TopdownApproach

Jinshi

May 19, 2019

# Limitation  
# 1. Savanna and grassland more than 20% area, but sample numbers are too small for Frrot, RC, and Ra/Gpp ratio  
# 2. Ra < 0 how to resolve this   
  
  
# Bottom-up approache to stimate GPP  
# Froot =   
# Fshoot =   
# FsFr =   
# Ra = Rroot + Rshoot

## Abbreviation

* Rroot - root respiration
* Rshoot - shoot respiration
* Ra - autotrophic respiration (Rroot + Rshoot)
* GPP - gross primary production
* NPP - net primary production
* NEP - net primary production
* Froot - Rroot / Ra (root autotrophic respiration to total autotrophic respiration fraction/ratio, calculation please see plot\_froot function)
* Fshoot - Rshoot / Ra (1-Froot)
* FsFr - (1-Froot)/Froot (FsFr = Rshoot/Rroot ratio)
* RC - (root respiration / soil respiration)
* RaGPP - (Ra/GPP) ratio

# install.packages('kableExtra')  
# Load required packages  
library(data.table)  
  
library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':  
##   
## hour, isoweek, mday, minute, month, quarter, second, wday,  
## week, yday, year

## The following object is masked from 'package:base':  
##   
## date

library(kableExtra)  
library(cowplot)

## Loading required package: ggplot2

##   
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggplot2':  
##   
## ggsave

library(knitr)  
library("ggpubr")

## Loading required package: magrittr

##   
## Attaching package: 'ggpubr'

## The following object is masked from 'package:cowplot':  
##   
## get\_legend

library(reshape)

##   
## Attaching package: 'reshape'

## The following object is masked from 'package:lubridate':  
##   
## stamp

## The following object is masked from 'package:data.table':  
##   
## melt

library(ggplot2)  
# install.packages("ggmap")  
library(ggmap)

## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.

## Please cite ggmap if you use it! See citation("ggmap") for details.

##   
## Attaching package: 'ggmap'

## The following object is masked from 'package:magrittr':  
##   
## inset

## The following object is masked from 'package:cowplot':  
##   
## theme\_nothing

# install.packages("maps")  
library(maps)  
# install.packages("mapdata")  
library(mapdata)  
library(tidyr)

##   
## Attaching package: 'tidyr'

## The following objects are masked from 'package:reshape':  
##   
## expand, smiths

## The following object is masked from 'package:magrittr':  
##   
## extract

library(hexbin)  
library(cowplot)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:reshape':  
##   
## rename

## The following object is masked from 'package:kableExtra':  
##   
## group\_rows

## The following objects are masked from 'package:lubridate':  
##   
## intersect, setdiff, union

## The following objects are masked from 'package:data.table':  
##   
## between, first, last

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# Source all needed functions  
library(ggplot2)  
theme\_set(theme\_bw())  
  
source('functions.R')

# input data  
srdb\_v4 <- read.csv(file.path(DATA\_DIR, 'srdbv4.csv')) # SRDB\_V4 data  
FlFsFr <- read.csv(file.path(DATA\_DIR, 'FlFsFrSummary.csv')) # Froot (Rroot/Rs) and Fshoot (Rshoot/Rs) from 39 papers  
GPP <- read.csv(file.path(DATA\_DIR, 'GlobalGPP\_Sum.csv')) # 65 GPP estimates from published paper  
NPP <- read.csv(file.path(DATA\_DIR, 'ITONPP.CSV')) # 251 NPP estimates from ITO (2011)  
GlobalRs <- read.csv(file.path(DATA\_DIR, 'GlobalRs.csv')) # 27 estimates of global Rs  
# (Ra/GPP) Piao, Shilong, et al. "Forest annual carbon cost: A global‐scale analysis of autotrophic respiration." Ecology 91.3 (2010): 652-661.  
RaGPP <- read.csv(file.path(DATA\_DIR, 'RaGPP.csv'))   
IGBP <- read.table(file.path(DATA\_DIR, 'IGBP.txt'), header = TRUE, sep = ",")  
# sort(unique(IGBP$IGBP2001Pr))  
# There are 16 IGBP group  
IGBP %>% mutate(IGBP = case\_when( IGBP2001Pr == 0 ~ "WAT",  
 IGBP2001Pr == 1 ~ "ENF",  
 IGBP2001Pr == 2 ~ "EBF",  
 IGBP2001Pr == 3 ~ "DNF",  
 IGBP2001Pr == 4 ~ "DBF",  
 IGBP2001Pr == 5 ~ "MF",  
 IGBP2001Pr == 6 ~ "CSH",  
 IGBP2001Pr == 7 ~ "OSH",  
 IGBP2001Pr == 8 ~ "WSA",  
 IGBP2001Pr == 9 ~ "SAV",  
 IGBP2001Pr == 10 ~ "GRA",  
 IGBP2001Pr == 11 ~ "WET",  
 IGBP2001Pr == 12 ~ "CRO",  
 IGBP2001Pr == 13 ~ "URB",  
 IGBP2001Pr == 14 ~ "CVM",  
 IGBP2001Pr == 15 ~ "SNO",  
 IGBP2001Pr == 16 ~ "BSV",  
 TRUE ~ "UNC" ) ) -> IGBP  
  
# BBL: Interesting; I didn't know case\_when! Aanother way to do this would be   
# IGBP\_list <- c("WAT", "ENF", ...)  
# IGBP$IGBP <- IGBP\_list[IGBP$IGBP2001Pr]  
  
# 16 IGBP group into 10 Ecosystems  
IGBP %>% mutate (Ecosystem = case\_when( IGBP2001Pr == 0 ~ "Water",  
 IGBP2001Pr == 1 ~ "Forest",  
 IGBP2001Pr == 2 ~ "Forest",  
 IGBP2001Pr == 3 ~ "Forest",  
 IGBP2001Pr == 4 ~ "Forest",  
 IGBP2001Pr == 5 ~ "Forest",  
 IGBP2001Pr == 6 ~ "Shrubland",  
 IGBP2001Pr == 7 ~ "Shrubland",  
 IGBP2001Pr == 8 ~ "Savanna",  
 IGBP2001Pr == 9 ~ "Savanna",  
 IGBP2001Pr == 10 ~ "Grassland",  
 IGBP2001Pr == 11 ~ "Wetland",  
 IGBP2001Pr == 12 ~ "Agriculture",  
 IGBP2001Pr == 13 ~ "Urbland",  
 IGBP2001Pr == 14 ~ "Agriculture",  
 IGBP2001Pr == 15 ~ "Snow",  
 IGBP2001Pr == 16 ~ "Desert",  
 TRUE ~ "Unclassified")) -> IGBP  
# DGRsD <- read.csv(paste0(DATA\_DIR,'/', 'DGRsD.csv'), header = TRUE)  
  
AGB <- read.table(paste0(DATA\_DIR,'/', 'GLC2000\_Point.txt'), header = T, sep = ",")  
AGB %>% filter(RASTERVALU != -9999) -> AGB  
SOC <- read.table(paste0(DATA\_DIR,'/', 'SOC\_Point.txt'), header = T, sep = ",")

head(AGB)

## FID\_ SiteID Latitude Longitude RASTERVALU  
## 1 NA 1 55.75 -73.25 19.657058  
## 2 NA 2 55.75 73.25 4.629761  
## 3 NA 3 -14.25 -43.75 9.440806  
## 4 NA 4 14.25 43.75 50.291114  
## 5 NA 5 -85.75 -177.75 0.000000  
## 6 NA 6 -84.25 -106.75 0.000000

max(AGB$RASTERVALU)

## [1] 373.3373

min(AGB$RASTERVALU)

## [1] 0

max(SOC$RASTERVALU)

## [1] 9

min(SOC$RASTERVALU)

## [1] 0

# IGBP area by type  
IGBP %>%   
 filter(Ecosystem != "Water", Ecosystem != "Snow",  
 Ecosystem != "Unclassified", Ecosystem != "Urbland") %>%   
 group\_by(Ecosystem) %>%   
 summarise(count = n()) %>%   
 mutate(percent = count / sum(count))

## # A tibble: 7 x 3  
## Ecosystem count percent  
## <chr> <int> <dbl>  
## 1 Agriculture 8171 0.191   
## 2 Desert 7269 0.170   
## 3 Forest 9132 0.214   
## 4 Grassland 6545 0.153   
## 5 Savanna 6628 0.155   
## 6 Shrubland 4850 0.113   
## 7 Wetland 173 0.00405

# sum(0.191,0.17,0.214,0.153,0.155,0.113,0.004)  
  
# Remove desert and shrubland because no data for Froot and Ra/GPP ratio  
IGBP %>%   
 filter(Ecosystem != "Water", Ecosystem != "Snow",  
 Ecosystem != "Unclassified", Ecosystem != "Urbland",  
 Ecosystem != "Desert", Ecosystem != "Shrubland") %>%   
 group\_by(Ecosystem) %>% summarise ( count = n() ) %>% mutate (percent = count/sum(count))

## # A tibble: 5 x 3  
## Ecosystem count percent  
## <chr> <int> <dbl>  
## 1 Agriculture 8171 0.267   
## 2 Forest 9132 0.298   
## 3 Grassland 6545 0.214   
## 4 Savanna 6628 0.216   
## 5 Wetland 173 0.00564

sum(0.267,0.298,0.213,0.216,0.006)

## [1] 1

# plot Ra Rs relationship  
# Ra\_Rs\_relationship <- function (sdata) {  
# sdata$Ra\_Norm <- ifelse (is.na(sdata$Ra\_Norm), sdata$RS\_Norm - sdata$Rh\_Norm, sdata$Ra\_Norm)  
# sdata %>% filter (!is.na(Ra\_Norm) & !is.na(RS\_Norm) & !is.na(Ecosystem\_type)  
# & Ecosystem\_type != 'Forest') -> sdata  
# ggplot(data = sdata, aes(Ra\_Norm, RS\_Norm)) + geom\_point() + facet\_wrap(~Ecosystem\_type)  
# }  
  
# Ra\_Rs\_relationship(DGRsD)

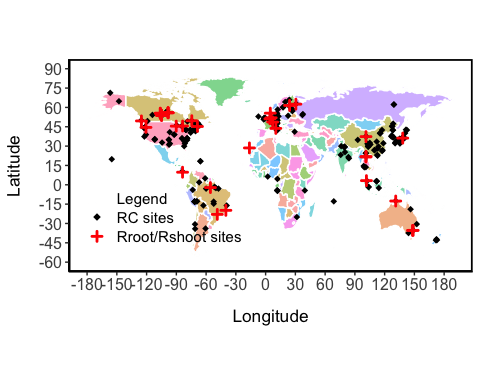
* 1. Introduction

\*\* Figure. Sites distribution – FlFsFr sites and RC/HC sites

# plot sites distribution   
plot\_sites(srdb\_v4, FlFsFr)

## Warning in qt(conf.interval/2 + 0.5, datac$N - 1): NaNs produced

## Warning: Removed 1 rows containing missing values (geom\_point).



# Get Froot and Fshoot information from FlFsFr dataset and srdb\_v4, see cal\_Froot() in functions for more information  
Froot <- cal\_Froot(FlFsFr, srdb\_v4)   
# check whether Froot + Fshoot = 1  
Froot %>% transmute (check = Froot + Fshoot) %>% max

## [1] 1

Froot %>% transmute (check = Froot + Fshoot) %>% min

## [1] 1

Froot %>% group\_by(Ecosystem) %>% summarise(count = n())

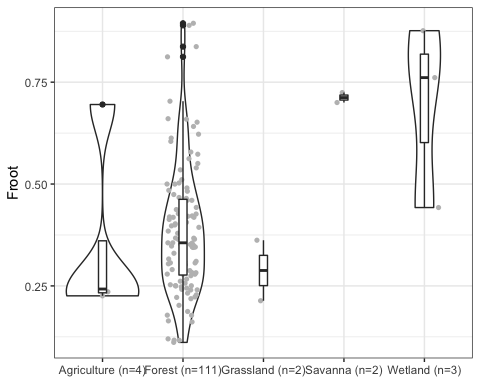
## # A tibble: 5 x 2  
## Ecosystem count  
## <fct> <int>  
## 1 Forest 111  
## 2 Grassland 2  
## 3 Savanna 2  
## 4 Agriculture 4  
## 5 Wetland 3

mean(Froot$Froot)

## [1] 0.394662

\*\* Plot Froot by ecosystem type

# potential issue: only two points from grassland  
# no point from tundra   
# BBL: difficult to see what's going on here; really needs more comments!  
  
plot\_froot <- function (sdata) {  
 var\_obs <- nrow(sdata)  
 # plot Froot (root respiration/aototrophic respiration ratio) by ecosystem type  
 Fl\_plot <- ggplot(sdata, aes(x = Ecosystem, y=Froot)) + geom\_violin() +  
 geom\_jitter(shape=16, position=position\_jitter(0.2), col = "gray") +  
 geom\_boxplot(width=.1) +  
 ylab("Froot") +  
 # stat\_summary(fun.y=median, geom="point", size=2, color="red") +  
 scale\_x\_discrete(limits = c("Agriculture", "Forest", "Grassland", "Savanna", "Wetland"),  
 labels = c("Agriculture (n=4)", "Forest (n=111)", "Grassland (n=2)", "Savanna (n=2)", "Wetland (n=3)") ) +  
 theme(axis.title.x=element\_blank())  
   
 print(Fl\_plot)  
}  
  
plot\_froot (Froot)



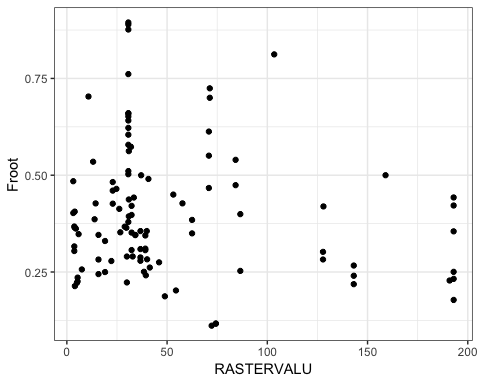
# Each row get the closese latitude and longitude values to AGB and SOC data  
AGB$Latitude[which.min(abs(61.85 - AGB$Latitude))]

## [1] 61.75

Froot %>% mutate(Latitude = round(Latitude\*2, 0)/2 - 0.25, Longitude = round(Longitude\*2, 0)/2-0.25) -> Froot  
# there are no clear relationship between Froot and above-ground biomass or soil organic carbon (SOC)  
left\_join(Froot, AGB) %>% ggplot() + aes(RASTERVALU, Froot) + geom\_point()

## Joining, by = c("Latitude", "Longitude")

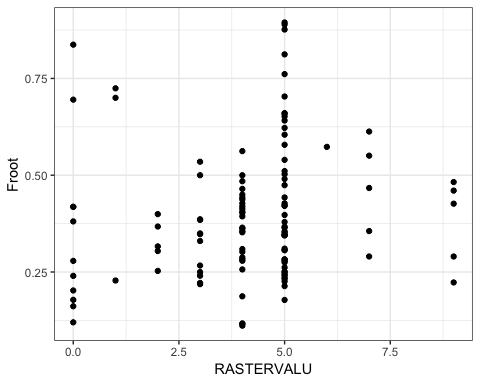
## Warning: Removed 14 rows containing missing values (geom\_point).



left\_join(Froot, SOC) %>% ggplot() + aes(RASTERVALU, Froot) + geom\_point()

## Joining, by = c("Latitude", "Longitude")

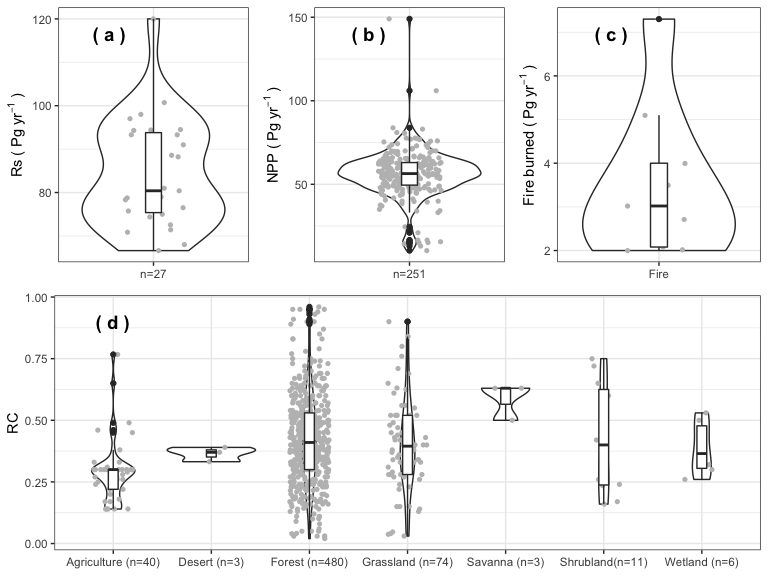
## Warning: Removed 4 rows containing missing values (geom\_point).



\*\* Plot global Rs, NPP, Fire-burned Carbon, and RC by ecosystem (root respiration to soil respiration ratio)

# more details please see function plot\_Rs()  
plot\_Rs (GlobalRs, srdb\_v4, NPP)

## # A tibble: 7 x 2  
## Ecosystem\_type n  
## <fct> <int>  
## 1 Agriculture 40  
## 2 Desert 3  
## 3 Forest 480  
## 4 Grassland 74  
## 5 Savanna 3  
## 6 Shrubland 11  
## 7 Wetland 6  
## [1] 0.4121125  
## [1] "RC se=0.00740339660633126"  
## [1] "RC 95% CI=0.015"  
## [1] "RC obs=617"  
## [1] 56.4  
## [1] "NPP se=0.902534118625694"  
## [1] "NPP 95% CI=1.778"

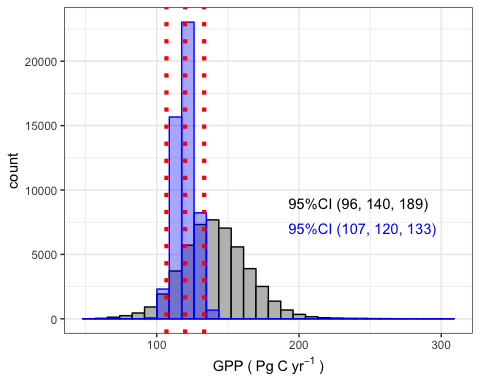


# Bottom-up approache to stimate GPP  
# Rroot = Rs \* RC (Rroot to Rs ratio, from srdb)  
# Froot = Rroot / Ra (root autotrophic respiration to total autotrophic respiration fraction/ratio, calculation please see plot\_froot function)  
# Fshoot = 1-Froot  
# FsFr = (1-Froot)/Froot (FsFr = Rshoot/Rroot ratio)  
# Ra = Rroot + Rshoot  
# GPP = NPP + Rroot + Rshoot  
  
# get subset data for Bottom-up approache to estimate GPP  
var\_RC <- srdb\_v4 %>% select(RC\_annual, Ecosystem\_type) %>% filter(RC\_annual > 0.01 & RC\_annual < 0.99 & !is.na(RC\_annual)) # dataset contain RC records  
var\_agr <- var\_RC %>% filter (Ecosystem\_type == "Agriculture") # dataset contain RC records only for agriculture ecosystem  
var\_des <- var\_RC %>% filter (Ecosystem\_type == "Desert") # dataset contain RC records only for desert  
var\_for <- var\_RC %>% filter (Ecosystem\_type == "Forest") # dataset contain RC records only for Forest ecosystem  
var\_gra <- var\_RC %>% filter (Ecosystem\_type == "Grassland") # dataset contain RC records only for grassland ecosystem  
var\_sav <- var\_RC %>% filter (Ecosystem\_type == "Savanna") # dataset contain RC records only for savanna ecosystem  
var\_shr <- var\_RC %>% filter (Ecosystem\_type == "Shrubland") # dataset contain RC records only for shrubland ecosystem  
var\_wet <- var\_RC %>% filter (Ecosystem\_type == "Wetland") # dataset contain RC records only for wetland ecosystem  
# var\_rest <- var\_RC %>% filter (Ecosystem\_type != "Forest") # dataset contain RC records for non-forest ecosystem  
var\_Rs <- GlobalRs %>% filter(!is.na(Rs) & is.na(SD) ) # dataset contain global Rs estimates (without standard diviation information)  
  
n\_samp <- 50000 # take 50000 samples  
# n\_rep <- 50000   
  
set.seed(1234567)  
gpp\_results <- tibble(  
 # Rs = replicate( n\_rep, sample(var\_Rs$Rs, n\_samp, replace = TRUE) %>% mean() ),  
 # Rs1 = rnorm(n = n\_samp, mean = 68, sd = 4),  
 Rs1 = sample(var\_Rs$Rs, n\_samp, replace = TRUE), # randomly take a sample from global Rs estimates, with replace=TRUE, repeat 50000 times  
 Rs2 = rnorm(n = n\_samp, mean = 80.4, sd = 16.9), # for those with standard diviation, 50000 samples were take from normal distribution ~ (mean=80.4, sd=16.9)  
 # repeat until all Rs estimates with sd samples were taken and hold in the outputs  
 Rs3 = rnorm(n = n\_samp, mean = 98, sd = 12), # same as above  
 Rs4 = rnorm(n = n\_samp, mean = 79, sd = 7.65), # same as above  
 Rs5 = rnorm(n = n\_samp, mean = 94.4, sd = 4.59), # same as above  
 Rs6 = rnorm(n = n\_samp, mean = 91, sd = 2.05), # same as above  
 Rs7 = rnorm(n = n\_samp, mean = 94.3, sd = 9.13), # same as above  
 Rs8 = rnorm(n = n\_samp, mean = 78.34, sd = 2.25), # same as above  
 Rs9 = rnorm(n = n\_samp, mean = 72.55, sd = 7.13), # same as above  
 # Rs was calculated as the mean of above 9 Rs estimates  
 Rs = (Rs1 + Rs2 + Rs3 + Rs4 + Rs5 + Rs6 + Rs7 + Rs8 + Rs9)/9,  
   
 # Rc = replicate( n\_rep, var\_RC$RC\_annual, n\_samp, replace = TRUE) %>% mean() ), # senario 0: Rc not seperate into different ecosystems, not used  
 # Senario 1: Rc seperate into 7 groups  
 Rc\_agr = sample(var\_agr$RC\_annual, n\_samp, replace = TRUE) , # randomly take a sample from RC estimates for agriculture, repeat 50000 times  
 Rc\_des = sample(var\_des$RC\_annual, n\_samp, replace = TRUE) , # randomly take a sample from RC estimates for desert, repeat 50000 times  
 Rc\_for = sample(var\_for$RC\_annual, n\_samp, replace = TRUE) , # randomly take a sample from RC estimates for forest, repeat 50000 times  
 Rc\_gra = sample(var\_gra$RC\_annual, n\_samp, replace = TRUE) , # randomly take a sample from RC estimates for grassland, repeat 50000 times  
 Rc\_sav = sample(var\_sav$RC\_annual, n\_samp, replace = TRUE) , # randomly take a sample from RC estimates for savanna, repeat 50000 times  
 Rc\_shr = sample(var\_shr$RC\_annual, n\_samp, replace = TRUE) , # randomly take a sample from RC estimates for shrubland, repeat 50000 times  
 Rc\_wet = sample(var\_wet$RC\_annual, n\_samp, replace = TRUE) , # randomly take a sample from RC estimates for wetland, repeat 50000 times  
 # calculation: Rc as a average of 7 ecosystem types, weighted by there area  
 Rc = (Rc\_agr\*0.191 + Rc\_des\*0.17 + Rc\_for\*0.214 + Rc\_gra\*0.153 + Rc\_sav\*0.155 + Rc\_shr\*0.113 + Rc\_wet\*0.004),   
 # Senario 2: Rc seperate into 4 groups: agriculture, forest, grassland, and rest (all other ecosystems with obs<40, take samples from all RC records)  
 Rc\_rest = sample(var\_RC$RC\_annual, n\_samp, replace = TRUE) , # randomly take a sample from all RC estimates, repeat 50000 times  
 # only seperate into four groups, agriculture, forest, grassland and the rest  
 Rc2 = (Rc\_agr\*0.191 + Rc\_for\*0.214 + Rc\_gra\*0.153 + Rc\_rest\*0.442),   
   
 # Froot = sample(Froot$Froot, n\_samp, replace = TRUE), # senario 0: Froot not seperate into different ecosystems, not used  
 Froot\_agr = sample(Froot[Froot$Ecosystem == "Agriculture",]$Froot, n\_samp, replace = TRUE), # take 50000 samples of Froot for agriculture   
 Froot\_for = sample(Froot[Froot$Ecosystem == "Forest",]$Froot, n\_samp, replace = TRUE), # take 50000 samples of Froot for forest   
 Froot\_gra = sample(Froot[Froot$Ecosystem == "Grassland",]$Froot, n\_samp, replace = TRUE), # take 50000 samples of Froot for grassland   
 Froot\_sav = sample(Froot[Froot$Ecosystem == "Savanna",]$Froot, n\_samp, replace = TRUE), # take 50000 samples of Froot for savanna   
 Froot\_wet = sample(Froot[Froot$Ecosystem == "Wetland",]$Froot, n\_samp, replace = TRUE), # take 50000 samples of Froot for agriculture   
 # take 50000 samples of Froot from all Froot recods, for senario2, those ecosystems have less than 40 obs use value from all Froot records  
 Froot\_rest = sample(Froot$Froot, n\_samp, replace = TRUE),   
   
 # senario1: Froot calculate from 5 ecosystems, weighted by area  
 Froot = Froot\_agr\*0.267 + Froot\_for\*0.298 + Froot\_gra\*0.213 + Froot\_sav\*0.216 + Froot\_wet\*0.006,  
 # Senario 2: seperate into 2 ecosystems and weighted by area  
 Froot2 = Froot\_rest \* 0.702 + Froot\_for\*0.298,  
 FsFr = (1 - Froot) / Froot, # calculate Froot to Fshoot ratio of senario 1  
 FsFr2 = (1 - Froot2) / Froot2, # calculate Froot to Fshoot ratio of senario 2  
 NPP = sample(NPP$NPP, n\_samp, replace = TRUE) , # take 500000 samples from NPP records  
   
 gpp1 = rnorm(n = n\_samp, mean = 124.6, sd = 2.7), # if gpp have mean and standard diviation, take 50000 samples from the normal distribution  
 gpp2 = rnorm(n = n\_samp, mean = 135.7, sd = 21.712), # same as above  
 gpp3 = rnorm(n = n\_samp, mean = 109.29, sd = 27.33), # same as above  
 gpp4 = rnorm(n = n\_samp, mean = 118, sd = 26), # same as above  
 gpp5 = rnorm(n = n\_samp, mean = 119, sd = 6), # same as above  
 gpp6 = rnorm(n = n\_samp, mean = 110.5, sd = 21.3), # same as above  
 gpp7 = rnorm(n = n\_samp, mean = 117, sd = 13), # same as above  
 gpp8 = sample(GPP[is.na(GPP$SD),]$GPP, n\_samp, replace = TRUE), # for those do not have SD information, take 50000 samples using boosting   
 GPP\_raw = (gpp1 + gpp2 + gpp3 + gpp4 + gpp5 + gpp6 + gpp7 + gpp8) / 8, # scenario 1: calculate   
   
 # Bottom up estimate of GPP, senario 1  
 Rroot = Rs \* Rc, # calculate root respiration  
 Rshoot = Rroot \* FsFr, # calculate shoot respiration  
   
 # Bottom up estimate of GPP, senario 2  
 Rroot2 = Rs \* Rc2,  
 Rshoot2 = Rroot2 \* FsFr2,  
   
 GPP = NPP + Rroot + Rshoot, # scenario 1: calculate GPP  
 GPP2 = NPP + Rroot2 + Rshoot2 # scenario 2: calculate GPP  
)

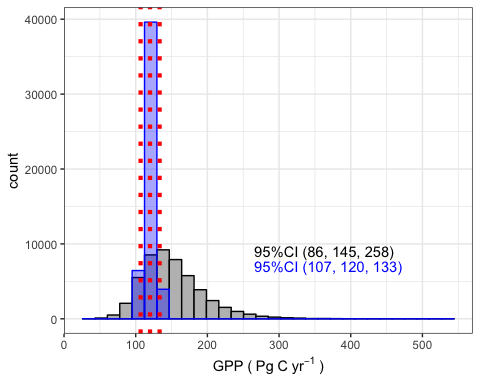
# check  
gpp\_results %>% mutate(check = (1 - Froot) / Froot, FsFr)

## # A tibble: 50,000 x 47  
## Rs1 Rs2 Rs3 Rs4 Rs5 Rs6 Rs7 Rs8 Rs9 Rs Rc\_agr  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 81.0 59.1 105. 74.2 92.6 90.9 100. 79.1 79.8 84.7 0.46  
## 2 93.3 81.9 110. 84.9 90.4 92.5 91.9 81.8 72.9 88.8 0.26  
## 3 97.0 59.4 95.4 81.8 99.9 89.9 94.3 80.7 71.9 85.6 0.3   
## 4 75 116. 101. 83.3 94.2 93.7 98.3 77.0 77.9 90.7 0.3   
## 5 93.3 70.9 93.0 81.7 88.8 88.0 99.0 77.7 68.5 84.5 0.14  
## 6 75.8 114. 90.9 55.8 92.8 92.2 81.8 77.1 80.5 84.5 0.3   
## 7 68 62.7 104. 67.0 86.8 93.5 82.4 74.1 76.2 79.4 0.38  
## 8 101. 96.1 101. 79.0 94.1 92.4 86.6 80.6 84.6 90.5 0.3   
## 9 75.8 74.5 94.9 75.2 95.7 91.2 97.8 76.5 86.7 85.4 0.3   
## 10 101. 88.4 81.4 77.9 92.4 91.0 90.6 81.9 61.6 85.1 0.28  
## # … with 49,990 more rows, and 36 more variables: Rc\_des <dbl>,  
## # Rc\_for <dbl>, Rc\_gra <dbl>, Rc\_sav <dbl>, Rc\_shr <dbl>, Rc\_wet <dbl>,  
## # Rc <dbl>, Rc\_rest <dbl>, Rc2 <dbl>, Froot\_agr <dbl>, Froot\_for <dbl>,  
## # Froot\_gra <dbl>, Froot\_sav <dbl>, Froot\_wet <dbl>, Froot\_rest <dbl>,  
## # Froot <dbl>, Froot2 <dbl>, FsFr <dbl>, FsFr2 <dbl>, NPP <dbl>,  
## # gpp1 <dbl>, gpp2 <dbl>, gpp3 <dbl>, gpp4 <dbl>, gpp5 <dbl>,  
## # gpp6 <dbl>, gpp7 <dbl>, gpp8 <dbl>, GPP\_raw <dbl>, Rroot <dbl>,  
## # Rshoot <dbl>, Rroot2 <dbl>, Rshoot2 <dbl>, GPP <dbl>, GPP2 <dbl>,  
## # check <dbl>

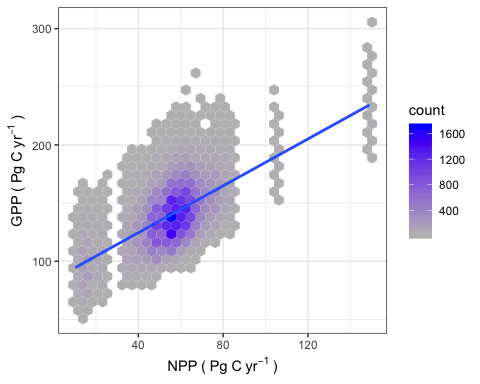
# hist(gpp\_results$GPP) + abline (v = quantile (gpp\_results$GPP, c(0.025, 0.5, 0.975)), col = "red", lty = "dashed")  
# mean(gpp\_results$GPP)  
# max(gpp\_results$GPP)  
  
# senario 1  
var\_CI <- quantile(gpp\_results$GPP, c(0.025, 0.5, 0.975)) %>% round(0)  
var\_CI\_raw <- quantile(gpp\_results$GPP\_raw, c(0.025, 0.5, 0.975)) %>% round(0)  
ggplot(gpp\_results) + aes(GPP) + geom\_histogram(col = "black", fill = "gray", bins = 30) +   
 geom\_histogram(aes(GPP\_raw), col = "blue", fill = "blue", bins = 30, alpha = 0.35) +  
 geom\_vline(xintercept = c(quantile (gpp\_results$GPP\_raw, c(0.025, 0.5, 0.975))),col = "red", linetype = "dotted", size = 1.5) +  
 annotate("text", x = quantile(gpp\_results$GPP, 0.98), y = 9000, label = paste0("95%CI (", var\_CI[1],", ", var\_CI[2], ", ", var\_CI[3], ")"), hjust = 0) +  
 annotate("text", x = quantile(gpp\_results$GPP, 0.98), y = 7000, label = paste0("95%CI (", var\_CI\_raw[1],", ", var\_CI\_raw[2], ", ", var\_CI\_raw[3], ")")  
 , hjust = 0, col = "blue") +  
 xlab(expression(GPP~"("~Pg~C~yr^{-1}~")"))



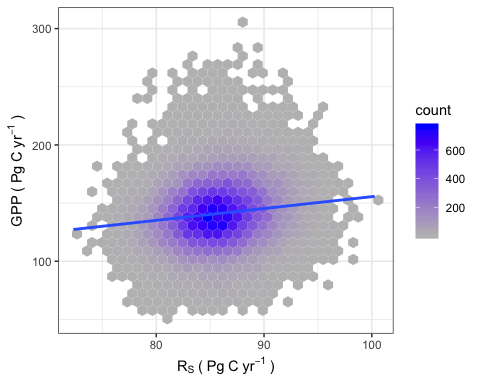
# senario 2, RC seperate into 4 groups, and Froot seperate into 2 groups  
var\_CI <- quantile(gpp\_results$GPP2, c(0.025, 0.5, 0.975)) %>% round(0)  
var\_CI\_raw <- quantile(gpp\_results$GPP\_raw, c(0.025, 0.5, 0.975)) %>% round(0)  
ggplot(gpp\_results) + aes(GPP2) + geom\_histogram(col = "black", fill = "gray", bins = 30) +   
 geom\_histogram(aes(GPP\_raw), col = "blue", fill = "blue", bins = 30, alpha = 0.35) +  
 geom\_vline(xintercept = c(quantile (gpp\_results$GPP\_raw, c(0.025, 0.5, 0.975))),col = "red", linetype = "dotted", size = 1.5) +  
 annotate("text", x = quantile(gpp\_results$GPP2, 0.98), y = 9000, label = paste0("95%CI (", var\_CI[1],", ", var\_CI[2], ", ", var\_CI[3], ")"), hjust = 0) +  
 annotate("text", x = quantile(gpp\_results$GPP2, 0.98), y = 7000, label = paste0("95%CI (", var\_CI\_raw[1],", ", var\_CI\_raw[2], ", ", var\_CI\_raw[3], ")")  
 , hjust = 0, col = "blue") +  
 xlab(expression(GPP~"("~Pg~C~yr^{-1}~")"))



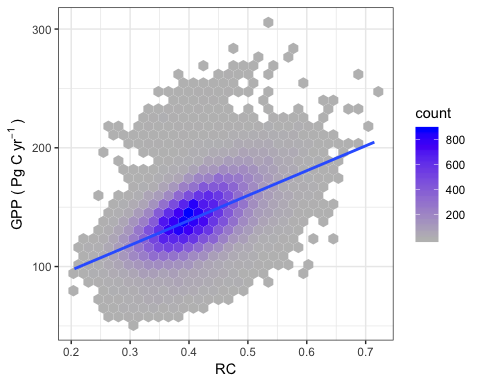
# GPP ~ NPP  
ggplot(gpp\_results) +  
 aes(x = NPP, y = GPP) +  
 geom\_hex() +  
 scale\_fill\_gradient(low = "gray", high = "blue") +  
 geom\_smooth(method = "lm") +   
 xlab(expression(NPP~"("~Pg~C~yr^{-1}~")")) +  
 ylab(expression(GPP~"("~Pg~C~yr^{-1}~")"))



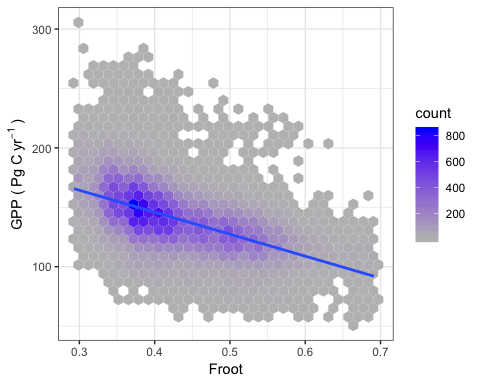
# GPP ~ Rs  
ggplot(gpp\_results) +  
 aes(x = Rs, y = GPP) +  
 geom\_hex() +  
 scale\_fill\_gradient(low = "gray", high = "blue") +  
 geom\_smooth(method = "lm") +  
 xlab(expression(R[S]~"("~Pg~C~yr^{-1}~")")) +  
 ylab(expression(GPP~"("~Pg~C~yr^{-1}~")"))



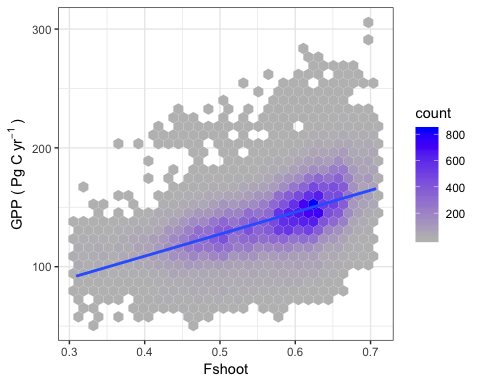
# GPP ~ Rc  
ggplot(gpp\_results) +  
 aes(x = Rc, y = GPP) +  
 geom\_hex() +  
 scale\_fill\_gradient(low = "gray", high = "blue") +  
 geom\_smooth(method = "lm") +  
 xlab(expression(RC)) +  
 ylab(expression(GPP~"("~Pg~C~yr^{-1}~")"))



# GPP ~ Froot  
ggplot(gpp\_results) +  
 aes(x = Froot, y = GPP) +  
 geom\_hex() +  
 scale\_fill\_gradient(low = "gray", high = "blue") +  
 geom\_smooth(method = "lm") +  
 xlab(expression(Froot)) +  
 ylab(expression(GPP~"("~Pg~C~yr^{-1}~")"))



# GPP ~ Fshoot  
ggplot(gpp\_results) +  
 aes(x = 1-Froot, y = GPP) +  
 geom\_hex() +  
 scale\_fill\_gradient(low = "gray", high = "blue") +  
 geom\_smooth(method = "lm") +  
 xlab(expression(Fshoot)) +  
 ylab(expression(GPP~"("~Pg~C~yr^{-1}~")"))



# Quantify the probability GPP following (107, 133)  
# about 29% chance GPP will between (107-133)  
y <- mean(gpp\_results$GPP)  
s <- sd(gpp\_results$GPP)  
# accume GPP following normal distribution, with mean = y and sd = s, we can calculate the properbility of GPP following (107-133) 95%CI  
p\_gpp\_less\_than\_133 <- pnorm ((133 - y) / (s))  
p\_gpp\_less\_than\_107 <- pnorm ((107 - y) / (s))  
p\_gpp <- p\_gpp\_less\_than\_133 -p\_gpp\_less\_than\_107  
p\_gpp

## [1] 0.2925003

# pnorm (-0.329)  
  
# senario 2 looks like a little bit right tail skew  
# about 18%  
y <- mean(gpp\_results$GPP2)  
s <- sd(gpp\_results$GPP2)  
p\_gpp\_less\_than\_133 <- pnorm ((133 - y) / (s))  
p\_gpp\_less\_than\_107 <- pnorm ((107 - y) / (s))  
p\_gpp <- p\_gpp\_less\_than\_133 -p\_gpp\_less\_than\_107  
p\_gpp

## [1] 0.1800701

\*\* GPP plot

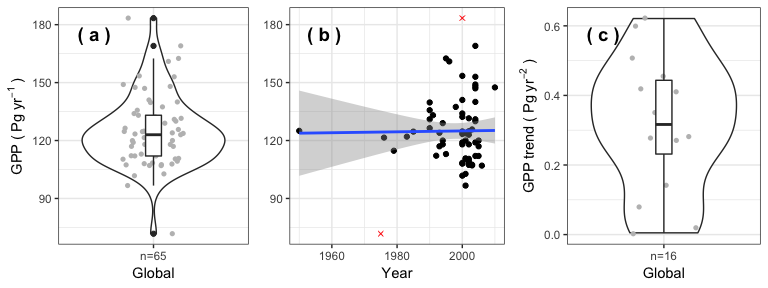
# panel (a). all 65 GPP estimates  
# panel (b). GPP vs year, the red dots in panel c are calculated trend from panel b (increase rate = 0.2 if outliers removed, increase rate = 0.42 if outlers kept)  
# panel (c). 16 GPP estimates reported GPP increase trend  
plot\_GPP(GPP)

## [1] "median=123"  
## [1] "se=2.29621351780485"  
## [1] "95% CI=4.587"  
## [1] "number of trend=16"  
##   
## Call:  
## lm(formula = GPP ~ Year, data = subGPP)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -28.327 -12.859 -2.018 7.810 43.935   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 77.9609 452.1737 0.172 0.864  
## Year 0.0235 0.2263 0.104 0.918  
##   
## Residual standard error: 16.03 on 61 degrees of freedom  
## Multiple R-squared: 0.0001768, Adjusted R-squared: -0.01621   
## F-statistic: 0.01079 on 1 and 61 DF, p-value: 0.9176  
##   
##   
## Call:  
## lm(formula = GPP ~ Year, data = GPP)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -47.315 -14.685 -2.175 9.840 57.759   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -401.2559 496.2209 -0.809 0.422  
## Year 0.2634 0.2484 1.061 0.293  
##   
## Residual standard error: 18.49 on 63 degrees of freedom  
## Multiple R-squared: 0.01754, Adjusted R-squared: 0.001946   
## F-statistic: 1.125 on 1 and 63 DF, p-value: 0.2929

## Warning: Removed 49 rows containing non-finite values (stat\_ydensity).

## Warning: Removed 49 rows containing non-finite values (stat\_boxplot).

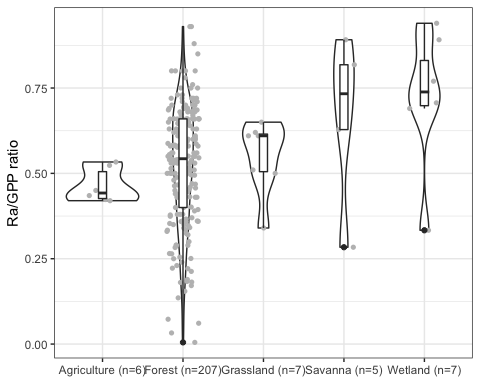
## Warning: Removed 49 rows containing missing values (geom\_point).



\*\* Plot Ra to GPP ratio

# only one point from tundra  
var\_RaGpp <- plot\_RaGPP (RaGPP, srdb\_v4)

## [1] 0.5303451  
## [1] "Ra/GPP se=0.0117196199746696"  
## [1] "Ra/GPP 95% CI=0.023"  
## [1] "Ra/GPP obs=232"



var\_RaGpp %>% group\_by (Ecosystem) %>% summarise ( count = n() ) %>% mutate (percent = count/sum(count)) %>% arrange (desc(Ecosystem))

## # A tibble: 5 x 3  
## Ecosystem count percent  
## <fct> <int> <dbl>  
## 1 Savanna 5 0.0216  
## 2 Agriculture 6 0.0259  
## 3 Wetland 7 0.0302  
## 4 Grassland 7 0.0302  
## 5 Forest 207 0.892

# topdowm approach to eatimate Rs  
# Ra = GPP \* Ra\_GPP\_ratio (calculated based on data from a summary paper and from srdb, see plot\_RaGPP function)  
# Ra = GPP - NPP (two approaches yield a very similar results, we thus used the average of two estimates)  
  
var\_Fire <- c(2, 3.5, 7.3, 4, 5.1, 2.02, 2.71, 3.02, 2.08)  
n\_samp <- 50000  
  
set.seed(1234567)  
rs\_results <- tibble(  
 # GPP = replicate(n\_rep, sample(GPP$GPP, n\_samp, replace = TRUE) %>% mean() ),  
 # GPP = sample(GPP$GPP, n\_samp, replace = TRUE),  
 gpp1 = rnorm(n = n\_samp, mean = 124.6, sd = 2.7),  
 gpp2 = rnorm(n = n\_samp, mean = 135.7, sd = 21.712),  
 gpp3 = rnorm(n = n\_samp, mean = 109.29, sd = 27.33),  
 gpp4 = rnorm(n = n\_samp, mean = 118, sd = 26),  
 gpp5 = rnorm(n = n\_samp, mean = 119, sd = 6),  
 gpp6 = rnorm(n = n\_samp, mean = 110.5, sd = 21.3),  
 gpp7 = rnorm(n = n\_samp, mean = 117, sd = 13),  
 gpp8 = sample(GPP[is.na(GPP$SD),]$GPP, n\_samp, replace = TRUE),  
 GPP = (gpp1 + gpp2 + gpp3 + gpp4 + gpp5 + gpp6 + gpp7 + gpp8)/8,  
   
 NPP = sample(NPP$NPP, n\_samp, replace = TRUE),  
 HerbComsum = rnorm(n = n\_samp, mean = 2.2, sd = 0.2),  
 Fire = sample(var\_Fire, n\_samp, replace = TRUE),  
 sink = rnorm(n = n\_samp, mean = 2.10, sd = 0.28),  
   
 # RaGpp = sample(var\_RaGpp$RaGPP\_ratio, n\_samp, replace = TRUE),  
 RaGpp\_agr = sample(var\_RaGpp[var\_RaGpp$Ecosystem == "Agriculture", ]$RaGPP\_ratio, n\_samp, replace = TRUE),  
 RaGpp\_for = sample(var\_RaGpp[var\_RaGpp$Ecosystem == "Forest", ]$RaGPP\_ratio, n\_samp, replace = TRUE),  
 RaGpp\_gra = sample(var\_RaGpp[var\_RaGpp$Ecosystem == "Grassland", ]$RaGPP\_ratio, n\_samp, replace = TRUE),  
 RaGpp\_sav = sample(var\_RaGpp[var\_RaGpp$Ecosystem == "Savanna", ]$RaGPP\_ratio, n\_samp, replace = TRUE),  
 RaGpp\_wet = sample(var\_RaGpp[var\_RaGpp$Ecosystem == "Wetland", ]$RaGPP\_ratio, n\_samp, replace = TRUE),  
 RaGpp\_rest = sample(var\_RaGpp$RaGPP\_ratio, n\_samp, replace = TRUE),  
   
 # senario 1  
 RaGpp = RaGpp\_agr\*0.267 + RaGpp\_for\*0.298 + RaGpp\_gra\*0.213 + RaGpp\_sav\*0.216 + RaGpp\_wet\*0.006,  
 # senario 2  
 RaGpp2 = RaGpp\_for\*0.298 + RaGpp\_rest\*(1-0.298),  
   
 # Froot = sample(Froot$Froot, n\_samp, replace = TRUE), # senario 0: Froot not seperate into different ecosystems, not used  
 Froot\_agr = sample(Froot[Froot$Ecosystem == "Agriculture",]$Froot, n\_samp, replace = TRUE),  
 Froot\_for = sample(Froot[Froot$Ecosystem == "Forest",]$Froot, n\_samp, replace = TRUE),  
 Froot\_gra = sample(Froot[Froot$Ecosystem == "Grassland",]$Froot, n\_samp, replace = TRUE),  
 Froot\_sav = sample(Froot[Froot$Ecosystem == "Savanna",]$Froot, n\_samp, replace = TRUE),  
 Froot\_wet = sample(Froot[Froot$Ecosystem == "Wetland",]$Froot, n\_samp, replace = TRUE),  
 Froot\_rest = sample(Froot$Froot, n\_samp, replace = TRUE),  
 Froot = Froot\_agr\*0.267 + Froot\_for\*0.298 + Froot\_gra\*0.213 + Froot\_sav\*0.216 + Froot\_wet\*0.006,  
 # seperate into 2 groupts  
 Froot2 = Froot\_rest \* 0.702 + Froot\_for\*0.298,  
   
 # Rs\_raw = sample(na.omit(GlobalRs$Rs), n\_samp, replace = TRUE) ,  
 Rs2 = rnorm(n = n\_samp, mean = 80.4, sd = 16.9),  
 Rs3 = rnorm(n = n\_samp, mean = 98, sd = 12),  
 Rs4 = rnorm(n = n\_samp, mean = 79, sd = 7.65),  
 Rs5 = rnorm(n = n\_samp, mean = 94.4, sd = 4.59),  
 Rs6 = rnorm(n = n\_samp, mean = 91, sd = 2.05),  
 Rs7 = rnorm(n = n\_samp, mean = 94.3, sd = 9.13),  
 Rs8 = rnorm(n = n\_samp, mean = 78.34, sd = 2.25),  
 Rs9 = rnorm(n = n\_samp, mean = 72.55, sd = 7.13),  
 Rs10 = sample(var\_Rs$Rs, n\_samp, replace = TRUE),  
 Rs\_raw = (Rs2 + Rs3 + Rs4 + Rs5 + Rs6 + Rs7 + Rs8 + Rs9 + Rs10)/9,  
   
 Ra1 = GPP \* RaGpp, # first way to calculate Ra, Ra = GPP \* RaGPP  
 Ra2 = GPP - NPP, # second way to calculate Ra, Ra = GPP - NPP  
 # BBL: what's this comma? Why doesn't it cause an error?  
 # response: because few Ra2 < 0 (when GPP < NPP), for those estimates, just drop them? or use Ra1 replace them? or keep them?  
 Ra3 = ifelse(Ra2 < 0, Ra1, Ra2),   
 Ra\_avg1 = (Ra1 + Ra3) / 2,  
   
 Rroot = Ra\_avg1 \* Froot,  
 Rshoot = Ra\_avg1 \* (1 - Froot),  
 Rs\_topdown = NPP - HerbComsum - Fire - sink + Rroot,   
   
 # senario2  
 Ra4 = GPP \* RaGpp2, # first way to calculate Ra, Ra = GPP \* RaGPP  
 Ra\_avg2 = (Ra4 + Ra3) / 2,  
 Rroot2 = Ra\_avg2 \* Froot2,  
 Rshoot2 = Ra\_avg2 \* (1 - Froot2),  
 Rs\_topdown2 = NPP - HerbComsum - Fire - sink + Rroot2,   
   
)

rs\_results$Ra2 %>% min(na.rm = TRUE)

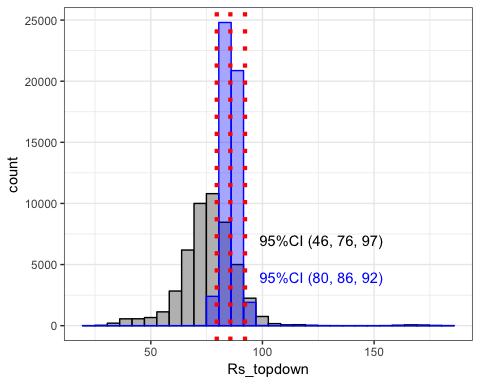
## [1] -44.95633

rs\_results$Ra3 %>% min(na.rm = TRUE)

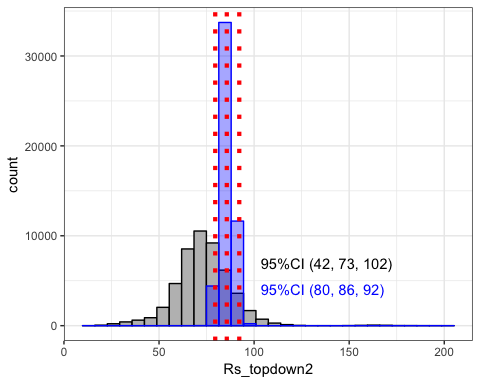
## [1] 0.3579088

### Topdow approach results diagnose

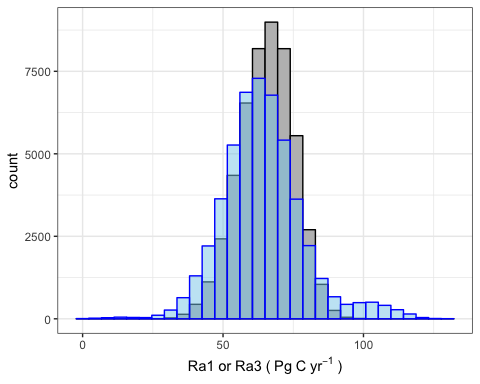
# mean(rs\_results$Rs)  
# max(rs\_results$Rs)  
  
var\_CI <- quantile(rs\_results$Rs\_topdown, c(0.025, 0.5, 0.975)) %>% round(0)  
var\_CI\_raw <- quantile(rs\_results$Rs\_raw, c(0.025, 0.5, 0.975)) %>% round(0)  
ggplot(rs\_results) + aes(Rs\_topdown) + geom\_histogram(col = "black", fill = "gray", bins = 30) +   
 geom\_histogram(aes(Rs\_raw), col = "blue", fill = "blue", bins = 30, alpha = 0.35) +  
 geom\_vline(xintercept = c(quantile (rs\_results$Rs\_raw, c(0.025, 0.5, 0.975))),col = "red", linetype = "dotted", size = 1.5) +  
 annotate("text", x = quantile(rs\_results$Rs\_topdown, 0.98), y = 7000, label = paste0("95%CI (", var\_CI[1],", ", var\_CI[2], ", ", var\_CI[3], ")"), hjust = 0) +  
 annotate("text", x = quantile(rs\_results$Rs\_topdown, 0.98), y = 4000, label = paste0("95%CI (", var\_CI\_raw[1],", ", var\_CI\_raw[2], ", ", var\_CI\_raw[3], ")")  
 , hjust = 0, col = "blue") +  
 ylab(expression(count))



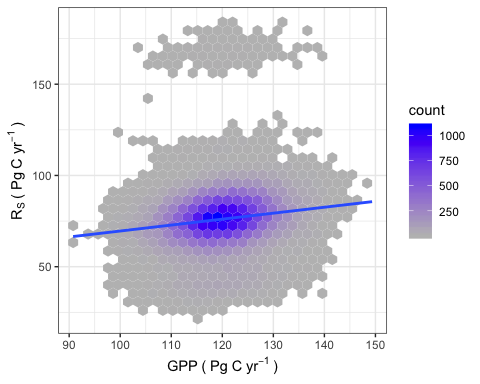
# senario two  
var\_CI <- quantile(rs\_results$Rs\_topdown2, c(0.025, 0.5, 0.975)) %>% round(0)  
var\_CI\_raw <- quantile(rs\_results$Rs\_raw, c(0.025, 0.5, 0.975)) %>% round(0)  
ggplot(rs\_results) + aes(Rs\_topdown2) + geom\_histogram(col = "black", fill = "gray", bins = 30) +   
 geom\_histogram(aes(Rs\_raw), col = "blue", fill = "blue", bins = 30, alpha = 0.35) +  
 geom\_vline(xintercept = c(quantile (rs\_results$Rs\_raw, c(0.025, 0.5, 0.975))),col = "red", linetype = "dotted", size = 1.5) +  
 annotate("text", x = quantile(rs\_results$Rs\_topdown2, 0.98), y = 7000, label = paste0("95%CI (", var\_CI[1],", ", var\_CI[2], ", ", var\_CI[3], ")"), hjust = 0) +  
 annotate("text", x = quantile(rs\_results$Rs\_topdown2, 0.98), y = 4000, label = paste0("95%CI (", var\_CI\_raw[1],", ", var\_CI\_raw[2], ", ", var\_CI\_raw[3], ")")  
 , hjust = 0, col = "blue") +  
 ylab(expression(count))



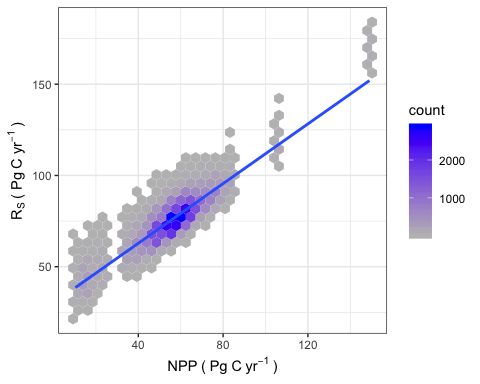
# Rs raw data plot  
# var\_CI <- quantile(rs\_results$Rs\_raw, c(0.025, 0.5, 0.975)) %>% round(0)  
# plot\_rs\_raw <- ggplot(rs\_results) + aes(Rs\_raw) + geom\_histogram(col = "black", fill = "gray", bins = 30) +   
# geom\_vline(xintercept = c(quantile (rs\_results$Rs, c(0.025, 0.5, 0.975))),col = "red", linetype = "dotted", size = 1.5) +  
# annotate("text", x = quantile(rs\_results$Rs, 0.98), y = 5000, label = paste0("95%CI (", var\_CI[1],", ", var\_CI[2], ", ", var\_CI[3], ")"), hjust = 0)  
  
# Ra1 ~ Ra2  
ggplot(rs\_results) +  
 aes(x = Ra1) + geom\_histogram(col = "black", fill = "gray", bins = 30) +  
 geom\_histogram(aes(Ra3), col = "blue", fill = "skyblue", bins = 30, alpha = 0.5) +  
 xlab(expression(Ra1~or~Ra3~"("~Pg~C~yr^{-1}~")")) +  
 ylab(expression(count))



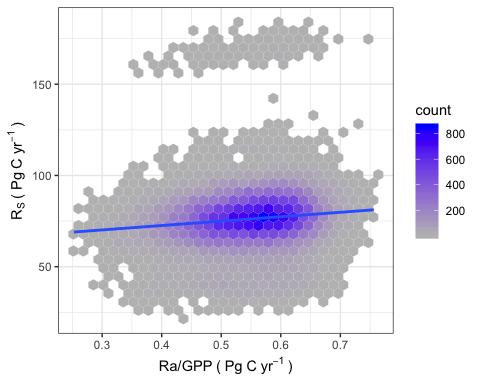
# Rs ~ GPP  
ggplot(rs\_results) +  
 aes(x = GPP, y = Rs\_topdown) +  
 geom\_hex() +  
 scale\_fill\_gradient(low = "gray", high = "blue") +  
 geom\_smooth(method = "lm") +  
 xlab(expression(GPP~"("~Pg~C~yr^{-1}~")")) +  
 ylab(expression(R[S]~"("~Pg~C~yr^{-1}~")"))



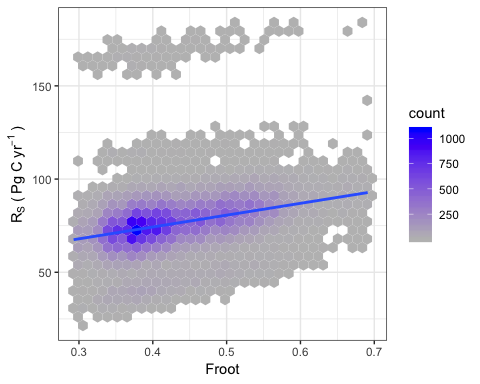
# Rs ~ NPP  
ggplot(rs\_results) +  
 aes(x = NPP, y = Rs\_topdown) +  
 geom\_hex() +  
 scale\_fill\_gradient(low = "gray", high = "blue") +  
 geom\_smooth(method = "lm") +  
 xlab(expression(NPP~"("~Pg~C~yr^{-1}~")")) +  
 ylab(expression(R[S]~"("~Pg~C~yr^{-1}~")"))



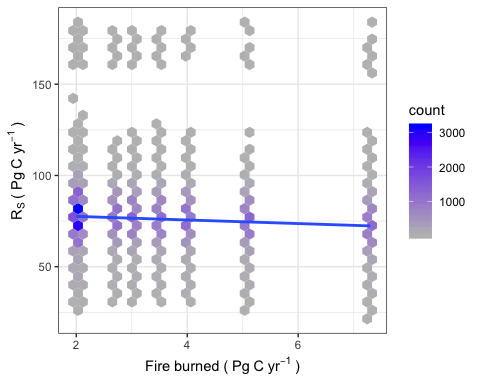
# Rs ~ RaGpp  
ggplot(rs\_results) +  
 aes(x = RaGpp, y = Rs\_topdown) +  
 geom\_hex() +  
 scale\_fill\_gradient(low = "gray", high = "blue") +  
 geom\_smooth(method = "lm") +  
 xlab(expression("Ra/GPP"~"("~Pg~C~yr^{-1}~")")) +  
 ylab(expression(R[S]~"("~Pg~C~yr^{-1}~")"))



# Rs ~ Froot  
ggplot(rs\_results) +  
 aes(x = Froot, y = Rs\_topdown) +  
 geom\_hex() +  
 scale\_fill\_gradient(low = "gray", high = "blue") +  
 geom\_smooth(method = "lm") +  
 xlab(expression(Froot)) +  
 ylab(expression(R[S]~"("~Pg~C~yr^{-1}~")"))



# Rs ~ Fire  
ggplot(rs\_results) +  
 aes(x = Fire, y = Rs\_topdown) +  
 geom\_hex() +  
 scale\_fill\_gradient(low = "gray", high = "blue") +  
 geom\_smooth(method = "lm") +  
 xlab(expression(Fire~burned~"("~Pg~C~yr^{-1}~")")) +  
 ylab(expression(R[S]~"("~Pg~C~yr^{-1}~")"))



# accume Rs following normal distribution, with mean = y and sd = s, we can calculate the properbility of Rs following (80-92) 95%CI  
# about 28% chance Rs will be between (80-92)  
y <- mean(rs\_results$Rs\_topdown)  
s <- sd(rs\_results$Rs\_topdown)  
p\_Rs\_higher\_than\_80 <- pnorm ((y - 80)/(s))  
p\_Rs\_higher\_than\_92 <- pnorm ((y - 92)/(s))  
p\_Rs <- p\_Rs\_higher\_than\_80 -p\_Rs\_higher\_than\_92  
p\_Rs

## [1] 0.2706582

# Senario 2: about 23% chance Rs will be between (80-92)  
y <- mean(rs\_results$Rs\_topdown2)  
s <- sd(rs\_results$Rs\_topdown2)  
p\_Rs\_higher\_than\_80 <- pnorm ((y - 80)/(s))  
p\_Rs\_higher\_than\_92 <- pnorm ((y - 92)/(s))  
p\_Rs <- p\_Rs\_higher\_than\_80 -p\_Rs\_higher\_than\_92  
p\_Rs

## [1] 0.2216852