Extra functionality for `rjd3modelling`

Mārtiņš Liberts (Central Statistical Bureau of Latvia)

2021-10-31 22:12:35

Contents

1 Holidays in Latvia

There are specific holidays in Latvia which are not possible to describe with `rjd3modelling` functions currently. The following extra functionality would be necessary.

1.1 Holidays with compensation

There are holidays with a so called compensation mechanism. If a holiday in question is on Saturday or Sunday, then the next working day is also a holiday.

1.1.1 May 4

May 4 (Restoration of Independence) is a public holiday since 2002. The compensation mechanism for the May 4 was introduced since 2008. If May 4 is on a Saturday or Sunday than the next working day is also a holiday. The next working day is always Monday as there is no clash with other holiday currently.

- If May 4 is on a weekday from Monday to Friday, only May 4 is a holiday.
- If May 4 is on Saturday, than May 4 is a holiday and also May 6 (the next Monday) is a holiday.
- If May 4 is on Sunday, than May 4 is a holiday and also May 5 (the next Monday) is a holiday.

Alternative description of the compensation mechanism:

- May 4 is always a holiday.
- May 5 is a holiday if May 5 is on Monday.
- May 6 is a holiday if May 6 is on Monday.

There have been four cases so far when May 4 has been compensated with an extra holiday. The extra holidays were:

- 2008-05-05
- 2013-05-06
- 2014-05-05
- 2019-05-06

1.1.2 November 18

November 18 (Proclamation Day of the Republic of Latvia) is a public holiday since 1990. The compensation mechanism for the November 18 was introduced since 2007. If November 18 is on a Saturday or Sunday than the next working day is also a holiday. The next working day is always Monday as there is no clash with other holiday currently.

- If November 18 is on a weekday from Monday to Friday, only November 18 is a holiday.
- If November 18 is on Saturday, than November 18 is a holiday and also November 20 (the next Monday) is a holiday.
- If November 18 is on Sunday, than November 18 is a holiday and also November 19 (the next Monday) is a holiday.

Alternative description of the compensation mechanism:

- November 18 is always a holiday.
- November 19 is a holiday if November 19 is on Monday.
- November 20 is a holiday if November 20 is on Monday.

There have been four cases so far when November 18 has been compensated with an extra holiday. The extra holidays were:

- 2007-11-19
- 2012-11-19
- 2017-11-20
- 2018-11-19

1.1.3 Implementation in `rjd3modelling`

Extra arguments for the function `calendar.fixedday` would be necessary to implement the compensation mechanism from the user interface. Two extra arguments for the function `calendar.fixedday` similar to arguments from the function `holidays` would be necessary:

- `nonworking = c(6, 7)`: Indexes of non working days (Monday=1, Sunday=7).
- `compensation = c("None", "NextWorkingDay", "PreviousWorkingDay")`

So, the description of the `calendar.fixedday` would be:

```
calendar.fixedday(
  calendar,
  month,
  day,
  weight = 1,
  start = NULL,
  end = NULL,
  nonworking = c(6, 7),
  compensation = c("None", "NextWorkingDay", "PreviousWorkingDay")
)
```

The May 4 would be defined as:

```
calendar.fixedday(
  calendar = cal_lv,
  month = 5,
  day   = 4,
  weight = 1,
  start = "2002-01-01",
  end   = "2007-12-31",
  nonworking = c(6, 7),
  compensation = "None"
)
calendar.fixedday(
  calendar = cal_lv,
  month = 5,
  day   = 4,
  weight = 1,
  start = "2008-01-01",
  end   = NULL,
  nonworking = c(6, 7),
  compensation = "NextWorkingDay"
)
```

The November 18 would be defined as:

```
calendar.fixedday(
  calendar = cal_lv,
  month = 11,
  day   = 18,
  weight = 1,
  start = NULL,
  end   = "2006-12-31",
  nonworking = c(6, 7),
  compensation = "None"
)
calendar.fixedday(
  calendar = cal_lv,
  month = 11,
  day   = 18,
  weight = 1,
  start = "2007-01-01",
  end   = NULL,
  nonworking = c(6, 7),
  compensation = "NextWorkingDay"
)
```

1.2 A holiday once in five years

We have a very recent holiday which will take place only once in a five years. General Latvian Song and Dance Celebration happens **once in 5 years**. The closing day of the celebration is a holiday since 2018. Traditionally the the closing day of the celebration is on Sunday of the first week of July. To make it more complicated - this is another holiday with a compensation. The next working day is also a holiday. So there are two holidays - Sunday of the 1st week of July and Monday of the 2nd week of July.

It has been a holiday only once so far. The July 8 (Sunday) and 9 (Monday) in 2018 were holidays for the first time because of the General Latvian Song and Dance Celebration. The next celebration will be in 2023. So, July 9 and 10 on 2023 should be holidays.

The 5 years cycle of the General Latvian Song and Dance Celebration is defined by the law. The date of the closing day is not defined by the law. So, it can happen that other date of the closing day is chosen. It has been almost always on Sunday of the 1st week of July since 1993. However, there was one exception. The closing day of the celebration in 2008 was on Saturday of the 2nd week of July (it was 2008-07-12). We will not be able to model it as a continuous holiday if the tradition will be broken once again.

The list closing days of the General Latvian Song and Dance Celebration since 1993:

- 1993-07-04
- 1998-07-05

- 2003-07-06
- 2008-07-12 (the exception - it was on Saturday of the 2nd week of July)
- 2013-07-07
- 2018-07-08
- *2023-07-09* (this is assumption, it has not been defined yet)

1.2.1 Implementation in `rjd3modelling`

Extra arguments for the function `calendar.fixedweekday` would be necessary to implement this holiday with a compensation mechanism from the user interface. Three extra arguments for the function `calendar.fixedweekday` would be necessary:

- `cycle = 1`: Cycle in years (by default one year).
- `nonworking = c(6, 7)`: Indexes of non working days (Monday=1, Sunday=7).
- `compensation = c("None", "NextWorkingDay", "PreviousWorkingDay")`

So, the description of the `calendar.fixedweekday` would be:

```
calendar.fixedweekday(
  calendar,
  month,
  week,
  dayofweek,
  cycle = 1,
  weight = 1,
  start = NULL,
  end = NULL,
  nonworking = c(6, 7),
  compensation = c("None", "NextWorkingDay", "PreviousWorkingDay")
)
```

The closing day of the General Latvian Song and Dance Celebration would be defined as:

```
calendar.fixedweekday(
  calendar = cal_lv,
  month = 7,
  week = 1,
  dayofweek = 7,
  cycle = 5,
  weight = 1,
  start = "2018-01-01",
  end = NULL,
  nonworking = c(6, 7),
  compensation = "NextWorkingDay"
)
```