

Procvičování 4 s řešením

1. V RStudiosu vytvořte nový projekt, umístěte ho do adresáře `cv04` a natáhněte si pracovní prostředí z minula. Najdete zde: <http://www.sci.muni.cz/~syrovat/cv04.RData>. Natažení můžete provést přímo z internetu pomocí příkazu `url()`.

```
load(url("http://www.sci.muni.cz/~syrovat/cv04.RData"))
```

2. Zjistěte, jaké objekty pracovní prostředí obsahuje.

```
ls()
## [1] "env" "jmena" "noha" "sex" "spe" "vyska"
```

3. Vytvořte vektor `vec` obsahující sekvenci celých čísel od 1 do 20.

```
vec <- 1:20
```

4. Přiřaďte vektoru `vec` rozměry: 5 řádků a 4 sloupce.

```
dim(vec) <- c(5, 4)
```

5. Zjistěte třídu objektu `vec` a zkontrolujte jeho rozměry.

```
class(vec)
## [1] "matrix"
dim(vec)
## [1] 5 4
```

6. Převeďte objekt `vec` zpět na vektor a nechte si vypsát jeho třídu.

```
vec <- as.numeric(vec)
# nebo vec <- as.vector(vec)
class(vec)
## [1] "numeric"
```

7. Vytvořte z vektoru `vec` matici `mat1` o 4 sloupcích tak, aby se hodnoty do matice skládaly po sloupcích.

```
mat1 <- matrix(vec, ncol = 4)
```

8. Vytvořte z vektoru *vec* matici *mat2* o 4 sloupcích tak, aby se hodnoty do matice skládaly po řádcích.

```
mat2 <- matrix(vec, ncol = 4, byrow = T)
```

9. Vizuálně porovnejte 10. hodnotu matic *mat1* a *mat2* (nechejte si je vypsat).

```
mat1[10]
## [1] 10
mat2[10]
## [1] 18
```

10. Stejně porovnejte první sloupec těchto matic.

```
mat1[, 1]
## [1] 1 2 3 4 5
mat2[, 1]
## [1] 1 5 9 13 17
```

11. Do matice *mat* spojte 1.-3. sloupec matice *mat1* a 3.-4. sloupec *mat2*. (vznikne čtvercová matice o 5 řádcích i sloupcích)

```
mat <- cbind(mat1[, 1:3], mat2[, 3:4])
mat
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  1   6  11   3   4
## [2,]  2   7  12   7   8
## [3,]  3   8  13  11  12
## [4,]  4   9  14  15  16
## [5,]  5  10  15  19  20
```

12. Vytvořte list *mat.info* obsahující matici *mat*, její rozměry, počet sloupců, počet řádků a aritmetický průměr hodnot matice *mat*. Jednotlivé elementy listu vhodně nazvěte.

```

mat.info <- list(mat = mat, dim.mat = dim(mat), ncol.mat = ncol(mat), nrow.mat = nrow(mat),
  mean.mat = mean(mat))
mat.info

## $mat
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   1   6  11   3   4
## [2,]   2   7  12   7   8
## [3,]   3   8  13  11  12
## [4,]   4   9  14  15  16
## [5,]   5  10  15  19  20
##
## $dim.mat
## [1] 5 5
##
## $ncol.mat
## [1] 5
##
## $nrow.mat
## [1] 5
##
## $mean.mat
## [1] 9.4

```

13. Horní i dolní trojúhelník matice *mat* (tedy všechny mimodiagonální elementy) vyplňte nulami.

```

mat[upper.tri(mat)] <- 0
mat[lower.tri(mat)] <- 0
mat

##      [,1] [,2] [,3] [,4] [,5]
## [1,]   1   0   0   0   0
## [2,]   0   7   0   0   0
## [3,]   0   0  13   0   0
## [4,]   0   0   0  15   0
## [5,]   0   0   0   0  20

```

14. Pojmenujte sloupce matice *mat* velkými písmeny.

```

colnames(mat) <- LETTERS[1:ncol(mat)]
mat

##      A B C D E
## [1,] 1 0 0 0 0
## [2,] 0 7 0 0 0
## [3,] 0 0 13 0 0
## [4,] 0 0 0 15 0
## [5,] 0 0 0 0 20

```

15. K matici *mat* připojte sloupec součtů hodnot matice *mat* po řádcích. Nazvěte jej 'sumy'. Funkci počítající sumy v řádcích najděte.

```
mat <- cbind(mat, rowSums(mat))
colnames(mat)[ncol(mat)] <- "sumy"
mat

##      A B C D E sumy
## [1,] 1 0 0 0 0  1
## [2,] 0 7 0 0 0  7
## [3,] 0 0 13 0 0 13
## [4,] 0 0 0 15 0 15
## [5,] 0 0 0 0 20 20

# nebo nejprve vytvorime sumy: sumy<- rowSums(mat) a az pak je pripojime:
# mat<- cbind(mat, sumy)
```

16. Vytvořte matici NA hodnot o počtu řádků rovném počtu sloupců objektu *mat* a dvou sloupcích. Řádky pojmenujte jmény sloupců matice *mat* a sloupce *r* a *p*.

```
# Napr.:
matrix(NA, nrow = ncol(mat), ncol = 2, dimnames = list(colnames(mat), c("r",
  "p")))

##      r p
## A   NA NA
## B   NA NA
## C   NA NA
## D   NA NA
## E   NA NA
## sumy NA NA
```

17. Vytvořte dataframe *miry*, obsahující proměnné *vyska*, *noha* a *sex* s hodnotami vektorů *vyska*, *noha* a *sex*.

```
miry <- data.frame(vyska = vyska, noha = noha, sex = sex)
miry

##   vyska noha sex
## 1   164 24.3  z
## 2   162 25.0  z
## 3   183 27.7  m
## 4   174 25.5  z
## 5   197 29.6  m
## 6   180 29.7  m
## 7   164 23.7  z
## 8   187 29.0  m
## 9   173 27.5  m
## 10  NA   NA  z
```

18. Řádky dataframu *miry* pojmenujte jmény měřených jedinců *jmena*.

```
rownames(miry) <- jmena
miry

##           vyska noha sex
## Michaela  164 24.3  z
## Petra     162 25.0  z
## Jaroslav  183 27.7  m
## Tereza    174 25.5  z
## Ondra     197 29.6  m
## Jan       180 29.7  m
## Lubica    164 23.7  z
## Jiri      187 29.0  m
## Jakub     173 27.5  m
## Zuzka     NA   NA   z
```

19. V dataframu *miry* vytvořte novou proměnnou *pomer* obsahující poměr délky nohy ku výšce.

```
miry$pomer <- miry$noha/miry$vyska
```

20. V dataframu *miry* vytvořte novou logickou proměnnou *nadprum* odlišující jedince s naprůměrným poměrem.

```
miry$nadprum <- miry$pomer > mean(miry$pomer, na.rm = T)
miry

##           vyska noha sex  pomer nadprum
## Michaela  164 24.3  z 0.1482  FALSE
## Petra     162 25.0  z 0.1543   TRUE
## Jaroslav  183 27.7  m 0.1514  FALSE
## Tereza    174 25.5  z 0.1466  FALSE
## Ondra     197 29.6  m 0.1503  FALSE
## Jan       180 29.7  m 0.1650   TRUE
## Lubica    164 23.7  z 0.1445  FALSE
## Jiri      187 29.0  m 0.1551   TRUE
## Jakub     173 27.5  m 0.1590   TRUE
## Zuzka     NA   NA   z    NA    NA
```

21. Dokažte se, zda objekt *spe* je matice. (odpověď má být logická hodnota)

```
is.matrix(spe)

## [1] TRUE
```

22. Převeďte matici *spe* na dataframe.

```
spe <- as.data.frame(spe)
class(spe)
```

```
## [1] "data.frame"
```

23. Je možné převést dataframe *miry* na matici? Co se stane, když to uděláme?

```
# je, ale vsechny hodnoty budou prevedeny na text
as.matrix(miry)
```

```
##      vyska noha  sex pomer  nadprum
## Michaela "164" "24.3" "z" "0.1482" "FALSE"
## Petra    "162" "25.0" "z" "0.1543" " TRUE"
## Jaroslav "183" "27.7" "m" "0.1514" "FALSE"
## Tereza   "174" "25.5" "z" "0.1466" "FALSE"
## Ondra    "197" "29.6" "m" "0.1503" "FALSE"
## Jan      "180" "29.7" "m" "0.1650" " TRUE"
## Lubica   "164" "23.7" "z" "0.1445" "FALSE"
## Jiri     "187" "29.0" "m" "0.1551" " TRUE"
## Jakub    "173" "27.5" "m" "0.1590" " TRUE"
## Zuzka    NA     NA     "z" NA     NA
```

24. Nahradte NA hodnoty dataframu *spe* nulami (0). (`is.na()`). Dataframe *spe* obsahuje početnosti - abundance - larev 56 druhů pakomárů (ve sloupcích) na 27 lokalitách (řádcích).

```
spe[is.na(spe)] <- 0
```

25. Vytvořte vektor *loc.sum* obsahující celkové abundance pakomárů na lokalitách. (Součty hodnot v řádcích dataframu *spe*).

```
loc.sum <- rowSums(spe)
loc.sum
```

```
##  s1  s2  s3  s4  s5  s6  s7  s8  s9  s10  s11  s12  s13  s14  s15  s16  s17  s18
## 274 432 164 220 535 630 258 146 301 212 181 540 632 249 747 122 243 791
## s19 s20 s21 s22 s23 s24 s25 s26 s27
## 207 221 657 194 145 705 65 957 444
```

26. Vytvořte vektor *spe.freq* obsahující počty lokalit, na kterých se jednotlivé druhy pakomárů vyskytovaly. (Počty hodnot vyšších než 0 ve sloupcích)

```
spe.freq <- colSums(spe > 0)
spe.freq
```

```
## ablabesp Apsetrif Brilbif Brilflav cladotsp corysp. Cricannu Cricbici
##      4      8      3      11      17      26      5      7
## cricbigr crictrgr Critriia cromusss demisp. Diplcult Eukibrev Eukicoer
##      16      12      3      9      1      1      11      13
## eukideil eukigrgr Eukilobi Eukimino Eukisimi Hetemarc mictrasp micrchgr
##      13      1      19      2      1      2      26      14
## Nanoreag Natasp. Nilodubi orthrigr Orthrubi Orthfrig Orthobum orththie
##      21      4      12      15      24      3      27      9
## paracrsp Parastyl paratasp paraalgr Pararufi phaepssp polyconv polylagr
##      2      6      2      2      9      2      5      18
## polyscgr Pottgaed Pottlong Prodoliv Rheofusc rheotasp stembrgr Synosemi
##      10      1      14      4      23      23      3      27
## tanybrun tanytasp thellasp thieGrGe tvetbaca tvetdive Xenoxeno zavviasp
##      24      24      24      25      22      22      1      5
```

27. Z dataframu *spe* vytvořte nový dataframu *spe2*, obsahující všechny lokality s celkovou abundancí pakomárů rovnou alespoň 200.

```
spe2 <- spe[loc.sum >= 200, ]
dim(spe2)

## [1] 20 56
```

28. Z dataframu *spe2* odstraňte druhy vyskytující se pouze na jedné (nebo žádné) lokalitě. (V dataframu zůstanou druhy vyskytující se na alespoň dvou lokalitách)

```
spe2 <- spe2[, colSums(spe2 > 0) > 1]
dim(spe2)

## [1] 20 49
```

29. Dataframu *env* obsahuje některé proměnné prostředí naměřené na stejných lokalitách, kde byly odebrány vzorky pakomárů dataframu *spe*. Proměnná *velocity* obsahuje naměřené rychlosti proudu, nicméně některé její hodnoty jsou záporné, což je nesmysl. Převeďte proměnnou *velocity* dataframu *env* na absolutní hodnoty.

```
env$velocity <- abs(env$velocity)
```

30. Vytvořte novou proměnnou *froude* v dataframu *env*, do níž vypočítáte Froudeho čísla pro příslušné lokality. Vzoreček pro výpočet Froudeho čísla je: $Fr = U/(gD)^{1/2}$, kde $g = 9.81$, U je absolutní hodnota rychlosti proudu *velocity* a D je hloubka *depth*. Všechny proměnné najdete v dataframu *env*.

```
env$froude <- env$velocity/sqrt(9.81 * env$depth)
env
```

```
##      gr_env depth velocity   froude
## s1      Ep 0.395    0.274 0.139193
## s2      Ep 0.422    0.358 0.175951
## s3      Ep 0.496    0.310 0.140536
## s4 Ep_FPOM 0.291    0.078 0.046165
## s5 Ep_FPOM 0.320    0.092 0.051925
## s6 Ep_FPOM 0.328    0.126 0.070242
## s7      Er 0.278    0.490 0.296715
## s8      Er 0.213    0.618 0.427528
## s9      Er 0.243    0.508 0.329023
## s10 Er_VEG 0.353    0.758 0.407331
## s11 Er_VEG 0.236    0.334 0.219511
## s12 Er_VEG 0.245    0.810 0.522477
## s13 Ep_CPOM 0.184    0.088 0.065500
## s14      Ep 0.155    0.224 0.181655
## s15 Er_VEG 0.279    0.496 0.299809
## s16      Ep 0.501    0.372 0.167799
## s17      Ep 0.454    0.286 0.135520
## s18 Er_VEG 0.200    0.340 0.242733
## s19      Ep 0.162    0.050 0.039662
## s20      Er 0.290    0.804 0.476675
## s21 Er_VEG 0.138    0.590 0.507082
## s22      Ep 0.340    0.246 0.134698
## s23      Ep 0.344    0.234 0.127380
## s24 Er_VEG 0.300    0.958 0.558432
## s25      Er 0.139    0.630 0.539509
## s26 Ep_CPOM 0.124    0.002 0.001813
## s27 Ep_CPOM 0.478    0.044 0.020319
```

31. Zjistěte, kolik lokalit spadá do kategorie peřej (riffle) a kolik do kategorie tůň (pool). Kritériem je Froudeho číslo, jehož hodnota větší než 0,23 indikuje peřej.

```
# pereje
sum(env$froude > 0.23)

## [1] 11

# tune
sum(env$froude <= 0.23)

## [1] 16
```

32. Spočítejte, kolik druhů se vyskytovalo ve všech peřejích (lokalitách s Froudeovým číslem vyšším než 0,23). (myšleno v každé peřeji)

```
# perejovy subset:
spe.perej <- spe[env$froude > 0.23, ]
# kolik druhu ma abundanci > 0 v kazdem radku:
sum(colSums(spe.perej > 0) == nrow(spe.perej))

## [1] 6
```

33. Spočítejte, kolik druhů se vyskytovalo aspoň v polovině tůní (lokalit s Foudovým číslem nižším nebo rovným 0,23).

```
# tunovy subset:  
spe.tun <- spe[env$froude <= 0.23, ]  
# kolik druhu ma abundanci > 0 v alespon polovine radku:  
sum(colSums(spe.tun > 0) >= nrow(spe.tun)/2)  
  
## [1] 21
```