

An introduction to WS 2013/2014

Dr. Noémie Becker (AG Metzler)
Dr. Sonja Grath (AG Parsch)

Special thanks to: Dr. Martin Hutzenthaler
(previously AG Metzler, now University of Frankfurt)
course development, lecture notes, exercises

Course outline – Day 4

Review session

Preparing a script in R

Basic statistics

Plotting

High- and low-level plotting functions and arguments

Mathematical symbols

Interacting with plots

Saving plots

Questions

Solutions to exercise sheet 2

Review Session

Some distributions implemented in R

For each distribution:

dxxx: density of the xxx distribution

pxxx: distribution function of the xxx distribution ('p' for probability)

qxxx: quantile function of the xxx distribution

rxxx: random number generator for the xxx distribution

Example: Normal distribution

`dnorm(x, mean = μ , sd = ρ)`

Standard normal distribution:

mean 0, standard deviation 1

```
>dnorm(x, mean = 0, sd = 1)
```

```
>dnorm(x)
```

Example: Normal distribution

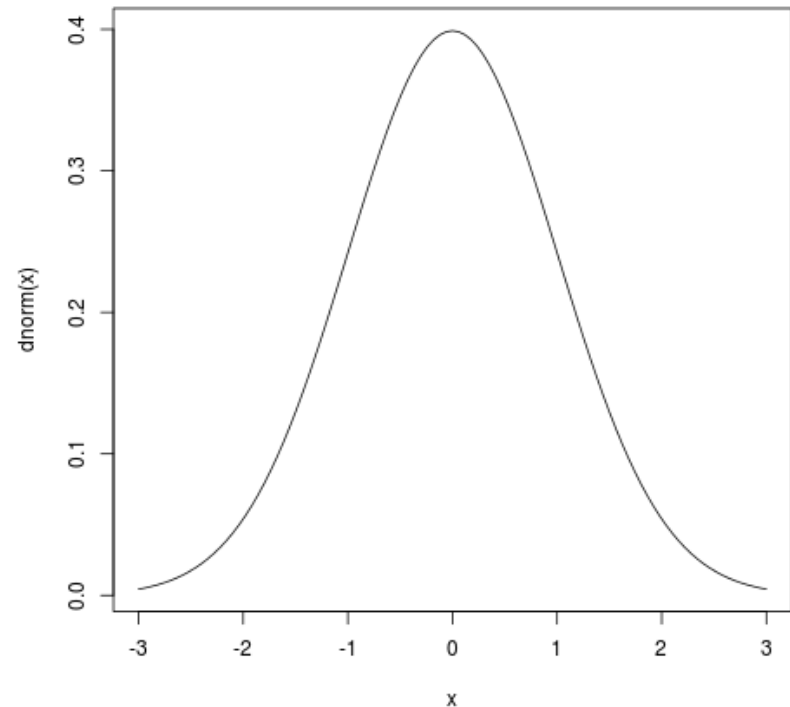
Recall:

`plot()`

`plot(fun)`

→ If *fun* is a function, then `plot(fun, from=a, to=b)` plots *fun* in the range [a, b]

```
>plot(dnorm, from = -3, to = 3)
```



Important functions

Imagine you have a vector v:

```
> v <- c(1:4)
```

```
> v
```

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

```
> mean(v)
```

```
[1] 2.5
```

```
> var(v)
```

```
[1] 1.666667
```

```
> sd(v)
```

```
[1] 1.290994
```

```
> median(v)
```

```
[1] 2.5
```

Important functions

Imagine you have a vector v:

```
> v <- c(1:4)
```

```
> v
```

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

```
> quantile(v)
```

```
0%  25%  50%  75% 100%
```

```
1.00 1.75 2.50 3.25 4.00
```

```
> summary(v)
```

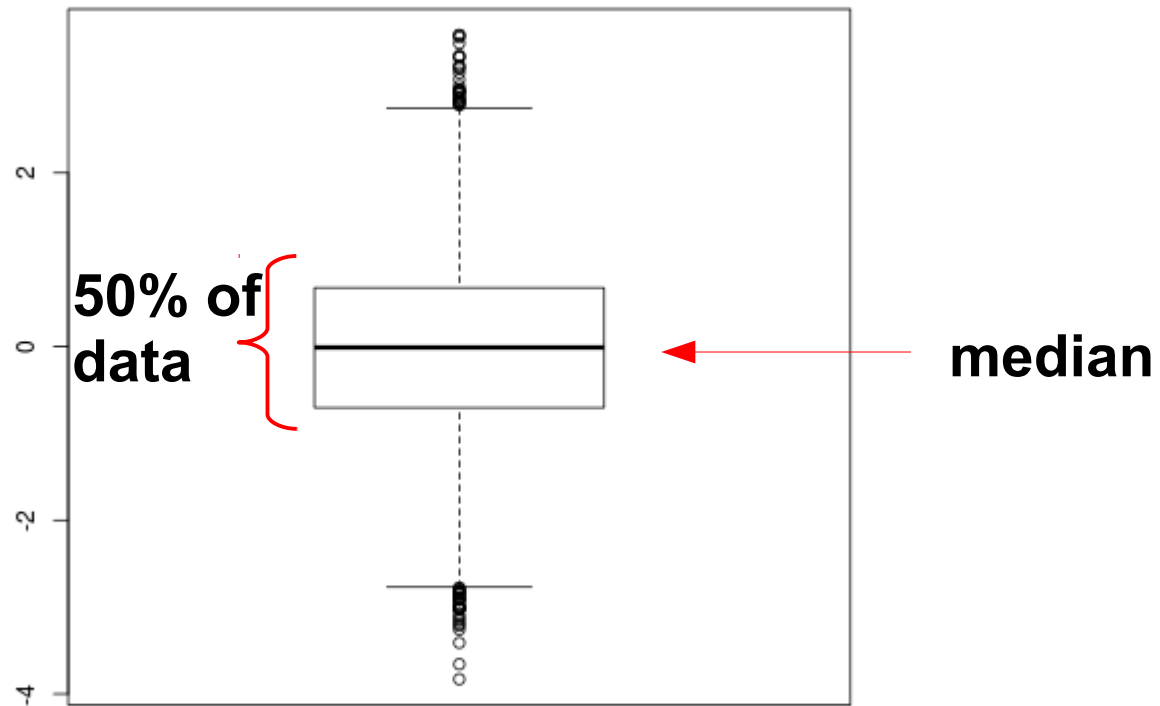
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.00	1.75	2.50	2.50	3.25	4.00

Box- and whisker plot (boxplot)

Get 10000 normally distributed values:

```
> rnorm(10000)
```

```
> boxplot(rnorm(10000))
```



Plotting

Plotting

There are three types of plotting commands:

High-level plotting functions create a new plot (usually with axes, labels, titles and so on)

Low-level plotting functions add more information to an existing plot, such as extra points, lines or labels

Interactive graphics functions allow you to interactively add information to an existing plot or to extract information from an existing plot using the mouse

High-level plotting functions

Function	Description
barplot()	Visualizes a vector with bars
boxplot()	Box- and whisker plot
contour()	The contour of a surface is plotted in 2D
coplot()	Conditioning-Plots
hist()	Histogram
mosaicplot()	Plot in form of a mosaic
pairs()	Produces a matrix of scatterplots
pie()	Circular pie charts
qqplot()	Quantile-quantile plot
...	

... many more – and R offers many packages for plotting (ggplot, lattice...)

We will cover now: plot(), hist(), boxplot()

High-level function – plot()

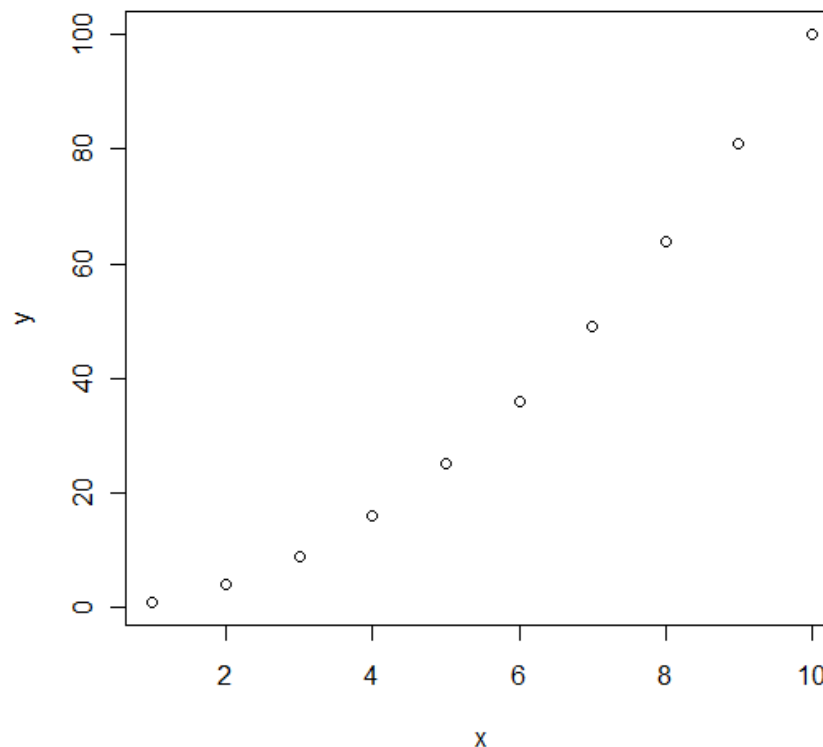
- Standard high-level plotting function
- Behaviour of plot() depends on the type of its argument

plot(x,y)

If x and y are numerical vectors, then plot(x,y) produces a scatterplot of y against x

Example:

```
x <- c(1:10)
y <- x^2
plot(x, y)
```



High-level function – plot()

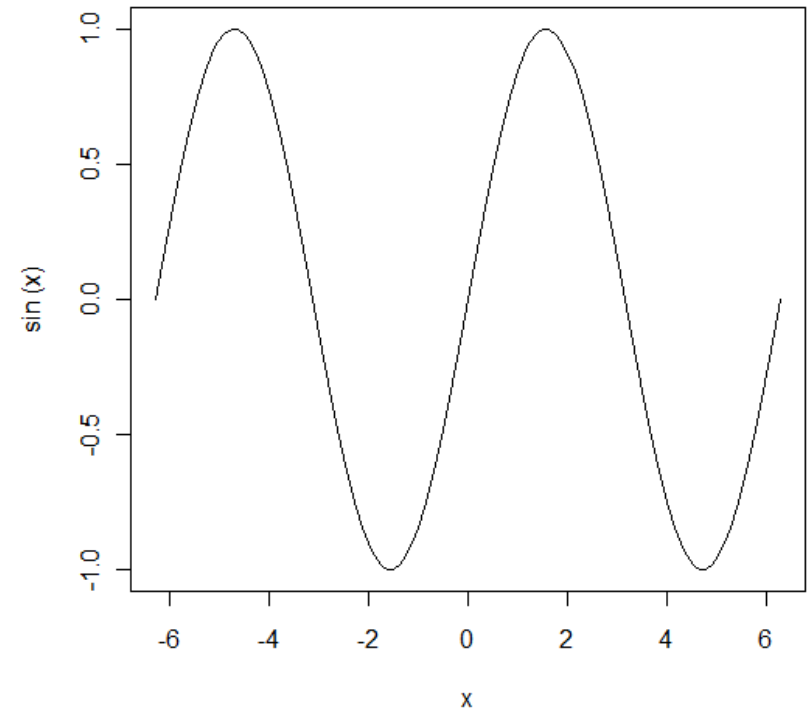
- Standard high-level plotting function
- Behaviour of plot() depends on the type of its argument

plot(*fun*)

If *fun* is a function, then `plot(fun, from=a, to=b)` plots *fun* in the range `[a, b]`

Example 1:

```
plot(sin, from=-2*pi, to=2*pi)
```



High-level function – plot()

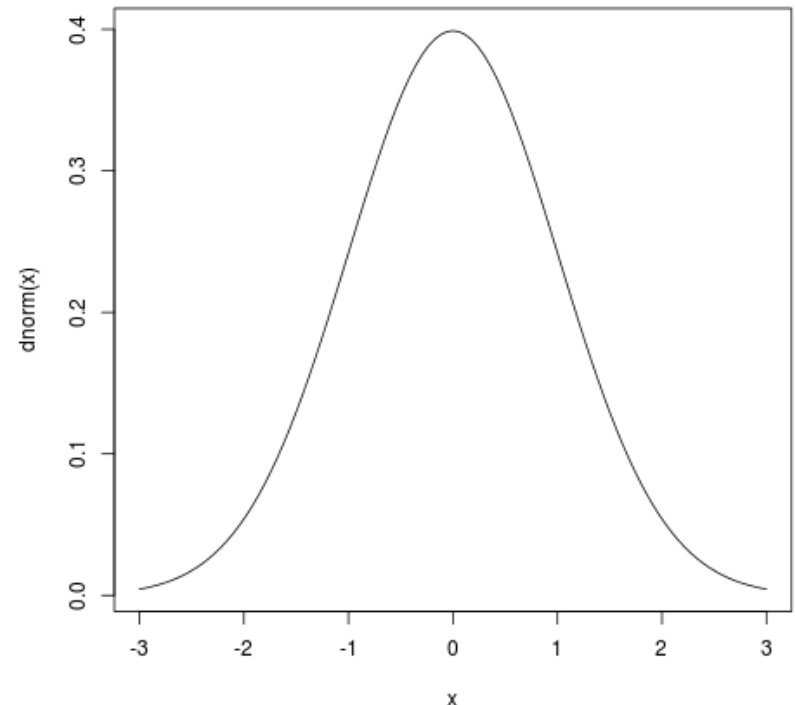
- Standard high-level plotting function
- Behaviour of plot() depends on the type of its argument

plot(*fun*)

If *fun* is a function, then plot(*fun*, from=*a*, to=*b*) plots *fun* in the range [*a*, *b*]

Example 2:

```
plot(dnorm, from = -3, to = 3)
```

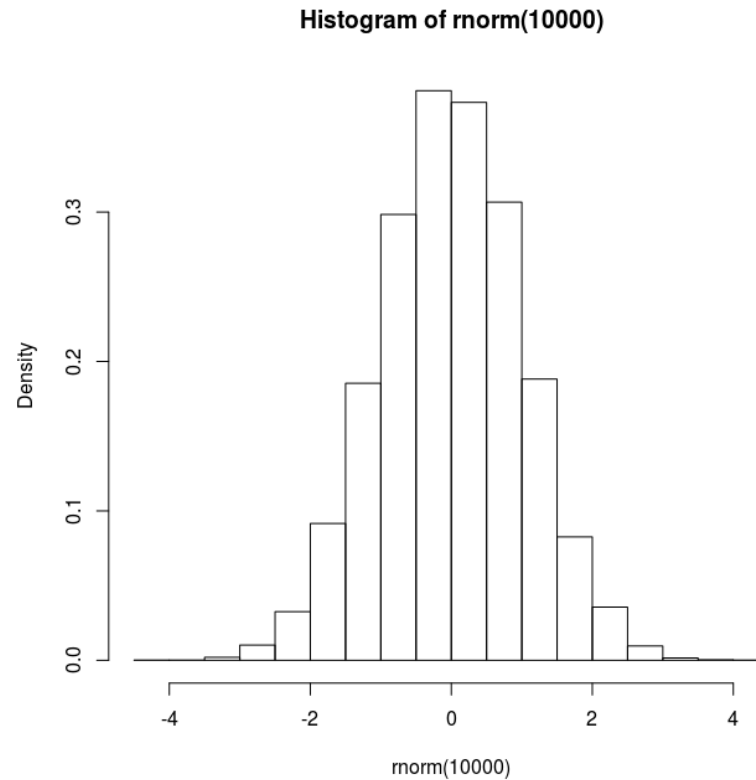


High-level function – hist()

→ Histogram

Example 1:

```
hist(rnorm(10000), probability = TRUE)
```

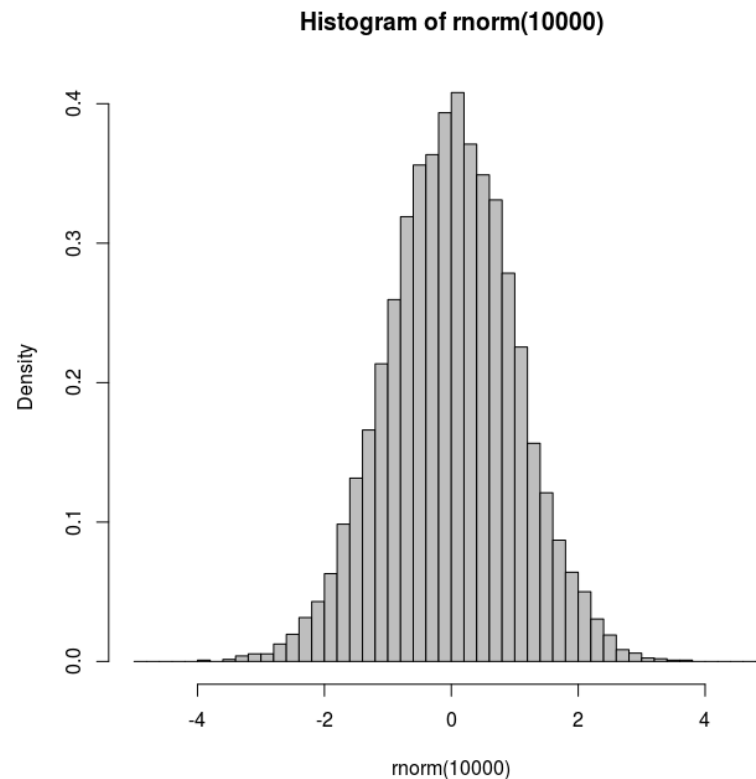


High-level function – hist()

→ Histogram

Example 2:

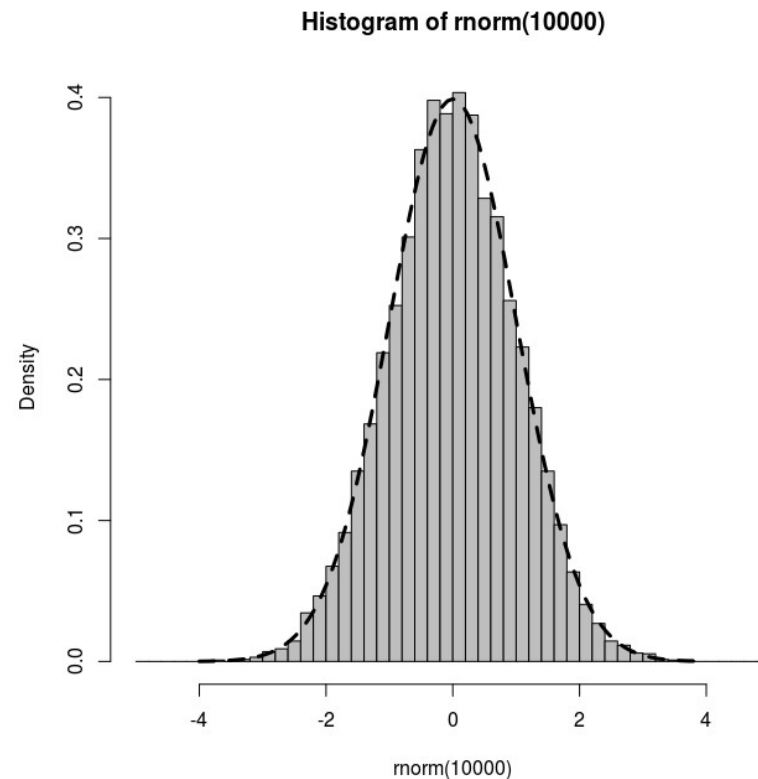
```
hist(rnorm(10000), probability=TRUE, col="grey",  
breaks=seq(-5, 5, by=0.2))
```



The histogram of 10000 simulated values is close to the density function

Example:

```
hist(rnorm(10000), probability=TRUE, col="grey",  
breaks=seq(-5, 5, by=0.2))  
plot(dnorm, from=-4, to=4, add=TRUE, lwd=3,  
lty="dashed")
```



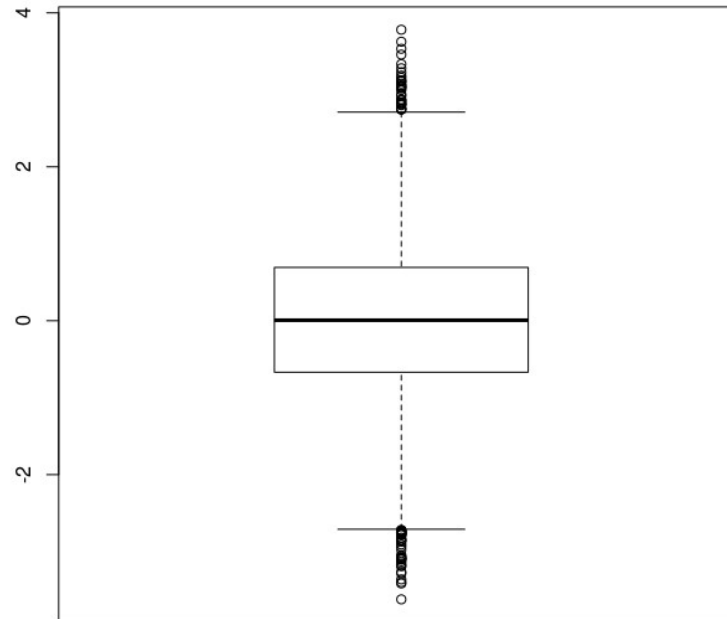
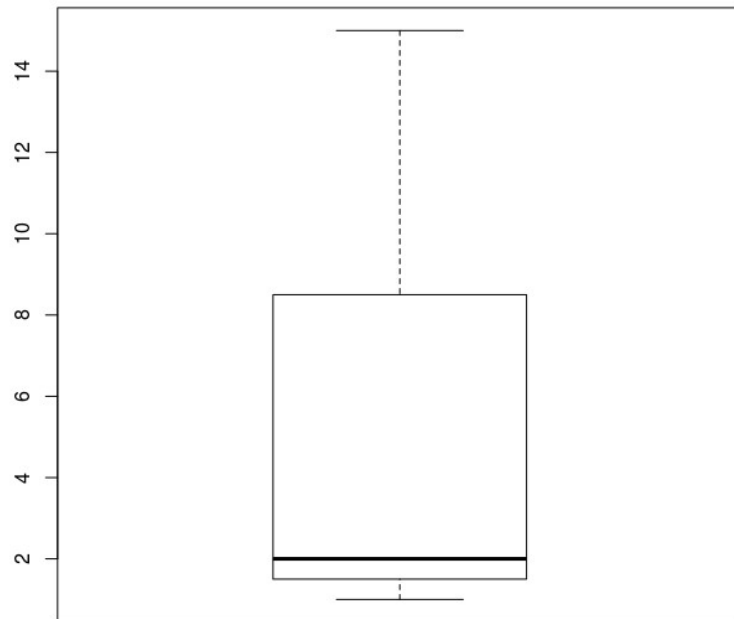
High-level function – boxplot()

→ Box and whisker plot

Example:

```
boxplot(c(1, 2, 15))
```

```
boxplot(rnorm(10000))
```



Saving plots

→ Several possibilities

(1) savePlot

Usage:

```
savePlot(filename = paste("Rplot", type, sep="."),  
         type = c("png", "jpeg", "tiff", "bmp"),  
         device = dev.cur())
```

Example:

```
savePlot(filename="Figure1.png", type="png")
```

(2) dev.print()

Example:

```
plot(...)  
#After you are finished with the plot:  
dev.print(device=pdf, file="filename.pdf")
```

(3) Plot directly into a file

Example:

```
pdf("filename.pdf")
```

When you are done you have to close the printing device!

```
dev.off()
```