

Package ‘LandUseQuantifieR’

October 8, 2019

Title This package is an (adapted) R implementation of the circle diagram approach from Hughes et al. 2018

Version 0.0.1.1000

Description This package is an (adapted) R implementation of the circle diagram approach from Hughes et al. 2018;

Depends R (>= 3.4)

Imports magrittr,
dplyr,
tibble,
tidyr,
utils

License GPL-3 + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests knitr,
rmarkdown,
ggplot2,
cowplot,
RColorBrewer,
grDevices

VignetteBuilder knitr

R topics documented:

animals	3
animal_products	3
arbori_selector_kcal	4
arbori_selector_protein	5
area_village	5
aux_animals	6
bronze_production	6
ceramics_production	7
cheese	7
cheese_selector_kcal	8
cheese_selector_protein	8

crop_selector_kcal	9
crop_selector_protein	9
data_checker_kcal	10
data_checker_protein	10
flax_production	11
game	11
game_selector_kcal	12
game_selector_protein	12
iron_production	13
kcal_req_age	13
kcal_req_detailed	14
kcal_req_human	14
milk	15
milk_selector_kcal	16
milk_selector_protein	16
nd_herd	17
nd_herd_protein	17
pasture_selector_kcal	18
pasture_selector_protein	18
plants	19
protein_req_age	20
protein_req_age_long	20
setup_auxiliary_animals_kcal	21
setup_auxiliary_animals_protein	21
setup_bronze	22
setup_ceramics_kcal	23
setup_ceramics_protein	23
setup_diet_arbori_gather	24
setup_diet_arbori_gather_protein	25
setup_diet_cheese	25
setup_diet_cheese_protein	26
setup_diet_crops	27
setup_diet_crops_protein	27
setup_diet_game	28
setup_diet_game_protein	29
setup_diet_milk	29
setup_diet_milk_protein	30
setup_diet_pasture	31
setup_diet_pasture_protein	31
setup_flax	32
setup_iron	32
setup_olive_oil	33
setup_population	34
setup_population_detailed_kcal	34
setup_population_detailed_protein	35
setup_protein_req	36
setup_wine	36
setup_wine_protein	37
setup_wood_consumption	37
spelling_checker_kcal	38
spelling_checker_protein	38
summarize_areas_kcal	39

<i>animals</i>	3
summarize_areas_protein	40
wood_cons	41
wood_prod	41

Index	42
--------------	-----------

<i>animals</i>	<i>Contains informations of the animal data</i>
----------------	---

Description

Contains informations of the animal data

Usage

`animals`

Format

A data.frame with 14 rows and 10 variables:

animal The type of animal

type The subtype of the animal

calory_requirement The caloric requirements in kcal/kg

meat_yield_factor The factor of the calculation of the animals weight that remains as meat

meat_yield The meat yield in kg; calculated as `weight * meat_yield_factor`

reference The references for the animal data

herd_composition The composition of the herd of each animal

reference_composition The references of the herd composition

slaughter_factor The slaughter factor

reference_sf The reference for the slaughter factor

```
animals <- read.table("./animals.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE) usethis::use_data(animals,
overwrite = TRUE)
```

<i>animal_products</i>	<i>Contains informations of the animal products data</i>
------------------------	--

Description

Contains informations of the animal products data

Usage

`animal_products`

Format

A data.frame with 15 rows and 6 variables:

animal The type of animal

product The specific animal product

calories The caloric content of the animal product in kcal/kg

protein The protein content of the animal product in g/kg

reference_calories The references of the caloric content

reference_protein The reference of the protein content

```
animal_products <- read.table("./animal_products.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE)
usethis::use_data(animal_products, overwrite = TRUE)
```

arbori_selector_kcal	<i>arbori_selector_kcal</i>
----------------------	-----------------------------

Description

Combining, filtering and selecting data sets

Usage

```
arbori_selector_kcal(food_df)
```

Arguments

food_df The consumed food and the proportion in % of the consumed food

Details

This small function called `arbori_selector_kcal` adds new variables to one dataset from another dataset, filters the data input and selects the references for the users data input from the packages data

Value

Parameter `food_df`

```
arbori_selector_protein
      abori_selector_protein
```

Description

Combining, filtering and selecting data sets

Usage

```
arbori_selector_protein(food_df)
```

Arguments

food_df The consumed food and the proportion in % of the consumed food

Details

This small function called abori_selector adds new variables to one dataset from another dataset, filters the data input and selects the references for the users data input from the packages data

Value

Parameter food_df

```
area_village                      area_village
```

Description

Calculates the area of a village

Usage

```
area_village(person_per_hectar = 200, population = 3800)
```

Arguments

person_per_hectar The amount of people per hectar or population density; default 200, based on Jameson et al. 1994

population The amount of people living in the village; default 3800, based on Jameson et al. 1994

Details

Calculates the area of the village in hectares based on the amount of inhabitants and a measure of their density

Value

A numeric, i.e. the size of a village in hectar

aux_animals	<i>Contains informations of the auxiliary animals</i>
-------------	---

Description

Contains informations of the auxiliary animals

Usage

```
aux_animals
```

Format

A data.frame with 3 rows and 1 variable:

type The type of auxiliary animal

Area_ploughed_day_ha The Area that can be ploughed per day and team of animals in ha

calory_req_day_individual The amount of calories needed to feed one individual per day; kcal

```
aux_animals <- read.table("./aux_animals.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE)
usethis::use_data(aux_animals, overwrite = TRUE)
```

bronze_production	<i>Contains informations of the wood requirements for bronze production</i>
-------------------	---

Description

Contains informations of the wood requirements for bronze production

Usage

```
bronze_production
```

Format

A data.frame with 1 row and 4 variables:

energy_req_melt_copper The amount of energy required to melt copper in KJ; default 60000 KJ

energy_req_melt_tin The amount of energy required to melt tin in KJ; default 59000 KJ

energy_req_melt_metals The amount of energy required to melt metal in KJ; default 59000 KJ

weight_melted_bronze_person The amount of weighted metal per person (kg/capita); default 2

```
bronze_production <- read.table("./bronze_production.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE)
usethis::use_data(bronze_production, overwrite = TRUE)
```

ceramics_production	<i>Contains informations of the ceramics production</i>
---------------------	---

Description

Contains informations of the ceramics production

Usage

```
ceramics_production
```

Format

A data.frame with 1 row and 5 variables:

vessels_person The number of vessels per person

mean_weight_vessel_kg The mean weight of a vessel in kg

mean_weight_wine_amphora_kg The mean weight of a wine amphora in kg

mean_weight_oil_amphora_kg The mean weight of an oil amphora in kg

ratio_wood_ceramics The ratio of wood to ceramics

```
ceramics_production <- read.table("./ceramics_production.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE)
usethis::use_data(ceramics_production, overwrite = TRUE)
```

cheese	<i>Contains informations of the cheese data</i>
--------	---

Description

Contains informations of the cheese data

Usage

```
cheese
```

Format

A data.frame with 13 rows and 8 variables:

product The type of cheese

calories_per_kg The amount of calories per kg cheese

reference_calories The references

protein The amount of protein per kg cheese

reference_protein The references

milk_per_indiv_and_year The amount of milk for human consumption per individual and year

liter_milk_needed_for_one_kg_cheese The amount of milk needed to produce 1 kg of cheese

calory_req_day_individual The amount of calories needed to feed one individual per day; kcal

```
cheese <- read.table("./cheese.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE)
usethis::use_data(cheese, overwrite = TRUE)
```

cheese_selector_kcal	<i>cheese_selector_kcal</i>
----------------------	-----------------------------

Description

Combining and selecting data sets

Usage

```
cheese_selector_kcal(food_df)
```

Arguments

food_df	The consumed food and the proportion in % of the consumed food
---------	--

Details

This small function called `cheese_selector_kcal` adds new variables to one dataset from another dataset and selects the references for the users data input from the packages data

Value

Parameter `food_df`

cheese_selector_protein	<i>cheese_selector_protein</i>
-------------------------	--------------------------------

Description

Combining and selecting data sets

Usage

```
cheese_selector_protein(food_df)
```

Arguments

food_df	The consumed food and the proportion in % of the consumed food
---------	--

Details

This small function called `cheese_selector_protein` adds new variables to one dataset from another dataset and selects the references for the users data input from the packages data

Value

Parameter `food_df`

crop_selector_kcal	<i>crop_selector_kcal</i>
--------------------	---------------------------

Description

Combining, filtering and selecting data sets

Usage

```
crop_selector_kcal(food_df)
```

Arguments

food_df	The consumed food and the proportion in % of the consumed food
---------	--

Details

This small function called `crop_selector_kcal` adds new variables to one dataset from another dataset, filters the data input and selects the references for the users data input from the packages data

Value

Parameter `food_df`

crop_selector_protein	<i>crop_selector_protein</i>
-----------------------	------------------------------

Description

Combining, filtering and selecting data sets

Usage

```
crop_selector_protein(food_df)
```

Arguments

food_df	The consumed food and the proportion in % of the consumed food
---------	--

Details

This small function called `crop_selector_protein` adds new variables to one dataset from another dataset, filters the data input and selects the references for the users data input from the packages data

Value

Parameter `food_df`

data_checker_kcal	<i>data_checker_kcal</i>
-------------------	--------------------------

Description

Convert a data frame to a lower case

Usage

```
data_checker_kcal(food_df)
```

Arguments

food_df	The consumed food and the proportion in % of the consumed food
---------	--

Details

This is a small helper function that is called data_checker_kcal and it converts the data given in the food_df into lower case

Value

Parameter food_df

data_checker_protein	<i>data_checker_protein</i>
----------------------	-----------------------------

Description

Convert a data frame to a lower case

Usage

```
data_checker_protein(food_df)
```

Arguments

food_df	The consumed food and the proportion in % of the consumed food
---------	--

Details

This is a small helper function that is called data_checker_protein and it converts the data given in the food_df into lower case

Value

Parameter food_df

flax_production	<i>Contains informations of the flax production</i>
-----------------	---

Description

Contains informations of the flax production

Usage

```
flax_production
```

Format

A data.frame with 1 row and 4 variables:

Productivity The productivity of the flax in kg/ha; default 804

Linen_need_kg The amount of linen needed (kg); default .1

Flax_kg_in_100g_linen The amount of flax (kg) per 100g linen; default 1

PropPop The proportion of the population that uses flax; default .65

```
flax_production <- read.table("./flax_production.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)
usethis::use_data(flax_production, overwrite = TRUE)
```

game	<i>Contains informations of the game data</i>
------	---

Description

Contains informations of the game data

Usage

```
game
```

Format

A data.frame with 11 observations and 8 variables:

animal The general category of the game animal: deer, wild boar, birds, fish

type The type of the animal

weight The weight of one individual in kg

meat_yield_factor The factor of the calculation of the animals weight that remains as meat

meat_yield The meat yield in kg; calculated as weight * meat_yield_factor

calories The caloric content in kcal/kg

protein The protein content in g/kg

references_weight The references of the weight data

references_calories The references of the caloric data

references_protein The references of the protein data

```
game <- read.table("./game.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE)
usethis::use_data(game, overwrite = TRUE)
```

game_selector_kcal	<i>game_selector_kcal</i>
--------------------	---------------------------

Description

Combining and selecting data sets

Usage

```
game_selector_kcal(food_df)
```

Arguments

food_df	The consumed food and the proportion in % of the consumed food
---------	--

Details

This small function called `game_selector_kcal` adds new variables to one dataset from another dataset and selects the references for the users data input from the packages data

Value

Parameter `food_df`

game_selector_protein	<i>game_selector_protein</i>
-----------------------	------------------------------

Description

Combining and selecting data sets

Usage

```
game_selector_protein(food_df)
```

Arguments

food_df	The consumed food and the proportion in % of the consumed food
---------	--

Details

This small function called `game_selector_protein` adds new variables to one dataset from another dataset and selects the references for the users data input from the packages data

Value

Parameter `food_df`

iron_production	<i>Contains informations of iron production</i>
-----------------	---

Description

Contains informations of iron production

Usage

```
iron_production
```

Format

A data.frame with 1 row and 3 variables:

iron_person_kg The amount of iron needed per person in kg; default 2; source: Weiberg et al. (in prep.)

ratio_charcoal_iron The ratio of charcoal to iron; default 82

ratio_charcoal_wood The ratio of charcoal to wood; default 4

```
iron_production <- read.table("./iron_production.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE)
usethis::use_data(iron_production, overwrite = TRUE)
```

kcal_req_age	<i>Kcal requirements of people for different activity levels and sex</i>
--------------	--

Description

Kcal requirements of people for different activity levels and sex

Usage

```
kcal_req_age
```

Format

A data.frame with 74 rows and 7 variables. # the following code is used to create the data set

```
library(rJava)
library(tabulizer)
report <- "https://www.cnpp.usda.gov/sites/default/files/usda_food_patterns/EstimatedCal"
kcal_req <- extract_tables(report, encoding="UTF-8")
kcal_req <- kcal_req[[1]]
kcal_req <- kcal_req[6:35,]
kcal_req <- gsub(pattern = ",", replacement = "", x = kcal_req)
kcal_req <- as.data.frame(kcal_req, stringsAsFactors = FALSE)
names(kcal_req) <- c("age", "male_sedentary", "male_moderately_active", "male_active", "female_sedentary", "female_moderately_active", "female_active")
kcal_req[,2:7] <- sapply(kcal_req[,2:7], as.numeric)
kcal_req_age <- kcal_req[18:29,]
dplyr::mutate(from = lapply(strsplit(kcal_req[18:29,1], split = "-"), function(x) x[1]), to = lapply(strsplit(kcal_req[18:29,1], split = "-"), function(x) x[2]))
dplyr::rowwise()
dplyr::do(data.frame(age = as.character(seq(. $from, . $to, by = 1)), male_sedentary = . $male_sedentary, male_moderately_active = . $male_moderately_active, male_active = . $male_active, female_sedentary = . $female_sedentary, female_moderately_active = . $female_moderately_active, female_active = . $female_active))
dplyr::bind_rows(kcal_req[1:17,], .)
usethis::use_data(kcal_req_age, overwrite = TRUE)
```

Source

<https://origin.www.cnpp.usda.gov/Publications/USDAFoodPatterns/EstimatedCalorieNeedsPerDayTable.pdf>

kcal_req_detailed	<i>Kcal requirements of people for different age, activity levels and sex</i>
-------------------	---

Description

Kcal requirements of people for different age, activity levels and sex

Usage

```
kcal_req_detailed
```

Format

A data.frame with 444 rows and 4 variables:

age The age of the person

sex The sex of the person

activity The activity level of the person

kcal_req The daily kcal requirements for this type of person

```
kcal_req_detailed <- read.table("./kcal_req_age_long.csv", header = TRUE, colClasses = c(age = "character"), sep = ",", stringsAsFactors = FALSE) dplyr::select(-X1)
```

```
usethis::use_data(kcal_req_detailed, overwrite = TRUE)
```

Source

<https://origin.www.cnpp.usda.gov/Publications/USDAFoodPatterns/EstimatedCalorieNeedsPerDayTable.pdf>

kcal_req_human	<i>Contains informations of the calory requirements for humans</i>
----------------	--

Description

Contains informations of the calory requirements for humans

Usage

```
kcal_req_human
```

Format

A data.frame with 80 rows and 6 variables:

pal The Total Energy Expenditure for 24 hours expressed as a multiple of Basal Metabolic Rate. Calculated as TEE/BMR for 24 hours (cf.FAO2001).

age_category The age categorized as a human younger than 19 years or older than 19 years

sex The gender

age The age expressed as category

kcal_day The caloric requirement as kcal/day

reference The sources of literature for the required kcal data

```
kcal_req_human <- read.table("/calory_requirement_human.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE) usethis::use_data(kcal_req_human, overwrite = TRUE)
```

milk	<i>Contains informations of the dairy products data</i>
------	---

Description

Contains informations of the dairy products data

Usage

```
milk
```

Format

A data.frame with 13 rows and 9 variables:

product The type of milk

milk_per_indiv_and_year The amount of milk available for human consumption per individual and year

unit_milk The unit of the milk; L

calories The amount of calories

unit_calories The unit of the calories; kcal/L

reference The references

protein The amount of protein in g/L

reference_protein The references

calory_req_day_individual The amount of calories needed to feed one individual per day; kcal

```
milk <- read.table("/milk.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE) usethis::use_data(milk, overwrite = TRUE)
```

milk_selector_kcal	<i>milk_selector_kcal</i>
--------------------	---------------------------

Description

Combining and selecting data sets

Usage

```
milk_selector_kcal(food_df)
```

Arguments

food_df	The consumed food and the proportion in % of the consumed food
---------	--

Details

This small function called milk_selector_kcal adds new variables to one dataset from another dataset and selects the references for the users data input from the packages data

Value

Parameter food_df

milk_selector_protein	<i>milk_selector_protein</i>
-----------------------	------------------------------

Description

Combining and selecting data sets

Usage

```
milk_selector_protein(food_df)
```

Arguments

food_df	The consumed food and the proportion in % of the consumed food
---------	--

Details

This small function called milk_selector adds new variables to one dataset from another dataset and selects the references for the users data input from the packages data

Value

Parameter food_df

nd_herd	<i>nd_herd</i>
---------	----------------

Description

Calculate herd characteristics when nothing specific is known

Usage

```
nd_herd(input_animals, kcal_req_village_year, proportion)
```

Arguments

input_animals	The animals input of the data.frame
kcal_req_village_year	The kcal requirements of the village per year
proportion	The proportion of the consumed foodtype in %

Details

This is a small helper function that is called when there are no specific information or empirical evidences to calculate the village meat requirements. To generate more realistic results, the number of required animals is rounded up to the nearest whole number. The function imports the data on animals and animal_products and takes the mean of those values; besides, fodder-plants are imported and all those plants that are also crops are removed from the dataset

Value

A data.frame

nd_herd_protein	<i>nd_herd_protein</i>
-----------------	------------------------

Description

Calculate herd characteristics when nothing specific is known

Usage

```
nd_herd_protein(input_animals, protein_req_village_year, proportion)
```

Arguments

input_animals	The animals input of the data.frame
protein_req_village_year	The kcal requirements of the village per year
proportion	The proportion of the consumed foodtype in %

Details

This is a small helper function that is called when there are no specific information or empirical evidences to calculate the village meat requirements. To generate more realistic results, the number of required animals is rounded up to the nearest whole number. The function imports the data on animals and animal_products and takes the mean of those values; besides, fodder-plants are imported and all those plants that are also crops are removed from the dataset

Value

A data.frame

pasture_selector_kcal *pasture_selector_kcal*

Description

Combining, filtering, selecting and renaming data sets

Usage

```
pasture_selector_kcal(food_df)
```

Arguments

food_df The consumed food and the proportion in % of the consumed food

Details

This small function called pasture_selector adds new variables to one dataset from another dataset, filters the data input and selects the references for the users data input from the packages data

Value

Parameter food_df

pasture_selector_protein
 pasture_selector_protein

Description

Combining, filtering, selecting and renaming data sets

Usage

```
pasture_selector_protein(food_df)
```

Arguments

food_df The consumed food and the proportion in % of the consumed food

Details

This small function called `pasture_selector` adds new variables to one dataset from another dataset, filters the data input and selects the references for the users data input from the packages data

Value

Parameter `food_df`

plants	<i>Contains informations of the plant data</i>
--------	--

Description

Contains informations of the plant data

Usage

```
plants
```

Format

A data.frame with 46 rows and 16 variables:

Type The type of plant

Species The specific plant species

productivity The productivity of the plant in kg/ha

produc_units The units of the productivity

calories The caloric content of the plant in kcal/kg

calories_units The units of the calories

protein The protein content of the plant in g/kg

planting Percentage of the yield needed for planting in the next year

waste_loss Percentage of the yield that is lost

storage Percentage of the yield that is stored

protein_units The units of the protein

yield_units The units of the yield

reference_calories The references of calories data

reference_productivity The references of productivity data

reference_protein The references of protein data

reference_planting_loss_storage The references of the planting, loss and storage data

```
plants <- read.table("./plants.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE) usethis::use_data(plants,
overwrite = TRUE)
```

protein_req_age	<i>Contains informations of the daily protein requirements</i>
-----------------	--

Description

Contains informations of the daily protein requirements

Usage

```
protein_req_age
```

Format

A data.frame with 75 rows and 3 variables:

age The age of the person

male_g_day The daily requirement of protein in g of a male persons

female_g_day The daily requirement of protein in g of a female persons

```
protein_req_age <- read.table("./protein_req_age.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE)
usethis::use_data(protein_req_age, overwrite = TRUE)
```

Source

<https://www.dge.de/wissenschaft/referenzwerte/protein/>

protein_req_age_long	<i>Contains informations of the daily protein requirements</i>
----------------------	--

Description

Contains informations of the daily protein requirements

Usage

```
protein_req_age_long
```

Format

A data.frame with 75 rows and 3 variables:

age The age of the person

sex The sex of the person

protein_req_g The daily requirement of protein in g of this type of person

```
protein_req_age_long <- read.table("./protein_req_age_long.csv", header = TRUE, colClasses = c(age = "character"), sep = ";", stringsAsFactors = FALSE)
usethis::use_data(protein_req_age_long, overwrite = TRUE)
```

Source

<https://www.dge.de/wissenschaft/referenzwerte/protein/>

Arguments

population	The number of people living in the village; default 3800
protein_req_human_year	The per capita protein requirement of the year; if nothing is supplied, it is calculated based on setup_population function
aux_animal	The type of auxiliary animals used; default "bull"
food_df	The consumed food and the proportion in % of the consumed food
...	further arguments passed to another function

Details

This function sets up the kcal etc. values for auxiliary animals in form of a tibble

Value

a tibble

setup_bronze	<i>setup_bronze</i>
--------------	---------------------

Description

Setup of the bronze production

Usage

```
setup_bronze(population, forest_type, ...)
```

Arguments

population	The number of people living in the village
forest_type	The type of the forest; choose between "oak_mixed_forest" and "olive_mixed_forest"; default "oak_mixed_forest"
...	Further arguments passed to other functions

Details

This function sets up the amount of wood required for bronze production

Value

A tibble

```
setup_ceramics_kcal      setup_ceramics_kcal
```

Description

Setup of the ceramics production

Usage

```
setup_ceramics_kcal(population, forest_type, food_df, households,
  olive_oil_yield, ...)
```

Arguments

population	The number of people living in the village
forest_type	The type of the forest; choose between "oak_mixed_forest" and "olive_mixed_forest"; default "oak_mixed_forest"
food_df	The product and the proportion of the consumed food
households	The number of households; default: population / 5, where 5 is the number of people living in a household
olive_oil_yield	The yield of the olive production
...	Further arguments passed to other functions

Details

This function sets up the required wood for ceramic production

Value

A tibble

```
setup_ceramics_protein      setup_ceramics_protein
```

Description

Setup of the ceramics production using protein

Usage

```
setup_ceramics_protein(population, forest_type, food_df, households,
  olive_oil_yield, ...)
```

Arguments

population	The number of people living in the village
forest_type	The type of the forest; choose between "oak_mixed_forest" and "olive_mixed_forest"; default "oak_mixed_forest"
food_df	The product and the proportion of the consumed food
households	The number of households; default: population / 5, where 5 is the number of people living in a household
olive_oil_yield	The yield of the olive production
...	Further arguments passed to other functions

Details

This function sets up the required wood for ceramic production

Value

A tibble

```
setup_diet_arbori_gather
      setup_diet_arbori_gather
```

Description

Setup the general arboriculture and gathered plants diet characteristics

Usage

```
setup_diet_arbori_gather(population, food_df, kcal_req_human_year)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
kcal_req_human_year	The per capita kcal requirement of the year; if nothing is supplied, it is calculated based on setup_population function

Details

This function creates a data.frame of the diet characteristics of a village that is used in subsequent calculations;

Value

A list with a numeric (required hectares to grow plants to sustain the population) and a data.frame that supplies the individual values

```
setup_diet_arbori_gather_protein
    setup_diet_arbori_gather_protein
```

Description

Setup the general arboriculture and gathered plants diet characteristics based on protein requirements

Usage

```
setup_diet_arbori_gather_protein(population, food_df,
    protein_req_human_year)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
protein_req_human_year	The per capita protein requirement of the year; if nothing is supplied, it is calculated based on setup_protein_req function

Details

This function creates a data.frame of the diet characteristics of a village that is used in subsequent calculations;

Value

A list with a numeric (required hectares to grow plants to sustain the population) and a data.frame that supplies the individual values

```
setup_diet_cheese    setup_diet_cheese
```

Description

Setup the general cheese characteristics

Usage

```
setup_diet_cheese(population, food_df, kcal_req_human_year)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
kcal_req_human_year	The per capita kcal requirement of the year; if nothing is supplied, it is calculated based on setup_population function

Details

This function creates a data.frame of the diet characteristics of a village that is used in subsequent calculations. To generate more realistic results, the number of required animals is rounded up to the nearest whole number;

Value

A list with a numeric (required hectares to grow plants to sustain the population) and a data.frame that supplies the individual values

```
setup_diet_cheese_protein
      setup_diet_cheese_protein
```

Description

Setup the general cheese characteristics

Usage

```
setup_diet_cheese_protein(population, food_df, protein_req_human_year)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
protein_req_human_year	The per capita kcal requirement of the year; if nothing is supplied, it is calculated based on setup_population function

Details

This function creates a data.frame of the diet characteristics of a village that is used in subsequent calculations. To generate more realistic results, the number of required animals is rounded up to the nearest whole number;

Value

A list with a numeric (required hectares to grow plants to sustain the population) and a data.frame that supplies the individual values

setup_diet_crops	<i>setup_diet_crops</i>
------------------	-------------------------

Description

Setup the crop and legume diet characteristics

Usage

```
setup_diet_crops(population, food_df, kcal_req_human_year)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
kcal_req_human_year	The per capita kcal requirement of the year; if nothing is supplied, it is calculated based on setup_population function

Details

This function creates a data.frame of the diet characteristics of a village that is used in subsequent calculations;

Value

A list with a numeric (required hectares to grow plants to sustain the population) and a data.frame that supplies the individual values

setup_diet_crops_protein	<i>setup_diet_crops_protein</i>
--------------------------	---------------------------------

Description

Setup the crop and legume diet characteristics based on protein requirements

Usage

```
setup_diet_crops_protein(population, food_df, protein_req_human_year)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
protein_req_human_year	The per capita protein requirement of the year; if nothing is supplied, it is calculated based on setup_protein_req function

Details

This function creates a data.frame of the diet characteristics of a village that is used in subsequent calculations;

Value

A list with a numeric (required hectares to grow plants to sustain the population) and a data.frame that supplies the individual values

setup_diet_game	<i>setup_diet_game</i>
-----------------	------------------------

Description

Setup the general game diet characteristics

Usage

```
setup_diet_game(population, food_df, kcal_req_human_year)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
kcal_req_human_year	The per capita kcal requirement of the year; if nothing is supplied, it is calculated based on setup_population function

Details

This function creates a data.frame of the diet characteristics of a village that is used in subsequent calculations. To generate more realistic results, the number of required animals is rounded up to the nearest whole number;

Value

A list with a numeric (required hectares to grow plants to sustain the population) and a data.frame that supplies the individual values

```
setup_diet_game_protein
    setup_diet_game_protein
```

Description

Setup the general game diet characteristics

Usage

```
setup_diet_game_protein(population, food_df, protein_req_human_year)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
protein_req_human_year	The per capita protein requirement of the year; if nothing is supplied, it is calculated based on setup_protein_req function

Details

This function creates a data.frame of the diet characteristics of a village that is used in subsequent calculations. To generate more realistic results, the number of required animals is rounded up to the nearest whole number;

Value

A list with a numeric (required hectares to grow plants to sustain the population) and a data.frame that supplies the individual values

```
setup_diet_milk    setup_diet_milk
```

Description

Setup the general milk characteristics

Usage

```
setup_diet_milk(population, food_df, kcal_req_human_year)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
kcal_req_human_year	The per capita kcal requirement of the year; if nothing is supplied, it is calculated based on setup_population function

Details

This function creates a data.frame of the diet characteristics of a village that is used in subsequent calculations. To generate more realistic results, the number of required animals is rounded up to the nearest whole number;

Value

A list with a numeric (required hectares to grow plants to sustain the population) and a data.frame that supplies the individual values

```
setup_diet_milk_protein
      setup_diet_milk_protein
```

Description

Setup the general milk characteristics

Usage

```
setup_diet_milk_protein(population, food_df, protein_req_human_year)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
protein_req_human_year	The per capita kcal requirement of the year; if nothing is supplied, it is calculated based on setup_population function

Details

This function creates a data.frame of the diet characteristics of a village that is used in subsequent calculations. To generate more realistic results, the number of required animals is rounded up to the nearest whole number;

Value

A list with a numeric (required hectares to grow plants to sustain the population) and a data.frame that supplies the individual values

setup_diet_pasture	<i>setup_diet_pasture</i>
--------------------	---------------------------

Description

Setup the general meat diet characteristics

Usage

```
setup_diet_pasture(population, food_df, kcal_req_human_year)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
kcal_req_human_year	The per capita kcal requirement of the year; if nothing is supplied, it is calculated based on setup_population function

Details

This function creates a data.frame of the diet characteristics of a village that is used in subsequent calculations;

Value

A list with a numeric (required hectares for pasture to sustain the animal herd) and a data.frame that supplies the individual values

setup_diet_pasture_protein	<i>setup_diet_pasture_protein</i>
----------------------------	-----------------------------------

Description

Setup the general meat diet characteristics

Usage

```
setup_diet_pasture_protein(population, food_df, protein_req_human_year)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
protein_req_human_year	The per capita protein requirement of the year; if nothing is supplied, it is calculated based on setup_protein_req function

Details

This function creates a data.frame of the diet characteristics of a village that is used in subsequent calculations;

Value

A list with a numeric (required hectares for pasture to sustain the animal herd) and a data.frame that supplies the individual values

setup_flax	<i>setup_flax</i>
------------	-------------------

Description

Setup the values for flax

Usage

```
setup_flax(population)
```

Arguments

population The number of people living in the village

Details

This function sets up the kcal etc. values for flax in form of a tibble

Value

A tibble

setup_iron	<i>setup_iron</i>
------------	-------------------

Description

Setup of the iron production

Usage

```
setup_iron(population, forest_type, ...)
```

Arguments

population The number of people living in the village

forest_type The type of the forest; choose between "oak_mixed_forest" and "olive_mixed_forest"; default "oak_mixed_forest"

... Further arguments passed to other functions

Details

This function sets up the amount of wood required for iron production

Value

A tibble with the characteristics of Iron

setup_olive_oil	<i>setup_olive_oil_kcal</i>
-----------------	-----------------------------

Description

Setup of the kcal etc. values for olive_oil

Usage

```
setup_olive_oil(population, households, olive_oil_storage = 0.25,
  rich_people_factor = 0.66, ...)
```

Arguments

population	The number of people living in a village
households	The number of households; default: population / 5, where 5 is the number of people living in a household
olive_oil_storage	The percentage of the yield that is stored
rich_people_factor	A factor to reflect the higher consumption of rich residents
...	Further arguments passed to other functions

Details

This function sets up the kcal etc. values for olive_oil in form of a tibble

Value

A tibble

setup_population	<i>setup_population</i>
------------------	-------------------------

Description

Calculate the energy requirements of a village

Usage

```
setup_population(population, mode = "simplified",
  kcal_requirement = NULL)
```

Arguments

population	The number of people living in a village
mode	The calculation mode; currently only "simplified" (the default) and "medium" is implemented; further modes will be added soon and require further input data simplified approach takes the mean of empirical values of daily kcal requirements for active females and males in the age between 10 and 50 (source for daily requirements: http://www.cnpp.usda.gov/sites/default/files/usda_food_patterns/EstimatedCalorieNeedsPerDayTable.pdf)
kcal_requirement	optional; default NULL

Details

Calculate the energy requirements of a village based on a given population size

This functions is a wrapper for more comprehensive functions calculating the nutritional (kcal) requirements of a village; These are not implemented yet; Note that more comprehensive approaches, e.g. considering a death-table of a village, require a lot more data that is often not available

Value

A list with the elements: Population, KcalHuman, KcalHumanYear, KcalVillageYear

setup_population_detailed_kcal	<i>setup_population_detailed_kcal</i>
--------------------------------	---------------------------------------

Description

Calculate the energy requirements of a village based on number, age, sex and activity level

Usage

```
setup_population_detailed_kcal(population)
```

Arguments

population A data.frame containing informations about number, age, sex and activity level of each demographic group

Details

Calculate the energy requirements of a village based on number, age, sex and activity level based on a data.frame containing these information

This functions uses more detailed demographic informations to caluculate the yearly kcal requirements of a village. To be able to do this, the data has to be stored in a data.frame.

Value

A list with the elements: Population, KcalHumanYear, KcalVillageYear

```
setup_population_detailed_protein
      setup_population_detailed_protein
```

Description

Calculate the protein requirements of a village based on number, age, sex and activity level

Usage

```
setup_population_detailed_protein(population)
```

Arguments

population A data.frame containing informations about number, age and sex of each demographic group

Details

Calculate the protein requirements of a village based on number, age, sex and activity level based on a data.frame containing these information

This functions uses more detailed demographic informations to caluculate the yearly protein requirements of a village. To be able to do this, the data has to be stored in a data.frame.

Value

A list with the elements: Population, ProtVillage, ProtHumanYear, ProtVillageyear

setup_protein_req	<i>setup_protein_req</i>
-------------------	--------------------------

Description

Calculate the protein requirements of a village

Usage

```
setup_protein_req(population, protein_requirement = NULL)
```

Arguments

population	The number of people living in a village
protein_requirement	optional; default NULL reference: https://www.dge.de/wissenschaft/referenzwerte/protein/

Details

Calculate the protein requirements of a village based on a given population size

This functions is a wrapper for more comprehensive functions calculating the protein (g) requirements of a village

Value

A list with the elements: Population, ProteinHuman, ProteinHumanYear, ProteinVillageYear

setup_wine	<i>setup_wine_kcal</i>
------------	------------------------

Description

Setup of the kcal etc. values for wine

Usage

```
setup_wine(population, food_df, ...)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
...	Further arguments passed to other functions

Details

This function sets up the kcal etc. values for wine in form of a tibble

Value

A tibble

setup_wine_protein	<i>setup_wine_protein</i>
--------------------	---------------------------

Description

Setup of the protein etc. values for wine

Usage

```
setup_wine_protein(population, food_df, ...)
```

Arguments

population	The number of people living in a village
food_df	The consumed food and the proportion in % of the consumed food
...	Further arguments passed to other functions

Details

This function sets up the kcal etc. values for wine in form of a tibble

Value

A tibble

setup_wood_consumption	<i>wood_consumption</i>
------------------------	-------------------------

Description

Setup of the wood consumption

Usage

```
setup_wood_consumption(population, forest_type, ...)
```

Arguments

population	The number of people living in the village; default: 3800;
forest_type	The type of the forest; choose between "oak_mixed_forest" and "olive_mixed_forest"; default "oak_mixed_forest"
...	Further arguments passed to other functions

Details

This function calculates the wood consumption of the village

Value

A tibble

spelling_checker_kcal *spelling_checker_kcal*

Description

checking the spelling of the animal data input

Usage

spelling_checker_kcal(animal_name)

Arguments

animal_name The name of the animal

Details

This is a small helper function that is called spelling_checker_kcal and it allows the user to use plural or singular for data input

Value

Parameter animal_name

spelling_checker_protein
 spelling_checker_protein

Description

checking the spelling of the animal data input

Usage

spelling_checker_protein(animal_name)

Arguments

animal_name The name of the animal

Details

This is a small helper function that is called spelling_checker_protein and it allows the user to use plural or singular for data input

Value

Parameter animal_name

summarize_areas_kcal *summarize_areas_kcal*

Description

Summarize_areas_kcal

Usage

```
summarize_areas_kcal(population = 3800, kcal_requirement = NULL,
  household_people = 5, food_df, aux_animal = "bull",
  forest_type = "oak_mixed_forest", bronze = FALSE, iron = FALSE,
  ceramics = FALSE, olive_oil = FALSE, flax = FALSE, ...)
```

Arguments

population	The number of people living in a village; default 3800
kcal_requirement	The kcal a human requires per day; default NULL
household_people	The number of people per household; default 5
food_df	The consumed food and the proportion in % of the consumed food
aux_animal	The type of auxiliary animals used; default "bull"
forest_type	The type of the forest; choose between "oak_mixed_forest" and "olive_mixed_forest"; default "oak_mixed_forest"
bronze	Whether the population melted/used bronze; default FALSE
iron	Whether the population melted/used iron; default FALSE
ceramics	Whether the population produced/used ceramic; default FALSE
olive_oil	Whether the population produced olive oil; default FALSE
flax	Whether the population needs flax; default FALSE
...	Further arguments passed to other functions

Details

Bring all function calculations together

Value

A tibble

```
summarize_areas_protein
      summarize_areas_protein
```

Description

Summarize_areas_protein

Usage

```
summarize_areas_protein(population = 3800, protein_requirement = NULL,
  household_people = 5, food_df, aux_animal = "bull",
  forest_type = "oak_mixed_forest", bronze = FALSE, iron = FALSE,
  ceramics = FALSE, olive_oil = FALSE, flax = FALSE, ...)
```

Arguments

population	The number of people living in a village; default 3800
protein_requirement	The protein a human requires per day; default NULL
household_people	The number of people per household; default 5
food_df	The consumed food and the proportion in % of the consumed food
aux_animal	The type of auxiliary animals used; default "bull"
forest_type	The type of the forest; choose between "oak_mixed_forest" and "olive_mixed_forest"; default "oak_mixed_forest"
bronze	Whether the population melted/used bronze; default FALSE
iron	Whether the population melted/used iron; default FALSE
ceramics	Whether the population produced/used ceramic; default FALSE
olive_oil	Whether the population produced olive oil; default FALSE
flax	Whether the population needs flax; default FALSE
...	Further arguments passed to other functions

Details

Bring all function calculations together

Value

A tibble

wood_cons

Contains informations of the wood consumption

Description

Contains informations of the wood consumption

Usage

wood_cons

Format

A data.frame with 1 row and 2 variables:

consumption_energy_person_a_GJ The amount of energy one person consumes per year by burning wood in GJ

consumption_fuelwood_person_a_kg The amount of fuelwood a person uses in kg

```
wood_cons <- read.table("./wood_cons.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE)
usethis::use_data(wood_cons, overwrite = TRUE)
```

wood_prod

Contains informations of the wood productivity

Description

Contains informations of the wood productivity

Usage

wood_prod

Format

A data.frame with 2 row and 4 variables:

forest_type The Type of wood that is used by the people/villagers

productivity_in_cbm_per_ha_per_a The productivity of the forest/woodland in cbm per ha and year

calorific_value_in_MJ_per_kg The calorific value of the wood in MJ per kg

references The references of the data

```
wood_prod <- read.table("./wood_prod.csv", header = TRUE, sep = ";", stringsAsFactors = FALSE)
usethis::use_data(wood_prod, overwrite = TRUE)
```

Index

*Topic **datasets**

- animal_products, [3](#)
- animals, [3](#)
- aux_animals, [6](#)
- bronze_production, [6](#)
- ceramics_production, [7](#)
- cheese, [7](#)
- flax_production, [11](#)
- game, [11](#)
- iron_production, [13](#)
- kcal_req_age, [13](#)
- kcal_req_detailed, [14](#)
- kcal_req_human, [14](#)
- milk, [15](#)
- plants, [19](#)
- protein_req_age, [20](#)
- protein_req_age_long, [20](#)
- wood_cons, [41](#)
- wood_prod, [41](#)

- animal_products, [3](#)
- animals, [3](#)
- arbori_selector_kcal, [4](#)
- arbori_selector_protein, [5](#)
- area_village, [5](#)
- aux_animals, [6](#)

- bronze_production, [6](#)

- ceramics_production, [7](#)
- cheese, [7](#)
- cheese_selector_kcal, [8](#)
- cheese_selector_protein, [8](#)
- crop_selector_kcal, [9](#)
- crop_selector_protein, [9](#)

- data_checker_kcal, [10](#)
- data_checker_protein, [10](#)

- flax_production, [11](#)

- game, [11](#)
- game_selector_kcal, [12](#)
- game_selector_protein, [12](#)

- iron_production, [13](#)

- kcal_req_age, [13](#)
- kcal_req_detailed, [14](#)
- kcal_req_human, [14](#)

- milk, [15](#)
- milk_selector_kcal, [16](#)
- milk_selector_protein, [16](#)

- nd_herd, [17](#)
- nd_herd_protein, [17](#)

- pasture_selector_kcal, [18](#)
- pasture_selector_protein, [18](#)
- plants, [19](#)
- protein_req_age, [20](#)
- protein_req_age_long, [20](#)

- setup_auxiliary_animals_kcal, [21](#)
- setup_auxiliary_animals_protein, [21](#)
- setup_bronze, [22](#)
- setup_ceramics_kcal, [23](#)
- setup_ceramics_protein, [23](#)
- setup_diet_arbori_gather, [24](#)
- setup_diet_arbori_gather_protein, [25](#)
- setup_diet_cheese, [25](#)
- setup_diet_cheese_protein, [26](#)
- setup_diet_crops, [27](#)
- setup_diet_crops_protein, [27](#)
- setup_diet_game, [28](#)
- setup_diet_game_protein, [29](#)
- setup_diet_milk, [29](#)
- setup_diet_milk_protein, [30](#)
- setup_diet_pasture, [31](#)
- setup_diet_pasture_protein, [31](#)
- setup_flax, [32](#)
- setup_iron, [32](#)
- setup_olive_oil, [33](#)
- setup_population, [34](#)
- setup_population_detailed_kcal, [34](#)
- setup_population_detailed_protein, [35](#)
- setup_protein_req, [36](#)
- setup_wine, [36](#)
- setup_wine_protein, [37](#)

setup_wood_consumption, [37](#)
spelling_checker_kcal, [38](#)
spelling_checker_protein, [38](#)
summarize_areas_kcal, [39](#)
summarize_areas_protein, [40](#)

wood_cons, [41](#)
wood_prod, [41](#)