Working with Data Frames

INFO 201

Joel Ross
Winter 2017

https://slides.com/joelross/info201w17-data-frames-ii/live

Today's Objectives

By the end of class, you should be able to

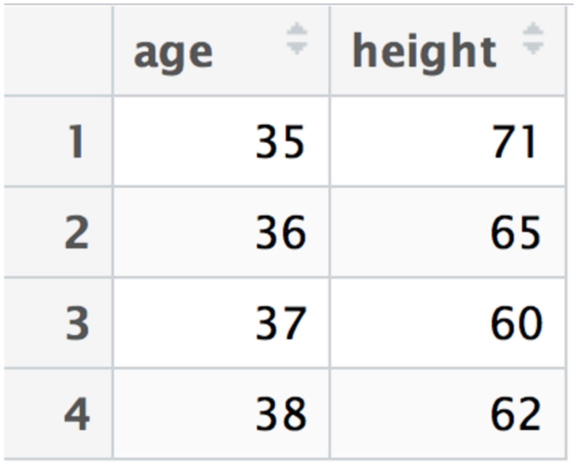
- Be comfortable working with data stored in **data frames**
- Load data sets from external **.csv** files in R
- Understand the purpose of **factors** in R
- Ask and answer simple questions about data sets

Review "Quiz"!

What is the difference between a **list** and a **vector**?

What is the difference between **single-bracket notation** and **double-bracket notation** when working with a **data frame**? With a **vector**?

How could you extract the value **35** from this data table?

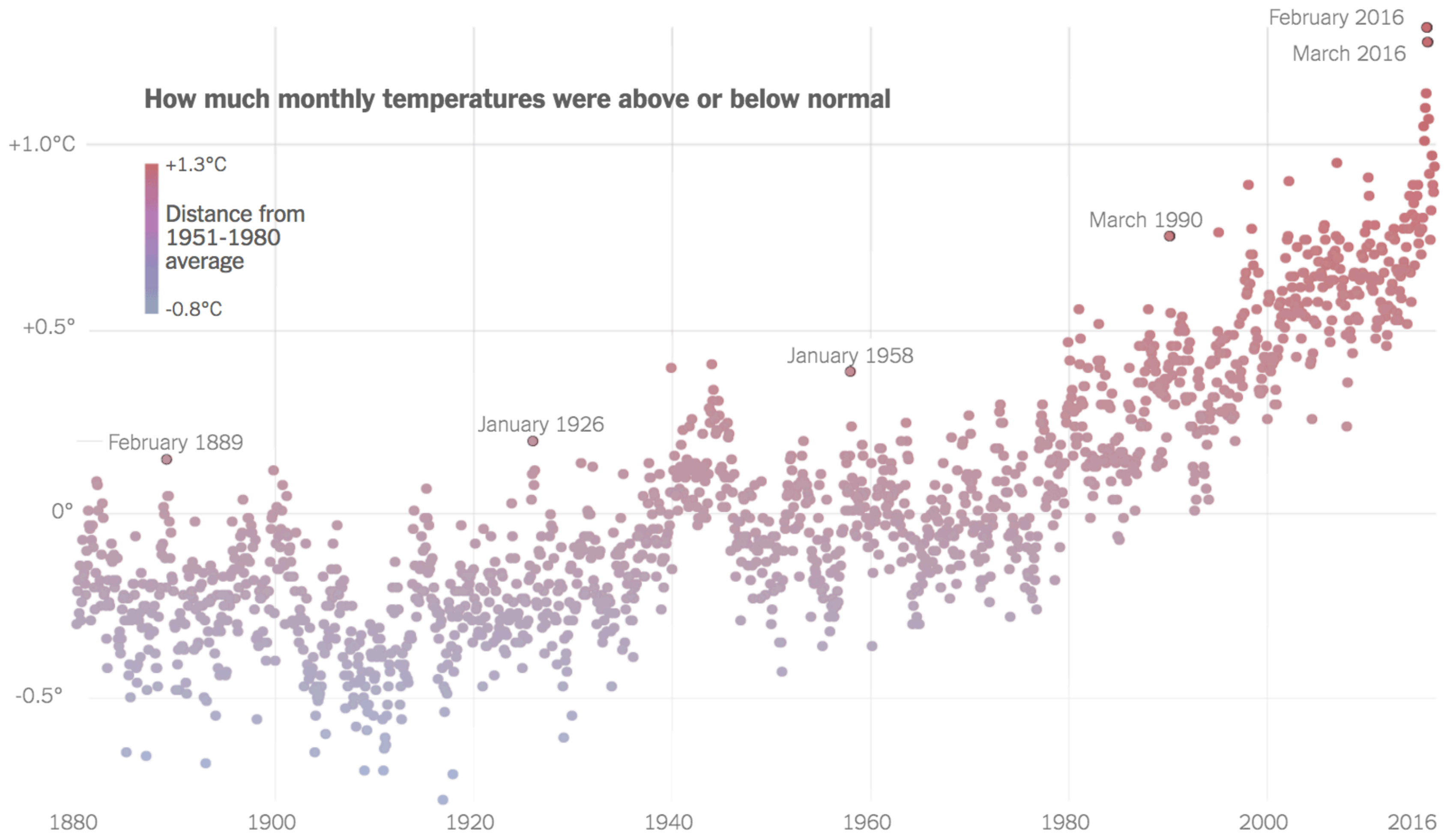


	age	height
1	35	71
2	36	65
3	37	60
4	38	62

```r
# some options...
people[1, 1]
people[1, 'age']
people$age[1]
people[1, people$height > 66]
people$age[people$height > 66]
```

Module 9 exercise-2

Data Analysis of the Day



https://nyti.ms/2jyS0aC

Loading Data

R Practice Data

R comes with a number of built-in data sets that can be used for practice, experimentation, and testing.

```
# view a list of included data sets
data()  # will open in new window

# Load e.g., the "Seatbelts" data set into memory
data("Seatbelts")  # quotes optional, but use them!

# The loaded data set is now available as a variable
print(Seatbelts)


# You may need to convert the data set into a data frame
# from another data type
seatbelts <- data.frame(Seatbelts)
```

CSV Files

R can also load data from external files, such as **comma-separated value** files (**.csv**).

```
Ada, 58, 115
Bob, 59, 117
Chris, 60, 120
Diya, 61, 123
Emma, 62, 126
```

record or observation

feature

Use **read.csv()** to read in a file at a given location (path)

```r
# Read data from the file `data/my_file.csv`
# into a data frame `my.data`
my.data <- read.csv('data/my_file.csv', stringsAsFactors=FALSE)
```

relative path!

RECALL

Paths

`/absolute/path/to/file`

leading slash

How to get there *starting from the root*

`relative/path/to/file`

How to get there *starting from here*

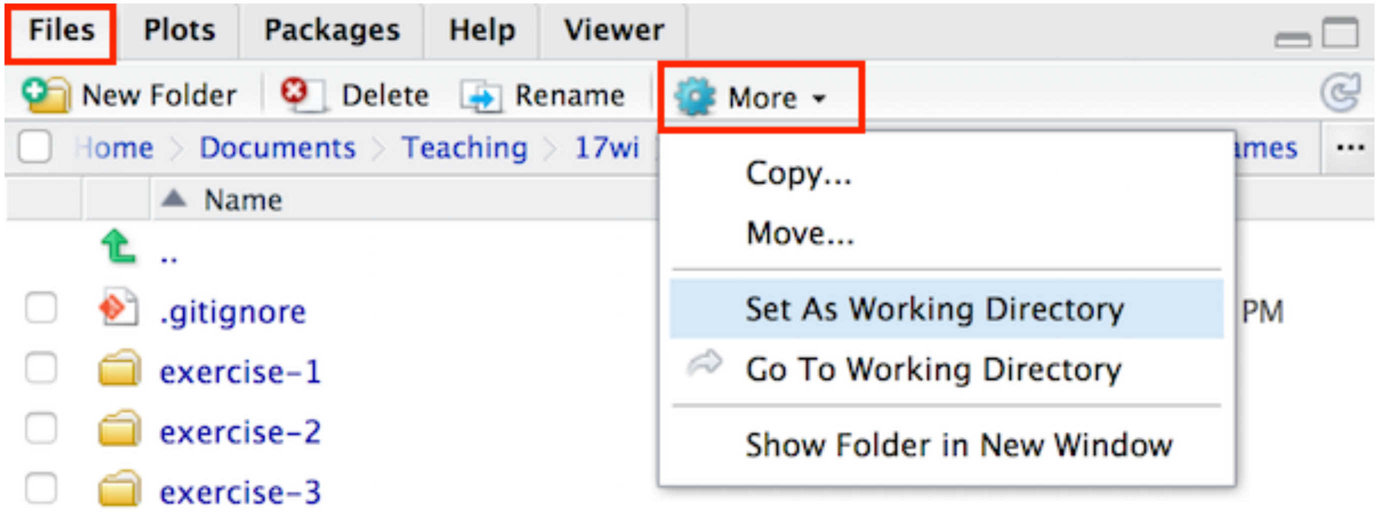
ALWAYS USE RELATIVE PATHS!

Working Directory

RStudio's **working directory** has nothing to do with which script you currently have open (if any!)

```
# get R's current working directory
getwd()  # like pwd, but not
```

Change the working directory through the executing environment: i.e., RStudio!



Module 9 exercise-3

Factors

Level of Measurement

A way of classifying the nature of data values. Applies to all data analysis, distinct from the R "data type".

Level	Example	Operations
★ **Nominal** unordered used for classification	*Fruits:* apples, bananas, oranges, etc.	**== !=** "same or different"
★ **Ordinal** ordered can comparison	*Grade of meat*: Grade A, Grade AA, Grade AAA, etc.	**== != < >** "bigger or smaller"
Interval ordered, no set "zero" can find difference	*Dates:* 05/15/2012, 04/17/2015, etc.	**== != < >** **+ –** "3 units bigger"
Ratio ordered, fixed "zero" can find magnitude	*Lengths:* 1 inch, 1.5 inches, 2 inches, etc.	**== != < >** **+ – * /** "twice as big"

Lots of Nominal Data

Imagine we're storing a **vector** of **nominal data** (e.g., to use in a **data frame**).

```
# Product categories (all media items)
prod.cat <- c("book","movie","book","music","music","movie")
```

4 chars

=

16 bytes

5 chars

=

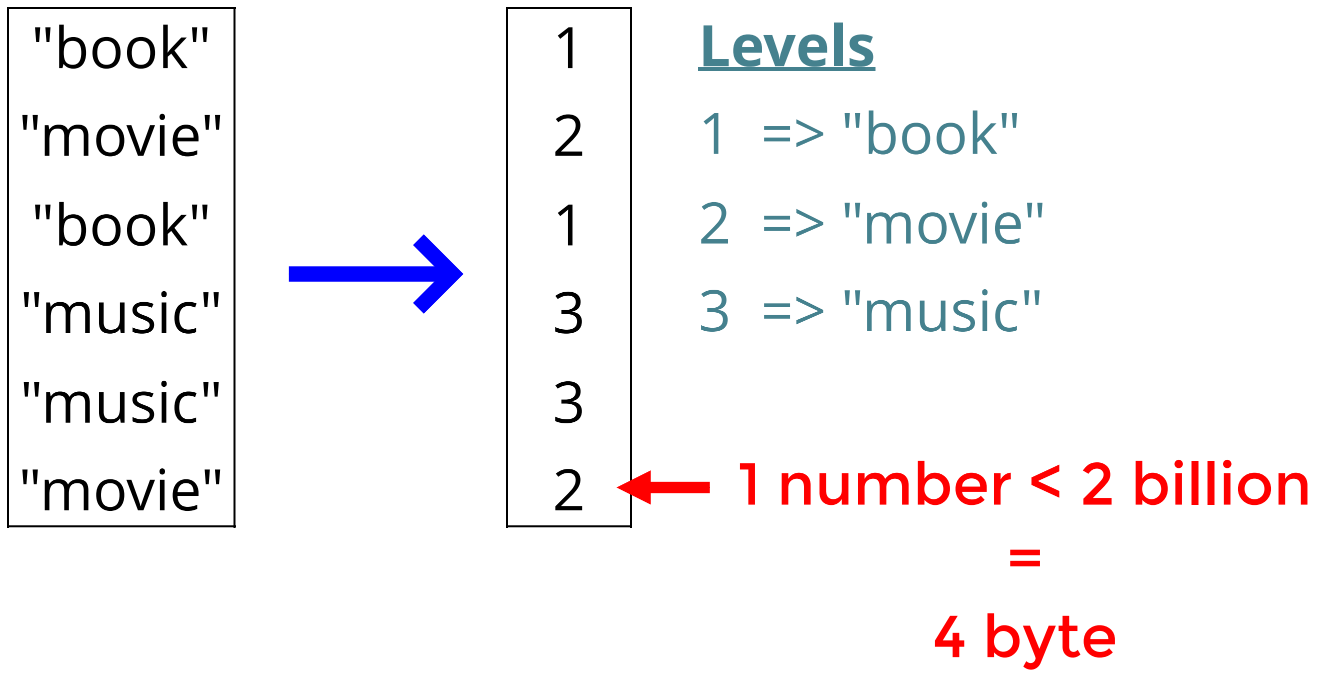
20 bytes

Total: 16+20+16+20+20+20 = 112 bytes of memory

What happens as this list gets big? Millions of items? (Amazon sells 480 million items, with categories like "Health, Household, and Baby Care")

Factors

Factor variables allow us to efficiently store data by storing **numbers** (called **levels**) instead of **strings**.



Factors

Factors are not vectors and do not support vector operations.

```
# create a factor of numbers (factors need not be strings)
num.factors <- as.factor(c(10,10,20,20,30,30,40,40))

# print the factor to see its levels
print(num.factors)

# multiply the numbers by 2
num.factors * 2  # Error: * not meaningful
                 # returns vector of NA instead

# changing entry to a level is fine
num.factors[1] <- 40

# change entry to a value that ISN'T a level fails
num.factors[1] <- 50  # Error: invalid factor level
                      # num.factors[1] is now NA
```

Factors and Data Frames

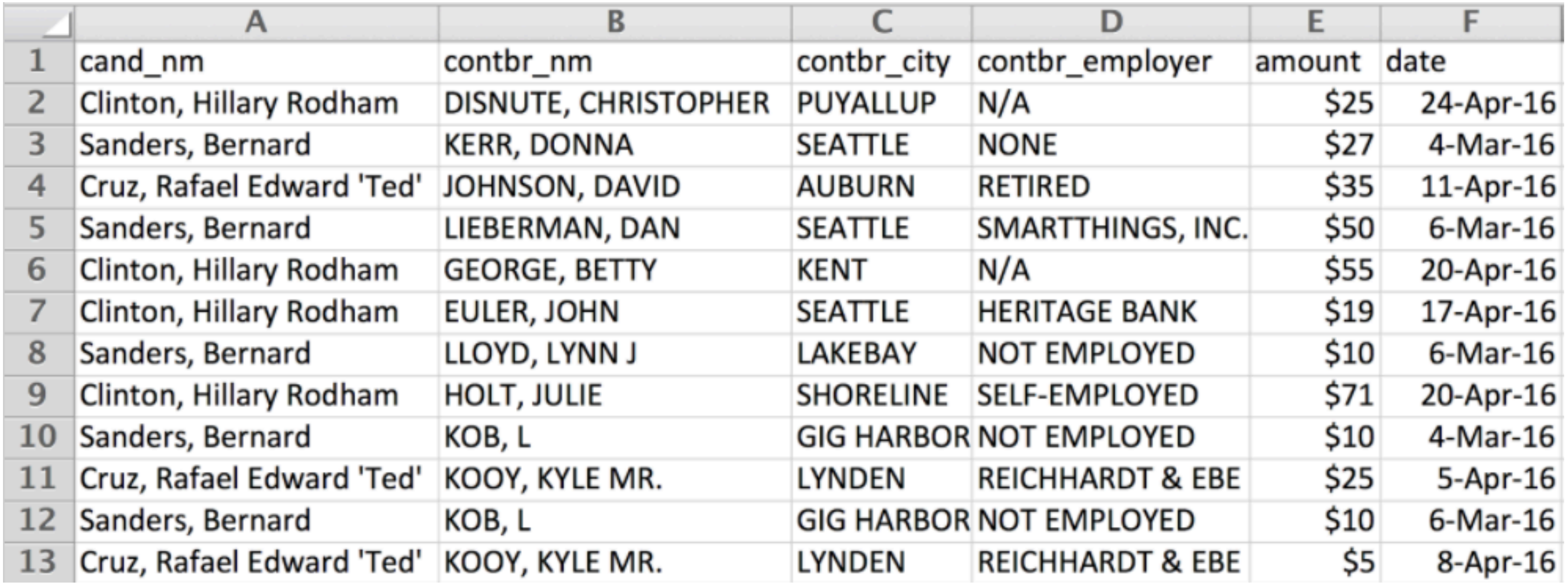
By default, R will store string columns as **factors** instead of **vectors** in a data frame. This disallows normal vector operations you may want to do.

```r
# Product categories (all media items)
prod.cat <- c("book","movie","book","music","music","movie")
# vector of costs (in dollars)
cost <- c(15.5, 17, 17, 14, 12, 23)

# data frame of products (with factors)
products <- data.frame(prod.cat, cost)

# the prod.factor column is a factor
is.factor(products$prod.cat)  # TRUE

# data frame of products (without factoring)
products <- data.frame(prod.cat, cost, stringsAsFactors=FALSE)

# the prod.cat column is NOT a factor
is.factor(products$prod.cat)  # FALSE
```

Tip: load data as a vector unless intentionally want a factor!

Working with Data Sets

An Example Data Set



	A	B	C	D	E	F
1	cand_nm	contbr_nm	contbr_city	contbr_employer	amount	date
2	Clinton, Hillary Rodham	DISNUTE, CHRISTOPHER	PUYALLUP	N/A	$25	24-Apr-16
3	Sanders, Bernard	KERR, DONNA	SEATTLE	NONE	$27	4-Mar-16
4	Cruz, Rafael Edward 'Ted'	JOHNSON, DAVID	AUBURN	RETIRED	$35	11-Apr-16
5	Sanders, Bernard	LIEBERMAN, DAN	SEATTLE	SMARTTHINGS, INC.	$50	6-Mar-16
6	Clinton, Hillary Rodham	GEORGE, BETTY	KENT	N/A	$55	20-Apr-16
7	Clinton, Hillary Rodham	EULER, JOHN	SEATTLE	HERITAGE BANK	$19	17-Apr-16
8	Sanders, Bernard	LLOYD, LYNN J	LAKEBAY	NOT EMPLOYED	$10	6-Mar-16
9	Clinton, Hillary Rodham	HOLT, JULIE	SHORELINE	SELF-EMPLOYED	$71	20-Apr-16
10	Sanders, Bernard	KOB, L	GIG HARBOR	NOT EMPLOYED	$10	4-Mar-16
11	Cruz, Rafael Edward 'Ted'	KOOY, KYLE MR.	LYNDEN	REICHHARDT & EBE	$25	5-Apr-16
12	Sanders, Bernard	KOB, L	GIG HARBOR	NOT EMPLOYED	$10	6-Mar-16
13	Cruz, Rafael Edward 'Ted'	KOOY, KYLE MR.	LYNDEN	REICHHARDT & EBE	$5	8-Apr-16

*What are three **questions** you could ask about this data set?*

(data source)

Sample Questions

- Who donated the most money?
- Which city did the largest donation come from?
- What was the average donation?

1. Select Columns

Who made the largest donation?

```
donations$amount
donations$contbr_nm
```

What was the average donation for Bernie?

```
$donation$amount
$donations$candidate == "Sanders"
```

Which cities donated in April?

```
donations$contbr_city
grepl("Apr",donations$date) == TRUE
```

2. Filter Rows

Who made the largest donation?

```
donations$contbr_nm[donations$amount == max(donations$amount)]
```

What was the average donation for Bernie?

```
mean(donations$amount[$donations$candidate == "Sanders"])
```

Which cities donated in April?

```
donations$contbr_city[grepl("Apr",donations$date) == TRUE]
```

Module 9 exercise-4

Module 9 exercise-5

Action Items!

- Be comfortable with **modules 0 - 9**
- Assignment 3 due ***Tuesday before class***

Tuesday: Data wrangling with DPLYR