

PHUSE EU Connect 2022
Coding Tips & Tricks



Decoding in R

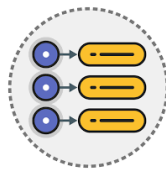
Himani Narang
Statistical Programmer
Novartis Healthcare Private Limited
Hyderabad, India

Disclaimer

- The views presented are the views of the presenter, not necessarily those of Novartis
- These slides are intended for educational purposes only and for the personal use of the audience. These slides are not intended for wider distribution outside the intended purpose without presenter approval
- The content of this slide deck is accurate to the best of the presenter's knowledge at the time of production

Storing Key-Value Pairs

Key	→	Value
Red	→	Santa
Golden	→	Star
White	→	Snowman
Green	→	Christmas Tree
Brown	→	Hot Chocolate
Orange	→	Fireplace



Associative Arrays

Maps

Symbol Tables

Dictionaries

R does not have a direct built-in data structure for storing key-value pairs.

Clinical Application

SUBJID	COUNTRY	COUNTRYL	COUNTRYN
1001	ALB	Albania	6
1002	AFG	Afghanistan	2
1003	ALA	Aland Islands	5

CODELIST Codelist name	KEY SDTM	VALE Extended	VALN Numeric
COUNTRY	ABW	Aruba	1
COUNTRY	AFG	Afghanistan	2
COUNTRY	AGO	Angola	3
COUNTRY	AIA	Anguilla	4
COUNTRY	ALA	Aland Islands	5
COUNTRY	ALB	Albania	6

Decoding values from codelists metadata as per CDISC submissions values.

Other examples:

- Convert raw RACE/ ETHNICITY/ SEX values to controlled terminology
- Derive PARAM/ PARAMCD/ PARAMN values
- Map VISITNUM values to VISIT values

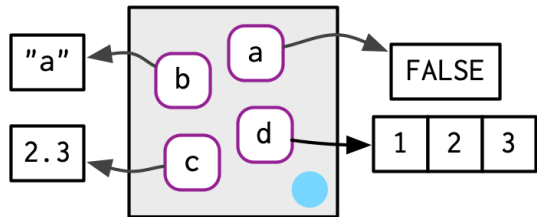
KEY

VALUE

Implementing Key-Value Pairs in R

1. R Hashed Environment

```
e1 <- env(  
  a = FALSE,  
  b = "a",  
  c = 2.3,  
  d = 1:3,  
)
```



2. Named Vectors/Lists

```
vec <- c('Apple', 'Banana', 'Carrot')  
names(vec) <- c('A', 'B', 'C')
```

```
> vec
```

```
      A      B      C  
"Apple" "Banana" "Carrot"
```

3. R Packages – dict, hash, ht

Package ‘Dict’

October 12, 2022

Title R6 Based Key-Value Dictionary Implementation

Package ‘hash’

October 13, 2022

Type Package

Title Full Featured Implementation of Hash Tables/Associative Arrays/Dictionaries

- 1 Package: ht
- 2 Type: Package
- 3 Title: Tiny Hash Table

Not a comparative analysis!

decode() Function

```
decode(codelist1,  
      variable,  
      intype,  
      outtype)
```

String value containing the name of the codelist to be used

Raw value as string / stored as variable e.g., "USA"

Column name of codelists_metadata that stores raw values

Column name of codelists_metadata that stores decoded values

Requirements:

- data.table package loaded
- codelists_metadata loaded in the environment as a data.table object

CODELIST Codelist name	KEY SDTM	VALE Extended	VALN Numeric	VALA Abbreviated
COUNTRY	ABW	Aruba	1	ABW
COUNTRY	AFG	Afghanistan	2	AFG
COUNTRY	AGO	Angola	3	AGO
COUNTRY	AIA	Anguilla	4	AIA
COUNTRY	ALA	Aland Islands	5	ALA
COUNTRY	ALB	Albania	6	ALB

decode() Function

```
library(data.table)
```

```
library(haven)
```

```
codelist_metadata = data.table(read_sas('codelist_metadata.sas7bdat'))
```

```
# Original value
```

```
origval = 'IND'
```

```
# Decoding to extended values
```

```
decodedval = decode(codelist1 = 'COUNTRY',  
                    variable = origval,  
                    intype = 'KEY',  
                    outtype = 'VALE')
```

```
# Printing decoded values
```

```
decodedval
```

```
[1] "India"
```

```
# Decoding to numeric values
```

```
decodedvaln = decode(codelist1 = 'COUNTRY',  
                     variable = origval,  
                     intype = 'KEY',  
                     outtype = 'VALN')
```

```
# Printing decoded values
```

```
decodedvaln
```

```
[1] 104
```

decode() Function

```
decode <- function(codelist1, variable, intype, outtype){  
  
  # Step 1 - Filter  
  codelist_metadata2 = codelist_metadata[codelist == codelist1]  
  
  # Step 2 - Assignment  
  temp = setNames(codelist_metadata2[[outtype]], codelist_metadata2[[intype]])  
  
  # Step 3 - Retrieval  
  out = temp[variable]  
  
  # If no match found, assign blank or NA  
  if(is.null(out) == T && mode(codelist_metadata2[[outtype]]) == 'character'){  
    out = '' } else if  
    (is.null(out) == T && mode(codelist_metadata2[[outtype]]) == 'numeric'){ out =  
    NA }  
  
  return(out)  
}
```

Name	Type	Value
temp	character [250]	'Aruba' 'Afghanistan' 'Angola' 'Anguilla' 'Aland Islands' 'Albania' ...
ABW	character [1]	'Aruba'
AFG	character [1]	'Afghanistan'
AGO	character [1]	'Angola'
AIA	character [1]	'Anguilla'
ALA	character [1]	'Aland Islands'
ALB	character [1]	'Albania'
AND	character [1]	'Andorra'
ARE	character [1]	'United Arab Emirates'
ARG	character [1]	'Argentina'

Decoding Multiple Columns At Once

Vector of new column names to be created from decode function

```
cols_after_decode = c('COUNTRYL', 'COUNTRYN', 'COUNTRYYS', 'ETHNICL', 'ETHNICN',  
                      'ETHNICS', 'RACEL', 'RACEN', 'RACES', 'SEXL', 'SEXN', 'SEXS')
```

	COUNTRY	ETHNIC	RACE	SEX
1	USA	NOT HISPANIC OR LATINO	WHITE	M
2	USA	HISPANIC OR LATINO	WHITE	F
3	USA	HISPANIC OR LATINO	WHITE	F
4	USA	NOT HISPANIC OR LATINO	WHITE	M
5	USA	HISPANIC OR LATINO	WHITE	F
6	USA	NOT HISPANIC OR LATINO	BLACK OR AFRICAN AMERICAN	F
7	USA	NOT HISPANIC OR LATINO	BLACK OR AFRICAN AMERICAN	F

The apply() Family

apply() mapply()
lapply() rapply()
sapply() tapply()
vapply()

Decoding Multiple Columns At Once

```
# Vector of new column names to be created from decode function
cols_after_decode = c('COUNTRYL', 'COUNTRYN', 'COUNTRYS', 'ETHNICL', 'ETHNICN',
                      'ETHNICS', 'RACEL', 'RACEN', 'RACES', 'SEXL', 'SEXN', 'SEXS')

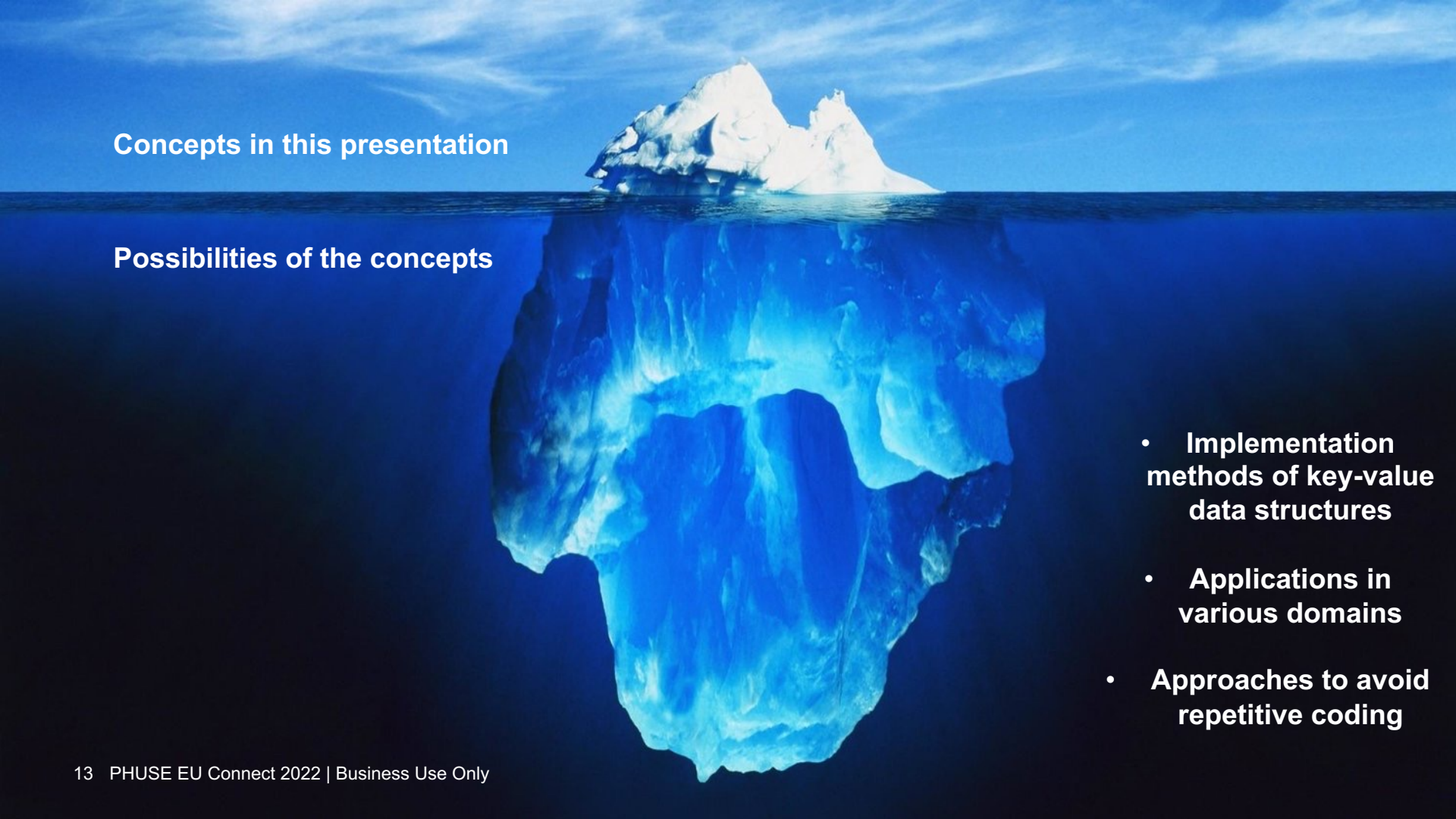
# Creating all decoded values in a single step
ads1 = dm[, (cols_after_decode) :=
            mapply(decode,
                    codelist1 = c(rep('COUNTRY', 3), rep('ETHNIC', 3),
                                rep('RACE', 3), rep('SEX', 3)),
                    variable = c(rep(COUNTRY, 3), rep(ETHNIC, 3),
                                rep(RACE, 3), rep(SEX, 3)),
                    intype = 'KEY',
                    outtype = c('VALE', 'VALN', 'VALA'),
                    SIMPLIFY = FALSE),
            by = row.names(dm)]
```

Decoding Multiple Columns at Once

Vector of new column names to be created from decode function

```
cols_after_decode = c('COUNTRYL', 'COUNTRYN', 'COUNTRYS', 'ETHNICL', 'ETHNICN',  
                      'ETHNICS', 'RACEL', 'RACEN', 'RACES', 'SEXL', 'SEXN', 'SEXS')
```

	COUNTRYL	COUNTRYN	COUNTRYS	ETHNICL	ETHNICN	ETHNICS	RACEL	RACEN	RACES	SEXL	SEXN	SEXS
1	United States	232	USA	Not Hispanic or Latino	281	NONHIS	White	90	WH	Male	1	M
2	United States	232	USA	Hispanic or Latino	100	HIS	White	90	WH	Female	2	F
3	United States	232	USA	Hispanic or Latino	100	HIS	White	90	WH	Female	2	F
4	United States	232	USA	Not Hispanic or Latino	281	NONHIS	White	90	WH	Male	1	M
5	United States	232	USA	Hispanic or Latino	100	HIS	White	90	WH	Female	2	F
6	United States	232	USA	Not Hispanic or Latino	281	NONHIS	Black or African American	70	BLAA	Female	2	F
7	United States	232	USA	Not Hispanic or Latino	281	NONHIS	Black or African American	70	BLAA	Female	2	F

A large iceberg floating in a blue ocean under a blue sky with light clouds. The tip of the iceberg is above the water, while the much larger, more complex structure is submerged below the surface. The water is a deep blue, and the sky is a lighter blue with wispy white clouds.

Concepts in this presentation

Possibilities of the concepts

- **Implementation methods of key-value data structures**
- **Applications in various domains**
- **Approaches to avoid repetitive coding**

Thank you

Do you have any questions?



Himani Narang

Statistical Programmer

Novartis Healthcare Private Limited

Hyderabad, India

Email: himani.narang@novartis.com