

Don't walk, run! `runner` package for rolling window functions.

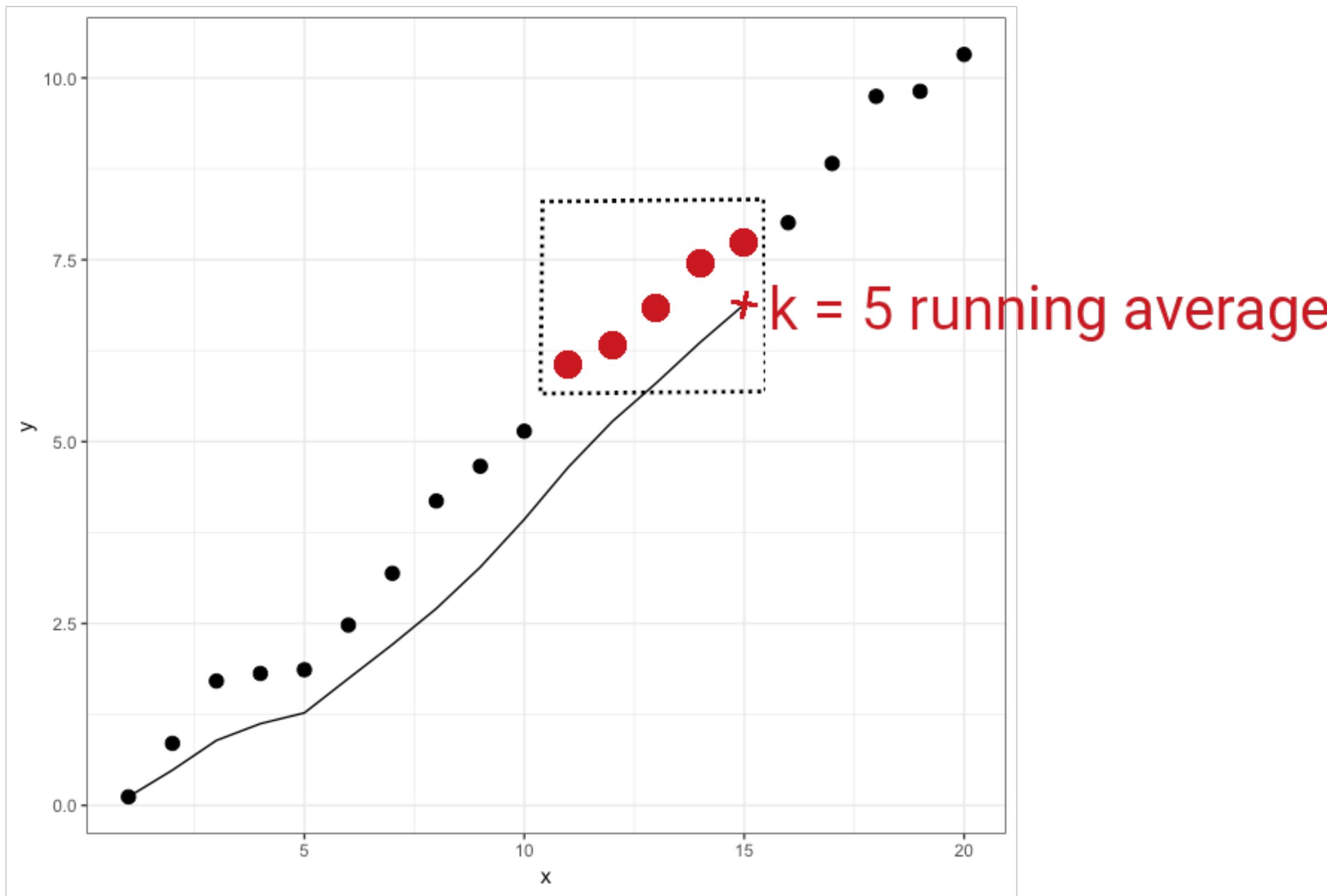
Dawid Kałędkowski @Roche/NEST

whoami

- Member of Roche/NEST team.
- I collect and predict speedway results.
- Speaches at WhyR, Data Mass, PAZUR, LondonR.
- Two packages `sport` and `runner` available on CRAN.



Running (rolling, moving) functions



R packages with running functions

``data.table``

``RcppRoll``

``runner``

``tibbletime``

``zoo``

and other minor implementations



Simple functions, complicated syntax

```
RcppRoll::roll_meanr(x,  
                      n = 4,  
                      fill = NA,  
                      na.rm = TRUE)
```

```
zoo::rollmeanr(x,  
                k = 4,  
                na.pad = TRUE,  
                na.rm = TRUE)
```



Simple functions, complicated syntax

```
RcppRoll::roll_meanr(x,  
                      n = 4,  
                      fill = NA,  
                      na.rm = TRUE)
```

```
zoo::rollmeanr(x,  
                k = 4,  
                na.pad = TRUE,  
                na.rm = TRUE)
```

Have to specify to keep
the same length as 'x'



Simple functions, complicated syntax

```
> library(tibbletime)  
> rolling_m5 <- rollify(~mean(.x, na.rm =TRUE), 5)  
> rolling_m13 <- rollify(~mean(.x, na.rm = TRUE), 13)  
> rolling_m21 <- rollify(~mean(.x, na.rm = TRUE), 21)
```

```
> rolling_m5(x)  
> rolling_m13(x)  
> rolling_m21(x)
```

Need to specify all calls - for each arguments combination



Simple functions, complicated syntax

```
> library(tibbletime)  
> rolling_m5 <- rollify(~mean(.x, na.rm =TRUE), 5)  
> rolling_m13 <- rollify(~mean(.x, na.rm = TRUE), 13)  
> rolling_m21 <- rollify(~mean(.x, na.rm = TRUE), 21)  
  
> rolling_m5(x)  
> rolling_m13(x)  
> rolling_m21(x)
```



Simple functions, complicated syntax

```
library(data.table)
frollmean(x,
          n = 4,
          fill = NA,
          align = "right",
          na.rm = TRUE)
```



Simple functions, complicated syntax

```
library(data.table)
frollmean(x,
          n = 4,
          fill = NA,
          align = "right",      default
          na.rm = TRUE)
```



`runner` doesn't fail on windows < k

```
> c(1, 2, 3, 4, 5) %>% runner::mean_run(k = 5, na_pad = FALSE)
```

```
[1] 1 1.5 2 2.5 3
```

```
> c(1, 2, 3, 4, 5) %>% data.table::frollmean(n = 5)
```

```
[1] NA NA NA NA 3
```

```
> c(1, 2, 3, 4, 5) %>% zoo::roll_mean(k = 5, fill = NA)
```

```
[1] NA NA NA NA 3
```



k is missing - simple cumulative fun

```
> c(1, 2, 3, 4) %>% runner::mean_run()  
[1] 1 1.5 2 2.5
```

```
> c(1, 2, 3, 4) %>% data.table::frollmean()  
error: argument "n" is missing, with no default
```

```
> c(1, 2, 3, 4) %>% zoo::roll_meanr(fill = NA)  
error: argument "k" is missing, with no default
```



runner is dplyr friendly

```
> data %>%  
  group_by(group) %>%  
  mutate(  
    runner_mean = runner::mean_run(close, k = 5),  
    data.table = frollmean(close, n = 5),  
    zoo_roll = zoo::rollmean(close, k = 5, na.pad = TRUE)  
)
```

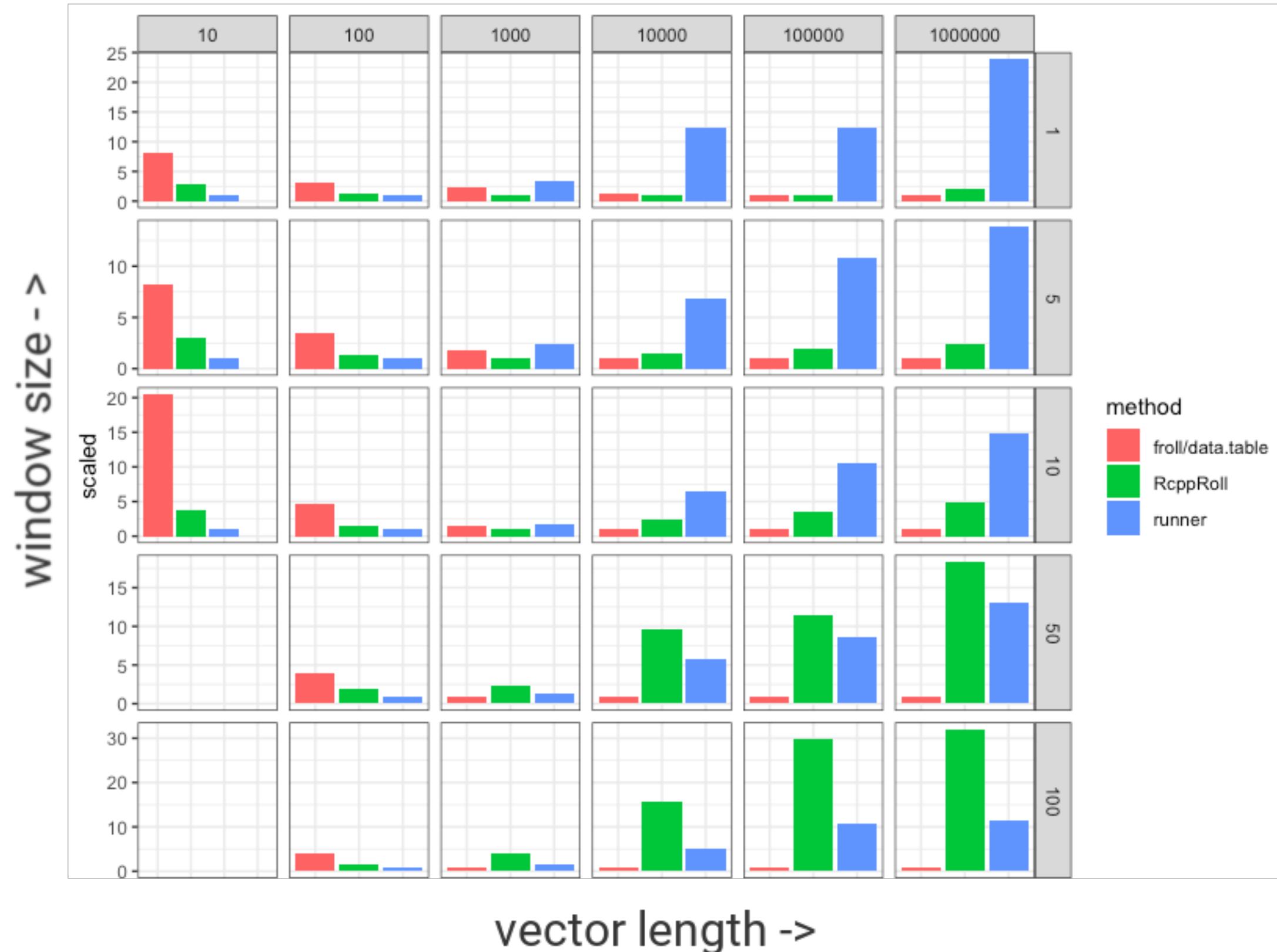


runner is dplyr friendly

```
> data %>%  
  group_by(group) %>%  
  mutate(  
    ✓ runner_mean = runner::mean_run(close),  
    ✗ data.table = frollmean(close, n = 5), Fail on groups < 5  
    ✗ zoo_roll = zoo::rollmean(close, k = 5, na.pad = TRUE)  
  )
```



Performance of the best runners



`runner` goes where other can't

```
> head(FANG, 10)
# A tibble: 10 x 8
  symbol date      open   high   low close volume adjusted
  <chr>  <date>    <dbl>  <dbl>  <dbl> <dbl>  <dbl>    <dbl>
1 FB     2013-01-02 27.4   28.2   27.4  28     69846400 28
2 FB     2013-01-03 27.9   28.5   27.6  27.8  63140600 27.8
3 FB     2013-01-04 28.0   28.9   27.8  28.8  72715400 28.8
4 FB     2013-01-07 28.7   29.8   28.6  29.4  83781800 29.4
5 FB     2013-01-08 29.5   29.6   28.9  29.1  45871300 29.1
6 FB     2013-01-09 29.7   30.6   29.5  30.6  104787700 30.6
7 FB     2013-01-10 30.6   31.5   30.3  31.3  95316400 31.3
8 FB     2013-01-11 31.3   32.0   31.1  31.7  89598000 31.7
9 FB     2013-01-14 32.1   32.2   30.6  31.0  98892800 31.0
10 FB    2013-01-15 30.6   31.7   29.9  30.1  173242600 30.1
```



`runner` goes where other can't



missing weekends



	symbol	date	open	high	low	close	volume	adjusted
	<chr>	<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	FB	2013-01-02	27.4	28.2	27.4	28	69846400	28
2	FB	2013-01-03	27.9	28.5	27.6	27.8	63140600	27.8
3	FB	2013-01-04	28.0	28.9	27.8	28.8	72715400	28.8
4	FB	2013-01-07	28.7	29.8	28.6	29.4	83781800	29.4
5	FB	2013-01-08	29.5	29.6	28.9	29.1	45871300	29.1
6	FB	2013-01-09	29.7	30.6	29.5	30.6	104787700	30.6
7	FB	2013-01-10	30.6	31.5	30.3	31.3	95316400	31.3
8	FB	2013-01-11	31.3	32.0	31.1	31.7	89598000	31.7
9	FB	2013-01-14	32.1	32.2	30.6	31.0	98892800	31.0
10	FB	2013-01-15	30.6	31.7	29.9	30.1	173242600	30.1



`runner` goes where other can't



missing weekends



	symbol	date	open	high	low	close	volume	adjusted
	<chr>	<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	FB	2013-01-02	27.4	28.2	27.4	28	69846400	28
2	FB	2013-01-03	27.9	28.5	27.6	27.8	63140600	27.8
3	FB	2013-01-04	28.0	28.9	27.8	28.8	72715400	28.8
4	FB	2013-01-07	28.7	29.8	28.6	29.4	83781800	29.4
5	FB	2013-01-08	29.5	29.6	28.9	29.1	45871300	29.1
6	FB	2013-01-09	29.7	30.6	29.5	30.6	104787700	30.6
7	FB	2013-01-10	30.6	31.5	30.3	31.3	95316400	31.3
8	FB	2013-01-11	31.3	32.0	31.1	31.7	89598000	31.7
9	FB	2013-01-14	32.1	32.2	30.6	31.0	98892800	31.0
10	FB	2013-01-15	30.6	31.7	29.9	30.1	173242600	30.1

how to compute 7-days average?



easy as this!

```
mean_run(close, k = 7, idx = date),
```



easy as this!

mean_run(close, k = 7, idx = date),



value to calculate



easy as this!

```
mean_run(close, k = 7, idx = date),
```



number of periods
in window



easy as this!

mean_run(close, k = 7, idx = date),



periods (date or integer)



Same API for other functions

◦

xxxx_run(close, k = 7, idx = date),



7-days lag

..

```
lag_run(close, k = 7, idx = date),
```



number of observations in last 7-days

•

length_run(close, k = 7, idx = date),



7-days which

..

which_run(close, k = 7, idx = date),



actual streak in 7-days window

streak_run(close, k = 7, idx = date),



7-days windows

•

```
window_run(close, k = 7, idx = date),
```



Future plans

- get community feedback
- more window aggregations
- runner::apply_run
- plotting extension



Thank you.



gogonzo



Dawid Kałędkowski

