Package 'finid'

November 25, 2026

Type Package

Title Validation and Normalization of Financial Identifiers

Version 0.1.1

Description Validate and normalize identifiers that are ubiquitous in financial technology data pipelines: payment card numbers (Luhn), International Bank Account Numbers (IBAN, ISO 13616), International Securities Identification Numbers (ISIN, ISO 6166), CUSIP, SEDOL, Legal Entity Identifiers (LEI, ISO 17442), Financial Instrument Global Identifiers (FIGI), SWIFT/BIC codes (ISO 9362) and Chinese Unified Social Credit Codes (GB 32100-2015). All checks run offline with base R only.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.6.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Shengxin Hao [aut, cre]

Maintainer Shengxin Hao <haosx9924@mails.jlu.edu.cn>

Repository CRAN

Date/Publication 2026-11-25 12:40:08 UTC

Contents

cusip_valid 2
figi_valid 3
finid_detect 3
finid_normalize 4
iban_valid 5

isin_valid 6
lei_valid 7
luhn_valid 8
sedol_valid 9
swift_valid 9
uscc_valid 10

Index **12**

`cusip_valid` *Validate and inspect CUSIPs*

Description

A CUSIP is a nine character identifier for North American securities: eight alphanumeric characters followed by a decimal check digit. `cusip_valid()` verifies the structure and checksum; `cusip_check_digit()` computes the check digit for an eight character prefix.

Usage

```
cusip_valid(x)

cusip_check_digit(x)
```

Arguments

`x`	A character vector of CUSIPs (or eight character prefixes for `cusip_check_digit()`). Spaces and hyphens are removed before validation.

Value

`cusip_valid()` returns a logical vector; `cusip_check_digit()` returns an integer vector, `NA` for malformed input.

Examples

```
cusip_valid("037833100")   # Apple Inc.
cusip_valid("037833101")
cusip_check_digit("03783310")
cusip_valid("594918104")   # Microsoft Corp.
```

figi_valid *Validate and inspect Financial Instrument Global Identifiers (FIGIs)*

Description

A FIGI is a 12 character identifier defined by the Object Management Group. The first two characters identify the certified provider, the third character is always "G", characters 4 to 11 are upper-case consonants (or digits) and character 12 is a check digit computed with the modulus 10 "double-add-double" technique.

Usage

```
figi_valid(x)

figi_check_digit(x)
```

Arguments

x	A character vector of FIGIs (or 11 character prefixes for figi_check_digit()). Spaces and hyphens are removed before validation.

Value

figi_valid() returns a logical vector; figi_check_digit() returns an integer vector, NA for malformed input.

Examples

```
figi_valid("BBG000BLNNH6")   # International Business Machines Corp.
figi_valid("BBG000BLNNH5")
figi_check_digit("BBG000BLNNH")
```

finid_detect *Detect the type of financial identifiers*

Description

finid_detect() classifies each element of a character vector as one of "lei", "uscc", "isin", "figi", "cusip", "sedol", "swift", "iban", "pan" or "unknown" by running the corresponding validator. finid_check() combines detection and validation into a data frame, which is convenient for profiling an identifier column before loading it.

Usage

```
finid_detect(x)

finid_check(x, type = NULL)
```

Arguments

x	A character vector of identifiers.
type	Optional type to use for every element instead of detection. One of "luhn", "pan", "iban", "isin", "cusip", "sedol", "lei", "figi", "swift" or "uscc".

Details

When an identifier could satisfy more than one specification, types are tested in the order shown above.

Value

- finid_detect(): a character vector of types; NA for NA input and "unknown" for values that match no supported type.
- finid_check(): a data frame with the columns input, normalized, type and valid.

Examples

```
finid_detect(c("US0378331005", "GB82WEST12345698765432", "not-an-id"))

finid_check(c(
  "US0378331005",
  "GB82 WEST 1234 5698 7654 32",
  "037833100",
  "4532015112830366",
  NA
))

finid_check("4532015112830366", type = "luhn")
```

---

finid_normalize *Normalize financial identifiers*

Description

Removes whitespace and hyphens and converts input to upper case. Financial identifiers are frequently copied from statements, invoices or web pages with decorative separators; every validator in finid applies this normalization before checking the identifier.

Usage

```
finid_normalize(x)
```

Arguments

x	A character vector. Factors are converted to character; numeric and logical input is converted with a warning-free `as.character()` call.

Value

A character vector of the same length as `x`, with `NA` values preserved.

Examples

```
finid_normalize("gb82 west 1234 5698 7654 32")
finid_normalize("4532-0151-1283-0366")
finid_normalize(c(NA, "US0378331005"))
```

---

iban_valid *Validate and inspect International Bank Account Numbers (IBAN)*

Description

`iban_valid()` validates IBANs against the ISO 13616 registry: country code, total length, national (BBAN) structure and the modulus 97 check digits. The remaining functions extract components and format a *valid* IBAN in groups of four characters.

Usage

```
iban_valid(x)

iban_format(x)

iban_country(x)

iban_check_digits(x)

iban_bban(x)
```

Arguments

x	A character vector of IBANs. Spaces and hyphens are removed before validation; see `finid_normalize()`.

Value

- `iban_valid()`: a logical vector.
- `iban_country()`, `iban_check_digits()`, `iban_bban()`: a character vector, `NA` for invalid IBANs.
- `iban_format()`: the IBAN in groups of four characters, `NA` for invalid IBANs.

Examples

```
iban_valid("GB82WEST12345698765432")
iban_valid("GB82WEST12345698765431")
iban_format("GB82 WEST 1234 5698 7654 32")
iban_country("DE89370400440532013000")
iban_bban("DE89370400440532013000")
```

---

isin_valid *Validate and inspect International Securities Identification Numbers (ISIN)*

Description

`isin_valid()` checks the structure (ISO 3166 country prefix plus nine alphanumeric characters and one decimal check digit) and the Luhn checksum of an ISIN. The special prefix `"XS"` used for internationally cleared securities is accepted in addition to ISO 3166-1 alpha-2 codes.

Usage

```
isin_valid(x)

isin_check_digit(x)

isin_country(x)
```

Arguments

`x`	A character vector of ISINs. Spaces and hyphens are removed before validation; see `finid_normalize()`.

Value

- `isin_valid()`: a logical vector.
- `isin_country()`: the two-character prefix, `NA` for invalid ISINs.
- `isin_check_digit()`: the check digit for an 11 character ISIN prefix (country code plus nine characters), `NA` if the prefix is malformed.

Examples

```
isin_valid("US0378331005")
isin_valid("US0378331004")
isin_country("US0378331005")
isin_check_digit("US037833100")

# Rebuild a valid ISIN from its prefix
paste0("US037833100", isin_check_digit("US037833100"))
```

`lei_valid` *Validate and inspect Legal Entity Identifiers (LEIs)*

Description

An LEI is a 20 character identifier for legal entities that take part in financial transactions (ISO 17442). Characters 1 to 4 identify the issuing Local Operating Unit, characters 5 and 6 are reserved and must be `"00"`, characters 7 to 18 identify the entity and characters 19 to 20 hold the modulus 97 check digits.

Usage

```
lei_valid(x)

lei_check_digits(x)
```

Arguments

`x`	A character vector of LEIs (or 18 character prefixes for `lei_check_digits()`). Spaces and hyphens are removed before validation.

Value

`lei_valid()` returns a logical vector; `lei_check_digits()` returns a two character vector, `NA` for malformed input.

Examples

```
lei_valid("529900T8BM49AURSDO55")
lei_valid("529900T8BM49AURSDO54")
lei_check_digits("529900T8BM49AURSDO")
```

luhn_valid *Luhn checksum validation*

Description

luhn_valid() checks whether a number satisfies the Luhn (modulus 10) checksum that protects payment card numbers and several other identifiers. pan_valid() additionally enforces the 12 to 19 digit length of a primary account number (PAN). luhn_check_digit() computes the check digit that has to be appended to an incomplete number.

Usage

```
luhn_valid(x)

pan_valid(x)

luhn_check_digit(x)
```

Arguments

x	A character vector of numbers. Non-digit characters (after normalization) make the corresponding element invalid.

Details

Input may contain spaces or hyphens; see finid_normalize().

Value

- luhn_valid() and pan_valid(): a logical vector. NA input yields NA.
- luhn_check_digit(): an integer vector with values 0 to 9, or NA when the input cannot be completed.

Examples

```
luhn_valid("4532015112830366")
luhn_valid("4532015112830367")
pan_valid("4532 0151 1283 0366")

# Check digit of a 15 digit body
luhn_check_digit("453201511283036")

# Round trip
paste0("123456789", luhn_check_digit("123456789"))
luhn_valid(paste0("123456789", luhn_check_digit("123456789")))
```

sedol_valid *Validate and inspect SEDOLs*

Description

A SEDOL (Stock Exchange Daily Official List) number is a seven character identifier for securities listed on the London Stock Exchange: six alphanumeric characters followed by a decimal check digit. The six characters are weighted 1, 3, 1, 7, 3, 9 and the check digit makes the weighted sum a multiple of ten.

Usage

```
sedol_valid(x)

sedol_check_digit(x)
```

Arguments

x	A character vector of SEDOLs (or six character prefixes for `sedol_check_digit()`). Spaces and hyphens are removed before validation.

Value

`sedol_valid()` returns a logical vector; `sedol_check_digit()` returns an integer vector, `NA` for malformed input.

Examples

```
sedol_valid("0263494")     # BAE Systems plc
sedol_valid("0263495")
sedol_check_digit("026349")
sedol_valid("B0YQ5W0")
```

swift_valid *Validate and inspect SWIFT/BIC codes*

Description

`swift_valid()` checks the structure of a Business Identifier Code (ISO 9362): four letters for the institution, a two letter ISO 3166-1 country code, a two character location code and an optional three character branch code.

Usage

```
swift_valid(x)

swift_country(x)

swift_bic8(x)

swift_branch(x)
```

Arguments

x	A character vector of BIC codes. Spaces and hyphens are removed before validation.

Value

- `swift_valid()`: a logical vector.
- `swift_country()`: the country code, `NA` for invalid codes.
- `swift_bic8()`: the eight character institution identifier, `NA` for invalid codes.
- `swift_branch()`: the three character branch code, `NA` for invalid codes or codes without a branch suffix (an eight character BIC refers to the primary office).

Examples

```
swift_valid("DEUTDEFF")
swift_valid("DEUTDEFF500")
swift_valid("DEUTDEFFXX")
swift_country("DEUTDEFF500")
swift_bic8("DEUTDEFF500")
swift_branch("DEUTDEFF500")
```

---

uscc_valid *Validate and inspect Chinese Unified Social Credit Codes*

Description

The Unified Social Credit Code (USCC) is the 18 character business identifier used in China (GB 32100-2015). The first 17 characters come from a 31 character alphabet (digits and upper-case letters excluding `I`, `O`, `S`, `V` and `Z`) and the last character is a check code weighted by `1, 3, 9, 27, ...` and taken modulo 31.

Usage

```
uscc_valid(x)

uscc_check_digit(x)
```

Arguments

x	A character vector of USCCs (or 17 character prefixes for `uscc_check_digit()`). Spaces and hyphens are removed before validation.

Value

`uscc_valid()` returns a logical vector; `uscc_check_digit()` returns a one character vector, `NA` for malformed input.

Examples

```
uscc_valid("91350100M000100Y43")
uscc_valid("91350100M000100Y44")
uscc_check_digit("91350100M000100Y4")
```

Index

cusip_check_digit *(*cusip_valid*)*, 2
cusip_valid, 2

figi_check_digit *(*figi_valid*)*, 3
figi_valid, 3
finid_check *(*finid_detect*)*, 3
finid_detect, 3
finid_normalize, 4
finid_normalize(), *5, 6*, *8*

iban_bban *(*iban_valid*)*, 5
iban_check_digits *(*iban_valid*)*, 5
iban_country *(*iban_valid*)*, 5
iban_format *(*iban_valid*)*, 5
iban_valid, 5
isin_check_digit *(*isin_valid*)*, 6
isin_country *(*isin_valid*)*, 6
isin_valid, 6

lei_check_digits *(*lei_valid*)*, 7
lei_valid, 7
luhn_check_digit *(*luhn_valid*)*, 8
luhn_valid, 8

pan_valid *(*luhn_valid*)*, 8

sedol_check_digit *(*sedol_valid*)*, 9
sedol_valid, 9
swift_bic8 *(*swift_valid*)*, 9
swift_branch *(*swift_valid*)*, 9
swift_country *(*swift_valid*)*, 9
swift_valid, 9

uscc_check_digit *(*uscc_valid*)*, 10
uscc_valid, 10