Advanced R Programming - Lecture 5

Leif Jonsson

Linköping University

leif.jonsson@ericsson.com
leif.r.jonsson@liu.se

September 8, 2016

Today

Input and output

Basic I/O

Cloud storage

web API:s

web scraping

Shiny

Relational Databases

Questions since last time?

Input and output

Input
Output

Format, localization

Input and output

Input
Output

Format, localization and encoding...... hell!

The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets (No Excuses!)

"Formats"

csv files
excel files
sas, spss, R, ... files
documents
html
json
xml
SQL
MongoDB
maps
pictures

More?

data files (see input)
reports (dyn)
html
update DB
graphs
"Shiny"
maps
pictures

Localization

🏠	🌐
own Computer	Cloud Storage
local network	web pages
local database	web scraping
	web APIs
	remote database

Table: Local - Remote

Files on your computer

```
# Input simple data
read.table()
read.csv()
read.csv2()

load()


# Output simple data
write.table()
write.csv()
write.csv2()

save()
```

More complex formats

software/data	**package**
Excel	XLConnect
SAS, SPSS, STATA, ...	foreign
XML	xml
JSON (GeoJSON)	rjsonio, RJSON
Documents	tm
Maps	sp
Images	raster

Table: Format - R package

Cloud storage

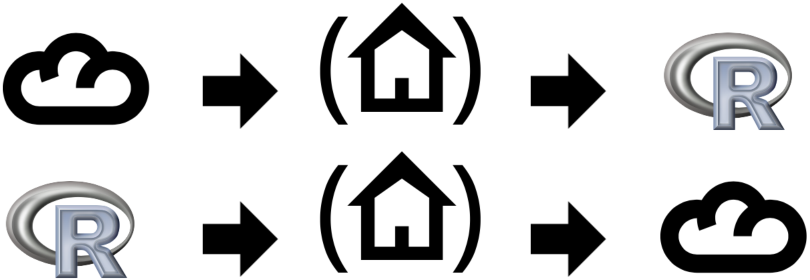


Table: Local - Remote

Why?

Robust

Backups

Cloud computing

... but how about safety?

... and can be tricky in the beginning

Localization

Arbitrary data

Structured data

API Packages

Remote	**package**
General	downloader
GitHub	repmis, downloader
Dropbox	rdrop
Amazon	RAmazonS3
Google Docs	googlesheets

web API:s

application program interface using http

"contract to 'get data' online"

more and more common

examples:

github

Riksdagen

Statistics Sweden

RESTful

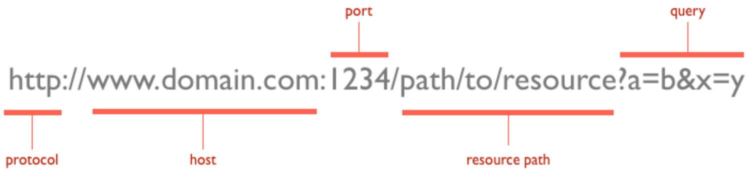
Basic principles:

Data is returned (JSON / XML)

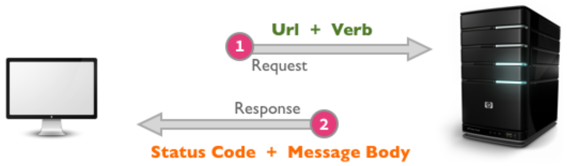
Each specific data has its own URI

Communication is based on HTTP verbs

Hypertext Transfer Protocol (http)



Hypertext Transfer Protocol (http)



Verbs

Verb	Description
GET	Get "data" from server.
POST	Post "data" to server (to get something)
PUT	Update "data" on server
DELETE	Delete resource on server

Status codes

Code	Description
1XX	Information from server
2XX	Yay! Gimme' data!
3XX	Redirections
4XX	You failed
5XX	Server failed

Example REST API's

Linköping Luftkvalitet API

Google Map Geocode API

Common API formats

JavaScript Object Notation (JSON)
Think of named lists in R
R Packages: RJSONIO, rjsonlite

Extensible Markup Language (XML)
Older format (using nodes)
xpath
R Packages: XML

JSON

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
        "streetAddress": "21␣2nd␣Street",
        "city": "New␣York",
        "state": "NY",
        "postalCode": "10021"
  },
  "phoneNumber": [
        { "type": "home", "number": "212␣555" },
        { "type": "fax", "number": "646␣555" }
  ],
  "newSubscription": false,
  "companyName": null
}
```

XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<wikimedia>
<projects>
<project name="Wikipedia" launch="2001-01-05">
<editions>
<edition language="English">en.wikipedia.org</edition>
<edition language="German">de.wikipedia.org</edition>
<edition language="French">fr.wikipedia.org</edition>
<edition language="Polish">pl.wikipedia.org</edition>
<edition language="Spanish">es.wikipedia.org</edition>
</editions>
</project>
<project name="Wiktionary" launch="2002-12-12">
<editions>
<edition language="English">en.wiktionary.org</edition>
<edition language="French">fr.wiktionary.org</edition>
<edition language="Vietnamese">vi.wiktionary.org</edition>
<edition language="Turkish">tr.wiktionary.org</edition>
<edition language="Spanish">es.wiktionary.org</edition>
</editions>
</project>
</projects>
</wikimedia>
```

web scraping

Unstructured http(s) data

Often HTML format

Spiders / scraping / web crawlers

Basics behind search engines

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

(har)rvest

JavaScript Object Notation (JSON)
Simplify spider activity

Download data
Parse data
Follow links
Fill out forms
Store crawling history

Figure: Spiderman

Difficulties and bad spiders

Scraping is fragile!
Difficulties and bad spiders
www.domain.se/robot.txt
Politeness

robot traps
javascript
delays

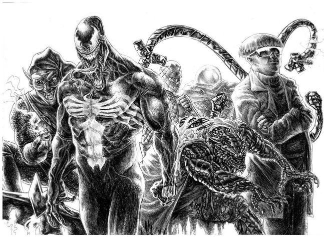


Figure: Bad spiders

Shiny?

Interactive dashboards made easy

online or local

R as "backend"

Shiny?

Shiny Examples

How it works

Application

Reactive

modify using HTML

```
MyAppName/server.R
MyAppName/ui.R
```

server.R define working directory

Shiny Example

```
library(shiny)
# Examples with code
runExample("01_hello")
runExample("03_reactivity")
```

Publish Shiny

locally
zip-file in cloud
github (see runGithub())

Publish Shiny

locally
zip-file in cloud
github (see runGithub())

your own server
shinyapps.io

Relational Databases

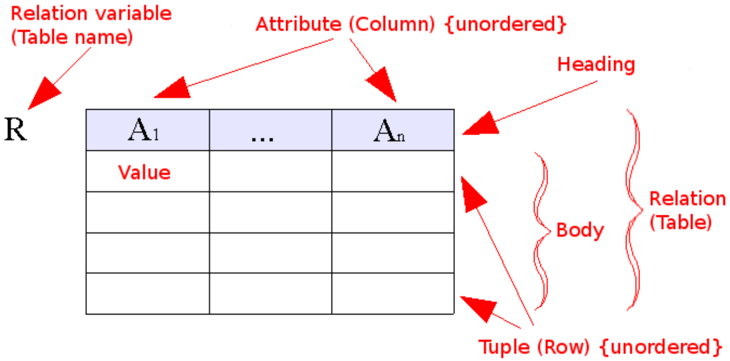
Structured datasbase in tables

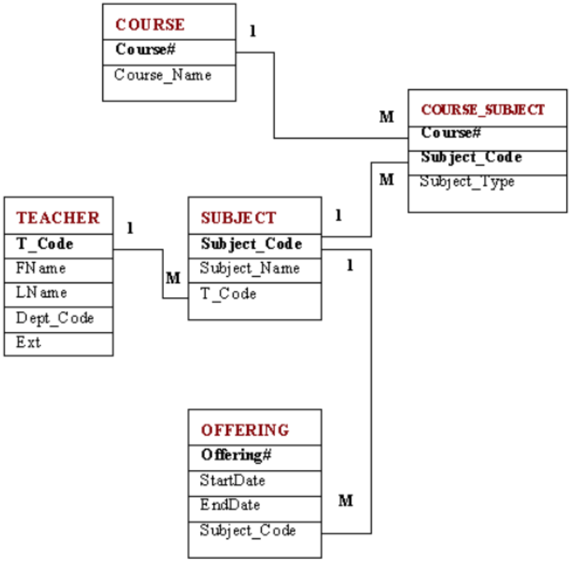
local or online

query language for I/O

effective for big data

difficult to design


Relation variable (Table name)
Attribute (Column) {unordered}
Heading
R
A1
...
An
Value
Body
Relation (Table)
Tuple (Row) {unordered}


COURSE
Course#
Course_Name
1
M
COURSE_SUBJECT
Course#
Subject_Code
Subject_Type
M
TEACHER
T_Code
FName
LName
Dept_Code
Ext
1
M
SUBJECT
Subject_Code
Subject_Name
T_Code
1
1
OFFERING
Offering#
StartDate
EndDate
Subject_Code
M

A good database

Can be difficult to design
No duplicates
No redundancies
Easy to query
Easy to update
"Normal forms"

Using databases from R

Database system	**R package**
ODBC (Microsoft Access)	RODBC
PostgreSQL	RPostgresql
Oracle	ROracle
MySQL	RMySql
MongoDB	rmongodb

Table: Database - R package

The End... for today.

Questions?

See you next time!