# Procvičování 10

1. Naimportujte do **R** dataframy *spe* a *env* z excelového souboru *dat09.xls* ve studijních materiálech (cv09).

```
spe<- read.delim('D:/My Dropbox/predmety/uvod do R/2014/cv09/spe.txt', row.names=1)
env<- read.delim('D:/My Dropbox/predmety/uvod do R/2014/cv09/env.txt', row.names=1)
```

2. Vytvořte sekvenci celých čísel od -5 do 5 (*vec*). Pomocí smyčky vynásobte každou hodnotu vektoru *vec* -1 a výsledné hodnoty uspořádejte do vektoru *res2*.

```
vec<- -5:5
res2<- numeric(length(vec))

for(i in 1:length(vec)){
  res2[i]<- vec[i] * -1
}
res2

##  [1]  5  4  3  2  1  0 -1 -2 -3 -4 -5
```

3. Pomocí smyčky upravte hodnoty vektoru *vec* následovně: $x^2 - x$.

```
res3<- numeric(length(vec))

for(i in 1:length(vec)){
  res3[i]<- vec[i]^2 - vec[i]
}
res3

##  [1] 30 20 12  6  2  0  0  2  6 12 20
```

4. Pomocí smyčky spočítejte kumulativní sumy vektoru *vec*. V základní instalaci **R** je funkce, která kumulativní sumy počítá. Najděte ji, na vektor *vec* použijte a výsledek porovnejte.

```
# nekolik (hodne) moznosti, napr.:
# a)
res4a<- numeric(length(vec))
res4a[1]<- vec[1]

for(i in 2:length(vec)){
  res4a[i]<- res4a[i-1] + vec[i]
}
res4a

##  [1]  -5  -9 -12 -14 -15 -15 -14 -12  -9  -5   0
```

```
# b)
res4b<- numeric(length(vec))

for(i in 1:length(vec)){
  res4b[i]<- c(0, res4b)[i] + vec[i]
}
res4b

##  [1]  -5  -9 -12 -14 -15 -15 -14 -12  -9  -5   0

# c) to vymyslel Honza Krausko
res4c<- numeric(length(vec))

for(i in 1:length(vec)){
  res4c[i]<- res4c[i - 1*(i>1)] + vec[i]
}
res4c

##  [1]  -5  -9 -12 -14 -15 -15 -14 -12  -9  -5   0

# cumsum()
cumsum(vec)

##  [1]  -5  -9 -12 -14 -15 -15 -14 -12  -9  -5   0
```

5. Pomocí smyčky spočítejte pro každou lokalitu dataframu *spe*, kolik druhů na ní bylo zaznamenáno.

```
n.spe<- numeric(nrow(spe))
names(n.spe)<- rownames(spe)

for(i in rownames(spe)){
  n.spe[i]<- sum(spe[i, ] > 0)
}
n.spe

## s01 s02 s03 s04 s05 s06 s07 s08 s09 s10 s11 s12 s13 s14 s15 s16 s17 s18
##  22  27  19  19  26  25  26  17  23  19  21  24  29  22  30  17  16  31
## s19 s20 s21 s22 s23 s24 s25 s26 s27
##  27  20  28  20  15  31  19  31  28
```

6. Pomocí smyčky spočítejte pro každý druh dataframu *spe* jeho celkovou abundanci.

```
abund<- numeric(ncol(spe))
names(abund)<- names(spe)

for(i in names(spe)){
  abund[i]<- sum(spe[,i])
}
abund
```

```
## ablabesp apsetrif brilmode brilflav cladotsp  corysp. cricannu cricbici
##        6       25       12       22      205      381        7        8
## cricbigr crictrgr critriia cromussp  demisp. diplcult eukibrev eukicoer
##       68       32        5       29        1        1       30       22
## eukideil eukigrgr eukilobi eukimino eukisimi hetemarc mictrasp micrchgr
##       94        2       69        3        1        3      419      809
## nanoreag  natasp. nilodubi orthrigr orthrubi orthfrig orthobum orththie
##      184        4       24      115      614        4      798       13
## paracrsp parastyl paratasp paraalgr pararufi phaepssp polyconv polylagr
##        2        9        4        3       16        6        8      200
## polyscgr pottgaed pottlong prodoliv rheofusc rheotasp stembrgr synosemi
##       39        1       57      147      172      897        3     1305
## tanybrun tanytasp thellasp thiegrge tvetbaca tvetdive
##      174      242      203      486      623      471
```

7. Pomocí smyčky zjistěte pro každý druh dataframu *spe*: počet nulových abundancí (počet lokalit, na kterých chyběl), minimální, mediánovou a maximální abundanci nenulových hodnot. Zjištěné hodnoty uspořádejte do matice s počtem řádků rovným počtu druhů a čtyřmi sloupci. Řádky i sloupce matice si rozumně pojmenujte.

```
# samozrejme opet nekolik moznosti
spe.stat<- matrix(NA, nrow= ncol(spe), ncol= 4)
dimnames(spe.stat)<- list(names(spe), c("zeros", "min", "median", "max"))

for(i in names(spe)){
  abund_i<- spe[, i]
  # abund_i obsahuje vytazene abundance i-teho druhu
  nuly_i<- abund_i == 0
  # nuly_i oznacuji nulove abundance i-teho druhu
  spe.stat[i, "zeros"]<- sum(nuly_i)
  spe.stat[i, "min"]<- min(abund_i[!nuly_i])
  spe.stat[i, "median"]<- median(abund_i[!nuly_i])
  spe.stat[i, "max"]<- max(abund_i[!nuly_i])
}
spe.stat

##          zeros min median max
## ablabesp    23   1    1.0   3
## apsetrif    19   1    2.0   9
## brilmode    24   1    2.0   9
## brilflav    16   1    1.0   4
## cladotsp    10   1    3.0  73
## corysp.      1   1   14.0  43
## cricannu    22   1    1.0   2
## cricbici    20   1    1.0   2
## cricbigr    11   1    2.5  13
## crictrgr    15   1    2.0   8
## critriia    24   1    1.0   3
## cromussp    18   1    2.0   7
## demisp.     26   1    1.0   1
```

```
## diplcult    26   1    1.0   1
## eukibrev    16   1    2.0  11
## eukicoer    14   1    1.0   5
## eukideil    14   1    2.0  25
## eukigrgr    26   2    2.0   2
## eukilobi     8   1    2.0  14
## eukimino    25   1    1.5   2
## eukisimi    26   1    1.0   1
## hetemarc    25   1    1.5   2
## mictrasp     1   1    6.5 217
## micrchgr    13   1   25.0 287
## nanoreag     6   1    3.0  58
## natasp.     23   1    1.0   1
## nilodubi    15   1    1.5   8
## orthrigr    12   1    2.0  32
## orthrubi     3   1   21.0  96
## orthfrig    24   1    1.0   2
## orthobum     0   1   22.0 133
## orththie    18   1    1.0   3
## paracrsp    25   1    1.0   1
## parastyl    21   1    1.5   2
## paratasp    25   1    2.0   3
## paraalgr    25   1    1.5   2
## pararufi    18   1    1.0   6
## phaepssp    25   2    3.0   4
## polyconv    22   1    1.0   4
## polylagr     9   1    5.0  48
## polyscgr    17   1    3.0  11
## pottgaed    26   1    1.0   1
## pottlong    13   1    2.5  16
## prodoliv    23   5    5.5 131
## rheofusc     4   1    4.0  28
## rheotasp     4   1    4.0 283
## stembrgr    24   1    1.0   1
## synosemi     0   1   32.0 183
## tanybrun     3   1    4.5  32
## tanytasp     3   1    4.5  47
## thellasp     3   1    5.0  34
## thiegrge     2   1    6.0 127
## tvetbaca     5   1    4.5 177
## tvetdive     5   1    5.0 116
```

8. Pro všechny druhy zaznamenané alespoň na třetině lokalit (9) zjistěte Adjusted R Squared linearního modelu závislosti $log(x+1)$ abundance na Froudeho čísle. Použijte polynom druhého řádu:

```
m1<- lm(log.abundance ~ poly(froude,2))
```

Adjusted R Squared najdete v `summary(m1)`, musíte jen přijít na to, jak si pro něj sáhnout. ($R^2$ představuje podíl variability modelované proměnné vysvětlené daným modelem)

```
# nejprve vytahneme ty druhy vyskytujici se aspon na tretine lokalit
spe9<- spe[, colSums(spe>0) >= nrow(spe)/3]
dim(spe9)

## [1] 27 31

# zjistime, jak sahnout na  Adjusted R2: nejprve si ulozime jeden model, prohlidneme
# strukturu jeho summary a tam najdeme hledane Adjuster R2:
m1<- lm(log1p(spe9[,1]) ~ poly(froude,2), data= env)
str(summary(m1))

## List of 11
##  $ call         : language lm(formula = log1p(spe9[, 1]) ~ poly(froude, 2), data = env)
##  $ terms        :Classes 'terms', 'formula' length 3 log1p(spe9[, 1]) ~ poly(froude, 2)
##   .. ..- attr(*, "variables")= language list(log1p(spe9[, 1]), poly(froude, 2))
##   .. ..- attr(*, "factors")= int [1:2, 1] 0 1
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:2] "log1p(spe9[, 1])" "poly(froude, 2)"
##   .. .. .. ..$ : chr "poly(froude, 2)"
##   .. ..- attr(*, "term.labels")= chr "poly(froude, 2)"
##   .. ..- attr(*, "order")= int 1
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(log1p(spe9[, 1]), poly(froude, 2, coefs = structure
##   .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "nmatrix.2"
##   .. .. ..- attr(*, "names")= chr [1:2] "log1p(spe9[, 1])" "poly(froude, 2)"
##  $ residuals    : Named num [1:27] -0.384 0.322 -0.383 -0.435 0.262 ...
##   ..- attr(*, "names")= chr [1:27] "s01" "s02" "s03" "s04" ...
##  $ coefficients : num [1:3, 1:4] 0.4099 0.0572 0.212 0.1143 0.5941 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "(Intercept)" "poly(froude, 2)1" "poly(froude, 2)2"
##   .. ..$ : chr [1:4] "Estimate" "Std. Error" "t value" "Pr(>|t|)"
##  $ aliased      : Named logi [1:3] FALSE FALSE FALSE
##   ..- attr(*, "names")= chr [1:3] "(Intercept)" "poly(froude, 2)1" "poly(froude, 2)2"
##  $ sigma        : num 0.594
##  $ df           : int [1:3] 3 24 3
##  $ r.squared    : num 0.00566
##  $ adj.r.squared: num -0.0772
##  $ fstatistic   : Named num [1:3] 0.0683 2 24
##   ..- attr(*, "names")= chr [1:3] "value" "numdf" "dendf"
##  $ cov.unscaled : num [1:3, 1:3] 3.70e-02 -2.67e-18 -1.34e-17 -2.67e-18 1.00 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "(Intercept)" "poly(froude, 2)1" "poly(froude, 2)2"
##   .. ..$ : chr [1:3] "(Intercept)" "poly(froude, 2)1" "poly(froude, 2)2"
##  - attr(*, "class")= chr "summary.lm"

# vidime, ze vysledkem summary(m1) je list s 11 elementy, mezi nimiz ten hledany
# je adj.r.squared.
# dosahneme na nej pomoci $
summary(m1)$adj.r.squared

## [1] -0.0772
```

```
# ted uz zbyva jen vytvorit cilovy objekt, ktery pomoci smycky naplnime R2
# modelu kazdeho druhu.
lm.R2<- numeric(ncol(spe9))
names(lm.R2)<- names(spe9)

for(i in names(spe9)){
  temp.model<- lm(log1p(spe9[, i]) ~ poly(froude,2), data= env)
  lm.R2[i]<- summary(temp.model)$adj.r.squared
}
lm.R2
```

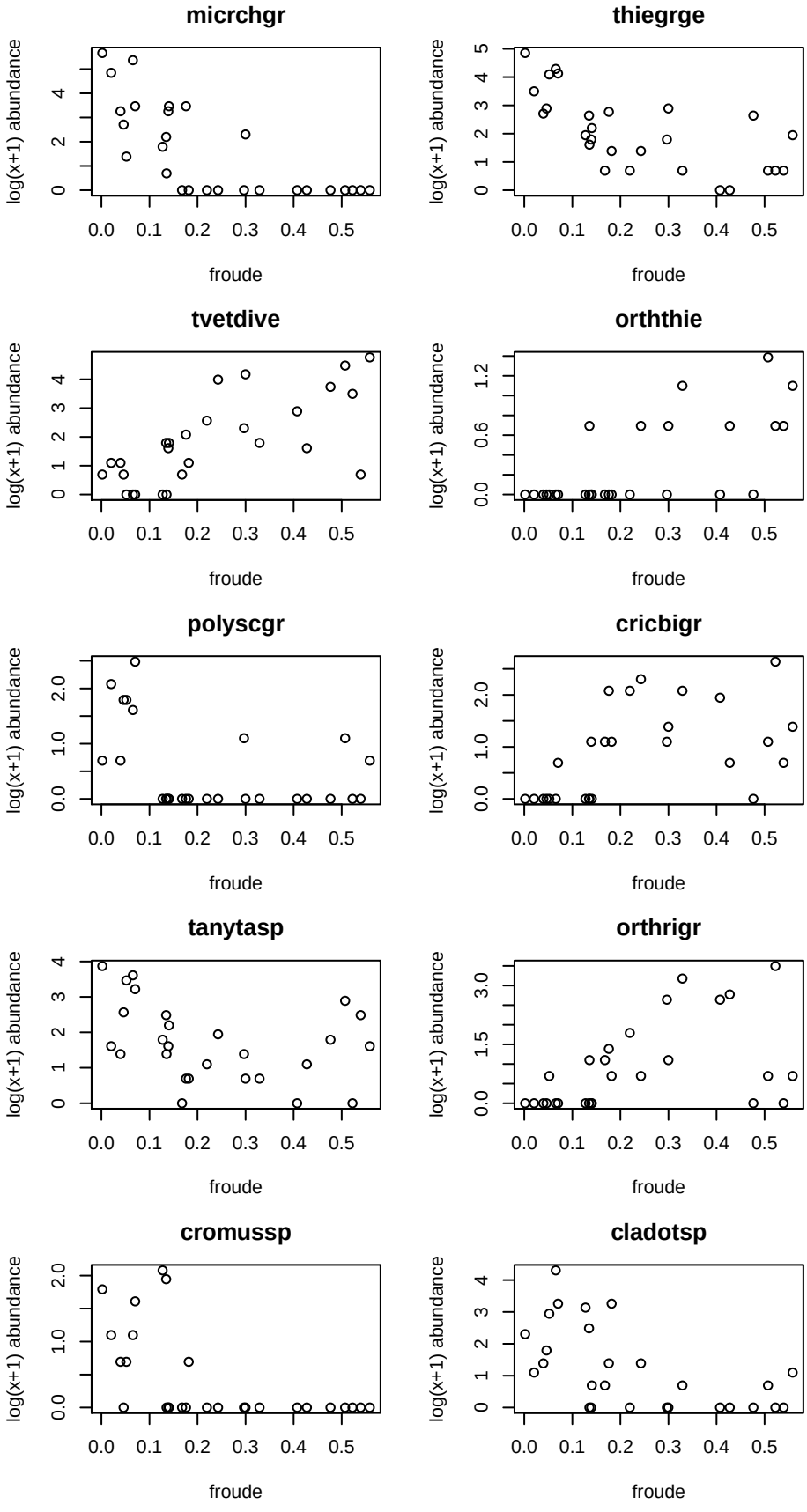
```
##   brilflav   cladotsp    corysp.   cricbigr   crictrgr   cromussp
## -0.0772013  0.3437445  0.1285199  0.3884363  0.1422570  0.3487249
##   eukibrev   eukicoer   eukideil   eukilobi   mictrasp   micrchgr
## -0.0502644  0.0290665  0.2585866  0.3004862  0.0682080  0.6292255
##   nanoreag   nilodubi   orthrigr   orthrubi   orthobum   orththie
## -0.0468079 -0.0650330  0.3514860  0.3140007  0.1441471  0.4162083
##   pararufi   polylagr   polyscgr   pottlong   rheofusc   rheotasp
## -0.0517413  0.1277036  0.3907850  0.2395388 -0.0006437  0.2257360
##   synosemi   tanybrun   tanytasp   thellasp   thiegrge   tvetbaca
##  0.1433001  0.2509443  0.3562075  0.0996563  0.5887470  0.3202542
##   tvetdive
##  0.4685363
```

9. Pro 10 druhů s nejvyšším Adjusted R Squared zobrazte bodový graf zavislosti jeho $log(x+1)$ abundance na Froudeho čísle.

```
# nejprve bychom meli vybrat tech 10 druhu s nejvyssim R2:
spe.best10<- spe9[, order(-lm.R2)[1:10]]

# a ted pro kazdy druh z noveho dataframu namalujeme graf.
# predtim si jeste nastavim graficke okno.
par(mfrow=c(5,2), mar= c(4,4,3,2))

for(i in names(spe.best10)){
  plot(log1p(spe.best10[, i]) ~ froude, data= env, main= i,
  xlab= 'froude', ylab= 'log(x+1) abundance')
}
```


micrchgr
thiegrge
tvetdive
orthhie
polyscgr
cricbigr
tanytasp
orthrigr
cromussp
cladotsp
log(x+1) abundance
froude