R Markdown

MD

3/14/23

Induktivní (vs. deduktivní) úsudky

- logické úsudky
  Jestliže je Bohouš v Bratrstvu, tak nemá rád Rychlé šípy.

  Bohouš je v Bratrstvu.

  $\models$ *Bohouš nemá rád Rychlé šípy.*

- chybný úsudek:
  Jestliže je Bohouš v Bratrstvu, tak nemá rád Rychlé šípy.

  Bohouš není v Bratrstvu.

  $\not\models$ *Bohouš má rád Rychlé šípy.*

- deduktivní platnost

```
#library(reticulate)
#py_install("nltk")
```

Python + Prover9

```
import nltk
read_expr = nltk.sem.Expression.fromstring

nltk.boolean_ops()

# modus ponens
```

```
negation        -
conjunction     &
disjunction     |
implication     ->
equivalence     <->
```

```
lp = nltk.sem.Expression.fromstring

BohJeBra = read_expr('BohJeBra')
NotRadRS = read_expr('-RadRS')
Rule = read_expr('BohJeBra -> -RadRS')
prover = nltk.Prover9()

print(prover.prove(NotRadRS, [BohJeBra, Rule]))
```

```
True
```

Monotonicita

```
BohJeBra = read_expr('BohJeBra')
Prsi = read_expr("Prsi")
NotRadRS = read_expr('-RadRS')
Rule = read_expr('BohJeBra -> -RadRS')
prover = nltk.Prover9()

print(prover.prove(NotRadRS, [BohJeBra, Prsi, Rule]))
```

```
True
```

Chybný úsudek

```
BohNeBra = read_expr('-BohJeBra')
RadRS = read_expr('RadRS')
Rule = read_expr('BohJeBra -> -RadRS')
prover = nltk.Prover9()

print(prover.prove(RadRS, [BohNeBra, Rule]))
```

```
False
```

Induktivní úsudky jsou ne-monotonní

```r
# Performs an exact test of a simple null hypothesis about the probability of success in a Bernoulli experiment
# first 62 cases are true (first 62 P are Q)
binom.test( x=62, n=100, p=.5 )
```

```
    Exact binomial test

data:  62 and 100
number of successes = 62, number of trials = 100, p-value = 0.02098
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.5174607 0.7152325
sample estimates:
probability of success
                  0.62
```

ale po prvních 100 pouze 38 z druhých 100 P je Q

```r
binom.test( x=100, n=200, p=.5 )
```

```
    Exact binomial test

data:  100 and 200
number of successes = 100, number of trials = 200, p-value = 1
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.4286584 0.5713416
sample estimates:
probability of success
                   0.5
```

Pravděpodobnosti součtů (2 kostky) i

```r
#vec1 <- sample(1:6, replace = TRUE)
#vec2 <- sample(1:6, replace = TRUE)

vec1 <- sample(1:6, 1000, replace=TRUE)
vec2 <- sample(1:6, 1000, replace=TRUE)

df <- cbind.data.frame(vec1, vec2)
str(df)
```

```
'data.frame':   1000 obs. of  2 variables:
 $ vec1: int  5 2 6 3 2 1 2 5 1 2 ...
 $ vec2: int  3 6 6 3 1 3 4 1 3 6 ...
```

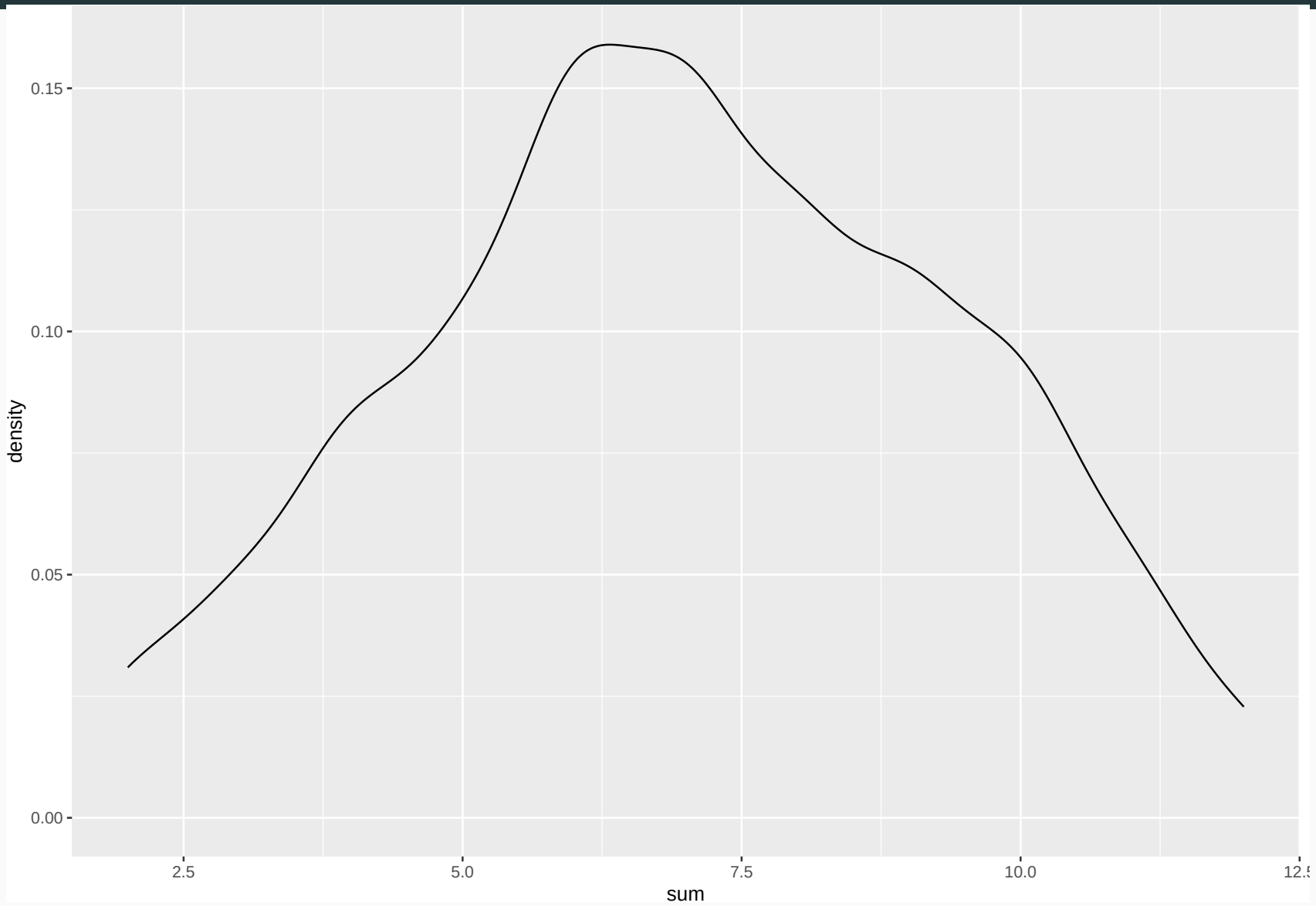
```r
library(dplyr)

df <- df %>%
  mutate(sum = rowSums(select(.,vec1:vec2)))

str(df)
```

```
'data.frame':   1000 obs. of  3 variables:
 $ vec1: int  5 2 6 3 2 1 2 5 1 2 ...
 $ vec2: int  3 6 6 3 1 3 4 1 3 6 ...
 $ sum : num  8 8 12 6 3 4 6 6 4 8 ...
```

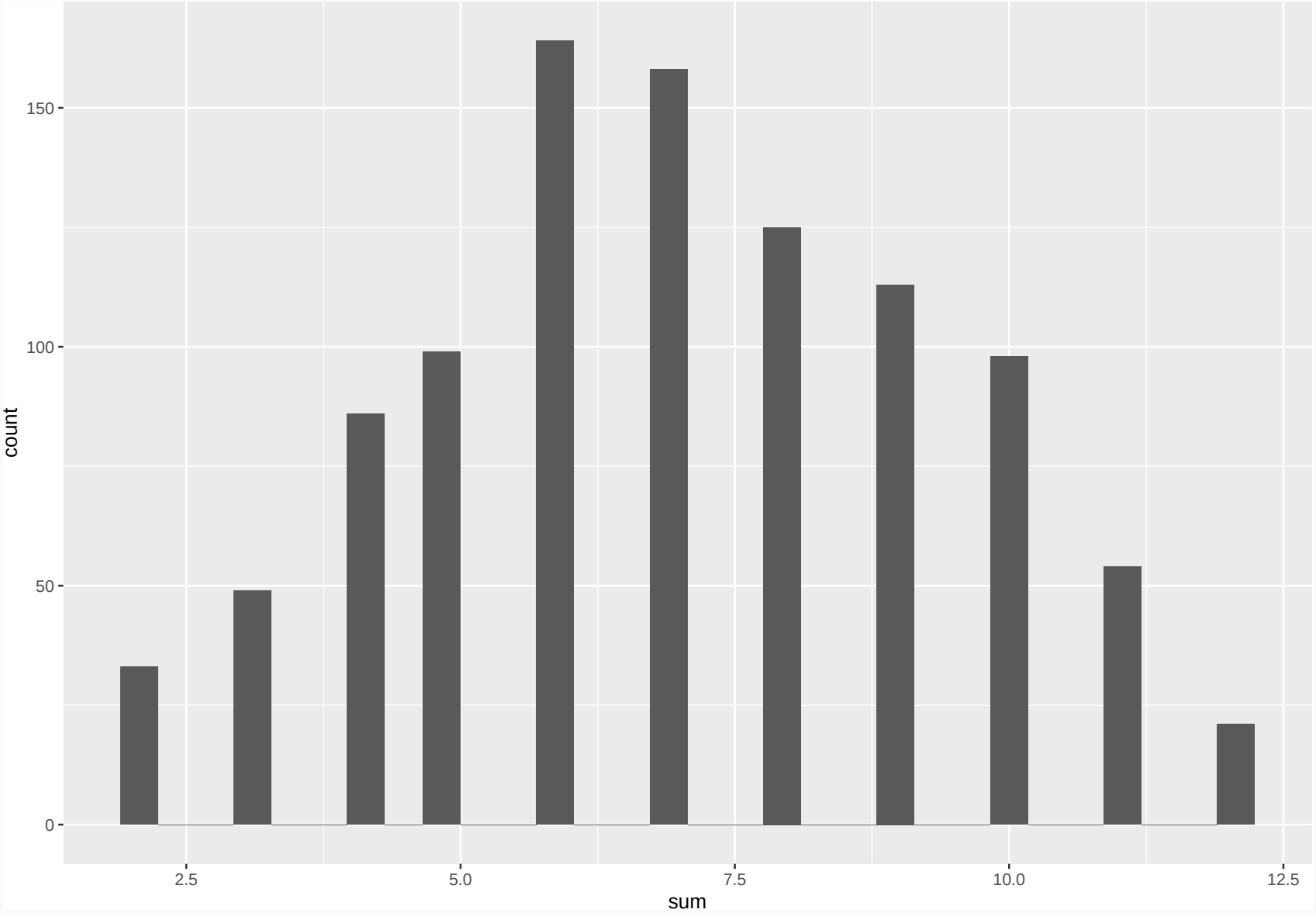
Pravděpodobnosti součtů (2 kostky) ii

```
library(ggplot2)
# Basic density
p <- ggplot(df, aes(x=sum)) +
  geom_density()
p
```

Pravděpodobnosti součtů (2 kostky) iii



```
q <- ggplot(df, aes(x=sum)) + geom_histogram()
q
```



R Markdown

- toto je příklad dokumentu v R Markdown
- Markdown je jednoduchý značkovací jazyk pro vytváření HTML, PDF a MS Word dokumentů
- jedna z adres: http://rmarkdown.rstudio.com.
- syntax je velmi jednoduchá: italika pomocí *výraz*
- bold: **výraz**
- odrážky pomocí: - - pod sebou
- více
    - zanořování
        - zanořování

- ale umožňuje TeX-zápis matematických formulí:

$$\forall x[\mathbf{man}(x) \rightarrow \exists y[\mathbf{woman}(y) \wedge \mathbf{love}(x,y)]]$$

- vzniklo z:

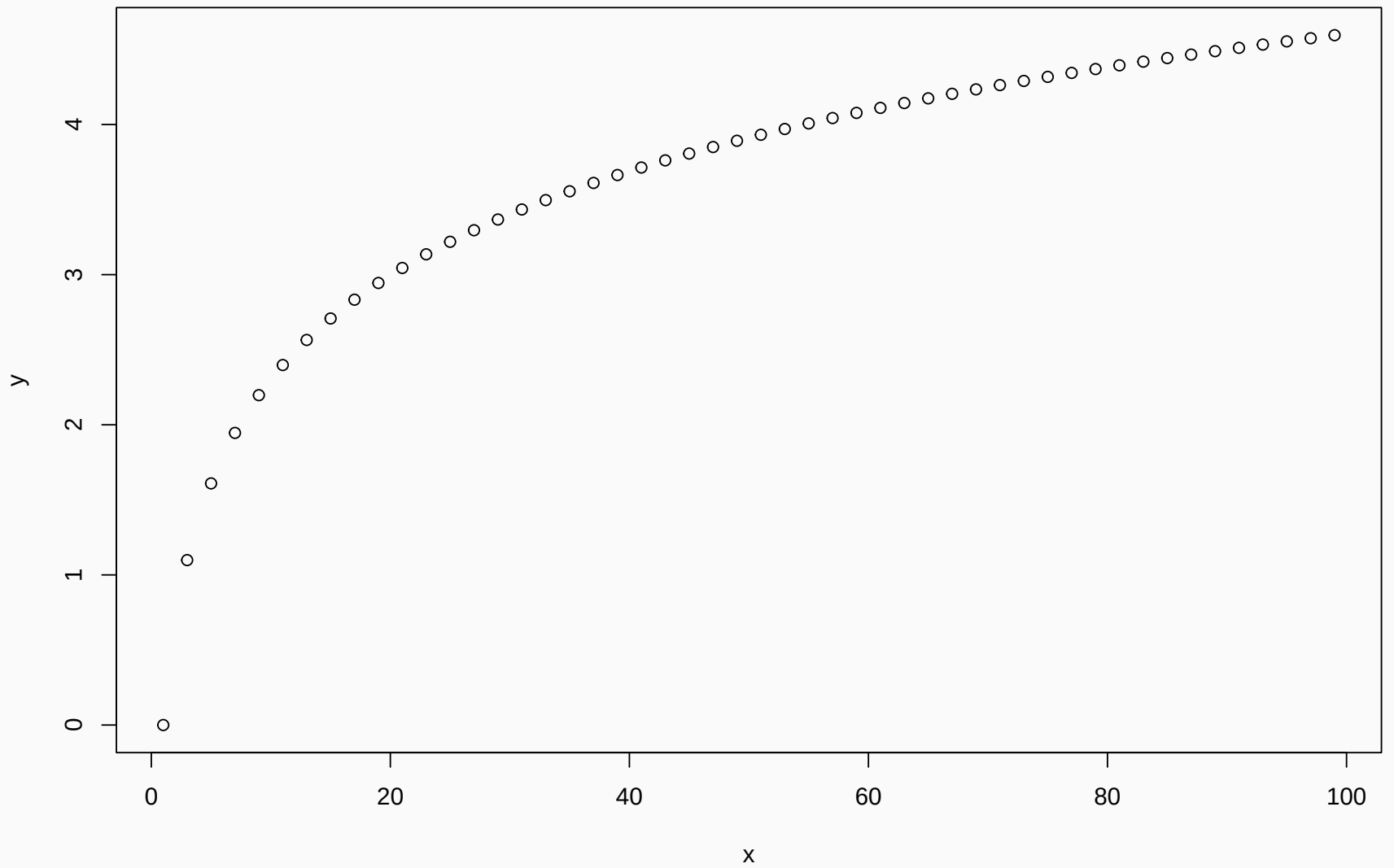
```
\forall x[\mathbf{man}(x) \rightarrow
\exists y[\mathbf{woman}(y) \wedge \mathbf{love}(x,y)]]
```

- asi nejpodstatnější: umožňuje interaktivní zanořování R kódu

```r
x <- seq(1, 100, by = 2)
y <- log(x)
y
```

```
 [1] 0.000000 1.098612 1.609438 1.945910 2.197225 2.397895 2.
 [9] 2.833213 2.944439 3.044522 3.135494 3.218876 3.295837 3.
[17] 3.496508 3.555348 3.610918 3.663562 3.713572 3.761200 3.
[25] 3.891820 3.931826 3.970292 4.007333 4.043051 4.077537 4.
[33] 4.174387 4.204693 4.234107 4.262680 4.290459 4.317488 4.
[41] 4.394449 4.418841 4.442651 4.465908 4.488636 4.510860 4.
[49] 4.574711 4.595120
```

```
plot(x,y)
```



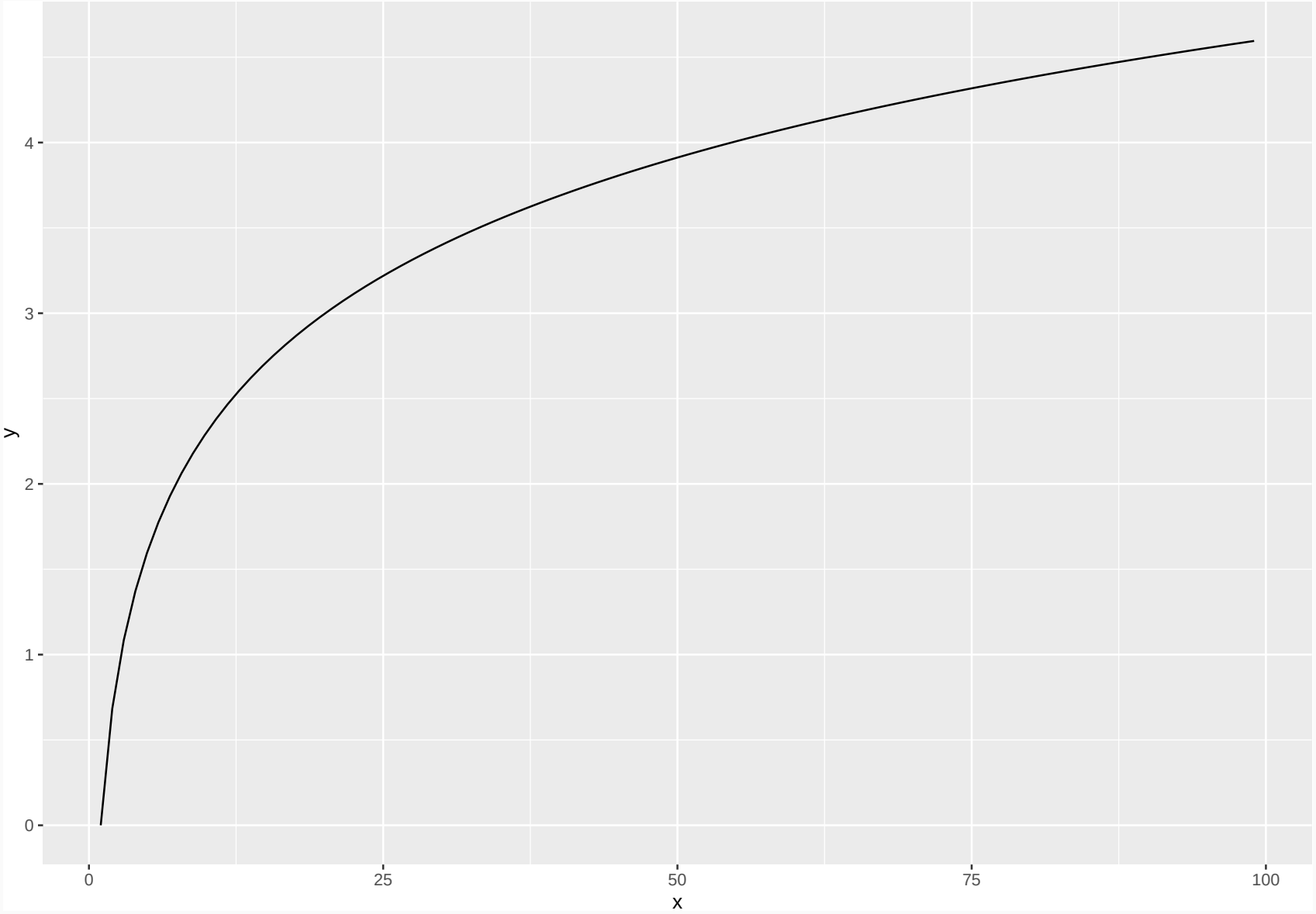

```
str(x)
```

```
 num [1:50] 1 3 5 7 9 11 13 15 17 19 ...
```

```r
library(ggplot2)
df <- data.frame(x)
str(df)
```

```
'data.frame':   50 obs. of  1 variable:
 $ x: num  1 3 5 7 9 11 13 15 17 19 ...
```

```
ggplot(df,aes(x)) + stat_function(fun=function(x) log(x))
```



- chunk Pythonu
- lepší na algebru, logiku, ...

```
A = {1, 2, 1}
B = {1, 2}
A == B  # Python will say 'True'
```

```
True
```

```
C = {3, 4}
A.union(C)
```

```
{1, 2, 3, 4}
```

```
A.intersection(C)
```

```
set()
```

```
len(A)  # tells you the size of A
```

```
2
```

```
A <= B  # checks if A is a subset of B
```

```
True
```

```
A < B   # checks if A is a proper subset of B
```

```
False
```

```
{(a, b) for a in A for b in B} # Cartesian product AxB
```

```
{(1, 1), (1, 2), (2, 1), (2, 2)}
```

```
emptyset = set()
A = {1, 2}
C = {3, 4}
A.intersection(C) == emptyset  # True
```



```
True
```

- pluralities

```
#Python program to find powerset
from itertools import combinations
def print_powerset(string):
    for i in range(0,len(string)+1):
        for element in combinations(string,i):
            print(''.join(element))
string=['a','b','c']
print_powerset(string)
```

```
a
b
c
ab
ac
bc
abc
```

- pointwise sum

```
# Python 3

list1 = ['a', 'b', 'c']
list2 = ['d', 'e', 'f']

assert len(list1) == len(list2)
result = [list1[i] + list2[i] for i in range(len(list1))]
print(result)
```

```
['ad', 'be', 'cf']
```

Knížky

https://bookdown.org/

- R Markdown: The Definitive Guide
- case studies

- moje nejčastější použití: slidy a reporty

```
---
title: "Habits"
author: John Doe
date: March 22, 2005
output: beamer_presentation
---

# In the morning

## Getting up

- Turn off alarm
- Get out of bed

 ...
---

## Going to sleep
...

---
```

In the morning

Getting up

- Turn off alarm
- Get out of bed

Breakfast

- Eat eggs
- Drink coffee

In the evening

Dinner

- Eat spaghetti
- Drink wine

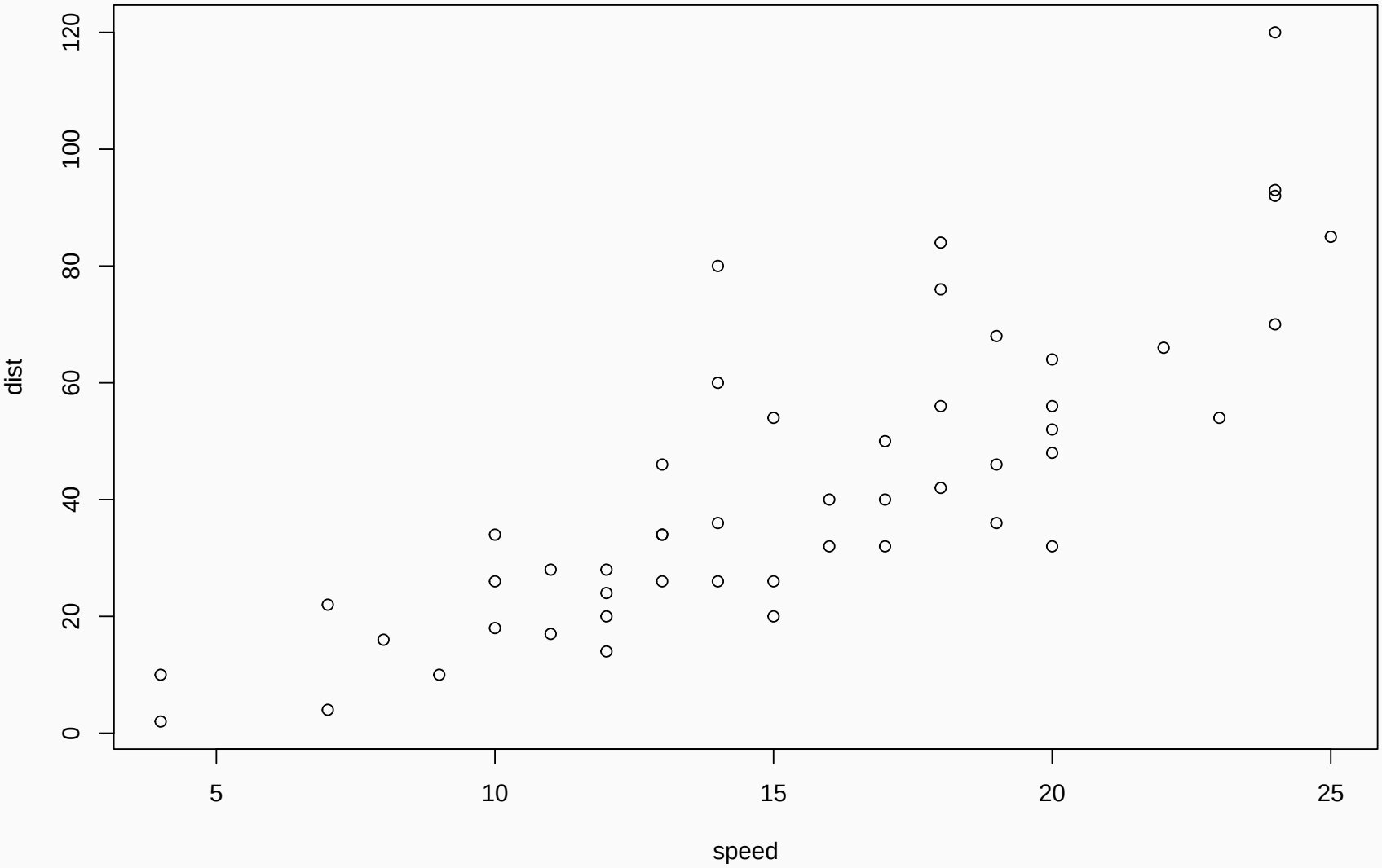


Figure 1: A scatterplot.

Going to sleep

- Get in bed
- Count sheep

- rmarkdown: trochu příliš lehký
- rozšíření směrem k akademickému použití: TeXové citace:

```
bibliography: biblio.bib
...
    includes:
      in_header: preamble.tex
    latex_engine: xelatex
    citation_package: natbib
```

  - citace: @R-base
  - vytvoří: R Core Team (2016)

- lingvistika: příklady
- expex makra:

(1) One brown mouse jumped over the fence.

```
\ex One brown mouse jumped over the fence.
\xe
```

- markdown je oficiální markovací jazyk na GitHubu
- dostanete se do polar vault
- ideální pro spolupráci:
- GitHub jako univerzálně použitelné úložiště pro experimenty, programování, etc.:

- největší plus: velmi propracovaný verzovací systém
- editovatelný zdroják experimentu
- samotný experiment na IBEXu
- historie verzí

- práce s open-source knihami:
- fork knihy Hands-On Programming with R
- https://github.com/MojmirDocekal/hopr
- změněná kapitola bacics.rmd

- vlastní report: File -> New File -> R Markdown
- insert R chunk
- ...
- rychlá kontrola: knit to html

R a TeX

- dizertace v Markdownu
- mezikrok k Overleafu: MUNI šablony na Overleafu
- první verze v markdownu, pak vyexportovat do TeXu
- případně Markdown v TeXu: https://www.overleaf.com/learn/latex/Articles/How_to_write_in_Markdown_on_Overleaf

- R Sweave: File -> New File -> R Sweave
- export do čistého TeXu
- propojení s Overleafem
- práce s verzemi v Overleafu
- článek z FANSB

Domácí úloha

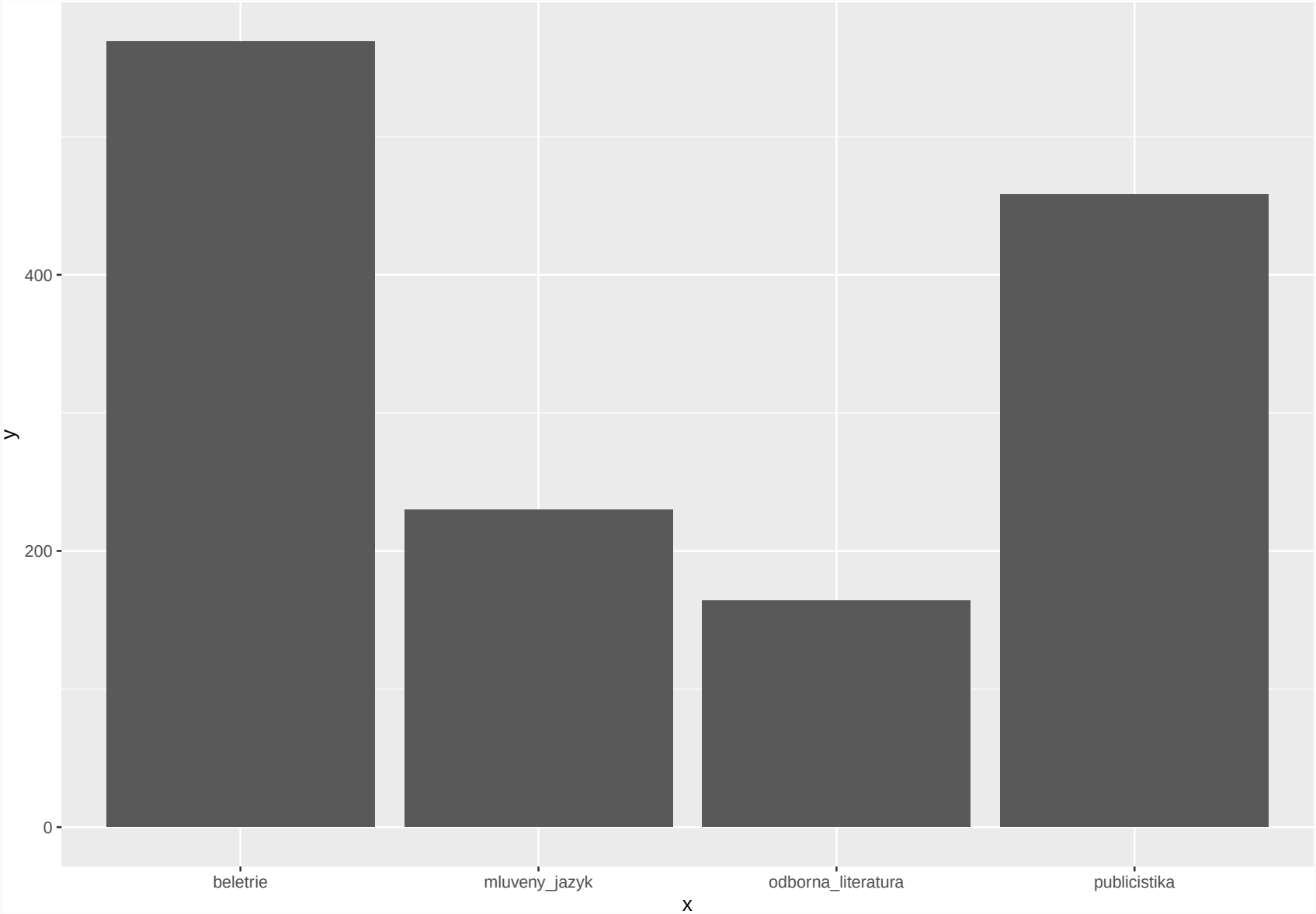
- v ČNK najít (např. ze jména) slovo: např. *čekat*:

čekat

- do dataframe hodit textové typy a ipm (zaokrouhlit):
- jednoduchý ggplot2 barplot graf
- zaslat výsledek reportu jako html
- obvyklé náležitosti: jméno, komentáře, ...

```r
x <- c("mluveny_jazyk","beletrie","publicistika","odborna_lit
y <- c(230, 569, 458, 164)

df <- data.frame(x,y)
```

```
library(ggplot2)

ggplot(aes(x = x, y = y), data = df) + geom_bar(stat="identit
```



References

R Core Team. 2016. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.