Sliding Windows and Calendars

Rolling Windows
Slide variants apply a function along sequential windows of a vector. Useful for moving averages!

Expanding Windows
Set .before = Inf to use an expanding window. Useful for cumulative functions!

Time-Aware Sliding
Respect the “gaps” in your time series by supplying a secondary index.

Period-Blocked Sliding
Slide in rolling period chunks to summarize at different frequencies.

Familiar Syntax
If you’ve used purrr, you’ll feel at home with slide!

Recurrence Rules
Construct holiday / weekend events with recurrence rules.

Schedules
Combine individual rules into comprehensive schedules.

lubridate Extensions
Use them to step by irregular periods!

Animations use .before = 2.
Sliding Windows and Calendars

{slider} - rstd.io/slider

Recurrence Rules

Construct holiday / weekend events with recurrence rules.

{almanac} - rstd.io/almanac

Rolling Windows

Slide variants apply a function along sequential windows of a vector. Useful for moving averages!

```r
slide(    , fn, .before = 1)
```

Expanding Windows

Set `.before = Inf` to use an expanding window. Useful for cumulative functions!

```r
slide(    , fn, .before = Inf)
```

Familiar Syntax

If you've used purrr, you'll feel at home with slide!

Variants such as:

- `slide2(.x, .y)`
- `pslide(.l)`

Multiple inputs with:

- `slide2(.x, .y)`

If you've used purrr, you'll feel at home with slide!

Expanding Windows

Set `.before = Inf` to use an expanding window. Useful for cumulative functions!

```r
slide(    , fn, .before = Inf)
```

Schedules

Combine individual rules into comprehensive schedules.

```r
on_weekends <- weekly() %>%
  recur_on_weekends()
```

```r
on_labor_day <- yearly() %>%
  recur_on_ymonth("September") %>%
  recur_on_wday("Monday", nth = 1)
```

```r
on_weekdays <- weekly() %>%
  recur_on_weekdays()
```

```r
sch_business <- schedule() %>%
  sch_rrule(on_labor_day) %>%
  sch_rrule(on_weekends) %>%
  sch_merge(hldy_christmas())
```

Prebuilt holidays

```r
alma_in <- c("2019-09-02", "2019-09-03")
```

```r
alma_stop(friday, n = 1, sch_business)
```

```r
alma <- as.Date("2019-09-03")
```

```r
on_labor_day <- yearly() %>%
  recur_on_ymonth("September") %>%
  recur_on_wday("Monday", nth = 1)
```

```r
friday <- as.Date("2019-08-30")
```

```r
alma_step(friday, n = 1, sch_business)
```

```r
alma_step(friday, n = 1, sch_business)
```

```r
alma_step(friday, n = 1, sch_business)
```

Correctly constructs business ranges of [Thur, Fri], [Fri, Mon], etc.
Rolling & Expanding Windows

Slide variants apply a function along sequential windows of a vector.
Useful for moving averages!

```
slide(x, fn, .before = 1)
```

- **Expanding**
  - `.before = Inf`
- **Rolling**
  - `.before = 2`

```
fn(    )
fn(    )
fn(    )
fn(    )
```

```
1  2  3  4  5  6  7  8  9  10
```
Familiar Syntax

If you’ve used purrr, you’ll feel at home with slide!

Variants such as:
- `slide dbl()`, `slide dfr()`

Multiple inputs with:
- `slide2(.x, .y)`, `pslide(.l)`
Time-Aware Sliding

Respect the “gaps” in your time series by supplying a secondary index.

```
slide_index( 1, 2 4 5, fn, .before = days(1))
```

Looking back 1 day creates a range of [3, 4]. Since day 2 isn’t in the range, it isn’t included.

```
slide() a
slide_index() a
```
Recurrence Rules

Construct holiday / weekend events with recurrence rules.

Start with a base frequency

```r
on_labor_day <- yearly() %>%
  recur_on_ymonth("September") %>%
  recur_on_wday("Monday", nth = 1)
```

Add recurrence conditions

Is a date in the recurrence set?

```r
alma_in(
  c("2019-09-02", "2019-09-03"),
  on_labor_day
)
#> [1] TRUE FALSE
```
lubridate Extensions

on_weekends <- weekly() %>%
  recur_on_weekends()

one_day <- days(1)

one_bday <- bdays(1, on_weekends)

Construct business day period objects.

Use them to step by irregular periods!

Fri + one_day = Sat
Fri + one_bday = Mon
\{slider\} + \{almanac\} = \heartsuit

Slide according to custom business schedule rules.

slide_index(Thur Fri Mon Tues), fn, .before = one_day)

slide_index(Thur Fri Mon Tues), fn, .before = one_bday)
{slider} - rstd.io/slider

{almanac} - rstd.io/almanac

slides - github.com/DavisVaughan/rstudio-conf-2020

@dvaughan32