg2
import nltk
import pandas as pd
import re
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import numpy as np
import matplotlib.pyplot as plt
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data] C:\Users\Marek\AppData\Roaming\nltk_data...
[nltk_data] Package vader_lexicon is already up-to-date!
```

1.0.1 Použité funkcie

create_function() - vytvorenie pripojenia do databázy
remove_emoji(tweet) - vymazanie emoji z tweetu
remove_mentions_and_hashtags(tweet) - vymazanie mentions a hashtagov z tweetu
clarify_tweet(tweet) - funkcia vycisti tweet volaním predoslych funkcií a vráti sentiment

```
[2]: def create_connection():
    return psycpg2.connect("dbname=twitter user=postgres password=postgres_
    →host=localhost port=5433")

def remove_emoji(tweet):
    emoji = re.compile("[
    u"\U0001F600-\U0001F64F"
    u"\U0001F300-\U0001F5FF"
    u"\U0001F680-\U0001F6FF"
    u"\U0001F1E0-\U0001F1FF"
    u"\U00002500-\U00002BEF"
    u"\U00002702-\U000027B0"
    u"\U00002702-\U000027B0"]")
```

```

        u"\U000024C2-\U0001F251"
        u"\U0001f926-\U0001f937"
        u"\U00010000-\U0010ffff"
        u"\u2640-\u2642"
        u"\u2600-\u2B55"
        u"\u200d"
        u"\u23cf"
        u"\u23e9"
        u"\u231a"
        u"\ufe0f"
        u"\u3030"
        "]" +", re.UNICODE)
    return re.sub(emoji, '', tweet)

def remove_mentions_and_hashtags(tweet):
    clarified_tweet = re.sub("(@[A-Za-z0-9]+)|(#[A-Za-z0-9_]+)", " ", tweet)
    return ' '.join(clarified_tweet.split())

def clarify_tweet(tweet):
    no_emoji = remove_emoji(tweet)
    text = remove_mentions_and_hashtags(no_emoji)
    return sid.polarity_scores(text)

```

1.1 Uloha 2

Vytvorenie potrebných stĺpcov (neg, neu, pos, compound):

```

[ ]: add_columns_query = """
ALTER TABLE tweets
ADD COLUMN neg DECIMAL DEFAULT 0,
ADD COLUMN neu DECIMAL DEFAULT 0,
ADD COLUMN pos DECIMAL DEFAULT 0,
ADD COLUMN compound DECIMAL DEFAULT 0
"""

try:
    conn = create_connection();
    cur = conn.cursor()
    cur.execute(add_columns_query)
    conn.commit()
    conn.close()
except:
    conn.rollback()

```

Select potrebných údajov pre ulohu c. 2.

```

[ ]: select_query = """
SELECT DISTINCT t.id, t.content, t.neg, t.neu, t.pos, t.compound
FROM tweets t

```

```

JOIN tweet_hashtags th ON th.tweet_id = t.id
JOIN hashtags h ON h.id = th.hashtag_id
WHERE h.value ILIKE any (array[
    '%DeepstateVirus%',
    '%DeepStateFauci%',
    '%DeepStateVaccine%',
    '%QAnon%',
    '%Agenda21%',
    '%CCPVirus%',
    '%ClimateChangeHoax%',
    '%GlobalWarmingHoax%',
    '%ChinaLiedPeopleDied%',
    '%SorosVirus%',
    '%5GCoronavirus%',
    '%MAGA%',
    '%WWG1WGA%',
    '%Chemtrails%',
    '%flatEarth%',
    '%MoonLandingHoax%',
    '%moonhoax%',
    '%illuminati%',
    '%pizzaGateIsReal%',
    '%PedoGateIsReal%',
    '%911truth%',
    '%911insidejob%',
    '%reptilians%'
])
"""

conn = create_connection()
df = pd.read_sql_query(select_query, con=conn)
conn.close()

```

Vycistenie tweetov od emoji, mentions, hashtagov a nasledne vypocet sentimentu:

```

[ ]: try:
    sid = SentimentIntensityAnalyzer()
    df['vader'] = df['content'].apply(lambda x : clarify_tweet(x))
    df['neg'] = df['vader'].apply(lambda x : x['neg'])
    df['neu'] = df['vader'].apply(lambda x : x['neu'])
    df['pos'] = df['vader'].apply(lambda x : x['pos'])
    df['compound'] = df['vader'].apply(lambda x : x['compound'])

except e:
    print(e)

```

Update jednotlivych hodnot selectnutych tweetov v database:

```
[1]: try:
    conn = create_connection()
    cur = conn.cursor()

    for index, row in df.iterrows():
        update_query = """
        UPDATE tweets
        SET neg = {},
            neu = {},
            pos = {},
            compound = {}
        WHERE '{}' = tweets.id;
        """.format(row['neg'], row['neu'], row['pos'], row['compound'],
        →row['id'])
        cur.execute(update_query)

    conn.commit()
    conn.close()

except:
    pass
```

```
[3]: select_uloha2 = """
SELECT t.id, t.neg, t.neu, t.pos, t.compound
FROM tweets t
LIMIT 5;
"""

conn = create_connection()
df = pd.read_sql_query(select_uloha2, con=conn)
conn.close()
df
```

```
[3]:
```

	id	neg	neu	pos	compound
0	1220798255098777601	0.0	0.0	0.0	0.0
1	1220790544009519104	0.0	0.0	0.0	0.0
2	1220798255992201217	0.0	0.0	0.0	0.0
3	1220798256608878598	0.0	0.0	0.0	0.0
4	1220798259431645185	0.0	0.0	0.0	0.0

1.2 Uloha 3

Vytvorenie potrebných tabuliek:

```
[1]: create_table_theories = """
CREATE TABLE theories (
    id SERIAL PRIMARY KEY,
    theory VARCHAR(50)
```

```

);
"""

create_table_tweets_theories = """
CREATE TABLE tweets_theories (
    id SERIAL PRIMARY KEY,
    tweet_id VARCHAR(20) NOT NULL,
    theory_id INTEGER NOT NULL,
    FOREIGN KEY (tweet_id) REFERENCES tweets(id) ON UPDATE CASCADE ON DELETE_
→RESTRICT,
    FOREIGN KEY (theory_id) REFERENCES theories(id) ON UPDATE CASCADE ON DELETE_
→RESTRICT
);
"""

conn = create_connection();
cur = conn.cursor()
cur.execute(create_table_theories)
cur.execute(create_table_tweets_theories)
conn.commit()
conn.close()

```

Naplnenie tabulky 'theories':

```

[: fill_theories = """
INSERT INTO theories(theory)
VALUES  ('Deep State'),
        ('Qanon'),
        ('New world order'),
        ('The virus escaped from a Chinese lab'),
        ('Global Warming is HOAX'),
        ('COVID19 and microchipping'),
        ('COVID19 is spreaded by 5G'),
        ('Moon landing is fake'),
        ('9/11 was inside job'),
        ('Pizzagate conspiracy theory'),
        ('Chemtrails'),
        ('FlatEarth'),
        ('Illuminati'),
        ('Reptilian conspiracy theory');
"""

conn = create_connection()
cur = conn.cursor()
cur.execute(fill_theories)
conn.commit()
conn.close()

```

Naplnenie tabulky 'tweets_theories', ktorá slúži na mapovanie jednotlivých tweetov na teórie:

```

[ ]: fill_tweets_theories = """
INSERT INTO tweets_theories (tweet_id, theory_id)
SELECT DISTINCT t.id, CASE
    WHEN h.value ILIKE any (array['%deepstatevirus%', '%deepstatefauci%', '%deepstatevaccine%'])
    THEN (SELECT id FROM theories WHERE theory ILIKE 'Deep State')
    WHEN h.value ILIKE any (array['%QAnon%', '%MAGA%', '%WWG1WGA%'])
    THEN (SELECT id FROM theories WHERE theory ILIKE 'Qanon')
    WHEN h.value ILIKE '%Agenda21%'
    THEN (SELECT id FROM theories WHERE theory ILIKE 'New world%')
    WHEN h.value ILIKE any (array['%CCPVirus%', '%ChinaLiedPeopleDied%'])
    THEN (SELECT id FROM theories WHERE theory ILIKE 'The virus escaped%')
    WHEN h.value ILIKE any (array['%ClimateChangeHoax%', '%GlobalWarmingHoax%'])
    THEN (SELECT id FROM theories WHERE theory ILIKE 'Global warming%')
    WHEN h.value ILIKE '%SorosVirus%'
    THEN (SELECT id FROM theories WHERE theory ILIKE 'COVID19 and %')
    WHEN h.value ILIKE '%5GCoronavirus%'
    THEN (SELECT id FROM theories WHERE theory ILIKE 'COVID19 is s%')
    WHEN h.value ILIKE any (array['%MoonLandingHoax%', '%moonhoax%'])
    THEN (SELECT id FROM theories WHERE theory ILIKE 'Moon landing%')
    WHEN h.value ILIKE any (array['%911truth%', '%911insidejob%'])
    THEN (SELECT id FROM theories WHERE theory ILIKE '9/11%')
    WHEN h.value ILIKE any (array['%pizzaGateIsReal%', '%PedoGateIsReal%'])
    THEN (SELECT id FROM theories WHERE theory ILIKE 'Pizzagate%')
    WHEN h.value ILIKE '%Chemtrails%'
    THEN (SELECT id FROM theories WHERE theory ILIKE 'Chemtrails')
    WHEN h.value ILIKE '%FlatEarth%'
    THEN (SELECT id FROM theories WHERE theory ILIKE 'FlatEarth')
    WHEN h.value ILIKE '%illuminati%'
    THEN (SELECT id FROM theories WHERE theory ILIKE 'illuminati')
    WHEN h.value ILIKE '%reptilians%'
    THEN (SELECT id FROM theories WHERE theory ILIKE 'reptilian%')
END AS theories_id
FROM tweets t
JOIN tweet_hashtags as th ON t.id = th.tweet_id
JOIN hashtags as h ON h.id = th.hashtag_id
WHERE h.value ILIKE any (array[
    '%DeepstateVirus%',
    '%DeepStateFauci%',
    '%DeepStateVaccine%',
    '%QAnon%',
    '%Agenda21%',
    '%CCPVirus%',
    '%ClimateChangeHoax%',
    '%GlobalWarmingHoax%',
    '%ChinaLiedPeopleDied%',
    '%SorosVirus%',

```

```

        '%5GCoronavirus%',
        '%MAGA%',
        '%WWG1WGA%',
        '%Chemtrails%',
        '%flatEarth%',
        '%MoonLandingHoax%',
        '%moonhoax%',
        '%illuminati%',
        '%pizzaGateIsReal%',
        '%PedoGateIsReal%',
        '%911truth%',
        '%911insidejob%',
        '%reptilians%'
    ])
    """

    conn = create_connection()
    cur = conn.cursor()
    cur.execute(fill_tweets_theories)
    conn.commit()
    conn.close()

```

```

[6]: select_uloha3 = """
SELECT t.id, th.theory
FROM tweets t
JOIN tweets_theories tt ON tt.tweet_id = t.id
JOIN theories th ON th.id = tt.theory_id
LIMIT 5;
"""

conn = create_connection()
df = pd.read_sql_query(select_uloha3, con=conn)
conn.close()
df

```

```

[6]:

```

	id	theory
0	1034289685227614208	Qanon
1	1034289685227614208	Pizzagate conspiracy theory
2	1037494012033343488	Qanon
3	1038674252940148736	Qanon
4	1039279051456831489	Qanon

1.3 Uloha 4

Vytvorenie potrebnej tabulky:

```

[ ]: create_table = """
CREATE TABLE sentiment_ratio_by_weeks (

```

```

        id SERIAL PRIMARY KEY,
        theory_id INTEGER NOT NULL,
        tweet_count INTEGER DEFAULT 0,
        tweet_extreme_count INTEGER DEFAULT 0,
        tweet_neutral_count INTEGER DEFAULT 0,
        ratio DECIMAL DEFAULT 0,
        week INTEGER,
        year INTEGER,
        FOREIGN KEY (theory_id) REFERENCES theories(id) ON UPDATE CASCADE ON DELETE_
        ↳RESTRIC
    )
    """

conn = create_connection();
cur = conn.cursor()
cur.execute(create_table)
conn.commit()
conn.close()

```

Naplnenie vytvorenej tabulky:

```

[ ]: select = """
INSERT INTO sentiment_ratio_by_weeks (theory_id, year, week,
↳tweet_extreme_count, tweet_neutral_count, tweet_count, ratio)
SELECT sub.*, CASE WHEN sub.neutral = 0 THEN NULL ELSE sub.extreme::DECIMAL /
↳sub.neutral::DECIMAL END
↳sub.neutral::DECIMAL END
FROM
(SELECT tt.theory_id, DATE_PART('year', t.happened_at) as year,
↳DATE_PART('week', t.happened_at) as week, COUNT(CASE WHEN t.compound < -0.5
↳OR t.compound > 0.5 THEN t.compound END) as extreme, COUNT(CASE WHEN t.
↳compound >= -0.5 AND t.compound <= 0.5 THEN t.compound END) as neutral,
↳count(tt.theory_id) as t_count
FROM tweets t
JOIN tweets_theories tt ON tt.tweet_id = t.id
GROUP BY tt.theory_id, year, week) as sub
"""

conn = create_connection();
cur = conn.cursor()
cur.execute(select)
conn.commit()
conn.close()

```

Select dat, potrebných pre vizualizáciu pomeru extrémnych a neutralných tweetov jednotlivých teórií podľa týždňov:

```

[7]: data = """
SELECT t.theory, s.tweet_extreme_count, s.tweet_neutral_count, s.week, s.year
FROM sentiment_ratio_by_weeks s

```



```
JOIN theories t ON t.id = s.theory_id
ORDER BY t.theory, s.year, s.week ASC;
"""
```

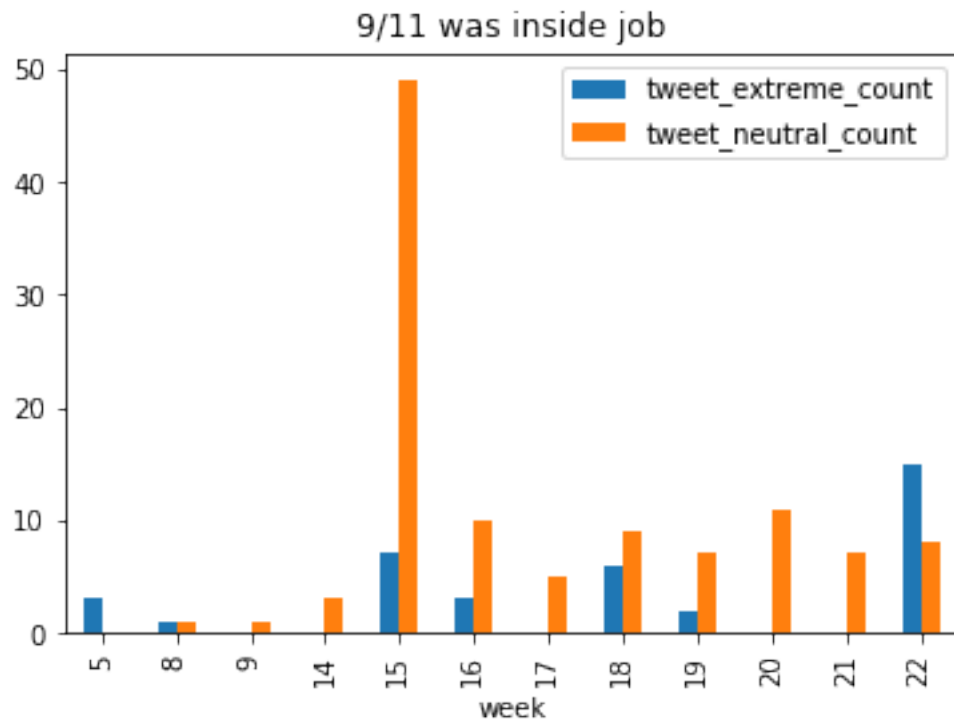
```
conn = create_connection()
df = pd.read_sql_query(data, con=conn)
conn.close()
```

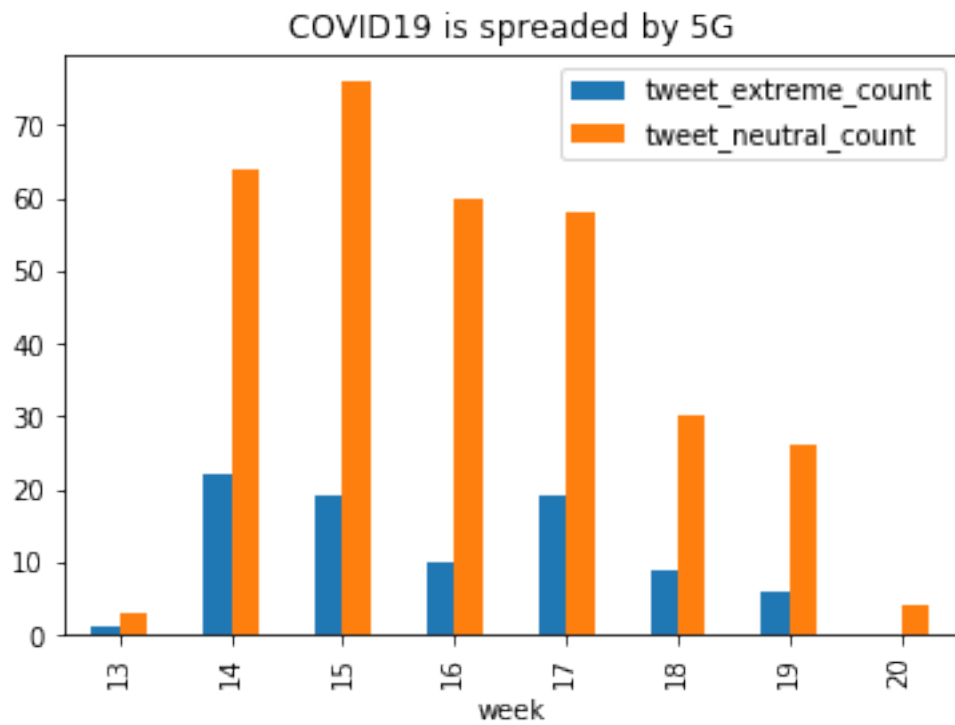
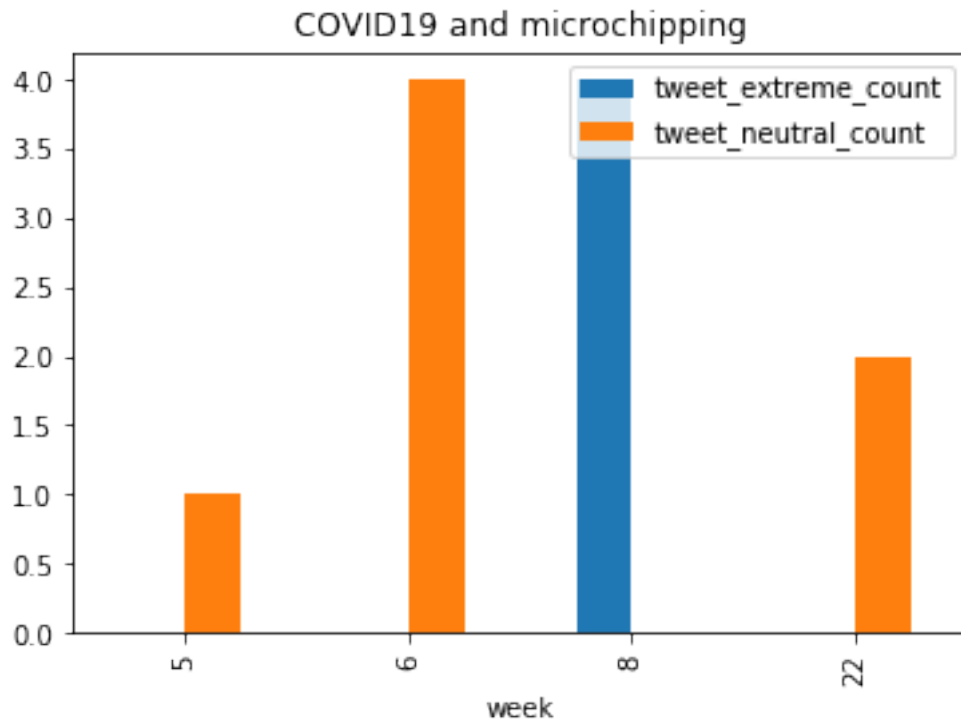
Rozdelenie datasetu na viacero mensich datasetov pre jednotlivé teórie a vykreslenie grafov:

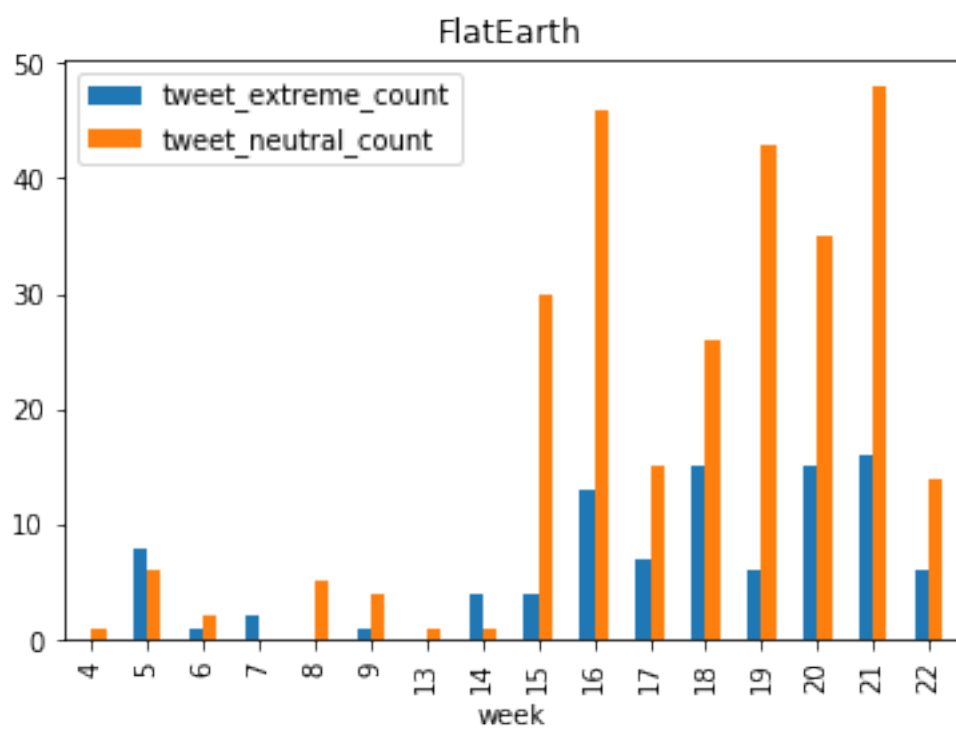
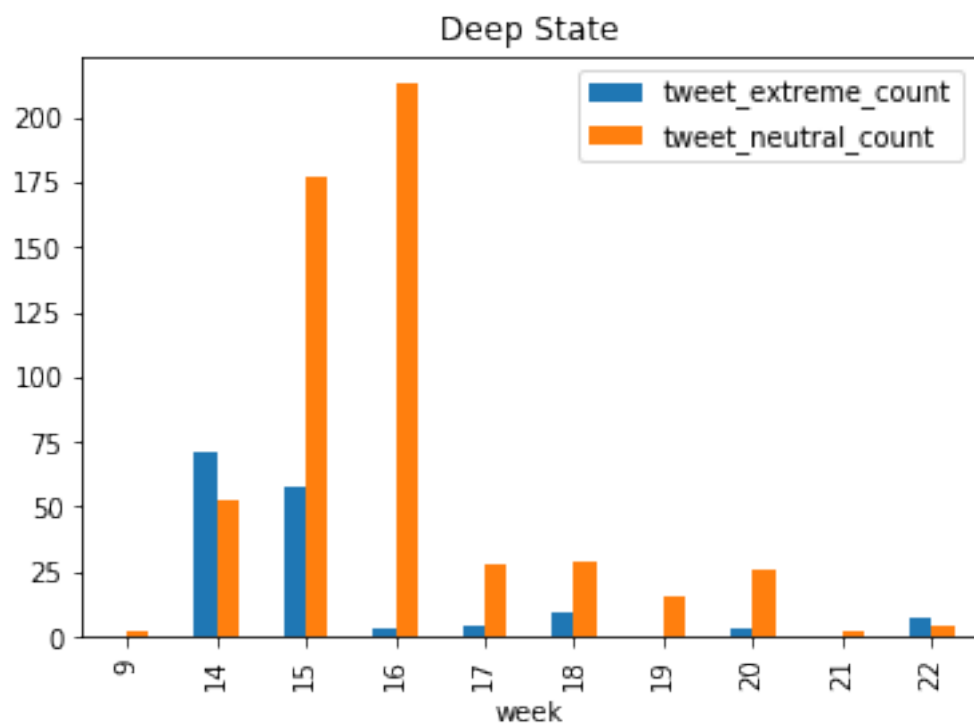
```
[8]: df_theory_1 = df[df['theory'] == '9/11 was inside job']
df_theory_2 = df[df['theory'] == 'COVID19 and microchipping']
df_theory_3 = df[df['theory'] == 'COVID19 is spreaded by 5G']
df_theory_4 = df[df['theory'] == 'Deep State']
df_theory_5 = df[df['theory'] == 'FlatEarth']
df_theory_6 = df[df['theory'] == 'Global Warming is HOAX']
df_theory_7 = df[df['theory'] == 'Chemtrails']
df_theory_8 = df[df['theory'] == 'Illuminati']
df_theory_9 = df[df['theory'] == 'Moon landing is fake']
df_theory_10 = df[df['theory'] == 'New world order']
df_theory_11 = df[df['theory'] == 'Pizzagate conspiracy theory']
df_theory_12 = df[df['theory'] == 'Qanon']
df_theory_13 = df[df['theory'] == 'Reptilian conspiracy theory']
df_theory_14 = df[df['theory'] == 'The virus escaped from a Chinese lab']
df_theory_1.plot.bar(title='9/11 was inside job', x='week',
    →y=['tweet_extreme_count', 'tweet_neutral_count'])
df_theory_2.plot.bar(title='COVID19 and microchipping', x='week',
    →y=['tweet_extreme_count', 'tweet_neutral_count'])
df_theory_3.plot.bar(title='COVID19 is spreaded by 5G', x='week',
    →y=['tweet_extreme_count', 'tweet_neutral_count'])
df_theory_4.plot.bar(title='Deep State', x='week',
    →y=['tweet_extreme_count', 'tweet_neutral_count'])
df_theory_5.plot.bar(title='FlatEarth', x='week',
    →y=['tweet_extreme_count', 'tweet_neutral_count'])
df_theory_6.plot.bar(title='Global Warming is HOAX', x='week',
    →y=['tweet_extreme_count', 'tweet_neutral_count'])
df_theory_7.plot.bar(title='Chemtrails', x='week',
    →y=['tweet_extreme_count', 'tweet_neutral_count'])
df_theory_8.plot.bar(title='Illuminati', x='week',
    →y=['tweet_extreme_count', 'tweet_neutral_count'])
df_theory_9.plot.bar(title='Moon landing is fake', x='week',
    →y=['tweet_extreme_count', 'tweet_neutral_count'])
df_theory_10.plot.bar(title='New world order', x='week',
    →y=['tweet_extreme_count', 'tweet_neutral_count'])
df_theory_13.plot.bar(title='Reptilian conspiracy theory', x='week',
    →y=['tweet_extreme_count', 'tweet_neutral_count'])
```

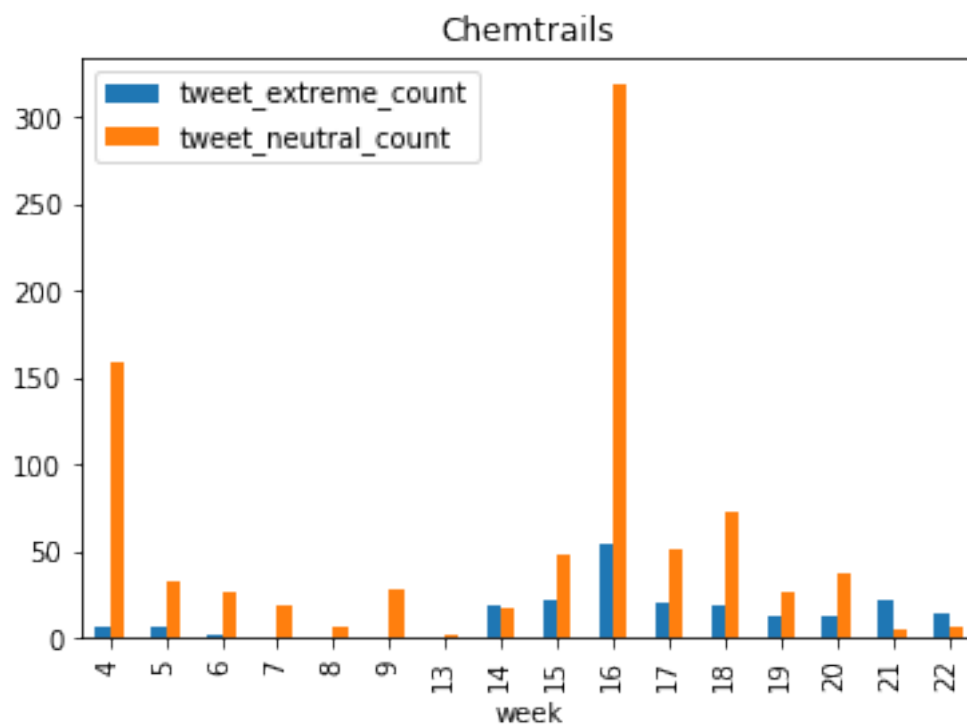
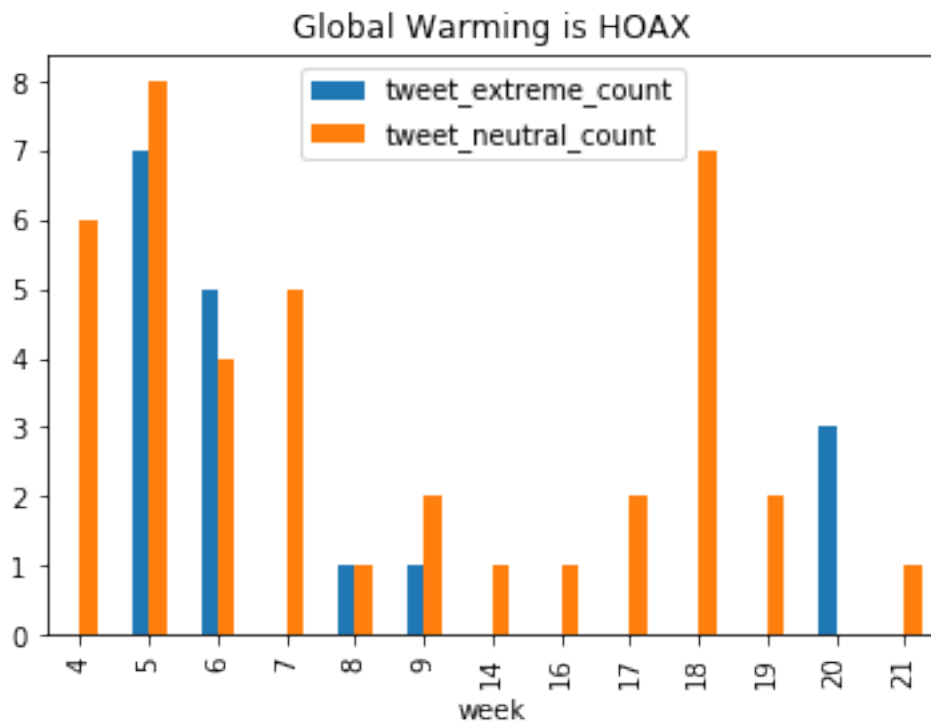
```
df_theory_14.plot.bar(title='The virus escaped from a Chinese lab', x='week',  
→y=['tweet_extreme_count','tweet_neutral_count'])
```

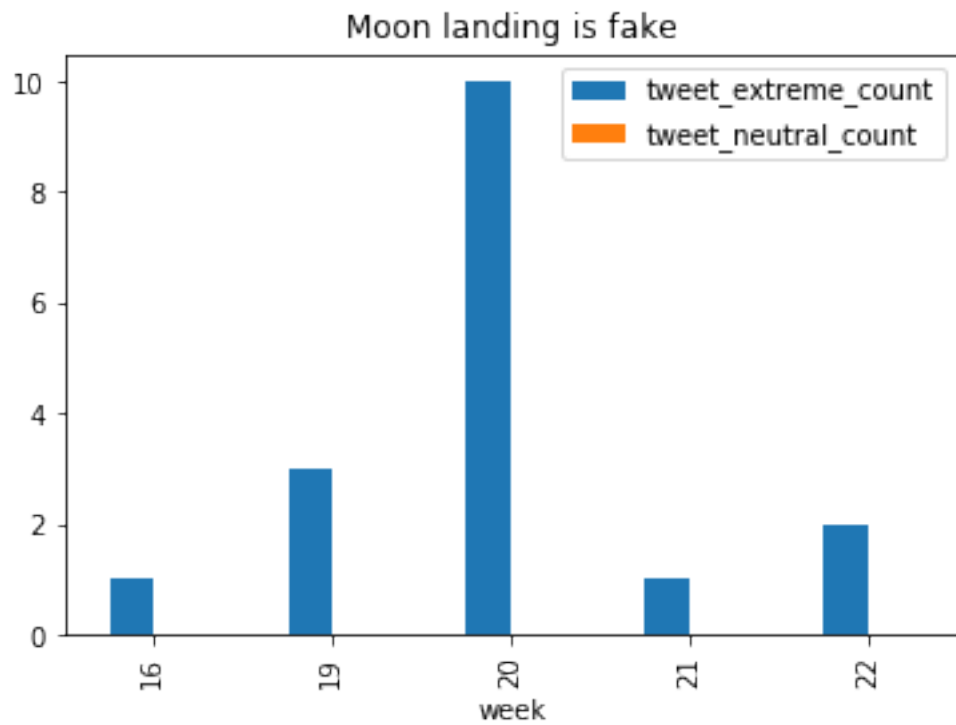
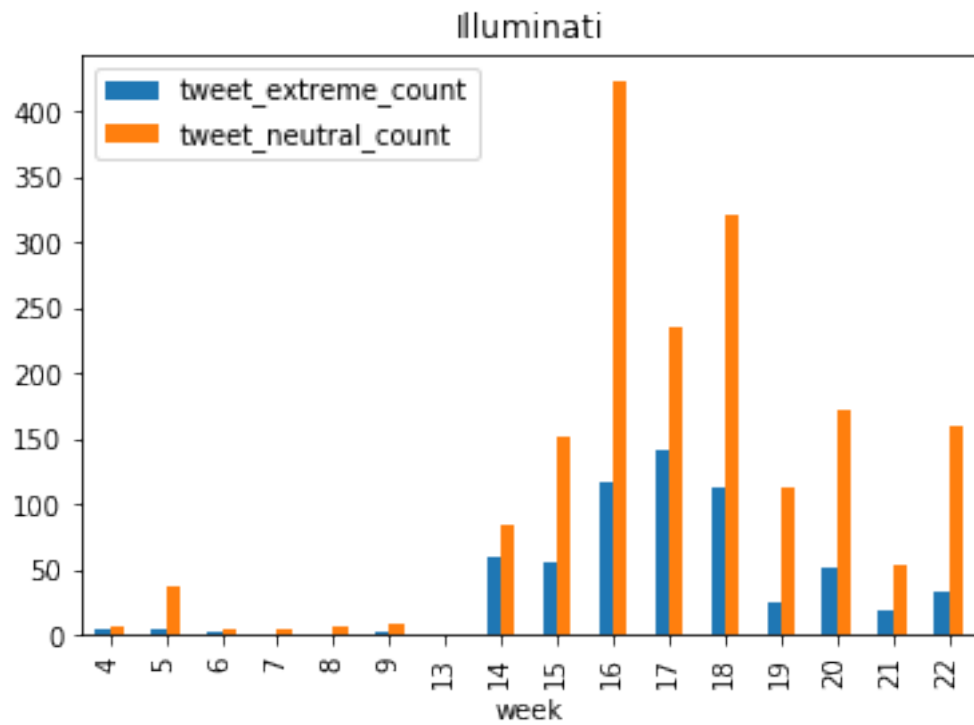
[8]: <matplotlib.axes._subplots.AxesSubplot at 0x1bedcb47fd0>

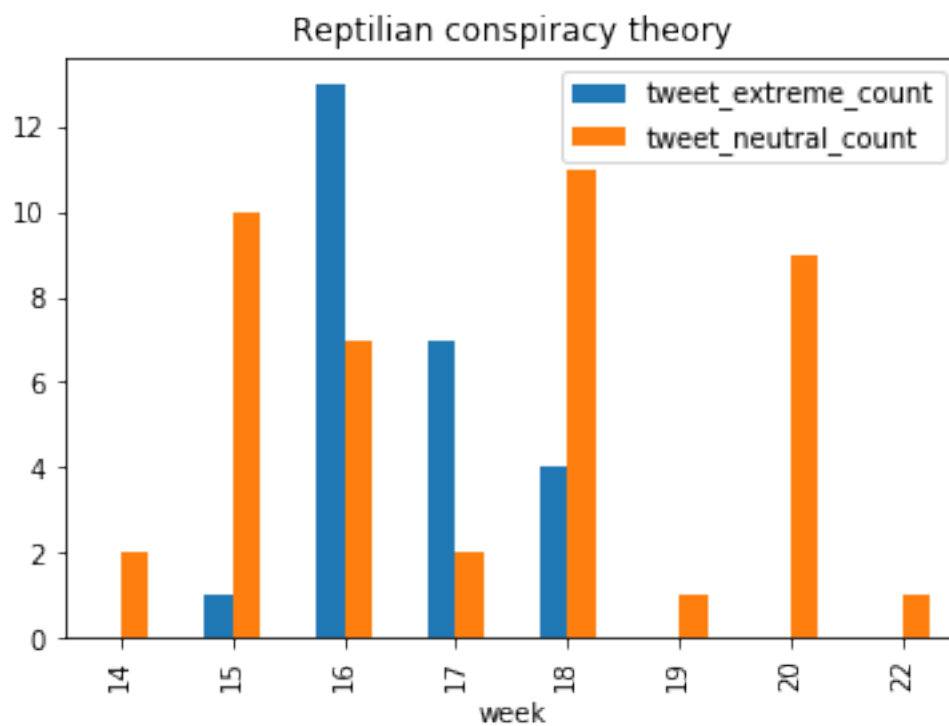
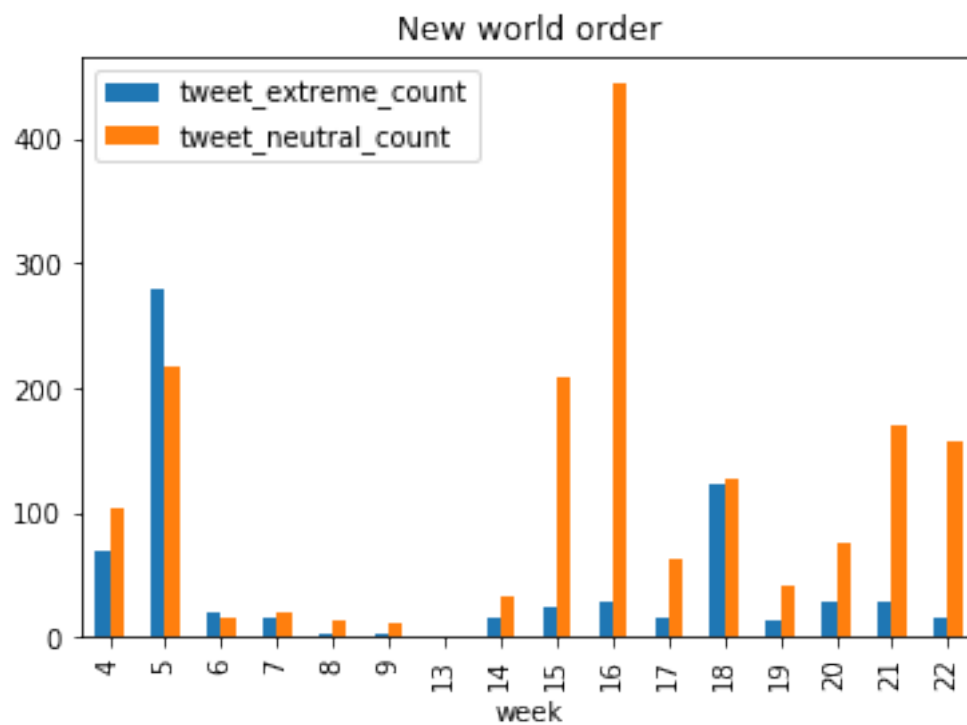


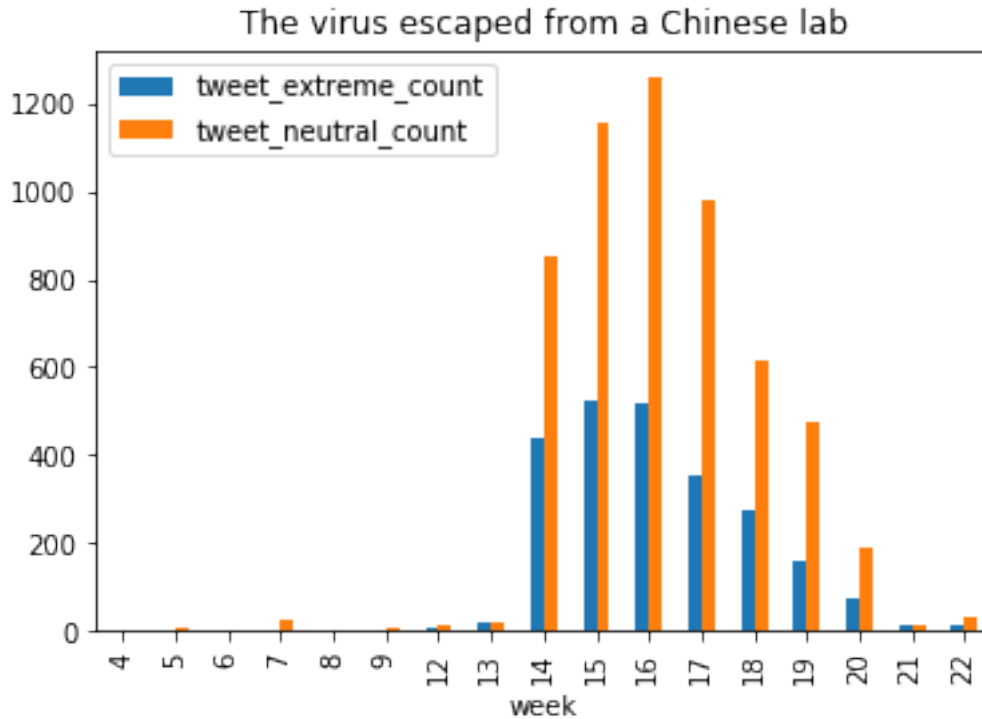












Nakolko teorie 'Pizzagate conspiracy theory' a 'Qanon' obsahuju tweety z viacerych rokov, je potrebné upraviť označenie týždňov (rok-týžden):

```
[9]: df_theory_11['year_week'] = df_theory_11['year'].astype(str) + '-' +
      df_theory_11['week'].astype(str)
```

C:\Users\Marek\Anaconda3\lib\site-packages\ipykernel_launcher.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

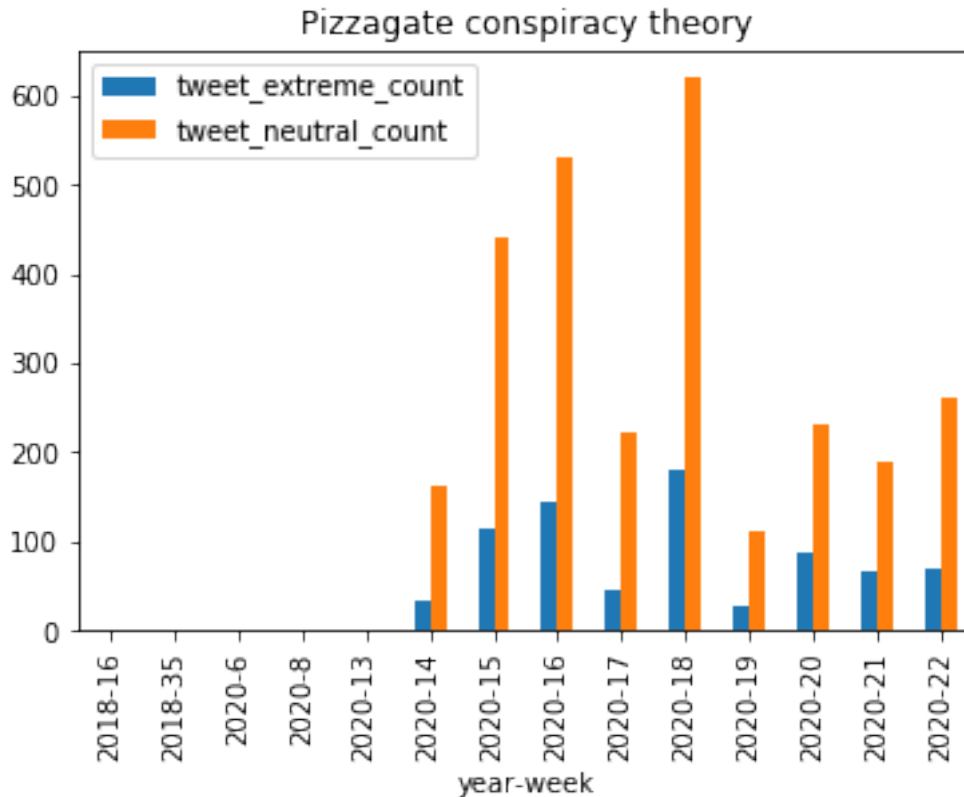
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

"""Entry point for launching an IPython kernel.

```
[10]: ax1 = df_theory_11.plot.bar(title='Pizzagate conspiracy theory', x='year_week',
      y=['tweet_extreme_count', 'tweet_neutral_count'])
      ax1.set_xlabel('year-week')
```

```
[10]: Text(0.5, 0, 'year-week')
```

```
[11]: df_theory_12['year_week'] = df_theory_12['year'].astype(str) + '-' +
      ↪df_theory_12['week'].astype(str)
```

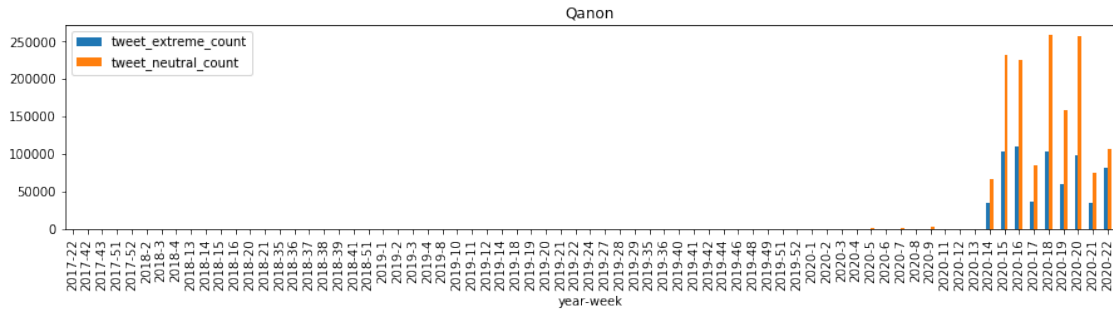
C:\Users\Marek\Anaconda3\lib\site-packages\ipykernel_launcher.py:1:
 SettingWithCopyWarning:
 A value is trying to be set on a copy of a slice from a DataFrame.
 Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

"""Entry point for launching an IPython kernel.

```
[12]: ax2 = df_theory_12.plot.bar(title='Qanon', x='year_week',
      ↪y=['tweet_extreme_count', 'tweet_neutral_count'], figsize=(15,3))
      ax2.set_xlabel('year-week')
```

```
[12]: Text(0.5, 0, 'year-week')
```



1.4 Uloha 5

Vytvorenie novej tabuľky pre ulohu 5 a jej naplnenie:

```
[ ]: create = """
CREATE TABLE top_extreme_accounts (
    id SERIAL PRIMARY KEY,
    author_id BIGINT NOT NULL,
    name CHARACTER VARYING(200),
    screen_name CHARACTER VARYING(200),
    tweet_count INTEGER,
    theory_id INTEGER NOT NULL,
    FOREIGN KEY (author_id) REFERENCES accounts(id) ON UPDATE CASCADE ON DELETE↵
    ↪RESTRICT,
    FOREIGN KEY (theory_id) REFERENCES theories(id) ON UPDATE CASCADE ON DELETE↵
    ↪RESTRICT
)
"""

insert = """
INSERT INTO top_extreme_accounts (author_id, name, screen_name, theory_id,↵
    ↪tweet_count)
SELECT sub.id, sub.name, sub.screen_name, sub.theory_id, sub.count
FROM
(SELECT ac.id, ac.name, ac.screen_name, th.id theory_id, count(t.id),↵
    ↪row_number() OVER (Partition BY th.theory ORDER BY count(t.id) DESC) AS rank
FROM tweets t
JOIN tweets_theories tt ON tt.tweet_id = t.id
JOIN theories th ON th.id = tt.theory_id
JOIN accounts ac ON ac.id = t.author_id
WHERE t.compound > 0.5 OR t.compound < -0.5
GROUP BY ac.id, ac.name, ac.screen_name, th.id) as sub
WHERE sub.rank <= 10;
"""
```

```

conn = create_connection()
cur = conn.cursor()
cur.execute(create)
cur.execute(insert)
conn.commit()
conn.close()

```

```

[13]: select_uloha5 = """
SELECT t.author_id, t.name, t.screen_name, th.theory, tweet_count
FROM top_extreme_accounts t
JOIN theories th ON th.id = t.theory_id
LIMIT 5;
"""

conn = create_connection()
df = pd.read_sql_query(select_uloha5, con=conn)
conn.close()
df

```

```

[13]:
      author_id      name      screen_name \
0      55905770  Miriam Selmi Reed      mimiselmi
1      2549547798    David Hooper    DaveHooper15
2  1242848729796935685    QuietBuck2    QuietBuck2
3  1243433781597548546      Karolin20775637
4  1237444146236919808    Be LVE SparrowHawkClwd

      theory  tweet_count
0  9/11 was inside job      2
1  9/11 was inside job      2
2  9/11 was inside job      1
3  9/11 was inside job      1
4  9/11 was inside job      1

```

1.5 Uloha 6

Select podľa zadania ulohy 6:

```

[14]: select = """
SELECT sub.theory, sub.value, sub.count
FROM
(SELECT th.theory, h.value, h.id, count(h.id), row_number() OVER (Partition by_
↳th.theory ORDER BY count(h.id) DESC) as rank
FROM tweets t
JOIN tweets_theories tt ON tt.tweet_id = t.id
JOIN theories th ON th.id = tt.theory_id
JOIN tweet_hashtags ths ON ths.tweet_id = t.id
JOIN hashtags h ON h.id = ths.hashtag_id
WHERE t.compound > 0.5 OR t.compound < -0.5

```

```
GROUP BY th.theory, h.value, h.id) as sub
WHERE sub.rank <= 10;
"""

conn = create_connection()
df = pd.read_sql_query(select, con=conn)
conn.close()
```

```
[15]: with pd.option_context('display.max_rows', None, 'display.max_columns', None):
      print(df)
```

	theory	value	count
0	9/11 was inside job	QAnon	27
1	9/11 was inside job	911Truth	24
2	9/11 was inside job	WWG1WGA	19
3	9/11 was inside job	TheGreatAwakening	16
4	9/11 was inside job	WTC7	16
5	9/11 was inside job	QAnonGermany	14
6	9/11 was inside job	WWG1WGA_WORLDWIDE	14
7	9/11 was inside job	911truth	8
8	9/11 was inside job	Qanon	5
9	9/11 was inside job	demolition	4
10	COVID19 and microchipping	SorosVirus	4
11	COVID19 and microchipping	coronavirus	4
12	COVID19 is spreaded by 5G	5GCoronavirus	85
13	COVID19 is spreaded by 5G	COVID19	25
14	COVID19 is spreaded by 5G	5G	18
15	COVID19 is spreaded by 5G	coronavirus	15
16	COVID19 is spreaded by 5G	QAnon	15
17	COVID19 is spreaded by 5G	qanon	9
18	COVID19 is spreaded by 5G	Agenda21	9
19	COVID19 is spreaded by 5G	COVID19	7
20	COVID19 is spreaded by 5G	Covid_19	6
21	COVID19 is spreaded by 5G	WWG1WGA	6
22	Deep State	DeepStateFauci	99
23	Deep State	Qanon	66
24	Deep State	Fauci	64
25	Deep State	RT	63
26	Deep State	QAnon	49
27	Deep State	DeepStateVirus	46
28	Deep State	TheGreatAwakening	46
29	Deep State	qanon	33
30	Deep State	wwg1wga	30
31	Deep State	COVID19	30
32	FlatEarth	FlatEarth	47
33	FlatEarth	QAnon	42
34	FlatEarth	flatearth	23
35	FlatEarth	NASA	22

36	FlatEarth	coronavirus	21
37	FlatEarth	ConspiracyTheory	17
38	FlatEarth	Qanon	16
39	FlatEarth	Woke	15
40	FlatEarth	NoVaccines	15
41	FlatEarth	Vaxxed	15
42	Global Warming is HOAX	ClimateChangeHoax	16
43	Global Warming is HOAX	coronavirus	10
44	Global Warming is HOAX	Agenda21	6
45	Global Warming is HOAX	QAnon	2
46	Global Warming is HOAX	FastandFurious	2
47	Global Warming is HOAX	WWG1WGA	2
48	Global Warming is HOAX	ObamaGate	2
49	Global Warming is HOAX	LockHimUp	2
50	Global Warming is HOAX	FakeNews	2
51	Global Warming is HOAX	TurkeyEarthquake	2
52	Chemtrails	chemtrails	155
53	Chemtrails	QAnon	106
54	Chemtrails	Qanon	57
55	Chemtrails	5G	56
56	Chemtrails	WWG1WGA	55
57	Chemtrails	BillGates	44
58	Chemtrails	NWO	44
59	Chemtrails	Chemtrails	39
60	Chemtrails	TheGreatAwakening	36
61	Chemtrails	HAARP	27
62	Illuminati	Illuminati	389
63	Illuminati	WWG1WGA	314
64	Illuminati	QAnon	300
65	Illuminati	TheGreatAwakening	207
66	Illuminati	Cabal	168
67	Illuminati	coronavirus	158
68	Illuminati	illuminati	125
69	Illuminati	qanon	124
70	Illuminati	nwo	117
71	Illuminati	2012olympicceremony	116
72	Moon landing is fake	QAnon	17
73	Moon landing is fake	NASA	11
74	Moon landing is fake	ObamaGate	11
75	Moon landing is fake	moonhoax	11
76	Moon landing is fake	researchflatearth	11
77	Moon landing is fake	spaceisfake	11
78	Moon landing is fake	Birther	6
79	Moon landing is fake	UraniumOne	6
80	Moon landing is fake	Soros	6
81	Moon landing is fake	chemtrails	6
82	New world order	Agenda21	370
83	New world order	coronavirus	262

84	New world order	agenda21	220
85	New world order	agenda2030	187
86	New world order	QAnon	173
87	New world order	Covid19	90
88	New world order	AGENDA21	81
89	New world order	NWO	79
90	New world order	CORONAVIRUS	78
91	New world order	AGENDA30	65
92	Pizzagate conspiracy theory	PIZZAGATEISREAL	356
93	Pizzagate conspiracy theory	QAnon	294
94	Pizzagate conspiracy theory	Qanon	290
95	Pizzagate conspiracy theory	PizzaGateIsReal	279
96	Pizzagate conspiracy theory	WWG1WGA	272
97	Pizzagate conspiracy theory	TheGreatAwakening	262
98	Pizzagate conspiracy theory	MAGA	210
99	Pizzagate conspiracy theory	qanon	180
100	Pizzagate conspiracy theory	KAG	172
101	Pizzagate conspiracy theory	PizzaGate	168
102	Qanon	QAnon	414745
103	Qanon	WWG1WGA	200157
104	Qanon	Qanon	130006
105	Qanon	MAGA	70378
106	Qanon	qanon	65599
107	Qanon	QANON	56674
108	Qanon	Trump	45340
109	Qanon	KAG	36806
110	Qanon	TheGreatAwakening	35410
111	Qanon	WWG1WGA WORLDWIDE	33598
112	Reptilian conspiracy theory	Reptilians	17
113	Reptilian conspiracy theory	TheGreatAwakening	16
114	Reptilian conspiracy theory	SSP	15
115	Reptilian conspiracy theory	QANON	15
116	Reptilian conspiracy theory	VictoryOfTheLight	15
117	Reptilian conspiracy theory	Romance	15
118	Reptilian conspiracy theory	SecretSpaceProgram	15
119	Reptilian conspiracy theory	SpaceMarines	15
120	Reptilian conspiracy theory	DragonShips	15
121	Reptilian conspiracy theory	FULLDisclosure	15
122	The virus escaped from a Chinese lab	CCPVirus	1301
123	The virus escaped from a Chinese lab	ChinaLiedPeopleDied	956
124	The virus escaped from a Chinese lab	China	370
125	The virus escaped from a Chinese lab	COVID19	262
126	The virus escaped from a Chinese lab	QAnon	258
127	The virus escaped from a Chinese lab	CCP	248
128	The virus escaped from a Chinese lab	coronavirus	237
129	The virus escaped from a Chinese lab	ChinaVirus	178
130	The virus escaped from a Chinese lab	ChineseVirus	173
131	The virus escaped from a Chinese lab	WuhanVirus	164

[: