Lecture 07: Sep 14, 2018

Data Oddities

- *Data Structures*
- *Coercion*
- *Missingness and NULL*

James Balamuta
STAT 385 @ UIUC

Announcements

- **hw02** is due on **Friday, Sep 14th, 2018 @ 6:00 PM**
  - Want to talk to a human? Visit OHs in IH 104!
- **hw03** slated to be released Friday/Saturday evening
  - Due on **Friday, Sep 21st, 2018 @ 6:00 PM**
- **Quiz 04** covers Week 3 contents @ CBTF.
  - Window: Sep 18th - 20th
  - Sign up: https://cbtf.engr.illinois.edu/sched
  - Demo of CBTF Environment: https://www.youtube.com/watch?v=6oaPvo4TIFk&t=8s
- **hw01** grade reports released on GitHub.
  - Post on forum detailing how to interpret the grade reports.
  - Got caught using GitHub's web interface? Let's chat.

Last Time

- **Derived Variables**
  - Variables created from other variables in a data.frame
- **Comparisons**
  - Make a choice.
- **Logical Operators**
  - Combine multiple choices
- **Control Structures**
  - Analyze where a program should go.
  - Common structures: *if-else*, *if-else if-else*, vectorized *ifelse*, and *switch*

Lecture Objectives

- **Create** and **apply** different data structures in *R*

- Understanding the **effects of implicit** and **explicit coercion** on differing data types and structures.

- Familiarity with **missingness** in data.

Data Structures

Previously

Vectors

… **1 D**imensional **collections** of the **same** kind of **element** …

```
# Vector of character elements
character_values = c("James", "summer", "Hi guys!")

# Vector of numeric elements
numeric_values = c(3.14, 8.2, -1.4123, 0.333)

# Vector of integer elements
integer_values = c(4L, -7L, 52L, 98L)

# Create sequences: 1, 2, ... , 9, 10
integer_sequence = 1L:10L
```

Colon Operator

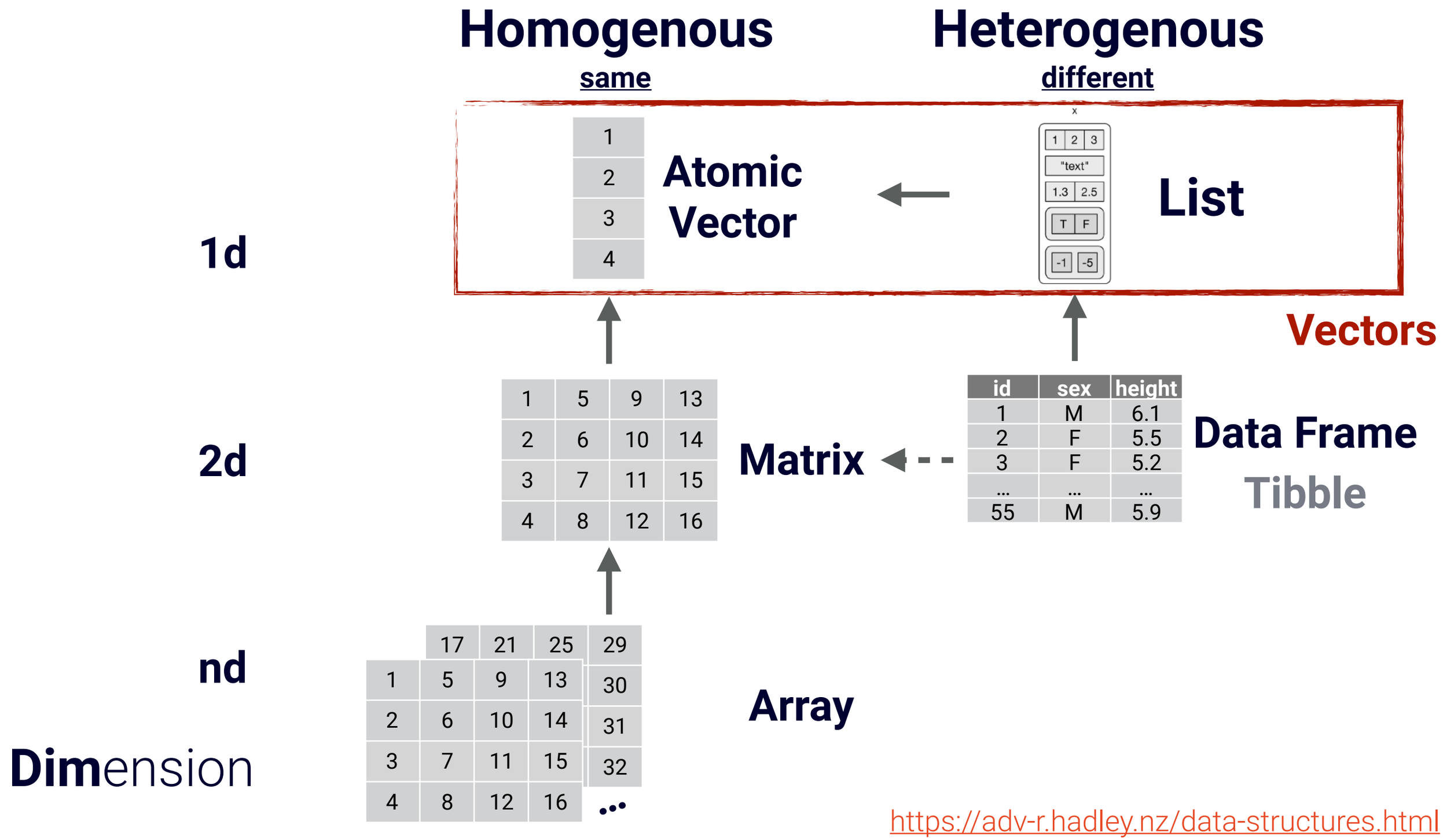
Previously

Everything in *R* is a

Vector

Data Structures

… how you can build data …



https://adv-r.hadley.nz/data-structures.html

4 Properties of a Vector

1. **Type**
"what it is"

2. **Length**
"amount of elements it contains"

3. **Attributes**
"additional arbitrary metadata"
… more later on about this …

4. **Class**
"blueprint"

https://adv-r.hadley.nz/data-structures.html#vectors

```
# Data
x = c(1, 2, 3, 4)

typeof(x)
# [1] "double"

length(x)
# [1] 4

attributes(x)
# NULL

class(x)
# [1] "numeric"
```

Checking Vector Types

… do **not** use is.vector() …

```
is.character(letters)          # Checks for characters
# [1] TRUE
is.double(c(1.2, 4.4))         # Checks for numerics (doubles)
# [1] TRUE
is.integer(c(1L, 5L))          # Checks for integers
# [1] TRUE
is.logical(c(TRUE, FALSE))  # Checks for logicals/booleans
# [1] TRUE
is.atomic(c(48, 21))           # Checks for atomic vector
# [1] TRUE
is.list(list(-2, 99))          # Checks for a generic vector
```

Atomic vectors must be flat
e.g. 1 dimensional (1d)

```
# Nested concatenation
c(1, c(2, c(3, 4)))
# [1] 1 2 3 4

# Traditional construction
c(1, 2, 3, 4)
# [1] 1 2 3 4
```

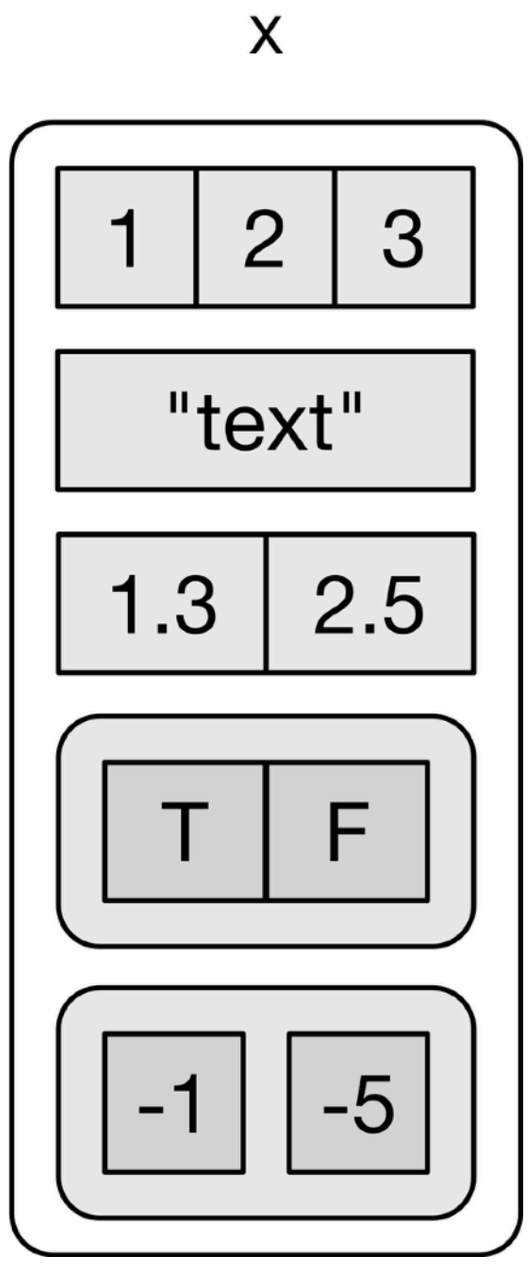
List

… **1 D**imensional *Heterogenous* …

```
x = list(c(1, 2, 3),
         "text",
          c(1.3, 2.5),
          list(c(TRUE, FALSE)),
          list(c(-1), c(-5))
)

length(x)
# [1] 5

dim(x)
# NULL
```



Matrix

... **2 D**imensional *Homogenous* ...

```
z = matrix(c(1, 2, 3, 4, 5, 6),
           nrow = 3,
           ncol  = 2
)

length(z)
# [1] 6

dim(z)
# [1] 3 2
```

$$z = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}_{3\times2}$$

Matrix Fill Order

… **2 D**imensional *Homogenous* …

Column (default)

$$z_{col} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}_{3\times2}$$

```
z_col = matrix(c(1, 2, 3, 4, 5, 6),
               nrow = 3,
               ncol  = 2,
               byrow = FALSE
)
```

Row

$$z_{row} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}_{3\times2}$$

```
z_row = matrix(c(1, 2, 3, 4, 5, 6),
               nrow = 3,
               ncol  = 2,
               byrow = TRUE
)
```

Coercion

Definition:
Coercion refers to converting values between data types.

https://www.youtube.com/watch?v=X5SkW7K0e3Y

Implicit Coercion

… automatic conversion (*R cares)* …

```
c(1.2, 4.4, -2.9)                    # Standard numeric vector
# [1] 1.2 4.4 -2.9

c(1.2, 4.4, "toad", -2.9)            # Introduce a character into a numeric vector
# [1] "1.2" "4.4" "toad" "-2.9"

c(1L, 10L, -1L)                      # Standard integer vector
# [1] 1 10 -1

c(1L, 4.4, 10L, -1L)                 # Introduce a numeric into an integer vector
# [1] 1.0 4.4 10.0 -1.0
```

Hierarchy of Implicit Conversion

… how *R* treats data …

logical →	integer →	numeric →	complex →	character
TRUE	1L	1.0	1.0 + 0i	"1.0 + 0i"
FALSE	0L	0.0	0.0 + 0i	"0.0 + 0i"
T	1L	1.0	1.0 + 0i	"1.0 + 0i"
F	0L	0.0	0.0 + 0i	"0.0 + 0i"
	42L	42.0	42.0 + 0i	"42.0 + 0i"
		32.9	32.9 + 0i	"32.9 + 0i"
			8.0 + 1.0i	"8.0 + 1.0i"
				"toad"

Explicit Coercion

… forcing data to one type …

```
as.character(c(TRUE, 1, 9.8))    # Force all types to character
# [1] "1"   "1"   "9.8"
as.integer(c(5.3, 8.8))          # Force numeric to integer
# [1] 5 8
as.logical(c(1L, 0L))            # Force to integer
# [1] TRUE FALSE
as.numeric(c(42L, 58L))          # Force to numeric
# [1] 42.0 58.0
```

Missingness

Definition:
Missingness indicates that no data has been recorded or was omitted. In *R*, we denote this by *NA*.

id	sex	height
1	M	6.1
2	F	5.5
3	F	5.2
…	…	…
55	M	5.9

Complete Cases

id	sex	height
1	M	6.1
2	F	*NA*
3	*NA*	5.2
…	…	…
55	*NA*	*NA*

Incomplete Cases

https://www.youtube.com/watch?v=1zwSO2uOYyI

Missingness *is*

Contagious & Propagates

* For checking missing values, use **is.na(value)** covered in a few slides.

```
# Operations with missingness (NA) yield more missingness (NA)...
NA + 2
# [1] NA

NA == 2
# [1] NA

12 - 2 + NA*5
# [1] NA

NA == NA *
# [1] NA
```

Injecting Missingness

id	sex	height
1	M	6.1
2	F	5.5
3	F	5.2
…	…	…
55	M	5.9

subject_heights

id	sex	height
1	M	6.1
2	F	*NA*
3	*NA*	5.2
…	…	…
55	*NA*	*NA*

subject_heights_na

```
# Add missingness to
# data frame
subject_heights_na =
  data.frame(
   id     = c(1, 2, 3, 55),
   sex    = c("M", "F", NA, NA),
   height = c(6.1, NA, 5.2, NA)
)
```

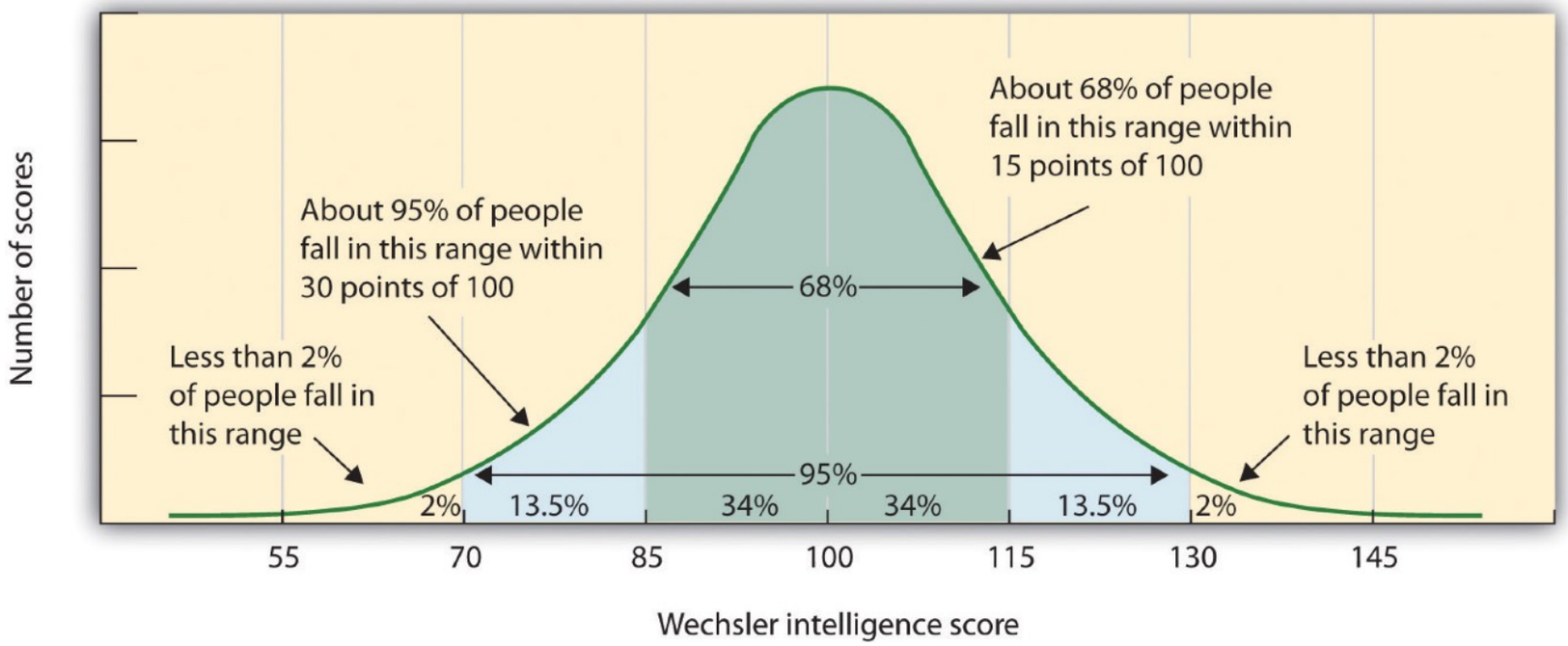
"An ***NA*** is the presence of an absence. Don't forget that some missing values are the absence of a presence."

– Hadley Wickham on Twitter

IQ Example

… dealing with test data …

IQ Range	IQ Classification
130 and above	Very superior
120–129	Superior
110–119	High average
90–109	Average
80–89	Low average
70–79	Borderline
69 and below	Extremely low



Source

Types of Missingness

… got data ???

Original

Age	IQ
18	112
19	108
19	94
22	87
25	132
28	79
30	103

Missing **C**ompletely **A**t **R**andom

Age	IQ
18	*NA*
19	108
19	94
22	87
25	*NA*
28	79
30	*NA*

????

Missing **A**t **R**andom

Age	IQ
18	*NA*
19	*NA*
19	*NA*
22	87
25	132
28	79
30	103

Non-response

Missing **N**ot **A**t **R**andom

Age	IQ
18	112
19	108
19	*NA*
22	*NA*
25	132
28	*NA*
30	103

Low IQ

* **MCAR** indicates that no relationship exists between missing values and any observed values

** **MAR** indicates a relationship exists between missing values and recorded values that can be inferred.

*** **MNAR** indicates a relationship exists between the value of the missing data.

```
# Detecting Missing Data

# Is there any observations with NA ?
checked_data = is.na(data_with_missing)
```

Age	IQ
18	*NA*
19	108
19	94
22	87
25	*NA*
28	79
30	*NA*

data_with_missing

Age	IQ
FALSE	TRUE
FALSE	FALSE
FALSE	FALSE
FALSE	FALSE
FALSE	TRUE
FALSE	FALSE
FALSE	TRUE

checked_data

```r
# Imputing Values
# ... assigning a value when missingness is abound ...

# Copy data
imputed_df = data_with_missing
# Create list of missing observations in IQ
index_na = is.na(data_with_missing$IQ)
# Impute (or set missing values to) the median of the data
imputed_df[index_na, "IQ"] = median(data_with_missing$IQ)
```

Age	IQ
18	*NA*
19	108
19	94
22	87
25	*NA*
28	79
30	*NA*

data_with_missing

index_na
TRUE
FALSE
FALSE
FALSE
TRUE
FALSE
TRUE

median
90.5
90.5
90.5

Age	IQ
18	*90.5*
19	108
19	94
22	87
25	*90.5*
28	79
30	*90.5*

imputed_df

```r
# Subsetting Missing Data
# By omitting any row with missingness...
data_present = na.omit(data_with_missing)

# By subsetting with logicals
# Create logical index of rows that are complete
index_complete = complete.cases(data_with_missing)
# Subset data frame with logical index
data_present = data_with_missing[index_complete, ]
```

Age	IQ
18	*NA*
19	108
19	94
22	87
25	*NA*
28	79
30	*NA*

data_with_missing

[FALSE, TRUE, TRUE, TRUE, FALSE, TRUE, FALSE ,]

index_complete

Subset brackets

=

Age	IQ
19	108
19	94
22	87
28	79

data_present

Let's add missing values to ...

twtr_stock_prices

time	price
09:30 AM	22.40
NA	22.38
09:50 AM	22.46
10:00 AM	*NA*

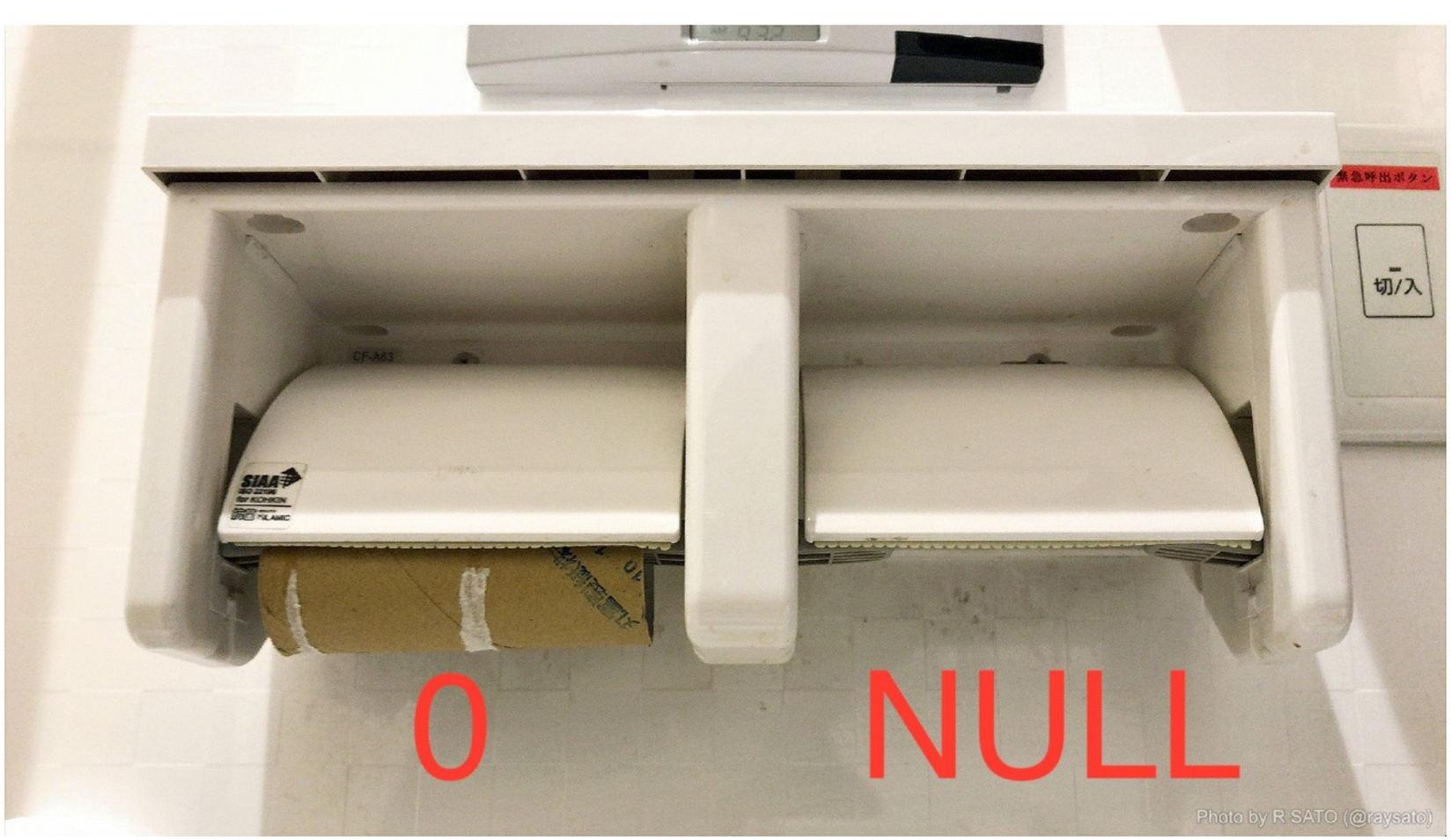
champaign_weather

date	temp	rain	wind
1/21	44	*NA*	*NA*
1/22	46	TRUE	19
1/23	*NA*	TRUE	*NA*
1/24	26	FALSE	*NA*
1/25	37	*NA*	14
1/26	44	FALSE	*NA*
1/27	*NA*	FALSE	12

TWTR Stock Price Jan 26, 2018

Champaign Weather Jan 21 - 27

Definition:
Null is a valueless object or an uninitialized object. (e.g. empty vector). In *R*, we denote this by NULL.



Source

NA is *not* **NULL**...

```
NA
# [1] NA

class(NA)
# [1] "logical"
```

```
NULL
# [1] NULL

class(NULL)
# [1] "NULL"
```

```
NA + 1
# [1] NA

c(NA, NULL, 3)
# [1] NA 3
```

```
NULL + 1
# numeric(0)

list(NA, NULL, 3)
# [[1]] [1] NA
# [[2]] NULL
# [[3]] [1] 3
```

Recap

- **Data Structures**
  - 1*D*, 2*D*, and n *D*imensions
  - Homogenous (Same) vs. Heterogenous (Different/Mixed)
- **Coercion**
  - Changing data from one form to the another either implicitly (R) or explicitly (You).
- **Missingness and NULL**
  - The lack of recorded data vs. the lack of an object being created.

Acknowledgements

Acknowledgements

- Hadley Wickham for various diagrams in his books.