Biomass sampling

Chaney Hart

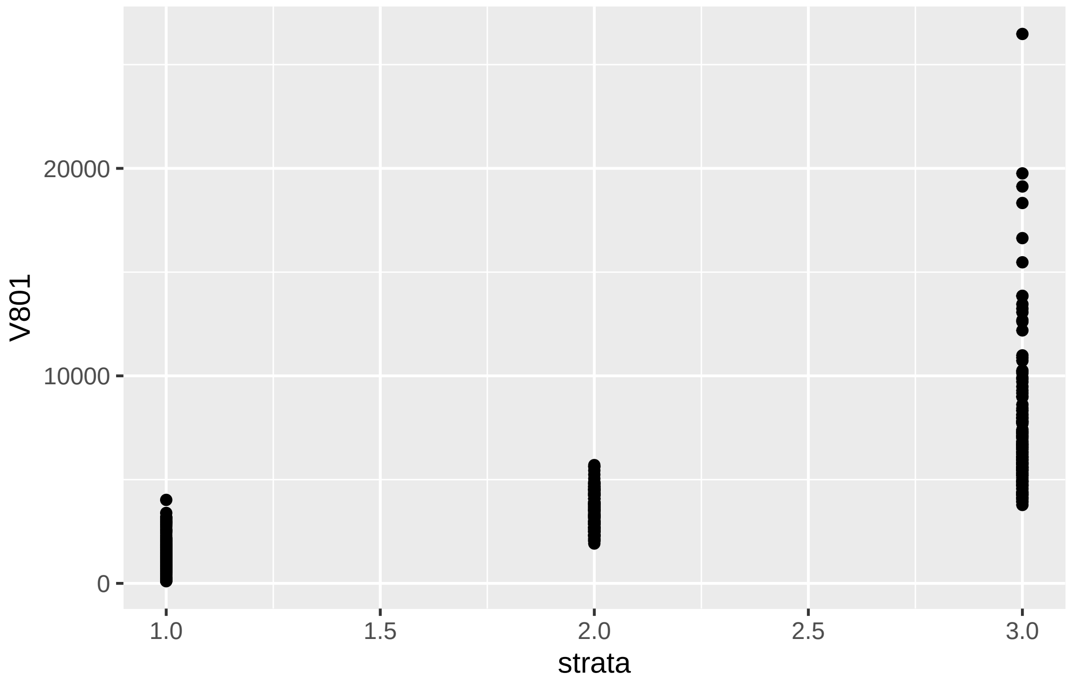
2023-11-07

read in data

Split trees into terciles based on measured volume index

```
biomass_set <- growth_dat %>% group_by(event_short) %>% mutate(
  strata = ntile(V801,3))

tercile_plot <- ggplot(biomass_set,aes(x=strata,y=V801))+
  geom_point()
tercile_plot + theme(plot.margin = unit(c(0.5,0.5,0.5,0.5),"inches"))
```



We see that the top tercile is much more variable, suggesting that simple random sampling may not be the most efficient and allocating samples equally to each tercile may not be the most efficient.

Allocating samples to each strata

number of samples determined by size of strata and variance of each strata

```
biomass_subset <- subset(biomass_set, event_short == "13-15E" | event_short == "2H" | event_short == "5A

# add info about the total number of trees in each event (N)
biomass_subset <- biomass_subset %>% mutate(N= case_when(
  event_short == "13-15E"~"32",
  event_short == "16-20"~"33",
  event_short == "5A"~"34",
  event_short == "4A"~"19",
  event_short == "2H"~ "29",
  event_short == "5C"~ "33",
  event_short == "8-9D"~"29",
  event_short == "CT3"~"34"
))

biomass_subset$N <- as.numeric(biomass_subset$N)
#for each event, we want 50% of the tree samples
biomass_subset$n_final <- biomass_subset$N*0.5

#brute force the denominator of Neyman's allocation formula for each event.
#this is calculated as the sum of standard deviation times the fractional size of each strata
biomass_subset <- biomass_subset %>% mutate(denom= case_when(
  event_short == "13-15E"~1159.79,
  event_short == "16-20"~1429.84,
  event_short == "5A"~2860.11,
  event_short == "4A"~2170.89,
  event_short == "2H"~ 846.708,
  event_short == "5C"~ 1461.4,
  event_short == "8-9D"~1073.33,
  event_short == "CT3"~724.72
))

#determine size and variance within each strata of each event. Use this to estimate how samples should
#larger and/or more variable strata get more samples
biomass_summary <- biomass_subset %>% group_by(event_short,strata) %>% summarize(
  ID = ID,
  V = V801,
  n_strata = n(),
  N = N,
  w_strata = n_strata/N,
  sd = sd(V801),
  n_neymans = (n_final)*((w_strata*sd)/denom),
  n_equal = n_final/3,
  n_prop = n_final*w_strata,
  tier = tier,
  block = block,
  construct = construct,
  construct2 = construct2
)
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## `summarise()` has grouped output by 'event_short', 'strata'. You can override
## using the `.groups` argument.
```

```
print(biomass_summary)
```

```
## # A tibble: 243 x 15
## # Groups:   event_short, strata [24]
##    event_short strata ID           V n_strata     N w_strata    sd n_neymans
##    <chr>        <int> <chr>    <dbl>    <int> <dbl>    <dbl> <dbl>     <dbl>
##  1 13-15E           1 LCOR-607  827.       11    32    0.344  554.      2.63
##  2 13-15E           1 LCOR-075  742.       11    32    0.344  554.      2.63
##  3 13-15E           1 LCOR-614 1248.       11    32    0.344  554.      2.63
##  4 13-15E           1 LCOR-080 1074.       11    32    0.344  554.      2.63
##  5 13-15E           1 LCOR-606 1735.       11    32    0.344  554.      2.63
##  6 13-15E           1 LCOR-605 1765.       11    32    0.344  554.      2.63
##  7 13-15E           1 LCOR-083 1862.       11    32    0.344  554.      2.63
##  8 13-15E           1 LCOR-297  584.       11    32    0.344  554.      2.63
##  9 13-15E           1 LCOR-304  123.       11    32    0.344  554.      2.63
## 10 13-15E           1 LCOR-615 1231.       11    32    0.344  554.      2.63
## # i 233 more rows
## # i 6 more variables: n_equal <dbl>, n_prop <dbl>, tier <chr>, block <chr>,
## #   construct <chr>, construct2 <chr>
```

The above chunk guides how many samples should be allocated to each strat

Monte Carlo simulation of sampling

Compared simple random sampling with stratified random sampling.

In stratified random sampling, allocated snumber of sampled to each strata based on variance in each strata.

```
library(nlme)
```

```
##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse
```

```r
#single event datasets

biomass_5A <- subset(biomass_subset, event_short == "5A")
biomass_4A <- subset(biomass_subset, event_short == "4A")
biomass_13_15E <- subset(biomass_subset, event_short == "13-15E")
biomass_16_20 <- subset(biomass_subset, event_short == "16-20")
biomass_8_9D <- subset(biomass_subset, event_short == "8-9D")
biomass_CT3 <- subset(biomass_subset, event_short == "CT3")


##################################################################################

#with a dataset for a single event...
#this function runs the simulation. Takes a random sample and a stratified random sample with the numbe
#runs this 1000 times
#compares the estimate of volume index you get from each sample to the observed value we have measured 
mcs_biomass <- function(data, colnames=c("ID","strata", "V801"), n,nh, reps=1000) {
  ## Match up column names
  colnames <- match(colnames, names(data))
  ID <-  as.matrix(data[,colnames[1]])  ## ID
  strata <- as.matrix(data[,colnames[2]])   ## Strata
  Volume <- as.matrix(data[,colnames[3]])   ## Volume


  ## Calculate population values
  N = length(Volume)
  SV <- mean(Volume)   ## mean biomass
  SD <- sd(Volume)

  ## Create initial values
  V <- matrix(nrow=reps, ncol=2)
  B <- double(2)
  MSE <- double(2)

  ## Run Monte Carlo simulation
  for(i in 1:reps)
  {
    ## SRS
    samp <- sample(Volume, n)
    V[i,1] <- (sum(samp/(n/N)))/N
    B[1] <- B[1] + (V[i,1] - SV)
    MSE[1] <- MSE[1] + (V[i,1] - SV)^2

    ## STRS (neymans)
    strt <- levels(factor(strata))
    Nh <- gapply(data.frame(strata), FUN=function(x) nrow(x), form=~strata) #Population total per strat

    V[i,2] <- 0
    for(j in 1:length(Nh))
    {
      samp <- sample(1:Nh[j], nh[j])
      V[i,2] <- V[i,2] + (sum((Volume[strata==strt[j]][samp])/(nh[j]/Nh[j])))/N
    }
    B[2] <- B[2] + (V[i,2] - SV)
    MSE[2] <- MSE[2] + (V[i,2] - SV)^2
  }
  B <- B/reps
  MSE <- MSE/reps
  RMSE <- sqrt(MSE)

  #create return list
  lst <- list(n=n, reps=reps,Bias=B, MSE=MSE, RMSE=RMSE, Vhat=V, Vobs=SV)
  return(lst)

}

#run simulation for each event of interest
mc_5A <- mcs_biomass(biomass_5A,n =17,nh=c(3,3,11),reps = 1000)
mc_4A <- mcs_biomass(biomass_4A,n =17,nh=c(2,2,6),reps = 1000)
mc_16_20 <- mcs_biomass(biomass_16_20,n =17,nh=c(2,3,11),reps = 1000)
mc_8_9D <- mcs_biomass(biomass_8_9D,n =17,nh=c(4,2,8),reps = 1000)
mc_CT3 <- mcs_biomass(biomass_CT3,n =17,nh=c(4,4,9),reps = 1000)
mc_13_15E <- mcs_biomass(biomass_13_15E,n =17,nh=c(3,3,10),reps = 1000)

#compile results into a dataframe to plot results
bias_SRS <- as.matrix(list(mc_5A[["Bias"]][1],mc_4A[["Bias"]][1],mc_16_20[["Bias"]][1],mc_8_9D[["Bias"]
bias_STRS <- as.matrix(list(mc_5A[["Bias"]][2],mc_4A[["Bias"]][2],mc_16_20[["Bias"]][2],mc_8_9D[["Bias"
MSE_SRS <- as.matrix(list(mc_5A[["MSE"]][1],mc_4A[["MSE"]][1],mc_16_20[["MSE"]][1],mc_8_9D[["MSE"]][1],
MSE_STRS <- as.matrix(list(mc_5A[["MSE"]][2],mc_4A[["MSE"]][2],mc_16_20[["MSE"]][2],mc_8_9D[["MSE"]][2]
RMSE_SRS <- as.matrix(list(mc_5A[["RMSE"]][1],mc_4A[["RMSE"]][1],mc_16_20[["RMSE"]][1],mc_8_9D[["RMSE"]
RMSE_STRS <- as.matrix(list(mc_5A[["RMSE"]][2],mc_4A[["RMSE"]][2],mc_16_20[["RMSE"]][2],mc_8_9D[["RMSE"


sample_comp <- as.data.frame(cbind(bias_SRS,bias_STRS,MSE_SRS,MSE_STRS,RMSE_SRS,RMSE_STRS))
sample_comp$event <- c("5A","4A","16-20","8-9D","CT3","13-15E")
colnames(sample_comp) <- c("bias_SRS","bias_STRS","MSE_SRS","MSE_STRS","RMSE_SRS","RMSE_STRS","event")

sample_comp_bias <- pivot_longer(sample_comp,cols = c("bias_SRS","bias_STRS"), names_to = "sampling_des
sample_comp_bias <- sample_comp_bias[,c(5:7)]
sample_comp_bias$bias <- as.matrix(unlist(sample_comp_bias$bias))
sample_comp_bias$sampling_design <- substr(sample_comp_bias$sampling_design,6,9)
str(sample_comp_bias)
```

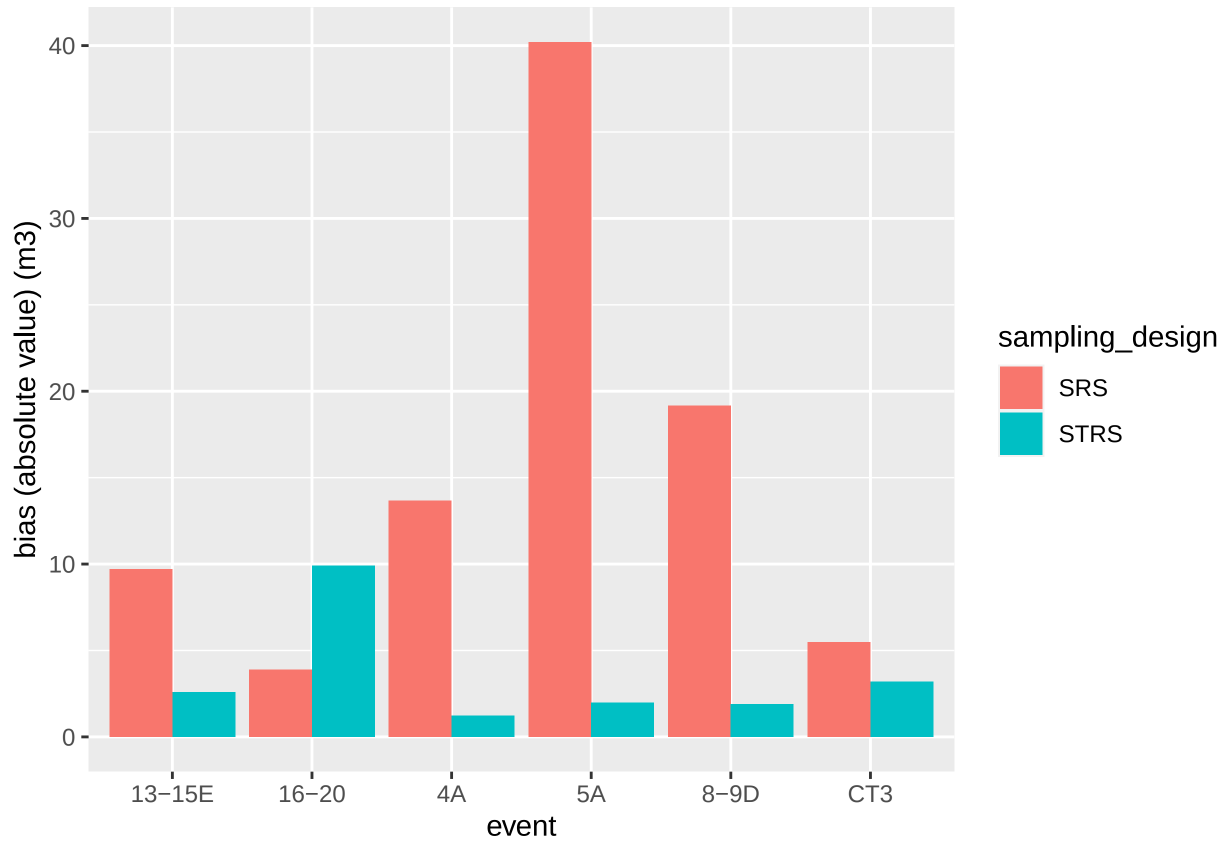
```
## tibble [12 x 3] (S3: tbl_df/tbl/data.frame)
##  $ event          : chr [1:12] "5A" "5A" "4A" "4A" ...
##  $ sampling_design: chr [1:12] "SRS" "STRS" "SRS" "STRS" ...
##  $ bias           : num [1:12, 1] 40.2 -1.97 13.68 1.24 3.88 ...
```

```
sample_comp_MSE <- pivot_longer(sample_comp,cols = c("MSE_SRS","MSE_STRS"), names_to = "sampling_design
sample_comp_MSE <- sample_comp_MSE[,c(5:7)]
sample_comp_MSE$MSE <- as.matrix(unlist(sample_comp_MSE$MSE))
sample_comp_MSE$sampling_design <- substr(sample_comp_MSE$sampling_design,5,9)

sample_comp_RMSE <- pivot_longer(sample_comp,cols = c("RMSE_SRS","RMSE_STRS"), names_to = "sampling_des
sample_comp_RMSE <- sample_comp_RMSE[,c(5:7)]
sample_comp_RMSE$RMSE <- as.matrix(unlist(sample_comp_RMSE$RMSE))
```

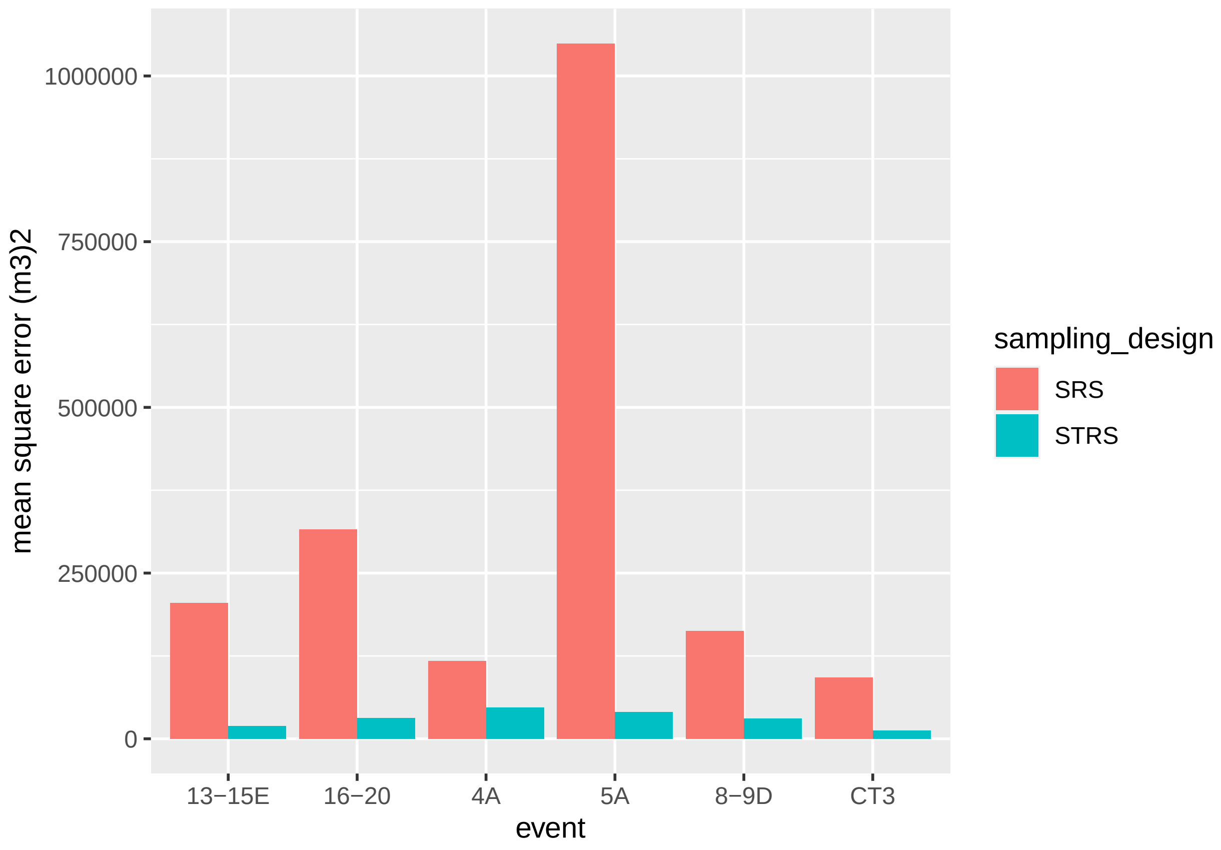
```
sample_comp_RMSE$sampling_design <- substr(sample_comp_RMSE$sampling_design,6,9)

# plot bias
bias_plot <- ggplot(sample_comp_bias, aes(event,abs(bias), fill = sampling_design))+
  geom_bar(stat = "identity",position = "dodge")+
  ylab("bias (absolute value) (m3)")
bias_plot
```



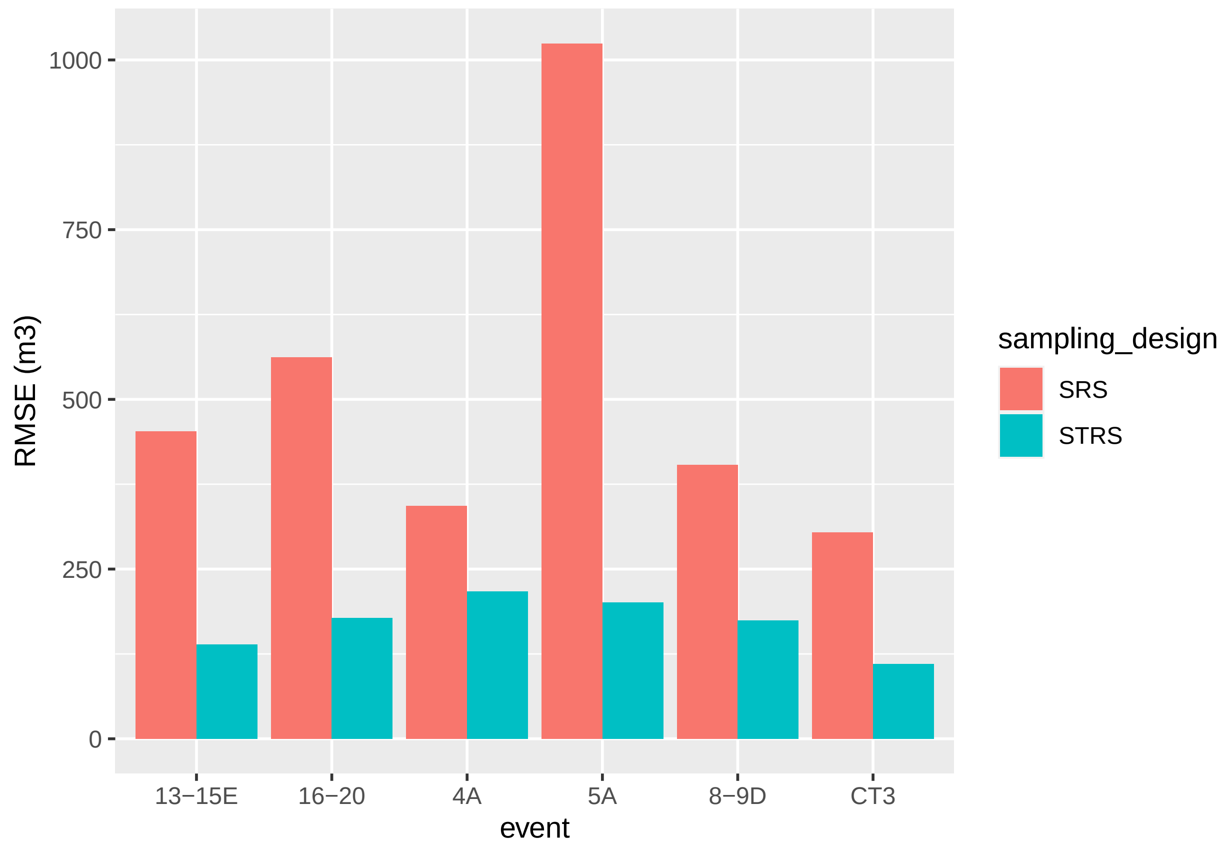
```
MSE_plot <- ggplot(sample_comp_MSE, aes(event,MSE, fill = sampling_design))+
  geom_bar(stat = "identity",position = "dodge")+
  ylab("mean square error (m3)2")

MSE_plot
```



```
RMSE_plot <- ggplot(sample_comp_RMSE, aes(event,RMSE, fill = sampling_design))+
  geom_bar(stat = "identity",position = "dodge")+
  ylab("RMSE (m3)")

RMSE_plot
```



Results of simulation

Stratified random sampling performed better in the monte carlo simulation.

Bias for both designs was small.

Accuracy (MSE and RMSE) was lesser for stratified random sampling than simple random sampling. This was especially true for the events that were most variable (5A and 4A).

Likely best to go with stratified random sampling.

Drawing a stratified random sample

```
##Take stratified sample.
#set seed
set.seed(54)
#make sure data is grouped correctly
#print(group_data(biomass_summary),n=24)

nested_neymans <- biomass_summary %>%
  group_by(event_short,strata) %>%
  nest() %>%
  ungroup() %>%
  mutate(n = c(3,3,10,2,3,11,3,4,6,2,2,6,3,3,11,2,3,11,4,2,8,4,4,9))
```

```
nested_neymans$event_short <- as.factor(nested_neymans$event_short)
nested_neymans$strata <- as.factor(nested_neymans$strata)
#print(nested_neymans,n=24)

sampled_neymans <- nested_neymans %>%
    mutate(samp = map2(data, n, sample_n))

sampled_neymans_list <- sampled_neymans %>%
  select(-data) %>%
  unnest(samp)


sampled_neymans_list <- subset(sampled_neymans_list, event_short != "2H" & event_short != "5C")
LC_meta <- read.csv("LC_2023/2023_growth_inventory_analysis/LC_9_20_growth_data_cleaned.csv")
LC_meta <- subset(LC_meta, select = c("row", "column","ID","event","event_short"))

sampled_neymans_list <- inner_join(sampled_neymans_list, LC_meta, by = c("ID","event_short"))


sample_list_priority <- subset(sampled_neymans_list, event_short == "5A" | event_short == "4A" | event_s

sample_list_priority <- subset(sample_list_priority, select = c("row","column","ID","event_short","bloc

print(sample_list_priority,n=57)
```

```
## # A tibble: 57 x 6
##      row column ID       event_short block construct
##    <int>  <int> <chr>    <chr>       <chr> <chr>
##  1     2     46 LCOR-112 16-20       small Escape
##  2     6     23 LCOR-106 16-20       large Escape
##  3     9     51 LCOR-108 16-20       small Escape
##  4    13     16 LCOR-105 16-20       large Escape
##  5    10     27 LCOR-322 16-20       large Escape
##  6    13     22 LCOR-514 16-20       large Escape
##  7    15     26 LCOR-508 16-20       large Escape
##  8     8     20 LCOR-101 16-20       large Escape
##  9    14      9 LCOR-511 16-20       large Escape
## 10    14      5 LCOR-314 16-20       large Escape
## 11     5     47 LCOR-315 16-20       small Escape
## 12    11     15 LCOR-507 16-20       large Escape
## 13    13     46 LCOR-102 16-20       small Escape
## 14     9     13 LCOR-320 16-20       large Escape
## 15     7     12 LCOR-111 16-20       large Escape
## 16     3     48 LCOR-505 16-20       small Escape
## 17    11     51 LCOR-225 4A          small LC-102
## 18    14     12 LCOR-002 4A          large LC-102
## 19    13     20 LCOR-534 4A          large LC-102
## 20     4     47 LCOR-232 4A          small LC-102
## 21     3      9 LCOR-001 4A          large LC-102
## 22     9     11 LCOR-233 4A          large LC-102
## 23    11      9 LCOR-231 4A          large LC-102
## 24    13     14 LCOR-014 4A          large LC-102
## 25    10     24 LCOR-546 4A          large LC-102
## 26     7      6 LCOR-003 4A          large LC-102
## 27     4      4 LCOR-249 5A          large LC-102
## 28     4     22 LCOR-165 5A          large LC-102
## 29     3     22 LCOR-245 5A          large LC-102
## 30     2      6 LCOR-577 5A          large LC-102
## 31     9     21 LCOR-584 5A          large LC-102
## 32     2     20 LCOR-162 5A          large LC-102
## 33     2      9 LCOR-581 5A          large LC-102
## 34     8      5 LCOR-575 5A          large LC-102
## 35    14     46 LCOR-155 5A          small LC-102
## 36     8      8 LCOR-250 5A          large LC-102
## 37     8     10 LCOR-157 5A          large LC-102
## 38     8      9 LCOR-163 5A          large LC-102
## 39     6      8 LCOR-582 5A          large LC-102
## 40    12     12 LCOR-241 5A          large LC-102
## 41    12     18 LCOR-243 5A          large LC-102
## 42     2     11 LCOR-164 5A          large LC-102
## 43     7     25 LCOR-160 5A          large LC-102
## 44    11     28 LCOR-096 8-9D        large Escape
## 45    15     51 LCOR-413 8-9D        small Escape
## 46     7     53 LCOR-097 8-9D        small Escape
## 47    14     50 LCOR-417 8-9D        small Escape
## 48     2      4 LCOR-087 8-9D        large Escape
## 49    10      5 LCOR-092 8-9D        large Escape
## 50    11     23 LCOR-090 8-9D        large Escape
## 51     7     10 LCOR-420 8-9D        large Escape
## 52     6     47 LCOR-419 8-9D        small Escape
## 53     2     21 LCOR-098 8-9D        large Escape
## 54    14     23 LCOR-290 8-9D        large Escape
## 55     7     17 LCOR-411 8-9D        large Escape
## 56     2     26 LCOR-293 8-9D        large Escape
## 57     4      5 LCOR-282 8-9D        large Escape
```

```
write.csv(sample_list_priority, file = "LC_2023/2023_growth_inventory_analysis/biomass_sample_priority1

sample_list_priority2 <- subset(sampled_neymans_list, event_short == "13-15E" | event_short == "CT3")
sample_list_priority2 <- subset(sample_list_priority2, select = c("row","column","ID","event_short","bl
print(sample_list_priority2,n=33)
```

```
## # A tibble: 33 x 6
##      row column ID       event_short block construct
##    <int>  <int> <chr>    <chr>       <chr> <chr>
##  1    15     28 LCOR-083 13-15E      large LC-102
##  2    12      9 LCOR-075 13-15E      large LC-102
##  3    15     16 LCOR-614 13-15E      large LC-102
##  4     5     17 LCOR-079 13-15E      large LC-102
##  5     7     26 LCOR-613 13-15E      large LC-102
##  6     3     16 LCOR-611 13-15E      large LC-102
##  7     7      9 LCOR-076 13-15E      large LC-102
##  8    13      7 LCOR-296 13-15E      large LC-102
##  9     3      7 LCOR-073 13-15E      large LC-102
## 10     5      8 LCOR-078 13-15E      large LC-102
## 11     7     16 LCOR-610 13-15E      large LC-102
## 12     3     14 LCOR-604 13-15E      large LC-102
## 13    10      13 LCOR-077 13-15E      large LC-102
## 14    13      24 LCOR-084 13-15E      large LC-102
## 15    10      18 LCOR-306 13-15E      large LC-102
## 16    15      25 LCOR-301 13-15E      large LC-102
## 17     6      22 LCOR-340 CT3         large Control
## 18     9      26 LCOR-341 CT3         large Control
## 19     6      24 LCOR-503 CT3         large Control
## 20     7      27 LCOR-197 CT3         large Control
## 21    13      50 LCOR-210 CT3         small Control
## 22     5      22 LCOR-209 CT3         large Control
## 23    10      28 LCOR-337 CT3         large Control
## 24     5      50 LCOR-208 CT3         small Control
## 25     7      13 LCOR-338 CT3         large Control
## 26    13      23 LCOR-202 CT3         large Control
## 27     4      21 LCOR-350 CT3         large Control
## 28     9       5 LCOR-346 CT3         large Control
## 29     7      24 LCOR-339 CT3         large Control
## 30    13      17 LCOR-502 CT3         large Control
## 31     4       9 LCOR-342 CT3         large Control
## 32    15      10 LCOR-203 CT3         large Control
## 33     7      14 LCOR-347 CT3         large Control
```

```
write.csv(sample_list_priority2, file = "LC_2023/2023_growth_inventory_analysis/biomass_sample_priority
```