

Procvičování 6

Natáhněte si pracovní prostředí z <http://www.sci.muni.cz/syrovat/cv06.RData>.

```
load(url("http://www.sci.muni.cz/~syrovat/cv06.RData"))
```

1. Vytvořte faktor *sex* s úrovněmi *m* a *z*, z nichž každá se ve faktoru bude desetkrát opakovat.

```
sex <- factor(rep(c("m", "z"), each = 10))
sex

## [1] m m m m m m m m m m z z z z z z z z z z
## Levels: m z
```

2. Přejmenujte úrovně faktoru *sex* na *male* (*m*) a *female* (*z*).

```
# dve moznosti. bud zmenime jmena urovni pomoci argumentu labels= funkce
# factor():
factor(sex, labels = c("male", "female"))

## [1] male male male male male male male male male male
## [11] female female female female female female female female female female
## Levels: male female

# v tomhle pripade jsem ale nezmenil faktor sex, jen jsem ho nechal zmeneny
# vypsati. nebo primo prepiseme urovne fatoru sex:
levels(sex) <- c("male", "female")
sex

## [1] male male male male male male male male male male
## [11] female female female female female female female female female female
## Levels: male female
```

3. Přehodte úrovně faktoru *sex* tak, že první bude *female*.

```
# opet vic moznost, bud pomoci argumentu levels= funkce factor():
factor(sex, levels = c("female", "male"))

## [1] male male male male male male male male male male
## [11] female female female female female female female female female female
## Levels: female male

# nebo pomoci funkce relevel()
relevel(sex, "female")

## [1] male male male male male male male male male male
## [11] female female female female female female female female female female
## Levels: female male

# samozrejme jsem sex nezmenil, jen nechal vypsati reseni.
```

4. Z vektoru celých čísel od 1 do 25 vytvořte faktor *petice* odlišující pětice hodnot, jak jdou po sobě (1-5, 6-10, ..., 21-25). Úrovně tohoto faktoru nazvěte *nejmensi*, *maly*, *stredni*, *velky*, *nejvetsi*.

```
petice <- cut(1:25, breaks = c(0, 5, 10, 15, 20, 25), labels = c("nejmensi",
  "maly", "stredni", "velky", "nejvetsi"))
petice

## [1] nejmensi nejmensi nejmensi nejmensi nejmensi maly      maly
## [8] maly      maly      maly      stredni  stredni  stredni  stredni
## [15] stredni  velky    velky    velky    velky    velky    nejvetsi
## [22] nejvetsi nejvetsi nejvetsi nejvetsi
## Levels: nejmensi maly stredni velky nejvetsi

# nebo taky
cut(1:25, breaks = 0:5 * 5, labels = c("nejmensi", "maly", "stredni", "velky",
  "nejvetsi"))

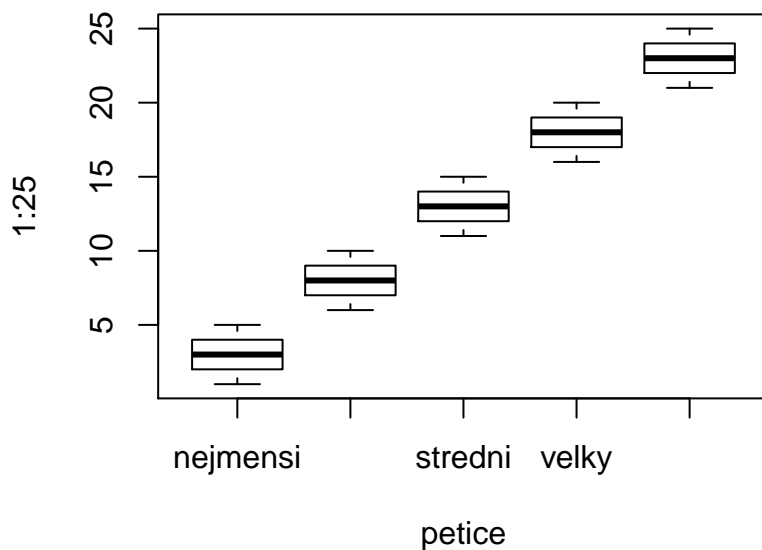
## [1] nejmensi nejmensi nejmensi nejmensi nejmensi maly      maly
## [8] maly      maly      maly      stredni  stredni  stredni  stredni
## [15] stredni  velky    velky    velky    velky    velky    nejvetsi
## [22] nejvetsi nejvetsi nejvetsi nejvetsi
## Levels: nejmensi maly stredni velky nejvetsi

# nebo taky treba takhle
cut(1:25, breaks = seq(0, 25, 5), labels = c("nejmensi", "maly", "stredni",
  "velky", "nejvetsi"))

## [1] nejmensi nejmensi nejmensi nejmensi nejmensi maly      maly
## [8] maly      maly      maly      stredni  stredni  stredni  stredni
## [15] stredni  velky    velky    velky    velky    velky    nejvetsi
## [22] nejvetsi nejvetsi nejvetsi nejvetsi
## Levels: nejmensi maly stredni velky nejvetsi
```

5. Pro kontrolu zobrazte vektor 1-25 proti faktoru *petice*.

```
par(mar = c(4, 4, 1, 0.1))
plot(1:25 ~ petice)
```



6. Zjistěte, jaké proměnné jsou v dataframě *env*.

```
names(env) # nebo colnames(env)
## [1] "gr_env" "depth" "velocity" "froude"
```

7. Přejmenujte proměnnou *gr_env* dataframě *env* na *hab*.

```
names(env)[names(env) == "gr_env"] <- "hab"
```

8. Přesvědčte se, že proměnná *hab* dataframě *env* je faktor (odpovědí **R** bude TRUE / FALSE).

```
is.factor(env$hab)
## [1] TRUE
```

9. Zjistěte, kolik úrovní tento faktor má a jaké to jsou.

```
nlevels(env$hab)
## [1] 5

levels(env$hab)
## [1] "Ep" "Ep_CPOM" "Ep_FPOM" "Er" "Er_VEG"
```

10. Přejmenujte úrovně faktoru *hab* dataframě *env* následovně: *Ep_CPOM* na *CPOM*, *Ep_FPOM* na *FPOM*, *Ep* na *P*, *Er* na *R*, *Er_VEG* na *VEG*.

```

# 2 možnosti, vždy ale musíme dodržet stejné pořadí úrovně, v jakém jsou v
# definici faktoru. buď použijeme factor():
factor(env$hab, labels = c("P", "CPOM", "FPOM", "R", "VEG"))

## [1] P P P FPOM FPOM FPOM R R R VEG VEG VEG CPOM P
## [15] VEG P P VEG P R VEG P P VEG R CPOM CPOM
## Levels: P CPOM FPOM R VEG

# nebo prepíšeme přímo úrovně:
levels(env$hab) <- c("P", "CPOM", "FPOM", "R", "VEG")
env$hab

## [1] P P P FPOM FPOM FPOM R R R VEG VEG VEG CPOM P
## [15] VEG P P VEG P R VEG P P VEG R CPOM CPOM
## Levels: P CPOM FPOM R VEG

```

11. V dataframu *env* vytvořte kopii faktoru *hab* *hab2*, v níž spojíte úrovně CPOM a FPOM do jedné OM.

```

env$hab2 <- env$hab
# vidím, že spojované úrovně jsou ne 2. a 3. místě, proto nahradím 2. a 3.
# úroveň:
levels(env$hab2)[2:3] <- "OM"
# pokud bych ale nechtěl manuálně odpocítávat, lze to uselat obecněji:
env$hab2 <- env$hab
levels(env$hab2)[levels(env$hab2) %in% c("CPOM", "FPOM")] <- "OM"
# výsledek můžeme zkontrolovat pomocí krizové tabulky z funkce table()
table(env$hab, env$hab2)

##
##      P OM R VEG
## P    9 0 0 0
## CPOM 0 3 0 0
## FPOM 0 3 0 0
## R    0 0 5 0
## VEG  0 0 0 7

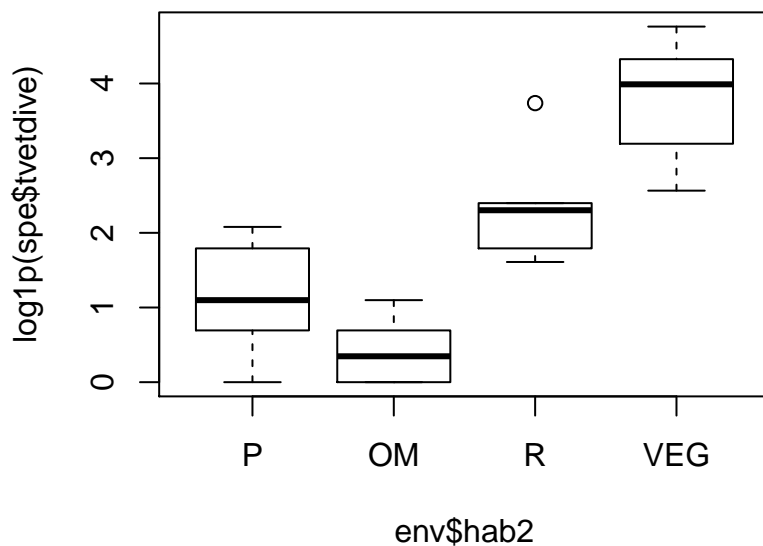
```

12. Zobrazte krabicový graf závislosti logaritmované abundance druhu *Tvetenia discoloripes*, *tvetdive* dataframu *spe* na typu habitatu.

```

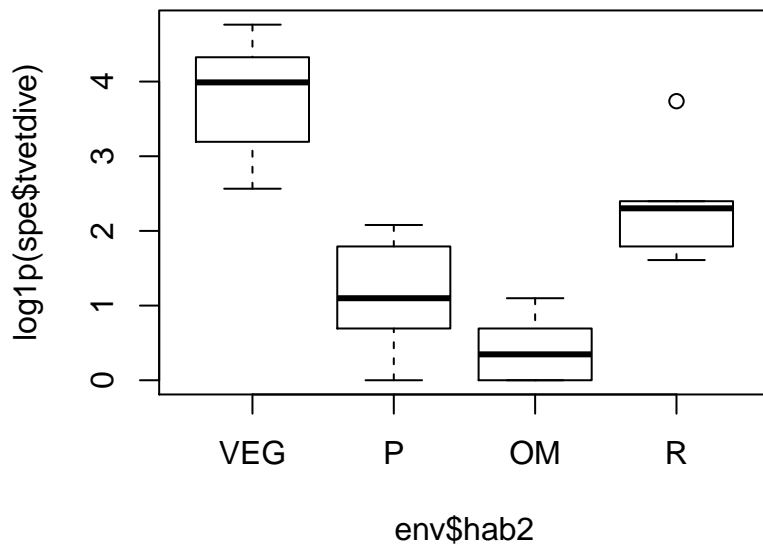
par(mar = c(4, 4, 1, 0.1))
plot(log1p(spe$tvetdive) ~ env$hab2)

```



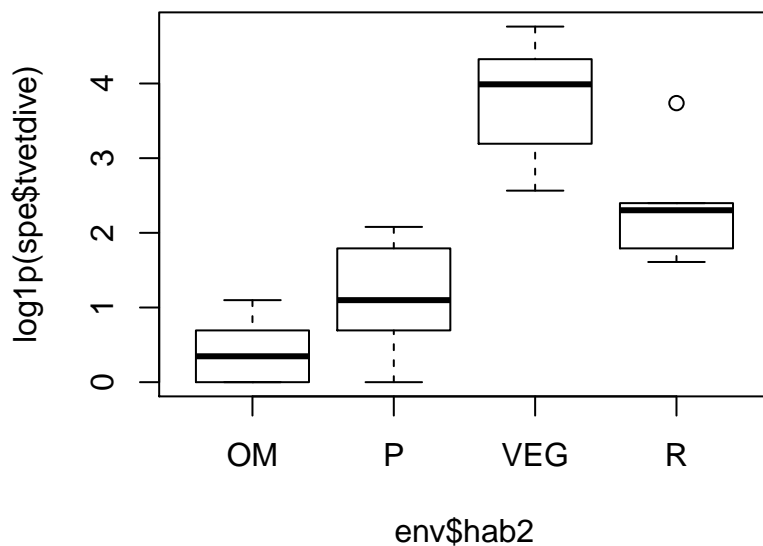
13. Přeařďte úrovně faktoru *hab2* tak, aby úroveň *VEG* byla první - referenční a vykreslete graf znovu.

```
env$hab2 <- relevel(env$hab2, "VEG")
par(mar = c(4, 4, 1, 0.1))
plot(log1p(spe$tvtdive) ~ env$hab2)
```



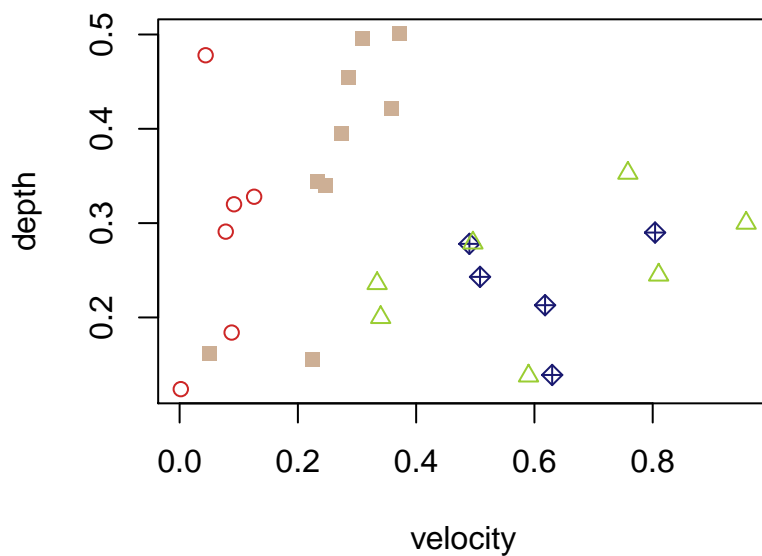
14. Seřadte úrovně faktoru *hab2* podle průměrného Froudeho čísla v jednotlivých úrovních a vykreslete graf znovu.

```
env$hab2 <- reorder(env$hab2, env$froude, mean)
par(mar = c(4, 4, 1, 0.1))
plot(log1p(spe$tvtdive) ~ env$hab2)
```



15. Zobrazte bodový graf závislosti hloubky *depth* na rychlosti proudu *velocity* a typem symbolů a barvou odlište vzorky z jednotlivých habitatů *hab2*. Typ symbolu vybírejte z hodnot 1:25, barvu z `colours()`.

```
par(mar = c(4, 4, 1, 0.1))
plot(depth ~ velocity, data = env, pch = c(1, 15, 2, 9)[hab2], col = c("firebrick3",
"peachpuff3", "olivedrab3", "midnightblue")[hab2])
```



16. V dataframě *env* vytvořte faktor *hydr* s úrovněmi *pool*, *run* a *riffle*, odlišující tůňové (*pool*), slapové (*run*) a peřejové (*riffle*) habitaty. Jako kritérium pro odlišení habitatů použijte Froudeho číslo (hraniční hodnoty $pool \leq 0,18 < run \leq 0,41 < riffle$).

```

env$hydr <- cut(env$froude, breaks = c(0, 0.18, 0.41, max(env$froude)), include.lowest = T,
  labels = c("pool", "run", "riffle"))
env$hydr

## [1] pool  pool  pool  pool  pool  pool  run   riffle run   run
## [11] run   riffle pool  run   run   pool  pool  run   pool  riffle
## [21] riffle pool  pool  riffle riffle pool  pool
## Levels: pool run riffle

```

17. Nechejte si vypsát faktor *hydr* jako čísla.

```

as.numeric(env$hydr)

## [1] 1 1 1 1 1 1 2 3 2 2 2 3 1 2 2 1 1 2 1 3 3 1 1 3 3 1 1

```

18. Seřadte úrovně faktoru *hydr* sestupně podle mediánu Froudeho čísla. (Když hodnoty převedeme na záporné, ty nejvyšší se stanou nejnižšími.)

```

env$hydr <- reorder(env$hydr, -env$froude, median)
levels(env$hydr)

## [1] "riffle" "run" "pool"

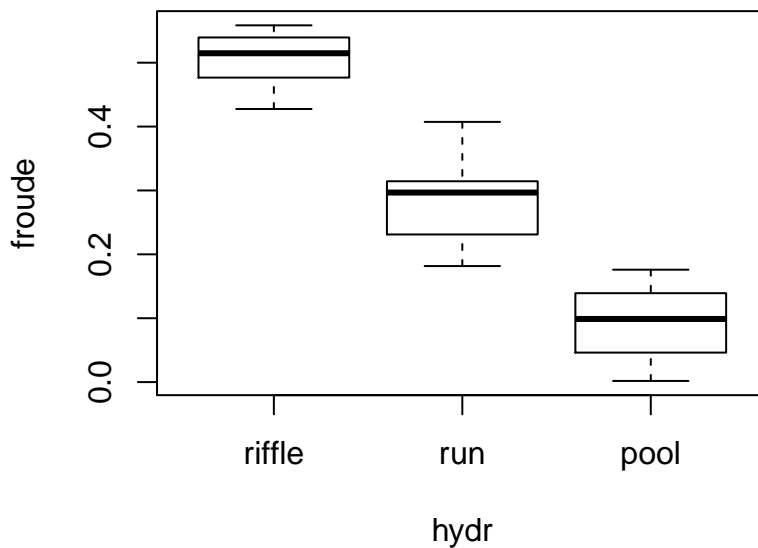
```

19. O správném seřazení se přesvědčte pomocí boxplotu Froudeho čísla v závislosti na *hydr*.

```

par(mar = c(4, 4, 1, 0.1))
plot(froude ~ hydr, data = env)

```



20. Obdobně jako v bodě 14 zobrazte bodový graf závislosti hloubky *depth* na rychlosti proudu *velocity*, ale barvou odlište lokality s odlišnými hydraulickými podmínkami (*hydr*). Barvy volte tak, aby vhodně ilustrovaly postupně se měnící hydraulické podmínky, tedy např. podobného odstínu, pool nejsvětlejší a riffle nejtmaší.

```
par(mar = c(4, 4, 1, 0.1))  
plot(depth ~ velocity, data = env, col = c("midnightblue", "royalblue2", "lightskyblue1")[hydr],  
      pch = 16, cex = 1.5)
```

