animatoR

dynamic graphics in R

A. Blejec
andrej.blejec@nib.si

National Institute of Biology
Ljubljana, Slovenia

useR! 2011

Outline

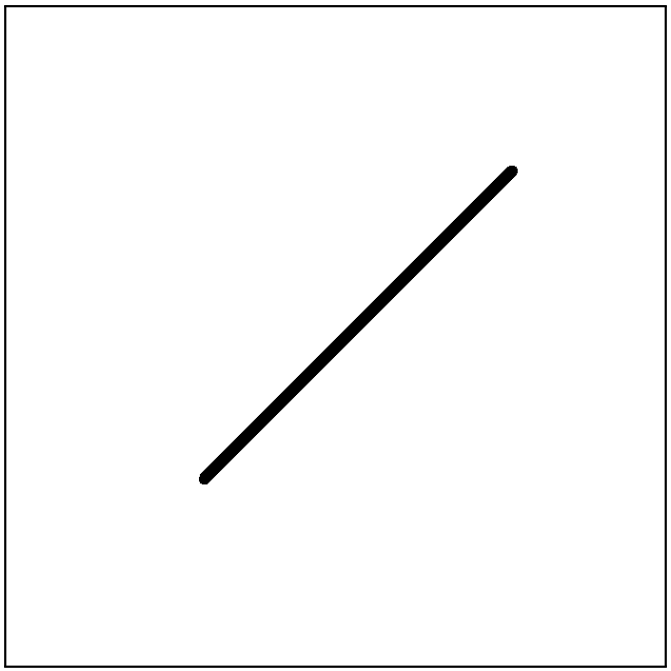
Animation solutions for R :

- the main idea is: plot frame by frame
- package **animation** provides an organized environment for subsequent plotting, see AniWiki

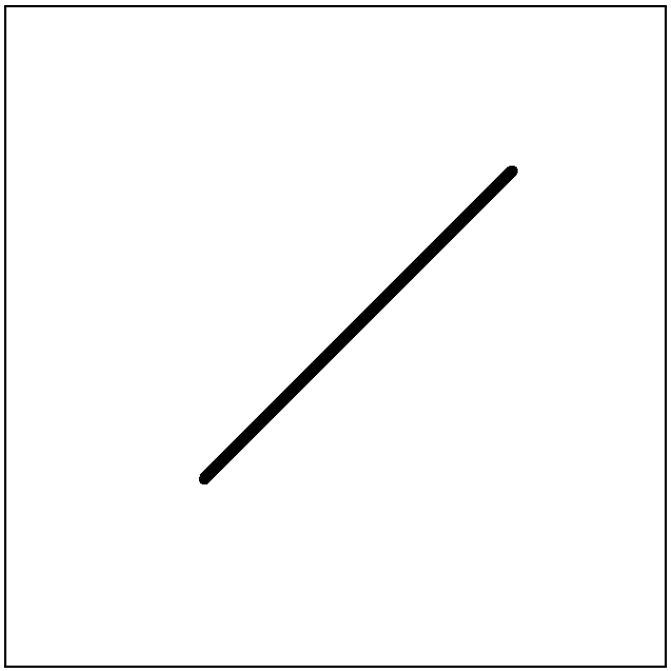
Drawbacks

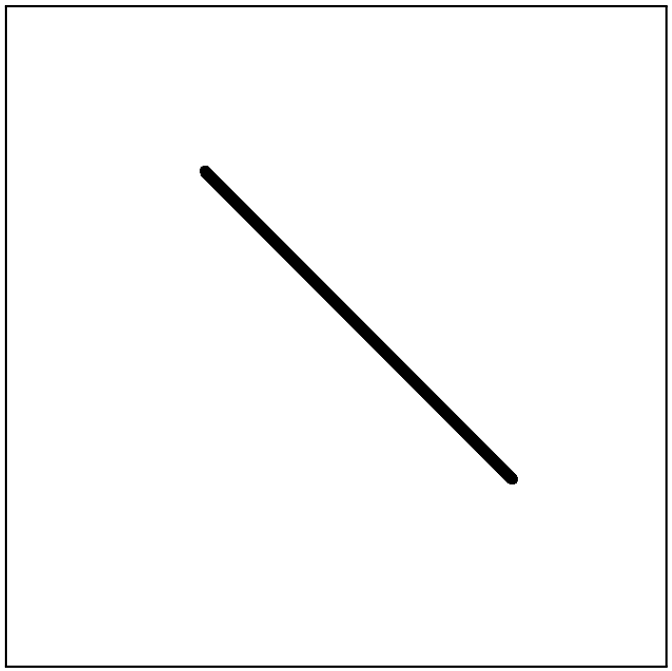
- plotting everything can be slow for complex figures
- smooth movements require intermediate plots, thus lot of frames to plot

XOR: 1+1=0

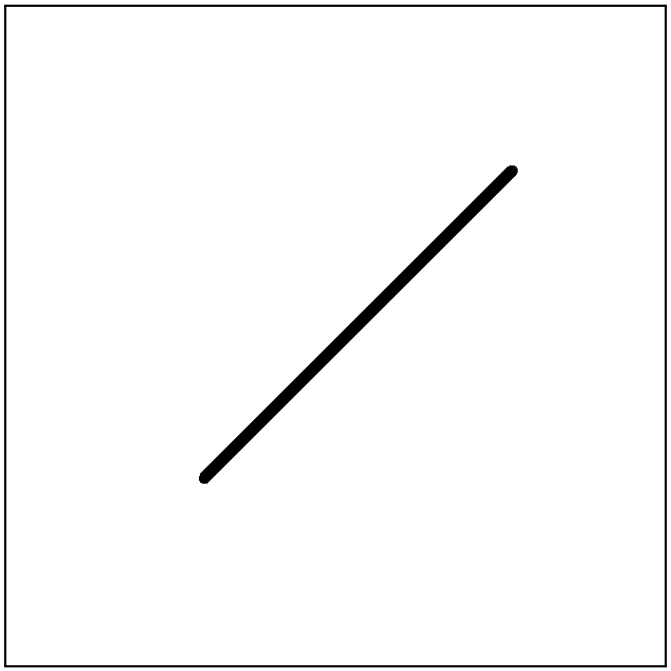


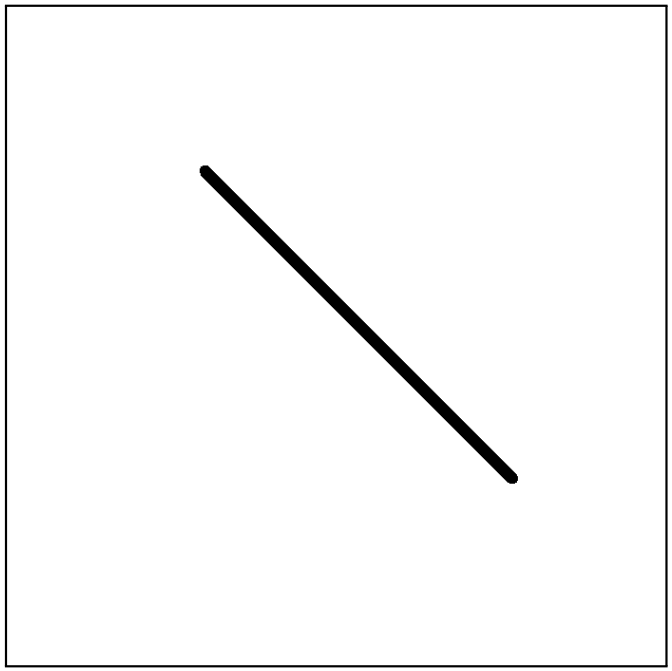
XOR: 1+1=0

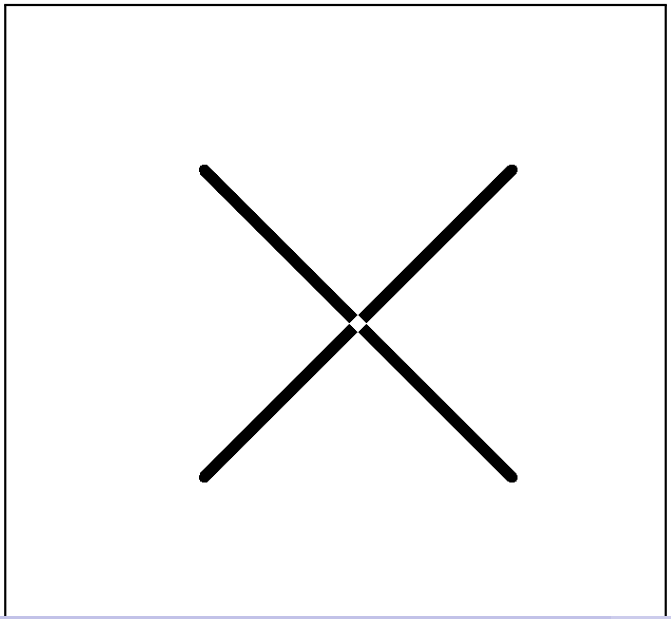




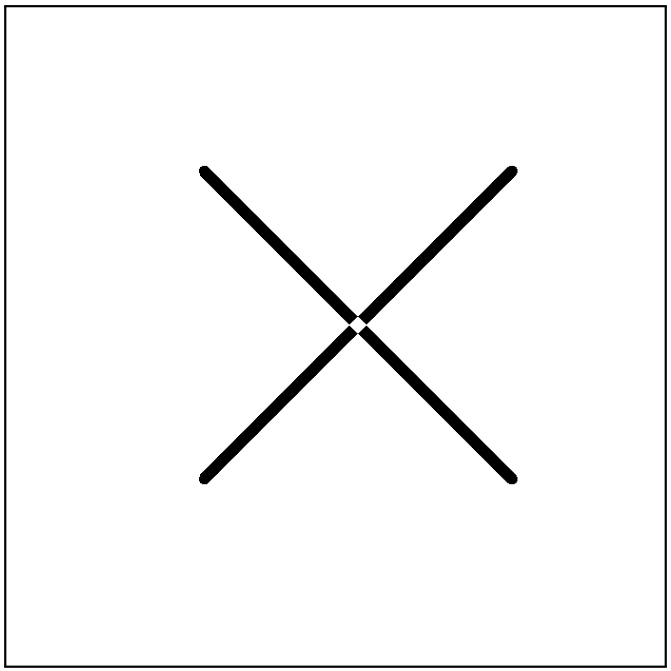
XOR: 1+1=0



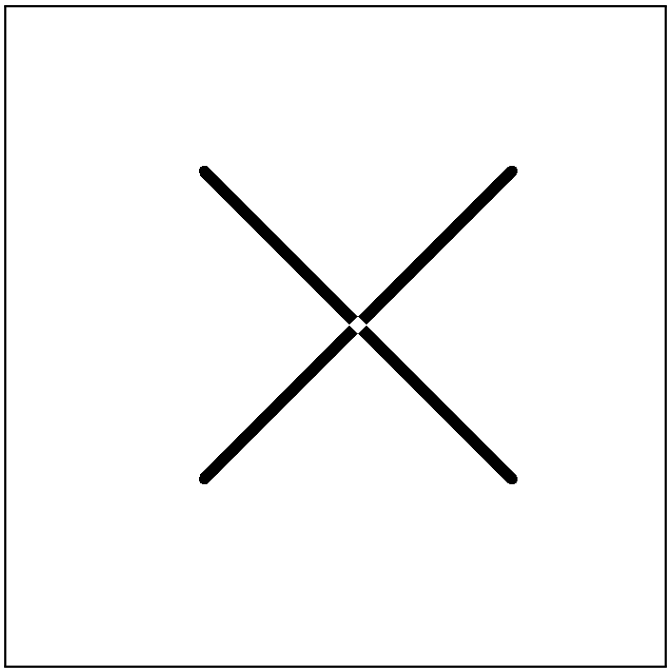


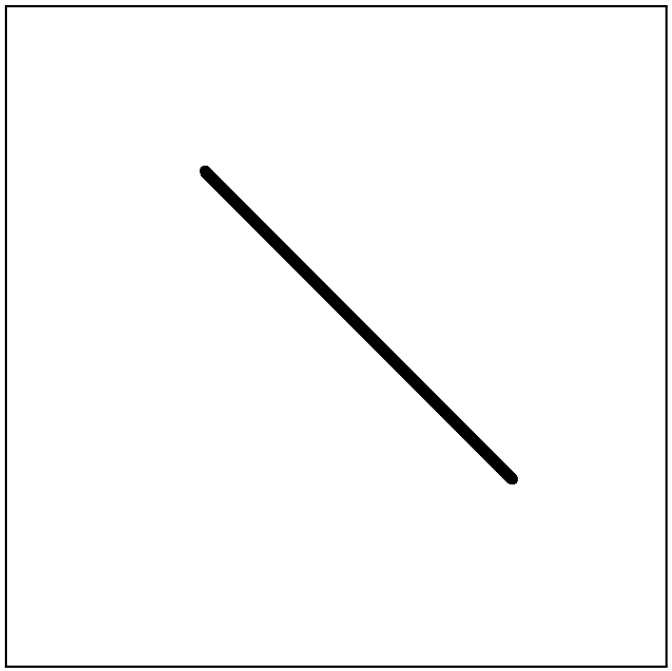


XOR: 1+1=0

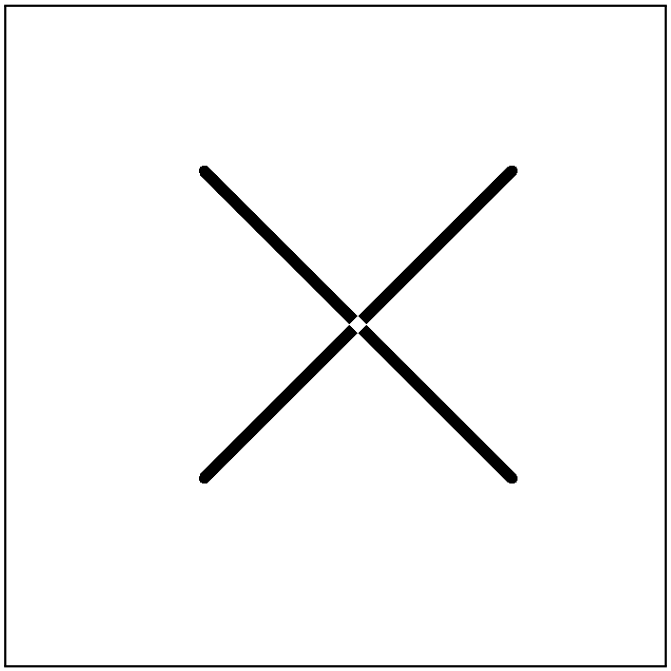


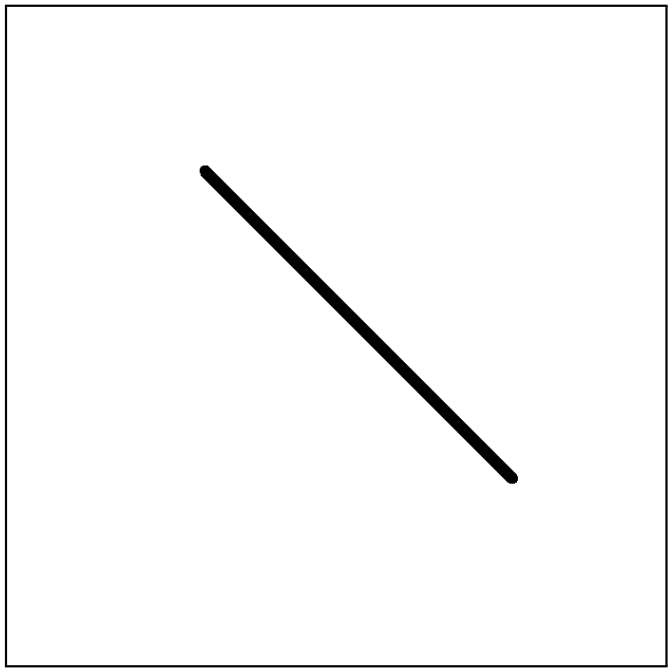
XOR: 1+1=0

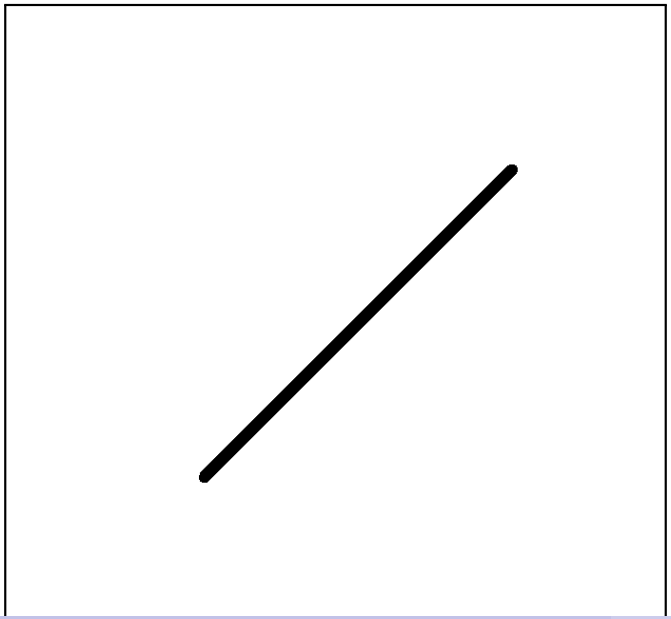




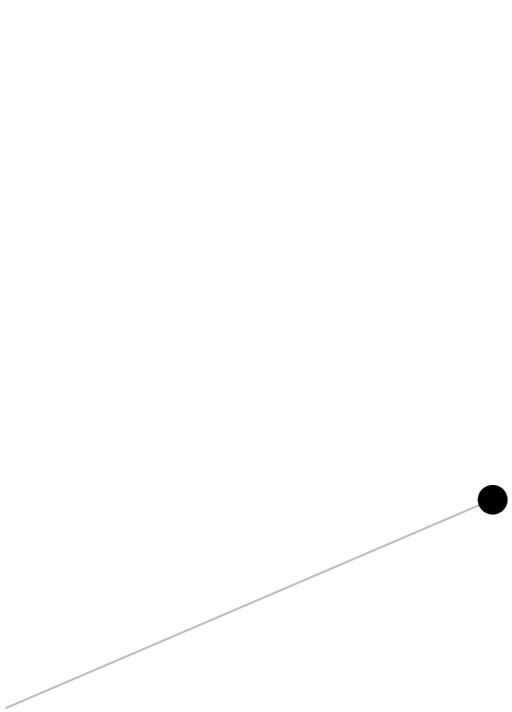
XOR: 1+1=0



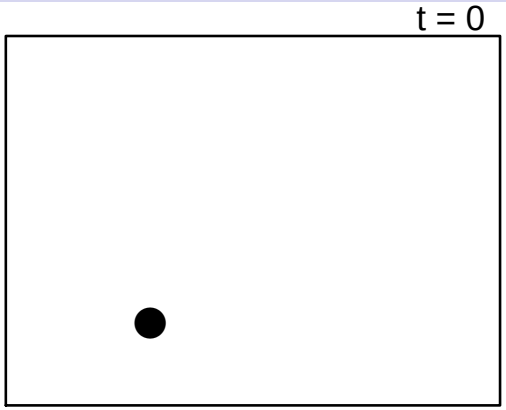


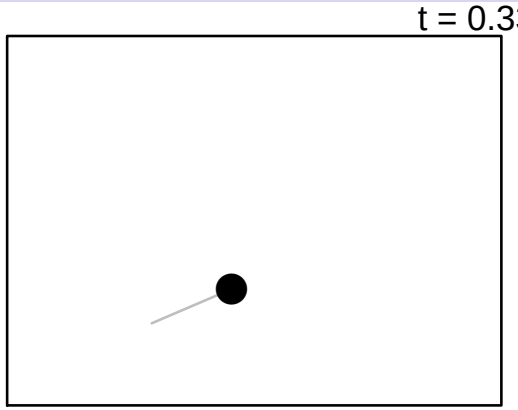


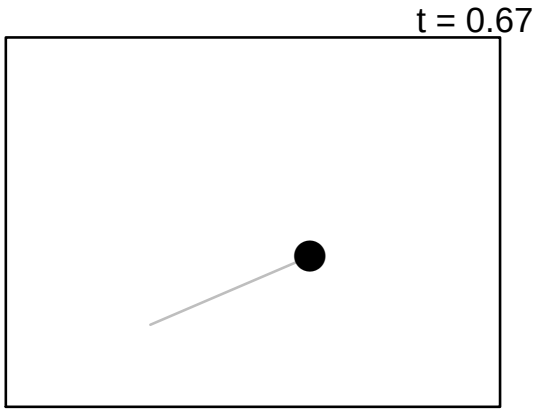
Moving a point, line traces the movement.

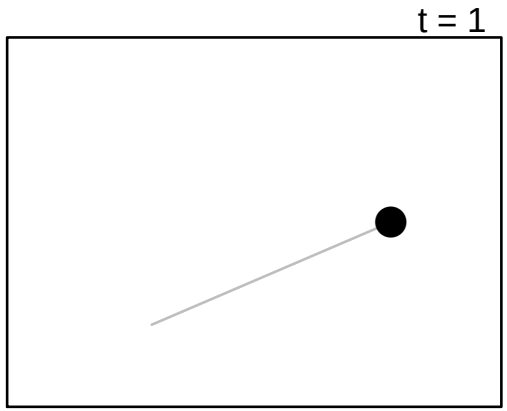


Moving a point ...









Linear homotopy

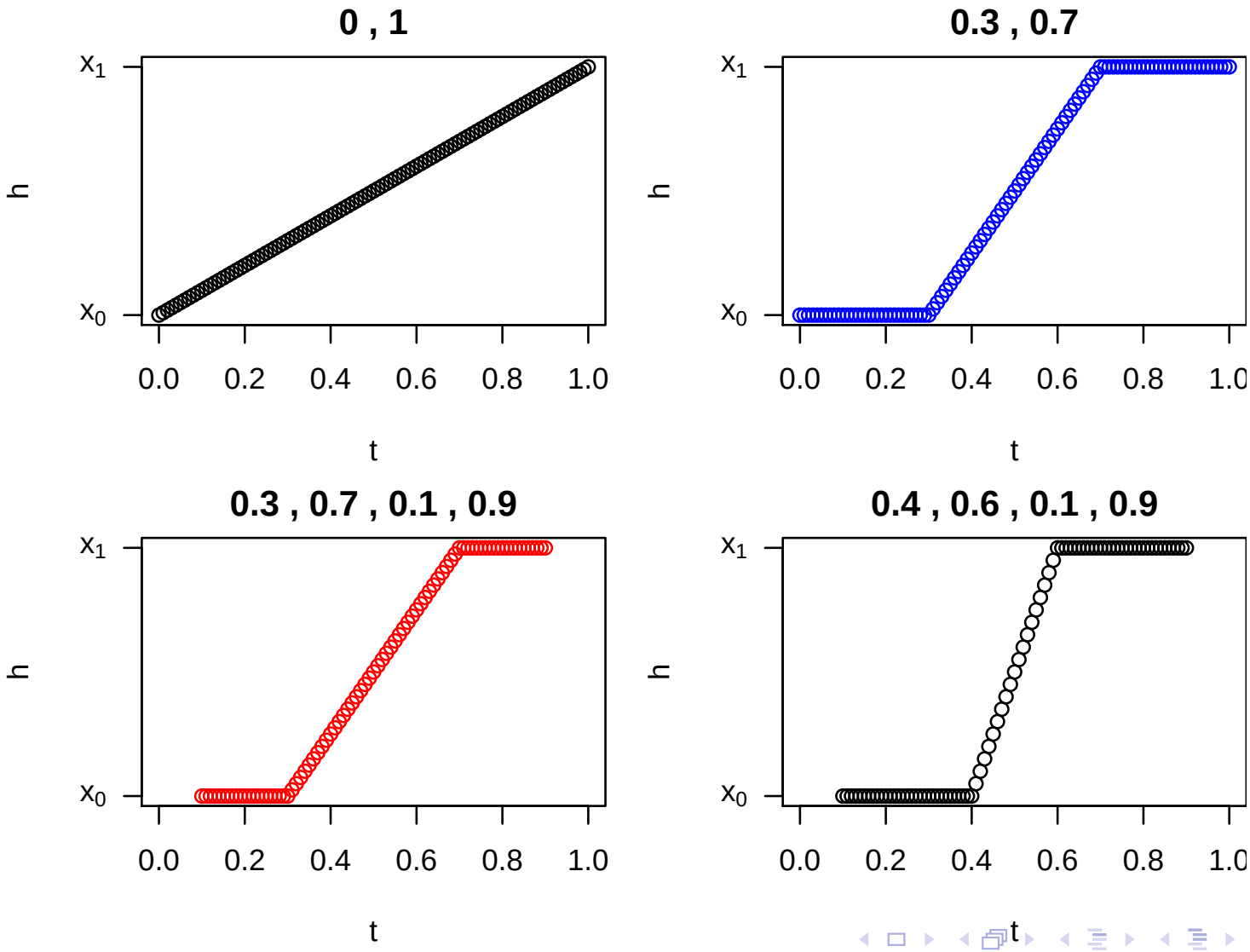
Start coordinate x_0
End coordinate x_1

For parameter $t \in [0, 1]$

$$x_t = h(x_0, x_1, t) = (1 - t) \cdot x_0 + t \cdot x_1$$

Linear homotopy

`when:  start, end, enter, exit ...  life`



Moving points
tpoints()

```
> for (t in seq(0, 1, length = 100)) {
+     newplot(stamp = TRUE)
+     tpoints(2, 2, 9, 5, t, pch = 16, cex = 2,
+         trace = TRUE)
+ }
```

Timed functions

```
tpoints    (x0, y0, x1 = x0, y1 = y0, t, when, trace = FALSE
tlines    (x0, y0, x1, y1, t, when, ...)
tsegments    (x0, y0, x1, y1, t, when, fixed = 1, ...)
tarrows    (x0, y0, x1, y1, t, when, fixed = 1, length = 0.1
tpolygon    (x0, y0, x1, y1, t, when, ...)
trect    (xleft0, ybottom0, xright0, ytop0, xleft1, ybottom1
ttext    (x0, y0, x1 = x0, y1 = y0, t, when, text = "", ...)
```

Timed parameters ...

```
tcex   (cex0 = 1, cex1 = 1, t, when, ...)

trgb   (x0, x1 = x0, t, when = c(0, 1), alpha0 = 1, alpha1

tmatrix   (X1, X0 = diag(nrow(X1)), t, when, ...)
```

Function animator()

```
animator   (block, life = 1, fps = 25, pause = 0.5, verbose
```

Accepts argument block as a character string

```
animator('
    newplot()
    tpoints(2,2,9,5,trace=TRUE,pch=16,cex=2)
')
```

Function animator()

Argument block as an expression

```
> block <- expression({
+     newplot()
+     tpoints(2, 2, 9, 5, trace = TRUE, pch = 16,
+         cex = 2)
+ })
> animator(block)
```

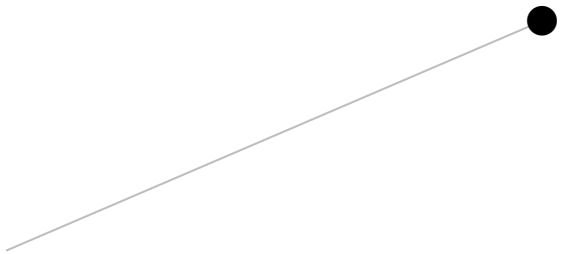
Class animator

A block of commands (expression or character version) can be changed to a class animator

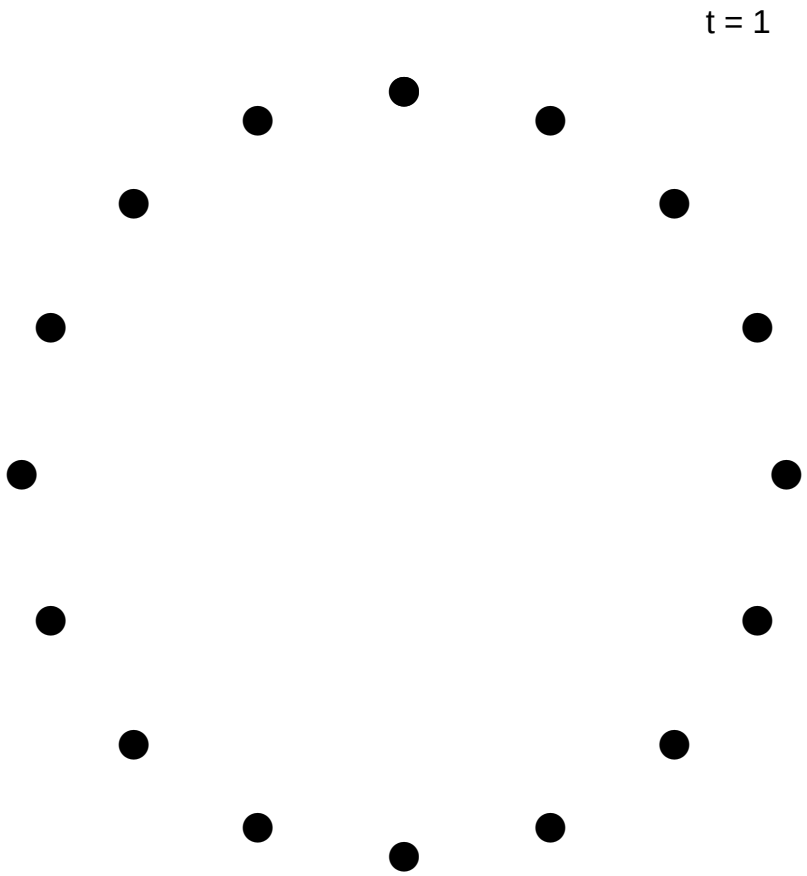
```
> x <- as.animator(block, life = 2)
> x

expression({
    newplot()
    tpoints(2, 2, 9, 5, trace = TRUE, pch = 16, cex = 2)
})
attr(,"class")
[1] "animator"
attr(,"life")
[1] 2

> is.animator(x)

[1] TRUE

> plot(x)
```

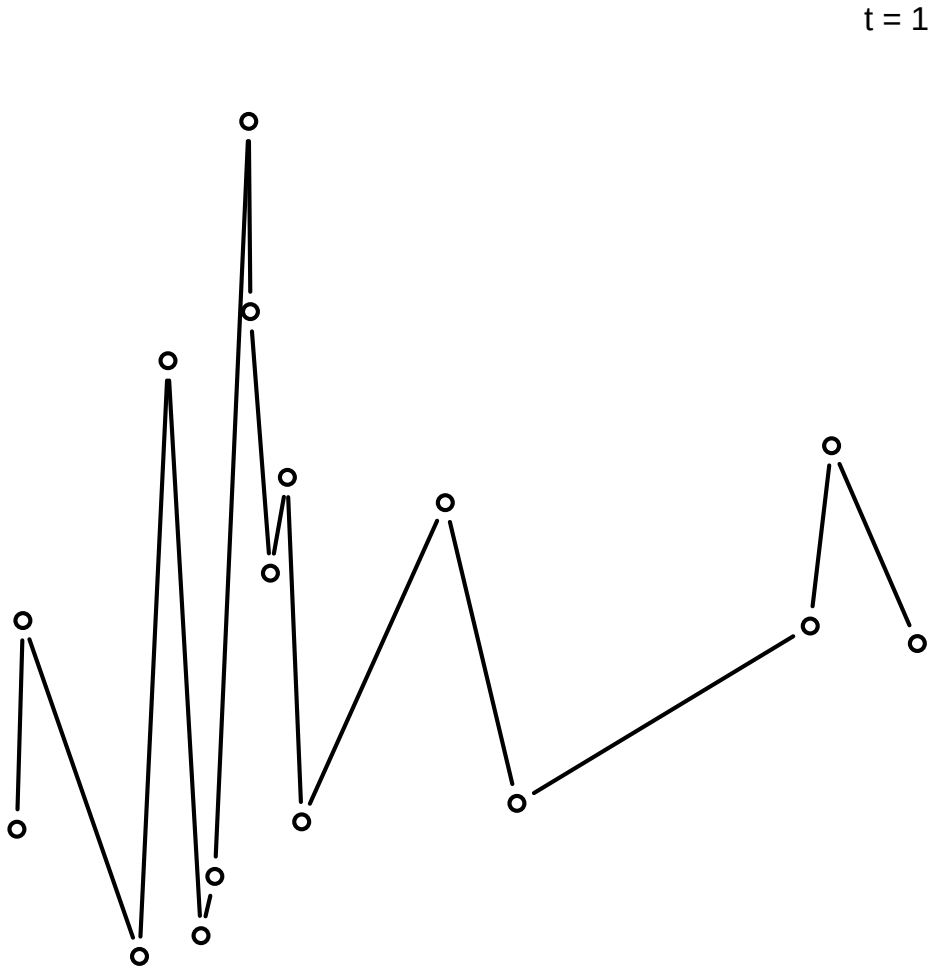
Animated graphics in a PDF file
use package animate in LaTeX

```
> plot(as.animator(block))
> includeLatex(vspace = "-2cm")
```

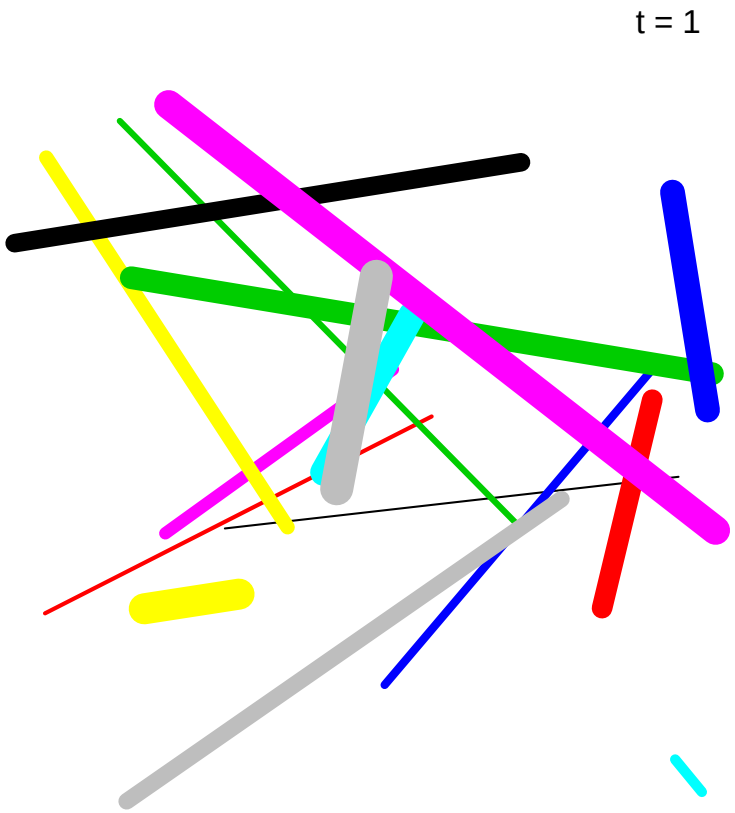


Moving points `tpoints()`

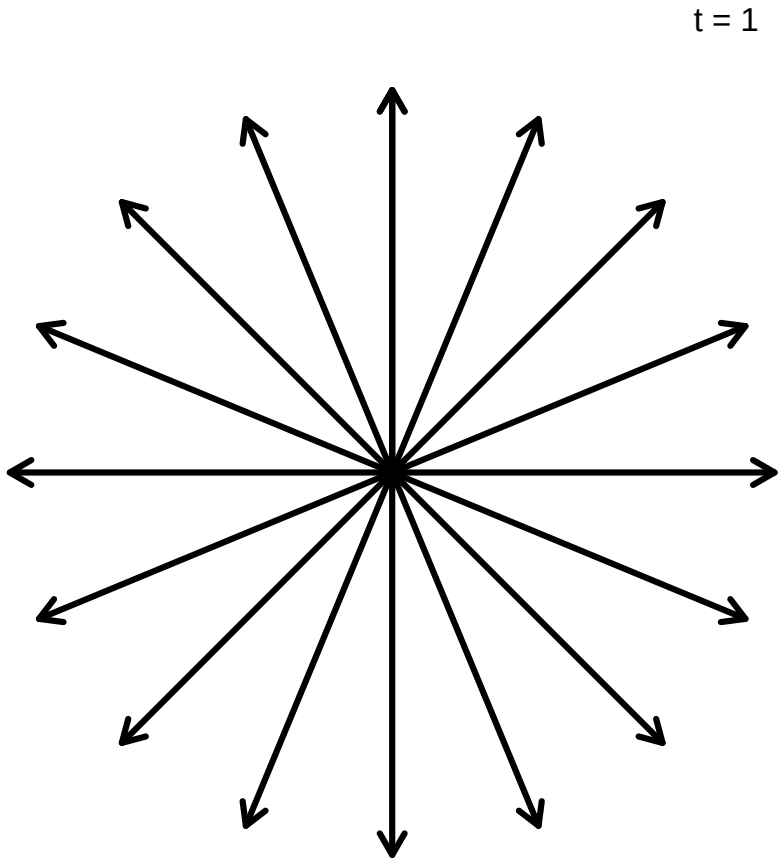


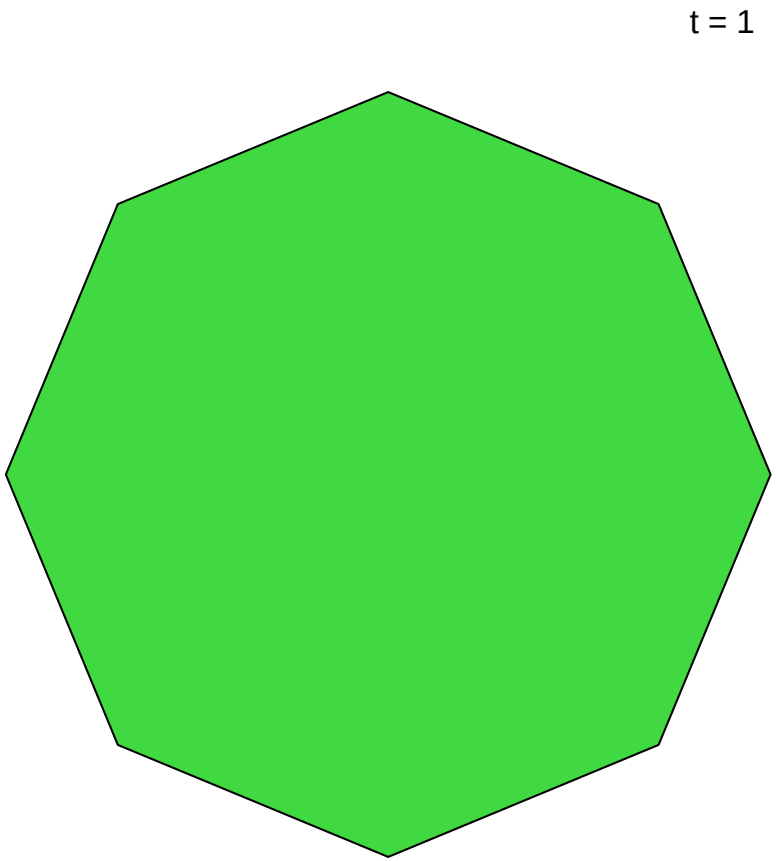

t = 1

Growing segments `tsegments()`

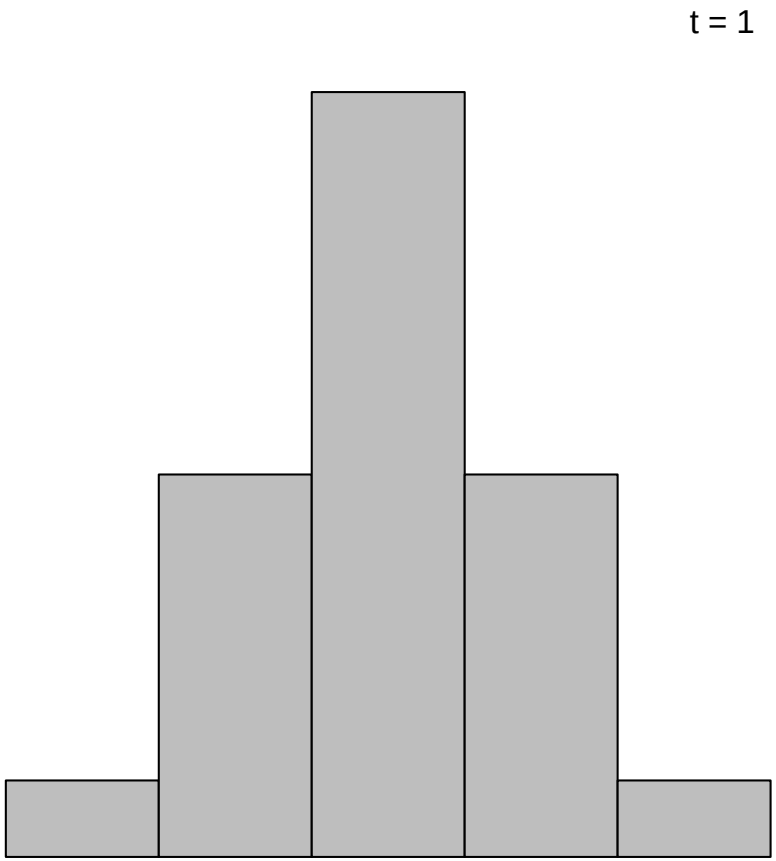


Growing arrows `tarrows()`



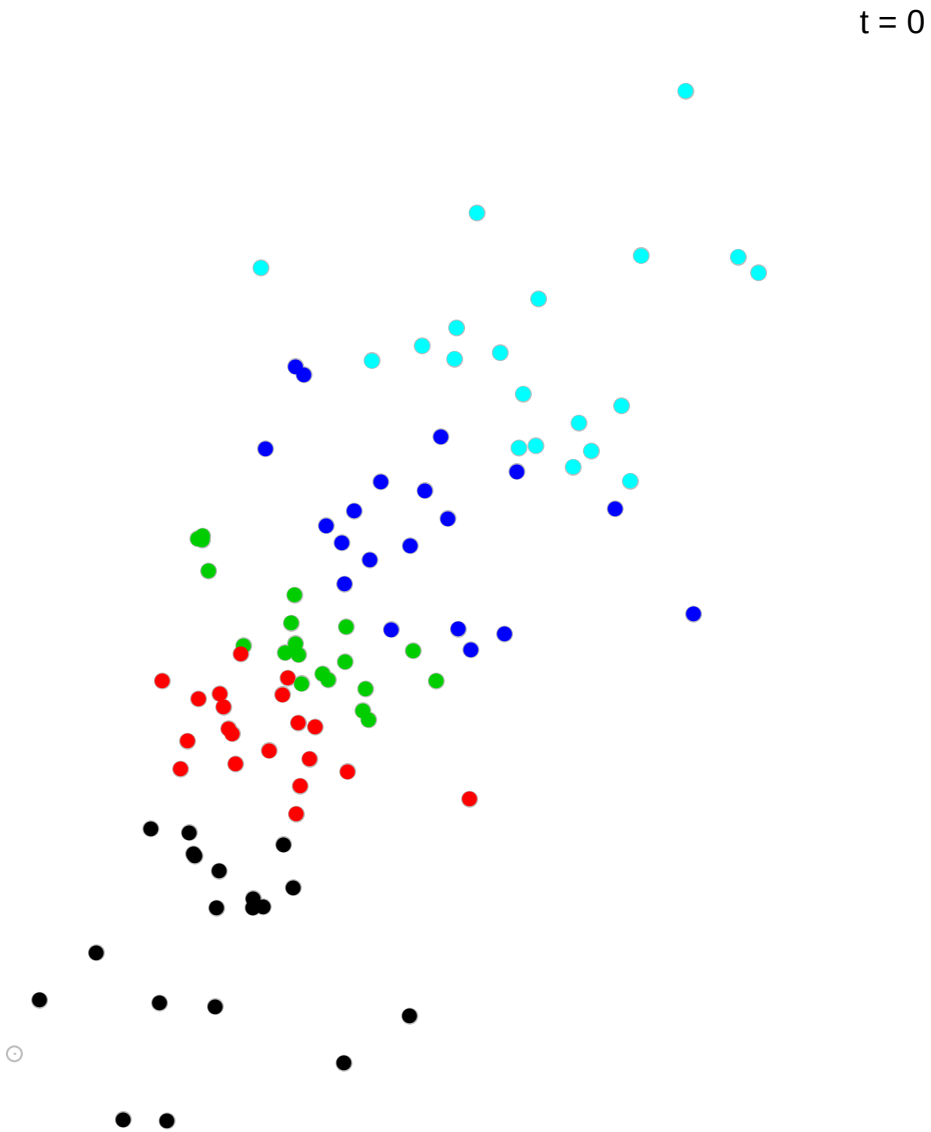

t = 1

Rectangles `trect()`



Fireworks

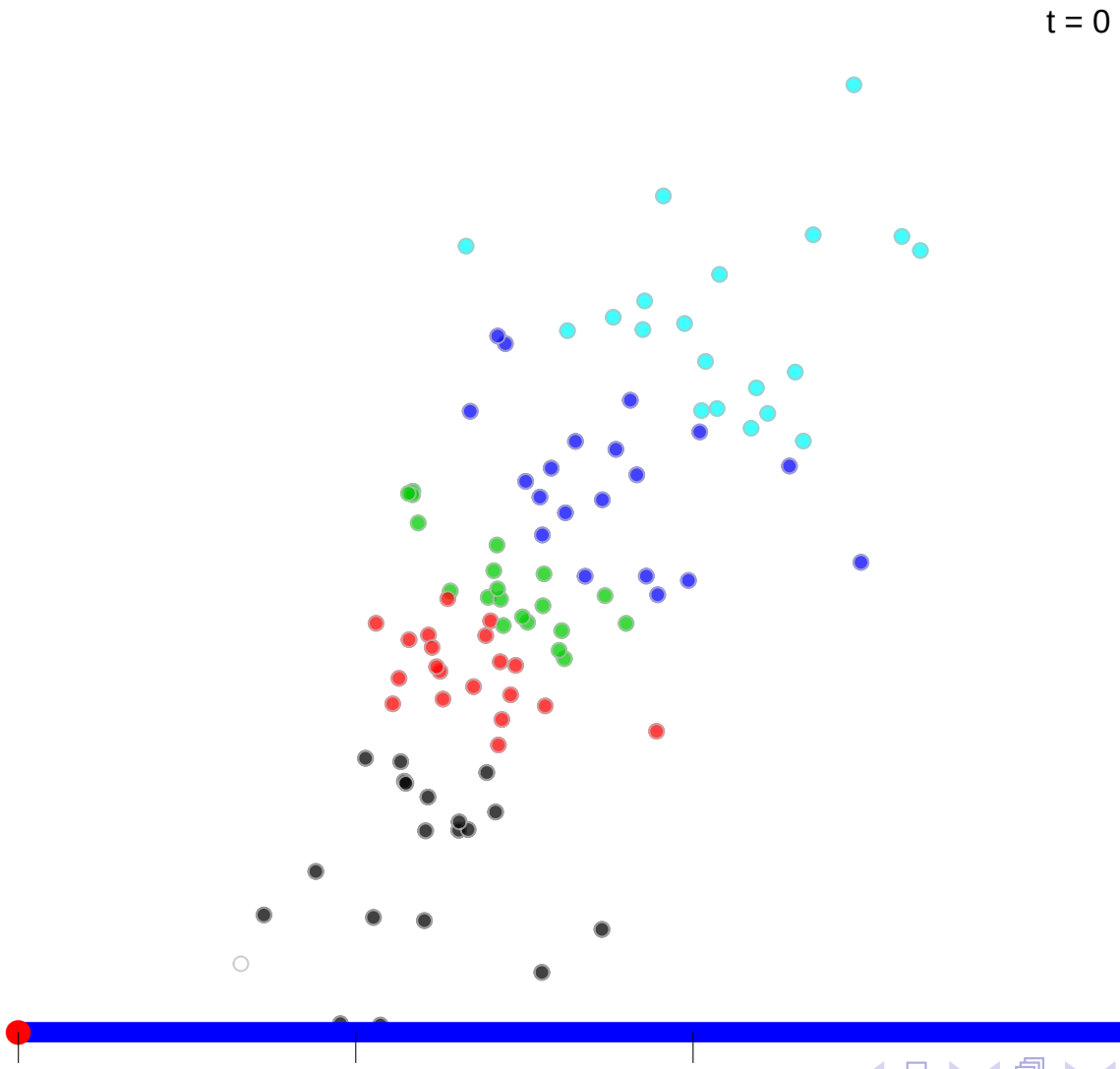
Projection to the first principal component



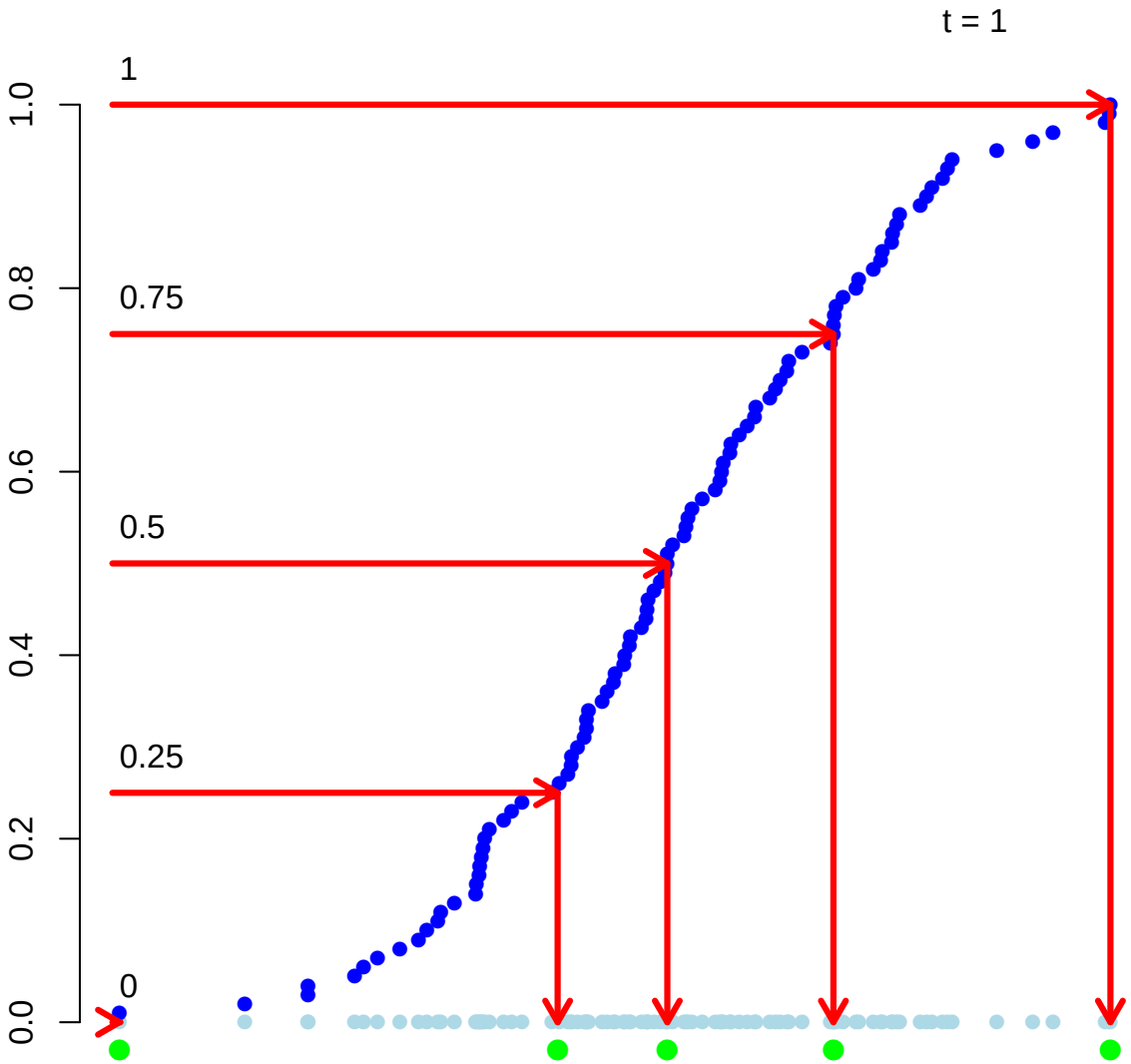
Move points to median then expand to mid-quintiles

(0.0, 0.3)

(0.6, 1.0)



Finding quartiles



Matrix multiplication: points move

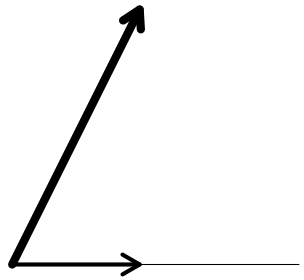
t = 0

```
     [,1] [,2]
[1,]    1    2
[2,]    2    1
```

Matrix multiplication: change of direction and magnitude

t = 1

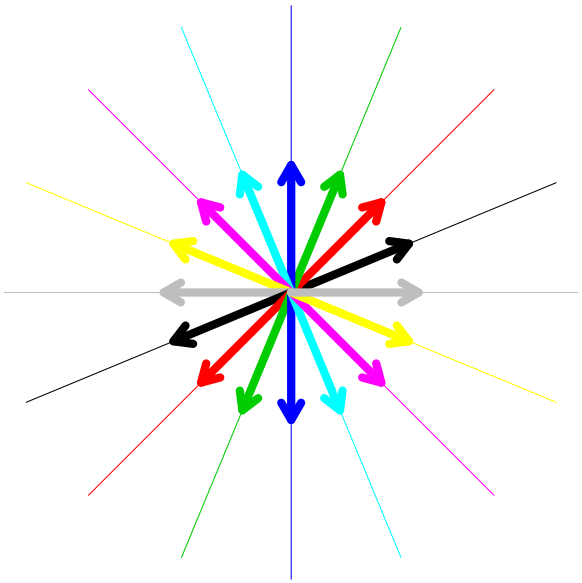
```
     [,1] [,2]
[1,]    1    2
[2,]    2    1
```



Matrix multiplication: find the eigenvectors

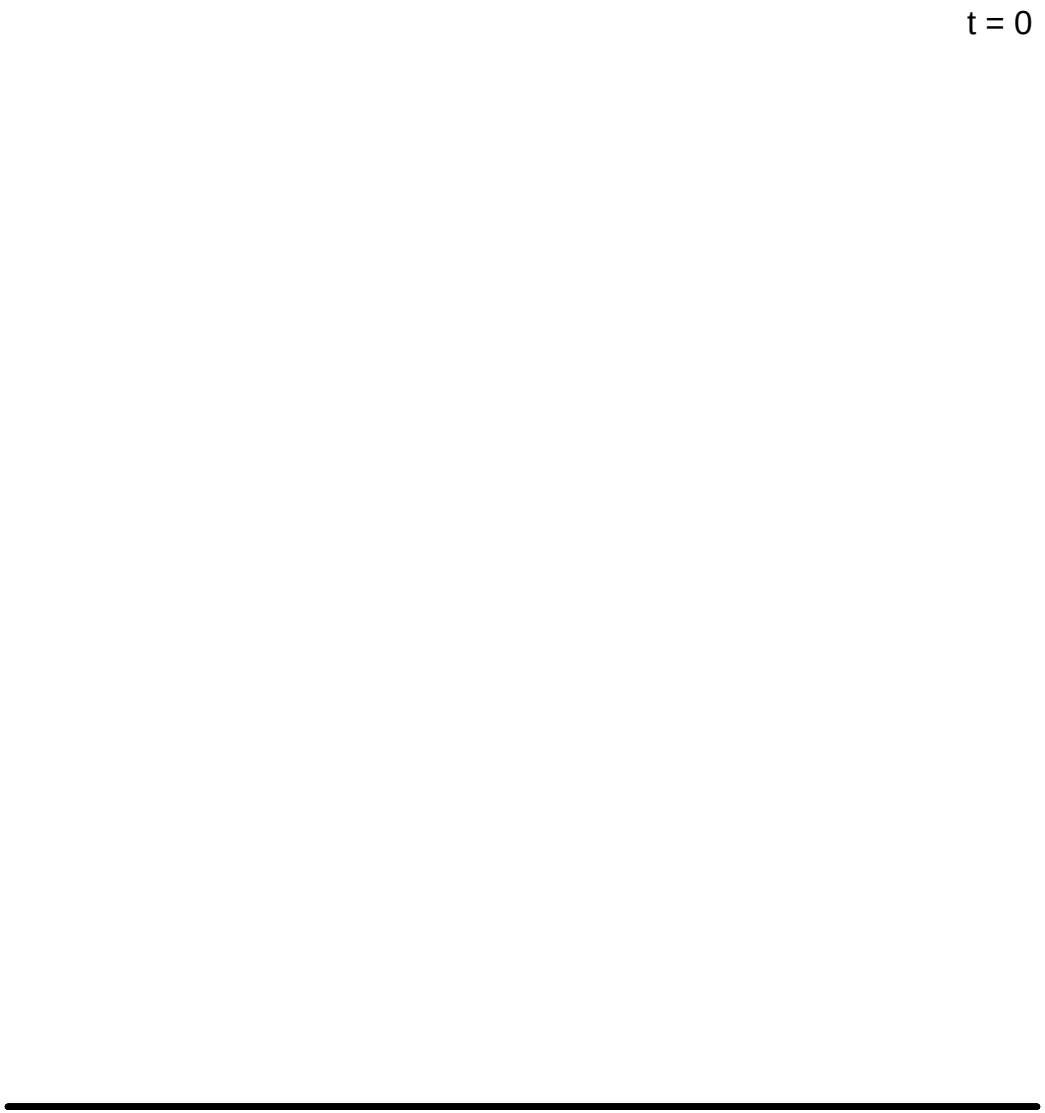
t = 0

```
     [,1] [,2]
[1,]    1    2
[2,]    2    1
```

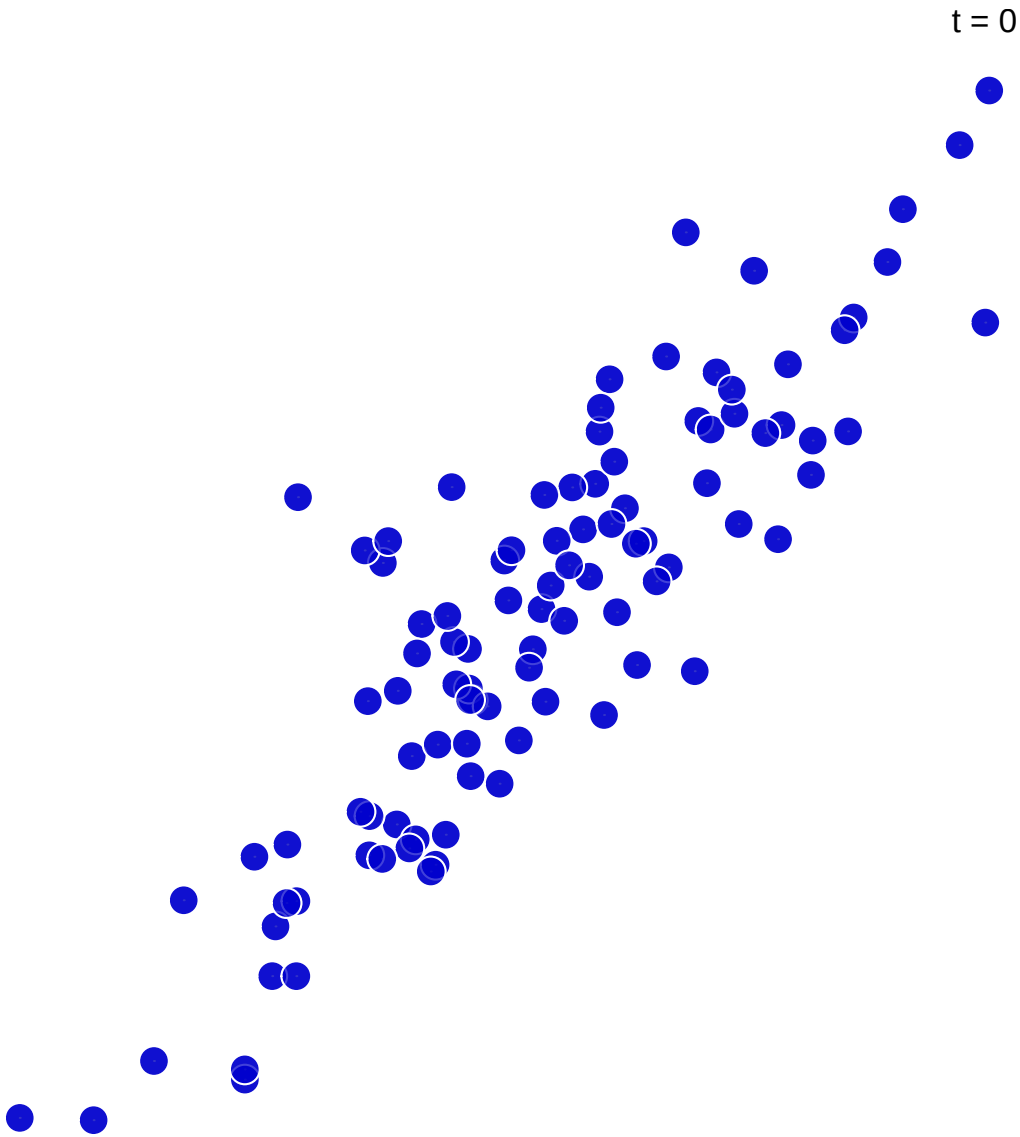


Overlapped distributions

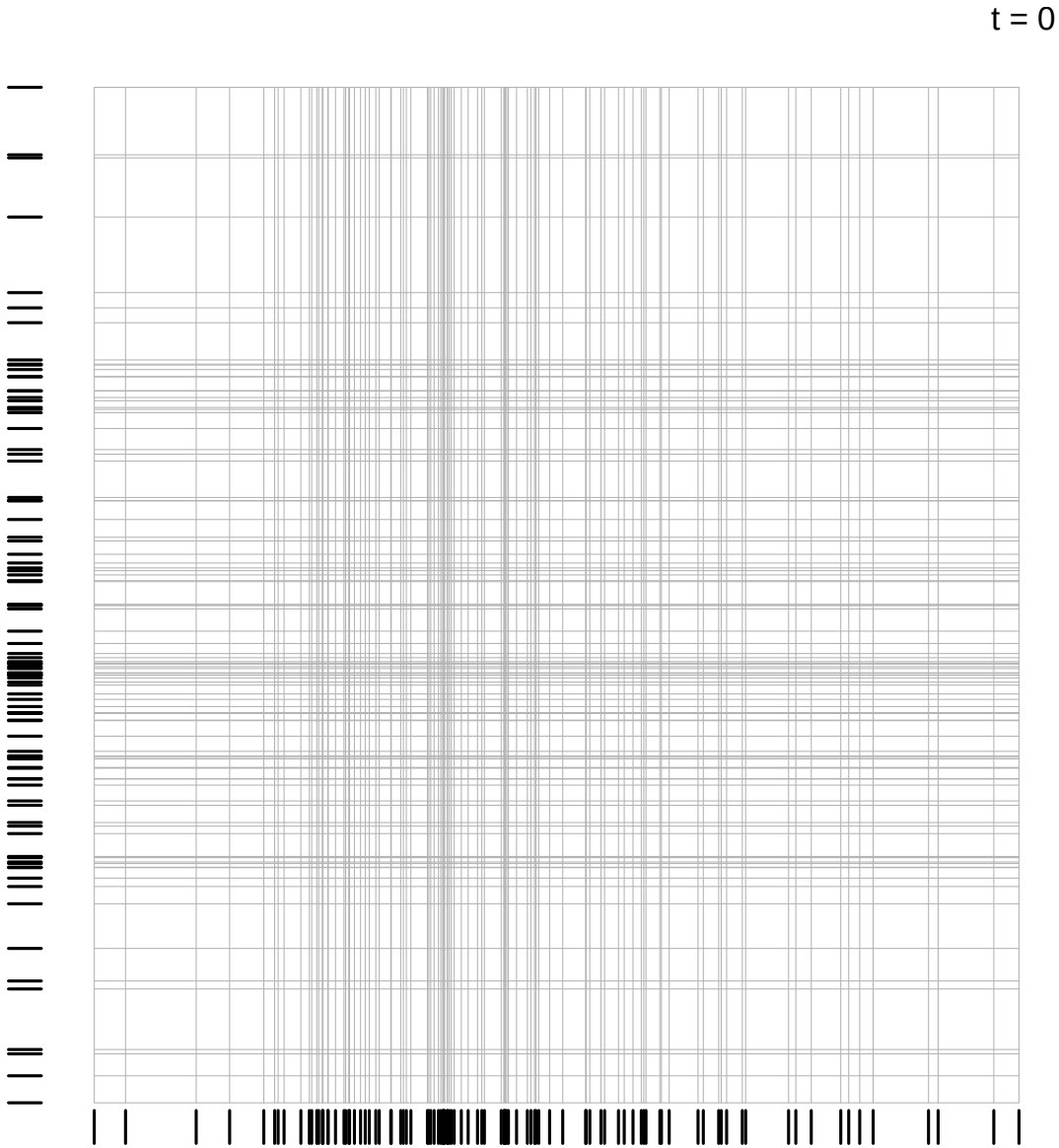
Overlapped distributions



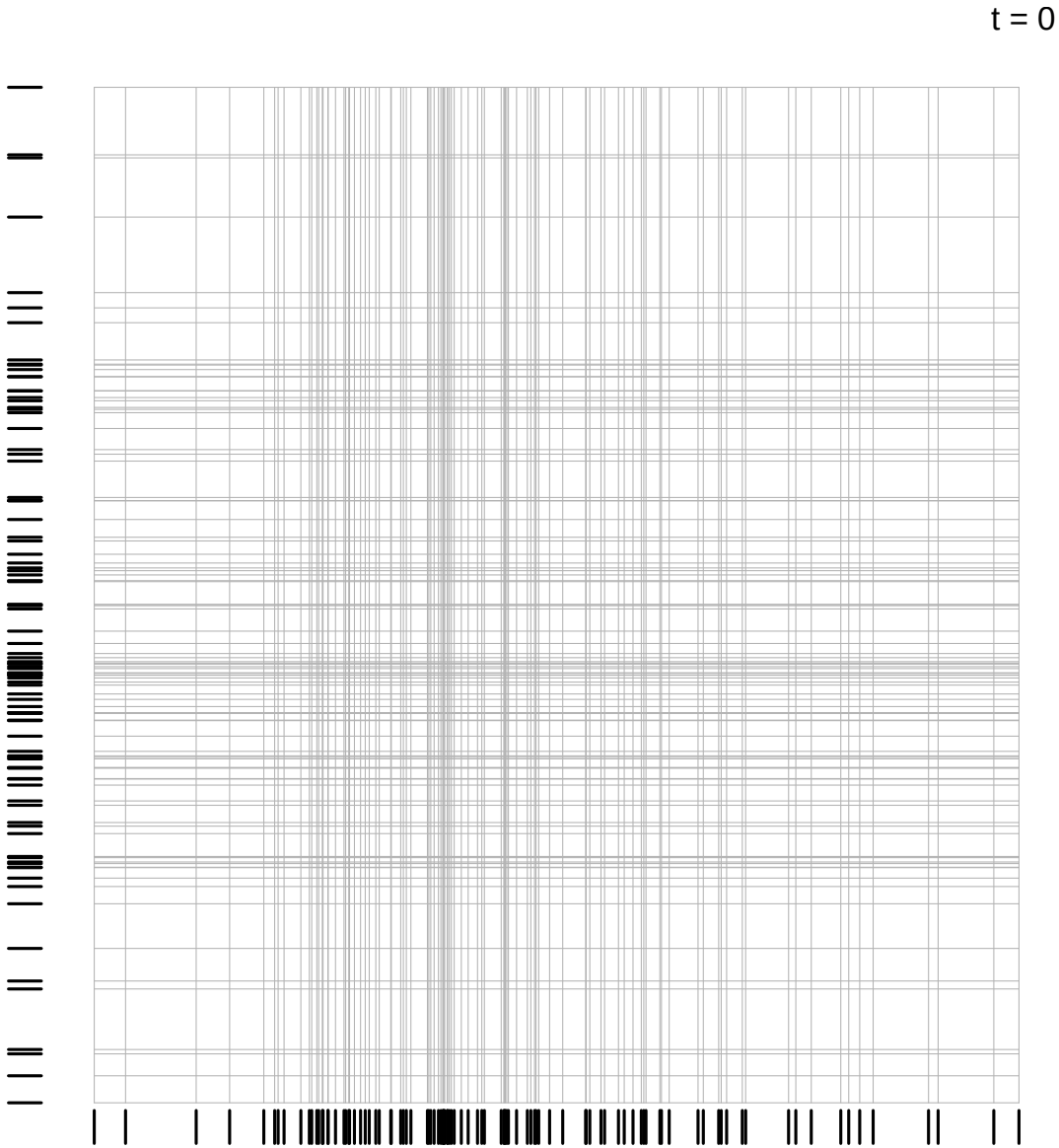
From bivariate distribution to marginal distributions



From marginal distributions to bivariate ?



From marginal distributions to bivariate ?



From marginal distributions to bivariate ?