

# Package ‘care4cmodel’

June 7, 2024

**Type** Package

**Title** Carbon-Related Assessment of Silvicultural Concepts

**Version** 1.0.2

**Date** 2024-06-03

**Description** A simulation model and accompanying functions that support assessing silvicultural concepts on the forest estate level with a focus on the CO<sub>2</sub> uptake by wood growth and CO<sub>2</sub> emissions by forest operations. For achieving this, a virtual forest estate area is split into the areas covered by typical phases of the silvicultural concept of interest. Given initial area shares of these phases, the dynamics of these areas is simulated. The typical carbon stocks and flows which are known for all phases are attributed post-hoc to the areas and upscaled to the estate level. CO<sub>2</sub> emissions by forest operations are estimated based on the amounts and dimensions of the harvested timber. Probabilities of damage events are taken into account.

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**LazyData** true

**Imports** deSolve, dplyr, stats, tibble, purrr, rlang, tidyr,  
tidyselect, ggplot2, Rdpack

**RdMacros** Rdpack

**Depends** R (>= 4.2.0)

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Peter Biber [aut, cre, cph] (<<https://orcid.org/0000-0002-9700-8708>>),  
Stefano Grigolato [ctb] (<<https://orcid.org/0000-0002-2089-3892>>),  
Julia Schmucker [ctb] (<<https://orcid.org/0000-0001-9996-4851>>),  
Enno Uhl [ctb] (<<https://orcid.org/0000-0002-7847-923X>>),  
Hans Pretzsch [ctb] (<<https://orcid.org/0000-0002-4958-1868>>)

**Maintainer** Peter Biber <p.biber@tum.de>

**Repository** CRAN

**Date/Publication** 2024-06-04 09:48:37 UTC

## Contents

aggregate_raw_sim_rslt . . . . .	2
avg_extraction_distance . . . . .	4
c4c_concept . . . . .	4
co2_eval_cutting . . . . .	5
co2_eval_moving . . . . .	6
exp_decay_rate . . . . .	7
fuel_and_co2_evaluation . . . . .	8
fuel_cons_forwarder_1 . . . . .	10
fuel_cons_forwarder_2 . . . . .	11
fuel_cons_harvester_1 . . . . .	12
fuel_cons_harvester_2 . . . . .	13
fuel_cons_road_maintenance . . . . .	14
fuel_to_co2 . . . . .	14
growth_and_yield_evaluation . . . . .	15
is_c4c_concept . . . . .	17
new_c4c_concept . . . . .	17
pine_no_thinning_and_clearcut_1 . . . . .	18
pine_thinning_from_above_1 . . . . .	18
plot.c4c_base_result . . . . .	19
plot.c4c_co2_result . . . . .	20
setup_parms . . . . .	21
setup_risk_events . . . . .	22
setup_statevars . . . . .	23
simulate_single_concept . . . . .	24
sim_area_single_concept_with_risk . . . . .	26
survival_weibull . . . . .	27
validate_c4c_concept . . . . .	28
wood_to_co2 . . . . .	29
<b>Index</b>	<b>30</b>

---

aggregate\_raw\_sim\_rslt

*Comprehensively Aggregate Raw Simulation Results*

---

## Description

Aggregate and prepare raw simulation output as obtained from `sim_area_single_concept_with_risk` in a way that makes them readable and appropriate for further processing.

**Usage**

```
aggregate_raw_sim_rslt(sim_areas_raw, concept_def)
```

**Arguments**

`sim_areas_raw` Raw simulation results as obtained from [sim\\_area\\_single\\_concept\\_with\\_risk](#)

`concept_def` The concept definition (a `c4c_concept` object) used for the simulation which generated `sim_areas_raw`

**Value**

A list of three matrices with named columns. Each row of these matrices represents a point in simulation time, but only integer times. The time distance from one row to the next one is one time unit, typically one year. The first column of the matrices is time, the other columns represent the stand development phases as defined in `concept_def`. The three list elements (matrices) are:

- *areas*: Contains the simulated areas of each stand development phase in the area units defined in `concept_def`, usually ha.
- *area\_inflows\_regular*: The area inflows into each stand development phase as caused by regular development, not by damage events. An entry at a given point in time represents the inflow between (and up to) this and the previous point in time. Therefore, the entries at time 0 are NA.
- *area\_outflows\_events*: Area outflows of each stand development phase as caused by damage events. An entry at a given point in time represents the inflow between (and up to) this and the previous point in time. Therefore, the entries at time 0 are NA.

**Examples**

```
# Make a simulation
state_vars <- setup_statevars(
  pine_thinning_from_above_1, c(1000, 0, 0, 0, 0, 0)
)
time_span <- 50
parms <- setup_parms(pine_thinning_from_above_1)
parms$risk_mat <- setup_risk_events(
  time_span, avg_event_strength = 1, parms$risk
)

# Simulate
sim_rslt_raw <- sim_area_single_concept_with_risk(
  state_vars,
  parms = parms,
  event_times = c(0:time_span),
  time_span = time_span
)

aggregate_raw_sim_rslt(sim_rslt_raw, pine_thinning_from_above_1)
```

---

avg\_extraction\_distance

*Estimate Average Wood Extraction Distance From Forest Road Density*

---

### Description

Estimate Average Wood Extraction Distance From Forest Road Density

### Usage

```
avg_extraction_distance(frd)
```

### Arguments

frd                    Forest road density (truck roads) in m/ha

### Value

The average extraction distance from the felling spot to the nearest landing at a truck road.

### Examples

```
frd <- c(15, 30, 60, 100) # Forest road densities m/ha
avg_extraction_distance(frd)
```

---

c4c\_concept

*User-Friendly Construction of a c4c\_concept Object*

---

### Description

For creating a c4c\_concept object under normal circumstances, you should not use the constructor [new\\_c4c\\_concept](#) directly, but this function.

### Usage

```
c4c_concept(growth_and_yield, concept_name)
```

### Arguments

growth\_and\_yield

data.frame with at least two rows and the columns "phase\_no", "phase\_name", "duration", "n\_subphases", "vol\_standing", "vol\_remove", "vol\_mort", "dbh\_standing", "dbh\_remove", "n\_standing", "n\_remove", "harvest\_interval", and "survival\_cum".

concept\_name    Character, name of the concept defined

## Details

Special attention needs to be paid to the definition of **vol\_remove** in cases when the standing volume decreases from one phase to the next. If the value given for `vol_remove` is too low, it will result in a negative volume increment for the respective phase. This will not pass the validation called inside this function.

## Value

A valid object of class `c4c_concept`, if it can be constructed from the input data; stops with an error otherwise. The object is basically a list. Its most important ingredient is a tibble named `growth_and_yield` which is a honed version of the input `growth_and_yield` to this function. It contains, in addition, phase wise periodical volume increments per ha (column `vol_increment`), which result from the given information. There is no option for user-provided volume increments in order to guarantee consistency.

## Examples

```
# construct dummy example (without real life relevance)
g_and_y <- data.frame(
  phase_no       = 1:2,
  phase_name     = c("young", "older"),
  duration       = c(10, 10),
  n_subphases    = c(3, 3),
  vol_standing   = c(166, 304),
  vol_remove     = c(0, 23.6),
  vol_mort       = c(0.01, 0.11),
  n_standing     = c(3200, 970),
  n_remove       = c(0, 306),
  dbh_standing   = c(9.4, 22.3),
  dbh_remove     = c(0, 12.3),
  harvest_interval = c(0, 5),
  survival_cum   = c(0.999, 0.852)
)

dummy_concept <- c4c_concept(g_and_y, "dummy_concept")
dummy_concept
```

## Description

Given the output of a simulation run (i.e. an object of class `c4c_base_result`) as created with the function `simulate_single_concept`, the fuel consumption and CO2 emissions for cutting (i.e. felling, limbing, cutting the trees into logs) are calculated. Currently, this function assumes only harvester operations.

**Usage**

```
co2_eval_cutting(x, mode = c("standard", "nordic"))
```

**Arguments**

**x** An object of class `c4c_base_result`

**mode** Character string to choose between "standard" (default) and "nordic". For "standard", the function `fuel_cons_harvester_1` is used. However, as this function does only provide values for operations with an average harvest tree diameter of 15 cm and more, the "nordic" function `fuel_cons_harvester_2` with the option `thinning = TRUE` is used for smaller average tree sizes. With the choice "nordic" only the function `fuel_cons_harvester_2` is used. However, if the harvested volume per ha per operation amounts to 90% and more of the standing volume per ha of the respective stand development phase, which is virutally a clearcut, the option `thinning = FALSE` is used. In case of damage- induced harvest, only the option `thinning = TRUE` is in effect in order to account for the more difficult conditions of such harvest operations.

**Value**

A data frame (tibble) with the columns `time`, `harvest_type` (damage or regular), `phase_no`, `phase_name` (numbers and names of the stand development phases), `fuel_cutting_l_per_m3` (liters of fuel consumed per m3 of harvested wood), `fuel_cutting_total_l` (liters of fuel consumed in total), `co2_cutting_total_kg` (kg CO2 emitted).

**Examples**

```
base_out <- simulate_single_concept(
  pine_thinning_from_above_1,
  init_areas = c(50, 100, 10, 50, 150, 600),
  time_span = 50,
  risk_level = 3
)

co2_eval_cutting(base_out, "standard")
co2_eval_cutting(base_out, "nordic")
```

---

 co2\_eval\_moving

*Fuel Consumption and CO2 Emissions for Moving Wood From the Felling Spot to the Forest Road*

---

**Description**

Given the output of a simulation run (i.e. an object of class `c4c_base_result`) as created with the function `simulate_single_concept`, the fuel consumption and CO2 emissions for moving the wood to a truck road are calculated. Currently, this function assumes only forwarder operations.

**Usage**

```
co2_eval_moving(
  x,
  road_density,
  rel_loss = 0.1,
  mode = c("standard", "nordic")
)
```

**Arguments**

x	An object of class <code>c4c_base_result</code>
road_density	Forest road density (m/ha), relating to truck-accessible roads
rel_loss	Relative amount of the standing tree volume that is lost during harvesting (default 0.1). Note that the harvested amount is reduced with the factor $1 - \text{rel\_loss}$ before upscaling from the fuel consumption per m <sup>3</sup> , because only the wood remaining after the harvest loss (mainly the stumps) is actually moved.
mode	Character string to choose between "standard" (default) and "nordic". With the option "standard", the function <code>fuel_cons_forwarder_1</code> is used, while "nordic" makes use of <code>fuel_cons_forwarder_2</code>

**Value**

A data frame (tibble) with the columns `time`, `harvest_type` (damage or regular), `phase_no`, `phase_name` (numbers and names of the stand development phases), `fuel_moving_l_per_m3` (liters of fuel consumed per m<sup>3</sup> of moved wood), `fuel_moving_total_l` (liters of fuel consumed in total), `co2_moving_total_kg` (kg CO<sub>2</sub> emitted).

**Examples**

```
base_out <- simulate_single_concept(
  pine_thinning_from_above_1,
  init_areas = c(50, 100, 10, 50, 150, 600),
  time_span = 50,
  risk_level = 3
)

co2_eval_moving(base_out, road_density = 35, mode = "standard")
co2_eval_moving(base_out, road_density = 35, mode = "nordic")
```

---

 exp\_decay\_rate

*Calculate an Exponential Decay Rate From Two Appropriate Pairs of Values*

---

**Description**

Assuming an exponential decay process  $y = \exp(-r * t)$ , this function calculates  $r$  if the following information is given:

$$y_1 = \exp(-r * t_1), y_2 = \exp(-r * t_2)$$

Hereby,  $t_1$  is the earlier,  $t_2$  the later point in time. This implies the following conditions:  $t_2 > t_1, y_2 \leq y_1$

If these conditions are not given, the function will terminate with an error.

**Usage**

```
exp_decay_rate(t_1, t_2, y_1, y_2)
```

**Arguments**

<code>t_1</code>	Earlier point in time, coupled to <code>y_1</code>
<code>t_2</code>	Later point in time, coupled to <code>y_2</code>
<code>y_1</code>	Earlier value, coupled to <code>t_1</code>
<code>y_2</code>	Later value, coupled to <code>t_2</code>

**Value**

The exponential decay rate  $r$ , relating to the time unit of  $t_1$  and  $t_2$

**Examples**

```
# Up to an age of t_1 = 30, a forest stand of interest has a survival
# probability of 0.95. Up to an age of t_2 = 80, it has a survival
# probability of 0.83. If we assume an exponential decay process for the
# 50-year period, what is the exponential decay rate r?
r <- exp_decay_rate(30, 80, 0.95, 0.83)
print(r)

# Check it
0.95 * exp(-r * (80 - 30)) # 0.83
```

---

fuel\_and\_co2\_evaluation

*Overarching Evaluation of Fuel Consumption and CO2 Emissions*

---

**Description**

Given the output of a simulation run generated with `simulate_single_concept`, i.e. an object of class `c4c_base_result`, a set of information related to CO2 emissions and storage is generated on different levels of aggregation.



**Usage**

```
fuel_and_co2_evaluation(
  x,
  road_density_m_ha,
  raw_density_kg_m3 = 520,
  harvest_loss = 0.1,
  bark_share = 0.12,
  mode = c("standard", "nordic")
)
```

**Arguments**

x	An object of class <code>c4c_base_result</code>
road_density_m_ha	The forest road density on the whole area in m/ha
raw_density_kg_m3	The raw wood density (kg/m <sup>3</sup> ) to be used for wood volume conversions (i.e. density of air-dry wood (12% water content)). Default is 520 kg/m <sup>3</sup> (typical for Scots pine). Internally, wood volume is converted into CO2 equivalents with <a href="#">wood_to_co2</a> .
harvest_loss	Relative loss fraction of wood volume during harvest, mainly comprising the stumps (default = 0.1). Does not include bark losses.
bark_share	Relative wood volume share of the bark. Required, as the CO2 equivalents of harvested wood are calculated for wood volume under bark.
mode	Character string indicating the mode of calculating fuel consumption due to harvest operations. This relates to the functions <a href="#">co2_eval_cutting</a> and <a href="#">co2_eval_moving</a> , see the documentation of these functions for details.

**Value**

An object of class `c4c_co2_result` which is, in essence, a list of three result data frames (and metadata about the underlying simulation), providing information about co2 emissions, storage, and fuel consumption on different levels of aggregation.

**Examples**

```
# Make a simulation run first
# The resulting object base_output (class c4c_base_result) comprises
# the simulated phase area dynamics as well as extended growth and yield
# information
base_output <- simulate_single_concept(
  pine_thinning_from_above_1,
  init_areas = c(1000, 0, 0, 0, 0, 0),
  time_span = 200,
  risk_level = 3
)

# Generate fuel and CO2 related information
```

```
fuel_and_co2_evaluation(  
  base_output,  
  road_density_m_ha = 35,  
  mode = "nordic"  
)
```

---

fuel\_cons\_forwarder\_1 *Fuel Consumption of a Forwarder per Cubic Meter Harvested Wood (Version #1)*

---

### Description

Fuel consumption per m<sup>3</sup> harvested wood derived from the data provided by Grigolato and Cadei (2022). Includes loading, transportation, and unloading.

### Usage

```
fuel_cons_forwarder_1(aed)
```

### Arguments

aed                    Average extraction distance to the nearest truck road

### Value

Fuel consumption of a forwarder in liters diesel fuel per m<sup>3</sup> wood to be handled

### References

Grigolato S, Cadei A (2022). "Full-mechanized CTL production data in Scots pine forest in Poland." <https://researchdata.cab.unipd.it/659/>.

### Examples

```
frd <- c(15, 30, 60, 100) # Forest road densities m/ha  
avg_extraction_distance(frd) |>  
  fuel_cons_forwarder_1()
```

---

fuel\_cons\_forwarder\_2 *Fuel Consumption of a Forwarder per Cubic Meter Harvested Wood (Version #2)*

---

### Description

Fuel consumption per m<sup>3</sup> harvested wood after Kärhä et al. (2023). Includes loading, transportation, and unloading.

### Usage

```
fuel_cons_forwarder_2(aed, harvest_vol_ha, mineral_soil = TRUE)
```

### Arguments

aed	Average extraction distance to the nearest truck road
harvest_vol_ha	Harvested merchantable wood volume over bark per ha (m <sup>3</sup> /ha)
mineral_soil	Logical, TRUE (default) if the operation takes place on mineral soil, FALSE if not

### Value

Fuel consumption of a forwarder in liters diesel fuel per m<sup>3</sup> wood to be handled

### References

Kärhä K, Haavikko H, Kääriäinen H, Palander T, Eliasson L, Roininen K (2023). “Fossil-fuel consumption and CO<sub>2</sub>eq emissions of cut-to-length industrial roundwood logging operations in Finland.” *European Journal of Forest Research*, 1–17.

### Examples

```
frd <- c(15, 30, 60, 100) # Forest road densities m/ha
aed <- avg_extraction_distance(frd)

fuel_cons_forwarder_2(aed, 100, TRUE)
fuel_cons_forwarder_2(aed, 100, FALSE)
```

---

fuel\_cons\_harvester\_1 *Fuel Consumption of a Harvester per Cubic Meter Harvested Wood (Version #1)*

---

## Description

Fuel consumption depends on the average tree volume. For tree diameters at breast height < 15 cm the function gives back NA, because the assumed machine does not work with such small trees. Estimated after Bacescu et al. (2022).

## Usage

```
fuel_cons_harvester_1(tree_vol, tree_dbh)
```

## Arguments

tree_vol	Average standing merchantable wood volume over bark (m <sup>3</sup> ) per harvested tree
tree_dbh	Average diameter at breast height (cm) per harvested tree

## Value

Fuel consumption of a harvester in liters diesel fuel per m<sup>3</sup> harvested wood

## References

Bacescu NM, Cadei A, Moskalik T, Wiśniewski M, Talbot B, Grigolato S (2022). “Efficiency Assessment of Fully Mechanized Harvesting System through the Use of Fleet Management System.” *Sustainability*, **14**(24). ISSN 2071-1050, doi:10.3390/su142416751, <https://www.mdpi.com/2071-1050/14/24/16751>.

## Examples

```
dbh <- seq(10, 70, 10) # Vector of tree dbh in cm
vol <- dbh ^ 2 / 1000 # Simple Volume estimate (m3) with Denzin's formula

fuel_cons_harvester_1(vol, dbh)
```

---

fuel\_cons\_harvester\_2 *Fuel Consumption of a Harvester per Cubic Meter Harvested Wood (Version #2)*

---

### Description

Fuel consumption of a harvester in liters diesel per m<sup>3</sup> harvested wood after Kärhä et al. (2023).

### Usage

```
fuel_cons_harvester_2(tree_vol, harvest_vol_ha, thinning = TRUE)
```

### Arguments

tree_vol	Average standing merchantable wood volume over bark (m <sup>3</sup> ) per harvested tree
harvest_vol_ha	Harvested merchantable wood volume over bark per ha (m <sup>3</sup> /ha)
thinning	Logical, TRUE (default) if the harvest is a thinning, or another kind of felling operation (FALSE)

### Value

Fuel consumption of a harvester in liters diesel fuel per m<sup>3</sup> harvested wood

### References

Kärhä K, Haavikko H, Kääriäinen H, Palander T, Eliasson L, Roininen K (2023). “Fossil-fuel consumption and CO<sub>2</sub>eq emissions of cut-to-length industrial roundwood logging operations in Finland.” *European Journal of Forest Research*, 1–17.

### Examples

```
tree_vol <- c(0.03, 0.10, 1.00, 2.00, 5.00)
harvest_vol <- c(5.00, 10.00, 50.00, 25.00, 10.00)

fuel_cons_harvester_2(tree_vol, harvest_vol, TRUE)
fuel_cons_harvester_2(tree_vol, harvest_vol, FALSE)
```

---

fuel\_cons\_road\_maintenance

*Annual Fuel Consumption for Truck Road Network Maintenance*

---

### Description

Estimate the annual diesel fuel consumption per ha for maintaining an existing truck road network in the forest. Estimate based on Enache and Stampfer (2015).

### Usage

```
fuel_cons_road_maintenance(frd)
```

### Arguments

frd                      Forest road density in m/ha

### Value

Diesel fuel consumption for truck road network maintenance in l/ha/year

### References

Enache A, Stampfer K (2015). "Machine utilization rates, energy requirements and greenhouse gas emissions of forest road construction and maintenance in Romanian mountain forests." *Journal of Green Engineering*, 4(4), 325–350.

### Examples

```
frd <- c(15, 30, 60, 100)
fuel_cons_road_maintenance(frd)
```

---

fuel\_to\_co2

*Convert Fuel Consumption into CO2 Emission*

---

### Description

Simple conversion assuming a factor of 2.61 kg CO<sub>2</sub> / l diesel fuel

### Usage

```
fuel_to_co2(fuel_cons_ltrs, fuel_type = c("diesel"))
```

**Arguments**

`fuel_cons_ltrs` Fuel amount consumed by a harvester in liters.

`fuel_type` Fuel type (character string), required to find the correct conversion factor. Currently, only "diesel" is accepted.

**Value**

The emitted amount of CO2 in kg coming from burning `fuel_cons_ltrs`

**Examples**

```
dbh <- seq(20, 70, 10) # Vector of tree dbh in cm
vol <- dbh ^ 2 / 1000 # Simple Volume estimate (m3) with Denzin's formula

fuel_cons_harvester_1(vol, dbh) |>
  fuel_to_co2()
```

---

growth\_and\_yield\_evaluation

*Extensive Growth and Yield Evaluation*

---

**Description**

Provide an extensive compilation of growth and yield related results.

**Usage**

```
growth_and_yield_evaluation(sim_agg, concept_def, detailed_out = FALSE)
```

**Arguments**

`sim_agg` Aggregated simulated area dynamics, more precisely, the output of [aggregate\\_raw\\_sim\\_rslt](#)

`concept_def` Concept definition matching `sim_agg`, a `c4c_concept` object

`detailed_out` Boolean, if TRUE, also pre-evaluation output (calculated by the internal function `growth_and_yield_pre_eval`) will be part of the result list (default = FALSE)

**Details**

The result object, a list, contains the following elements:

- *gyield\_summary*: A tibble that contains phase overarching growth and yield information. In this tibble, each row is a point in time. The columns (in addition to time) are: *vol\_standing*, the standing volume on the total area of interest; *vol\_rmv\_cont*, the continuous removals that take place, as long as an area is in a given phase; *vol\_rmv\_trans*, the volume removals occurring at phase transitions from phases with more to such with less standing volume;

*vol\_rmv\_damage*, the volume losses due to damage events; *vol\_rmv\_harvest*, regular harvest volume, the sum of *vol\_rmv\_cont* and *vol\_rmv\_trans*; *vol\_rmv\_total*, all removed volume, the sum of *vol\_rmv\_harvest* and *vol\_rmv\_damage*; *vol\_mort*, the mortality volume (normal, not event-triggered); *vol\_inc*, the volume increment on the whole area resulting from the *vol\_standing*, *vol\_rmv\_total*, and *vol\_mort*.

- *gyield\_phases*: A list of tibbles, one for each variable as contained in *gyield\_summary* (except the volume increment) which makes no sense in a phase-wise context), but in phase-wise resolution. In each tibble, each row is a point in time, the columns represent the stand development phases.
- *gyield\_pre* (in case the user has chosen `detailed_out = TRUE`): A tibble, where each row is a point in time. The columns (in addition to time) are: *vol\_standing*, the standing volume on the total area of interest; *vol\_rmv\_cont*, the continuous removals that take place, as long as an area is in a given phase; *vol\_rmv\_trans*, the volume removals occurring at phase transitions from phases with more to such with less standing volume; *vol\_rmv\_damage*, the volume losses due to damage events; *vol\_rmv\_harvest*, regular harvest volume, the sum of *vol\_rmv\_cont* and *vol\_rmv\_trans*; *vol\_rmv\_total*, all removed volume, the sum of *vol\_rmv\_harvest* and *vol\_rmv\_damage*; *vol\_mort*, the mortality volume (normal, not event-triggered); *vol\_inc*, the volume increment on the whole area resulting from the *vol\_standing*, *vol\_rmv\_total*, and *vol\_mort*.

## Value

A list with two elements (see also details),

- *gyield\_summary*: A tibble that contains phase overarching growth and yield information.
- *gyield\_phases*: A list of tibbles, each one representing one of the growth and yield variables also found in *gyield\_summary*, but here on the level of the single stand development phases.

If the user has selected `detailed_out = TRUE`, there will also be another list element, *gyield\_pre*, which contains the interim information from which *gyield\_summary* and *gyield\_phases* are calculated.

## Examples

```
# Run a simulation and store the aggregated outcome
state_vars <- setup_statevars(
  pine_thinning_from_above_1, c(1000, 0, 0, 0, 0, 0)
)
time_span <- 200
parms <- setup_parms(pine_thinning_from_above_1)
parms$risk_mat <- setup_risk_events(
  time_span, avg_event_strength = 1, parms$risk
)

sim_areas_agg <- sim_area_single_concept_with_risk(
  state_vars,
  parms = parms,
  event_times = c(0:time_span),
  time_span = time_span
) |>
```



```
aggregate_raw_sim_rslt(pine_thinning_from_above_1)

# Growth and yield evaluation
growth_and_yield_evaluation(sim_areas_agg, pine_thinning_from_above_1)
```

---

is\_c4c\_concept            *Check if an Object is of Class c4c\_concept*

---

### Description

Check if an Object is of Class c4c\_concept

### Usage

```
is_c4c_concept(x)
```

### Arguments

x                    object to check

### Value

TRUE, if x has class c4c\_concept, FALSE if not

### Examples

```
data(pine_thinning_from_above_1)
x <- unclass(pine_thinning_from_above_1)

is_c4c_concept(pine_thinning_from_above_1)
is_c4c_concept(x)
```

---

new\_c4c\_concept            *Constructor for a c4c\_concept Object*

---

### Description

Constructor for a c4c\_concept Object

### Usage

```
new_c4c_concept(x = list())
```

### Arguments

x                    a list object

**Value**

Returns an object of class `c4c_concept`

**Examples**

```
# remove the c4c_class attribute for the example's sake
x <- unclass(pine_thinning_from_above_1)

x <- new_c4c_concept(x)
```

---

```
pine_no_thinning_and_clearcut_1
      pine_no_thinning_and_clearcut_1
```

---

**Description**

Generated from simulation runs to mimic a concept of treating a Scots pine stand with no thinnings and a short final harvest phase with the model SILVA.

**Usage**

```
pine_no_thinning_and_clearcut_1
```

**Format**

An object of class `c4c_concept` of length 3.

---

```
pine_thinning_from_above_1
      pine_thinning_from_above_1
```

---

**Description**

Generated from simulation runs to mimic a concept similar to the Scots pine management of the Bavarian State Forest with the model SILVA.

**Usage**

```
pine_thinning_from_above_1
```

**Format**

An object of class `c4c_concept` of length 3.

---

plot.c4c\_base\_result *Plot Function for c4c\_base\_result Objects*

---

## Description

Plot Function for c4c\_base\_result Objects

## Usage

```
## S3 method for class 'c4c_base_result'
plot(
  x,
  variable = c("area", "vol_standing", "vol_inc_ups", "vol_rmv_total", "vol_rmv_cont",
              "vol_rmv_damage", "vol_mort", "hrvst_det_reg", "hrvst_det_dam"),
  ...
)
```

## Arguments

x	Object of class c4c_base_result
variable	Character string, specifies the variable to be plotted. The options are: <ul style="list-style-type: none"> <li>• "area": The areas covered by the different stand development phases (SDPH) over time</li> <li>• "vol_standing": The standing volume by SDPH over time</li> <li>• "vol_inc_ups": The volume increment by SDPH (upscaled from the concept definition) over time</li> <li>• "vol_rmv_total": The total removed volume (comprising regular and damage-induced harvest) by SDPH over time</li> <li>• "vol_rmv_cont": The volume removed due to regular harvest by SDPH over time</li> <li>• "vol_rmv_damage": The volume removed due to damage-induced harvest by SDPH over time</li> <li>• "vol_mort": The volume losses due to mortality by SDPH over time</li> <li>• "hrvst_det_reg": The regular harvest volume over time by tree size classes (mean harvest diameter)</li> <li>• "hrvst_det_dam": The damage-induced harvest volume over time by tree size classes (mean harvest diameter)</li> </ul>
...	Other parameters, currently not used

## Value

A ggplot object

**Examples**

```

sim_base_out <- simulate_single_concept(
  pine_thinning_from_above_1,
  init_areas = c(1000, 0, 0, 0, 0, 0),
  time_span = 200,
  risk_level = 3
)

# Make a plot
plot(sim_base_out, variable = "area")
# Also try the following options for the parameter "variable":
# "vol_standing", "vol_inc_ups", "vol_rmv_total", "vol_rmv_cont",
# "vol_rmv_damage", "vol_mort", "hrvst_det_reg", "hrvst_det_dam"

```

---

plot.c4c\_co2\_result    *Plot Function for c4c\_co2\_result Objects*

---

**Description**

Plot Function for c4c\_co2\_result Objects

**Usage**

```

## S3 method for class 'c4c_co2_result'
plot(
  x,
  plot_type = c("em_by_type", "fl_by_type", "em_by_phase", "fl_by_phase", "em_vs_inc",
    "em_vs_hrv", "em_inc_ratio"),
  ...
)

```

**Arguments**

x	Object of class c4c_base_result
plot_type	Character string, specifies the kind of diagram to be plotted. The options are: <ul style="list-style-type: none"> <li>"em_by_type": The CO2 emissions by operation type (cutting, moving, forest road maintenance) over time</li> <li>"fl_by_type": The fuel consumption by operation type (cutting, moving, forest road maintenance) over time</li> <li>"em_by_phase": The CO2 emissions by stand development phase over time, not including forest road maintenance</li> <li>"fl_by_phase": The fuel consumption by stand development phase over time, not including forest road maintenance</li> <li>"em_vs_inc": The CO2 emissions plotted against the wood increment (in CO2 equivalents)</li> </ul>

- "em\_vs\_hrv": The CO2 emissions plotted against the harvest (in CO2 equivalents)
- "em\_inc\_ratio": The ratio of all CO2 emissions and the wood increment (in CO2 equivalents) over time
- ... Other parameters; currently not used

**Value**

A ggplot object

**Examples**

```
sim_co2_out <- simulate_single_concept(
  pine_thinning_from_above_1,
  init_areas = c(1000, 0, 0, 0, 0, 0),
  time_span = 200,
  risk_level = 3
) |>
  fuel_and_co2_evaluation(road_density_m_ha = 35, mode = "nordic")

# Make a plot
plot(sim_co2_out, plot_type = "em_by_type")
# Also try the plot types "fl_by_type", "em_by_phase", "fl_by_phase",
# "em_vs_inc", "em_vs_hrv", "em_inc_ratio"
```

---

 setup\_parms

*Parameter Setup*


---

**Description**

Given a `c4c_concept` concept definition, a list of parameter elements is handed back. This information is required for simulations and subsequent evaluations.

**Usage**

```
setup_parms(concept_def)
```

**Arguments**

`concept_def`      Concept definition as a `c4c_concept` object

**Details**

The element `risk` as part of the output describe as 'normal' risk as assumed for the silvicultural concept defined in `concept_def`. This can be adjusted with the parameter `avg_event_strength` of the function [setup\\_risk\\_events](#), which has to be called in any case after the parameter setup.

**Value**

A list with three elements. The first, `dwell_time`, is a vector of dwell times for each subphase area, i.e. it indicates the average time a unit area is dwelling in this subphase (assuming an exponential distribution over time). The second element, `risk`, is a vector of the same length and order. It represents, for each subphase area, the average relative loss rate per year. It is derived from the cumulative survival probabilities (`survival_com`) given in the data frame `growth_and_yield` which is part of the concept definition (`concept_def`). The third element, `phase_indexes`, is a tibble which contains, for each stand development phase in `concept_def`, a vector of indexes which can be used to easier access the phase wise information in the different kinds of simulation outputs.

**Examples**

```
parms <- pine_thinning_from_above_1 |> setup_parms()
parms
```

---

```
setup_risk_events      setup_risk_events
```

---

**Description**

Low-level function for setting up a risk matrix for a simulation run. Available for users who want to build simulation runs out of single elements. Regular users are recommended to use the function `simulate_single_concept` for running a simulation with one single command (where this function is internally used).

**Usage**

```
setup_risk_events(time_span, avg_event_strength = 1, area_risks)
```

**Arguments**

<code>time_span</code>	Simulation time span to be covered (integer)
<code>avg_event_strength</code>	Number which indicates the average strength of a damage event in the simulation. Default is 1 which means that the survival probabilities as defined in the silvicultural concept of interest are applied exactly as they are. A value of 2 would mean that one damage event would have the same effect as would two subsequent events with normal strength. A value of 0 would trigger no damage events at all.
<code>area_risks</code>	Vector of subphase-wise baseline damage risks, contained in the list made with <code>setup_parms</code> under the name <code>risk</code> .

## Details

The function uses exponentially distributed random numbers (with expectation = 1) for simulating the strength of damaging events. Such kind of distribution where small events are much more frequent than strong ones is a realistic assumption for forest damages. Such a random number is drawn for each simulation point in time. The actual damage strength (i.e. relative area loss) for a given subphase is then calculated as follows:

$$\text{rel\_area\_loss} = 1 - ((1 - x) \wedge \text{avg\_event\_strength}) \wedge \text{event\_strength},$$

where

- `x`: The baseline area loss risk of a given stand development subphase as resulting from the silvicultural concept definition of interest
- `avg_event_strength`: The user defined overall average event strength
- `event_strength`: Exponentially distributed random number with expectation 1, indicating the damage event strength in a given year

## Value

A matrix where each row is a point in simulation time, and each column represents a subphase of the silvicultural concept of interest (in increasing order). Each matrix element describes the relative area loss that will happen at a given time to a given subphase.

## Examples

```
parms <- setup_parms(pine_no_thinning_and_clearcut_1)
setup_risk_events(time_span = 200,
                 avg_event_strength = 3,
                 area_risks = parms$risk)
```

---

setup_statevars	<i>State Variable Setup and Initialization Given a c4c_concept concept definition, and a matching vector of initial areas the state variables required for simulating the concept are set up and initialized.</i>
-----------------	---

---

## Description

The state variables to be created are the areas attributed to the single stand development phases defined in the concept definition of interest. More precisely, each subphase has an area which is a state variable. When initializing, the initial areas can be given for each phase in `init_areas`, or for each subphase. In the former case the initial phase areas are equally divided among the respective subphases.

In order to allow post-hoc reconstruction of the area flows, the function also creates the cumulative inflows and outflows to each subphase area as state variables and initializes them with 0.

## Usage

```
setup_statevars(concept_def, init_areas, detailed = FALSE)
```

**Arguments**

concept_def	The concept definition of interest as a c4c_object
init_areas	A vector providing the initial areas for the stand development (sub) phases defined in concept_def in the same order. In case init_areas relates to the phases, detailed must be FALSE. If it relates directly to the subphases, detailed must be TRUE. In the former case, the areas given for a phase are equally divided among its subphases. The areas must be given in the same area unit as named in concept_def, usually ha.
detailed	Logical, FALSE (default) indicates that init_areas is given for each stand development phase (and will be equally distributed among the subphases). TRUE indicates that init_areas is directly given for each subphase.

**Value**

A vector which is actually a sequence of three different blocks. This format is required for simulations with `deSolve`. Each block has as many elements as the total number of subphases defined in concept\_def. Each element refers to each subphase in the order of the phase sequence. The first block, contains the initial areas attributed to all subphases in the order of the phase sequence. The second and the third block will track the cumulative in- and outflows of each area during the simulation. They are initialized with 0.

**Examples**

```
# Initialize with phase wise initial areas
init_areas <- c(1000, 400, 250, 125, 125, 100)
state_vars <- pine_thinning_from_above_1 |> setup_statevars(init_areas)
state_vars

# Initialize with subphase wise initial areas
# Assume, we are afforesting 1000 ha, so all area has to be initially in
# the first subphase of the first stand development phase
n_sub <- sum(pine_thinning_from_above_1$growth_and_yield$n_subphases)
init_areas <- rep(0, n_sub)
init_areas[1] <- 1000
state_vars <- setup_statevars(pine_thinning_from_above_1,
                             init_areas,
                             detailed = TRUE)

state_vars
```

---

simulate\_single\_concept

*Run a Simulation for a Single Silvicultural Concept*

---

**Description**

Top level function for running a simulation and obtaining all fundamental results, i.e. the simulated area dynamics and all growth and yield related outcomes.



**Usage**

```
simulate_single_concept(
  concept_def,
  init_areas,
  time_span,
  risk_level = 1,
  detailed_init = FALSE,
  detailed_out = FALSE,
  ...
)
```

**Arguments**

concept_def	Silvicultural Concept definition to be used in the simulation; a <code>c4c_concept</code> object
init_areas	The initial areas for each stand development phase defined in <code>concept_def</code> (if <code>detailed_init == FALSE</code> , default) or the initial areas for each stand development subphase as defined in <code>concept_def</code> (if <code>detailed_init == TRUE</code> )
time_span	Time span to be covered by the simulation
risk_level	Risk level relative to the standard risk level as defined in <code>concept_def</code> . The default is 1 which means, the standard risk level will be applied. If <code>risk_level == 0</code> , no damaging events will happen; e.g. <code>risk_level == 2</code> will increase damage probabilities as if the standard level risk events would occur two times.
detailed_init	Logical; is <code>init_areas</code> provided for each stand development phase (default, <code>FALSE</code> ), or for each subphase ( <code>TRUE</code> )?
detailed_out	Logical; should the output also include growth and yield pre-evaluation results (which are a very detailed interim evaluation output that is usually only required for internal efficiency)? The default is <code>FALSE</code> .
...	Additional arguments to <a href="#">sim_area_single_concept_with_risk</a>

**Details**

The output of this function is an object of class `c4c_base_result`. There is no other way to generate such an object, therefore there is neither a constructor nor a validator available to the user.

**Value**

An object of class `c4c_base_result` which is actually a named list containing all information that was used to define and set up a simulation, as well as all fundamental simulation results, i.e. the simulated area dynamics, and all growth and yield related results.

**Examples**

```
simulate_single_concept(
  pine_thinning_from_above_1,
  init_areas = c(1000, 0, 0, 0, 0, 0),
  time_span = 200,
```

```

    risk_level = 3
)

```

---

```

sim_area_single_concept_with_risk
    sim_area_single_concept_with_risk

```

---

### Description

Low-level simulation function for area-phase dynamics, available for users who want to compose simulations out of the single steps. Regular users are recommended to use the function [simulate\\_single\\_concept](#).

### Usage

```

sim_area_single_concept_with_risk(
  state_vars,
  parms,
  event_times,
  time_span = 100L,
  time_frac = 4L,
  integ_method = "lsoda"
)

```

### Arguments

state_vars	list of state variable(s) (vectors), all initialized
parms	list of parameter(s) (vectors)
event_times	vector of integers specifying the points in time when damage events can happen. Usually all numbers between (and including) zero and the endpoint of the simulation (i.e. $c(0, \text{time\_span})$ ).
time_span	simulation time span, integer, in the chosen time unit (typically years)
time_frac	integer $\geq 1$ , defines the time step to be used for numerical integration (time step = $1 / \text{time\_frac}$ ), i.e. one time unit will be split into <code>time_frac</code> substeps. Too small values of <code>time_frac</code> may cause the model showing chaotic dynamics, too large values increase computation time without meaningful gains in precision. The default value of 4 is mostly a good choice; in doubt, increase <code>time_frac</code> until the results do not change meaningfully anymore (risk events should be turned off during that procedure).
integ_method	integration method, passed to <a href="#">ode</a>

### Value

an object of class `deSolve`

**Examples**

```

# Work with the example data pine_thinning_from_above_1
# Initialize state variables (areas per stand development phase)
state_vars <- setup_statevars(pine_thinning_from_above_1,
                             c(1000, 0, 0, 0, 0, 0))

# Set time frame
time_span <- 200

# Initialize parameters
parms <- setup_parms(pine_thinning_from_above_1)

# Build risk matrix and add it to parms
parms$risk_mat <- setup_risk_events(
  time_span, avg_event_strength = 1, parms$risk
)

# Simulate
sim_area_single_concept_with_risk(
  state_vars,
  parms = parms,
  event_times = c(0:time_span),
  time_span = time_span
)

```

---

survival\_weibull

*Weibull-Based Estimates of Stand Survival*


---

**Description**

Estimates the probability of a forest stand to survive a period  $t$  after its establishment, based on the Weibull-Method published by Staupendahl (2011) Forstarchiv 82, 10-19.

**Usage**

```
survival_weibull(t, alpha, s_100)
```

**Arguments**

<code>t</code>	Time in years after stand establishment
<code>alpha</code>	Shape parameter. According to Staupendahl, Values $< 1$ indicate a high risk at young ages, $\alpha = 1$ indicates an indifference of the risk to $t$ (exponential distribution), values $> 1$ indicate high risk at old ages, where $1 < \alpha < 2$ means a degressively increasing, $\alpha = 2$ a constantly increasing, and $\alpha > 2$ a progressively increasing risk.
<code>s_100</code>	Survival probability up to $t = 100$

**Details**

The parameter `s_100` represents the survival probability after  $t = 100$  years, and `alpha` is the shape parameter, indicating the risk profile of the stand (type) of interest.

**Value**

The probability to survive up to  $t = 100$

**Examples**

```
# Calculations for Common oak, European beech, Norway spruce, Douglas fir,  
# and Scots pine with parameters after Staupendahl and Zucchini (AFJZ 2011)  
t <- seq(0, 120, 5)  
survival_weibull(t, alpha = 2.75, s_100 = 0.971) # oak  
survival_weibull(t, alpha = 1.76, s_100 = 0.967) # beech  
survival_weibull(t, alpha = 2.78, s_100 = 0.726) # spruce  
survival_weibull(t, alpha = 3.11, s_100 = 0.916) # Douglas  
survival_weibull(t, alpha = 2.45, s_100 = 0.923) # pine
```

---

validate\_c4c\_concept *Validator for a c4c\_concept Object*

---

**Description**

Validator for a `c4c_concept` Object

**Usage**

```
validate_c4c_concept(x)
```

**Arguments**

`x` an object of class `c4c_concept` to be validated

**Value**

Returns the input object if it passes validation, stops with an error otherwise

**Examples**

```
pine_thinning_from_above_1 |> validate_c4c_concept()  
pine_no_thinning_and_clearcut_1 |> validate_c4c_concept()
```

---

`wood_to_co2`*Convert Wood Volume to CO2 Equivalents*

---

**Description**

Convert Wood Volume to CO2 Equivalents

**Usage**

```
wood_to_co2(volume_m3, raw_density_kg_m3, water_perc = 12)
```

**Arguments**

<code>volume_m3</code>	Wood volume to be converted in cubic meters
<code>raw_density_kg_m3</code>	The raw density of the wood in kg/m <sup>3</sup>
<code>water_perc</code>	The water content of the wood at raw density in percent (usually defined as 12 %, i.e. air-dry)

**Value**

The CO2 equivalent of the input wood volume in kg

**Examples**

```
# Conversion of 1 m3 wood with typical values for important tree species
wood_to_co2(1, raw_density_kg_m3 = 520) # Scots pine
wood_to_co2(1, raw_density_kg_m3 = 470) # Norway spruce
wood_to_co2(1, raw_density_kg_m3 = 600) # European larch
wood_to_co2(1, raw_density_kg_m3 = 700) # sessile/pedunculate oak
wood_to_co2(1, raw_density_kg_m3 = 730) # European beech
```

# Index

## \* datasets

pine\_no\_thinning\_and\_clearcut\_1,  
18  
pine\_thinning\_from\_above\_1, 18

aggregate\_raw\_sim\_rslt, 2, 15  
avg\_extraction\_distance, 4

c4c\_concept, 4  
co2\_eval\_cutting, 5, 9  
co2\_eval\_moving, 6, 9

deSolve, 24

exp\_decay\_rate, 7

fuel\_and\_co2\_evaluation, 8  
fuel\_cons\_forwarder\_1, 7, 10  
fuel\_cons\_forwarder\_2, 7, 11  
fuel\_cons\_harvester\_1, 6, 12  
fuel\_cons\_harvester\_2, 6, 13  
fuel\_cons\_road\_maintenance, 14  
fuel\_to\_co2, 14

growth\_and\_yield\_evaluation, 15

is\_c4c\_concept, 17

new\_c4c\_concept, 4, 17

ode, 26

pine\_no\_thinning\_and\_clearcut\_1, 18  
pine\_thinning\_from\_above\_1, 18  
plot.c4c\_base\_result, 19  
plot.c4c\_co2\_result, 20

setup\_parms, 21, 22  
setup\_risk\_events, 21, 22  
setup\_statevars, 23  
sim\_area\_single\_concept\_with\_risk, 2, 3,  
25, 26

simulate\_single\_concept, 5, 6, 8, 22, 24, 26  
survival\_weibull, 27

validate\_c4c\_concept, 28

wood\_to\_co2, 9, 29