Package: fkr (via r-universe)

October 22, 2026

Type Package

Title Realistic Fake Data Generator for the Philippines

Version 0.1.0

Description Generates realistic, locale-aware synthetic data for the Philippines, covering both personal data (names, ages, birthdays, contact details, addresses, government-ID-style numbers) and business data (company names, industries, job titles, products, prices, and permit numbers). Every value, including anything formatted to resemble a government identifier, is drawn uniformly at random for software testing, prototyping, and teaching, and does not correspond to any real person, business, or record.

License MIT + file LICENSE

URL https://github.com/yng-me/fkr

BugReports https://github.com/yng-me/fkr/issues

Encoding UTF-8

Depends R (>= 3.5.0)

Imports stats, phscs, psgc

Suggests curl, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.3

NeedsCompilation no

Author Bhas Abdulsamad [aut, cre, cph] (ORCID: <https://orcid.org/0009-0002-5891-8124>)

Maintainer Bhas Abdulsamad <aeabdulsamad@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-10-22 15:24:06 UTC

RemoteUrl https://github.com/cran/fkr

RemoteRef HEAD

RemoteSha 4f7cfb309c5677f0c9b1acc8e75c0b4e043ef4fd

Contents

fkr . . . 3
fkr_address . . . 4
fkr_age . . . 5
fkr_barangay . . . 5
fkr_binary . . . 6
fkr_birthday . . . 7
fkr_birthday_split . . . 7
fkr_boolean . . . 8
fkr_business_permit_number . . . 9
fkr_city . . . 9
fkr_civil_status . . . 10
fkr_company . . . 10
fkr_company_suffix . . . 11
fkr_coordinates . . . 11
fkr_department . . . 12
fkr_email . . . 13
fkr_factor . . . 13
fkr_first_name . . . 14
fkr_full_name . . . 15
fkr_industry . . . 16
fkr_int . . . 16
fkr_job_title . . . 17
fkr_landline_number . . . 17
fkr_last_name . . . 18
fkr_mobile_number . . . 19
fkr_multiple_response . . . 19
fkr_num . . . 20
fkr_pagibig_number . . . 21
fkr_philhealth_number . . . 21
fkr_philsys_number . . . 22
fkr_price . . . 23
fkr_product . . . 23
fkr_province . . . 24
fkr_psgc_code . . . 24
fkr_psic_industry . . . 25
fkr_psoc_occupation . . . 26
fkr_region . . . 27
fkr_seed . . . 28
fkr_sex . . . 28
fkr_sss_number . . . 29
fkr_street_address . . . 29
fkr_tin_number . . . 30
fkr_uuid . . . 30
fkr_zip_code . . . 31

Index **32**

fkr — *Generate a complete fake dataset*

Description

A dataset-level abstraction over the individual `fkr_*()` generators: describe each column with a `specs` list and get back a single, ready-to-use `data.frame` with `n` rows.

Usage

```
fkr(n, specs)
```

Arguments

`n` Number of rows to generate. A single non-negative whole number.

`specs` A named list describing each column. Names become column names and must be unique and non-empty. Each element can be:

- A **function of** `n` – the common case – called as `spec(n)`. This covers every `fkr_*()` generator directly, including ones with extra optional arguments (their defaults just apply), and `function(n) fkr_city(n, province = "Cebu")` to fix extra arguments.
- A **function with a** `data` **argument** – for a column that depends on columns generated earlier in `specs` – called as `spec(n, data = data)`, where `data` is the partial result built so far (a `data.frame` with one column per already-processed spec, in `specs` order). The argument must be named exactly `data` for this to trigger; its position doesn't matter.
- An **already-generated vector or data.frame** with length/`nrow` equal to `n`.
- A **single scalar**, recycled to length `n` (e.g. a constant column like `country = "Philippines"`).

When a spec (whether a function's result or a literal value) is a `data.frame` – e.g. `fkr_address()` or `fkr_coordinates()` – its columns are flattened into the output, prefixed with the spec's own name (`address` becomes `address_street`, `address_city`, ...).

Value

A `data.frame` with `n` rows and one column per (flattened) spec, in the order given in `specs`.

Age/birthday consistency

If `fkr_age()` and either `fkr_birthday()` or `fkr_birthday_split()` both appear *unwrapped* (i.e. passed as the bare function, not e.g. `function(n) fkr_age(n, age_range = 18:65)`) among `specs`, `fkr()` draws the age once and feeds it to the birthday spec(s), so the generated age and the age implied by the birthdate always agree – instead of each independently drawing its own random age. Wrap either in an anonymous function (even a trivial `function(n) fkr_age(n)`) to opt out and get independent draws.

Examples

```
fkr_seed(1)
fkr(5, list(
  first_name = fkr_first_name,
  last_name = fkr_last_name,
  sex = fkr_sex,
  age = fkr_age,
  date_of_birth = fkr_birthday,
  address = fkr_address,
  coordinates = fkr_coordinates
))

# A column that depends on one generated earlier in `specs`:
fkr(5, list(
  age = fkr_age,
  civil_status = function(n, data) {
    ifelse(data$age < 18, "Single", fkr_civil_status(n))
  }
))
```

---

fkr_address *Generate complete, internally-consistent Philippine addresses*

Description

Each generated row shares a single, real region/province/city/ZIP combination, so unlike calling the individual `fkr_*` geo functions separately, the fields are always geographically consistent with each other.

Usage

```
fkr_address(n, region = NULL)
```

Arguments

`n`	Number of addresses to generate. A single non-negative whole number.
`region`	Optional region name to restrict the draw to.

Value

A `data.frame` with `n` rows and character columns `street`, `barangay`, `city`, `province`, `region`, `zip`.

Examples

```
fkr_address(5)
```

fkr_age — *Generate ages*

Description

Draws `n` ages from `age_range`, weighted by an exponential decay so that, like a real population pyramid, younger ages are more common than older ones by default.

Usage

```
fkr_age(n, age_range = 0:130, bias = -0.05)
```

Arguments

`n`	Number of ages to generate. A single non-negative whole number.
`age_range`	Integer vector of candidate ages. Default `0:130`. All values must be non-negative.
`bias`	Exponential decay rate applied over `age_range` (`exp(bias * age)`). Negative values (the default, `-0.05`) make younger ages more likely; `0` makes every age in `age_range` equally likely; positive values make older ages more likely.

Value

An integer vector of length `n`.

Examples

```
fkr_age(5)
fkr_age(5, age_range = 18:65, bias = 0)
```

fkr_barangay — *Generate barangay names*

Description

Draws from a curated list of barangay names that recur across many Philippine cities and municipalities (e.g. "Poblacion", "San Isidro").

Usage

```
fkr_barangay(n)
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.

Value

A character vector of length `n`.

Examples

```
fkr_barangay(5)
```

---

fkr_binary *Generate binary (0/1) values*

Description

Generate binary (0/1) values

Usage

```
fkr_binary(n, values = c(0L, 1L), prob = 0.5)
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.
`values`	The two values to draw from, as `c(value_for_0, value_for_1)`. Default `c(0L, 1L)`.
`prob`	Probability of drawing `values[2]`. Default `0.5`.

Value

A vector of length `n`, with the same type as `values`.

Examples

```
fkr_binary(5)
fkr_binary(5, values = c("No", "Yes"), prob = 0.8)
```

fkr_birthday *Generate birthdates*

Description

Generates a birthdate for each of n simulated people by first drawing an age with fkr_age(), then drawing a uniformly random date within that age's exact 1-year window relative to ref_date.

Usage

```
fkr_birthday(n, ..., age = NULL, ref_date = Sys.Date())
```

Arguments

n	Number of birthdates to generate. A single non-negative whole number.
...	Passed on to fkr_age() (e.g. age_range, bias). Ignored if age is supplied.
age	Optional precomputed ages (e.g. from a prior fkr_age() call) to generate birthdates for, instead of drawing new ones. Must be a non-negative numeric vector of length n. Passing this keeps an age column and a date_of_birth column consistent with each other instead of each being drawn independently; fkr() does this automatically when both fkr_age() and fkr_birthday() appear (unwrapped) in the same specs.
ref_date	Reference "today" that ages are computed against. Default Sys.Date().

Value

A Date vector of length n.

Examples

```
fkr_birthday(5)
fkr_birthday(5, age_range = 18:65)

ages <- fkr_age(5, age_range = 18:65)
fkr_birthday(5, age = ages) # birthdates consistent with `ages`
```

fkr_birthday_split *Generate birthdates as separate year/month/day components*

Description

A convenience wrapper around fkr_birthday() that splits each generated date into its year, month, and day parts – handy for forms that collect date of birth as three separate fields.

Usage

```
fkr_birthday_split(n, ..., age = NULL, ref_date = Sys.Date())
```

Arguments

n	Number of birthdates to generate. A single non-negative whole number.
...	Passed on to fkr_age() (e.g. age_range, bias). Ignored if age is supplied.
age	Optional precomputed ages to generate birthdates for; see fkr_birthday()'s age argument.
ref_date	Reference "today" that ages are computed against. Default Sys.Date().

Value

A data.frame with n rows and character columns year ("YYYY"), month ("MM"), and day ("DD").

Examples

```
fkr_birthday_split(5)
```

---

fkr_boolean *Generate boolean values*

Description

Generate boolean values

Usage

```
fkr_boolean(n, prob = 0.5)
```

Arguments

n	Number of values to generate. A single non-negative whole number.
prob	Probability of drawing TRUE. Default 0.5.

Value

A logical vector of length n.

Examples

```
fkr_boolean(5)
fkr_boolean(5, prob = 0.9)
```

fkr_business_permit_number

Generate business/mayor's permit-style numbers

Description

Produces values matching the `BP-YYYY-NNNNNN` format commonly used by local government Business Permits and Licensing Offices (BPLOs). **The digits are uniformly random** and do not correspond to any real business; see the package-level disclaimer in `?fkr`.

Usage

```
fkr_business_permit_number(n, year = as.integer(format(Sys.Date(), "%Y")))
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.
`year`	Permit year. Default is the current year. Must be a single 4-digit whole number.

Value

A character vector of length `n`.

Examples

```
fkr_business_permit_number(5)
fkr_business_permit_number(5, year = 2023)
```

fkr_city

Generate Philippine cities/municipalities

Description

Generate Philippine cities/municipalities

Usage

```
fkr_city(n, province = NULL, region = NULL)
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.
`province`	Optional province name to restrict the draw to.
`region`	Optional region name to restrict the draw to.

Value

A character vector of length `n`.

Examples

```
fkr_city(5)
fkr_city(5, province = "Cebu")
```

`fkr_civil_status` *Generate civil status*

Description

Draws from the civil status categories used on Philippine government and civil registry forms.

Usage

```
fkr_civil_status(n)
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.

Value

A character vector of length `n`, one of `"Single"`, `"Married"`, `"Widowed"`, `"Separated"`, or `"Annulled"`.

Examples

```
fkr_civil_status(5)
```

`fkr_company` *Generate company names*

Description

Builds realistic Philippine business names by combining one or two surnames, a business-type word, and a corporate suffix (e.g. `"Dela Cruz Trading Corp."`, `"Santos & Reyes Enterprises, Inc."`).

Usage

```
fkr_company(n)
```

Arguments

`n`	Number of company names to generate. A single non-negative whole number.

Value

A character vector of length `n`.

Examples

```
fkr_company(5)
```

---

fkr_company_suffix *Generate company suffixes*

Description

Generate company suffixes

Usage

```
fkr_company_suffix(n)
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.

Value

A character vector of length `n` (e.g. `"Inc."`, `"Corp."`, `"OPC"`).

Examples

```
fkr_company_suffix(5)
```

---

fkr_coordinates *Generate Philippine latitude/longitude coordinates*

Description

Draws a point uniformly at random within the decimal-degree bounding box of the Philippines, or of a single region if `region` is given. Bounding boxes are region-level approximations (not survey-grade), so a point is guaranteed to fall in roughly the right part of the archipelago, not necessarily on land.

Usage

```
fkr_coordinates(n, region = NULL, precision = 10)
```

Arguments

n	Number of coordinate pairs to generate. A single non-negative whole number.
region	Optional region name to restrict the draw to (one of fkr_region()'s possible outputs). If NULL (default), draws from the bounding box of the whole country.
precision	Number of decimal places to round latitude/ longitude to. Default 10.

Value

A data.frame with n rows and numeric columns latitude and longitude, each rounded to precision decimal places.

Examples

```
fkr_coordinates(5)
fkr_coordinates(5, region = "National Capital Region (NCR)")
```

---

fkr_department *Generate company department names*

Description

Generate company department names

Usage

```
fkr_department(n)
```

Arguments

n	Number of values to generate. A single non-negative whole number.

Value

A character vector of length n.

Examples

```
fkr_department(5)
```

fkr_email — *Generate email addresses*

Description

Builds realistic email addresses from (optionally supplied) first and last names, using the mix of free email providers most commonly seen in the Philippines.

Usage

```
fkr_email(n, first_name = NULL, last_name = NULL, domain = NULL)
```

Arguments

`n`	Number of email addresses to generate. A single non-negative whole number.
`first_name`	Optional character vector of first names to build addresses from. Must have length 1 or length `n`. If `NULL` (default), names are drawn with `fkr_first_name()`.
`last_name`	Optional character vector of last names, same recycling rule as `first_name`. If `NULL` (default), names are drawn with `fkr_last_name()`.
`domain`	Optional character vector of email domains (e.g. `"gmail.com"`) to use instead of the default weighted provider mix. Must have length 1 or length `n`.

Value

A character vector of length `n`.

Examples

```
fkr_email(5)
fkr_email(3, first_name = "Juan", last_name = "Dela Cruz")
```

fkr_factor — *Generate a factor variable with value labels and an optional variable label*

Description

Draws from `values` (the underlying, often numeric, codes) and returns a `factor()` over them, optionally relabeled with human-readable `labels` (the value set) and tagged with a `label` (a variable-level description), mirroring the "variable label" / "value labels" distinction common in survey/statistical data (e.g. SPSS, Stata, **haven**).

Usage

```
fkr_factor(n, values, labels = NULL, label = NULL, prob = NULL)
```

Arguments

n	Number of values to generate. A single non-negative whole number.
values	A non-empty atomic vector of underlying codes/levels, with no NAs (e.g. c(1, 2) or c("M", "F")).
labels	Optional character vector of display labels, the same length as values and in the same order (e.g. c("Male", "Female") for values = c(1, 2)). If NULL (default), values are used as their own labels, as in a plain factor().
label	Optional single string describing the variable itself (e.g. "Respondent's biological sex"), attached as attr(x, "label") – not to be confused with labels, the per-value display text.
prob	Optional numeric vector of sampling weights, the same length as values. Default NULL draws every value with equal probability.

Value

A factor of length n with levels values (coerced to character, per factor()'s usual behavior) and, if given, an attr(x, "label") variable description. Note that as.integer() on the result gives each value's *position* in values, not the original code – use levels() or as.character() to recover the original values.

Examples

```
fkr_factor(5, values = c(1, 2), labels = c("Male", "Female"))
fkr_factor(5,
  values = c(1, 2), labels = c("Male", "Female"),
  label = "Respondent's biological sex"
)
fkr_factor(5, values = c(1, 2, 3), prob = c(0.7, 0.2, 0.1))
```

---

fkr_first_name *Generate Filipino first names*

Description

Draws n given names from a lookup of ~700 male and ~760 female Filipino first names, optionally restricted by sex.

Usage

```
fkr_first_name(n, sex = c("any", "male", "female"))
```

Arguments

n	Number of names to generate. A single non-negative whole number.
sex	Restrict the draw to "male" or "female" names, or "any" (the default) to draw from both pools combined.

Value

A character vector of length n.

Examples

```
fkr_first_name(5)
fkr_first_name(5, sex = "female")
```

---

fkr_full_name *Generate full Filipino names*

Description

Combines fkr_first_name() and fkr_last_name(), optionally with a middle initial, in the common Filipino "First Middle. Last" convention.

Usage

```
fkr_full_name(n, sex = c("any", "male", "female"), middle_initial = FALSE)
```

Arguments

n	Number of names to generate. A single non-negative whole number.
sex	Restrict the given name to "male", "female", or "any" (default).
middle_initial	If TRUE, include a random middle initial (e.g. "Juan D. Dela Cruz"). Default FALSE.

Value

A character vector of length n.

Examples

```
fkr_full_name(5)
fkr_full_name(5, sex = "male", middle_initial = TRUE)
```

fkr_industry *Generate industry names*

Description

Draws from industries prominent in the Philippine economy (e.g. BPO, agriculture, remittance services).

Usage

```
fkr_industry(n)
```

Arguments

n	Number of values to generate. A single non-negative whole number.

Value

A character vector of length n.

Examples

```
fkr_industry(5)
```

fkr_int *Generate whole numbers*

Description

Draws integers uniformly at random between min and max, inclusive.

Usage

```
fkr_int(n, min = 0, max = 100)
```

Arguments

n	Number of values to generate. A single non-negative whole number.
min	Minimum value (inclusive). Default 0.
max	Maximum value (inclusive). Default 100. Must be >= min.

Value

An integer vector of length n.

Examples

```
fkr_int(5)
fkr_int(5, min = 18, max = 65)
```

---

fkr_job_title *Generate job titles*

Description

Generate job titles

Usage

```
fkr_job_title(n)
```

Arguments

n	Number of values to generate. A single non-negative whole number.

Value

A character vector of length n.

Examples

```
fkr_job_title(5)
```

---

fkr_landline_number *Generate Philippine landline numbers*

Description

Draws landline numbers using real area codes for Metro Manila and a curated set of major provincial cities.

Usage

```
fkr_landline_number(n, area_code = NULL)
```

Arguments

n	Number of landline numbers to generate. A single non-negative whole number.
area_code	Optional city name to restrict the area code to (one of `names(fkr:::.ph_landline_area_codes)`, e.g. `"Cebu"`, `"Davao"`). If `NULL` (default), a city is drawn at random for each number.

Value

A character vector of length n, formatted as "(area code) subscriber number", e.g. "(02) 8123 4567" or "(032) 234 5678".

Examples

```
fkr_landline_number(5)
fkr_landline_number(3, area_code = "Cebu")
```

---

fkr_last_name *Generate Filipino last names (surnames)*

Description

Draws n surnames from the ~1,000 most common Philippine surnames, ranked by real-world frequency (Spanish-derived names like "Dela Cruz", "Garcia", and "Reyes" dominate the Philippine namescape). By default the draw is weighted by that ranking, so rank-1 surnames come up far more often than rank-1000 ones, the way they would in an actual population sample.

Usage

```
fkr_last_name(n, weighted = TRUE)
```

Arguments

n	Number of surnames to generate. A single non-negative whole number.
weighted	If TRUE (default), weight the draw by real-world frequency rank. If FALSE, draw uniformly from all ~1,000 surnames.

Value

A character vector of length n.

Examples

```
fkr_last_name(5)
fkr_last_name(5, weighted = FALSE)
```

fkr_mobile_number *Generate Philippine mobile numbers*

Description

Draws mobile numbers using real network prefixes (Smart/TNT, Globe/TM, or DITO).

Usage

```
fkr_mobile_number(
  n,
  network = c("any", "smart", "globe", "dito"),
  format = c("local", "international")
)
```

Arguments

n	Number of mobile numbers to generate. A single non-negative whole number.
network	Restrict the draw to "smart", "globe", or "dito" prefixes, or "any" (default) to mix all networks.
format	"local" (default) returns "09XXXXXXXXX"; "international" returns "+639XXXXXXXXX".

Value

A character vector of length n.

Examples

```
fkr_mobile_number(5)
fkr_mobile_number(5, network = "globe", format = "international")
```

fkr_multiple_response *Generate multiple-response (select-all-that-apply) values*

Description

Simulates a "select all that apply" survey item: each generated value is one or more entries from values combined together, e.g. with values = LETTERS[1:4] a result might be "A", "AC", or "ABCD". Selected entries always keep their original order from values (so "AC" can occur but "CA" cannot), and never repeat within a single generated value.

Usage

```
fkr_multiple_response(n, values, multiple = TRUE, sep = "")
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.
`values`	A non-empty vector of candidate entries, with no `NA`s, to combine (e.g. `LETTERS[1:4]`, or option labels like `c("Email", "Phone", "SMS")`).
`multiple`	If `TRUE` (default), each generated value may combine more than one entry from `values`. If `FALSE`, exactly one entry is drawn per value, as in a single-response ("select one") item.
`sep`	String used to join multiple selected entries together. Default `""`, which reads naturally when `values` are single characters (e.g. `"AC"`); for multi-character `values`, pass something like `sep = ", "` to keep entries distinguishable (e.g. `"Email, SMS"`).

Details

The number of entries selected for each value is drawn uniformly from `1:length(values)` (not uniformly over all non-empty subsets), so picking exactly 1 entry is just as likely as picking all of them.

Value

A character vector of length `n`.

Examples

```
fkr_multiple_response(5, values = LETTERS[1:4])
fkr_multiple_response(5, values = c("Email", "Phone", "SMS"), sep = ", ")
fkr_multiple_response(5, values = LETTERS[1:4], multiple = FALSE)
```

`fkr_num` *Generate numbers*

Description

Draws numbers uniformly at random between `min` and `max`.

Usage

```
fkr_num(n, min = 0, max = 1, decimals = 2)
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.
`min`	Minimum value (inclusive). Default `0`.
`max`	Maximum value (inclusive). Default `1`. Must be `>= min`.
`decimals`	Number of decimal places to round to. Default `2`.

Value

A numeric vector of length `n`.

Examples

```
fkr_num(5)
fkr_num(5, min = 0, max = 100, decimals = 1)
```

`fkr_pagibig_number` *Generate Pag-IBIG (HDMF) number-style identifiers*

Description

Produces values matching the `NNNN-NNNN-NNNN` format used by the Home Development Mutual Fund (Pag-IBIG Fund). **The digits are uniformly random** and do not correspond to any real member; see the package-level disclaimer in `?fkr`.

Usage

```
fkr_pagibig_number(n)
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.

Value

A character vector of length `n`.

Examples

```
fkr_pagibig_number(5)
```

`fkr_philhealth_number` *Generate PhilHealth number-style identifiers*

Description

Produces values matching the `NN-NNNNNNNNN-N` format used by the Philippine Health Insurance Corporation. **The digits are uniformly random** and do not correspond to any real member; see the package-level disclaimer in `?fkr`.

Usage

```
fkr_philhealth_number(n)
```

Arguments

n	Number of values to generate. A single non-negative whole number.

Value

A character vector of length n.

Examples

```
fkr_philhealth_number(5)
```

---

fkr_philsys_number *Generate PhilSys Number (PSN) card-style identifiers*

Description

Produces values matching the NNNN-NNNN-NNNN-NNNN format printed on the Philippine national ID (PhilSys). **The digits are uniformly random** and do not correspond to any real registrant; see the package-level disclaimer in ?fkr.

Usage

```
fkr_philsys_number(n)
```

Arguments

n	Number of values to generate. A single non-negative whole number.

Value

A character vector of length n.

Examples

```
fkr_philsys_number(5)
```

fkr_price — *Generate prices in Philippine pesos*

Description

Draws prices on a log scale between `min` and `max` so that, like real retail pricing, low values are much more common than high ones.

Usage

```
fkr_price(n, min = 10, max = 10000, format = c("currency", "numeric"))
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.
`min`	Minimum price. Default `10`.
`max`	Maximum price. Default `10000`. Must be greater than `min`.
`format`	`"currency"` (default) returns a formatted character vector like `"PHP 1,234.56"`; `"numeric"` returns a raw numeric vector rounded to 2 decimal places.

Value

A character vector (if `format = "currency"`) or numeric vector (if `format = "numeric"`) of length `n`.

Examples

```
fkr_price(5)
fkr_price(5, min = 100, max = 500, format = "numeric")
```

fkr_product — *Generate product names*

Description

Draws from a curated list of products typical of Philippine retail and sari-sari (small variety) stores.

Usage

```
fkr_product(n)
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.

Value

A character vector of length n.

Examples

```
fkr_product(5)
```

---

fkr_province *Generate Philippine provinces*

Description

Generate Philippine provinces

Usage

```
fkr_province(n, region = NULL)
```

Arguments

n	Number of values to generate. A single non-negative whole number.
region	Optional region name to restrict the draw to (one of fkr_region()'s possible outputs).

Value

A character vector of length n.

Examples

```
fkr_province(5)
fkr_province(5, region = "Region VII (Central Visayas)")
```

---

fkr_psgc_code *Generate Philippine Standard Geographic Code (PSGC) codes*

Description

Draws real PSGC codes – the Philippine Statistics Authority's official hierarchical coding system for regions, provinces, cities, municipalities, and barangays – at the requested level, optionally restricted to a specific region/province/city.

Usage

```
fkr_psgc_code(
  n,
  level = c("city", "region", "province", "municipality", "barangay"),
  region = NULL,
  province = NULL,
  city = NULL
)
```

Arguments

`n`	Number of codes to generate. A single non-negative whole number.
`level`	Geographic level of the code: `"region"`, `"province"`, `"city"`, `"municipality"`, or `"barangay"`. Default `"city"`.
`region`	Optional region name to restrict the draw to (see `fkr_region()`).
`province`	Optional province name to restrict the draw to (see `fkr_province()`).
`city`	Optional city/municipality name to restrict the draw to (useful with `level = "barangay"` to get barangay codes within a specific city, or on its own to look up a specific city's code).

Details

Codes are fetched live via the **psgc** package (which caches its data locally after the first call) and kept in memory for the rest of the R session, so the first call in a session may require network access.

Value

A character vector of length `n` of 9- or 10-digit PSGC codes.

Examples

```
fkr_psgc_code(5)
fkr_psgc_code(5, level = "region")
fkr_psgc_code(5, level = "barangay", city = "Davao City")
```

`fkr_psic_industry` *Generate PSIC-classified industries*

Description

Draws industry/economic-activity titles from the Philippine Standard Industrial Classification (PSIC), the Philippine Statistics Authority's official industry taxonomy (aligned with ISIC Rev. 4), together with their real classification code. Complements `fkr_industry()`, which draws from an informal, uncoded list.

Usage

```
fkr_psic_industry(
  n,
  level = c("divisions", "sections", "groups", "classes", "sub-classes"),
  section = NULL
)
```

Arguments

`n`	Number of industries to generate. A single non-negative whole number.
`level`	PSIC hierarchy level to draw from: `"sections"` (broadest, e.g. "Manufacturing"), `"divisions"`, `"groups"`, `"classes"`, or `"sub-classes"` (most specific). Default `"divisions"`.
`section`	Optional section title to restrict the draw to (e.g. `"Manufacturing"`, `"Construction"`). See `fkr_psic_industry()`'s `section` output column for valid values.

Details

Data is fetched live via the **phscs** package (which caches its data locally after the first call) and kept in memory for the rest of the R session, so the first call in a session may require network access.

Value

A `data.frame` with `n` rows and columns `title` (the industry/activity title), `psic_code` (its PSIC code), and `section` (its top-level PSIC section title).

Examples

```
fkr_psic_industry(5)
fkr_psic_industry(5, level = "sections")
fkr_psic_industry(5, section = "Manufacturing")
```

`fkr_psoc_occupation` *Generate PSOC-classified occupations*

Description

Draws occupation titles from the Philippine Standard Occupational Classification (PSOC), the Philippine Statistics Authority's official occupational taxonomy (aligned with ISCO-08), together with their real classification code.

Usage

```
fkr_psoc_occupation(
  n,
  level = c("unit", "minor", "sub-major", "major"),
  major_group = NULL
)
```

Arguments

n	Number of occupations to generate. A single non-negative whole number.
level	PSOC hierarchy level to draw from: "unit" (most specific, e.g. "Finance managers"), "minor", "sub-major", or "major" (broadest, e.g. "Managers"). Default "unit".
major_group	Optional major-group title to restrict the draw to (e.g. "Professionals", "Elementary Occupations"). See fkr_psoc_occupation()'s major_group output column for valid values.

Details

Data is fetched live via the **phscs** package (which caches its data locally after the first call) and kept in memory for the rest of the R session, so the first call in a session may require network access.

Value

A data.frame with n rows and columns title (the occupation/group title), psoc_code (its PSOC code), and major_group (its top-level PSOC major-group title).

Examples

```
fkr_psoc_occupation(5)
fkr_psoc_occupation(5, level = "major")
fkr_psoc_occupation(5, major_group = "Professionals")
```

---

fkr_region *Generate Philippine regions*

Description

Generate Philippine regions

Usage

```
fkr_region(n)
```

Arguments

n	Number of values to generate. A single non-negative whole number.

Value

A character vector of length n, drawn from the Philippines' 17 administrative regions.

Examples

```
fkr_region(5)
```

fkr_seed — *Seed fkr's random number generation*

Description

A thin, self-documenting wrapper around `set.seed()` so a whole sequence of `fkr_*()` calls can be made reproducible.

Usage

```
fkr_seed(seed)
```

Arguments

`seed`	A single whole number.

Value

`NULL`, invisibly. Called for its side effect.

Examples

```
fkr_seed(123)
fkr_first_name(3)
```

fkr_sex — *Generate biological sex*

Description

Generate biological sex

Usage

```
fkr_sex(n, levels = c("Male", "Female"))
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.
`levels`	Labels to use for the two sexes, as `c(male_label, female_label)`. Defaults to `c("Male", "Female")`.

Value

A character vector of length `n`.

Examples

```
fkr_sex(5)
```

---

fkr_sss_number *Generate SSS number-style identifiers*

Description

Produces values matching the `NN-NNNNNNN-N` format used by the Philippine Social Security System. **The digits are uniformly random** and do not correspond to any real SSS member; see the package-level disclaimer in `?fkr`.

Usage

```
fkr_sss_number(n)
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.

Value

A character vector of length `n`.

Examples

```
fkr_sss_number(5)
```

---

fkr_street_address *Generate street addresses*

Description

Generates a house/unit number followed by a common Philippine street name and street type (e.g. `"482 Mabini Street"`).

Usage

```
fkr_street_address(n)
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.

Value

A character vector of length `n`.

Examples

```
fkr_street_address(5)
```

---

fkr_tin_number *Generate TIN-style identifiers*

Description

Produces values matching the `NNN-NNN-NNN-NNN` format used for Philippine Bureau of Internal Revenue Tax Identification Numbers. **The digits are uniformly random** and do not correspond to any real taxpayer; see the package-level disclaimer in `?fkr`.

Usage

```
fkr_tin_number(n)
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.

Value

A character vector of length `n`.

Examples

```
fkr_tin_number(5)
```

---

fkr_uuid *Generate UUIDs*

Description

Generates random (version 4, RFC 4122) universally unique identifiers.

Usage

```
fkr_uuid(n, version = 4)
```

Arguments

`n`	Number of UUIDs to generate. A single non-negative whole number.
`version`	UUID version to generate. Only `4` (random) is currently supported.

Value

A character vector of length `n`, formatted as `"xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx"` (lowercase hex, with the version nibble fixed to `4` and the variant nibble one of `8`, `9`, `a`, `b`).

Examples

```
fkr_uuid(5)
```

---

fkr_zip_code *Generate Philippine ZIP codes*

Description

Generate Philippine ZIP codes

Usage

```
fkr_zip_code(n, province = NULL, city = NULL)
```

Arguments

`n`	Number of values to generate. A single non-negative whole number.
`province`	Optional province name to restrict the draw to.
`city`	Optional city/municipality name to restrict the draw to.

Value

A character vector of length `n` of 4-digit ZIP codes.

Examples

```
fkr_zip_code(5)
fkr_zip_code(5, city = "Davao City")
```

Index

factor(), *13, 14*
fkr, 3
fkr(), *7*
fkr_address, 4
fkr_address(), *3*
fkr_age, 5
fkr_age(), *3*, *7, 8*
fkr_barangay, 5
fkr_binary, 6
fkr_birthday, 7
fkr_birthday(), *3*, *7, 8*
fkr_birthday_split, 7
fkr_birthday_split(), *3*
fkr_boolean, 8
fkr_business_permit_number, 9
fkr_city, 9
fkr_civil_status, 10
fkr_company, 10
fkr_company_suffix, 11
fkr_coordinates, 11
fkr_coordinates(), *3*
fkr_department, 12
fkr_email, 13
fkr_factor, 13
fkr_first_name, 14
fkr_first_name(), *13*, *15*
fkr_full_name, 15
fkr_industry, 16
fkr_industry(), *25*
fkr_int, 16
fkr_job_title, 17
fkr_landline_number, 17
fkr_last_name, 18
fkr_last_name(), *13*, *15*
fkr_mobile_number, 19
fkr_multiple_response, 19
fkr_num, 20
fkr_pagibig_number, 21
fkr_philhealth_number, 21
fkr_philsys_number, 22
fkr_price, 23
fkr_product, 23
fkr_province, 24
fkr_province(), *25*
fkr_psgc_code, 24
fkr_psic_industry, 25
fkr_psic_industry(), *26*
fkr_psoc_occupation, 26
fkr_psoc_occupation(), *27*
fkr_region, 27
fkr_region(), *12*, *24, 25*
fkr_seed, 28
fkr_sex, 28
fkr_sss_number, 29
fkr_street_address, 29
fkr_tin_number, 30
fkr_uuid, 30
fkr_zip_code, 31

set.seed(), *28*
Sys.Date(), *7, 8*