Package ‘sweidnumbr’

January 13, 2024

Type Package
Title Handling of Swedish Identity Numbers
Version 1.5.0
Date 2024-01-13
Author Mans Magnusson and Erik Bulow
Maintainer Mans Magnusson <mons.magnusson@gmail.com>
Description Structural handling of identity numbers used in the Swedish administration such as personal identity numbers (‘personnummer’) and organizational identity numbers (‘organisationsnummer’).
VignetteBuilder rmarkdown, knitr
URL https://github.com/rOpenGov/sweidnumbr,
http://ropengov.github.io/sweidnumbr/
BugReports https://github.com/rOpenGov/sweidnumbr/issues
License BSD_2_clause + file LICENSE
Depends R (>= 3.2)
Encoding UTF-8
Imports lubridate (>= 1.5), stringr, checkmate
Suggests testthat (>= 1.0.0), rmarkdown, knitr
LazyData true
RoxygenNote 7.2.3
X-schema.org-isPartOf http://ropengov.org/
X-schema.org-keywords ropengov
NeedsCompilation no
Repository CRAN
Date/Publication 2024-01-13 20:20:02 UTC
### R topics documented:

as.oin ......................................................... 2
as.pin .......................................................... 3
fake_pins ....................................................... 4
format_pin ...................................................... 5
is.oin ........................................................... 6
luhn_algo ....................................................... 6
oin_ctrl ......................................................... 7
oin_group ....................................................... 8
pin_age .......................................................... 9
pin_birthplace ............................................... 10
pin_coordn ..................................................... 11
pin_ctrl ......................................................... 11
pin_date ........................................................ 12
pin_sex .......................................................... 13
roin ............................................................. 13
rpin .............................................................. 14
sweidnumbr .................................................... 15

Index 16

| as.oin | Parse organizational identity numbers |

### Description

Check and convert a vector of organizational identity numbers.

### Usage

as.oin(oin)

### Arguments

- **oin** Vector with swedish organizational identity numbers in character format. See details.

### Details

The following format is accepted:

- character: GNNNNN–NNNC

### Value

Character vector (of class oin and AsIs) with swedish organizational identity numbers.
as.pin

References
Lag (1974:174) om identitetsbeteckning för juridiska personer m.fl.

Examples
ex_oin <- c("556000-4615", "232100-0156", "802002-4280", "8020024280", "AA2002-4280")
as.oin(ex_oin)

---

as.pin

Parse personal identity numbers to ABS format

Description
as.pin Converts personal identity numbers of different formats to standard (ABS) pin format YYYYMMDDNNNC where YYYYMMDD is the date of birth, NNN is the birth number and C is the control number. is.pin checks whether an R object is of class "pin".

Usage
as.pin(pin)
is.pin(pin)

Arguments
pin Vector with swedish personal identity numbers in character or numeric format. See details.

Details
as.pin converts different formats of swedish personal identity numbers to the standard ABS format. The formats that can be converted are:
- numeric: YYYYMMDDNNNC
- numeric: YYYYMMDDNNNC (assuming < 100 years of age)
- character: "YYYYMMDDNNNC"
- character: "YYYYMMDD-NNNC", "YYYYMMDD+NNNC"
- character: "YYYYMMDD-NNNC" (assuming < 100 years of age)
(where "C" can be substituted by characters "A", "T" or "X" if "YYYY" < 1967).

Value
as.pin returns a vector of class "pin" (with additional classes "AsIs" and character) with swedish personal identity numbers with standard ABS format "YYYYMMDDNNNC". is.pin returns TRUE if pin is of class "pin", otherwise FALSE.
References

- Skatteverket, Personnummer, SKV 704 (2007)

Examples

# Examples taken from SKV 704 (see references)
ex_pin1 <- c("196408233234", "640823-3234", "19640823-3234")
as.pin(pin = ex_pin1)
ex_pin2 <- c("6408233234")
as.pin(ex_pin2)
ex_pin3 <- c(6408233234, 196408233234)
as.pin(ex_pin3)
ex_pin4 <- rep(c("20121209-0122", "201212090122", "121209-0122", "1212090122"),250)
as.pin(ex_pin4)
ex_pin5 <- c("205012090122", "186512090122", "121209-0122", "121209-012A")
as.pin(pin = ex_pin5)
pin <- c("201212090122", "201212090122", "121209-0122", "1212090122")
ex_pin <- rpin(3)
is.pin(ex_pin)

is.pin(ex_pin)

ex_pin_char <- as.character(ex_pin)
is.pin(ex_pin_char)

---

fake_pins  
Fake personal identity numbers and names

Description

Data set with fake personal identity numbers and names to use as example.

Format

A data frame with 62 rows and 2 variables:

- **pin**  Personal identification number, as character
- **name**  Fictional Swedish names
format_pin

Formatting pin

Description

Format pin for pretty printing

Usage

format_pin(x, format. = "%Y%m%d%N", ...)

Arguments

- **x**: vector of class "pin" (see as.pin) or a vector that can be coerced to such
- **format.**: character string specifying the output format. %N is used as a reference for the last four digits of the pin. Format of the date is handled via strftime. ("%Y%m%d%N" by default). %P is an available shorthand for "(%C) %y-%m-%d-%N", a format aimed for maximal readability when used in long lists
- ... arguments passed to format.Date

Value

character vector of same length as x

Examples

```r
x <- as.pin(fake_pins$pin[1:10])

# Separate elements with hyphens:
format_pin(x, "%Y-%m-%d-%N")

# Separate even further
format_pin(x, "%C-%y-%m-%d-%N")

# The special P-format for maximal readability
format_pin(x, "%P")

# A custom representation
format_pin(x, "Borned %d of %B in %Y (a %A in week %U) with suffix no: %N")

# Extract only the year
format_pin(x, "%Y")
```
is.oin  

*Test if a character vector contains correct oin*

### Description
Test which elements in a text vector that contains organization identity number.

### Usage

```r
is.oin(oin)
```

### Arguments

- `oin`  
  Character vector to be tested if it is an oin of the right format.

### Value
Logical vector indicating if the elements can be an organization identity number.

### Examples

```r
ex_oin <- roin(3)
is.oin(ex_oin)

ex_oin_char <- as.character(ex_oin)
is.oin(ex_oin_char)
```

---

luhn_algo  

*The Luhn algorithm*

### Description
Calculates the control number for a Swedish personal/organisational identity number using the Luhn algorithm.

### Usage

```r
luhn_algo(id, multiplier)
```

### Arguments

- `id`  
  Element with swedish personal identity number.
- `multiplier`  
  What should each element in id be multiplied with
**oan_ctrl**

**Value**

The control number (last digit in the personal identification number) calculated from id (as integer).

**References**

- Luhn Algorithm.

**Examples**

```r
luhn_algo("121212121212", c(0,0,2,1,2,1,2,1,2,1,2,0))
luhn_algo( "121212121", c( 2,1,2,1,2,1,2,1,2))
```

```r
## If no multiplier, the default is
## to find one that match the format of id
luhn_algo("1212121212")
luhn_algo("12121212121")
luhn_algo("121212121")
luhn_algo("1212121211")
```

```r
## Also for multiple pin
## (as long they are all of the same format)
luhn_algo(c("121212121", "19850504333"))
```

```r
## Not run:
try(luhn_algo(c("121212121", "850504333"))) # Different formats should fail!
```

## End(Not run)

---

**oan_ctrl**  
*Check the control numbers for oan*

**Description**

Calculates the control number using the Luhn algorithm and compare it with the control number in the organization identity number (oan).

**Usage**

`oan_ctrl(oan, force_logical = FALSE)`

**Arguments**

- `oan` A vector of class oan. See `as.oan`.
- `force_logical` If TRUE, force all NA in oan to be FALSE. Default is FALSE.
**oin_group**

**Value**

Logical vector indicating if a oin is correct (TRUE) or not (FALSE)

**References**

Organisationsnummer Skatteverket

**Examples**

```r
ex_oin <- c("556000-4615", "232100-0156", "802002-4280", "232100-0157", "802002-4281")
oin_ctrl(ex_oin)
```

---

**oin_group**  
*Calculate organization group from oin*

**Description**

Calculates the organization group from the organization number.

**Usage**

```r
oin_group(oin)
```

**Arguments**

- `oin`  
  A vector of class `oin`. See `as.oin`.

**Value**

Factor with organization categories.

**References**

Organisationsnummer Skatteverket

**Examples**

```r
ex_oin <- c("556000-4615", "232100-0156", "802002-4280")
oin_group(ex_oin)
```
**pin_age**  

*Calculate age of pin for a given date*

**Description**

Calculate the age in full years for a given date.

**Usage**

```r
pin_age(pin, date = Sys.Date(), timespan = "years", verbose = TRUE)
```

**Arguments**

- **pin**: A vector of class pin. See `as.pin`
- **date**: Date at which age is calculated. If a vector is provided it must be of the same length as the `pin` argument.
- **timespan**: Timespan to use to calculate age. The actual timespans are:
  - years (Default)
  - months
  - weeks
  - days
- **verbose**: Should messages be printed? Default is `TRUE`.

**Value**

Age as an integer vector.

**References**


**Examples**

```r
# Example with someone born today
today_pin <-
paste(paste(unlist(strsplit(as.character(Sys.Date()),split = "-")), collapse = ""),
  "0000",sep="")
pin_age(today_pin)

# Examples taken from SKV 704 (see references)
ex_pin <- c("196408233234", "186408833224")
pin_age(ex_pin, date = "2012-01-01")
```
Description

Calculate the birthplace for a given personal identity number born before 1990. See details.

Usage

```
pin_birthplace(pin)
```

Arguments

- **pin**: A vector of class `pin`. See `as.pin`.

Details

It is possible to calculate where people were born (and/or if a person has immigrated) through their personal identity number. This is possible for people that was born before 1990 and after 1945.

For people born before 1946 the birthplace identifier contains information on where one was registered the 1st of November 1946.

Personal identity numbers for people born after 1989 do not contain any information on birthplace.

During the period 1946 - 1989 the pin also contains information on whether one has immigrated to Sweden during the period.

Value

Birthplace as factor.

References

SOU 2008:60: Personnummer och samordningsnummer

Examples

```
# Example with someone born today and from SKV 704 (see references)
today_pin <- paste0(format(Sys.Date(), "%Y%m%d"), "0000")
ex_pin <- c("19640823234", today_pin)
pin_birthplace(ex_pin)
```
**pin_coordin**

*Check if *pin* is a coordination number*

**Description**

Calculate if the personal identity number is a coordination number.

**Usage**

```
pin_coordin(pin)
```

**Arguments**

- `pin` A vector of class `pin`. See `as.pin`.

**Value**

Logical vector indicating if the pin is a coordination number (TRUE) or pin (FALSE).

**References**


**Examples**

```
# Examples taken from SKV 704 (see references)
ex_pin <- c("196408233234", "196408833224")
pin_coordin(ex_pin)
```

**pin_ctrl**

*Check control number from *pin*

**Description**

Calculates the control number using the Luhn algorithm and compare it with the control number in the personal identity number.

**Usage**

```
pin_ctrl(pin, force_logical = FALSE)
```
Arguments

pin A vector of class pin. See as.pin.
force_logical If TRUE, force all NA in pin to be FALSE. Default is FALSE.

Value

Logical vector indicating if a pin is correct (TRUE) or not (FALSE)

References


Examples

# Examples taken from SKV 704 (see references)
ex_pin <- c("196408233234", "196408233235")
pin_ctrl(ex_pin)

Description

Calculate the date of birth from a pin

Usage

pin_date(pin)

Arguments

pin A vector of class pin. See as.pin.

Value

Date of birth as a vector in date format.

Examples

# Examples taken from SKV 704 (see references)
ex_pin <- c("196408233234", "186408883224")
pin_date(ex_pin)
**pin_sex**  
*Calculate sex from pin*

**Description**  
Calculates the sex from the personal identification number.

**Usage**  
`pin_sex(pin)`

**Arguments**  
- **pin**: A vector of class `pin`. See `as.pin`.

**Value**  
Factor with label 'Male' and 'Female'.

**References**

**Examples**

```r
# Examples taken from SKV 704 (see references)
ex_pin <- c("196408233234", "186408233224")
pin_sex(ex_pin)
```

---

**roin**  
*Generate a vector of random oin*

**Description**  
A function that generates random oins (see `as.pin`). The generated oin is uniformly distributed over all possible oins.

**Usage**  
`roin(n)`
Arguments

- **n**: number of observations. If `length(n) > 1`, the length is taken to be the number required.

Value

- a vector of generated oins.

Examples

```r
x <- roin(3)
oin_ctrl(x)
oin_group(x)
```

---

**rpin**

*Generate a vector of random pin*

Description

A function that generates random pins (see `as.pin`). The generated pin is uniformly distributed over the time period.

Usage

```r
rpin(
  n,
  start_date = "1900-01-01",
  end_date = Sys.Date(),
  p.male = 0.5,
  p.coordn = 0.1
)
```

Arguments

- **n**: number of observations. If `length(n) > 1`, the length is taken to be the number required.
- **start_date**: Smallest possible pin. Default is 1900-01-01.
- **end_date**: Largest possible pin. Default is the current date.
- **p.male**: Proportion of males. Default is 0.5.
- **p.coordn**: Proportion of coordination numbers. Default is 0.1.

Value

- a vector of generated pins.
Examples

```r
x <- rpin(3)
pin_ctrl(x)
pin_sex(x)
pin_age(x)
```

Description

Handling of swedish identity numbers. For a quick tutorial see `vignette("sweidnumbr")`. For more information see [https://github.com/rOpenGov/sweidnumbr](https://github.com/rOpenGov/sweidnumbr).
Index

as.oin, 2, 7, 8
as.pin, 3, 5, 9–14

fake_pins, 4
format.Date, 5
format_pin, 5

is.oin, 6
is.pin(as.pin), 3

luhn_algo, 6
ooin_ctrl, 7
ooin_group, 8

pin_age, 9
pin_birthplace, 10
pin_coordn, 11
pin_ctrl, 11
pin_date, 12
pin_sex, 13

roin, 13
rpin, 14

strptime, 5
sweidnumbr, 15