

R / C++

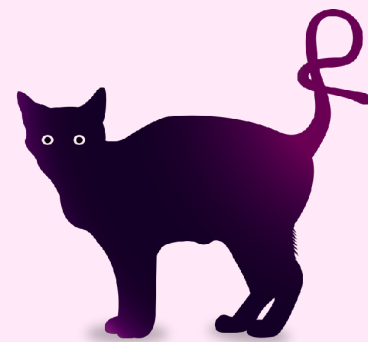
Romain FRANÇOIS

#eRum2016

romain@purrple.cat
@romain_francois

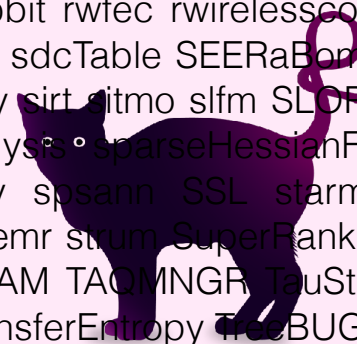


updates

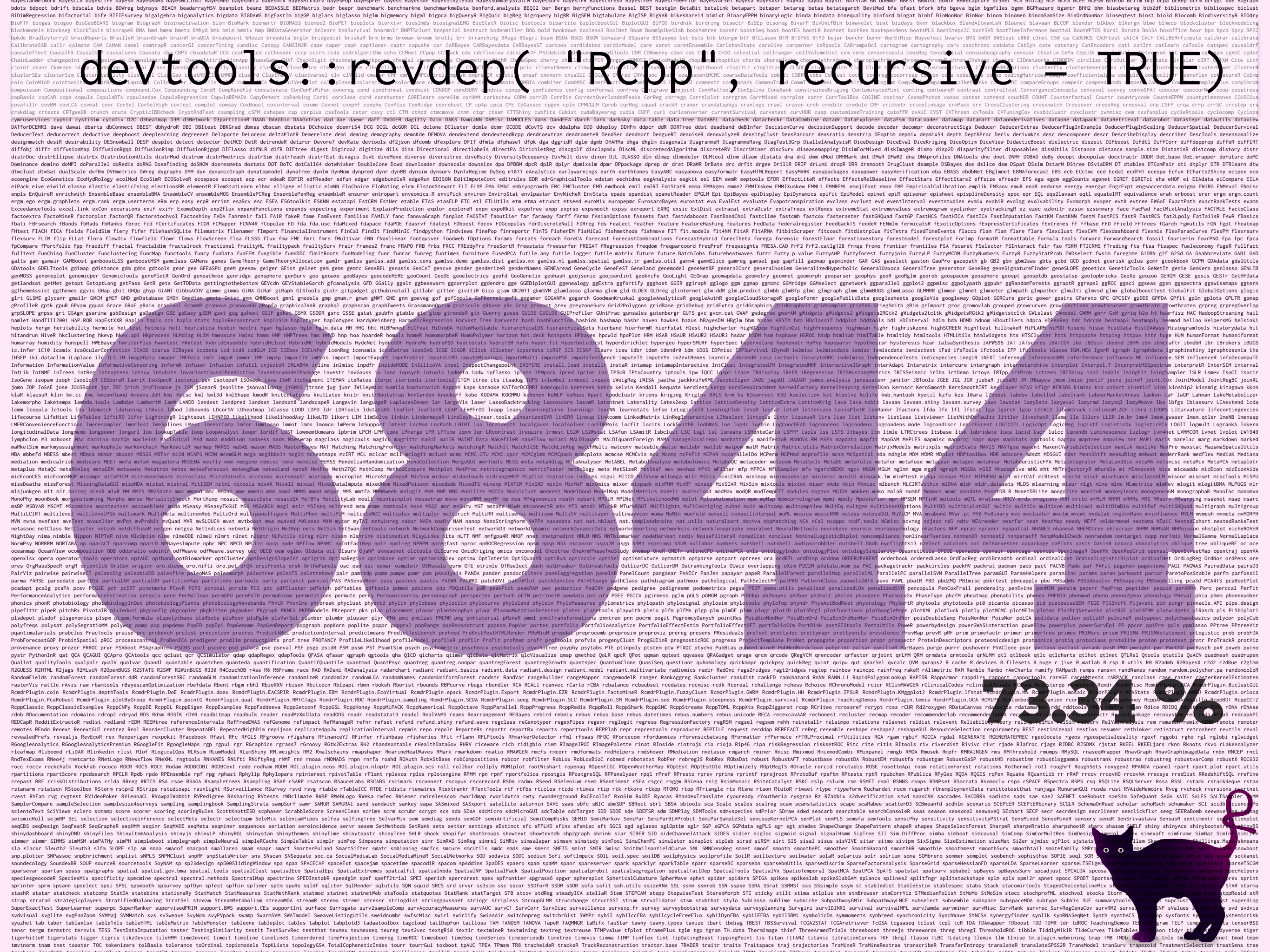


devtools::revdep("Rcpp")

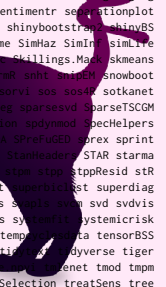
8.60%



devtools::revdep("Rcpp", recursive = TRUE)



73.34 %



*.h

*.cpp

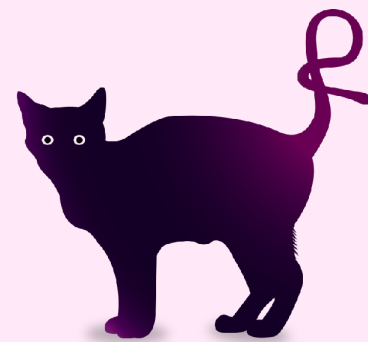
*.R

94390

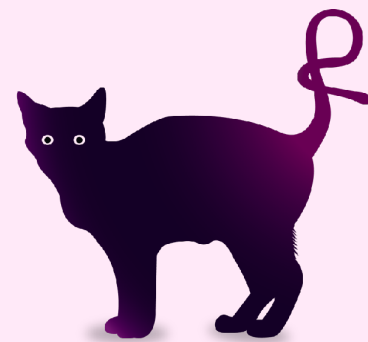
5575

99965

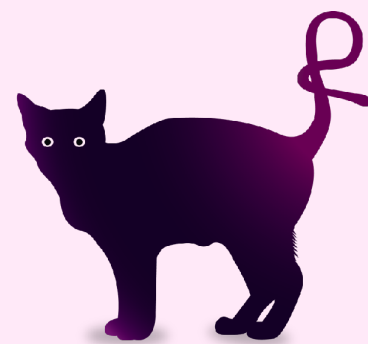
3134



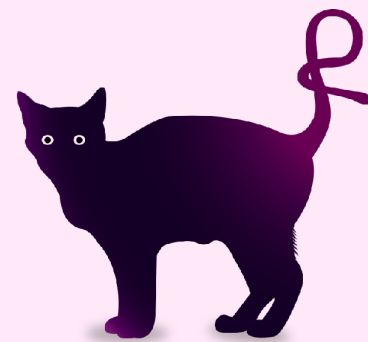
motivation



```
int add( int a, int b ){  
    return a + b ;  
}
```



C/R API




```
#include <R.h>
#include <Rinternals.h>

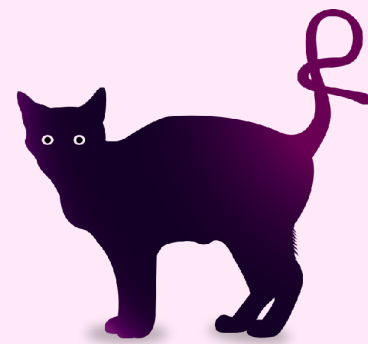
int add( int a, int b ){
    return a + b ;
}

SEXP add_fun( SEXP a_, SEXP b_ ){
    int a = INTEGER(a_)[0], b = INTEGER(b_)[0] ;

    int res = add( a, b ) ;

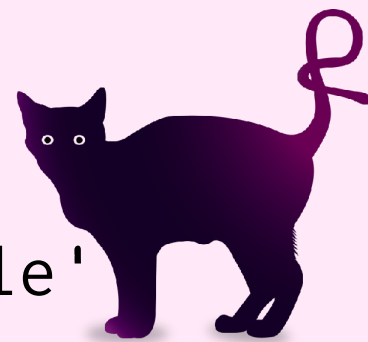
    SEXP result = PROTECT(allocVector(INTSXP, 1) ) ;
    INTEGER(result)[0] = res ;
    UNPROTECT(1) ;

    return result ;
}
```

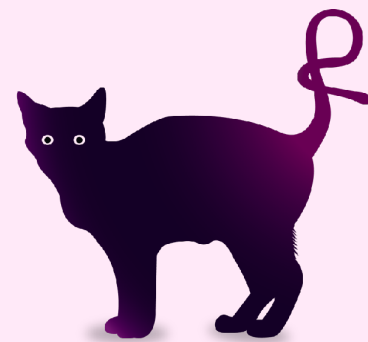


```
$ R CMD SHLIB add.c
clang -I/Library/Frameworks/R.framework/Resources/include -DNDEBUG -I/usr/local/include -I/usr/local/include/freetype2 -I/opt/X11/include -fPIC -Wall -mtune=core2 -g -O2 -c add.c -o add.o
clang -dynamiclib -Wl,-headerpad_max_install_names -undefined dynamic_lookup -single_module -multiply_defined suppress -L/Library/Frameworks/R.framework/Resources/lib -L/usr/local/lib -o add.so add.o -F/Library/Frameworks/R.framework/.. -framework R -Wl,-framework -Wl,CoreFoundation
```

```
> dyn.load( "add.so" )
> add <- function(a, b){
+   .Call( "add_c", a, b )
+ }
> add( 33L, 9L )
[1] 42
>
> add( 33, 9 )
Error in add(33, 9) :
  INTEGER() can only be applied to a 'integer', not a 'double'
```



Rcpp



```
#include <Rcpp.h>
```

```
// [[Rcpp::export]]
```

```
int add( int a, int b ){  
    return a + b ;  
}
```

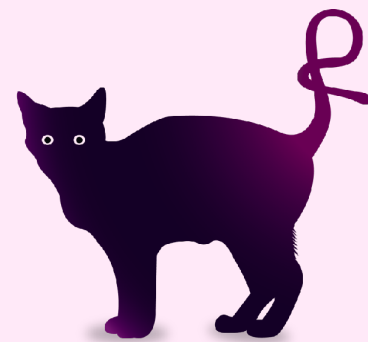
```
> sourceCpp( "add.cpp" )
```

```
> add( 17L, 25L )
```

```
[1] 42
```

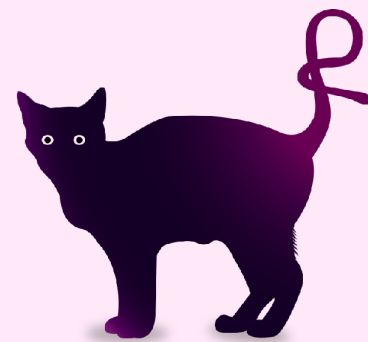
```
> add( 17, 25 )
```

```
[1] 42
```

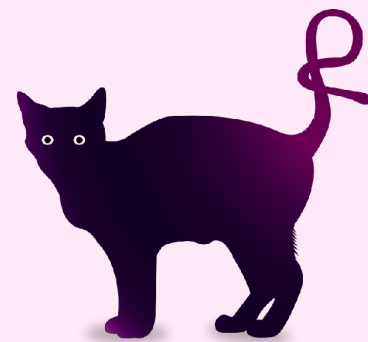


```
#include <Rcpp.h>
using namespace Rcpp

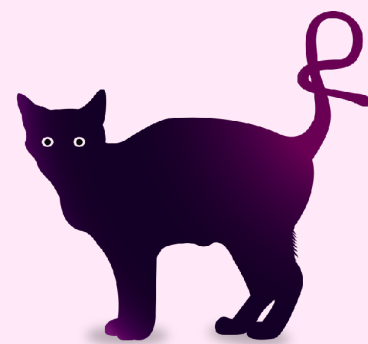
// [[Rcpp::export]]
double sum_cpp( NumericVector x){
    int n = x.size() ;
    double res = 0.0 ;
    for( int i=0; i<n; i++){
        res += x[i] ;
    }
    return res ;
}
```



sugar

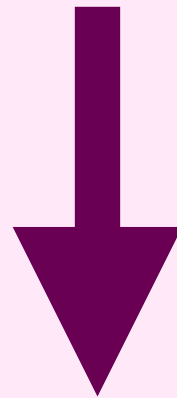



```
pdistR <- function(x, ys) {  
  sqrt((x - ys) ^ 2)  
}
```

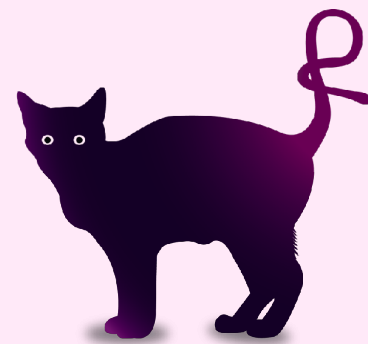


sugar

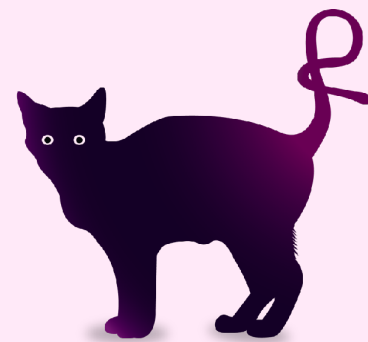
```
NumericVector pdistCpp( double x, NumericVector ys) {  
    int n = ys.size() ;  
    NumericVector res(n) ;  
    for(int i=0; i<n; i++){  
        res[i] = sqrt( pow( x-ys[i], 2.0 ) ) ;  
    }  
    return res ;  
}
```



```
NumericVector pdistCpp2( double x, NumericVector ys){  
    return sqrt( pow( x - ys, 2 ) ) ;  
}
```



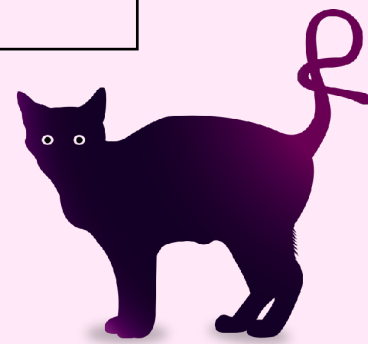
modules



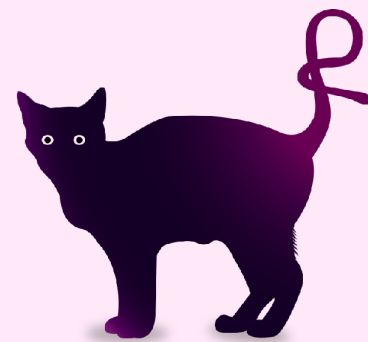
```
class Kittens {  
public:  
    Kittens( int n_ ) : n(n_) ;  
    void meow() { ... };  
    void purr() { ... };  
    int n ;  
} ;
```

```
RcppModule(Cats){  
    class_<Kittens>( "Kittens" )  
        .constructor<int>()  
        .field( "n", &Kittens::n )  
        .method( "meow", &Kittens::meow )  
        .method( "purr", &Kittens::purr )  
    ;  
}
```

```
kittens <- new( Kittens, 10 )  
kittens$n  
kittens$meow()  
kittens$purr()
```

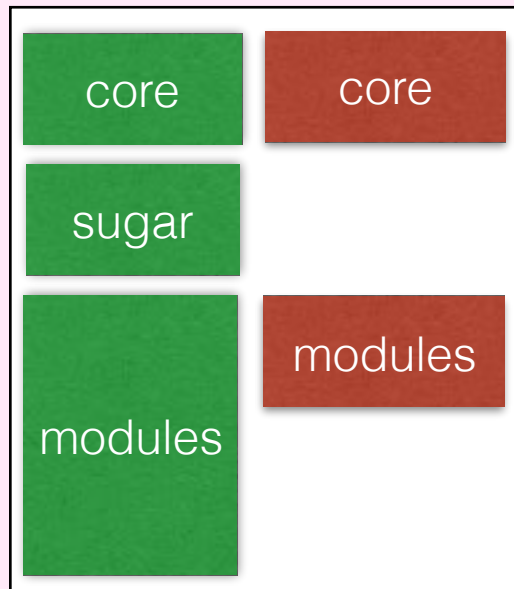


dependency problem

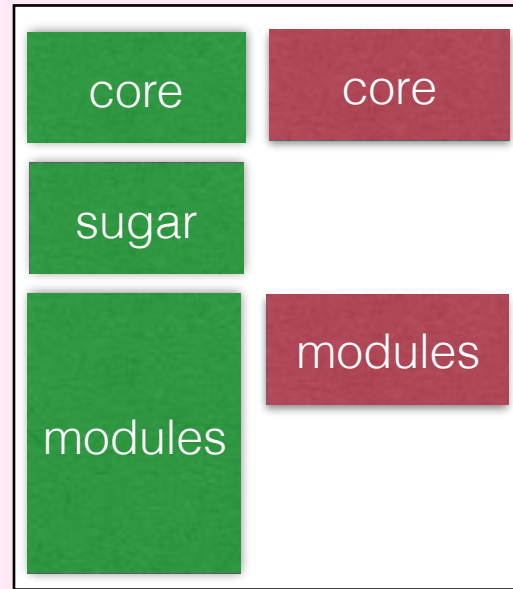


cran

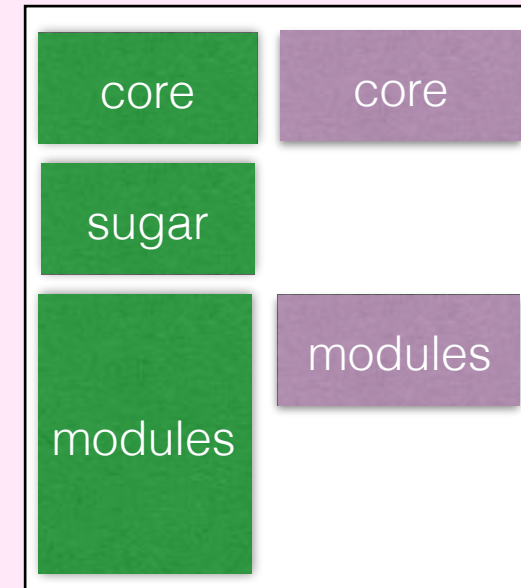
Rcpp n



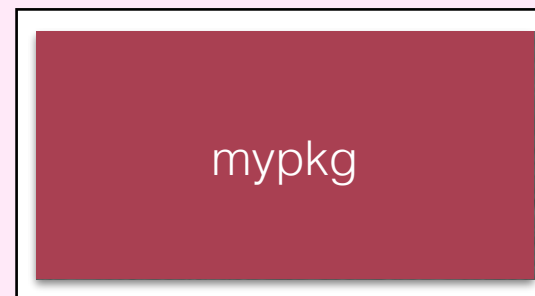
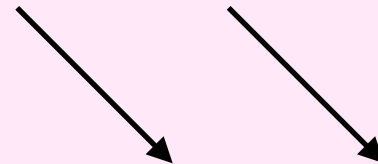
Rcpp n + 1



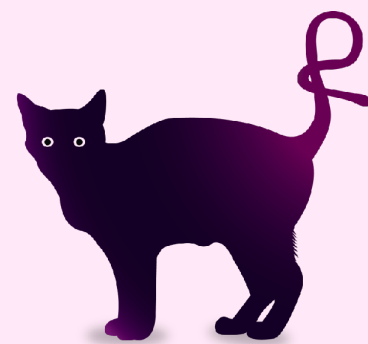
Rcpp n + 2

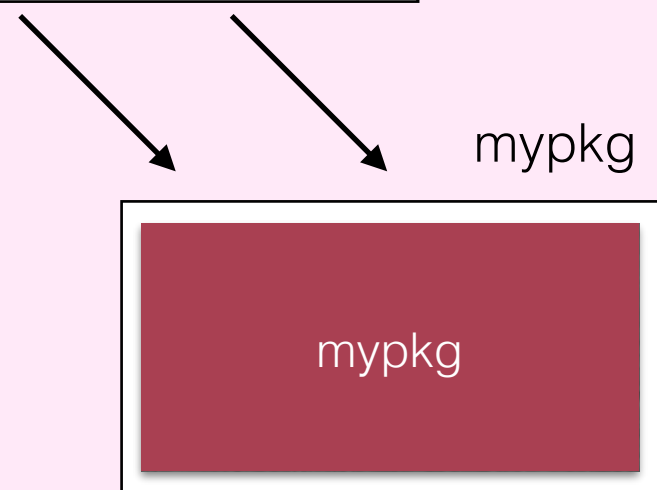
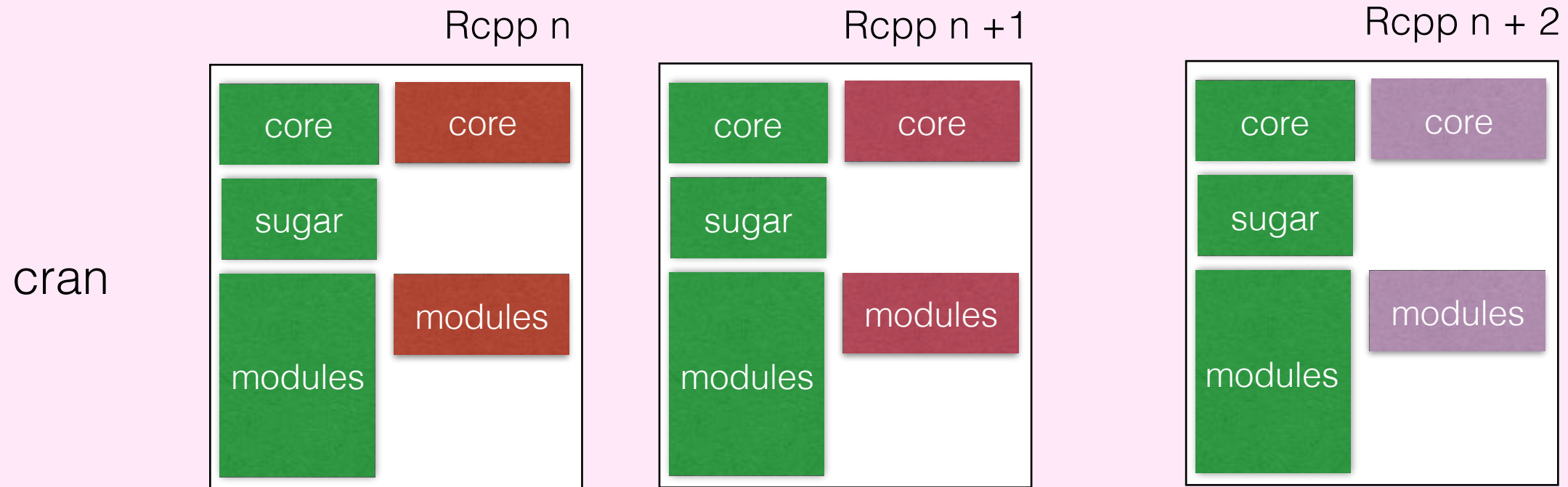


mypkg



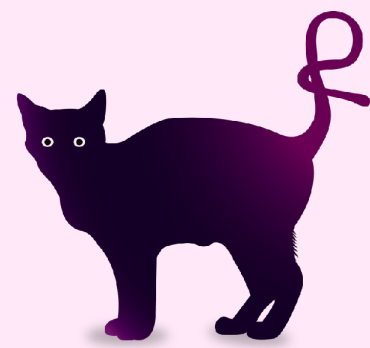
developper

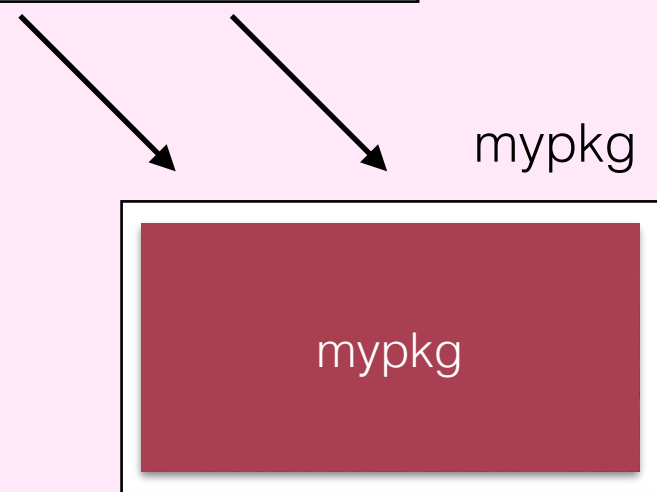
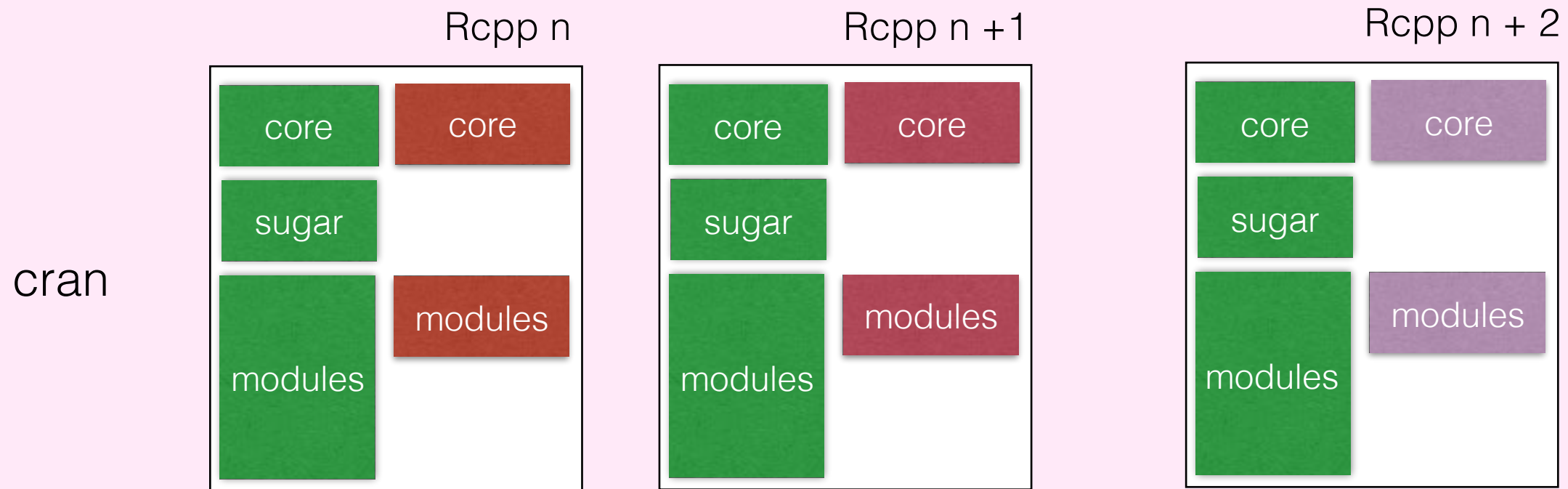




developper

user1



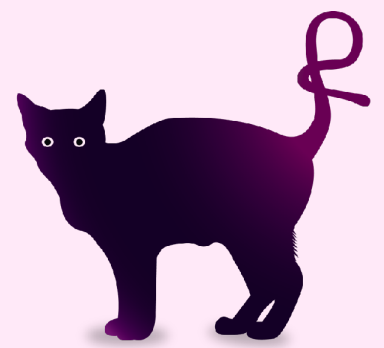
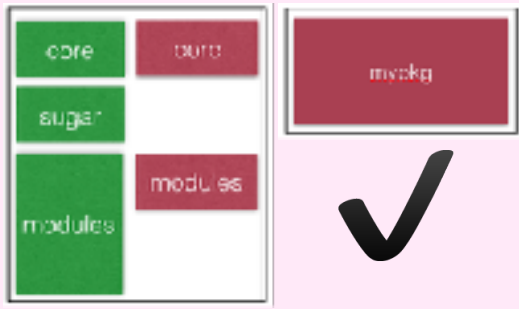


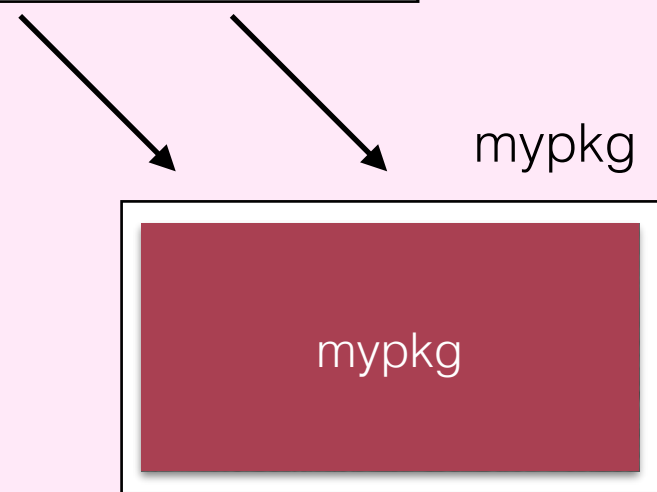
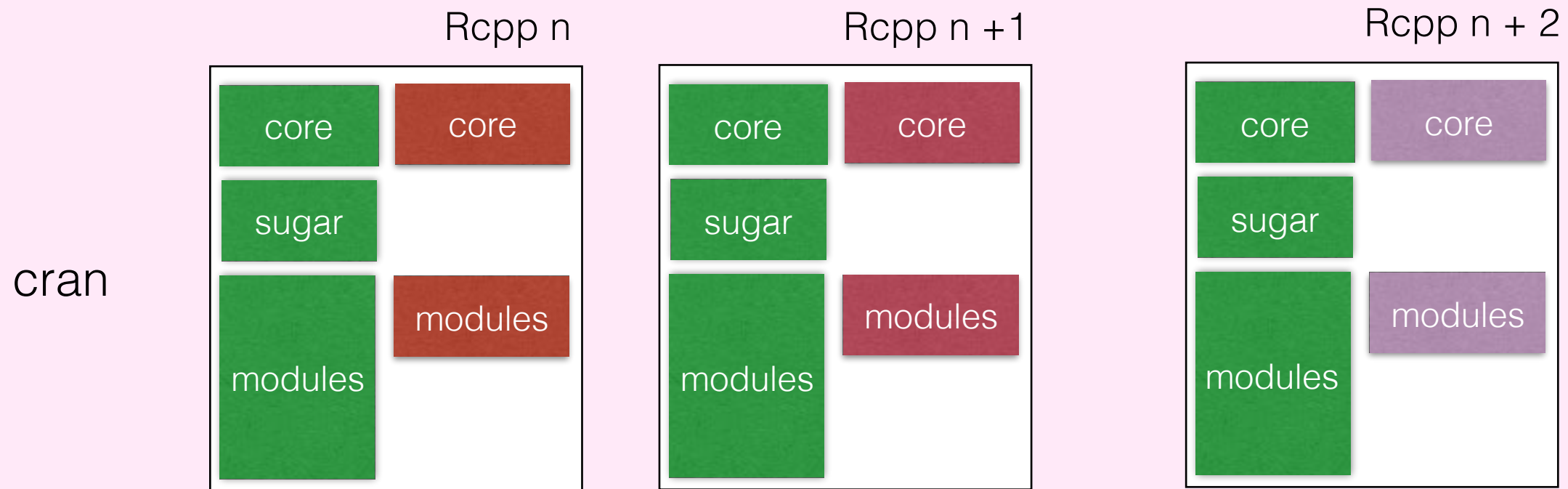
developper

user1



user2



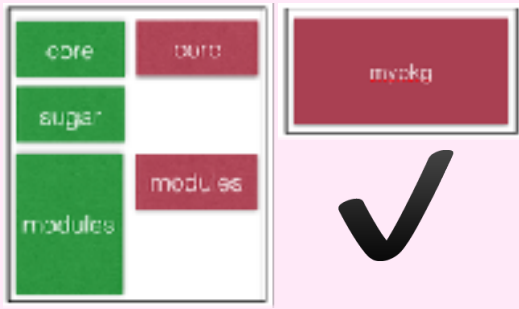


developper

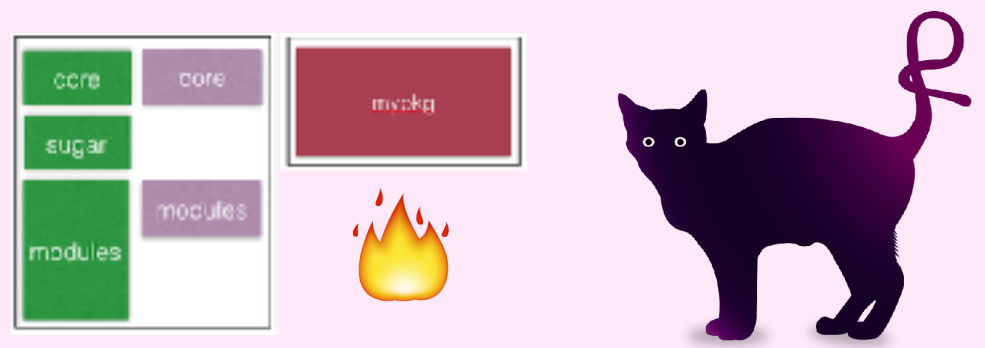
user1



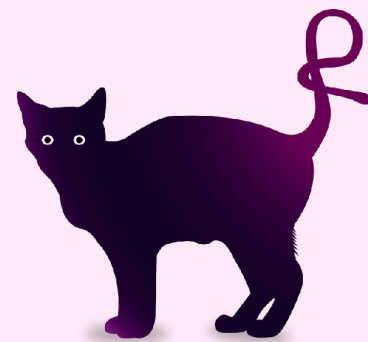
user2



user3



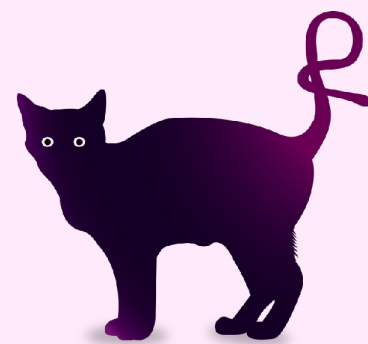
proposal



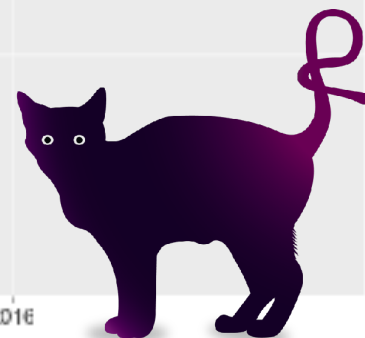
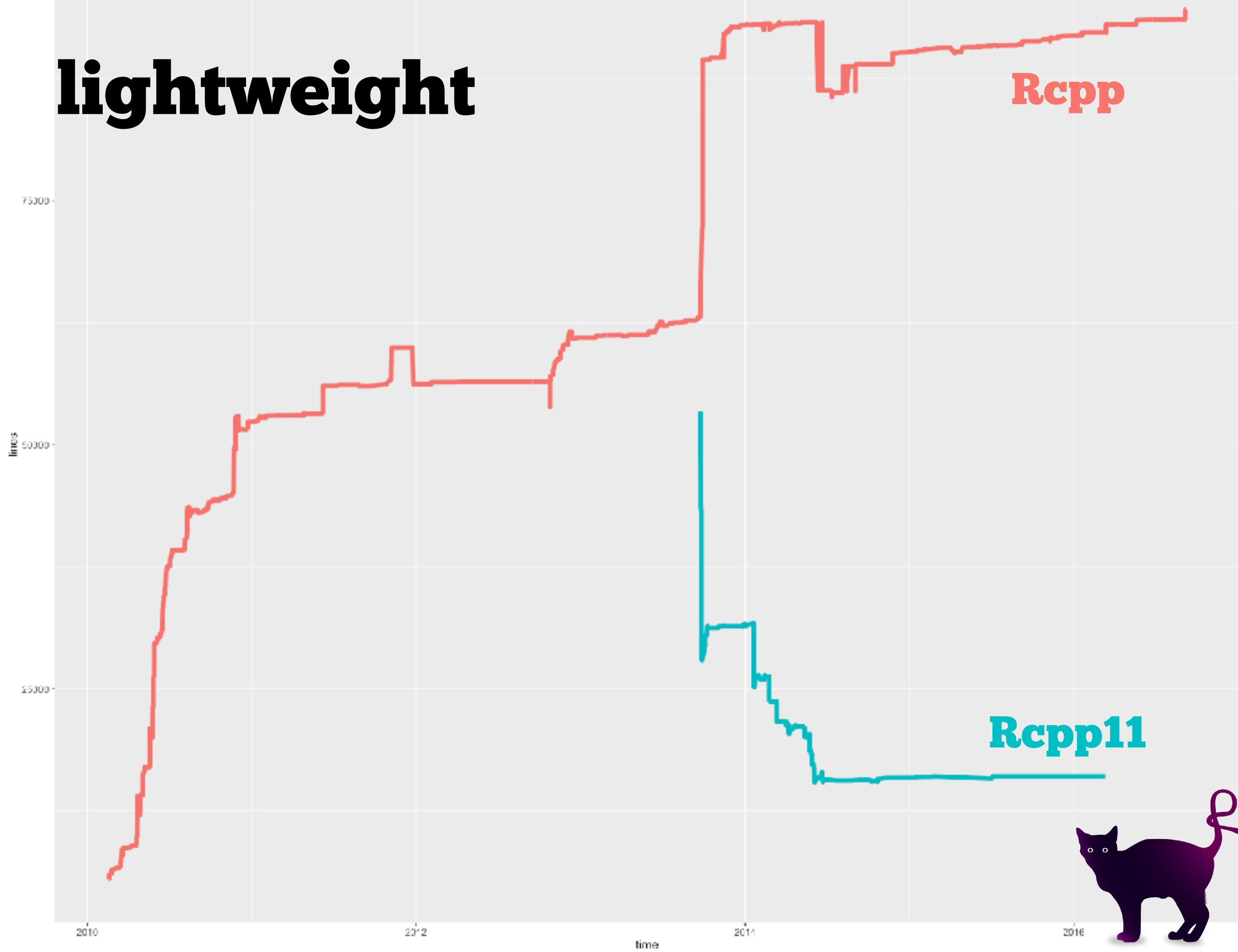


Rcpp11

- C++ = C++11
- header only / no runtime dependency
- lightweight & modular



lightweight



modular

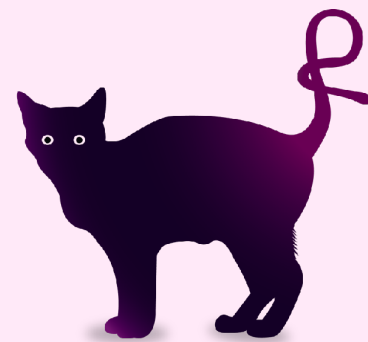
Rcpp11

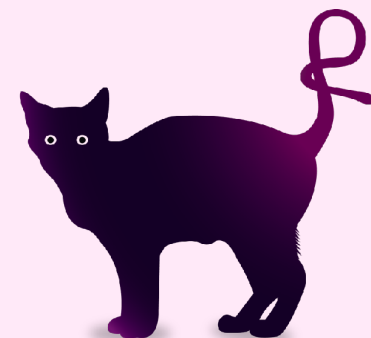
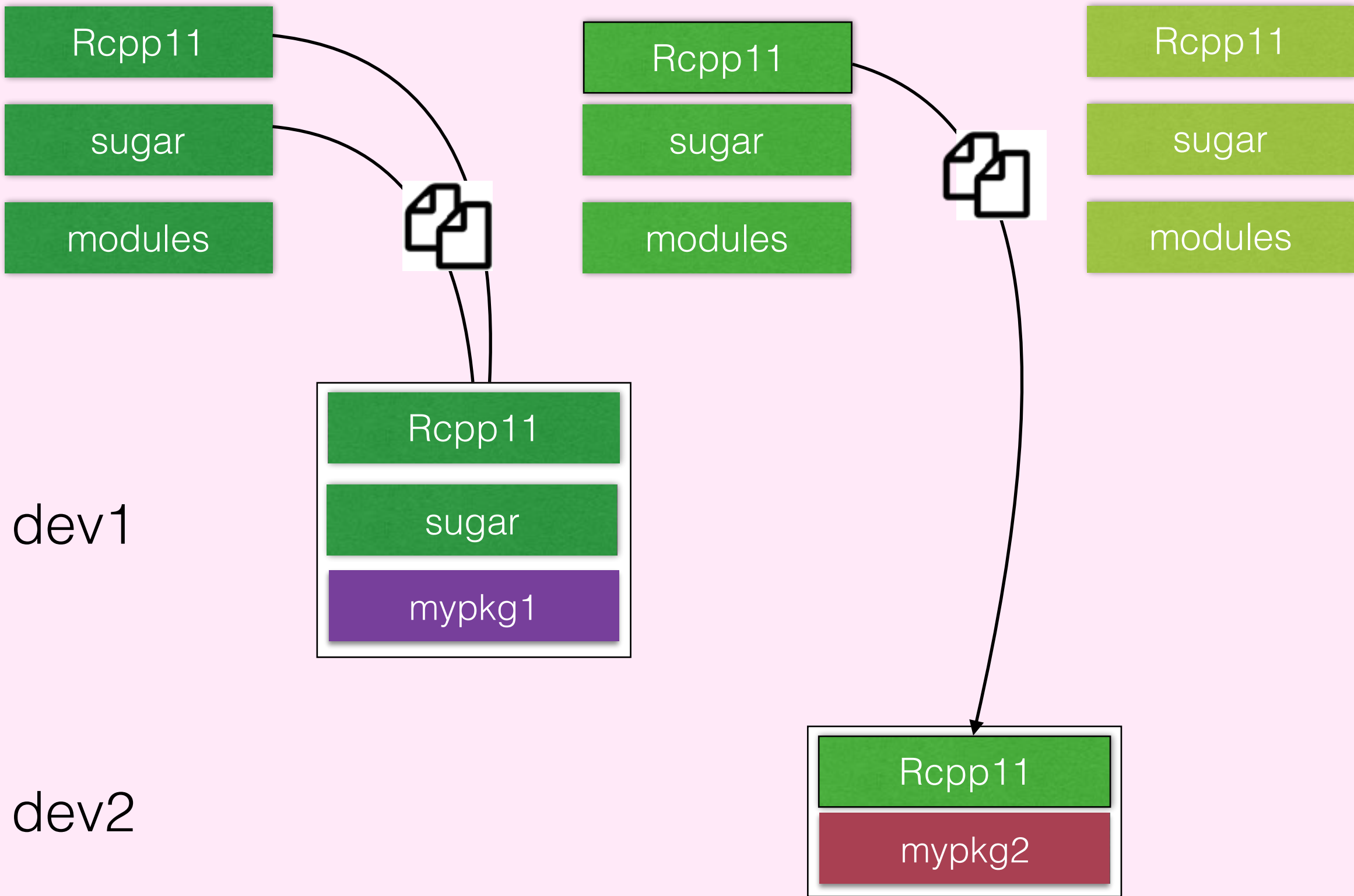
sugar

modules

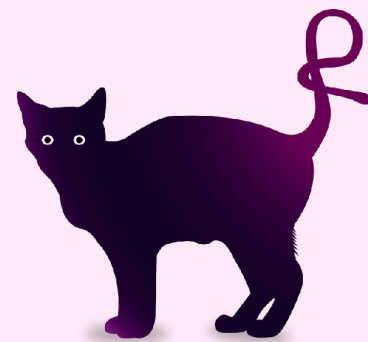
• • •

...



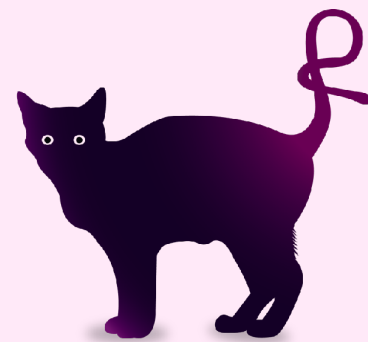


why C++11



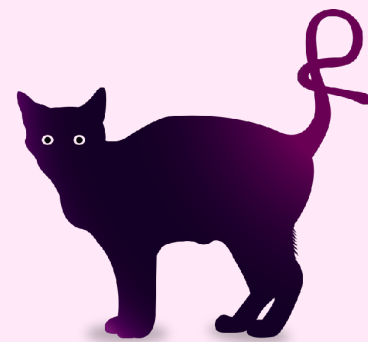
uniform initialization

```
NumericVector x = { 1.0, 2.3, 3.3 } ;
```



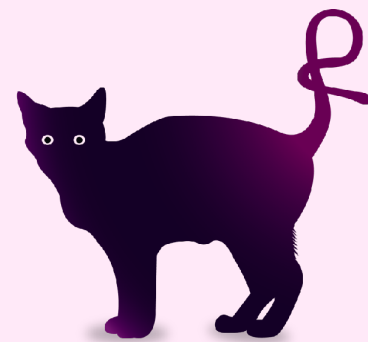
for

```
double sum( NumericVector x ){  
    double res = 0.0 ;  
    for( double d: x) res += d ;  
    return res ;  
}
```



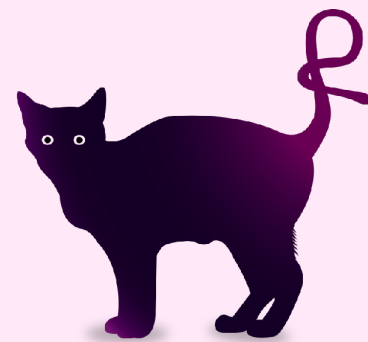
lambdas

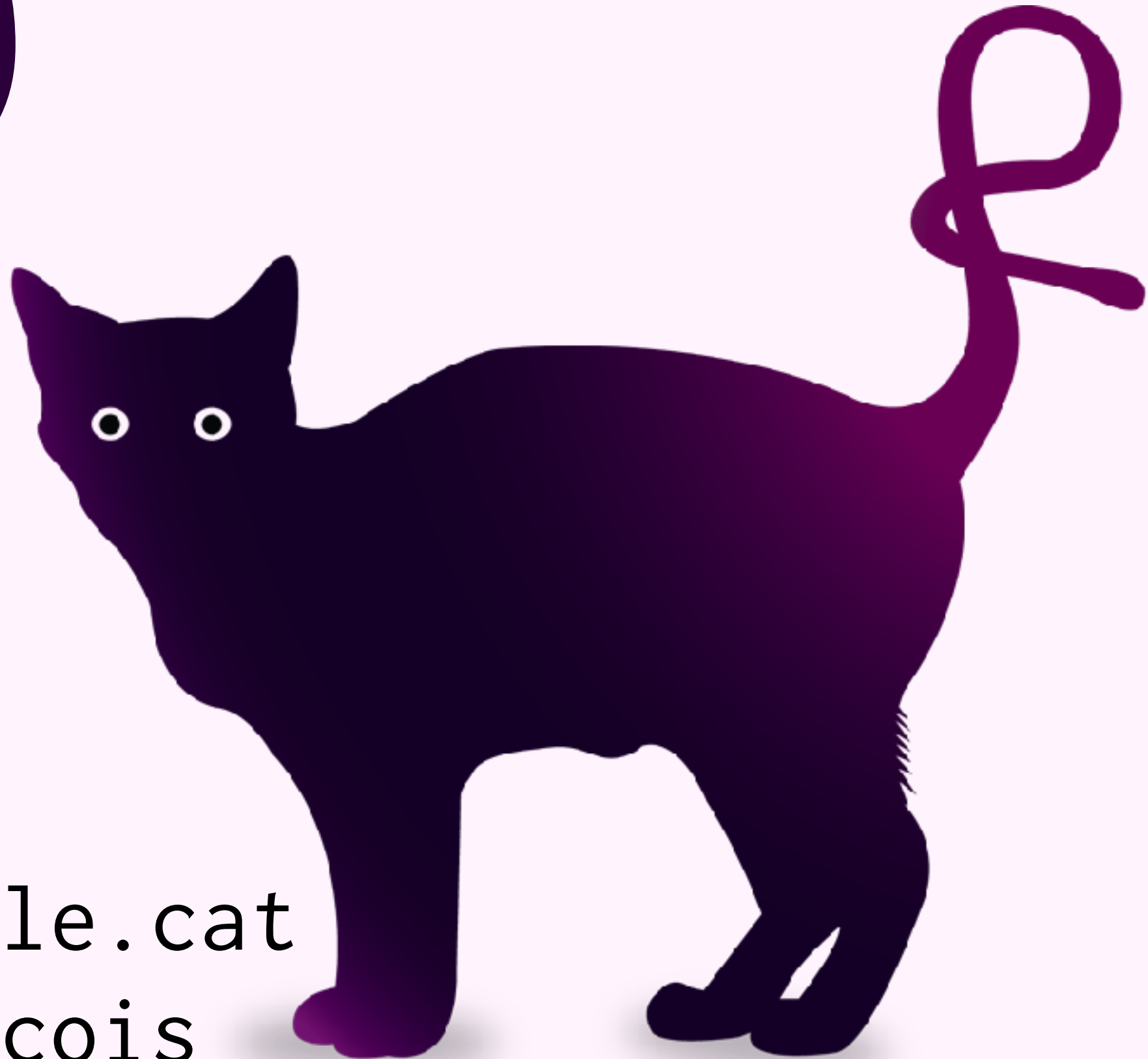
```
NumericVector pdist( double x, NumericVector ys ){  
    return sapply( ys, [&x]( double y ){  
        return sqrt( pow(x-y, 2.0) ) ;  
    } ) ;  
}
```



auto

```
NumericVector pdist( double x, NumericVector ys ){  
    auto fun = [&x]( double y ){  
        return sqrt( pow(x-y, 2.0) ) ;  
    } ;  
    return sapply( ys, fun) ;  
}
```





romain@purrple.cat
@romain_francois