Bringing structure into data processing work-flows in R

Jan Philipp Dietrich, Lavinia Baumstark, Anastasis Giannousakis, Benjamin Leon Bodirsky, David Klein

8 September 2017
The problem
Our attempt to solve it
Our attempt to solve it

```
101101
110001
012010
```

```
101101
110001
011010
```

“blackbox” script
Our attempt to solve it

```
101101 110001 012010
```

“blackbox” script

```
101101 110001 011010
```

```
101101 110001 012010
```

```
101101 110001 011010
```
Our attempt to solve it

101101 110001 012010

“blackbox” script

101101 110001 012010

101101 110001 011010

101101 110001 012010

101101 110001 011010

101101 110001 011010
The derived framework

1) Download data
2) Read data and convert to standardized data format
3) Bring data to country-resolution
The derived framework

- Calculate required data
  - Filtering of data
  - Merging of data from different data sources
  - Data harmonization
- Provide spatial aggregation (e.g. weight)
The derived framework

- Collecting data sets
- Coordinate packaging of aggregated data
The derived framework
The derived framework
The derived framework
Unanticipated side effects

• A lot of low hanging fruits
  • Meta-data generation
  • Sanity checks
  • Data processing network
  • Data caching
  • Structured log file

• User report faster development
• Broader usage than planned

• Change in focus:
  Spatial aggregation → reproducibility & transparency
Additional Information

• The R package:
  • Name: MADRaT
    “May All Data be Reproducible and Transparent”
  • License: BSD2
  • Git: https://github.com/pik-piam/madrat
  • CRAN: https://CRAN.R-project.org/package=madrat

• License of this presentation: CC-BY-4.0
  https://creativecommons.org/licenses/by/4.0/

• Contact: dietrich@pik-potsdam.de
Wrapper functions:

```r
calcOutput("ours")
```

User functions:

```r
calcOurs <- function() {
a <- readSource("ours")
  #do some fancy calculations
  return(list(x=x, weight=weight, unit="-", description="Some example calculations"))
}
```

```r
readYours() {
x <- read.csv("example.csv")
  return(as.magpie(x))
}
```

```r
convertYours(x) {
y <- toolAggregate(x, "mapping.csv")
  return(y)
}
```

```r
downloadYours() {
download.file("http://exam.ple/data.zip", destfile = "data.zip")
  unzip("data.zip")
  unlink("data.zip")
}
**Backup Slides**

<table>
<thead>
<tr>
<th>Wrapper functions</th>
<th>User functions</th>
</tr>
</thead>
</table>
| `retrievedata("example", rev=1.2, modelfolder="example", regionmapping="example.csv")` | `fullEXAMPLE <- function(rev=0) {
  if(rev>=1) {
    calcOutput("ours", round=2, file="ours.cs4", destination="testfolder")
  } else {
    stop("No calculations for rev<1 available!")
  }
}` |