

Using for statistical analyses

Robert Bauer

Warnemünde, 04/18/2012



Day 2 - Agenda:

- ▶ Last Exercise
- ▶ Defining Vectors
- ▶ aggregating and saving data
- ▶ Exercises
- ▶ Histograms
- ▶ multiple plots in a single window
- ▶ The “plot” function
- ▶ Exercises

Exercises

1. Import “Fish.csv” (mind not available values!)
2. Which was the maximum, which was the minimum size of each caught species?
3. Create box-plots of the length distribution of both species and both sexes
4. Create species specific box-plots of the full, empty and liver weight
5. Define a new vector: Hepta Somatic Index $HSI = \frac{liver\ weight * 100}{total\ weight}$
6. What is the median & the range of the HSI per species?

Results

```
# 1. Import "Fish.csv" (mind not available values!)
setwd('~/.Dropbox/R_course/day1')
fish <- read.table("Fish.csv", header=T, sep=',', dec=".") #
  load dataframe
head(fish)
attach(fish)

# 2. Which was the maximum, which was the minimum size of
  each caught species?
range(total.length[species == "dab"])
range(total.length[species == "cod"])
```

Results

```
# 3. Create box-plots of the length distribution of both
species and both sexes

boxplot(total.length[species == "dab"]~sex[species == "dab"]
, data = fish, las=1, names = c("male", "female"), ylab=
"total length [mm]", main = "dab")

boxplot(total.length[species == "cod"]~sex[species == "cod"]
, data = fish, las=1, names = c("male", "female", "
juvenile"), ylab="total length [mm]", main = "cod")
```

boxplots

```
length.j <- total.length[species == "cod" & sex == 0]
length.m <- total.length[species == "cod" & sex == 1]
length.f <- total.length[species == "cod" & sex == 2]

boxplot(length.f, length.m, length.j, las=1, names = c("
  female", "male", "juvenile"), ylab="total length [mm]",
  main = "cod")
```

Results

```
# 4. Create species specific box-plots of the full, empty
    and liver weight
boxplot(full.weight[species == "dab"], empty.weight[species
    == "dab"], liver.weight[species == "dab"], data = fish,
    las=1, names = c("full", "empty", "liver"), ylab="weight
    [g]", main = "dab")

boxplot(full.weight[species == "cod"], empty.weight[species
    == "cod"], liver.weight[species == "cod"], data = fish,
    las=1, names = c("full", "empty", "liver"), ylab="weight
    [g]", main = "cod")
```

Results

```
# 5. Define a new vector: Hepta Somatic Index
HSI <- liver.weight*100/full.weight

# 6. What is the median & the range of the HSI per species?
median(HSI[species == "dab"], na.rm=T)
median(HSI[species == "cod"], na.rm=T)

range(HSI[species == "dab"], na.rm=T)
range(HSI[species == "cod"], na.rm=T)
```


Defining Vectors

Function	Description	Example
c()	combining elements	c(1,2,3,4)
a:b	sequence	1:10; 10:1
seq(start, end, step size)	sequence	seq(10,20,2)
rep(vector, replicates)	replicating data	rep(0,10) rep(0:10,10)

Combination of commands:

```
test <- c(1:10, rep(0,10), seq(10,20,2))
```

Vectors and Data Types

```
# character vectors
xlab <- "sea urchins"
names <- c("available", "not available")

# logical vectors
logical <- c(TRUE, FALSE, TRUE, F, T)

# integer vectors
numbers <- c(1,2,3,4)

# numeric vectors
numbers <- c(10.2,67.2)
```

Function	Description
mode()	data type
str()	data type with levels
class()	class or type of an object

Combining Vectors

```
# combining vectors rowwise
c(xlab, names)
rbind(xlab, names)

# combining vectors as columns
cbind(xlab, names)
cbind(xlab, names, logical, numbers) # won't work since
    vectors have different numbers of rows

# defining new dataframes
data.frame("col1" = seq(1,10), 21:30)
```

aggregating and saving data

```
aggregate(total.length, by = list(species=species, sex=sex), FUN="
  max", na.rm=T)

max.lengths <- aggregate(total.length, by = list(species=species,
  sex=sex), FUN="max", na.rm=T)

write.table(max.lengths, "max.lengths.csv", sep=',', row.names=F)
```

Exercises

1. calculate the minimum full weight per sex and species using the aggregate function
2. calculate the maximum size & weight per sex and species using the aggregate function (hint: use bracket selection!)
3. save the results of ex1 and ex2 as dataframe

Results

```
# 1. calculate the minimum full weight per sex and species
      using the aggregate function
min.weight <- aggregate(full.weight, by = list(species=
      species, sex=sex), FUN="min", na.rm=T)

# 2. calculate the maximum size & weight per sex and species
      using the aggregate function (hint: use bracket
      selection!)
max.size.and.weight <- aggregate(fish[,3:4], by = list(
      species=species, sex=sex), FUN="max", na.rm=T)

# 3. save the results of ex1 and ex2 as dataframe
write.table(min.weight, "min.weight.csv", sep=',')
write.table(max.size.and.weight, "max.size.and.weight.csv",
      sep=',')
```

aggregating data

```
max.size.and.weight <- aggregate(fish[,3:4], by = list(
  species=species, sex=sex), FUN="max", na.rm=T)

max.size.and.gonad.weight <- aggregate(fish[,c(3,7)], by =
  list(species=species, sex=sex), FUN="max", na.rm=T)

max.size.and.gonad.weight <- aggregate(cbind(total.length,
  gonad.weight), by = list(species=species, sex=sex), FUN=
  "max", na.rm=T)
```

Histograms

```
hist(total.length[species == "dab"], xlab = "total length [mm]", main="length distribution of dab") # absolute frequencies (counts)
```


Histograms

```
hist(total.length[species == "dab"], xlab = "total length [mm]", main="length distribution of dab") # absolute frequencies (counts)

hist(total.length[species == "dab"], freq = F, xlab = "total length [mm]", main="length distribution of dab") # relative frequencies (counts)
```

Histograms - Changing "breaks"

```
hist(total.length[species == "dab"], breaks=10, xlab = "
  total length [mm]", main="length distribution of dab") #
  plot histogram with 10 breaks

hist(total.length[species == "dab"], breaks=seq(80, 560, 5),
  xlab = "total length [mm]", main="length distribution
  of dab") # plot histogram with breaks from 80 till 560
  mm (breaks width = 5mm)

box(bty="l") # plot L-like box to align axes
```

2 Histograms in a single window

```
par(mfrow=c(3,1)) # plot rowwise by deviding the display in
  a "c(rows, columns)" matrix
hist(total.length[species == "dab"], xlab = "total length [
  mm]", main="length distribution of dab") # first
  histogram

hist(total.length[species == "cod"], xlab = "total length [
  mm]", main="length distribution of cod") # second
  histogram
```

Loading CTD data

```
# the plot function
# load CTD data
setwd("~/Dropbox/R_course/day2") # set working directory
CTD <- read.table("CTD.csv", header=T, sep=',', dec=".") # load
  dataframe
head(CTD)
colnames(CTD) <- c(colnames(CTD)[1:5], "Pressure", "Conductivity", "
  Temp", "Sal", "OxySat")
head(CTD)
attach(CTD)
```

plotting a temperature profile

```
Depth <- Pressure -1
```

```
par(mfrow=c(1,1)) # single plot window
```

```
plot(Temp, Depth, ylab="depth [m]", xlab="temperature [degrees C]")
```

plotting a temperature profile as a line

```
Depth <- Pressure -1
```

```
par(mfrow=c(1,1)) # single plot window
```

```
plot(Temp, Depth, ylab="depth [m]", xlab="temperature [degrees C]")
```

```
plot(Temp, Depth, type="l", ylab="depth [m]", xlab="temperature [
  degrees C]") # plot line instead of dots
```

plotting a temperature profile - editing the y-axis

```
Depth <- Pressure -1
```

```
par(mfrow=c(1,1)) # single plot window
```

```
plot(Temp, Depth, ylab="depth [m]", xlab="temperature [degrees C]")
```

```
plot(Temp, Depth, type="l", ylab="depth [m]", xlab="temperature [degrees C]") # plot line instead of dots
```

```
plot(Temp, Depth, xlim = c(0, 25), ylim=c(8, 0), type="l", ylab="depth [m]", xlab="temperature [degrees C]") # plot inverse y-axis
```

plotting a temperature profile

```
Depth <- Pressure -1
```

```
par(mfrow=c(1,1)) # single plot window
```

```
plot(Temp, Depth, ylab="depth [m]", xlab="temperature [degrees C]")
```

```
plot(Temp, Depth, type="l", ylab="depth [m]", xlab="temperature [degrees C]") # plot line instead of dots
```

```
plot(Temp, Depth, xlim = c(0, 25), ylim=c(8, 0), type="l", ylab="depth [m]", xlab="temperature [degrees C]") # plot inverse y-axis
```

```
xlabel <- "temperature [degrees C]"
```

```
ylabel <- "depth [m]"
```

```
plot(Temp, Depth, ylim=c(max(Depth), 0), type="l", xlab=xlabel, ylab=ylabel)
```


plotting a temperature profile - editing the axes positions

```
xlabel <- "temperature [degrees C]"
ylabel <- "depth [m]"
plot(Temp, Depth, ylim=c(max(Depth), 0), type="l", xlab=xlabel, ylab
      =ylabel)
```

```
plot(Temp, Depth, ylim=c(max(Depth), 0), type="l", xlab=xlabel, ylab
      =ylabel, axes=F) # plot no axes
axis(2) # plot y-axis to the left
# axis(4) # plot y-axis to the right
# axis(1) # plot x-axis at the bottom
axis(3) # plot x-axis above the plot
```

plotting text, points and lines

```
xlabel <- "temperature [degrees C]"
ylabel <- "depth [m]"

plot(Temp, Depth, ylim=c(max(Depth), 0), type="l", xlab=xlabel, ylab=
      =ylabel, axes=F) # plot no axes
axis(2) # plot y-axis to the left
# axis(4) # plot y-axis to the right
# axis(1) # plot x-axis at the bottom
axis(3) # plot x-axis above the plot

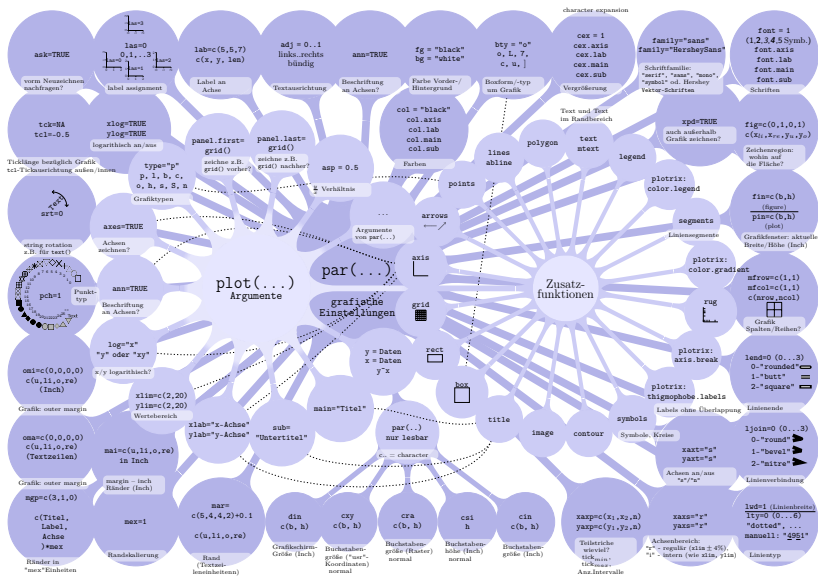
text(10, 5, "text") # add text
points(15, 5, cex=3.5, col="red", pch=16) # add a point (cex:
      point size; pch: point type)
lines(0:20, rep(15, 21), lwd=2, lty=1) # add line
lines(0:20, rep(Depth[Temp == min(Temp)], 21), lwd=2, lty=2, col="blue
      ") # add line
```

plotting a legend

```
plot(Temp, Depth, ylim=c(max(Depth), 0), type="l", xlab=xlabel, ylab  
      =ylabel, axes=F) # plot inverse y-axis  
axis(2) # plot y-axis to the left  
axis(3) # plot x-axis above the plot
```

```
legend("bottom", legend = c("refline 1", "refline 2"), col=c("black"  
      , "blue"), lwd = 2)
```

plotting commands



Exercises

CTD.xls

1. plot the salinity profile
2. plot the temperature and salinity profiles above each other
3. plot the profiles of temperature, salinity, conductivity and oxygen in a 2x2 plotting window

Parasite.xls

1. Import Parasite.csv
2. plot the age distribution of both sexes as 2 histograms but in one window
3. redraw the age distribution as colored lines using the density and plot function
4. add a legend
5. create boxplots of the weight of infected and non infected organisms and both sexes