HUDK 4050: CORE METHODS IN EDM

9/19/19 1:24 PM

Today

- CAs
- Checkin Test
- Vectr Consent Form
- Data wrangling

CAs

- Aidi Bian (ab4499@tc.columbia.edu)
- Anna Lizard (al3868@tc.columbia.edu)

CAs

- Open RStudio
- Create a vector with Aidi & Anna’s name
- Randomly draw from that vector one name
- That is the person you should email if you have a question

What should I be able to do at this point in the course?

- Open RStudio
- Connect to Github
- Fork/Clone/Commit/Push/ Pull Request
- Start Swirl

- If you can’t do these thing you should talk to Prof, CAs, the internet or a classmate to figure it out

Analytics Discretion

- Can we find out if you have done these things?
- Should we?

Data Wrangling

Data Wrangling
with dplyr and tidyr
Cheat Sheet

RStudio

Syntax - Helpful conventions for wrangling

dplyr::tbl_df(iris)

Converts data to tbl class. tbl's are easier to examine than data frames. R displays only the data that fits onscreen:

```
Source: local data frame [150 x 5]

   Sepal.Length Sepal.Width Petal.Length
1           5.1         3.5          1.4
2           4.9         3.0          1.4
3           4.7         3.2          1.3
4           4.6         3.1          1.5
5           5.0         3.6          1.4
..          ...         ...          ...
Variables not shown: Petal.Width (dbl),
  Species (fctr)
```

dplyr::glimpse(iris)

Information dense summary of tbl data.

utils::View(iris)

View data set in spreadsheet-like display (note capital V).

iris

Filter

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

dplyr::%>%

Passes object on left hand side as first argument (or . argument) of function on righthand side.

`x %>% f(y)` *is the same as* `f(x, y)`

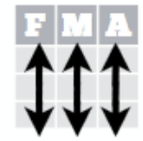
`y %>% f(x, ., z)` *is the same as* `f(x, y, z )`

"Piping" with %>% makes code more readable, e.g.

```
iris %>%
  group_by(Species) %>%
  summarise(avg = mean(Sepal.Width)) %>%
  arrange(avg)
```

Tidy Data - A foundation for wrangling in R

In a tidy data set:

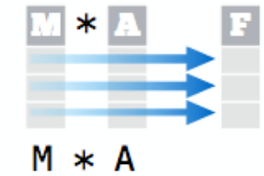


Each **variable** is saved in its own **column**

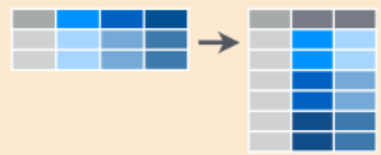
&

Each **observation** is saved in its own **row**

Tidy data complements R's **vectorized operations**. R will automatically preserve observations as you manipulate variables. No other format works as intuitively with R.



Reshaping Data - Change the layout of a data set



tidyr::gather(cases, "year", "n", 2:4)

Gather columns into rows.

tidyr::spread(pollution, size, amount)

Spread rows into columns.

tidyr::separate(storms, date, c("y", "m", "d"))

Separate one column into several.

tidyr::unite(data, col, ..., sep)

Unite several columns into one.

dplyr::data_frame(a = 1:3, b = 4:6)

Combine vectors into data frame (optimized).

dplyr::arrange(mtcars, mpg)

Order rows by values of a column (low to high).

dplyr::arrange(mtcars, desc(mpg))

Order rows by values of a column (high to low).

dplyr::rename(tb, y = year)

Rename the columns of a data frame.

Subset Observations (Rows)

dplyr::filter(iris, Sepal.Length > 7)

Extract rows that meet logical criteria.

dplyr::distinct(iris)

Remove duplicate rows.

dplyr::sample_frac(iris, 0.5, replace = TRUE)

Randomly select fraction of rows.

dplyr::sample_n(iris, 10, replace = TRUE)

Randomly select n rows.

dplyr::slice(iris, 10:15)

Select rows by position.

dplyr::top_n(storms, 2, date)

Select and order top n entries (by group if grouped data).

Logic in R - ?Comparison, ?base::Logic

<	Less than	!=	Not equal to
>	Greater than	%in%	Group membership
==	Equal to	is.na	Is NA
<=	Less than or equal to	!is.na	Is not NA
>=	Greater than or equal to	&, |, !, xor, any, all	Boolean operators

Subset Variables (Columns)

dplyr::select(iris, Sepal.Width, Petal.Length, Species)

Select columns by name or helper function.

Helper functions for select - ?select

select(iris, contains("."))

Select columns whose name contains a character string.

select(iris, ends_with("Length"))

Select columns whose name ends with a character string.

select(iris, everything())

Select every column.

select(iris, matches(".t."))

Select columns whose name matches a regular expression.

select(iris, num_range("x", 1:5))

Select columns named x1, x2, x3, x4, x5.

select(iris, one_of(c("Species", "Genus")))

Select columns whose names are in a group of names.

select(iris, starts_with("Sepal"))

Select columns whose name starts with a character string.

select(iris, Sepal.Length:Petal.Width)

Select all columns between Sepal.Length and Petal.Width (inclusive).

select(iris, -Species)

Select all columns except Species.

devtools::install_github("rstudio/EDAWR") for data sets

Learn more with browseVignettes(package = c("dplyr", "tidyr")) • dplyr 0.4.0• tidyr 0.2.0 • Updated: 1/15

Tidy Data

Data Frames, Vectors, Lists, Matrices, (Arrays)

numeric, character, factor

- character: "a", "swc"
- numeric: 2, 15.5
- integer: 2L (the L tells R to store this as an integer)
- logical: TRUE, FALSE
- complex: 1+4i

(complex numbers with real and imaginary parts)

Atomic vs generic*

All naked numbers are double-width floating-point atomic vectors of length one* (There 2 is not an integer)

TRUE/FALSE vs T/F

Why is tidy data?

- Difference between “clean” and “tidy”
- Data comes in a lot different structures, some which are difficult to analyze
- We want to make them manageable
- We want them to be “intuitive” to R (vectorized)
- BUT we want to keep a very precise record of how we did that

What is tidy data?

1. Observations are in rows
2. Variables are in columns
3. In a single data set

But…?

- What is a variable?
- What is an observation?
- What goes where in a data matrix?

Wide Format

- Repeated measures are in a single row

Student	Quiz 2-1-16	Quiz 2-10-16	Quiz 2-20-16
Francis	10	10	11
Alex	14	15	18
Kaji	11	17	14
Miriam	8	10	8

Long (Narrow) Format

- Each row is one time point per subject

Student	Quiz	Date
Francis	10	2-1-16
Francis	10	2-10-16
Francis	11	2-20-16
Alex	14	2-1-16

Generalize

Male	Female
4	10
7	10

How many variables are in the above matrix?

1. Male
2. Female
3. Count

Types of Messiness

- Column headers are values, not variable names
- Multiple variables are stored in one column
- Variables are stored in both rows and columns
- Multiple types of experimental unit stored in the same table
- One type of experimental unit stored in multiple tables

Tidy System

- There are many commands and several packages for doing this in R
- We are going to try to stick to two: tidyr & dplyr (we may end up using more)
- Reshape, Subset, Variable generation, Combine, Summarize

Subset

- Splitting data frames

Student	Quiz	Date
Francis	10	2-1-16
Francis	10	2-10-16
Francis	11	2-20-16
Alex	14	2-1-16

Student	Quiz	Date
Francis	10	2-1-16
Francis	10	2-10-16
Francis	11	2-20-16

Student	Quiz	Date
Alex	14	2-1-16

Variable Generation

- Create new variable from current variables

Student	Quiz 2-1-16	Quiz 2-10-16	Quiz 2-20-16
Francis	10	10	11
Alex	14	15	18
Kaji	11	17	14
Miriam	8	10	8

→

mean
10.3
15.7
14
8.7

Combine

- Merge and bind dataframes
- Mutate or Filter

Student	Quiz 2-1-16	Quiz 2-10-16
Francis	10	10
Alex	14	15
Kaji	11	17

+

Student	Quiz 2-1-16	Quiz 2-20-16
Francis	10	9
Suchi	14	5
Kaji	11	10

Summarize

- Collapse data into a limited number of values according to a function

Student	Quiz 2-1-16	Quiz 2-10-16
Francis	10	10
Alex	14	15
Kaji	11	17

→

Av(Score/ Quiz/ Student)
12.8

Data Wrangling
with dplyr and tidyr
Cheat Sheet

RStudio

Syntax - Helpful conventions for wrangling

dplyr::tbl_df(iris)

Converts data to tbl class. tbl's are easier to examine than data frames. R displays only the data that fits onscreen:

```
Source: local data frame [150 x 5]

   Sepal.Length Sepal.Width Petal.Length
1           5.1         3.5          1.4
2           4.9         3.0          1.4
3           4.7         3.2          1.3
4           4.6         3.1          1.5
5           5.0         3.6          1.4
..          ...         ...          ...
Variables not shown: Petal.Width (dbl),
  Species (fctr)
```

dplyr::glimpse(iris)

Information dense summary of tbl data.

utils::View(iris)

View data set in spreadsheet-like display (note capital V).

iris

Filter

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

dplyr::%>%

Passes object on left hand side as first argument (or . argument) of function on righthand side.

`x %>% f(y)` *is the same as* `f(x, y)`

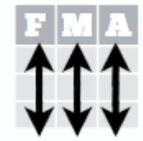
`y %>% f(x, ., z)` *is the same as* `f(x, y, z )`

"Piping" with %>% makes code more readable, e.g.

```
iris %>%
  group_by(Species) %>%
  summarise(avg = mean(Sepal.Width)) %>%
  arrange(avg)
```

Tidy Data - A foundation for wrangling in R

In a tidy data set:

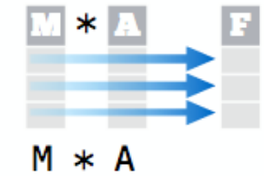


Each **variable** is saved in its own **column**

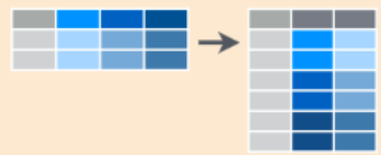
&

Each **observation** is saved in its own **row**

Tidy data complements R's **vectorized operations**. R will automatically preserve observations as you manipulate variables. No other format works as intuitively with R.



Reshaping Data - Change the layout of a data set



tidyr::gather(cases, "year", "n", 2:4)

Gather columns into rows.

tidyr::spread(pollution, size, amount)

Spread rows into columns.

tidyr::separate(storms, date, c("y", "m", "d"))

Separate one column into several.

tidyr::unite(data, col, ..., sep)

Unite several columns into one.

dplyr::data_frame(a = 1:3, b = 4:6)

Combine vectors into data frame (optimized).

dplyr::arrange(mtcars, mpg)

Order rows by values of a column (low to high).

dplyr::arrange(mtcars, desc(mpg))

Order rows by values of a column (high to low).

dplyr::rename(tb, y = year)

Rename the columns of a data frame.

Subset Observations (Rows)

dplyr::filter(iris, Sepal.Length > 7)

Extract rows that meet logical criteria.

dplyr::distinct(iris)

Remove duplicate rows.

dplyr::sample_frac(iris, 0.5, replace = TRUE)

Randomly select fraction of rows.

dplyr::sample_n(iris, 10, replace = TRUE)

Randomly select n rows.

dplyr::slice(iris, 10:15)

Select rows by position.

dplyr::top_n(storms, 2, date)

Select and order top n entries (by group if grouped data).

Logic in R - ?Comparison, ?base::Logic

<	Less than	!=	Not equal to
>	Greater than	%in%	Group membership
==	Equal to	is.na	Is NA
<=	Less than or equal to	!is.na	Is not NA
>=	Greater than or equal to	&, |, !, xor, any, all	Boolean operators

Subset Variables (Columns)

dplyr::select(iris, Sepal.Width, Petal.Length, Species)

Select columns by name or helper function.

Helper functions for select - ?select

select(iris, contains("."))

Select columns whose name contains a character string.

select(iris, ends_with("Length"))

Select columns whose name ends with a character string.

select(iris, everything())

Select every column.

select(iris, matches(".t."))

Select columns whose name matches a regular expression.

select(iris, num_range("x", 1:5))

Select columns named x1, x2, x3, x4, x5.

select(iris, one_of(c("Species", "Genus")))

Select columns whose names are in a group of names.

select(iris, starts_with("Sepal"))

Select columns whose name starts with a character string.

select(iris, Sepal.Length:Petal.Width)

Select all columns between Sepal.Length and Petal.Width (inclusive).

select(iris, -Species)

Select all columns except Species.

devtools::install_github("rstudio/EDAWR") for data sets

Learn more with browseVignettes(package = c("dplyr", "tidyr")) • dplyr 0.4.0• tidyr 0.2.0 • Updated: 1/15

Reshape

- Similar to generating pivot tables
- Long format ⟷ Wide format

Student	Quiz 1	Quiz 2	Quiz 3
Francis	10	10	11
Alex	14	15	18
Kaji	11	17	14
Miriam	8	10	8

Spread ←

→ **Gather**

Student	Quiz	Date
Francis	10	Quiz 1
Francis	10	Quiz 2
Francis	11	Quiz 3
Alex	14	Quiz 1

tidyr::spread()

Specify:

- dataframe
- key: the column that the reshape will be based on
- value: column whose values will populate the cells

tidyr::spread()

Key **Value**

Student	Quiz	Date
Francis	10	Quiz 1
Francis	10	Quiz 2
Francis	11	Quiz 3
Alex	14	Quiz 1

→

Student	Quiz 1	Quiz 2	Quiz 3
Francis	10	10	11
Alex	14	15	18
Kaji	11	17	14
Miriam	8	10	8

tidyr::gather()

Specify:

- dataframe
- key: the new column that the reshape will be based on
- value: new column to store the values that will be generated for new data frame
- gather_cols: the columns that are reshaped to accommodate the new structure
    - ✳ Can also identify key using “-“ and all other columns will be gathered

tidyr::gather()

Student	Quiz 1	Quiz 2	Quiz 3
Francis	10	10	11
Alex	14	15	18
Kaji	11	17	14
Miriam	8	10	8

Student	Quiz	Date
Francis	10	Quiz 1
Francis	10	Quiz 2
Francis	11	Quiz 3
Alex	14	Quiz 1

key

Student
Francis
Francis
Alex

Quiz 1	Quiz 2	Quiz 3	10	11	14	15	18
10	10	11	Quiz 1	Quiz 3	NA	NA	NA
10	10	11	Quiz2	NA	NA	NA	NA
14	15	18	NA	NA	Quiz 1	Quiz 2	Quiz 3

Class Activity 1

core-methods-in-edm/class-activity-1