Homework 1

Yu-Jen Lin

b04b01036@ntu.edu.tw (mailto:b04b01036@ntu.edu.tw)

---

read the data

First, I have to read the data in R. Thus, I use “read.table” function to read the txt file. The “header=T” means I set the first row as header. On the other hand, the “header=F” means that first row is not a header.

```
composition <- read.table("~/R/copepod_composition.txt", header=T)
head(composition)
```

```
##   X.p1 p3 p4 p6 p13  p16  p19 p21  p23  p25  s18  s19  s20  s22  s23  s25
## 1    0  0  0  0   0 0.00 0.00   0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.30
## 2    0  0  0  0   0 0.00 0.00   0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## 3    0  0  0  0   0 0.22 2.34   0 2.51 1.62 0.00 0.00 0.00 0.00 0.00 0.00
## 4    0  0  0  0   0 0.00 0.00   0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## 5    0  0  0  0   0 0.00 0.00   0 0.00 0.00 4.07 1.56 1.08 4.83 8.49 1.49
## 6    0  0  0  0   0 0.00 0.00   0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##    s27  s29   sA   sB   sC   sD   sE   sF  sG   w22   w23  w25  w27  w29
## 1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0  0.00  0.00 0.00 0.00 0.00
## 2 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.0  0.00  0.00 0.00 0.00 0.42
## 3 1.52 0.30 3.06 1.35 1.24 0.62 2.92 0.31 1.4  0.00  0.00 0.00 0.00 0.00
## 4 0.00 0.00 0.26 0.00 0.00 0.00 0.00 0.00 0.0  0.00  0.00 0.00 0.00 0.00
## 5 0.00 0.00 0.26 0.00 0.00 0.00 0.32 1.53 0.0 19.42 51.76 2.81 3.85 6.28
## 6 0.76 1.51 0.00 0.00 0.00 0.00 0.00 0.00 0.0  0.00  0.00 0.00 0.00 0.00
##     wA   wB   wC   wD
## 1 0.00 0.00 0.00 0.00
## 2 0.00 0.00 0.00 0.00
## 3 0.00 2.22 1.13 0.93
## 4 0.00 0.00 0.00 0.00
## 5 0.55 0.28 0.00 0.62
## 6 0.00 0.00 0.00 0.00
```

```
density <- read.table("~/R/cop_density.txt", header=F)
head(density)
```

```
##     V1
## 1 1119
## 2 1153
## 3 1719
## 4  855
## 5 1246
## 6 2123
```

1. Calculate the copepod density for each species for each cruise-station

transpose

First, in order to make these two dataset multipliable (“composition” and “density”), I do the matrix transpose on the “desity”.

```
density.t <- t(density)
```

copepod density

Second, I calculate the copepod density by mutiply “density.t” and “composition”, and then mutiply 0.01 to make “composition” become a porpotion.

```
density.each <- density.t*composition*0.01
head(density.each)
```

```
##   X.p1 p3 p4 p6 p13     p16     p19 p21      p23      p25     s18      s19
## 1    0  0  0  0   0  0.0000  0.0000   0   0.0000 0.000000  0.0000   0.0000
## 2    0  0  0  0   0  0.0000  0.0000   0   0.0000 0.000000  0.0000   0.0000
## 3    0  0  0  0   0 17.4196 26.1846   0 137.8492 0.756054  0.0000   0.0000
## 4    0  0  0  0   0  0.0000  0.0000   0   0.0000 0.000000  0.0000   0.0000
## 5    0  0  0  0   0  0.0000  0.0000   0   0.0000 0.000000 61.3756 123.5208
## 6    0  0  0  0   0  0.0000  0.0000   0   0.0000 0.000000  0.0000   0.0000
##       s20   s22      s23      s25     s27     s29       sA      sB      sC
## 1  0.0000  0.00   0.0000 0.125220  0.0000  0.0000 0.000000  0.0000  0.0000
## 2  0.0000  0.00   0.0000 0.000000  0.0000  0.0000 0.000000  0.0000  0.0000
## 3  0.0000  0.00   0.0000 0.000000 20.5352  8.6850 0.283356 20.2095 19.6912
## 4  0.0000  0.00   0.0000 0.000000  0.0000  0.0000 0.129220  0.0000  0.0000
## 5 12.0852 91.77 466.2708 0.695383  0.0000  0.0000 0.066924  0.0000  0.0000
## 6  0.0000  0.00   0.0000 0.000000 14.4400 82.9292 0.000000  0.0000  0.0000
##         sD      sE      sF      sG      w22      w23      w25     w27
## 1 0.000000  0.0000  0.0000 0.00000   0.0000   0.0000 0.000000  0.0000
## 2 0.000000  0.0000  0.0000 0.00000   0.0000   0.0000 0.000000  0.0000
## 3 0.258788 33.8428 14.9513 0.64484   0.0000   0.0000 0.000000  0.0000
## 4 0.000000  0.0000  0.0000 0.00000   0.0000   0.0000 0.000000  0.0000
## 5 0.000000  4.3232 44.2935 0.00000 290.7174 821.9488 1.172894 44.6215
## 6 0.000000  0.0000  0.0000 0.00000   0.0000   0.0000 0.000000  0.0000
##        w29      wA      wB      wC       wD
## 1   0.0000 0.00000  0.0000  0.0000 0.000000
## 2  20.6598 0.00000  0.0000  0.0000 0.000000
## 3   0.0000 0.00000 18.9810 55.5847 3.330330
## 4   0.0000 0.00000  0.0000  0.0000 0.000000
## 5 302.8844 0.25333  5.9444  0.0000 0.500154
## 6   0.0000 0.00000  0.0000  0.0000 0.000000
```

2. For each cruise-station, calculate the species richness (number of species) and Shannon diversity index

species richness (number of species)

First, I create “density.each.zero” to distinguish items larger than 0 from the others items. If the items are larger than 0, which means that species does exist in that cruise-station.

```
density.each.zero <- density.each>0
head(density.each.zero)
```

```
##       X.p1    p3    p4    p6   p13   p16   p19   p21   p23   p25   s18
## [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
## [4,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [5,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## [6,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##        s19   s20   s22   s23   s25   s27   s29    sA    sB    sC    sD
## [1,] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [4,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## [5,]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE
## [6,] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE
##         sE    sF    sG   w22   w23   w25   w27   w29    wA    wB    wC
## [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## [3,]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
## [4,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [5,]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## [6,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##         wD
## [1,] FALSE
## [2,] FALSE
## [3,]  TRUE
## [4,] FALSE
## [5,]  TRUE
## [6,] FALSE
```

Second, when I use “apply” to do the FUN (function) by each column (MARGIN=2), I can count the items that are larger than 0 by summing up the “density.each.zero” by row (“True” stands for 1). And then correct it by minus 1.

```
species.number <- apply(density.each.zero, MARGIN=2, FUN = sum)
species.number.correction <- species.number-1
species.number.correction
```

```
## X.p1   p3   p4   p6  p13  p16  p19  p21  p23  p25  s18  s19  s20  s22  s23 
##    5   11    7    8   30   28   42    6   45   38   38   40   31   24   31 
##  s25  s27  s29   sA   sB   sC   sD   sE   sF   sG  w22  w23  w25  w27  w29 
##   40   39   22   46   48   43   45   43   37   24   17   15   23   32   26 
##   wA   wB   wC   wD 
##   43   53   63   47
```

Shannon diversity index

First, I sum up all the density in “density.each” in order to know the total amount of all the species. It’s like we assume there’s only 1 cubic meter of the sumation of all cruise-stations, and to see the propotion of each species in this 1 cubic meter(“density.propotion”).

```
density.sum <- sum(density.each)
density.propotion <- density.each/density.sum
head(density.propotion)
```

```
##   X.p1 p3 p4 p6 p13          p16         p19 p21         p23          p25
## 1    0  0  0  0   0 0.0000000000 0.000000000   0 0.000000000 0.000000e+00
## 2    0  0  0  0   0 0.0000000000 0.000000000   0 0.000000000 0.000000e+00
## 3    0  0  0  0   0 0.0002577082 0.000387379   0 0.002039362 1.118518e-05
## 4    0  0  0  0   0 0.0000000000 0.000000000   0 0.000000000 0.000000e+00
## 5    0  0  0  0   0 0.0000000000 0.000000000   0 0.000000000 0.000000e+00
## 6    0  0  0  0   0 0.0000000000 0.000000000   0 0.000000000 0.000000e+00
##        s18         s19          s20         s22         s23          s25
## 1 0.000000 0.000000000 0.0000000000 0.000000000 0.000000000 1.852524e-06
## 2 0.000000 0.000000000 0.0000000000 0.000000000 0.000000000 0.000000e+00
## 3 0.000000 0.000000000 0.0000000000 0.000000000 0.000000000 0.000000e+00
## 4 0.000000 0.000000000 0.0000000000 0.000000000 0.000000000 0.000000e+00
## 5 0.000908 0.001827386 0.0001787903 0.001357659 0.006898081 1.028760e-05
## 6 0.000000 0.000000000 0.0000000000 0.000000000 0.000000000 0.000000e+00
##            s27          s29           sA           sB           sC
## 1 0.0000000000 0.0000000000 0.000000e+00 0.0000000000 0.0000000000
## 2 0.0000000000 0.0000000000 0.000000e+00 0.0000000000 0.0000000000
## 3 0.0003038009 0.0001284872 4.192012e-06 0.0002989824 0.0002913146
## 4 0.0000000000 0.0000000000 1.911700e-06 0.0000000000 0.0000000000
## 5 0.0000000000 0.0000000000 9.900839e-07 0.0000000000 0.0000000000
## 6 0.0002136276 0.0012268672 0.000000e+00 0.0000000000 0.0000000000
##             sD           sE           sF           sG         w22
## 1 0.000000e+00 0.000000e+00 0.0000000000 0.000000e+00 0.000000000
## 2 0.000000e+00 0.000000e+00 0.0000000000 0.000000e+00 0.000000000
## 3 3.828549e-06 5.006755e-04 0.0002211918 9.539861e-06 0.000000000
## 4 0.000000e+00 0.000000e+00 0.0000000000 0.000000e+00 0.000000000
## 5 0.000000e+00 6.395808e-05 0.0006552848 0.000000e+00 0.004300918
## 6 0.000000e+00 0.000000e+00 0.0000000000 0.000000e+00 0.000000000
##          w23          w25          w27          w29           wA
## 1 0.00000000 0.000000e+00 0.0000000000 0.0000000000 0.000000e+00
## 2 0.00000000 0.000000e+00 0.0000000000 0.0003056442 0.000000e+00
## 3 0.00000000 0.000000e+00 0.0000000000 0.0000000000 0.000000e+00
## 4 0.00000000 0.000000e+00 0.0000000000 0.0000000000 0.000000e+00
## 5 0.01216004 1.735197e-05 0.0006601373 0.0044809180 3.747803e-06
## 6 0.00000000 0.000000e+00 0.0000000000 0.0000000000 0.000000e+00
##             wB           wC           wD
## 1 0.000000e+00 0.0000000000 0.000000e+00
## 2 0.000000e+00 0.0000000000 0.000000e+00
## 3 2.808078e-04 0.0008223285 4.926941e-05
## 4 0.000000e+00 0.0000000000 0.000000e+00
## 5 8.794236e-05 0.0000000000 7.399354e-06
## 6 0.000000e+00 0.0000000000 0.000000e+00
```

Second, I mutiply the pi (proportion of individuals belonging to the ith species (wikipedia)) by log pi, to get the Shannon diversity index for each species. After that, I can easily do the calculation of the real Shannon diversity index. In these process, I do the sumation, use “na.rm=TRUE” to let r not to sum the NAs. Also, I multiply Shannon.sum by -1 because it’s in the equation of Shannon diversity index.

```
Shannon <- density.propotion*log(density.propotion)
Shannon.sum <- apply(Shannon, MARGIN=2, FUN = sum, na.rm=TRUE)
Shannon.sum.positive <- Shannon.sum*-1
Shannon.sum.positive
```

```
##       X.p1         p3         p4         p6        p13        p16 
## 0.16469465 0.06229167 0.07636589 0.25321949 0.11446908 0.10110450 
##        p19        p21        p23        p25        s18        s19 
## 0.23145994 0.13705752 0.15637975 0.20678515 0.16107689 0.19324929 
##        s20        s22        s23        s25        s27        s29 
## 0.15841820 0.24547149 0.15081181 0.18038614 0.18957190 0.12323629 
##         sA         sB         sC         sD         sE         sF 
## 0.17266506 0.14366028 0.10490866 0.13361624 0.24555600 0.20697436 
##         sG        w22        w23        w25        w27        w29 
## 0.12548789 0.12755105 0.10595493 0.13185746 0.19128235 0.11006828 
##         wA         wB         wC         wD 
## 0.17675987 0.19607985 0.16499248 0.21005275
```

3. Find dominant species (species >=2% of total composition in any cruise-station) and calculate the average density for the spring, summer, and winter cruise for each dominant species.

dominant species

First, I sum up all the propotion in every station (“MARGIN=1” = by row) for each species, and add this vector to the table that contains the density of each species for each cruise-station.

```
density.propotion.sum <- apply(density.propotion, MARGIN=1, FUN = sum) # species sum
density.each.propotion.sum.add <- cbind(density.each,density.propotion.sum)
head(density.each.propotion.sum.add)
```

```
##   X.p1 p3 p4 p6 p13     p16     p19 p21      p23      p25     s18      s19
## 1    0  0  0  0   0  0.0000  0.0000   0   0.0000 0.000000  0.0000   0.0000
## 2    0  0  0  0   0  0.0000  0.0000   0   0.0000 0.000000  0.0000   0.0000
## 3    0  0  0  0   0 17.4196 26.1846   0 137.8492 0.756054  0.0000   0.0000
## 4    0  0  0  0   0  0.0000  0.0000   0   0.0000 0.000000  0.0000   0.0000
## 5    0  0  0  0   0  0.0000  0.0000   0   0.0000 0.000000 61.3756 123.5208
## 6    0  0  0  0   0  0.0000  0.0000   0   0.0000 0.000000  0.0000   0.0000
##       s20   s22      s23      s25     s27     s29       sA      sB      sC
## 1  0.0000  0.00   0.0000 0.125220  0.0000  0.0000 0.000000  0.0000  0.0000
## 2  0.0000  0.00   0.0000 0.000000  0.0000  0.0000 0.000000  0.0000  0.0000
## 3  0.0000  0.00   0.0000 0.000000 20.5352  8.6850 0.283356 20.2095 19.6912
## 4  0.0000  0.00   0.0000 0.000000  0.0000  0.0000 0.129220  0.0000  0.0000
## 5 12.0852 91.77 466.2708 0.695383  0.0000  0.0000 0.066924  0.0000  0.0000
## 6  0.0000  0.00   0.0000 0.000000 14.4400 82.9292 0.000000  0.0000  0.0000
##         sD      sE      sF      sG      w22      w23      w25     w27
## 1 0.000000  0.0000  0.0000 0.00000   0.0000   0.0000 0.000000  0.0000
## 2 0.000000  0.0000  0.0000 0.00000   0.0000   0.0000 0.000000  0.0000
## 3 0.258788 33.8428 14.9513 0.64484   0.0000   0.0000 0.000000  0.0000
## 4 0.000000  0.0000  0.0000 0.00000   0.0000   0.0000 0.000000  0.0000
## 5 0.000000  4.3232 44.2935 0.00000 290.7174 821.9488 1.172894 44.6215
## 6 0.000000  0.0000  0.0000 0.00000   0.0000   0.0000 0.000000  0.0000
##        w29      wA      wB      wC       wD density.propotion.sum
## 1   0.0000 0.00000  0.0000  0.0000 0.000000          1.852524e-06
## 2  20.6598 0.00000  0.0000  0.0000 0.000000          3.056442e-04
## 3   0.0000 0.00000 18.9810 55.5847 3.330330          5.610053e-03
## 4   0.0000 0.00000  0.0000  0.0000 0.000000          1.911700e-06
## 5 302.8844 0.25333  5.9444  0.0000 0.500154          3.361889e-02
## 6   0.0000 0.00000  0.0000  0.0000 0.000000          1.440495e-03
```

Second, I pick up the rows that species >=2%.

```
dominant <- subset(density.each.propotion.sum.add, density.propotion.sum >= 0.02, select=c(1:35))
```

Third, I do the sumation for each species in each season. Eventually, use “avg” table to show the results.

```
springsum <- apply(dominant, MARGIN=1, FUN = sum, select=c(1:10) )
springavg <- springsum/10
summersum <- apply(dominant, MARGIN=1, FUN = sum, select=c(11:25) )
summeravg <- summersum/15
wintersum <- apply(dominant, MARGIN=1, FUN = sum, select=c(26:34) )
winteravg <- wintersum/9
avg <- cbind(springavg, summeravg, winteravg)
avg
```

```
##     springavg summeravg winteravg
## 5    232.7478  169.4985  282.4975
## 14   461.5675  322.0450  536.7416
## 15   166.1722  125.1148  208.5246
## 60   432.1845  302.4563  504.0938
## 72   179.2770  133.8513  223.0855
## 84   264.0621  190.3747  317.2912
## 85  1628.9125 1100.2750 1833.7917
## 88   481.3737  335.2491  558.7486
## 112  238.4705  173.3136  288.8561
## 126  187.8175  139.5450  232.5750
## 142  264.3490  190.5660  317.6100
## 169  311.3546  221.9030  369.8384
```