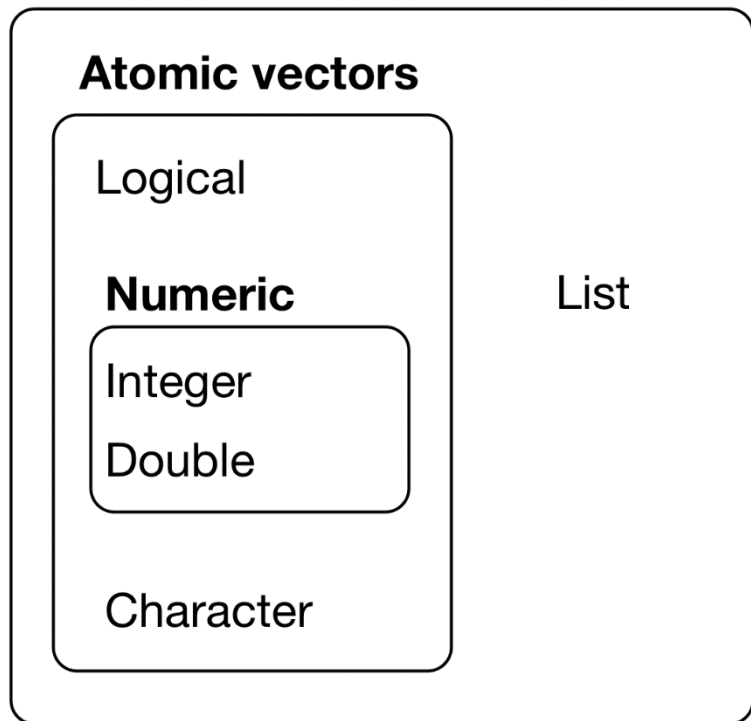


Úvod do R - Vektory

Vít Syrovátka

Vektory

Vectors



NULL

2 typy vektorů:

- **Atomic** (atomické) – homogenní, mohou obsahovat pouze hodnoty jednoho typu
- **List** (seznam) - může být heterogenní, obsahovat hodnoty více typů

NULL značí prázdný vektor

2 vlastnosti vektorů:

- **typ** hodnot
- **délka** `length()` – počet elementů

Typ hodnot

`typeof()`

- zjistí typ hodnot
- rozlišuje mezi `numeric` (double) a `integer`

`mode()`

- zjistí, ale i nastaví typ hodnot
- nekonzistentní v rozlišování `numeric` a `integer`

	<code>mode()</code>	<code>typeof()</code>
1:5	"numeric"	"integer"
1.2	"numeric"	"double"
"a"	"character"	"character"
TRUE	"logical"	"logical"

```
x <- 1:5
```

```
> mode(x)
[1] "numeric"
```

```
> typeof(x)
[1] "integer"
```

```
> mode(x) <- "numeric"
> mode(x)
[1] "numeric"
```

```
> typeof(x)
[1] "double"
```

Numeric, double, integer

Numeric

Integer

Double

- celá čísla, pouze jeden typ NA hodnot
- explicitně značíme velkým L: 2L

- plovoucí čísla (floating numbers)
- pozor na porovnávání hodnot pomocí `==`, raději použít speciální funkce `all.equal()`, nebo `dplyr::near()`
- 4 typy NA hodnot: NA, NaN, -Inf, Inf

```
> c(-1, 0, 1) / 0  
[1] -Inf NaN Inf
```

```
> x<- sqrt(2)^2  
> x  
[1] 2
```

```
> x == 2  
[1] FALSE
```

```
> x - 2  
[1] 4.440892e-16
```

```
> all.equal(x, 2)  
[1] TRUE
```

```
> dplyr::near(x, 2)  
[1] TRUE
```

NA a NaN hodnoty

NA – (Not Available) neznámá, chybějící hodnota

NaN – (Not a Number) neexistující, nemožná hodnota, např. výsledek
0/0

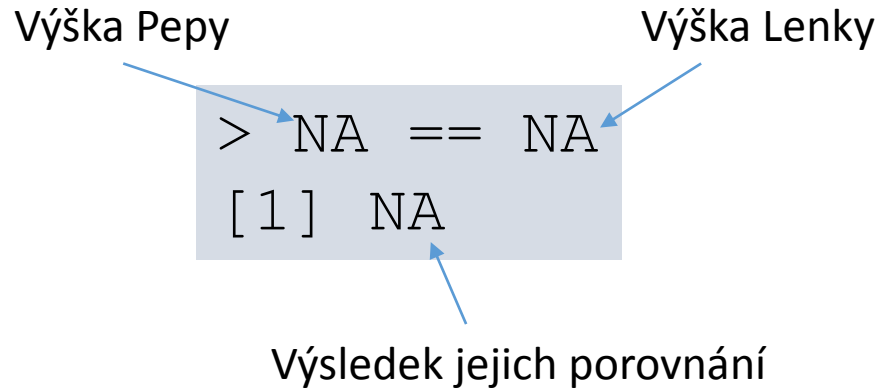
-Inf a Inf – (Infinite) (mínus) nekonečno

Testování pomocí speciálních funkcí: `is.na()`, `is.nan()`,
`is.finite()`, `is.infinite()`

```
> is.na(NA)
[1] TRUE
```

NA hodnoty

„Výšku Pepy neznáme, je **NA**. Výšku Lenky také neznáme, také je **NA**. **Je Lenka stejně vysoká jako Pepa?** Těžko říct, nevíme, je také **NA**.“



The diagram illustrates the R code and its output for comparing two NA values. A light blue rectangular box contains the code and its output. Three blue arrows point from external text labels to specific parts of the code and output.

```
> NA == NA
[1] NA
```

Výška Pepy

Výška Lenky

Výsledek jejich porovnání

NA hodnoty

Neznáme hodnoty → neznáme výsledek

```
> x <- c(4, 2, 0, NA)
> mean(x)
[1] NA
```



“Ok, odstraňme ty NA hodnoty!”

```
> mean(x, na.rm = TRUE)
[1] 2
```

Logical

- nejjednodušší typ
- pouze 3 hodnoty: FALSE, TRUE, NA
- nejčastěji výsledkem porovnání hodnot pomocí logického výrazu
- lze převést na čísla: FALSE = 0, TRUE = 1

```
> c(T, F, NA)
[1]  TRUE FALSE   NA

> x <- c(4, 2, 0, NA)
> x > 0
[1]  TRUE  TRUE FALSE   NA

> (x > 0) + 1
[1]  2  2  1 NA
```


Character

- nejsložitější typ
- každý element tvořen libovolným řetězcem znaků libovolné délky
- elementy uzavřeny v uvozovkách dvojitých "text", nebo jednoduchých 'text'.

```
> c("Pepicek", "a", "Marenka", "se maji radi tuze radi.",  
"To je ale dlouhy retezec!")  
  
[1] "Pepicek"           "a"  
[3] "Marenka"          "se maji radi tuze radi."  
[5] "To je ale dlouhy retezec!"
```

Coercion – změna typu hodnot

- **explicitní** – funkce rodiny `as.:` `as.numeric()`, `as.integer()`, `as.double()`, `as.logical()`, `as.character()`
- **implicitní** – při použití v daném kontextu, např.:
 - logické hodnoty se při matematickém výpočtu převedou na čísla: `FALSE = 0`, `TRUE = 1`

```
> x <- sample(6, 50, replace= T)
> sum(x > 3) # počet hodnot, které splňují podmínku
[1] 25

> mean(x > 3)
[1] 0.5 # podíl hodnot, které splňují podmínku
```

- spojení vektorů odlišných typů vede k převodu na složitější z typů

Pojmenování hodnot vektoru

- během vytváření vektoru

```
> c(a = 1, b = 5, c = 14)
a  b  c
1  5 14
```

- po vytvoření

```
> x <- c(1, 5, 14)
> names(x) <- c("a", "b", "c")
> x
a  b  c
1  5 14
```

- odstranění jmen

```
> names(x) <- NULL
> x
[1] 1 5 14
```

Podvýběry

Podvýběr se provádí pomocí funkce `[`.

```
vektor[subscript]
```

subscript:

1. numerický

- pozice vybraných hodnot
- záporný subscript označí vyloučené hodnoty
- pozice se mohou opakovat

```
> x <- letters[1:5]
> x
[1] "a" "b" "c" "d" "e"

> x[c(1, 3, 2)]
[1] "a" "c" "b"

> x[-1]
[1] "b" "c" "d" "e"

> x[c(1, 3, 2, 3, 5, 5)]
[1] "a" "c" "b" "c" "e" "e"
```

Podvýběry

Podvýběr se provádí pomocí funkce `[`.

```
vektor[subscript]
```

subscript:

1. numerický

- pozice vybraných hodnot
- záporný subscript označí vyloučené hodnoty
- pozice se mohou opakovat

2. logický

- TRUE hodnoty budou zachované
- obvykle stejné délky jako cílový vektor

```
> x <- c(5, 4, 1, 8, 0, NA, 3)
> x[x > 4]
[1] 5 8 NA

> x[!is.na(x)]
[1] 5 4 1 8 0 3

> x[x%%2 == 0]
[1] 4 8 0 NA
```

Podvýběry

Podvýběr se provádí pomocí funkce `[]`.

```
vektor[subscript]
```

subscript:

1. numerický

- pozice vybraných hodnot
- záporný subscript označí vyloučené hodnoty
- pozice se mohou opakovat

2. logický

- TRUE hodnoty budou zachované
- obvykle stejné délky jako cílový vektor

3. jmenný

- pokud je vektor pojmenovaný
- jména je možné opakovat

```
> x <- c(aaa = 1, bbb = 6, iou = 44)
> x
aaa bbb iou
  1   6  44

> x[c("iou", "aaa", "iou")]
iou aaa iou
 44   1  44
```