Package 'oRaklE'

April 29, 2025

Title Multi-Horizon Electricity Demand Forecasting in High Resolution

Version 1.0.0

Description Advanced forecasting algorithms for long-term energy demand at the national or regional level. The methodology is based on Grandón et al. (2024) <doi:10.1016/j.apenergy.2023.122249>; Zimmermann & Ziel (2024) <doi:10.2139/ssrn.4823013>. Real-time data, including power demand, weather conditions, and macroeconomic indicators, are provided through automated API integration with various institutions. The modular approach maintains transparency on the various model selection processes and encompasses the ability to be adapted to individual needs. 'oRaklE' tries to help facilitating robust decision-making in energy management and planning.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

LazyData true

LazyDataCompression xz

Imports ggplot2, scales, MLmetrics, MuMIn, R.utils, caret, survival, countrycode, doParallel, dplyr, ggthemes, glmnet, httr, jsonlite, lubridate, mgcv, patchwork, purrr, xml2, zoo

Depends R (>= 3.5)

Suggests knitr, rmarkdown, roxygen2 (>= 7.2.3), spelling, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Language en-US

NeedsCompilation no

Author Johannes Schwenzer [aut, cre, cph] (<https://orcid.org/0009-0006-9618-8889>), Simone Maxand [aut] (<https://orcid.org/0000-0002-3153-7922>), Tatiana Gonzalez Grandón [aut] (<https://orcid.org/0000-0001-6587-0144>)

Maintainer Johannes Schwenzer <schwenzer@europa-uni.de>

Repository CRAN

Date/Publication 2025-04-29 08:50:02 UTC

Contents

add_holidays_mid_term *Add holidays to the mid-term series*

Description

This function adds a dummy variable for holidays to the mid-term data series. Information on the holidays is retrieved from "https://date.nager.at/api/v3/publicholidays/".

Usage

```
add_holidays_mid_term(midterm_data)
```

Arguments

midterm_data	The mid-term data series resulting from the function decompose_load_data.

Value

The mid-term series with an additional column of holiday dummies.

See Also

See also mid_term_lm for the prediction model.

Examples

```
example_midterm_demand_data <- add_holidays_mid_term(example_decomposed_data$midterm)
head(example_midterm_demand_data)
```

add_holidays_short_term *Add holidays to the short-term series*

Description

This function adds a dummy variable for holidays to the short-term data series. Information on the holidays is retrieved from "https://date.nager.at/api/v3/publicholidays/".

Usage

```
add_holidays_short_term(shortterm)
```

Arguments

shortterm	The short-term data series resulting from decompose_load_data

Value

The short-term series with an additional column of holiday dummies.

Examples

```
example_shortterm_demand_data <- add_holidays_short_term(example_decomposed_data$shortterm)
example_shortterm_demand_data[1:5, c(1, 2, 11)]
```

`combine_models` *Combine forecast models*

Description

This function combines the three separate forecasts for the low, mid and high frequency model. The three separate forecasts need to be run first.

Usage

```
combine_models(
  longterm_predictions,
  midterm_predictions,
  shortterm_predictions,
  longterm_model_number = 1,
  data_directory = tempdir(),
  verbose = FALSE
)
```

Arguments

`longterm_predictions`	Dataframe. The object resulting from function `long_term_lm`.
`midterm_predictions`	Dataframe. The object resulting from function `mid_term_lm`.
`shortterm_predictions`	Dataframe. The object resulting from function `short_term_lm`.
`longterm_model_number`	Integer. Specifies which of the 3 best long-term models should be used.
`data_directory`	The path to the directory where the results, metrics and plots will be saved. The default is set to a temporary directory.
`verbose`	A boolean value indicating if you want the generated plots and final result metrics to be shown (set to TRUE if yes).

Value

A list with the dataframe with the combined model results. A dataframe with selected model metrics. And a list with the plotted results. The combined model predictions, plots, and metrics are saved in the respective folder for the country.

combined_model_predictions A dataframe with the combined model results.

combined_model_metrics A dataframe with model metrics.

combined_model_plots A list with the plot for the full timeseries, a plot with two sample weeks, and a stacked plot with both.

Examples

```
example_full_model_predictions <- combine_models(example_longterm_predictions,
  example_midterm_predictions, example_shortterm_predictions,
  longterm_model_number = 1
)
```

`combine_models_future` *Combine forecast models for future predictions*

Description

This function combines the three separate future forecasts for the low, mid and high frequency model. The three separate forecasts need to be run first and should have the same end_year.

Usage

```
combine_models_future(
  longterm_future_predictions,
  midterm_future_predictions,
  shortterm_future_predictions,
  longterm_model_number = 1,
  data_directory = tempdir(),
  verbose = FALSE
)
```

Arguments

`longterm_future_predictions`
: Dataframe. The dataframe object resulting from function `long_term_future`.

`midterm_future_predictions`
: Dataframe. The dataframe object resulting from function `mid_term_future`.

`shortterm_future_predictions`
: Dataframe. The dataframe object resulting from function `short_term_future`.

`longterm_model_number`
: Integer. Specifies which of the 3 best long-term models should be used.

`data_directory` The path to the directory where the results and plots will be saved. The default is set to a temporary directory.

`verbose` A boolean value indicating if you want the generated plots to be shown (set to TRUE if yes).

Value

A list with the dataframe with the combined model results. And a list with the plotted results. The combined model predictions and plots are saved in the respective folder for the country.

combined_model_future_predictions A dataframe with the combined model results.

combined_model_future_plots A list with the plot for the full timeseries, a plot with two sample weeks, and a stacked plot with both.

See Also

See also functions `long_term_future`, `mid_term_future`, and `short_term_future` for the prediction models.

Examples

```
example_full_model_future_predictions <- combine_models_future(example_longterm_future_predictions,
  example_midterm_future_predictions, example_shortterm_future_predictions,
  longterm_model_number = 1
)
```

---

decompose_load_data *Decomposing the load data into long-, mid- and short-term component*

Description

This function decomposes the load data into three components: a yearly long-term trend, a daily mid-term seasonality, and an hourly short-term seasonality. If the data is available only at a daily resolution, the calculation of hourly seasonality is skipped. The results of the decomposition are returned as a list of dataframes. The series are plotted additionally.

Usage

```
decompose_load_data(load_data, data_directory = tempdir(), verbose = FALSE)
```

Arguments

`load_data` A data frame object with "load", "date", "unit", and "country" columns

load Consisting of the load values, numeric.

date Consisting of the datetime values, datetime (e.g. POSIXct).

unit Indicating the unit, e.g. MW, character.

country Indicating the country's ISO2C code, character.

`data_directory` The path to the directory where the data will be saved. The default is set to a temporary directory.

`verbose` A boolean value indicating if you want the generated plots to be shown (set to TRUE if yes)

Value

A list of three data frames with

longterm A data frame of the long-term trend, including columns for country, year, and yearly average hourly demand.

midterm A data frame of the mid-term component, including country, date, year, month, day, weekday, average hourly demand, and seasonal average hourly demand. Where seasonal average hourly demand corresponds to the difference between the yearly average demand per hour and the daily average demand per hour of the respective day.

shortterm A data frame of the short-term component, including country, date, year, month, day, weekday, hour, hourly demand, and hourly demand trend and trend and season corrected. Where hourly demand trend and season corrected corresponds to the difference between the daily average demand per hour and the actual demand in the respective hour, effectively showing the intra-day pattern.

plots A list with all created plots.

Examples

```
example_decomposed_data <- decompose_load_data(example_demand_data_filled)
```

---

example_decomposed_data

Example Decomposed Data

Description

This dataset contains the decomposed data of `example_demand_data_filled` into a long-term trend, a mid-term seasonality, and a short-term seasonality. It contains a list of three dataframes, one for each component.

Usage

```
example_decomposed_data
```

Format

A list with three dataframes:

`longterm` A dataframe with long-term trend specific data

`midterm` A dataframe with mid-term seasonality specific data

`shortterm` A dataframe with short-term seasonality specific data

Longterm Data Frame:

`longterm.country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`longterm.year` The respective year.

`longterm.avg_hourly_demand` The average hourly electricity demand (in megawatts) for each year.

`example` A boolean indicator to mark this dataset as an example dataset.

Midterm Data Frame:

`midterm.country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`midterm.date` The date of the demand measurement (in `YYYY-MM-DD` format).

`midterm.year` The respective year.

`midterm.month` The respective month.

`midterm.day` The respective day.

`midterm.wday` The type of weekday (e.g., `Sun`, `Mon`)

`midterm.avg_hourly_demand` The average hourly electricity demand (in megawatts) for each day.

`midterm.seasonal_avg_hourly_demand` The seasonal mid-term component of the demand (in megawatts).

`example` A boolean indicator to mark this dataset as an example dataset.

Shortterm Data Frame

`shortterm.country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`shortterm.date` The date of the demand measurement (in `YYYY-MM-DD HH:MM:SS` format).

`shortterm.year` The respective year.

`shortterm.month` The respective month.

`shortterm.day` The respective day.

`shortterm.wday` The type of weekday (e.g., `Sun`, `Mon`)

`shortterm.hour` The respective hour (from 0 to 23).

`shortterm.hourly_demand` The actual hourly electricity demand (in megawatts) for each hour.

`shortterm.hourly_demand_trend_corrected` The demand substracted by the long-term trend.

`shortterm.yearly` The yearly average electricity demand in the respective year.

`shortterm.daily` The daily average electricity demand in the respective day.

`shortterm.hourly_demand_trend_and_season_corrected` The short-term seasonal component which is the hourly demand, substracted by both the long-term trend and the mid-term seasonality.

`example` A boolean indicator to mark this dataset as an example dataset.

Source

Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/)

---

example_demand_data *Example Demand Data*

Description

This dataset contains the hourly electricity demand data of France from 2017 until 2021.

Usage

```
example_demand_data
```

Format

A data frame with 43,769 rows and 7 columns:

`date` The date and time of the demand measurement (in `YYYY-MM-DD HH:MM:SS` format).

`load` The electricity demand.

`unit` The unit of measurement (MW).

`year` The year of the respective timepoint.

`time_interval` The time interval at which the demand was reported (e.g., `60 mins`).

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`example` A boolean indicator to mark this dataset as an example dataset.

Source

Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/)

---

example_demand_data_filled *Example Demand Data Filled*

Description

This dataset contains the same data as `example_demand_data` with any missing values filled.

Usage

```
example_demand_data_filled
```

Format

A data frame with 43,824 rows and 7 columns:

`date` The date and time of the demand measurement (in `YYYY-MM-DD HH:MM:SS` format).

`load` The electricity demand.

`unit` The unit of measurement (MW).

`year` The year of the respective timepoint.

`time_interval` The time interval at which the demand was reported (e.g., `60 mins`).

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`example` A boolean indicator to mark this dataset as an example dataset.

Source

Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/)

`example_full_model_future_predictions`
Example Full Model Future Predictions Data

Description

This dataset extends the full model predictions from `example_full_model_predictions` until the year 2028.

Usage

```
example_full_model_future_predictions
```

Format

A data frame with 43800 rows and 12 columns:

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`date` The date (in `YYYY-MM-DD` format).

`year` The respective year.

`month` The respective month.

`day` The respective day.

`wday` The type of weekday (e.g., `Sun`, `Mon`)

`hour` The respective hour of the day.

`hourly_demand` The actual hourly electricity demand (in megawatts).

`long_term_model` The predicted long-term trend (yearly average of hourly demand) from the best long-term forecasting model.

`mid_term_model` The predicted mid-term seasonality (daily minus yearly average of hourly demand) from the best mid-term forecasting model.

`short_term_model` The predicted short-term seasonality (actual hourly demand minus the long-term trend minus the mid-term seasonality) from the best short-term forecasting model.

`complete_model` Final predicted electricity demand for each hour. Derived by adding the results from the long-, mid-, and short-term components.

Source

demand data: Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/); holidays: https://date.nager.at/api/v3/publicholidays/

`example_full_model_predictions`
Example Full Model Predictions Data

Description

This dataset combines the results from `long_term_lm`, `mid_term_lm`, and `short_term_lm` into the final predictions of hourly electricity demand for France from 2017 until 2021.

Usage

```
example_full_model_predictions
```

Format

A data frame with 43800 rows and 12 columns:

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`date` The date (in `YYYY-MM-DD` format).

`year` The respective year.

`month` The respective month.

`day` The respective day.

`wday` The type of weekday (e.g., `Sun`, `Mon`)

`hour` The respective hour of the day.

`hourly_demand` The actual hourly electricity demand (in megawatts).

`long_term_model` The predicted long-term trend (yearly average of hourly demand) from the best long-term forecasting model.

`mid_term_model` The predicted mid-term seasonality (daily minus yearly average of hourly demand) from the best mid-term forecasting model.

`short_term_model` The predicted short-term seasonality (actual hourly demand minus the long-term trend minus the mid-term seasonality) from the best short-term forecasting model.

`complete_model` Final predicted electricity demand for each hour. Derived by adding the results from the long-, mid-, and short-term components.

Source

demand data: Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/); holidays: https://date.nager.at/api/v3/publicholidays/

`example_longterm_and_macro_data`
Example Longterm and Macro Data

Description

This dataset extends the long-term average hourly electricity demand data from `example_longterm_data` with ten macro-economic indicators. The macro-economic data is taken from the World Development Indicators (WDI) of the World Bank (https://databank.worldbank.org/source/world-development-indicators).

Usage

```
example_longterm_and_macro_data
```

Format

A data frame with 16 rows and 14 columns:

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`year` The year of the observation.

`avg_hourly_demand` The average hourly electricity demand (in megawatts) for the respective year.

`population` The total population in the respective year.

`GDP` Gross Domestic Product (in constant 2015 USD) in the respective year.

`industrial_value_added` The percentage of GDP attributed to industrial value-added activities.

`manufacturing_value_added` The percentage of GDP attributed to manufacturing value-added activities.

`GDP_growth` The GDP growth rate (in percentage) for the respective year.

`GDP_deflator` The GDP deflator (in percentage), which measures price inflation or deflation.

`service_value_added` The percentage of GDP attributed to service sector value-added activities.

`GNI` Gross National Income (in constant 2015 USD) in the respective year.

`household_consumption_expenditure` The percentage of GDP attributed to household consumption expenditure.

`rural_population` The rural population in the respective year.

`example` A boolean indicator to mark this dataset as an example dataset.

Source

World Development Indicators (WDI) of the World Bank (https://databank.worldbank.org/source/world-development-indicators); Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/); ENTSO-E Power Stats Archive (https://www.entsoe.eu/data/power-stats/)

example_longterm_data *Example Longterm Data*

Description

This dataset contains the average hourly electricity demand per year for France from 2006 until 2021. It is an extension of the long-term component of example_decomposed_data with historical data from the ENTSO-E Power Stats archive (https://www.entsoe.eu/data/power-stats/).

Usage

```
example_longterm_data
```

Format

A data frame with 16 rows and 4 columns:

country The country, represented by the ISO2C country code (e.g., FR for France).

year The year of the observation.

avg_hourly_demand The average hourly electricity demand (in megawatts) for each year.

example A boolean indicator to mark this dataset as an example dataset.

Source

Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/) ; ENTSO-E Power Stats Archive (https://www.entsoe.eu/data/power-stats/)

example_longterm_future_macro_data
Example Longterm Future Macro Data

Description

This dataset extends the macro-economic data from example_longterm_predictions until the year 2028. The macro-econmic data for the years 2023 until 2028 is derived from the World Economic Outlook Database (April 2023 edition) of the International Monetary Fund (IMF) (https://www.imf.org/en/Publications
database/2023/October).

Usage

```
example_longterm_future_macro_data
```

Format

A data frame with 23 rows and 18 columns:

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`year` The year of the observation.

`avg_hourly_demand` The average hourly electricity demand (in megawatts) for the respective year.

`population` The total population in the respective year.

`GDP` Gross Domestic Product (in constant 2015 USD) in the respective year.

`industrial_value_added` The percentage of GDP attributed to industrial value-added activities.

`manufacturing_value_added` The percentage of GDP attributed to manufacturing value-added activities.

`GDP_growth` The GDP growth rate (in percentage) for the respective year.

`GDP_deflator` The GDP deflator (in percentage), which measures price inflation or deflation.

`service_value_added` The percentage of GDP attributed to service sector value-added activities.

`GNI` Gross National Income (in constant 2015 USD) in the respective year.

`household_consumption_expenditure` The percentage of GDP attributed to household consumption expenditure.

`rural_population` The rural population in the respective year.

`longterm_model_predictions1` Predictions for the long-term trend component of electricity demand based on Model 1.

`longterm_model_predictions2` Predictions for the long-term trend component of electricity demand based on Model 2.

`longterm_model_predictions3` Predictions for the long-term trend component of electricity demand based on Model 3.

`test_set_steps` The number of years used in the test or validation set for the model evaluation.

`example` A boolean indicator to mark this dataset as an example dataset.

Source

World Economic Outlook Database (April 2023 edition) of the International Monetary Fund (IMF) (https://www.imf.org/en/Publications/WEO/weo-database/2023/October); World Development Indicators (WDI) of the World Bank (https://databank.worldbank.org/source/world-development-indicators); Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/); ENTSO-E Power Stats Archive of https://www.entsoe.eu/data/power-stats/

example_longterm_future_predictions
Example Longterm Future Predictions Data

Description

This dataset extends the long-term trend from example_longterm_predictions until the year 2028.

Usage

```
example_longterm_future_predictions
```

Format

A data frame with 23 rows and 18 columns:

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`year` The year of the observation.

`avg_hourly_demand` The average hourly electricity demand (in megawatts) for the respective year.

`population` The total population in the respective year.

`GDP` Gross Domestic Product (in constant 2015 USD) in the respective year.

`industrial_value_added` The percentage of GDP attributed to industrial value-added activities.

`manufacturing_value_added` The percentage of GDP attributed to manufacturing value-added activities.

`GDP_growth` The GDP growth rate (in percentage) for the respective year.

`GDP_deflator` The GDP deflator (in percentage), which measures price inflation or deflation.

`service_value_added` The percentage of GDP attributed to service sector value-added activities.

`GNI` Gross National Income (in constant 2015 USD) in the respective year.

`household_consumption_expenditure` The percentage of GDP attributed to household consumption expenditure.

`rural_population` The rural population in the respective year.

`longterm_model_predictions1` Predictions for the long-term trend component of electricity demand based on Model 1.

`longterm_model_predictions2` Predictions for the long-term trend component of electricity demand based on Model 2.

`longterm_model_predictions3` Predictions for the long-term trend component of electricity demand based on Model 3.

`test_set_steps` The number of years used in the test or validation set for the model evaluation.

`example` A boolean indicator to mark this dataset as an example dataset.

@source Demand predictions until 2021: long_term_lm; Demand predictions from 2022-2028: long_term_future ;World Economic Outlook Database (April 2023 edition) of the International Monetary Fund (IMF) (https://www.imf.org/en/Publications/WEO/weo-database/2023/October); World Development Indicators (WDI) of the World Bank (https://databank.worldbank.org/source/world-development-indicators); Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/); ENTSO-E Power Stats Archive of https://www.entsoe.eu/data/power-stats/

`example_longterm_predictions`
Example Longterm Predictions Data

Description

This dataset extends the long-term trend component and the macro-economic data from example_longterm_and_macro_dat with the prediction results of the three best derived trend models.

Usage

```
example_longterm_predictions
```

Format

A data frame with 16 rows and 18 columns:

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`year` The year of the observation.

`avg_hourly_demand` The average hourly electricity demand (in megawatts) for the respective year.

`population` The total population in the respective year.

`GDP` Gross Domestic Product (in constant 2015 USD) in the respective year.

`industrial_value_added` The percentage of GDP attributed to industrial value-added activities.

`manufacturing_value_added` The percentage of GDP attributed to manufacturing value-added activities.

`GDP_growth` The GDP growth rate (in percentage) for the respective year.

`GDP_deflator` The GDP deflator (in percentage), which measures price inflation or deflation.

`service_value_added` The percentage of GDP attributed to service sector value-added activities.

`GNI` Gross National Income (in constant 2015 USD) in the respective year.

`household_consumption_expenditure` The percentage of GDP attributed to household consumption expenditure.

`rural_population` The rural population in the respective year.

`longterm_model_predictions1` Predictions for the long-term trend component of electricity demand based on Model 1.

longterm_model_predictions2 Predictions for the long-term trend component of electricity demand based on Model 2.

longterm_model_predictions3 Predictions for the long-term trend component of electricity demand based on Model 3.

test_set_steps The number of years used in the test or validation set for the model evaluation.

example A boolean indicator to mark this dataset as an example dataset.

Source

Demand predictions: long_term_lm ;World Development Indicators (WDI) of the World Bank (https://databank.worldbank.org/source/world-development-indicators); Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/); ENTSO-E Power Stats Archive (https://www.entsoe.eu/data/power-stats/)

---

example_midterm_demand_and_weather_data

Example Midterm Demand and Weather Data

Description

This dataset extends the example_midterm_demand_data by adding a weighted average temperature column. The dataset is divided into two parts: demand and temperature_data. The demand dataframe contains the added weighted average temperature column and the other demand related data. The temperature_data dataframe contains the daily temperature observations for the 20 most populated regions. This data is provided to show from which locations the weather data was taken.

Usage

```
example_midterm_demand_and_weather_data
```

Format

A list containing two data frames:

demand A data frame with 1,825 rows and 10 columns, representing mid-term electricity demand data.

temperature_data A data frame with 1,826 rows and 22 columns, representing temperature measurements across multiple cities.

Demand Data Frame:

demand.country The country, represented by the ISO2C country code (e.g., FR for France).

demand.date The date of the demand measurement (in YYYY-MM-DD format).

demand.year The respective year.

demand.month The respective month.

demand.day The respective day.

demand.wday The type of weekday (e.g., Sun, Mon)

demand.avg_hourly_demand The average hourly electricity demand (in megawatts) for each day.

demand.seasonal_avg_hourly_demand The seasonal mid-term component of the demand (in megawatts).

demand.weighted_temperature The weighted average temperature for France on that day (in degrees Celsius).

example A boolean indicator to mark this dataset as an example dataset.

Temperature Data Frame:

date The date of the temperature observation (in YYYY-MM-DD format).

Paris, Marseille, Lyon, Toulouse, Nice, Nantes, Montpellier, Strasbourg, Bordeaux,Cergy-Pontoise,Toulon,Reims The daily average temperature readings (in degrees Celsius) for various cities or city districts on that date.

weighted_mean_temperature The weighted (by share of population) mean temperature across the country for the respective date.

Source

demand data: Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/); holidays: https://date.nager.at/api/v3/publicholidays/ ; area population: https://wft-geo-db.p.rapidapi.com ; daily average temperatures: https://meteostat.p.rapidapi.com;

---

example_midterm_demand_data

Example Midterm Demand Data This dataset contains the seasonal mid-term demand (the difference between the yearly average hourly electricity demand and the daily average hourly electricity demand) for each day for France from 2017 until 2021.

Description

Example Midterm Demand Data This dataset contains the seasonal mid-term demand (the difference between the yearly average hourly electricity demand and the daily average hourly electricity demand) for each day for France from 2017 until 2021.

Usage

```
example_midterm_demand_data
```

Format

A data frame with 1,825 rows and 10 columns:

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`date` The date of the mid-term demand measurement (in `YYYY-MM-DD` format).

`year` The year of the observation.

`month` The month of the observation.

`day` The day of the month for the observation.

`wday` The day of the week for the observation (where 1 represents Sunday and 7 represents Saturday).

`avg_hourly_demand` The average hourly electricity demand (in megawatts) for the day.

`seasonal_avg_hourly_demand` The seasonal average hourly demand (in megawatts) for the day.

`holiday` Indicates whether the day is a public holiday (`1` for holiday, `0` for non-holiday).

`example` A boolean indicator to mark this dataset as an example dataset.

Source

demand data: Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/); holidays: https://date.nager.at/api/v3/publicholidays/

`example_midterm_future_predictions` *Example Midterm Future Predictions Data*

Description

This dataset extends the mid-term electricity demand predictions from `example_midterm_predictions` until the year 2028.

Usage

```
example_midterm_future_predictions
```

Format

A data frame with 4,380 rows and 46 columns:

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`date` The date (in `YYYY-MM-DD` format).

`year` The respective year.

`month` The respective month.

`day` The respective day.

`wday` The type of weekday (e.g., `Sun`, `Mon`)

`avg_hourly_demand` The average hourly electricity demand (in megawatts) for the day.

`seasonal_avg_hourly_demand` The seasonal average hourly demand (in megawatts) for the day.

`holiday` Indicates whether the day is a public holiday (`1` for holiday, `0` for non-holiday).

`weighted_temperature` The weighted average temperature for France on that day (in degrees Celsius).

`Jan`, `Feb`, `Mar`, `Apr`, `May`, `Jun`, `Jul`, `Aug`, `Sep`, `Nov`, `Dec` Monthly dummy variables for January through December, indicating the respective month (`1` if the date belongs to the month, `0` otherwise).

`Sun`, `Mon`, `Tue`, `Wed`, `Thu`, `Fri`, `Sat` Weekly dummy variables for Sunday through Saturday, indicating the respective weekday (`1` if the date is the specific weekday, `0` otherwise).

`HD` The weighted temperature converted to heating degree days.

`CD` The weighted temperature converted to cooling degree days.

`HD2` The squared heating degree days (HD).

`HD3` The cubed heating degree days (HD).

`CD2` The squared cooling degree days (CD).

`CD3` The cubed cooling degree days (CD).

`weighted_temperature2` The squared weighted temperature.

`weighted_temperature3` The cubed weighted temperature.

`HDlag1` Lagged value of heating degree days (1 day).

`HDlag2` Lagged value of heating degree days (2 days).

`CDlag1` Lagged value of cooling degree days (1 day).

`CDlag2` Lagged value of cooling degree days (2 days).

`weighted_temperaturelag1` Lagged weighted temperature (1 day).

`weighted_temperaturelag2` Lagged weighted temperature (2 days).

`midterm_model_fit` model predictions for the seasonal mid-term component.

`end_of_year` Binary dummy variable to account for lower demand between Christmas and New Year's Evening. Starts at 22nd December.

`test_set_steps` Number of days used in the test set for model evaluation.

`example` A boolean indicator to mark this dataset as an example dataset.

Source

demand data: Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/); holidays: https://date.nager.at/api/v3/publicholidays/ ; area population: https://wft-geo-db.p.rapidapi.com ; daily average temperatures: https://meteostat.p.rapidapi.com;

example_midterm_predictions

Example Midterm Predictions Data

Description

This dataset extends the demand dataframe from example_midterm_demand_and_weather_data with the prediction results from the best derived mid-term seasonality model. It also includes all used covariates for the model selection process.

Usage

```
example_midterm_predictions
```

Format

A data frame with 1,825 rows and 46 columns:

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`date` The date (in `YYYY-MM-DD` format).

`year` The respective year.

`month` The respective month.

`day` The respective day.

`wday` The type of weekday (e.g., `Sun`, `Mon`)

`avg_hourly_demand` The average hourly electricity demand (in megawatts) for the day.

`seasonal_avg_hourly_demand` The seasonal average hourly demand (in megawatts) for the day.

`holiday` Indicates whether the day is a public holiday (`1` for holiday, `0` for non-holiday).

`weighted_temperature` The weighted average temperature for France on that day (in degrees Celsius).

`Jan`, `Feb`, `Mar`, `Apr`, `May`, `Jun`, `Jul`, `Aug`, `Sep`, `Nov`, `Dec` Monthly dummy variables for January through December, indicating the respective month (`1` if the date belongs to the month, `0` otherwise).

`Sun`, `Mon`, `Tue`, `Wed`, `Thu`, `Fri`, `Sat` Weekly dummy variables for Sunday through Saturday, indicating the respective weekday (`1` if the date is the specific weekday, `0` otherwise).

`HD` The weighted temperature converted to heating degree days.

`CD` The weighted temperature converted to cooling degree days.

`HD2` The squared heating degree days (HD).

`HD3` The cubed heating degree days (HD).

`CD2` The squared cooling degree days (CD).

`CD3` The cubed cooling degree days (CD).

`weighted_temperature2` The squared weighted temperature.

`weighted_temperature3` The cubed weighted temperature.

`HDlag1` Lagged value of heating degree days (1 day).

`HDlag2` Lagged value of heating degree days (2 days).

`CDlag1` Lagged value of cooling degree days (1 day).

`CDlag2` Lagged value of cooling degree days (2 days).

`weighted_temperaturelag1` Lagged weighted temperature (1 day).

`weighted_temperaturelag2` Lagged weighted temperature (2 days).

`midterm_model_fit` model predictions for the seasonal mid-term component.

`end_of_year` Binary dummy variable to account for lower demand between Christmas and New Year's Evening. Starts at 22nd December.

`test_set_steps` Number of days used in the test set for model evaluation.

`example` A boolean indicator to mark this dataset as an example dataset.

Source

demand data: Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/); holidays: https://date.nager.at/api/v3/publicholidays/ ; area population: https://wft-geo-db.p.rapidapi.com ; daily average temperatures: https://meteostat.p.rapidapi.com;

`example_shortterm_demand_data` *Example Short-term Demand Data*

Description

This dataset contains the seasonal short-term demand (the difference between the measured hourly demand and the yearly average hourly electricity demand minus the daily average hourly electricity demand). for each hour for France from 2017 until 2021. The short-term seasonality corresponds to the intra-day pattern.

Usage

```
example_shortterm_demand_data
```

Format

A data frame with 43,800 rows and 14 columns:

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`date` The date (in `YYYY-MM-DD` format).

`year` The respective year.

`month` The respective month.

`day` The respective day.

`wday` The type of weekday (e.g., `Sun`, `Mon`)

`hour` The respective hour of the day.

`hourly_demand` The actual hourly electricity demand (in megawatts).

`hourly_demand_trend_corrected` The hourly demand corrected for long-term trends.

`yearly` The yearly average electricity demand.

`daily` The daily average electricity demand.

`hourly_demand_trend_and_season_corrected` The hourly demand corrected for both long-term trends and seasonal variations.

`holiday` A binary indicator for whether the day is a public holiday (`1` for holiday, `0` for non-holiday).

`example` A boolean indicator to mark this dataset as an example dataset.

Source

demand data: Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/); holidays: https://date.nager.at/api/v3/publicholidays/

`example_shortterm_future_predictions`

Example Shortterm Future Predictions Data

Description

This dataset extends the short-term electricity demand predictions from `example_shortterm_predictions` until the year 2028.

Usage

```
example_shortterm_future_predictions
```

Format

A data frame with 105,120 rows and 40 columns:

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`date` The date (in `YYYY-MM-DD` format).

`year` The respective year.

`month` The respective month.

`day` The respective day.

`wday` The type of weekday (e.g., `Sun`, `Mon`)

`hour` The respective hour of the day.

`hourly_demand` The actual hourly electricity demand (in megawatts).

`hourly_demand_trend_corrected` The hourly demand corrected for long-term trends.

`yearly` The yearly average electricity demand.

`daily` The daily average electricity demand.

`hourly_demand_trend_and_season_corrected` The hourly demand corrected for both long-term trends and seasonal variations.

`holiday` A binary indicator for whether the day is a public holiday (`1` for holiday, `0` for non-holiday).

`Hour0`, `Hour1`, `Hour2`, `Hour3`, `Hour4`, `Hour5`, `Hour6`, `Hour7`, `Hour8`, `Hour9`, `Hour10`, `Hour11`, `Hour12`, `Hour13`, `Hour14`, `Hour` Binary variables indicating the hour of the day, where each variable represents a specific hour (e.g., `Hour0` for 00:00 to 00:59, `Hour1` for 01:00 to 01:59, and so on up to `Hour23` for 23:00 to 23:59).

`short_term_lm_model_predictions` Model predictions for the short-term seasonality (the intra-day pattern).

`test_set_steps` Number of hours used in the test set for model evaluation.

`example` A boolean indicator to mark this dataset as an example dataset.

Source

demand data: Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/); holidays: https://date.nager.at/api/v3/publicholidays/

`example_shortterm_predictions`

Example Shortterm Predictions Data

Description

This dataset extends the data from `example_shortterm_demand_data` with the prediction results from the best derived short-term seasonality model. The only covariates used for the model selection process are the hour of the day and an indicator if its a holiday or not.

Usage

```
example_shortterm_predictions
```

Format

A data frame with 43,800 rows and 40 columns:

`country` The country, represented by the ISO2C country code (e.g., `FR` for France).

`date` The date (in `YYYY-MM-DD` format).

`year` The respective year.

`month` The respective month.

`day` The respective day.

`wday` The type of weekday (e.g., `Sun`, `Mon`)

`hour` The respective hour of the day.

`hourly_demand` The actual hourly electricity demand (in megawatts).

`hourly_demand_trend_corrected` The hourly demand corrected for long-term trends.

`yearly` The yearly average electricity demand.

`daily` The daily average electricity demand.

`hourly_demand_trend_and_season_corrected` The hourly demand corrected for both long-term trends and seasonal variations.

`holiday` A binary indicator for whether the day is a public holiday (`1` for holiday, `0` for non-holiday).

`Hour0`, `Hour1`, `Hour2`, `Hour3`, `Hour4`, `Hour5`, `Hour6`, `Hour7`, `Hour8`, `Hour9`, `Hour10`, `Hour11`, `Hour12`, `Hour13`, `Hour14`, `Hour` Binary variables indicating the hour of the day, where each variable represents a specific hour (e.g., `Hour0` for 00:00 to 00:59, `Hour1` for 01:00 to 01:59, and so on up to `Hour23` for 23:00 to 23:59).

`short_term_lm_model_predictions` Model predictions for the short-term seasonality (the intra-day pattern).

`test_set_steps` Number of hours used in the test set for model evaluation.

`example` A boolean indicator to mark this dataset as an example dataset.

Source

demand data: Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/); holidays: https://date.nager.at/api/v3/publicholidays/

`fill_missing_data` *Replace missing values in the load data set*

Description

This function substitutes missing values with the corresponding values at the same hour exactly one week prior. For example, if there is no load value available for May 12th at 20:00, the value recorded on May 7th at 20:00 will be used as a replacement. This function is primarily designed to handle minor gaps in the dataset acquired using `get_entsoE_data`. To use this function with other datasets, it is important that the input data frame adheres to the required column naming conventions.

Usage

```
fill_missing_data(load_data, data_directory = tempdir())
```

Arguments

`load_data`	Data Frame with load data. Data Frame must contain the following columns: **date** Consisting of the datetime values, date formatted (e.g. POSIXct). **load** Consisting of the load values, numeric. **unit** Indicating the measured unit (e.g., MW), character. **country** Indicating the country's ISO2C code, character.
`data_directory`	The path to the directory where the data will be saved. The default is set to a temporary directory.

Value

Data Frame with completed load values, date, unit, year, time resolution, ISO2C Country Code

Examples

```
suppressWarnings(
  library(ggplot2)
)
example_demand_data_filled <- fill_missing_data(example_demand_data)
example_df <- as.data.frame(seq.POSIXt(
  example_demand_data$date[841],
  example_demand_data$date[870], "hour"
))
example_df$before <- NA
example_df$before[example_df[, 1] %in% example_demand_data$date] <-
  example_demand_data$load[example_demand_data$date %in% example_df[, 1]]
example_df$after <- example_demand_data_filled$load[example_demand_data_filled$date
  %in% example_df[, 1]]
ggplot(example_df, aes(x = example_df[, 1])) +
  geom_line(aes(y = after, colour = "after data filling")) +
  geom_line(aes(y = before, colour = "before data filling")) +
  xlab("\nHour") +
  ylab("Load [MW]\n") +
  theme(legend.title = element_blank()) +
  scale_x_continuous(
    breaks = c(example_df[1, 1], example_df[25, 1]),
    labels = c(as.Date(example_df[1, 1]), as.Date(example_df[25, 1]))
  )
```

---

full_forecast *Title*

Description

Title

Usage

```
full_forecast(
  start_year,
  end_year_data,
  country,
  test_set_steps = 2,
  future = "yes",
  end_year = 2028,
  data_directory = tempdir(),
  verbose = FALSE
)
```

Arguments

start_year	Specifies the starting year for which predictions and models will be generated.
end_year_data	Specifies the final year for which data from the Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/) is retrieved and for which models will be generated.
country	String. Specifies the country.
test_set_steps	Integer. Specifies how many years are used for generating the test/validation set for the model selection.
future	String. Option to enable or disable the future forecasts. If set to "yes" forecasts will be made until the specified end_year. If set to anything else, forecasts will only be generated until the specified end_year_data value.
end_year	Specifies the final year for which future predictions will be generated.
data_directory	The path to the directory where the results and plots will be saved. The default is set to a temporary directory. It is highly recommended to set it to something that is not a temporary directory if you want to use this function.
verbose	A boolean value indicating if you want the generated plots and detailed status updates to be shown (set to TRUE if yes). #' @seealso See also functions long_term_lm, mid_term_lm, and short_term_lm for the prediction models and long_term_future, mid_term_future, and short_term_future for the future prediction models.

Value

Returns a list with the combined model results and plots. And a list with the results, models and plots for the long-term trend, mid-term seasonality, and short-term seasonality respectively. The combined model predictions and plots are saved in the respective folder for the country.

full_model_predictions The combined model results and plots.

longterm The long-term trend models, results, and plots.

midterm The mid-term seasonality models, results, and plots.

shortterm The short-term seasonality models, results, and plots.

Examples

```
library(ggplot2)
## Without future predictions

forecast_data <- full_forecast(
  start_year = 2017, end_year = 2021, country = "France", test_set_steps = 2,
  future = "no"
)

ggplot(example_full_model_predictions) +
  geom_line(aes(date, hourly_demand, color = "actual")) +
  geom_line(aes(date, complete_model, color = "fitted")) +
  xlab("\nYear") +
```

```
  ylab("Hourly Demand\n [MW]\n") +
  geom_vline(xintercept = example_full_model_predictions$date[26280], linetype = 2) +
  ggthemes::theme_foundation(base_size = 14, base_family = "sans") +
  xlab("\nHour") +
  ylab("Hourly Demand\n [MW]\n") +
  ggtitle(paste("Complete Model Results - FR\n")) +
  theme(
    plot.title = element_text(
      face = "bold",
      size = rel(1.2), hjust = 0.5
    ),
    text = element_text(),
    panel.background = element_rect(colour = NA),
    plot.background = element_rect(colour = NA),
    panel.border = element_rect(colour = NA),
    axis.title = element_text(face = "bold", size = rel(1)),
    axis.title.y = element_text(angle = 90, vjust = 2),
    axis.title.x = element_text(vjust = -0.2),
    axis.text = element_text(),
    axis.line.x = element_line(colour = "black"),
    axis.line.y = element_line(colour = "black"),
    axis.ticks = element_line(),
    panel.grid.major = element_line(colour = "#f0f0f0"),
    panel.grid.minor = element_blank(),
    legend.key = element_rect(colour = NA),
    legend.position = "bottom",
    legend.direction = "horizontal",
    legend.key.size = unit(0.2, "cm"),
    plot.margin = unit(c(10, 5, 5, 5), "mm"),
    strip.background = element_rect(colour = "#f0f0f0", fill = "#f0f0f0"),
    strip.text = element_text(face = "bold")
  ) +
  theme(legend.title = element_blank())

## With future predictions

forecast_data <- full_forecast(
  start_year = 2017, end_year = 2021, country = "France", test_set_steps = 2,
  future = "yes", end_year = 2028
)

suppressWarnings(
  ggplot(example_full_model_future_predictions) +
    geom_line(aes(1:nrow(example_full_model_future_predictions),
      hourly_demand,
      color = "actual"
    )) +
    geom_line(aes(1:nrow(example_full_model_future_predictions), complete_model,
      color = "fitted"
    )) +
    xlab("\nYear") +
    ylab("Hourly Demand\n [MW]\n") +
    geom_vline(xintercept = 26280, linetype = 2) +
```

```
    geom_vline(xintercept = 43800, linetype = 3) +
    ggthemes::theme_foundation(base_size = 14, base_family = "sans") +
    xlab("\nHour") +
    ylab("Hourly Demand\n [MW]\n") +
    scale_y_continuous(labels = scales::label_number(scalar = 1)) +
    ggtitle(paste("Complete Model Results - FR\n")) +
    theme(
      plot.title = element_text(
        face = "bold",
        size = rel(1.2), hjust = 0.5
      ),
      text = element_text(),
      panel.background = element_rect(colour = NA),
      plot.background = element_rect(colour = NA),
      panel.border = element_rect(colour = NA),
      axis.title = element_text(face = "bold", size = rel(1)),
      axis.title.y = element_text(angle = 90, vjust = 2),
      axis.title.x = element_text(vjust = -0.2),
      axis.text = element_text(),
      axis.line.x = element_line(colour = "black"),
      axis.line.y = element_line(colour = "black"),
      axis.ticks = element_line(),
      panel.grid.major = element_line(colour = "#f0f0f0"),
      panel.grid.minor = element_blank(),
      legend.key = element_rect(colour = NA),
      legend.position = "bottom",
      legend.direction = "horizontal",
      legend.key.size = unit(0.2, "cm"),
      plot.margin = unit(c(10, 5, 5, 5), "mm"),
      strip.background = element_rect(colour = "#f0f0f0", fill = "#f0f0f0"),
      strip.text = element_text(face = "bold")
    ) +
    theme(legend.title = element_blank()) +
    scale_x_continuous(
    breaks = c(1, 8761, 17521, 26281, 35041, 43801, 52561, 61321, 70081, 78841, 87601, 96361),
      labels = c(2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028)
    ) +
   annotate("text", x = 13140, y = 99216.6, label = "Training", size = 4, hjust = 0.5, vjust = 0) +
   annotate("text", x = 35040, y = 99216.6, label = "Test", size = 4, hjust = 0.5, vjust = 0) +
   annotate("text", x = 74460, y = 99216.6, label = "Unknown", size = 4, hjust = 0.5, vjust = 0)
)
```

`get_entsoE_data` *Load data from the ENTSO-E Transparency Platform*

Description

This function makes various API requests to the Transparency Platform of the European Network of Transmission System Operators for Electricity (ENTSO-E, https://transparency.entsoe.eu/) and stores the downloaded load data in a data frame. The earliest possible year for the requested load time series is 2017.

Usage

```
get_entsoE_data(
  start_year,
  end_year,
  country,
  api_key = "default",
  dry_run = FALSE
)
```

Arguments

start_year	Numeric. The starting year for which load data will be requested.
end_year	Numeric. The final year for which load data will be requested.
country	Character. The country name for which load data will be requested provided as the English name of the country.
api_key	Character. A valid API key for the ENTSO-E Transparency Platform. If set to "default", one of the deposited keys will be used.
dry_run	Boolean. Defaults to FALSE. This is only set to TRUE for the example run.

Value

A Data Frame with the following columns

date The series of dates, POSIXct format.

load The series of load data, numeric

unit The series of units in which the load data is provided, character.

year The year of each load data point, numeric

time_interval The time resolution of each load data point, character

country The ISO2C Country Code, character

Examples

```
example_demand_data <- get_entsoE_data(2017, 2021, "France", api_key = "default", dry_run = TRUE)
print(example_demand_data[1:20, ])
```

---

get_historic_load_data

Load historic yearly average load data

Description

This function utilizes two historical load datasets obtained from the archive of https://www.entsoe.eu/data/power-stats/. It calculates the hourly average load for each year and appends the results to the long-term dataframe generated by decompose_load_data.

Usage

```
get_historic_load_data(longterm)
```

Arguments

longterm	A data frame object with "country", "year", and "avg_hourly_demand" columns resulting from the function decompose_load_data.

Value

Data frame with "country", "year", and "avg_hourly_demand" (yearly average of hourly demand).

Examples

```
print("Input dataframe should be in the following form:")
example_decomposed_data$longterm
example_longterm_data <- get_historic_load_data(example_decomposed_data$longterm)
print("Historic load data is added starting from 2006.")
example_longterm_data
```

---

get_macro_economic_data

Load a list of macroeconomic data from WDI

Description

This function downloads a set of ten macroeconomic variables via API from the World Development Indicators (WDI) of the World Bank (https://databank.worldbank.org/source/world-development-indicators). The variables are suspected to have a predictive capacity for the load data.

Usage

```
get_macro_economic_data(longterm_data)
```

Arguments

longterm_data	The long-term data series resulting from the function decompose_load_data. Contains information on country (longterm$country) and years (longterm$year).

Value

Data frame with the original time series and 10 additional columns with macroeconomic indicators.

See Also

See function decompose_load_data for the generation of the long-term series and long_term_lm for the selection of covariates.

Examples

```
example_longterm_and_macro_data <- get_macro_economic_data(example_longterm_data)
print("Macro economic variables are added from the World Bank Developer Indicators:")
example_longterm_and_macro_data
```

`get_weather_data` *Load weather data via API*

Description

This function loads weather data which is used to forecast the mid-term load seasonalities. First the 20 most populated areas in the country are obtained from https://wft-geo-db.p.rapidapi.com . Then the closest weather stations of each area are identified and average daily temperature values are downloaded from https://meteostat.p.rapidapi.com for the provided time period. From this data a weighted daily average temperature based on population is calculated for the provided country.

Usage

```
get_weather_data(
  midterm_demand_data,
  api_key = "default",
  data_directory = tempdir()
)
```

Arguments

`midterm_demand_data`	Dataframe. The mid-term data series from `decompose_load_data` with added holidays resulting from the function `add_holidays_mid_term`.
`api_key`	Character. A valid API key from rapidapi that is subscribed to wft-geo-db and meteostat. If set to "default", one of the deposited keys will be used.
`data_directory`	The path to the directory where the data will be saved.

Value

A list containing the mid-term data and temperature data.

See Also

See function `decompose_load_data` for the generation of the mid-term series.

Examples

```
example_midterm_demand_and_weather_data <- get_weather_data(example_midterm_demand_data,
  api_key = "default"
)
head(example_midterm_demand_and_weather_data$demand)
head(example_midterm_demand_and_weather_data$temperature_data)
```

`long_term_future` *Long-term trend predictions for future years*

Description

This function extends the long-term trend predictions generated by `long_term_lm` until a specified future year. The unknown macro-economic covariates are either obtained from the WEO Outlook with `long_term_future_data` or can be supplied manually. The function also produces and saves visualizations of the actual and the predicted demand over the training, test, and future periods.

Usage

```
long_term_future(
  longterm_future_macro_data,
  data_directory = tempdir(),
  model_list = NULL,
  verbose = FALSE
)
```

Arguments

`longterm_future_macro_data`	Dataframe. Generated by `long_term_future_data`
`data_directory`	The path to the directory where the data will be saved and where the function will look for the long-term models from `long_term_lm`. The default is set to a temporary directory.
`model_list`	A list with the models from `long_term_lm`. Only needs to be specified if the models are not in the data directory.
`verbose`	A boolean value indicating if you want the generated plots to be shown (set to TRUE if yes).

Value

A list with the extended initial dataframe with the future predictions for each of the 3 best long term models and one plot with the respective results per model.

longterm_future_predictions A dataframe with the input data and additional columns for test_set_steps and for best three models longterm_model_predictions1, longterm_model_predictions2 and longterm_model_predictions3.

logterm_future_plots A list with the respective plots for each model.

See Also

See also function `mid_term_future` and `short_term_future` for the other prediction models and `long_term_future_data` for the covariate download.

Examples

```
example_longterm_future_predictions <- long_term_future(example_longterm_future_macro_data)
```

`long_term_future_data` *Get future predictions for the macro economic covariates*

Description

This function gets predictions from the World Economic Outlook Database (April 2025 edition) for the covariates utilized in the long-term models. If the dataset argument is not set to "WEO", the function will indicate which covariates require user-supplied predictions for `long_term_future`.

Usage

```
long_term_future_data(
  longterm_predictions,
  end_year,
  dataset = "WEO",
  data_directory = tempdir(),
  model_list = NULL
)
```

Arguments

`longterm_predictions`	Dataframe or list. Generated by `long_term_lm`. Either the prediction dataframe or the complete output list can be used. If the full list is supplied the function will extract the necessary models automatically.
`end_year`	Integer. Specifies the final year for which future predictions will be generated.
`dataset`	Character. By default the World Economic Outlook (WEO) Database April 2025 edition is used to generate covariate predictions for the long term models. If the dataset option is set to anything else than "WEO" the function will tell you for which covariates data will be needed to run `long_term_future`.
`data_directory`	The path to the directory where the function will look for the long-term models from `long_term_lm`. Only needed if dataset is not set to "WEO" and no model list is supplied.
`model_list`	A list with the models from `long_term_lm`. Only needs to be specified if dataset is not set to "WEO" and if the models are not in the data directory.

Value

The extended initial dataframe until the specified end year with the covariate predictions if dataset = "WEO" . Otherwise, the dataframe will have empty covariate predictions and a note for which covariates a forecast must be supplied.

Examples

```
example_longterm_future_macro_data <- long_term_future_data(example_longterm_predictions,
  end_year = 2028, dataset = "WEO"
)
```

`long_term_lm` *Long-term forecast*

Description

This function predicts the long-term load data based on the provided time series and a set of macroeconomic variables.

Usage

```
long_term_lm(
  longterm_and_macro_data,
  test_set_steps = 2,
  testquant = 500,
  rdm_seed = sample(1:10000, 1),
  data_directory = tempdir(),
  verbose = FALSE
)
```

Arguments

`longterm_and_macro_data`	Dataframe. Containing the load data and macroeconomic indicators derived from `get_macro_economic_data`.
`test_set_steps`	Integer. Number of time periods in the test set.
`testquant`	Integer. Determines how many of the best ranked models are evaluated with cross validation.
`rdm_seed`	A random seed to keep results consistent
`data_directory`	The path to the directory where the data, plots, and models will be saved. The default is set to a temporary directory.
`verbose`	A boolean value indicating if you want the generated plots to be shown (set to TRUE if yes).

Details

The model corresponds to

$$\bar{D}_L(t_L) = \beta_{L,1} + \beta_{L,2}x_1(t_L) + ... + \beta_{L,10}x_{10}(t_L)\epsilon_L(t_L).$$

where the covariates correspond to the loaded macroeconomic variables from `get_macro_economic_data`. The three best models out of all possible covariate combinations are chosen and saved. The predicted and actual time series of the three best models are plotted and saved as well.

Value

A list with the dataframe with the input data and results. A list with the plotted results of the three best models. And a list with the three best models. The dataset, plots, and the models are saved in the respective folder for the country.

longterm_predictions A dataframe with the input data and additional columns for test_set_steps and for best three models longterm_model_predictions1, longterm_model_predictions2 and longterm_model_predictions3.

longterm_plots A list with the respective plots for each model.

longterm_models A list with the three best models.

See Also

See also function `mid_term_lm` and `short_term_lm` for the other prediction models and `get_macro_economic_data` for the covariate download.

Examples

```
example_longterm_predictions <- long_term_lm(example_longterm_and_macro_data,
  test_set_steps = 2, testquant = 500, rdm_seed = 421
)
```

---

mid_term_future *Generate future mid-term demand predictions*

Description

This function extends the mid-term demand predictions generated by `mid_term_lm` until a specified future year. The unknown temperature-based covariates for future days are obtained by averaging over the past 3 years of the dataset. The function also produces and saves visualizations of the actual and the predicted demand over the training, test, and future periods.

Usage

```
mid_term_future(
  midterm_predictions,
  end_year,
  Tref = 18,
  data_directory = tempdir(),
  midterm_model = NULL,
  verbose = FALSE
)
```

Arguments

midterm_predictions	Dataframe or list. Generated by mid_term_lm. Either the prediction dataframe or the complete output list can be used. If the full list is supplied the function will extract the necessary models automatically.
end_year	Integer. Specifies the final year for which future predictions will be generated.
Tref	Numeric. Reference temperature as basis for the calculation of cooling and heating days.
data_directory	The path to the directory where the data will be saved and where the function will look for the mid-term model from mid_term_lm. The default is set to a temporary directory.
midterm_model	The mid-term seasonality model from mid_term_lm. Only needs to be specified if the model is not in the data directory.
verbose	A boolean value indicating if you want the generated plot to be shown (set to TRUE if yes).

Value

A list with the extended initial dataframe with the future predictions for the mid term model. And the plot with the midterm seasonality future forecast. The dataset and the plot are saved in the respective folder for the country.

midterm_future_predictions A dataframe with the input and prediction data for the future mid-term seasonality.

midterm_future_plot A plot with the prediction results.

See Also

See also function long_term_future and short_term_future for the other prediction models.

Examples

```
example_midterm_future_predictions <- mid_term_future(example_midterm_predictions,
  end_year = 2028, Tref = 18
)
```

---

mid_term_lm *Mid-term forecast*

Description

The mid-term load series is forecasted based on the provided load time series and weather data. The prediction is either based on the (lagged) temperature data in combination with transformed variables for heating and cooling days or on a spline regression applied on the temperature data to account for non-linear effects.

Usage

```
mid_term_lm(
  demand_and_weather_data,
  Tref = 18,
  test_set_steps = 730,
  method = "temperature transformation",
  data_directory = tempdir(),
  verbose = FALSE
)
```

Arguments

demand_and_weather_data	Dataframe. Containing the mid-term load data, the holidays and weather data obtained from get_weather_data.
Tref	Numeric. Reference temperature as basis for the calculation of cooling and heating days.
test_set_steps	Integer. Number of time periods in the test set.
method	String. Indicates which model selection process is used. If method="temperature transformation", the temperature values are transformed to heating and cooling degree days to capture the non-linear relationship of temperature and electricity demand. If the method is set to "spline" a spline regression is instead used without the transformation of the temperature data.
data_directory	The path to the directory where the data, plots, and models will be saved. The default is set to a temporary directory.
verbose	A boolean value indicating if you want the generated plots to be shown (set to TRUE if yes).

Value

A list with the dataframe with the input data and results. The plot with the midterm seasonality forecast. And the midterm model. The dataset, the plot, and the model are saved in the respective folder for the country.

midterm_predictions A dataframe with the input and prediction data for the mid-term seasonality.

midterm_plot A plot with the prediction results.

midterm_model The mid-term seasonality model.

Examples

```
example_midterm_predictions <- mid_term_lm(example_midterm_demand_and_weather_data$demand,
  Tref = 18, test_set_steps = 730, method = "temperature transformation"
)
```

short_term_future *Generate future short-term demand predictions*

Description

This function extends the short-term demand predictions generated by short_term_lm until a specified future year. The function also produces and saves visualizations of the actual and the predicted demand over the training, test, and future periods.

Usage

```
short_term_future(
  shortterm_predictions,
  end_year,
  data_directory = tempdir(),
  model_list = NULL,
  verbose = FALSE
)
```

Arguments

shortterm_predictions	Dataframe. Generated by short_term_lm
end_year	Integer. Specifies the final year for which future predictions will be generated
data_directory	The path to the directory where the data will be saved and where the function will look for the short-term models from short_term_lm. The default is set to a temporary directory.
model_list	A list with the models from short_term_lm. Only needs to be specified if the models are not in the data directory.
verbose	A boolean value indicating if you want the generated plots to be shown (set to TRUE if yes).

Value

The extended initial dataframe with the future predictions for the short term model.

A list with the extended initial dataframe with the future predictions for the short term model. And the plot with the shortterm seasonality future forecast. The dataset and the plot are saved in the respective folder for the country.

shortterm_future_predictions A dataframe with the input and prediction data for the future short-term seasonality.

shortterm_future_plot A plot with the prediction results.

See Also

See also function long_term_future and mid_term_future for the other prediction models.

Examples

```
example_shortterm_future_predictions <- short_term_future(example_shortterm_predictions,
  end_year = 2028
)
```

`short_term_lm` *Short-term forecast*

Description

The short-term load series is forecasted based on the provided hourly load data.

Usage

```
short_term_lm(
  shortterm_demand_data,
  test_set_steps = 17520,
  data_directory = tempdir(),
  verbose = FALSE
)
```

Arguments

`shortterm_demand_data`	Dataframe. Containing the short-term load data from `decompose_load_data` and the added holiday dummy resulting from `add_holidays_short_term`.
`test_set_steps`	Integer. Number of hours used for the test set. The default value of 17520 equals two years (2 * 8760 hours).
`data_directory`	The path to the directory where the data, plots, and models will be saved. The default is set to a temporary directory.
`verbose`	A boolean value indicating if you want the generated plots to be shown (set to TRUE if yes).

Value

A list with the dataframe with the input data and results. A list with the plotted result for the complete timeseries and two sample weeks. And a list with the the best model for each type of month and type of day (84 in total with 12 different months times 7 types of weekdays). The dataset, plots, and the models are saved in the respective folder for the country.

shortterm_predictions A dataframe with the input data and additional columns for the respective hour, test_set_steps, and for the model predictions.

shortterm_plots A list with the full plot and a plot for two sample weeks.

shortterm_models A list with the respective models for each month and type of day.

Examples

```
example_shortterm_predictions <- short_term_lm(example_shortterm_demand_data)
```

`weo_data` *World Economic Outlook (WEO) Data*

Description

This dataset contains macroeconomic data and projections from the International Monetary Fund (IMF) World Economic Outlook (WEO). It includes annual data for multiple countries and economic indicators, spanning from 1980 to 2030.

Usage

```
weo_data
```

Format

A data frame with 588 rows and 61 columns:

`WEO Country Code` Unique code assigned to each country by the WEO database.

`ISO` The ISO3 country code (e.g., `FRA` for France).

`WEO Subject Code` Unique code representing the economic indicator or subject in the WEO database.

`Country` The name of the country.

`Subject Descriptor` Description of the economic indicator.

`Subject Notes` Additional notes or details about the economic indicator.

`Units` The unit of measurement for the indicator (e.g., percentage, persons, national currency).

`Scale` The scaling factor for the indicator values (e.g., "Billions").

`Country/Series-specific Notes` Country-specific notes about the data series.

`1980`, `1981`, `1982`, `1983`, `1984`, `1985`, `1986`, `1987`, `1988`, `1989`, `1990`, `1991`, `1992`, `1993`, `1994`, `1995`, `1996`, `1997`, `1998`, `1999`
: The annual value of the economic indicator for the respective year, starting from 1980 (`1980`) to 2028 (`2028`).

`Estimates Start After` The year after which data values are based on projections instead of past values.

Source

World Economic Outlook Database (April 2025 edition) of the International Monetary Fund (IMF) (https://www.imf.org/en/Publications/WEO/weo-database/2023/October)

Index