

# RserveCLI

An Rserve Client Implementation for CLI/.NET

Oliver M. Haynold

R/Finance 2011

April 2011

# An Impedance Mismatch

**R**

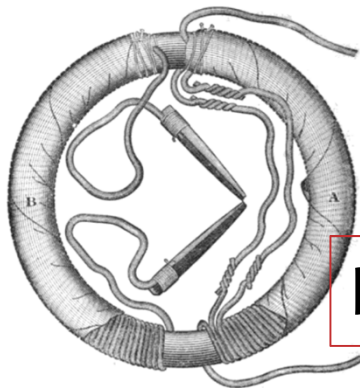


- GNU toolchain
- More at home on \*nix
- Dynamic typing
- GNU licenses

**.NET/CLI**



- .NET or Mono toolchain
- More at home on Windows
- Mostly static typing
- Permissive open source and proprietary licenses



How can we connect the two worlds?

# Rserve to the Rescue!

- TCP/IP server to access R remotely
- Important Commands
  - Evaluate expression
  - Move R objects to and from the remote session
  - Move files
- Clients for Java, C++, and PHP
- <http://www.rforge.net/Rserve/>

All we need is a CLI client!  
RserveCLI

# A Simple Example

```
using (var s = new RConnection(  
    new System.Net.IPAddress(  
        new byte[] { 192, 168, 37, 10 })),  
    port: 6311,  
    user: "ruser", password: "rpwd"))  
{  
    // Generate some example data  
    var x = Enumerable.Range(1, 20).ToArray();  
    var y = (from a in x select (0.5 * a * a) + 2).ToArray();  
  
    // Build an R data frame  
    var d = Sexp.MakeDataFrame();  
    d["x"] = Sexp.Make(x);  
    d["y"] = Sexp.Make(y);  
    s["d"] = d;
```

We make a data  
frame locally

Now we send the data  
frame to the R server

Sexp.Make() turns  
native data into Sexp

# A Simple Example (2)

```
// Run a linear regression, obtain the summary,  
// and print the result  
var linearModelSummary = s.Eval["summary(lm(y ~ x, d))"];  
Console.WriteLine(linearModelSummary.Count);  
var coefs = linearModelSummary["coefficients"];  
var rSquared = (double) linearModelSummary["r.squared"];  
Console.WriteLine(  
    "y = {0} x + {1}. R^2 = {2,4:F}%",  
    coefs[1, 0], coefs[0, 0], rSquared * 100);
```

Evaluate a  
command

Sexp typecast to  
native type

Sexps can be indexed in  
one or two dimensions, by  
names or by indices

# A Simple Example (3)

```
// Make a chart and transfer it to the local machine
```

```
s.VoidEval("library(ggplot2)");  
s.VoidEval("pdf(\"outfile.pdf\")");  
s.VoidEval("print(qplot(x,y, data=d))");  
s.VoidEval("dev.off()");
```

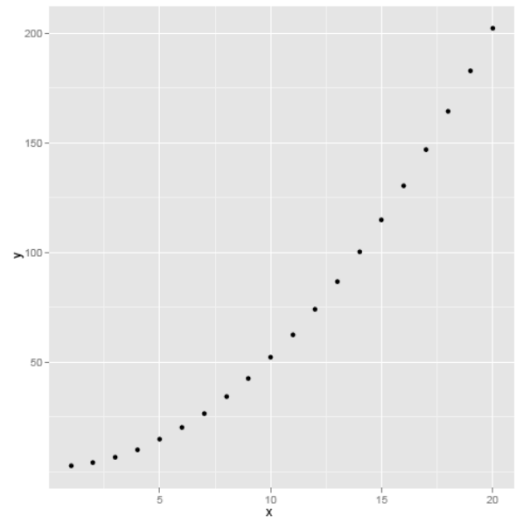
Generate a PDF  
file on the server

```
using (var f = File.Create("Data Plot.pdf"))  
{  
    s.ReadFile("outfile.pdf").CopyTo(f);  
}
```

Copy the file to the  
local machine

```
s.RemoveFile("outfile.pdf");
```

Remove the file  
from the server



# A Simple Example (4)

```
// Now let's do some linear algebra
var matA = new double[,] {
    { 14, 9, 3 }, { 2, 11, 15 },
    { 0, 12, 17 }, { 5, 2, 3 } };
var matB = new double[,] { { 12, 25 }, { 9, 10 }, { 8, 5 } };
s["a"] = Sexp.Make(matA);
s["b"] = Sexp.Make(matB);
Console.WriteLine(s["a %*% b"].ToString());
```

Comfortable syntax.  
Sexp.Make() does the  
Right Thing.

Sexp.ToString()  
formats Sexps in a  
user-friendly way

[ 273, 243, 244, 102, 455, 235, 205, 160 ]

Try it out!

RserveCLI

<http://rservecli.codeplex.com/>