smires – Calculating Hydrological Metrics for Univariate Time Series

Tobias Gauster, Gregor Laaha

Institute of Applied Statistics and Computing
BOKU, Vienna

useR!2017 BRUSSELS

Motivation

Happy families are all alike; ...
– Leo Tolstoi, Anna Karenina

Hydrological/ecological metrics are all alike.

SMIRES Science and Management of Intermittent Rivers and Ephemeral Streams

Working Group 1: *Prevalence, distribution and trends of IRES*

The R package `smires`

- Provides a framework for computing hydro-/ecological metrics.
- Contains sample datasets of every participating European country.

- Is aimed at unexperienced useRs.
- Has only a few requirements on input data.
- Can work with binary data (flow, no-flow).

`https://github.com/mundl/smires`

Functions

Preprocessing

`is.intermittent()`	checks for intermittency
`validate()`	validates input time series

Computing Metrics

`metric()`	continuous time series
`smires()`	binary time series, e.g. for intermittent rivers

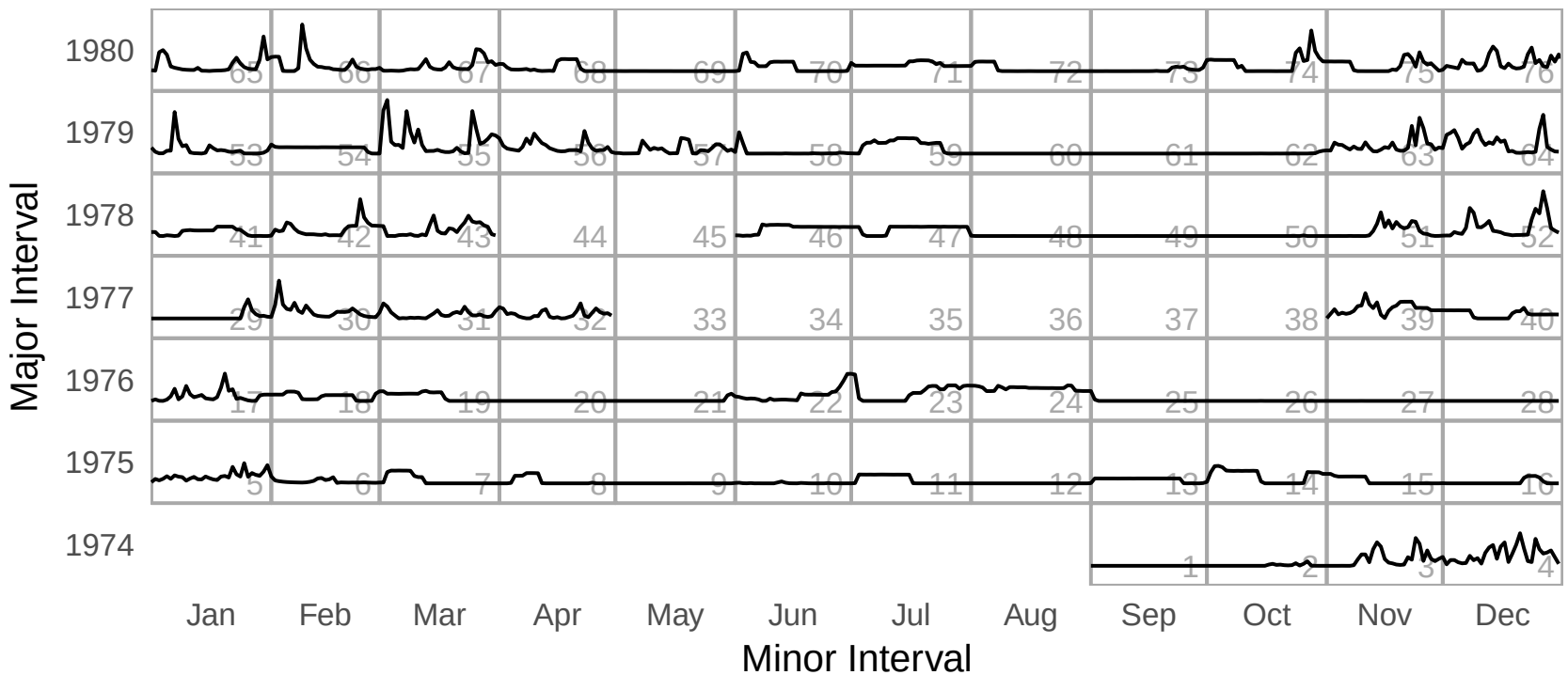
Low level functions

`group_by_interval()`	assigns indices and groups
`find_events()`	derives a binary time series

Metrics for continuous time series

E.g. mean annual maximum discharge

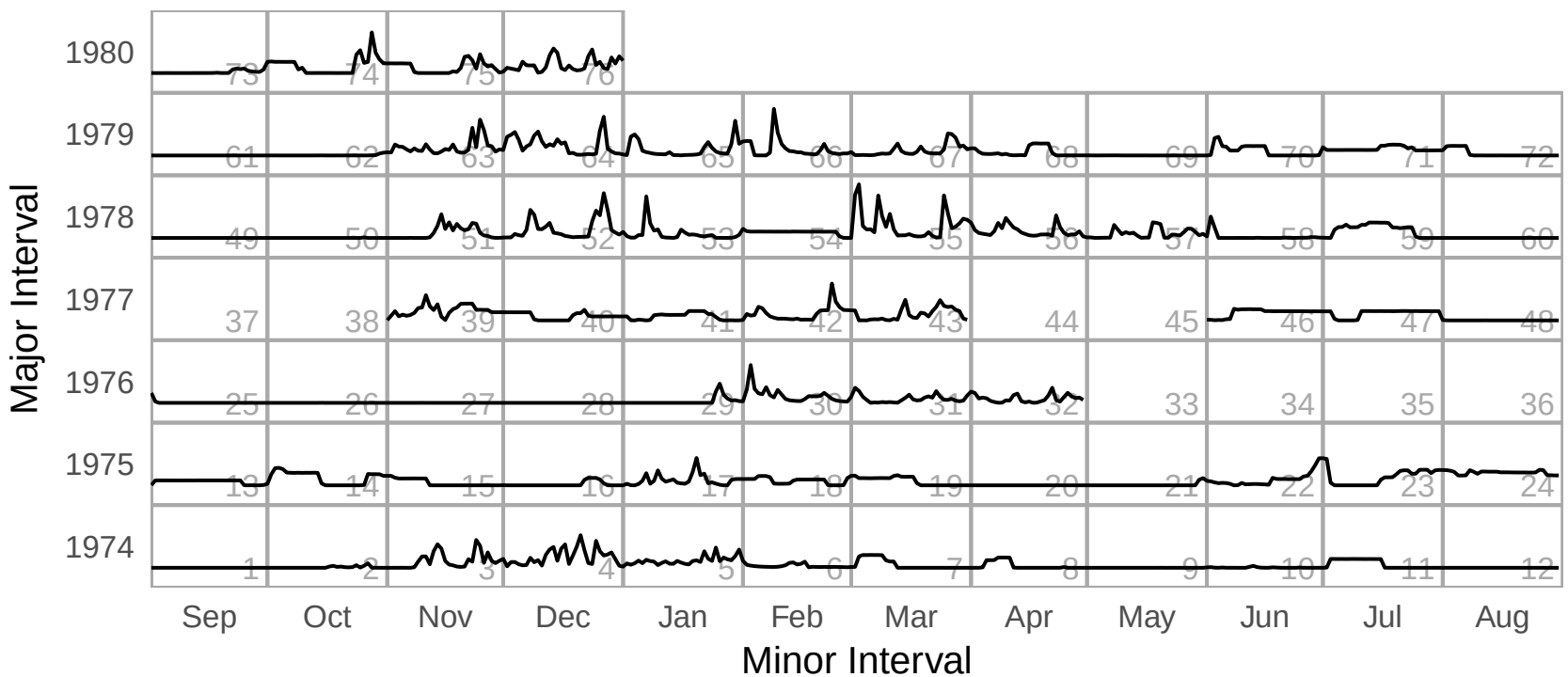
```
> metric(balder,
+        fun_major = max, fun_total = mean,
+        plot = T, drop = T, outvar = "mean.annual.max")

mean.annual.max
       5.438429
```



Metrics for continuous time series

E.g. mean annual maximum discharge, hydrological year starting in September

```
> metric(balder, major = 244,
+        fun_major = max, fun_total = mean,
+        plot = T, drop = T, outvar = "mean.annual.max")

mean.annual.max
       5.696429
```



Metrics for continuous time series

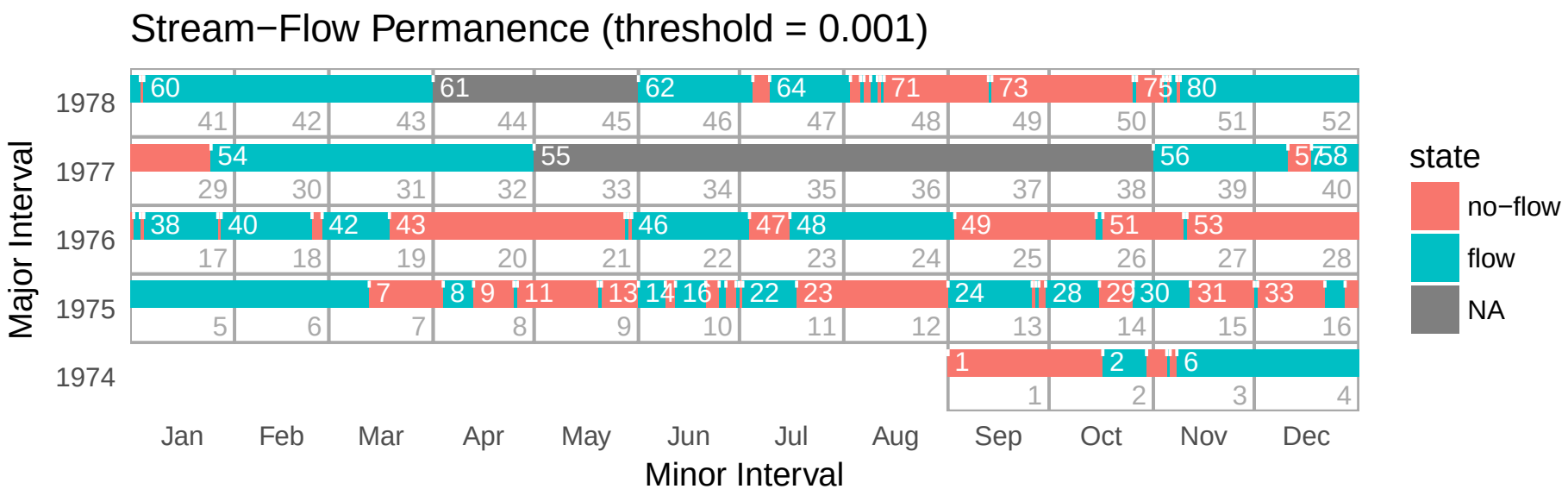
E.g. maximum annual discharge, hydrological year starting in September

```
> metric(balder, major = 244, fun_major = max)

# A tibble: 7 x 2
  major variable
  <ord>    <dbl>
1  1974    4.720
2  1975    3.955
3  1976    5.471
4  1977    5.334
5  1978    7.757
6  1979    6.753
7  1980    5.885
```

Metrics for binary time series

Threshold is 1 l/s

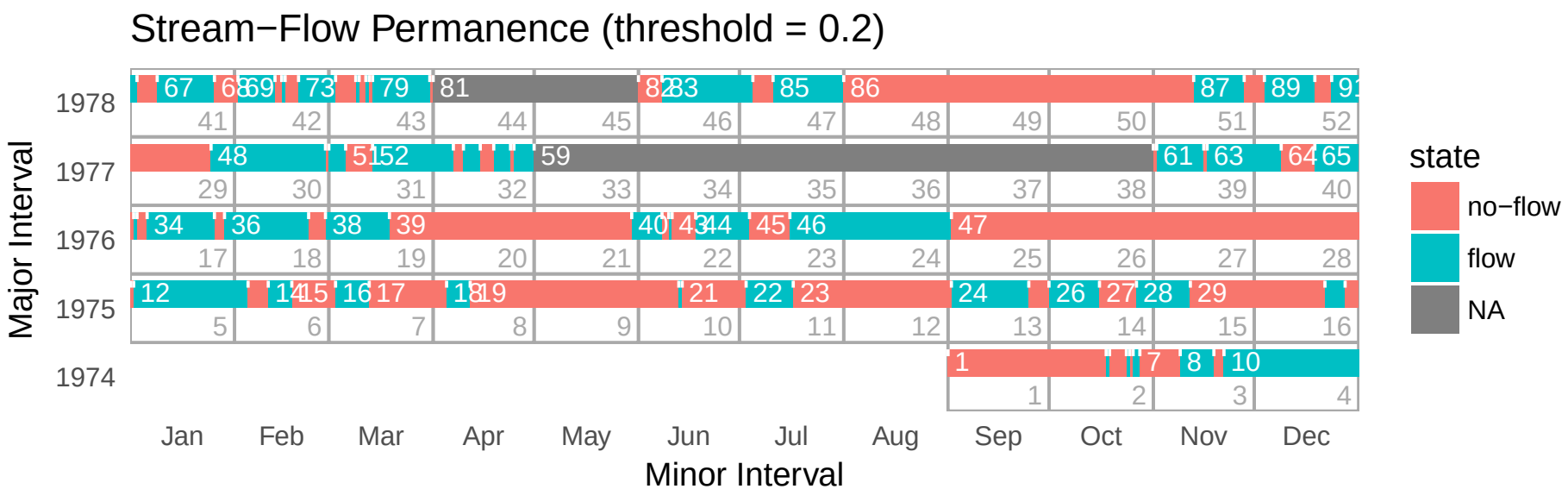
```
> smires(balder, plot = T) %>% head(3)

# A tibble: 3 x 9
  event   state      start         end group duration major minor variabl
  <ord>  <fctr>     <date>     <date> <dbl>   <time> <ord> <ord>    <time
1     1 no-flow 1974-09-01 1974-10-17     1  46 days  1974   Sep  46 day
2     2    flow 1974-10-17 1974-10-30     2  13 days  1974   Oct  13 day
3     3 no-flow 1974-10-30 1974-11-05     2   6 days  1974   Oct   6 day
```



Metrics for binary time series

Threshold is 20 l/s

```
> smires(balder, threshold = 0.2, plot = T) %>% head(3)

# A tibble: 3 x 9
  event   state      start         end group duration major minor variabl
  <ord>  <fctr>     <date>     <date> <dbl>   <time> <ord> <ord>   <time
1     1 no-flow 1974-09-01 1974-10-18     1  47 days  1974   Sep  47 day
2     2    flow 1974-10-18 1974-10-19     2   1 days  1974   Oct   1 day
3     3 no-flow 1974-10-19 1974-10-24     2   5 days  1974   Oct   5 day
```



Metrics for binary time series

Mean annual maximum duration of events

```
> rm(balder)
```

```
> smires(balder,
+        fun_major = max,
+        drop_na = "major")

# A tibble: 10 x 3
   major   state variable
   <ord>  <fctr>   <time>
 1  1974 no-flow  46 days
 2  1974    flow 125 days
 3  1975 no-flow  45 days
 4  1975    flow  25 days
 5  1976 no-flow  76 days
 6  1976    flow  49 days
 7  1979 no-flow  20 days
 8  1979    flow 214 days
 9  1980 no-flow  12 days
10  1980    flow  46 days
```

```
> smires(balder,
+        fun_major = max,
+        fun_total = mean,
+        drop_na = "major")

# A tibble: 2 x 2
    state  variable
   <fctr>    <time>
1 no-flow 39.8 days
2    flow 91.8 days
```

Metrics for binary time series

Low level functions

```
> # Appending the group and interval indices
> grouped <- group_by_interval(balder)
> head(grouped, 3)

# A tibble: 3 x 6
        time discharge major minor group  hday
      <date>     <dbl> <ord> <ord> <dbl> <dbl>
1 1974-09-01         0  1974   Sep     1   244
2 1974-09-02         0  1974   Sep     1   245
3 1974-09-03         0  1974   Sep     1   246

> # Detecting events
> find_events(grouped, rule = "start") %>% head(3)

# A tibble: 3 x 8
  event   state      start        end group duration major minor
  <ord>  <fctr>     <date>     <date> <dbl>   <time> <ord> <ord>
1     1 no-flow 1974-09-01 1974-10-17     1  46 days  1974   Sep
2     2    flow 1974-10-17 1974-10-30     2  13 days  1974   Oct
3     3 no-flow 1974-10-30 1974-11-05     2   6 days  1974   Oct
```

Varying the minor interval: Seasonal analysis

```
> seasons <- c(spring = 60, summer = 152,
+              autumn = 244, winter = 335)

> smires(balder, minor = seasons, fun_minor = max)

# A tibble: 8 x 3
   minor   state variable
   <ord>  <fctr>   <time>
1 spring no-flow  70 days
2 spring    flow  96 days
3 summer no-flow  45 days
4 summer    flow  49 days
5 autumn no-flow  76 days
6 autumn    flow 214 days
7 winter no-flow  20 days
8 winter    flow  96 days
```

Summary

- The package **smires** provides a framework to compute metrics of univariate time series.
- Either continuous or binary time series.
- Free choice of the aggregation period (calendar years, hydrological years, months, seasons, ...).
- Free choice of the aggregation function.
- github: `https://github.com/mundl/smires`

Acknowledgements

This research is based upon work from COST Action CA15113 (SMIRES, Science and Management of Intermittent Rivers and Ephemeral Streams, www.smires.eu), supported by COST (European Cooperation in Science and Technology)

SMIRES Science and Management of Intermittent Rivers and Ephemeral Streams

cost EUROPEAN COOPERATION IN SCIENCE AND TECHNOLOGY