Procvičování 1 s řešením

R jako kalkulačka

1. Sečtěte 12 a 5

Normálně použijeme **R** jako kalkulačku:

```
12 + 5
```

```
## [1] 17
```

2. Vynásobte 5 a 8

```
5 * 8
```

```
## [1] 40
```

3. Spočtěte druhou mocninu rozdílu dvou čísel z bodů 1) a 2)

Buď můžeme opravit předchozí příkazy tak, aby se jejich výsledky uložily do objektů, třeba a a b, a jejich rozdíl následně umocníme:

```
a <- 12 + 5
b <- 5 * 8
(a - b)^2
```

```
## [1] 529
```

Nebo je opíšeme do jednoho příkazu:

```
((12 + 5) - (5 * 8))^2
```

```
## [1] 529
```

4. Vypočtěte Froudeho číslo Fr pro rychlost proudu $U = 0.65$ m.s-1 a hloubku $D = 0.24$ m. $Fr = U/(gD)^{1/2}$, kde $g = 9.81$. (Froudeho číslo je hydraulický parametr, o němž bude řeč příště)

Pro odmocnění máme několik možností, jinak je to kalkulačková záležitost:

```
0.65/(9.81 * 0.24)^(1/2)
```

```
## [1] 0.4236
```

```
0.65/(9.81 * 0.24)^(0.5)
```

```
## [1] 0.4236
```

```
0.65/sqrt(9.81 * 0.24)
```

```
## [1] 0.4236
```

Nápověda

5. Zjistěte, jak se vypočítá logaritmus (logarithm) při základě 10 a vypočtěte ho pro číslo 1000 (měli byste dostat hodnotu 3). K čemu je funkce `log1p()`?

Nejprve vyhledáme funkci, která počítá logaritmus, anglicky logarithm. `??logarithm` nám otevře okno se seznamem funkcí, v jejichž nápovědě se řetězec logarithm vyskytuje. Mezi nimi je funkce `log()`, podle popisu počítá *Logarithms and Exponentials*, to je co hledáme. Když otevřeme nápovědu k této funkci `?log` (případně kliknutím na jméno funkce, pokud používáme RStudio), dočteme se, že pro výpočet logaritmu při základu 10 máme 2 způsoby. Buď použít funkci `log()` s argumentem `base = 10`, nebo funkci `log10()`..

Ve stejné nápovědě se také dočteme, že `log1p()` počítá přirozený loaritmus z log(x+1).

```
log(1000, 10)

## [1] 3

log10(1000)

## [1] 3
```

6. Zjistěte, na co je funkce `rep()`?

Vyvoláme nápovědu k funkci `?rep`, kde se dočteme, že funkce `rep()` slouží k vytváření repeticí.

7. Vytvořte sekvenci čísel od 0 do 1 po 0.1. (v **R** musí být použita desetinná tečka, nikoliv čárka)

Nejprve musíme najít funkci, která vytváří sekvence, anglicky sequence. Vyhledáme tedy `??sequence`. V otevřeném okně pak najdeme, že ke generování sekvencí slouží funkce `seq()`. Vyvoláme nápovědu k této funkci ?seq a přečteme si o jejím použití, můžeme si vyzkoušet i příklady na konci nápovědy, které stačí jen vkládat do **R**. Podle nápovědy bychom měli pochopit, že sekvenci od 0 do 1 po 0,1 vytvoříme zadáním argumentů `from =`, `to =` a `by =` (od, do, po).

```
seq(from = 0, to = 1, by = 0.1)

##  [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

Ze cvíka už víme, že pokud zachováme pořadí argumentů shodné s pořadím uvedeným v nápovědě, nemusíme vypisovat jejich názvy. Stačí tedy:

```
seq(0, 1, 0.1)

##  [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

Nápověda + kalkulačka

8. Vypočtěte, jaká musí být hloubka, aby Fr bylo 0.23, když rychlost proudu bude 0.35 $m.s^{-1}$ a zaokrouhlete výsledek na 2 desetinná místa ($D = (U/Fr)^2/g$). (anglicky zaokrouhlit je *round*)

Vypočítat hloubku nebude problém, stačí dosadit do rovnice:

```
(0.35/0.23)^2/8.91
```

```
## [1] 0.2599
```

Pro zaokrouhlení zase musíme najít funkci. Pomocí `??round` zjistíme, že k zaokrouhlování slouží trochu překvapivě funkce `ceiling()` (v překladu "strop"). Nicméně v nápovědě `?ceiling` se dozvíme, že funkce `ceiling()` zaokrouhluje nahoru a klasické zaokrouhlení provádí funkce `round()`, argument `digits =` pak udává počet desetinných míst (nemusíme ho ale jmenovat, pokud bude na druhém místě).

```
D <- (0.35/0.23)^2/8.91
round(D, 2)
```

```
## [1] 0.26
```

Nebo celé naráz:

```
round((0.35/0.23)^2/8.91, 2)
```

```
## [1] 0.26
```