PREPARING DATETIME DATA WITH PADR

Edwin Thoen
useR 2017 Brussels
Number of traffic accidents in four neighbourhoods

- **ABINGTON**
- **CHELTENHAM**
- **LOWER MERION**
- **UPPER MERION**

**Dayparts:**
- Night
- Morning
- Afternoon
- Evening

**Dates:**
- Jul 11
- Jul 13
- Jul 15
- Jul 17
```r
head(one_week_of_traffic, 10)
```

<table>
<thead>
<tr>
<th>title</th>
<th>time_stamp</th>
<th>twp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic: VEHICLE ACCIDENT -</td>
<td>2016-07-10 08:17:02</td>
<td>ABINGTON</td>
</tr>
<tr>
<td>Traffic: VEHICLE ACCIDENT -</td>
<td>2016-07-10 10:47:02</td>
<td>LOWER MERION</td>
</tr>
<tr>
<td>Traffic: VEHICLE ACCIDENT -</td>
<td>2016-07-10 11:17:01</td>
<td>UPPER MERION</td>
</tr>
<tr>
<td>Traffic: VEHICLE ACCIDENT -</td>
<td>2016-07-10 11:22:01</td>
<td>LOWER MERION</td>
</tr>
<tr>
<td>Traffic: VEHICLE ACCIDENT -</td>
<td>2016-07-10 11:37:00</td>
<td>LOWER MERION</td>
</tr>
<tr>
<td>Traffic: VEHICLE ACCIDENT -</td>
<td>2016-07-10 11:51:01</td>
<td>UPPER MERION</td>
</tr>
<tr>
<td>Traffic: VEHICLE ACCIDENT -</td>
<td>2016-07-10 13:12:01</td>
<td>UPPER MERION</td>
</tr>
<tr>
<td>Traffic: VEHICLE ACCIDENT -</td>
<td>2016-07-10 13:42:03</td>
<td>ABINGTON</td>
</tr>
<tr>
<td>Traffic: VEHICLE ACCIDENT -</td>
<td>2016-07-10 14:06:02</td>
<td>ABINGTON</td>
</tr>
<tr>
<td>Traffic: VEHICLE ACCIDENT -</td>
<td>2016-07-10 14:56:00</td>
<td>ABINGTON</td>
</tr>
</tbody>
</table>
TWO PROBLEMS WHEN PREPARING DATETIME DATA

• Measurements are on too low a time level

• When there is no event, there is no record
THE INTERVAL

• Think of observations as having a heartbeat

• Highest repetitive pattern that explains all observations

• Does not need an observation at each heartbeat
```{r}
get_interval(one_week_of_traffic$time_stamp)
```

```
[1] "sec"
```

```
```{r}
get_interval(as.Date(c("2017-07-01", "2017-07-03", "2017-07-07")))
```

```
[1] "2 day"
```

```
```{r}
get_interval(as.Date(c("2017-07-01", "2017-07-03", "2017-07-06")))
```

```
[1] "day"
```
• **finds** the datetime variable in a df

• **spans** a variable of a **higher** interval around it

• **assigns** each original value to a spanned value

• **adds** the spanned value as a column to the df
```r
one_week_of_traffic %>%
  thicken("6 hour") %>%
  count(twp, time_stamp_6_hour)
```
PAD

• **finds** the datetime variable in a df

• **spans** a variable of the same interval

• **fills** the missing records according to the interval

• **returns** NA values for all other columns
```r
one_week_of_traffic %>%
  thicken("6 hour") %>%
  count(twp, time_stamp_6_hour) %>%
  pad(group = "twp")
```
FILL_BY_SUITE

- How to fill NAs after padding depends on the nature of the data
- Forward-filling can be done with `tidyr::fill()`
- Value-filling with `fill_by_value()`
- Other options are `fill_by_function()` and `fill_by_prevalent()`
```r
one_week_of_traffic %>%
  thicken("6 hour") %>%
  count(twp, time_stamp_6_hour) %>%
  pad(group = "twp") %>%
  fill_by_value()
```

<table>
<thead>
<tr>
<th>twp</th>
<th>time_stamp_6_hour</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABINGTON</td>
<td>2016-07-10 06:00:00</td>
<td>1</td>
</tr>
<tr>
<td>ABINGTON</td>
<td>2016-07-10 12:00:00</td>
<td>6</td>
</tr>
<tr>
<td>ABINGTON</td>
<td>2016-07-10 18:00:00</td>
<td>1</td>
</tr>
<tr>
<td>ABINGTON</td>
<td>2016-07-11 00:00:00</td>
<td>0</td>
</tr>
<tr>
<td>ABINGTON</td>
<td>2016-07-11 06:00:00</td>
<td>1</td>
</tr>
<tr>
<td>ABINGTON</td>
<td>2016-07-11 12:00:00</td>
<td>1</td>
</tr>
<tr>
<td>ABINGTON</td>
<td>2016-07-11 18:00:00</td>
<td>3</td>
</tr>
<tr>
<td>ABINGTON</td>
<td>2016-07-12 00:00:00</td>
<td>0</td>
</tr>
<tr>
<td>ABINGTON</td>
<td>2016-07-12 06:00:00</td>
<td>1</td>
</tr>
<tr>
<td>ABINGTON</td>
<td>2016-07-12 12:00:00</td>
<td>1</td>
</tr>
</tbody>
</table>
4 STEPS OF PADR

• **add** a column of a higher interval with **thicken**

• **aggregate** your data grouped by the added column with dplyr, data.table or base R

• **insert** rows for missing observations with **pad**

• **replace** the missing values after padding with **tidyr::fill** or one of the **fill_by_** functions
MORE INFORMATION

• There are two vignettes: padr and padr_implementation

• I blog at: thats-so-random.com

• @edwin_thoen

• https://github/EdwinTh/padr