reddit_time_dist

August 21, 2021

[1]:

```
import sys

sys.path.append('../..')
from plotting.matplotlib_setup import configure_latex, savefig,␣
 ↪set_size_decorator, savefig, thiner_border

tex_dir, images_dir = 'porocilo/main.tex', 'porocilo/images'

configure_latex(style=['science', 'notebook'], global_save_path=images_dir)

%config InlineBackend.figure_format = 'pdf'
```

[2]:

```
import os

sys.path.insert(0, os.getcwd() + '/reddit_download')
```

[3]:

```
from reddit_download.RWV.pushshift.time_utils import timestamp_to_utc

from datetime import datetime
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[4]:

```
from reddit_download.RWV.pushshift.utils import build_df,␣
 ↪apply_df_time_transforms

df = build_df(content_type='comment', file_path=os.getcwd() + '/
 ↪reddit_download')
```

[5]:

```
df["datetime"] = df["created_utc"].apply(datetime.fromtimestamp)
df = df.rename(columns={"created_utc": "timestamp"})
```

[6]:

```
ind = df[df['author'] == '[deleted]'].index
df.drop(ind, inplace=True)

ind = df[df['author'] == 'AutoModerator'].index
df.drop(ind, inplace=True)
```

[7]:

```
def get_dates(df):
    return df['datetime'].apply(lambda datetime: datetime.date())


def post_time_dist(df, sub):
    df_ = df[df['subreddit'] == sub].copy()
    dates = get_dates(df_)
    df_['date'] = dates

    post_time_dist_dct = {}
    for d in dates.unique():
        post_time_dist_dct[str(d)] = df_[df_['date'] == d]['time_in_day'].values

    return post_time_dist_dct


# subreddits = df['subreddit'].unique()
# post_time_dist_dct = post_time_dist(df, sub=subreddits[0])
```

[8]:

```
weekdays = {0:'monday', 1:'tuesday', 2:'wednesday', 3:'thursday', 4:'friday', 5:
 ↪'saturday', 6:'sunday'}

def day_hists(df):
    unique_subs = df['subreddit'].unique()
    results = []

    for sub in unique_subs:
        results.append(post_time_dist(df, sub))

    unique_dates = get_dates(df).unique()

    for d in unique_dates:
        plt.title(f'{weekdays[d.weekday()]} {d}')
        for i in range(len(unique_subs)):
            try:
                y = results[i][str(d)]
                plt.hist(y, alpha=0.9, histtype='step',␣
 ↪label=f'{unique_subs[i]}, sum={len(y)}')
            except KeyError:
                pass

        plt.legend()
        plt.show()

# day_hists(df)
```

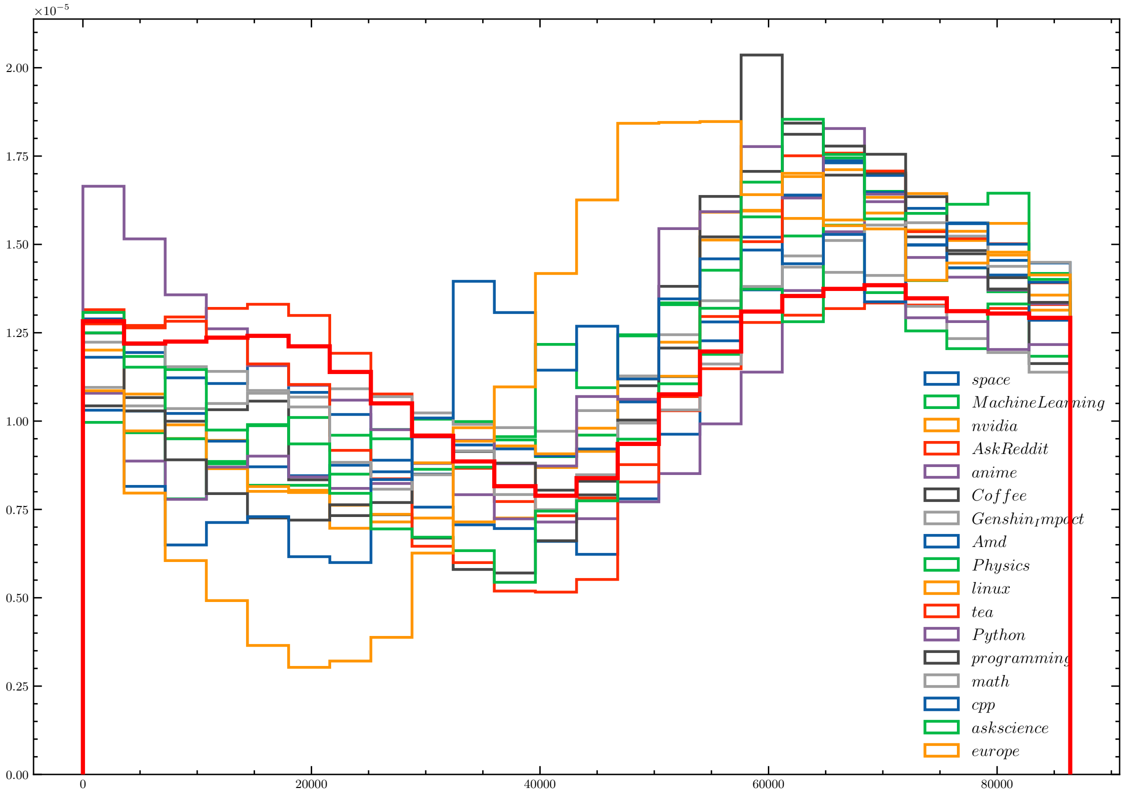
[9]:

```
def get_time_dist(df):
    ys = []

    for sub in df['subreddit'].unique():
        df_ = df[df['subreddit'] == sub]
        y = df_['time_in_day'].values
        ys.append(y)
        print(f"{len(y)} {sub}")

    flat_ys = []
    for sublist in ys:
        for item in sublist:
            flat_ys.append(item)

    return ys, flat_ys
```

[10]:

```
from RWV.pushshift.time_utils import seconds_in_day
df["time_in_day"] = df["datetime"].apply(seconds_in_day)

ys, flat_ys = get_time_dist(df)

plt.figure(figsize=(14, 10))
for (sub, y) in zip(df['subreddit'].unique(), ys):
    plt.hist(y, histtype='step', lw=2, density=True, bins=24, label=f'${sub}$')

plt.hist(flat_ys, histtype='step', bins=24, density=True, lw=3, zorder=100,␣
 ↪color='r')
plt.legend(fontsize=12, loc='lower right')
plt.show()
```

```
100429 space
12323 MachineLearning
75684 nvidia
6747049 AskReddit
548727 anime
33182 Coffee
2023068 Genshin_Impact
100277 Amd
13832 Physics
43657 linux
18628 tea
17950 Python
65955 programming
29248 math
10053 cpp
24828 askscience
66736 europe
```

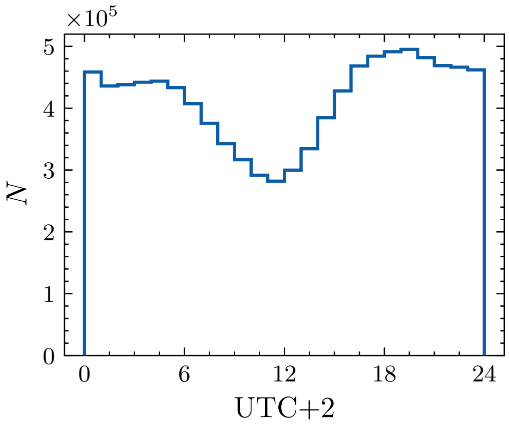


[11]:

```
sub_lst = list(df['subreddit'].unique())
```

[12]:

```
fig, ax = set_size_decorator(plt.subplots, fraction=0.5, ratio='4:3')(1, 1)

ax.hist(flat_ys, histtype='step', bins=24, lw=1.2, zorder=10)

x_ = np.arange(0, 86400, 1)
x = x_[::len(x_)//4]
x = np.append(x, x_[-1] + 1)

ax.set_xticks(x)
ax.set_xticklabels((x / (60 * 60)).astype(int))

ax.ticklabel_format(style='sci', axis='y', scilimits=(0, 0))

ax.set_xlabel('UTC+2')
ax.set_ylabel('$N$')

# savefig('reddit_times_dist_all', tight_layout=False)

plt.show()
```



[13]:

```
y = np.array(flat_ys)
# y = ys[sub_lst.index('europe')]

hist, bin_edges = np.histogram(y, bins=24)
```

[14]:

```
bin_edges = bin_edges[:-1]
```

[15]:

```
shift = - 6 * 60 * 60

b = bin_edges + shift
```

[16]:

```
ind_neg = np.where(b < 0)[0]
ind_pos = np.where(b >= 0)[0]
ind = np.concatenate((ind_pos, ind_neg))
```

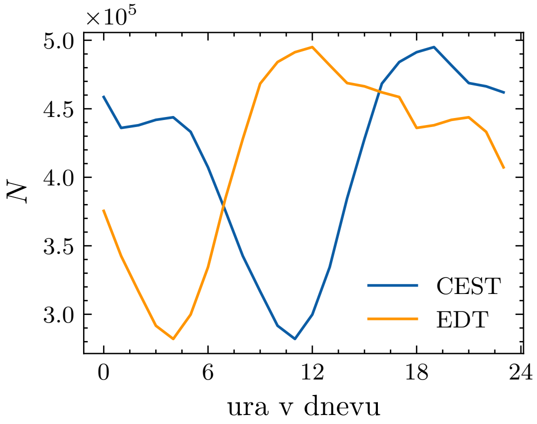
[17]:

```
new_hist = hist[ind]
```

[18]:

```
fig, ax = set_size_decorator(plt.subplots, fraction=0.5, ratio='4:3')(1, 1)

ax.plot(bin_edges, hist, lw=1, c='C0', label='CEST')
ax.plot(bin_edges, new_hist, lw=1, c='C2', label='EDT')

x_ = np.arange(0, 86400, 1)
x = x_[::len(x_)//4]
x = np.append(x, x_[-1] + 1)

ax.set_xticks(x)
ax.set_xticklabels((x / (60 * 60)).astype(int))

ax.ticklabel_format(style='sci', axis='y', scilimits=(0, 0))

ax.set_xlabel('ura v dnevu')
ax.set_ylabel('$N$')

ax.legend()

# savefig('reddit_timezones_dist', tight_layout=False)
```

```
[18]: <matplotlib.legend.Legend at 0x7f74effe4e80>
```



```
[ ]: df_posts = build_df(content_type='post', file_path=os.getcwd() + '/
      ↪reddit_download')
```

```
[ ]: df_posts["datetime"] = df_posts["created_utc"].apply(datetime.fromtimestamp)
     df_posts = df_posts.rename(columns={"created_utc": "timestamp"})
     df_posts["time_in_day"] = df_posts["datetime"].apply(seconds_in_day)
```

```
[ ]: ys, flat_ys = get_time_dist(df_posts)
```

```
[ ]: plt.figure(figsize=(14, 10))
     for (sub, y) in zip(df_posts['subreddit'].unique(), ys):
         plt.hist(y, histtype='step', lw=2, density=True, bins=24, label=f'${sub}$')

     plt.legend(fontsize=12, loc='lower right')
```

```
[ ]: plt.hist(flat_ys, histtype='step', bins=24)
     plt.show()
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```