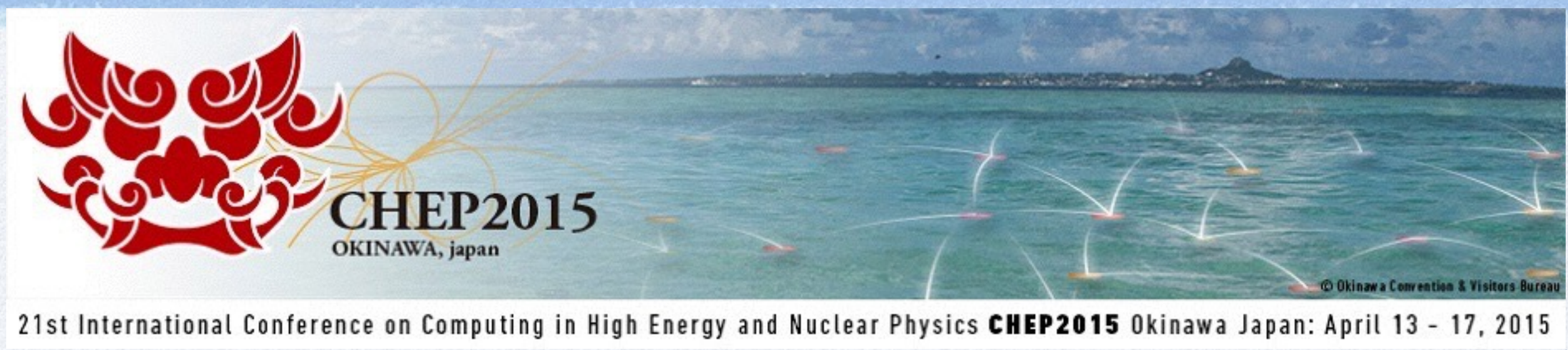
Using R in ROOT with the ROOT-R package

Lorenzo Moneta (CERN),
Omar A. Zapata M. (GSOC student)



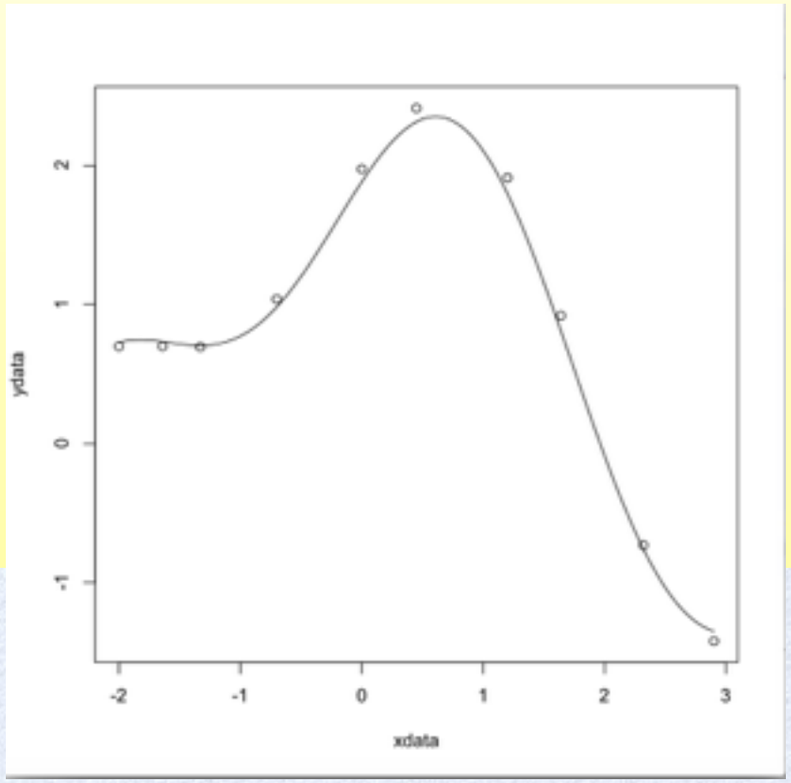
What is R ?

- R is an open source language and environment for statistical computing and graphics.
  - open source implementation of S language developed by J. Chambers
    - R popularity and usage increased largely in recent years
  - R provides a large variety of statistical tools
    - linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, …
  - Environment is highly extensible with a large number of existing packages

Example R Code

- Fitting (regression) of data points in R

```
> xdata = c(-2,-1.64,-1.33,-0.7,0,0.45,1.2,1.64,2.32,2.9)
> ydata = c(0.699369,0.700462,0.695354,1.03905,1.97389,2.41143,1.91091,0.919576,-0.730975,-1.42001)
> fit = nls(ydata ~ p1*cos(p2*xdata) + p2*sin(p1*xdata), start=list(p1=1,p2=0.2))
> summary(fit)
Formula: ydata ~ p1 * cos(p2 * xdata) + p2 * sin(p1 * xdata)

Parameters:
   Estimate Std. Error t value Pr(>|t|)
p1 1.881851   0.027430   68.61 2.27e-12 ***
p2 0.700230   0.009153   76.51 9.50e-13 ***
> confint(fit)
Waiting for profiling to be done...
        2.5%     97.5%
p1 1.8206081 1.9442365
p2 0.6794193 0.7209843
> plot(xdata,ydata)
> xgrid=seq(min(xdata),max(xdata),len=200))
> lines(xgrid,predict(fit,xgrid)
```



Introduction

- ROOT-R package (*developed by O. A. Zapata, GSOC student 2014)*
  - a new ROOT package to interface with R environment
    - use R, its functions and tools from the ROOT prompt or any C++ application
      - give access to ROOT users to the R capabilities and its rich functionality
      - focus more to use R statistical methods than graphics

ROOT ⇔ R Interfaces

- Goal of ROOT-R is to use R in the ROOT environment
- Could also be extended to access ROOT tools inside R
  - e.g. ROOT bindings in a R package to access ROOT files
    - this is not yet implemented
- Already exists an R package to use ROOT within the R environment
  - Bioconductor xps package *(Christian Stratowa, Austria)*
    - http://www.bioconductor.org/packages/2.11/bioc/html/xps.html

C++ Interface to R

- ROOT-R makes use of the R packages Rcpp and RInside
  - packages providing integration of R with C++
    - mapping back and forth of R object to C++ classes
    - calling R commands from C++ code
  - ROOT-R provides an additional extra layer to Rcpp and RInside
    - simple way to call R functions from ROOT prompt or C++ code
    - facilitate conversion between ROOT/C++ objects and R objects
  - Plug-ins can be developed to hide detail of ROOT-R interface
    - ROOT Minimizer plug-in using optimisation packages from R (RMinimizer)

How does it work ?

- TRInterface: main class interfacing to R
  - static instance that can be made available at the ROOT prompt
  - can be used to execute R commands
    - using TRInterface::Execute

```
root[] auto r = ROOT::R::TRInterface::Instance();

root[] r.Execute("print(version$version.string)");
[1] "R version 3.1.0 (2014-04-10)"
```

    - using TRInterface::Eval

```
root[] r.Eval("c(1,2,3,4)");
(ROOT::R::TRObjectProxy) @0x7fd9691a30a0
```

R Object -> ROOT

- TRObjectProxy class for converting from R objects to ROOT/C++ objects
- Automatic conversions with operator= after evaluating a R command

```
root[] double std_dev  = r.Eval("sd(rnorm(100))");
root[] std::vector<double> v = r.Eval("c(1,2,3,4)");
root[] TMatrixD m = r.Eval("matrix(c(1,2,3,4),2,2)");
root[] std::array<double,10> a = r.Eval("seq(1:10)");
```

- conversion with TRInterface::operator[] using the name of the R object

```
root[] r.Execute("mat<-matrix(c(1,2,3,4),2,2,byrow=TRUE)");
root[] TMatrixD m = r["mat"];
```

Using Operator << and >>

- We can also use operator<< to pass an R command to the TRInterface

```
root[] r << "mat<-matrix(c(1,2,3,4),2,2,byrow=TRUE)";
```

- And use operator >> to transform an R object into a ROOT/C++ object

```
root[] TMatrixD m;
root[] r["mat"] >> m;
root[] m.Print()
2x2 matrix is as follows
     |      0    |      1    |
------------------------------
   0 |           1           2
   1 |           3           4
```

ROOT/C++ -> R

- ROOT/C++ objects can be passed to R using the operator[] of TRInterface

```
root[] std::vector<double> v{1,2,3,4,5};
root[] r["v"] = v;
root[] r << "print(v)";
[1] 1 2 3 4 5
```

- or in combination with operator<<

```
root[]TMatrixD A(2,2,v.data() );
root[] r["A"] <<  A;
root[] r << "print(A)";
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

Supported Object Conversions

- C++ fondamental types (int, float,double)
  - R scalar (vectors of size 1)
- std::vector, std::array, std::list, TVectorD
  - R vector types
- TMatrixD ↔ R matrix
- Direct conversion from/to more complex R object (e.g. list or data frames) is not supported yet

Passing Functions to R

- If you have free C/C++ functions

```
Double_t myfun(Double_t x){
    return 2*cos(x);
}
Double_t myfun2(const std::vector<Double_t> & x){
    return x[1]*cos(x[0]);
}
```

- you can pass it directly to R

```
r["dilog"] = TMath::DiLog;
r["myfun"] = myfun;
r["myfun2"] = myfun2;
r << "print(dilog(0))";
r << "print(myfun(0))";
r << "print(myfun2(c(0,4)))";
```

- Can be used for passing functions to numerical algorithms from R (Numerical integration, minimisation, etc…)
- plan to support C++ lambda functions

Example: Integration in R

- Integrate a function using R numerical integration
  - *note:* R uses vectors as input to functions

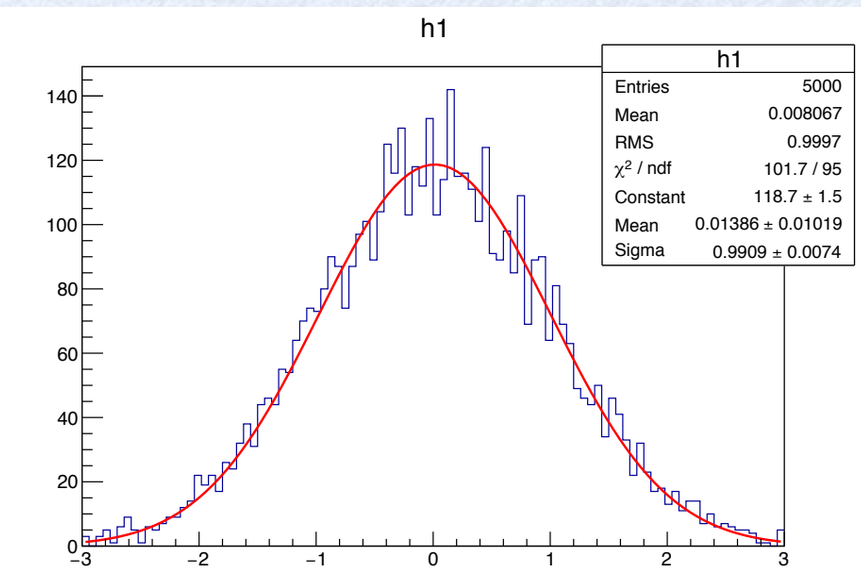
```
// function definition
std::vector<double>  BreitWignerVectorized(const std::vector<double> & xx){
  std::vector<double> result(xx.size());
  for(Int_t i=0;i<xx.size();i++)
  {
    result[i]=TMath::BreitWigner(xx[i]);
  }
  return result;
}

void Integrate() {
  // integrate a function using R
  auto r = ROOT::R::TRInterface::Instance();
  r["BreitWigner"]= BreitWignerVectorized;
  double value=r.Eval("integrate(BreitWigner, lower = -2, upper = 2)$value");

  std::cout<<"Integral of the BreitWigner in [-2, 2] =" << value << std::endl;
}
```

RMinimizer

- ROOT plugin for Minimisation implemented using R
  - *developed by Kirby Hermann (GSOC student 2014)*
  - give access to R optimisation tools when fitting or multi-dimensional function minimisation
  - based on R optim and optimx packages

```
ROOT::Math::MinimizerOptions::SetDefaultMinimizer("RMinimizer","L-BFGS-B");
hist->Fit("gaus");
```

```
root [4] h1.Fit("gaus")
Value at minimum =101.673
****************************************
Minimizer is RMinimizer / L-BFGS-B
Chi2                      =      101.673
NDf                       =           95
NCalls                    =          265
Constant                  =      118.694   +/-   1.47659
Mean                      =    0.0138555   +/-   0.0101907
Sigma                     =     0.990906   +/-   0.00741443
```



Future Developments

- Develop wrapper classes (plugin) in ROOT for R classification tools
  - integrate some of R classification tools in TMVA
    - GSOC project for 2015
- Extend ROOT-R to map also R data frames
  - Data frame: container of vectors of different types
    - TTree ⟷ data.frame conversion

```
> d = data.frame(x=c(1,2,3),y=c(4,5,6),var=c("a","b","c"))
 x y var
1 1 4   a
2 2 5   b
3 3 6   c
```

Conclusions

- ROOTR provides easy access to R tools in ROOT and C++
- Easy to use directly R from ROOT prompt
  - extend the functionality of ROOT giving access to a large variety of statistical methods from R
    - can be used also to verify and cross-check some of existing ROOT tools
    - new ROOT plugins or wrapper classes implemented using R can be easily created
- Package is ready do be released in the next ROOT production version (6.0.4)

References

- ROOT-R code
  - available now as a github branch
    - https://github.com/lmoneta/root/tree/master-root-R
    - will be soon added in central ROOT git master (6.03.05)
    - with example tutorials in `tutorials/r` directory
- ROOT-R User Guide
  - http://gfif.udea.edu.co/web/tiki-print.php?page=ROOTR%20Cling%20Users%20Guide
- ROOT-R Web page
  - http://root.cern.ch/drupal/content/how-use-r-root-root-r-interface
  - http://gfif.udea.edu.co/web/tiki-index.php?page=ROOTR+Cling