**فصل 7. مدیریت تاریخ‌ها و زمان‌ها**

**مقدمه**

در یادگیری ماشین اغلب در مرحله‌ی پیش‌پردازش داده‌ها با تاریخ‌ها و زمان‌ها مواجه می‌شویم. ممکن است زمان فروش بخصوصی باشد یا سال تعدادی آمار مربوط به سلامت عمومی. در این فصل، جعبه ابزاری برای مدیریت داده‌های زمانی خواهیم ساخت که شامل دستکاری کردن منطقه زمانی و ایجاد مختصه‌های زمانی وقفه‌دار می‌شود. به ویژه، بر ابزارهای سری زمانی در کتابخانه‌ی Pandas تمرکز خواهیم کرد که قابلیت‌های بسیاری از کتابخانه‌های دیگر را دارد.

**7-1- تبدیل رشته‌ها[[1]](#footnote-1) به تاریخ‌ها**

**مسئله**

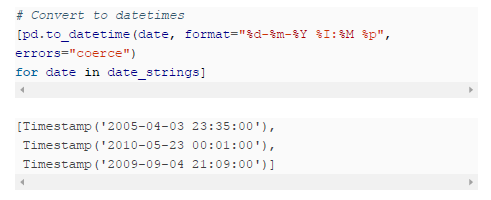
فرض کنید یک بردار از رشته‌های تاریخ‌ و زمان‌ در اختیار دارید و می‌خواهید آن‌ها را به داده‌‌ی سری‌های زمانی تبدیل کنید.

**راه حل**

از دستور to\_datetime در Pandas با قالب تاریخ و/یا زمان که در پارامتر format مشخص شده است استفاده کنید:



شاید بخواهیم شناسه[[2]](#footnote-2)‌ای به پارامتر errors اضافه کنیم تا مشکلات را مدیریت کند:



اگر errors="coerce"، پس هر مشکلی ایجاد شود خطا نمی‌دهد (رفتار پیش‌فرض) بلکه به جای آن، مقداری که باعث ایجاد خطا شده را برابر با NaT (یعنی مقدار مفقود شده) می‌گذارد.

بحث

وقتی تاریخ‌ها و زمان‌ها به صورت رشته در می‌آیند، باید آن‌ها را به نوعی از داده تبدیل کنیم که برای پایتون قابل فهم باشد. با وجود آنکه پایتون تعدادی ابزار برای تبدیل رشته‌ها به تاریخ‌زمان‌ها دارد، اما با توجه به کابرد Pandas در دستورالعمل‌های دیگر، می‌توانیم از to\_datetime برای تبدیل استفاده کنیم. رشته‌هایی که تاریخ‌ها و زمان‌ها را نشان می‌دهند مشکلی دارند و آن این است که فرمت رشته‌ها ممکن است در منابع داده تفاوت معناداری داشته باشد. برای مثال، ممکن است یک بردار تاریخ‌ها 23ام مارس 2015 را به صورت “03-23-15” نشان دهد اما دیگری ممکن است از “3|23|2015” استفاده کند. می‌توانیم از پارامتر format برای مشخص کردن فرمت دقیق رشته استفاده کنیم. اینجا برخی کدهای فرمت تاریخ و زمان رایج را مشاهده می‌کنید:

|  |  |  |
| --- | --- | --- |
| کد | توضیح | مثال |
| %Y | کل سال | 2001 |
| %m | ماه پوشش صفر[[3]](#footnote-3) w/ | 04 |
| %d | روزِ ماه پوشش صفر w/ | 09 |
| %I | ساعت (12 ساعته) پوشش صفر w/ | 02 |
| %p | AM یا PM | AM |
| %M | دقیقه پوشش صفر w/ | 05 |
| %S | ثانیه پوشش صفر w/ | 09 |

برای اطلاعات بیشتر:

* [فهرست کاملی از کدهای رشته زمانی پایتون](https://strftime.org/)

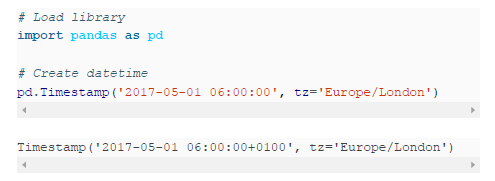
**7-2- مدیریت منطقه‌های زمانی**

**مسئله**

داده‌ی سری‌های زمانی را در اختیار دارید و می‌خواهید اطلاعات منطقه زمانی را تغییر دهید یا اطلاعاتی به آن اضافه کنید.

**راه حل**

اگر منطقه زمانی مشخص نباشد، اشیای Pandas منطقه زمانی ندارند. با این حال می‌توانیم با استفاده از tz یک منطقه زمانی‌ را بسازیم و اضافه کنیم:



می‌توانیم با استفاده از tz\_localize یک منطقه زمانی‌ را به تاریخ‌زمانی که قبلاً ساخته شده اضافه کنیم:



همچنین می‌توانیم منطقه زمانی‌اش را تغییر دهیم:

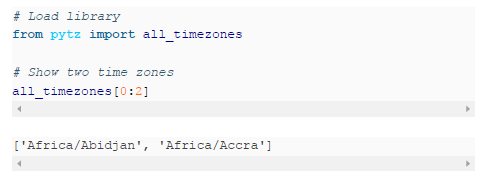


سرانجام، اشیای سری‌های Pandas می‌توانند tz\_localize و tz\_convert را به هر اِلِمان اعمال کنند:



**بحث**

Pandas از دو مجموعه رشته برای نشان دادن منطقه‌های زمانی پشتیبانی می‌کند. با این حال پیشنهاد می‌کنم از رشته‌های کتابخانه‌ی pytz استفاده کنید. می‌توانیم با وارد کردن all\_timezones همه‌ی رشته‌هایی را که برای نشان دادن منطقه‌های زمانی به کار می‌روند، ببینیم:



**7-3- انتخاب تاریخ‌ها و زمان‌ها**

**مسئله**

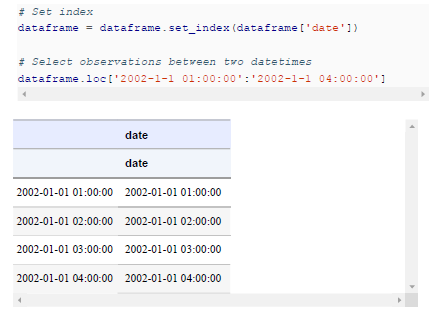
یک بردار تاریخ‌ دارید و می‌خواهید یک یا تعداد بیشتری را انتخاب کنید.

**راه‌حل**

از دو شرط منطقی[[4]](#footnote-4) برای شروع و پایان تاریخ‌ها استفاده کنید:



یا به جای آن می‌توانیم ستون تاریخ را به عنوان شاخص دیتا‌فریم در نظر بگیریم و سپس با استفاده از loc تقسیم کنیم:



**بحث**

برای استفاده از شروط منطقی یا تقسیم‌ شاخص باید به موقعیت توجه کنیم. اگر می‌خواهیم کار پیچیده‌ای روی سری‌های زمانی انجام دهیم، شاید بهتر است بالای ستون تاریخ را به عنوان شاخص دیتافریم (DataFrame) در نظر بگیریم. اما اگر می‌خواهیم داده‌ورزی[[5]](#footnote-5) ساده‌ای انجام دهیم، شروط منطقی شاید آسان‌تر باشد.

**4-7- شکستن داده‌ی تاریخ به چندین مختصه[[6]](#footnote-6)**

**مشکل**

ستونی از تاریخ‌ها و زمان‌ها در اختیار شماست و می‌خواهید مختصه‌هایی برای سال، ماه، روز، ساعت و دقیقه بسازید.

**راه حل**

از ویژگی‌های زمانی Series.dt در Pandas استفاده کنید:



**بحث**

گاهی شکستن یک ستون از تاریخ به اجزایش می‌تواند مفید باشد.

به عنوان مثال، ممکن است مختصه‌ای را بخواهیم که فقط شامل سال مشاهده یا نمونه باشد یا ممکن است بخواهیم فقط ماه برخی از مشاهدات را در نظر بگیریم تا بتوانیم آن‌ها را فارغ از سال با هم مقایسه کنیم.

**5-7- محاسبه‌ی اختلاف بین تاریخ‌ها**

**مسئله**

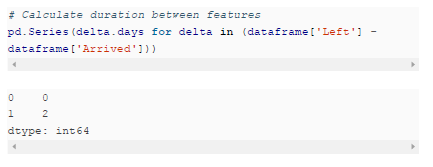
دو مختصه‌ی زمان‌تاریخ دارید و می‌خواهید زمان بین دو زمان‌تاریخ را برای هر مشاهده محاسبه کنید.

**راه‌حل**

با استفاده از Pandas دو مختصه‌ی تاریخ را تفریق کنید:



اغلب می‌خواهیم خروجی days را حذف کنیم و فقط مقدار عددی را نگه داریم:



**بحث**

در مواقعی خواستار مختصه‌ی تغییر (دلتا) بین دو نقطه در زمان هستیم. به عنوان مثال شاید تاریخ ورود و خروج یک مشتری به هتل را در اختیار داشته باشیم اما می‌خواهیم مدت زمان اقامتش را بدانیم. به کمک TimeDelta در Pandas می‌توان به راحتی محاسبه‌اش کرد.

**برای اطلاعات بیشتر:**

* [اسناد پاندا: Time Deltas](http://bit.ly/2HOS3sJ)

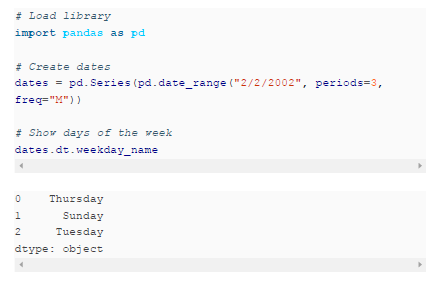
**7-6- کدگذاری کردن روزهای هفته**

**مسئله**

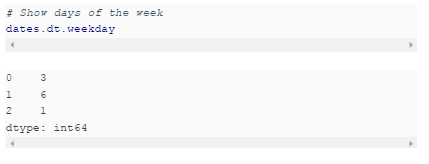
یک بردار تاریخ دارید و می‌خواهید بدانید هر تاریخ چندشنبه است.

**راه‌حل**

از Series.dt در Pandas و ویژگی weekday\_name استفاده کنید:



اگر بخواهیم خروجی مقداری عددی باشد تا به عنوان مختصه‌ی یادگیری ماشین بیشتر به کار بیاید، می‌توانیم از weekday استفاده کنیم که روزهای هفته را به صورت یک عدد صحیح نمایش می‌دهد (دوشنبه برابر با 0 است):



**بحث**

دانستن روز هفته می‌تواند مفید باشد. به عنوان مثال اگر بخواهیم کل فروش یک‌شنبه‌های سه سال اخیر را با هم مقایسه کنیم. به کمک Pandas به راحتی می‌توان برداری ساخت که شامل اطلاعات روزهای هفته باشد.

**برای اطلاعات بیشتر:**

* [pandas Series datetimelike properties](http://bit.ly/2HShhq1)

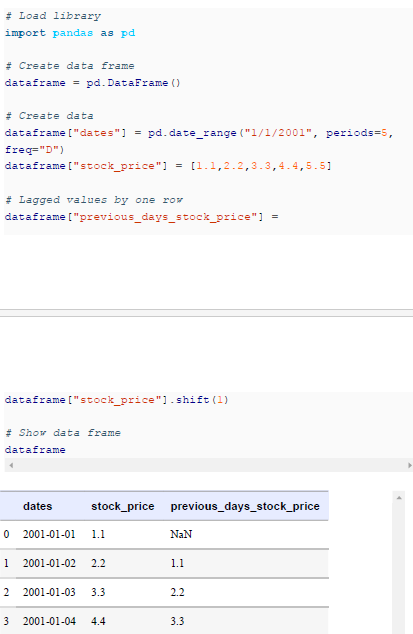
**7-7 ساختن یک مختصه‌ی وقفه‌دار[[7]](#footnote-7)**

**مسئله**

می‌خواهید مختصه‌ای بسازید که n دوره‌ی زمانی وقفه دارد.

**راه حل**

از shift در Pandas استفاده کنید:



**بحث**

بسیاری از اوقات داده‌ها بر اساس دوره‌‌های زمانی منظمی هستند ( مثلاً، هر روز، هر ساعت، هر سه ساعت) و ما می‌خواهیم از داده‌های گذشته برای پیش‌بینی استفاده کنیم (اغلب به آن مختصه‌ی وقفه‌دار می‌گویند). برای مثال، می‌خواهیم قیمت سهام را با استفاده از قیمت روز قبلش پیش‌بینی کنیم. به کمک Pandas می‌توانیم از shift برای وقفه‌انداختن مقادیر در هر ردیف استفاده کنیم و مختصه‌ی جدیدی بسازیم که حاوی مقادیر گذشته باشد.

در این راه حل، اولین ردیف previous\_days\_stock\_price مقداری ندارد زیرا مقدار stock\_price قبلی وجود ندارد.

**7-8- استفاده از پنجره‌های رولینگ تایم یا زمان‌غلتان[[8]](#footnote-8)**

**مسئله**

داده‌های سری زمانی در اختیار شماست و می‌خواهید از رولینگ تایم استفاده کنید:

**راه‌حل**



**بحث**

پنجره‌های رولینگ تایم (به آن پنجره‌ی زمانیِ متحرک هم می‌گویند) از نظر مفهومی ساده هستند اما ممکن است در نگاه اول به آسانی فهمیده نشوند. تصور کنید داده‌ای داریم که حاوی مشاهده‌ی ماهانه‌ی قیمت یک سهام است. داشتن یک پنجره‌ی زمانی برای تعداد مشخصی از ماه‌ها و بعد حرکت‌دادن روی داده‌های مشاهده و انجام محاسبات آماری روی تمام مشاهدات اغلب می‌تواند مفید باشد.

برای مثال اگر یک پنجره‌ی زمانی برای سه ماه داشته باشیم و یک میانگین رولینگ بخواهیم، به صورت زیر محاسبه می‌کنیم:

1. mean(January, February, March)

2. mean(February, March, April)

3. mean(March, April, May)

4. etc.

راه دیگرش این است: پنجره‌ی زمانی سه ‌ماه‌ بر روی داده‌های مشاهده حرکت کند (راه برود) و میانگین را در هر قدم محاسبه کند.

Rolling در Pandas این امکان را برای ما فراهم می‌کند که اندازه‌ی پنجره را با استفاده از window مشخص کنیم و بعد به سرعت تعدادی محاسبات آماری انجام دهیم که شامل مقدار ماکسیسمم (max())، مقدار میانگین (mean())، شمارش مقادیر (count()) و همبستگی رولینگ (corr()) است.

از میانگین‌های رولینگ اغلب برای هموارکردن داده‌های سری زمانی استفاده می‌کنند زیرا وقتی از میانگین سرتاسر پنجره‌ی زمانی استفاده می‌کنیم، اثر نوسانات کوتاه‌مدت کاهش پیدا می‌کند.

**برای اطلاعات بیشتر:**

* [اسناد پاندا: پنجره‌های رولینگ](http://bit.ly/2HTIicA)
* [میانگین متحرک[[9]](#footnote-9) یا تکنیک‌های هموار کردن چیست؟](http://bit.ly/2HRRHBn)

**7-9- مدیریت داده‌های ناموجود در سری‌های زمانی**

**مسئله**

در داده‌های سری‌های زمانی مقادیری موجود نیستند.

**راه‌حل**

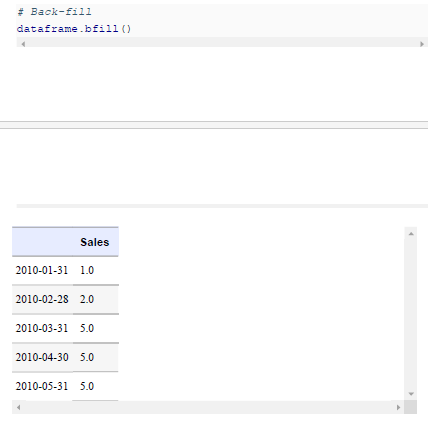
علاوه بر استراتژی‌هایی که پیش‌تر درباره‌ی داده‌های ناموجود بیان کردیم، وقتی با داده‌های سری زمانی مواجه هستیم می‌توانیم از همپوشانی برای پر کردن جای خالی آن‌ها استفاده کنیم:



یا می‌توانیم به جای مقادیری که موجود نیستند آخرین مقدار موجود قبل از جای خالی را قرار دهیم (forward-filling):



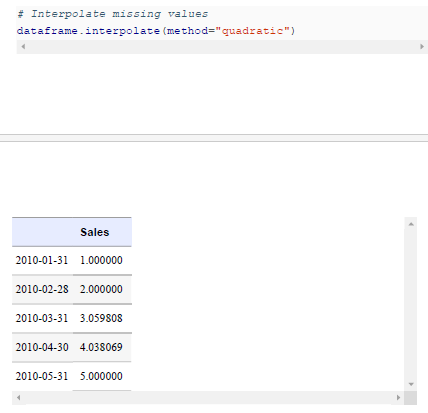
همچنین می‌توانیم مقادیر ناموجود را با اولین مقدار موجود بعد از جای خالی جایگزین کنیم (back-filling):



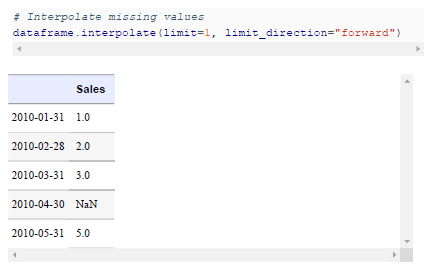
**بحث**

همپوشانی تکنیکی است که از آن برای پر کردن جای خالی مقادیر ناموجود استفاده می‌کنند. در این تکنیک یک خط یا منحنی بین مقادیر موجود می‌کشند که جای خالی را پر می‌کند و با استفاده از آن خط یا منحنی مقادیر مناسبی برای آن داده‌های ناموجود می‌یابند. همپوشانی به ویژه زمانی مفید است که فواصل زمانی ثابت هستند و داده مستعد افت‌وخیزهای نوفه‌ای نیست و مقادیر ناموجود کوچک هستند. برای مثال، برای به دست آوردن دو مقدار ناموجود بین 2 و 5، خطی بین این مقدار می‌کشیم که از 2 شروع و به 5 ختم می‌شود. حدس معقولانه این است که دو مقدار ناموجود بین 3 و 4 قرار دارند.

اگر مطمئن باشیم که خطِ بین دو مقدار موجود غیرخطی است، می‌توانیم از interpolate method استفاده کنیم تا روش همپوشانی را مشخص کنیم:



در آخر، شاید در مواقعی تعداد مقادیر ناموجود زیاد باشند و نخواهیم همه‌ی مقادیر ناموجود را همپوشانی کنیم. در این موارد می‌توانیم از limit برای محدود کردن تعداد مقادیر همپوشانی‌شده استفاده کنیم و به کمک limit\_direction مشخص کنیم که مقادیر ناموجود را از کدام طریق پیش‌بینی کند. از طریق آخرین مقدار موجود قبل از فاصله یا برعکس:



ممکن است این طور گمان شود که Back-filling یا forward-filling شکل ساده‌لوحانه‌ای از همپوشانی هستند؛ یک خط صاف از مقدار موجود بکشیم تا جای مقادیر ناموجود را پر کنیم. یک مزیت کوچک این دو روش به همپوشانی این است که نیازی نیست مقادیرموجود هر دو سر مقادیر ناموجود را داشته باشیم.

1. String [↑](#footnote-ref-1)
2. argument [↑](#footnote-ref-2)
3. zero padding [↑](#footnote-ref-3)
4. boolean conditions [↑](#footnote-ref-4)
5. data wrangling [↑](#footnote-ref-5)
6. feature [↑](#footnote-ref-6)
7. Lagged Feature [↑](#footnote-ref-7)
8. Rolling Time [↑](#footnote-ref-8)
9. Moving Average [↑](#footnote-ref-9)