

# Wetland Bird Analysis

Dan McGlinn

First created on 2022-02-24. Updated on 2023-02-27

```
library(mobr)
```

```
## Warning: replacing previous import 'dplyr::filter' by 'stats::filter' when
## loading 'mobr'
```

```
## Warning: replacing previous import 'dplyr::lag' by 'stats::lag' when loading
## 'mobr'
```

```
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      intersect, setdiff, setequal, union
```

```
library(nlme)
```

```
##
```

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

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

```
##
```

```
##      collapse
```

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
```

```
## Attaching package: 'lme4'
```

```
## The following object is masked from 'package:nlme':
```

```
##
```

```
##      lmList
```

```
library(MuMIn)
```

Read in data

```
dat <- read.csv('./data/filtered_data/clean_bird_dat.csv')
comm <- read.csv('./data/filtered_data/clean_bird_comm.csv')
row.names(comm) <- comm[, 1]
comm <- comm[, -1]
```

```
#head(dat)
#head(comm)
```

```
dim(dat)
```

```
## [1] 186 122
```

```
dim(comm)
```

```
## [1] 186 55
```

#Q1: is bird diversity higher in wetlands and uplands

```
#div <- calc_biodiv(comm, dat$uni_id_date, effort = 5, extrapolate = TRUE)
```

```
dat$N <- rowSums(comm)
dat$S <- rowSums(comm > 0 )
dat$S_n <- apply(comm, 1, rarefaction, 'IBR', effort = 5, extrapolate = F,
                 quiet_mode = TRUE)
# singletons will result in rarefaction that results in 1 species which isn't useful
dat$S_n <- ifelse(dat$S_n == 1, NA, dat$S_n)
dat$S_PIE <- calc_SPIE(comm)
```

```
## Warning in calc_PIE(x, replace = replace): NA was returned because the sample
## contains one or zero individuals.
```

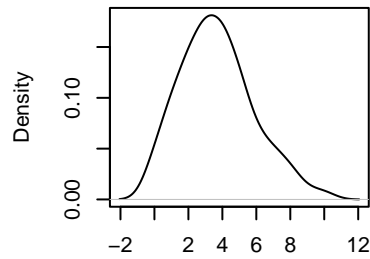
```
## Warning in calc_SPIE(comm): NA was returned because PIE = 1. This happens in
## samples where all species are singletons.
```

```
dat$S_asymp <- apply(comm, 1, calc_chao1)
```

```
par(mfrow=c(2,3))
plot(density(dat$N))
plot(density(dat$S))
plot(density(dat$S_n, na.rm = TRUE))
plot(density(dat$S_PIE, na.rm = TRUE))
plot(density(dat$S_asymp, na.rm = TRUE))
```

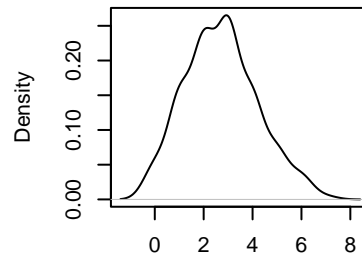
```
par(mfrow=c(2,3))
```

**density.default(x = dat\$N)**



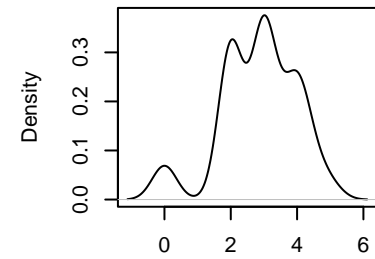
N = 186 Bandwidth = 0.6858

**density.default(x = dat\$S)**



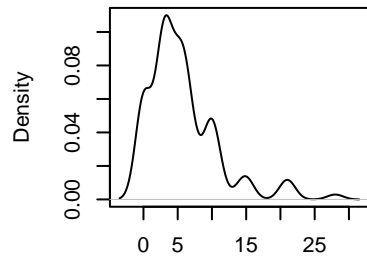
N = 186 Bandwidth = 0.4676

**density.default(x = dat\$S\_n, na.rm = TRUE)**

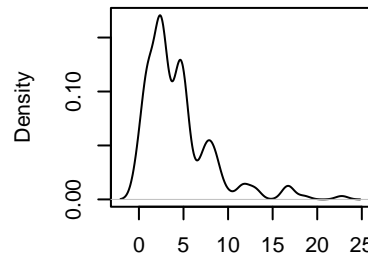


N = 156 Bandwidth = 0.3714

**density.default(x = dat\$S\_PIE, na.rm = TRUE)**



N = 117 Bandwidth = 1.166



N = 186 Bandwidth = 0.6888

```
boxplot(N ~ site_type, data = dat)
boxplot(S ~ site_type, data = dat)
boxplot(S_n ~ site_type, data = dat)
boxplot(S_PIE ~ site_type, data = dat)
boxplot(S_asymp ~ site_type, data = dat)

div_mods <- list()
div_mods$N <- glm(N ~ site_type + site + block, data = dat, family = 'poisson')
div_mods$S <- glm(S ~ site_type + site + block, data = dat, family = 'poisson')
div_mods$S_n <- glm(S_n ~ site_type + site + block, data = dat, family = 'quasipoisson')
div_mods$S_PIE <- glm(S_PIE ~ site_type + site + block, data = dat, family = 'quasipoisson')
div_mods$S_asymp <- glm(S_asymp ~ site_type + site + block, data = dat, family = 'quasipoisson')

lapply(div_mods, summary)
```

```
## $N
##
## Call:
## glm(formula = N ~ site_type + site + block, family = "poisson",
##      data = dat)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.8633  -0.8068  -0.1066   0.6317   2.8676
##
```

```
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.10352    0.12855   8.585 < 2e-16 ***
## site_typewetland 0.50385    0.11376   4.429 9.46e-06 ***
## sitestono       0.44957    0.11073   4.060 4.90e-05 ***
## block          -0.09830    0.02747  -3.578 0.000346 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 262.39  on 185  degrees of freedom
## Residual deviance: 225.10  on 182  degrees of freedom
## AIC: 778.66
##
## Number of Fisher Scoring iterations: 5
##
##
## $S
##
## Call:
## glm(formula = S ~ site_type + site + block, family = "poisson",
##      data = dat)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.4485  -0.7239   0.0014   0.4246   1.7675
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.76363    0.15100   5.057 4.26e-07 ***
## site_typewetland 0.52744    0.13363   3.947 7.91e-05 ***
## sitestono       0.47275    0.12872   3.673 0.00024 ***
## block          -0.09663    0.03204  -3.016 0.00256 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 170.24  on 185  degrees of freedom
## Residual deviance: 140.39  on 182  degrees of freedom
## AIC: 648.45
##
## Number of Fisher Scoring iterations: 5
##
##
## $S_n
##
## Call:
## glm(formula = S_n ~ site_type + site + block, family = "quasipoisson",
##      data = dat)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
```

```

## -2.48251 -0.40217 0.01867 0.38502 1.30213
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.95435    0.10852   8.794 2.91e-15 ***
## site_typewetland 0.33176    0.09391   3.533 0.000545 ***
## sitestono      0.34386    0.08679   3.962 0.000114 ***
## block         -0.08036    0.02165  -3.712 0.000288 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 0.4067537)
##
## Null deviance: 95.599 on 155 degrees of freedom
## Residual deviance: 83.199 on 152 degrees of freedom
## (30 observations deleted due to missingness)
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
##
##
## $S_PIE
##
## Call:
## glm(formula = S_PIE ~ site_type + site + block, family = "quasipoisson",
##      data = dat)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.7961 -1.7757 -0.4976  0.7580  5.8673
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.00516    0.34043   2.953 0.003833 **
## site_typewetland 1.11310    0.31065   3.583 0.000502 ***
## sitestono      0.79034    0.22440   3.522 0.000619 ***
## block         -0.14347    0.05579  -2.572 0.011418 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 3.744872)
##
## Null deviance: 525.93 on 116 degrees of freedom
## Residual deviance: 412.39 on 113 degrees of freedom
## (69 observations deleted due to missingness)
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
##
##
## $S_asymp
##
## Call:
## glm(formula = S_asymp ~ site_type + site + block, family = "quasipoisson",

```

```
##      data = dat)
##
## Deviance Residuals:
##      Min        1Q      Median        3Q        Max
## -3.1355  -1.2231  -0.4269   0.6542   5.4000
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.13689    0.20835   5.457 1.57e-07 ***
## site_typewetland 0.73496    0.18566   3.959 0.000108 ***
## sitestono       0.70080    0.17275   4.057 7.37e-05 ***
## block          -0.13970    0.04332  -3.225 0.001493 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 2.696207)
##
##      Null deviance: 538.33  on 185  degrees of freedom
## Residual deviance: 445.36  on 182  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

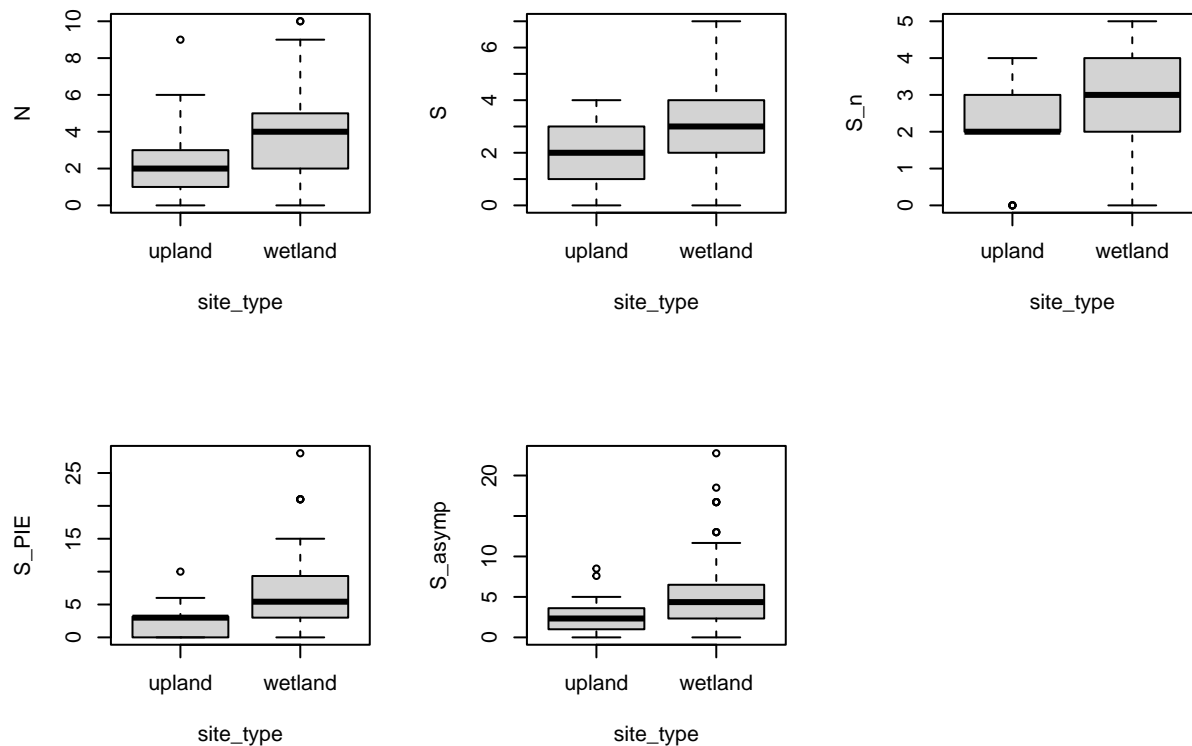
```
lapply(div_mods, anova)
```

```
## $N
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: N
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev
## NULL                185      262.39
## site_type  1  18.4853      184      243.91
## site       1   5.3338      183      238.57
## block      1  13.4746      182      225.10
##
## $S
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: S
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev
## NULL                185      170.24
## site_type  1  15.0157      184      155.23
```

```

## site      1    5.2693      183    149.96
## block     1    9.5634      182    140.39
##
## $S_n
## Analysis of Deviance Table
##
## Model: quasipoisson, link: log
##
## Response: S_n
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                        155      95.599
## site_type  1     4.8916      154     90.707
## site       1     1.6632      153     89.044
## block      1     5.8451      152     83.199
##
## $S_PIE
## Analysis of Deviance Table
##
## Model: quasipoisson, link: log
##
## Response: S_PIE
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                        116     525.93
## site_type  1    61.954      115     463.97
## site       1    24.779      114     439.19
## block      1    26.804      113     412.39
##
## $S_asymp
## Analysis of Deviance Table
##
## Model: quasipoisson, link: log
##
## Response: S_asymp
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                        185     538.33
## site_type  1    43.422      184     494.90
## site       1    19.299      183     475.61
## block      1    30.245      182     445.36

```



use mixed effect model to account for pseudo-replicates

```
indices <- c('N', 'S', 'S_n', 'S_PIE', 'S_asymp')
div_mods_me <- vector('list', length(indices))
names(div_mods_me) <- indices
for(i in seq_along(indices)) {
  div_mods_me[[i]] <- lme(as.formula(paste(indices[i], "~ site_type + site")),
    random = ~1 | year / block / wetland_id, data = dat,
    na.action = na.omit)
}

lapply(div_mods_me, summary)
```

```
## $N
## Linear mixed-effects model fit by REML
##   Data: dat
##       AIC      BIC    logLik
##   811.0088 833.4752 -398.5044
##
## Random effects:
## Formula: ~1 | year
##      (Intercept)
## StdDev: 0.0001173841
##
## Formula: ~1 | block %in% year
##      (Intercept)
```



```

## StdDev: 0.4173081
##
## Formula: ~1 | wetland_id %in% block %in% year
## (Intercept) Residual
## StdDev: 0.3368384 2.010724
##
## Fixed effects: as.formula(paste(indices[i], "~ site_type + site"))
## Value Std.Error DF t-value p-value
## (Intercept) 2.2150991 0.3917406 124 5.654504 0.0000
## site_typewetland 1.5260030 0.3912132 47 3.900695 0.0003
## sitestono 0.6645412 0.4051353 11 1.640294 0.1292
## Correlation:
## (Intr) st_typ
## site_typewetland -0.796
## sitestono -0.340 -0.018
##
## Standardized Within-Group Residuals:
## Min Q1 Med Q3 Max
## -2.0389669 -0.7058639 -0.0496332 0.6020161 3.0320740
##
## Number of Observations: 186
## Number of Groups:
## year block %in% year
## 2 14
## wetland_id %in% block %in% year
## 62
##
## $S
## Linear mixed-effects model fit by REML
## Data: dat
## AIC BIC logLik
## 662.6468 685.1132 -324.3234
##
## Random effects:
## Formula: ~1 | year
## (Intercept)
## StdDev: 6.802325e-05
##
## Formula: ~1 | block %in% year
## (Intercept)
## StdDev: 0.3287625
##
## Formula: ~1 | wetland_id %in% block %in% year
## (Intercept) Residual
## StdDev: 0.3134864 1.319824
##
## Fixed effects: as.formula(paste(indices[i], "~ site_type + site"))
## Value Std.Error DF t-value p-value
## (Intercept) 1.5648940 0.2727718 124 5.737008 0.0000
## site_typewetland 1.1782482 0.2670069 47 4.412800 0.0001
## sitestono 0.5469688 0.2923930 11 1.870663 0.0882
## Correlation:
## (Intr) st_typ
## site_typewetland -0.780

```

```

## sitestono          -0.352 -0.017
##
## Standardized Within-Group Residuals:
##      Min          Q1          Med          Q3          Max
## -2.17031301 -0.69785818 -0.03965633  0.60949336  2.94846031
##
## Number of Observations: 186
## Number of Groups:
##              year              block %in% year
##              2              14
## wetland_id %in% block %in% year
##              62
##
## $S_n
## Linear mixed-effects model fit by REML
##   Data: dat
##      AIC      BIC    logLik
##  480.235 501.4481 -233.1175
##
## Random effects:
## Formula: ~1 | year
##      (Intercept)
## StdDev: 0.0002281663
##
## Formula: ~1 | block %in% year
##      (Intercept)
## StdDev: 0.3858321
##
## Formula: ~1 | wetland_id %in% block %in% year
##      (Intercept) Residual
## StdDev: 0.2455886 1.006719
##
## Fixed effects: as.formula(paste(indices[i], "~ site_type + site"))
##              Value Std.Error DF   t-value p-value
## (Intercept)  1.9804196 0.2647115 94  7.481427  0.0000
## site_typewetland 0.8822926 0.2411433 47  3.658789  0.0006
## sitestono      0.2895579 0.2867775 11  1.009695  0.3343
## Correlation:
##              (Intr) st_typ
## site_typewetland -0.766
## sitestono        -0.379 -0.004
##
## Standardized Within-Group Residuals:
##      Min          Q1          Med          Q3          Max
## -2.637875504 -0.729688200 -0.004977302  0.682763807  1.994511074
##
## Number of Observations: 156
## Number of Groups:
##              year              block %in% year
##              2              14
## wetland_id %in% block %in% year
##              62
##
## $S_PIE

```

```

## Linear mixed-effects model fit by REML
##   Data: dat
##       AIC      BIC    logLik
##   707.7393 726.8927 -346.8696
##
## Random effects:
##   Formula: ~1 | year
##           (Intercept)
## StdDev: 3.025363e-05
##
##   Formula: ~1 | block %in% year
##           (Intercept)
## StdDev: 0.9871521
##
##   Formula: ~1 | wetland_id %in% block %in% year
##           (Intercept) Residual
## StdDev: 0.0004759956 4.762546
##
## Fixed effects: as.formula(paste(indices[i], "~ site_type + site"))
##               Value Std.Error DF   t-value p-value
## (Intercept)    1.305681  1.197342 58 1.090482   0.280
## site_typewetland 4.246144  1.207645 44 3.516054   0.001
## sitestono       2.103339  1.082547 11 1.942953   0.078
## Correlation:
##           (Intr) st_typ
## site_typewetland -0.839
## sitestono        -0.320 -0.008
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -1.4957699 -0.5681909 -0.1885212  0.3766772  4.5235716
##
## Number of Observations: 117
## Number of Groups:
##               year                block %in% year
##               2                  14
## wetland_id %in% block %in% year
##               59
##
## $S_asymp
## Linear mixed-effects model fit by REML
##   Data: dat
##       AIC      BIC    logLik
##   1018.597 1041.063 -502.2983
##
## Random effects:
##   Formula: ~1 | year
##           (Intercept)
## StdDev: 0.0001698476
##
##   Formula: ~1 | block %in% year
##           (Intercept)
## StdDev: 0.5630449
##

```

```
## Formula: ~1 | wetland_id %in% block %in% year
## (Intercept) Residual
## StdDev: 1.090982 3.464422
##
## Fixed effects: as.formula(paste(indices[i], "~ site_type + site"))
## Value Std.Error DF t-value p-value
## (Intercept) 1.935012 0.7144766 124 2.708294 0.0077
## site_typewetland 2.419427 0.7355654 47 3.289207 0.0019
## sitestono 1.392572 0.6920710 11 2.012182 0.0693
## Correlation:
## (Intr) st_typ
## site_typewetland -0.822
## sitestono -0.319 -0.019
##
## Standardized Within-Group Residuals:
## Min Q1 Med Q3 Max
## -1.5002000 -0.6112896 -0.2493320 0.3356174 4.7779029
##
## Number of Observations: 186
## Number of Groups:
## year block %in% year
## 2 14
## wetland_id %in% block %in% year
## 62
```

```
lapply(div_mods_me, r.squaredGLMM)
```

```
## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help page.
```

```
## $N
## R2m R2c
## [1,] 0.09895059 0.1587914
##
## $S
## R2m R2c
## [1,] 0.1303852 0.2224927
##
## $S_n
## R2m R2c
## [1,] 0.09121738 0.246697
##
## $S_PIE
## R2m R2c
## [1,] 0.1302593 0.1660863
##
## $S_asymp
## R2m R2c
## [1,] 0.09348444 0.194625
```

```
lapply(div_mods_me, anova)
```

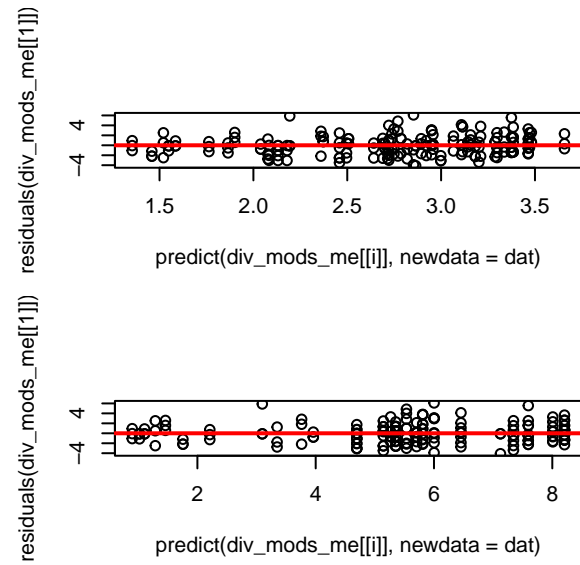
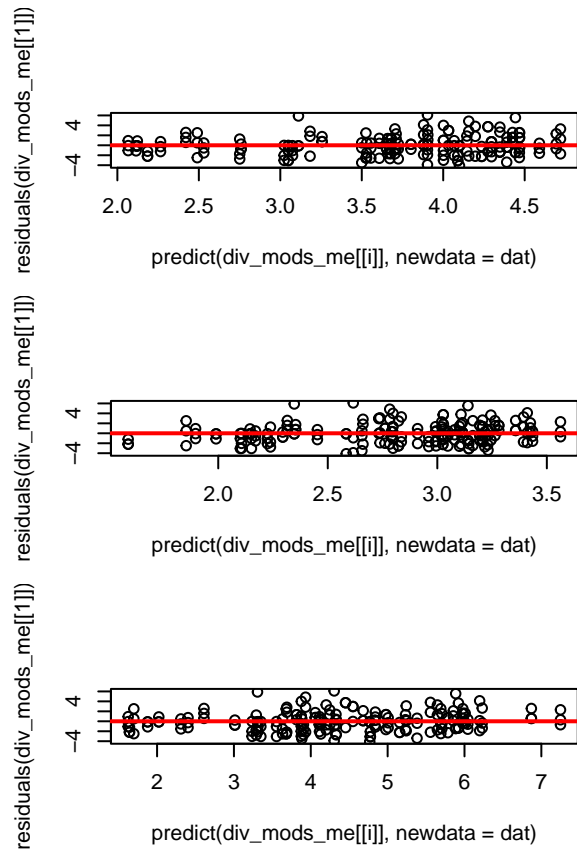
```
## $N
```

```
##          numDF denDF  F-value p-value
## (Intercept)      1   124 364.4230 <.0001
## site_type        1    47  15.4504 0.0003
## site             1    11   2.6906 0.1292
##
## $S
##          numDF denDF  F-value p-value
## (Intercept)      1   124 379.0111 <.0001
## site_type        1    47  19.7680 0.0001
## site             1    11   3.4994 0.0882
##
## $S_n
##          numDF denDF  F-value p-value
## (Intercept)      1    94 425.1705 <.0001
## site_type        1    47  13.4141 0.0006
## site             1    11   1.0195 0.3343
##
## $S_PIE
##          numDF denDF  F-value p-value
## (Intercept)      1    58 116.31570 <.0001
## site_type        1    44  12.47608 0.001
## site             1    11   3.77507 0.078
##
## $S_asymp
##          numDF denDF  F-value p-value
## (Intercept)      1   124 175.73953 <.0001
## site_type        1    47  11.07695 0.0017
## site             1    11   4.04887 0.0693
```

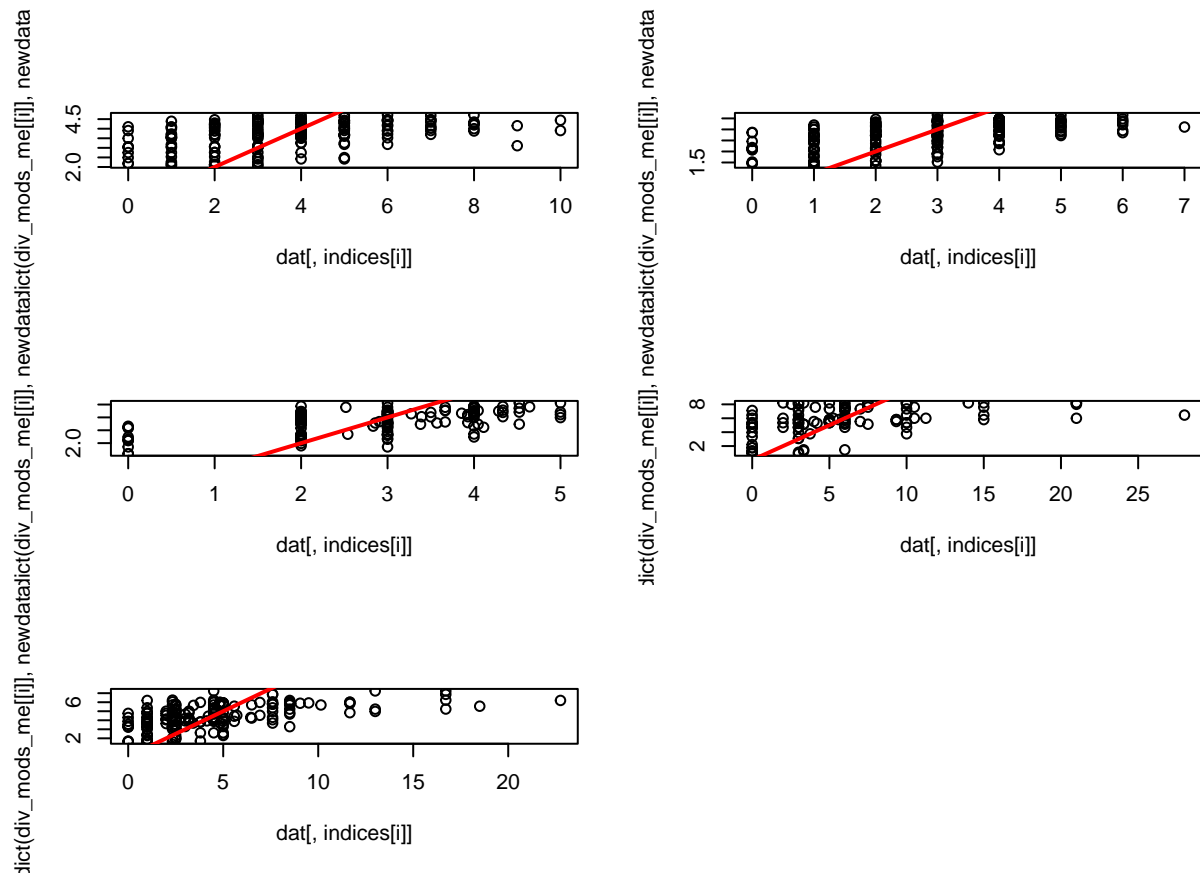
## Model Diagnostics

```
# fitted values vs residuals
par(mfrow=c(3,2))
for(i in seq_along(div_mods_me)) {
  plot(predict(div_mods_me[[i]], newdata = dat),
        residuals(div_mods_me[[1]]))
  abline(h= 0, col='red', lwd = 2)
}

# obs vs predicted plots
par(mfrow=c(3,2))
```



```
for(i in seq_along(div_mods_me)) {
  plot(dat[, indices[i]],
       predict(div_mods_me[[i]], newdata = dat))
  abline(a = 0, b = 1, col='red', lwd = 2)
}
```



run again but after aggregating across replicates within a year for all strings just take first value because these do not change during repeat visits

```
dat_chr_agg <- dat %>%
  group_by(wetland_id, year) %>%
  summarise_if(is.character, first)
```

for numeric values this mostly don't change visit to visit but a few do like temp and wind speed so take an average for the species we want to sum their counts though.

```
names(dat)
```

```
##      [1] "wetland_id"      "site"            "site_type"
##      [4] "block"           "date.x"          "year"
##      [7] "time"            "temp"            "water"
##     [10] "wind"            "uni_id_date"     "NOMO"
##     [13] "OROR"            "COYE"            "PIWA"
##     [16] "EAKI"            "CHSP"            "BLGR"
##     [19] "RBWO"            "TUTI"            "EATO"
##     [22] "NOCA"            "YBCU"            "BGGN"
##     [25] "CARW"            "YTVI"            "CACH"
##     [28] "GCFL"            "WOTH"            "DOWO"
##     [31] "BASP"            "WBNU"            "SUTA"
##     [34] "YBCH"            "NOPA"            "WEVI"
##     [37] "EAWP"            "YTWA"            "REVI"
```

```
## [40] "EABL"          "RHW"          "AMRO"
## [43] "RSHA"          "BHCO"          "INBU"
## [46] "BHNU"          "BRTH"          "MIKI"
## [49] "MODO"          "RWBL"          "COGR"
## [52] "PABU"          "RTHA"          "PIWO"
## [55] "BGNN"          "LBHE"          "BLJA"
## [58] "AMCR"          "ACFL"          "OVEN"
## [61] "WODU"          "GREG"          "GRHE"
## [64] "EWPE"          "KILL"          "BCSP"
## [67] "tree_sr"       "tree_dist"     "tree_dbh"
## [70] "tree_ba"       "tupelo_dbh"    "pine_dbh"
## [73] "other_dbh"     "date.y"        "time_start"
## [76] "time_end"      "property"      "type"
## [79] "bird_sr"       "shan_sr_1"     "shan_sr_2"
## [82] "agriculture"   "ditched"       "rain_24"
## [85] "inch_24"       "wetland_type"  "hydrology"
## [88] "last_dry"      "canopy_cover"  "canopy_dom"
## [91] "midstory_cover" "midstory_dom"  "debris_cover"
## [94] "debris_dom"    "groundcover"   "herb_layer"
## [97] "dom_herb_group" "water_cover"   "current_water_max"
## [100] "high_water_mark" "burn_scars"    "duff"
## [103] "duff_depth"    "hydro_dist"    "wetland_disturbance"
## [106] "natural_community" "canopy_dom_upland" "canopy_cover_upland"
## [109] "midstory_cover_upland" "midstory_dom_upland" "herb_cover_upland"
## [112] "herb_dom_upland" "stumps"        "burn_scars_upland"
## [115] "duff.1"        "duff_depth_upland" "time_last_fire"
## [118] "tree_height_mean" "tree_height_sd" "crown_hull_mean"
## [121] "canopy_cover_sim" "canopy_cover_sim.1" "N"
## [124] "S"             "S_n"           "S_PIE"
## [127] "S_asymp"
```

```
sp_cols <- 12:66
dat_num_agg <- dat[, -(12:66)] %>%
  group_by(wetland_id, year) %>%
  summarise_if(is.numeric, mean, na.rm = TRUE)
dat_sp_agg <- dat[, c('wetland_id', 'year', names(dat)[sp_cols])] %>%
  group_by(wetland_id, year) %>%
  summarise_all(sum)
# note on next line some of the names get changed with column binding
dat_agg <- cbind(dat_chr_agg, dat_num_agg, dat_sp_agg)
```

```
## New names:
## * 'wetland_id' -> 'wetland_id...1'
## * 'year' -> 'year...2'
## * 'wetland_id' -> 'wetland_id...35'
## * 'year' -> 'year...36'
## * 'wetland_id' -> 'wetland_id...67'
## * 'year' -> 'year...68'
```

```
# fix wonky names
names(dat_agg) <- c('wetland_id', 'year', names(dat_agg)[-(1:2)])
dim(dat_agg)
```

```
## [1] 62 123
```



```
comm_agg <- dat_sp_agg[ , -(1:2)]
comm_agg
```

```
## # A tibble: 62 x 55
##   NOMO  OROR  COYE  PIWA  EAKI  CHSP  BLGR  RBWO  TUTI  EATO  NOCA  YBCU  BGGN
##   <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1     2     1     3     1     0     0     1     0     0     0     0     0     0
## 2     0     0     1     0     0     0     0     0     2     0     0     2     3
## 3     0     0     1     0     0     0     0     1     3     0     4     1     3
## 4     2     0     0     0     0     0     0     0     4     1     0     0     5
## 5     0     0     1     0     0     0     0     0     0     0     0     0     4
## 6     0     0     3     0     0     0     0     0     3     1     0     0     1
## 7     0     0     0     0     0     0     0     1     2     0     0     0     0
## 8     0     0     0     0     0     0     0     0     0     0     0     0     1
## 9     1     0     2     0     0     0     0     2     2     1     1     0     2
## 10    0     0     2     0     0     0     1     0     0     3     0     0     6
## # ... with 52 more rows, and 42 more variables: CARW <int>, YTVI <int>,
## #   CACH <int>, GCFL <int>, WOTH <int>, DOWO <int>, BASP <int>, WBNU <int>,
## #   SUTA <int>, YBCH <int>, NOPA <int>, WEVI <int>, EAWP <int>, YTWA <int>,
## #   REVI <int>, EABL <int>, RHWO <int>, AMRO <int>, RSHA <int>, BHCO <int>,
## #   INBU <int>, BHNU <int>, BRTH <int>, MIKI <int>, MODO <int>, RWBL <int>,
## #   COGR <int>, PABU <int>, RTHA <int>, PIWO <int>, BGNN <int>, LBHE <int>,
## #   BLJA <int>, AMCR <int>, ACFL <int>, OVEN <int>, WODU <int>, GREG <int>, ...
```

```
sum(comm_agg)
```

```
## [1] 684
```

```
sum(comm)
```

```
## [1] 684
```

```
rowSums(comm_agg)
```

```
## [1]  8 11 14 14 10 12  6  7 14 19  7 11 14 12  8 10  6  5 15 16 11  8 14  4  5
## [26] 13 12 13 12 18 18 13 16 17 10 16 13 17 14 14 17 12 13 12  8 14 17  4 12  8
## [51]  6  6  6  5  8  1 12  7 15 12  5  7
```

```
rowSums(comm_agg[dat_agg$wetland_id == "UP03", ])
```

```
## [1] 8 1
```

```
dat_agg$S <- rowSums(comm_agg > 0 )
dat_agg$N <- rowSums(comm_agg)
summary(dat_agg$N)
```

```
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.00   7.25   12.00   11.03   14.00   19.00
```

```

dat_agg$S_n <- apply(comm_agg, 1, rarefaction, 'IBR', effort = 10, extrapolate = F,
                    quiet_mode = TRUE)
dat_agg$S_n <- ifelse(dat_agg$S_n == 1, NA, dat_agg$S_n)
dat_agg$S_PIE <- calc_SPIE(comm_agg)

```

```

## Warning in calc_PIE(x, replace = replace): NA was returned because the sample
## contains one or zero individuals.

```

```

## Warning in calc_SPIE(comm_agg): NA was returned because PIE = 1. This happens in
## samples where all species are singletons.

```

```

dat_agg$S_asymp <- apply(comm_agg, 1, calc_chao1)
dat_agg$pct_rare <- apply(comm_agg, 1, calc_div, 'pct_rare', rare_thres = 0.2)

```

```

par(mfrow=c(2,3))
plot(density(dat_agg$N))
plot(density(dat_agg$S))
plot(density(dat_agg$S_n, na.rm = TRUE))
plot(density(dat_agg$S_PIE, na.rm = TRUE))
plot(density(dat_agg$S_asymp, na.rm = TRUE))

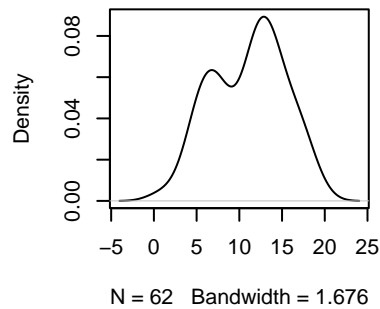
```

```

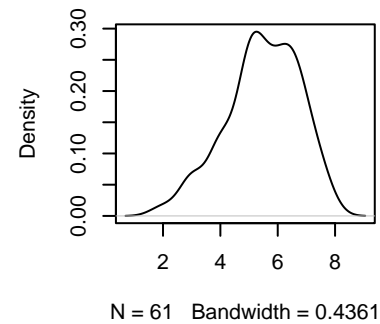
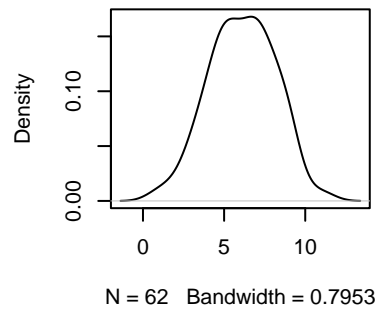
par(mfrow=c(2,3))

```

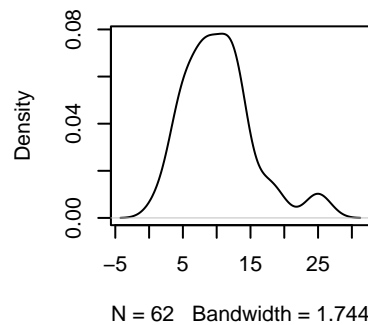
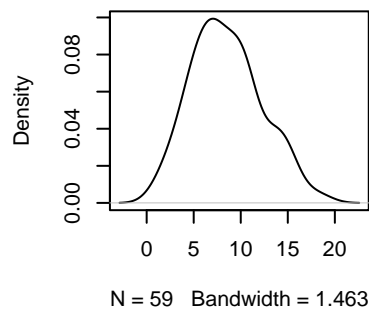
**density.default(x = dat\_agg\$N**



**density.default(x = dat\_agg\$S, na.rm = TRUE)**



**y.default(x = dat\_agg\$S\_PIE, na.rm = TRUE)**



```

boxplot(N ~ site_type, data = dat_agg)
boxplot(S ~ site_type, data = dat_agg)
boxplot(S_n ~ site_type, data = dat_agg)
boxplot(S_PIE ~ site_type, data = dat_agg)
boxplot(S_asymp ~ site_type, data = dat_agg)

div_mods <- list()
div_mods$N <- glm(N ~ site_type + site + block, data = dat_agg, family = 'poisson')
div_mods$S <- glm(S ~ site_type + site + block, data = dat_agg, family = 'poisson')
div_mods$S_n <- glm(S_n ~ site_type + site + block, data = dat_agg, family = 'quasipoisson')
div_mods$S_PIE <- glm(S_PIE ~ site_type + site + block, data = dat_agg, family = 'quasipoisson')
div_mods$S_asymp <- glm(S_asymp ~ site_type + site + block, data = dat_agg, family = 'quasipoisson')

lapply(div_mods, summary)

```

```

## $N
##
## Call:
## glm(formula = N ~ site_type + site + block, family = "poisson",
##      data = dat_agg)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.93009  -0.80634   0.04221   0.56992   2.10503
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      2.20213     0.12855  17.131 < 2e-16 ***
## site_typewetland  0.50385     0.11376   4.429 9.46e-06 ***
## sitestono        0.44957     0.11073   4.060 4.90e-05 ***
## block           -0.09830     0.02747  -3.578 0.000346 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 110.047  on 61  degrees of freedom
## Residual deviance:  72.753  on 58  degrees of freedom
## AIC: 338.56
##
## Number of Fisher Scoring iterations: 4
##
## $S
##
## Call:
## glm(formula = S ~ site_type + site + block, family = "poisson",
##      data = dat_agg)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.77151  -0.48100  -0.02403   0.47540   1.17356
##

```

```

## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.67171    0.16923   9.878 < 2e-16 ***
## site_typewetland 0.41004    0.14785   2.773 0.005549 **
## sitestono       0.51924    0.14967   3.469 0.000522 ***
## block          -0.10014    0.03747  -2.673 0.007519 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 43.815  on 61  degrees of freedom
## Residual deviance: 24.010  on 58  degrees of freedom
## AIC: 255.96
##
## Number of Fisher Scoring iterations: 4
##
##
## $S_n
##
## Call:
## glm(formula = S_n ~ site_type + site + block, family = "quasipoisson",
##      data = dat_agg)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.11761  -0.32006   0.04689   0.34502   0.76836
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.64100    0.07747  21.182 < 2e-16 ***
## site_typewetland 0.25127    0.06796   3.697 0.000491 ***
## sitestono       0.35961    0.06868   5.236 2.46e-06 ***
## block          -0.07151    0.01703  -4.200 9.48e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 0.1945479)
##
##      Null deviance: 19.220  on 60  degrees of freedom
## Residual deviance: 11.485  on 57  degrees of freedom
## (1 observation deleted due to missingness)
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
##
##
## $S_PIE
##
## Call:
## glm(formula = S_PIE ~ site_type + site + block, family = "quasipoisson",
##      data = dat_agg)
##
## Deviance Residuals:

```

```

##      Min      1Q   Median      3Q      Max
## -2.3363 -0.8948 -0.1626   0.9011   2.5594
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.12582    0.17642  12.050  <2e-16 ***
## site_typewetland 0.17213    0.15845   1.086   0.2821
## sitestono      0.33400    0.15724   2.124   0.0382 *
## block         -0.06293    0.03880  -1.622   0.1105
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.575539)
##
##      Null deviance: 97.113  on 58  degrees of freedom
## Residual deviance: 88.312  on 55  degrees of freedom
## (3 observations deleted due to missingness)
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
##
##
## $S_asymp
##
## Call:
## glm(formula = S_asymp ~ site_type + site + block, family = "quasipoisson",
##      data = dat_agg)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.7184 -0.9733 -0.2812   0.8734   3.4653
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.34427    0.17458  13.428  < 2e-16 ***
## site_typewetland 0.36849    0.15015   2.454 0.017146 *
## sitestono      0.78297    0.16445   4.761 1.32e-05 ***
## block         -0.15721    0.04146  -3.792 0.000358 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.815888)
##
##      Null deviance: 157.88  on 61  degrees of freedom
## Residual deviance: 103.20  on 58  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4

lapply(div_mods, anova)

## $N
## Analysis of Deviance Table
##

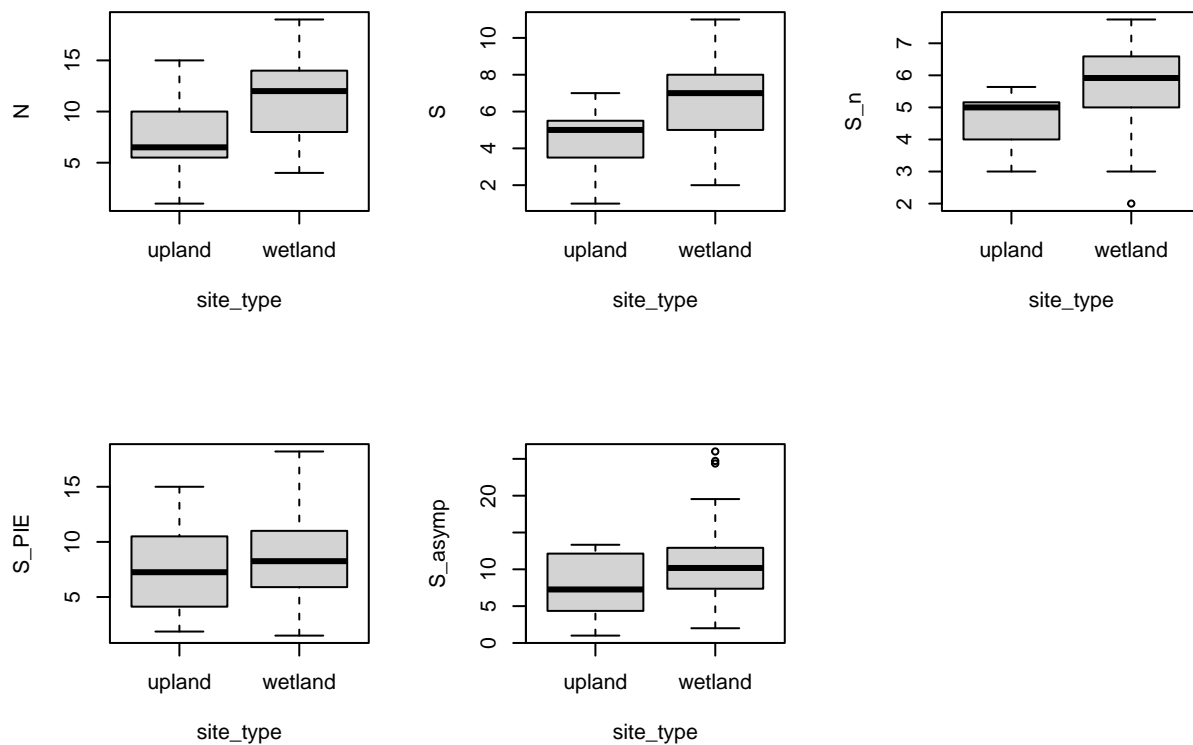
```

```

## Model: poisson, link: log
##
## Response: N
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                                61    110.047
## site_type  1  18.4853          60     91.562
## site       1   5.3338          59     86.228
## block      1  13.4746          58     72.753
##
## $S
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: S
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                                61     43.815
## site_type  1   6.8272          60     36.988
## site       1   5.4531          59     31.535
## block      1   7.5251          58     24.010
##
## $S_n
## Analysis of Deviance Table
##
## Model: quasipoisson, link: log
##
## Response: S_n
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                                60     19.220
## site_type  1   2.0036          59     17.216
## site       1   2.1724          58     15.043
## block      1   3.5584          57     11.485
##
## $S_PIE
## Analysis of Deviance Table
##
## Model: quasipoisson, link: log
##
## Response: S_PIE
##
## Terms added sequentially (first to last)
##

```

```
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                      58      97.113
## site_type 1      1.2998      57      95.814
## site      1      3.2238      56      92.590
## block     1      4.2782      55      88.312
##
## $S_asymp
## Analysis of Deviance Table
##
## Model: quasipoisson, link: log
##
## Response: S_asymp
##
## Terms added sequentially (first to last)
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                      61     157.88
## site_type 1      8.2868      60     149.59
## site      1     17.9724      59     131.62
## block     1     28.4180      58     103.20
```



use mixed effect model to account for pseudo-replicates

```

indices <- c('N', 'S', 'S_n', 'S_PIE', 'S_asymp')
div_mods_me <- vector('list', length(indices))
names(div_mods_me) <- indices
for(i in seq_along(indices)) {
  div_mods_me[[i]] <- lme(as.formula(paste(indices[i], "~ site_type + site")),
    random = ~1 | wetland_id, data = dat_agg,
    na.action = na.omit)
}

lapply(div_mods_me, summary)

```

```

## $N
## Linear mixed-effects model fit by REML
##   Data: dat_agg
##       AIC      BIC    logLik
##  340.0714 350.459 -165.0357
##
## Random effects:
##   Formula: ~1 | wetland_id
##           (Intercept) Residual
## StdDev:    2.677176 2.844505
##
## Fixed effects: as.formula(paste(indices[i], "~ site_type + site"))
##               Value Std.Error DF   t-value p-value
## (Intercept)    6.800623  1.424839 34  4.772907  0.0000
## site_typewetland 4.012611  1.507555 34  2.661668  0.0118
## sitestono       2.098131  1.205062 34  1.741097  0.0907
## Correlation:
##               (Intr) st_typ
## site_typewetland -0.865
## sitestono        -0.282 -0.019
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -1.661267734 -0.548558359 -0.002660407  0.630454407  1.600202633
##
## Number of Observations: 62
## Number of Groups: 37
##
## $S
## Linear mixed-effects model fit by REML
##   Data: dat_agg
##       AIC      BIC    logLik
##  246.572 256.9597 -118.286
##
## Random effects:
##   Formula: ~1 | wetland_id
##           (Intercept) Residual
## StdDev:    1.217573 1.284999
##
## Fixed effects: as.formula(paste(indices[i], "~ site_type + site"))
##               Value Std.Error DF   t-value p-value
## (Intercept)    3.962688  0.6464338 34  6.130076  0.0000

```



```

## site_typedwetland 1.791461 0.6839328 34 2.619353 0.0131
## sitestono 1.611935 0.5465777 34 2.949142 0.0057
## Correlation:
## (Intr) st_typ
## site_typedwetland -0.865
## sitestono -0.282 -0.019
##
## Standardized Within-Group Residuals:
## Min Q1 Med Q3 Max
## -1.53941634 -0.50339028 0.03941161 0.45482944 2.01125084
##
## Number of Observations: 62
## Number of Groups: 37
##
## $S_n
## Linear mixed-effects model fit by REML
## Data: dat_agg
## AIC BIC logLik
## 196.7961 207.0984 -93.39807
##
## Random effects:
## Formula: ~1 | wetland_id
## (Intercept) Residual
## StdDev: 0.7762853 0.8873765
##
## Fixed effects: as.formula(paste(indices[i], "~ site_type + site"))
## Value Std.Error DF t-value p-value
## (Intercept) 4.175404 0.4361708 34 9.572863 0.0000
## site_typedwetland 1.023825 0.4586816 34 2.232104 0.0323
## sitestono 1.000348 0.3611240 34 2.770095 0.0090
## Correlation:
## (Intr) st_typ
## site_typedwetland -0.869
## sitestono -0.290 -0.006
##
## Standardized Within-Group Residuals:
## Min Q1 Med Q3 Max
## -2.042306741 -0.553260806 0.002882615 0.633662486 2.181641930
##
## Number of Observations: 61
## Number of Groups: 37
##
## $S_PIE
## Linear mixed-effects model fit by REML
## Data: dat_agg
## AIC BIC logLik
## 323.799 333.9257 -156.8995
##
## Random effects:
## Formula: ~1 | wetland_id
## (Intercept) Residual
## StdDev: 0.7517033 3.612896
##
## Fixed effects: as.formula(paste(indices[i], "~ site_type + site"))

```

```
##               Value Std.Error DF   t-value p-value
## (Intercept)      7.098330   1.226187 33  5.788948  0.0000
## site_typewetland 1.066820   1.302129 33  0.819289  0.4185
## sitestono        1.490199   1.031926 33  1.444094  0.1581
## Correlation:
##               (Intr) st_typ
## site_typewetland -0.872
## sitestono        -0.254 -0.035
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -1.7682743 -0.6775067 -0.1699467  0.5930000  2.6918685
##
## Number of Observations: 59
## Number of Groups: 36
##
## $S_asymp
## Linear mixed-effects model fit by REML
##   Data: dat_agg
##       AIC       BIC    logLik
##  373.0426 383.4303 -181.5213
##
## Random effects:
## Formula: ~1 | wetland_id
##      (Intercept) Residual
## StdDev:    1.620125  4.60685
##
## Fixed effects: as.formula(paste(indices[i], "~ site_type + site"))
##               Value Std.Error DF   t-value p-value
## (Intercept)      6.755800   1.552152 34  4.352537  0.0001
## site_typewetland 2.693178   1.650187 34  1.632044  0.1119
## sitestono        3.663552   1.352067 34  2.709593  0.0105
## Correlation:
##               (Intr) st_typ
## site_typewetland -0.855
## sitestono        -0.290 -0.021
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -1.45446430 -0.72890474 -0.06958884  0.50291385  3.03069515
##
## Number of Observations: 62
## Number of Groups: 37
```

```
lapply(div_mods_me, r.squaredGLMM)
```

```
## $N
##           R2m           R2c
## [1,] 0.1930309 0.5720834
##
## $S
##           R2m           R2c
## [1,] 0.2664325 0.6134663
##
```

```
## $S_n
##           R2m           R2c
## [1,] 0.2195279 0.5578792
##
## $S_PIE
##           R2m           R2c
## [1,] 0.04825874 0.08774955
##
## $S_asymp
##           R2m           R2c
## [1,] 0.1544808 0.2475425
```

```
lapply(div_mods_me, anova)
```

```
## $N
##           numDF denDF  F-value p-value
## (Intercept)      1    34 354.5348 <.0001
## site_type        1    34   7.2658 0.0108
## site             1    34   3.0314 0.0907
##
## $S
##           numDF denDF  F-value p-value
## (Intercept)      1    34 528.7682 <.0001
## site_type        1    34   7.1625 0.0114
## site             1    34   8.6974 0.0057
##
## $S_n
##           numDF denDF  F-value p-value
## (Intercept)      1    34 967.8606 <.0001
## site_type        1    34   5.0571 0.0311
## site             1    34   7.6734 0.0090
##
## $S_PIE
##           numDF denDF  F-value p-value
## (Intercept)      1    33 302.52919 <.0001
## site_type        1    33   0.75844 0.3901
## site             1    33   2.08541 0.1581
##
## $S_asymp
##           numDF denDF  F-value p-value
## (Intercept)      1    34 250.66098 <.0001
## site_type        1    34   2.85646 0.1002
## site             1    34   7.34189 0.0105
```

```
# beta diversity analysis -----
# loop through each block and year and pull 1 upland and 1 wetland
# Note: two blocks do not have an upland and a wetlands sites, blocks 7 & 8
# Note: block 3 the upland site UP03 only has 1 individual across the 3 visits
# in the 0-25m range so drop this site and block from analysis

nboot <- 500
uni_yrs <- unique(dat_agg$year)
uni_blocks <- 1:6 # not including blocks 7 & 8 here b/c these only had wetland sites
```

```

#uni_blocks <- 1:4 # to only do halidon sites

betas <- data.frame()
curves <- data.frame()

Navg_up <- sum(comm_agg[dat_agg$site_type == 'upland', ]) /
  nrow(comm_agg[dat_agg$site_type == 'upland', ])
Navg_we <- sum(comm_agg[dat_agg$site_type == 'wetland', ]) /
  nrow(comm_agg[dat_agg$site_type == 'wetland', ])

for (i in 1:nboot) {
  for (j in seq_along(uni_yrs)) {
    uplands <- data.frame()
    wetlands <- data.frame()
    for (k in seq_along(uni_blocks)) {
      good_rows <- dat_agg$year == uni_yrs[j] &
        dat_agg$block == uni_blocks[k]
      sample_ids <- unique(dat_agg$wetland_id[good_rows])
      # from this list draw a single wetland and a single upland
      upland_id <- sample_ids[grepl('UP', sample_ids)]
      wetland_id <- sample(sample_ids[!(sample_ids %in% upland_id)], 1)
      upland_samples <- comm_agg[good_rows & dat_agg$wetland_id == upland_id, ]
      wetland_samples <- comm_agg[good_rows & dat_agg$wetland_id == wetland_id, ]
      # keep just one of the sites in the wetland samples
      #random_wetland_id <- sample(unique(wetland_samples$wetland_id), 1)
      #wetland_samples <- subset(wetland_samples, wetland_id == random_wetland_id)
      # ok now we have 3 samples from the upland and 3 samples from a wetland
      # in a specific year
      uplands <- rbind(uplands, upland_samples)
      wetlands <- rbind(wetlands, wetland_samples)
    }
    # now capture rarefaction curve and compute beta div at two scales point count to block & block to
    #up_N_min <- min(rowSums(uplands))
    #we_N_min <- min(rowSums(wetlands))
    betas <- rbind(betas,
      data.frame(boot = i, site_type = 'upland',
        calc_comm_div(uplands, index = c('S', 'S_n', 'S_PIE'),
          effort = 10, scale = 'beta'))))
    betas <- rbind(betas,
      data.frame(boot = i, site_type = 'wetland',
        calc_comm_div(wetlands, index = c('S', 'S_n', 'S_PIE'),
          effort = 10, scale = 'beta'))))
    nmin <- max(min(rowSums(uplands)), 5)
    S_up <- apply(uplands, 1, rarefaction, 'IBR', effort = 1:nmin, extrapolate = TRUE,
      quiet_mode = TRUE)
    if (is.matrix(S_up)) S_up <- rowMeans(S_up)
    curves <- rbind(curves,
      data.frame(boot = i, site_type = 'upland', scale = 'alpha',
        effort = 1:nmin, S = S_up))
    curves <- rbind(curves,
      data.frame(boot = i, site_type = 'upland', scale = 'gamma',
        effort = 1:sum(uplands), S = rarefaction(uplands, 'IBR'))))
  }
}

```

```

nmin <- max(min(rowSums(wetlands)), 5)
S_we <- apply(wetlands, 1, rarefaction, 'IBR', effort = 1:nmin, extrapolate = TRUE,
             quiet_mode = TRUE)
if (is.matrix(S_we)) S_we <- rowMeans(S_we)

curves <- rbind(curves,
               data.frame(boot = i, site_type = 'wetland', scale = 'alpha',
                           effort = 1:nmin, S = S_we))
curves <- rbind(curves,
               data.frame(boot = i, site_type = 'wetland', scale = 'gamma',
                           effort = 1:sum(wetlands), S = rarefaction(wetlands, 'IBR'))))
}
}

```

```
save(betas, curves, file = './results/div_bootstrap_results.Rdata')
```

```
load(file = './results/div_bootstrap_results.Rdata')
```

aggregated across bootstraps

```

beta_sum <- betas %>% group_by(index, site_type) %>%
  summarize(beta_avg = mean(value), beta_lo = quantile(value, 0.025),
            beta_hi = quantile(value, 0.975))

```

## 'summarise()' has grouped output by 'index'. You can override using the  
## '.groups' argument.

```

curves_sum <- curves %>% group_by(site_type, scale, effort) %>%
  summarize(S_avg = mean(S), S_lo = quantile(S, 0.025), S_hi = quantile(S, 0.975))

```

## 'summarise()' has grouped output by 'site\_type', 'scale'. You can override  
## using the '.groups' argument.

Here are the computed beta-diversity metrics with 95% CI

```
beta_sum
```

```

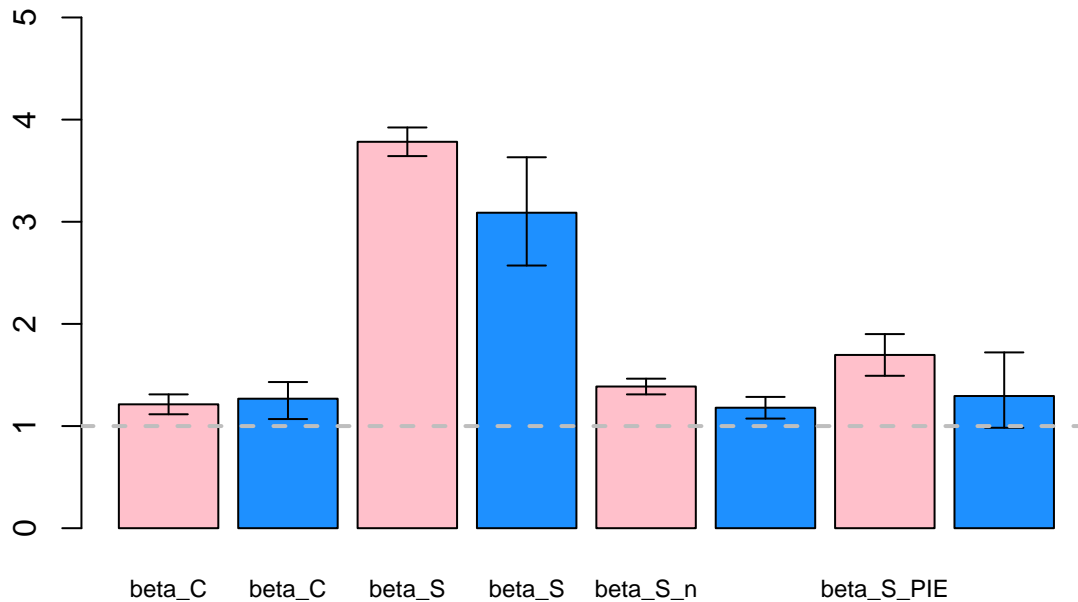
## # A tibble: 8 x 5
## # Groups:   index [4]
##   index      site_type beta_avg beta_lo beta_hi
##   <chr>      <chr>      <dbl>  <dbl>  <dbl>
## 1 beta_C    upland        1.21   1.12   1.31
## 2 beta_C    wetland       1.27   1.07   1.43
## 3 beta_S    upland        3.78   3.64   3.92
## 4 beta_S    wetland       3.09   2.57   3.63
## 5 beta_S_n  upland        1.39   1.31   1.46
## 6 beta_S_n  wetland       1.18   1.07   1.29
## 7 beta_S_PIE upland       1.70   1.49   1.90
## 8 beta_S_PIE wetland       1.29   0.983  1.72

```

```

# make parplot
par(mfrow=c(1,1))
tmp <- barplot(height = beta_sum$beta_avg, col = c('pink', 'dodgerblue'), ylim = c(0, 5),
               names = beta_sum$index, cex.names = 0.75)
arrows(x0 = tmp,
       y0 = beta_sum$beta_lo,
       y1 = beta_sum$beta_hi,
       angle = 90,
       code = 3,
       length = 0.1)
abline(h = 1, col='grey', lty =2, lwd =2)

```



```

nmax <- curves %>% group_by(site_type, scale, boot) %>%
  summarize(N_max = max(effort)) %>%
  summarize(N_min = min(N_max))

```

```

## 'summarise()' has grouped output by 'site_type', 'scale'. You can override
## using the '.groups' argument.
## 'summarise()' has grouped output by 'site_type'. You can override using the
## '.groups' argument.

```

```

plot(S_hi ~ effort, curves_sum, subset = site_type == 'wetland' & scale == 'gamma',
     type = 'l', col= 'dodgerblue', log = 'xy')
lines(S_lo ~ effort, curves_sum, subset = site_type == 'wetland' & scale == 'gamma',

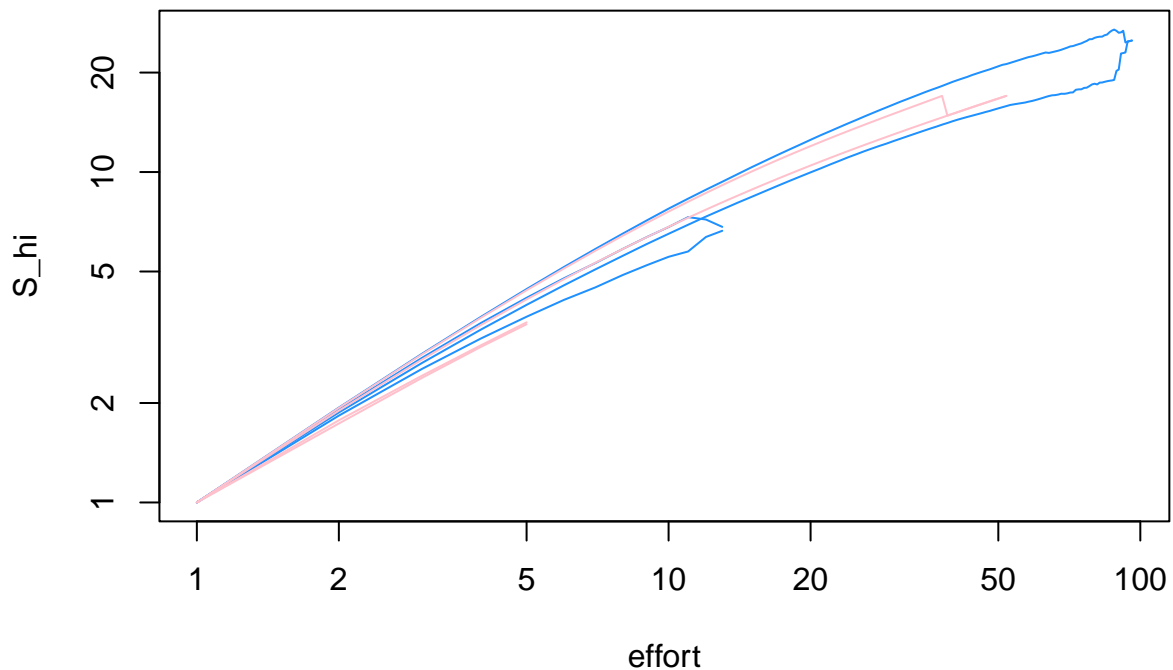
```

```

col = 'dodgerblue')
lines(S_hi ~ effort, curves_sum, subset = site_type == 'wetland' & scale == 'alpha',
col = 'dodgerblue')
lines(S_lo ~ effort, curves_sum, subset = site_type == 'wetland' & scale == 'alpha',
col = 'dodgerblue')

lines(S_hi ~ effort, curves_sum, subset = site_type == 'upland' & scale == 'gamma',
col = 'pink')
lines(S_lo ~ effort, curves_sum, subset = site_type == 'upland' & scale == 'gamma',
col = 'pink')
lines(S_hi ~ effort, curves_sum, subset = site_type == 'upland' & scale == 'alpha',
col = 'pink')
lines(S_lo ~ effort, curves_sum, subset = site_type == 'upland' & scale == 'alpha',
col = 'pink')

```

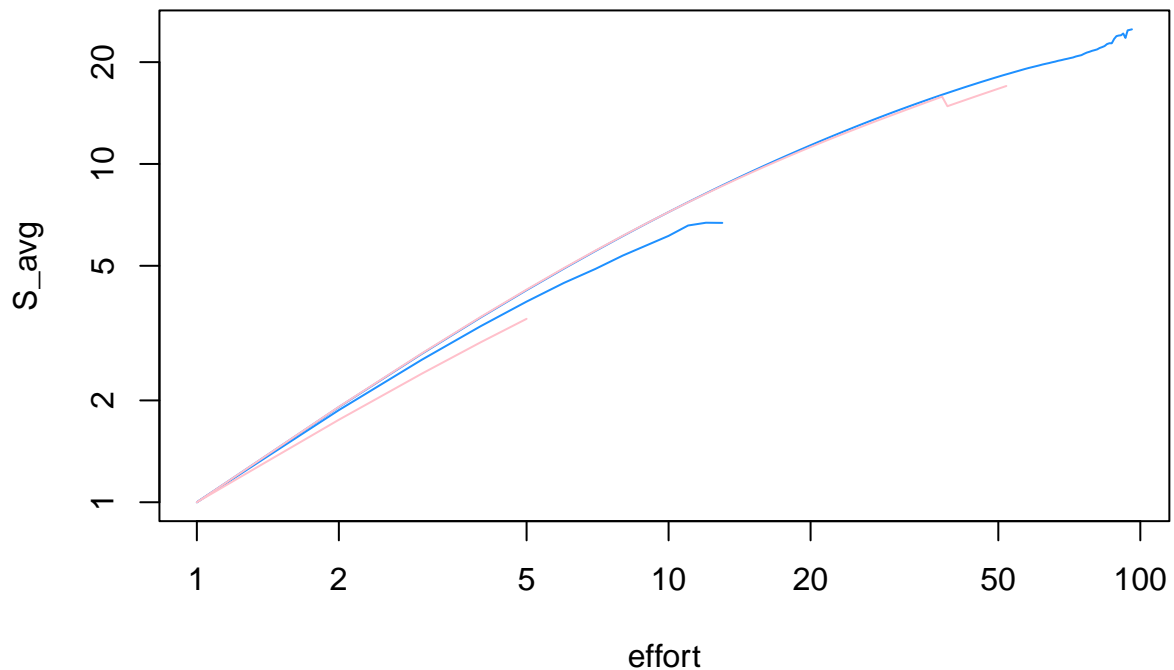


```

plot(S_avg ~ effort, curves_sum, subset = site_type == 'wetland' & scale == 'gamma',
type = 'l', col = 'dodgerblue', log = 'xy')
lines(S_avg ~ effort, curves_sum, subset = site_type == 'wetland' & scale == 'alpha',
col = 'dodgerblue')

lines(S_avg ~ effort, curves_sum, subset = site_type == 'upland' & scale == 'gamma',
col = 'pink')
lines(S_avg ~ effort, curves_sum, subset = site_type == 'upland' & scale == 'alpha',
col = 'pink')

```



classic un-balanced rarefaction comparison

```
dat_mob_in <- make_mob_in(dat[, 12:66], dat)
```

```
## Warning in make_mob_in(dat[, 12:66], dat): Some species have zero occurrences
## and will be dropped from the community table
```

```
## Warning in make_mob_in(dat[, 12:66], dat): Note: 'coord_names' was not supplied
## and therefore spatial aggregation will not be examined in downstream analyses
```

```
par(mfrow=c(1,3))
plot_rarefaction(dat_mob_in, group_var = 'site_type', method = 'IBR', avg= TRUE,
  log='xy')
```



