Todo list

EDUARD SZÖCS

QUANTITATIVE ECOTOXICOLOGY

WITH R!

This document was created using LaTeX, knitr and the tufte book class.

First draft, December 2013

Contents

List of Figures

List of Tables

Preface

1
Introduction

```
require(devtools)
install_github("qetx", "EDiLD")
```

```
require(qetx)
```

2
The Measurement Process

2.1 Winsorized Mean and Standard Deviation

Example data

The following sulfate concentrations (mg/L) were measured during a routine water quality survey of the Savannah River (South Carolina). The data is available in the `qetx` package [1]:

[1] Note that in this case you do not have to assign the data to a name.

```
data(so4)
```

```
so4

##  [1] 1.3 2.3 2.6 3.3 3.5 3.5 3.6 4.0 4.1 4.5 5.2 5.6
## [13] 5.7 6.1 6.2 6.5 6.9 7.1 7.7 7.9 9.9

length(so4)

## [1] 21

mean(so4)

## [1] 5.119

sd(so4)

## [1] 2.137
```

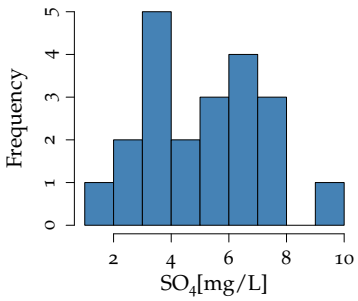


Figure 2.1: A histogramm of the so4 data.

So there are 21 measurements with a mean of 5.12 mg/L and a standard deviation of 2.14 mg/L.

Winsorization

Suppose we have a detection limit of 2.5 mg/L and want to winsorize values below LOD, i.e. replace the two lowest values by 2.6 mg/L and the two highest values by 7.7 mg/L.

Happily there is function in the `qetx`-package to do this for us: `winsor()`. This function takes a vector of values and a second argument specifying how many values should be winsorized (either by giving a LOD-value or the number of values on each side) [2].

[2] Look at the source of this function - type the function name into the console - to see which computations are done.

```
so4_w <- winsor(so4, lod = 2.5)
so4_w

##  [1] 2.6 2.6 2.6 3.3 3.5 3.5 3.6 4.0 4.1 4.5 5.2 5.6
## [13] 5.7 6.1 6.2 6.9 6.5 7.1 7.7 7.7 7.7
## attr(,"width")
## [1] 2
```

This give the expected results, moreover we see that on each end two observations where modified [3].

[3] stored within the attribute 'width' of the resulting vector. **TODO: verbatim within sidenote.**

```
mean(so4_w)

## [1] 5.081

sd(so4_w)

## [1] 1.792

sd_winsor(so4_w)

## [1] 2.24
```

The Winsorized mean ($\bar{x}_w$) now is 5.08 mg/L, the standard deviation of the modified data set (s) is 1.79 mg/L and the Winsorized standard deviation (s_w) 2.24 mg/L.

2.2 Probability Plotting

3
Bioaccumulation

4
Tests for Detection of Chronic Lethal and Sub-lethal Stress

5
Lethal and Other Quantal Responses to Stress

Figure 5.1: LD50. Source: http://xkcd.com/1260/

5.1 Fitting dose-response models

Example data

Newman and Aplin (1992) exposed mosquitofish *Gambusia holbrooki* to a series of NaCl concentrations for 96h. The data is available from the `qetx` package.

```
data(salt)
str(salt)

## 'data.frame': 6 obs. of  3 variables:
##  $ dead : int  16 22 40 69 78 77
##  $ total: int  76 79 77 76 78 77
##  $ conc : num  10.3 10.8 11.6 13.2 15.8 20.1
```

There are three columns:

dead Number of fish died.

total Total number of fish exposed.

cocn NaCl concentration (g/L).

First we calculate the proportion of died fish and safe it as a new column in our data.frame:

```
salt$prop <- salt$dead/salt$total
```

As always we first take a look at the data, to produce Fig. 5.2:

```
plot(prop ~ conc, data = salt, log = 'x', pch = 16,
     xlab = "NaCl-Concentration", ylab = "Proportion dead")
```

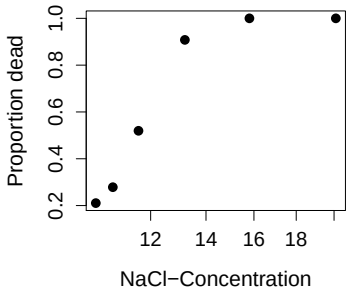


Figure 5.2: Proportion of fish died at different NaCl concentrations. The x axis is on a log scale.

Introduction

[1] Ritz, 2010

6
Population and Metapopulation Effects

7
Community Effects

7.1 Species Richness

Example data

Ten species were sampled at eight sites around an outfall and the number of individuals per species noted. The data is available from the `qetx` package has a usual species x sites format and the row-names give the distance to the outfall[1]:

[1] Most functions for multivariate techniques need the data in this format - rows = samples, columns = species.

```
data(abu)
abu
```

```
##      Sp1 Sp2 Sp3 Sp4 Sp5 Sp6 Sp7 Sp8 Sp9 Sp10
## -1    12  11  10   8   8   5   2   2   1    0
## -0.5  12  12  10   8   8   4   3   2   2    1
## 0     58  21   3   2   1   0   0   0   0    0
## 0.5   18  16  15   5   4   0   0   0   0    0
## 1     11  11  11   8   7   3   2   1   1    0
## 1.7    8  12  13   3   3   3   1   2   0    0
## 2.7   12  10   8  11   8   4   1   1   2    0
## 5.3   10  11   8   6   6   5   4   2   1    1
```

For our analyses we will use two packages:

vegan A package for Community Ecology.

BiodiversityR A package forbiodiversity, suitability and community ecology analysis[2]

[2] Advisable further reading: Kindt and Coe (2005) freely availabe at `http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis`

```
require(vegan)
require(BiodiversityR)
```

Rarefaction

The total number of individuals per sample can be calculated as the row sums

```
rowSums(abu)

##   -1 -0.5    0  0.5    1  1.7  2.7  5.3
##   59   62   85   58   55   45   57   54
```

As the total number of individuals varies we will use the rarefaction method to estimate species richness at a sample size of 40 individuals.

```
rarefy(abu, 40, se = TRUE)

##        -1  -0.5      0     0.5      1    1.7    2.7
## S  8.4757 9.347 4.0496 4.99092 8.3665 7.8767 8.3122
## se 0.6225 0.687 0.7458 0.09492 0.6788 0.3313 0.6962
##       5.3
## S  9.4139
## se 0.6572
## attr(,"Subsample")
## [1] 40
```

As can be seen species richness drops at the outfall, but increases again downstream (Fig. 7.1). Rarefaction curves can be easily created as well as species accumulation curves:

```
rarecurve(abu, sample = 40)

env <- data.frame(dist = rownames(abu))
rankabuncomp(abu, env, 'dist', scale = 'accumfreq')
```

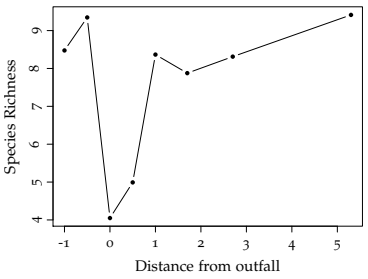


Figure 7.1: Species Richness along outfall

7.2 Species Diversity

Function `diversity()` from the vegan package can calculate the Shannon and Simpson diversity indices [3]:

```
diversity(abu, base = 2)

##        -1      -0.5         0       0.5         1
## 2.8673512 2.9862404 1.2475503 2.1119181 2.7892213
##       1.7       2.7       5.3
## 2.5720769 2.8102132 3.0159575
```

[3] The logarithm with base 2 is used for the Shannon index

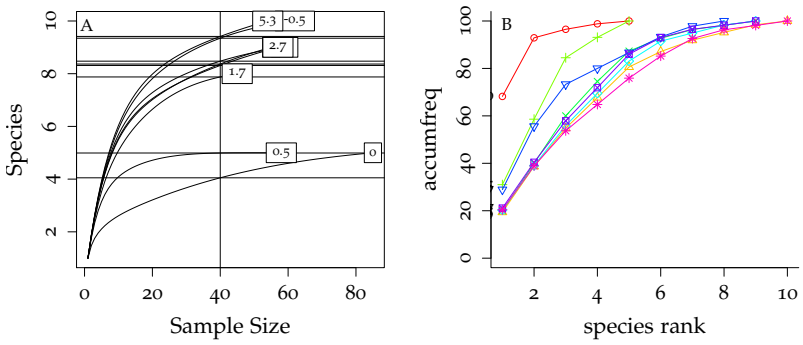


Figure 7.2: A) Rarefaction Curve. B) Rank abundance curve

```
diversity(abu, index = "simpson")

##         -1       -0.5          0        0.5          1
## 0.84860672 0.85691988 0.47141869 0.74851367 0.83768595
##        1.7        2.7        5.3
## 0.79802469 0.84148969 0.86145405
```

Brillouin's index is not implemented so have to calculate it by hand:

```
N <- rowSums(abu)
log(factorial(N)/apply(factorial(abu), 1, prod), base = 2)/N

##        -1      -0.5         0       0.5         1
## 2.5566733 2.6578637 1.1447406 1.9283402 2.4710966
##       1.7       2.7       5.3
## 2.2413558 2.4986847 2.6527313
```

Or use the function `brillouin` from the `qetx` package, which runs the two above lines of code for us:

```
brillouin(abu, 2)
```

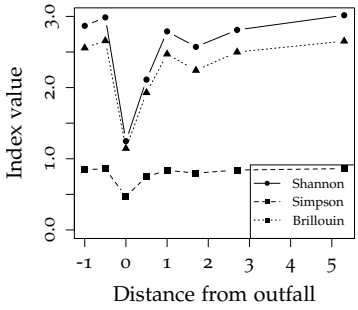


Figure 7.3: Diversity indices along outfall

Species Eveness

Pielou's J can be easyily calculated from the shannon index by dividing it with the number of species in each sample:

```
diversity(abu)/log(rowSums(abu > 0))

##         -1       -0.5          0        0.5          1
## 0.90454859 0.89894792 0.53729068 0.90955361 0.87990134
```

```
##        1.7        2.7        5.3
## 0.85735897 0.88652356 0.90789368
```

7.3 Dissimilarity indices

```
data(oil)
oil

##    Abra nitida Abra tenuis Acanthocardia echinata
## 1            0           1                      0
## 2            0           0                      2
## 3            0           0                      0

....

## 19                0                   0
## 20                2                   0
```

7.4 Ordination techniques - PCA, RDA, NMDS, MDS, dbRDA

4

[4] Further reading: Borcard et al. (2011)

7.5 Testing hypotheses - PERMANOVA

7.6 Analyzing mesocosm data - Principal Response Curves (PRC)

Example data

Here we will analyze the pyrifos data set from Van den Brink and Ter Braak (1999) which is shipped with the vegan package.

Describe experiment

```
require(vegan)
data(pyrifos)
head(pyrifos[, c(1:8)])

##        Simve Daplo Cerpu Alogu Aloco Alore Aloaf Copsp
## w.4.c1 3.951     0     0     0     0     0     0 2.773
## w.4.c2 2.303     0     0     0     0     0     0 2.079
## w.4.c3 4.595     0     0     0     0     0     0 3.761
## w.4.c4 2.398     0     0     0     0     0     0 3.296
## w.4.c5 4.025     0     0     0     0     0     0 3.466
## w.4.c6 2.303     0     0     0     0     0     0 2.197
```

So rows correspond to samples and columns are the species (with abbreviated names), a usual species x sites matrix. The column names code treatment and time, but we will create a separate data.frame with information about experimental ditch, sampling time and treatment:

```
ditch <- gl(12, 1, length = 132)
week <- gl(11, 12, labels = c(-4, -1, 0.1, 1, 2, 4, 8, 12,
    15, 19, 24))
dose <- factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44,
    0.9, 0, 6), 11))
pyrifos_env <- data.frame(ditch, week, dose)
```

Introduction

Principal Response Curves (PRC) [5] are commonly used for analyzing ecotoxicological mesocosm experiments. PRC analyses the change of a community due to a treatment over time and is a special form of Redundancy Analysis (RDA) [6].

[5] Van den Brink and Ter Braak, 1999

[6] **?**

Overall pattern

With this a hand we can easily calculate and plot (Figure 7.4) [7] the PRC using the prc() function:

[7] Note that only species with scores greater or smaller than 1 are displayed to avoid cluttering of the plot

```
pyrifos_prc <- prc(response = pyrifos, treatment = dose,
    time = week)
pyrifos_prc_sum <- summary(pyrifos_prc, scaling = 1)
```

```
plot(pyrifos_prc, select = abs(pyrifos_prc_sum$sp) > 1,
    scaling = 1)
```

The plot shows on the x axis the time and on the y-axis the difference from the control treatments. The farther apart from the x-axis the more different are the communities compared to the control (you can say the x axis represents the control).

We see a clearly treatment-related effect: After application at week 0 the treated communities rapidly change treatment dependent. However to the end of the experiment the treated and the control get similar again, which indicates a 'recovery'.

On the right-hand side we see the species names and their scores. The more extreme the scores the more this species contributed to the plotted pattern. However, you cannot directly infer from these

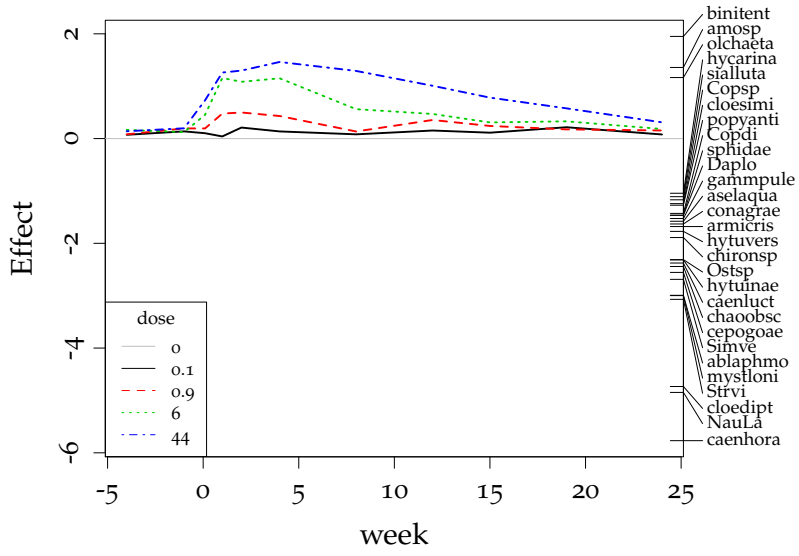


Figure 7.4: Principal response curves (PRC) with species weights for the pyrifos data set indicating effects of the insecticide on the invertebrate community.

species scores which species are more susceptible. For example *Gammarus pulex* (gammapule) has a relatively low scores, although it's response pattern (Figure 7.5) shows a strong response, but with no recovery. PRC displays global pattern in the community, but the pattern of *G. pulex* is different from most other species, therefore it gets a lower species score.

We can also look at the numerical output[8] for this plot using the summary method:

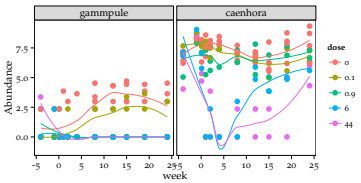


Figure 7.5: Responses of *G. pulex* and *C. horaria* to chlorpyrifos.

[8] Only a shortened output is given here.

```
pyrifos_prc_sum
```

```
##
## Call:
## prc(response = pyrifos, treatment = dose, time = week)
## Species scores:
##    Simve    Ostsp    NauLa    Strvi binitent caenhora
##   -2.688   -2.312   -4.847   -3.070    1.951   -5.768
## caenluct cloedipt hytuinae ablaphmo cepogoae chaoobsc
##   -2.376   -4.734   -2.316   -2.993   -2.555   -2.442
## mystloni
##   -2.998
##
## Coefficients for dose + week:dose interaction
```

```
## which are contrasts to dose 0
## rows are dose, columns are week
##          -4     -1    0.1       1      2      4
## 0.1 0.07218 0.1375 0.1020 0.04068 0.2101 0.1364
## 0.9 0.08106 0.1935 0.1936 0.47699 0.4977 0.4306
## 6   0.16616 0.1232 0.4539 1.15638 1.0835 1.1511
## 44  0.13979 0.1958 0.7308 1.26088 1.2978 1.4627
```

The output of `prc()` gives us more detailed information about the RDA model:

```
pyrifos_prc
```

```
## Call: prc(response = pyrifos, treatment = dose,
## time = week)
##
##                 Inertia Proportion Rank
## Total         288.99200    1.00000
## Conditional    63.34900    0.21921   10
## Constrained    96.68400    0.33456   44
## Unconstrained 128.95900    0.44624   77
## Inertia is variance
##
## Eigenvalues for constrained axes:
##    RDA1    RDA2    RDA3    RDA4    RDA5    RDA6
## 25.2823  8.2969  6.0442  4.7662  4.1482  3.8568
##    RDA7    RDA8
##  3.5875  3.3341
##
## Eigenvalues for unconstrained axes:
##     PC1     PC2     PC3     PC4     PC5     PC6
## 17.1561  9.1894  7.5849  6.0636  5.7298  4.8434
##     PC7     PC8
##  4.5184  4.1046
## (Showed only 8 of all 77 unconstrained eigenvalues)
```

We see that 21.9 % of the variance can be attributed to time (Conditional), 33.5 % can be explained by the treatment regime (Constrained) and 44.6 % of residual variance (Unconstrained), which cannot be explained by time and treatment.

The first RDA axis has an eigenvalue of 25.3. If we divide this eigenvalue by the sum of all eigenvalues, we get the proportion of explained variance which is displayed on the first axis[9]:

[9] rda() (and therefore also prc()) returns a huge object with all kind of information stored in it. See ?cca.object for the internal structure. Here I directly access the eigenvalues from this object

```
pyrifos_prc$CCA$eig[1]/sum(pyrifos_prc$CCA$eig) * 100

##     RDA1
## 26.14946
```

The significance of the PRC diagram can be tested via permutations. However observations from a experimental ditch are not independent, since the same ditch was measured repeatedly during the experiment. We have to take this into account: each ditch represents a time-series. We will permute the whole series of one ditch, keeping the temporal order (see Tab. 7.1).

Table 7.1: 3 ditches observed for 4 weeks and a possible permutation.

Week	Ditch	Perm
1	1	3
2	1	3
3	1	3
4	1	3
1	2	1
2	2	1
3	2	1
4	2	1
1	3	2
2	3	2
3	3	2
4	3	2

To setup such a permutation scheme we use the permute package, which is automatically loaded with vegan[10]:

```
control = how(plots = Plots(strata = ditch, type = "free"),
    within = Within(type = "none"), nperm = 99)
```

This sets up our permutation scheme:

plots We will permute ditches, without any restrictions.

within But within one ditch there will be no permutations.

nperm We want 99 permutations.

Permutation tests for PRC can be performed using `anova`:

```
set.seed(1234)
anova(pyrifos_prc, permutations = control, first = TRUE)

## Permutation test for rda under reduced model
## Plots: ditch, plot permutation: free
## Permutation: none
## Number of permutations: 99
##
## Model: prc(response = pyrifos, treatment = dose, time = week)
##          Df Variance       F Pr(>F)
## RDA1      1  25.2823 15.0958   0.01 **
## Residual 77 128.9589
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This runs a permutation test for the first eigenvalue of our model. We see that our first axis explains a statistically significant proportion of variation (Fig. 7.6). The minimum p-value that we could get is 0.01 (=1/no. of permutations).

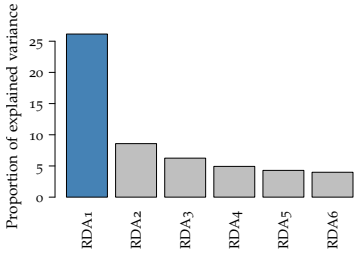


Figure 7.6: Proportion of explained variance of the first 6 RDA-axes. Note that treatment and time were factors and (internally) dummy-coded, therefore we have a total of 44 axes.

[10] vegan is in active development and you should always use the newest version.

You can also test the significance of each axis separately[11] [12]

[11] Legendre et al., 2011

[12] The results for the first axis are identical, but testing only the first eigenvalue is computationally more efficient since we are generally not interested in all axes.

```
anova(pyrifos_prc, permutations = control, by = "axis")
```

or the terms in the model:

```
anova(pyrifos_prc, permutations = control, by = "terms")

## Permutation test for rda under reduced model
## Terms added sequentially (first to last)
## Plots: ditch, plot permutation: free
## Permutation: none
## Number of permutations: 99
##
## Model: rda(formula = pyrifos ~ dose * week + Condition(week))
##           Df Variance       F Pr(>F)
## dose       4  30.6951 4.58193   0.04 *
## week       0   0.0000
## dose:week 40  65.9886 0.98503   0.05 *
## Residual  77 128.9589
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This shows that the interaction term in the model is statistically significant. Note that no output for week is generated - this is because with the specified permutation scheme the significance of a week effect cannot be assessed.

Effects per week

Now that we have looked at the overall effects, we may want to look at effects at individual time-points and explore the nature of the treatment x time interaction.

We follow here [13] and use the ln-transformed nominal dose as continuous explanatory variable [14].

[13] Van den Brink and Ter Braak, 1999

[14] But before we have to convert dose from a factor to a numeric vector via as.numeric(levels(x))[x]

```
dose_c <- log(20 * as.numeric(levels(dose))[dose] + 1)
```

Now we can write a for-loop and compute for every week a RDA and a permutation test [15].

[15] First we create an empty object (rdas) that will hold our results. Next we run on a subset of data (based on week) a RDA and permutation test. The results of both are stored as a list entry.

```
rdas <- NULL
for (i in levels(week)) {
  rdas[[i]]$rda <- rda(pyrifos[week == i, ] ~
                         dose_c[week == i])
  rdas[[i]]$anova <- anova(rdas[[i]]$rda, by = 'terms',
```

```
                              step = 199)
}
```

However there is also convenience function in the `qetx` package:

```
rdas <- rda_per_time(pyrifos, dose_c, week)
```

This returns a very big list: one list entry per week and each entry itself contains two lists: `rda` (RDA-Model) and `anova` (permutation test)). Digging through this manually is to .... therefore we need to extract only the information we need.

We can use `sapply()` to apply a function to every list entry and return results in a vector. For example to extract the p-values for each week we can use:

```
sapply(rdas, function(x) x$anova[1, 4])

##    -4    -1   0.1     1     2     4     8    12    15
## 0.334 0.343 0.039 0.001 0.007 0.001 0.001 0.001 0.001
##    19    24
## 0.001 0.001
```

Have a look at the object structure to write a custom function to extract the information you need:

```
str(rdas[[1]]$anova)
```

Other methods

Other methods to analyse mesocosm experiments include:

multivariate GLMs [16] In R: mvabund-package.

trait-based indicators [17] Currently no package, but have look at `rspear-package` (section 7.8).

community endpoints [18] You can use `vegan` for the computations.

[16] Warton et al., 2011; and Wang et al., 2012

[17] Liess and Beketov, 2011

[18] Sanchez-Bayo and Goka, 2012

7.7 Species Sensitivity Distributions (SSD)

Introduction

Species Sensitivity Distributions (SSD) are a central tool for risk assessment in ecotoxicology.

introduction

Example data

```
data(ec50)
```

Calculate SSD

We have data for a total of 35 taxa, with multiple EC50 values per taxon. We could aggregate/average these values into a single observation per taxon.

Here we aggregate using the geometric mean (with base 10).

```
logmean <- function(x) 10^(mean(log10(x)))
agg <- aggregate(ec50$conc, by = list(species = ec50$species),
    logmean)
```

The result is a data.frame with 35 species and the mean EC50/LC50 values.

We can now fit a parametric distribution to this data using the `fitdistrplus` package. The choice of distribution is a crucial one - to help the choice on can fit different distributions to the data and the the one that fits best to the data. Here we fit a normal, a log-normal, a logistic and a weibull distribution to the data:

```
require(fitdistrplus)
ft_norm <- fitdist(agg$x, distr = "norm")
ft_lnorm <- fitdist(agg$x, distr = "lnorm")
ft_logis <- fitdist(agg$x, distr = "logis")
ft_weib <- fitdist(agg$x, distr = "weibull")
```

To compare different distributions we can use Information Criteria like AIC or BIC. A graphical comparison may also help decision.

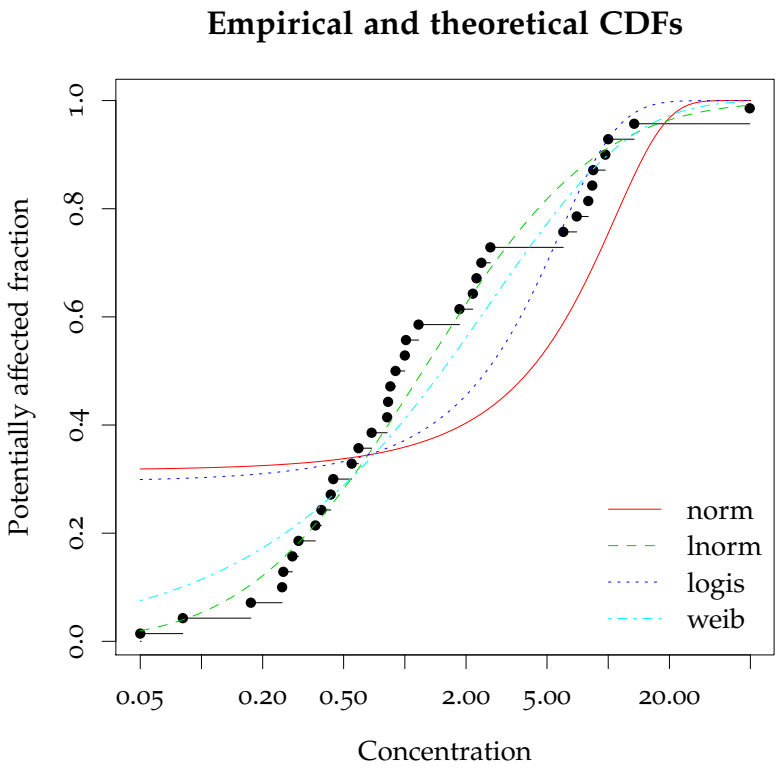
```
gofstat(list(ft_norm, ft_lnorm, ft_logis, ft_weib), fitnames = c("norm",
    "lnorm", "logis", "weib"))

## Goodness-of-fit statistics
##                               norm  lnorm  logis
## Kolmogorov-Smirnov statistic 0.3188 0.1201 0.2992
## Cramer-von Mises statistic   1.0916 0.0797 0.6914
## Anderson-Darling statistic   5.8682 0.4702 3.9085
##                               weib
## Kolmogorov-Smirnov statistic 0.1575
## Cramer-von Mises statistic   0.1690
## Anderson-Darling statistic   0.9821
```

```
##
## Goodness-of-fit criteria
##                                norm lnorm logis  weib
## Aikake's Information Criterion 253.8 148.2 227.1 155.0
## Bayesian Information Criterion 256.9 151.3 230.2 158.1
```

```
cdfcomp(list(ft_norm, ft_lnorm, ft_logis, ft_weib), xlogscale = TRUE,
    legendtext = list("norm", "lnorm", "logis", "weib"),
    xlab = "Concentration", ylab = "Potentially affected fraction")
```

Figure 7.7: SSD



From the plot we can clearly see that the normal and logistic distributions are not appropiate here. AIC and BIC tell the same story and the log-normal distribution fits bests.

The fitted distribution has the following parameters:

```
ft_lnorm

## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters:
```

```
##         estimate Std. Error
## meanlog   0.2024     0.2620
## sdlog     1.5503     0.1853
```

The 5% quantile of this distribution is the HC5 (Figure 7.8):

```
quantile(ft_lnorm, 0.05)

## Estimated quantiles for each specified probability (non-censored data)
##          p=0.05
## estimate 0.0956
```

Confidence Intervals can be produced using bootstrap:

```
bt_lnorm <- bootdist(ft_lnorm, niter = 999)
quantile(bt_lnorm, 0.05)

## (original) estimated quantiles for each specified probability (non-censored data)
##          p=0.05
## estimate 0.0956
## Median of bootstrap estimates
##          p=0.05
## estimate 0.1034
##
## two-sided 95 % CI of each quantile
##         p=0.05
## 2.5 %  0.04625
## 97.5 % 0.22548
```

So the lower CI for the HC5 is 0.047 (Figure 7.8).

`fitdistrplus` allows also to use interval and censored data. However first we need to restructure our dataset:

```
require(plyr)
int <- ddply(ec50, .(species), summarise, left = min(conc),
    right = max(conc))
```

```
ft_lnorm_int <- fitdistcens(int, "lnorm")
quantile(ft_lnorm_int, 0.05)

## Estimated quantiles for each specified probability (censored data)
##          p=0.05
## estimate 0.1124
```

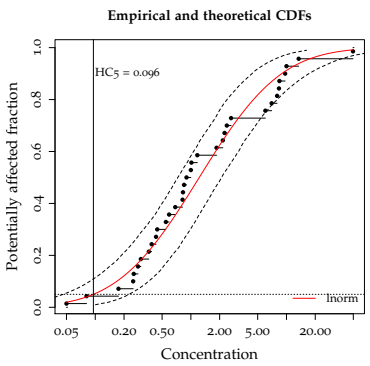


Figure 7.8: SSD

7.8 Species At Risk (SPEAR)

```
require(rspear)
```

R Session Info

```
sessionInfo()
```

```
## R version 3.0.2 (2013-09-25)
## Platform: x86_64-pc-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8
##  [2] LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8
##  [4] LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8
##  [6] LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8
##  [8] LC_NAME=C
##  [9] LC_ADDRESS=C
## [10] LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8
## [12] LC_IDENTIFICATION=C
##
## attached base packages:
##  [1] splines   tcltk     grid      stats     graphics
##  [6] grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] rspear_0.1-2        fitdistrplus_1.0-1
##  [3] survival_2.37-4     ggplot2_0.9.3.1.99
##  [5] reshape2_1.2.2      BiodiversityR_2.3-6
##  [7] tikzDevice_0.6.3    filehash_2.2-1
##  [9] qetx_0.0.1          vegan_2.1-40
## [11] lattice_0.20-24     permute_0.8-0
## [13] knitr_1.5
##
## loaded via a namespace (and not attached):
##  [1] codetools_0.2-8    colorspace_1.2-4
```

```
##  [3] dichromat_2.0-0    digest_0.6.4
##  [5] evaluate_0.5.1     formatR_0.10
##  [7] gtable_0.1.2       highr_0.3
##  [9] labeling_0.2       MASS_7.3-29
## [11] munsell_0.4.2      plyr_1.8
## [13] proto_0.3-10       Rcmdr_2.0-2
## [15] RColorBrewer_1.0-5 scales_0.2.3
## [17] stringr_0.6.2      tcltk2_1.2-9
## [19] tools_3.0.2
```

Bibliography

Borcard, D., Gillet, F., and Legendre, P. (2011). *Numerical Ecology with R (Use R)*. Springer, 1st edition. edition.

Kindt, R. and Coe, R. (2005). *Tree diversity analysis: [a manual and software for common statistical methods for ecological and biodiversity studies]*. World Agrofirestry Centre, Nairobi and Kenya.

Legendre, P., Oksanen, J., and ter Braak, C. J. F. (2011). Testing the significance of canonical axes in redundancy analysis. *Methods in Ecology and Evolution*, 2(3):269âĂŞ277.

Liess, M. and Beketov, M. (2011). Traits and stress: keys to identify community effects of low levels of toxicants in test systems. *Ecotoxicology*, 20(6):1328–1340.

Newman, M. C. and Aplin, M. S. (1992). Enhancing toxicity data interpretation and prediction of ecological risk with survival time modeling: an illustration using sodium chloride toxicity to mosquitofish (gambusia holbrooki). *Aquatic toxicology*, 23(2):85–96.

Ritz, C. (2010). Toward a unified approach to dose-response modeling in ecotoxicology. *Environmental Toxicology and Chemistry*, 29(1):220–229.

Sanchez-Bayo, F. and Goka, K. (2012). Evaluation of suitable endpoints for assessing the impacts of toxicants at the community level. *Ecotoxicology*, 21(3):667–80.

Van den Brink, P. and Ter Braak, C. (1999). Principal response curves: Analysis of time-dependent multivariate responses of biological community to stress. *Environmental Toxicology and Chemistry*, 18(2):138–148.

Wang, Y., Naumann, U., Wright, S. T., and Warton, D. I. (2012). mvabund- an r package for model-based analysis of multivariate abundance data. *Methods in Ecology and Evolution*, 3(3):471–474.

Warton, D. I., Wright, S. T., and Wang, Y. (2011). Distance-based multivariate analyses confound location and dispersion effects. *Methods in Ecology and Evolution*, 3(1):89–101.