Biostatistics 615 - Statistical Computing

Lecture 17
Write R packages

Jian Kang

Nov 24, 2015

Introduction

- The R packaging system has been one of the key factors of the overall success of the R project
- Packages allow for easy, transparent and cross-platform extension of the R base system.
- R packages are (after a short learning phase) a comfortable way to maintain collections of R functions and data sets.

Step 1: Load all functions and data sets into an R file

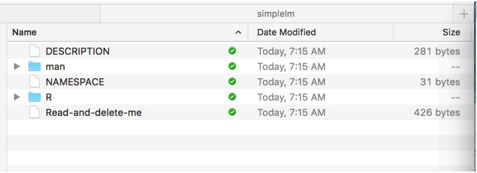
For example, in “simplelm.R” , we have the following code for fast simple linear regression

```
fastSimpleLinearRegression = function(y, x)  {
  y = y - mean(y)
  x = x - mean(x)
  n = length(y)
  stopifnot(length(x) == n) # for error handling
  s2y = sum( y * y ) / ( n - 1 ) # \sigma_y^2
  s2x = sum( x * x ) / ( n - 1 ) # \sigma_x^2
  sxy = sum( x * y ) / ( n - 1 ) # \sigma_xy
  rxy = sxy/sqrt( s2y * s2x ) # \rho_xy
  b = rxy * sqrt( s2y / s2x )
  se.b = sqrt( s2y * ( 1 - rxy * rxy ) / (n-2) / s2x )
  tstat = rxy * sqrt( ( n - 2 ) / ( 1 - rxy * rxy ) )
  p = pt( abs(tstat) , n - 2 , lower.tail=FALSE )*2
  return(list( beta = b , se.beta = se.b , t.stat = tstat, p.value = p ))
}
```

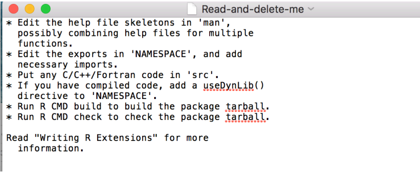
Step 2: Run package.skeletion()

In a clean R session, and run package.skeleton()

```
> package.skeleton("simplelm",code_files="simplelm.R")
Creating directories ...
Creating DESCRIPTION ...
Creating NAMESPACE ...
Creating Read-and-delete-me ...
Copying code files ...
Making help files ...
Done.
Further steps are described in './simplelm/Read-and-delete-me'.
```



Name	Date Modified	Size
DESCRIPTION	Today, 7:15 AM	281 bytes
▶ man	Today, 7:15 AM	--
NAMESPACE	Today, 7:15 AM	31 bytes
▶ R	Today, 7:15 AM	--
Read-and-delete-me	Today, 7:15 AM	426 bytes



```
* Edit the help file skeletons in 'man',
  possibly combining help files for multiple
  functions.
* Edit the exports in 'NAMESPACE', and add
  necessary imports.
* Put any C/C++/Fortran code in 'src'.
* If you have compiled code, add a useDynLib()
  directive to 'NAMESPACE'.
* Run R CMD build to build the package tarball.
* Run R CMD check to check the package tarball.

Read "Writing R Extensions" for more
  information.
```

Step 3: Edit help file skeletons in ‘man‘

``simplelm-package.Rd"

```
\name{simplelm-package}
\alias{simplelm-package}
\alias{simplelm}
\docType{package}
\title{
\packageTitle{simplelm}
}
\description{
\packageDescription{simplelm}
}
\details{

The DESCRIPTION file:
\packageDESCRIPTION{simplelm}
\packageIndices{simplelm}
This package provides a fast simple linear regression fitting

}
\author{
\packageAuthor{simplelm}

Maintainer: \packageMaintainer{simplelm}
}
\examples{
y  = rnorm(100)
x = rnorm(100)
fit=fastSimpleLinearRegression(y,x)
}
```

Step 3: Edit help file skeletons in ‘man‘ – continued

``fastSimpleLinearRegression.Rd"

```
\name{fastSimpleLinearRegression}
\alias{fastSimpleLinearRegression}
\title{
This function fits a simple linear regression
}
\description{
Fast implementation of simple linear regression models
}
\usage{
fastSimpleLinearRegression(y, x)
}
\arguments{
  \item{y}{
numeric vector represents the outcome variable
}
  \item{x}{
numeric vector represents the predictor
}
}
\details{
Compute the sufficient statistics
}

...
```

Step 3: Edit help file skeletons in ‘man‘– continued

``fastSimpleLinearRegression.Rd"

```
...
\value{
\item{beta}{coeffiicent estimate}
\item{se.beta}{standard error of coefficient estimate}
\item{t.stat}{t statistic}
\item{p.value}{p value}
}

\references{
Kang, J. (2016) Fast simple linear regression, working paper.
}
\author{
Jian Kang (jiankang@umich.edu)
}
\note{
%%  ~~further notes~~
}

\seealso{
%% ~~objects to See Also as \code{\link{help}}, ~~~
}
\examples{
y = rnorm(100)
x = rnorm(100)
fit = fastSimpleLinearRegression(y,x)
}
\keyword{Linear regression }
\keyword{Fast}% __ONLY ONE__ keyword per line
```

Step 4: Modify The package DESCRIPTION file

```
Package: simplelm
Type: Package
Title: fast simple linear regression
Version: 1.0
Date: 2015-11-24
Author: Jian Kang
Maintainer: Jian Kang <jiankang@umich.edu>
Description: This package is an example on how to create R package
License: GPL-3
```

Step 5: Check and Build

In shell window / terminal (windows users need to download Rtools: https://cran.r-project.org/bin/windows/Rtools/)

```
user$ R CMD check simplelm
* using log directory '/Users/jkang30/Dropbox/Umich/Biostat_615_Material/Fall_2015/Rcode/simplelm
* using R version 3.2.2 (2015-08-14)
* using platform: x86_64-apple-darwin13.4.0 (64-bit)
* using session charset: UTF-8
* checking for file 'simplelm/DESCRIPTION' ... OK
* checking extension type ... Package
* this is package 'simplelm' version '1.0'
* checking package namespace information ... OK
* checking package dependencies ... OK
....
user$ R CMD build simplelm
* checking for file 'simplelm/DESCRIPTION' ... OK
* preparing 'simplelm':
* checking DESCRIPTION meta-information ... OK
* installing the package to process help pages
* saving partial Rd database
* checking for LF line-endings in source and make files
* checking for empty or unneeded directories
* building 'simplelm_1.0.tar.gz'
```

Step 6: Install

In R,

```
> install.packages("simplelm_1.0.tar.gz",repos=NULL)
* installing *source* package 'simplelm' ...
** R
** preparing package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded
* DONE (simplelm)
```

How about C++ code?

Using the Rcpp package, we can improve the efficiency of R code
In ``cppfastLM.cpp", we have the following code:

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

// [[Rcpp::export]]
List cppfastLM(NumericVector y, NumericVector x){
  y = y - mean(y);
  x = x - mean(x);
  int  n = y.size();
  double s2y = sum( y * y );
  s2y /= n - 1.0;
  double s2x = sum( x * x );
  s2x /= n - 1.0;
  double sxy = sum( x * y );
  sxy /= n - 1.0;
  double rxy = sxy;
  rxy /= sqrt( s2y * s2x );
  double b = rxy * sqrt( s2y / s2x );
  double se_b = sqrt(s2y * ( 1.0 - rxy * rxy ) / (n-2.0) / s2x);
  NumericVector tstat;
  tstat.push_back(rxy * sqrt( ( n - 2 ) / ( 1 - rxy * rxy ) ));

  NumericVector p = pt(abs(tstat), n - 2 ,0, 0)*2;
  return Rcpp::List::create(Rcpp::Named("beta") = b,
                            Rcpp::Named("se.beta") = se_b,
                            Rcpp::Named("t.stat") = tstat[0],
                            Rcpp::Named("p.value") = p[0]);
}
```

Using Rcpp.packages.skeleton()

All the steps are the same except **Step 2**, now we use `Rcpp.packages.skeleton()`

```
> Rcpp.package.skeleton("fastSimpleLM",code_files = "simplelm.R",
cpp_files="cppFastLM.cpp")
Creating directories ...
Creating DESCRIPTION ...
Creating NAMESPACE ...
Creating Read-and-delete-me ...
Copying code files ...
Making help files ...
Done.
Further steps are described in './fastSimpleLM/Read-and-delete-me'.

Adding Rcpp settings
 >> added Imports: Rcpp
 >> added LinkingTo: Rcpp
 >> added useDynLib directive to NAMESPACE
 >> added importFrom(Rcpp, evalCpp) directive to NAMESPACE
 >> copied cppFastLM.cpp to src directory
 >> added example src file using Rcpp attributes
 >> compiled Rcpp attributes
 >> added Rd file for rcpp_hello_world
```

Performance Comparison

```
> library(microbenchmark)
> library(fastSimpleLM)
> n = 1000000
> x = rnorm(n)
> y = 2*x+rnorm(n)

> microbenchmark(lm(y~0+x),
+               fastSimpleLinearRegression(y,x),
+              cppfastLM(y,x),
+              times = 10L)

Unit: milliseconds
                             expr        min
                    lm(y ~ 0 + x) 630.289271
 fastSimpleLinearRegression(y, x)  18.127215
                  cppfastLM(y, x)   7.176658
         lq      mean     median         uq
 686.148319 810.42767 756.022109 802.849028
  31.825939  96.65432  61.273852 190.640336
   7.350376   7.72509   7.672616   8.228796
         max neval cld
 1225.938566    10   b
  228.754991    10  a
    8.321194    10  a
```

Summary

- Step 1: write R functions and C++ functions in separate files
- Step 2: Run `package.skeleton()` or `Rcpp.package.skeleton()`
- `Step 3: Edit help files`
- `Step 4: Edit DESCRIPTION file`
- `Step 5: Check and Build`
- `Step 6: Install`