Lecture 5

The tidyverse!

agenda

Announcements	HW 1 due tonight, how ungrading will work
Slow reveal graphs	Whole class activity (linked on Canvas after class)
Mini-lecture	magrittr, dplyr, ggplot2
R activity	Practice what we just learned
Wrap-up	Reminders

EMRN Rubric

http://eric.ed.gov/?id=EJ717675

For the first two Canvas assignments, Dr. R will decide which category you fall into using the EMRN Rubric. After that, you will assess your own hw and decide the category.

Does the work demonstrate thorough understanding of the concepts?

Yes

Is the work complete and well-communicated?

- **Yes** → **E**
- **No** → **M**

No

Is there evidence of partial understanding?

- **Yes** → **R**
- **No** → **N**

E

Excellent/Exemplary
The work meets or exceeds the expectations of the assignment. Communication is clear and complete. Mastery of the concepts is evident. There are no non-trivial errors. This work could be used as a classroom example.

M

Meets Expectations
Understanding of the concepts is evident through correct work and clear, audience-appropriate explanations. Some revision or expansion is needed, but no significant gaps/errors are present. No additional instruction on the concepts is needed.

R

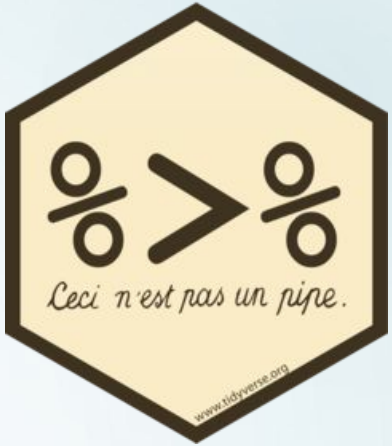
Revision Requested
Partial understanding of the concepts is evident, but there are significant gaps that remain. Needs further work, more review, and/or improved explanations.

N

Not Assessable
Not enough information is present in the work to determine if there is an understanding of the concepts. The work is fragmentary or contains significant omissions.

Slow reveal graph class activity

magrittr



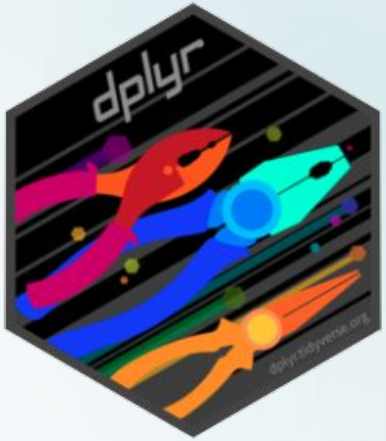
R for DS Chapter

“ the magrittr package offers a set of operators which make your code more readable…”

Namely, the pipe

%>%

dplyr



"dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges"

learn more about dplyr by running the code:

`vignette("dplyr")`

the main dplyr crew

01

mutate()

adds/changes new variables

02

select()

picks variables (columns) based on their names

03

filter()

picks cases (rows) based on their values

04

summarise()

reduces multiple values down to a single summary

05

arrange()

changes the ordering of the rows

06

group_by()

works with all the previous functions to perform operations “by group”

dplyr

cheat sheet

link

mutate()

```
data %>%
   mutate(newcol1 = function1(existing_col1),
          newcol2 = function2(existing_col2),
          existing_col3 = function3(existing_col3
          ...)
```

summarise()

```
data %>%
    summarise(name1 = function(col),
              name2 = function2(col),
          ...)
```

arrange()

```
# arrange tibble by col1 and col2 ascending
data %>%
    arrange(col1, col2)

# arrange tibble by col in descending order
data %>%
    arrange(desc(col))
```

group_by()

```
data %>%
   group_by(col1, col2,...) %>%
            ...
```

- summarise()
- arrange()
- slice()
- ...

other useful functions: slice()

```
data %>%
    slice(row_number) # any row(s) based on index

data %>%
    slice(n()) # last row

data %>%
    slice_min(col, n = __) # min or max

data %>%
    slice_sample(n = __) # random sample of n = __ rows
```

To R!

redownload intro_tidyverse.Rmd
(new version since Tuesday)

ggplot2

"ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics. You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details."

ggplot2

cheat sheet

link

how ggplot2 works

ggplot2 contains a wealth of functions made for high-quality data visualization. It “works” by building data graphics ***incrementally***

Some key terms:

- **aesthetics (aes)**
- **layers**
- **geometries (geoms)**
- scales
- guides
- facets

(a few) geoms to know

01

geom_point()

scatterplot

02

geom_histogram()

histogram

03

geom_col()

barplot

04

geom_boxplot()

boxplot

05

geom_density()

density plot

06

geom_line()

useful for time series or longitudinal plots

ggplot()

```
data %>%
   ggplot() # creates a blank plot
OR
ggplot(data)
```

- aes(): to assign aesthetics (variables) globally
- +: after creating the blank plot space you need to *add* layers using the plus sign

geom_point()

```
data %>%
    ggplot() +
    geom_point(aes(x = x, y = y))
```

- x = x variable
- y = y variable

geom_histogram()

```
data %>%
    ggplot() +
    geom_histogram(aes(x = x),
                   binwidth = number, # size of interval
                   bins = number) # number of bins
```

- x = numeric variable

geom_col()

```
data %>%
    ggplot() +
    geom_col(aes(y = y))
```

- y = height of bar

geom_boxplot()

```
data %>%
    ggplot() +
    geom_boxplot(aes(x = x, y = y))
```

- Either x or y should be numeric and the other variable is usually a factor

other arguments

col/color/colour

color (of point or border)

shape

shape of point

fill

fill color

alpha

transparency

size

size of point

linetype

type of line (e.g. dotted, dashed, solid)

adding labels

I usually use `labs()` to specify all labels

```
data %>%
    ggplot() +
    geom_point() +
    labs(title = “Title”,
         x = “x-label”,
         y = “y-label”,
         col = “color-label”, ...)
```

To R!

summarizing_visualizing.Rmd