



# **Parę słów o Rrrr...**



**Andrzej Hirko**

# Co to jest ten R ?

- Interpretowany język programowania
- Środowisko programistyczne do obliczeń statystycznych i wizualizacji danych

Poszukujemy estymatorów rozkładu normalnego, funkcja gęstości prawdopodobieństwa:

$$f(x; m, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right).$$

Wyznaczamy logarytm w celu uproszczenia obliczeń:

$$\ln f(x; m, \sigma^2) = -\ln \sqrt{2\pi\sigma^2} - \frac{(x-m)^2}{2\sigma^2} = -\frac{1}{2} \ln 2\pi\sigma^2 - \frac{(x-m)^2}{2\sigma^2} = -\frac{1}{2} \ln 2\pi - \frac{1}{2} \ln \sigma^2 - \frac{(x-m)^2}{2\sigma^2}.$$

Funkcja wiarygodności  $L$ :

$$\ln L(x_1, \dots, x_N; m, \sigma^2) = \sum_{i=1}^N \ln f(x_i; m, \sigma^2).$$

Maksymalizujemy  $L$  ze względu na  $m$ :

$$0 = \frac{\partial \ln L}{\partial m} = \sum_{i=1}^N \frac{\partial \ln f(x_i; m, \sigma^2)}{\partial m} = \sum_{i=1}^N \left( \frac{x_i - m}{\sigma^2} \right) = \frac{1}{\sigma^2} \left( \left( \sum_{i=1}^N x_i \right) - Nm \right).$$

Co daje standardowy estymator średniej, jako estymator parametru  $m$ :

$$m = \frac{1}{N} \sum_{i=1}^N x_i.$$



# Informacje ogólne

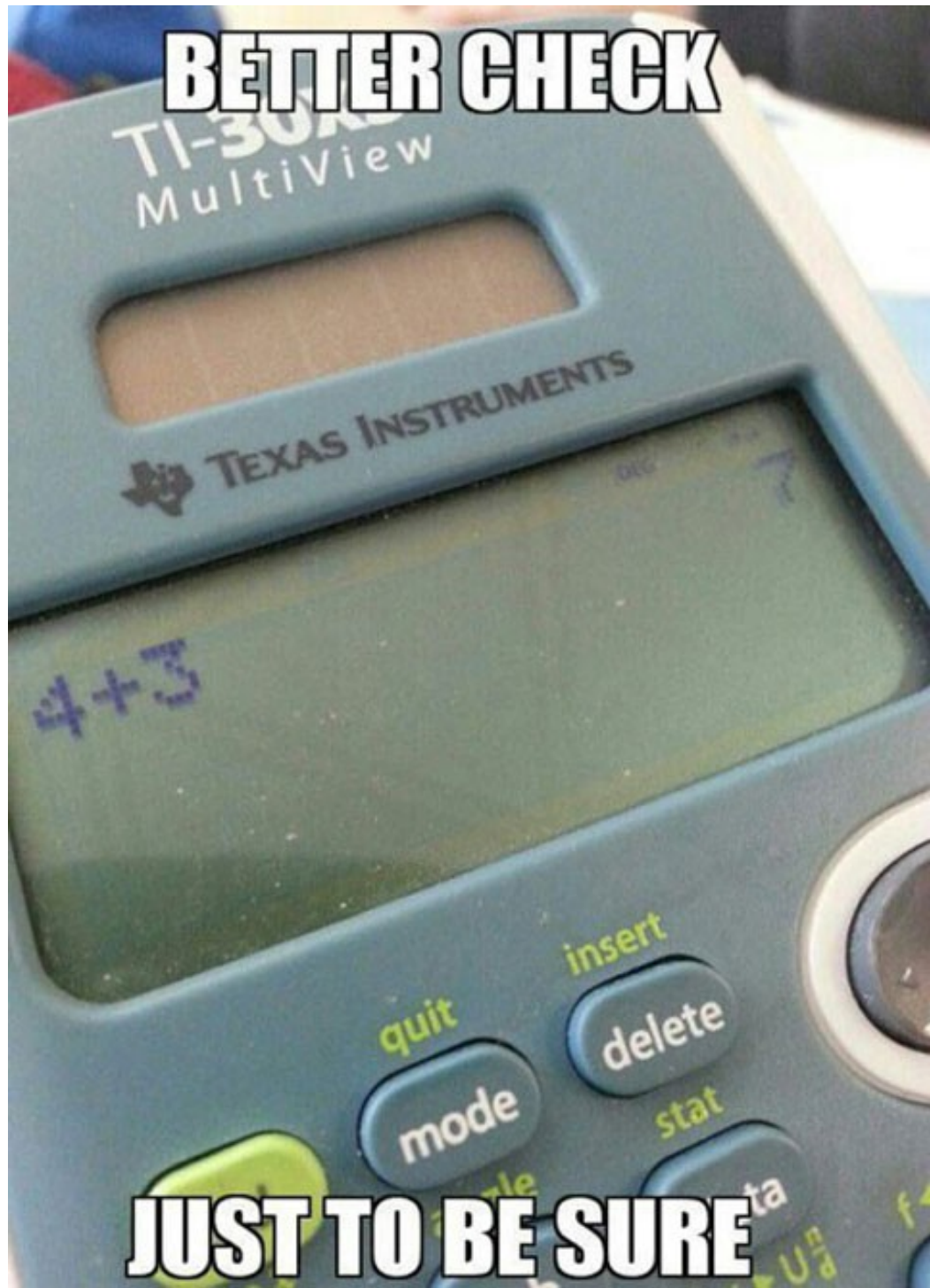
- Stworzony przez Ross Ihaka i Robert Gentleman, Uniwersytet Auckland, 1993
- Język programowania S
- R Development Core Team
- Projekt GNU napisany w C, Fortran
- Dostępny na: MS Windows, Mac OS., Linux
- Aktualna Wersja Stabilna: 3.2.3 / 10 grudnia 2015

# GUI

- Architect
- DataJoy
- Deducer
- Java GUI for R
- Number Analytics
- R Commander
- RGUI
- RKWard
- **RStudio**



**BETTER CHECK**



**JUST TO BE SURE**



# R jako kalkulator

```
> 2+2  
[1] 4
```

```
> sin(45)  
[1] 0.8509035
```

```
> exp(10)  
[1] 22026.47
```

```
> pi  
[1] 3.141593
```

```
> log(35)  
[1] 3.555348
```

# Atomowe typy danych

- Liczbowe
- Logiczne
- Znakowe
- Wartości brakujące i puste

# Liczbowe

```
> x<-55
```

```
> x
```

```
[1] 55
```

```
> x=55.10^40
```

```
> x
```

```
[1] 4.426536e+69
```

```
> y<-sqrt(-1+(0+0i))
```

```
> y
```

```
[1] 0+1i
```

# Logiczne

```
> -3+3==0
```

```
[1] TRUE
```

```
> 10!=5
```

```
[1] TRUE
```

```
> x<-4
```

```
> x > 1 & x < 2
```

```
[1] FALSE
```

# Znakowy

```
> x<-"To\n też jest napis"  
> x  
[1] 'To\n też jest napis'  
> cat(x)  
To  
  też jest napis  
  
> paste("Napis", "+", "napis")  
[1] "Napis + napis"  
  
> strsplit("To jest fajny napis", ' ')  
[[1]]  
[1] "To"      "jest"    "fajny"  "napis"
```

**SYNTAX ERRORS**

**SYNTAX ERRORS EVERYWHERE**

# Struktura danych

- Wektory
- Listy
- Macierze/ tablice wielowymiarowe
- Ramki



# Wektory

```
> d<-c(2, 3, 4, 5)
```

```
> d
```

```
[1] 2 3 4 5
```

```
> d*5
```

```
[1] 10 15 20 25
```

```
> d[4]
```

```
[1] 5
```

```
> d[d>=3]
```

```
[1] 3 4 5
```

```
> d==3
```

```
[1] FALSE TRUE FALSE FALSE
```

# Listy

```
> x <- list(a = 1, 2, d=c(T,F,T))
```

```
> x
```

```
$a
```

```
[1] 1
```

```
[[2]]
```

```
[1] 2
```

```
$d
```

```
[1] TRUE FALSE TRUE
```

```
> x$a
```

```
[1] 1
```

```
> x[[2]]
```

```
[1] 2
```

```
> x[[3]][2]
```

```
[1] FALSE
```

# Macierze

```
> mat<-matrix(1:8, 2, 4)
```

```
> mat
```

```
      [,1] [,2] [,3] [,4]  
[1,]    1    3    5    7  
[2,]    2    4    6    8
```

```
> mat[1,2]
```

```
[1] 3
```

# Ramki

```
> ramka = data.frame(  
  t = c("H", "W", "L"),  
  a = c(T, F, F),  
  n = c(450, 875, 500)  
)
```

```
> ramka  
  t      a      n  
1 H  TRUE  450  
2 W FALSE  875  
3 L FALSE  500
```

```
> ramka$n  
[1] 450 875 500
```



**I'm bored!**

# Instrukcje warunkowe

```
> liczba <- 11
> if (liczba %% 2 == 0) {
  print("ta liczba jest parzysta")
} else {
  print("ta liczba jest nieparzysta")
}
[1] "ta liczba jest nieparzysta"
```

# Pętla for

```
> for (i in 1:3) {  
  cat(paste("aktualna wartość i to ", i,  
"\n"))  
}
```

```
aktualna wartość i to 1
```

```
aktualna wartość i to 2
```

```
aktualna wartość i to 3
```



# Funkcje

```
x <- 10  
f1 <- function(x) {  
  return(x+10)  
}
```

```
> f1(30)  
[1] 40
```

```
f2 <- function() {  
  x+20  
}
```

```
> f2()  
[1] 30
```

**CHALLENGE ACCEPTED**



**LET'S CODE IT**

# Podsumowanie

- Zalety:
  - Darmowy (GNU GPL)
  - Zewnętrzna biblioteka
  - Ponad 4000 zewnętrznych pakietów
  - Wykresy wysokiej jakości
  - Dokumentacja
- Wady:
  - Nie można stworzyć bezpośrednio .exe lub .dll

