Basic statistics and plots

Presenter: Steve Baskauf
steve.baskauf@vanderbilt.edu

DiSC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

CodeGraf landing page

- vanderbi.lt/codegraf

Missing data in R

DISC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

Missing data indicators

- R's built-in indicators for **missing data**:

  NA ("not available") means there is a value, but it's missing; length =1

  NULL means no value; length=0

```
vector_with_missing <- c(1, 2, NA, 3)
```

- NA will prevent some calculations from displaying a value. Example:

```
mean(vector_with_missing)
```

- This behavior can be overridden by removing NAs:

```
mean(vector_with_missing, na.rm = TRUE)
```

- "NA" can be used for missing data in tables instead of (ambiguous) blank cells

What happens when we read in a CSV?

- Files do not have a "magic" missing data indicator, only text.

- Missing data in a CSV may be:
  - an empty cell (a cell containing the empty string "")
  - a special series of characters ("-9999", "N/A", etc.)
  - coded data (particular numbers for particular reasons, e.g. age = "999" means age not known, age = "998" means participant refused to provide age)

CSV Import rules

- Rule for tibbles **`read_csv()`** is: **`"NA", "" -> NA`**

- Default rule for local file **`read.csv()`** is:
  - **`number columns: "NA", "" -> NA`**
  - **`character columns: "NA" -> NA`**

- **`read.csv()`** other values can be set using the argument:

```
na.strings = c("-9999", "NaN")
```

- **`read.csv()`** to suppress all conversion and read everything in as a character string (not factor) except **`"NA"`**, use the argument:

```
colClasses = "character"
```

Basic statistical quantities

DiSC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

Basic stats on a vector of continuous numbers

- Count of observations: **`length()`**
- Average (no NAs): **`mean()`**
- Standard deviation (no NAs): **`sd()`**
- Summary: **`summary()`**
- Quantiles (no NAs): **`quantile()`**

Problems with stats on Nashville schools data

- **`mean(schools_data$Asian)`** outputs NA when missing data present

- **`quantile(schools_data$Asian)`** will not do analysis with missing data

- Examine data! Should there be missing data ???
  - Should high schools report no first graders as missing data?
  - Should any school report no Asian students as missing data?

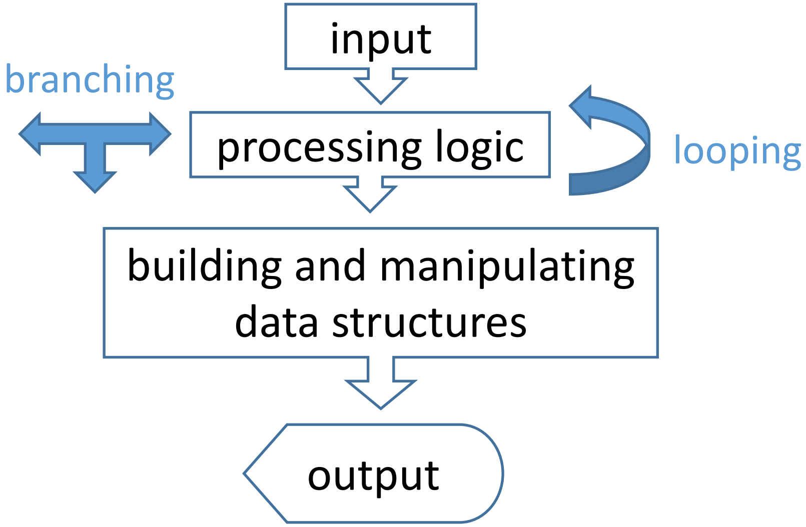
Procedural vs. vectorized paradigm

DISC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

Procedural vs. vectorized programming

beginner **Python** advanced



input

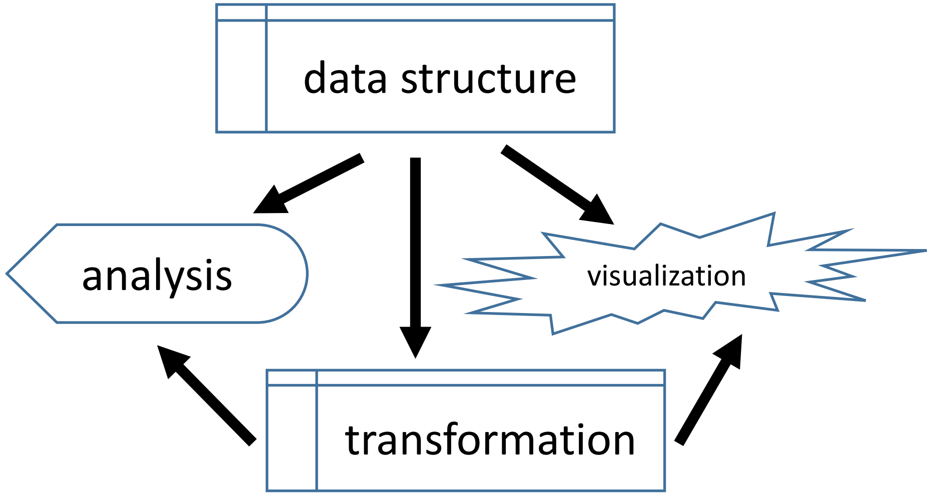
branching

processing logic

looping

building and manipulating data structures

output



data structure

analysis

visualization

transformation

advanced **R** beginner

Python script to read CSV and replace missing data with zeros

```
import csv
from statistics import mean

# read from a CSV file into a list of dictionaries
def read_dict(filename):
    with open(filename, 'r', newline='', encoding='utf-8') as file_object:
        dict_object = csv.DictReader(file_object)
        array = []
        for row in dict_object:
            array.append(row)
    return array

filename = 'Metro_Nashville_Schools.csv'
schools_data = read_dict(filename)
asian_no_missing = []
for school in schools_data:
    if school['Asian'] == '':
        asian_no_missing.append(0)
    else:
        asian_no_missing.append(int(school['Asian']))
mean(asian_no_missing)
```

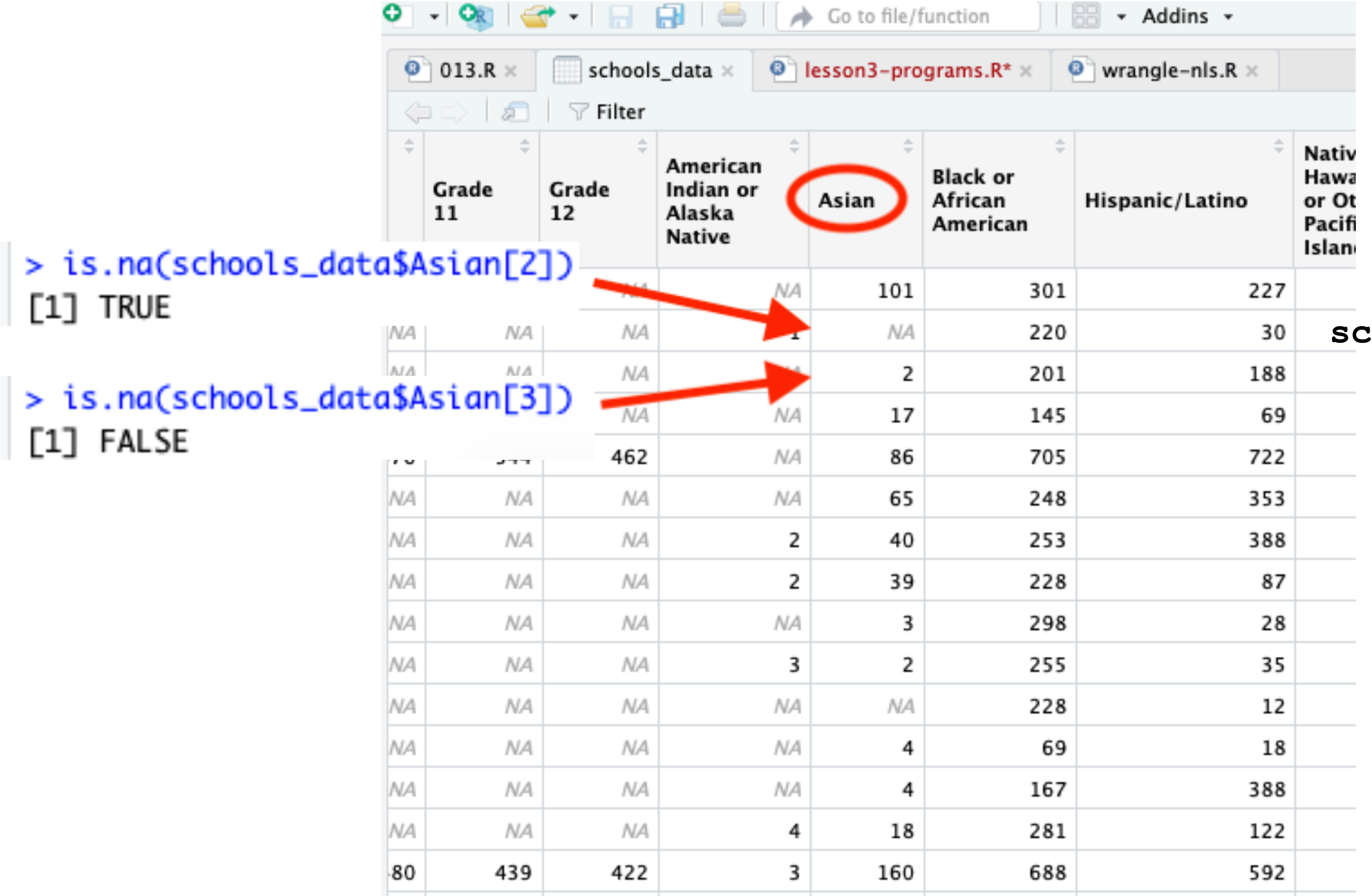
is.na() function

The function returns **TRUE** when the argument is NA and **FALSE** when it's anything else

```
> is.na(NA)
[1] TRUE
> is.na(3)
[1] FALSE
```

R script to read CSV and replace missing data with zeros

```
schools_data <- read_csv("Metro_Nashville_Schools.csv")
schools_data$Asian[is.na(schools_data$Asian)] <- 0
mean(schools_data$Asian)
```



	Grade 11	Grade 12	American Indian or Alaska Native	Asian	Black or African American	Hispanic/Latino	Nativ Hawa or Ot Pacifi Islan
		NA	NA	101	301	227	
NA	NA	NA	1	NA	220	30	
NA	NA	NA	NA	2	201	188	
		NA	NA	17	145	69	
70	544	462	NA	86	705	722	
NA	NA	NA	NA	65	248	353	
NA	NA	NA	2	40	253	388	
NA	NA	NA	2	39	228	87	
NA	NA	NA	NA	3	298	28	
NA	NA	NA	3	2	255	35	
NA	NA	NA	NA	NA	228	12	
NA	NA	NA	NA	4	69	18	
NA	NA	NA	NA	4	167	388	
NA	NA	NA	4	18	281	122	
80	439	422	3	160	688	592	

```
> is.na(schools_data$Asian[2])
[1] TRUE
```

```
> is.na(schools_data$Asian[3])
[1] FALSE
```

Vectorized operation:

```
schools_data$Asian[c(FALSE, TRUE, FALSE,…)] <- 0
```

Set to zero every item in the vector (i.e. column) where the condition has a value of **TRUE**

Basic plots

DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

"Built-in" plots vs. ggplot

- **Built-in R plots** are very easy to use but are limited in one's ability to customize them.

- The ggplot2 library, part of the tidyverse package, embodies "a deep philosophy of visualization". The **`ggplot()` function** produces highly customizable plots but has a much greater learning curve.

- ggplot will be covered in a later series of lessons.

`hist()` function

- **`hist()`** generates a histogram showing the distribution of data in a vector.

- The plot appears in RStudio's lower right pane under the **plots** tab.

`plot(y ~ x)`

- The **`plot()`** function is a simple way to generate a two-dimensional plot.

- The dependent (**`y`**) variable is listed before the tilde
- The independent (**`x`**) variable is listed after the tilde

- If **`x`** is a:
  - **discontinuous** factor (i.e. categories), **`plot()`** generates a box-and-whisker plot.
  - **continuous** variable (i.e. numbers), **`plot()`** generates an x-y scatter plot.

- In either case, **`y`** must be continuous.

generate a linear model with **`lm()`**

- The **`lm(y ~ x)`** function is used to generate a variety of **linear models** depending on the values of **`y`** and **`x`**. Linear models analyze the relationship between two variables.

- When **`x`** and **`y`** are both continuous, **`lm()`** performs a **linear regression**.
  - the model provides the slope and intercept
  - **`abline(model)`** inserts a **trendline** on a scatterplot.
  - **`summary(model)`** provides the results of the linear regression **statistical test**.