Harvested Wood Products vR Documentation

Jeremiah Groom

2022-04-08

Contents

1 Summary **5**

1.1 Acknowledgements 5

1.2 Abbreviations 6

2 Introduction **7**

2.1 General description of the HWP model 7

2.2 Model history 7

2.3 Model utility 10

2.4 HWP model assumptions and shortcomings 10

2.5 Future work 11

3 HWP Model Operations **13**

3.1 HWP model function 13

3.2 Monte Carlo simulation 25

4 Parts **41**

5 Footnotes and citations **43**

5.1 Footnotes 43

5.2 Citations 43

6 Blocks **45**

6.1 Equations 45

6.2 Theorems and proofs 45

6.3 Callout blocks 45

7 Sharing your book **47**

7.1 Publishing 47

7.2 404 pages 47

7.3 Metadata for sharing 47

Chapter 1

Summary

The Harvested Wood Products (HWP) model is a deterministic model which relies on optional and required user-supplied information. The model is described in (Stockmann et al. [2012]). There have been several builds of the model, on platforms including Microsoft Excel, C++, and now R [R Core Team, 2021]. Another effort is currently underway to construct an interactive version of the model in Python. This article describes a build of the model based on data from the U.S. west coast states of California, Oregon, and Washington. This R-based platform allows for user interaction with California and Oregon data via a web application (https://groomanalyticsllc.shinyapps.io/HWP_Shared_Test/). Users may also upload their own data onto the app, run their data through the HWP model and associated Monte Carlo simulation, interact with their data, and construct and save figures. Alternatively, users may access the full Shiny app code as well as a stand-alone version of the HWP model implementation. The stand-alone version allows users the opportunity to step through the code more easily, save model arrays for hand-calculation of outcomes, and alter the code to meet their own data needs. Both versions allow users to download or generate the tables necessary to recreate figures to their own specifications. This document describes the model and structure of the code, explains how to generate input files, and provides instruction on accessing model code.

1.1 Acknowledgements

Many thanks to Nadia Tase and Andrew Yost for supporting the development of this work through feedback, discussion, and their own efforts to dive into the code and understand the model at its core. Jimmy Kagan and Amanda Reynolds provided administrative support. We received feedback and support for the earlier iteration of the R-version of the model from Todd Morgan, Dan

Loffler, Keith Stockmann, [OTHERS].
This project was funded by the California Department of Forestry and Fire Protection Purchase Order 3540-0000277546. Earlier work on the project was funded by the Oregon Department of Forestry and supported by the Institute for Natural Resources, Oregon State University.

1.2 Abbreviations

The following description uses many acronyms for brevity. Here is a list of the most common:

BF: Board Feet
BBF: Billion Board Feet
BLM: Bureau of Land Management
BWEC: Burned with Energy Capture
EWEC: Emitted with Energy Capture
EWOEC: Emitted without Energy Capture
EUR: End-use Product Ratios
HWP: Harvested Wood Products
CCF: Hundred Cubic Feet
IPCC: Intergovernmental Panel on Climate Change
MBF: Thousand Board Feet), MT: Metric Tons
MMT C: Million Metric Tons of Carbon
PPR: Primary Product Ratios
PIU; Products in Use
SWDS: Solid Waste Disposal Sites
Tg C: Teragrams of Carbon: Tg C
Tg CO2e: Teragrams of Carbon Dioxide Equivalent
TPO: Timber Product Output
TPR: Timber Product Ratios
USFS: United States Forest Service

A note about units for carbon:
Publications from the United States have frequently described carbon storage and emissions using the metric million metric tons carbon, or MMT C. We are using the equivalent metric, teragrams of carbon (Tg C) in the HWP model and this manual to align with IPCC reporting requirements.

Chapter 2

Introduction

2.1 General description of the HWP model

An overview schematic of the HWP model is provided in Figure 2.1. For every year of consideration, timber harvest results in board-feet of timber arriving at mills. These harvest values may be subdivided by the land ownership that produced the timber. Timber values are multiplied by the timber product classes, then the primary product ratios for each timber products class, and finally the end-use ratios for each primary products class. These resulting values are transformed to Tg C. Some end-use ratios involve material that is burned for energy capture. Other end-use products become products in use, although the process of creating those results in a percentage of the material ending up as discarded products. Material entering discarded products is proportionally allocated to dumps, landfill, compost, or are recovered (recycled) or burned without energy capture.

In the subsequent year more timber enters the cycle. However, the previous year's carbon in products in use, dumps, recovered products, and the decaying portion of landfills are subjected to a half-life decay. The outputs either end up in the discard category or are emitted to the atmosphere as CO_2.

2.2 Model history

The HWP model began as an Excel macro for the USFS, as described by Stockmann et al. [2012]. A newer version of the tool (v1) was made available in 2012 [Loeffler et al., 2014]. The states of Oregon, California and Washington are signatories to a Memorandum of Understanding for Pacific Coast Temperate Forests (MOU). The MOU partners have a common requirement to generate reliable estimates of the stock and flux of carbon in harvested wood products.

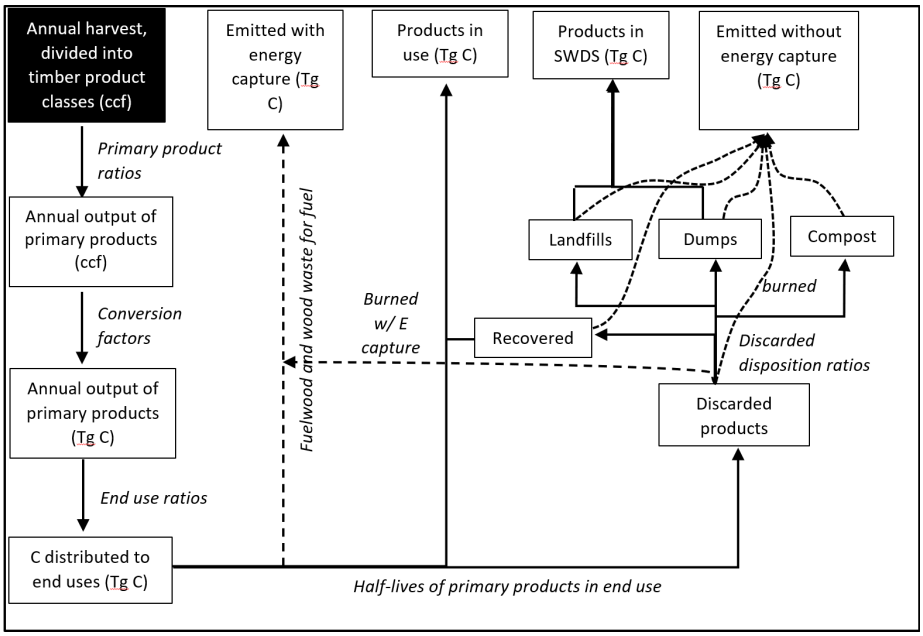


Figure 2.1: General schematic for calculations to quantify HWP storage and emissions, based on @stockmann2012 Figure 7. Note that in this version Compost is not connected to SWDS, an arrow to SWDS has been added for emissions from Recovered, and that the code calculates end-use C distribution in one step after organizing TPO, PPR, EUR, and conversion factors.

A west-coast specific state-level variant of the model was developed in C++; however, it did not fully account for the carbon mass entering the cycle from timber harvest, nor did it generate reliable error estimates using a Monte Carlo simulation approach. Perhaps most importantly, the software was not in a form that could be readily viewed, understood, or changed by others. Representatives from each west-coast state and the USFS agreed that the software needed to be re-written in order to correct these errors.

Groom Analytics LLC undertook programming the Stockmann et al. [2012] model in R for Oregon, and subsequently deployed the R version to a web application (https://groomanalyticsllc.shinyapps.io/HWP_Shared_Test/). This document and the available code are publicly released to enable model transparency and assist in further model development.

The USFS is currently developing a Python-based version of the model to be released soon. A schematic of this history is presented in Figure 2.2.

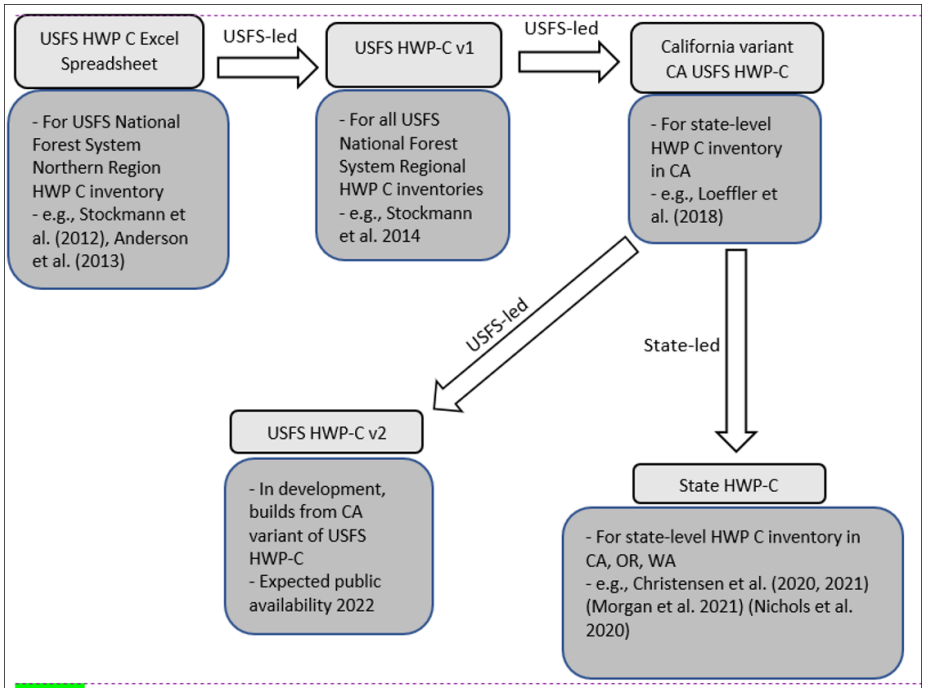


Figure 2.2: Schematic of the HWP model development pathway. The version described here is the State HWP-C variant.

2.3 Model utility

The HWP model (and associated visualizations) is a long-term accounting tool for carbon emissions and storage from harvested wood products. Users can easily understand how harvest levels have changed over time, by ownership and in total. Given changes in disposal methods and the assumed lifetime use of wood products, users can examine what proportion of harvested wood products remain in use, in disposal sites, or as CO_2 emitted to the atmosphere. The model may be useful for planners and decision-makers to understand the emissions consequences of promoting wood products with different use lifespans.

2.4 HWP model assumptions and shortcomings

The HWP model is a closed model, and tracks only carbon harvested within a specified area. Therefore, at a state level, emissions from HWP may be greater than expected from timber harvested within the state if the state imports timber or timber products. Similarly, exported timber or timber products may experience different utilization or decay processes.

The model assumes that harvested timber volume and ratios such as TPR, PPR, and EUR are valid. The model can be expected to perform poorly if the ratios or other values are not based on solid evidence. The Monte Carlo attempts to explore the range of possible outcomes given different levels of certainty in the data. The model does, ultimately, rely on many different data types. Some of these, such as landfill decay rates, may be difficult to enumerate and thus we are unsure of their true values.

The model does not (currently) differentiate between carbon in landfills that is emitted as methane vs. CO_2. Methane is a much more potent greenhouse gas than CO_2, albeit with a shorter residence time.

The Monte Carlo takes a specific approach towards varying the ratios, constraining all ratios to sum to one. Given the selected range of possible values for each variable, the resulting randomly selected values may not be independent from one another and may be unintentionally biased. That is, within a ratio type that must sum to one and for a particular model iteration, if one value is randomly selected to equal 1.0, the remaining values for that ratio set must equal 0.

Finally, we are not aware of any method for overall model validation. Estimates for individual pieces of the model, such as product or decay half-lives, can be improved. But, like all models, the HWP model simplifies reality greatly and hopefully provides users with useful insight by tracking and summarizing complicated calculation outputs in a reasonable way.

2.5 Future work

Users can improve the model's utility by researching and altering model input tables with their own regional information. There is interest in incorporating a means for tracking the production of CO_2 separately from methane as methane is a much more powerful greenhouse gas that results from anoxic decomposition such as occurs in landfills. In the short term we plan to update the model so that users can control the discard ratios for paper and wood products. Finally, work is currently being done at the University of Massachusetts to explore the consequences of errors in estimating the capability of landfills to retain carbon without decay.

Chapter 3

HWP Model Operations

3.1 HWP model function

3.1.1 Model inputs

The HWP model is a deterministic model which relies on optional and required user-supplied information. The user must supply annual timber harvest volumes in board feet (BF) by ownership, annual timber product ratios (TPR) that allocate harvest to different timber product classes, annual primary product ratios (PPR) that allocate timber products to different primary products and residue uses, and annual end-use product ratios (EUR) which are subcategories of PPR that represent the final products produced.

The user may, at a minimum, provide timber harvest volumes and then rely on default California and Oregon values. These include a board foot to cubic foot conversion factors for sets of years and conversion factors between hundred cubic feet (CCF) and metric tons of carbon (MT C) or teragrams of carbon (Tg C) for the primary products. The model uses end-use half-life values to determine how long carbon resides in Products in Use (PIU) before becoming discarded.

The user may also provide ratios for discard fates. These ratios determine, by year, what fraction of discarded paper and wood are channeled into one of six discard categories: burned (Emitted without Energy Capture, EWOEC), burned with energy capture (BWEC), composted, sent to a dump, enter the non-decaying portion of a landfill, or become landfill subject to decay. Users may also adjust half-life values for paper and wood discards in landfills and dumps and the fraction of material entering landfills that is subject to decay or remains permanently.

3.1.2 Summary of model operations

In the next section I provide a detailed schematic of how the HWP model code operates. A more general schematic of the process, very similar to Figure 7 in Stockmann et al. [2012], is presented in the Introduction (Figure 1). Here I provide a written description of the code's actions. We start out with data sets and variable information that are arranged by different combinations of year, ownership, and ratio types. First the model combines (i.e., performs a database join function) harvest data (units = BBF) with board feet to cubic feet conversion factors. The model combines Primary Product Ratios (PPR) with Timber Product Ratios (TPR) and then joins those with the timber harvest volume in cubic feet by year. These are then joined with conversion values for cubic feet to metric tons of carbon, and the End Use Ratios (EUR). The result is a data set called eu_ratios, and values for EUR, TPR, PPR, the harvest amount, and the cubic foot to metric tons of carbon conversion factor are multiplied together to produce a variable called MTC for Metric Tons of Carbon. This variable serves as the carbon input into the model and has the correct units of carbon input per year for each EUR category.

For each year, some of the carbon, fuel wood, is immediately burned and is Emitted with Energy Capture (EEC). A certain percentage (8% for California and Oregon data) of the carbon allocated to each of the remaining EUR categories is discarded and enters the Discarded Products process and the remaining 92% become the annual carbon addition to the Products in Use (PIU) process. The model currently assumes that 100% of pulp wood becomes paper with no loss.

The Products in Use process calculates, for every year, the amount of carbon remaining from the year prior and the amount of carbon entering the process as new PIU. A decay function considers the previous year's carbon amounts in each EUR/ownership category, applies a decay calculation, adds the result to the amount of incoming PIU carbon, and saves the value as the amount of PIU carbon for that year. It also records the amount of carbon that was discarded from the process and sends that carbon to the Discarded Products process.

The Discarded Products process uses proportions to divide incoming discarded products among six fates: burn, burn with energy capture, compost, dump, landfill, and recover (recycle). Burned and composted proportions are treated as EWOEC. The proportion that is burned with energy capture will be treated as EEC (for California and Oregon all carbon is currently modeled as BWOEC). Discarded Products that are recovered or sent to dumps are processed with the same decay function used for Products in Use, but with different half-life values. The outputs that do not decay in a given year are treated either as Products in Use (recovered) or Solid Waste Disposal Site (SWDS) material (dumps). Discarded Products that are sent to Landfill are treated like Dumps, except that a proportion of the landfill is not subject to decay. That amount will become one of the carbon pools that comprise SWDS. The remainder of carbon

in the landfill decays and is treated as EWOEC or remains in the landfill and is counted as another SWDS carbon pool: landfill material subject to decay.

The HWP code's outputs are EEC and combined versions of EWOEC, PIU, and SWDS. The final EEC values are the original EEC values for fuel wood plus the fraction of the discarded materials that are burned with energy capture. The final EWOEC amounts are combinations of emissions from dumps, recovered materials, landfill decay, burnt refuse, and compost materials. PIU is the combination of the original PIU values plus recovered carbon that has not decayed. SWDS is the non-decaying and decaying carbon in landfills plus remaining carbon in dumps.

3.1.3 Model schematic description

Below I provide a schematic of how the HWP model operates. The schematic color codes items and flows from the top to bottom of each page. Inputs, such as worksheets from the Excel data file, are yellow. Calculated matrices or arrays are blue. Final output, which may be manipulated to produce tables and figures, are purple. Calculations or data transformations are peach colored.

The file and object names listed in the schematic may be found in the provided model code. My intent is to provide a written description (above) that assists readers with understanding the schematic, which in turn can be used to interpret how the R code (HWP_Model_Function.R) works. The R code itself is annotated. Readers may alter the code to incorporate their own files or improve upon the model (see Operating and Changing the Model).

Transformations include database functions such as pivots and joins. A pivot in this case usually involves lengthening a matrix by taking several columns of data and stacking them into one column, with a second column identifying which of the original columns they came from. This format is useful for combining data with different levels of information. For instance, we might have data with rows = different ownerships and columns = years. We can pivot those data such that there is a column of ownership, a column for the years (years will be repeated for each block of ownerships), and a column for the values. Say we have another pivoted data set that has data for ownership, years, and Timber Product Ratios. We can join the two data sets by matching years and ownership values and duplicating the data values from the first data set across all of the Timber Product Ratios for each year/ownership combination. A somewhat more complicated version of this example is described in the Model Start page of the schematic.

Ultimately, the code may produce a final pivot length of well over 100,000 rows. This is equal to the number of End Use Ratio values * number of ownerships (including a Total column) * number of years. For instance, the eu_ratio matrix has this many rows and several useful variables that are used to calculate metric tons of carbon from harvest as distributed across EURs, years, and ownerships.

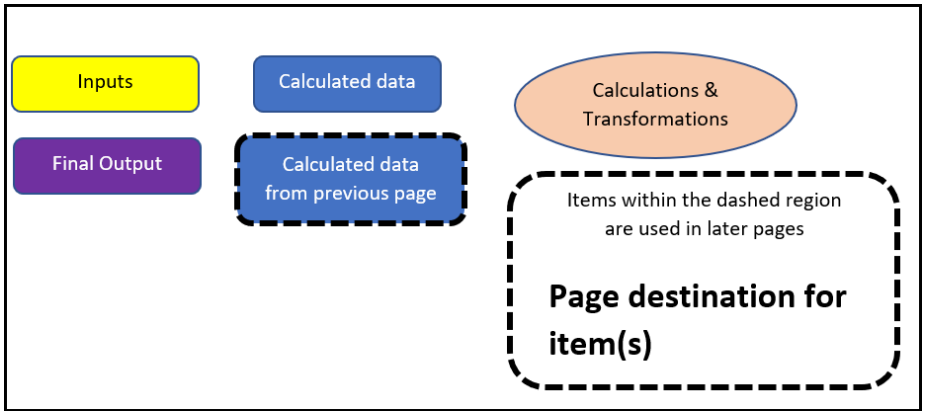
This value is then placed in a three-dimensional array with row number = number of EUR rows, depth = ownership number, and column number = number of years of data. The HWP code frequently uses arrays of these dimensions for storing values from many types of calculated data.

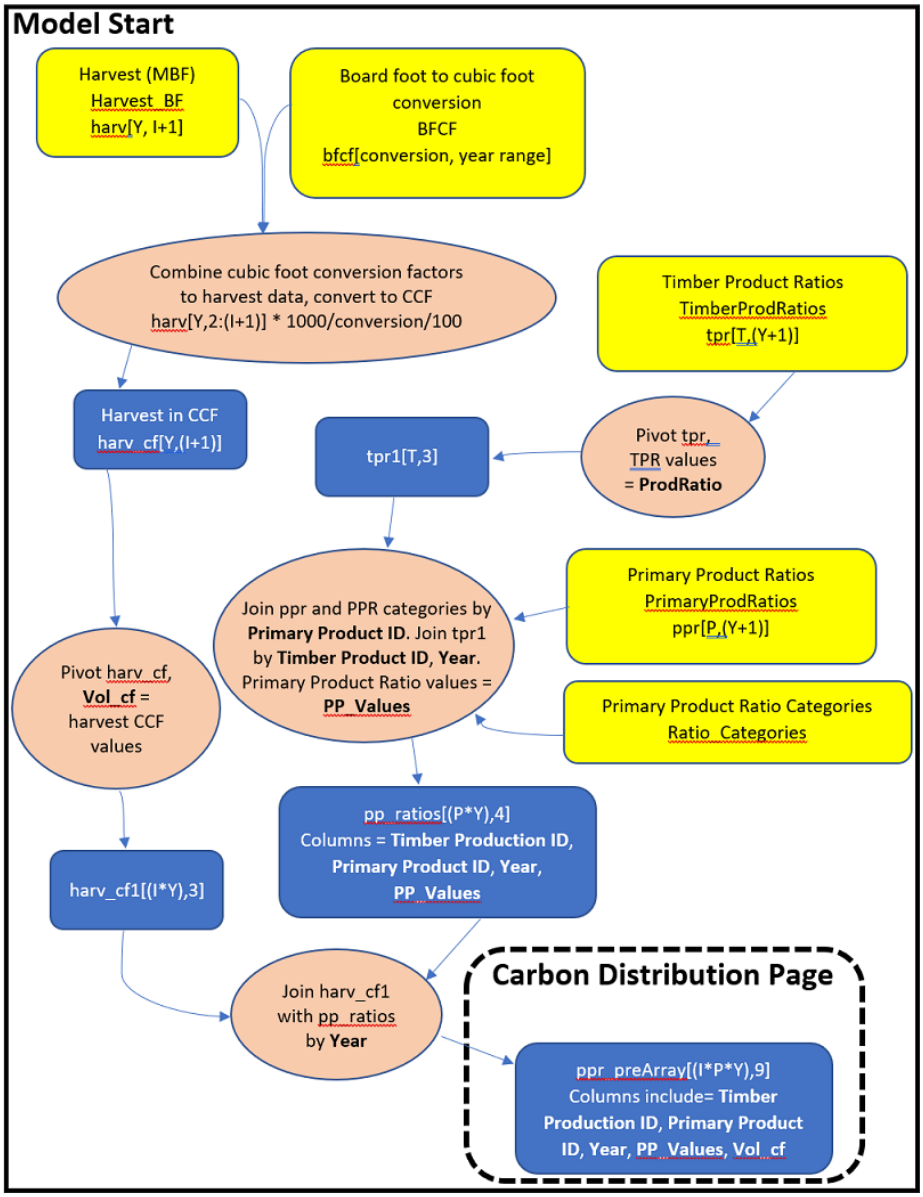
Each page of the following schematic has a title. These titles may be referred to within the schematic, as certain program products are used in later pages. If a calculated data set's boarders have thick black dashes, the data set is from a previous page. If a calculated data set lies within some white space with a thick black dashed boarder, the white space includes information on which page that product is destined for.

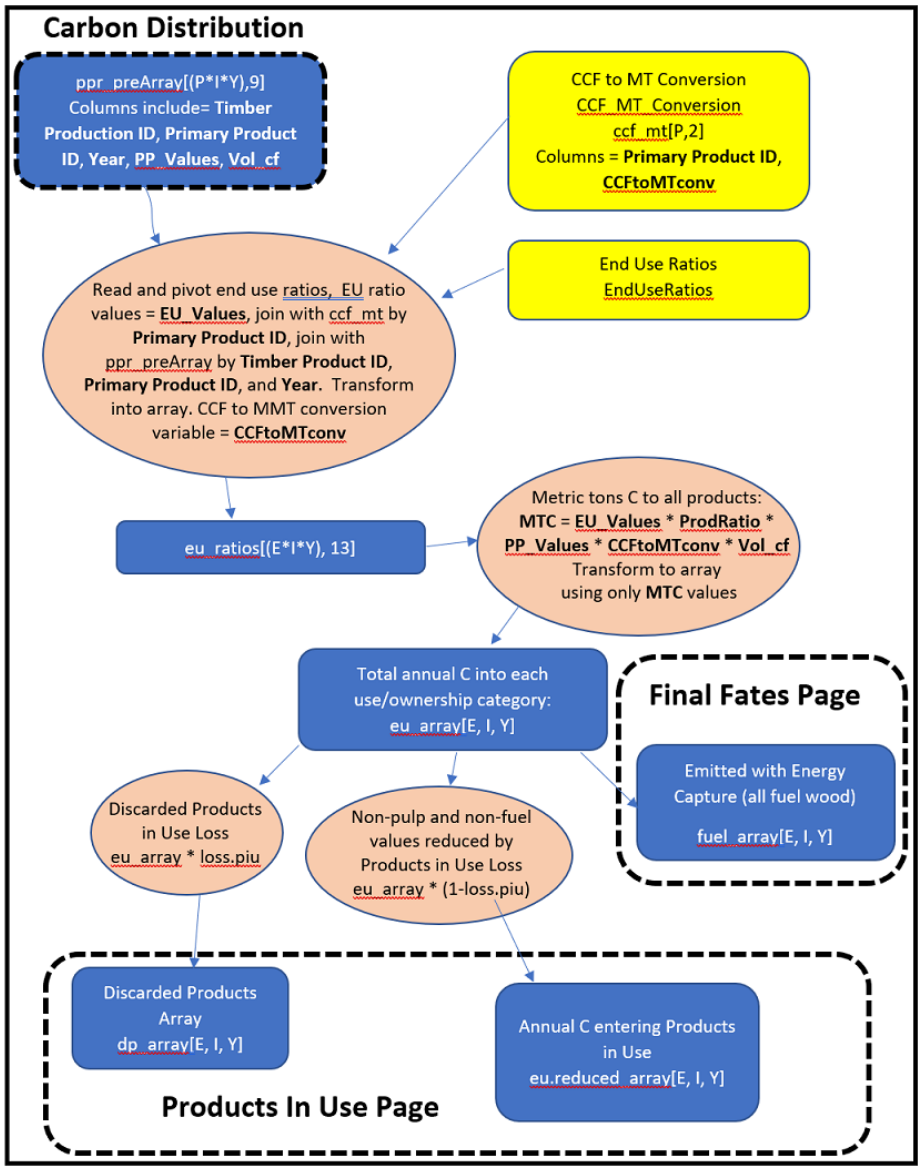
The HWP program has a decay function that is used repeatedly. The same function operates on Products in Use and the decaying portion of landfill, dumps, and recovered (recycled) products. As an example, for PIU, the function iterates through years for a given ownership/EUR combination. It calculates the amount of PIU carbon carried over from a previous year by subjecting that carbon to its appropriate half-life value. Then it adds that carbon to the incoming PIU carbon for that year. Then it moves forward one year and repeats.

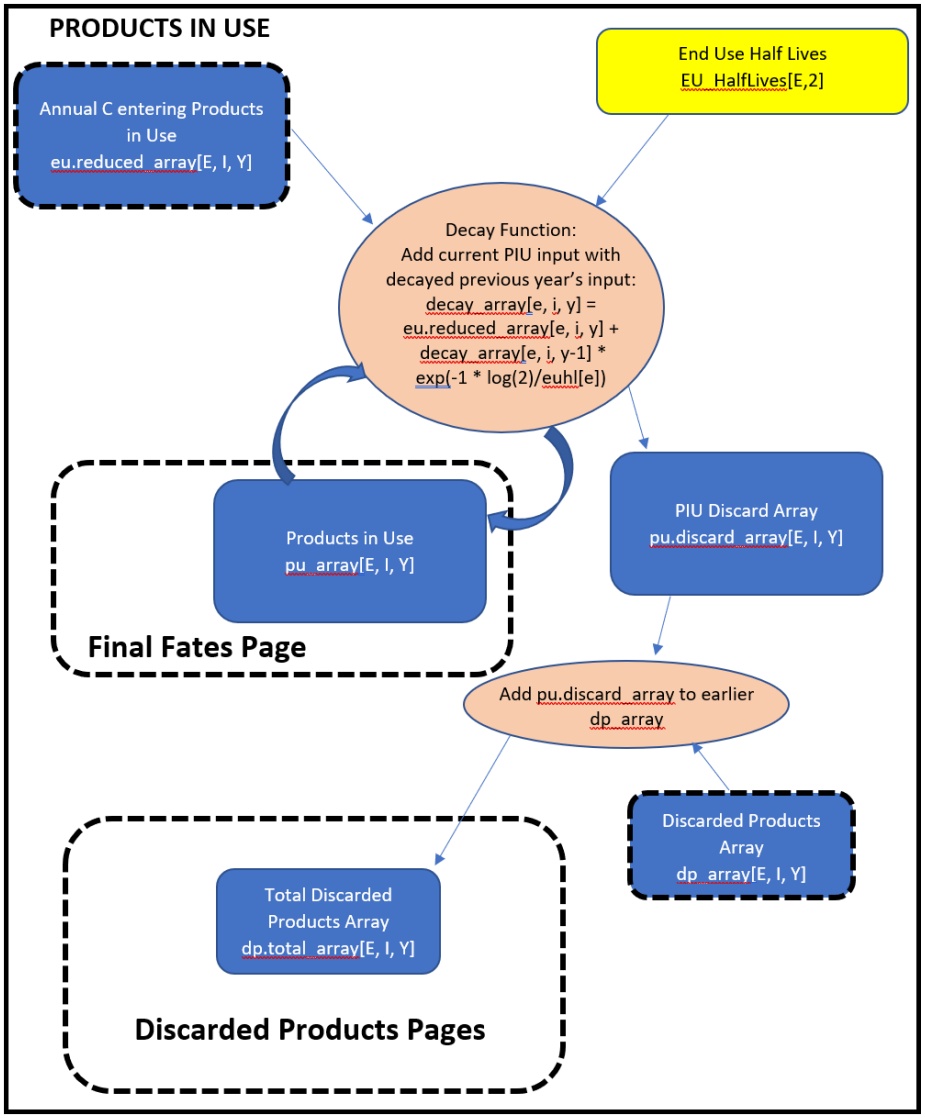
3.1.4 Model Schematic

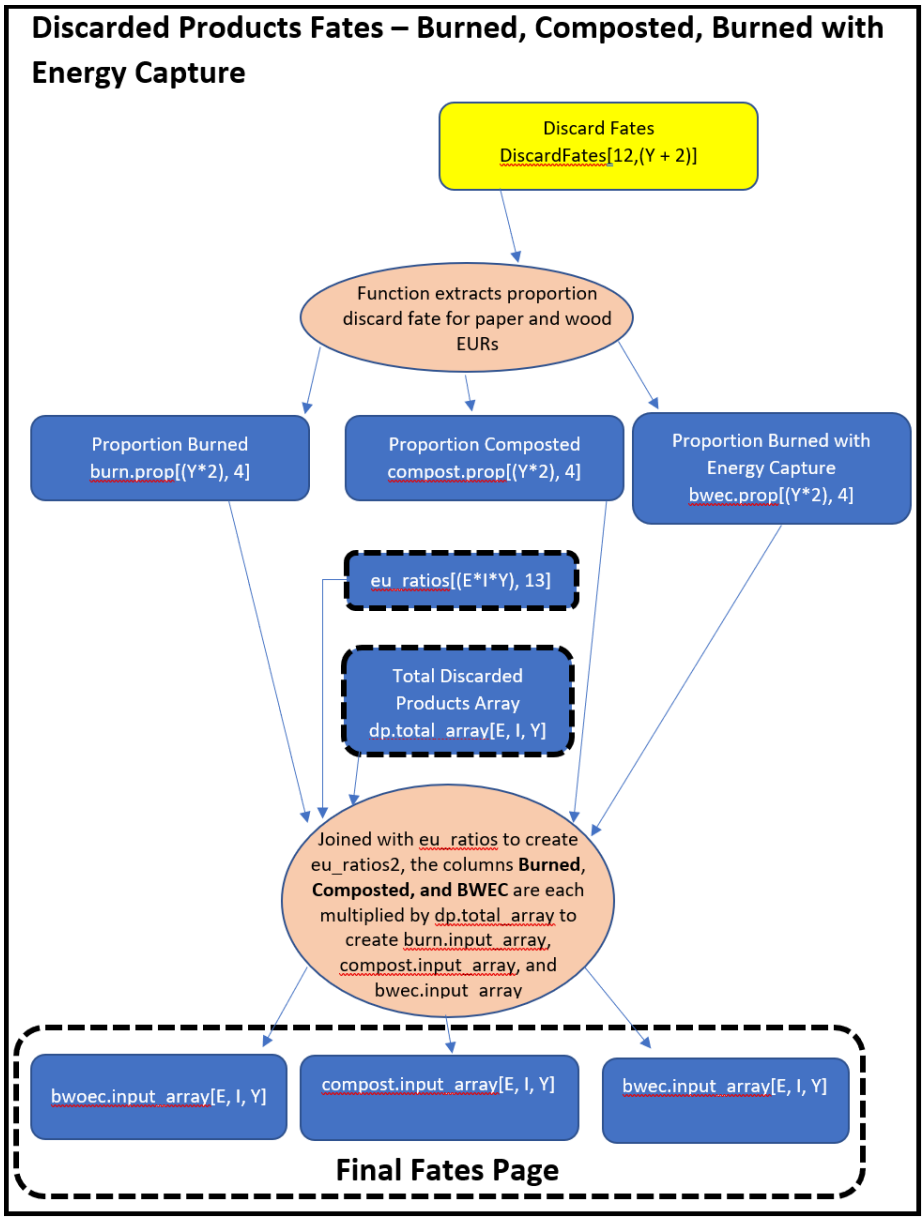
Model Dimensions Key:
Y = number of years (California = 68, from 1952 to 2019)
y = particular year
I = ownerships including Total (California = 6)
i = particular ownership
T = Timber product ratio number (California = 40)
P = Primary product ratio number (California = 64)
E = End use ratio number (California = 224)
e = particular EUR
loss.piu = Loss percentage for products entering PIU (California = 8%)

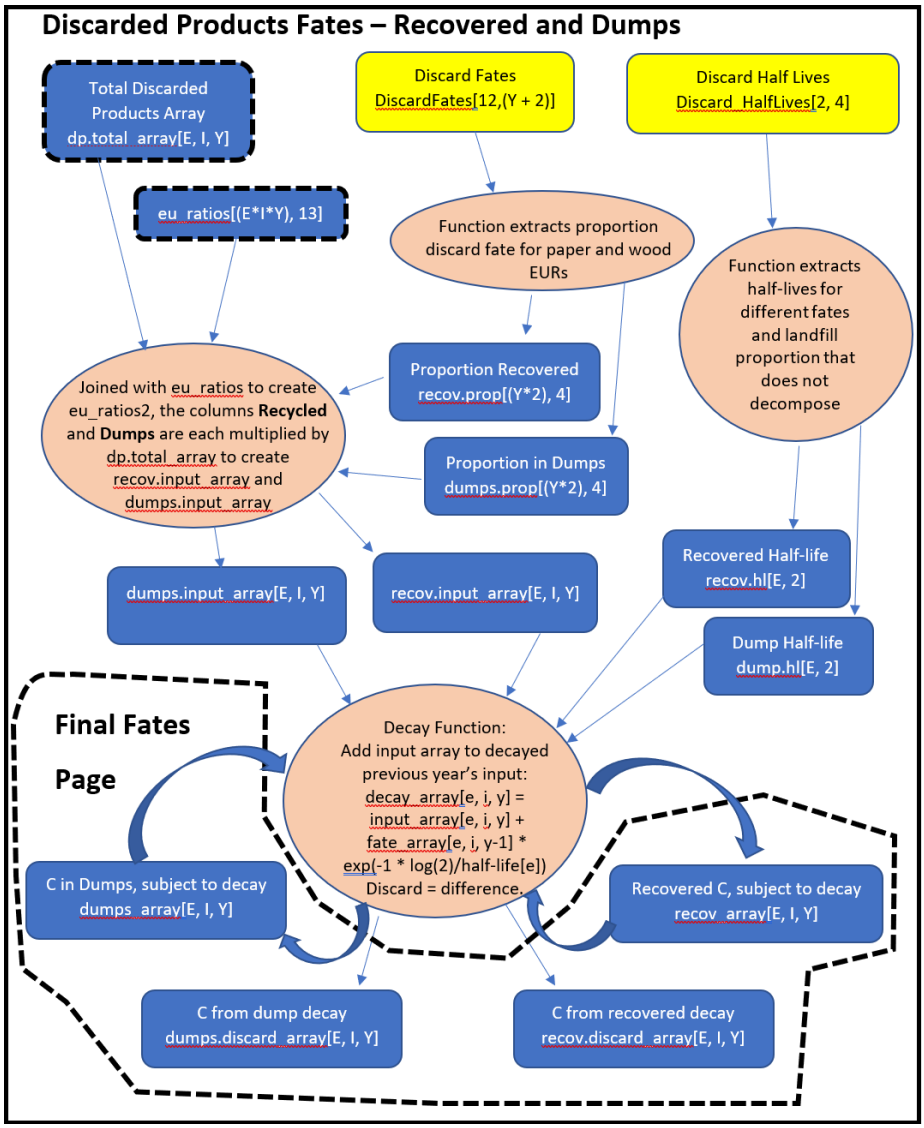


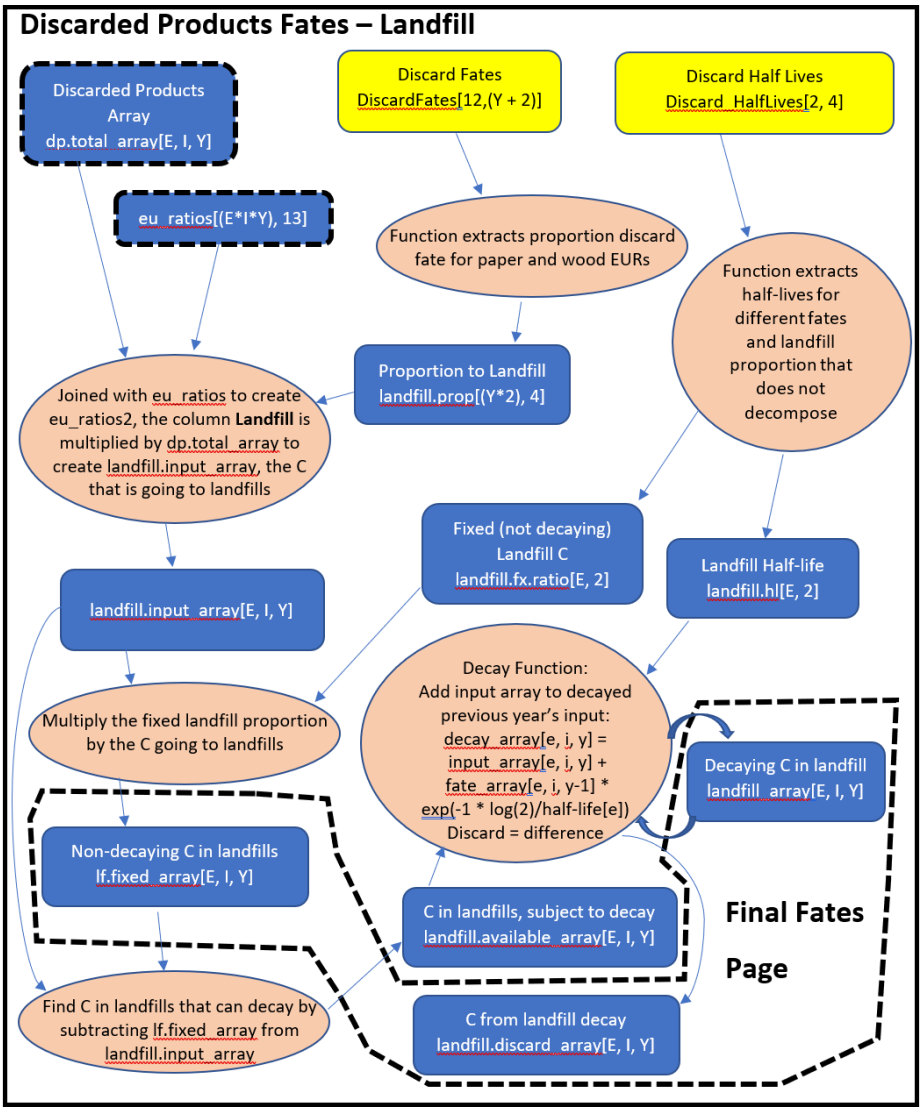

Model Start
Harvest (MBF)
Harvest_BF
harv[Y, I+1]
Board foot to cubic foot conversion
BFCF
bfcf[conversion, year range]
Combine cubic foot conversion factors to harvest data, convert to CCF
harv[Y,2:(I+1)] * 1000/conversion/100
Timber Product Ratios
TimberProdRatios
tpr[T,(Y+1)]
Harvest in CCF
harv_cf[Y,(I+1)]
tpr1[T,3]
Pivot tpr, TPR values = ProdRatio
Join ppr and PPR categories by Primary Product ID. Join tpr1 by Timber Product ID, Year. Primary Product Ratio values = PP_Values
Primary Product Ratios
PrimaryProdRatios
ppr[P,(Y+1)]
Pivot harv_cf, Vol_cf = harvest CCF values
Primary Product Ratio Categories
Ratio_Categories
pp_ratios[(P*Y),4]
Columns = Timber Production ID, Primary Product ID, Year, PP_Values
harv_cf1[(I*Y),3]
Join harv_cf1 with pp_ratios by Year
Carbon Distribution Page
ppr_preArray[(I*P*Y),9]
Columns include= Timber Production ID, Primary Product ID, Year, PP_Values, Vol_cf

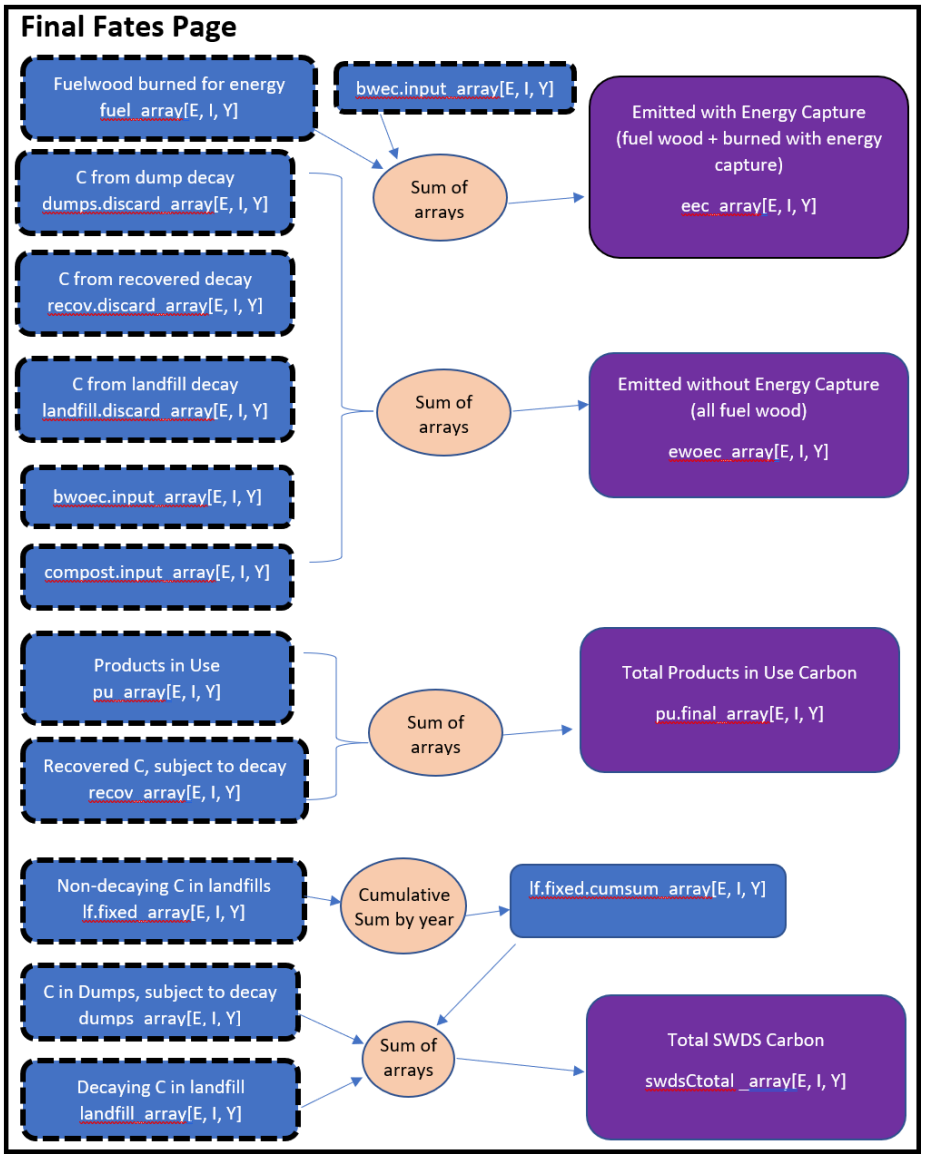

Carbon Distribution
ppr_preArray[(P*I*Y),9]
Columns include= Timber Production ID, Primary Product ID, Year, PP_Values, Vol_cf
CCF to MT Conversion
CCF_MT_Conversion
ccf_mt[P,2]
Columns = Primary Product ID, CCFtoMTconv
End Use Ratios
EndUseRatios
Read and pivot end use ratios, EU ratio values = EU_Values, join with ccf_mt by Primary Product ID, join with ppr_preArray by Timber Product ID, Primary Product ID, and Year. Transform into array. CCF to MMT conversion variable = CCFtoMTconv
eu_ratios[(E*I*Y), 13]
Metric tons C to all products:
MTC = EU_Values * ProdRatio * PP_Values * CCFtoMTconv * Vol_cf
Transform to array using only MTC values
Total annual C into each use/ownership category:
eu_array[E, I, Y]
Final Fates Page
Emitted with Energy Capture (all fuel wood)
fuel_array[E, I, Y]
Discarded Products in Use Loss
eu_array * loss.piu
Non-pulp and non-fuel values reduced by Products in Use Loss
eu_array * (1-loss.piu)
Discarded Products Array
dp_array[E, I, Y]
Annual C entering Products in Use
eu.reduced_array[E, I, Y]
Products In Use Page


PRODUCTS IN USE
Annual C entering Products in Use
eu.reduced_array[E, I, Y]
End Use Half Lives
EU_HalfLives[E,2]
Decay Function:
Add current PIU input with decayed previous year's input:
decay_array[e, i, y] =
eu.reduced_array[e, i, y] +
decay_array[e, i, y-1] *
exp(-1 * log(2)/euhl[e])
Products in Use
pu_array[E, I, Y]
Final Fates Page
PIU Discard Array
pu.discard_array[E, I, Y]
Add pu.discard_array to earlier dp_array
Discarded Products Array
dp_array[E, I, Y]
Total Discarded Products Array
dp.total_array[E, I, Y]
Discarded Products Pages


Discarded Products Fates – Burned, Composted, Burned with Energy Capture
Discard Fates
DiscardFates[12,(Y + 2)]
Function extracts proportion discard fate for paper and wood EURs
Proportion Burned
burn.prop[(Y*2), 4]
Proportion Composted
compost.prop[(Y*2), 4]
Proportion Burned with Energy Capture
bwec.prop[(Y*2), 4]
eu_ratios[(E*I*Y), 13]
Total Discarded Products Array
dp.total_array[E, I, Y]
Joined with eu_ratios to create eu_ratios2, the columns Burned, Composted, and BWEC are each multiplied by dp.total_array to create burn.input_array, compost.input_array, and bwec.input_array
bwoec.input_array[E, I, Y]
compost.input_array[E, I, Y]
bwec.input_array[E, I, Y]
Final Fates Page


Discarded Products Fates – Recovered and Dumps
Total Discarded Products Array dp.total_array[E, I, Y]
Discard Fates DiscardFates[12,(Y + 2)]
Discard Half Lives Discard_HalfLives[2, 4]
eu_ratios[(E*I*Y), 13]
Function extracts proportion discard fate for paper and wood EURs
Function extracts half-lives for different fates and landfill proportion that does not decompose
Proportion Recovered recov.prop[(Y*2), 4]
Joined with eu_ratios to create eu_ratios2, the columns Recycled and Dumps are each multiplied by dp.total_array to create recov.input_array and dumps.input_array
Proportion in Dumps dumps.prop[(Y*2), 4]
Recovered Half-life recov.hl[E, 2]
dumps.input_array[E, I, Y]
recov.input_array[E, I, Y]
Dump Half-life dump.hl[E, 2]
Final Fates Page
Decay Function: Add input array to decayed previous year's input: decay_array[e, i, y] = input_array[e, i, y] + fate_array[e, i, y-1] * exp(-1 * log(2)/half-life[e]) Discard = difference.
C in Dumps, subject to decay dumps_array[E, I, Y]
Recovered C, subject to decay recov_array[E, I, Y]
C from dump decay dumps.discard_array[E, I, Y]
C from recovered decay recov.discard_array[E, I, Y]


Discarded Products Fates – Landfill
Discarded Products Array dp.total_array[E, I, Y]
Discard Fates DiscardFates[12,(Y + 2)]
Discard Half Lives Discard_HalfLives[2, 4]
eu_ratios[(E*I*Y), 13]
Function extracts proportion discard fate for paper and wood EURs
Function extracts half-lives for different fates and landfill proportion that does not decompose
Proportion to Landfill landfill.prop[(Y*2), 4]
Joined with eu_ratios to create eu_ratios2, the column Landfill is multiplied by dp.total_array to create landfill.input_array, the C that is going to landfills
Fixed (not decaying) Landfill C landfill.fx.ratio[E, 2]
Landfill Half-life landfill.hl[E, 2]
landfill.input_array[E, I, Y]
Decay Function: Add input array to decayed previous year's input: decay_array[e, i, y] = input_array[e, i, y] + fate_array[e, i, y-1] * exp(-1 * log(2)/half-life[e]) Discard = difference
Multiply the fixed landfill proportion by the C going to landfills
Decaying C in landfill landfill_array[E, I, Y]
Non-decaying C in landfills lf.fixed_array[E, I, Y]
C in landfills, subject to decay landfill.available_array[E, I, Y]
Final Fates Page
Find C in landfills that can decay by subtracting lf.fixed_array from landfill.input_array
C from landfill decay landfill.discard_array[E, I, Y]



3.1.5 Model Outputs

The model described above treats timber harvest, product manufacture, and disposal as processes that occur within the same year. To match other reports, outputs are altered (SHIFTYEAR = TRUE; see Providing Your Own Data / Creation of the input file). All trees harvested in a given year are treated as processed within that year and the carbon is distributed according to the end use product ratios of that same year. Most of the carbon is immediately burned

or turned into Products in Use. A fraction (the default 8%) is immediately discarded and distributed to the different discard fates using the discarded disposition ratios for the same year. The carbon fate for a given year is reported for the following year. Carbon that enters the model in a given year does not undergo a half-life decay until the following year. For example, the model uses the Timber Production Output for 1906 and all product and discard ratios for 1906 to distribute the carbon in Products in Use, Burned with Energy Capture, and the discard fates. The sum of those fates for 1906 is recorded in tables as occurring by January 1, 1907. Any carbon from 1906 that was distributed to Products in Use, Landfill, or Dumps was subject to a half-life decay or disposal in 1907.

When users upload their data to the Shiny application or run the stand-alone model, they have the opportunity to download or generate summary tables of the model output. The provided tables should allow, with effort, the user to reproduce all Shiny application figures except for those associated with the Sankey diagram.

The stand-alone code can optionally save the arrays used to generate the tables in the folder "Arrays". Unlike the table outputs for stocks and emissions the arrays are not year-shifted. The arrays are provided for results-checking and data exploration. The arrays are three dimensional but are saved as Excel workbooks. Excel worksheet rows represent the end use ratios, columns represent years, and each worksheet tab is an ownership with the final tab representing the "Total" carbon value. Some arrays are combinations of other arrays, described below. File "eu_array.xslx" is the distribution of carbon (metric tons) to end uses. Values are used to convey how harvested wood in each ownership category translates to Tg C for each of the end use products. This array is effectively the starting point for all C fates.

Users are encouraged to generate the arrays (OUTPUT_ARRAYS = TRUE) in the stand-alone model and verify calculations. The above flow diagram provides array names and a general description of how they are used. The code file HWP_Model_Function.r provides more detail about how they are used.

3.2 Monte Carlo simulation

3.2.1 Simulation specifications

The Monte Carlo (MC) simulation alters the values of at least 16 different parameters, listed in Table 3.1. The columns Min Value, Peak Value, and Max Value describe the desired 90% confidence interval for the range of random proportions against which the parameters of interest will be adjusted. In Table 3.1 harvest, timber product ratios and primary product ratios each have two parameters with confidence intervals. These parameters represent different year

Table 3.1: Monte Carlo simulation target parameters and ranges for provided California data.

Parameter ID	Parameter Name	First Year	Last Year	Min Value	Peak Value	Max Value	Sum to One
1	CCF to MgC conversion factors	n/a	n/a	0.95	1	1.05	No
2	End use product half lives	n/a	n/a	0.85	1	1.15	No
3	End use product ratios	n/a	n/a	0.85	1	1.15	Yes
4	Discarded disposition ratios (paper)	n/a	n/a	0.85	1	1.15	Yes
5	Discarded disposition ratios (wood)	n/a	n/a	0.85	1	1.15	Yes
6	Landfill decay limits (paper)	n/a	n/a	0.85	1	1.15	No
7	Landfill decay limits (wood)	n/a	n/a	0.85	1	1.15	No
8	Landfill half lives (paper)	n/a	n/a	0.85	1	1.15	No
9	Landfill half lives (wood)	n/a	n/a	0.85	1	1.15	No
10	Dump half lives (paper)	n/a	n/a	0.85	1	1.15	No
11	Dump half lives (wood)	n/a	n/a	0.85	1	1.15	No
12	Recovered half lives (paper)	n/a	n/a	0.85	1	1.15	No
13	Recovered half lives (wood)	n/a	n/a	0.85	1	1.15	No
14	Harvest	1952	1979	0.80	1	1.20	No
15	Harvest	1980	2100	0.85	1	1.15	No
16	Timber product ratios	1952	1979	0.80	1	1.20	Yes
17	Timber product ratios	1980	2100	0.85	1	1.15	Yes
18	Primary product ratios	1952	1979	0.80	1	1.20	Yes
19	Primary product ratios	1980	2100	0.85	1	1.15	Yes

sets. The Oregon data have three year sets for harvest, timber production, and primary product ratios. Each parameter requires a separate MC random variable. Finally, parameters for timber product ratios and end use product ratios have a sum-to-one characteristic. For instance, timber product ratios (TPRs) represent the proportion of total harvest for a given year that is allocated to each of the timber product ratio values. While the proportions change during the time series the full set of TPR proportions will always sum to one.

3.2.2 Simulation Sampling

As in other HWP publications such as Stockmann et al. [2012] and Anderson et al. [2013], the random variables in the Monte Carlo simulations are drawn from triangular distributions. The distributions all have a peak value of 1.0 (Table 3.1) and symmetrically taper to given 90% confidence interval bounds. The values from the triangular distribution are used as proportions for adjusting parameter values.

Drawing random variables from triangular distributions is a multi-step process. The first step involves drawing random variables from a standard uniform (0, 1) distribution using a Latin Hypercube Sampling (LHS) process. The standard uniform distribution is an excellent starting place for randomly selecting points along other distributions. The random uniform points can be translated as locations along some other distribution's Cumulative Distribution Function (CDF). However, with unorganized, truly random draws of random uniform variables for more than one distribution (we are using at least 16 separate distributions, with one to three distributions per parameter) there is a strong probability that samples will by chance generally fall in the same local region for several of the distributions. In other words, a purely random sample may be inefficient. Many

Table 3.2: Translation of 90% Confidence Intervals to endpoints for triangular distributions.

90% Confidence Interval Range	Endpoints Used
0.7, 1.3	0.57, 1.43
0.8, 1.2	0.71, 1.29
0.85, 1.15	0.78, 1.22
0.95, 1.05	0.93, 1.07

iterations are needed to obtain a full suite of the possible MC values across distributions. LHS offers a method for partitioning the uniform distributions and forcing the random selections across distributions to avoid clumping. As a result, many fewer iterations are needed to achieve stable estimates from the MC simulation.

For this analysis I used the function "randomLHS" in the package "lhs" [Carnell, 2021] to conduct a random LHS sample. When we obtain random draws for a given distribution, the package divides the uniform distribution into as many partitions as there are iterations and then randomly selects a value from within that range. The partitions themselves are randomly ordered.

Once we have a matrix from the LHS (rows = number of iterations, columns = desired number of distributions) we can transform the cell values into draws from triangular distributions Wikipedia: Triangular Distribution.

$$\text{Triangular Random Variable} = \begin{cases} if 0 < U < F(c) & a + \sqrt{U(b-a)(c-a)} \\ else & b - \sqrt{(1-U)(b-a)(c-a)} \end{cases}$$

In this equation U is the random uniform variable (the LHS draw), $F(c) = (c-a)/(b-a), a$ is the minimum value for the triangle, b is the maximum value, and c is the midpoint. Note that the specifications require 90% confidence intervals and do not state what the endpoints for the triangular distributions should be. I derived the endpoints that corresponded with the desired 90% confidence intervals (Table 3.2) using an optimization algorithm.

Each random draw from a triangular distribution was used to adjust an MC parameter (Table 3.1) for all years in a single iteration. The next iteration would use the subsequent random draw. The only exception occurs when the 90% confidence intervals changed within the year set for an iteration. In those instances (i.e., for Harvest, Timber Production Ratios, and Primary Production Ratios) we simulate random draws for three random variables if there were three sets of years identified for each parameter. For each iteration we use the appropriate random triangular variable draw for the given year set (e.g., 1906 to 1945).

There is a final complication: we would like some random variables to be correlated with one another (e.g., Stockmann et al. [2012]). For instance, Harvest has diminishing confidence intervals for annual timber production over time. We have a separate random variable for each of those three sets of years. In a given iteration we can imagine that, had a practitioner developed those values, the intervals would likely have been constructed in similar ways and therefore would have been correlated (biased). Therefore, we force a correlation among the three Harvest variables. Timber Product Ratios and End Use Product Ratios can each have sets of correlated variables and I created pairs of correlated random triangular variables for all paper/wood discard disposition ratios.

I used a Pearson's correlation of 0.5 for all correlations. This was also done by Anderson et al. [2013] and Stockmann et al. [2012]. I used the Pearson's correlation to create a Pearson's correlation matrix, with the number of columns and rows corresponding to the number of random variables to correlate. For a three-variable correlation I would use the following matrix and create a Choleski decomposition from it:

$$\text{Pearson's Correlation Matrix} = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{bmatrix}$$

$$\text{Choleski Decomposition} = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0 & 0.866 & 0.289 \\ 0 & 0 & 0.816 \end{bmatrix}$$

Linear multiplication of the three variables against the Choleski decomposition alters the second and third variables such that they are correlated to the first. I create the correlated triangular random variables by first transforming an iteration's three LHS random uniform variables into values from a standard normal distribution by treating the random uniform variables as probability quantiles from a standard normal distribution. The three standard normal values are then altered and become correlated by linearly multiplying them by the Choleski decomposition. The values are transformed back to a standard uniform distribution by finding the normal cumulative distribution function values of the three normally distributed and correlated random variables. These random variables are then ready to be processed as described above into draws from a triangular distribution.

3.2.3 Sampling and the HWP program

The MC code operates by first running the original HWP program. This creates many values and arrays that the MC simulation relies upon and which we do not want to duplicate with each simulation iteration. I simplified the MC simulation code by moving several functions outside of the code and removing specific

outputs and their precursors that were no longer needed. The goal of the MC simulation was to produce estimates and confidence intervals for overall values for SWDS, PIU, EWOEC, EEC, and their combinations.

Before entering the MC loop, the program performs all LHS draws, forces correlations among specific variables, and transforms the random draws into draws from triangular distributions. Many of the parameter values are then manipulated so that they may be combined into the MC loop's creation of the eu_ratio matrix. Several of the parameters adjusted for each MC iteration affect calculations of the values for the matrix eu_ratios. Specifically, the eu_ratios value for MTC is calculated as follows:

MTC = Harvest * CCF to MTC conversion * TPR * PPR * EUR

This matrix, in the original version of the HWP program, is used to create the eu_array, which distributes all HWP inputs for a given year into PIU, discarded products (EWOEC, SWDS), and EEC. The eu_array object has a dimension for the ownership categories. Since the MC loop only focuses on total values across ownerships (the final category), the eu_array item becomes a two-dimensional matrix with EUR categories for rows and years for the columns.

Below I describe how each parameter is altered.

CCF to MgC Conversion Factors
This is a simple alteration. Within a single iteration, one random triangular variable value is multiplied across the conversion factor CCFtoMTconv values for all years.

Harvest
For Harvest we use one or more correlated random variables for every MC iteration. Each random variable is used to adjust its corresponding range of years. Subsequent iterations use a different draw of correlated random variables. The matrix of random triangular variables, with years = rows, iterations = columns, is multiplied by a vector of the harvest volume per year. For each iteration in the MC loop a different vector of these altered values is used.

Timber Product Ratios
This parameter is more complicated than Harvest because the values within a year must sum to one. It relies on three correlated random variables which are processed as described above. California and Oregon have 40 TPR categories. For each iteration, the code examines all 40 proportion values and determines which one, for a given year, is the maximum value. It finds the product of the random triangular variable value and the maximum proportion. If the resulting value is 1 then the maximum proportion is set to 1.0 and all of the other 39 values are set to 0.0. If it is < 1 then the code calculates the ratio of (1 – new maximum proportion) / (1 – old maximum proportion) and multiplies this proportion against all of the remaining proportions. All 40 altered values should sum to 1.0.

The result is that for each year there are 40 altered TPR values. When the

MTC variable is calculated for each iteration, the altered values are effectively propagated across all PPR and EUR categories per year. This MC sum-to-one approach does have some substantial drawbacks. For instance, Timber Product Ratio 2 is Softwood Sawtimber. It accounts for 77% to 97% of harvest in a given year. Multiplying the triangular distribution value by Timber Product Ratio 2 frequently results in values that are 1.0. Therefore, the alteration of this variable frequently results in the same value, and the distribution of Timber Product Ratio 2 is triangular until the distribution crosses the 1.0 boundary, at which point the remainder of the distribution is stacked at 1.0.

Primary Product Ratios
Three correlated random variables were used to alter PPR for each MC iteration, as was done for TPR. California and Oregon have 64 Primary Product Ratios (PPRs). The 64 PPRs differ from the TPR values in that they sum by year to 40, not 1. Sets of PPR ratios sum to 1; most sets include only one ratio. The PPRs were altered in the same fashion as TPR values with one difference. The code found, for each PPR set with more than one value, the maximum proportion and then adjusted that proportion with the random triangular parameter value. It adjusted the remainder of the values in the PPR set as described for TPR. If there was only one value in the set, it was always equal to 1.0 and the code never adjusted it. These 1.0 values allowed the TPR value to be carried through PPR. Similarly, if the maximum ratio in a set was already 1.0, no alteration occurred.

These values are propagated across all EUR categories per year so that they may be multiplied against other values in the eu_ratio matrix to create the altered MTC values.

End Use Product Ratios
End Use Product ratios were altered as described for PPRs. One difference is that the EUR alterations only used one random triangular variable value for all years in an iteration (i.e., not three correlated variables) because the triangular distribution confidence intervals were held constant across years (Table 3.1). California and Oregon EUR values for a given year summed to 64. The code therefore examined sets of EUR values for each PPR category. The values were combined with other values in the eu_ratio matrix to assist in altering the MTC variable.

Product Half Lives
This was a straightforward alteration. California and Oregon have 224 half-life values, one per EUR. The same random triangular random variable value was multiplied by all 224 values per iteration. Each iteration of the MC loop used a different vector of those 224 values in the decay function that is used to determine PIU over time.

Decay and Half-Life Limits for Landfills, Dumps, and Recovered Pools
Description of these parameters is combined because they relied on the same process. Each one has two correlated random triangular variables, for paper and wood. A function inputs the original 224 ratios for each, where most of the

224 ratios are for wood and a few are for paper. The function also obtains the two vectors of correlated random variables. Each iteration has a single random variable value multiplied by the paper ratio and another for the wood ratio. The altered vectors are then combined into a matrix. When the MC loop runs, each iteration obtains a differently altered vector of 224 values to enter into the decay function.

Discarded Disposition Ratios for Wood and Paper
These ratios control what fraction of waste ends up as burned, burned with energy capture, recycled, composted, placed in a landfill, or sent to a dump (the six fates). The values can differ by year. I wrote a function that operated on the ratios for wood and paper separately. The function performed the sum-to-one approach on the six ratios for either wood or paper. The original HWP analysis process relied on a function for each of these six fates that was time consuming. This function was removed for the MC simulation. Prior to the MC loop I created a matrix that contained altered values for wood and paper across all years and for each iteration for the six fates. The MC loop calls, for each iteration, a different subset of the matrix to use for calculations involving the six fates.

3.2.4 Monte Carlo results and development

It appears that the full version of the MC simulation reaches convergence with 2000 iterations (Figure 3.1). Convergence was assessed visually by examining, over the iterations, the estimated values for the mean, standard error (SE), and the lower and upper empirical 90% confidence interval limits. All four values appear to have reached their stable values by 1500 iterations, as verified with iterations 1501 - 2000. (Note: the calculated SE values are not used beyond this figure.)

The Monte Carlo outputs are summarized in Figures 3.2 and 3.3. In Figure 3.2 we see the mean Tg C value for emissions (Emitted with Energy Capture, Emitted Without Energy Capture) and pools (Products in Use, SWDS) along with the empirical 90% confidence intervals. Figure 3.3 combines Products in Use and SWDS into a single carbon pool. The four carbon fates in Figure 3.2 were obtained by calculating, for each iteration, the cumulative sum of carbon for each fate across all years. Each fate has 2000 MC iterations for each year. In Figure 3.3, it is difficult to tell, but the confidence interval range between 1906 and 1920 are between -37% and + 42% of the mean. In 2017 the range is between -17% and +17% of the mean.

The correlation procedure described above appeared to work. Table 3.3 contains the estimated correlation coefficients for the three triangular distribution random variables used for Primary Products Ratios. The target correlation coefficient was 0.5.

Histograms of the three vectors of Primary Products Ratios triangular random

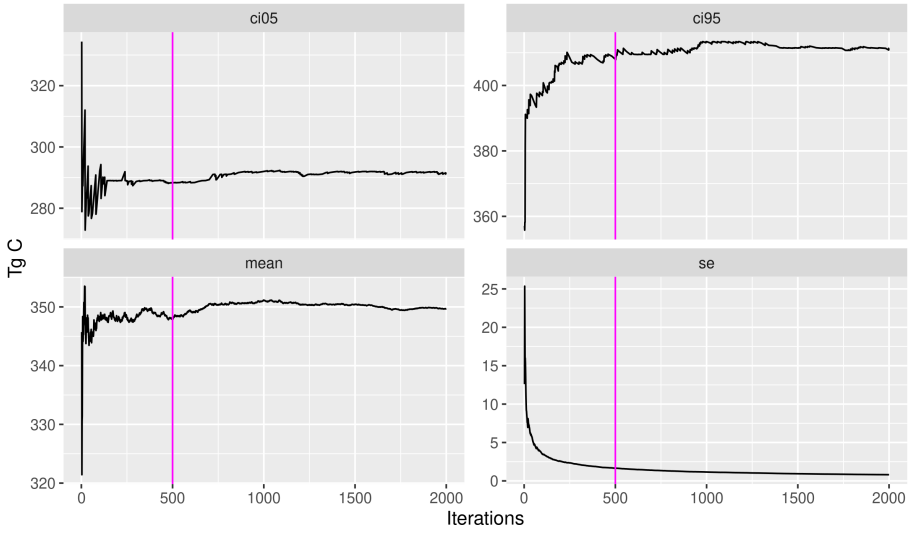


Figure 3.1: Monte Carlo convergence assessment for the upper and lower bounds of the empirical 90% confidence interval, the mean, and the standard error for the full model.

Table 3.3: Correlation matrix for Primary Products Ratios random variables, 2000 draws.

	Variable 1	Variable 2	Variable 3
Variable 1	1.000	0.505	0.508
Variable 2	0.505	1.000	0.521
Variable 3	0.508	0.521	1.000

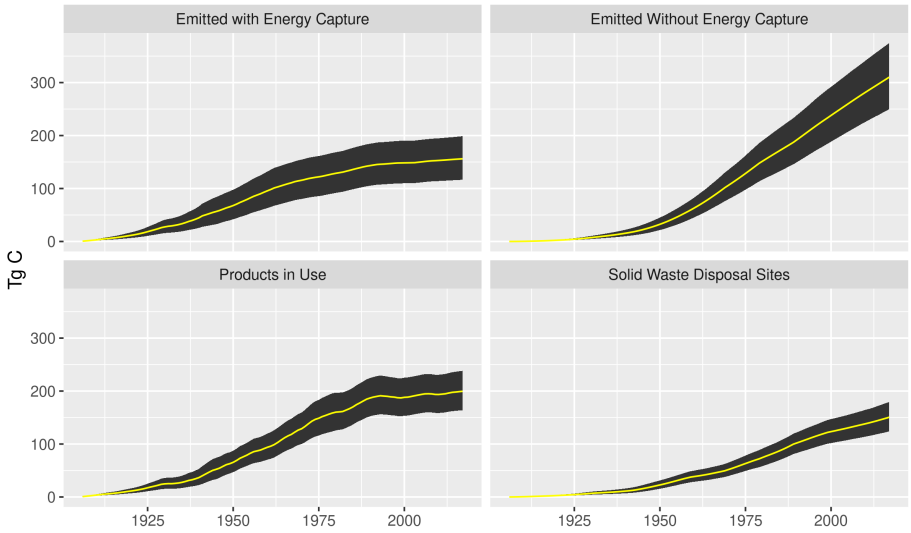


Figure 3.2: Monte Carlo simulation mean (yellow) and 90% confidence interval ranges for four HWP carbon fates across years.

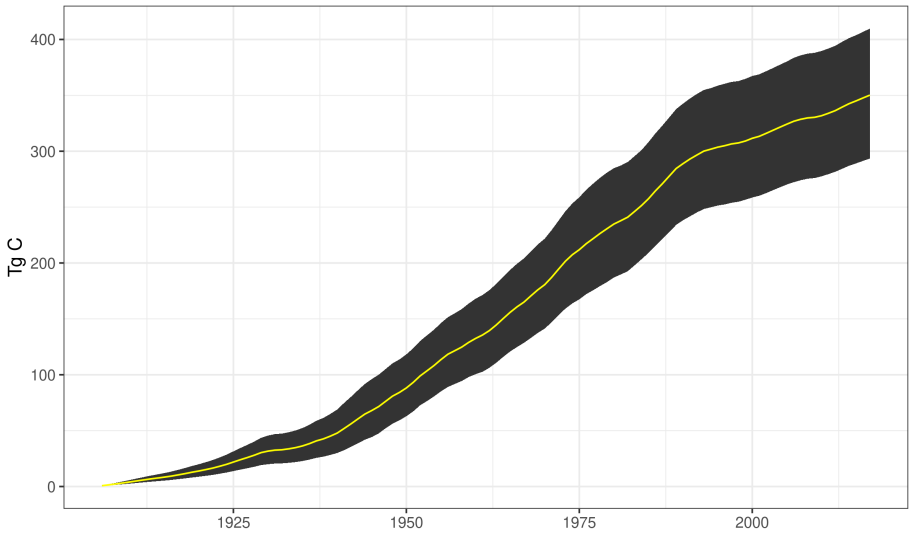


Figure 3.3: Monte Carlo simulation mean (yellow) and 90% confidence interval ranges for the total Tg C value for SWDS and Products in Use.

variables are presented in Figure 3.4. The values are correlated (Table 3.3) and the histogram distributions correspond to 90% confidence intervals and endpoints (see Table 3.2 for triangular distribution endpoints for 0.7-1.3, 0.8-1.2, and 0.85-1.15 90% confidence intervals). Although Variables 2 and 3 do not evidence as tight a triangular distribution as Variable 1, their distributions appear approximately triangular and do not appear skewed.

Figures 3.5 and 3.6 examine triangular distribution changes on sum-to-one ratio sets. In Figure 3.5, PPR.21 is the largest proportion in its sum-to-one group of proportions and was directly adjusted by the random triangular variable draws. PPR.17 was adjusted to ensure that it and the other variables in the group still summed to one. Note that although PPR.21 was the only one directly altered, PPR.17 evidences a triangular distribution as well. The percentage range of PPR.17 values was broader, however, than the percentage range of values for PPR.21.

Figure 3.6 shows the consequences of altering a sum-to-one ratio beyond 1.0, in this case using TPR.2 during the year 2002. The code truncates the value at 1.0, forcing TPR.1 to become zero. The two ratio sets appear to otherwise conform to triangular distributions.

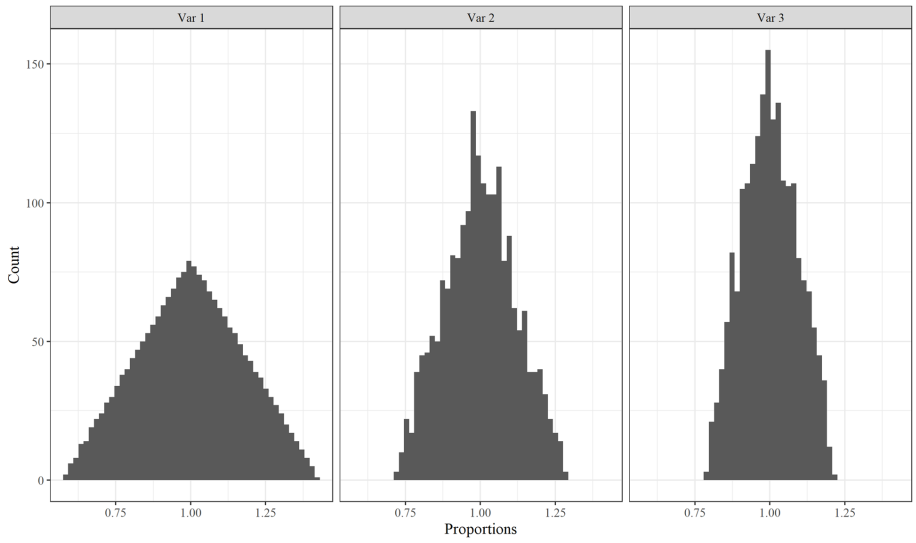


Figure 3.4: Histograms of correlated triangular distribution random variables for Primary Product Ratios.

The MC simulation was developed in pieces. I first altered the values for the non-sum-to-one variables: CCF to MgC conversion factors, annual harvest amounts, product half-lives, landfill decay limits, and the half-lives of decaying paper and wood in landfill, dumps, and recovered products. These alterations should

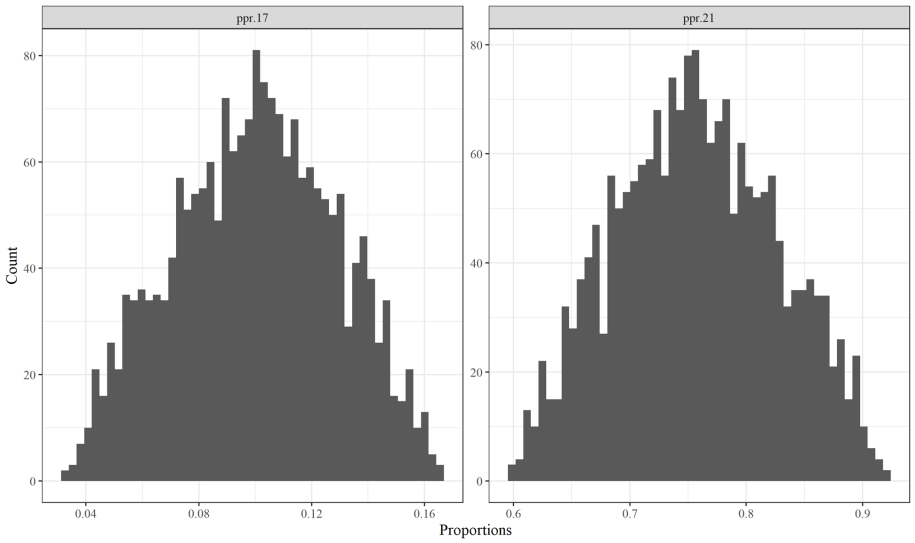


Figure 3.5: Histograms of sum-to-one Primary Product Ratios variables PPR.17 and PPR.21 that do not overlap with 0.0 or 1.0. Both variables are for the year 1994.

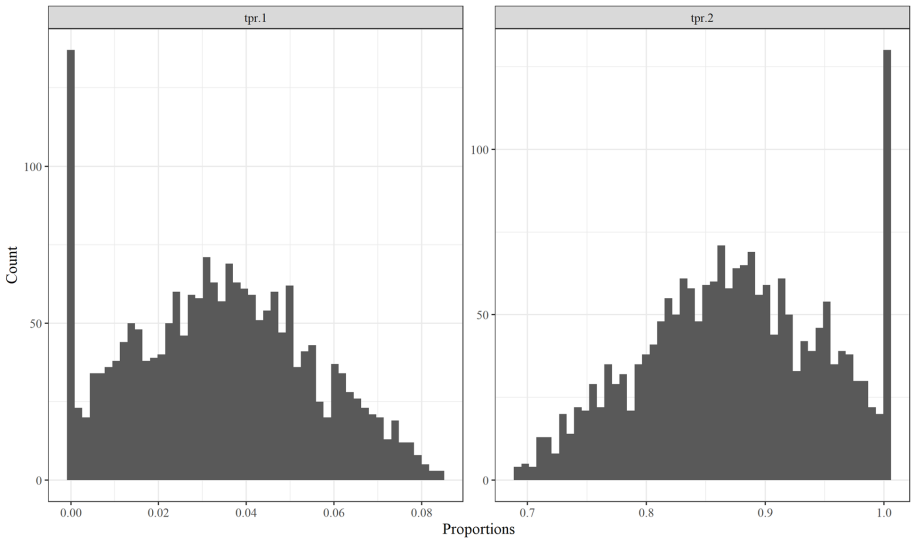


Figure 3.6: Histograms of sum-to-one Timber Product Ratios variables TPR.1 and TPR2; the values for TPR.2 were truncated at 1.0. Both variables are for the year 2002.

affect all categories, since harvest amounts and the conversion factors control all carbon input into the model, product half-lives affect PIU residence and SWDS input, and the SWDS decay limits and half lives affect residency in SWDS.

I next introduced Timber Production Ratios as the first sum-to-one MC alterations, then Primary Production Ratios, then End Use Ratios. Finally, I added the Discarded Disposition Ratios (DDR). These additions did not affect MC results for all carbon fates (Emitted with Energy Capture, Emitted without Energy Capture, Solid Waste Disposal Sites, and Products in Use) equally.

Figure 3.7 provides results for Emitted with Energy Capture. We can see that the MC 90% confidence intervals increase between the first (No Sum-to-One MC Variables) and second (TPR) versions of the model. When I introduced variation into the Timber Production Ratios and then Primary Product Ratios the ratios for fuel wood were altered. Those fuel wood ratios are not altered by changes to EUR, as EUR in this case effectively serves as a "pass-through" category for fuel wood where the ratios remain 1.0 and are not altered by the simulation.

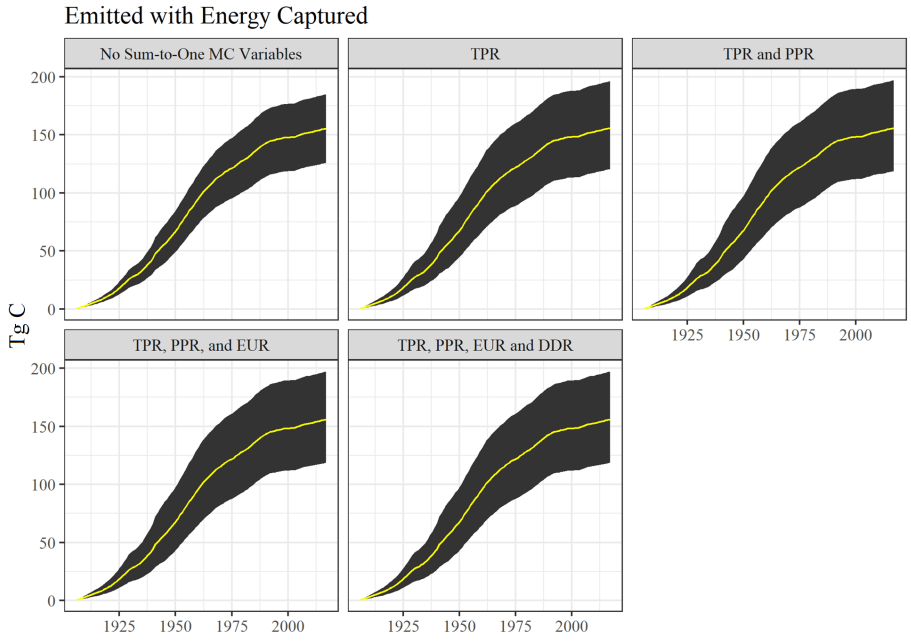


Figure 3.7: Monte Carlo simulation outputs for Emitted with Energy Captured for different Monte Carlo construction stages. See text for acronym definitions.

Figure 3.8, Emitted without Energy Capture, is interesting because graphically there does not appear to be a large change between MC construction iterations. However, the confidence interval ranges do increase across all versions. Each

version differs by between 1 and 4% of the previous version. Therefore, most of the gain in variability was achieved by altering the first set of variables. All other variable alterations did affect the amount of carbon emitted without energy capture, be it affecting the discard dispositions or the End Use Product Ratios (which would in turn be affected by EUR product half-lives and all other variables affecting the next steps of disposal and decay). The effect was not large, however.

Figure 3.9, depicting the MC results for carbon stored in Solid Waste Disposal Sites, really does not evidence an increase in the MC confidence interval until the final iteration when Discarded Disposition Ratios are changed. Changing the DDR affects where disposed carbon winds up. The initial version of the model is already altering carbon entering the system and product half-lives. These are likely the two major drivers of variation for SWDS. Altering the ratios of which products carbon goes to affects things relatively slightly.

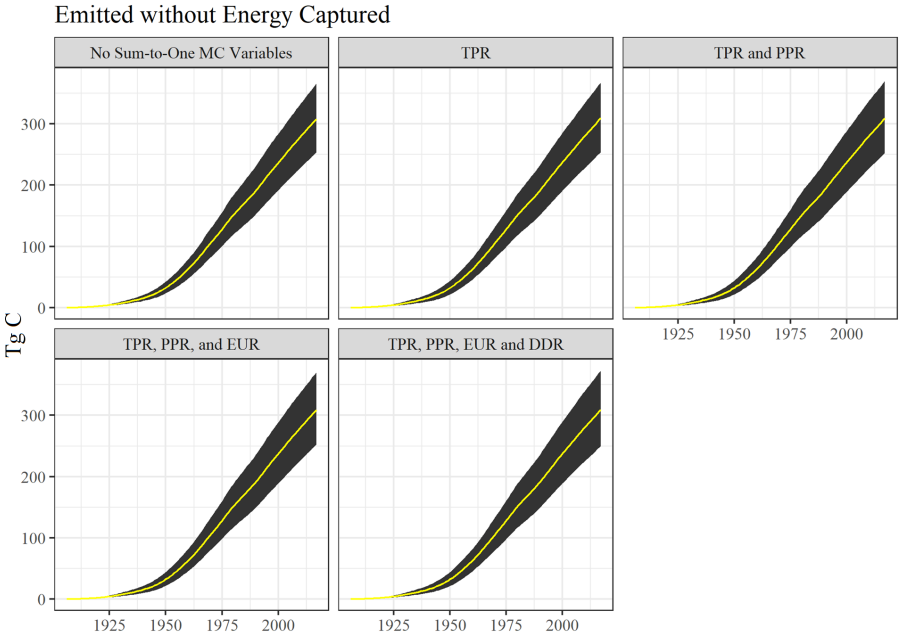


Figure 3.8: Monte Carlo simulation outputs for Emitted without Energy Captured for different Monte Carlo construction stages. See text for acronym definitions.

Products in Use (Figure 3.10) outcomes for the different MC model versions has similar results as EWOEC. Subsequent additions of sum-to-one variables do increase the confidence intervals, by about 1 to 2%. Therefore, adding the sum-to-one variables did not affect the model much. As we might expect, including

the DDR variable changes did not affect PIU confidence intervals at all since those changes only interact with discarded carbon.

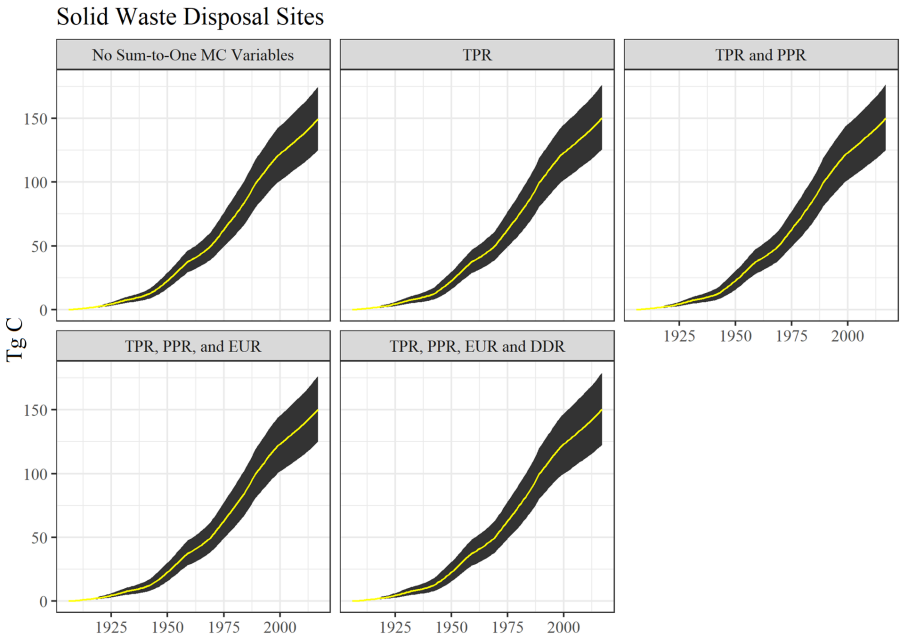


Figure 3.9: Monte Carlo simulation outputs for Solid Waste Disposal Sites for different Monte Carlo construction stages. See text for acronym definitions.

The simulation also did not use different random triangular variable values within a parameter for a given iteration. For instance, the California and Oregon have 224 product decay ratios. In a single iteration they were all multiplied by the same draw value from a triangular distribution. Conceivably, there could be a separate random draw for every product decay ratio, and every other sort of parameter subset. This could result in creating a LHS procedure with hundreds of separate random variables instead of the number used here.

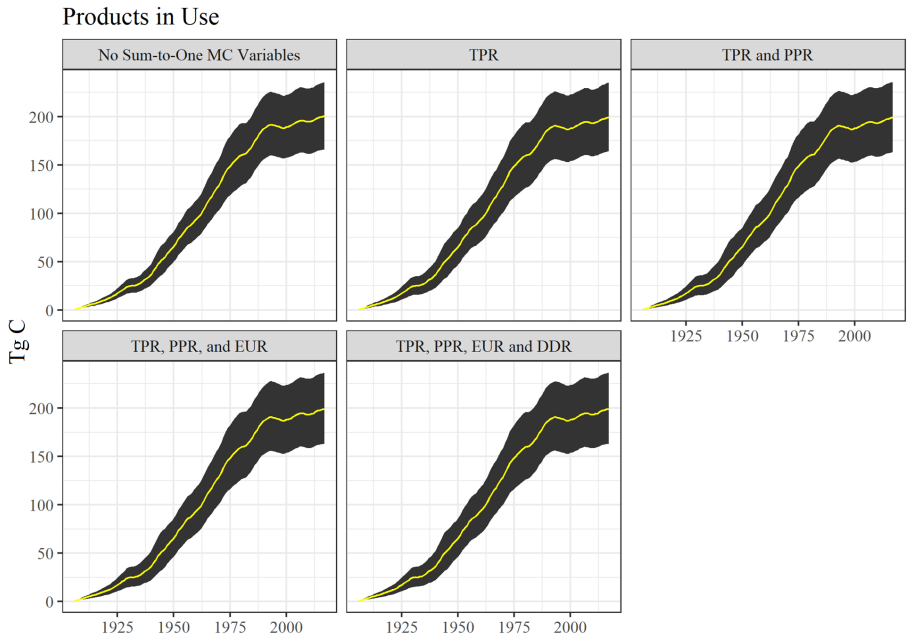


Figure 3.10: Monte Carlo simulation outputs for Products in Use for different Monte Carlo construction stages. See text for acronym definitions.

Chapter 4

Parts

You can add parts to organize one or more book chapters together. Parts can be inserted at the top of an .Rmd file, before the first-level chapter heading in that same file.

Add a numbered part: `# (PART) Act one {-}` (followed by `# A chapter`)

Add an unnumbered part: `# (PART\*) Act one {-}` (followed by `# A chapter`)

Add an appendix as a special kind of un-numbered part: `# (APPENDIX) Other stuff {-}` (followed by `# A chapter`). Chapters in an appendix are prepended with letters instead of numbers.

Chapter 5

Footnotes and citations

5.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one [1].

5.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [Xie, 2022] (check out the last code chunk in index.Rmd to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [Xie, 2015] (this citation was added manually in an external file book.bib). Note that the `.bib` files need to be listed in the index.Rmd with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: https://rstudio.github.io/visual-markdown-editing/#/citations

[1] This is a footnote.

Chapter 6

Blocks

6.1 Equations

Here is an equation.

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{6.1}$$

You may refer to using `\@ref(eq:binom)`, like see Equation (6.1).

6.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref(thm:tri)`, for example, check out this smart theorem 6.1.

Theorem 6.1. *For a right triangle, if c denotes the* length *of the hypotenuse and a and b denote the lengths of the* ***other*** *two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html.

6.3 Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html

Chapter 7

Sharing your book

7.1 Publishing

HTML books can be published online, see: https://bookdown.org/yihui/bookdown/publishing.html

7.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

7.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book- all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

https://pkgs.rstudio.com/bookdown/reference/gitbook.html

Or use:

```
?bookdown::gitbook
```

Bibliography

Nathaniel Anderson, Jesse Young, Keith Stockmann, Kenneth Skog, Sean Healey, Daniel Loeffler, J Greg Jones, and James Morrison. Regional and forest-level estimates of carbon stored in harvested wood products from the united states forest service northern region, 1906-2010. *Gen. Tech. Rep. RMRS-GTR-311. Fort Collins, CO: US Department of Agriculture, Forest Service, Rocky Mountain Research Station. 114 p.*, 311, 2013.

Rob Carnell. *lhs: Latin Hypercube Samples*, 2021. URL https://github.com/bertcarnell/lhs. R package version 1.1.3.

Dan Loeffler, Nathaniel Anderson, Keith Stockmann, Ken Skog, Sean Healey, J Greg Jones, James Morrison, and Jesse Young. Estimates of carbon stored in harvested wood products from united states forest service southern region, 1911-2012. *Unpublished report. Missoula, MT: US Department of Agriculture, Forest Service, Rocky Mountain Research Station, Forestry Sciences Laboratory. 27 p.*, 2014.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021. URL https://www.R-project.org/.

Keith D Stockmann, Nathaniel M Anderson, Kenneth E Skog, Sean P Healey, Dan R Loeffler, Greg Jones, and James F Morrison. Estimates of carbon stored in harvested wood products from the united states forest service northern region, 1906-2010. *Carbon Balance and Management*, 7(1):1–16, 2012.

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL http://yihui.org/knitr/. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2022. URL https://CRAN.R-project.org/package=bookdown. R package version 0.25.