

Using for statistical analyses

Robert Bauer

Warnemünde, 05/02/2012



Day 4 - Agenda:

- ▶ Last Exercises
- ▶ Loops
- ▶ Conditional Statements
- ▶ Exercises

Exercises

1. generate random Poisson-distributed numbers
`lambda <- c(0.1, 1, 2, 3, 10)`
2. plot associated probability density functions (PDF)
(one plot with multiple lines of different type)
`dpois(range, lambda)`
3. add a legend
4. perform Kolmogorov-Smirnov tests for Normal Distribution

Exercises

```
# 1. generate random Poisson-distributed numbers
n <- 100      # sample size
lambda <- c(0.1, 1, 2, 3, 10)

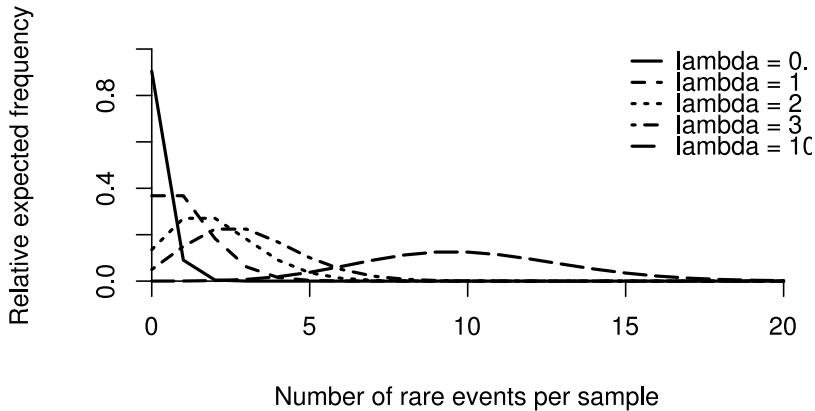
x <- rpois(n, lambda[1]) # rpois(n, lambda)
```

Exercises

```
# 2. plot associated probability density functions (PDF)
plot(0:20, dpois(0:20, lambda[1]), type="l", lwd=2, axes=F,
     xlim=c(0,20), ylim=c(0,1),
     xlab="Number of rare events per sample", ylab="Relative
         expected frequency")
axis(1,pos=0)
axis(2,pos=0)
lines(0:20, dpois(0:20, lambda[2]), lwd=2, lty=2)
lines(0:20, dpois(0:20, lambda[3]), lwd=2, lty=3)
lines(0:20, dpois(0:20, lambda[4]), lwd=2, lty=4)
lines(0:20, dpois(0:20, lambda[5]), lwd=2, lty=5)

# 3. add a legend
legend("topright", bty="n", lwd=2, lty=c(1,2,3,4,5),
      legend = c("lambda = 0.1", "lambda = 1", "lambda = 2", "
        lambda = 3", "lambda = 10"))
```

Exercises



Exercises

```
# 4. perform Kolmogorov-Smirnov tests for Normal Distribution
n <- 100      # sample size
lambda <- c(0.1, 1, 2, 3, 10)

x <- rpois(n, lambda[5]) # rpois(n, lambda)

# H0: normal distributed data
ks.test(x, "pnorm", mean = mean(x), sd = sqrt(var(x)))
# if p-value < 0.05: reject H0, else: keep H0

# result: Poisson random numbers with lambda = 10 appear to be
#         normal distributed!
```

One-sample Kolmogorov-Smirnov test

data: x D = 0.1074, p-value = 0.1987
alternative hypothesis: two-sided

for loops

```
for(i in 1:5) i           # no output shown  
for(i in 1:5) print(i)    # return index  
for(i in 5:1) print(i)    # inverse index  
index <- c(9,2,21,3,4)    # specified index  
for(i in index) print(i)
```


for loops

```
for(i in 1:5) i           # no output shown  
for(i in 1:5) print(i)    # return index  
for(i in 5:1) print(i)    # inverse index  
index <- c(9,2,21,3,4)    # specified index  
for(i in index) print(i)
```

```
for(i in index) # multiple operations using {}  
{  
  x <- rnorm(i)  
  print(x)  
  print(sum(x^2))  
}
```

while loops

```
# 2.) while loop
i <- 1
while(i <= 13){
  print(i)
  i <- i+1
}
```

Conditional Statements

```
# Conditional Statements
x <- 5
if(x <= 99) print("x <= 99")

x <- 100
if(x <= 99){
  print("x <= 99")
}else{
  print("x > 99")
}
```

repeat loops

```
# repeat loop
x <- 0
repeat{
  x <- x+1
  if(x == 7) next
  print(x)
  # if condition, necessary to end (break) loop
```

Conditional Statements

```
# Conditional Statements  
x <- 5  
if(x <= 99) print("x <= 99")
```

```
x <- 100  
if(x <= 99){  
  print("x <= 99")  
}else{  
  print("x > 99")  
}
```

```
ifelse(x<=99, print("x <= 99"), print("x > 99"))
```

Conditional Statements

```
x <- 3  
switch(x, 2+2, mean(1:10), rnorm(5))
```

```
y <- "fruit"  
switch(y, fruit="banana", vegetable="broccoli", "Neither")  
  
k <- switch(y, fruit="banana", vegetable="broccoli", "Neither")
```

Loops

Function	Description
<code>repeat{statement}</code>	repeating statement
<code>while(condition){statement}</code>	repeating statement, while condition is true
<code>for{i in seq}{statement}</code>	repeat statement for each element <i>i</i> of seq
<code>next</code>	jumps to next iteration step
<code>break</code>	immediately exits the loop
<code>print</code>	show selected values from enclosed operations (e.g. loops, functions)

Conditional Statements

Function	Description
<code>if(condition){statement1}</code>	if condition is TRUE, perform statement1
<code>else{statement2}</code>	else perform statement2
<code>ifelse(condition,statement1,statement2)</code>	alternative listing of statements
<code>switch(statement,list)</code>	return corresponding element from the list of alternatives

Combining *for* loops

```
# combining loops!  
setwd("~/Dropbox/R_course/day4") # set working directory  
file = "DataC13.csv"  
data <- read.csv(file, header=T, dec=".", sep=',') # load  
  dataframe  
head(data)
```

	species	regime	depth	atom13C
1	Lobophora variegata	N1	5	NA
2	Dictyota pinnatifida	N1	5	1.469
3	Halimeda opuntia	N1	5	1.231
4	Turfalgae	N1	5	1.345
5	Fluffy balls	N1	5	1.231
6	Red hairy	N1	5	1.498

Combining *for* loops

```
levels.depth <- as.numeric(levels(factor(depth)))
levels.species <- levels(data$species)
max.values <- aggregate(data$atom13C, list(species=data$species), FUN=
  max, na.rm=T)

for(s in levels.species)
{
  par(mfrow = c(2,2))
  for(d in 1:length(levels.depth))
  {
    # select data to plot
    plot.data <- subset(data, species == s & depth == levels.depth[d])

    # define title and y-axis limits
    Title <- paste('depth: ', levels.depth[d], ' m')
    ylimits <- c(1.06, max.values$x[max.values$species == s])

    # create boxplots
    boxplot(atom13C ~ regime, data = plot.data, main=Title,
            ylim = ylimits, xlab = 'nutrient regime', ylab = 'atom
              percentage 13C')

    # add species information
    mtext(s, NORTH<-3, line=0.2)
  }
}
```

Exercises

1. edit shown loops for multiple boxplots
 - 1.1 don't show x-axis label on upper plots
 - 1.2 don't show y-axis label on right plots

no x-axis label on upper plots and no y-axis label on right plots

```
# 1. don't show x-axis label on upper plots and
# 2. don't show y-axis label on right plots
# option 1. if-else
if(d > 2)
{
  if(d == 3){
    xlabel <- 'nutrient regime'
    ylabel <- 'atom percentage 13C'
  }else{
    xlabel <- 'nutrient regime'
    ylabel <- ''
  }
}
else{
  if(d == 1){
    xlabel <- ''
    ylabel <- 'atom percentage 13C'
  }else{
    xlabel <- ''
    ylabel <- ''
  }
}

boxplot(atom13C ~ regime, data = plot.data,
        ylim = ylimits, xlab = xlabel, ylab = ylabel)
```

no x-axis label on upper plots and no y-axis label on right plots

```
# option 2. vector based
xlabel <- c("", "", 'nutrient regime', 'nutrient regime')
ylabel <- c('atom percentage 13C', "", 'atom percentage 13C', "")

boxplot(atom13C ~ regime, data = plot.data, ylim = ylims,
        xlab = xlabel[d], ylab = ylabel[d])

# option 3. switch
xlabel <- switch(d, "", "", 'nutrient regime', 'nutrient regime')
ylabel <- switch(d, 'atom percentage 13C', "", 'atom percentage 13C',
               "")

boxplot(atom13C ~ regime, data = plot.data, ylim = ylims,
        xlab = xlabel, ylab = ylabel)
```