# Building Packages for

and

python™

Dr. Mirco Schönfeld
mirco.schoenfeld@tum.de
@TWlyY29

# Building Packages for

and

python™

Dr. Mirco Schönfeld
mirco.schoenfeld@tum.de
@TWlyY29

base64().encode(“Mirco”)

++++++++[>+++++[>++>+++>+++>+<<<<-]>+>+>->>+[<]<-]>>.>---.+++++++..+++.>>.<-.<.+++.------.--------.>>+.>++.

Hello World!

```
[ This program prints "Hello World!" and a newline to the screen, its
  length is 106 active command characters. [It is not the shortest.]

  This loop is an "initial comment loop", a simple way of adding a comment
  to a BF program such that you don't have to worry about any command
  characters. Any ".", ",", "+", "-", "<" and ">" characters are simply
  ignored, the "[" and "]" characters just have to be balanced. This
  loop and the commands it contains are ignored because the current cell
  defaults to a value of 0; the 0 value causes this loop to be skipped.
]
++++++++               Set Cell #0 to 8
[
    >++++               Add 4 to Cell #1; this will always set Cell #1 to 4
    [                   as the cell will be cleared by the loop
        >++             Add 2 to Cell #2
        >+++            Add 3 to Cell #3
        >+++            Add 3 to Cell #4
        >+              Add 1 to Cell #5
        <<<<-           Decrement the loop counter in Cell #1
    ]                   Loop till Cell #1 is zero; number of iterations is 4
    >+                  Add 1 to Cell #2
    >+                  Add 1 to Cell #3
    >-                  Subtract 1 from Cell #4
    >>+                 Add 1 to Cell #6
    [<]                 Move back to the first zero cell you find; this will
                        be Cell #1 which was cleared by the previous loop
    <-                  Decrement the loop Counter in Cell #0
]                       Loop till Cell #0 is zero; number of iterations is 8

The result of this is:
Cell No :   0   1   2   3   4   5   6
Contents:   0   0  72 104  88  32   8
Pointer :   ^

>>.                     Cell #2 has value 72 which is 'H'
>---.                   Subtract 3 from Cell #3 to get 101 which is 'e'
+++++++..+++.           Likewise for 'llo' from Cell #3
>>.                     Cell #5 is 32 for the space
<-.                     Subtract 1 from Cell #4 for 87 to give a 'W'
<.                      Cell #3 was set to 'o' from the end of 'Hello'
+++.------.--------.    Cell #3 for 'rl' and 'd'
>>+.                    Add 1 to Cell #5 gives us an exclamation point
>++.                    And finally a newline from Cell #6
```

# Brainfuck.

(That is the name of the programming language.)

“For "readability", this code has been spread across many lines, and blanks and comments have been added.” (Wikipedia Article on Brainfuck)

https://en.wikipedia.org/wiki/Brainfuck

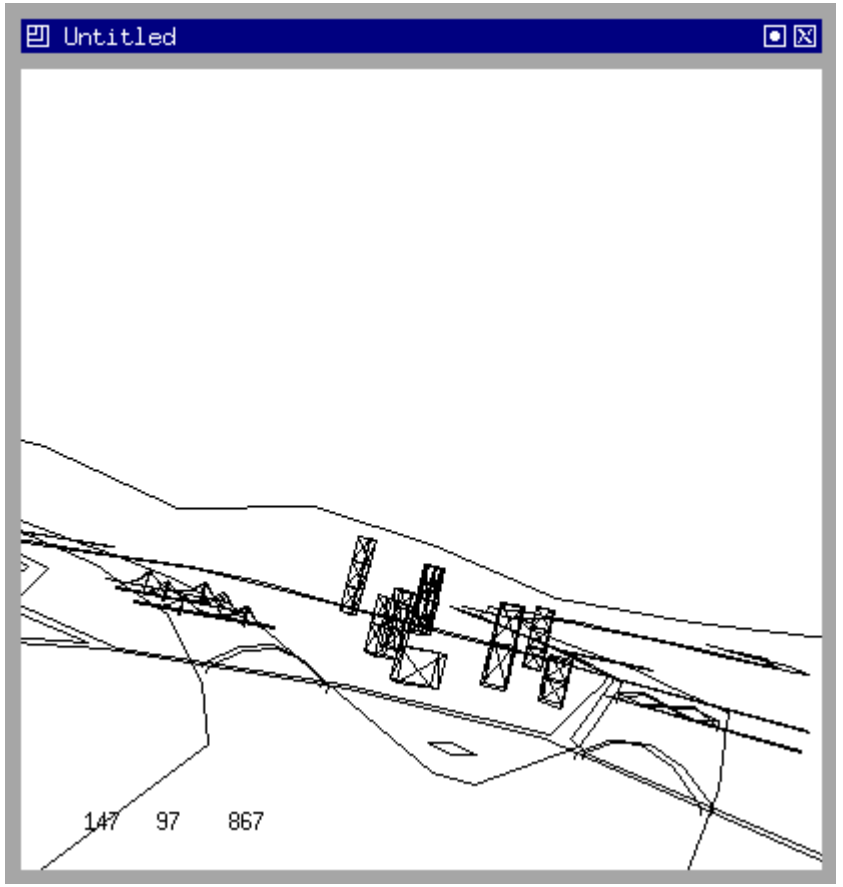
```
#include                                  <math.h>
#include                                 <sys/time.h>
#include                                 <X11/Xlib.h>
#include                                <X11/keysym.h>
                                        double L ,o ,P
                                       ,_=dt,T,Z,D=1,d,
                                       s[999],E,h= 8,I,
                                       J,K,w[999],M,m,O
                                      ,n[999],j=33e-3,i=
                                      1E3,r,t, u,v ,W,S=
                                      74.5,l=221,X=7.26,
                                      a,B,A=32.2,c, F,H;
                                      int N,q, C, y,p,U;
                                     Window z; char f[52]
                                   ; GC k; main(){ Display*e=
 XOpenDisplay( 0); z=RootWindow(e,0); for (XSetForeground(e,k=XCreateGC (e,z,0,0),BlackPixel(e,0))
; scanf("%lf%lf%lf",y +n,w+y, y+s)+1; y ++); XSelectInput(e,z= XCreateSimpleWindow(e,z,0,0,400,400,
0,0,WhitePixel(e,0) ),KeyPressMask); for(XMapWindow(e,z); ; T=sin(O)){ struct timeval G={ 0,dt*1e6}
; K= cos(j); N=1e4; M+= H*_; Z=D*K; F+=_*P; r=E*K; W=cos( O); m=K*W; H=K*T; O+=D*_*F/ K+d/K*E*_; B=
sin(j); a=B*T*D-E*W; XClearWindow(e,z); t=T*E+ D*B*W; j+=d*_*D-_*F*E; P=W*E*B-T*D; for (o+=(I=D*W+E
*T*B,E*d/K *B+v+B/K*F*D)*_; p<y; ){ T=p[s]+i; E=c-p[w]; D=n[p]-L; K=D*m-B*T-H*E; if(p [n]+w[ p]+p[s
]== 0|K <fabs(W=T*r-I*E +D*P) |fabs(D=t *D+Z *T-a *E)> K)N=1e4; else{ q=W/K *4E2+2e2; C= 2E2+4e2/ K
 *D; N-1E4&& XDrawLine(e ,z,k,N ,U,q,C); N=q; U=C; } ++p; } L+=_* (X*t +P*M+m*l); T=X*X+ l*l+M *M;
  XDrawString(e,z,k ,20,380,f,17); D=v/l*15; i+=(B *l-M*r -X*Z)*_; for(; XPending(e); u *=CS!=N){
                                  XEvent z; XNextEvent(e ,&z);
                                     ++*((N=XLookupKeysym
                                       (&z.xkey,0))-IT?
                                       N-LT? UP-N?& E:&
                                       J:& u: &h); --*(
                                       DN -N? N-DT ?N==
                                       RT?&u: & W:&h:&J
                                        ); } m=15*F/l;
                                        c+=(I=M/ l,l*H
                                        +I*M+a*X)*_; H
                                        =A*r+v*X-F*l+(
                                        E=.1+X*4.9/l,t
                                        =T*m/32-I*T/24
                                         )/S; K=F*M+(
                                         h* 1e4/l-(T+
                                         E*5*T*E)/3e2
                                         )/S-X*d-B*A;
                                         a=2.63 /l*d;
                                         X+=( d*l-T/S
                                          *(.19*E +a
                                          *.64+J/1e3
                                          )-M* v +A*
                                          Z)*_; l +=
                                          K *_; W=d;
                                          sprintf(f,
                                          "%5d  %3d"
                                          "%7d",p =l
                                         /1.7,(C=9E3+
                                  O*57.3)%0550,(int)i); d+=T*(.45-14/l*
                                 X-a*130-J* .14)*_/125e2+F*_*v; P=(T*(47
                                 *I-m* 52+E*94 *D-t*.38+u*.21*E) /1e2+W*
                                 179*v)/2312; select(p=0,0,0,0,&G); v-=(
                                  W*F-T*(.63*m-I*.086+m*E*19-D*25-.11*u
                                   )/107e2)*_; D=cos(o); E=sin(o); } }
```

```
#include                                  <math.h>
#include                                <sys/time.h>
#include                                <X11/Xlib.h>
#include                               <X11/keysym.h>
                                       double L ,o ,P
                                      ,_=dt,T,Z,D=1,d,
                                      s[999],E,h= 8,I,
                                      J,K,w[999],M,m,O
                                     ,n[999],j=33e-3,i=
                                     1E3,r,t, u,v ,W,S=
                                     74.5,l=221,X=7.26,
                                     a,B,A=32.2,c, F,H;
                                      int N,q, C, y,p,U;
                                    Window z; char f[52]
                                  ; GC k; main(){ Display*e=
 XOpenDisplay( 0); z=RootWindow(e,0); for (XSetForeground(e,k=XCreateGC (e,z,0,0),BlackPixel(e,0))
; scanf("%lf%lf%lf",y +n,w+y, y+s)+1; y ++); XSelectInput(e,z= XCreateSimpleWindow(e,z,0,0,400,400,
0,0,WhitePixel(e,0) ),KeyPressMask); for(XMapWindow(e,z); ; T=sin(O)){ struct timeval G={ 0,dt*1e6}
; K= cos(j); N=1e4; M+= H*_; Z=D*K; F+=_*P; r=E*K; W=cos( O); m=K*W; H=K*T; O+=D*_*F/ K+d/K*E*_; B=
sin(j); a=B*T*D-E*W; XClearWindow(e,z); t=T*E+ D*B*W; j+=d* *D- _*F*E; P=W*E*B-T*D; for (o+=(I=D*W+E
*T*B,E*d/K *B+v+B/K*F*D)*_; p<y; ){ T=p[s]+i; E=c-p[w]; D=n[p]-L; K=D*m-B*T-H*E; if(p [n]+w[ p]+p[s
]== 0|K <fabs(W=T*r-I*E +D*P) |fabs(D=t *D+Z *T-a *E)> K)N=1e4; else{ q=W/K *4E2+2e2; C= 2E2+4e2/ K
 *D; N-1E4&& XDrawLine(e ,z,k,N ,U,q,C); N=q; U=C; } ++p; } L+=_* (X*t +P*M+m*l); T=X*X+ l*l+M *M;
  XDrawString(e,z,k ,20,380,f,17); D=v/l*15; i+=(B *l-M*r -X*Z)*_; for(; XPending(e); u *=CS!=N){
                                  XEvent z; XNextEvent(e ,&z);
                                      ++*((N=XLookupKeysym
                                        (&z.xkey,0))-IT?
                                        N-LT? UP-N?& E:&
                                        J:& u: &h); --*(
                                        DN -N? N-DT ?N==
                                        RT?&u: & W:&h:&J
                                         ); } m=15*F/l;
                                         c+=(I=M/ l,l*H
                                         +I*M+a*X)*_; H
                                         =A*r+v*X-F*l+(
                                         E=.1+X*4.9/l,t
                                         =T*m/32-I*T/24
                                          )/S; K=F*M+(
                                          h* 1e4/l-(T+
                                          E*5*T*E)/3e2
                                          )/S-X*d-B*A;
                                          a=2.63 /l*d;
                                          X+=( d*l-T/S
                                           *(.19*E +a
                                           *.64+J/1e3
                                           )-M* v +A*
                                           Z)*_; l +=
                                           K *_; W=d;
                                           sprintf(f,
                                           "%5d  %3d"
                                           "%7d",p =l
                                           /1.7,(C=9E3+
                                  O*57.3)%0550,(int)i); d+=T*(.45-14/l*
                                 X-a*130-J* .14)*_/125e2+F* *v; P=(T*(47
                                 *I-m* 52+E*94 *D-t*.38+u*.21*E) /1e2+W*
                                 179*v)/2312; select(p=0,0,0,0,&G); v-=(
                                  W*F-T*(.63*m-I*.086+m*E*19-D*25-.11*u
                                   )/107e2)*_; D=cos(o); E=sin(o); } }
```

A fully functional

# flight simulator.

“Best of Show” in the 1998
International Obfuscated C Code Challenge.




http://blog.aerojockey.com/post/iocccsim
http://www.de.ioccc.org/years.html#1998

```python
import random

class Citizen:
    def __init__(self, sex, age):
        self.sex = sex
        self.age = age

    def __repr__(self):
        return 'Citizen is a {} {} years old'.format(self.sex, self.age)

    def grow(self):
        self.age += 1

    def getAge(self):
        return self.age


class City:
    def __init__(self, population):
        self.pop = population

    def __repr__(self):
        return 'City population is {}, random: {}'.format(
            len(self.pop), self.pop[int(random.random() * len(self.pop))])

    def age(self):
        pop = self.pop
        for citizen in pop:
            citizen.grow()
            # citizens die
            if citizen.getAge() > 80:
                self.pop.remove(citizen)
                self.breed()  # creates a new citizen

    def breed(self):
        sex = 'male' if random.random() >= 0.5 else 'female'
        age = random.randint(18, 30)
        self.pop.append(Citizen(sex, age))
```

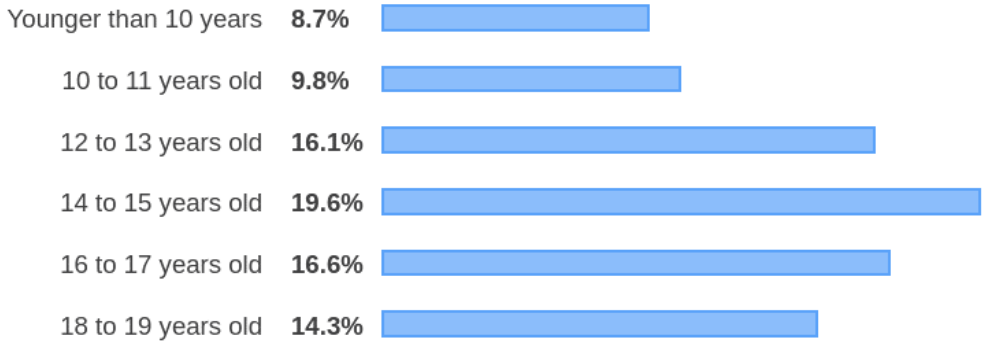
https://www.freecodecamp.org/news/make-your-code-understandable-by-using-abstraction-4b522307130c/

# Why Should You Care?

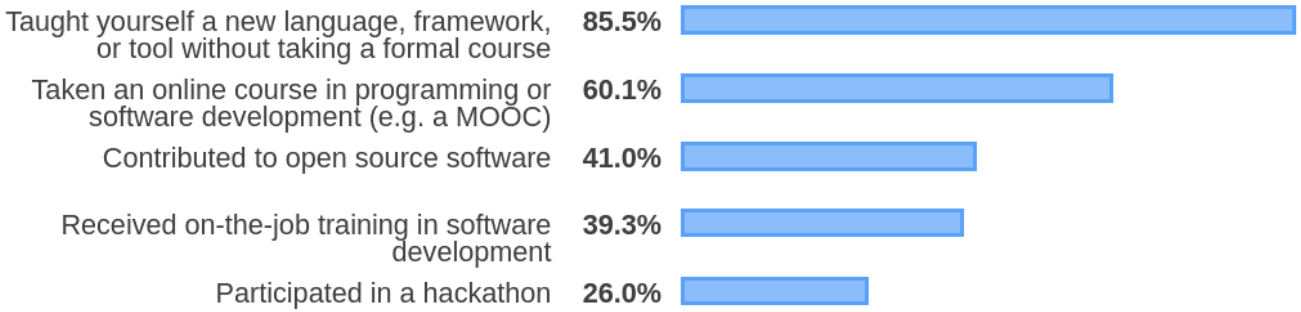
Barrier to entry the world of programming has never been lower before

Free access to tutorials, courses, MOOCs, social support platforms, ….

Programming languages increasingly accessible also for non computer-scientists




"How old were you when you wrote your first line of code or program?" 87,634 responses

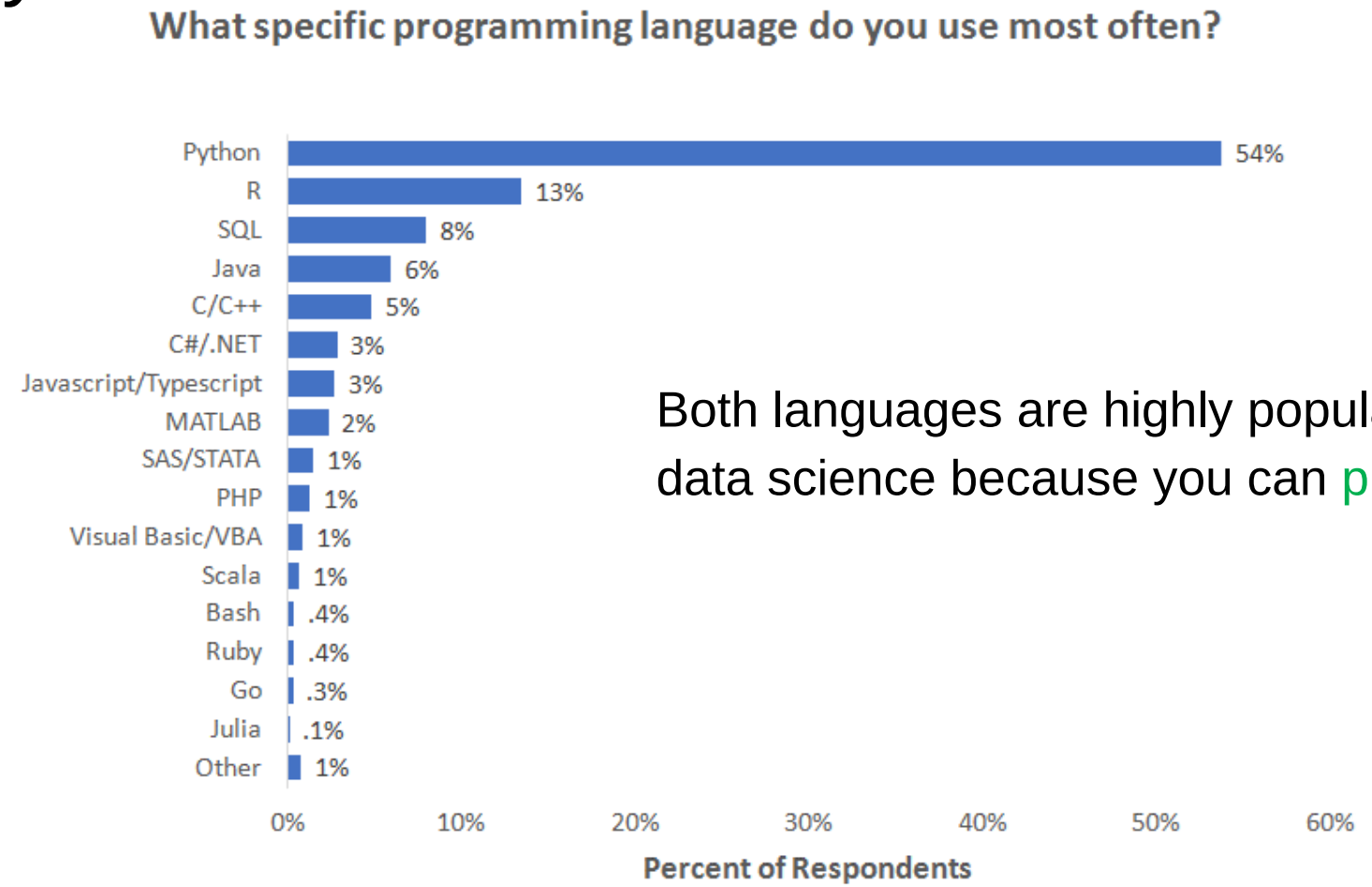



Education in Programming

84,260 responses

https://insights.stackoverflow.com/survey/2019

# Why R and Python?




Both languages are highly popular in the field of data science because you can program with data

Note: Data are from the 2018 Kaggle ML and Data Science Survey. You can learn more about the study here: http://www.kaggle.com/kaggle/kaggle-survey-2018.
A total of 23859 respondents completed the survey; the percentages in the graph are based on a total of 15222 respondents who provided an answer to this question.

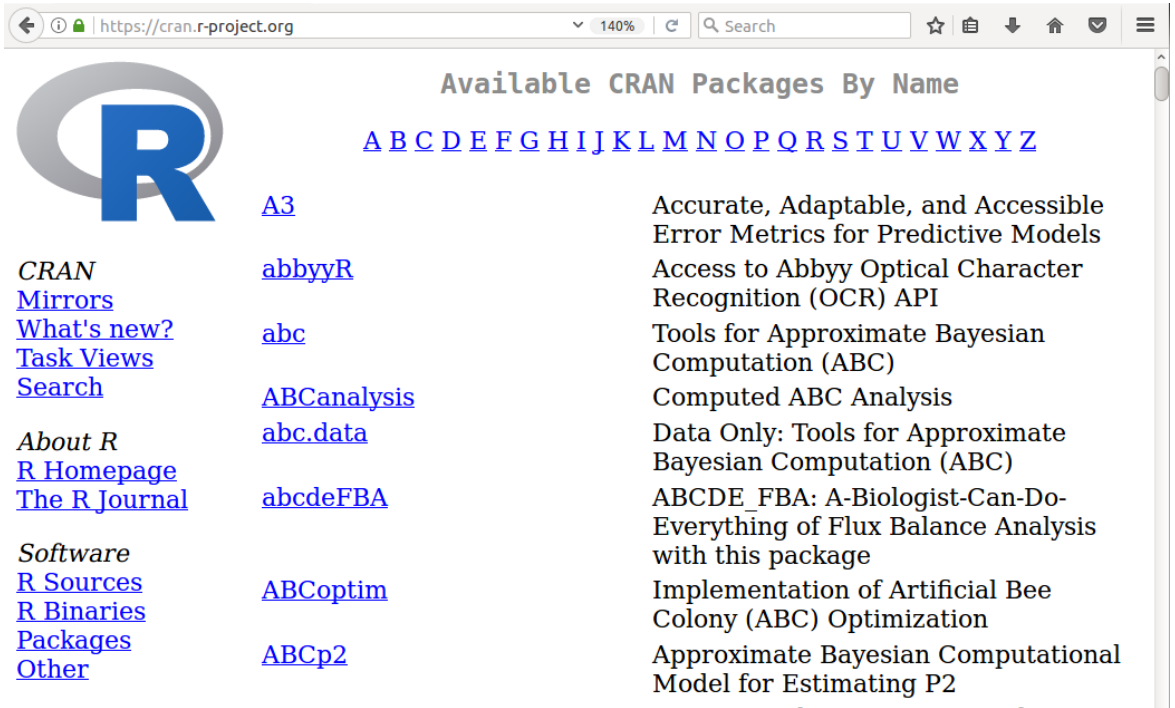
https://businessoverbroadway.com/2019/01/13/programming-languages-most-used-and-recommended-by-data-scientists/

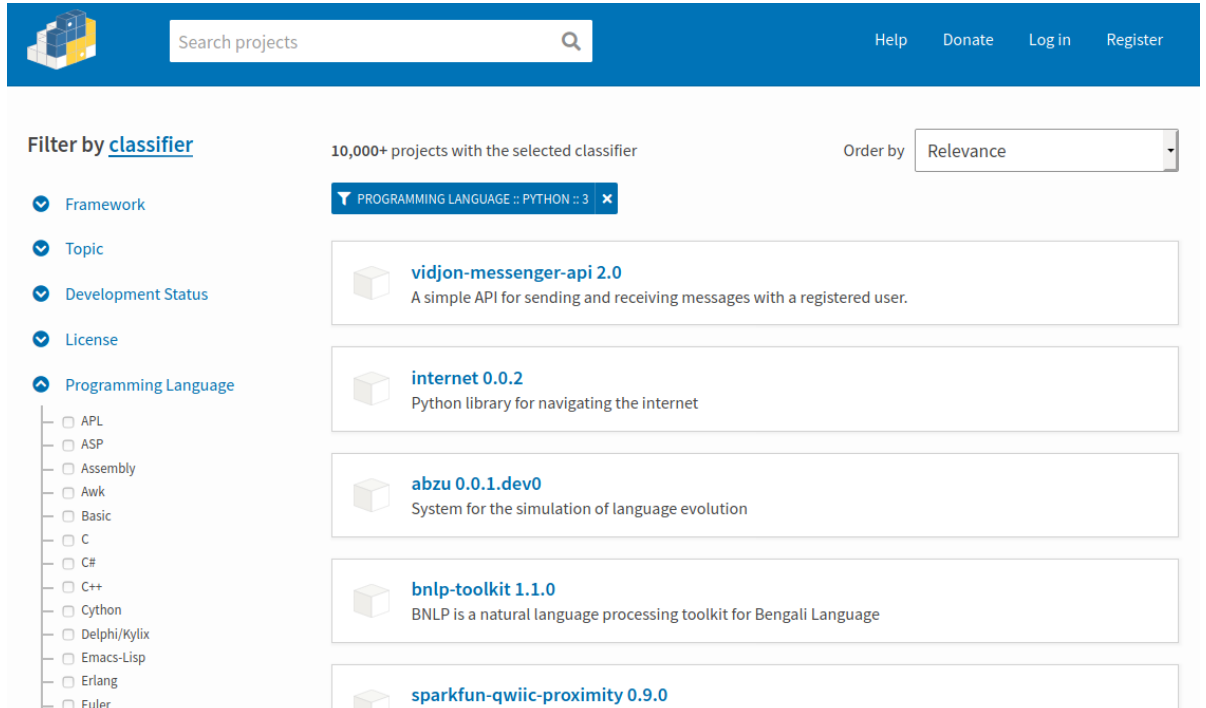
# Extending R and Python

Packages encapsulate complex or new functionality

Provide easy access to this functionality

One of the reasons for the popularity of open-source programming languages



# The Goal of This Tutorial

Create a package for R and Python that

We reach that goal by

# The Goal of This Tutorial

Create a package for R and Python that

… is suitable for big amount of data

We reach that goal by

# The Goal of This Tutorial

Create a package for R and Python that

- … is suitable for big amount of data
- … shares a common codebase

We reach that goal by

# The Goal of This Tutorial

Create a package for R and Python that

- … is suitable for big amount of data
- … shares a common codebase
- … requires as little additional code as possible

We reach that goal by

# The Goal of This Tutorial

Create a package for R and Python that

- … is suitable for big amount of data
- … shares a common codebase
- … requires as little additional code as possible

We reach that goal by

- … implementing the core functionality in C++

# The Goal of This Tutorial

Create a package for R and Python that

- … is suitable for big amount of data
- … shares a common codebase
- … requires as little additional code as possible

We reach that goal by

- … implementing the core functionality in C++
- … use Boost.Python to compile that code for Python

# The Goal of This Tutorial

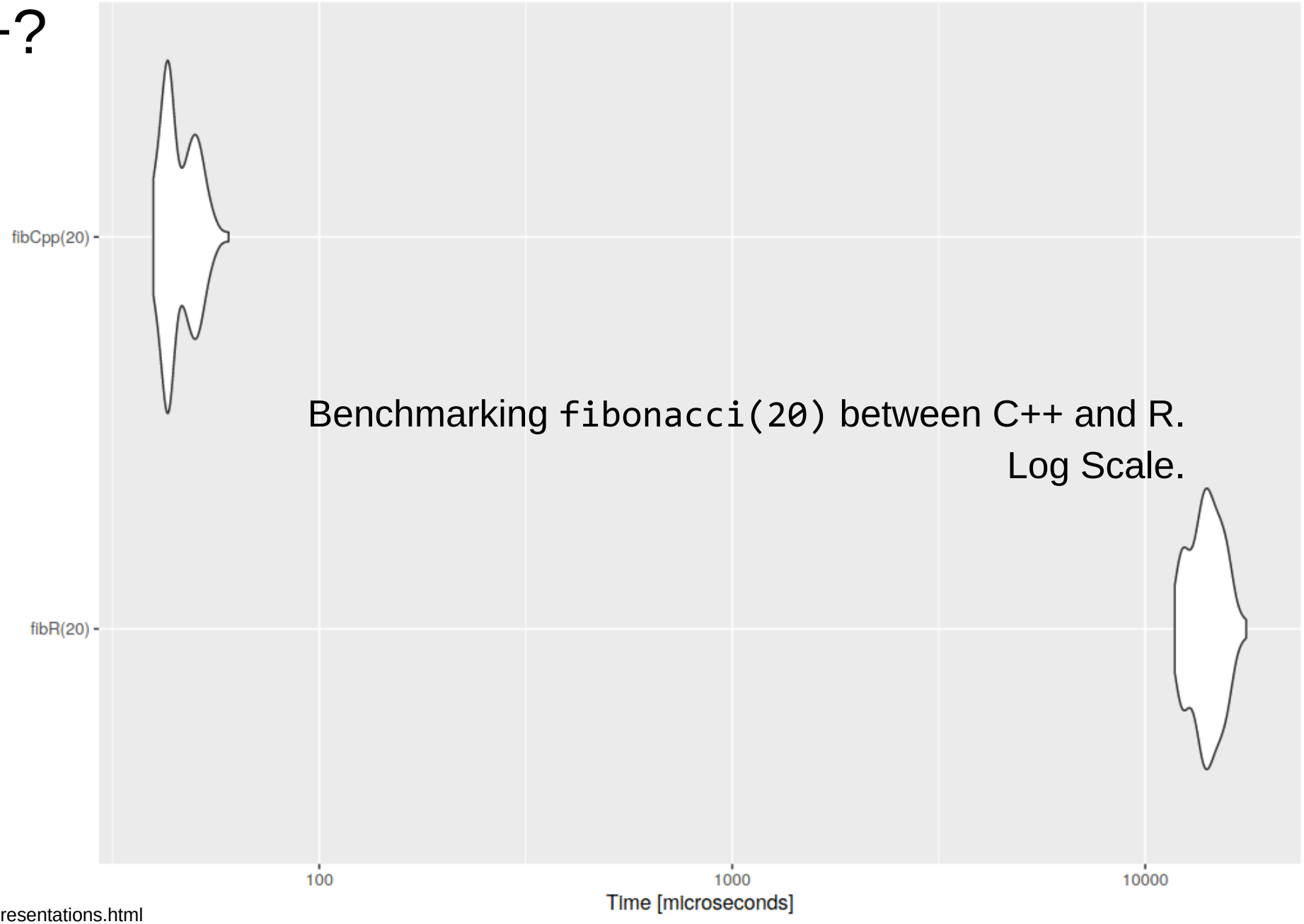
Create a package for R and Python that

- … is suitable for big amount of data
- … shares a common codebase
- … requires as little additional code as possible

We reach that goal by

- … implementing the core functionality in C++
- … use Boost.Python to compile that code for Python
- … use Rcpp to compile that code for R

# Why C++?




Benchmarking `fibonacci(20)` between C++ and R.
Log Scale.

https://dirk.eddelbuettel.com/presentations.html

Let’s get started!

https://github.com/TwlyY29/openmunich_2019

# Requirements

Clone https://github.com/TwlyY29/openmunich_2019

Code was tested using Ubuntu 18.04.1 LTS 64 Bit, Boost 1.65.1 & R 3.6.1

```
sudo apt install build-essential libboost1.65-all-dev r-base
Rscript --slave --no-save --no-restore-history \
 -e 'install.packages(c("Rcpp", "RUnit"), dependencies = T)'
```

# A Simple Example

`fancyalgorithms/fancy_functions.hpp`

```
int fancy_increment(int i){
  srand (time(NULL));
  return i + (rand() % 10 + 1);
}
void fancy_increment_container(std::vector<int>& more_i){
  srand (time(NULL));
  for (auto& i:more_i)
    i += (rand() % 10 + 1);
}
```

```
git checkout 000-get_started
cd core/
make run
```

```
class FancyObject{
  private:
    int min, max;

  public:
    FancyObject():min(1),max(10)
    {
      srand(time(NULL));
    }
    FancyObject(int min, int max):min(min), max(max)
    {
      srand(time(NULL));
    }
    int random_increment(int nr)
    {
      return nr + (rand() % max + min);
    }

    int get_min(){return min;}
    void set_min(int new_min){min=new_min;}

    int get_max(){return max;}
    void set_max(int new_max){max=new_max;}

};
```

# First Python Package

```python
import fancymodule.fancymodule as fancy
i = fancy.fancy_increment(1)
```

```
#include "fancyalgorithms/fancy_functions.hpp"
#include <boost/python.hpp>
namespace py = boost::python;
BOOST_PYTHON_MODULE(fancymodule)
{
  py::def("fancy_increment", fancy_increment);
}
```

```
git checkout 100-simple_python
cd pkg-python/
make python3
cd test
python3 main.py
```

# First Python Package

```python
import fancymodule fancymodule as fancy
i = fancy.fancy_increment(1)
```

```
#include "fancyalgorithms/fancy_functions.hpp"
#include <boost/python.hpp>
namespace py = boost::python;
BOOST_PYTHON_MODULE(fancymodule)
{
  py::def("fancy_increment", fancy_increment);
}
```

```
git checkout 100-simple_python
cd pkg-python/
make python3
cd test
python3 main.py
```

# First Python Package

```python
import fancymodule.fancymodule as fancy
i = fancy.fancy_increment(1)
```

```
#include "fancyalgorithms/fancy_functions.hpp"
#include <boost/python.hpp>
namespace py = boost::python;
BOOST_PYTHON_MODULE(fancymodule)
{
  py::def("fancy_increment", fancy_increment);
}
```

```
git checkout 100-simple_python
cd pkg-python/
make python3
cd test
python3 main.py
```

# Exposing a C++ Class to Python

```python
import fancymodule.fancymodule as fancy
f = fancy.FancyObject()
i = f.random_increment(1)
f2 = fancy.FancyObject(-2,2)
i2 = f2.random_increment(0)
```

```cpp
BOOST_PYTHON_MODULE(fancymodule)
{
  py::def("fancy_increment", fancy_increment);

  py::class_<FancyObject>("FancyObject", py::init<>())
    .def(py::init<int,int>())
    .def("random_increment", &FancyObject::random_increment);

}
```

```
git checkout 200-a_class
cd pkg-python/
make python3
cd test
python3 main.py
```

# Exposing a C++ Class to Python

```python
import fancymodule.fancymodule as fancy
f = fancy.FancyObject()
i = f.random_increment(1)
f2 = fancy.FancyObject(-2,2)
i2 = f2.random_increment(0)
```

```cpp
BOOST_PYTHON_MODULE(fancymodule)
{
  py::def("fancy_increment", fancy_increment);

  py::class_<FancyObject>("FancyObject", py::init<>())
    .def(py::init<int,int>())
    .def("random_increment", &FancyObject::random_increment);

}
```

```
git checkout 200-a_class
cd pkg-python/
make python3
cd test
python3 main.py
```

# Exposing a C++ Class to Python

```python
import fancymodule.fancymodule as fancy
f = fancy.FancyObject()
i = f.random_increment(1)
f2 = fancy.FancyObject(-2,2)
i2 = f2.random_increment(0)
```

```cpp
BOOST_PYTHON_MODULE(fancymodule)
{
  py::def("fancy_increment", fancy_increment);

  py::class_<FancyObject>("FancyObject", py::init<>())
    .def(py::init<int,int>())
    .def("random_increment", &FancyObject::random_increment);

}
```

```
git checkout 200-a_class
cd pkg-python/
make python3
cd test
python3 main.py
```

# Exposing a C++ Class to Python

```python
import fancymodule.fancymodule as fancy
f = fancy.FancyObject()
i = f.random_increment(1)
f2 = fancy.FancyObject(-2,2)
i2 = f2.random_increment(0)
```

```cpp
BOOST_PYTHON_MODULE(fancymodule)
{
  py::def("fancy_increment", fancy_increment);

  py::class_<FancyObject>("FancyObject", py::init<>())
    .def(py::init<int,int>())
    .def("random_increment", &FancyObject::random_increment);

}
```

```
git checkout 200-a_class
cd pkg-python/
make python3
cd test
python3 main.py
```

# Exposing a C++ Class to Python

```python
import fancymodule.fancymodule as fancy
f = fancy.FancyObject()
i = f.random_increment(1)
f2 = fancy.FancyObject(-2,2)
i2 = f2.random_increment(0)
```

```cpp
BOOST_PYTHON_MODULE(fancymodule)
{
  py::def("fancy_increment", fancy_increment);

  py::class_<FancyObject>("FancyObject", py::init<>())
    .def(py::init<int,int>())
    .def("random_increment", &FancyObject::random_increment);

}
```

```
git checkout 200-a_class
cd pkg-python/
make python3
cd test
python3 main.py
```

At this point you can expose C++ functions which take `FancyObject&` as parameters and call them from python. The python program takes ownership of a `FancyObject` and can pass references.

# Class Members Become Object Properties

```python
import fancymodule.fancymodule as fancy
f = fancy.FancyObject()
f.min = -2
f.max = 2
```

```
py::class_<FancyObject>("FancyObject", py::init<>())
  .def(py::init<int,int>())
  .add_property("min", &FancyObject::get_min, &FancyObject::set_min)
  .add_property("max", &FancyObject::get_max, &FancyObject::set_max)
  .def("random_increment", &FancyObject::random_increment);
```

```
git checkout 300-class_members
cd pkg-python/
make python3
cd test
python3 main.py
```

# Class Members Become Object Properties

```
import fancymodule.fancymodule as fancy
f = fancy.FancyObject()
f.min = -2
f.max = 2
```

```
py::class_<FancyObject>("FancyObject", py::init<>())
  .def(py::init<int,int>())
  .add_property("min", &FancyObject::get_min, &FancyObject::set_min)
  .add_property("max", &FancyObject::get_max, &FancyObject::set_max)
  .def("random_increment", &FancyObject::random_increment);
```

```
git checkout 300-class_members
cd pkg-python/
make python3
cd test
python3 main.py
```

# C++ Containers…

```python
import fancymodule.fancymodule as fancy
numbers = fancy.IntList()
numbers[:] = [0,1,2,3,4]
fancy.fancy_increment_container(numbers)
```

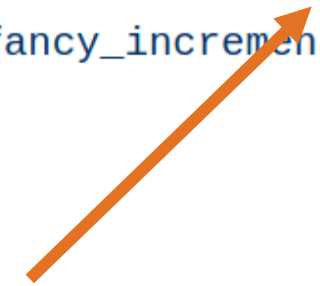
```
py::class_< std::vector<int> > ("IntList")
  .def(py::vector_indexing_suite< std::vector<int> >());
py::def("fancy_increment_container", fancy_increment_container);
```

```
git checkout 400-container
cd pkg-python/
make python3
cd test
python3 main.py
```

# C++ Containers…

```python
import fancymodule.fancymodule as fancy
numbers = fancy.IntList()
numbers[:] = [0,1,2,3,4]
fancy.fancy_increment_container(numbers)
```

```
py::class_< std::vector<int> > ("IntList")
  .def(py::vector_indexing_suite< std::vector<int> >());
py::def("fancy_increment_container", fancy_increment_container);
```



```
git checkout 400-container
cd pkg-python/
make python3
cd test
python3 main.py
```

This enables iterators, accessors, deletion, and addition of elements such that `IntList` behaves like a typical python list.

# C++ Containers…

```python
import fancymodule.fancymodule as fancy
numbers = fancy.IntList()
numbers[:] = [0,1,2,3,4]
fancy.fancy_increment_container(numbers)
```

This doesn't look very pythonic…
But it prevents copying the list and instead allows for passing its reference to `fancy_increment_container`!

```
py::class_< std::vector<int> > ("IntList")
  .def(py::vector_indexing_suite< std::vector<int> >());
py::def("fancy_increment_container", fancy_increment_container);
```

```
git checkout 400-container
cd pkg-python/
make python3
cd test
python3 main.py
```

This enables iterators, accessors, deletion, and addition of elements such that `IntList` behaves like a typical python list.

Building python packages with Boost.Python:

- Expose Functions and Classes
- Supports References and Pointers
- Exports C++ Iterators as Python Iterators
- and much more…

# Moving on…

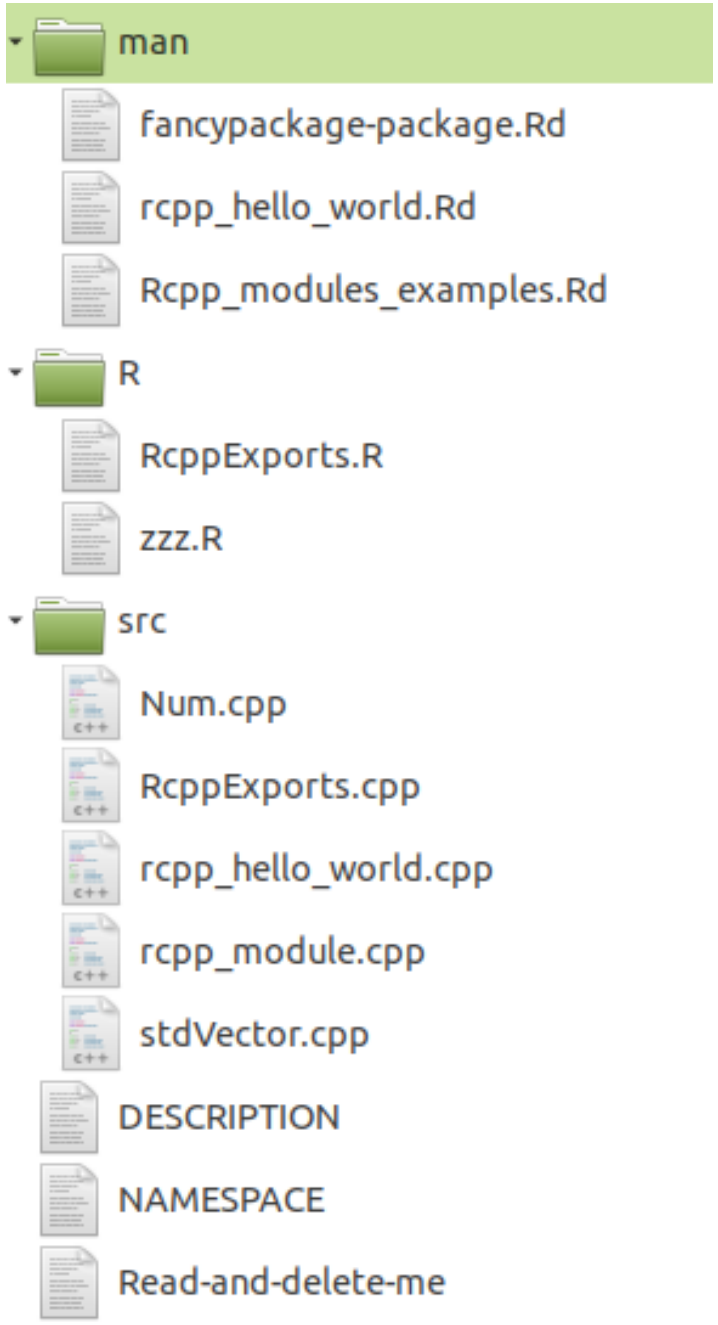
# A Simple R Package

We need Rcpp to create a standard package:

```
library(Rcpp)
Rcpp.package.skeleton(name = "fancypackage",
  author = "Mirco Schoenfeld",
  email = "twlyy29@gmail.com",
  module=T)
```

This standard package also serves
as an “applied documentation”

```
git checkout 500-getting_started_r
cd pkg-r/
make init
```

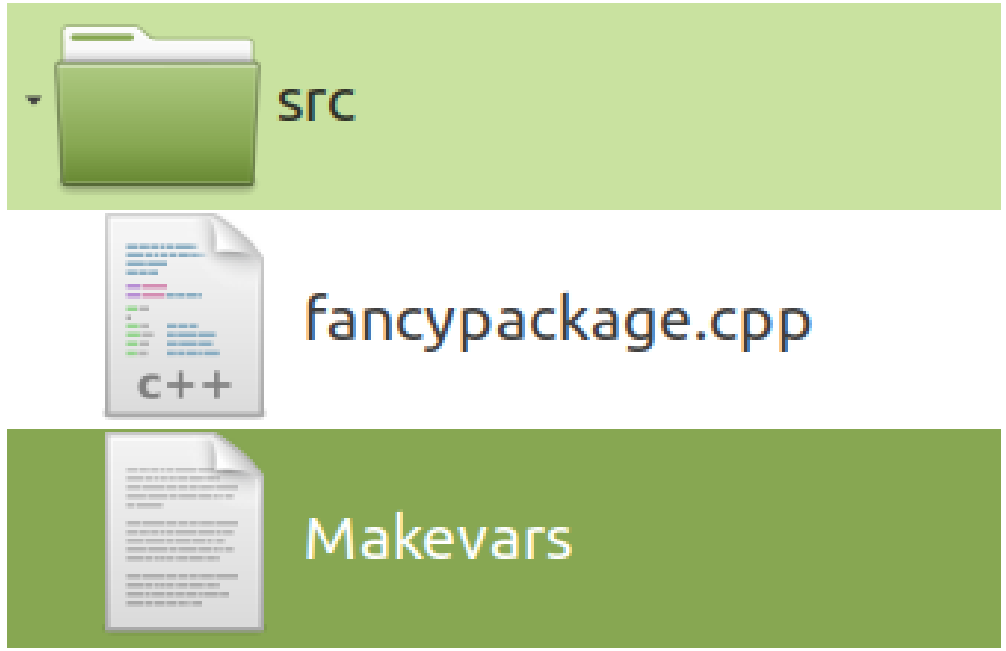


# Our First R Package

Again, we only need one source-file to bridge R and C++: `fancypackage.cpp`

Use `Makevars` or `Makevars.win` to include the library header files:

```
PKG_CXXFLAGS = -I"/path/to/lib"
```




```
git clean -f -d
git checkout 600-first_function
```

# Our First R Package

```cpp
#include <Rcpp.h>
#include "fancyalgorithms/fancy_functions.hpp"

using namespace Rcpp;

RCPP_MODULE(fancy){

  function("fancy_increment" , &fancy_increment  , "documentation for fancy_increment ");

}
```

```
cd pkg-r/
make install
Rscript test.R
```

```
library(fancypackage)
fancy_increment(3)
```

# Our First R Package

```cpp
#include <Rcpp.h>
#include "fancyalgorithms/fancy_functions.hpp"

using namespace Rcpp;

RCPP_MODULE(fancy){

  function("fancy_increment", &fancy_increment  , "documentation for fancy_increment ");

}
```

```
cd pkg-r/
make install
Rscript test.R
```

```r
library(fancypackage)
fancy_increment(3)
```

# Our First R Package

```
#include <Rcpp.h>
#include "fancyalgorithms/fancy_functions.hpp"

using namespace Rcpp;

RCPP_MODULE(fancy){

  function("fancy_increment" , &fancy_increment  , "documentation for fancy_increment ");

}
```

Don’t forget to add this line to man/zzz.R

```
loadModule("fancy", TRUE)
```

```
cd pkg-r/
make install
Rscript test.R
```

```
library(fancypackage)
fancy_increment(3)
```

# Our First R Package

```
#include <Rcpp.h>
#include "fancyalgorithms/fancy_functions.hpp"

using namespace Rcpp;

RCPP_MODULE(fancy){

  function("fancy_increment" , &fancy_increment  , "documentation for fancy_increment ");

}
```

Don’t forget to add this line to man/zzz.R

```
loadModule("fancy", TRUE)
```

```
cd pkg-r/
make install
Rscript test.R
```

```
library(fancypackage)
fancy_increment(3)
```

# Exposing C++ Classes to R

```
RCPP_MODULE(fancy){

  function("fancy_increment" , &fancy_increment  , "documentation for fancy_increment ");

  class_<FancyObject>("FancyObject")
    .constructor()
    .constructor<int,int>()

    .method("random_increment", &FancyObject::random_increment , "increment by random number")

    ;
}
```

```
git checkout 700-a_class_in_r
cd pkg-r
make install
Rscript test.R
```

```
library(fancypackage)
fo <- new(FancyObject)
inc <- fo$random_increment(3)

fo <- new(FancyObject,2,5)
inc <- fo$random_increment(3)
```

# Exposing C++ Classes to R

```
RCPP_MODULE(fancy){

  function("fancy_increment" , &fancy_increment  , "documentation for fancy_increment ");

  class_<FancyObject>("FancyObject")
    .constructor()
    .constructor<int,int>()

    .method("random_increment", &FancyObject::random_increment , "increment by random number")

    ;
}
```

```
git checkout 700-a_class_in_r
cd pkg-r
make install
Rscript test.R
```

```
library(fancypackage)
fo <- new(FancyObject)
inc <- fo$random_increment(3)

fo <- new(FancyObject,2,5)
inc <- fo$random_increment(3)
```

# Exposing C++ Classes to R

```
RCPP_MODULE(fancy){

  function("fancy_increment" , &fancy_increment  , "documentation for fancy_increment ");

  class_<FancyObject>("FancyObject")
    .constructor()
    .constructor<int,int>()

    .method("random_increment", &FancyObject::random_increment , "increment by random number")

    ;
}
```

```
git checkout 700-a_class_in_r
cd pkg-r
make install
Rscript test.R
```

```
library(fancypackage)
fo <- new(FancyObject)
inc <- fo$random_increment(3)

fo <- new(FancyObject,2,5)
inc <- fo$random_increment(3)
```

# Exposing C++ Classes to R

```
RCPP_MODULE(fancy){

  function("fancy_increment" , &fancy_increment  , "documentation for fancy_increment ");

  class_<FancyObject>("FancyObject")
    .constructor()
    .constructor<int,int>()

    .method("random_increment", &FancyObject::random_increment , "increment by random number")

    ;
}
```

```
git checkout 700-a_class_in_r
cd pkg-r
make install
Rscript test.R
```

```
library(fancypackage)
fo <- new(FancyObject)
inc <- fo$random_increment(3)

fo <- new(FancyObject,2,5)
inc <- fo$random_increment(3)
```

# Class Members in R

```
class_<FancyObject>("FancyObject")
  .constructor()
  .constructor<int,int>()

  .method("random_increment", &FancyObject::random_increment , "increment by random number")

  .property("min", &FancyObject::get_min, &FancyObject::set_min)
  .property("max", &FancyObject::get_max, &FancyObject::set_max)
  ;
```

```
git checkout 800-class_members
cd pkg-r
make install
Rscript test.R
```

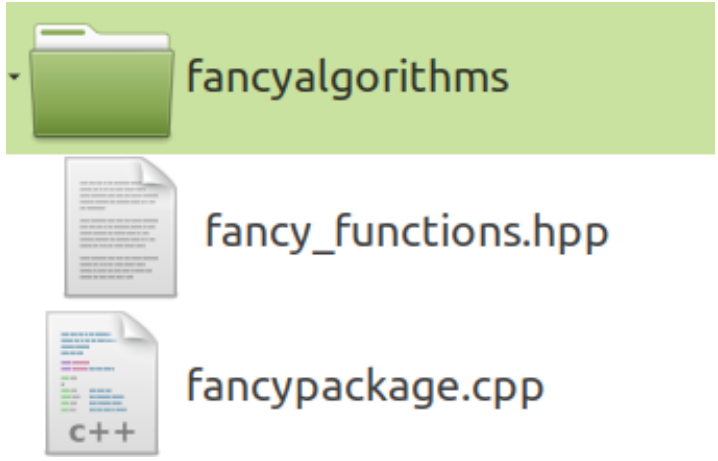
```
library(fancypackage)
fo <- new(FancyObject)
fo$min <- 0
fo$max <- 1
```

# Containers in R

To support passing references and modifying R-datastructures in-place, we need to change the signature of `fancy_increment_container` slightly:

```
void fancy_increment_container(std::vector<int>& more_i){
  srand (time(NULL));
  for (auto& i:more_i)
    i += (rand() % 10 + 1);
}
```

```
git reset --hard
git checkout 900-container_r
cd pkg-r
make install
Rscript test.R
```




The library has been copied into the package and `Makevars` has been removed.

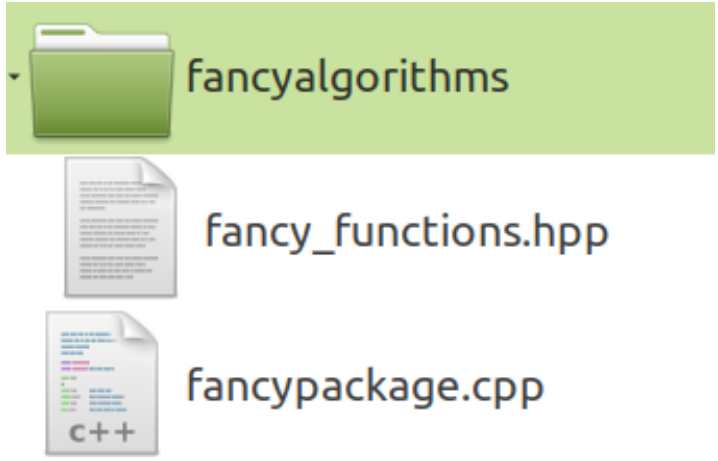
# Containers in R

To support passing references and modifying R-datastructures in-place, we need to change the signature of `fancy_increment_container` slightly:

```
void fancy_increment_container(std::vector<int>& more_i){
  srand (time(NULL));
  for (auto& i:more_i)
    i += (rand() % 10 + 1);
}
```

```
git reset --hard
git checkout 900-container_r
cd pkg-r
make install
Rscript test.R
```

This might be necessary in case you adapted Makevars to point to the library files…



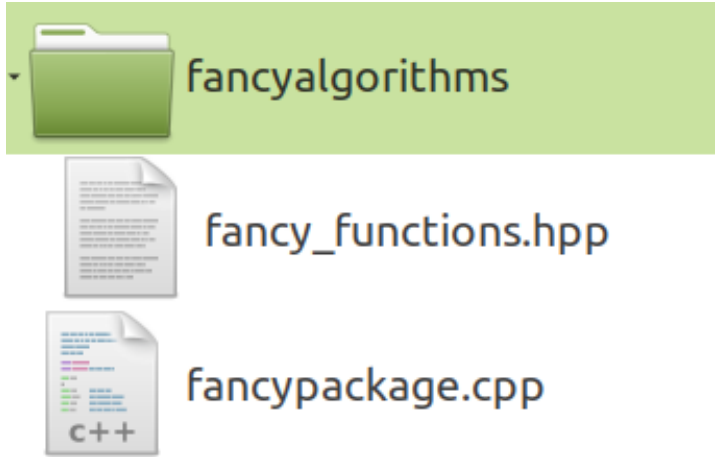

The library has been copied into the package and `Makevars` has been removed.

# Containers in R

To support passing references and modifying R-datastructures in-place, we need to change the signature of `fancy_increment_container` slightly:

```
void fancy_increment_container(std::vector<int>& more_i){
  srand (time(NULL));
  for (auto& i:more_i)
    i += (rand() % 10 + 1);
}
```

```
git reset --hard
git checkout 900-container_r
cd pkg-r
make install
Rscript test.R
```



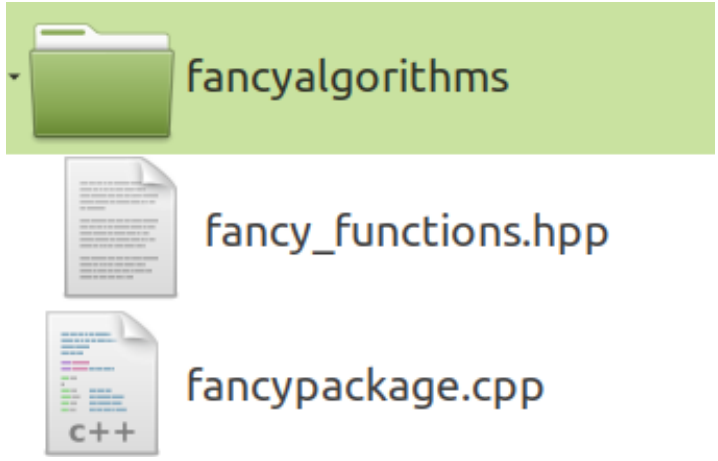

The library has been copied into the package and `Makevars` has been removed.

# Containers in R

To support passing references and modifying R-datastructures in-place, we need to change the signature of `fancy_increment_container` slightly:

```
void fancy_increment_container(Rcpp::NumericVector more_i){
  srand (time(NULL));
  for (auto& i:more_i)
    i += (rand() % 10 + 1);
}
```

```
git reset --hard
git checkout 900-container_r
cd pkg-r
make install
Rscript test.R
```




The library has been copied into the package and `Makevars` has been removed.

Building R packages with Rcpp:

- Seamless integration of C++ into R
- Standard R types are mapped to corresponding C++ types
- Around 45% of all packages on CRAN rely on Rcpp
- Several flavors exist (RcppArmadillo, RcppParallel, RcppEigen…)

# What we Saw Today

Create a package for R and Python that

We reach that goal by

# What we Saw Today

Create a package for R and Python that

… is suitable for big amount of data

We reach that goal by

# What we Saw Today

Create a package for R and Python that

… is suitable for big amount of data

… shares a common codebase

We reach that goal by

# What we Saw Today

Create a package for R and Python that

… is suitable for big amount of data

… shares a common codebase

… requires as little additional code as possible

We reach that goal by

# What we Saw Today

Create a package for R and Python that

… is suitable for big amount of data

… shares a common codebase

… requires as little additional code as possible

We reach that goal by

… implementing the core functionality in C++

# What we Saw Today

Create a package for R and Python that

- … is suitable for big amount of data
- … shares a common codebase
- … requires as little additional code as possible

We reach that goal by

- … implementing the core functionality in C++
- … use Boost.Python to compile that code for Python

# What we Saw Today

Create a package for R and Python that

- … is suitable for big amount of data
- … shares a common codebase
- … requires as little additional code as possible

We reach that goal by

- … implementing the core functionality in C++
- … use Boost.Python to compile that code for Python
- … use Rcpp to compile that code for R

# Helpful Resources

Boost.Python

Rcpp

# Helpful Resources

Boost.Python

- https://github.com/TNG/boost-python-examples

Rcpp

# Helpful Resources

Boost.Python

- https://github.com/TNG/boost-python-examples
- https://wiki.python.org/moin/boost.python/GettingStarted

Rcpp

# Helpful Resources

Boost.Python

- https://github.com/TNG/boost-python-examples
- https://wiki.python.org/moin/boost.python/GettingStarted
-

Rcpp

# Helpful Resources

Boost.Python

- https://github.com/TNG/boost-python-examples
- https://wiki.python.org/moin/boost.python/GettingStarted
-
-

Rcpp

# Helpful Resources

Boost.Python

- https://github.com/TNG/boost-python-examples
- https://wiki.python.org/moin/boost.python/GettingStarted
-
-
-

Rcpp

# Helpful Resources

Boost.Python

- https://github.com/TNG/boost-python-examples
- https://wiki.python.org/moin/boost.python/GettingStarted
-
-
-
-

Rcpp

# Helpful Resources

Boost.Python

- https://github.com/TNG/boost-python-examples
- https://wiki.python.org/moin/boost.python/GettingStarted
-
-
-
-
- https://www.boost.org/doc/libs/1_46_1/libs/python/doc/tutorial/doc/html/index.html

Rcpp

# Helpful Resources

Boost.Python

- https://github.com/TNG/boost-python-examples
- https://wiki.python.org/moin/boost.python/GettingStarted
-
-
-
-
- https://www.boost.org/doc/libs/1_46_1/libs/python/doc/tutorial/doc/html/index.html

(really, Boost documentation is no fun to read)

Rcpp

# Helpful Resources

Boost.Python

- https://github.com/TNG/boost-python-examples
- https://wiki.python.org/moin/boost.python/GettingStarted
-
-
-
-
- https://www.boost.org/doc/libs/1_46_1/libs/python/doc/tutorial/doc/html/index.html

(really, Boost documentation is no fun to read)

Rcpp

- http://www.rcpp.org/

# Helpful Resources

Boost.Python

- https://github.com/TNG/boost-python-examples
- https://wiki.python.org/moin/boost.python/GettingStarted
-
-
-
-
- https://www.boost.org/doc/libs/1_46_1/libs/python/doc/tutorial/doc/html/index.html

(really, Boost documentation is no fun to read)

Rcpp

- http://www.rcpp.org/
- https://gallery.rcpp.org/

# Helpful Resources

Boost.Python

- https://github.com/TNG/boost-python-examples
- https://wiki.python.org/moin/boost.python/GettingStarted
-
-
-
-
- https://www.boost.org/doc/libs/1_46_1/libs/python/doc/tutorial/doc/html/index.html

(really, Boost documentation is no fun to read)

Rcpp

- http://www.rcpp.org/
- https://gallery.rcpp.org/
- http://dirk.eddelbuettel.com/presentations/

# Thanks.

@TwlyY29
mirco.schoenfeld@tum.de