

# Package ‘hetu’

October 13, 2022

**Type** Package

**Title** Structural Handling of Finnish Personal Identity Codes

**Version** 1.0.7

**Date** 2022-05-20

**MailingList** rOpenGov <ropengov-forum@googlegroups.com>

**Description** Structural handling of Finnish identity codes (natural persons and organizations); extract information, check ID validity and diagnostics.

**License** BSD\_2\_clause + file LICENSE

**VignetteBuilder** knitr

**Encoding** UTF-8

**BugReports** <https://github.com/ropengov/hetu/issues>

**URL** <https://ropengov.github.io/hetu/>, <https://github.com/ropengov/hetu>

**Depends** R (>= 3.6.0)

**Imports** lubridate, checkmate, parallel

**Suggests** Cairo, knitr, testthat, rmarkdown, covr, dplyr

**RoxygenNote** 7.2.0

**X-schema.org-isPartOf** <http://ropengov.org/>

**X-schema.org-keywords** ropengov

**Config/Needs/website** magick, ropengov/rogtemplate

**NeedsCompilation** no

**Author** Pyry Kantanen [aut, cre] (<<https://orcid.org/0000-0003-2853-2765>>),  
Mans Magnusson [aut],  
Jussi Paananen [aut],  
Juho Kopra [ctb],  
Oskari Luomala [ctb],  
Tuomo Nieminen [ctb],  
Leo Lahti [aut] (<<https://orcid.org/0000-0001-5537-637X>>)

**Maintainer** Pyry Kantanen <pyry.kantanen@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-05-21 23:10:11 UTC

**R topics documented:**

bid_ctrl	2
hetu	3
hetu_control_char	4
hetu_diagnostic	5
pin_age	6
pin_ctrl	7
pin_date	8
pin_sex	8
rbid	9
rpip	10
satu_control_char	11
satu_ctrl	12

<b>Index</b>	<b>13</b>
--------------	-----------

---

bid_ctrl	<i>Check Validity of Finnish Business ID (Y-tunnus)</i>
----------	---

---

**Description**

A function that checks whether a bid (Finnish Business ID) is valid. Returns TRUE or FALSE.

**Usage**

```
bid_ctrl(bid)
```

**Arguments**

bid                    a vector of 1 or more business identity numbers

**Examples**

```
bid_ctrl(c("0000000-0", "0000001-9")) # TRUE TRUE
bid_ctrl("0737546-1") # FALSE
```

---

 hetu

*Generic Extraction Tool for Finnish Personal Identity Codes*


---

### Description

Extract embedded information from Finnish personal identity codes (hetu).

### Usage

```
hetu(pin, extract = NULL, allow.temp = FALSE, diagnostic = FALSE)
```

### Arguments

pin	Finnish personal identity code(s) as a character vector
extract	Extract only selected part of the information. Valid values are "hetu", "sex", "p.num", "ctrl.char", "date", "day", "month", "year", "century", "is.temp". If NULL (default), returns all information.
allow.temp	Allow artificial or temporary PINs (personal numbers 900-999). If FALSE (default), only PINs intended for official use (personal numbers 002-899) are allowed.
diagnostic	Print additional information about possible problems in PINs. The checks are "valid.p.num", "valid.ctrl.char", "correct.ctrl.char", "valid.date", "valid.day", "valid.month", "valid.length", "valid.century". Default is FALSE which returns no diagnostic information.

### Value

Finnish personal identity code data.frame, or if extract parameter is set, the requested part of the information as a vector. Returns an error or NA if the given character vector is not a valid Finnish personal identity code.

hetu	Finnish personal identity code as a character vector. A correct pin should be in the form DDMMYYCZZZQ, where DDMMYY stands for date, C for century sign, ZZZ for personal number and Q for control character.
sex	sex of the person as a character vector ("Male" or "Female").
p.num	Personal number part of the identity code.
ctrl.char	Control character for the personal identity code.
date	Birthdate.
day	Day of the birthdate.
month	Month of the birthdate.
year	Year of the birthdate.
century	Century character of the birthdate: + (1800), - (1900) or A (2000).
valid.pin	Does the personal identity code pass all validity checks: (TRUE or FALSE)

**Author(s)**

Pyry Kantanen, Jussi Paananen

**See Also**

[pin\\_ctrl](#) For validating Finnish personal identity codes.

**Examples**

```

hetu("111111-111C")
hetu("111111-111C")$date
hetu("111111-111C")$sex
# Same as previous, but using extract argument
hetu("111111-111C", extract="sex")
# Process a vector of hetu's
hetu(c("010101-0101", "111111-111C"))
# Process a vector of hetu's and extract sex information from each
hetu(c("010101-0101", "111111-111C"), extract="sex")

```

---

hetu_control_char	<i>Calculate Control Character for Personal Identity Code</i>
-------------------	---

---

**Description**

Calculate a valid control character for an incomplete Finnish personal identity codes (hetu).

**Usage**

```
hetu_control_char(pin, with.century = TRUE)
```

**Arguments**

pin	An incomplete PIN that ONLY has a date, century marker (optional, see parameter with.century) and personal number
with.century	If TRUE (default), the function assumes that the PIN input contains a century marker (DDMMYYQZZZ). If FALSE, the function assumes that the PIN contains only date and personal number (DDMMYYZZZ).

**Details**

This method of calculating the control character was devised by mathematician Erkki Pale (1962) to detect input errors but also to detect errors produced by early punch card machines. The long number produced by writing the birth date and the personal number together are divided by 31 and the remainder is used to look up the control character from a separate table containing alphanumeric characters except letters G, I, O, Q and Z.

The method of calculating the control character does not need century character and therefore the function has an option to omit it.

**Value**

Control character, either a number 0-9 or a letter.

**Author(s)**

Pyry Kantanen

**See Also**

[hetu](#) For extracting information from Finnish personal identity codes.

**Examples**

```
hetu_control_char("010101-010")  
hetu_control_char("010101010", with.century = FALSE)
```

---

hetu\_diagnostic

*Diagnostics Tool for Personal Identity Codes*

---

**Description**

Prints information on the tests that are used to confirm or reject the validity of each personal identity code.

**Usage**

```
hetu_diagnostic(pin, extract = NULL)
```

```
pin_diagnostic(pin, extract = NULL)
```

**Arguments**

pin	Finnish personal identification number as a character vector, or vector of identification numbers as a character vectors
extract	Extract only selected part of the diagnostic information. Valid values are "hetu", "is.temp", "valid.p.num", "valid.ctrl.char", "correct.ctrl.char", "valid.date", "valid.day", "valid.month", "valid.length", "valid.century". If NULL (default), returns all information.

**Value**

A data.frame containing diagnostic checks about PINs.

**See Also**

[hetu](#) for the main function on which hetu\_diagnostic relies on.

**Examples**

```

diagnosis_example <- c("010101-0102", "111111-111Q",
"010101B0101", "320101-0101", "011301-0101",
"010101-01010", "010101-0011")
## Print all diagnostics for various fake personal identity codes
hetu_diagnostic(diagnosis_example)
# Extract century-related checks
hetu_diagnostic(diagnosis_example, extract = "valid.century")
diagnosis_example <- c("010101-0102", "111111-111Q",
"010101B0101", "320101-0101", "011301-0101",
"010101-01010", "010101-0011")
## Print all diagnoses
pin_diagnostic(diagnosis_example)

```

---

pin\_age

---

*Extract Age from Personal Identity Code*


---

**Description**

Calculate age in years, months, weeks or days from personal identity codes.

**Usage**

```
pin_age(pin, date = Sys.Date(), timespan = "years", allow.temp = FALSE)
```

```
hetu_age(pin, date = Sys.Date(), timespan = "years", allow.temp = FALSE)
```

**Arguments**

pin	Finnish personal identity code(s) as a character vector
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 possible timespans are: <ul style="list-style-type: none"> <li>• years (Default)</li> <li>• months</li> <li>• weeks</li> <li>• days</li> </ul>
allow.temp	Allow artificial or temporary PINs (personal numbers 900-999). If FALSE (default), only PINs intended for official use (personal numbers 002-899) are allowed.

**Value**

Age as an integer vector.

**Examples**

```
ex_pin <- c("010101-0101", "111111-111C")
pin_age(ex_pin, date = "2012-01-01")
```

```
ex_pin <- c("010101-0101", "111111-111C")
hetu_age(ex_pin, date = "2012-01-01")
```

---

pin_ctrl	<i>Check Validity of Personal Identity Code</i>
----------	---

---

**Description**

Validate Finnish personal identity codes (hetu).

**Usage**

```
pin_ctrl(pin, allow.temp = FALSE)
```

```
hetu_ctrl(pin, allow.temp = FALSE)
```

**Arguments**

pin	Finnish personal identity code(s) as a character vector
allow.temp	If TRUE, temporary PINs (personal numbers 900-999) are handled similarly to regular PINs (personal numbers 002-899), meaning that otherwise valid temporary PIN will return a TRUE. Default is FALSE.

**Value**

A logical vector indicating whether the input vector contains valid Finnish personal identity codes.

**Author(s)**

Pyry Kantanen

**See Also**

[hetu](#) For extracting information from Finnish personal identity codes.

**Examples**

```
pin_ctrl("010101-0101") # TRUE
pin_ctrl("010101-010A") # FALSE
pin_ctrl(c("010101-0101", "010101-010A")) # TRUE FALSE
hetu_ctrl("010101-0101") # TRUE
hetu_ctrl("010101-010A") # FALSE
hetu_ctrl(c("010101-0101", "010101-010A")) # TRUE FALSE
```

---

pin_date	<i>Extract Date of Birth from Personal Identity Code</i>
----------	--

---

**Description**

Returns the date of birth in date format.

**Usage**

```
pin_date(pin, allow.temp = FALSE)
```

```
hetu_date(pin, allow.temp = FALSE)
```

**Arguments**

pin	Finnish personal identity code(s) as a character vector
allow.temp	Allow artificial or temporary PINs (personal numbers 900-999). If FALSE (default), only PINs intended for official use (personal numbers 002-899) are allowed.

**Value**

Date of birth as a vector in date format.

**Examples**

```
pin_date(c("010101-0101", "111111-111C"))
```

```
hetu_date(c("010101-0101", "111111-111C"))
```

---

pin_sex	<i>Extract Sex from Personal Identity Code</i>
---------	--

---

**Description**

Extract sex (as binary) from Finnish personal identification code.

**Usage**

```
pin_sex(pin, allow.temp = TRUE)
```

```
hetu_sex(pin, allow.temp = TRUE)
```



**Arguments**

`pin` Finnish personal identity code(s) as a character vector

`allow.temp` Allow artificial or temporary PINs (personal numbers 900-999). If FALSE (default), only PINs intended for official use (personal numbers 002-899) are allowed.

**Value**

Factor with label 'Male' and 'Female'.

**Author(s)**

Pyry Kantanen, Leo Lahti

**See Also**

[hetu](#) For general information extraction

**Examples**

```
pin_sex("010101-010A")
hetu_sex("010101-010A")
```

---

rbid

*Generate Random Finnish Business ID's (Y-tunnus)*

---

**Description**

A function that generates random Finnish Business ID's, bid-numbers (Y-tunnus).

**Usage**

```
rbid(n)
```

**Arguments**

`n` number of generated BIDs

**Value**

a vector of generated BID-numbers.

**Examples**

```
x <- rbid(3)
bid_ctrl(x)
```

---

`rpin`*Generate Random Personal Identity Codes*

---

### Description

A function that generates random Finnish personal identity codes (hetu codes).

### Usage

```
rpin(  
  n,  
  start.date = as.Date("1895-01-01"),  
  end.date = Sys.Date(),  
  p.male = 0.4,  
  p.temp = 0,  
  num.cores = 1  
)
```

```
rhetu(  
  n,  
  start.date = as.Date("1895-01-01"),  
  end.date = Sys.Date(),  
  p.male = 0.4,  
  p.temp = 0,  
  num.cores = 1  
)
```

### Arguments

<code>n</code>	number of generated hetu-pins
<code>start.date</code>	Lower limit of generated hetu dates, character string in ISO 8601 standard, for example "2001-02-03". Default is "1895-01-01".
<code>end.date</code>	Upper limit of generated hetu. Default is current date.
<code>p.male</code>	Probability of males, between 0.0 and 1.0. Default is 0.4.
<code>p.temp</code>	Probability of temporary identification numbers, between 0.0 and 1.0. Default is 0.0.
<code>num.cores</code>	The number of cores for parallel processing. The number of available cores can be determined with <code>detectCores()</code> . Default is 1.

### Details

There is a finite number of valid personal identity codes available per day. More specifically, there are 498 odd personal numbers for males and 498 even personal numbers for females from range 002-899. Additionally there are 50 odd numbers for males and 50 even numbers for females in the temporary personal identity code number range 900-999 that is not normally in use. This function

will return an error "too few positive probabilities" in `sample.int` function if you try to generate too many codes in a short enough timeframe.

The theoretical upper limit of valid PINs is in the millions since there are 898 PINs available for each day, 327770 for each year. In practice this number is much lower since same personal number component cannot be "recycled" if it has been used in the past. To illustrate, if an identity code "010101-0101" has already been assigned to someone born in 1901-01-01, a similar code "010101A0101" for someone born in 2001-01-01 could not be used.

### Value

a vector of generated hetu-pins.

### Author(s)

Pyry Kantanen, Jussi Paananen

### Examples

```
x <- rpin(3)
hetu(x)
hetu(x, extract = "sex")
hetu(x, extract = "ctrl.char")

x <- rhetu(3)
x
```

---

satu\_control\_char

*Finnish Unique Identification Number Control Character Calculator*

---

### Description

Calculate a valid control character for an incomplete Finnish Unique Identification Number (FINUID, or sähköinen asiointitunnus SATU).

### Usage

```
satu_control_char(pin, print.full = FALSE)
```

### Arguments

<code>pin</code>	An incomplete FINUID that has 8 first numbers.
<code>print.full</code>	Should the function print only the whole FINUID-number (TRUE) or only the control character (FALSE). Default is FALSE.

### Details

This method of calculating the control character was devised by mathematician Erkki Pale (1962) to detect input errors but also to detect errors produced by early punch card machines. The long number produced by writing the birth date and the personal number together are divided by 31 and the remainder is used to look up the control character from a separate table containing alphanumeric characters except letters G, I, O, Q and Z.

The method of calculating the control character does not need century character and therefore the function has an option to omit it.

### Value

Control character, either a number 0-9 or a letter (length 1 character). If parameter `print.full` is set to `TRUE`, the function returns a complete FINUID / SATU number (length 9 characters).

### Author(s)

Pyry Kantanen

### See Also

For more detailed information about FINUID, see Finnish Digital and population data services agency website: <https://dvv.fi/en/citizen-certificate-and-electronic-identity>

### Examples

```
# The first assigned FINUID number, 10000001N.  
satu_control_char("10000001")
```

---

satu\_ctrl

*Check Validity of Finnish Unique Identification Number (SATU)*

---

### Description

A function that checks whether a `satu` (Finnish Unique Identification Number) is valid. Returns `TRUE` or `FALSE`.

### Usage

```
satu_ctrl(satu)
```

### Arguments

`satu` a vector of 1 or more Unique Identification Numbers

### Examples

```
satu_ctrl("10000001N") # TRUE  
satu_ctrl(c("10000001N", "20000001B")) # TRUE FALSE
```

# Index

bid\_ctrl, 2

hetu, 3, 5, 7, 9

hetu\_age (pin\_age), 6

hetu\_control\_char, 4

hetu\_ctrl (pin\_ctrl), 7

hetu\_date (pin\_date), 8

hetu\_diagnostic, 5

hetu\_sex (pin\_sex), 8

pin\_age, 6

pin\_ctrl, 4, 7

pin\_date, 8

pin\_diagnostic (hetu\_diagnostic), 5

pin\_sex, 8

rbid, 9

rhetu (rpin), 10

rpin, 10

satu\_control\_char, 11

satu\_ctrl, 12