Max-Planck-Institut für Bildungsforschung
Max Planck Institute for Human Development

Automatic Codebooks from Survey Metadata Encoded in Attributes

Ruben C. Arslan
(with help and advice from Martin Brümmer and Kai Horstmann)

https://github.com/rubenarslan/codebook

ruben.arslan@gmail.com
@rubenarslan

survey software: formr.org
blog: http://the100.ci

Metadata for scientific data

- data about data
- help to document datasets for large teams or future you
- help others discover your dataset to reuse it or find papers based on it
- help others learn about what's inside before requesting access
- help machines and algorithms use and merge data

Background

- too little metadata about scientific data is available
- powerful techniques like machine learning have to be supplemented by lots of stupid manual labour (e.g. coding for meta analysis)
- standards and guidelines exist
    - people don't know and don't care
    - if they did, they still wouldn't spend time on this (the scientific system for the most part doesn't encourage data sharing and reuse (i.e. "better" to have two papers from your data than to have one data publication with ten papers citing it)

Make it easy and they will come?

- Goal of this package is to solve a coordination problem
- Supply something data holders want:
  - nice, readable, shareable, searchable documentation
  - descriptives and reliabilities
  - missingness patterns
- Get something we all (should) want
  - nice, machine- and human-readable metadata that can be indexed in search engines, fed into bigger databases

Precursor Concepts

- rmarkdown
    - a way of blending Markdown (a low-tech way of marking up plain text) with R commands and outputs (model summaries, plots, etc.)
- rmarkdown partials
    - putting repetitive sections in your document in a partial (name starts with _) and then calling it via a function

Precursor Concepts

- informal metadata standards
  - e.g. a PDF with variable names and descriptions
  - slightly better: an xlsx or csv file
  - Stata/SPSS: pretty useful if you have the right SW, not open
  - R attributes, semantics transferred from Stata/SPSS

```
> attributes(s1_demo$has_children)
$labels
                                  Nein                                     Ja
                                     0                                      1
Item was never rendered for this user.
                                    NA

$class
[1] "labelled"

$label
[1] "Haben Sie Kinder?"
```

Metadata standards

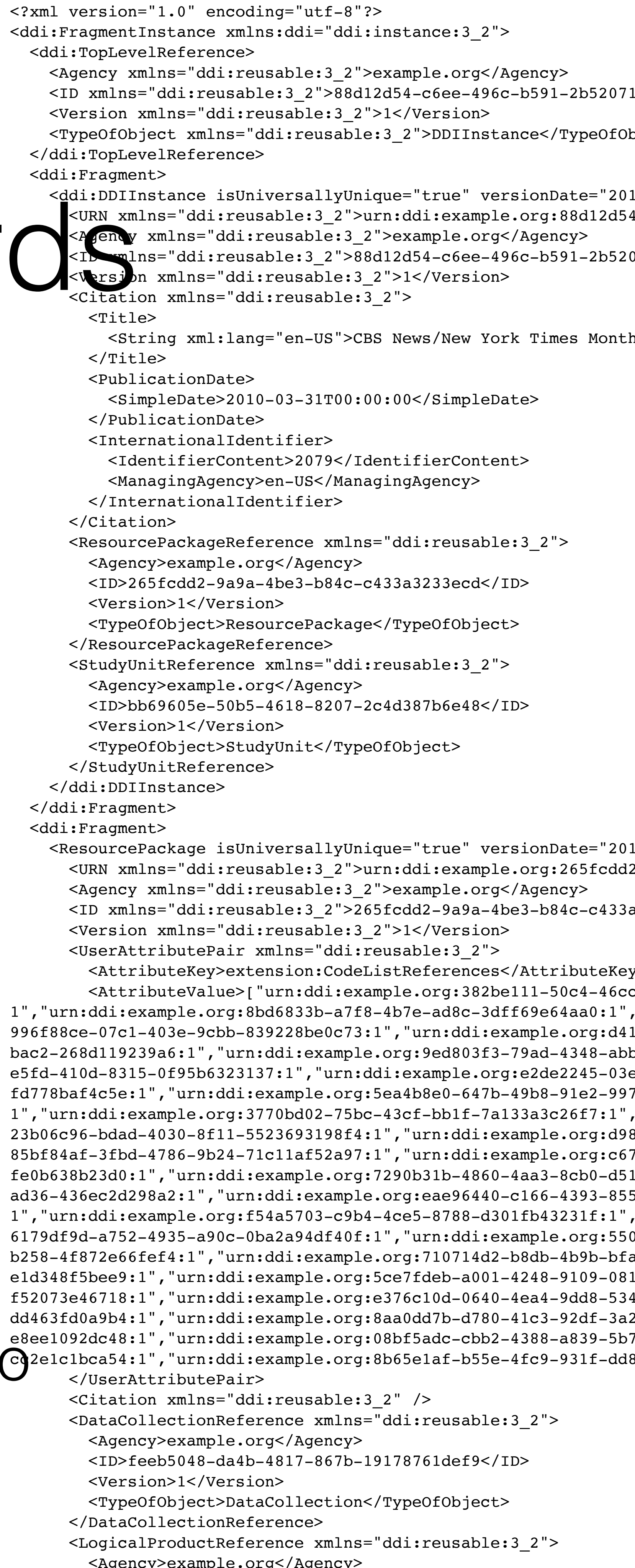
- **JSON-LD**: JSON linked data
  - lightweight metadata markup
  - can be embedded in an HTML webpage
  - will be indexed by Google
  - human-readable

```
{
  "@context":"http://schema.org/",
  "@type":"Dataset",
  "name":"NCDC Storm Events Database",
  "description":"Storm Data is provided by the National Weather Service (NWS) and contain statistics or
  "url":"https://catalog.data.gov/dataset/ncdc-storm-events-database",
  "sameAs":"https://gis.ncdc.noaa.gov/geoportal/catalog/search/resource/details.page?id=gov.noaa.ncdc:
  "keywords":[
     "ATMOSPHERE > ATMOSPHERIC PHENOMENA > CYCLONES",
     "ATMOSPHERE > ATMOSPHERIC PHENOMENA > DROUGHT",
     "ATMOSPHERE > ATMOSPHERIC PHENOMENA > FOG",
     "ATMOSPHERE > ATMOSPHERIC PHENOMENA > FREEZE"
  ],
  "creator":{
     "@type":"Organization",
     "url": "https://www.ncei.noaa.gov/",
     "name":"OC/NOAA/NESDIS/NCEI > National Centers for Environmental Information, NESDIS, NOAA, U.S.
     "contactPoint":{
        "@type":"ContactPoint",
        "contactType": "customer service",
        "telephone":"+1-828-271-4800",
        "email":"ncei.orders@noaa.gov"
     }
  },
```

https://developers.google.com/search/docs/data-types/dataset#complete-example-dataset-markup-in-json-ld

Metadata standards

- **DDI**: Documenting data Initiative
  - **heavy**weight metadata markup
  - only indexed by custom search engines for datasets they store
  - few open implementations
  - I guess some humans can read this
  - hard to like, hard for small teams to get into



```
<?xml version="1.0" encoding="utf-8"?>
<ddi:FragmentInstance xmlns:ddi="ddi:instance:3_2">
  <ddi:TopLevelReference>
    <Agency xmlns="ddi:reusable:3_2">example.org</Agency>
    <ID xmlns="ddi:reusable:3_2">88d12d54-c6ee-496c-b591-2b52071
    <Version xmlns="ddi:reusable:3_2">1</Version>
    <TypeOfObject xmlns="ddi:reusable:3_2">DDIInstance</TypeOfOb
  </ddi:TopLevelReference>
  <ddi:Fragment>
    <ddi:DDIInstance isUniversallyUnique="true" versionDate="201
      <URN xmlns="ddi:reusable:3_2">urn:ddi:example.org:88d12d54
      <Agency xmlns="ddi:reusable:3_2">example.org</Agency>
      <ID xmlns="ddi:reusable:3_2">88d12d54-c6ee-496c-b591-2b520
      <Version xmlns="ddi:reusable:3_2">1</Version>
      <Citation xmlns="ddi:reusable:3_2">
        <Title>
          <String xml:lang="en-US">CBS News/New York Times Month
        </Title>
        <PublicationDate>
          <SimpleDate>2010-03-31T00:00:00</SimpleDate>
        </PublicationDate>
        <InternationalIdentifier>
          <IdentifierContent>2079</IdentifierContent>
          <ManagingAgency>en-US</ManagingAgency>
        </InternationalIdentifier>
      </Citation>
      <ResourcePackageReference xmlns="ddi:reusable:3_2">
        <Agency>example.org</Agency>
        <ID>265fcdd2-9a9a-4be3-b84c-c433a3233ecd</ID>
        <Version>1</Version>
        <TypeOfObject>ResourcePackage</TypeOfObject>
      </ResourcePackageReference>
      <StudyUnitReference xmlns="ddi:reusable:3_2">
        <Agency>example.org</Agency>
        <ID>bb69605e-50b5-4618-8207-2c4d387b6e48</ID>
        <Version>1</Version>
        <TypeOfObject>StudyUnit</TypeOfObject>
      </StudyUnitReference>
    </ddi:DDIInstance>
  </ddi:Fragment>
  <ddi:Fragment>
    <ResourcePackage isUniversallyUnique="true" versionDate="201
      <URN xmlns="ddi:reusable:3_2">urn:ddi:example.org:265fcdd2
      <Agency xmlns="ddi:reusable:3_2">example.org</Agency>
      <ID xmlns="ddi:reusable:3_2">265fcdd2-9a9a-4be3-b84c-c433a
      <Version xmlns="ddi:reusable:3_2">1</Version>
      <UserAttributePair xmlns="ddi:reusable:3_2">
        <AttributeKey>extension:CodeListReferences</AttributeKey
        <AttributeValue>["urn:ddi:example.org:382be111-50c4-46cc
1","urn:ddi:example.org:8bd6833b-a7f8-4b7e-ad8c-3dff69e64aa0:1",
996f88ce-07c1-403e-9cbb-839228be0c73:1","urn:ddi:example.org:d41
bac2-268d119239a6:1","urn:ddi:example.org:9ed803f3-79ad-4348-abb
e5fd-410d-8315-0f95b6323137:1","urn:ddi:example.org:e2de2245-03e
fd778baf4c5e:1","urn:ddi:example.org:5ea4b8e0-647b-49b8-91e2-997
1","urn:ddi:example.org:3770bd02-75bc-43cf-bb1f-7a133a3c26f7:1",
23b06c96-bdad-4030-8f11-5523693198f4:1","urn:ddi:example.org:d98
85bf84af-3fbd-4786-9b24-71c11af52a97:1","urn:ddi:example.org:c67
fe0b638b23d0:1","urn:ddi:example.org:7290b31b-4860-4aa3-8cb0-d51
ad36-436ec2d298a2:1","urn:ddi:example.org:eae96440-c166-4393-855
1","urn:ddi:example.org:f54a5703-c9b4-4ce5-8788-d301fb43231f:1",
6179df9d-a752-4935-a90c-0ba2a94df40f:1","urn:ddi:example.org:550
b258-4f872e66fef4:1","urn:ddi:example.org:710714d2-b8db-4b9b-bfa
e1d348f5bee9:1","urn:ddi:example.org:5ce7fdeb-a001-4248-9109-081
f52073e46718:1","urn:ddi:example.org:e376c10d-0640-4ea4-9dd8-534
dd463fd0a9b4:1","urn:ddi:example.org:8aa0dd7b-d780-41c3-92df-3a2
e8ee1092dc48:1","urn:ddi:example.org:08bf5adc-cbb2-4388-a839-5b7
cc2e1c1bca54:1","urn:ddi:example.org:8b65e1af-b55e-4fc9-931f-dd8
      </UserAttributePair>
      <Citation xmlns="ddi:reusable:3_2" />
      <DataCollectionReference xmlns="ddi:reusable:3_2">
        <Agency>example.org</Agency>
        <ID>feeb5048-da4b-4817-867b-19178761def9</ID>
        <Version>1</Version>
        <TypeOfObject>DataCollection</TypeOfObject>
      </DataCollectionReference>
      <LogicalProductReference xmlns="ddi:reusable:3_2">
        <Agency>example.org</Agency>
```

Getting variable and value labels into R

- don’t use foreign
- readstata13 is a bit too hard to use
- Easy to install and remember (uses haven under the hood):
    - `jugendl <- rio::import(“jugendl.dta")`
- Even easier if you use formr.org:
    - `s1_demo <- formr_results("s1_demo")`

Try it out

- rubenarslan.ocpu.io/codebook (toy files)
- tiny.cc/cbocpu (slightly bigger files)
- `devtools::install_github("rubenarslan/codebook") # big files`

Examples

Similar efforts

- dataMaid package (focused more on finding errors in data)
- emldown/dataspice package (focused a bit more on ecological data)
- paneldata.org, DDI On Rails (hard to set up for simpler projects, not available in R?)
- ICPSR and other big players (mostly proprietary?)

Future plans

- Addin in RStudio to quickly find variables
- Additional plots, correlograms
- DDI support with DDIwR?
- Your suggestions?