nMDS

Pablo E. Gutiérrez-Fonseca

9/25/2021

```
libraries <- c("vegan", "ggplot2", "dplyr")
lapply(libraries, require, character.only = TRUE)
```

```
## [[1]]
## [1] TRUE
##
## [[2]]
## [1] TRUE
##
## [[3]]
## [1] TRUE
```

Paso 1. LLamar a la tabla (como .csv)

```
Moth_full  <- read.csv("data/fullmatrix.csv")
head(Moth_full)
```

Paso 2. Seleccionar las especies.

```
moth_sp <- select(Moth_full, M1:A248)
str(moth_sp)
ncol(moth_sp)
nrow(moth_sp)
```

Paso 3. Vamos hacer el nMDS

As a rule of thumb literature has identified the following cut-off values for stress-level:

Higher than 0.2 is poor (risks for false interpretation). 0.1 - 0.2 is fair (some distances can be misleading for interpretation). 0.05 - 0.1 is good (can be confident in inferences from plot). Less than 0.05 is excellent (this can be rare).

An important number to note is the stress, which is roughly the "goodness of fit" of your NMDS ordination. For a good representation of your data, the stress value should ideally be less than 0.2.

```
set.seed(1) # Con este comando, siempre comenzara del mismo lugar.

moth.mds <- metaMDS(moth_sp, distance = "bray", k = 2, trymax=100)  #using all the defaults
```

```
## Square root transformation
## Wisconsin double standardization
```

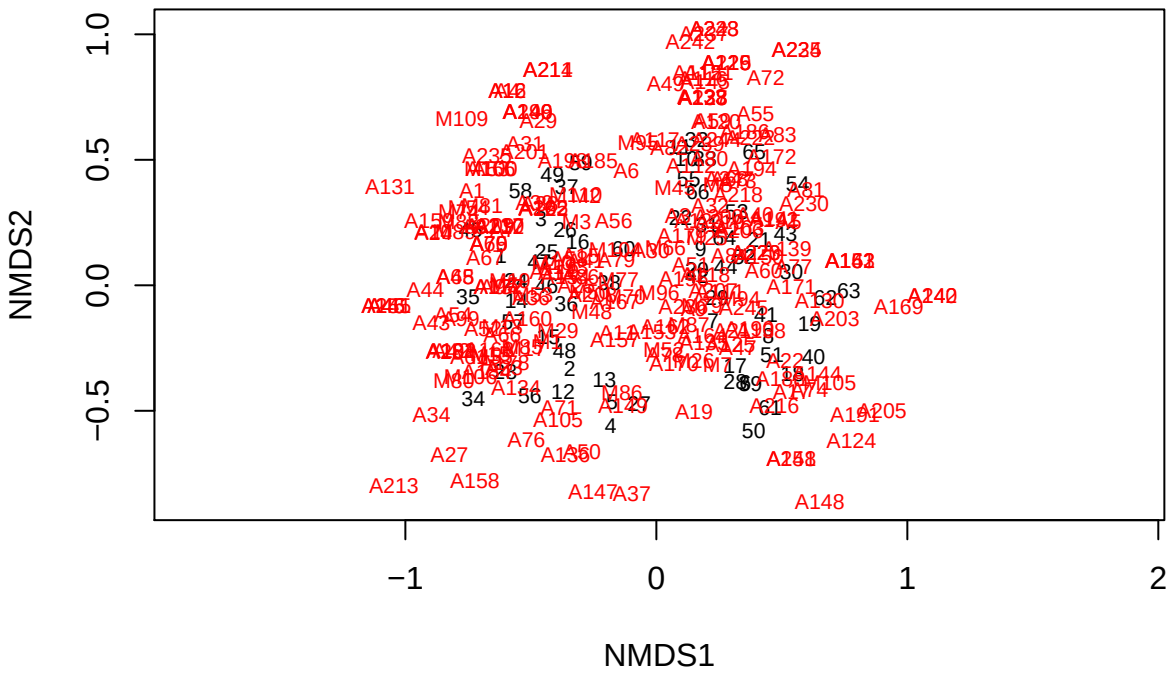
```
## Run 0 stress 0.2509353
## Run 1 stress 0.2595334
## Run 2 stress 0.2523396
## Run 3 stress 0.282036
## Run 4 stress 0.2509339
## ... New best solution
## ... Procrustes: rmse 0.0009701113  max resid 0.006667077
## ... Similar to previous best
## Run 5 stress 0.250961
## ... Procrustes: rmse 0.003294134  max resid 0.01957031
## Run 6 stress 0.2566342
## Run 7 stress 0.2510057
## ... Procrustes: rmse 0.007469064  max resid 0.04719079
## Run 8 stress 0.2530351
## Run 9 stress 0.251111
## ... Procrustes: rmse 0.01288445  max resid 0.05411702
## Run 10 stress 0.2563251
## Run 11 stress 0.2509585
## ... Procrustes: rmse 0.006726555  max resid 0.04626972
## Run 12 stress 0.2522903
## Run 13 stress 0.2509695
## ... Procrustes: rmse 0.003379292  max resid 0.02477938
## Run 14 stress 0.2523395
## Run 15 stress 0.286507
## Run 16 stress 0.2572426
## Run 17 stress 0.2566301
## Run 18 stress 0.2563038
## Run 19 stress 0.2579864
## Run 20 stress 0.2810398
## *** Solution reached
```

```
moth.mds
```

```
##
## Call:
## metaMDS(comm = moth_sp, distance = "bray", k = 2, trymax = 100)
##
## global Multidimensional Scaling using monoMDS
##
## Data:     wisconsin(sqrt(moth_sp))
## Distance: bray
##
## Dimensions: 2
## Stress:     0.2509339
## Stress type 1, weak ties
## Two convergent solutions found after 20 tries
## Scaling: centring, PC rotation, halfchange scaling
## Species: expanded scores based on 'wisconsin(sqrt(moth_sp))'
```

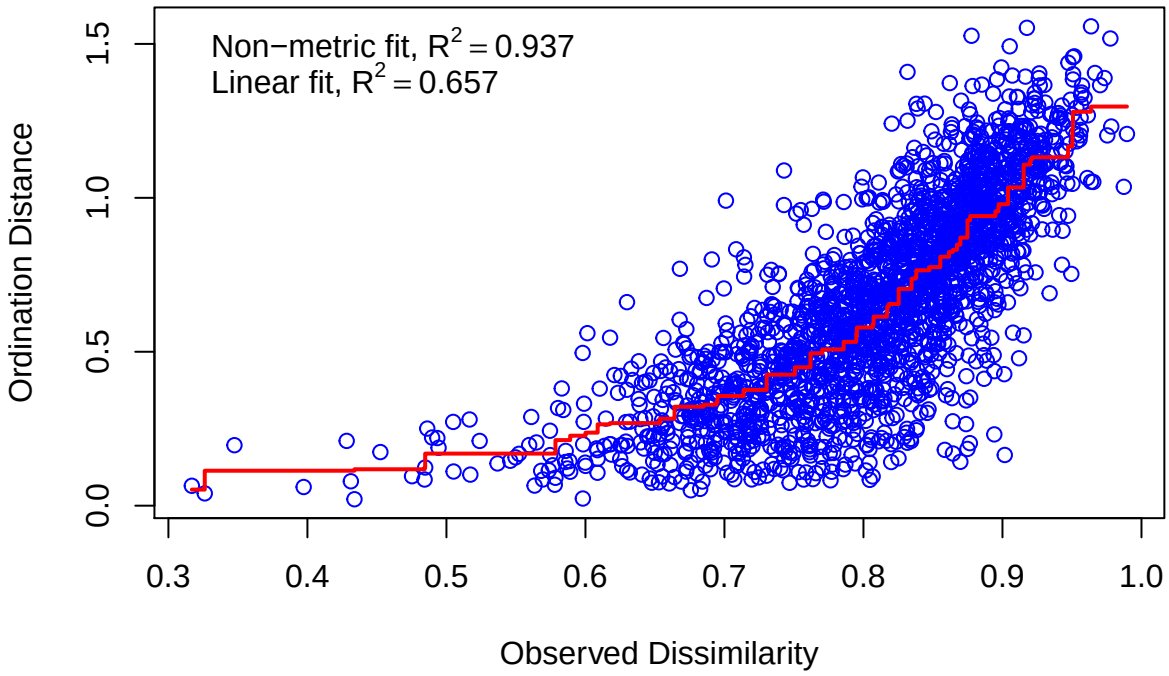
Paso 4. Vamos hacer el grafico. Un muy sensillo Plot

```
plot(moth.mds, type="t")
```



Paso 5. Mirar el stressplot Large scatter around the line suggests that original dissimilarities are not well preserved in the reduced number of dimensions. Looks pretty good in this case.

```
stressplot(moth.mds)
```



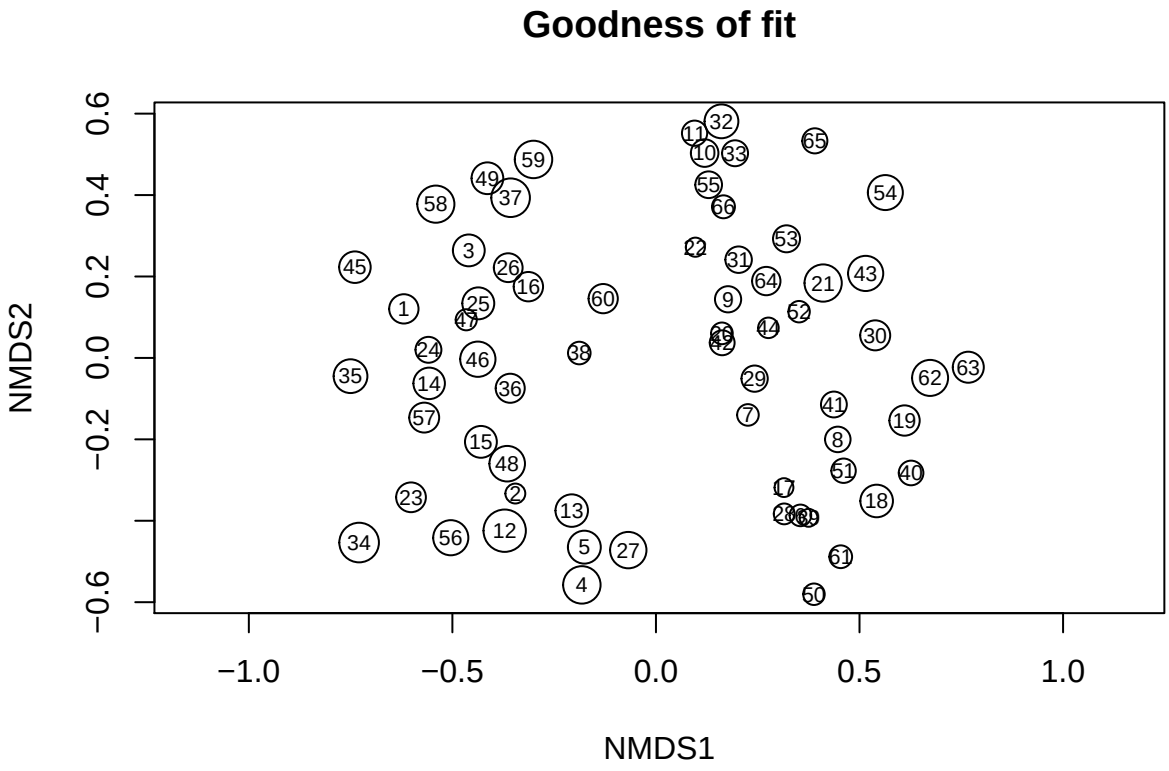
Paso 6. Otra metrica para evaluar el ajuste (buen desempeno) del nMDS es mirar el "Goodness of fit".

Visitar esta pagina para mas informacion: https://rdrr.io/rforge/vegan/man/goodness.metaMDS.html

```
gof <- goodness(moth.mds)
gof
```

```
##  [1] 0.03081585 0.02092671 0.03345693 0.03934151 0.03456932 0.02270405
##  [7] 0.02267995 0.02680810 0.02753120 0.02897304 0.02650191 0.04450752
## [13] 0.03412675 0.03305425 0.03345389 0.03087429 0.01976678 0.03423633
## [19] 0.03182412 0.02277720 0.03916362 0.02009878 0.03125329 0.02720130
## [25] 0.03354618 0.03025820 0.03822878 0.02183686 0.02757183 0.03151484
## [31] 0.02780051 0.03553918 0.02725503 0.04161688 0.03548390 0.03050237
## [37] 0.04057359 0.02372956 0.01882180 0.02589446 0.02716969 0.02614204
## [43] 0.03696792 0.02157502 0.03311220 0.03707901 0.02217101 0.03736177
## [49] 0.03334138 0.02258202 0.02564067 0.02251966 0.02835295 0.03679571
## [55] 0.02805723 0.03691240 0.03143149 0.03904002 0.03934138 0.03073778
## [61] 0.02391046 0.03786065 0.03260026 0.02980762 0.02637525 0.02409839
```

```
{plot (moth.mds, display = 'sites', type = 't', main = 'Goodness of fit') # this function draws NMDS or
points (moth.mds, display = 'sites', cex = 2*gof/mean(gof))} # and this adds the points with size refle
```



Paso 7. Chequear el % variacion explicado por cada Axis.

```
MDS <- cmdscale(vegdist(moth_sp, method = "bray"), k = 2, eig = T, add = T )
round(MDS$eig*100/sum(MDS$eig),1)
```

```
##  [1] 18.8 11.6  7.8  4.6  4.4  3.4  3.1  2.6  2.4  2.2  2.1  2.0  1.8  1.7  1.7
## [16]  1.5  1.4  1.3  1.3  1.2  1.2  1.1  1.1  1.1  1.0  1.0  1.0  0.9  0.9  0.8
## [31]  0.8  0.8  0.7  0.7  0.7  0.6  0.6  0.6  0.6  0.5  0.5  0.5  0.4  0.4  0.4
## [46]  0.4  0.4  0.3  0.3  0.3  0.3  0.3  0.2  0.2  0.2  0.2  0.2  0.2  0.2  0.1
## [61]  0.1  0.1  0.1  0.1  0.0  0.0
```

Paso 8. Cuantas dimenciones neceita para reducir el stress.

To decide how much dimensions you need you might plot the stress as a function of the number of dimensions. Note that, in contrast with a traditional scree plot, each bar does not represent the variance associated with each axis but the total stress (a function of the squared difference between d and d_hat) for all the dimensions. For example, the "3Dim" bar represent the stress of a solution in 3 dimensions, not the stress associated with the 3rd axis... Here the improvements of the representation for dimensions > than 2 dimensions are low.

https://stackoverflow.com/questions/49223740/cumulative-variance-explained-for-nmds-in-r

```r
n = 10
stress <- vector(length = n)
for (i in 1:n) {
    stress[i] <- metaMDS(moth_sp, distance = "bray", k = i)$stress
}
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.3847327
## Run 1 stress 0.5681109
## Run 2 stress 0.5681115
## Run 3 stress 0.5639052
## Run 4 stress 0.5639781
## Run 5 stress 0.5683156
## Run 6 stress 0.5680508
## Run 7 stress 0.5671818
## Run 8 stress 0.5674588
## Run 9 stress 0.5629774
## Run 10 stress 0.559712
## Run 11 stress 0.5668505
## Run 12 stress 0.5636168
## Run 13 stress 0.5661957
## Run 14 stress 0.5615393
## Run 15 stress 0.5684886
## Run 16 stress 0.5669772
## Run 17 stress 0.5646981
## Run 18 stress 0.568075
## Run 19 stress 0.3843358
## ... New best solution
## ... Procrustes: rmse 0.00638009  max resid 0.0351071
## Run 20 stress 0.5673483
## *** No convergence -- monoMDS stopping criteria:
##     20: scale factor of the gradient < sfgrmin
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.2509353
## Run 1 stress 0.258066
## Run 2 stress 0.2509987
## ... Procrustes: rmse 0.007661097  max resid 0.04605655
## Run 3 stress 0.2814577
## Run 4 stress 0.2588473
## Run 5 stress 0.2509702
## ... Procrustes: rmse 0.003888509  max resid 0.0271706
## Run 6 stress 0.2509692
## ... Procrustes: rmse 0.003497855  max resid 0.02415927
## Run 7 stress 0.2673446
## Run 8 stress 0.2509989
## ... Procrustes: rmse 0.007659982  max resid 0.04573383
## Run 9 stress 0.2578538
## Run 10 stress 0.2534028
## Run 11 stress 0.2551326
## Run 12 stress 0.2509339
## ... New best solution
```

```
## ... Procrustes: rmse 0.000967898  max resid 0.006665365
## ... Similar to previous best
## Run 13 stress 0.2579912
## Run 14 stress 0.2626952
## Run 15 stress 0.2509339
## ... Procrustes: rmse 4.658996e-05  max resid 0.0002181994
## ... Similar to previous best
## Run 16 stress 0.2509987
## ... Procrustes: rmse 0.007519338  max resid 0.04598299
## Run 17 stress 0.2509585
## ... Procrustes: rmse 0.006738279  max resid 0.04632735
## Run 18 stress 0.2557038
## Run 19 stress 0.2509696
## ... Procrustes: rmse 0.003402007  max resid 0.02494837
## Run 20 stress 0.2509688
## ... Procrustes: rmse 0.003111686  max resid 0.02277722
## *** Solution reached
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1867517
## Run 1 stress 0.1947603
## Run 2 stress 0.1918322
## Run 3 stress 0.1866932
## ... New best solution
## ... Procrustes: rmse 0.008649084  max resid 0.0414615
## Run 4 stress 0.1882354
## Run 5 stress 0.1866932
## ... Procrustes: rmse 4.927403e-05  max resid 0.0002261834
## ... Similar to previous best
## Run 6 stress 0.1890492
## Run 7 stress 0.1866934
## ... Procrustes: rmse 0.0003592932  max resid 0.00146563
## ... Similar to previous best
## Run 8 stress 0.1882352
## Run 9 stress 0.1895299
## Run 10 stress 0.1867541
## ... Procrustes: rmse 0.008016863  max resid 0.04209774
## Run 11 stress 0.1877962
## Run 12 stress 0.1928321
## Run 13 stress 0.1866933
## ... Procrustes: rmse 9.8263e-05  max resid 0.0005561325
## ... Similar to previous best
## Run 14 stress 0.1903924
## Run 15 stress 0.1916521
## Run 16 stress 0.186728
## ... Procrustes: rmse 0.007100508  max resid 0.03145465
## Run 17 stress 0.1921838
## Run 18 stress 0.1866927
## ... New best solution
## ... Procrustes: rmse 0.0004705958  max resid 0.002627182
## ... Similar to previous best
## Run 19 stress 0.1866929
## ... Procrustes: rmse 0.0002655013  max resid 0.001408088
## ... Similar to previous best
```

```
## Run 20 stress 0.186729
## ... Procrustes: rmse 0.007302493  max resid 0.03198173
## *** Solution reached
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1472134
## Run 1 stress 0.1535865
## Run 2 stress 0.1519037
## Run 3 stress 0.1471878
## ... New best solution
## ... Procrustes: rmse 0.007631452  max resid 0.05327874
## Run 4 stress 0.1471745
## ... New best solution
## ... Procrustes: rmse 0.0165164  max resid 0.08749151
## Run 5 stress 0.1492772
## Run 6 stress 0.1472137
## ... Procrustes: rmse 0.01519826  max resid 0.09220426
## Run 7 stress 0.1481417
## Run 8 stress 0.147223
## ... Procrustes: rmse 0.01686351  max resid 0.08661913
## Run 9 stress 0.1473786
## ... Procrustes: rmse 0.01237671  max resid 0.05924728
## Run 10 stress 0.1471345
## ... New best solution
## ... Procrustes: rmse 0.007832771  max resid 0.0535424
## Run 11 stress 0.1471882
## ... Procrustes: rmse 0.0151391  max resid 0.08556419
## Run 12 stress 0.1472143
## ... Procrustes: rmse 0.0175421  max resid 0.0915777
## Run 13 stress 0.1471878
## ... Procrustes: rmse 0.01517898  max resid 0.0858641
## Run 14 stress 0.1482029
## Run 15 stress 0.1471877
## ... Procrustes: rmse 0.01526721  max resid 0.08664122
## Run 16 stress 0.1471353
## ... Procrustes: rmse 0.00479566  max resid 0.02469199
## Run 17 stress 0.1488962
## Run 18 stress 0.147188
## ... Procrustes: rmse 0.01523096  max resid 0.08658252
## Run 19 stress 0.1471352
## ... Procrustes: rmse 0.004509947  max resid 0.02449091
## Run 20 stress 0.1472134
## ... Procrustes: rmse 0.01735544  max resid 0.09071273
## *** No convergence -- monoMDS stopping criteria:
##     10: no. of iterations >= maxit
##     10: stress ratio > sratmax
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1229536
## Run 1 stress 0.1231511
## ... Procrustes: rmse 0.04888813  max resid 0.1936297
## Run 2 stress 0.1229082
## ... New best solution
## ... Procrustes: rmse 0.04398667  max resid 0.2053497
```

```
## Run 3 stress 0.1234641
## Run 4 stress 0.1227772
## ... New best solution
## ... Procrustes: rmse 0.05293927  max resid 0.2085025
## Run 5 stress 0.1239301
## Run 6 stress 0.1230582
## ... Procrustes: rmse 0.04373959  max resid 0.1608896
## Run 7 stress 0.1234682
## Run 8 stress 0.1230263
## ... Procrustes: rmse 0.03987507  max resid 0.1582695
## Run 9 stress 0.1228837
## ... Procrustes: rmse 0.02735616  max resid 0.111629
## Run 10 stress 0.1224974
## ... New best solution
## ... Procrustes: rmse 0.03385775  max resid 0.1092351
## Run 11 stress 0.1224844
## ... New best solution
## ... Procrustes: rmse 0.05790252  max resid 0.1880256
## Run 12 stress 0.1234664
## Run 13 stress 0.1238022
## Run 14 stress 0.1228796
## ... Procrustes: rmse 0.05438419  max resid 0.1526529
## Run 15 stress 0.1231978
## Run 16 stress 0.1305964
## Run 17 stress 0.1229829
## ... Procrustes: rmse 0.03371464  max resid 0.1575952
## Run 18 stress 0.1229163
## ... Procrustes: rmse 0.04085033  max resid 0.1617605
## Run 19 stress 0.1228371
## ... Procrustes: rmse 0.05374827  max resid 0.1534564
## Run 20 stress 0.1226291
## ... Procrustes: rmse 0.01063919  max resid 0.03260354
## *** No convergence -- monoMDS stopping criteria:
##     16: no. of iterations >= maxit
##      4: stress ratio > sratmax
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1041799
## Run 1 stress 0.1045279
## ... Procrustes: rmse 0.05062653  max resid 0.1886346
## Run 2 stress 0.1040836
## ... New best solution
## ... Procrustes: rmse 0.02771921  max resid 0.1011892
## Run 3 stress 0.103802
## ... New best solution
## ... Procrustes: rmse 0.03887743  max resid 0.1292979
## Run 4 stress 0.1037921
## ... New best solution
## ... Procrustes: rmse 0.002179864  max resid 0.006317243
## ... Similar to previous best
## Run 5 stress 0.1045391
## Run 6 stress 0.1050911
## Run 7 stress 0.1042464
## ... Procrustes: rmse 0.05132673  max resid 0.1784793
```

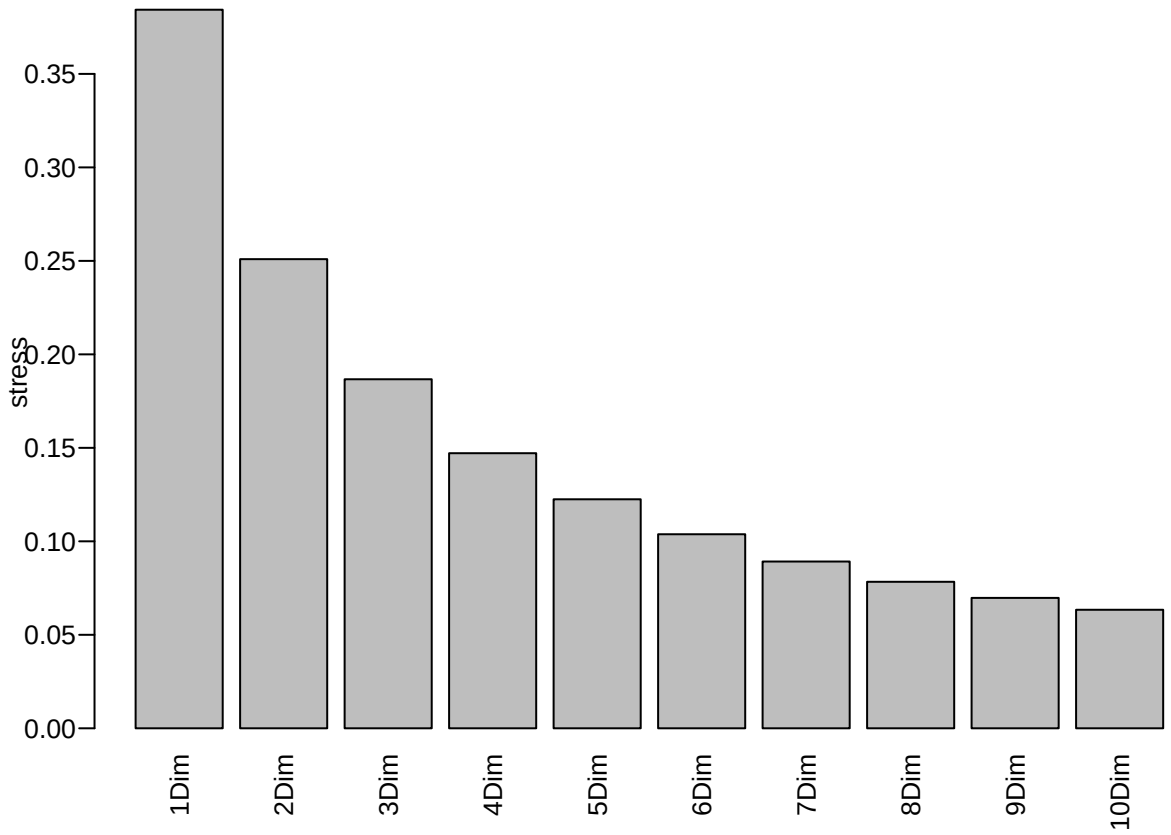
```
## Run 8 stress 0.1038427
## ... Procrustes: rmse 0.008348228  max resid 0.04870888
## Run 9 stress 0.1038317
## ... Procrustes: rmse 0.005297702  max resid 0.01886154
## Run 10 stress 0.1038867
## ... Procrustes: rmse 0.02657405  max resid 0.09325475
## Run 11 stress 0.1041321
## ... Procrustes: rmse 0.03709033  max resid 0.1431816
## Run 12 stress 0.1040332
## ... Procrustes: rmse 0.03017842  max resid 0.09295283
## Run 13 stress 0.1038088
## ... Procrustes: rmse 0.003847844  max resid 0.01821796
## Run 14 stress 0.1039769
## ... Procrustes: rmse 0.03251159  max resid 0.1279609
## Run 15 stress 0.1038635
## ... Procrustes: rmse 0.02679485  max resid 0.09285018
## Run 16 stress 0.1038336
## ... Procrustes: rmse 0.007457331  max resid 0.04384425
## Run 17 stress 0.1056353
## Run 18 stress 0.1037989
## ... Procrustes: rmse 0.001784536  max resid 0.005500061
## ... Similar to previous best
## Run 19 stress 0.1038029
## ... Procrustes: rmse 0.004336353  max resid 0.02352524
## Run 20 stress 0.1043564
## *** Solution reached
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.08922516
## Run 1 stress 0.09098486
## Run 2 stress 0.09001057
## Run 3 stress 0.0892564
## ... Procrustes: rmse 0.01497927  max resid 0.0940726
## Run 4 stress 0.08923379
## ... Procrustes: rmse 0.001990524  max resid 0.009204329
## ... Similar to previous best
## Run 5 stress 0.08946439
## ... Procrustes: rmse 0.01313187  max resid 0.04344846
## Run 6 stress 0.09005837
## Run 7 stress 0.08939116
## ... Procrustes: rmse 0.02449299  max resid 0.1383703
## Run 8 stress 0.08936454
## ... Procrustes: rmse 0.0105927  max resid 0.05375773
## Run 9 stress 0.08933036
## ... Procrustes: rmse 0.02184052  max resid 0.1291654
## Run 10 stress 0.09008214
## Run 11 stress 0.09064976
## Run 12 stress 0.08920284
## ... New best solution
## ... Procrustes: rmse 0.01910009  max resid 0.1278456
## Run 13 stress 0.08944697
## ... Procrustes: rmse 0.02345653  max resid 0.1164047
## Run 14 stress 0.09123979
## Run 15 stress 0.08945458
```

```
## ... Procrustes: rmse 0.02373917  max resid 0.1162705
## Run 16 stress 0.08928464
## ... Procrustes: rmse 0.02079494  max resid 0.1348639
## Run 17 stress 0.08925024
## ... Procrustes: rmse 0.02008263  max resid 0.1339664
## Run 18 stress 0.09164843
## Run 19 stress 0.08939769
## ... Procrustes: rmse 0.01445275  max resid 0.09284891
## Run 20 stress 0.08931765
## ... Procrustes: rmse 0.02118339  max resid 0.1232517
## *** No convergence -- monoMDS stopping criteria:
##     19: no. of iterations >= maxit
##      1: stress ratio > sratmax
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.07840797
## Run 1 stress 0.07836671
## ... New best solution
## ... Procrustes: rmse 0.009717777  max resid 0.02545575
## Run 2 stress 0.07889332
## Run 3 stress 0.07948725
## Run 4 stress 0.07927069
## Run 5 stress 0.07965848
## Run 6 stress 0.07864306
## ... Procrustes: rmse 0.01934034  max resid 0.06968534
## Run 7 stress 0.07837537
## ... Procrustes: rmse 0.003154258  max resid 0.01331049
## Run 8 stress 0.08124367
## Run 9 stress 0.0785121
## ... Procrustes: rmse 0.01245831  max resid 0.04197671
## Run 10 stress 0.07897184
## Run 11 stress 0.0788555
## ... Procrustes: rmse 0.03672841  max resid 0.1648051
## Run 12 stress 0.0783662
## ... New best solution
## ... Procrustes: rmse 0.00299924  max resid 0.01072812
## Run 13 stress 0.07837938
## ... Procrustes: rmse 0.006605765  max resid 0.02638621
## Run 14 stress 0.07851252
## ... Procrustes: rmse 0.01371164  max resid 0.04787773
## Run 15 stress 0.07838557
## ... Procrustes: rmse 0.004374073  max resid 0.01373191
## Run 16 stress 0.08019961
## Run 17 stress 0.07904616
## Run 18 stress 0.07845023
## ... Procrustes: rmse 0.01059562  max resid 0.03271408
## Run 19 stress 0.07864578
## ... Procrustes: rmse 0.02200873  max resid 0.07500325
## Run 20 stress 0.07934495
## *** No convergence -- monoMDS stopping criteria:
##     19: no. of iterations >= maxit
##      1: stress ratio > sratmax
## Square root transformation
## Wisconsin double standardization
```

```
## Run 0 stress 0.06974
## Run 1 stress 0.06974409
## ... Procrustes: rmse 0.002785319  max resid 0.007565117
## ... Similar to previous best
## Run 2 stress 0.07163843
## Run 3 stress 0.0713707
## Run 4 stress 0.07021654
## ... Procrustes: rmse 0.01836479  max resid 0.08510465
## Run 5 stress 0.07000843
## ... Procrustes: rmse 0.01163488  max resid 0.02950614
## Run 6 stress 0.06979022
## ... Procrustes: rmse 0.005970457  max resid 0.02443095
## Run 7 stress 0.07058828
## Run 8 stress 0.06984001
## ... Procrustes: rmse 0.008074552  max resid 0.01968537
## Run 9 stress 0.07007787
## ... Procrustes: rmse 0.01733833  max resid 0.041012
## Run 10 stress 0.06974427
## ... Procrustes: rmse 0.00275597  max resid 0.006635771
## ... Similar to previous best
## Run 11 stress 0.07063419
## Run 12 stress 0.07054387
## Run 13 stress 0.06976668
## ... Procrustes: rmse 0.003767149  max resid 0.008534269
## ... Similar to previous best
## Run 14 stress 0.07030616
## Run 15 stress 0.06983126
## ... Procrustes: rmse 0.006986279  max resid 0.01701897
## Run 16 stress 0.07006814
## ... Procrustes: rmse 0.01453451  max resid 0.04764786
## Run 17 stress 0.06979519
## ... Procrustes: rmse 0.008444138  max resid 0.02537364
## Run 18 stress 0.07093319
## Run 19 stress 0.0697837
## ... Procrustes: rmse 0.00508733  max resid 0.0123812
## Run 20 stress 0.07002471
## ... Procrustes: rmse 0.01144351  max resid 0.02723927
## *** Solution reached
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.06343135
## Run 1 stress 0.06340261
## ... New best solution
## ... Procrustes: rmse 0.01716874  max resid 0.05519689
## Run 2 stress 0.06351589
## ... Procrustes: rmse 0.03233192  max resid 0.07305745
## Run 3 stress 0.06342442
## ... Procrustes: rmse 0.02941419  max resid 0.06543713
## Run 4 stress 0.06434077
## Run 5 stress 0.06348554
## ... Procrustes: rmse 0.03317476  max resid 0.07189175
## Run 6 stress 0.06345187
## ... Procrustes: rmse 0.02222414  max resid 0.06491989
## Run 7 stress 0.06346003
```

```
## ... Procrustes: rmse 0.02018651  max resid 0.06564787
## Run 8 stress 0.06368081
## ... Procrustes: rmse 0.02632952  max resid 0.1026811
## Run 9 stress 0.06504183
## Run 10 stress 0.06394857
## Run 11 stress 0.06404723
## Run 12 stress 0.06350122
## ... Procrustes: rmse 0.02859978  max resid 0.06575821
## Run 13 stress 0.06386401
## ... Procrustes: rmse 0.03073777  max resid 0.07687876
## Run 14 stress 0.06366366
## ... Procrustes: rmse 0.02348809  max resid 0.08113178
## Run 15 stress 0.06348515
## ... Procrustes: rmse 0.02362809  max resid 0.08189401
## Run 16 stress 0.06374267
## ... Procrustes: rmse 0.04035174  max resid 0.1000998
## Run 17 stress 0.06568226
## Run 18 stress 0.0634738
## ... Procrustes: rmse 0.02846266  max resid 0.06979868
## Run 19 stress 0.06360368
## ... Procrustes: rmse 0.03848785  max resid 0.1128543
## Run 20 stress 0.06413267
## *** No convergence -- monoMDS stopping criteria:
##     20: no. of iterations >= maxit
```

```
names(stress) <- paste0(1:n, "Dim")
# x11(width = 10/2.54, height = 7/2.54)
par(mar = c(3.5,3.5,1,1), mgp = c(2, 0.6, 0), cex = 0.8, las = 2)
barplot(stress, ylab = "stress")
```



Paso 9. Vamos a poner Habitat, Site y Periodo en el grafico. Primer las busco en la Matrix Orignal

```
Habitat <- select(Moth_full, Habitat)
Site <- select(Moth_full, Site)
Period <- select(Moth_full, Period)
```

Paso 10. Extraer las cordenadas de los Axis del nmds.

```
data.scores <- as.data.frame(scores(moth.mds))  #Using the scores function from vegan to extract the si
data.scores
```

```
##          NMDS1        NMDS2
## 1  -0.61922141  0.120470413
## 2  -0.34538334 -0.333130972
## 3  -0.45971354  0.264104192
## 4  -0.18201886 -0.557503052
## 5  -0.17590801 -0.464825120
## 6   0.35481540 -0.386808065
## 7   0.22592884 -0.140024092
## 8   0.44666523 -0.200156166
## 9   0.17711216  0.144021119
## 10  0.11958516  0.503172409
## 11  0.09487077  0.551938903
## 12 -0.37156920 -0.424413360
## 13 -0.20690131 -0.375013455
## 14 -0.55706899 -0.062499904
```

```
## 15 -0.42991704 -0.206186892
## 16 -0.31339020  0.174980913
## 17  0.31447227 -0.318507583
## 18  0.54209725 -0.351384668
## 19  0.61061747 -0.153736801
## 20  0.16198514  0.060347906
## 21  0.41066018  0.183934887
## 22  0.09682790  0.271984593
## 23 -0.60161771 -0.342357057
## 24 -0.55879744  0.019911137
## 25 -0.43636878  0.134019714
## 26 -0.36289717  0.221650212
## 27 -0.06797333 -0.472073388
## 28  0.31500121 -0.383138638
## 29  0.24237375 -0.050896154
## 30  0.53854390  0.055414562
## 31  0.20303820  0.241272327
## 32  0.16085810  0.581079898
## 33  0.19431873  0.502681123
## 34 -0.72911115 -0.453525961
## 35 -0.75026191 -0.044802339
## 36 -0.35799216 -0.074521103
## 37 -0.35700524  0.393474356
## 38 -0.18829658  0.012001047
## 39  0.37490714 -0.392733021
## 40  0.62664081 -0.282610667
## 41  0.43705481 -0.114411375
## 42  0.16265790  0.037138414
## 43  0.51508528  0.207516576
## 44  0.27622272  0.073897836
## 45 -0.73941782  0.222885589
## 46 -0.43749124 -0.003163544
## 47 -0.46597979  0.093647975
## 48 -0.36540117 -0.259838129
## 49 -0.41398258  0.441319299
## 50  0.38814421 -0.580466912
## 51  0.46084588 -0.277086944
## 52  0.35158363  0.113565102
## 53  0.32069228  0.292551847
## 54  0.56344938  0.406146054
## 55  0.12945544  0.425346242
## 56 -0.50397694 -0.441660091
## 57 -0.56920890 -0.146571940
## 58 -0.54067252  0.377662103
## 59 -0.30077513  0.487495974
## 60 -0.12962218  0.145606296
## 61  0.45381256 -0.488343932
## 62  0.67336409 -0.049121680
## 63  0.76718389 -0.022874961
## 64  0.27098802  0.188769811
## 65  0.39025161  0.533190976
## 66  0.16583030  0.371188161
```

Paso 11. Unir los datos de Habitat, Site y Periodo a mi nueva dataframe (i.e., data.scores)

```
data.scores$Site <- unlist(Site)  #   create a column of site names
data.scores$Period <- unlist(Period)
data.scores$Habitat <- unlist(Habitat)
head(data.scores) #look at the data
```

```
##        NMDS1      NMDS2 Site         Period  Habitat
## 1 -0.6192214  0.1204704   T1  Pre-Hurricane Tabonuco
## 2 -0.3453833 -0.3331310   T1  Pre-Hurricane Tabonuco
## 3 -0.4597135  0.2641042   T1  Pre-Hurricane Tabonuco
## 4 -0.1820189 -0.5575031   T1  Pre-Hurricane Tabonuco
## 5 -0.1759080 -0.4648251   T1  Pre-Hurricane Tabonuco
## 6  0.3548154 -0.3868081   T1 Post-Hurricane Tabonuco
```

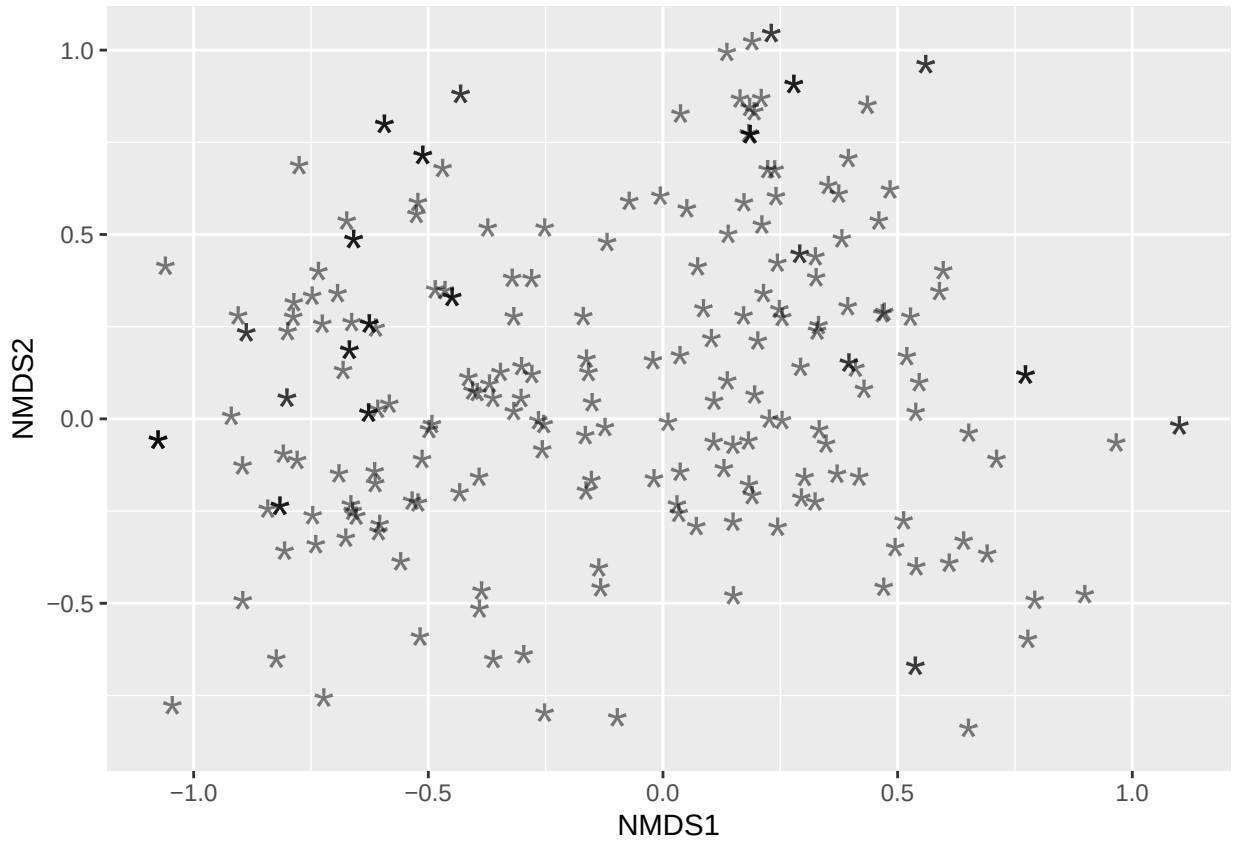
Paso 12. Extraer las cordenadas de las especies y ponerlas en una nueva matrix.

```
species.scores <- as.data.frame(scores(moth.mds, "species"))
species.scores$species <- rownames(species.scores)  # create a column of species, from the rownames of
```

Paso 13. Vamos hacer graficos bonitos en ggplot2. Solo Especies.

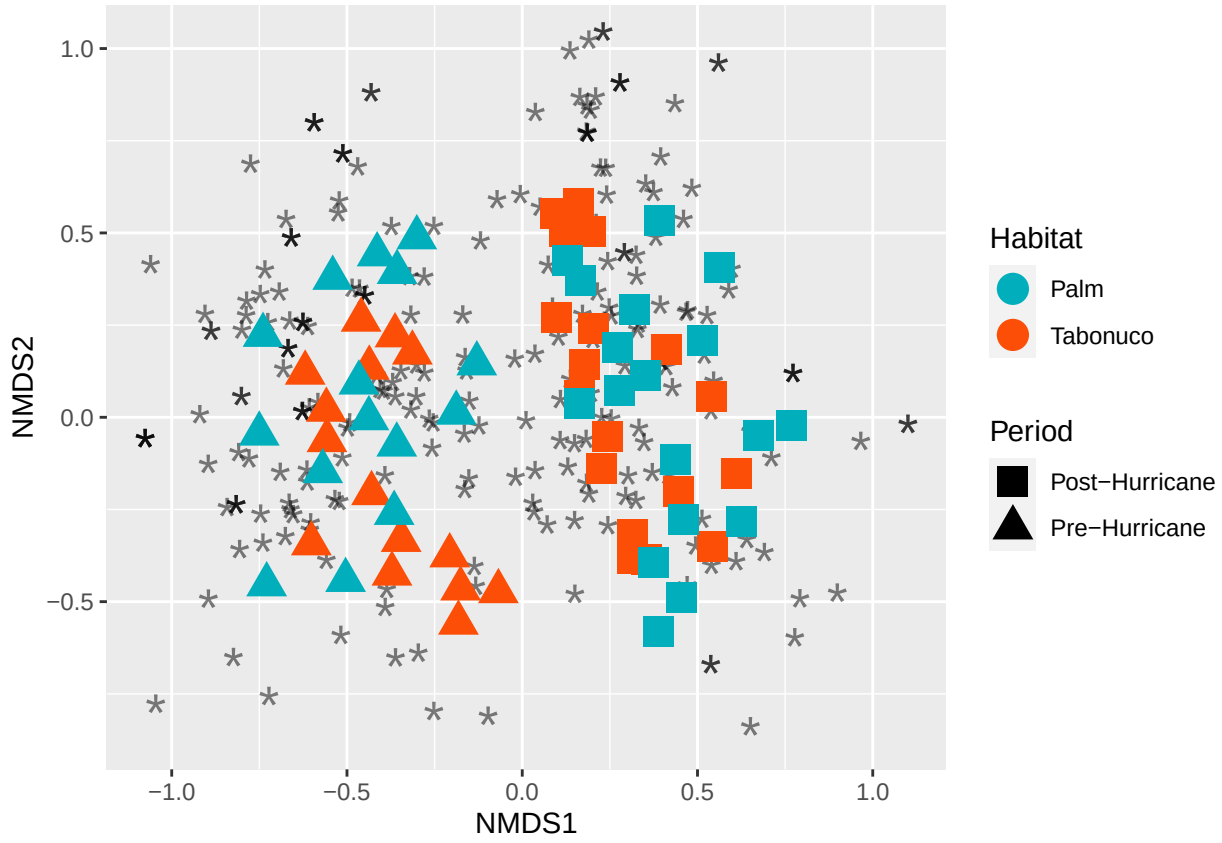
```
p <- ggplot() +
  geom_text(data=species.scores, aes(x=NMDS1,y=NMDS2, label = "*"),size=7, alpha=0.5) # add the species

p
```



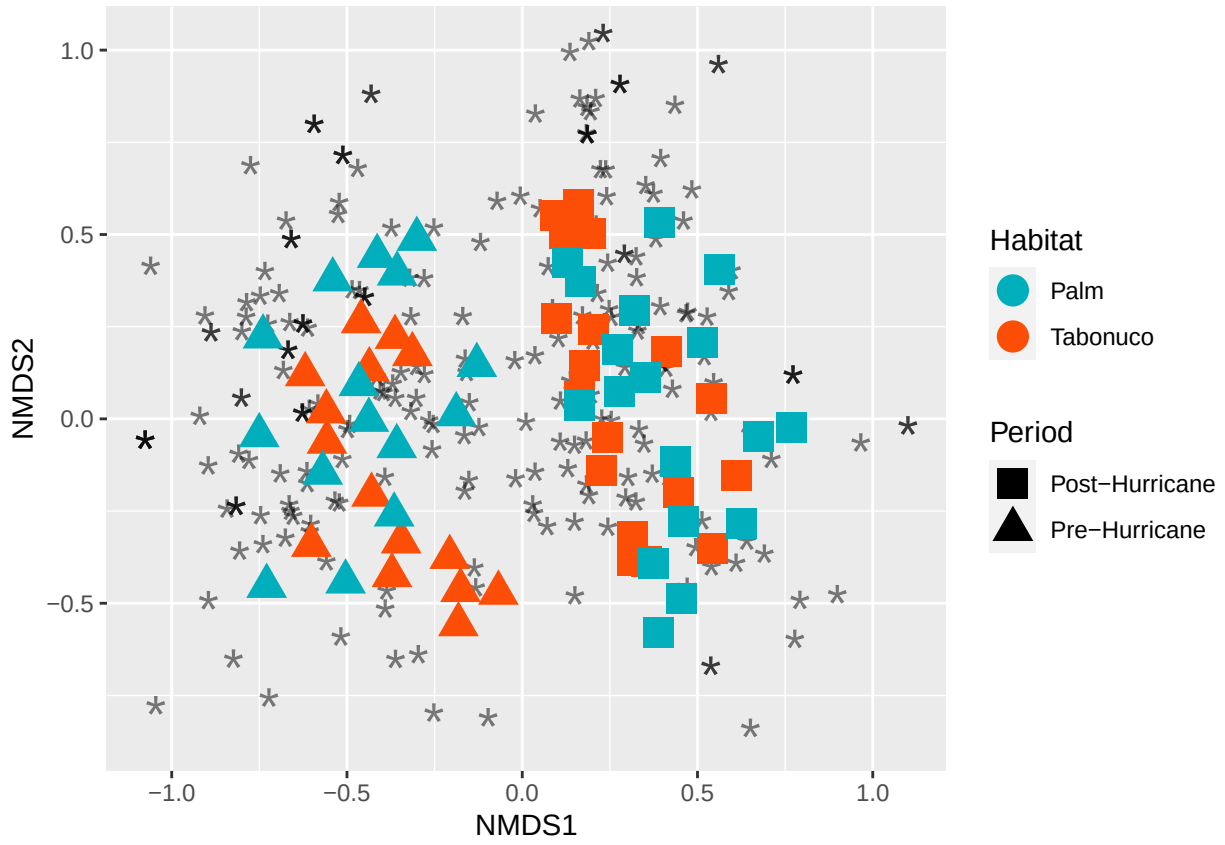
Paso 14. Vamos a poner etiquetas. En este caso solo Habitats y Periordo

```
p1 <- p + geom_point(data=data.scores,aes(x=NMDS1,y=NMDS2,
                                          shape=Period,colour=Habitat),size=5) + # add the point marker
        scale_color_manual(values=c("#00AFBB", "#FC4E07")) +
        scale_shape_manual(values=c(15, 17))
p1
```



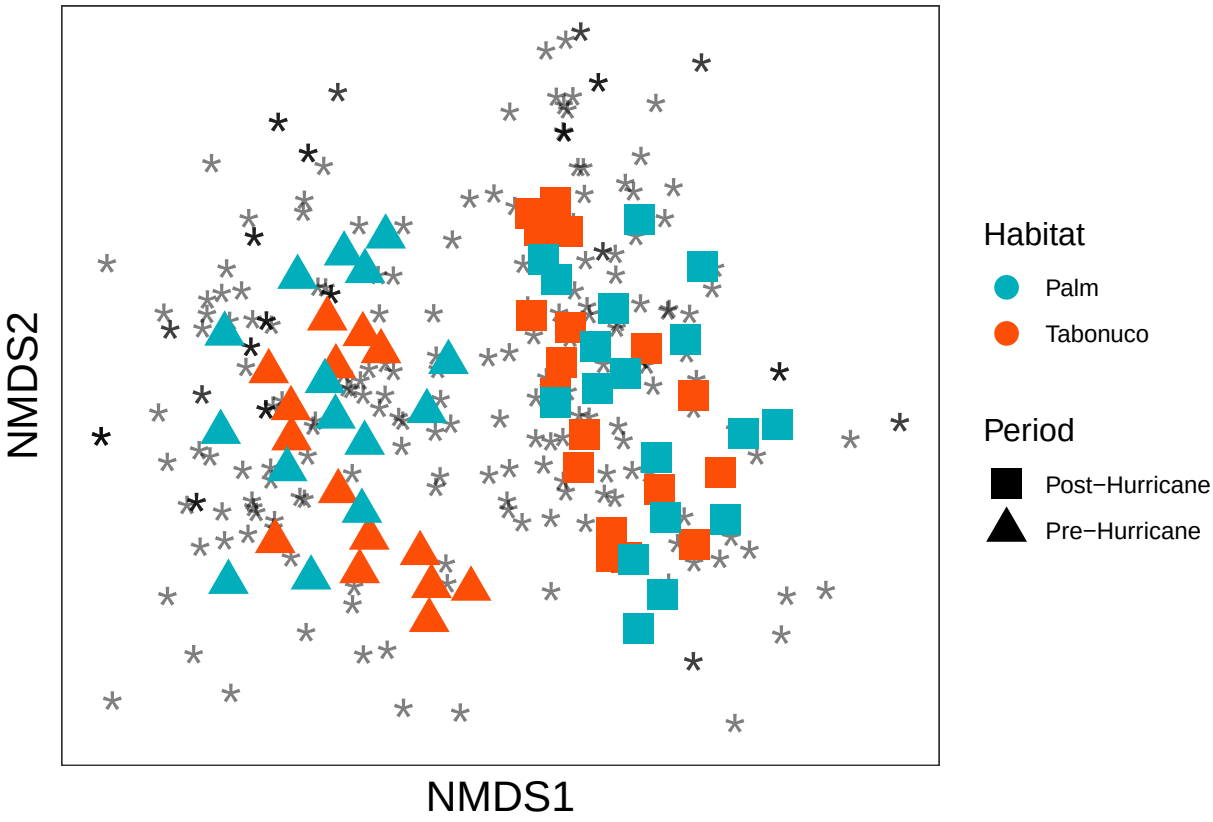
Paso 15. Agrerar sitios.

```
p2 <- p1 + geom_text(data=data.scores,aes(x=NMDS1,y=NMDS2,label=""),size=6,vjust=0)  # add the site lab
p2
```



Paso 16. Graficos de Publicacion.

```
p3 <- p1 + coord_equal() +
  theme_bw() +
  theme(legend.title = element_text(size=12),
        axis.text.x = element_blank(),  # remove x-axis text
        axis.text.y = element_blank(), # remove y-axis text
        axis.ticks = element_blank(),  # remove axis ticks
        axis.title.x = element_text(size=16), # remove x-axis labels
        axis.title.y = element_text(size=16), # remove y-axis labels
        panel.background = element_blank(),
        panel.grid.major = element_blank(),  #remove major-grid labels
        panel.grid.minor = element_blank(),  #remove minor-grid labels
        plot.background = element_blank()) +
  guides(color = guide_legend(override.aes = list(shape = 16, size = 4)))
p3
```


NMDS2
NMDS1
Habitat
Palm
Tabonuco
Period
Post−Hurricane
Pre−Hurricane