

Advanced R Programming - Lecture 5

Krzysztof Bartoszek
(slides based on Leif Jonsson's and Måns Magnusson's)

Linköping University
krzysztof.bartoszek@liu.se

26 September 2019 (KEY1)

Today

Input and output

Basic I/O

Cloud storage

web APIs: Lab

web scraping

Shiny

Relational Databases

Questions since last time?

Input and output



Input and output



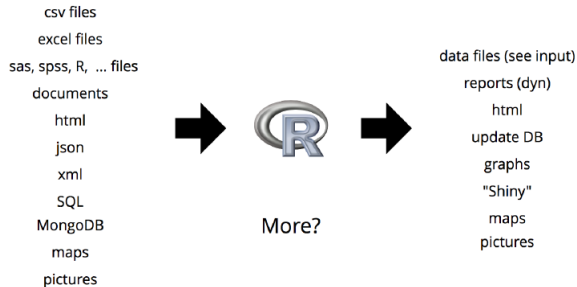
<http://www.joelonsoftware.com/articles/Unicode.html>
The Absolute Minimum Every Software Developer Absolutely, Positively
Must Know About Unicode and Character Sets (No Excuses!)

Unicode defines codes for **all (?)** characters—multiple encodings
(for a given language only small fraction of characters used)

Content-Type tag for HTML

BUT e-mail, .txt, .csv

"Formats"



Localization



own Computer
local network
local database



Cloud Storage
web pages
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

Excel

SAS, SPSS, STATA, ...

XML

JSON (GeoJSON)

Documents

Maps

Images

package

XLConnect

foreign

xml

rjsonio, RJSON

tm

sp

raster

Table: Format - R package

Cloud storage



Table: Local - Remote

Why?

Robust

Backups

Cloud computing

can be tricky in the beginning

but how about safety?

But control on what is going on?

BUT requires internet connection

Localization

Arbitrary data



Structured data



API Packages

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

web APIs

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)

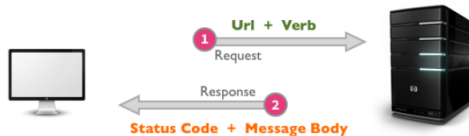
Each specific data has its own URI

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

<http://www.linkoping.se/open/data/Luftkvalitet/>
Linköping Luftkvalitet API

<https://developers.google.com/maps/documentation/geocoding/intro>
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 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

Difficulties and bad spiders

Scraping is fragile!

Difficulties and bad spiders

`www.domain.se/robot.txt`

Politeness

robot traps

javascript

delays

Shiny?

Interactive dashboards made easy

online or local

R as "backend"

Shiny?

<https://www.rstudio.com/products/shiny/shiny-user-showcase/>
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

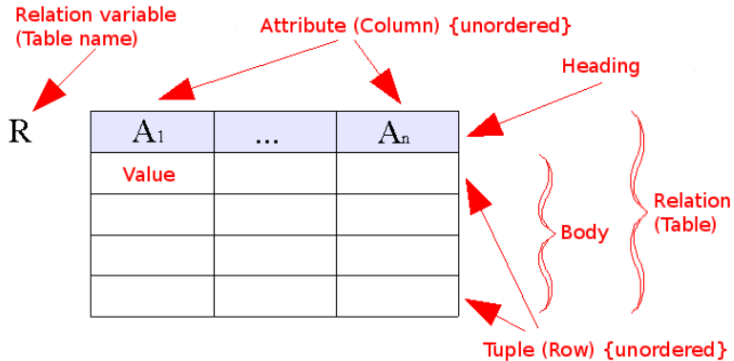
Structured database in tables

local or online

query language for I/O

effective for big data

difficult to design



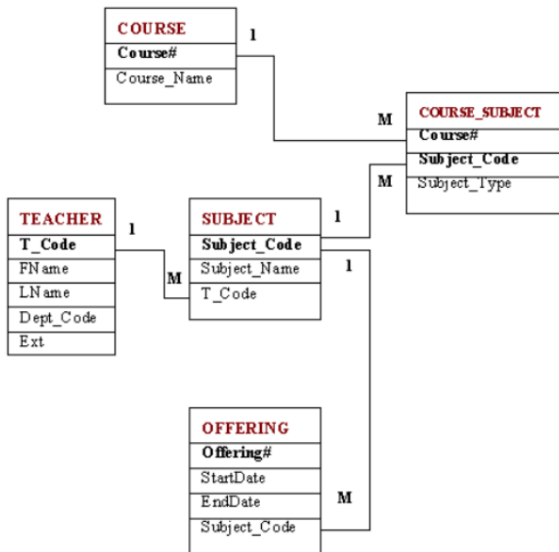
Keys

Superkey “set of attributes such that two distinct rows that do not have the same values for these attributes”

Primary key (attribute): choice of superkey, relationships between tables are done through the primary key

<https://en.wikipedia.org/wiki/Superkey>

https://en.wikipedia.org/wiki/Primary_key



A good database

Can be difficult to design ?

No duplicates

No redundancies

Easy to update

"Normal forms"

Easy to query

A good database: normalization

Database normalization: “is the process of restructuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity”

(usually divide table into separate tables linked by primary keys)

Denormalization: create redundancies for increased performance:
(preferred) store normalized data and allow DBMS to create additional redundancies (DBMS is responsible for inconsistencies)
(common) designed denormalized DB (designer is responsible for inconsistencies)

https://en.wikipedia.org/wiki/Database_normalization

<https://en.wikipedia.org/wiki/Denormalization>

A good database: normal forms in brief

First normal form: in each attribute (column) entry there is a single atomic value:

for Telephone Number you cannot have two telephone numbers

Second normal form: 1NF and each non-primary attribute depends functionally only on the primary attribute and not on any other attribute:

(Course_code, Course_name, University, University_country) is not in 2NF as University_country is defined through University here (Course_code, University) is the (composite) primary key

https://en.wikipedia.org/wiki/X_normal_form, X appropriate form

A good database: normal forms in brief

Third normal form: 2NF and “Every non–prime attribute of R is non–transitively dependent on every key of R.”: (University, Year, Vice–Chancellor, Vice–Chancellor DOB)
composite primary key (University,Year)
Vice–Chancellor DOB depends on key via Vice–Chancellor
(what if someone made a typo when entering a second time?)

Boyce–Codd normal form or 3.5NF: more strict than 3NF, no functional dependencies between two attributes of which neither is a superkey:
(city, land_plot, postal_code) fails due to relationship between city and postal_code

https://en.wikipedia.org/wiki/X_normal_form, X appropriate form

A good database: normal forms in brief

Fourth normal form: 3NF and no multiple multivalued dependencies:

(Teacher, Language, Course), primary key is whole entry

Version 1 (redundant)

KB, Polish, 732A94

KB, Polish, 732A63

KB, English, 732A94

KB, English, 732A63

KB, Swedish, 732A94

KB, Swedish, 732A63

Version 2 (what if I stop teaching R?)

KB, Polish, 732A94

KB, English, 732A94

KB, Swedish, 732A63

https://en.wikipedia.org/wiki/X_normal_form, X appropriate form

A good database: normal forms in brief

Fifth normal form: when there are complex constraints on the possible combinations of values

Sixth normal form: when there are temporal dependencies in data (can lead to table explosion)

Domain–key normal form: values only constrained by permissible values for attributes and key uniquely identifying row: (Lecturer, Lecturer_description, University) fails (but 1NF?):

KB, **LiU** Statistician, **LiU**

TB, **SU** Mathematician, **SU**

TE, **LiU** Mathematician, **LiU**

FR, **SU** Biologist, **SU**

https://en.wikipedia.org/wiki/X_normal_form, X appropriate form

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!