Computer Intensive Statistics in Ecology – HW1

R06546049 Pei-En Lai

2018/3/1

Setting Path and Importing Data Sets

Before importing these data sets from my computer, let's setup the path by using setwd(). By using setwd(), I enable the computer to know which folder it should check to get the data sets I want. And then I think we should assign these data sets into some variables so that I can call them more easily next time I want.

```
setwd("C:\\Users\\Hugo\\Documents\\Courses\\Quantity_Related\\Computer_Intensive_Statistics_in_Ecology")
copepod.composition <- read.table("copepod_composition.txt", header = TRUE)
# To make the cop_density possible to mutiply with value in the other data set, I import it and extract its first column, and then se
t it as a vector.
cop.density <- as.vector(read.table("cop_density.txt", header = TRUE)[, 1])
```

Installing The "knitr" Package

In R Markdown, we can use the package "knitr" and "kableExtra" to display a data set, so let me install it first and then put it into my library.

```
library(knitr)
library(kableExtra)
```

Displaying Part of Data Sets by Using kable()

Since the data sets are a bit too long to display all of the values, let me just display the first 8 rows for each data set by using head(). To make them easily to recognize, I also set the captions for them.

```
length.head <- 1:8
kable(head(copepod.composition, length(length.head)), caption = "First 8 Rows of Copepod Composition", format = "html", align = 'l')
%>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

First 8 Rows of Copepod Composition

p1	p3	p4	p6	p13	p16	p19	p21	p23	p25	s18	s19	s20	s22	s23	s25	s27	s29	sA	sB	sC	sD	sE	sF
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	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
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	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
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	1.52	0.30	3.06	1.35	1.24	0.62	2.92	0.3
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	0.00	0.00	0.26	0.00	0.00	0.00	0.00	0.0
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	0.00	0.00	0.26	0.00	0.00	0.00	0.32	1.5
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	0.76	1.51	0.00	0.00	0.00	0.00	0.00	0.0
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	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
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	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0

```
kable(head(cop.density, length(length.head)), caption = "First 8 Rows of Cop Density", col.names = "Density", format = "html" , align
= 'l') %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

First 8 Rows of Cop Density

Density
1119
1153

Density
1719
855
1246
2123
1159
1497

Calculating The Copepod Density for Each Species for Each Cruise-Station

To iterate over two data sets, I use apply() instead of the for loop since R isn't that efficient to run a for loop and I'm pretty bad at writing any for loop also. apply() is the function that returns a vector or array or list of values obtained by applying a function to margins of an array or matrix.

```
# Let the density can be calculated directly
copepod.composition.per <- copepod.composition * 0.01

# Calculate copepod density for each species for each cruise station
# function(x) means the method I want to apply to the data, and here the data is copepod.composition.per.
# 1 means to calculate by row.
# To make dimnames' length the same, transpose it.
cop.density.e.e <- t(apply(copepod.composition.per, 1, function(x){cop.density * x}))

# Print out the result
kable(t(head(cop.density.e.e, length(length.head))), col.names = 1:length(length.head), format = "html", align = "l", digits = 3) %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

	1	2	3	4	5	6	7	8
p1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
p3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
p4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
p6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
p13	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
p16	0.000	0.000	4.671	0.000	0.000	0.000	0.000	0.000
p19	0.000	0.000	27.121	0.000	0.000	0.000	0.000	0.000
p21	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
p23	0.000	0.000	33.910	0.000	0.000	0.000	0.000	0.000
p25	0.000	0.000	15.552	0.000	0.000	0.000	0.000	0.000
s18	0.000	0.000	0.000	0.000	119.902	0.000	0.000	0.000
s19	0.000	0.000	0.000	0.000	29.640	0.000	0.000	0.000
s20	0.000	0.000	0.000	0.000	16.286	0.000	0.000	0.000
s22	0.000	0.000	0.000	0.000	195.277	0.000	0.000	0.000
s23	0.000	0.000	0.000	0.000	417.623	0.000	0.000	0.000
s25	18.996	0.000	0.000	0.000	94.347	0.000	0.000	0.000
s27	0.000	0.000	31.935	0.000	0.000	15.968	0.000	0.000
s29	0.000	0.000	14.469	0.000	0.000	72.827	0.000	0.000
sA	0.000	0.000	48.593	4.129	4.129	0.000	0.000	0.000

	1	2	3	4	5	6	7	8
sB	0.000	0.000	39.083	0.000	0.000	0.000	0.000	0.000
sC	0.000	0.000	47.988	0.000	0.000	0.000	0.000	0.000
sD	0.000	0.000	8.395	0.000	0.000	0.000	0.000	0.000
sE	0.000	0.000	160.366	0.000	17.574	0.000	0.000	0.000
sF	0.000	0.000	24.546	0.000	121.145	0.000	0.000	0.000
sG	0.000	0.000	17.780	0.000	0.000	0.000	0.000	0.000
w22	0.000	0.000	0.000	0.000	69.543	0.000	0.000	0.000
w23	0.000	0.000	0.000	0.000	135.497	0.000	0.000	0.000
w25	0.000	0.000	0.000	0.000	2.267	0.000	0.000	0.000
w27	0.000	0.000	0.000	0.000	1.773	0.000	0.000	0.000
w29	0.000	0.175	0.000	0.000	2.621	0.000	0.000	0.000
wA	0.000	0.000	0.000	0.000	0.051	0.000	0.076	0.076
wB	0.000	0.000	1.103	0.000	0.139	0.000	0.000	0.000
wC	0.000	0.000	0.291	0.000	0.000	0.000	0.000	0.000
wD	0.000	0.000	0.434	0.000	0.289	0.000	0.000	0.000

For Each Cruise-Station, Calculate The Species Richness (Number of Species) and Shannon Diversity Index

I Use the length() to extract elements greater than 0 in each station so that I can know the number of species in each station.

```
species.richness <- apply(copepod.composition.per, 2, function(x){length(x[x > 0])})
kable(species.richness, col.names = "Species Richness", format = "html", align = "l") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

	Species Richness
p1	6
p3	12
p4	8
p6	9
p13	31
p16	29
p19	43
p21	7
p23	46
p25	39
s18	39
s19	41
s20	32
s22	25

	Species Richness
s23	32
s25	41
s27	40
s29	23
sA	47
sB	49
sC	44
sD	46
sE	44
sF	38
sG	25
w22	18
w23	16
w25	24
w27	33
w29	27
wA	44
wB	54
wC	64
wD	48

Now, let's calculate the Shannon Diversity Index. The Shannon entropy quantifies the uncertainty (entropy or degree of surprise) associated with this prediction. It is most often calculated as follows: H' = ?????? pi * ln(pi), where pi is the proportion of characters belonging to the ith type of letter in the string of interest. Thus, I calculate through column this time to get every pi in each station. Also, since the log0 is meaningless, I also remove na value.

```
shannon.divers <- apply(copepod.composition.per, 2, function(x){-sum(x * log(x), na.rm = TRUE)})
kable(shannon.divers, col.names = "Shannon Diversity Index", format = "html", align = "l", digits = 3) %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

	Shannon Diversity Index
p1	1.081
p3	1.256
p4	1.045
p6	1.115
p13	2.145
p16	1.413
p19	2.582
p21	1.567
p23	2.995
p25	2.485

	Shannon Diversity Index
s18	2.841
s19	2.983
s20	2.569
s22	2.567
s23	2.879
s25	3.001
s27	2.803
s29	2.118
sA	3.106
sB	2.983
sC	2.816
sD	2.938
sE	3.021
sF	2.890
sG	1.692
w22	1.979
w23	1.616
w25	1.842
w27	2.580
w29	2.359
wA	2.613
wB	3.000
wC	3.214
wD	3.006

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.

Let's get the dominant species in each station first by using the combination of the slice and the logigic operator. When I get any repetitive species index, I just count it once in each season since I only want the know which species is dominant during every season instead of the appearing times. Yet before finding the dominant species, let's get the season of each station from the column namea of data by using some functions first.

```
c.names <- colnames(copepod.composition)
spring <- grep("p", c.names)
summer <- grep("s", c.names)
winter <- grep("w", c.names)
```

In spring, the dominant species are:

```
dominant.spe.p <- apply(copepod.composition.per[, spring] >= 0.02, 2, function(x){which(x == TRUE)})
sorted.dom.spe.p <- sort(unique(unlist(dominant.spe.p)))
kable(sorted.dom.spe.p, col.names = "Dominant Species", format = "html", align ="l") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Dominant Species

Dominant Species
3
14
16
35
40
54
60
72
76
84
85
88
111
123
126
135
142
158
161
169

After finding the dominant speicies in spring, I want to check thier average densities in each season.

```
p.dominant.avg.p <- apply(cop.density.e.e[sorted.dom.spe.p, spring], 1, mean)
p.dominant.avg.s <- apply(cop.density.e.e[sorted.dom.spe.p, summer], 1, mean)
p.dominant.avg.w <- apply(cop.density.e.e[sorted.dom.spe.p, winter], 1, mean)
p.dominant.cbind <- cbind(p.dominant.avg.p, p.dominant.avg.s, p.dominant.avg.w)
rownames(p.dominant.cbind) <- sorted.dom.spe.p

kable(p.dominant.cbind, col.names = c("Spring", "Summer", "Winter"), format = "html", align ="l", digits = 3) %>%
        kable_styling(bootstrap_options = c("striped", "hover"))
```

	Spring	Summer	Winter
3	8.125	26.210	0.203
14	86.466	2.500	4.603
16	4.501	1.904	0.137
35	11.927	63.691	0.672
40	11.446	23.416	0.681
54	4.873	40.485	0.229
60	28.353	41.598	23.154
72	10.711	156.407	0.996
76	5.550	3.362	0.000

	Spring	Summer	Winter
84	54.093	18.917	9.400
85	626.171	414.205	18.136
88	11.223	419.718	0.038
111	9.938	4.661	0.005
123	28.781	83.105	0.991
126	115.089	38.427	0.000
135	7.362	197.810	0.000
142	161.229	6.446	0.000
158	3.378	16.906	0.011
161	21.429	0.000	0.000
169	33.968	341.827	1.254

And again, getting dominant species in summer:

```
dominant.spe.s <- apply(copepod.composition.per[, summer] >= 0.02, 2, function(x){which(x == TRUE)})
sorted.dom.spe.s <- sort(unique(unlist(dominant.spe.s)))
kable(sorted.dom.spe.s, col.names = "Dominant Species", format = "html", align ="l") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Dominant Species
3
5
15
20
35
40
51
54
60
72
73
79
80
81
84
85
86
88
106
112

Dominant Species
117
118
120
123
126
135
145
147
148
151
158
164
165
169

Again, after finding the dominant speicies in summer, I check thier average densities in each season.

```
s.dominant.avg.p <- apply(cop.density.e.e[sorted.dom.spe.s, spring], 1, mean)
s.dominant.avg.s <- apply(cop.density.e.e[sorted.dom.spe.s, summer], 1, mean)
s.dominant.avg.w <- apply(cop.density.e.e[sorted.dom.spe.s, winter], 1, mean)
s.dominant.cbind <- cbind(s.dominant.avg.p, s.dominant.avg.s, s.dominant.avg.w)
rownames(s.dominant.cbind) <- sorted.dom.spe.s

kable(s.dominant.cbind, col.names = c("Spring", "Summer", "Winter"), format = "html", align ="l", digits = 3) %>%
        kable_styling(bootstrap_options = c("striped", "hover"))
```

	Spring	Summer	Winter
3	8.125	26.210	0.203
5	0.000	67.728	23.576
15	3.426	204.692	1.267
20	1.274	42.367	0.170
35	11.927	63.691	0.672
40	11.446	23.416	0.681
51	0.000	28.013	0.864
54	4.873	40.485	0.229
60	28.353	41.598	23.154
72	10.711	156.407	0.996
73	0.307	31.915	0.054
79	1.459	26.637	0.000
80	3.744	9.886	0.000
81	1.760	13.302	0.005
84	54.093	18.917	9.400

	Spring	Summer	Winter
85	626.171	414.205	18.136
86	0.000	107.269	0.025
88	11.223	419.718	0.038
106	0.768	20.458	4.463
112	3.691	267.765	0.970
117	0.473	134.511	0.000
118	0.237	53.422	0.000
120	1.248	16.453	1.225
123	28.781	83.105	0.991
126	115.089	38.427	0.000
135	7.362	197.810	0.000
145	1.753	97.267	0.086
147	0.000	75.731	1.564
148	0.973	21.038	0.023
151	0.000	39.107	0.186
158	3.378	16.906	0.011
164	1.123	116.336	0.950
165	1.915	14.383	0.154
169	33.968	341.827	1.254

And again, getting dominant species during winter:

```
dominant.spe.w <- apply(copepod.composition.per[, winter] >= 0.02, 2, function(x){which(x == TRUE)})
sorted.dom.spe.w <- sort(unique(unlist(dominant.spe.w)))
kable(sorted.dom.spe.w, col.names = "Dominant Species", format = "html", align ="l") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Dominant Species
3
5
14
15
16
35
40
51
52
54
55
60

Dominant Species
72
84
85
106
112
120
123
147
166
169

Again, after finding the dominant speicies in winter, I check thier average densities in each season.

```
w.dominant.avg.p <- apply(cop.density.e.e[sorted.dom.spe.w, spring], 1, mean)
w.dominant.avg.s <- apply(cop.density.e.e[sorted.dom.spe.w, summer], 1, mean)
w.dominant.avg.w <- apply(cop.density.e.e[sorted.dom.spe.w, winter], 1, mean)
w.dominant.cbind <- cbind(w.dominant.avg.p, w.dominant.avg.s, w.dominant.avg.w)
rownames(w.dominant.cbind) <- sorted.dom.spe.w

kable(w.dominant.cbind, col.names = c("Spring", "Summer", "Winter"), format = "html", align ="l", digits = 3) %>%
        kable_styling(bootstrap_options = c("striped", "hover"))
```

	Spring	Summer	Winter
3	8.125	26.210	0.203
5	0.000	67.728	23.576
14	86.466	2.500	4.603
15	3.426	204.692	1.267
16	4.501	1.904	0.137
35	11.927	63.691	0.672
40	11.446	23.416	0.681
51	0.000	28.013	0.864
52	0.000	5.324	1.412
54	4.873	40.485	0.229
55	0.301	0.727	0.421
60	28.353	41.598	23.154
72	10.711	156.407	0.996
84	54.093	18.917	9.400
85	626.171	414.205	18.136
106	0.768	20.458	4.463
112	3.691	267.765	0.970
120	1.248	16.453	1.225
123	28.781	83.105	0.991

	Spring	Summer	Winter
147	0.000	75.731	1.564
166	3.540	4.435	0.263
169	33.968	341.827	1.254