

Raport

Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet

```
"1"
```

```
## [1] "1"
```

```
"2"
```

```
## [1] "2"
```

```
"3"
```

```
## [1] "3"
```

```
2 + 2
```

```
## [1] 4
```

```
x <- sum(1:10) # wyznacz sume 10 pierwszych liczb calkowitych  
print(x)
```

```
## [1] 55
```

Sam output, bez polecen: echo=FALSE

```
## Warning: package 'stringi' was built under R version 3.0.1
```

```
## stringi (pl_PL.windows-1250)
```

```
## [1] 26
```

Bez pierwszego polecenia: echo=-1

```
stri_length("ja mam psa, a ty nie masz!")
```

```
## [1] 26
```

Kilka wybranych:

```
1
```

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

2

```
## [1] 2
```

```
## [1] 3
```

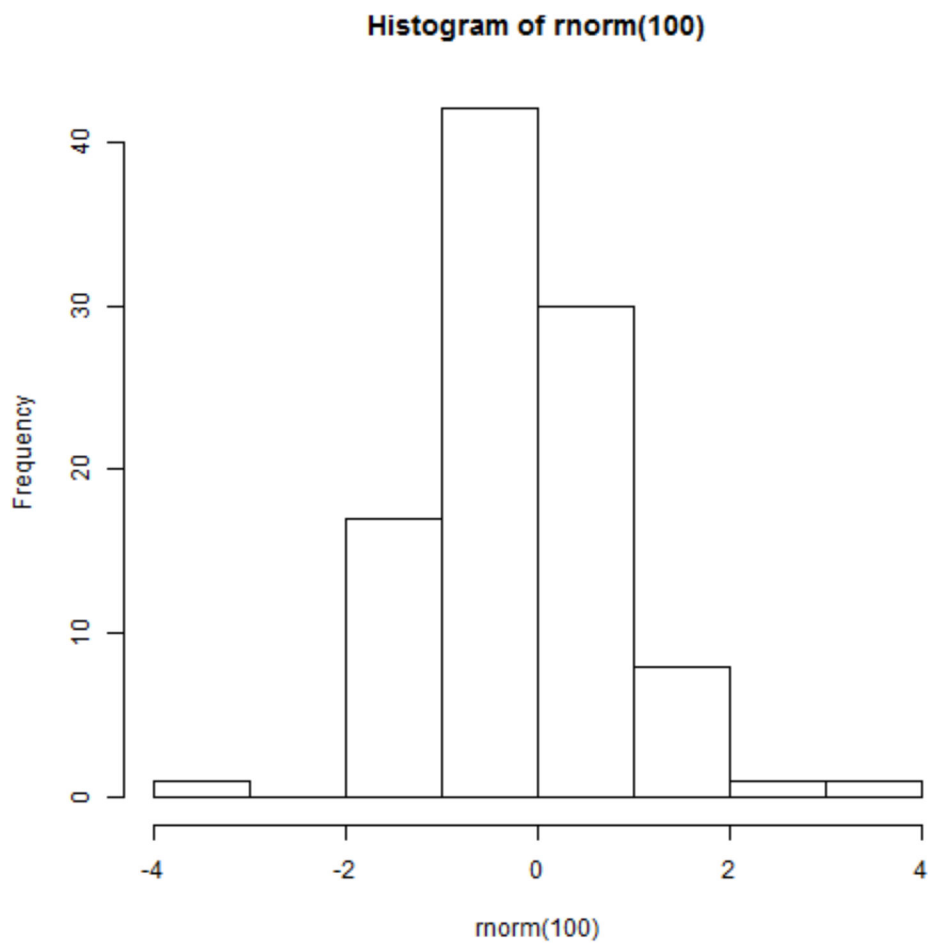
4

```
## [1] 4
```

```
## [1] 5
```

Obrazek tez sie wyswietli:

```
hist(rnorm(100))
```



Ukryje nam output:

```
print("aaa")
```

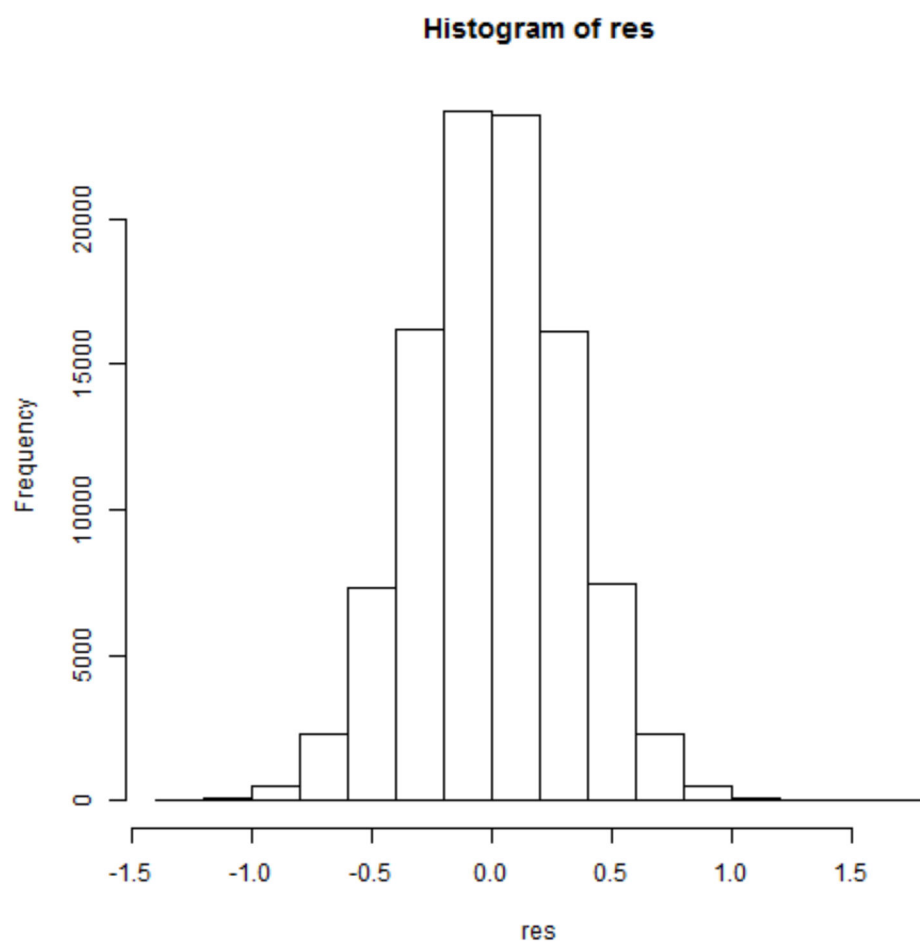
asis → potraktuje to jak htmla i robi ladna liste

- 1
- 2

3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10

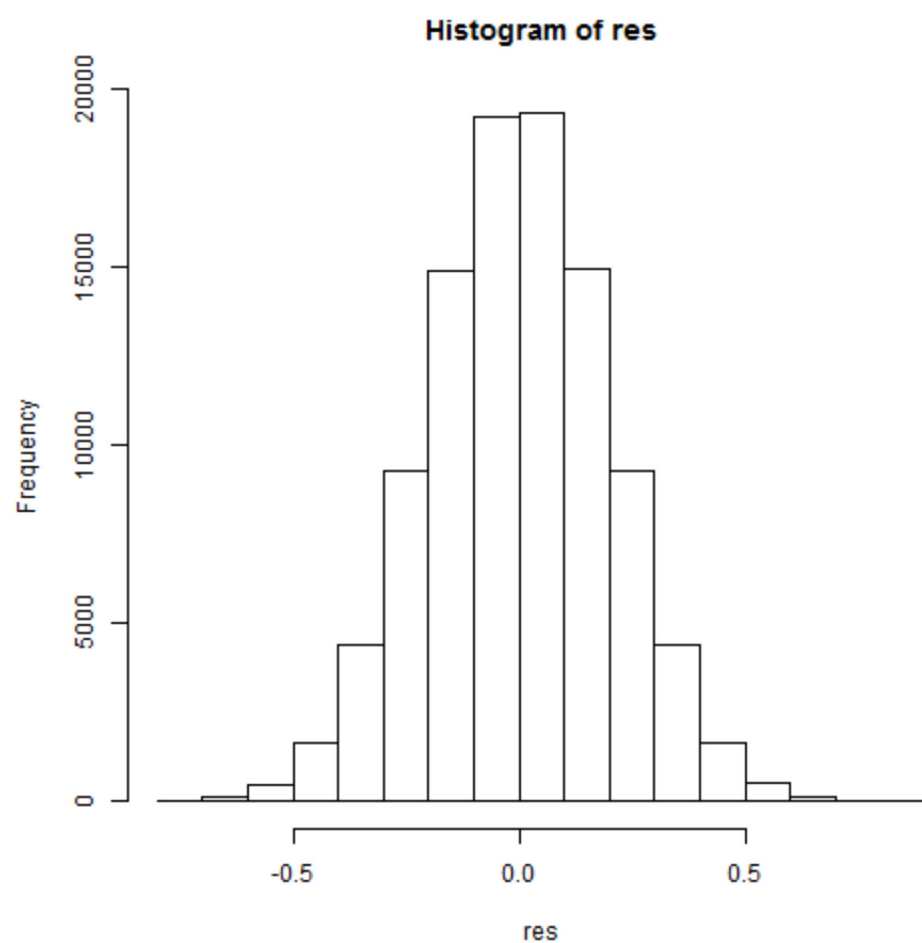
Warto włączyc czasem pamięć podręczna, żeby raz policzone wyniki zostały zachowane (cache=TRUE) -> za drugim razem otwiera się już dużo szybciej:

n= 10



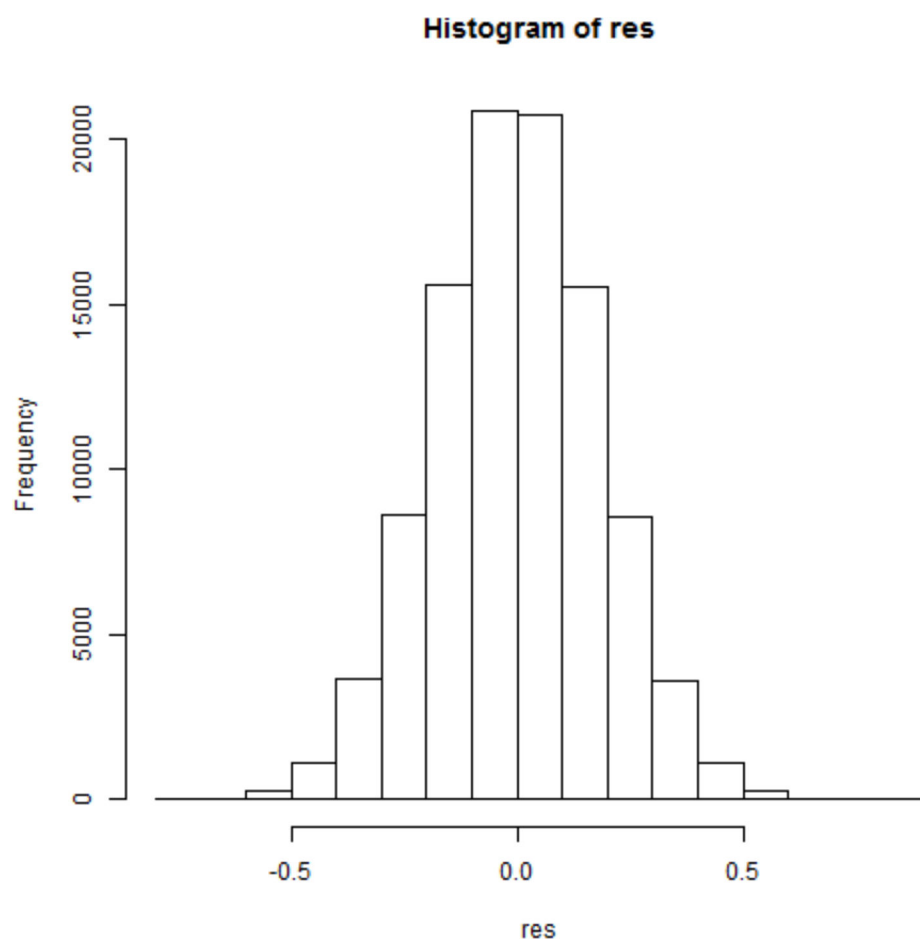
Srednia to: 0.0002771

n= 25



Srednia to: 0.0002024

n= 30



Srednia to: -0.000325

Wstawki nazwane: - eval i echo oznacza, ze nie bedzie sie wyswietlalo nic - ani output ani kod

```
print("ala ma kota")
```

```
## [1] "ala ma kota"
```

```
print("ala ma kota")
```

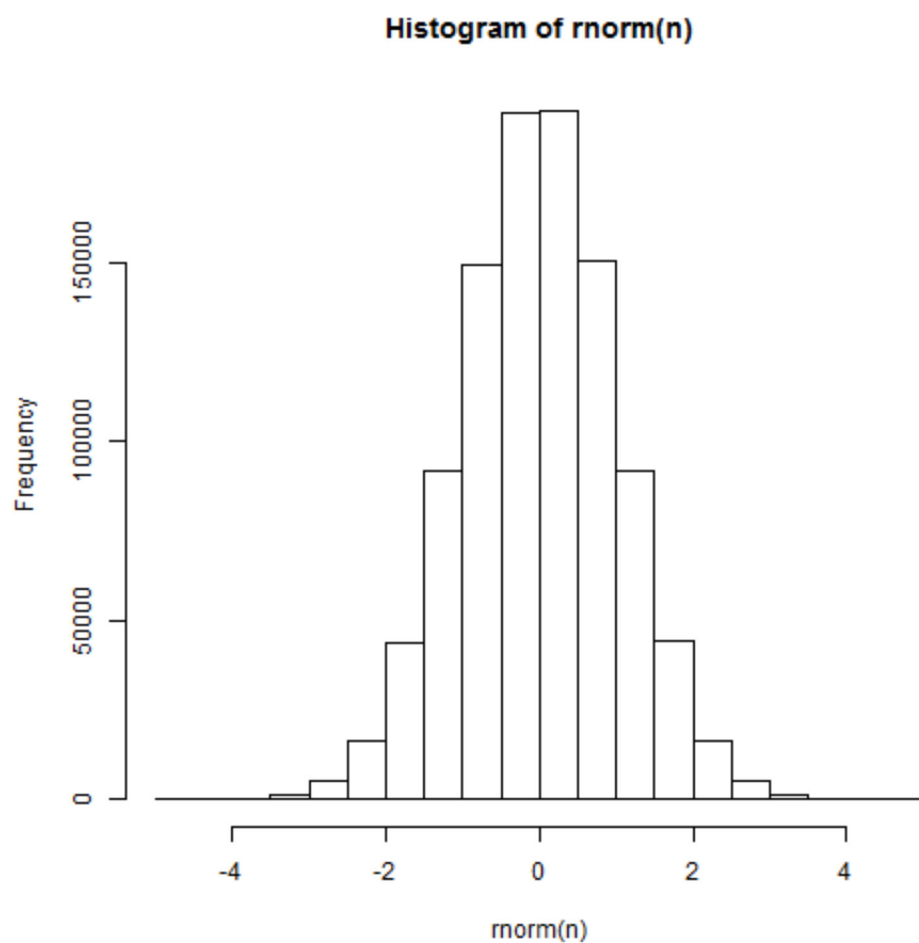
```
## [1] "ala ma kota"
```

```
print("ala ma kota")
```

```
## [1] "ala ma kota"
```

```
n <- 1e+06
```

```
hist(rnorm(n))
```



```
sum(gsdgsd)  # takiej zmiennej nie ma
```

```
## Error: nie znaleziono obiektu 'gsdgsd'
```

```
log(-1)
```

```
## Warning: wyprodukowano wartości NaN
```

```
## [1] NaN
```

Można powstrzymać warningi:

```
sum(gsdgsd)  # takiej zmiennej nie ma
```

```
## Error: nie znaleziono obiektu 'gsdgsd'
```

```
log(-1)
```

```
## [1] NaN
```

```
for(i in 1: 10) print(  
  i)
```

```
## [1] 1  
## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5  
## [1] 6  
## [1] 7  
## [1] 8  
## [1] 9  
## [1] 10
```

```
for (i in 1:10) print(i)
```

```
## [1] 1  
## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5  
## [1] 6  
## [1] 7  
## [1] 8  
## [1] 9  
## [1] 10
```