A Day Has Only 24±1 Hours

import pytz

Miroslav Šedivý

eumiro

A Day Has Only 24±1 Hours

A Day Has Only 24±1 Hours

- check the time

A Day Has Only 24±1 Hours

- check the time
- don't check the time too often

A Day Has Only 24±1 Hours

- check the time
- don't check the time too often
- check what your government does

A Day Has Only 24±1 Hours

- check the time
- don't check the time too often
- check what your government does
- TIME ZONES!

Miroslav Šedivý

['mɪrɔslaʊ 'ʃɛɟɪviː]

- `@eumiro`
- born in Bratislava, Czechoslovakia (`TZ=Europe/Bratislava`)
- M.Sc. at INSA Lyon, France (`TZ=Europe/Paris`)
- Senior Software Architect in Karlsruhe, Germany (`TZ=Europe/Berlin`)

Miroslav Šedivý

['mɪrɔslaʊ 'ʃɛɟɪviː]

- @eumiro
- born in Bratislava, Czechoslovakia (`TZ=Europe/Bratislava`)
- M.Sc. at INSA Lyon, France (`TZ=Europe/Paris`)
- Senior Software Architect in Karlsruhe, Germany (`TZ=Europe/Berlin`)
- using Python to make the sun shine and the wind blow (in many timezones)

Friday, 2nd November 2018

15:03 MSK

(Saint Petersburg, Russia)

```
>>> import time
>>> time.time()
1541160180.0
```

```
>>> import time
>>> time.time()
1541160180.0
```

```
>>> import datetime
>>> datetime.datetime.now()
datetime.datetime(2018, 11, 2, 15, 3, 0, 0)
```

```
>>> import time
>>> time.time()
1541160180.0
```

```
>>> import datetime
>>> datetime.datetime.now()
datetime.datetime(2018, 11, 2, 15, 3, 0, 0)
```

```
datetime.datetime(2018, 11, 2, 12, 3, 0, 0)  # on a server set to UTC
```

```
>>> import time
>>> time.time()
1541160180.0
```

```
>>> import datetime
>>> datetime.datetime.now()
datetime.datetime(2018, 11, 2, 15, 3, 0, 0)
```

```
datetime.datetime(2018, 11, 2, 12, 3, 0, 0)  # on a server set to UTC
```

```
>>> datetime.datetime.utcnow()
datetime.datetime(2018, 11, 2, 12, 3, 0, 0)
```

```
>>> import time
>>> time.time()
1541160180.0
```

```
>>> import datetime
>>> datetime.datetime.now()
datetime.datetime(2018, 11, 2, 15, 3, 0, 0)
```

```
datetime.datetime(2018, 11, 2, 12, 3, 0, 0)  # on a server set to UTC
```

```
>>> datetime.datetime.utcnow()
datetime.datetime(2018, 11, 2, 12, 3, 0, 0)
```

```
>>> datetime.datetime.now(datetime.timezone.utc)
datetime.datetime(2018, 11, 2, 12, 3, 0, 0, tzinfo=datetime.timezone.utc)
```

```
>>> import time
>>> time.time()
1541160180.0
```

```
>>> import datetime
>>> datetime.datetime.now()
datetime.datetime(2018, 11, 2, 15, 3, 0, 0)
```

```
datetime.datetime(2018, 11, 2, 12, 3, 0, 0)  # on a server set to UTC
```

```
>>> datetime.datetime.utcnow()
datetime.datetime(2018, 11, 2, 12, 3, 0, 0)
```

```
>>> datetime.datetime.now(datetime.timezone.utc)
datetime.datetime(2018, 11, 2, 12, 3, 0, 0, tzinfo=datetime.timezone.utc)
```

```
>>> datetime.datetime.now(datetime.timezone(datetime.timedelta(hours=3)))
datetime.datetime(2018, 11, 2, 15, 3, 0, 0, tzinfo=datetime.timezone(datetime.timedelta(0, 10800)))
```

```
>>> import time
>>> time.time()
1541160180.0
```

```
>>> import datetime
>>> datetime.datetime.now()
datetime.datetime(2018, 11, 2, 15, 3, 0, 0)
```

```
datetime.datetime(2018, 11, 2, 12, 3, 0, 0)  # on a server set to UTC
```

```
>>> datetime.datetime.utcnow()
datetime.datetime(2018, 11, 2, 12, 3, 0, 0)
```

```
>>> datetime.datetime.now(datetime.timezone.utc)
datetime.datetime(2018, 11, 2, 12, 3, 0, 0, tzinfo=datetime.timezone.utc)
```

```
>>> datetime.datetime.now(datetime.timezone(datetime.timedelta(hours=3)))
datetime.datetime(2018, 11, 2, 15, 3, 0, 0, tzinfo=datetime.timezone(datetime.timedelta(0, 10800)))
```

```
>>> import pytz
>>> datetime.datetime.now(pytz.timezone('Europe/Moscow'))
datetime.datetime(2018, 11, 2, 15, 3, 0, 0, tzinfo=<DstTzInfo 'Europe/Moscow' MSK+3:00:00 STD>)
```

```
>>> today = datetime.datetime.utcnow().strftime('%Y-%m-%d')
2018-11-02
```

```
>>> today = datetime.datetime.utcnow().strftime('%Y-%m-%d')
2018-11-02
```

```
>>> yesterday = (datetime.datetime.utcnow() - datetime.timedelta(days=1)).strftime('%Y-%m-%d')
2018-11-01
```

```
>>> today = datetime.datetime.utcnow().strftime('%Y-%m-%d')
2018-11-02
```

```
>>> yesterday = (datetime.datetime.utcnow() - datetime.timedelta(days=1)).strftime('%Y-%m-%d')
2018-11-01
```

```
>>> now = datetime.datetime.utcnow()
>>> today = now.strftime('%Y-%m-%d')
>>> yesterday = (now - datetime.timedelta(days=1)).strftime('%Y-%m-%d')
2018-11-02
2018-11-01
```

```
>>> start = datetime.datetime.utcnow()
>>> expensive_operation()
>>> end = datetime.datetime.utcnow()
>>> elapsed = (end - start).total_seconds()
```

```
>>> start = datetime.datetime.utcnow()
>>> expensive_operation()
>>> end = datetime.datetime.utcnow()
>>> elapsed = (end - start).total_seconds()
```

```
>>> start = time.time()
>>> expensive_operation()
>>> end = time.time()
>>> elapsed = end - start
```

```
>>> start = datetime.datetime.utcnow()
>>> expensive_operation()
>>> end = datetime.datetime.utcnow()
>>> elapsed = (end - start).total_seconds()
```

```
>>> start = time.time()
>>> expensive_operation()
>>> end = time.time()
>>> elapsed = end - start
```

```
>>> start = time.monotonic()
>>> expensive_operation()
>>> end = time.monotonic()
>>> elapsed = end - start
```

```
>>> start = datetime.datetime.utcnow()
>>> expensive_operation()
>>> end = datetime.datetime.utcnow()
>>> elapsed = (end - start).total_seconds()
```

```
>>> start = time.time()
>>> expensive_operation()
>>> end = time.time()
>>> elapsed = end - start
```

```
>>> start = time.monotonic()
>>> expensive_operation()
>>> end = time.monotonic()
>>> elapsed = end - start
```

```
>>> time.monotonic_ns()  # Python 3.7+
```

Time

- local solar time (equation of time)

Time

- local solar time (equation of time)
- Greenwich

Time

- local solar time (equation of time)
- Greenwich
- France:
  - 1891: l'heure de Paris (railroads at -00:05)
  - 1911: Greenwich Time

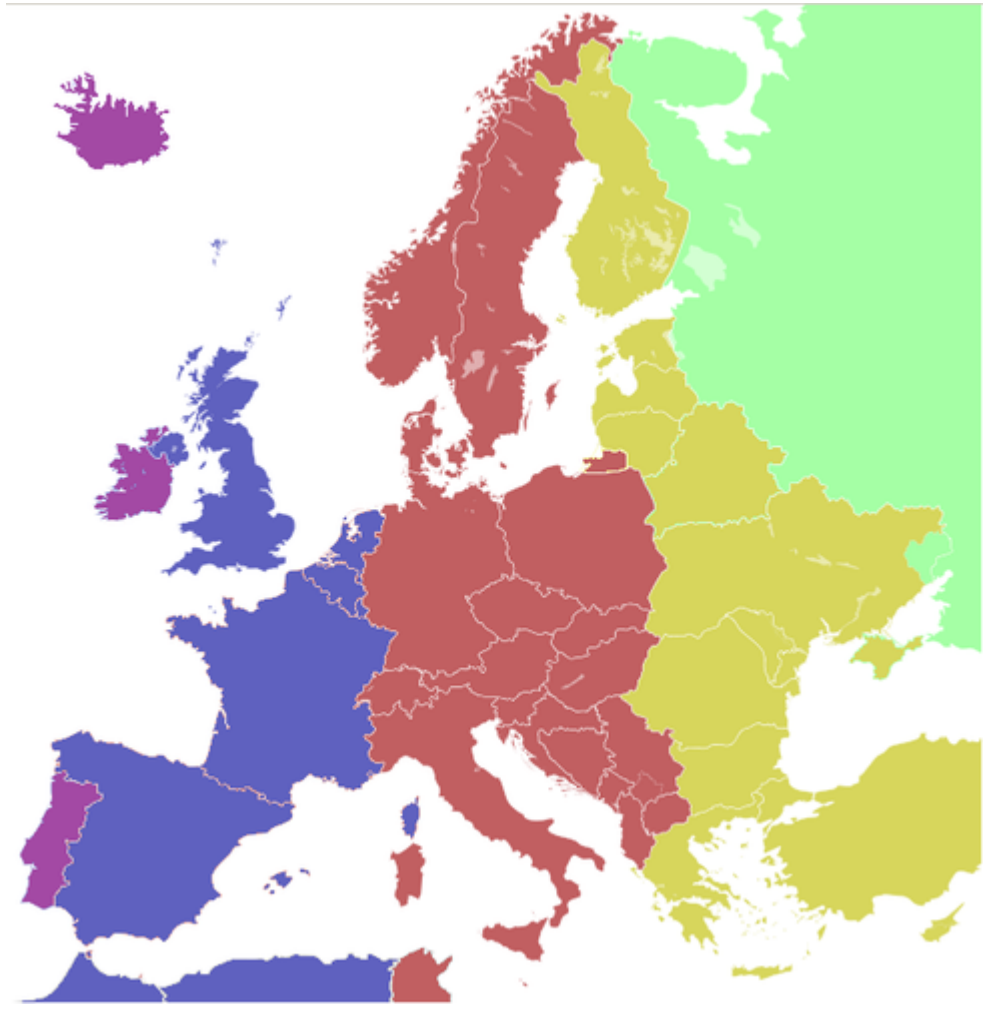
Time

- local solar time (equation of time)
- Greenwich

- France:

  - 1891: l'heure de Paris (railroads at -00:05)
  - 1911: Greenwich Time

- Greenwich and 25 time zones (-12 … +12)



Source: StuartBrady / Wikipedia, CC BY-SA 3.0

Time zones

- no central authority (countries, regions, counties, …)
- IANA.org (Internet Assigned Numbers Authority) → `tzdata`

Time zones

- no central authority (countries, regions, counties, …)
- IANA.org (Internet Assigned Numbers Authority) → `tzdata`

```
>>> len(pytz.all_timezones)
591
```

Time zones

- no central authority (countries, regions, counties, …)
- IANA.org (Internet Assigned Numbers Authority) → `tzdata`

```
>>> len(pytz.all_timezones)
591
```

```
>>> len(pytz.common_timezones)
439
```

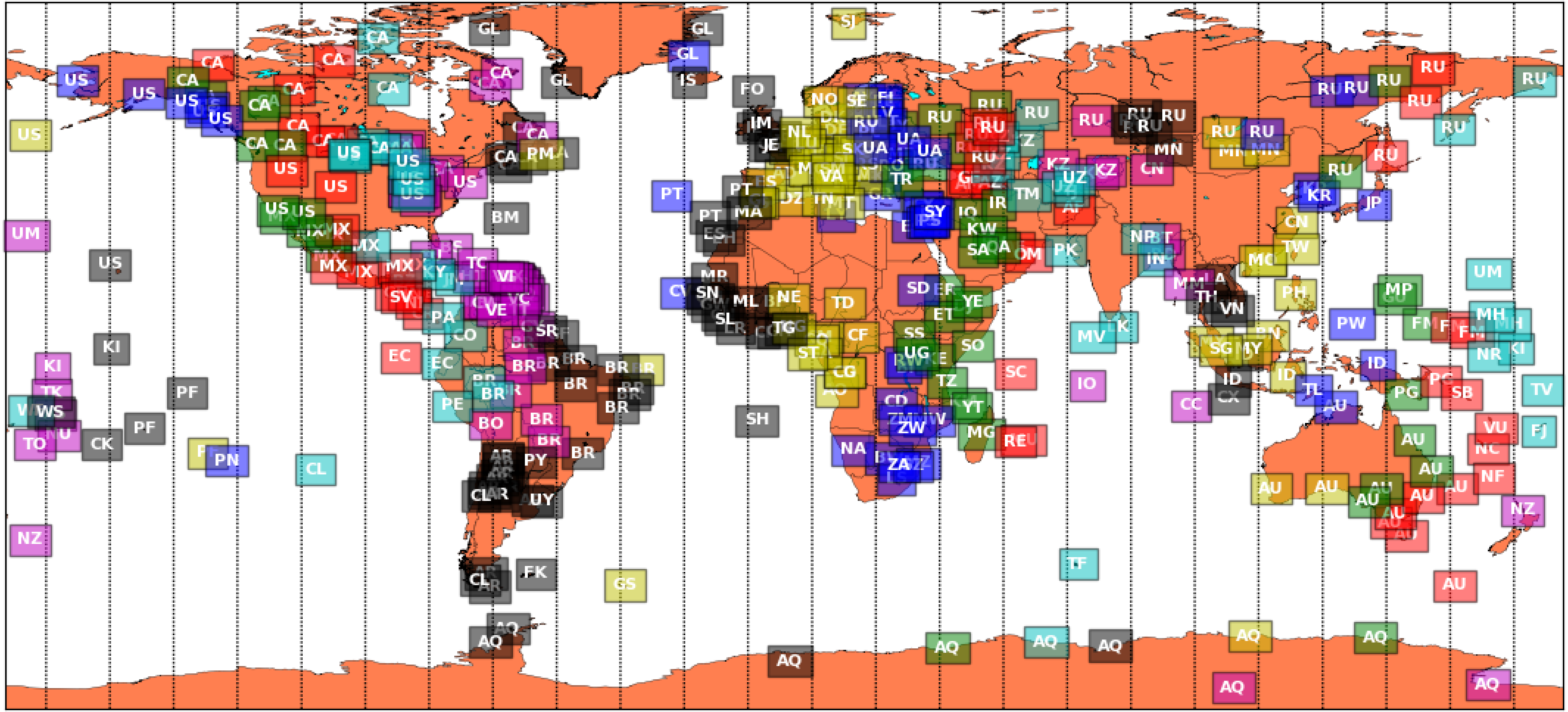
Time zones

- no central authority (countries, regions, counties, …)
- IANA.org (Internet Assigned Numbers Authority) → `tzdata`

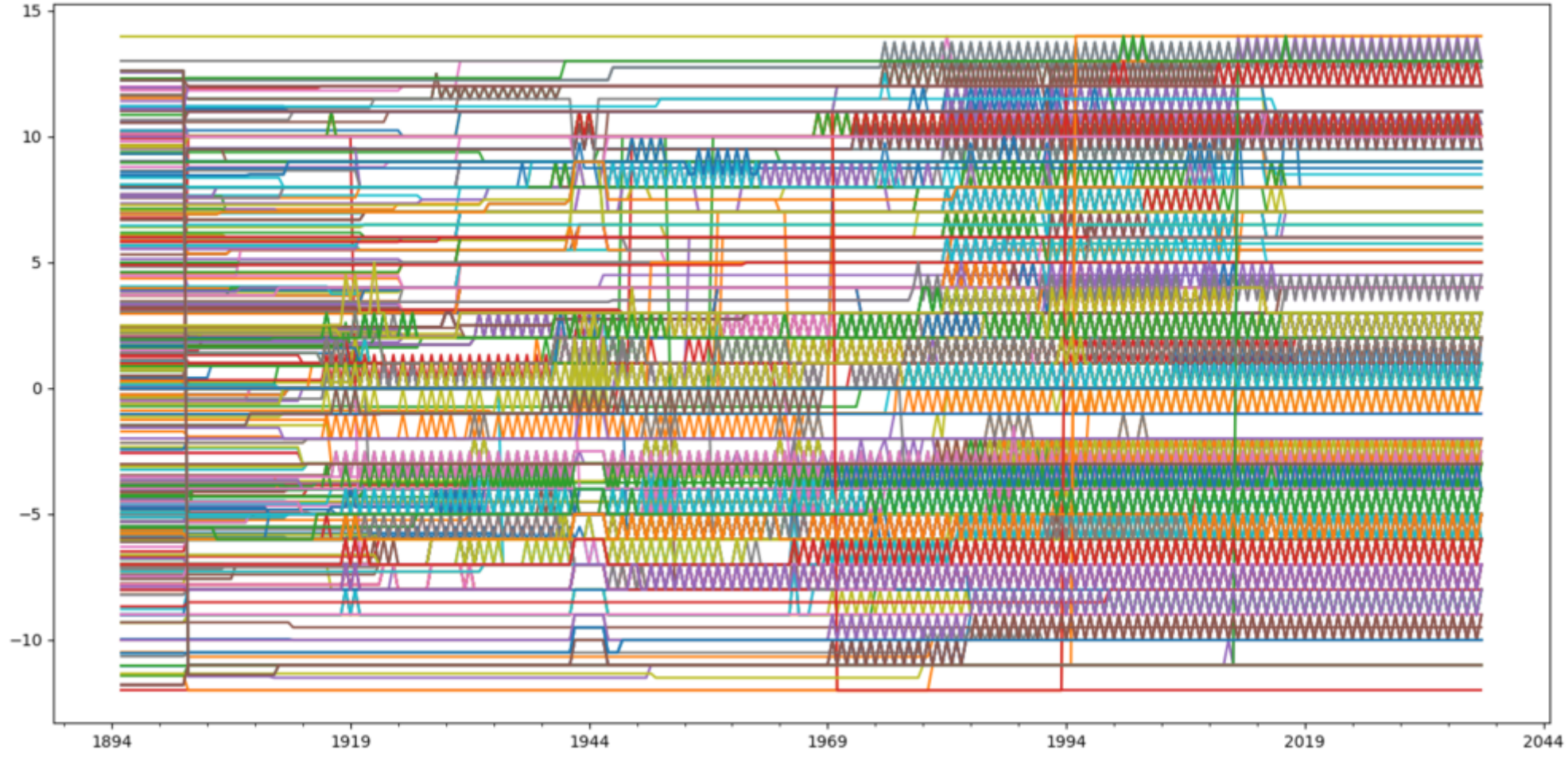
```
>>> len(pytz.all_timezones)
591
```

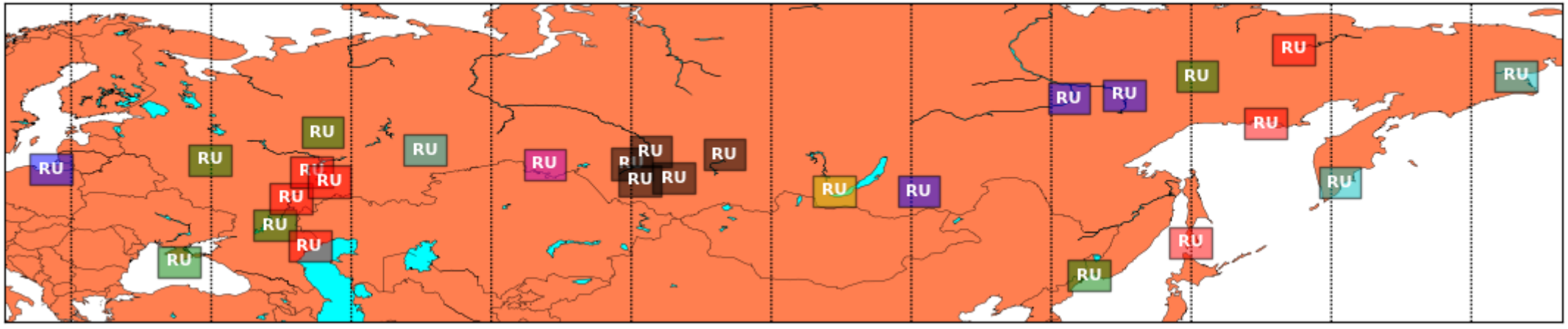
```
>>> len(pytz.common_timezones)
439
```

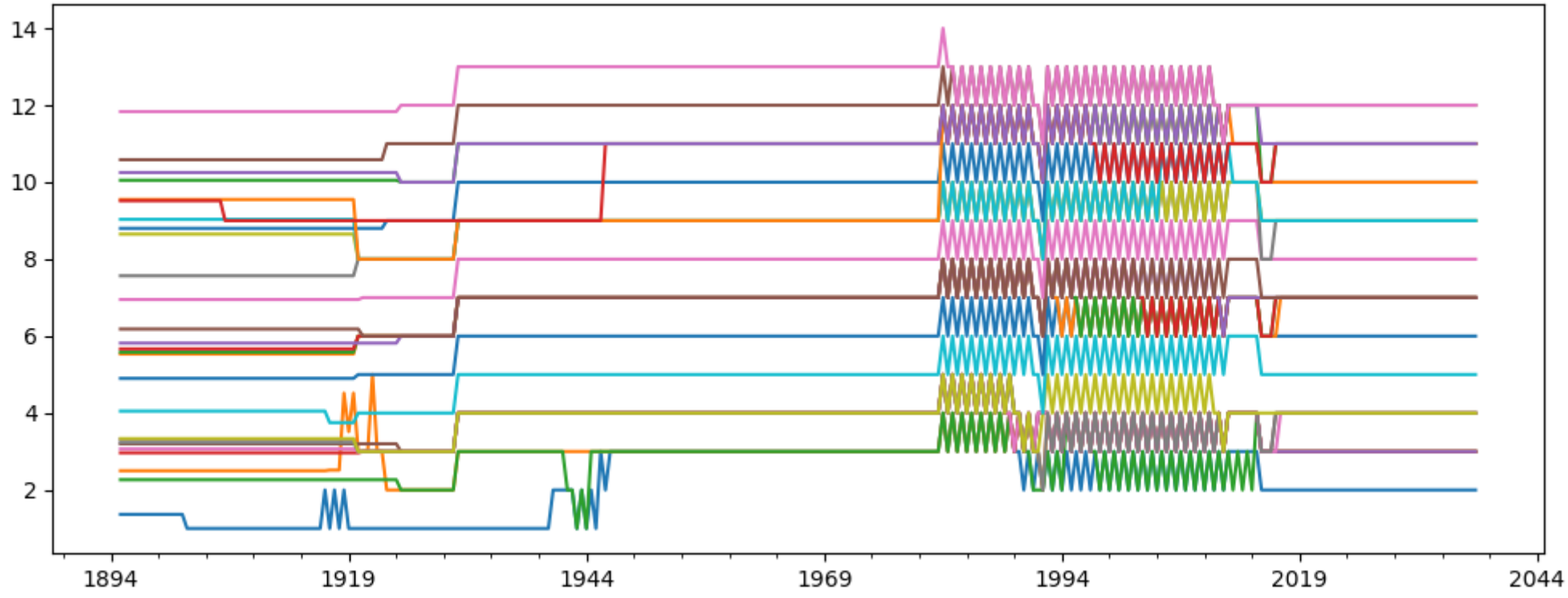
- America (147), Asia (82), Europe (60), Africa (52), Australia (12), Antarctica (11)
- Pacific (38), Indian (11), Atlantic (10), Arctic (1)
- US (7), Canada (6)
- GMT, UTC



SJ
GL
GL
GL
GL
IS
FO
CA
CA
CA
CA
CA
CA
CA
CA
CA
CA
US
US
US
US
US
US
US
US
US
US
UM
US
BM
PT
PT
MA
NO
SE
NL
IM
JE
UA
UA
TR
VA
SY
IQ
IR
KW
SA
OM
PK
TM
UZ
KZ
KZ
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
MN
CN
CN
KR
JP
TW
MO
PH
VN
NP
IN
MX
MX
MX
MX
MX
SV
KY
TC
VI
VE
PA
CO
SR
EC
EC
PE
BR
BR
BR
BR
BR
BR
BR
BR
BR
BO
PY
UY
CL
FK
GS
AQ
AQ
AQ
AQ
AQ
AQ
AQ
AQ
AQ
AQ
SN
ML
NE
TD
SL
TG
CF
ST
CG
SD
YE
ET
SS
SO
UG
TZ
CD
ZW
YT
MG
NA
ZA
RE
SC
IO
MV
SH
TF
KI
KI
WS
TO
CK
PF
PF
PN
CL
NZ
PW
MP
UM
MH
FM
NR
ID
ID
SG
MY
CC
CX
TL
PG
SB
TV
VU
NC
FJ
AU
AU
AU
AU
AU
AU
AU
NF
NZ


15
10
5
0
−5
−10
1894
1919
1944
1969
1994
2019
2044


RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU
RU


14
12
10
8
6
4
2
1894
1919
1944
1969
1994
2019
2044

```
>>> pytz.timezone('Europe/Moscow')
<DstTzInfo 'Europe/Moscow' LMT+2:30:00 STD>

>>> pytz.timezone('Europe/Berlin')
<DstTzInfo 'Europe/Berlin' LMT+0:53:00 STD>

>>> pytz.timezone('Europe/Paris')
<DstTzInfo 'Europe/Paris' LMT+0:09:00 STD>

>>> pytz.timezone('Europe/London')
<DstTzInfo 'Europe/London' LMT-1 day, 23:59:00 STD>
```

```
>>> pytz.timezone('Europe/Moscow')
<DstTzInfo 'Europe/Moscow' LMT+2:30:00 STD>

>>> pytz.timezone('Europe/Berlin')
<DstTzInfo 'Europe/Berlin' LMT+0:53:00 STD>

>>> pytz.timezone('Europe/Paris')
<DstTzInfo 'Europe/Paris' LMT+0:09:00 STD>

>>> pytz.timezone('Europe/London')
<DstTzInfo 'Europe/London' LMT-1 day, 23:59:00 STD>
```

```
>>> datetime.datetime(2018, 11, 2, 15, 3, tzinfo=pytz.timezone('Europe/Moscow')).astimezone(pytz.utc)
datetime.datetime(2018, 11, 2, 12, 33, tzinfo=<UTC>)
```

```
>>> pytz.timezone('Europe/Moscow')
<DstTzInfo 'Europe/Moscow' LMT+2:30:00 STD>

>>> pytz.timezone('Europe/Berlin')
<DstTzInfo 'Europe/Berlin' LMT+0:53:00 STD>

>>> pytz.timezone('Europe/Paris')
<DstTzInfo 'Europe/Paris' LMT+0:09:00 STD>

>>> pytz.timezone('Europe/London')
<DstTzInfo 'Europe/London' LMT-1 day, 23:59:00 STD>
```

```
>>> datetime.datetime(2018, 11, 2, 15, 3, tzinfo=pytz.timezone('Europe/Moscow')).astimezone(pytz.utc)
datetime.datetime(2018, 11, 2, 12, 33, tzinfo=<UTC>)
```

```
>>> pytz.timezone('Europe/Moscow').localize(datetime.datetime(2018, 11, 2, 15, 3)).astimezone(pytz.utc)
datetime.datetime(2018, 11, 2, 12, 3, tzinfo=<UTC>)
```

tzdata2018e.tar.gz (346k)

```
 52800 Apr 23 01:52 africa
 12934 Mar 14 23:49 antarctica
138358 Apr 30 21:28 asia
 76124 Apr  1 18:27 australasia
  4543 Jul 16  2017 backward
 21945 Mar 18 22:48 backzone
  5567 Oct  2  2017 calendars
  1008 Jun  2  2017 checklinks.awk
  4203 Jan 30 18:16 checktab.awk
  2895 Mar  2 18:43 CONTRIBUTING
  2740 Aug 24  2016 etcetera
168081 Apr 23 01:52 europe
   367 Aug 24  2016 factory
  4445 Feb 28  2017 iso3166.tab
  2196 Jan 15 18:19 leapseconds
  2380 Jul 18  2017 leapseconds.awk
 10666 Jan 15 18:19 leap-seconds.list
   252 May 25  2017 LICENSE
 38969 Apr 30 21:46 Makefile
156727 May  2 08:44 NEWS
150084 Mar 20 23:31 northamerica
  1182 Jun 20  2014 pacificnew
  2340 Jan 13 08:05 README
 86505 Feb 21 00:50 southamerica
  1538 Jun 16  2014 systemv
 53216 Apr 15 20:02 theory.html
     6 May  2 08:44 version
   678 Jun 18  2014 yearistype.sh
  2869 Apr 23 01:52 ziguard.awk
  4207 Jan 23 23:54 zishrink.awk
 17781 Feb 16 22:59 zone1970.tab
  1409 Aug 17  2014 zoneinfo2tdf.pl
 19165 Feb 16 18:21 zone.tab
```

Europe/Moscow

```
Zone Europe/Moscow  2:30:17 -       LMT      1880
                    2:30:17 -       MMT      1916 Jul  3  # Moscow Mean Time
                    2:31:19 Russia  %s       1919 Jul  1  0:00u
                    3:00    Russia  %s       1921 Oct
                    3:00    Russia  MSK/MSD  1922 Oct
                    2:00    -       EET      1930 Jun 21
                    3:00    Russia  MSK/MSD  1991 Mar 31  2:00s
                    2:00    Russia  EE%sT    1992 Jan 19  2:00s
                    3:00    Russia  MSK/MSD  2011 Mar 27  2:00s
                    4:00    -       MSK      2014 Oct 26  2:00s
                    3:00    -       MSK
```

Europe/Istanbul

```
Zone  Europe/Istanbul  1:55:52  -       LMT      1880
                       1:56:56  -       IMT      1910 Oct # Istanbul Mean Time?
                       2:00     Turkey  EE%sT    1978 Oct 15
                       3:00     Turkey  +03/+04  1985 Apr 20
                       2:00     Turkey  EE%sT    2007
                       2:00     EU      EE%sT    2011 Mar 27  1:00u
                       2:00     -       EET      2011 Mar 28  1:00u
                       2:00     EU      EE%sT    2014 Mar 30  1:00u
                       2:00     -       EET      2014 Mar 31  1:00u
                       2:00     EU      EE%sT    2015 Oct 25  1:00u
                       2:00     1:00    EEST     2015 Nov  8  1:00u
                       2:00     EU      EE%sT    2016 Sep  7
                       3:00     -       +03
```

Europe/Istanbul

```
Zone  Europe/Istanbul  1:55:52  -       LMT      1880
                       1:56:56  -       IMT      1910 Oct # Istanbul Mean Time?
                       2:00     Turkey  EE%sT    1978 Oct 15
                       3:00     Turkey  +03/+04  1985 Apr 20
                       2:00     Turkey  EE%sT    2007
                       2:00     EU      EE%sT    2011 Mar 27  1:00u
                       2:00     -       EET      2011 Mar 28  1:00u
                       2:00     EU      EE%sT    2014 Mar 30  1:00u
                       2:00     -       EET      2014 Mar 31  1:00u
                       2:00     EU      EE%sT    2015 Oct 25  1:00u
                       2:00     1:00    EEST     2015 Nov  8  1:00u
                       2:00     EU      EE%sT    2016 Sep  7
                       3:00     -       +03
```

(2011-03-10): […] Turkey will change into summer time zone (GMT+3) on March 28, 2011 at 3:00 a.m. instead of March 27. This change is due to a nationwide exam on 27th. [URL] Turkish: [URL]

```
2:00     EU      EE%sT    2014 Mar 30  1:00u
2:00     -       EET      2014 Mar 31  1:00u
```

[…] (2014-02-14): The DST for Turkey has been changed for this year because of the Turkish Local election.... [URL] ... so Turkey will move clocks forward one hour on March 31 at 3:00 a.m.

[…] (2014-04-15): Having landed on a flight from the states to Istanbul (via AMS) on March 31, I can tell you that NOBODY (even the airlines) respected this timezone DST change delay. Maybe the word just didn't get out in time.

[…] (2014-06-15): The press reported massive confusion, as election officials obeyed the rule change but cell phones (and airline baggage systems) did not. See: [URL from 2014-03-30] I guess the best we can do is document the official time.

```
2:00     EU      EE%sT    2015 Oct 25  1:00u
2:00     1:00    EEST     2015 Nov  8  1:00u
```

[…] (2015-09-29): It's officially announced now by the Ministry of Energy. Turkey delays winter time to 8th of November 04:00 [URL]

BBC News (2015-10-25): Confused Turks are asking "what's the time?" after automatic clocks defied a government decision … "For the next two weeks #Turkey is on EEST… Erdogan Engineered Standard Time," said Twitter user @aysekarahasan. [URL]

```
2:00     EU      EE%sT    2016 Sep  7
3:00     -       +03
```

[…] (2016-09-08): Turkey will stay in Daylight Saving Time even in winter…. [URL]

[…] (2016-09-07): The change is permanent, so this is the new standard time in Turkey. It takes effect today, which is not much notice.

[…] (2017-10-28): Turkey will go back to Daylight Saving Time starting 2018-10. [URL]

[…] (2017-11-08): … today it was announced that the DST will become "continuous": [URL]

[…] (2017-11-08): Although Google Translate misfires on that source, it looks like Turkey reversed last month's decision, and so will stay at +03.

America/Caracas

```
Zone  America/Caracas  -4:27:44  -  LMT    1890
                       -4:27:40  -  CMT    1912 Feb 12 # Caracas Mean Time?
                       -4:30     -  -0430  1965 Jan  1  0:00
                       -4:00     -  -04    2007 Dec  9  3:00
                       -4:30     -  -0430  2016 May  1  2:30
                       -4:00     -  -04
```

> […] (2016-04-15): Clocks advance 30 minutes on 2016-05-01 at 02:30.… […] [URL from Reuters]
>
> […] (2016-04-20): … published in the official Gazette [2016-04-18], here: [URL from .ve]

America/Port-au-Prince

```
Rule  Haiti  2005  2006  -  Apr Sun>=1  0:00  1:00  D
Rule  Haiti  2005  2006  -  Oct lastSun 0:00  0     S
Rule  Haiti  2012  2015  -  Mar Sun>=8  2:00  1:00  D
Rule  Haiti  2012  2015  -  Nov Sun>=1  2:00  0     S
Rule  Haiti  2017  max   -  Mar Sun>=8  2:00  1:00  D
Rule  Haiti  2017  max   -  Nov Sun>=1  2:00  0     S
```

> […] (2005-04-15) […] wrote me that Haiti is now on DST. I searched for confirmation, and I found a press release on the Web page of the Haitian Consulate in Chicago (2005-03-31), […]
>
> […] (2006-04-04) I have been informed by users that Haiti observes DST this year like last year […]
>
> […] (2012-03-11) According to several news sources, Haiti will observe DST this year, apparently using the same start and end date as USA/Canada. […]
>
> […] (2013-03-10) It appears that Haiti is observing DST this year as well, same rules as US/Canada. They did it last year as well, and it looks like they are going to observe DST every year now… […]
>
> […] (2016-03-12) […] informed us that Haiti are not going on DST this year. […]
>
> […] (2017-03-12) We have received 4 mails from different people telling that Haiti has started DST again today, and this source seems to confirm that, I have not been able to find a more authoritative source: [URL]

America/Montevideo

[…] (1993-11-18) Uruguay wins the prize for the strangest peacetime manipulation of the rules.

1974: Decreto […], citing "the international rise in the price of oil", advanced clocks by 90 minutes (to UT-01:30). Decreto […] returned 60 of those minutes (to UT-02:30), and the remaining 30 minutes followed in Decreto […].

1987: "better use of national tourist attractions" to advance clocks one hour from Monday 1987-12-14 00:00.

(2005-03-11): Uruguay's DST was scheduled to end on Sunday, 2005-03-13, but in order to save energy … it was postponed two weeks….

(2015-06-30): … it looks like they will not be using DST the coming summer: Apparently restaurateurs complained that DST caused people to go to the beach instead of out to dinner.

Asia/Pyongyang

[…] (2015-08-07) According to many news sources, North Korea is going to change to the 8:30 time zone on August 15

[…] (2015-08-15) Bells rang out midnight (00:00) Friday as part of the celebrations. […]

[…] (2018-04-29) North Korea will revert its time zone from UTC+8:30 (PYT; Pyongyang Time) back to UTC+9 (KST; Korea Standard Time).

[…] (2018-04-30) […] It appears to be the front page story at the top in the right-most column.

Asia/Pyongyang

```
>>> datetime.datetime.now(pytz.timezone('Asia/Pyongyang'))
datetime.datetime(2018, 10, 25, 20, 28, 0, tzinfo=<DstTzInfo 'Asia/Pyongyang' KST+8:30:00 STD>)
```

Asia/Pyongyang

```
>>> datetime.datetime.now(pytz.timezone('Asia/Pyongyang'))
datetime.datetime(2018, 10, 25, 20, 28, 0, tzinfo=<DstTzInfo 'Asia/Pyongyang' KST+8:30:00 STD>)
```

```
$ TZ='Asia/Pyongyang' date
Thu Oct 25 20:58:00 KST 2018
```

Asia/Pyongyang

```
>>> datetime.datetime.now(pytz.timezone('Asia/Pyongyang'))
datetime.datetime(2018, 10, 25, 20, 28, 0, tzinfo=<DstTzInfo 'Asia/Pyongyang' KST+8:30:00 STD>)
```

```
$ TZ='Asia/Pyongyang' date
Thu Oct 25 20:58:00 KST 2018
```

```
$ pacman -Qs tz
community/python-pytz 2018.4-1
    Cross platform time zone library for Python
core/tzdata 2018e-1
    Sources for time zone and daylight saving time data
```

Asia/Pyongyang

```
>>> datetime.datetime.now(pytz.timezone('Asia/Pyongyang'))
datetime.datetime(2018, 10, 25, 20, 28, 0, tzinfo=<DstTzInfo 'Asia/Pyongyang' KST+8:30:00 STD>)
```

```
$ TZ='Asia/Pyongyang' date
Thu Oct 25 20:58:00 KST 2018
```

```
$ pacman -Qs tz
community/python-pytz 2018.4-1
    Cross platform time zone library for Python
core/tzdata 2018e-1
    Sources for time zone and daylight saving time data
```

- Android 7: 2016f

Reading data

```
>>> datetime.datetime.strptime('02.11.2018 15:03 +0300', '%d.%m.%Y %H:%M %z')
datetime.datetime(2018, 11, 2, 15, 3, tzinfo=datetime.timezone(datetime.timedelta(seconds=10800)))
```

Reading data

```
>>> datetime.datetime.strptime('02.11.2018 15:03 +0300', '%d.%m.%Y %H:%M %z')
datetime.datetime(2018, 11, 2, 15, 3, tzinfo=datetime.timezone(datetime.timedelta(seconds=10800)))
```

```
>>> from dateutil import parser, tz
>>> tzinfos = {'SP': tz.gettz('Europe/Moscow')}
>>> parser.parse('02.11.2018 15:03 SP', tzinfos=tzinfos, dayfirst=True)

datetime.datetime(2018, 11, 2, 15, 3, tzinfo=tzfile('/usr/share/zoneinfo/Europe/Moscow'))
```

Reading data

```
>>> datetime.datetime.strptime('02.11.2018 15:03 +0300', '%d.%m.%Y %H:%M %z')
datetime.datetime(2018, 11, 2, 15, 3, tzinfo=datetime.timezone(datetime.timedelta(seconds=10800)))
```

```
>>> from dateutil import parser, tz
>>> tzinfos = {'SP': tz.gettz('Europe/Moscow')}
>>> parser.parse('02.11.2018 15:03 SP', tzinfos=tzinfos, dayfirst=True)

datetime.datetime(2018, 11, 2, 15, 3, tzinfo=tzfile('/usr/share/zoneinfo/Europe/Moscow'))
```

Data without timezone info?

- check end of March/October for missing/repeating entries

Reading data

```
>>> datetime.datetime.strptime('02.11.2018 15:03 +0300', '%d.%m.%Y %H:%M %z')
datetime.datetime(2018, 11, 2, 15, 3, tzinfo=datetime.timezone(datetime.timedelta(seconds=10800)))
```

```
>>> from dateutil import parser, tz
>>> tzinfos = {'SP': tz.gettz('Europe/Moscow')}
>>> parser.parse('02.11.2018 15:03 SP', tzinfos=tzinfos, dayfirst=True)

datetime.datetime(2018, 11, 2, 15, 3, tzinfo=tzfile('/usr/share/zoneinfo/Europe/Moscow'))
```

Data without timezone info?

- check end of March/October for missing/repeating entries
- if you have some solar data, match sunrise/sunset

- Arrow and other convenient libs

- Arrow and other convenient libs
- PostgreSQL: TIMESTAMP WITH TIME ZONE / TIMESTAMPTZ
  - stored as UTC, converts in SQL
  - `SELECT colname AT TIME ZONE 'Europe/Berlin' FROM …`

- Arrow and other convenient libs
- PostgreSQL: TIMESTAMP WITH TIME ZONE / TIMESTAMPTZ
  - stored as UTC, converts in SQL
  - `SELECT colname AT TIME ZONE 'Europe/Berlin' FROM …`
- AoE (Anywhere on Earth)
  - “2018-11-02 AoE”
  - 2018-11-02 23:59:59 Pacific/Samoa
  - 2018-11-02 10:59:59 UTC
  - 2018-11-02 13:59:59 MSK

Best practices

- don't invent your own time zones

Best practices

- don't invent your own time zones
- don't hard code any rules

Best practices

- don't invent your own time zones
- don't hard code any rules
- keep your time zone libs up-to-date

Best practices

- don't invent your own time zones
- don't hard code any rules
- keep your time zone libs up-to-date
- convert from local time to UTC as soon as possible (be precise or guess)

Best practices

- don't invent your own time zones
- don't hard code any rules
- keep your time zone libs up-to-date
- convert from local time to UTC as soon as possible (be precise or guess)
- convert from UTC to local time as late as possible

Best practices

- don't invent your own time zones
- don't hard code any rules
- keep your time zone libs up-to-date
- convert from local time to UTC as soon as possible (be precise or guess)
- convert from UTC to local time as late as possible
- follow your government's intentions to modify your time zone and inform `tz@iana.org`

Best practices

- don't invent your own time zones
- don't hard code any rules
- keep your time zone libs up-to-date
- convert from local time to UTC as soon as possible (be precise or guess)
- convert from UTC to local time as late as possible
- follow your government's intentions to modify your time zone and inform `tz@iana.org`
- store future local events “every day at 10:00 local time” in original form and convert them to datetimes

Best practices

- don't invent your own time zones
- don't hard code any rules
- keep your time zone libs up-to-date
- convert from local time to UTC as soon as possible (be precise or guess)
- convert from UTC to local time as late as possible
- follow your government's intentions to modify your time zone and inform `tz@iana.org`
- store future local events “every day at 10:00 local time” in original form and convert them to datetimes
- AVOID TIME ZONES IF YOU CAN!

“The enjoyment of one's tools

is an essential ingredient of successful work.”

Donald E. Knuth

Miroslav Šedivý

['mɪrɔslaʊ 'ʃɛɟɪviː]

eumiro eumiro šedivý