Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. doi: 10.18637/jss.v000.i00

Working with Missing, Approximate, Uncertain, Sets and Ranges of Dates with messydates

James Hollway
Graduate Institute of
International and Development Studies

Henrique Sposito
Graduate Institute of
International and Development Studies

Abstract

This paper presents the **messydates** package for R, which facilitates working with 'messy' dates. Messy dates are common when studying historical and sometimes even current phenomena, and can create various technical problems for the data analyst. The paper highlights these problems and offers practical advice on how to solve them using **messydates**. The paper also introduces a conceptual framework for resolving messydates into more familiar date classes in R ready for analysis.

Keywords: dates, ISO, R.

1. Introduction

Dates are often messy. Whether historical (or ancient), future, or even recent, we often only know approximately when an event occurred, that it happened within a particular period, an unreliable source means a date should be flagged as uncertain, or sources offer multiple, competing dates.

As researchers, we often recognize this messiness but are forced to force non-existent precision on data so we can proceed with analysis. For example, if we only know something happened in a given month or year, we might just opt for the start of that month (e.g. `2021-07-01`) or year (`2021-01-01`), assuming that to err on the earlier (or later) side is a justifiable bias. However, this can create issues for inference in which sequence or timing is important. The goal of **messydates** is to help with this problem by retaining and working with various kinds of date imprecision.

1.1. A quick overview

messydates implements the extended annotation standard for dates, the Extended Date/Time Format (EDTF), outlined in ISO 8601-2_2019(E) for R. These include standardised notation for:

- unspecified date(component)s, e.g. `2012-XX-01` for the first of some unknown month in 2012 or 2012-01 for some unknown day in January 2012
- approximate date(component)s, e.g. `2012-01-12~` for approximately the 12th of January 2012
- uncertain date(component)s, e.g. `2012-01-12?` where this data point is based on an unreliable source
- sets of dates, e.g. `{2012-01-01,2012-01-12}` where the date can be both 1 January 2012 and 12 January 2012
- ranges of dates, e.g. `2012-01-01..2012-01-12` for all dates between the 1 January 2012 and 12 January 2012 inclusive

messydates contains a set of tools for constructing and coercing into and from the `messydt` class. This date class allows regular dates to be annotated to express unspecified date components, approximate or uncertain date components, date ranges, and sets of dates.

Importantly, the package also includes a function for unpacking or expanding sets or ranges of dates into all dates consistent with how the date or set of dates is specified or annotated. Methods are also offered that can be used to make explicit how researchers convert date imprecision into precise dates for analysis, such as getting the `min()`, `max()`, or even a `random()` date from among the dates consistent with a set or range of dates. This greatly facilitates research transparency as well as robustness checks.

1.2. Relationship to other packages

messydates offers a new date class, but one that comes with methods for converting from and into common date classes such as `Date`, `POSIXct`, and `POSIXlt.` It is thus fully compatible with packages such as **lubridate** (Grolemund and Wickham 2011) and **anytime** (Eddelbuettel 2019). It is also compatible with **unstruwwel**, which also parses historic dates in R, though the emphasis of our package is on working with these dates.

2. R code

2.1. A new class

messydates contains a set of tools for constructing and coercing into and from the messydt class. This date class implements ISO 8601-2:2019(E) and allows regular dates to be annotated to express unspecified date components, approximate or uncertain date components, date ranges, and sets of dates. The function `as_messydate()` handles the coercion to messydt class.

```
R> library(messydates)
R> suppressPackageStartupMessages(library(lubridate))
R> library(tibble)
```

```
R> suppressPackageStartupMessages(library(dplyr))
R> dates_comparison <- tibble::tribble(~Example, ~OriginalDate,
+                                      "Normal date", "2010-01-01",
+                                      "Historical date", "1291-08-01",
+                                      "Very historical date", "476",
+                                      "Really historical date", "33 BC",
+                                      "Clearly future date", "9999-12-31",
+                                      "Not so clearly future date", "2599-12-31",
+                                      "Range of dates", "2019-11-01:2020-01-01",
+                                      "Uncertain date", "2001-01-01?",
+                                      "Set of dates", "2021-5-26, 2021-6-10, 2021-11-19, 2
R> dates_comparison %>% dplyr::mutate(base = as.Date(OriginalDate),
+                                     lubridate = suppressWarnings(lubridate::as_date(Origi
+                                     messydates = messydates::as_messydate(OriginalDate))
+   print()

# A tibble: 9 x 5
  Example                    OriginalDate        base       lubridate  messydates
  <chr>                      <chr>               <date>     <date>     <messydt>
1 Normal date                2010-01-01          2010-01-01 2010-01-01 2010-01-0~
2 Historical date            1291-08-01          1291-08-01 1291-08-01 1291-08-0~
3 Very historical date       476                 NA         NA         0476     ~
4 Really historical date     33 BC               NA         NA         -0033    ~
5 Clearly future date        9999-12-31          9999-12-31 9999-12-31 9999-12-3~
6 Not so clearly future date 2599-12-31          2599-12-31 2599-12-31 2599-12-3~
7 Range of dates             2019-11-01:2020-0~ 2019-11-01 2019-11-01 2019-11-0~
8 Uncertain date             2001-01-01?         2001-01-01 2001-01-01 2001-01-0~
9 Set of dates               2021-5-26, 2021-6~ 2021-05-26 NA         {2021-05--
```

2.2. Annotate

Some datasets have, for example, an arbitrary cut off point for start and end points, but these are often coded as precise dates when they are not necessarily the real start or end dates. The annotate functions helps annotate uncertainty and approximation to dates. Inaccurate start or end dates can be represented by an affix indicating "on or before", if used as a prefix (e.g. `..1816-01-01`), or indicating "on or after", if used as a suffix (e.g. `2016-12-31..`). Approximate dates are indicated by adding a `~` to year, month, or day components, as well as groups of components or whole dates to estimate values that are possibly correct (e.g. `2003-03-03~`). Day, month, or year, uncertainty can be indicated by adding a `?` to a possibly dubious date (e.g. `1916-10-10?`) or date component (e.g. `1916-?10-10`).

```
R> dates_annotate <- tibble::tibble(Beg = as_messydate(c("1816-01-01", "1916-01-01", "2016
+                                   End = as_messydate(c("1816-12-31", "1916-12-31", "2016-
R> dplyr::mutate(dates_annotate, Beg = ifelse(Beg <= "1816-01-01", on_or_before(Beg), Beg)

# A tibble: 3 x 2
  Beg          End
```

```
  <chr>        <messydt>
1 ..1816-01-01 1816-12-31
2 1916-01-01   1916-12-31
3 2016-01-01   2016-12-31

R> dplyr::mutate(dates_annotate, End = ifelse(End >= "2016-01-01", on_or_after(End), End))

# A tibble: 3 x 2
  Beg        End
  <messydt>  <chr>
1 1816-01-01 1816-12-31
2 1916-01-01 1916-12-31
3 2016-01-01 2016-12-31..

R> dplyr::mutate(dates_annotate, Beg = ifelse(Beg == "1916-01-01", as_approximate(Beg), Be

# A tibble: 3 x 2
  Beg         End
  <chr>       <messydt>
1 1816-01-01  1816-12-31
2 1916-01-01~ 1916-12-31
3 2016-01-01  2016-12-31

R> dplyr::mutate(dates_annotate, End = ifelse(End == "1916-12-31", as_uncertain(End), End)

# A tibble: 3 x 2
  Beg        End
  <messydt>  <chr>
1 1816-01-01 1816-12-31
2 1916-01-01 1916-12-31?
3 2016-01-01 2016-12-31
```

2.3. Expand

Expand functions transform date ranges, sets of dates, and unspecified or approximate dates (annotated with '..', '{ , }', 'XX' or '~') into lists of dates. As these dates may refer to several possible dates, the function "opens" these values to include all the possible dates implied.

```
R> dates_expand <- as_messydate(c("2008-03-25", "2001-01?", "2001",
+                                  "2001-01-01..2001-02-02", "{2001-01-01,2001-02-02}",
+                                  "2008-XX-31", "28 BC"))
R> expand(dates_expand)

[[1]]
[1] "2008-03-25"
```

```
[[2]]
 [1] "2001-01-01" "2001-01-02" "2001-01-03" "2001-01-04" "2001-01-05"
 [6] "2001-01-06" "2001-01-07" "2001-01-08" "2001-01-09" "2001-01-10"
[11] "2001-01-11" "2001-01-12" "2001-01-13" "2001-01-14" "2001-01-15"
[16] "2001-01-16" "2001-01-17" "2001-01-18" "2001-01-19" "2001-01-20"
[21] "2001-01-21" "2001-01-22" "2001-01-23" "2001-01-24" "2001-01-25"
[26] "2001-01-26" "2001-01-27" "2001-01-28" "2001-01-29" "2001-01-30"
[31] "2001-01-31"

[[3]]
  [1] "2001-01-01" "2001-01-02" "2001-01-03" "2001-01-04" "2001-01-05"
  [6] "2001-01-06" "2001-01-07" "2001-01-08" "2001-01-09" "2001-01-10"
 [11] "2001-01-11" "2001-01-12" "2001-01-13" "2001-01-14" "2001-01-15"
 [16] "2001-01-16" "2001-01-17" "2001-01-18" "2001-01-19" "2001-01-20"
 [21] "2001-01-21" "2001-01-22" "2001-01-23" "2001-01-24" "2001-01-25"
 [26] "2001-01-26" "2001-01-27" "2001-01-28" "2001-01-29" "2001-01-30"
 [31] "2001-01-31" "2001-02-01" "2001-02-02" "2001-02-03" "2001-02-04"
 [36] "2001-02-05" "2001-02-06" "2001-02-07" "2001-02-08" "2001-02-09"
 [41] "2001-02-10" "2001-02-11" "2001-02-12" "2001-02-13" "2001-02-14"
 [46] "2001-02-15" "2001-02-16" "2001-02-17" "2001-02-18" "2001-02-19"
 [51] "2001-02-20" "2001-02-21" "2001-02-22" "2001-02-23" "2001-02-24"
 [56] "2001-02-25" "2001-02-26" "2001-02-27" "2001-02-28" "2001-03-01"
 [61] "2001-03-02" "2001-03-03" "2001-03-04" "2001-03-05" "2001-03-06"
 [66] "2001-03-07" "2001-03-08" "2001-03-09" "2001-03-10" "2001-03-11"
 [71] "2001-03-12" "2001-03-13" "2001-03-14" "2001-03-15" "2001-03-16"
 [76] "2001-03-17" "2001-03-18" "2001-03-19" "2001-03-20" "2001-03-21"
 [81] "2001-03-22" "2001-03-23" "2001-03-24" "2001-03-25" "2001-03-26"
 [86] "2001-03-27" "2001-03-28" "2001-03-29" "2001-03-30" "2001-03-31"
 [91] "2001-04-01" "2001-04-02" "2001-04-03" "2001-04-04" "2001-04-05"
 [96] "2001-04-06" "2001-04-07" "2001-04-08" "2001-04-09" "2001-04-10"
[101] "2001-04-11" "2001-04-12" "2001-04-13" "2001-04-14" "2001-04-15"
[106] "2001-04-16" "2001-04-17" "2001-04-18" "2001-04-19" "2001-04-20"
[111] "2001-04-21" "2001-04-22" "2001-04-23" "2001-04-24" "2001-04-25"
[116] "2001-04-26" "2001-04-27" "2001-04-28" "2001-04-29" "2001-04-30"
[121] "2001-05-01" "2001-05-02" "2001-05-03" "2001-05-04" "2001-05-05"
[126] "2001-05-06" "2001-05-07" "2001-05-08" "2001-05-09" "2001-05-10"
[131] "2001-05-11" "2001-05-12" "2001-05-13" "2001-05-14" "2001-05-15"
[136] "2001-05-16" "2001-05-17" "2001-05-18" "2001-05-19" "2001-05-20"
[141] "2001-05-21" "2001-05-22" "2001-05-23" "2001-05-24" "2001-05-25"
[146] "2001-05-26" "2001-05-27" "2001-05-28" "2001-05-29" "2001-05-30"
[151] "2001-05-31" "2001-06-01" "2001-06-02" "2001-06-03" "2001-06-04"
[156] "2001-06-05" "2001-06-06" "2001-06-07" "2001-06-08" "2001-06-09"
[161] "2001-06-10" "2001-06-11" "2001-06-12" "2001-06-13" "2001-06-14"
[166] "2001-06-15" "2001-06-16" "2001-06-17" "2001-06-18" "2001-06-19"
[171] "2001-06-20" "2001-06-21" "2001-06-22" "2001-06-23" "2001-06-24"
[176] "2001-06-25" "2001-06-26" "2001-06-27" "2001-06-28" "2001-06-29"
[181] "2001-06-30" "2001-07-01" "2001-07-02" "2001-07-03" "2001-07-04"
```

```
[186] "2001-07-05" "2001-07-06" "2001-07-07" "2001-07-08" "2001-07-09"
[191] "2001-07-10" "2001-07-11" "2001-07-12" "2001-07-13" "2001-07-14"
[196] "2001-07-15" "2001-07-16" "2001-07-17" "2001-07-18" "2001-07-19"
[201] "2001-07-20" "2001-07-21" "2001-07-22" "2001-07-23" "2001-07-24"
[206] "2001-07-25" "2001-07-26" "2001-07-27" "2001-07-28" "2001-07-29"
[211] "2001-07-30" "2001-07-31" "2001-08-01" "2001-08-02" "2001-08-03"
[216] "2001-08-04" "2001-08-05" "2001-08-06" "2001-08-07" "2001-08-08"
[221] "2001-08-09" "2001-08-10" "2001-08-11" "2001-08-12" "2001-08-13"
[226] "2001-08-14" "2001-08-15" "2001-08-16" "2001-08-17" "2001-08-18"
[231] "2001-08-19" "2001-08-20" "2001-08-21" "2001-08-22" "2001-08-23"
[236] "2001-08-24" "2001-08-25" "2001-08-26" "2001-08-27" "2001-08-28"
[241] "2001-08-29" "2001-08-30" "2001-08-31" "2001-09-01" "2001-09-02"
[246] "2001-09-03" "2001-09-04" "2001-09-05" "2001-09-06" "2001-09-07"
[251] "2001-09-08" "2001-09-09" "2001-09-10" "2001-09-11" "2001-09-12"
[256] "2001-09-13" "2001-09-14" "2001-09-15" "2001-09-16" "2001-09-17"
[261] "2001-09-18" "2001-09-19" "2001-09-20" "2001-09-21" "2001-09-22"
[266] "2001-09-23" "2001-09-24" "2001-09-25" "2001-09-26" "2001-09-27"
[271] "2001-09-28" "2001-09-29" "2001-09-30" "2001-10-01" "2001-10-02"
[276] "2001-10-03" "2001-10-04" "2001-10-05" "2001-10-06" "2001-10-07"
[281] "2001-10-08" "2001-10-09" "2001-10-10" "2001-10-11" "2001-10-12"
[286] "2001-10-13" "2001-10-14" "2001-10-15" "2001-10-16" "2001-10-17"
[291] "2001-10-18" "2001-10-19" "2001-10-20" "2001-10-21" "2001-10-22"
[296] "2001-10-23" "2001-10-24" "2001-10-25" "2001-10-26" "2001-10-27"
[301] "2001-10-28" "2001-10-29" "2001-10-30" "2001-10-31" "2001-11-01"
[306] "2001-11-02" "2001-11-03" "2001-11-04" "2001-11-05" "2001-11-06"
[311] "2001-11-07" "2001-11-08" "2001-11-09" "2001-11-10" "2001-11-11"
[316] "2001-11-12" "2001-11-13" "2001-11-14" "2001-11-15" "2001-11-16"
[321] "2001-11-17" "2001-11-18" "2001-11-19" "2001-11-20" "2001-11-21"
[326] "2001-11-22" "2001-11-23" "2001-11-24" "2001-11-25" "2001-11-26"
[331] "2001-11-27" "2001-11-28" "2001-11-29" "2001-11-30" "2001-12-01"
[336] "2001-12-02" "2001-12-03" "2001-12-04" "2001-12-05" "2001-12-06"
[341] "2001-12-07" "2001-12-08" "2001-12-09" "2001-12-10" "2001-12-11"
[346] "2001-12-12" "2001-12-13" "2001-12-14" "2001-12-15" "2001-12-16"
[351] "2001-12-17" "2001-12-18" "2001-12-19" "2001-12-20" "2001-12-21"
[356] "2001-12-22" "2001-12-23" "2001-12-24" "2001-12-25" "2001-12-26"
[361] "2001-12-27" "2001-12-28" "2001-12-29" "2001-12-30" "2001-12-31"

[[4]]
 [1] "2001-01-01" "2001-01-02" "2001-01-03" "2001-01-04" "2001-01-05"
 [6] "2001-01-06" "2001-01-07" "2001-01-08" "2001-01-09" "2001-01-10"
[11] "2001-01-11" "2001-01-12" "2001-01-13" "2001-01-14" "2001-01-15"
[16] "2001-01-16" "2001-01-17" "2001-01-18" "2001-01-19" "2001-01-20"
[21] "2001-01-21" "2001-01-22" "2001-01-23" "2001-01-24" "2001-01-25"
[26] "2001-01-26" "2001-01-27" "2001-01-28" "2001-01-29" "2001-01-30"
[31] "2001-01-31" "2001-02-01" "2001-02-02"

[[5]]
```

```
[1] "2001-01-01" "2001-02-02"

[[6]]
 [1] "2008-01-31" "2008-02-28" "2008-03-31" "2008-04-30" "2008-05-31"
 [6] "2008-06-30" "2008-07-31" "2008-08-31" "2008-09-30" "2008-10-31"
[11] "2008-11-30" "2008-12-31"

[[7]]
  [1] "-0028-01-01" "-0028-01-02" "-0028-01-03" "-0028-01-04" "-0028-01-05"
  [6] "-0028-01-06" "-0028-01-07" "-0028-01-08" "-0028-01-09" "-0028-01-10"
 [11] "-0028-01-11" "-0028-01-12" "-0028-01-13" "-0028-01-14" "-0028-01-15"
 [16] "-0028-01-16" "-0028-01-17" "-0028-01-18" "-0028-01-19" "-0028-01-20"
 [21] "-0028-01-21" "-0028-01-22" "-0028-01-23" "-0028-01-24" "-0028-01-25"
 [26] "-0028-01-26" "-0028-01-27" "-0028-01-28" "-0028-01-29" "-0028-01-30"
 [31] "-0028-01-31" "-0028-02-01" "-0028-02-02" "-0028-02-03" "-0028-02-04"
 [36] "-0028-02-05" "-0028-02-06" "-0028-02-07" "-0028-02-08" "-0028-02-09"
 [41] "-0028-02-10" "-0028-02-11" "-0028-02-12" "-0028-02-13" "-0028-02-14"
 [46] "-0028-02-15" "-0028-02-16" "-0028-02-17" "-0028-02-18" "-0028-02-19"
 [51] "-0028-02-20" "-0028-02-21" "-0028-02-22" "-0028-02-23" "-0028-02-24"
 [56] "-0028-02-25" "-0028-02-26" "-0028-02-27" "-0028-02-28" "-0028-02-29"
 [61] "-0028-03-01" "-0028-03-02" "-0028-03-03" "-0028-03-04" "-0028-03-05"
 [66] "-0028-03-06" "-0028-03-07" "-0028-03-08" "-0028-03-09" "-0028-03-10"
 [71] "-0028-03-11" "-0028-03-12" "-0028-03-13" "-0028-03-14" "-0028-03-15"
 [76] "-0028-03-16" "-0028-03-17" "-0028-03-18" "-0028-03-19" "-0028-03-20"
 [81] "-0028-03-21" "-0028-03-22" "-0028-03-23" "-0028-03-24" "-0028-03-25"
 [86] "-0028-03-26" "-0028-03-27" "-0028-03-28" "-0028-03-29" "-0028-03-30"
 [91] "-0028-03-31" "-0028-04-01" "-0028-04-02" "-0028-04-03" "-0028-04-04"
 [96] "-0028-04-05" "-0028-04-06" "-0028-04-07" "-0028-04-08" "-0028-04-09"
[101] "-0028-04-10" "-0028-04-11" "-0028-04-12" "-0028-04-13" "-0028-04-14"
[106] "-0028-04-15" "-0028-04-16" "-0028-04-17" "-0028-04-18" "-0028-04-19"
[111] "-0028-04-20" "-0028-04-21" "-0028-04-22" "-0028-04-23" "-0028-04-24"
[116] "-0028-04-25" "-0028-04-26" "-0028-04-27" "-0028-04-28" "-0028-04-29"
[121] "-0028-04-30" "-0028-05-01" "-0028-05-02" "-0028-05-03" "-0028-05-04"
[126] "-0028-05-05" "-0028-05-06" "-0028-05-07" "-0028-05-08" "-0028-05-09"
[131] "-0028-05-10" "-0028-05-11" "-0028-05-12" "-0028-05-13" "-0028-05-14"
[136] "-0028-05-15" "-0028-05-16" "-0028-05-17" "-0028-05-18" "-0028-05-19"
[141] "-0028-05-20" "-0028-05-21" "-0028-05-22" "-0028-05-23" "-0028-05-24"
[146] "-0028-05-25" "-0028-05-26" "-0028-05-27" "-0028-05-28" "-0028-05-29"
[151] "-0028-05-30" "-0028-05-31" "-0028-06-01" "-0028-06-02" "-0028-06-03"
[156] "-0028-06-04" "-0028-06-05" "-0028-06-06" "-0028-06-07" "-0028-06-08"
[161] "-0028-06-09" "-0028-06-10" "-0028-06-11" "-0028-06-12" "-0028-06-13"
[166] "-0028-06-14" "-0028-06-15" "-0028-06-16" "-0028-06-17" "-0028-06-18"
[171] "-0028-06-19" "-0028-06-20" "-0028-06-21" "-0028-06-22" "-0028-06-23"
[176] "-0028-06-24" "-0028-06-25" "-0028-06-26" "-0028-06-27" "-0028-06-28"
[181] "-0028-06-29" "-0028-06-30" "-0028-07-01" "-0028-07-02" "-0028-07-03"
[186] "-0028-07-04" "-0028-07-05" "-0028-07-06" "-0028-07-07" "-0028-07-08"
[191] "-0028-07-09" "-0028-07-10" "-0028-07-11" "-0028-07-12" "-0028-07-13"
```

```
[196] "-0028-07-14" "-0028-07-15" "-0028-07-16" "-0028-07-17" "-0028-07-18"
[201] "-0028-07-19" "-0028-07-20" "-0028-07-21" "-0028-07-22" "-0028-07-23"
[206] "-0028-07-24" "-0028-07-25" "-0028-07-26" "-0028-07-27" "-0028-07-28"
[211] "-0028-07-29" "-0028-07-30" "-0028-07-31" "-0028-08-01" "-0028-08-02"
[216] "-0028-08-03" "-0028-08-04" "-0028-08-05" "-0028-08-06" "-0028-08-07"
[221] "-0028-08-08" "-0028-08-09" "-0028-08-10" "-0028-08-11" "-0028-08-12"
[226] "-0028-08-13" "-0028-08-14" "-0028-08-15" "-0028-08-16" "-0028-08-17"
[231] "-0028-08-18" "-0028-08-19" "-0028-08-20" "-0028-08-21" "-0028-08-22"
[236] "-0028-08-23" "-0028-08-24" "-0028-08-25" "-0028-08-26" "-0028-08-27"
[241] "-0028-08-28" "-0028-08-29" "-0028-08-30" "-0028-08-31" "-0028-09-01"
[246] "-0028-09-02" "-0028-09-03" "-0028-09-04" "-0028-09-05" "-0028-09-06"
[251] "-0028-09-07" "-0028-09-08" "-0028-09-09" "-0028-09-10" "-0028-09-11"
[256] "-0028-09-12" "-0028-09-13" "-0028-09-14" "-0028-09-15" "-0028-09-16"
[261] "-0028-09-17" "-0028-09-18" "-0028-09-19" "-0028-09-20" "-0028-09-21"
[266] "-0028-09-22" "-0028-09-23" "-0028-09-24" "-0028-09-25" "-0028-09-26"
[271] "-0028-09-27" "-0028-09-28" "-0028-09-29" "-0028-09-30" "-0028-10-01"
[276] "-0028-10-02" "-0028-10-03" "-0028-10-04" "-0028-10-05" "-0028-10-06"
[281] "-0028-10-07" "-0028-10-08" "-0028-10-09" "-0028-10-10" "-0028-10-11"
[286] "-0028-10-12" "-0028-10-13" "-0028-10-14" "-0028-10-15" "-0028-10-16"
[291] "-0028-10-17" "-0028-10-18" "-0028-10-19" "-0028-10-20" "-0028-10-21"
[296] "-0028-10-22" "-0028-10-23" "-0028-10-24" "-0028-10-25" "-0028-10-26"
[301] "-0028-10-27" "-0028-10-28" "-0028-10-29" "-0028-10-30" "-0028-10-31"
[306] "-0028-11-01" "-0028-11-02" "-0028-11-03" "-0028-11-04" "-0028-11-05"
[311] "-0028-11-06" "-0028-11-07" "-0028-11-08" "-0028-11-09" "-0028-11-10"
[316] "-0028-11-11" "-0028-11-12" "-0028-11-13" "-0028-11-14" "-0028-11-15"
[321] "-0028-11-16" "-0028-11-17" "-0028-11-18" "-0028-11-19" "-0028-11-20"
[326] "-0028-11-21" "-0028-11-22" "-0028-11-23" "-0028-11-24" "-0028-11-25"
[331] "-0028-11-26" "-0028-11-27" "-0028-11-28" "-0028-11-29" "-0028-11-30"
[336] "-0028-12-01" "-0028-12-02" "-0028-12-03" "-0028-12-04" "-0028-12-05"
[341] "-0028-12-06" "-0028-12-07" "-0028-12-08" "-0028-12-09" "-0028-12-10"
[346] "-0028-12-11" "-0028-12-12" "-0028-12-13" "-0028-12-14" "-0028-12-15"
[351] "-0028-12-16" "-0028-12-17" "-0028-12-18" "-0028-12-19" "-0028-12-20"
[356] "-0028-12-21" "-0028-12-22" "-0028-12-23" "-0028-12-24" "-0028-12-25"
[361] "-0028-12-26" "-0028-12-27" "-0028-12-28" "-0028-12-29" "-0028-12-30"
[366] "-0028-12-31"
```

2.4. Contract

The `contract()` function operates as the opposite of `expand()`. It contracts a list of dates into the abbreviated annotation of **messydates**.

```
R> tibble::tibble(contract = contract(expand(dates_expand)))

# A tibble: 7 x 1
  contract
  <messydt>
```

```
1 2008-03-25                                                                      ~
2 2001-01                                                                         ~
3 2001                                                                            ~
4 2001-01-01..2001-02-02                                                          ~
5 {2001-01-01,2001-02-02}                                                         ~
6 {2008-01-31,2008-02-28,2008-03-31,2008-04-30,2008-05-31,2008-06-30,2008-07-31~
7 -0028                                                                           ~
```

2.5. Coerce from messydates

Coercion functions coerce objects of `messydt` class to common date classes such as `Date`, `POSIXct`, and `POSIXlt`. Since `messydt` objects can hold multiple individual dates, an additional function must be passed as an argument so that multiple dates are "resolved" into a single date.

For example, one might wish to use the earliest possible date in any ranges of dates (`min`), the latest possible date (`max`), some notion of a central tendency (`mean`, `median`, or `modal`), or even a `random` selection from amongst the candidate dates.

These functions are particularly useful for use with existing methods and models, especially for checking the robustness of results.

```
R> tibble::tibble(min = as.Date(dates_expand, min),
+                 max = as.Date(dates_expand, max),
+                 median = as.Date(dates_expand, median),
+                 mean = as.Date(dates_expand, mean),
+                 modal = as.Date(dates_expand, modal),
+                 random = as.Date(dates_expand, random))

# A tibble: 7 x 6
  min        max        median     mean       modal      random
  <date>     <date>     <date>     <date>     <date>     <date>
1 2008-03-25 2008-03-25 2008-03-25 2008-03-25 2008-03-25 2008-03-25
2 2001-01-01 2001-01-31 2001-01-16 2001-01-16 2001-01-01 2001-01-05
3 2001-01-01 2001-12-31 2001-07-02 2001-07-02 2001-01-01 2001-02-14
4 2001-01-01 2001-02-02 2001-01-17 2001-01-17 2001-01-01 2001-01-15
5 2001-01-01 2001-02-02 2001-02-02 2001-01-17 2001-01-01 2001-01-01
6 2008-01-31 2008-12-31 2008-07-31 2008-07-15 2008-01-31 2008-04-30
7 -028-01-01 -028-12-31 -028-07-02 -028-07-01 -028-01-01 -028-01-23
```

2.6. Additional functionality

Several other functions are also offered in the **messydates** package.

For example, one can check various logical tests for messy date objects. `is_messydate()` tests whether the object inherits the `messydt` class. `is_intersecting()` tests whether there is any intersection between two messy dates. `is_element()` similarly tests whether a messy date can be found within a messy date range or set. `is_similar()` tests whether two dates contain similar components.

```
R> is_messydate(as_messydate("2012-01-01"))
```

```
[1] TRUE
```

```
R> is_messydate(as.Date("2012-01-01"))
```

```
[1] FALSE
```

```
R> is_intersecting(as_messydate("2012-01"), as_messydate("2012-01-01..2012-02-22"))
```

```
[1] FALSE
```

```
R> is_intersecting(as_messydate("2012-01"), as_messydate("2012-02-01..2012-02-22"))
```

```
[1] FALSE
```

```
R> is_element(as_messydate("2012-01-01"), as_messydate("2012-01"))
```

```
[1] TRUE
```

```
R> is_element(as_messydate("2012-01-01"), as_messydate("2012-02"))
```

```
[1] FALSE
```

```
R> is_similar(as_messydate("2012-06-02"), as_messydate("2012-02-06"))
```

```
[1] TRUE
```

```
R> is_similar(as_messydate("2012-06-22"), as_messydate("2012-02-06"))
```

```
[1] FALSE
```

Additionally, one can perform intersection (`md_intersect()`) and union (`md_union()`) on, inter alia, messy date class objects. Or 'join' that retains all elements, even if duplicated, with `md_multiset`.

```
R> md_intersect(as_messydate("2012-01-01..2012-01-20"),as_messydate("2012-01"))
```

```
 [1] "2012-01-01" "2012-01-02" "2012-01-03" "2012-01-04" "2012-01-05"
 [6] "2012-01-06" "2012-01-07" "2012-01-08" "2012-01-09" "2012-01-10"
[11] "2012-01-11" "2012-01-12" "2012-01-13" "2012-01-14" "2012-01-15"
[16] "2012-01-16" "2012-01-17" "2012-01-18" "2012-01-19" "2012-01-20"
```

```
R> md_union(as_messydate("2012-01-01..2012-01-20"),as_messydate("2012-01"))
```

```
 [1] "2012-01-01" "2012-01-02" "2012-01-03" "2012-01-04" "2012-01-05"
 [6] "2012-01-06" "2012-01-07" "2012-01-08" "2012-01-09" "2012-01-10"
[11] "2012-01-11" "2012-01-12" "2012-01-13" "2012-01-14" "2012-01-15"
[16] "2012-01-16" "2012-01-17" "2012-01-18" "2012-01-19" "2012-01-20"
[21] "2012-01-21" "2012-01-22" "2012-01-23" "2012-01-24" "2012-01-25"
[26] "2012-01-26" "2012-01-27" "2012-01-28" "2012-01-29" "2012-01-30"
[31] "2012-01-31"
```

```
R> md_multiset(as_messydate("2012-01-01..2012-01-20"),as_messydate("2012-01"))
```

```
 [1] "2012-01-01" "2012-01-02" "2012-01-03" "2012-01-04" "2012-01-05"
 [6] "2012-01-06" "2012-01-07" "2012-01-08" "2012-01-09" "2012-01-10"
[11] "2012-01-11" "2012-01-12" "2012-01-13" "2012-01-14" "2012-01-15"
[16] "2012-01-16" "2012-01-17" "2012-01-18" "2012-01-19" "2012-01-20"
[21] "2012-01-01" "2012-01-02" "2012-01-03" "2012-01-04" "2012-01-05"
[26] "2012-01-06" "2012-01-07" "2012-01-08" "2012-01-09" "2012-01-10"
[31] "2012-01-11" "2012-01-12" "2012-01-13" "2012-01-14" "2012-01-15"
[36] "2012-01-16" "2012-01-17" "2012-01-18" "2012-01-19" "2012-01-20"
[41] "2012-01-21" "2012-01-22" "2012-01-23" "2012-01-24" "2012-01-25"
[46] "2012-01-26" "2012-01-27" "2012-01-28" "2012-01-29" "2012-01-30"
[51] "2012-01-31"
```

2.7. Case Study 1 - The size of the population of the Roman Empire

Several historical facts are contested. However, up to this point, dealing with negative historical dates in R was difficult and only manageable if authors made arbitrary choices to complete dates, making them biased, or left these in character class, instead of date class, limiting their analytical usefulness. Take, for example, the population sizes in the Roman Empire, which has been a topic of heated debates among some academics (see Scheidel 2008). Various accounts diverge in how they estimate population size and, therefore, the conclusions about the of the population at certain points in time differs (Scheidel 2008). Below, we build a dataset for the populations of mainland Italy during the Roman Empire (Scheidel 2008, p. 31) to illustrate how **messydates** facilitate the annotation and the conversion of historical dates for analysis.

```
R> roman_population <- tibble::tibble(period = as_messydate(c("200 BC:100 BC", "100 BC:1 A
+                                  population_min = c(3900000, 4400000, 5500000, 6000000
+                                  population_max = c(8000000, 10000000, 16000000, 20000
R> roman_population_md <- roman_population %>%
+   dplyr::mutate(period_min = as.Date(period, min),
+                 period_max = as.Date(period, max),
+                 period_mean = as.Date(period, mean),
+                 period_median = as.Date(period, median),
+                 period_random = as.Date(period, random)) %>%
+   print()
```

```
# A tibble: 4 x 8
  period        population_min population_max period_min period_max period_mean
  <messydt>              <dbl>          <dbl> <date>     <date>     <date>
1 -0200..-0100         3900000        8000000 -100-01-01 -200-12-31 -150-07-02
2 -0100..0001          4400000       10000000 -001-01-01 -100-12-31 -050-12-31
3 0001..0150           5500000       16000000 0001-01-01 0150-12-31 0075-12-31
4 0150..0200           6000000        2000000 0150-01-01 0200-12-31 0175-07-02
# ... with 2 more variables: period_median <date>, period_random <date>
```

2.8. Case Study 2 - Roman Emperors Rise to Power: Inference and robustness checks with "messy" historical dates

Getting the timing can be important for researchers, however, when faced with date imprecision, researchers usually have to chose between making arbitrary choices (e.g. adding "-01-01" to all incomplete dates) or work imprecise dates (i.e. year only). Yet, both choices may lead to bias results. This is especially true if researchers' are looking to generate inferences where getting "timing" right is important. Assume, for example, certain researcher is interested in the relationship between how Roman Emperors rose to power and how long they stayed in powe. The researcher theorizes that Roman Emperors that rose to power through birthrights were less contested politically and popularly and, thus, tended to stay in power for longer than those Roman Emperors that rose to power in other ways (e.g. coup or appointment). The **manydata** contains a database on Roman Emperors composed of three datasets. We can also use **manydata** to "consolidate" the database into a single dataset using some combination of the rows and columns available.

Below, we use the `consolidate()` function to transform the Emperors database into a single dataset by, first, favoring the United Nations of Rome Victrix (UNVR) dataset over others in the database. Second, we select all rows that are present in "every" one of the data sets (i.e. rows present in all the datasets only) and columns that are present in "any" of the datasets in the database (i.e. rows present in at least one of the datasets). Third, we resolve the conflicts between observations using the "coalesce" method that takes the first non-NA value. Finally, we select "ID" as the key variable to which observations will be matched. Then, we use **dplyr** in combination with **messydates** to create three different date variables to be our dependent variables: imprecise (year only), arbitrary (the minimal value for a data range), and random (a random value within the date range). With these variables in hand, we build three simple linear regression models correlating each of the dependent variables to the variable on how Roman Emperors rose to power. We use **stargazer** to visualize and compare the regression results.

```
R> library(manydata)
R> library(dplyr)
R> library(stargazer)
```

```
Please cite as:

 Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
 R package version 5.2.3. https://CRAN.R-project.org/package=stargazer

R> emperors <- manydata::consolidate(database = manydata::favour(manydata::emperors, "UNRV
+                                    rows = "every",
+                                    cols = "any",
+                                    resolve = "coalesce",
+                                    key = "ID") %>%
+   select(c(ID, Rise, Birth, Beg, End)) %>%
+   mutate(years_only = as.numeric(End) - as.numeric(Beg),
+          arbitrary = md_duration(Beg, End, "min"),
+          random = md_duration(Beg, End, "random"))
R> year_only <- lm(years_only ~ Rise , emperors)
R> arbitrary <- lm(arbitrary ~ Rise, emperors)
R> random <- lm(random ~ Rise, emperors)
R> stargazer(year_only, arbitrary, random, type = "text")


======================================================================
                                             Dependent variable:
                                    ----------------------------------
                                    years_only  arbitrary    random
                                       (1)         (2)         (3)
----------------------------------------------------------------------
RiseAppointment by Emperor            2.200      803.200     768.400
                                     (9.779)   (3,571.848) (3,600.736)

RiseAppointment by Praetorian Guard  -3.800   -1,387.800  -1,129.600
                                     (9.779)   (3,571.848) (3,600.736)

RiseAppointment by Senate            -2.300    -839.967    -830.433
                                     (5.406)   (1,974.413) (1,990.382)

RiseBirthright                       9.064**   3,310.473** 3,336.127**
                                     (4.423)   (1,615.429) (1,628.494)

RiseElection                          7.200     2,630.200   2,620.400
                                     (9.779)   (3,571.848) (3,600.736)

RiseSeized Power                      3.000     1,095.800   1,131.400
                                     (5.646)   (2,062.207) (2,078.886)

Constant                              3.800     1,387.800   1,339.600
                                     (3.992)   (1,458.201) (1,469.994)

----------------------------------------------------------------------
Observations                           41          41          41
R2                                    0.253       0.253       0.249
```

```
Adjusted R2                            0.121       0.121       0.117
Residual Std. Error (df = 34)          8.927     3,260.636   3,287.007
F Statistic (df = 6; 34)               1.920       1.920       1.882
======================================================================
Note:                                       *p<0.1; **p<0.05; ***p<0.01
```

Notice how the regression coefficients change when we take the year only of dates, add an arbitrary cut off point to dates, or get random dates from the range. However, in this case, we can say that the relationship between Roman Emperors rising to power through birthrights and time in power is both positive and statistically significant across the three cases; therefore, more robust than otherwise.

3. Acknowledgements

We would like to thank the Swiss National Science Foundation. This work was supported by grant number 188976.

References

Eddelbuettel D (2019). "Anytime: Easier Date and Time Conversion." p. 4.

Grolemund G, Wickham H (2011). "Dates and Times Made Easy with Lubridate." **40**(3), 1–25. ISSN 1548-7660. doi:10.18637/jss.v040.i03. URL http://www.jstatsoft.org/v40/i03/.

Affiliation:

James Hollway
Graduate Institute of
International and Development Studies
Chemin Eugène-Rigot 2A
PO Box 1672
1211 Geneva 1
Switzerland
E-mail: james.hollway@graduateinstitute.ch
URL: http://jameshollway.com