ДЕРЕКТЕРДІ ИНТЕЛЛЕКТУАЛДЫ ТАЛДАУ

Дәріс 10

Python және pandas ішіндегі күндер мен уақыт

Оспан Ә.Ғ., PhD, доцент-зерттеуші

ҚазҰУ, Жасанды интеллект және Big Data кафедрасы

2026-2027 оқу жылы

Дәрістің мақсаты мен мазмұны

Мақсат және оқу нәтижелері

- Күннің жазылуын, уақыт мезетін және ұзақтықты ажырату.
- Айқын форматтарды, аралықтар мен уақыт белдеулерін қолдану.
- Оқиғаларды тану мен күндік агрегаттауды тексеру.

Дәріс мазмұны

1. **Типтер мен форматтар**
2. **Күнтізбелік арифметика**
3. **Күн терезелері**
4. **UTC және Unix уақыты**
5. **pandas уақыт бағандары**

Мәтіндік жол, мезет және ұзақтық — әртүрлі нысандар

Нысан	Мағына мысалы	Не істейміз
str	Күннің мәтіндік жазылуы	Белгілі форматпен танимыз
datetime / Timestamp	Оқиға күні мен уақыты	Салыстырып, құрамдастарын аламыз
date	Күнтізбелік күн	Күндермен жұмыс істейміз
timedelta	Мезеттер арасындағы ұзақтық	Толық секунд санын есептейміз
Unix timestamp	Эпохадан бергі бірліктер саны	Бірлік пен UTC көрсетеміз

Түрлендіруге дейін өріс мағынасын, өлшем бірліктерін және уақыт белдеуін анықтаңыз.

Күн үлгісі: регистр мағынаны өзгертеді

Белгілеу	Мағынасы	Жиі шатастыру
%Y	Төрт таңбалы жыл	Қысқа %y емес
%m / %M	Ай / минут	Регистр шешуші
%d	Айдың күні	Ай нөмірі емес
%H	Сағат 00-23	24 сағаттық жазу
%I + %p	Сағат 01-12 + AM/PM	Тәулік бөлігі қажет
%S	Секунд	Толық ұзақтық емес

Күнді тану және есеп мәтінін дайындау

Python мысалы

```
from datetime import datetime, timedelta, timezone
import pandas as pd

text = '2019-07-07T18:59:33'
moment = datetime.strptime(text, '%Y-%m-%dT%H:%M:%S')
display_text = moment.strftime('%d.%m.%Y')
print(display_text)
print(moment.hour, moment.minute, moment.second)
```

Нәтиже: 07.07.2019; 18 59 33

Тану

- strptime: str -> datetime.
- Үлгі құрамдастар мағынасын айқын береді.

Пішімдеу

- strftime: datetime -> str.
- Қысқа нәтиже бастапқы moment мәнін өзгертпейді.

Екіұшты формат және мәтіндік жолдарды сұрыптау

Екіұштылық

- 01/02/2024 келісімді талап етеді.
- Автоматты болжау қате болуы мүмкін.
- Он екінші күннен кейінгі мысалдарды тексереміз.

Рет

- Мәтіндік рет әрдайым күнтізбелік емес.
- Сұрыптау мен салыстыруға дейін күндерді танимыз.
- Айқын формат ережені тексеруге мүмкіндік береді.

Шындыққа ұқсас күн бастапқы мәтін форматын қате түсіндіруден алынуы мүмкін.

Ұзақтық: толық мезеттердің айырмасы

duration_seconds = (end - start).total_seconds()

Толық айырма

- Минут, күн, ай және жыл ауысуын ескереді.
- Мәтіндік жолдар емес, уақыт нысандары бойынша есептеледі.

Тексеру

- seconds бүкіл аралықты емес, қалдықты сақтайды.
- Теріс нәтиже бөлек талдауды талап етеді.

timedelta(days=2, seconds=7): seconds=7; total_seconds()=172807.

Себебін анықтамай, теріс ұзақтықты модулімен алмастырмаңыз.

Түн ортасынан және 29 ақпанға өту

Python мысалы

```
start = datetime(2024, 2, 28, 23, 59, 50)
end = start + timedelta(seconds=20)
print(end.isoformat())
print((end - start).total_seconds())
long_gap = timedelta(days=2, seconds=7)
print(long_gap.seconds, long_gap.total_seconds())
```

Нәтиже: 2024-02-29T00:00:10; 20.0; 7 172807.0

Күнтізбе

- 28 ақпаннан кейін 29 ақпан келеді.
- Ай ауысуын кітапхана өңдейді.

Аралық

- Айырма жиырма секундқа тең.
- Толық секундтар күндерді де ескереді.

Күн терезесі: шектердің енуі алдын ала беріледі

1

Есеп белдеуін нақтылау

Күнтізбелік күннің қай белдеуде басталатынын анықтау.

2

Басталуы мен соңын беру

Мысалы, [1 наурыз 00:00, 2 наурыз 00:00).

3

Тану және салыстыру

Әртекті мәтіндік жолдарды емес, уақыт нысандарын салыстыру.

4

Шектерді тексеру

Дәл соңындағы оқиға келесі терезеге кіреді.

- Көршілес [start, end) терезелері қиылыспайды.
- Үш күнтізбелік күн мен 72 сағат — есептің әртүрлі тұжырымдары.

Белдеусіз және белдеуі белгілі уақыт

Белдеусіз datetime

- Белдеу жоқ.
- Дереккөз келісімі қажет.
- tzinfo тағайындау түсіндірмені береді.

Белдеуі белгілі datetime

- Мезет UTC-пен байланысты.
- astimezone мезетті сақтап, сағатты өзгертеді.
- Аударғанда есеп күні өзгеруі мүмкін.

Бір мезеттің әртүрлі белдеулер мен есеп терезелерінде күнтізбелік күні әртүрлі болуы мүмкін.

Белдеуді аудару және Unix уақытының қайтымдылығы

Python мысалы

```
utc_moment = datetime(2024, 3, 1, tzinfo=timezone.utc)
plus_five = timezone(timedelta(hours=5))
local_moment = utc_moment.astimezone(plus_five)
stamp = utc_moment.timestamp()
restored = datetime.fromtimestamp(stamp, tz=timezone.utc)
print(local_moment.isoformat())
assert restored == utc_moment
```

Нәтиже: 2024-03-01T05:00:00+05:00

Аудару

- UTC 00:00 уақыты +05:00 кезінде 05:00 болады.
- Мезет сол күйінде қалады.

Қайтымдылық

- Unix саны белдеуі белгілі нысаннан жасалады.
- Қалпына келтіруде UTC айқын көрсетіледі.

Unix уақыты және күнтізбелік аралықтар

Есеп	Айқын келісім	Жиі қате
Unix секундтары	unit="s", UTC	Санды миллисекунд деп қабылдау
Unix миллисекундтары	unit="ms", UTC	Санды секунд деп қабылдау
Соңғы 72 сағат	Бекітілген ұзақтық	Күнтізбелік күндермен алмастыру
Келесі ай	Күнтізбелік ығысу	Әрдайым 30 күн қосу
Есеп күні	Шектер есеп белдеуінде	Кездейсоқ белдеуде топтау

pandas уақыт бағаны: тану және бақылау

1

Бастапқы мәтінді сақтау

NaT себебін түсіндіру мен диагностикаға қажет.

2

Формат пен UTC беру

Параметрлер дереккөз келісіміне сәйкес болуы тиіс.

3

Қателерді санау

Агрегаттауға дейін дұрыс жолдар мен карантинді бөлу.

4

Реттеу және агрегаттау

Уақыт индексін және берілген күнтізбелік терезелерді қолдану.

- dt.hour және dt.date таңдалған белдеуге тәуелді.
- Енгізілген оқиғалар саны күндік count қосындысына тең болуы тиіс.

Оқу оқиғалары: NaT және күндік қосындылар

Python мысалы

```
events = pd.DataFrame({
    'time_text': ['2024-02-28T23:00:00Z',
                  '2024-02-29T01:00:00Z',
                  '2024-02-30T12:00:00Z'],
    'amount': [10, 20, 30]
})
events['time'] = pd.to_datetime(
    events['time_text'], format='%Y-%m-%dT%H:%M:%SZ',
    errors='coerce', utc=True)
valid = events.loc[events['time'].notna()].copy()
daily = valid.set_index('time')['amount'].resample('D').sum(
    min_count=1)
print(events['time'].isna().sum())
print(daily)
```

Нәтиже: NaT саны: 1; UTC күндік қосындылары: 10 және 20.

Бақылау

- Бір NaT: жоқ 30 ақпан күні.
- Екі дұрыс оқу жазбасы сақталған.

Күндік есеп

- 28 ақпан: 10; 29 ақпан: 20.
- Бос терезені автоматты түрде нөл демейміз.

Уақыт қателері және тексеру жағдайлары

Қате	Салдары	Бақылау жағдайы
%M орнына %m	Минут орнына ай	Белгілі құрамдастар
total_seconds орнына seconds	Толық күндер жоғалады	Бір тәуліктен ұзақ аралық
Аударудың орнына белдеу тағайындау	Басқа мезет	UTC-ке кері аудару
s пен ms шатастыру	Басқа күн	Белгілі уақыт белгісі мен ауқым
Екі шекті де қосу	Екі рет есептеу	Дәл түн ортасындағы оқиға
Санауға дейін NaT жоғалту	Теңгерімнің бұзылуы	Кіріс = дұрыс оқиғалар + карантин

Дәрістің негізгі нәтижелері

1. Мәтіндік жол, уақыт мезеті және ұзақтықтың мағынасы әртүрлі.
2. Айқын формат күнді екіұшты болжаудың алдын алады.
3. total_seconds күндерді қоса толық ұзақтықты ескереді.
4. Белдеуді аудару мезетті сақтайды, бірақ есеп күнін өзгертуі мүмкін.
5. NaT және терезе шектері күндік агрегаттауға дейін тексеріледі.

Бақылау сұрақтары және өзіндік жұмыс

1. strptime пен strftime қалай ерекшеленеді?
2. %H пен %I неге бір таңғы мысал бойынша таңдалмайды?
3. timedelta.seconds қолданғанда не жоғалады?
4. Белдеу тағайындау мезетті аударудан қалай ерекшеленеді?
5. [start, end) көршілес терезелерге неге ыңғайлы?

Өзіндік жұмыс

UTC түн ортасының екі жағында екі оқиға жасаңыз. Бекітілген +05:00 белдеуіне аударып, күндік жиіліктерді салыстырыңыз және мезеттердің сақталуын дәлелдеңіз.

Дереккөздер және құжаттама

Құжаттама

1. Python: datetime

https://docs.python.org/3/library/datetime.html

2. pandas: уақыт қатарлары

https://pandas.pydata.org/docs/user_guide/timeseries.html

3. pandas: уақыт аралықтары

https://pandas.pydata.org/docs/user_guide/timedeltas.html

Дәрістің негізі

- Берілген 10-дәрістің презентациясы
- Берілген 10-зертханалық жұмыс