Data Structure

Kwang-Yeol Park

2018 03 15

References:
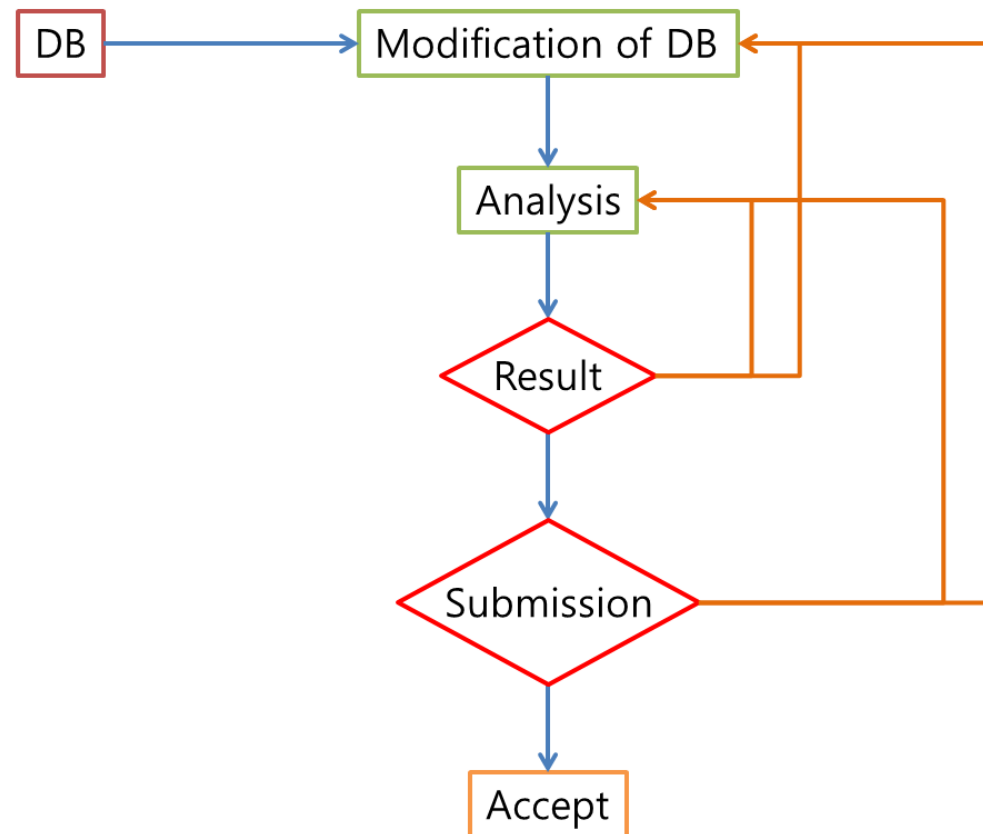
- http://r4ds.had.co.nz/

- MOOC: https://www.coursera.org/specializations/jhu-data-science

Report submission:

- sbaram1@naver.com

- Title: Student_ID, Name, Report_title

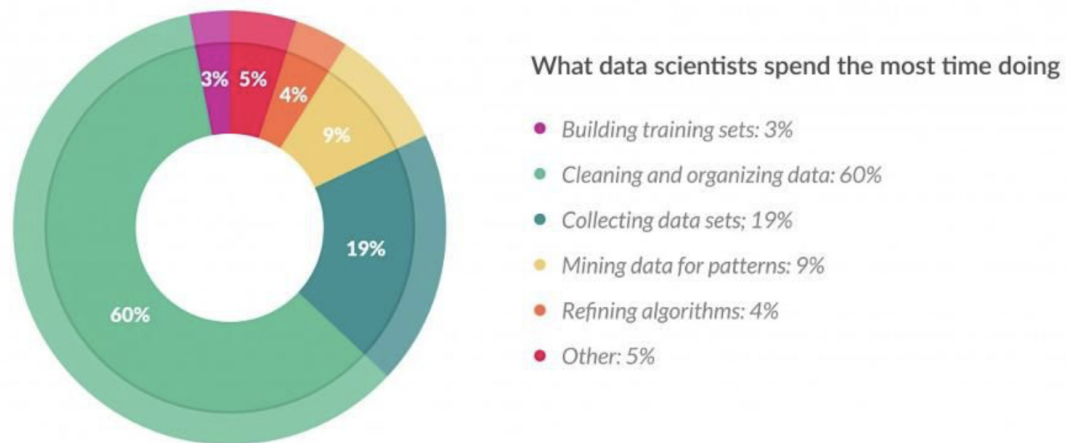Lecture material: https://github.com/sbaram1/Presentation

Research Process

Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says

A new survey of data scientists found that they spend most of their time massaging rather than mining or modeling data. Still, most are happy with having the sexiest job of the 21st century. The survey of about 80 data scientists was conducted for the second year in a row by CrowdFlower, provider of a "data enrichment" platform for data scientists. Here are the highlights:

Data preparation *accounts for about 80% of the work of data scientists*

What data scientists spend the most time doing

- Building training sets: 3%
- Cleaning and organizing data: 60%
- Collecting data sets; 19%
- Mining data for patterns: 9%
- Refining algorithms: 4%
- Other: 5%

```r
# install.packages("dplyr")
# install.packages("magrittr")

library(dplyr)
```

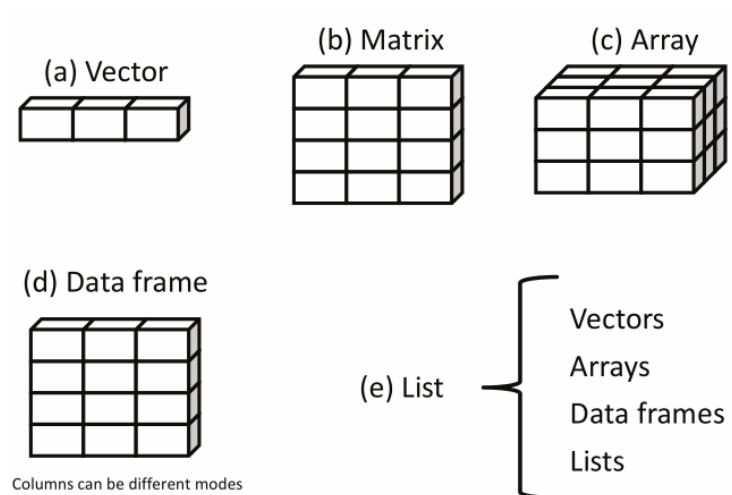# Data structure and importing DB

Data Preparation for analysis

· Entering data from the keyboard

```
a <- c(1, 2, 3, 4, 5)
a
```

```
## [1] 1 2 3 4 5
```

Data Structure

(a) Vector
(b) Matrix
(c) Array
(d) Data frame
Columns can be different modes
(e) List — Vectors, Arrays, Data frames, Lists

- Types
    - Logical
    - Integer
    - Numeric (Double)
    - Character

Matrix

```
mat1 <- matrix(data=c(1:6), nr=3, nc=2)
mat1
```

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    2    5
## [3,]    3    6
```

Data frame

```
ID <- c(11:15)
Age <- c(32, 35, 45, 21, 58)
Sex <- c("M", "M", "F", "M", "F")
DB <- data.frame(ID, Age, Sex)
DB
```

```
##    ID Age Sex
## 1 11  32   M
## 2 12  35   M
## 3 13  45   F
## 4 14  21   M
## 5 15  58   F
```

Data frame 2

```
DB[1,]
```

```
##   ID Age Sex
## 1 11  32   M
```

```
DB[,1]
```

```
## [1] 11 12 13 14 15
```

```
DB$ID
```

```
## [1] 11 12 13 14 15
```

```
str(DB)
```

```
## 'data.frame':    5 obs. of  3 variables:
##  $ ID : int  11 12 13 14 15
```

Factor

```
a <- c(1, 2, 2, 1, 2, 1, 1, 1, 1, 2)
class(a)
```

```
## [1] "numeric"
```

```
b <- factor(a)
b
```

```
##  [1] 1 2 2 1 2 1 1 1 1 2
## Levels: 1 2
```

```
class(b)
```

```
## [1] "factor"
```

Factor 2

```
?factor


b <- factor(a, levels=c(1, 2), labels=c("M", "F"))
b


## [1] M F F M F M M M F
## Levels: M F


# back_to_a <- as.numeric(as.character(b))
# droplevels()
```

Exercise 1

- Make a numeric vector
    - What is the 3rd element?
- Make a 3x3 matrix
    - What is in the 3rd row and column?
- make vector b (1, 2, 3, 7, 8)
    - turn b into factor b
    - turn b into numeric vector b again
- Make a data frame
    - variable name: ID, age, sex
    - number of rows: 5
    - sex should be factor
    - Display the content of ID

Managing data frame

Read Data file

- Importing DB files

    - CSV file (comma separated values)

    - SPSS file (read.spss() in foreign package, spss.get() in Hmisc package, read_sav in heven package)

    - MS-EXCEL file

```
# Check Working directory
# getwd()
DB1 <- read.csv("db/VitDdb_example.csv", sep=",", header=TRUE)

#install.packages("Hmisc")
# library(Hmisc)
# DB2 <- spss.get("db/VitDdb_example.sav", use.value.labels=TRUE)
# library(haven)
# VitDdb_example <- read_sav("~/Dropbox/RBook/ERC_KSS_R_lecture/2nd_20180209/db/VitDdb_example.sav")
```

Change the name of variable

```
names(DB1)
```

```
##  [1] "Age"               "Gender_F"          "TOAST.classification"
##  [4] "NIHSS.admission"    "mRS.admission"     "WBC"
##  [7] "Hemoglobin"         "hsCRP"             "FBS"
## [10] "T.Chol"             "HDL"               "LDL"
## [13] "TG"                 "VitD"
```

```
names(DB1)[4] <- c("NIHSS")
names(DB1)[which(names(DB1) == "mRS.admission")] <- c("Prev_mRS")
DB1 <- rename(DB1, TOAST = TOAST.classification)
# DB1 <- rename(DB1, NHISS = NIHSS.admission, Prev_mRS = mRS.admission,
#                 TOAST = TOAST.classification)
```

Look at TOAST

- set "Undetermined" as reference

DB1$TOAST

```
##    [1] CE            Undetermined    TIA     Undetermined
##    [5] LAA           CE              LAA     Undetermined
##    [9] TIA           Undetermined    TIA     TIA
##   [13] LAA           SVO             LAA     LAA
##   [17] LAA           LAA             TIA     CE
##   [21] SVO           CE              LAA     LAA
##   [25] Undetermined  TIA             LAA     CE
##   [29] LAA           LAA             TIA     LAA
##   [33] TIA           CE              CE      TIA
##   [37] CE            TIA             SVO     CE
##   [41] TIA           Undetermined    LAA     Undetermined
##   [45] LAA           SVO             LAA     CE
##   [49] LAA           SVO             SVO     CE
##   [53] LAA           CE              LAA     LAA
##   [57] Undetermined  LAA             TIA     LAA
##   [61] Undetermined  TIA             LAA     LAA
```

Change reference of a factor

```
levels(DB1$TOAST)
```

```
## [1] "CE"              "LAA"              "Other determined"
## [4] "SVO"             "TIA"              "Undetermined"
```

```
# library(magrittr)
```

```
DB1$TOAST <- factor(DB1$TOAST) %>% relevel(ref = "Undetermined")
# DB1$TOAST <- relevel(DB1$TOAST, ref = "Undetermined")
```

```
levels(DB1$TOAST)
```

```
## [1] "Undetermined"     "CE"              "LAA"
## [4] "Other determined" "SVO"             "TIA"
```

Making new variable

- New variable: SVO vs. non-SVO

DB1$TOAST

```
##    [1] CE              Undetermined    TIA          Undetermined
##    [5] LAA             CE              LAA          Undetermined
##    [9] TIA             Undetermined    TIA          TIA
##   [13] LAA             SVO             LAA          LAA
##   [17] LAA             LAA             TIA          CE
##   [21] SVO             CE              LAA          LAA
##   [25] Undetermined    TIA             LAA          CE
##   [29] LAA             LAA             TIA          LAA
##   [33] TIA             CE              CE           TIA
##   [37] CE              TIA             SVO          CE
##   [41] TIA             Undetermined    LAA          Undetermined
##   [45] LAA             SVO             LAA          CE
##   [49] LAA             SVO             SVO          CE
##   [53] LAA             CE              LAA          LAA
##   [57] Undetermined    LAA             TIA          LAA
##   [61] Undetermined    TIA             LAA          LAA
```

```
DB1$SVO <- ifelse(DB1$TOAST == "SVO", 1, 0) %>%
            factor(levels=c(0, 1), labels=c("SVO(-)", "SVO(+)"))
DB1$SVO
```

```
##   [1] SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-)
##  [11] SVO(-) SVO(-) SVO(-) SVO(+) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-)
##  [21] SVO(+) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-)
##  [31] SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(+) SVO(-)
##  [41] SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(+) SVO(-) SVO(-) SVO(-) SVO(+)
##  [51] SVO(+) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-)
##  [61] SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(+)
##  [71] SVO(-) SVO(-) SVO(+) SVO(-) SVO(+) SVO(-) SVO(-) SVO(-) SVO(+) SVO(-)
##  [81] SVO(-) SVO(+) SVO(-) SVO(-) SVO(-) SVO(-) SVO(+) SVO(-) SVO(-) SVO(-)
##  [91] SVO(-) SVO(-) SVO(+) SVO(-) SVO(-) SVO(+) SVO(+) SVO(-) SVO(-) SVO(-)
## [101] SVO(+) SVO(-) SVO(-) SVO(+) SVO(+) SVO(+) SVO(-) SVO(-) SVO(-) SVO(-)
## [111] SVO(-) SVO(-) SVO(-) SVO(+) SVO(-) SVO(-) SVO(-) SVO(+) SVO(-) SVO(-)
## [121] SVO(-) SVO(-) SVO(-) SVO(+) SVO(+) SVO(-) SVO(+) SVO(-) SVO(-) SVO(-)
## [131] SVO(+) SVO(-) SVO(+) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-)
## [141] SVO(-) SVO(-) SVO(+) SVO(-) SVO(-) SVO(+) SVO(-) SVO(+) SVO(-) SVO(+)
## [151] SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-)
## [161] SVO(-) SVO(-) SVO(-) SVO(-) SVO(+) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-)
## [171] SVO(-) SVO(+) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-)
## [181] SVO(+) SVO(-) SVO(-) SVO(-) SVO(+) SVO(-) SVO(-) SVO(-) SVO(-) SVO(-)
```

```
table(DB1$TOAST, DB1$SVO)
```

```
##
##                     SVO(-) SVO(+)
##   Undetermined       121      0
##   CE                 202      0
##   LAA                186      0
##   Other determined    16      0
##   SVO                  0    148
##   TIA                 86      0
```

Making Stroke subset

```
summary(DB1$TOAST)
```

```
##      Undetermined              CE          LAA Other determined
##               121             202          186                16
##               SVO             TIA
##               148              86
```

Using subset function

```
subset(DB1, TOAST != "TIA") %>% nrow()
```

```
## [1] 673
```

```
subset(DB1, TOAST == "TIA") %>% nrow()
```

```
## [1] 86
```

```
nrow(DB1)
```

```
## [1] 759
```

Exercise 2

- Use your own DB
  - Disply the summary of DB
  - Display the names of variables
  - Change variable name
  - Make a factor and change reference value
  - Make a new variable
  - Make a subset
- You can use data()