Package 'standardlastprofile'

June 11, 2026

Title BDEW Standard Load Profiles for Electricity and Gas

Description Provides representative standard load profiles (SLPs) for electricity and gas published by the German Association of Energy and Water Industries (BDEW Bundesverband der Energie- und Wasserwirtschaft e.V.) in a tidy format. The electricity profiles cover the 1999 profiles — households (H0), commercial (G0–G6), and agriculture (L0–L2) — and the updated 2025 profiles (H25, G25, L25, P25, S25), which additionally represent households with photovoltaic systems and battery storage. An interface generates an electricity load profile over a user-defined date range. A second interface generates daily gas load profiles using the BDEW/VKU/GEODE synthetic 'SigLinDe' procedure from daily temperatures and a customer value ('Kundenwert'). The 1999 data and methodology are described in VDEW (1999), ``Repräsentative VDEW-Lastprofile'', <https://www.bdew.de/media/documents/1999_Repraesentative-VDEW-Lastprofile.pdf>. The electricity generation algorithm is described in VDEW (2000), ``Anwendung der Repräsentativen VDEW-Lastprofile step-by-step'', <https://www.bdew.de/media/documents/2000131_Anwendung-repraesentativen_Lastprofile-Step-by-step.pdf>. The 2025 profiles are described in BDEW (2025), ``Standardlastprofile Strom'', <https://www.bdew.de/energie/standardlastprofile-strom/>. The gas procedure is described in BDEW (2025), ``Standardlastprofile Gas'', <https://www.bdew.de/energie/standardlastprofile-gas/>.

Version 2.0.0

License CC0

Encoding UTF-8

Suggests covr, ggplot2, httr2, testthat (>= 3.0.0), vdiffr, withr

Config/testthat/edition 3

Config/Needs/website jsonlite, knitr, leaflet, plotly, rdwd, rmarkdown, scales

URL https://github.com/flrd/standardlastprofile, https://flrd.github.io/standardlastprofile/

BugReports https://github.com/flrd/standardlastprofile/issues

Imports lifecycle
Depends R (>= 4.1)
LazyData true
Language en-GB
Config/roxygen2/version 8.0.0
NeedsCompilation no
Author Markus Döring [aut, cre, cph]
Maintainer Markus Döring <m4rkus.doering@gmail.com>
Repository CRAN
Date/Publication 2026-06-11 11:00:02 UTC

Contents

`slp_electricity` *Generate a Standard Load Profile for Electricity*

Description

Generate a standard load profile in watts, normalised to an annual consumption of 1,000 kWh.

Usage

```
slp_electricity(profile_id, start_date, end_date, holidays = NULL)
```

Arguments

`profile_id`	load profile identifier, required
`start_date`	start date in ISO 8601 format, required
`end_date`	end date in ISO 8601 format, required
`holidays`	controls public holiday treatment: • `NULL` (default): built-in nationwide German holidays are used. • `NA`: no dates are treated as public holidays. • a character or Date vector in ISO 8601 format (`"YYYY-MM-DD"`): only these dates are treated as public holidays; the built-in data are ignored entirely.

Details

In the German electricity market, a standard load profile is a representative pattern of electricity consumption used to forecast demand for customer groups that are not continuously metered. For each distinct combination of `profile_id`, `period`, and `day` there are 96 quarter-hourly measurements of electrical power, normalised to an annual consumption of 1,000 kWh.

See the electricity algorithm article for more details.

Profile IDs:

There are 16 profile IDs across two generations:

1999 profiles:

- `H0`: Households
- `G0`, `G1`, `G2`, `G3`, `G4`, `G5`, `G6`: Commercial
- `L0`, `L1`, `L2`: Agriculture

2025 profiles

In 2025, BDEW published an updated set of standard load profiles reflecting changes in electricity consumption patterns since the original 1999 study. Five new profiles are included:

- `H25`: Households — updated version of `H0`
- `G25`: Commercial (general) — updated version of `G0`
- `L25`: Agriculture — updated version of `L0`
- `P25`: combination profile for households with a photovoltaic (PV) system
- `S25`: combination profile for households with a PV system and battery storage

For descriptions of each profile, call `slp_info()`.

Periods and day types:

1999 profiles use three seasonal periods:

- `summer`: May 15 to September 14
- `winter`: November 1 to March 20
- `transition`: March 21 to May 14, and September 15 to October 31

2025 profiles use calendar months (`january` ... `december`) instead of seasons.

Within each period, days are classified as:

- `workday`: Monday to Friday
- `saturday`: Saturdays; Dec 24th and Dec 31st are also treated as Saturdays unless they fall on a Sunday
- `sunday`: Sundays and all public holidays

Public holidays:

By default, the following nine public holidays observed nationwide across all German states are treated as Sundays:

- New Year's Day (1 January)
- Good Friday
- Easter Monday
- Labour Day (1 May)
- Ascension Day

- Whit Monday
- German Unity Day (3 October)
- Christmas Day (25 December)
- Boxing Day (26 December)

State-level holidays are **not** included by default. These vary by state and can change — for example, Berlin observed a one-time holiday on 8 May 2025 (end of World War II anniversary). Use the `holidays` argument to supply your own dates instead; the built-in data are then ignored entirely.

The built-in holiday data cover the years 1990 to 2099. For dates outside this range, `holidays = NULL` will yield no public holiday adjustments; pass `holidays` explicitly if needed.

Units and conversion:

The 1999 source file stores values in watts (W), normalised to 1,000 kWh/a. The 2025 source file stores values in kWh per 15-minute interval, normalised to 1,000,000 kWh/a. To keep all profiles consistent, the 2025 values are converted to watts normalised to 1,000 kWh/a.

To convert to energy consumed per interval in kWh:

```
kwh <- out$watts / 4 / 1000
```

Value

A data.frame with four variables:

- `profile_id`, character, load profile identifier
- `start_time`, POSIXct / POSIXlt, start time
- `end_time`, POSIXct / POSIXlt, end time
- `watts`, numeric, average electric power in watts per 15-minute interval, normalised to an annual consumption of 1,000 kWh

Source

https://www.bdew.de/energie/standardlastprofile-strom/

https://www.bdew.de/media/documents/1999_Repraesentative-VDEW-Lastprofile.pdf

https://www.bdew.de/media/documents/2000131_Anwendung-repraesentativen_Lastprofile-Step-by-step.pdf

Examples

```
start <- "2026-01-01"
end <- "2026-12-31"

# multiple profile IDs are supported
L <- slp_electricity(c("L0", "L1", "L2"), start, end)
head(L)

# supply custom holiday dates (e.g. only treat New Year's Day as a holiday)
H0_custom <- slp_electricity("H0", start, end, holidays = "2026-01-01")
```

```
# Fetch state-level holidays from the nager.Date API and pass them in.
# Each entry in the API response contains two relevant fields:
#   $global  — logical; TRUE = nationwide holiday, FALSE = state-specific
#   $counties — list of ISO 3166-2 state codes (e.g. "DE-BE" for Berlin)
#               when global is FALSE; NULL otherwise
#
# Berlin (DE-BE) observes International Women's Day (March 8) in addition
# to all nationwide holidays. The example below fetches 2027 holidays,
# keeps entries where global is TRUE or "DE-BE" appears in counties, and
# passes the resulting dates to slp_electricity().
## Not run:
resp <- httr2::request("https://date.nager.at/api/v3") |>
  httr2::req_url_path_append("PublicHolidays", "2027", "DE") |>
  httr2::req_perform() |>
  httr2::resp_body_json()

is_berlin <- \(x) isTRUE(x$global) || "DE-BE" %in% unlist(x$counties)
holidays_berlin_2027 <- as.Date(
  vapply(Filter(is_berlin, resp), \(x) x$date, character(1))
)

H0_berlin_2027 <- slp_electricity(
  "H0", "2027-01-01", "2027-12-31",
  holidays = holidays_berlin_2027
)

## End(Not run)

# consider only nationwide public holidays (default)
H0_2026 <- slp_electricity("H0", start, end)

# electric power values are normalised to consumption of ~1,000 kWh/a
sum(H0_2026$watts / 4 / 1000)

# convert watts to kWh per interval using a wrapper
slp_generate_kwh <- \(...) {
  out <- slp_electricity(...)
  out$kwh <- out$watts / 4 / 1000
  out
}
H0_kwh <- slp_generate_kwh("H0", start, end)
head(H0_kwh)
```

`slp_electricity_profiles`

Standard Load Profile Data for Electricity from BDEW

Description

Data about representative, standard load profiles for electricity from the German Association of Energy and Water Industries (BDEW Bundesverband der Energie- und Wasserwirtschaft e.V.) in a tidy format.

Usage

```
slp_electricity_profiles
```

Format

A data.frame with 26,784 observations and 5 variables:

profile_id character, identifier for load profile, see 'Details'

period character, one of `'summer'`, `'winter'`, `'transition'` for 1999 profiles; one of `'january'` through `'december'` for 2025 profiles

day character, one of `'saturday'`, `'sunday'`, `'workday'`

timestamp character, format: %H:%M

watts numeric, electric power in watts, normalised to 1,000 kWh/a

Details

There are 96 x 1/4h measurements of electrical power for each combination of `profile_id`, `period` and `day`, which we refer to as the "standard load profile".

In total there are 16 `profile_id` across two generations of profiles:

1999 profiles (based on analysis of 1,209 load profiles of low-voltage electricity consumers in Germany):

- Households: `H0`
- Commercial: `G0`, `G1`, `G2`, `G3`, `G4`, `G5`, `G6`
- Agriculture: `L0`, `L1`, `L2`

2025 profiles (updated profiles published by BDEW in 2025):

- Households: `H25`
- Commercial: `G25`
- Agriculture: `L25`
- Combination profile PV: `P25`
- Combination profile storage and PV: `S25`

The 2025 profiles use calendar months rather than seasons for the `period` column (`'january'` through `'december'`).

Call `slp_info()` for more information and examples.

Period definitions (1999 profiles):

- `summer`: May 15 to September 14

- `winter`: November 1 to March 20
- `transition`: March 21 to May 14, and September 15 to October 31

Day definitions:

- `workday`: Monday to Friday
- `saturday`: Saturdays; Dec 24th and Dec 31st are also treated as Saturdays unless they fall on a Sunday
- `sunday`: Sundays and all public holidays

Units and normalisation:

The source Excel file for the 1999 profiles stores values in watts (W), normalised to an annual consumption of 1,000 kWh/a. The source Excel file for the 2025 profiles stores values in kilowatt-hours (kWh) per 15-minute interval, normalised to 1,000,000 kWh/a. To keep the internal representation consistent and backwards compatible, all 2025 values have been converted to watts normalised to 1,000 kWh/a.

As a result, the `watts` column in both this dataset and the output of `slp_electricity()` always represents average electric power in watts, normalised to 1,000 kWh/a. To convert to energy consumed per 15-minute interval in kWh, divide by 4 and by 1,000:

```
watts_to_kwh <- \(x) x / 4 / 1000
```

Source

https://www.bdew.de/energie/standardlastprofile-strom/

https://www.bdew.de/media/documents/Profile.zip

https://www.bdew.de/media/documents/1999_Repraesentative-VDEW-Lastprofile.pdf

Examples

```
head(slp_electricity_profiles)
```

`slp_gas` *Generate a Standard Load Profile for Gas*

Description

Generate daily gas consumption values using the BDEW/VKU/GEODE synthetic standard load profile procedure (SigLinDe method).

Usage

```
slp_gas(
  profile_id,
  dates,
  temperatures,
  kundenwert,
  variant = c("34", "33"),
  holidays = NULL
)
```

Arguments

`profile_id`	gas load profile identifier, required. One of `"HEF"`, `"HMF"`, `"HKO"`, `"GKO"`, `"GHA"`, `"GMK"`, `"GBD"`, `"GBH"`, `"GWA"`, `"GGA"`, `"GBA"`, `"GGB"`, `"GPD"`, `"GMF"`, `"GHD"`. Multiple values are supported.
`dates`	a Date vector or character vector in ISO 8601 format (`"YYYY-MM-DD"`). Each element is the **start date** of a gas day (German: *Gastag*, 06:00–06:00). Must have the same length as `temperatures`.
`temperatures`	a numeric vector of daily temperatures in degrees Celsius, one value per gas day. Must have the same length as `dates`. The temperature should be the allocation temperature (German: *Allokationstemperatur*) for that gas day. Two options are supported by the Leitfaden (see Details): • **Simple daily mean** (*Tagesmitteltemperatur*): arithmetic average of hourly values over the gas day. • **Geometrically-weighted 4-day mean**: recommended by BDEW for distribution network operators. In production contexts, distribution network operators increasingly use the **gas forecast temperature** (German: *Gasprognosetemperatur*, GPT) published by DWD or DTN instead of a raw daily mean. The GPT incorporates a multi-day weighted average and seasonal adjustment that reduces the systematic seasonal allocation bias of pure temperature-based profiles (VKU SLP evaluation reports 2023, 2025). This function accepts whichever temperature values are passed; the choice of method is the caller's responsibility.
`kundenwert`	numeric scalar, required. Customer value (Kundenwert) in kWh/day — the daily gas consumption at the reference temperature of 8 °C. Derive it once from a full reference year with `slp_gas_kundenwert()`, or supply a value you already know. See Details.
`variant`	SigLinDe variant (German: *Ausprägung*) to use. Either `"34"` (default) or `"33"`. Variant 34 has a 57 % linear component and a steeper heating slope; variant 33 has a 45 % linear component. The BDEW Leitfaden recommends that distribution network operators test both variants against their own grid data and select the better fit. See Details. The `"HKO"` profile is a pure sigmoid with no linear part and is unaffected by this argument.
`holidays`	controls public holiday treatment: • `NULL` (default): built-in nationwide German holidays are used.

- NA: no dates are treated as public holidays.
- a character or Date vector in ISO 8601 format ("YYYY-MM-DD"): only these dates are treated as public holidays; the built-in data are ignored entirely.

Details

Background:

In the (German) gas market, standard load profiles (Standardlastprofile, SLP) are used to allocate gas volumes to low-pressure customers who are not continuously metered. The synthetic procedure computes a daily gas quantity as:

$$Q(D) = KW \times h(\vartheta_D) \times F_{WT}$$

where:

- KW is a customer-specific scaling factor in kWh/day (German: *Kundenwert*).
- $h(\vartheta_D)$ is the SigLinDe profile function value for the daily temperature ϑ_D.
- F_{WT} is the weekday factor for the profile and day type.

SigLinDe Profile Function:

The SigLinDe function is defined in two variants (German: *Ausprägungen*). The pure sigmoid term was introduced by TU München (Geiger / Hellwig 2002); the linear envelope on top — together with the 33 / 34 variant split — was added by FfE in the 2015 research report *Weiterentwicklung des Standard- lastprofilverfahrens Gas* (Appendix 7.1). The current operational coefficient set is published in the BDEW Leitfaden, Appendix 6 (as of 2025-10-28):

$$h(\vartheta) = \frac{A}{1 + \left(\frac{B}{\vartheta - \vartheta_0}\right)^C} + D + \max(m_H\vartheta + b_H,\ m_W\vartheta + b_W)$$

The first four terms form the sigmoid part; the last term is the linear part (space-heating and hot water lines). Variant 34 (57 % linear component, steeper heating slope) is the default. Variant 33 (45 % linear component) is an alternative for distribution network areas where it fits better. Distribution network operators are advised to test both against their own grid data.

The HKO profile (Kochgasprofil) is a pure sigmoid retained from the pre-SigLinDe era; it has no 33/34 variant and its linear part is always zero.

Allocation temperature:

The allocation temperature can be computed in two ways:

Simple daily mean — arithmetic mean of hourly temperatures:

$$\vartheta_D = \frac{1}{24}\sum_{h=1}^{24} T_h$$

Geometrically-weighted 4-day mean (recommended by BDEW for network operators):

$$\vartheta_D = \frac{T_D + 0.5 \times T_{D-1} + 0.25 \times T_{D-2} + 0.125 \times T_{D-3}}{1.875}$$

This function accepts whichever temperature values the user provides in temperatures; the choice of method is the user's responsibility.

Kundenwert:

The Kundenwert KW scales the dimensionless profile to a customer's actual consumption and is a **required** input. The recommended workflow is two steps:

1. Derive KW once from a full reference year of temperatures with `slp_gas_kundenwert()`:

$$KW = \frac{E_a}{\sum_D h(\vartheta_D) \times F_{WT,D}}$$

   where E_a is the annual consumption.
2. Pass that KW to `slp_gas()` for any period you want to generate.

Keeping the two steps separate is deliberate: `kundenwert` is a property of the customer and their climate zone, computed from a representative (ideally multi-year) temperature mean. Deriving it from the same short series you are generating over would collapse the seasonal denominator and bias the result — for a single day the h values cancel entirely.

Profile IDs:

There are 15 gas profile IDs defined in the BDEW Leitfaden:

Residential:

- `HEF`: single-family home (Einfamilienhaus)
- `HMF`: multi-family home (Mehrfamilienhaus)
- `HKO`: cooking and hot water only (Kochen / Warmwasser)

Commercial / industrial:

- `GKO`: small commercial (Kleinstgewerbe)
- `GHA`: trade and commerce (Handel)
- `GMK`: metal and automotive (Metall / Kfz)
- `GBD`: services (Dienstleistung)
- `GBH`: accommodation (Beherbergung)
- `GWA`: laundries (Wäscherei)
- `GGA`: gastronomy (Gastronomie)
- `GBA`: bakeries (Bäckerei)
- `GGB`: mixed commercial (gemischtes Gewerbe)
- `GPD`: paper and printing (Papier / Druck)
- `GMF`: large multi-family / mixed use (Mehrfamilienhaus groß)
- `GHD`: trade, commerce and services aggregate (GHD-Stützpunkt)

Weekday Factors:

Unlike the electricity profiles, gas weekday factors use seven individual weekdays (Mo, Tu, We, Th, Fr, Sa, Su) rather than three day types. Public holidays are treated as Sunday (`Su`); 24 December and 31 December are treated as Saturday (`Sa`) unless they fall on a Sunday.

For the residential profiles `HEF`, `HMF`, and `HKO` all weekday factors are 1, meaning no day-of-week differentiation is applied.

The built-in holiday data cover the years 1990 to 2099. For dates outside this range, `holidays = NULL` will yield no public holiday adjustments; pass `holidays` explicitly if needed.

Value

A data.frame with three variables:

- `profile_id`, character, gas load profile identifier
- `date`, Date, start date of the gas day (06:00 local time)
- `kwh`, numeric, estimated gas consumption in kWh for that gas day

Source

https://www.bdew.de/energie/standardlastprofile-gas/

BDEW/VKU/GEODE. *Leitfaden Abwicklung von Standardlastprofilen Gas*, Kooperationsvereinbarung Gas, Annex XIV.2, as of 2025-10-28, Appendix 6. https://www.bdew.de/media/documents/251028_LF_SLP_Gas_KoV_XIV.2.pdf

See Also

`slp_gas_kundenwert()` to derive the kundenwert; `slp_gas_coefficients()` and `slp_gas_siglinde()` for the underlying coefficients and profile function.

Examples

```
dates <- seq.Date(as.Date("2026-01-01"), as.Date("2026-01-07"), by = "day")
temps <- c(2.1, -1.3, 0.5, 3.8, 5.2, 4.0, 1.9)

# Supply the Kundenwert directly (kWh/day)
slp_gas("HEF", dates, temps, kundenwert = 55.1)

# Multiple profiles
slp_gas(c("HEF", "HMF", "GKO"), dates, temps, kundenwert = 55.1)
```

---

slp_gas_coefficients *Retrieve SigLinDe Coefficients for Gas Standard Load Profiles*

Description

Returns the SigLinDe profile function coefficients for one or more gas standard load profiles as a data frame. These are the values used internally by `slp_gas()` and `slp_gas_siglinde()`.

Usage

```
slp_gas_coefficients(profile_id = NULL, variant = NULL)
```

Arguments

`profile_id`	character vector of gas profile identifiers. One or more of `"HEF"`, `"HMF"`, `"HKO"`, `"GKO"`, `"GHA"`, `"GMK"`, `"GBD"`, `"GBH"`, `"GWA"`, `"GGA"`, `"GBA"`, `"GGB"`, `"GPD"`, `"GMF"`, `"GHD"`. Pass `NULL` (the default) to retrieve all 15 profiles.
`variant`	character vector of SigLinDe variants to include. One or both of `"34"` (57 % linear component) and `"33"` (45 % linear component). Pass `NULL` (the default) to retrieve both variants. Duplicate values are silently ignored.

Details

The `HKO` profile (Kochgasprofil) is a pure sigmoid with no linear component; its `mH`, `bH`, `mW`, and `bW` are all zero for both variants.

The returned coefficients can be passed directly to `slp_gas_siglinde()` for custom calculations. When selecting a single profile and variant the result is a one-row data frame, so use `[[ ]]` or `$` to extract scalars:

```
p <- slp_gas_coefficients("HEF", variant = "34")
slp_gas_siglinde(0, p$A, p$B, p$C, p$D, p$theta0, p$mH, p$bH, p$mW, p$bW)
```

[ ]: R:%20

Value

A data frame with one row per profile–variant combination and 11 variables:

profile_id character, gas profile identifier

variant character, SigLinDe variant (`"34"` or `"33"`)

A, B, C, D numeric, sigmoid coefficients

theta0 numeric, pole temperature (40 °C for all published profiles)

mH, bH numeric, slope and intercept of the space-heating line (*Heizgas-Gerade*)

mW, bW numeric, slope and intercept of the hot-water line (*Warmwasser-Gerade*)

Source

BDEW/VKU/GEODE (2025). *Leitfaden Abwicklung von Standardlastprofilen Gas*, Kooperationsvereinbarung Gas, Annex XIV.2, as of 2025-10-28, Appendix 6. https://www.bdew.de/media/documents/251028_LF_SLP_Gas_KoV_XIV.2.pdf

See Also

`slp_gas_siglinde()`, `slp_gas()`, `slp_gas_weekday_factors()`; all values are listed in tabular form in the SigLinDe parameters article.

Examples

```
# Single profile, both variants
slp_gas_coefficients("HEF")

# Single profile, single variant
slp_gas_coefficients("HEF", variant = "34")

# Both variants explicitly — same as NULL
slp_gas_coefficients(c("HEF", "GKO"), variant = c("34", "33"))
```

`slp_gas_kundenwert` *Compute the Kundenwert for a Gas Standard Load Profile*

Description

Compute the customer value (Kundenwert, KW) that scales a gas standard load profile to a specific annual consumption. The result can be passed directly to `slp_gas()` via its `kundenwert` argument, enabling a two-step workflow: derive KW from a representative full-year reference temperature series, then generate profiles for any shorter period using that fixed KW.

Usage

```
slp_gas_kundenwert(
  profile_id,
  dates = NULL,
  temperatures = NULL,
  annual_consumption = 1000,
  variant = c("34", "33"),
  holidays = NULL
)
```

Arguments

`profile_id`	gas load profile identifier, required. Same values as `slp_gas()`. Multiple values are supported; the result is a named numeric vector with one element per profile.
`dates`	a Date vector or character vector in ISO 8601 format (`"YYYY-MM-DD"`), representing a **full reference year** of daily dates. For a meaningful Kundenwert the series should ideally cover 365 (or 366) days. Must have the same length as `temperatures`.
`temperatures`	a numeric vector of daily temperatures in degrees Celsius. Must have the same length as `dates`.
`annual_consumption`	numeric scalar, annual gas consumption in kWh. Defaults to `1000`.
`variant`	SigLinDe variant, either `"34"` (default) or `"33"`. Must match the `variant` passed to `slp_gas()` when applying the resulting Kundenwert.
`holidays`	controls public holiday treatment. Same semantics as in `slp_gas()`. The reference year used here should apply the same holiday calendar as the generation step.

Details

The Kundenwert is defined as:

$$KW = \frac{E_a}{\sum_D h(\vartheta_D) \cdot F_{WT,D}}$$

where E_a is `annual_consumption` and the sum $\sum_D h(\vartheta_D) \cdot F_{WT,D}$ runs over all days in the temperature series. For the result to be meaningful the denominator must reflect a full seasonal cycle.

Reference temperature series:

For a robust Kundenwert the temperature series should represent a **full reference year**, ideally a multi-year climatological mean rather than a single year, so that no individual-year anomaly distorts the scaling factor; with fewer than 365 days a message is shown.

Daily mean temperatures can be downloaded from the DWD (Deutscher Wetterdienst) open-data archive, e.g. via the rdwd package. The gas SLP article on the package website walks through fetching DWD data, deriving the Kundenwert, and generating profiles.

Recommended workflow:

`slp_gas()` requires a `kundenwert`. If you do not already know it, compute it first with `slp_gas_kundenwert()` from a full reference year and the customer's annual consumption, then pass the result into `slp_gas()` to generate the profile for whatever period you need:

```
# Step 1 — derive KW from a full-year reference temperature series
kw <- slp_gas_kundenwert("HEF", dates_year, temps_year, annual_consumption = 15000)

# Step 2 — generate a profile for any shorter period
slp_gas("HEF", dates_jan_mar, temps_jan_mar, kundenwert = kw)
```

Value

A named numeric vector of length `length(profile_id)`. Each element is the Kundenwert in kWh/day for the corresponding profile. Names match the input `profile_id` values.

See Also

`slp_gas()`

Examples

```
# Derive KW from a full-year reference temperature series
dates_ref <- seq.Date(as.Date("2024-01-01"), as.Date("2024-12-31"), by = "day")
doy       <- as.integer(format(dates_ref, "%j"))

# fake temperature data for demonstration here only
temps_ref <- 10 - 11 * cos(2 * pi * (doy - 15) / 365)
slp_gas_kundenwert("HEF", dates = dates_ref, temperatures = temps_ref,
                   annual_consumption = 15000)

# Multiple profiles at once
```

```
slp_gas_kundenwert(c("HEF", "GKO", "GWA"), dates_ref, temps_ref,
                   annual_consumption = 15000)
```

`slp_gas_siglinde` *Compute Dimensionless Daily Heating Demand (SigLinDe)*

Description

Computes the dimensionless daily heating demand $h(\vartheta)$ for a given outdoor temperature using the SigLinDe method.

Usage

```
slp_gas_siglinde(theta, A, B, C, D, theta0, mH, bH, mW, bW)
```

Arguments

`theta`	Numeric vector of daily mean outdoor temperatures in °C (the allocation temperature).
`A`, `B`, `C`, `D`	Numeric scalars — sigmoid coefficients.
`theta0`	Numeric scalar — pole temperature (40 °C for all published profiles). The function is undefined at $\vartheta = \vartheta_0$ and physically meaningless above it.
`mH`, `bH`	Numeric scalars — slope and intercept of the heating linear component (*Heizgas-Gerade*).
`mW`, `bW`	Numeric scalars — slope and intercept of the hot-water linear component (*Warmwasser-Gerade*).

Details

The function value is the sum of a sigmoid part and a linear part:

$$h(\vartheta) = \frac{A}{1 + \left(\frac{B}{\vartheta - \vartheta_0}\right)^C} + D + \max(m_H\vartheta + b_H,\ m_W\vartheta + b_W)$$

The sigmoid captures the non-linear relationship between outdoor temperature and heating demand. The linear envelope of two lines represents space-heating demand (*Heizgas-Gerade*, slope `mH`) and hot-water demand (*Warmwasser-Gerade*, slope `mW`).

For residential profiles (e.g. `HEF`, `HMF`) both parts contribute. For the `HKO` (TUM) profile the linear coefficients are all zero, so only the sigmoid part remains.

This is the low-level building block used internally by `slp_gas()`. It is exported so that users with custom or region-specific coefficients (e.g. state-level parameters such as `BW_HEF03` for Baden-Württemberg) can compute $h(\vartheta)$ directly and build their own profiles.

Published coefficients for all 15 standard profiles are listed in the SigLinDe parameters article.

Value

A numeric vector the same length as `theta` giving the dimensionless profile value $h(\vartheta)$ for each temperature.

References

BDEW/VKU/GEODE (2025). *Abwicklung von Standardlastprofilen Gas*, Kooperationsvereinbarung Gas, Annex XIV.2, as of 2025-10-28. The unified SigLinDe profile function is shown on p. 42 (Abbildung 12; PDF page 54); the per-profile coefficients are tabulated in Appendix 6, pp. 145–166. https://www.bdew.de/media/documents/251028_LF_SLP_Gas_KoV_XIV.2.pdf

See Also

`slp_gas()`; SigLinDe parameters article

Examples

```
# h value at 0 °C for HEF (single-family home), variant 34
slp_gas_siglinde(
  theta = 0,
  A = 1.3819663, B = -37.4124155, C = 6.1723179, D = 0.0396284,
  theta0 = 40,
  mH = -0.0672159, bH = 1.1167138,
  mW = -0.0019982, bW = 0.1355070
)

# h values across a temperature range
temps <- seq(-15, 30, by = 5)
slp_gas_siglinde(
  theta = temps,
  A = 1.3819663, B = -37.4124155, C = 6.1723179, D = 0.0396284,
  theta0 = 40,
  mH = -0.0672159, bH = 1.1167138,
  mW = -0.0019982, bW = 0.1355070
)
```

---

slp_gas_weekday_factors

Retrieve Weekday Factors for Gas Standard Load Profiles

Description

Returns the weekday scaling factors (F_{WT}) for one or more gas standard load profiles as a data frame. These are the values used internally by `slp_gas()`.

Usage

```
slp_gas_weekday_factors(profile_id = NULL)
```

Arguments

profile_id	character vector of gas profile identifiers. Same values as slp_gas(). Pass NULL (the default) to retrieve all 15 profiles.

Details

For the residential profiles HEF, HMF, and HKO all weekday factors are 1: gas consumption in households is assumed not to vary by day of the week. Commercial profiles show sector-specific patterns — for example, GWA (laundries) has high Monday–Wednesday factors (busy wash days) and very low weekend factors.

Public holidays are treated as Sunday ("Su"); 24 and 31 December are treated as Saturday ("Sa") unless they fall on a Sunday. See slp_gas() for details.

Value

A data frame with one row per profile–day combination and 3 variables:

profile_id character, gas profile identifier

day character, abbreviated weekday: "Mo", "Tu", "We", "Th", "Fr", "Sa", "Su"

f_wt numeric, weekday scaling factor

Source

BDEW/VKU/GEODE (2025). *Leitfaden Abwicklung von Standardlastprofilen Gas*, Kooperationsvereinbarung Gas, Annex XIV.2, as of 2025-10-28, Appendix 6. https://www.bdew.de/media/documents/251028_LF_SLP_Gas_KoV_XIV.2.pdf

See Also

slp_gas(), slp_gas_coefficients(); all values are listed in tabular form in the SigLinDe parameters article.

Examples

```
slp_gas_weekday_factors(c("HEF", "GWA"))
```

---

slp_info *Retrieve information on standard load profiles*

Description

Returns descriptions for electricity and gas standard load profiles defined by BDEW. Accepts both electricity profile IDs (H0, G0–G6, L0–L2, H25, G25, L25, P25, S25) and gas profile IDs (HEF, HMF, HKO, GKO, GHA, GMK, GBD, GBH, GWA, GGA, GBA, GGB, GPD, GMF, GHD).

Usage

```
slp_info(profile_id, language = c("EN", "DE"))
```

Arguments

profile_id	character vector of profile identifiers. Electricity and gas IDs can be mixed freely.
language	one of "EN" (default) or "DE".

Value

A named list with one element per profile_id. Each element is a list with character components profile (the identifier), description (a short label), and — for electricity profiles only — details (a longer explanation).

Source

https://www.bdew.de/energie/standardlastprofile-strom/

https://www.bdew.de/energie/standardlastprofile-gas/

Examples

```
# Electricity profile
slp_info("H0")

# Gas profile
slp_info("HEF")

# Mixed
slp_info(c("H0", "HEF", "GKO"))

# German descriptions
slp_info("HEF", language = "DE")
```

Index