

niche_differences

Katharine Prata

20/01/2022

Nich Differences

```
library(vcfR)
```

```
##
##      *****      ***   vcfR   ***      *****
##      This is vcfR 1.12.0
##      browseVignettes('vcfR') # Documentation
##      citation('vcfR') # Citation
##      *****      *****      *****      *****
```

```
library(ade4)
```

```
## Loading required package: ade4
```

```
## Registered S3 method overwritten by 'spdep':
```

```
##   method      from
```

```
##   plot.mst ape
```

```
##
```

```
##   /// adegenet 2.1.3 is loaded //////////////////////////////////
```

```
##
```

```
##   > overview: '?adegenet'
```

```
##   > tutorials/doc/questions: 'adegenetWeb()'
```

```
##   > bug reports/feature requests: adegenetIssues()
```

```
library(ggplot2)
```

```
library(lme4)
```

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

```
library(emmeans)
```

```
# Functions
```

```
combine_metadata <- function(struc.complex, clusters, coordinates) {
```

```
  # match
```

```

struc.complex.subset <- subset(struc.complex, X %in% clusters$Individual)
clusters.subset <- subset(clusters, Individual %in% struc.complex.subset$X)
coordinates.subset <- subset(coordinates, X %in% struc.complex.subset$X)

# sort
sorted.clusters.subset <- clusters.subset[order(clusters.subset[,1]),]
sorted.coordinates.subset <- coordinates.subset[order(coordinates.subset[,1]),]
sorted.struc.complex.subset <- struc.complex.subset[order(struc.complex.subset[,1]),]
# check if identical
cat("\nChecking clusters and struc complex: ", identical(sorted.clusters.subset[,1], sorted.struc.complex.subset[,1]), "\n")
cat("\nChecking clusters and coordinates: ", identical(sorted.clusters.subset[,1], sorted.coordinates.subset[,1]), "\n")
# Put metadata together
metadata <- sorted.clusters.subset
metadata[,20:25] <- sorted.struc.complex.subset[,2:7]
metadata[, 26:28] <- sorted.coordinates.subset[,2:4]
return(metadata)
}

overhang_barplot <- function(metadata, factor) {
  overhang <- table(metadata$overhang, factor) %>% matrix(nrow = 2, ncol = length(levels(factor)))
  colnames(overhang) <- levels(factor)
  rownames(overhang) <- levels(metadata$overhang)
  barplot(overhang, main = "Presence of overhang", xlab = "Taxa",
          col = c("Red", "Green"))
  legend("topright", c("No", "Yes"), fill = c("Red", "Green"))
}

k_boxplot <- function(df, cat, cont) {

  levels <- levels(cat)
  sample.sizes <- NA

  for (i in 1:length(levels)) {
    sample.sizes[i] <- length(cat[cat == levels[i]])
  }

  label = paste0(levels, "\n n = ", sample.sizes)

  p <- ggplot(df, aes(x = cat, y = cont)) +
    geom_boxplot(notch = TRUE) +
    geom_point(size = 3, shape = 1) +
    scale_x_discrete(labels = label) +
    theme_classic(base_size = 12)
  return(p)
}

# Import datasets
struc.complex <- read.csv("~/git/coralscape_open3d/results/sample_metadata.csv")
clusters <- read.csv("~/Dropbox/agaricia_project_2019/shalo_ag/gen_project/data/all-aga_1d_nc_20_6.csv")
coordinates <- read.csv("~/git/coralscape_open3d/results/annotations.csv")

# Organise data
metadata <- combine_metadata(struc.complex, clusters, coordinates)

```

```
##
## Checking clusters and struc complex: TRUE
## Checking clusters and coordinates: TRUE

# Clean working space
rm(struc.complex)
rm(clusters)
rm(coordinates)

# Data structure
str(metadata)

## 'data.frame': 264 obs. of 28 variables:
## $ Individual: chr "KP0057_AC_SB10" "KP0058_AC_SB10" "KP0061_AC_SB10" "KP0072_AC_SB10" ...
## $ Species : chr "AC" "AC" "AC" "AC" ...
## $ Population: chr "AC_SB10" "AC_SB10" "AC_SB10" "AC_SB10" ...
## $ Site : chr "SB10" "SB10" "SB10" "SB10" ...
## $ Loc : chr "SB" "SB" "SB" "SB" ...
## $ Depth : int 10 10 10 10 10 10 10 10 10 10 ...
## $ V1 : num 16.563 16.62 16.536 16.564 0.663 ...
## $ V2 : num -10.6 -10.8 -10.7 -10.5 28.5 ...
## $ V3 : num 1.288 1.792 0.485 0.736 0.41 ...
## $ V4 : num -0.101 -0.1155 -0.0998 -0.058 3.6592 ...
## $ V5 : num 0.0639 0.0403 -0.0839 -0.042 -0.2469 ...
## $ V6 : num 0.013 0.0686 0.1197 -0.1803 -0.1964 ...
## $ Clusters : int 1 1 1 1 3 1 6 2 1 3 ...
## $ Q.Clust1 : num 0.95599 0.99995 0.99995 0.99995 0.00001 ...
## $ Q.Clust2 : num 1e-05 1e-05 1e-05 1e-05 1e-05 ...
## $ Q.Clust3 : num 0.00001 0.00001 0.00001 0.00001 0.79146 ...
## $ Q.Clust4 : num 0.04397 0.00001 0.00001 0.00001 0.00001 ...
## $ Q.Clust5 : num 1e-05 1e-05 1e-05 1e-05 1e-05 ...
## $ Q.Clust6 : num 0.00001 0.00001 0.00001 0.00001 0.2085 ...
## $ xy : num 3.82 10.95 16.92 2.02 10.6 ...
## $ xz : num 3.34 15.31 18.81 7.42 35.52 ...
## $ yz : num 11.26 3.7 9.29 22.18 3.83 ...
## $ prop : num 0.546 0.512 0.692 0.453 0.688 ...
## $ overhang : chr "No" "No" "No" "No" ...
## $ range : num 0.0666 0.0712 0.1161 0.0387 0.0204 ...
## $ x : num -0.392 0.105 -0.448 1.276 1.384 ...
## $ y : num 7.82 6.92 6.67 5.68 6.02 ...
## $ z : num -1.89 -1.6 -1.58 -1.87 -1.57 ...

metadata[, c(2:6, 13, 24)] <- data.frame(lapply(metadata[, c(2:6, 13, 24)], as.factor))
levels(metadata$Clusters) <- c("AA1", "AL2", "AH1", "AA2", "AL1", "AH2")
str(metadata)

## 'data.frame': 264 obs. of 28 variables:
## $ Individual: chr "KP0057_AC_SB10" "KP0058_AC_SB10" "KP0061_AC_SB10" "KP0072_AC_SB10" ...
## $ Species : Factor w/ 3 levels "AC","HU","LM": 1 1 1 1 2 1 1 3 1 1 ...
## $ Population: Factor w/ 12 levels "AC_SB10","AC_SB20",...: 1 1 1 1 6 1 1 9 1 1 ...
## $ Site : Factor w/ 5 levels "SB10","SB20",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ Loc : Factor w/ 2 levels "SB","WP": 1 1 1 1 1 1 1 1 1 1 ...
## $ Depth : Factor w/ 3 levels "5","10","20": 2 2 2 2 2 2 2 2 2 2 ...
```

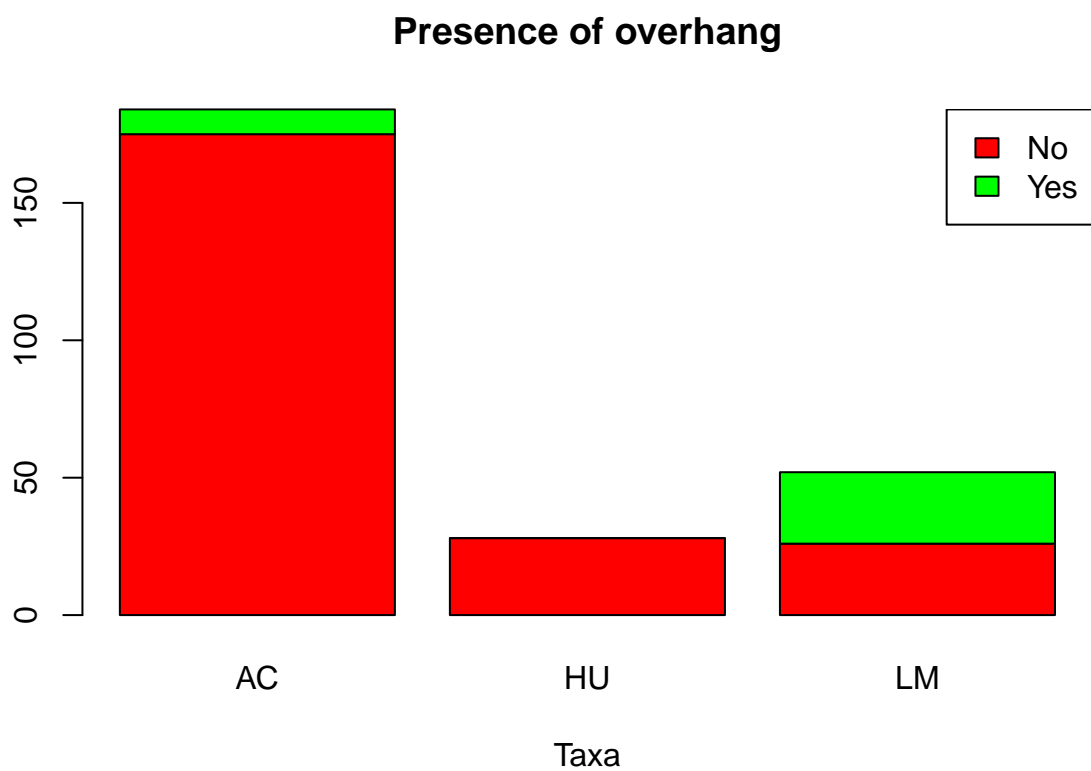
```
## $ V1      : num  16.563 16.62 16.536 16.564 0.663 ...
## $ V2      : num  -10.6 -10.8 -10.7 -10.5 28.5 ...
## $ V3      : num   1.288 1.792 0.485 0.736 0.41 ...
## $ V4      : num  -0.101 -0.1155 -0.0998 -0.058 3.6592 ...
## $ V5      : num   0.0639 0.0403 -0.0839 -0.042 -0.2469 ...
## $ V6      : num   0.013 0.0686 0.1197 -0.1803 -0.1964 ...
## $ Clusters : Factor w/ 6 levels "AA1","AL2","AH1",...: 1 1 1 1 3 1 6 2 1 3 ...
## $ Q.Clust1 : num   0.95599 0.99995 0.99995 0.99995 0.00001 ...
## $ Q.Clust2 : num   1e-05 1e-05 1e-05 1e-05 1e-05 ...
## $ Q.Clust3 : num   0.00001 0.00001 0.00001 0.00001 0.79146 ...
## $ Q.Clust4 : num   0.04397 0.00001 0.00001 0.00001 0.00001 ...
## $ Q.Clust5 : num   1e-05 1e-05 1e-05 1e-05 1e-05 ...
## $ Q.Clust6 : num   0.00001 0.00001 0.00001 0.00001 0.2085 ...
## $ xy       : num   3.82 10.95 16.92 2.02 10.6 ...
## $ xz       : num   3.34 15.31 18.81 7.42 35.52 ...
## $ yz       : num  11.26 3.7 9.29 22.18 3.83 ...
## $ prop     : num   0.546 0.512 0.692 0.453 0.688 ...
## $ overhang : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
## $ range    : num   0.0666 0.0712 0.1161 0.0387 0.0204 ...
## $ x        : num  -0.392 0.105 -0.448 1.276 1.384 ...
## $ y        : num   7.82 6.92 6.67 5.68 6.02 ...
## $ z        : num  -1.89 -1.6 -1.58 -1.87 -1.57 ...
```

```
# choosing steepest angle
metadata$theta <- NA
for (row in 1:length(metadata$xz)) {
  if (is.na(metadata$xz[row])) {
    next
  } else if (is.na(metadata$yz[row])) {
    next
  } else if (metadata$xz[row] > metadata$yz[row]) {
    metadata$theta[row] <- metadata$xz[row]
  } else {
    metadata$theta[row] <- metadata$yz[row]
  }
}
```

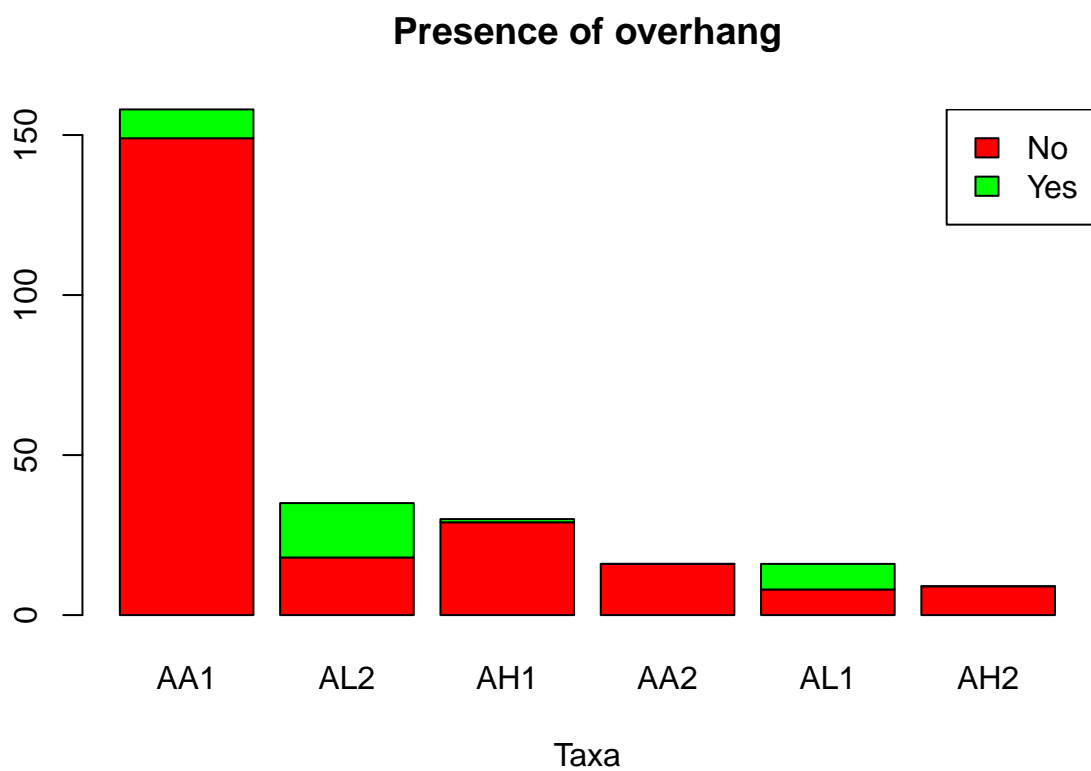
Environmental data plots

Overhang - presence and absense at the moment but will be % cover.

```
overhang_barplot(metadata, metadata$Species) # Species
```

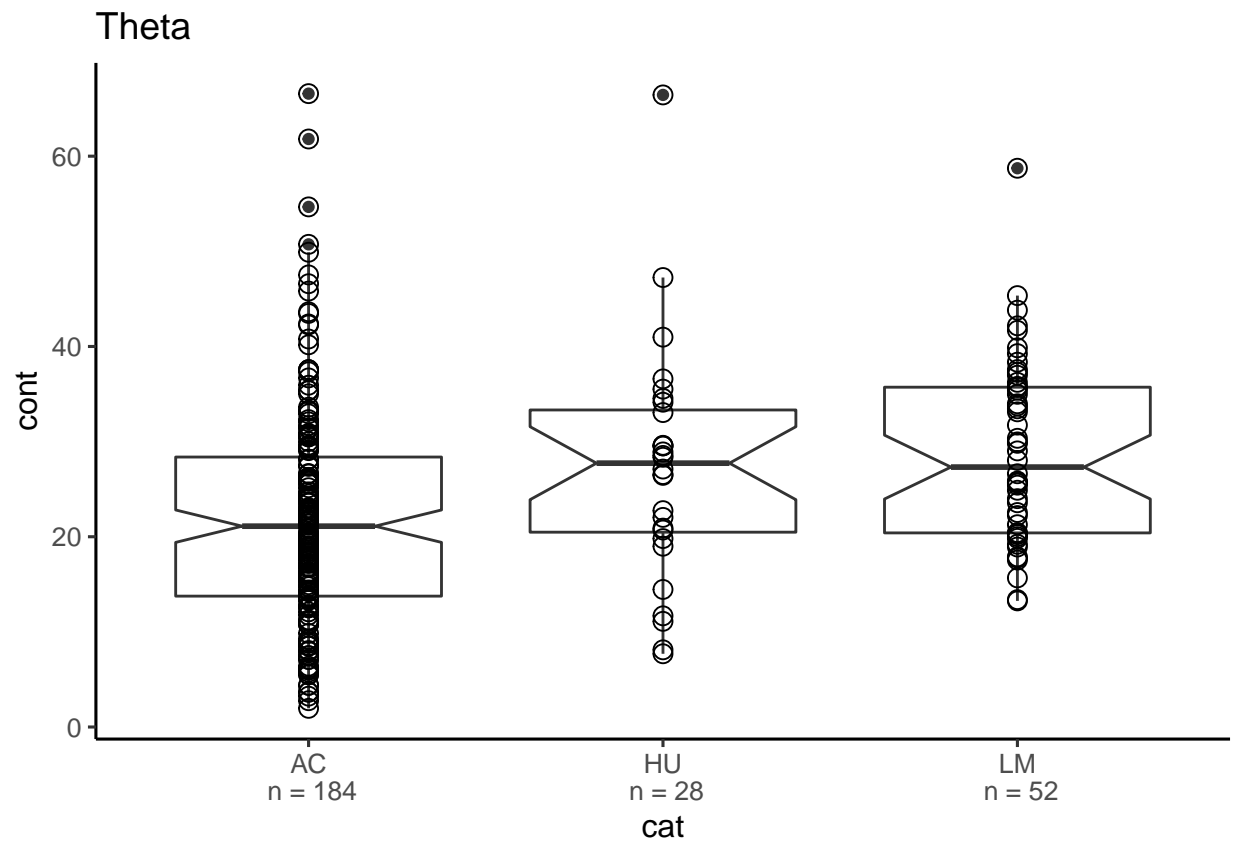


```
overhang_barplot(metadata, metadata$Clusters) # Cluster
```



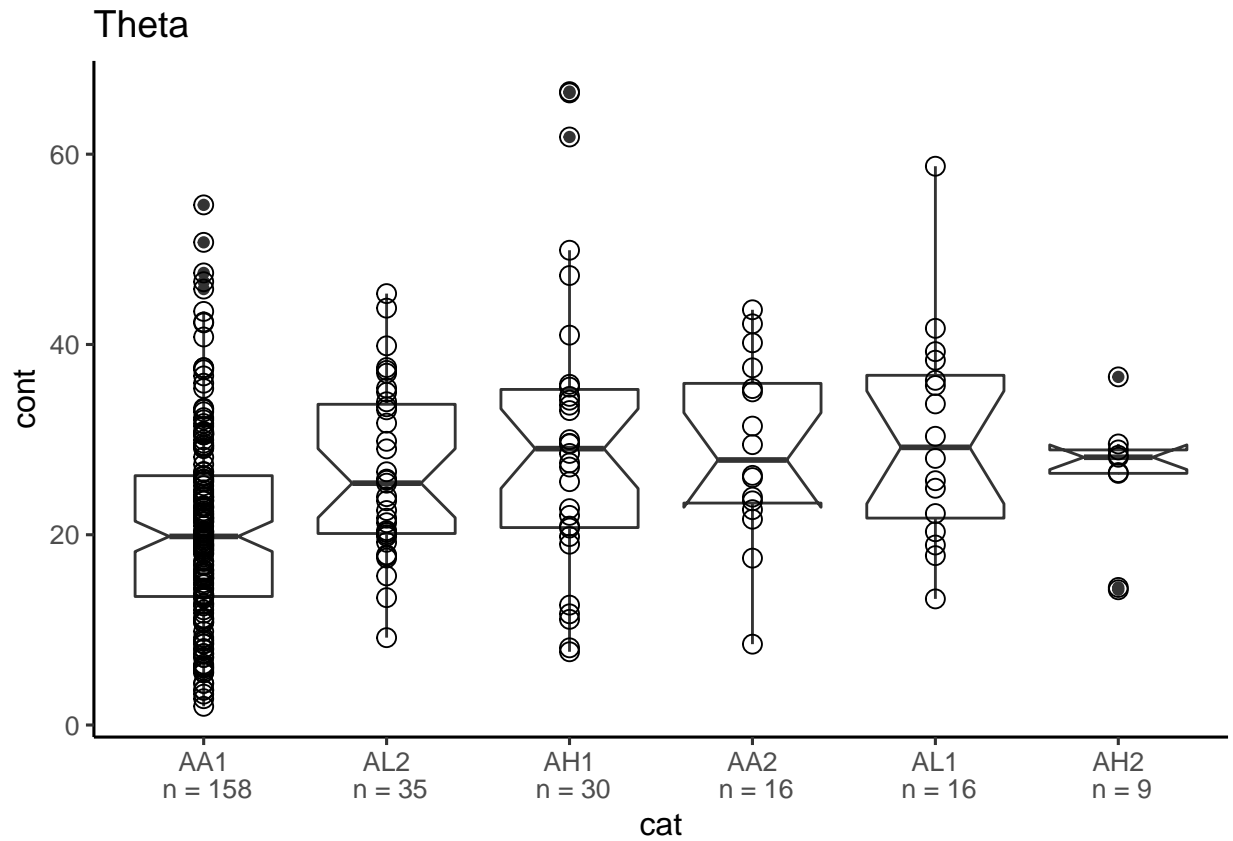
Theta - angle of attachment. Might have interaction with overhang and colony rugosity.

```
k_boxplot(metadata, metadata$Species, metadata$theta) + ggtitle('Theta')
```



```
k_boxplot(metadata, metadata$Clusters, metadata$theta) + ggtitle('Theta')
```

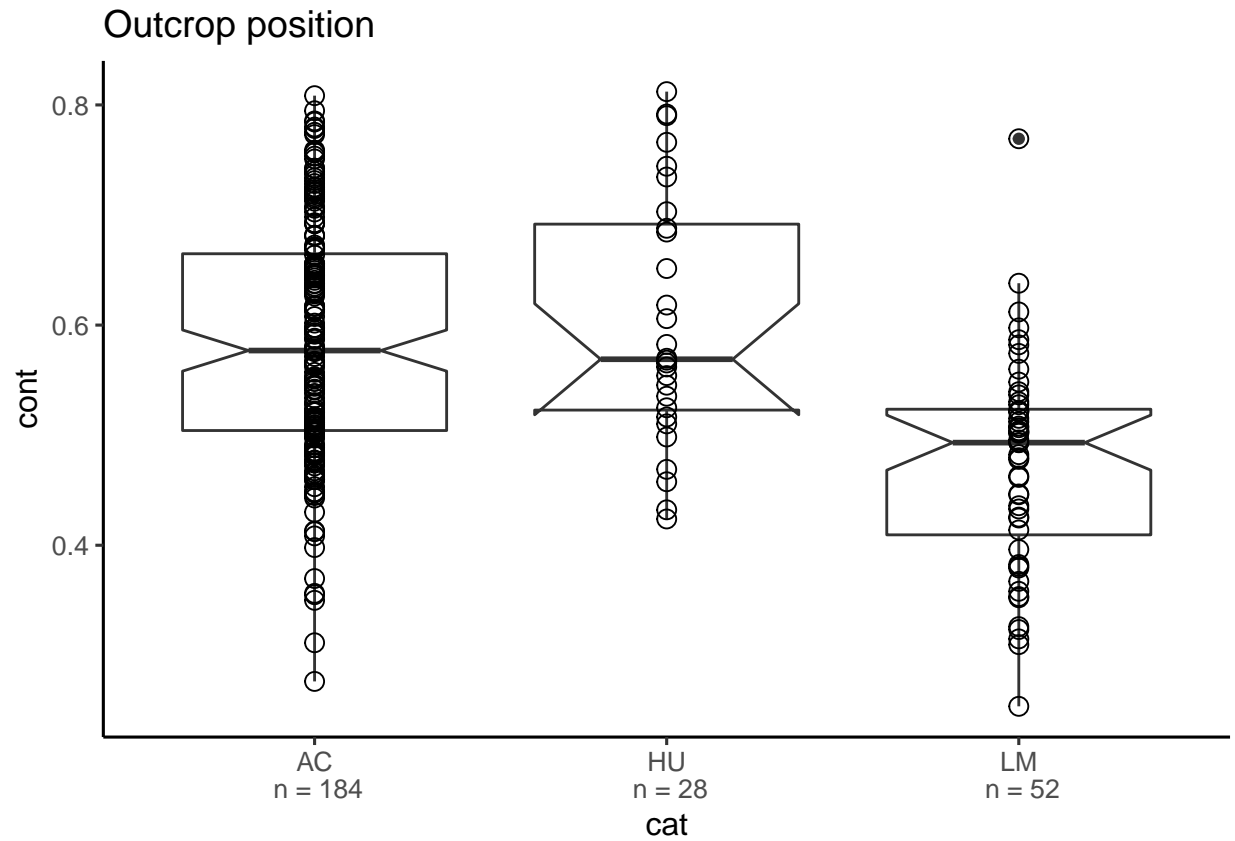
```
## notch went outside hinges. Try setting notch=FALSE.  
## notch went outside hinges. Try setting notch=FALSE.
```



Outcrop position - proportion of height the colony sits at in relation to it's local environment.

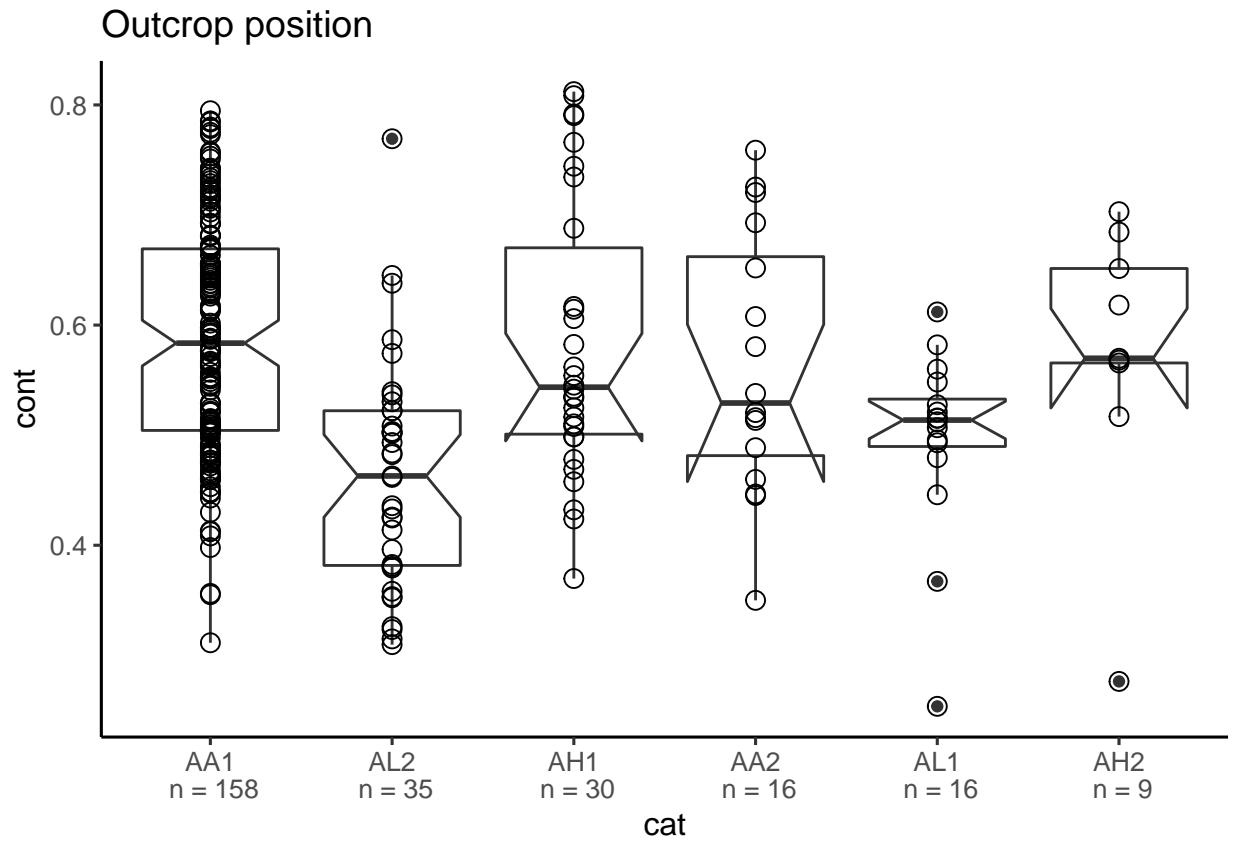
```
k_boxplot(metadata, metadata$Species, metadata$prop) + ggtitle('Outcrop position')
```

```
## notch went outside hinges. Try setting notch=FALSE.
```

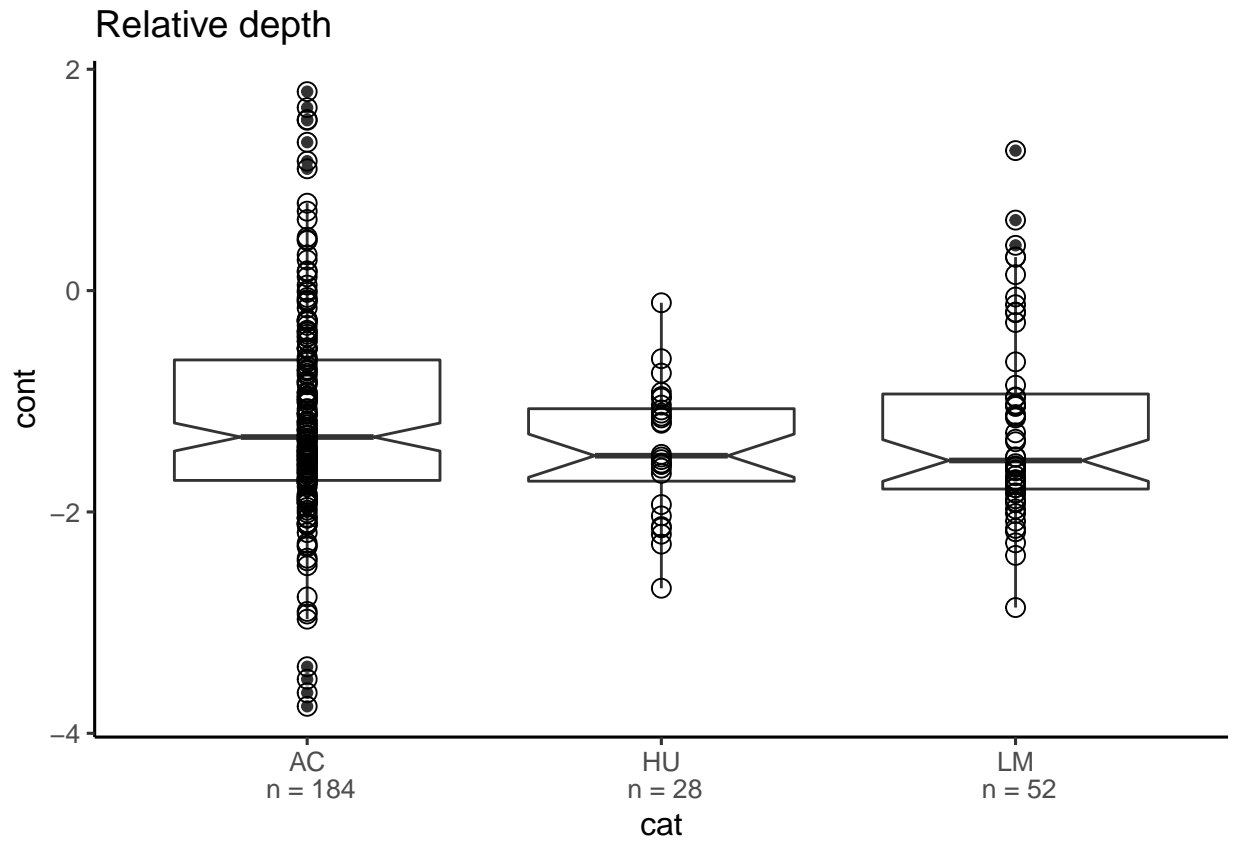
```
k_boxplot(metadata, metadata$Clusters, metadata$prop) + ggtitle('Outcrop position')
```

```
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
```



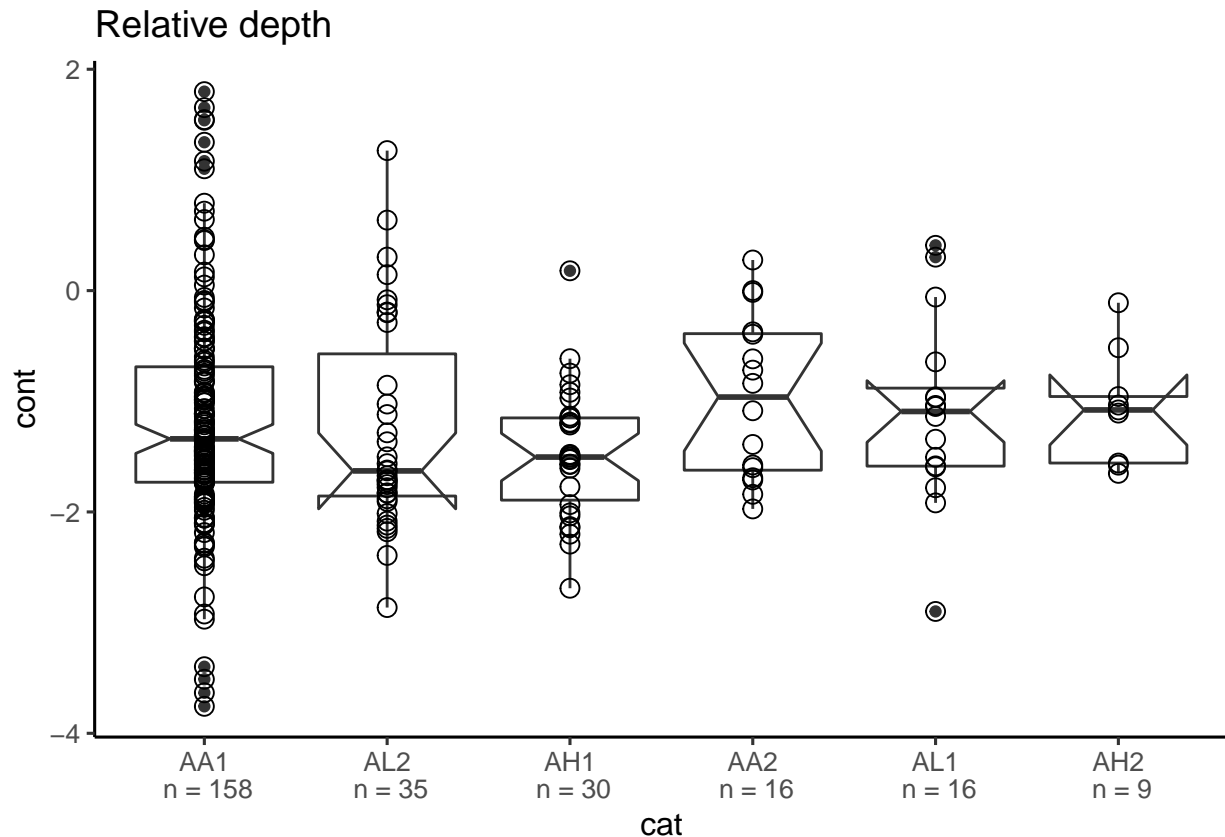
Raw depth, local depth measures within plots. Comparisons between plots don't make sense but comparisons with are real depth differences.

```
k_boxplot(metadata, metadata$Species, metadata$z) + ggtitle('Relative depth')
```



```
k_boxplot(metadata, metadata$Clusters, metadata$z) + ggtitle('Relative depth')
```

```
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
```



Linear models

Overhang

Presence and absence thus a binomial family.

```
# 1. overhang #### will become overhang %
metadata$overhang.bin <- ifelse(metadata$overhang == "Yes", 1, 0)
glmer1.overhang <- glmer(overhang ~ Clusters + (1|Loc/Depth), data = metadata, family = binomial) # won
```

```
## boundary (singular) fit: see ?isSingular
```

```
# boundary (singular) fit: see ?isSingular - means the random effects don't use too much variation
summary(glmer1.overhang)
```

```
## Warning in vcov.merMod(object, use.hessian = use.hessian): variance-covariance matrix computed from Hessian is
## not positive definite or contains NA values: falling back to var-cov estimated from RX
```

```
## Warning in vcov.merMod(object, correlation = correlation, sigma = sigma): variance-covariance matrix computed from Hessian is
## not positive definite or contains NA values: falling back to var-cov estimated from RX
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace)
```

```
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: overhang ~ Clusters + (1 | Loc/Depth)
## Data: metadata
##
##      AIC      BIC    logLik deviance df.resid
##    164.5    193.1    -74.2    148.5      256
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.0000 -0.2458 -0.2458 -0.1857  5.3852
##
## Random effects:
## Groups      Name                Variance Std.Dev.
## Depth:Loc (Intercept) 6.47e-16 2.544e-08
## Loc      (Intercept) 0.00e+00 0.000e+00
## Number of obs: 264, groups: Depth:Loc, 5; Loc, 2
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.807e+00  3.433e-01  -8.177 2.91e-16 ***
## ClustersAL2  2.750e+00  4.819e-01   5.706 1.16e-08 ***
## ClustersAH1 -5.606e-01  1.073e+00  -0.522  0.602
## ClustersAA2 -3.743e+01  1.678e+07   0.000  1.000
## ClustersAL1  2.807e+00  6.065e-01   4.628 3.69e-06 ***
## ClustersAH2 -2.849e+01  2.087e+06   0.000  1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) ClsAL2 ClsAH1 ClsAA2 ClsAL1
## ClustersAL2 -0.712
## ClustersAH1 -0.320  0.228
## ClustersAA2  0.000  0.000  0.000
## ClustersAL1 -0.566  0.403  0.181  0.000
## ClustersAH2  0.000  0.000  0.000  0.000  0.000
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

#Warning messages:

```
# 1: In vcov.merMod(object, use.hessian = use.hessian) :
# variance-covariance matrix computed from finite-difference Hessian is
#not positive definite or contains NA values: falling back to var-cov estimated from RX
#2: In vcov.merMod(object, correlation = correlation, sigm = sig) :
# variance-covariance matrix computed from finite-difference Hessian is
#not positive definite or contains NA values: falling back to var-cov estimated from RX
# ClustersAL1  2.807e+00  6.065e-01   4.628 3.69e-06 ***
# ClustersAL2  2.750e+00  4.819e-01   5.706 1.16e-08 ***
```

```
emmeans(glmer1.overhang, list(pairwise ~ Clusters), adjust = "tukey")
```

```
## Warning in vcov.merMod(object, correlation = FALSE): variance-covariance matrix computed from finite-
## not positive definite or contains NA values: falling back to var-cov estimated from RX
```

```
## $`emmeans of Clusters`
## Clusters      emmean      SE df asymp.LCL asymp.UCL
## AA1          -2.8067      0 Inf      -3      -2
## AL2          -0.0572      0 Inf      -1       1
## AH1          -3.3673      1 Inf      -5      -1
## AA2          -40.2376 16777216 Inf -32882779 32882699
## AL1           0.0000      1 Inf      -1       1
## AH2          -31.2994 2086724 Inf -4089935 4089873
##
## Results are given on the logit (not the response) scale.
## Confidence level used: 0.95
##
## $`pairwise differences of Clusters`
## 1      estimate      SE df z.ratio p.value
## AA1 - AL2  -2.7496      0 Inf -5.706 <.0001
## AA1 - AH1   0.5606      1 Inf  0.522 0.9953
## AA1 - AA2  37.4309 16777216 Inf  0.000 1.0000
## AA1 - AL1  -2.8067      1 Inf -4.628 0.0001
## AA1 - AH2  28.4927 2086724 Inf  0.000 1.0000
## AL2 - AH1   3.3101      1 Inf  3.088 0.0246
## AL2 - AA2  40.1804 16777216 Inf  0.000 1.0000
## AL2 - AL1  -0.0572      1 Inf -0.095 1.0000
## AL2 - AH2  31.2423 2086724 Inf  0.000 1.0000
## AH1 - AA2  36.8703 16777216 Inf  0.000 1.0000
## AH1 - AL1  -3.3673      1 Inf -2.971 0.0352
## AH1 - AH2  27.9321 2086724 Inf  0.000 1.0000
## AA2 - AL1 -40.2376 16777216 Inf  0.000 1.0000
## AA2 - AH2  -8.9382 16906490 Inf  0.000 1.0000
## AL1 - AH2  31.2994 2086724 Inf  0.000 1.0000
##
## Results are given on the log odds ratio (not the response) scale.
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```
#AA1 - AL2  -2.7496      0 Inf -5.706 <.0001 ***
#AL2 - AH1   3.3101      1 Inf  3.088 0.0246 *
#AH1 - AL1  -3.3673      1 Inf -2.971 0.0352 *

# probability space with confidence interval?
```

Theta - angle of attachment

```
shapiro.test(metadata$theta) # not normal
```

```
##
## Shapiro-Wilk normality test
##
## data:  metadata$theta
## W = 0.96897, p-value = 1.72e-05
```

Linear model without mixed effects.

```
lm.theta <- lm(theta ~ Clusters, data = metadata)
summary(lm.theta) # all diff to AA1 except AH2 :D
```

```
##
## Call:
## lm(formula = theta ~ Clusters, data = metadata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.768  -7.267  -0.864   5.919  36.109
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20.6926     0.8716   23.741 < 2e-16 ***
## ClustersAL2    5.8931     2.0467    2.879 0.004319 **
## ClustersAH1    9.7738     2.1818    4.480 1.12e-05 ***
## ClustersAA2    8.3621     2.8742    2.909 0.003938 **
## ClustersAL1    9.6290     2.8742    3.350 0.000929 ***
## ClustersAH2    5.2203     3.7544    1.390 0.165591
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.96 on 258 degrees of freedom
## Multiple R-squared:  0.1217, Adjusted R-squared:  0.1047
## F-statistic: 7.152 on 5 and 258 DF, p-value: 2.754e-06
```

```
summary(aov(lm.theta))
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Clusters      5  4292   858.4    7.152 2.75e-06 ***
## Residuals    258 30966   120.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(aov(lm.theta))
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = lm.theta)
##
## $Clusters
##              diff              lwr              upr              p adj
## AL2-AA1  5.8931198  0.01654314 11.769696 0.0489002
## AH1-AA1  9.7738198  3.50915167 16.038488 0.0001624
## AA2-AA1  8.3621034  0.10943483 16.614772 0.0449977
## AL1-AA1  9.6290353  1.37636675 17.881704 0.0118319
## AH2-AA1  5.2203349 -5.55961517 16.000285 0.7327936
## AH1-AL2  3.8807000 -3.94584960 11.707250 0.7126074
## AA2-AL2  2.4689836 -7.02392692 11.961894 0.9757809
## AL1-AL2  3.7359155 -5.75699499 13.228826 0.8685473
## AH2-AL2 -0.6727848 -12.42932073 11.083751 0.9999832
```

```
## AA2-AH1 -1.4117164 -11.14964555 8.326213 0.9983897
## AL1-AH1 -0.1447845 -9.88271362 9.593145 1.0000000
## AH2-AH1 -4.5534848 -16.50873684 7.401767 0.8836473
## AL1-AA2 1.2669319 -9.85456689 12.388431 0.9994993
## AH2-AA2 -3.1417684 -16.24858049 9.965044 0.9831420
## AH2-AL1 -4.4087004 -17.51551242 8.698112 0.9282997
```

```
#AL2-AA1 5.8931198 0.01654314 11.769696 0.0489002 *
#AH1-AA1 9.7738198 3.50915167 16.038488 0.0001624 ***
#AA2-AA1 8.3621034 0.10943483 16.614772 0.0449977 *
#AL1-AA1 9.6290353 1.37636675 17.881704 0.0118319 *
```

Linear model 1 - with mixed effects.

```
# mixed model
lmer1.theta <- lmer(theta ~ Clusters + (1|Loc/Depth), data = metadata)
```

```
## boundary (singular) fit: see ?isSingular
```

```
# boundary (singular) fit: see ?isSingular - means that the random effects don't use too much variation
summary(lmer1.theta)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: theta ~ Clusters + (1 | Loc/Depth)
## Data: metadata
##
## REML criterion at convergence: 1987.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.0782 -0.6633 -0.0788  0.5403  3.2960
##
## Random effects:
## Groups Name Variance Std.Dev.
## Depth:Loc (Intercept) 0 0.00
## Loc (Intercept) 0 0.00
## Residual 120 10.96
## Number of obs: 264, groups: Depth:Loc, 5; Loc, 2
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 20.6926 0.8716 23.741
## ClustersAL2 5.8931 2.0467 2.879
## ClustersAH1 9.7738 2.1818 4.480
## ClustersAA2 8.3621 2.8742 2.909
## ClustersAL1 9.6290 2.8742 3.350
## ClustersAH2 5.2203 3.7544 1.390
##
## Correlation of Fixed Effects:
## (Intr) ClsAL2 ClsAH1 ClsAA2 ClsAL1
## ClustersAL2 -0.426
## ClustersAH1 -0.399 0.170
```



```
## ClustersAA2 -0.303  0.129  0.121
## ClustersAL1 -0.303  0.129  0.121  0.092
## ClustersAH2 -0.232  0.099  0.093  0.070  0.070
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

```
# extract coefficients
coefs <- data.frame(coef(summary(lmer1.theta)))
# use normal distribution to approximate p-value
coefs$p.z <- 2 * (1 - pnorm(abs(coefs$t.value)))
coefs
```

	Estimate	Std..Error	t.value	p.z
## (Intercept)	20.692570	0.8715783	23.741493	0.000000e+00
## ClustersAL2	5.893120	2.0466851	2.879348	3.984977e-03
## ClustersAH1	9.773820	2.1818490	4.479604	7.478164e-06
## ClustersAA2	8.362103	2.8742268	2.909340	3.621926e-03
## ClustersAL1	9.629035	2.8742268	3.350131	8.077345e-04
## ClustersAH2	5.220335	3.7544246	1.390449	1.643927e-01

```
# post-hoc
emmeans(lmer1.theta, list(pairwise ~ Clusters), adjust = "tukey")
```

```
## $`emmeans of Clusters`
## Clusters emmean SE df lower.CL upper.CL
## AA1 20.7 1.07 1.55 14.6 26.8
## AL2 26.6 1.97 32.81 22.6 30.6
## AH1 30.5 2.36 20.64 25.6 35.4
## AA2 29.1 3.10 65.75 22.9 35.2
## AL1 30.3 2.87 102.58 24.6 36.0
## AH2 25.9 3.94 128.69 18.1 33.7
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
##
## $`pairwise differences of Clusters`
## 1 estimate SE df t.ratio p.value
## AA1 - AL2 -5.893 2.15 237 -2.737 0.0718
## AA1 - AH1 -9.774 2.41 164 -4.054 0.0011
## AA1 - AA2 -8.362 3.02 232 -2.771 0.0659
## AA1 - AL1 -9.629 2.94 249 -3.274 0.0152
## AA1 - AH2 -5.220 3.99 159 -1.308 0.7802
## AL2 - AH1 -3.881 3.15 117 -1.231 0.8206
## AL2 - AA2 -2.469 3.51 237 -0.703 0.9814
## AL2 - AL1 -3.736 3.32 256 -1.125 0.8705
## AL2 - AH2 0.673 4.48 144 0.150 1.0000
## AH1 - AA2 1.412 3.76 104 0.375 0.9990
## AH1 - AL1 0.145 3.74 151 0.039 1.0000
## AH1 - AH2 4.553 4.21 249 1.081 0.8886
## AA2 - AL1 -1.267 3.98 256 -0.318 0.9996
## AA2 - AH2 3.142 4.93 114 0.638 0.9879
## AL1 - AH2 4.409 4.91 151 0.899 0.9463
##
```

```
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```
#AA1 - AL2    -5.893 2.15 237 -2.737 0.0718
#AA1 - AH1    -9.774 2.41 164 -4.054 0.0011 **
#AA1 - AA2    -8.362 3.02 232 -2.771 0.0659
#AA1 - AL1    -9.629 2.94 249 -3.274 0.0152 *
```

Linear model 2 - with mixed effects.

```
# mixed model 2
lmer2.theta <- lmer(theta ~ Clusters + (Clusters|Loc/Depth), data = metadata)
```

```
## boundary (singular) fit: see ?isSingular
```

```
# boundary (singular) fit: see ?isSingular - means that the random effects don't use too much variation
summary(lmer2.theta)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: theta ~ Clusters + (Clusters | Loc/Depth)
## Data: metadata
##
## REML criterion at convergence: 1972
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.6471 -0.6252 -0.0699  0.5681  3.5951
##
## Random effects:
## Groups      Name                Variance Std.Dev. Corr
## Depth:Loc (Intercept) 5.861e+00  2.4209
##              ClustersAL2 1.952e+01  4.4182 -1.00
##              ClustersAH1 7.436e+01  8.6235 -1.00  1.00
##              ClustersAA2 4.378e+01  6.6169 -1.00  1.00  1.00
##              ClustersAL1 3.688e+01  6.0728 -1.00  1.00  1.00  1.00
##              ClustersAH2 4.748e-03  0.0689  1.00 -1.00 -1.00 -1.00 -1.00
## Loc (Intercept) 6.566e+00  2.5624
##              ClustersAL2 5.742e+00  2.3962 -1.00
##              ClustersAH1 2.779e+01  5.2712 -1.00  1.00
##              ClustersAA2 1.020e+02 10.0998 -1.00  1.00  1.00
##              ClustersAL1 1.100e+02 10.4866 -1.00  1.00  1.00  1.00
##              ClustersAH2 2.210e+01  4.7014  1.00 -1.00 -1.00 -1.00 -1.00
## Residual          1.096e+02 10.4707
## Number of obs: 264, groups: Depth:Loc, 5; Loc, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 20.858075  2.291482  9.102
## ClustersAL2  4.861974  3.337676  1.457
## ClustersAH1 13.257667  5.825082  2.276
## ClustersAA2  7.475353  8.384737  0.892
## ClustersAL1  8.019348  8.456738  0.948
```

```
## ClustersAH2 -0.002366 5.123218 0.000
##
## Correlation of Fixed Effects:
##      (Intr) ClsAL2 ClsAH1 ClsAA2 ClsAL1
## ClustersAL2 -0.800
## ClustersAH1 -0.871 0.744
## ClustersAA2 -0.904 0.707 0.792
## ClustersAL1 -0.900 0.693 0.783 0.898
## ClustersAH2 0.453 -0.279 -0.409 -0.544 -0.549
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

```
# extract coefficients
coefs <- data.frame(coef(summary(lmer2.theta)))
# use normal distribution to approximate p-value
coefs$p.z <- 2 * (1 - pnorm(abs(coefs$t.value)))
coefs
```

	Estimate	Std..Error	t.value	p.z
## (Intercept)	20.858075456	2.291482	9.1024380522	0.00000000
## ClustersAL2	4.861974310	3.337676	1.4566947141	0.14520067
## ClustersAH1	13.257666710	5.825082	2.2759623126	0.02284827
## ClustersAA2	7.475353198	8.384737	0.8915429619	0.37263795
## ClustersAL1	8.019347530	8.456738	0.9482790874	0.34298739
## ClustersAH2	-0.002366269	5.123218	-0.0004618716	0.99963148

```
# post-hoc
emmeans(lmer2.theta, list(pairwise ~ Clusters), adjust = "tukey")
```

```
## Warning in ptukey(sqrt(2) * abst, fam.size, zapsmall(df), lower.tail = FALSE):
## NaNs produced
```

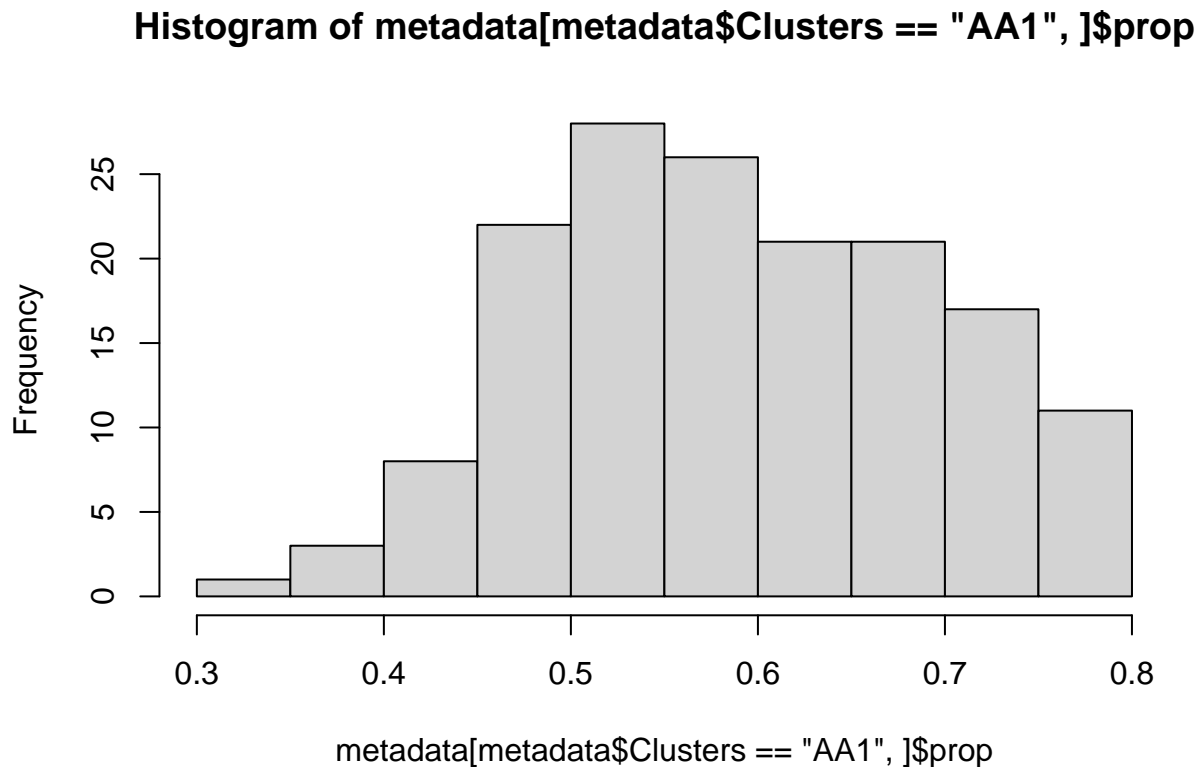
```
## $`emmeans of Clusters`
## Clusters emmean SE df lower.CL upper.CL
## AA1 20.9 2.74 0.95 -18.4 60.1
## AL2 25.7 4.01 0.99 -26.1 77.5
## AH1 34.1 8.93 0.47 -2098.3 2166.5
## AA2 28.3 13.70 0.78 -315.1 371.7
## AL1 28.9 9.96 0.99 -100.7 158.4
## AH2 20.9 18.48 0.36 -27209.8 27251.5
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
##
## $`pairwise differences of Clusters`
## 1 estimate SE df t.ratio p.value
## AA1 - AL2 -4.86197 5.16 0.96 -0.943 NaN
## AA1 - AH1 -13.25767 10.18 0.69 -1.302 NaN
## AA1 - AA2 -7.47535 14.67 0.87 -0.510 NaN
## AA1 - AL1 -8.01935 11.43 0.99 -0.702 NaN
## AA1 - AH2 0.00237 18.19 0.20 0.000 NaN
## AL2 - AH1 -8.39569 9.38 0.50 -0.895 NaN
## AL2 - AA2 -2.61338 14.08 0.80 -0.186 NaN
```

```
## AL2 - AL1 -3.15737 10.60 0.99 -0.298 NaN
## AL2 - AH2 4.86434 18.87 0.40 0.258 NaN
## AH1 - AA2 5.78231 15.36 0.43 0.376 NaN
## AH1 - AL1 5.23832 12.06 0.67 0.434 NaN
## AH1 - AH2 13.26003 21.35 0.51 0.621 NaN
## AA2 - AL1 -0.54399 14.54 0.43 -0.037 NaN
## AA2 - AH2 7.47772 23.38 0.72 0.320 NaN
## AL1 - AH2 8.02171 21.89 0.76 0.366 NaN
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```
# Warning message:
# In ptukey(sqrt(2) * abst, fam.size, zapsmall(df), lower.tail = FALSE) :
# NaNs produced
```

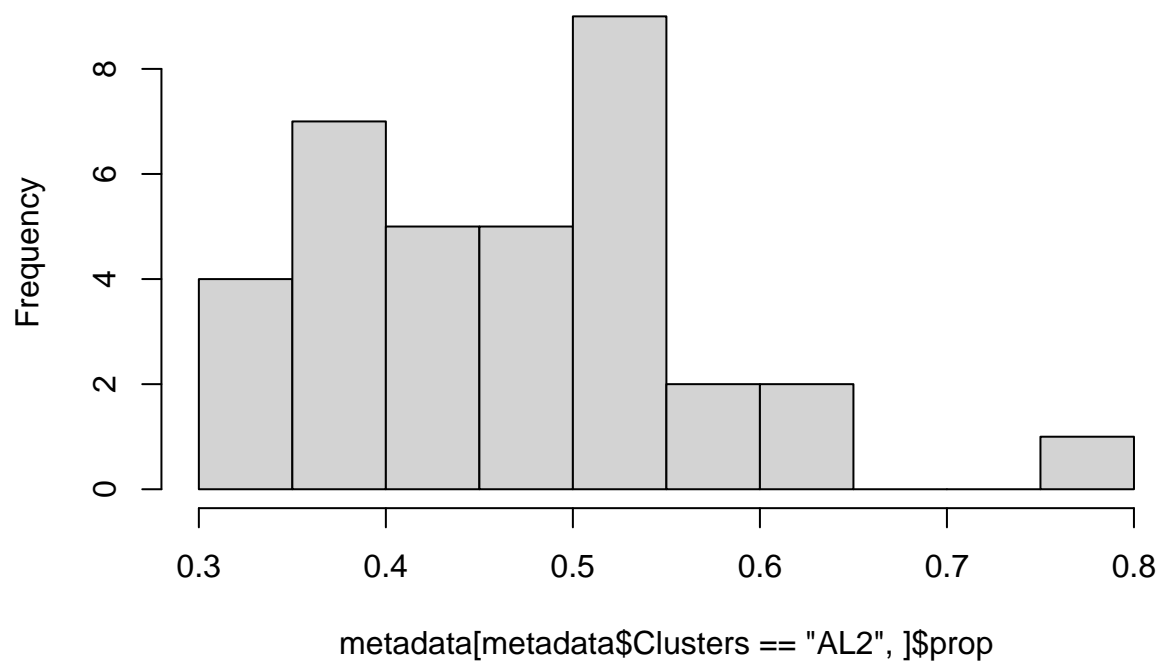
Outcrop position

```
hist(metadata[metadata$Clusters == "AA1",]$prop)
```



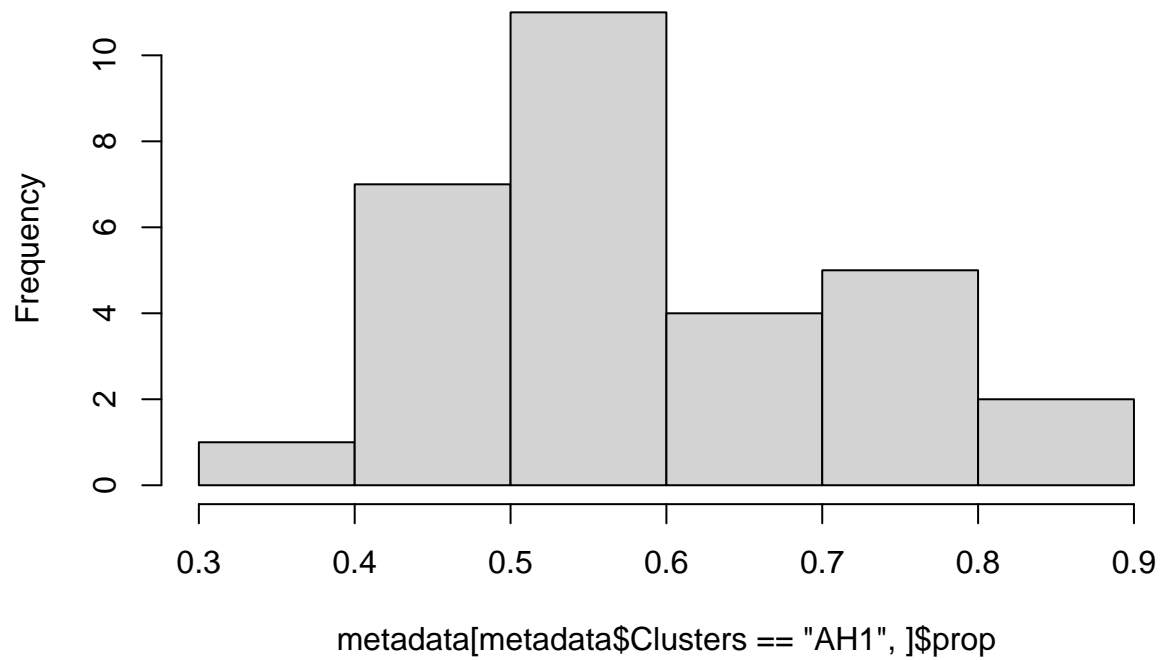
```
hist(metadata[metadata$Clusters == "AL2",]$prop)
```

Histogram of metadata[metadata\$Clusters == "AL2",]\$prop



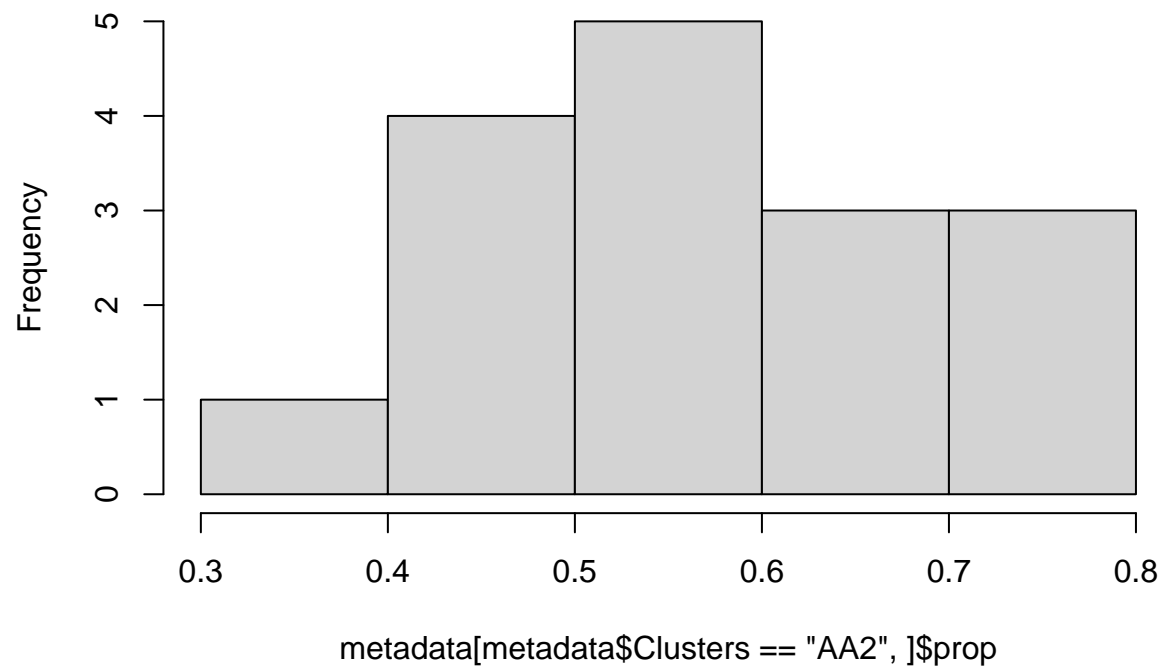
```
hist(metadata[metadata$Clusters == "AH1", ]$prop)
```

Histogram of metadata[metadata\$Clusters == "AH1",]\$prop



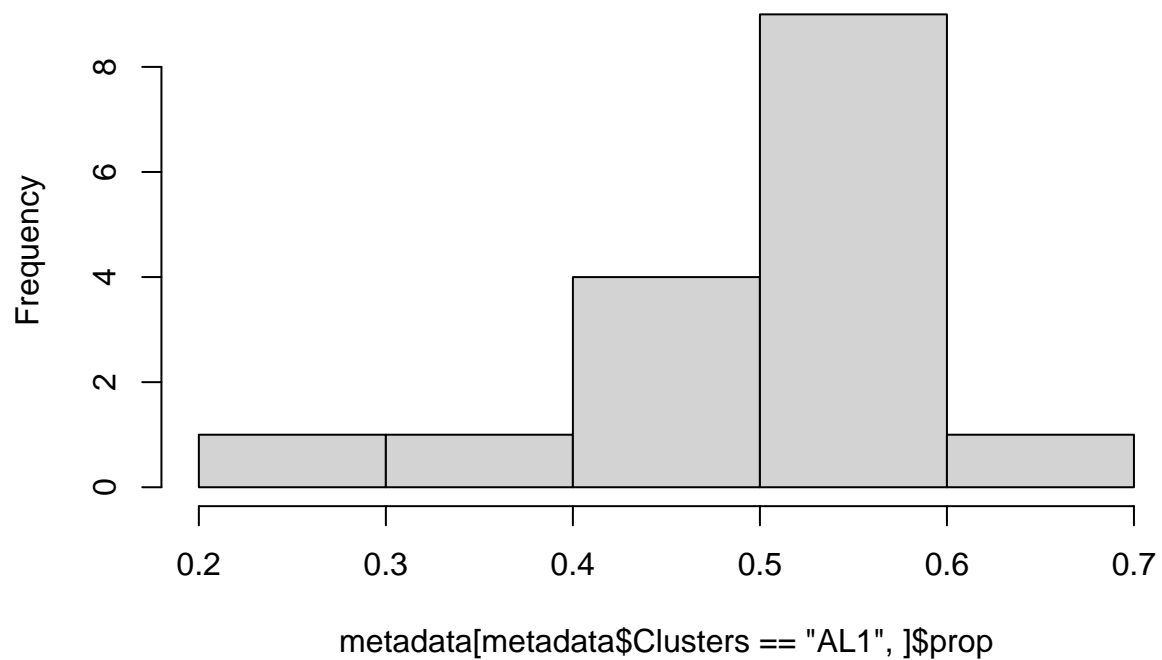
```
hist(metadata[metadata$Clusters == "AA2", ]$prop)
```

Histogram of metadata[metadata\$Clusters == "AA2",]\$prop



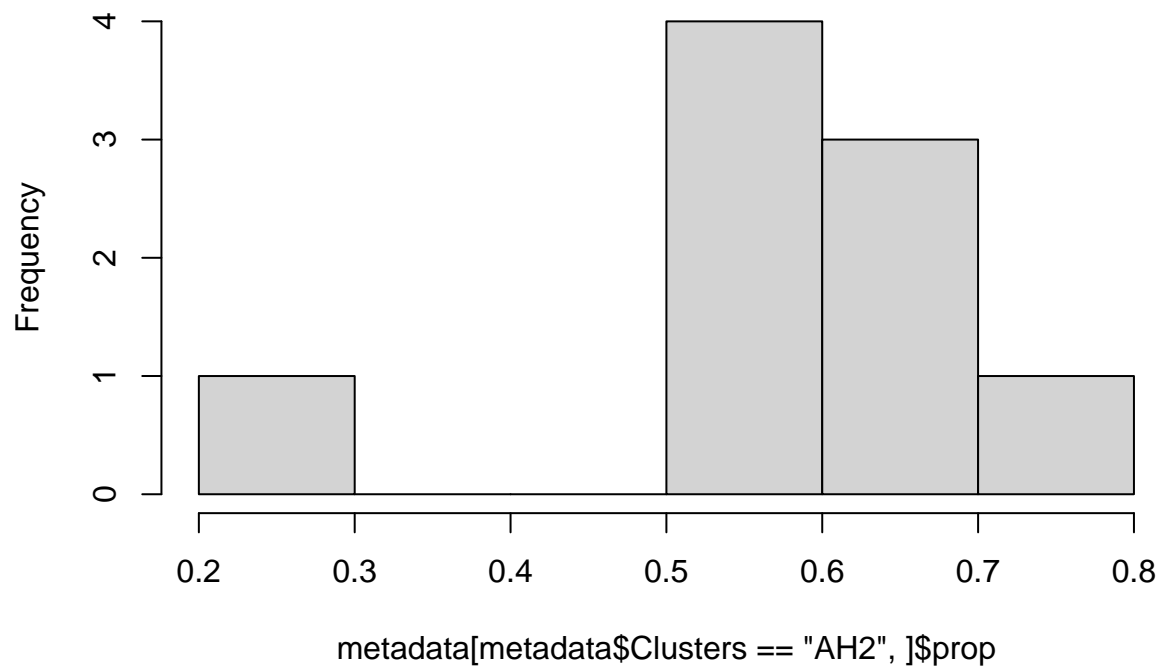
```
hist(metadata[metadata$Clusters == "AL1", ]$prop)
```

Histogram of metadata[metadata\$Clusters == "AL1",]\$prop



```
hist(metadata[metadata$Clusters == "AH2", ]$prop)
```

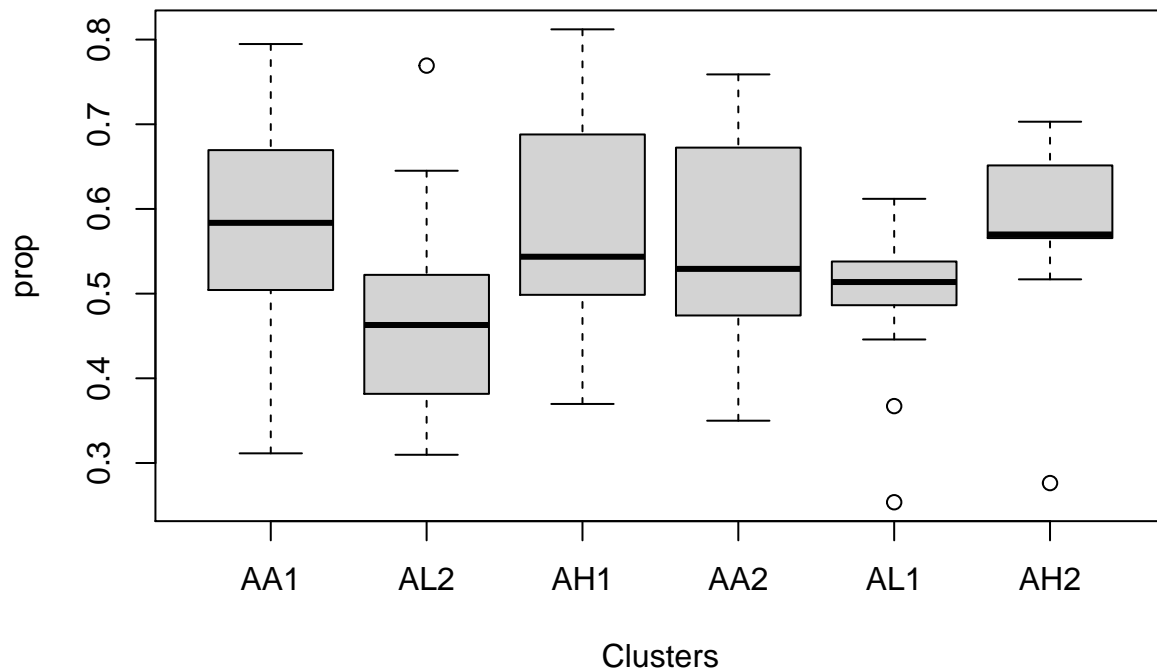

Histogram of metadata[metadata\$Clusters == "AH2",]\$prop



sort of normally distributed

Boxplot

```
boxplot(prop ~ Clusters, data = metadata)
```



Linear model

```
lm.prop <- lm(prop ~ Clusters, data = metadata)
summary(lm.prop) # only AL1 and AL2 diff to AA1
```

```
##
## Call:
## lm(formula = prop ~ Clusters, data = metadata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.296354 -0.082646 -0.002843  0.072356  0.304312
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.587928   0.008599  68.370 < 2e-16 ***
## ClustersAL2 -0.122971   0.020193  -6.090 4.08e-09 ***
## ClustersAH1 -0.004123   0.021527  -0.192  0.84827
## ClustersAA2 -0.024452   0.028358  -0.862  0.38935
## ClustersAL1 -0.091953   0.028358  -3.243  0.00134 **
## ClustersAH2 -0.015284   0.037042  -0.413  0.68023
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1081 on 258 degrees of freedom
## Multiple R-squared:  0.1472, Adjusted R-squared:  0.1307
```

```
## F-statistic: 8.905 on 5 and 258 DF, p-value: 8.117e-08
```

```
summary(aov(lm.prop))
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Clusters      5  0.5202  0.10405    8.905 8.12e-08 ***
## Residuals    258  3.0143  0.01168
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(aov(lm.prop))
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = lm.prop)
##
## $Clusters
##              diff              lwr              upr              p adj
## AL2-AA1 -0.122970975 -0.180950729 -0.064991220 0.0000001
## AH1-AA1 -0.004122691 -0.065931452  0.057686070 0.9999641
## AA2-AA1 -0.024451650 -0.105874514  0.056971213 0.9550322
## AL1-AA1 -0.091952545 -0.173375409 -0.010529682 0.0167064
## AH2-AA1 -0.015284147 -0.121641793  0.091073500 0.9984561
## AH1-AL2  0.118848284  0.041629615  0.196066952 0.0002101
## AA2-AL2  0.098519324  0.004859928  0.192178721 0.0327950
## AL1-AL2  0.031018429 -0.062640967  0.124677826 0.9326510
## AH2-AL2  0.107686828 -0.008306055  0.223679711 0.0858314
## AA2-AH1 -0.020328960 -0.116405770  0.075747851 0.9904322
## AL1-AH1 -0.087829854 -0.183906665  0.008246956 0.0949951
## AH2-AH1 -0.011161456 -0.129114921  0.106792009 0.9997983
## AL1-AA2 -0.067500895 -0.177228345  0.042226555 0.4894185
## AH2-AA2  0.009167504 -0.120147536  0.138482544 0.9999515
## AH2-AL1  0.076668399 -0.052646642  0.205983439 0.5314969
```

```
# AL2-AA1 -0.122970975 -0.180950729 -0.064991220 0.0000001 ***
# AL1-AA1 -0.091952545 -0.173375409 -0.010529682 0.0167064 *
# AH1-AL2  0.118848284  0.041629615  0.196066952 0.0002101 ***
# AA2-AL2  0.098519324  0.004859928  0.192178721 0.0327950 *
```

Mixed effect

```
lmer1.prop <- lmer(prop ~ Clusters + (1|Loc/Depth), data = metadata )
```

```
## boundary (singular) fit: see ?isSingular
```

```
summary(lmer1.prop)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: prop ~ Clusters + (1 | Loc/Depth)
```

```

## Data: metadata
##
## REML criterion at convergence: -397.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.84734 -0.80341 -0.04555  0.69878  2.76218
##
## Random effects:
##   Groups      Name      Variance Std.Dev.
## Depth:Loc (Intercept) 0.0002457 0.01568
## Loc      (Intercept) 0.0000000 0.00000
## Residual                0.0115107 0.10729
## Number of obs: 264, groups: Depth:Loc, 5; Loc, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  0.5881997  0.0114419  51.408
## ClustersAL2 -0.1228823  0.0203151  -6.049
## ClustersAH1 -0.0006249  0.0220327  -0.028
## ClustersAA2 -0.0283745  0.0284647  -0.997
## ClustersAL1 -0.0925561  0.0283957  -3.260
## ClustersAH2 -0.0159789  0.0380313  -0.420
##
## Correlation of Fixed Effects:
##              (Intr) ClsAL2 ClsAH1 ClsAA2 ClsAL1
## ClustersAL2 -0.300
## ClustersAH1 -0.336  0.136
## ClustersAA2 -0.195  0.135  0.092
## ClustersAL1 -0.207  0.147  0.093  0.103
## ClustersAH2 -0.226  0.076  0.142  0.048  0.051
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular

# extract coefficients
coefs <- data.frame(coef(summary(lmer1.prop)))
# use normal distribution to approximate p-value
coefs$p.z <- 2 * (1 - pnorm(abs(coefs$t.value)))
coefs

##              Estimate Std..Error      t.value      p.z
## (Intercept)  0.5881996692 0.01144187  51.40763868 0.000000e+00
## ClustersAL2 -0.1228822616 0.02031510 -6.04881253 1.459173e-09
## ClustersAH1 -0.0006248916 0.02203266 -0.02836206 9.773734e-01
## ClustersAA2 -0.0283744939 0.02846474 -0.99682955 3.188473e-01
## ClustersAL1 -0.0925560890 0.02839571 -3.25950958 1.116050e-03
## ClustersAH2 -0.0159789198 0.03803126 -0.42015223 6.743742e-01

# post-hoc
emmeans(lmer1.prop, list(pairwise ~ Clusters), adjust = "tukey")

## $`emmeans of Clusters`
## Clusters emmean      SE      df lower.CL upper.CL

```

```

## AA1      0.588 0.0129  1.26    0.486    0.690
## AL2      0.465 0.0211 12.72    0.420    0.511
## AH1      0.588 0.0241  9.89    0.534    0.641
## AA2      0.560 0.0303 39.07    0.498    0.621
## AL1      0.496 0.0294 43.85    0.436    0.555
## AH2      0.572 0.0404 64.64    0.492    0.653
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
##
## $`pairwise differences of Clusters`
## 1      estimate      SE  df t.ratio p.value
## AA1 - AL2  0.122882 0.0209 255  5.879 <.0001
## AA1 - AH1  0.000625 0.0236 208  0.026 1.0000
## AA1 - AA2  0.028374 0.0290 256  0.980 0.9240
## AA1 - AL1  0.092556 0.0288 256  3.213 0.0184
## AA1 - AH2  0.015979 0.0406 176  0.394 0.9988
## AL2 - AH1 -0.122257 0.0307 173 -3.983 0.0014
## AL2 - AA2 -0.094508 0.0333 257 -2.835 0.0551
## AL2 - AL1 -0.030326 0.0325 255 -0.934 0.9373
## AL2 - AH2 -0.106903 0.0453 162 -2.358 0.1775
## AH1 - AA2  0.027750 0.0362 198  0.767 0.9727
## AH1 - AL1  0.091931 0.0365 208  2.516 0.1241
## AH1 - AH2  0.015354 0.0418 252  0.368 0.9991
## AA2 - AL1  0.064182 0.0384 257  1.670 0.5529
## AA2 - AH2 -0.012396 0.0491 168 -0.252 0.9999
## AL1 - AH2 -0.076577 0.0495 178 -1.548 0.6335
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 6 estimates

# AA1 - AL2  0.122882 0.0209 255  5.879 <.0001 ***
# AA1 - AL1  0.092556 0.0288 256  3.213 0.0184 *
# AL2 - AH1 -0.122257 0.0307 173 -3.983 0.0014 **

```