phinterval: An R package for representing and manipulating timespans with gaps

Ethan Sansom[1]

1 Department of Statistical Sciences, University of Toronto

Summary

`phinterval` is an R (R Core Team 2022) package for representing and manipulating time spans that may contain gaps. It implements the *phinterval* vector class, designed as an extension of the `lubridate` (Grolemund and Wickham 2011) package's *Interval* class, to represent continuous, disjoint, empty, and unknown spans of time.

Functionality for manipulating these spans includes:

- Performing set operations: union, intersection, difference, and complement.
- Merging overlapping or adjacent intervals into non-overlapping sets of time spans.
- Testing whether time spans, dates, or times fall within one another or overlap.

Statement of Need

Because of the complexities of accurately representing dates and times, including adjustments for time zones, daylight saving transitions, and leap years or seconds, manipulating time spans is a common source of frustration and error-prone code for analysts Tiwari et al. (2025). Several R packages, notably `lubridate` and `ivs` (Vaughan 2023), provide intuitive interfaces for representing and manipulating time spans that handle these complexities internally, reducing the cognitive load required of users and the likelihood of mistakes.

To the author's knowledge, however, no existing package supports empty or discontinuous time spans. Users encountering these spans - for example, the intersection of two non-overlapping intervals - receive an error (as in `ivs`) or unintuitive results (as in `lubridate`), forcing workarounds or potentially leading to uncaught mistakes. The `phinterval` package addresses this gap by providing explicit representations of disjoint and empty time spans, enabling operations such as union, intersection, and set difference to be performed accurately on arbitrary intervals. Its interface closely mirrors that of `lubridate`, and all `phinterval` functions accept `lubridate` *Interval* vectors as inputs, allowing analysts to safely integrate `phinterval` into their existing workflows and work with a broader range of temporal data.

Examples

To demonstrate the utility of the *phinterval* class, consider a modified example from the `lubridate` package vignettes. Suppose two colleagues are taking 5-day vacations: one to Greece in early January and the other to Brazil in mid-February.

```
greece <- interval(ymd("2020-01-01"), ymd("2020-01-06"))
brazil <- interval(ymd("2020-02-11"), ymd("2020-02-16"))
```

If we compute the union of these intervals using the `union()` method from `lubridate`:

```
union(greece, brazil)
```

```
## [1] 2020-01-01 UTC--2020-02-16 UTC
```

The resulting interval includes the intervening time between the disjoint vacations. The `phinterval` package provides a drop-in replacement, `phint_union()`, which accepts the same arguments but returns a *phinterval* vector.

```
phint_union(greece, brazil)
```

```
## <phinterval<UTC>[1]>
## [1] {2020-01-01--2020-01-06, 2020-02-11--2020-02-16}
```

The result is a disjoint time span, preserving the gap between the two vacations.

In simple calculations, this distinction can easily lead to unexpected results. For example, to calculate the number of days that either employee is out of the office, one might take the union of their vacation spans and then calculate the duration in days:

```
as_duration(union(greece, brazil)) / ddays()
```

```
## [1] 46
```

```
as_duration(phint_union(greece, brazil)) / ddays()
```

```
## [1] 10
```

While experienced `lubridate` users can anticipate and work around these cases, `phinterval` provides an intuitive alternative which can easily be substituted into existing analyses.

References

Grolemund, Garrett, and Hadley Wickham. 2011. "Dates and Times Made Easy with lubridate." *Journal of Statistical Software* 40 (3): 1–25. https://doi.org/10.18637/jss.v040.i03.

R Core Team. 2022. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://doi.org/10.32614/r.manuals.

Tiwari, Shrey, Serena Chen, Alexander Joukov, Peter Vandervelde, Ao Li, and Rohan Padhye. 2025. "It's about Time: An Empirical Study of Date and Time Bugs in Open-Source Python Software." In *2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)*, 39–51. https://doi.org/10.1109/MSR66628.2025.00020.

Vaughan, Davis. 2023. *Ivs: Interval Vectors*. https://doi.org/10.32614/cran.package.ivs.