



# Elevating Data Documentation Proficiency

Harnessing the Power of DDI for Official Statistics

Adrian Duşa  
University of Bucharest  
2023-12-12

# Slides location

<https://github.com/dusadrian/uRos2023>

# Typical data

```
dfm <- data.frame(  
  V1 = sample(c(1:5, -91), 10, replace = TRUE),  
  V2 = sample(c(1:2, -92), 10, replace = TRUE)  
)  
dfm
```

	V1	V2
1	-91	2
2	3	-92
3	2	2
4	4	2
5	2	1
6	5	-92
7	3	2
8	2	2
9	-91	1
10	3	2

What are the variables all about?

What do the values stand for?

Why are there negative numbers there?

Metadata powered data

# Variable and value labels

```
library(declared)
dfm$V1 <- declared(
  dfm$V1,
  label = "Opinion about this tutorial",
  labels = c(Bad = 1, Good = 5, DK = -91),
  na_values = -91
)
```

```
dfm$V1
```

```
<declared<numeric>[10]> Opinion about this tutorial
 [1] NA(-91)      3      2      4      2      5      3      2 NA(-91)      3
Missing values: -91
```

Labels:

value	label
1	Bad
5	Good
-91	DK

# Variable and value labels

```
dfm$V2 <- declared(  
  dfm$V2,  
  label = "Respondent's gender",  
  labels = c(Males = 1, Females = 2, NR = -92),  
  na_values = -92  
)
```

```
dfm$V2
```

```
<declared<numeric>[10]> Respondent's gender  
 [1]          2 NA(-92)          2          2          1 NA(-92)          2          2          1          2  
Missing values: -92
```

Labels:

value	label
1	Males
2	Females
-92	NR

# All others have it



STATA



# More typical questions

When was the data collected?

Under which study?

For what purpose?

From which population?

What do the cases represent? (people, companies etc.)

What was the sampling procedure?

If any, how were the weight variables calculated?





# DDI - Data Documentation Initiative

# What is DDI

<https://ddialliance.org/>

A metadata standard used to describe research data.

An international standard for describing the data produced by (but is not limited to) **surveys** and other observational methods in the social, behavioral, and economic (SBE) sciences, health sciences, and in **official statistics**

Used to document and manage different stages in the research **data lifecycle**, such as conceptualization, collection, processing, distribution/dissemination, discovery, and archiving.

Has a great community of researchers who want their research data to be:

**F**indable  
**A**ccessible  
**I**nteroperable  
**R**eusable

# What is DDI

- DDI facilitates the creation and use of metadata
- It is expressed in **XML**, and validated against an XML Schema.
- The XML schema supports the tagging of text (the metadata) for meaning, not formatting/appearance
- DDI establishes a clear boundary between the information required to define and understand data and the data itself

# DDI Products

## DDI Codebook

A light-weight version of the standard intended to document simple survey data.

[https://ddialliance.org/Specification/DDI-Codebook/2.5/XMLSchema/field\\_level\\_documentation.html](https://ddialliance.org/Specification/DDI-Codebook/2.5/XMLSchema/field_level_documentation.html)

## DDI Lifecycle

Next generation, *reusing* information from one study to another, from conceptualisation to analysis, making extensive use of *Controlled Vocabularies*, and monitoring the *change over time*

## DDI-CDI Cross-Domain Integration

Facilitates the joint analysis of data from multiple different domains (e.g. combining official statistics, administrative data, and data from environment agencies etc.)

# Interoperable/Aligned with Other Standards

- Dublin Core and MARC
- Generic Statistical Information Model (GSIM)
- ISO/IEC 11179 (Geography)
- ISO 19118 (Geography)
- ISO 17369 SDMX – Statistical Data and Metadata Exchange
- METS and PREMIS
- Structured Data Transformation Language (SDTL)
- Validation and Transformation Language (VTL)
- Others (you name it)

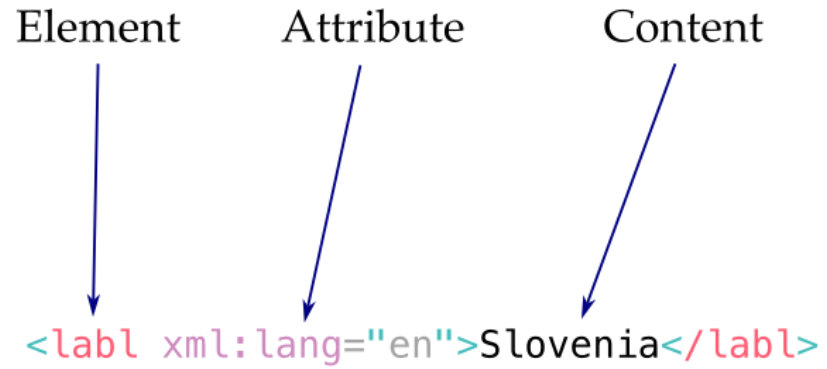
# DDI Codebook

# Five sections

1. Document Description
2. Study Description
3. Data Files Description
4. **Variable Description** (90% of the effort)
5. Other Study Related Materials



# Structure of an XML element



# Structure of an XML element

Parent element

Child element

Child element

```
<catgry>  
  <catValu>SI</catValu>  
  <labl xml:lang="en">Slovenia</labl>  
</catgry>
```

The diagram illustrates the structure of an XML element. It shows a parent element, <catgry>, which contains two child elements: <catValu>SI</catValu> and <labl xml:lang="en">Slovenia</labl>. The parent element is highlighted in red, and the child elements are highlighted in black. Blue arrows point from the labels 'Parent element', 'Child element', and 'Child element' to their respective elements in the XML code.

# Corresponding structure in R

```
$catgry  
$catgry$catValu  
$catgry$catValu[[1]]  
[1] "SI"
```

```
$catgry$labl  
$catgry$labl[[1]]  
[1] "Slovenia"
```

```
attr(,"xmlang")  
[1] "en"
```

Parent element

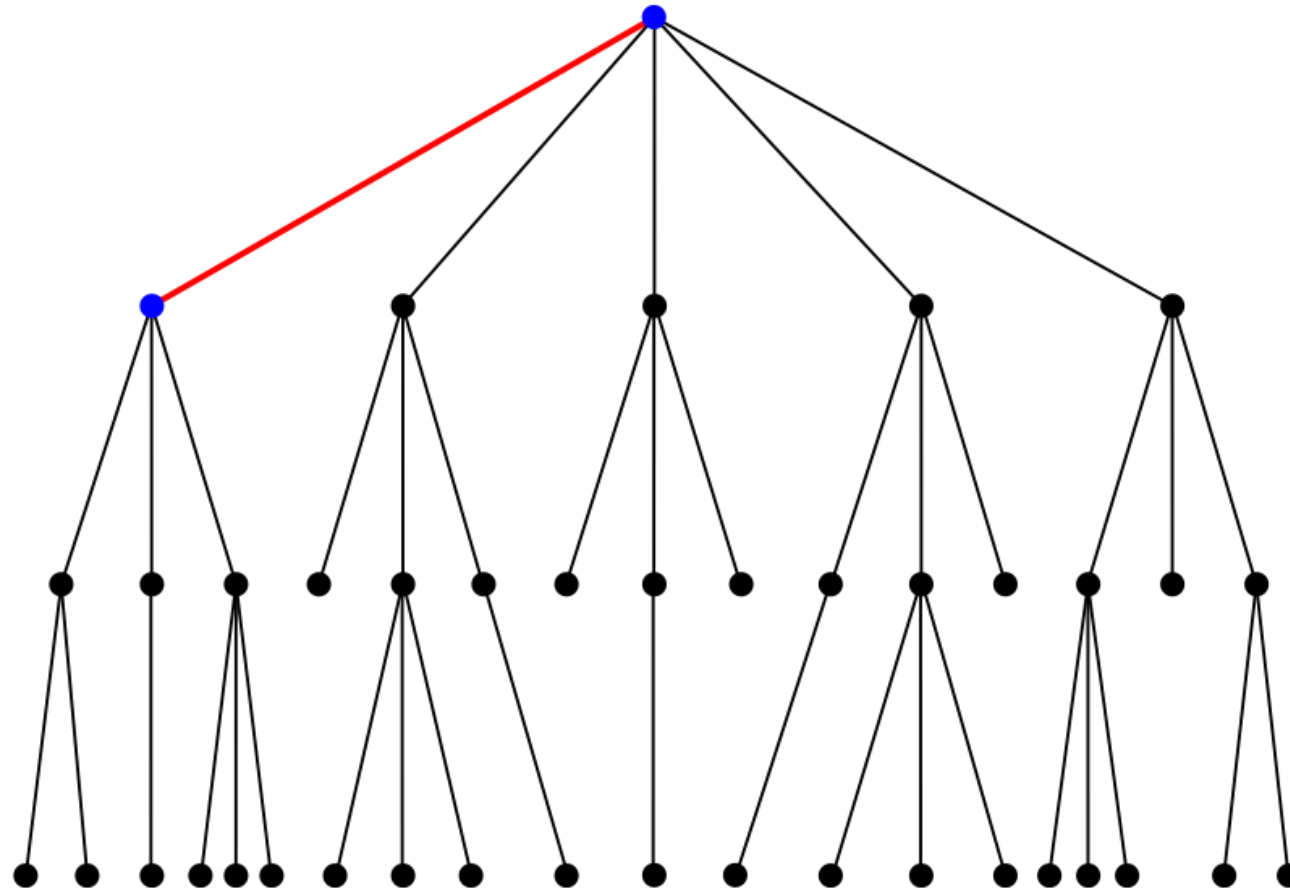
Child element

Child element

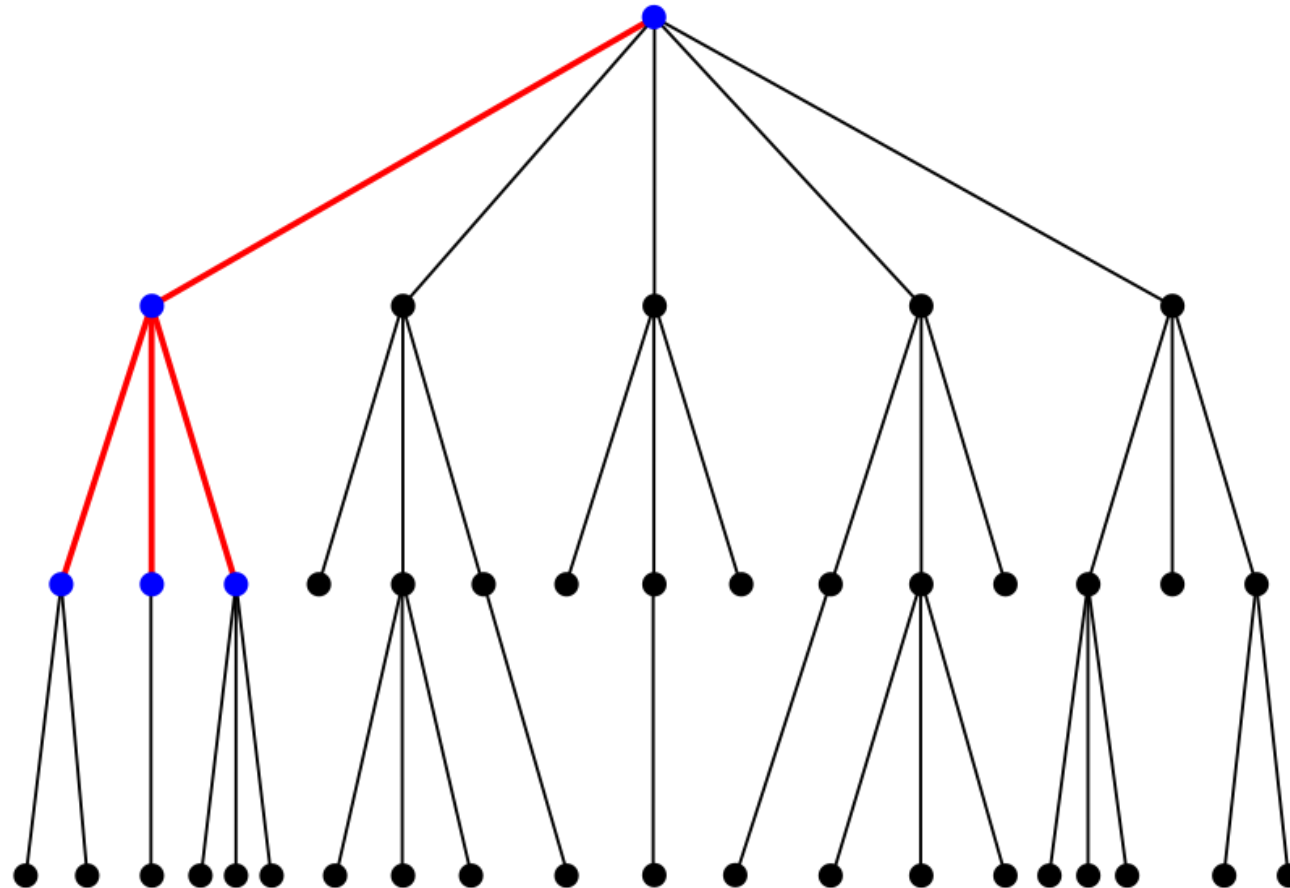
```
<catgry>  
  <catValu>SI</catValu>  
  <labl xml:lang="en">Slovenia</labl>  
</catgry>
```

The diagram illustrates the mapping between R code and XML structure. Three labels with arrows point to the XML code: 'Parent element' points to the opening <catgry> tag, the first 'Child element' points to the <catValu>SI</catValu> element, and the second 'Child element' points to the <labl xml:lang="en">Slovenia</labl> element.

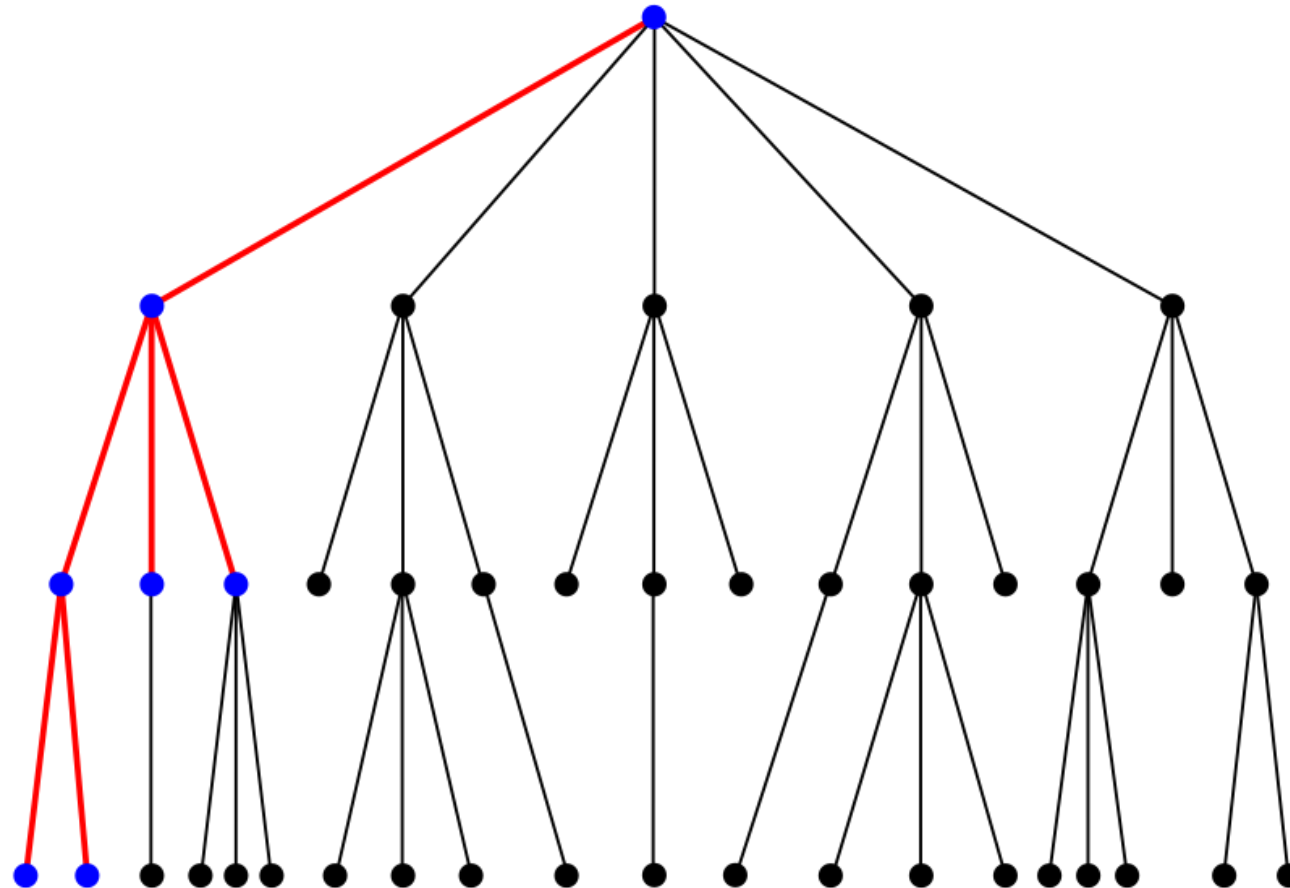
# Top-down approach



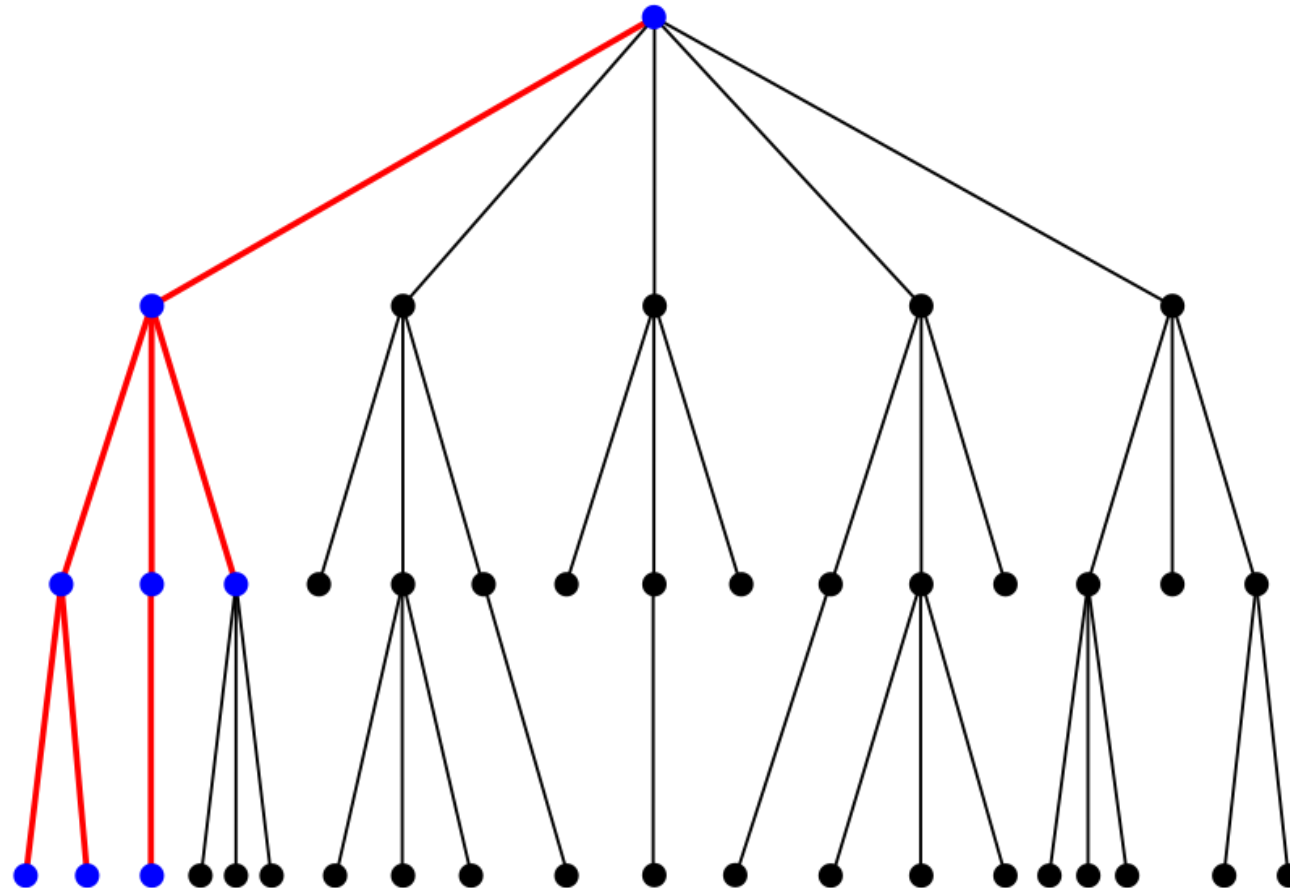
# Top-down approach



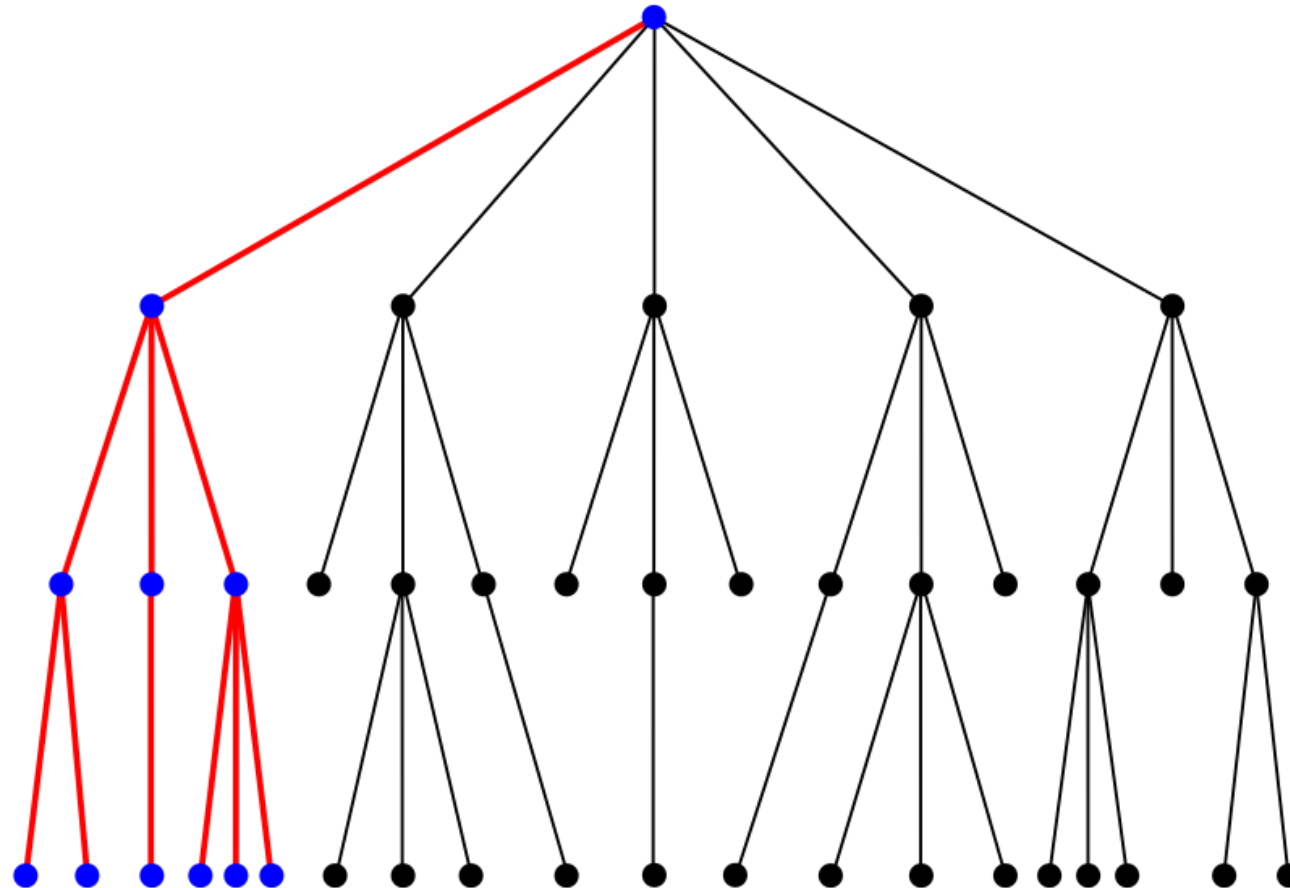
# Top-down approach



# Top-down approach

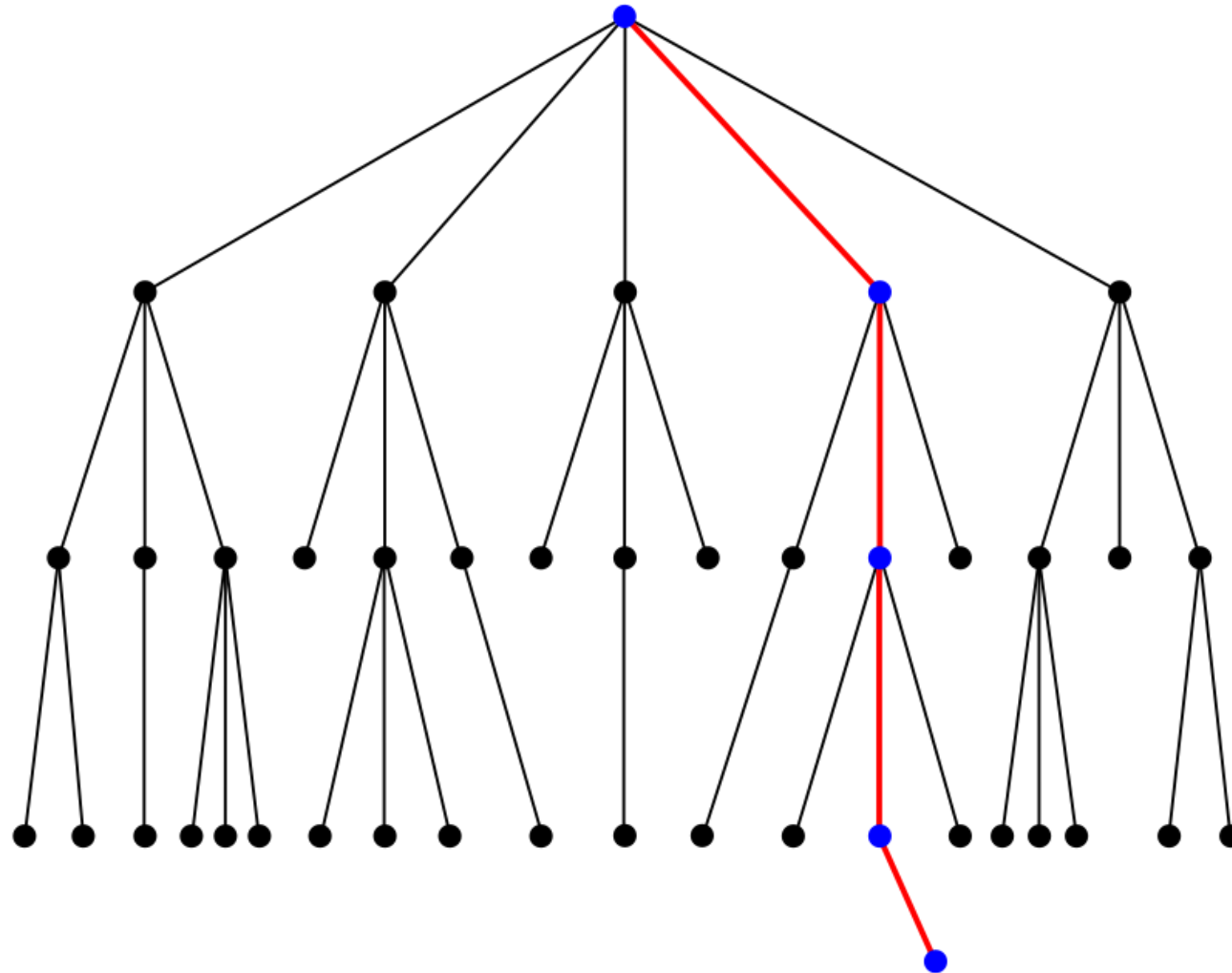


# Top-down approach

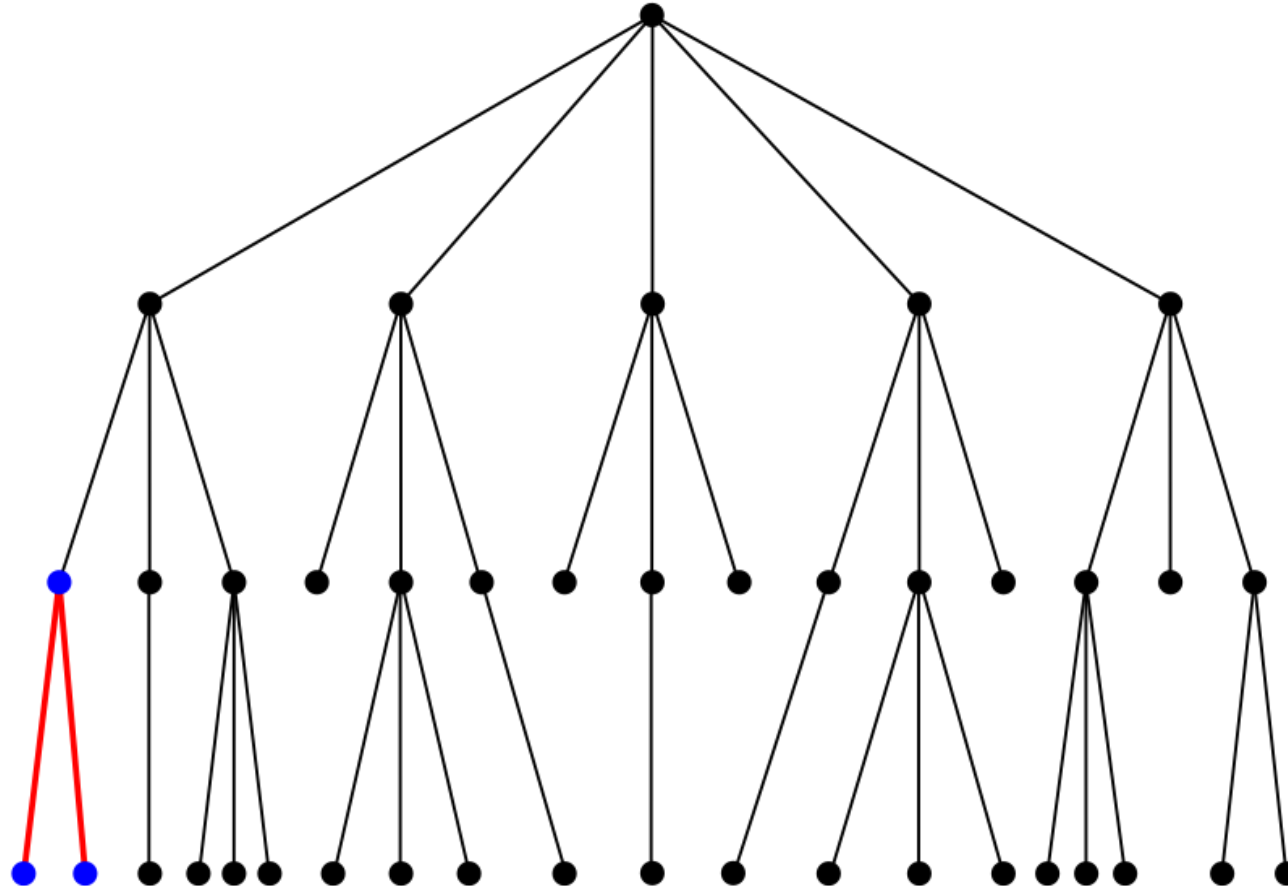




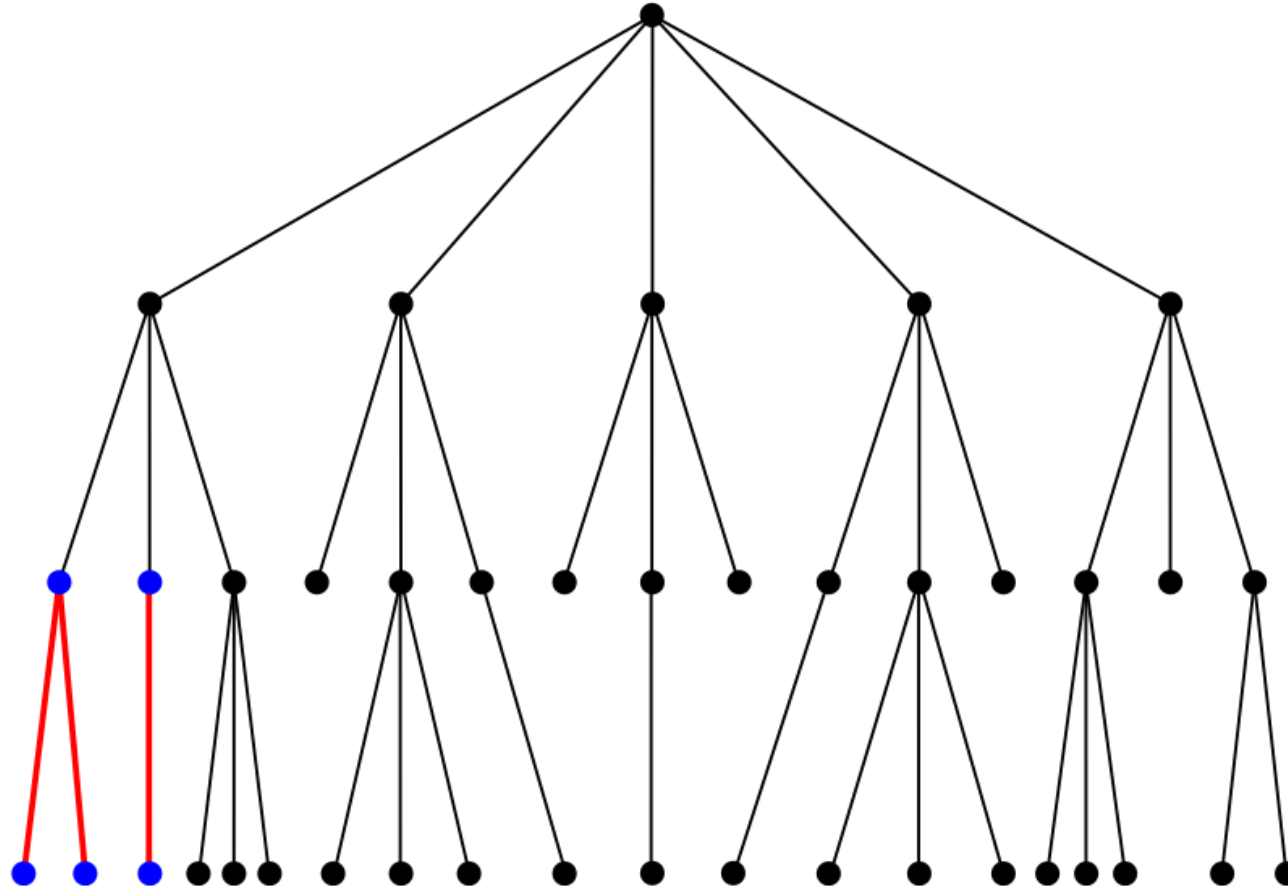
# Top-down approach



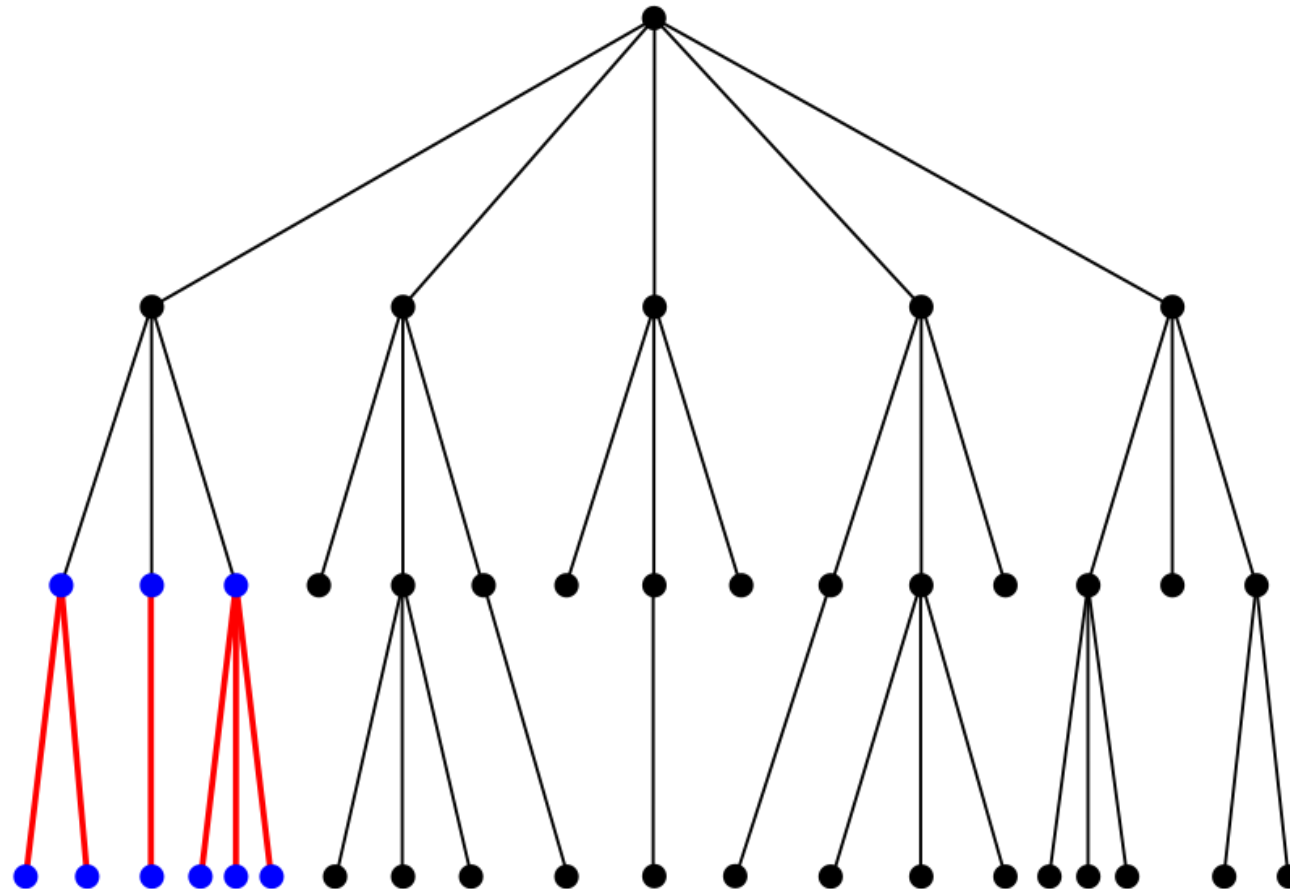
# Bottom-up approach



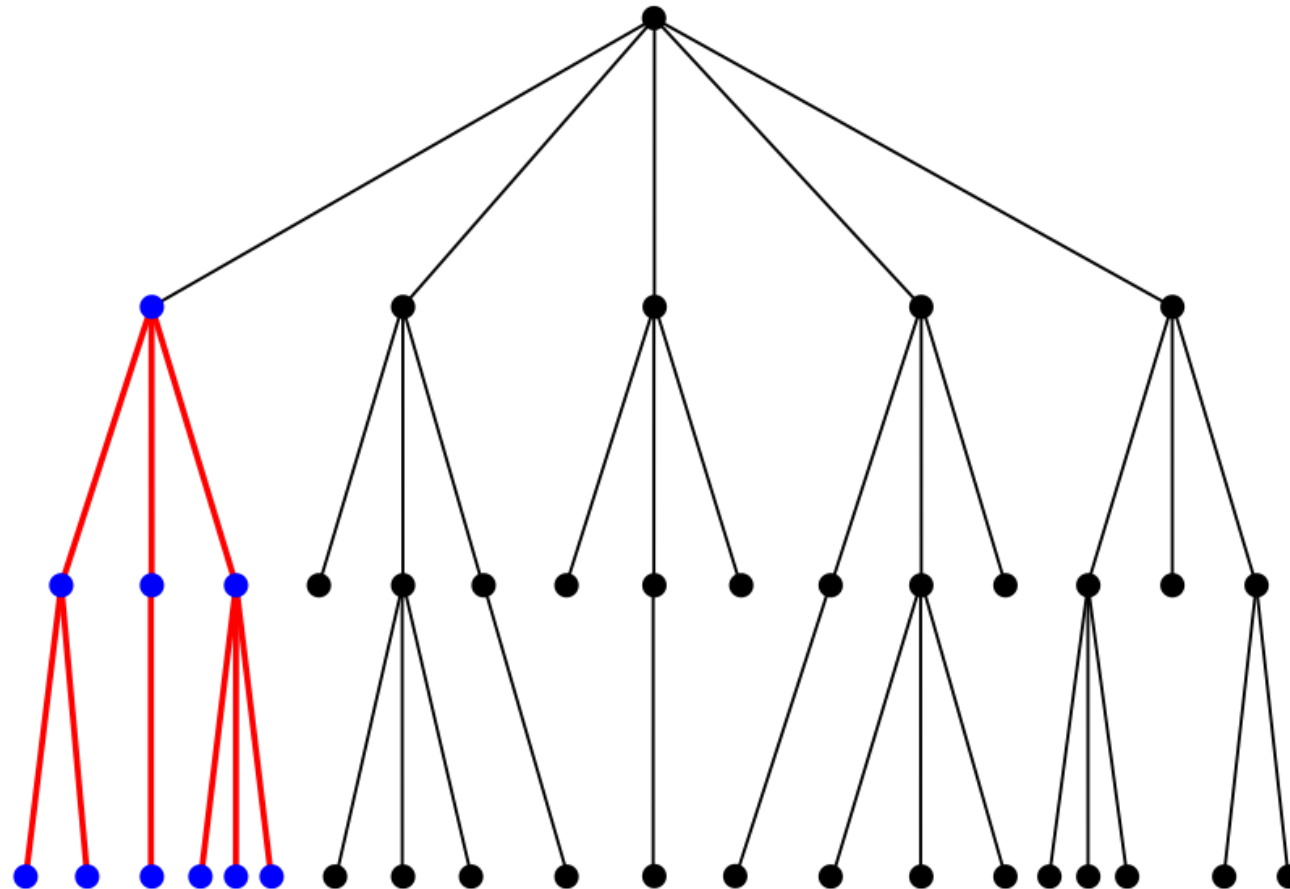
# Bottom-up approach



# Bottom-up approach



# Bottom-up approach





# Bottom-up R code

```
catValu <- makeElement("catValu", content = "SI")

labl <- makeElement(
  "labl",
  content = "Slovenia",
  attributes = c(xmlang = "en")
)

catgry <- makeElement(
  "catgry",
  children = list(catValu, labl)
)
```

```
<catgry>
  <catValu>SI</catValu>
  <labl xml:lang="en">Slovenia</labl>
</catgry>
```

# StatConverter

StatConverter

Encoding

Input type


Convert from

---

Output type

Convert to

---



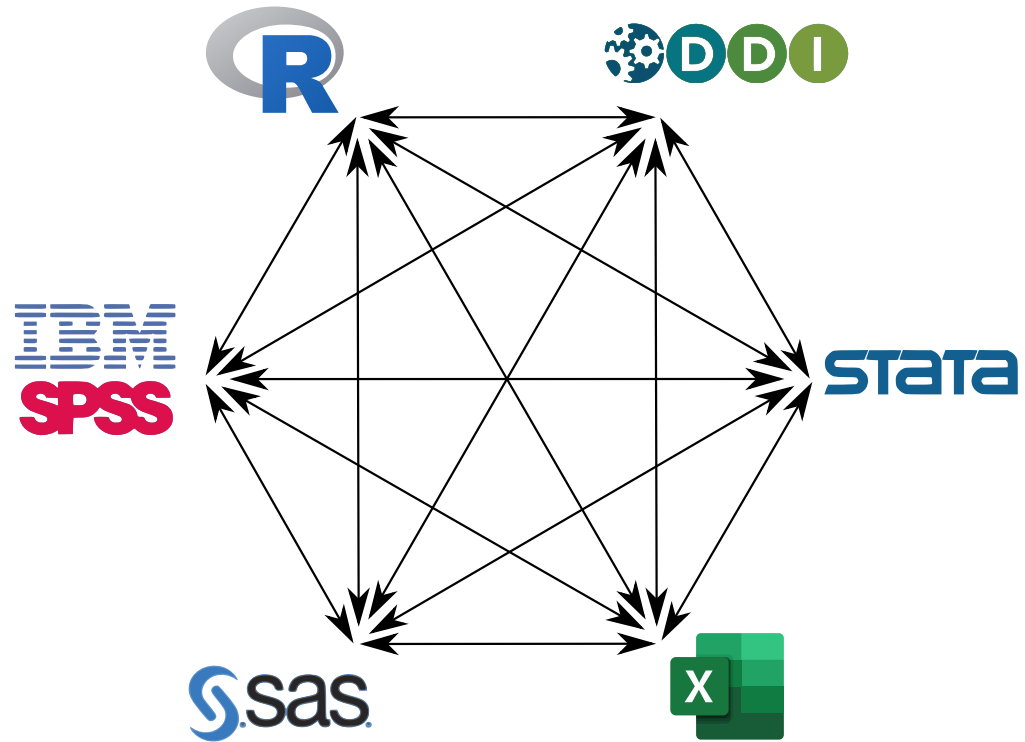
[Check for Updates](#)  
version 1.1.8



# R package DDIwR

- handles DDI Codebook v2.6
- much finer and grained control over the DDI XML output
- robust solution to the problem of categories and missing values in R
- imports and export to and from: SPSS, Stata, SAS, R and Excel
- capable of reading metadata from social science datasets
- translates the metadata into an R list, compatible with DDI's XML structure

# Achieves this



Hands on computers...