Replication in R of SimVoi's Monte Carlo Simulations of Guyana's REDD+ Uncertainty Esimates

2024-03-24

Contents

Objective . . . 1
SimVoi syntax . . . 1
SimVoi replication . . . 3
Replication results . . . 8
Distribution analysis . . . 9
Recommendations . . . 13
References . . . 14
Annex I: SimVoi replication in base-R . . . 15
Annex II: Review of Monte Carlo methods in REDD+ . . . 16
Annex III: Runtime snapshot . . . 17

Objective

Using an R-based approach, this analysis replicates the Monte Carlo simulations originally performed with SimVoi in Excel. It details the code used in the analysis, compares simulation results between R and Excel, and proposes next steps for enhancement based on statistical tests. Specifically, we identified updates needed to address empirical distributions of the input data. Such updates are recommended to both strengthen compliance with REDD+ requirements and VVB audits, and to provide an effective basis for reducing uncertainty and increasing revenue from emissions credits.

SimVoi syntax

SimVoi provides seventeen random number generator functions[1] that are operated with the following syntax:

- `RandBeta(alpha,beta,,[MinValue],[MaxValue])`
- `RandBinomial(trials,probability_s)`
- `RandBiVarNormal(mean1,stdev1,mean2,stdev2,correl12)`
- `RandCumulative(value_cumulative_table)`
- `RandDiscrete(value_discrete_table)`
- `RandExponential(lambda)`
- `RandInteger(bottom,top)`
- `RandLogNormal(Mean,StDev)`
- `RandNormal(mean,standard_dev)`
- `RandPoisson(mean)`
- `RandSample(population)`
- `RandTriangular(minimum,most_likely,maximum)`
- `RandTriBeta(minimum,most_likely,maximum,[shape])`
- `RandTruncBiVarNormal(mean1,stdev1,mean2,stdev2,correl12, [min1],[max1],[min2],[max2])`
- `RandTruncLogNormal(Mean,StDev,[MinValue],[MaxValue])`
- `RandTruncNormal(Mean,StDev,[MinValue],[MaxValue])`

[1] SimVoi-313-Guide.pdf

- `RandUniform(minimum,maximum)`

In this workflow, we attempt to replicate the following SimVoi function as identified in Guyana's emissions workbook:

`=[1]!randtruncnormal(CarbonStocks.B2,CarbonStocks.B3,0)`

According to the documentation, *RandTruncNormal()* returns a random value from a truncated normal distribution, modeling an uncertain quantity with a bell-shaped density while excluding extreme tail values. If no simulation count is provided, the function repeatedly samples until a value falls between the specified minimum and maximum—or until it reaches 10,000 attempts. In the above example, only a minimum value of 0 is provided, so the default iteration limit is used.

Import data

```r
workbook  = "./data/art/GuyanaARTWorkbookMC-thru2022-April2024_values_V2.xlsx"
CarbonStocks = readxl::read_excel(workbook, "CarbonStocks") |>
  janitor::clean_names() |> mutate(across(where(is.numeric), ~ round(.x, 1)))
DegradationEF = readxl::read_excel(workbook,"Degradation EFs") |> janitor::clean_names()|>
  mutate(across(where(is.numeric), ~ round(.x, 1)))
ActivityData = readxl::read_excel(workbook, "Activity Data") |> janitor::clean_names() |>
  mutate(across(where(is.numeric), ~ round(.x, 1)))
flextable(head(CarbonStocks[, 1:8]))|>fontsize(size=8,part="all")
flextable(DegradationEF[, 1:8])|>fontsize(size=8,part="all")
flextable(ActivityData[, 1:8])|>fontsize(size=8,part="all")
```

Table 1: Input values from CarbonStocks tabsheet

Statistic	AG Tree	BG Tree	Saplings	Standing Dead	Lying Dead	Litter	Sum w/o Soil	Soil
mean	205.8	48.3	3.7	2.6	8.6	3.3	272.3	58.7
std_dev	60.4	14.3	2.0	4.0	8.1	1.3	90.0	61.5
minimum	91.6	21.2	0.5	0.0	0.0	1.2	114.4	10.1
maximum	353.7	83.1	18.8	13.7	42.3	8.7	520.3	502.4
90%_CI	9.2	2.2	0.3	0.6	1.2	0.2	N/A	11.0
CI_%_of_mean	0.0	0.0	0.1	0.2	0.1	N/A	N/A	0.2

Table 2: Input values from Degradation EFs tabsheet

Statistic	LDF	Wood Density	LIF	For. Infrastr.	Mining	Mining Infrastr.	Infrast.
Factor_tC	1.05	0.4	46.87	NA	NA	NA	NA
StdDev_tC	0.68	0.03	8.08	NA	NA	NA	NA
CI90_tC	0.08	0	1.6	NA	NA	NA	NA
Factor_tCO2	3.85	1.47	171.84	NA	NA	NA	NA
StdDev_tCO2	2.4900000000000002	0.11	29.63	NA	NA	NA	NA
CI90_tCO2	0.28999999999999998	0.01	5.87	NA	NA	NA	NA
EF_tCO2_AD	NA	NA	NA	8.1	8.1	8.1	8.1
MAD_tCO2	NA	NA	NA	8.1	8.1	8.1	8.1

Table 3: Absolute input values from Activity Data tabsheet

drivers	units	x2011	x2012	x2013	x2014	x2015	x2016	x2017	x2018	x2019	x2020	x2021	x2022	x2023
Forestry infrastructure	ha	186	240	330	204	313	313	227	356	226	195	228	155.6	339
Agriculture	ha	41	440	424	817	379	379	477	512	246	489	216	281.6	475
Mining (medium and large scale)	ha	7340	13664	11518	10434	6782	6782	7442	7624	5821	6,452	6,825	5,264.3	5,853
Mining infrastructure	ha													
Infrastructure	ha	118	127	342	141	217	217	195	67	52	102	117	110.6	541
Settlements	ha	-	-	23	71	8	8	7	7	22	60	105	169.4	201

SimVoi replication

CarbonStocks data

```
A_MEAN = CarbonStocks$ag_tree_t_c_ha[1]
A_SD   = CarbonStocks$ag_tree_t_c_ha[2]
B_MEAN = CarbonStocks$bg_tree_t_c_ha[1]
B_SD   = CarbonStocks$bg_tree_t_c_ha[2]
C_MEAN = CarbonStocks$saplings_t_c_ha[1]
C_SD   = CarbonStocks$saplings_t_c_ha[2]
D_MEAN = CarbonStocks$standing_dead_wood_t_c_ha[1]
D_SD   = CarbonStocks$standing_dead_wood_t_c_ha[2]
E_MEAN = CarbonStocks$lying_dead_wood_t_c_ha[1]
E_SD   = CarbonStocks$lying_dead_wood_t_c_ha[2]
F_MEAN = CarbonStocks$litter_t_c_ha[1]
F_SD   = CarbonStocks$litter_t_c_ha[2]
G_MEAN = CarbonStocks$sum_pools_w_o_soil[1]
G_SD   = CarbonStocks$sum_pools_w_o_soil[2]
H_MEAN = CarbonStocks$soil_t_c_ha[1]
H_SD   = CarbonStocks$soil_t_c_ha[2]

# 100 simulations sampling 1 observation each iteration.
# NOTE: #no. of simulations reduced to match estimates (see "Observations")
A_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,91.6,353.7,A_MEAN, A_SD)
B_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,21.2,83.1,B_MEAN, B_SD)
C_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,0.5,18.8,C_MEAN, C_SD)
D_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,0.0,13.7,D_MEAN, D_SD)
E_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,0.0,42.3,E_MEAN, E_SD)
F_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,1.2,8.7,F_MEAN,F_SD)
G_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,114.4,520.3,G_MEAN, G_SD)
H_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,10.1,502.4,H_MEAN, H_SD)

# --- Simulation Estimates ---
AG_tree_tC_ha           = mean(A_rtruncnormal_100)
AG_tree_tCO2_ha         = mean(A_rtruncnormal_100)*(44/12)
BG_tree_tC_ha           = mean(B_rtruncnormal_100)
BG_tree_tCO2_ha         = mean(B_rtruncnormal_100)*(44/12)
Saplings_tC_ha          = mean(C_rtruncnormal_100)
Saplings_tCO2_ha        = mean(C_rtruncnormal_100)*(44/12)
StandDead_tC_ha         = mean(D_rtruncnormal_100)
StandDead_tCO2_ha       = mean(D_rtruncnormal_100)*(44/12)
```

```
LyingDead_tC_ha          = mean(E_rtruncnormal_100)
LyingDead_tCO2_ha        = mean(E_rtruncnormal_100)*(44/12)
Litter_tC_ha             = mean(F_rtruncnormal_100)
Litter_tCO2_ha           = mean(F_rtruncnormal_100)*(44/12)
Sum_wo_Soil_tC_ha        = mean(G_rtruncnormal_100)
Sum_wo_Soil_tCO2_ha      = mean(G_rtruncnormal_100)*(44/12)
Soil_tC_ha               = mean(H_rtruncnormal_100)
Soil_tCO2_ha             = mean(H_rtruncnormal_100)*(44/12)

CarbonStocks_MC_R_df <- data.frame(
  Units                    = c("tC/ha", "tCO2/ha"),
  `AG Tree`                = c(AG_tree_tC_ha,  AG_tree_tCO2_ha),
  `BG Tree`                = c(BG_tree_tC_ha,  BG_tree_tCO2_ha),
  `Saplings`               = c(Saplings_tC_ha, Saplings_tCO2_ha),
  `Standing Dead`          = c(StandDead_tC_ha, StandDead_tCO2_ha),
  `Lying Dead`             = c(LyingDead_tC_ha, LyingDead_tCO2_ha),
  `Litter`                 = c(Litter_tC_ha, Litter_tCO2_ha),
  `Sum w/o Soil`           = c(Sum_wo_Soil_tC_ha, Sum_wo_Soil_tCO2_ha),
  `Soil`                   = c(Soil_tC_ha, Soil_tCO2_ha)
  )
```

Degradation data

```
A_MEAN = DegradationEF$ldf[4]
A_SD   = DegradationEF$ldf[5]
B_MEAN = DegradationEF$wood_density[4]
B_SD   = DegradationEF$wood_density[5]
C_MEAN = DegradationEF$lif[4]
C_SD   = DegradationEF$lif[5]
D_MEAN = DegradationEF$forestry_infrastructure[7]
D_SD   = DegradationEF$forestry_infrastructure[8]
E_MEAN = DegradationEF$mining[7]
E_SD   = DegradationEF$mining[8]
F_MEAN = DegradationEF$mining_infrastructure[7]
F_SD   = DegradationEF$mining_infrastructure[8]
G_MEAN = DegradationEF$infrastructure[7]
G_SD   = DegradationEF$infrastructure[8]

# 100 simulations sampling 1 observation each iteration.
A_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,0,Inf,A_MEAN, A_SD)
B_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,0,Inf,B_MEAN, B_SD)
C_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,0,Inf,C_MEAN, C_SD)
D_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,0,Inf,D_MEAN, D_SD)
E_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,0,Inf,E_MEAN, E_SD)
F_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,0,Inf,F_MEAN,F_SD)
G_rtruncnormal_100 = truncnorm::rtruncnorm(n=100,0,Inf,G_MEAN, G_SD)

LDF_EF_tCO2_m2           = mean(A_rtruncnormal_100)
WD_EF_tCO2_m2            = mean(B_rtruncnormal_100)
LIF_EF_tCO2_km           = mean(C_rtruncnormal_100)
ForInfr_EF_tCO2_ha       = mean(D_rtruncnormal_100)
Mining_EF_tCO2_ha        = mean(E_rtruncnormal_100)
MiningInfr_EF_tCO2_ha    = mean(F_rtruncnormal_100)
Infrastructure_EF_tCO2_ha= mean(G_rtruncnormal_100)
```

```
df_logging <- data.frame(
  component = c("LDF", "Wood Density of Harvest", "LIF (Skid Trails)"),
  unit      = c("per m3", "per m3", "per km"),
  tco2      = c(LDF_EF_tCO2_m2, WD_EF_tCO2_m2, LIF_EF_tCO2_km)
  )

df_degrading <- data.frame(degrading_activity = c(
    "Forestry infrastructure", "Mining (medium & large scale)",
    "Mining infrastructure", "Infrastructure"),
  ef_tco2_ha = c(ForInfr_EF_tCO2_ha, Mining_EF_tCO2_ha,
    MiningInfr_EF_tCO2_ha, Infrastructure_EF_tCO2_ha)
  )

max_rows <- max(nrow(df_logging), nrow(df_degrading))
logging_nas         = rep(NA, max_rows - nrow(df_logging))
degrading_nas       = rep(NA, max_rows - nrow(df_degrading))
Degradation_MC_R_df = data.frame(
  Component         = c(df_logging$component, logging_nas),
  Unit              = c(df_logging$unit, logging_nas),
  tCO2              = c(df_logging$tco2, logging_nas),
  Degrading_Activity= c(df_degrading$degrading_activity, degrading_nas),
  EF_tCO2_ha        = c(df_degrading$ef_tco2_ha, degrading_nas)
  )
```

Activity data

Please note that for purpose of saving space, columns and cells C1-M8 were implemented with `echo-F` settings, and can be located in the R-markdown.rmd file used to derive this PDF.

```
A1_MEAN = ActivityData$x2011[1]
A1_SD   = ActivityData$x2011[17]
A2_MEAN = ActivityData$x2011[2]
A2_SD   = ActivityData$x2011[18]
A3_MEAN = ActivityData$x2011[3]
A3_SD   = ActivityData$x2011[19]
A4_MEAN = ActivityData$x2011[4]
A4_SD   = ActivityData$x2011[20]
A5_MEAN = ActivityData$x2011[5]
A5_SD   = ActivityData$x2011[21]
A6_MEAN = ActivityData$x2011[6]
A6_SD   = ActivityData$x2011[22]
A7_MEAN = ActivityData$x2011[7]
A7_SD   = ActivityData$x2011[23]
A8_MEAN = ActivityData$x2011[8]
A8_SD   = ActivityData$x2011[24]

B1_MEAN = ActivityData$x2012[1]
B1_SD   = ActivityData$x2012[17]
B2_MEAN = ActivityData$x2012[2]
B2_SD   = ActivityData$x2012[18]
B3_MEAN = ActivityData$x2012[3]
B3_SD   = ActivityData$x2012[19]
B4_MEAN = ActivityData$x2012[4]
B4_SD   = ActivityData$x2012[20]
B5_MEAN = ActivityData$x2012[5]
```

```
B5_SD   = ActivityData$x2012[21]
B6_MEAN = ActivityData$x2012[6]
B6_SD   = ActivityData$x2012[22]
B7_MEAN = ActivityData$x2012[7]
B7_SD   = ActivityData$x2012[23]
B8_MEAN = ActivityData$x2012[8]
B8_SD   = ActivityData$x2012[24]

A1_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, A1_MEAN, A1_SD)
A2_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, A2_MEAN, A2_SD)
A3_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, A3_MEAN, A3_SD)
A4_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, A4_MEAN, A4_SD)
A5_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, A5_MEAN, A5_SD)
A6_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, A6_MEAN, A6_SD)
A7_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, A7_MEAN, A7_SD)
A8_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, A8_MEAN, A8_SD)

B1_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, B1_MEAN, B1_SD)
B2_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, B2_MEAN, B2_SD)
B3_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, B3_MEAN, B3_SD)
B4_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, B4_MEAN, B4_SD)
B5_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, B5_MEAN, B5_SD)
B6_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, B6_MEAN, B6_SD)
B7_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, B7_MEAN, B7_SD)
B8_rtruncnormal_100 = truncnorm::rtruncnorm(n=100, 0, Inf, B8_MEAN, B8_SD)

# --- Simulation Estimates ---
A1 = mean(A1_rtruncnormal_100)
A2 = mean(A2_rtruncnormal_100)
A3 = mean(A3_rtruncnormal_100)
A4 = mean(A4_rtruncnormal_100)
A5 = mean(A5_rtruncnormal_100)
A6 = mean(A6_rtruncnormal_100)
A7 = mean(A7_rtruncnormal_100)
A8 = mean(A8_rtruncnormal_100)

B1 = mean(B1_rtruncnormal_100)
B2 = mean(B2_rtruncnormal_100)
B3 = mean(B3_rtruncnormal_100)
B4 = mean(B4_rtruncnormal_100)
B5 = mean(B5_rtruncnormal_100)
B6 = mean(B6_rtruncnormal_100)
B7 = mean(B7_rtruncnormal_100)
B8 = mean(B8_rtruncnormal_100)
```

Below, we presented organize our results into a table and compare with absolute input values (CarbonStocks), and Monte Carlo estimates generated with SimVoi (CarbonStocks (MC)). For external comparisons, we have also saved these results in a new excel tab called “CarbonStocks (MC-R)”.

```
CarbonStocks_MC_R = flextable(head(CarbonStocks_MC_R_df[, 1:8])) |>
  fontsize(size = 8, part = "all")
CarbonStocks_MC_R

Degradation_MC_R = flextable(head(Degradation_MC_R_df[, 1:8])) |>
  fontsize(size = 8, part = "all")
```

```
Degradation_MC_R
```

Table 4: Results of Monte Carlo simulations of CarbonStocks tabsheet using R

```
CarbonStocks_MC_R = flextable(CarbonStocks_MC_R_df) |>
  width(width = 1) |> fit_to_width(max_width = 6) |>
  colformat_double(big.mark = ",", digits = 1, na_str = "N/A")
CarbonStocks_MC_R
```

Units	AG.Tree	BG.Tree	Saplings	Standing.Dead	Lying.Dead	Litter	Sum.w.o.Soil	Soil
tC/ha	203.6	48.3	4.1	4.5	11.3	3.4	281.1	86.1
tCO2/ha	746.5	177.2	15.1	16.7	41.5	12.4	1,030.8	315.7

Table 5: Results of Monte Carlo simulations of CarbonStocks tabsheet using SimVoi

Units	AG Tree	BG Tree	Saplings	Standing Dead	Lying Dead	Litter	Sum w/o Soil	Soil
tC/ha	181.1	65.0	3.5	7.3	17.1	3.7	277.7	60.6
tCO2/ha	664.2	238.2	12.8	26.9	62.6	13.7	1,018.4	222.3

Table 6: Results of Monte Carlo simulations of DegradationEFs tabsheet using R

Component	Unit	tCO2	Degrading_Activity	EF_tCO2_ha
LDF	per m3	4.5	Forestry infrastructure	10.9
Wood Density of Harvest	per m3	1.5	Mining (medium & large scale)	10.3
LIF (Skid Trails)	per km	175.0	Mining infrastructure	10.0
		N/A	Infrastructure	10.9

Table 7: Results of Monte Carlo simulations of Degradation EFs tabsheet using SimVoi

Component	Unit	t CO2
LDF	per m^3	4.54
Wood Density of timber harvested	per m^3	1.39
LIF (Skid Trails)	per km	185.88

Degrading Activity	EF (tCO2/ha)
Forestry infrastructure	7.2
Mining (med & large scale)	7.2
Mining infrastructure	7.2
Infrastructure	7.2

Table 8: Results of Monte Carlo simulations of Activity Data tabsheet using R

Drivers	units	X2011	X2012	X2013	X2014	X2015
Forestry infrastructure	ha	188	237	331	205	317
Agriculture	ha	41	443	417	824	376
Mining (med & large scale)	ha	7,508	13,624	11,594	10,453	6,874
Mining infrastructure	ha	N/A	N/A	N/A	N/A	N/A
Infrastructure	ha	120	127	342	141	215
Settlements	ha	N/A	N/A	23	71	8

Drivers	units	X2011	X2012	X2013	X2014	X2015
Fire-Biomass burning	ha	47	184	96	258	1,518
Shifting Cultivation	ha	N/A	N/A	N/A	N/A	N/A

Table 9: Results of Monte Carlo simulations of Activity Data tabsheet using SimVoi

Drivers	units	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
Forestry infrastructure		225	194	270	229	325	288	224	244	228	185	215	83
Agriculture		36	384	463	882	436	401	487	291	261	126	436	305
Mining (med & large scale)		8,835	13,157	7,687	12,583	7,673	7,082	6,619	6,295	5,518	6,980	7,303	6,163
Infrastructure	ha	117	121	331	143	182	236	196	72	33	188	101	117
Settlements		-	-	25	73	7	7	7	9	29	53	110	210
Fire-Biomass burning		53	188	89	301	1,833	1,553	460	681	6,194	3,364	221	365
Shifting Cultivation								451	361	509	478	456	109
Deforestation		9,266	14,044	8,863	14,210	10,457	9,568	7,993	7,592	12,772	11,373	8,842	7,353
Logging - harvest volume	m3	608,730	585,108	624,287	759,684	655,406	500,788	533,106	546,242	521,172	545,355	547,516	547,517
Logging - skid trail length	km	2,302	2,212	2,360	2,872	2,478	1,893	2,016	2,065	1,971	2,062	2,070	2,070
Illegal logging	m3	2776	2,306	2,371	2,836	1,505	2,249	2,706	3,719	2,149	1,281	1,281	1,281
Mining and Infrastructure	ha						36,647	31,919	28,185	23,028	22,795	26,651	26,651

Replication results

In the following chunk, we compute the final uncertainty estimates for the simulated emission reductions (GHG ER) using a truncated normal distribution. With 10,000 simulation trials, we calculate key statistics, mean, standard deviation, and mean standard error, along with distributional percentiles. By extracting the 5th and 95th percentiles, we form a 90% confidence interval (CI) from which we derive the margin of error (ME) and its corresponding percentage error relative to the mean. These computed metrics replicate the SimVoi univariate summary and provide a statistical summary of the uncertainty associated with the emissions reduction estimates.

```r
# Emission Reductions (MC-R)
ER_values <- c(7715885, 10371977, 10040723, 6358705, 7174999, 6977178, 9223423, 7299024)
ER_mean_emp <- mean(ER_values)
ER_sd_emp    <- sd(ER_values)
n_sim <- 10000

# Simulate a truncated normal distribution & compute stats
sim_ER <- rtruncnorm(n = n_sim, a = 0, b = Inf, mean = ER_mean_emp, sd = ER_sd_emp)
sim_mean <- mean(sim_ER)
sim_sd    <- sd(sim_ER)
sim_se    <- sim_sd / sqrt(length(sim_ER))
sim_skew <- moments::skewness(sim_ER)
sim_quant <- quantile(sim_ER, probs = c(0, 0.25, 0.5, 0.75, 1))
lower90 <- quantile(sim_ER, probs = 0.05)
upper90 <- quantile(sim_ER, probs = 0.95)
ci90 <- upper90 - lower90
ME <- ci90 / 2
pct_error <- ME / sim_mean * 100

ER_summary <- data.frame(
  Metric = c("Mean", "St. Dev.", "Mean St. Error", "Skewness",
             "Minimum", "1st Quartile", "Median", "3rd Quartile", "Maximum",
             "5th Percentile", "95th Percentile", "90% CI", "Margin of Error", "% Error"),
  Value  = c(round(sim_mean),      round(sim_sd),
             round(sim_se),        round(sim_skew, 3),
```

```
              round(sim_quant[1]), round(sim_quant[2]),
              round(sim_quant[3]), round(sim_quant[4]),
              round(sim_quant[5]), round(lower90),
              round(upper90), round(ci90),
              round(ME), round(pct_error, 2))
  )
```

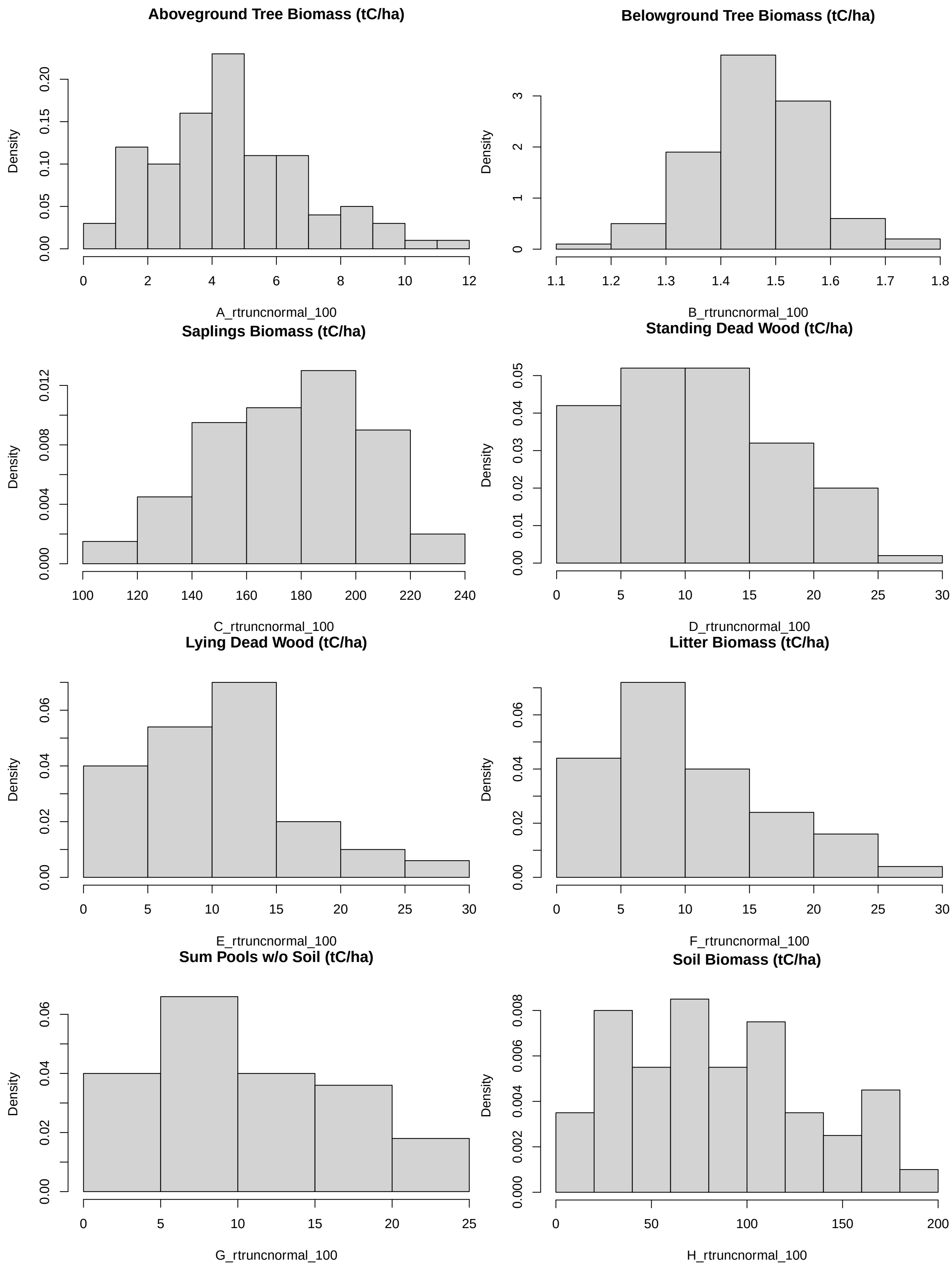
Table 10: Simulated Univariate Summary of Emission Reductions (GHG ER)

Metric	Value
Mean	8,126,521.00
St. Dev.	1,517,506.00
Mean St. Error	15,175.00
Skewness	0.02
Minimum	2,814,075.00
1st Quartile	7,121,279.00
Median	8,103,663.00
3rd Quartile	9,144,927.00
Maximum	14,821,484.00
5th Percentile	5,649,961.00
95th Percentile	10,631,361.00
90% CI	4,981,401.00
Margin of Error	2,490,700.00
% Error	30.65

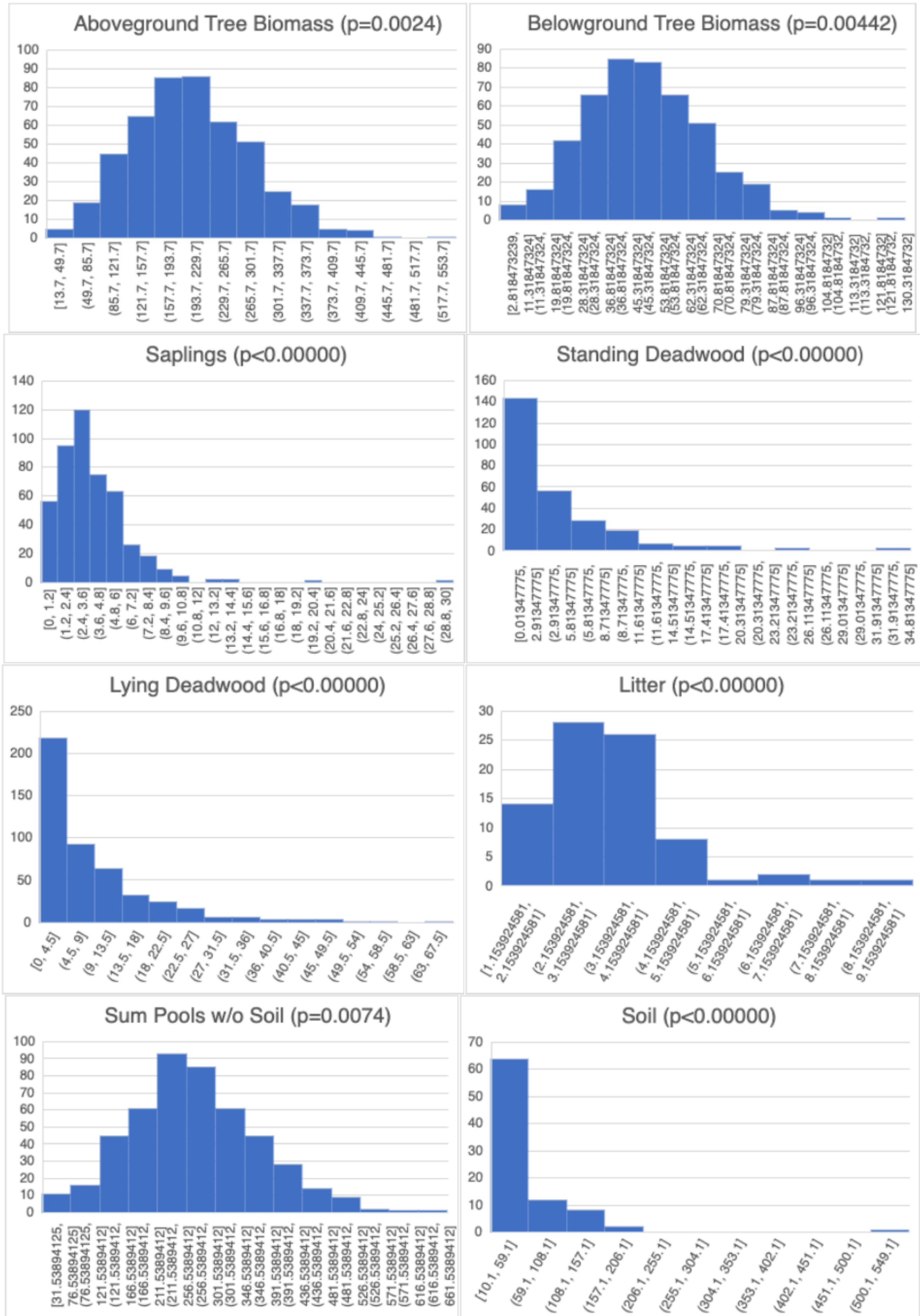
Distribution analysis

Distribution analysis is critical for ensuring Monte Carlo simulations accurately reflect the empirical characteristics of input data. Visual and statistical checks, such as normality tests (Shapiro, Sandford & Wilk, 1965) and distribution plots (Figures 9–16), verify data shape, spread, skewness, and outliers. Failure to perform this step risks masking or exaggerating inherent biases, which could undermine the reliability of our estimates. Figures 1–8 display the simulated distributions, while Figures 9–16 show the original data distributions. Discrepancies between them highlight opportunities for effectively reducing uncertainty. We recommend to refine the current simulation approach by incorporating data-specific models fitted with log-normal or beta distributions.

```
# Distribution of Monte Carlo derived estimates
hist(A_rtruncnormal_100, freq=F, main="Aboveground Tree Biomass (tC/ha)")
hist(B_rtruncnormal_100, freq=F, main="Belowground Tree Biomass (tC/ha)")
hist(C_rtruncnormal_100, freq=F, main="Saplings Biomass (tC/ha)")
hist(D_rtruncnormal_100, freq=F, main="Standing Dead Wood (tC/ha)")
hist(E_rtruncnormal_100, freq=F, main="Lying Dead Wood (tC/ha)")
hist(F_rtruncnormal_100, freq=F, main="Litter Biomass (tC/ha)")
hist(G_rtruncnormal_100, freq=F, main="Sum Pools w/o Soil (tC/ha)")
hist(H_rtruncnormal_100, freq=F, main="Soil Biomass (tC/ha)")
```


Aboveground Tree Biomass (tC/ha)
Density
A_rtruncnormal_100
Belowground Tree Biomass (tC/ha)
Density
B_rtruncnormal_100
Saplings Biomass (tC/ha)
Density
C_rtruncnormal_100
Standing Dead Wood (tC/ha)
Density
D_rtruncnormal_100
Lying Dead Wood (tC/ha)
Density
E_rtruncnormal_100
Litter Biomass (tC/ha)
Density
F_rtruncnormal_100
Sum Pools w/o Soil (tC/ha)
Density
G_rtruncnormal_100
Soil Biomass (tC/ha)
Density
H_rtruncnormal_100

Figures 1-8: Distribution analysis of simulated estimates of carbon stock variables



***Figures 9-16: Distribution analysis of carbon stock input variables (Bhanti, 2025-03-17)* {#sec-figures-9-16-distribution-analysis-of-carbon-stock-input-variables-(bhanti-2025-03-17}**

In addition, these distributional visualizations above offer auditors useful diagnostic tools, enabling rapid identification and characterization of biases commonly encountered in biomass data. Such diagrams help auditors efficiently assess the statistical approaches implemented by the project to monitor and manage uncertainty. Winrock recommends incorporating distribution analyses early in a project's reporting lifecycle with the aim of reducing audit findings, lowering uncertainty, and enhancing financial outcomes for Guyana's REDD+ activities. Early distribution analysis likewise informs simulations & parameters in SimVoi.

To guide practitioners in appropriate simulation designs, the following two tables present findings from a rapid literature review of Monte Carlo methods in forestry and REDD+ contexts (Annex II).

Table 9: Continuous data distributions, example cases & equations used in Monte Carlo simulations.

Distribution	Statistical Use Cases	PDF
Normal	Symmetric, bell-shaped distribution used for modeling continuous variables: biomass/ha	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
Lognormal	Right-skewed distribution suitable for variables constrained to positive values (e.g., emission rates).	$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x-\mu)^2}{2\sigma^2}\right)$
Exponential	Models waiting times between independent events, such as forest fire occurrences or logging events.	$f(x) = \lambda e^{-\lambda x}, \quad x \geq 0$
Cont. Uniform	Assumes all values in an interval [a, b] are equally likely; useful for random spatial sampling in forests.	$f(x) = \frac{1}{b-a}, \quad a \leq x \leq b$
Chi-Square	Often used in goodness-of-fit tests to evaluate model accuracy in biomass estimation.	$f(x) = \frac{1}{2^{k/2}\Gamma(k/2)} x^{\frac{k}{2}-1} e^{-x/2}, \quad x > 0$
t-Distribution	Suitable for small sample sizes with unknown population stdev (e.g., limited forest carbon data).	$f(x) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{\sqrt{v\pi}\,\Gamma\left(\frac{v}{2}\right)} \left(1+\frac{x^2}{v}\right)^{-\frac{v+1}{2}}$
Gamma	Models positively skewed data, such as biomass growth rates or carbon accumulation over time.	$f(x) = \frac{x^{k-1} e^{-x/\theta}}{\theta^k \Gamma(k)}$
Weibull	Flexible distribution used in reliability analysis, e.g., modeling tree mortality.	$f(x) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}$

Table 10: Discrete data distributions, example cases & equations used in Monte Carlo simulations.

Distribution	Statistical Use Cases	PMF
Bernoulli	Binary outcome probability, e.g., presence/absence of deforestation in an area.	$P(X=x) = p^x(1-p)^{1-x}$, $x \in \{0,1\}$
Binomial	Probability of fixed #no. of successes over n Bernoulli trials, e.g., no. of heads in ten coin flips.	$P(X=k) = \binom{n}{k} p^k (1-p)^{n-k}$, $k = 0,1,\ldots,n$
Poisson	Models counts of independent events within an interval, e.g., number of wildfire incidents per year.	$P(X=k) = \frac{\lambda^k e^{-\lambda}}{k!}$, $k = 0,1,2,\ldots$
Geometric	Models #no. of trials until the first success, e.g., number of inspections until detecting deforestation.	$P(X=k) = (1-p)^{k-1}p$, $k = 1,2,\ldots$
Negative Binomial	Counts #no. failures until r successes occur, treats overdispersed or repeated deforestation detections.	$P(X=k) = \binom{k+r-1}{k}(1-p)^r p^k$, $k = 0,1,2,\ldots$
Discrete Uniform	Assumes outcome in a finite set is equally likely, e.g., random sampling of inventory across a forest.	$P(X=x) = \frac{1}{n}$, $x = 1,2,\ldots,n$

Discrete distributions model countable events in forestry, such as deforestation counts, wildfire occurrences, or logged trees, using tools like the Binomial, Poisson, or Negative Binomial distributions to improve prediction accuracy and uncertainty assessments. For example, a Poisson distribution can enhance precision in estimating deforestation emissions from illegal logging.

Continuous distributions apply to variables that assume any value within a range, such as tree heights, carbon stock densities, or biomass. Common models, including Normal, Lognormal, Weibull, and Gamma distributions, capture ecological variability effectively; for instance, a Lognormal distribution often provides more reliable biomass estimates for right-skewed data.

The core mathematical concepts are the Probability Mass Function (PMF) for discrete data and the Probability Density Function (PDF) for continuous data. Accurate use of PMFs and PDFs is essential in Monte Carlo simulations, as they underpin random sampling processes that directly influence the reliability of uncertainty estimates. Rigorous selection of these functions enhances biomass and emissions estimates, reduces uncertainty, and supports the credibility of REDD+ reporting (Morgan & Henrion, 1990; IPCC, 2019; ART, 2021).

Early exploratory data analysis, including statistical normality tests and visual assessments (histograms, kernel density plots, Q-Q plots), is recommended to diagnose data distributions, optimize model selection, and reduce audit findings, ultimately improving the financial and environmental outcomes of national REDD+ monitoring programs.

Recommendations

- Regarding simulation parameters, *"RandTruncNormal(Mean,StDev,MinValue,MaxValue)) uses values of RandNormal until a value is found between MinValue and MaxValue or until it has made 10,000 attempts."* The formula applied in Guyana's verified workbook uses a minimum value of `0`. Considering the levels of quantitative variance between input values in the "CarbonStocks" tabsheet and SimVoi-simulated estimates in "CarbonStocks (MC)" tabsheet, we may assume that approximately less than ~100 simulations were completed before values were reached "*between MinValue and MaxValue*" . It is useful to consider that with increasing numbers of simulations the closer the mean values generated should be to absolute input values. Therefore, in order to test this assumption we simply ran 10,000 simulations, which generated mean estimates of within `2 tC ha-1` of input values (`205 tC ha-1`) 10 times of 10 reruns. By reducing the number of iterations per simulation, we confirm our assumptions that the current SimVoi setup provided estimates from ~<100 simulations.
- We recommend increasing the number of simulation iterations from 100 to a higher number, in order to

reduce variance in and beteween simulated estimates, which thereby provide more robust uncertainty quantification.

- It is advised to consider performing sensitivity analyses in order to identify which input parameters contribute most to overall uncertainty, thereby informing targeted data collection and model refinement efforts.
- Furthermore, distribution analysis confirms that our simulation outputs approximate the expected statistical distributions of the underlying data. In the next phase, we recommend incorporating the observed distributional characteristics into the Monte Carlo simulation framework. This adjustment, using distribution-specific models (e.g., log-normal, beta, etc.), is essential for reducing uncertainty and ensuring that our estimates carry statistical significance.

References

(1) ART, S. *The REDD+ Environmental Excellence Standard*; 2021. https://www.artredd.org/wp-content/uploads/2021/12/TREES-2.0-August-2021-Clean.pdf.

(2) Bolker, B. (2008). *Ecological Models and Data in R.* Princeton University Press.

(3) Brown, I. F.; Foster Brown, I.; Martinelli, L. A.; Wayt Thomas, W.; Moreira, M. Z.; Cid Ferreira, C. A.; Victoria, R. A. Uncertainty in the Biomass of Amazonian Forests: An Example from Rondônia, Brazil. *Forest Ecology and Management* 1995, *75* (1–3), 175–189. https://doi.org/10.1016/0378-1127(94)03512-u.

(4) Cohen, R.; Kaino, J.; Okello, J. A.; Bosire, J. O.; Kairo, J. G.; Huxham, M.; Mencuccini, M. Uncertainty to Estimates of Above-Ground Biomass for Kenyan Mangroves: A Scaling Procedure from Tree to Landscape Level. In *Forest ecology and management*; 2013; Vol. 310, pp 968–982. https://doi.org/10.1016/j.foreco.2013.09.047.

(5) Chen, Q.; Laurin, G. V.; Valentini, R. Uncertainty of Remotely Sensed Aboveground Biomass over an African Tropical Forest: Propagating Errors from Trees to Plots to Pixels. *Remote Sensing of Environment* 2015, *160*, 134–143. https://doi.org/10.1016/j.rse.2015.01.009.

(6) Holdaway, R. J.; McNeill, S. J.; Mason, N. W. H.; Carswell, F. E. Propagating Uncertainty in Plot-Based Estimates of Forest Carbon Stock and Carbon Stock Change. *Ecosystems* 2014, *17*, 627–640. https://doi.org/10.1007/s10021-014-9749-5.

(7) Chen, Q.; McRoberts, R. E.; Wang, C.; Radtke, P. J. Forest Aboveground Biomass Mapping and Estimation Across Multiple Spatial Scales Using Model-Based Inference. *Remote Sensing of Environment* 2016, *184*, 350–360. https://doi.org/10.1016/j.rse.2016.07.023.

(8) Chave, J.; Condit, R.; Aguilar, S.; Hernandez, A.; Lao, S.; Perez, R. Error Propagation and Scaling for Tropical Forest Biomass Estimates. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 2004, *359* (1443), 409–420.

(9) Keller, M.; Palace, M.; Hurtt, G. Biomass Estimation in the Tapajos National Forest, Brazil. *Forest Ecology and Management* 2001, *154*, 371–382.

(10) McRoberts, R. E.; Moser, P.; Oliveira, L. Z.; Vibrans, A. C. A General Method for Assessing the Effects of Uncertainty in Individual-Tree Volume Model Predictions on Large-Area Volume Estimates 222 with a Subtropical Forest Illustration. *Canadian Journal of Forest Research* 2015, *45*.

(11) Melson, S. L.; Harmon, M. E.; Fried, J. S.; Domingo, J. B. Estimates of Live-Tree Carbon Stores in the Pacific Northwest Are Sensitive to Model Selection. *Carbon Balance and Management* 2011, *6*, 2.

(12) Molto, Q.; Rossi, V.; Blanc, L. Error Propagation in Biomass Estimation in Tropical Forests. *Methods in Ecology and Evolution* 2013, *4*, 175–183. https://doi.org/10.1111/j.2041-210x.2012.00266.x.

(13) Picard, N.; Bosela, F. B.; Rossi, V. Reducing the Error in Biomass Estimates Strongly Depends on Model Selection. *Annals of Forest Science* 2015, *72* (6), 811–823. https://doi.org/10.1007/s13595-014-0434-9.

(14) Yanai, R. D.; Battles, J. J.; Richardson, A. D.; Blodgett, C. A.; Wood, D. M.; Rastetter, E. B. Estimating Uncertainty in Ecosystem Budget Calculations. *Ecosystems* 2010, *13*, 239–248. https://doi.org/10.1007/s10021-010-9315-8.

(15) Limpert, E., Stahel, W. A., & Abbt, M. (2001). "Log-normal distributions across the sciences: Keys and clues." *BioScience*, 51(5), 341–352.

(16) Morgan, M. G., & Henrion, M. (1990). *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis.* Cambridge University Press.

(17) Ross, S. M. (2019). *Introduction to Probability Models* (12th ed.). Academic Press.

(18) Shapiro, Samuel Sanford, and Martin B. Wilk. "An analysis of variance test for normality (complete samples)." *Biometrika* 52, no. 3-4 (1965): 591-611.

Annex I: SimVoi replication in base-R

Using the following base functions from R avoids need for any package installations and limits risk of bugs or common problems with R versioning arising in the workflow. However, the base functions below are a little clunkier and may require additional guidance. Regarding R and package versions, please consult the Annex II which presents record of the current runtime used in this analysis.

```r
randtruncnormal_sim_10000 <- rnorm(n=10,mean=MEAN,sd=SD)
hist(randtruncnormal_sim_10000, freq=F)
AG_Tree_tC_ha   = mean(randtruncnormal_sim_10000)
AG_Tree_tCO2_ha = AG_Tree_tC_ha*(44/12)
AG_Tree_tC_ha
AG_Tree_tCO2_ha
#curve(dnorm(x, mean=MEAN, sd=SD), from=0, to=450, add=T, col="red")

# 10,000 simulations sampling 10 observations
randtruncnormal_sim_10000_10 = replicate(n=10000, rnorm(n=10,mean=MEAN,sd=SD))
hist(apply(X = randtruncnormal_sim_10000_10, MARGIN=2, FUN=mean))
sd(apply(X = randtruncnormal_sim_10000_10, MARGIN=2, FUN=mean))
mean(apply(X = randtruncnormal_sim_10000_10, MARGIN=2, FUN=mean))
(mean(apply(X = randtruncnormal_sim_10000_10, MARGIN=2, FUN=mean)))*(44/12)

# 10,000 simulations sampling 100 observations
randtruncnormal_sim_10000_100 = replicate(n=10000,rnorm(n=100,mean=MEAN,sd=SD))
hist(apply(X = randtruncnormal_sim_10000_100, MARGIN=2, FUN=mean))
sd(apply(X = randtruncnormal_sim_10000_100, MARGIN=2, FUN=mean))
mean(apply(X = randtruncnormal_sim_10000_100, MARGIN=2, FUN=mean))
(mean(apply(X = randtruncnormal_sim_10000_100, MARGIN=2, FUN=mean)))*(44/12)

# 10,000 simulations sampling 1,000 observations
randtruncnormal_sim_10000_1000 = replicate(n=10000, rnorm(n=1000,mean=MEAN,sd=SD))
hist(apply(X = randtruncnormal_sim_10000_1000, MARGIN=2, FUN=mean))
sd(apply(X = randtruncnormal_sim_10000_1000, MARGIN=2, FUN=mean))
mean(apply(X = randtruncnormal_sim_10000_1000, MARGIN=2, FUN=mean))
(mean(apply(X = randtruncnormal_sim_10000_1000, MARGIN=2, FUN=mean)))*(44/12)

# 10,000 simulations sampling 10,000 observations
randtruncnormal_sim_10000_10000 = replicate(n=10000,rnorm(n=10000,mean=MEAN,sd=SD))
```

```r
hist(apply(X = randtruncnormal_sim_10000_10000, MARGIN=2, FUN=mean))
sd(apply(X = randtruncnormal_sim_10000_10000, MARGIN=2, FUN=mean))
mean(apply(X = randtruncnormal_sim_10000_10000, MARGIN=2, FUN=mean))
(mean(apply(X = randtruncnormal_sim_10000_10000, MARGIN=2, FUN=mean)))*(44/12)
```

Annex II: Review of Monte Carlo methods in REDD+

Table A.2: Search parameters, resource scope, and objectives informing search

REDD+[2]	**MC Application**	**Region**	**Key Findings**
ADD	Uncertainty of SAAB estimate	Rondônia, Brazil	Estimated $\pm$ 20% measurement error in SAAB using Monte Carlo simulations; emphasized large trees' role in biomass.
ADD	AGB Uncertainty	Kenya, Mozambique	Assessed mixed-effects models in estimating mangrove biomass.
ADD	Blanket uncertainty propagation	Ghana	AGB prediction error >20%; addressed error propagation from trees to pixels in remote sensing.
ADD	Plot-based uncertainty	New Zealand	Cross-plot variance greatest magnitude of uncertainty
JNR	Multi-scale AGB uncertainty modeling	Minnesota, USA	Cross-scale tests showing effects of spatial resolution on AGB uncertainty.
N/A	Allometric uncertainty modeling	Panama	Allometric models identified as largest source of biomass estimation error.
ADD	Sampling and allometric uncertainty	Tapajos Nat Forest, Brazil	Significance of allometric models on uncertainty of root biomass, 95% CI, 21 plots.
ADD	Uncertainty of volume estimates	Santa Catarina, Brazil	Negligible effects of residual uncertainty on large-area estimates
N/A	Uncertainty metrics in model selection	Oregon, USA	Uncertainty estimates call for local validation or new local model development
ADD	AGB model uncertainty	French Guiana	AGB sub-model errors dominate uncertainty; height and wood-specific gravity errors are minor but can cause bias.

[2] 1. ADD: Avoided deforestation degradation, IFM: Improved forest management, JNR: Jurisdictional nested REDD+

IFM	Emission factor uncertainty	Central Africa	Model selection is the largest error source (40%); weighting models reduces uncertainty in emission factors.
NA	Uncertainty in ecosystem nutrient estimate	New Hampshire, USA	Identified 8% uncertainty in nitrogen budgets, mainly from plot variability (6%) and allometric errors (5%).

Annex III: Runtime snapshot

```r
devtools::session_info()
```

```
- Session info ---------------------------------------------------------------
 setting  value
 version  R version 4.3.0 (2023-04-21)
 os       macOS 15.4.1
 system   aarch64, darwin20
 ui       X11
 language (EN)
 collate  en_US.UTF-8
 ctype    en_US.UTF-8
 tz       America/Vancouver
 date     2025-05-05
 pandoc   3.6.1 @ /usr/local/bin/ (via rmarkdown)

- Packages -------------------------------------------------------------------
 package           * version    date (UTC) lib source
 animation         * 2.7        2021-10-07 [1] CRAN (R 4.3.3)
 askpass             1.2.1      2024-10-04 [1] CRAN (R 4.3.3)
 assertthat          0.2.1      2019-03-21 [1] CRAN (R 4.3.0)
 backports           1.5.0      2024-05-23 [1] CRAN (R 4.3.3)
 BIOMASS           * 2.2.3      2025-02-24 [1] CRAN (R 4.3.3)
 boot                1.3-31     2024-08-28 [1] CRAN (R 4.3.3)
 broom             * 1.0.7      2024-09-26 [1] CRAN (R 4.3.3)
 c2z               * 0.2.0      2023-08-10 [1] CRAN (R 4.3.0)
 cachem              1.1.0      2024-05-16 [1] CRAN (R 4.3.3)
 caret             * 7.0-1      2024-12-10 [1] CRAN (R 4.3.3)
 cellranger          1.1.0      2016-07-27 [1] CRAN (R 4.3.0)
 chromote            0.4.0      2025-01-25 [1] CRAN (R 4.3.3)
 class               7.3-23     2025-01-01 [1] CRAN (R 4.3.3)
 classInt            0.4-11     2025-01-08 [1] CRAN (R 4.3.3)
 cli                 3.6.3      2024-06-21 [1] CRAN (R 4.3.3)
 codetools           0.2-20     2024-03-31 [1] CRAN (R 4.3.1)
 colorspace          2.1-1      2024-07-26 [1] CRAN (R 4.3.3)
 data.table          1.16.4     2024-12-06 [1] CRAN (R 4.3.3)
 dataMaid          * 1.4.1      2021-10-08 [1] CRAN (R 4.3.0)
 DBI                 1.2.3      2024-06-02 [1] CRAN (R 4.3.3)
 DEoptimR            1.1-3-1    2024-11-23 [1] CRAN (R 4.3.3)
 DescTools         * 0.99.59    2025-01-26 [1] CRAN (R 4.3.3)
 devtools            2.4.5      2022-10-11 [1] CRAN (R 4.3.0)
```

```
dials             * 1.3.0     2024-07-30 [1] CRAN (R 4.3.3)
DiceDesign          1.10      2023-12-07 [1] CRAN (R 4.3.1)
digest              0.6.37    2024-08-19 [1] CRAN (R 4.3.3)
dplyr             * 1.1.4     2023-11-17 [1] CRAN (R 4.3.1)
e1071               1.7-16    2024-09-16 [1] CRAN (R 4.3.3)
easypackages        0.1.0     2016-12-05 [1] CRAN (R 4.3.0)
ellipsis            0.3.2     2021-04-29 [1] CRAN (R 4.3.0)
evaluate            1.0.3     2025-01-10 [1] CRAN (R 4.3.3)
Exact               3.3       2024-07-21 [1] CRAN (R 4.3.3)
expm                1.0-0     2024-08-19 [1] CRAN (R 4.3.3)
extrafont         * 0.19      2023-01-18 [1] CRAN (R 4.3.3)
extrafontdb         1.0       2012-06-11 [1] CRAN (R 4.3.3)
fastmap             1.2.0     2024-05-15 [1] CRAN (R 4.3.3)
flextable         * 0.9.7     2024-10-27 [1] CRAN (R 4.3.3)
fontBitstreamVera   0.1.1     2017-02-01 [1] CRAN (R 4.3.3)
fontLiberation      0.1.0     2016-10-15 [1] CRAN (R 4.3.3)
fontquiver          0.2.1     2017-02-01 [1] CRAN (R 4.3.3)
forcats           * 1.0.0     2023-01-29 [1] CRAN (R 4.3.0)
foreach             1.5.2     2022-02-02 [1] CRAN (R 4.3.0)
formatR           * 1.14      2023-01-17 [1] CRAN (R 4.3.3)
fs                  1.6.5     2024-10-30 [1] CRAN (R 4.3.3)
furrr               0.3.1     2022-08-15 [1] CRAN (R 4.3.0)
future              1.34.0    2024-07-29 [1] CRAN (R 4.3.3)
future.apply        1.11.3    2024-10-27 [1] CRAN (R 4.3.3)
gdtools             0.4.1     2024-11-04 [1] CRAN (R 4.3.3)
generics            0.1.3     2022-07-05 [1] CRAN (R 4.3.0)
ggplot2           * 3.5.1     2024-04-23 [1] CRAN (R 4.3.1)
gld                 2.6.7     2025-01-17 [1] CRAN (R 4.3.3)
globals             0.16.3    2024-03-08 [1] CRAN (R 4.3.1)
glue                1.8.0     2024-09-30 [1] CRAN (R 4.3.3)
gower               1.0.2     2024-12-17 [1] CRAN (R 4.3.3)
GPfit               1.0-8     2019-02-08 [1] CRAN (R 4.3.0)
gridExtra           2.3       2017-09-09 [1] CRAN (R 4.3.0)
gtable              0.3.6     2024-10-25 [1] CRAN (R 4.3.3)
hardhat             1.4.0     2024-06-02 [1] CRAN (R 4.3.3)
haven               2.5.4     2023-11-30 [1] CRAN (R 4.3.1)
hms                 1.1.3     2023-03-21 [1] CRAN (R 4.3.0)
htmltools         * 0.5.8.1   2024-04-04 [1] CRAN (R 4.3.1)
htmlwidgets         1.6.4     2023-12-06 [1] CRAN (R 4.3.1)
httpuv              1.6.15    2024-03-26 [1] CRAN (R 4.3.1)
httr                1.4.7     2023-08-15 [1] CRAN (R 4.3.0)
infer             * 1.0.7     2024-03-25 [1] CRAN (R 4.3.1)
ipred               0.9-15    2024-07-18 [1] CRAN (R 4.3.3)
iterators           1.0.14    2022-02-05 [1] CRAN (R 4.3.0)
janitor           * 2.2.1     2024-12-22 [1] CRAN (R 4.3.3)
jsonlite          * 1.8.9     2024-09-20 [1] CRAN (R 4.3.3)
kableExtra        * 1.4.0     2024-01-24 [1] CRAN (R 4.3.1)
kernlab           * 0.9-33    2024-08-13 [1] CRAN (R 4.3.3)
KernSmooth          2.23-26   2025-01-01 [1] CRAN (R 4.3.3)
knitr             * 1.49      2024-11-08 [1] CRAN (R 4.3.3)
later               1.4.1     2024-11-27 [1] CRAN (R 4.3.3)
latex2exp         * 0.9.6     2022-11-28 [1] CRAN (R 4.3.0)
latexpdf          * 0.1.8     2023-12-19 [1] CRAN (R 4.3.3)
lattice           * 0.22-6    2024-03-20 [1] CRAN (R 4.3.1)
```

```
lava                 1.8.1      2025-01-12 [1] CRAN (R 4.3.3)
lhs                  1.2.0      2024-06-30 [1] CRAN (R 4.3.3)
lifecycle            1.0.4      2023-11-07 [1] CRAN (R 4.3.1)
listenv              0.9.1      2024-01-29 [1] CRAN (R 4.3.1)
lmom                 3.2        2024-09-30 [1] CRAN (R 4.3.3)
lubridate          * 1.9.4      2024-12-08 [1] CRAN (R 4.3.3)
magrittr           * 2.0.3      2022-03-30 [1] CRAN (R 4.3.0)
MASS               * 7.3-58.4   2023-03-07 [2] CRAN (R 4.3.0)
Matrix               1.6-5      2024-01-11 [1] CRAN (R 4.3.1)
memoise              2.0.1      2021-11-26 [1] CRAN (R 4.3.0)
mime                 0.12       2021-09-28 [1] CRAN (R 4.3.0)
miniUI               0.1.1.1    2018-05-18 [1] CRAN (R 4.3.0)
minpack.lm           1.2-4      2023-09-11 [1] CRAN (R 4.3.3)
modeldata          * 1.4.0      2024-06-19 [1] CRAN (R 4.3.3)
ModelMetrics         1.2.2.2    2020-03-17 [1] CRAN (R 4.3.0)
moments            * 0.14.1     2022-05-02 [1] CRAN (R 4.3.3)
munsell              0.5.1      2024-04-01 [1] CRAN (R 4.3.1)
mvtnorm              1.3-3      2025-01-10 [1] CRAN (R 4.3.3)
nlme                 3.1-166    2024-08-14 [1] CRAN (R 4.3.3)
nnet                 7.3-20     2025-01-01 [1] CRAN (R 4.3.3)
officer            * 0.6.7      2024-10-09 [1] CRAN (R 4.3.3)
openssl              2.3.1      2025-01-09 [1] CRAN (R 4.3.3)
pander               0.6.6      2025-03-01 [1] CRAN (R 4.3.3)
parallelly           1.41.0     2024-12-18 [1] CRAN (R 4.3.3)
parsnip            * 1.2.1      2024-03-22 [1] CRAN (R 4.3.1)
pillar               1.10.1     2025-01-07 [1] CRAN (R 4.3.3)
pkgbuild             1.4.6      2025-01-16 [1] CRAN (R 4.3.3)
pkgconfig            2.0.3      2019-09-22 [1] CRAN (R 4.3.0)
pkgload              1.4.0      2024-06-28 [1] CRAN (R 4.3.3)
plyr                 1.8.9      2023-10-02 [1] CRAN (R 4.3.1)
pROC                 1.18.5     2023-11-01 [1] CRAN (R 4.3.1)
processx             3.8.5      2025-01-08 [1] CRAN (R 4.3.3)
prodlim              2024.06.25 2024-06-24 [1] CRAN (R 4.3.3)
profvis              0.4.0      2024-09-20 [1] CRAN (R 4.3.3)
promises             1.3.2      2024-11-28 [1] CRAN (R 4.3.3)
proxy                0.4-27     2022-06-09 [1] CRAN (R 4.3.0)
ps                   1.8.1      2024-10-28 [1] CRAN (R 4.3.3)
purrr              * 1.0.2      2023-08-10 [1] CRAN (R 4.3.0)
R6                   2.5.1      2021-08-19 [1] CRAN (R 4.3.0)
ragg                 1.3.3      2024-09-11 [1] CRAN (R 4.3.3)
rappdirs             0.3.3      2021-01-31 [1] CRAN (R 4.3.0)
Rcpp                 1.0.14     2025-01-12 [1] CRAN (R 4.3.3)
readr              * 2.1.5      2024-01-10 [1] CRAN (R 4.3.1)
readxl             * 1.4.3      2023-07-06 [1] CRAN (R 4.3.0)
recipes            * 1.1.0      2024-07-04 [1] CRAN (R 4.3.3)
remotes              2.5.0      2024-03-17 [1] CRAN (R 4.3.1)
reshape2           * 1.4.4      2020-04-09 [1] CRAN (R 4.3.0)
rlang                1.1.4      2024-06-04 [1] CRAN (R 4.3.3)
rmarkdown          * 2.29       2024-11-04 [1] CRAN (R 4.3.3)
robustbase           0.99-4-1   2024-09-27 [1] CRAN (R 4.3.3)
rootSolve            1.8.2.4    2023-09-21 [1] CRAN (R 4.3.3)
rpart                4.1.24     2025-01-07 [1] CRAN (R 4.3.3)
rsample            * 1.2.1      2024-03-25 [1] CRAN (R 4.3.1)
rstudioapi           0.17.1     2024-10-22 [1] CRAN (R 4.3.3)
```

```
Rttf2pt1           1.3.12    2023-01-22 [1] CRAN (R 4.3.3)
rvest              1.0.4     2024-02-12 [1] CRAN (R 4.3.1)
scales           * 1.3.0     2023-11-28 [1] CRAN (R 4.3.1)
sessioninfo        1.2.2     2021-12-06 [1] CRAN (R 4.3.0)
sf                 1.0-21    2025-04-09 [1] Github (r-spatial/sf@ac49ec2)
shiny              1.10.0    2024-12-14 [1] CRAN (R 4.3.3)
snakecase          0.11.1    2023-08-27 [1] CRAN (R 4.3.0)
stringi            1.8.4     2024-05-06 [1] CRAN (R 4.3.1)
stringr          * 1.5.1     2023-11-14 [1] CRAN (R 4.3.1)
survival           3.8-3     2024-12-17 [1] CRAN (R 4.3.3)
svglite            2.1.3     2023-12-08 [1] CRAN (R 4.3.1)
systemfonts        1.1.0     2024-05-15 [1] CRAN (R 4.3.3)
terra              1.8-42    2025-04-02 [1] CRAN (R 4.3.0)
textshaping        0.4.1     2024-12-06 [1] CRAN (R 4.3.3)
tibble           * 3.2.1     2023-03-20 [1] CRAN (R 4.3.0)
tidymodels       * 1.2.0     2024-03-25 [1] CRAN (R 4.3.1)
tidyr            * 1.3.1     2024-01-24 [1] CRAN (R 4.3.1)
tidyselect         1.2.1     2024-03-11 [1] CRAN (R 4.3.1)
tidyverse        * 2.0.0     2023-02-22 [1] CRAN (R 4.3.0)
timechange         0.3.0     2024-01-18 [1] CRAN (R 4.3.1)
timeDate           4041.110  2024-09-22 [1] CRAN (R 4.3.3)
tinytex          * 0.54      2024-11-01 [1] CRAN (R 4.3.3)
truncnorm        * 1.0-9     2023-03-20 [1] CRAN (R 4.3.3)
tune             * 1.2.1     2024-04-18 [1] CRAN (R 4.3.1)
tzdb               0.4.0     2023-05-12 [1] CRAN (R 4.3.0)
units              0.8-7     2025-03-11 [1] CRAN (R 4.3.3)
urlchecker         1.0.1     2021-11-30 [1] CRAN (R 4.3.0)
useful           * 1.2.6.1   2023-10-24 [1] CRAN (R 4.3.1)
usethis            3.1.0     2024-11-26 [1] CRAN (R 4.3.3)
uuid               1.2-1     2024-07-29 [1] CRAN (R 4.3.3)
vctrs              0.6.5     2023-12-01 [1] CRAN (R 4.3.1)
viridisLite        0.4.2     2023-05-02 [1] CRAN (R 4.3.0)
webshot          * 0.5.5     2023-06-26 [1] CRAN (R 4.3.0)
webshot2         * 0.1.1     2023-08-11 [1] CRAN (R 4.3.0)
websocket          1.4.2     2024-07-22 [1] CRAN (R 4.3.3)
withr              3.0.2     2024-10-28 [1] CRAN (R 4.3.3)
workflows        * 1.1.4     2024-02-19 [1] CRAN (R 4.3.1)
workflowsets     * 1.1.0     2024-03-21 [1] CRAN (R 4.3.1)
xfun               0.50      2025-01-07 [1] CRAN (R 4.3.3)
xml2               1.3.6     2023-12-04 [1] CRAN (R 4.3.1)
xtable             1.8-4     2019-04-21 [1] CRAN (R 4.3.0)
yaml               2.3.10    2024-07-26 [1] CRAN (R 4.3.3)
yardstick        * 1.3.1     2024-03-21 [1] CRAN (R 4.3.1)
zip                2.3.1     2024-01-27 [1] CRAN (R 4.3.1)

 [1] /Library/Frameworks/R.framework/Versions/4.1-arm64/Resources/library
 [2] /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library

------------------------------------------------------------------------------
```

```
#Sys.getenv()
#.libPaths()
```