Introduction

- R: a powerful, free, open-source, reliable, statistical computing environment; large community of developers; language used for prototyping much modern methodology
- Web: ubiquitous, convenient, simple-to-use infrastructure

Web Service

W3C  A software system designed to support interoperable machine-to-machine interaction over a network
Interface described in Web Services Description Language (WSDL)
Interaction via messages (perhaps enclosed in a SOAP envelope)
Transport layer is HTTP comprised of XML and other standards
Why R as a web service?

- Data often resides in a separate place from where R is available. People can find downloading, formatting, and setting up can become chores
- Many routine computations can be done using R (data quality checks, trends, recruitment breakdowns) and easily provided to users if we could have R as a web service
- Large systems doing complex tasks can be built using distributed web services; these become necessary for high-throughput analyses in genomics, proteomics etc.

- Other Approaches
- The lingua franca
- XML Bindings
- R Data XML Binding
- R Data Schema (contd.)
- R Data Schema (contd.)
- StatDataML DTD
- Other StatDataML Bindings
- uStatDataML
- API (contd.)
- API (contd.)
- Transformations
- XSLT Language
- XSLT Example Input
- XSLT Transform
- XSLT Output
- R as an XSLT Transform

- Introduction
- Why R as a web service?
- Other Approaches
- The lingua franca
- XML Bindings
- R Data XML Binding
- R Data Schema (contd.)
- R Data Schema (contd.)
- StatDataML DTD
- Other StatDataML Bindings
- uStatDataML
- API (contd.)
- API (contd.)
- Transformations
- XSLT Language
- XSLT Example Input
- XSLT Transform
- XSLT Output
- R as an XSLT Transform

- Integration
- Why R as a web service?
- Other Approaches
- The lingua franca
- XML Bindings
- R Data XML Binding
- R Data Schema (contd.)
- R Data Schema (contd.)
- StatDataML DTD
- Other StatDataML Bindings
- uStatDataML
- API (contd.)
- API (contd.)
- Transformations
- XSLT Language
- XSLT Example Input
- XSLT Transform
- XSLT Output
- R as an XSLT Transform

- Continuation
- Why R as a web service?
- Other Approaches
- The lingua franca
- XML Bindings
- R Data XML Binding
- R Data Schema (contd.)
- R Data Schema (contd.)
- StatDataML DTD
- Other StatDataML Bindings
- uStatDataML
- API (contd.)
- API (contd.)
- Transformations
- XSLT Language
- XSLT Example Input
- XSLT Transform
- XSLT Output
- R as an XSLT Transform

- The various xslt examples can be used to show how R can be used as a web service.
R need not be used only as web service. Other possibilities are available.

- Linking R with your own C, Fortran code. There is a well-established API for calling R from these languages.
- Embedding R in your favorite language; R has been embedded in Perl, Python, Java etc.
- CGI scripts can be used with R in a web server and R embedded inside Apache is also available. Educational use of R has been made with these technologies for teaching purposes.
R need not be used only as web service. Other possibilities are available.

- Linking R with your own C, Fortran code. There is a well-established API for calling R from these languages.
- Embedding R in your favorite language; R has been embedded in Perl, Python, Java etc.
- CGI scripts can be used with R in a web server and R embedded inside Apache is also available. Educational use of R has been made with these technologies for teaching purposes.

Embedding R in a language couples R to the language somewhat closer than what a web service suggests.

The first question: what language will the web services speak? Answer: XML (eXtensible Markup Language). XML is a (text-based) way of describing data. Here is an example:

```xml
<?xml version="1.0"?>
<data>
  <array type="integer">
    <values>1 2 3 4 5 6 7 8 9 10</values>
  </array>
</data>
```

Key details: arbitrary but matching tags, attributes, free format. XML documents must be well-formed! In addition, they must also be valid. This leads to XML DTDs and Schemas.
Many modern languages have XML Bindings. A Binding allows the XML data to be treated as a native object in the language.

For example, Java has JAXB (Java for XML Binding) which allows XML data to be treated as objects. Python has PyXML that allows XML data to be treated as Python Objects.

With a binding, one can

unmarshal read XML data and create an object/methods to access the contents

marshal write out an object as XML that others can consume

Important: Bindings are based on an XML DTD/Schema
R has a package called StatDataML that described some common data types that it understands: vectors, lists, matrices.

Creating StatDataML
library(StatDataML)
x <- c(1,2,3)
writeSDML(x, file="int.xml")

To read the XML back in:
library(StatDataML)
x = readSDML("int.xml")
x

[1] 1 2 3
To appear on Omegahat:

**JStatDataML**  Java StatDataML binding

**PyStatDataML**  Python StatDataML binding

Java can grok StatDataML.
To appear on Omegahat:

**JStatDataML**  Java StatDataML binding  
**PyStatDataML**  Python StatDataML binding

Java can grok StatDataML.

- In Java, you can create R data objects such as lists, vectors, data frames, matrices.

- These objects can be *marshalled* into XML.

- In Java, you are able to read StatDataML and represent them as pedigreed Java objects. In other words, StatDataML can be *unmarshalled* into Java objects.

You can do the same in Python, but the focus here is Java.
JStatDataML

- JStatDataML is layered on top of StatDataML so that any changes to StatDataML can be easily reworked
- Handles missing values, factors etc.
- Helper Java classes are available to take results of a database query and marshall it into StatDataML for R computations
- Simple Application Programming Interface that is easy to use

Caveat: The StatDataML data types represent a subset of R data objects. In particular, you don't want to use this for large expression arrays, proteomics data etc.
Transformations are a general approach to convert XML data to convert to other formats (including XML). They are specified by means of a language, XSLT (Extensible Stylesheet Language Transformations), an XML-based language used for the transformation of XML documents. The original document is untouched and the output is a transformed document that can be converted to HTML, plain text, or PDF.
Transformations

Transformations are a general approach to convert XML data to convert to other formats (including XML). They are specified by means of a language, XSLT (Extensible Stylesheet Language Transformations), an XML-based language used for the transformation of XML documents. The original document is untouched and the output is a transformed document that can be converted to HTML, plain text, or PDF.

XSLT Language

- Declarative
- Applies transform to nodes that meet conditions
- Results are added to result tree
- Nodes not matching conditions are left untouched

XSLT Example Input

<?xml version="1.0"?>
<persons>
  <person username="naras">
    <name>Balasubramanian</name>
    <family-name>Narasimhan</family-name>
  </person>
  <person username="shenry">
    <name>Solomon</name>
    <family-name>Henry</family-name>
  </person>
</persons>
Now imagine if you can extend XSLT to do R computations. Enter Sxslt, an omegahat project which is an extension of XSLT to R. This allows us to do processing such as:

```xml
<xsl:template match="sqrt">
  <xsl:value-of select="r:call('sqrt', number(.))"/>
</xsl:template>
```

In an xml document, if Sxslt sees a tag like:

```xml
<lm y="creatine" x="predictors" />
```

it will replace it with the value of the R function `myfun` taking y and x as arguments.

**R as an XSLT Transform**

- Now imagine if you can extend XSLT to do R computations.
- Enter Sxslt, an omegahat project which is an extension of XSLT to R.
- This allows us to do processing such as:

  ```xml
  <xsl:template match="sqrt">
    <xsl:value-of select="r:call('sqrt', number(.))"/>
  </xsl:template>
  ```

  In an xml document, if Sxslt sees a tag like:

  ```xml
  <lm y="creatine" x="predictors" /> 
  ```

  it will replace it with the value of the R function `myfun` taking y and x as arguments.

**A pipeline**

- Introduction
- Why R as a web service?
- Other Approaches
- The Inqua franca
- XML Bindings
- R Data XML Binding
- R Data Schema (contd.)
- API (contd.)
- Transformations
- XSLT Language
- XSLT Example Input
- XSLT Transform
- Persistent Data Storage
- Oracle
- PostgreSQL
- MySQL

**Continuing work**

- Being deployed in small settings at Stanford
- Technical issues being worked out
- RSXMLObjects is an omegahat project for serializing general R objects to XML. These might be more general than StatDataML
- Integration of this web services into a Struts framework that is widely used at the DCC and elsewhere
- Security issues although there are good solutions available
- Strengthening the technology for enterprise use

If you are interested in playing with these technologies, I can provide you some downloads.