

HUDK 4050: CORE METHODS IN EDM

Today

- CAs
- Checkin Test
- Data wrangling

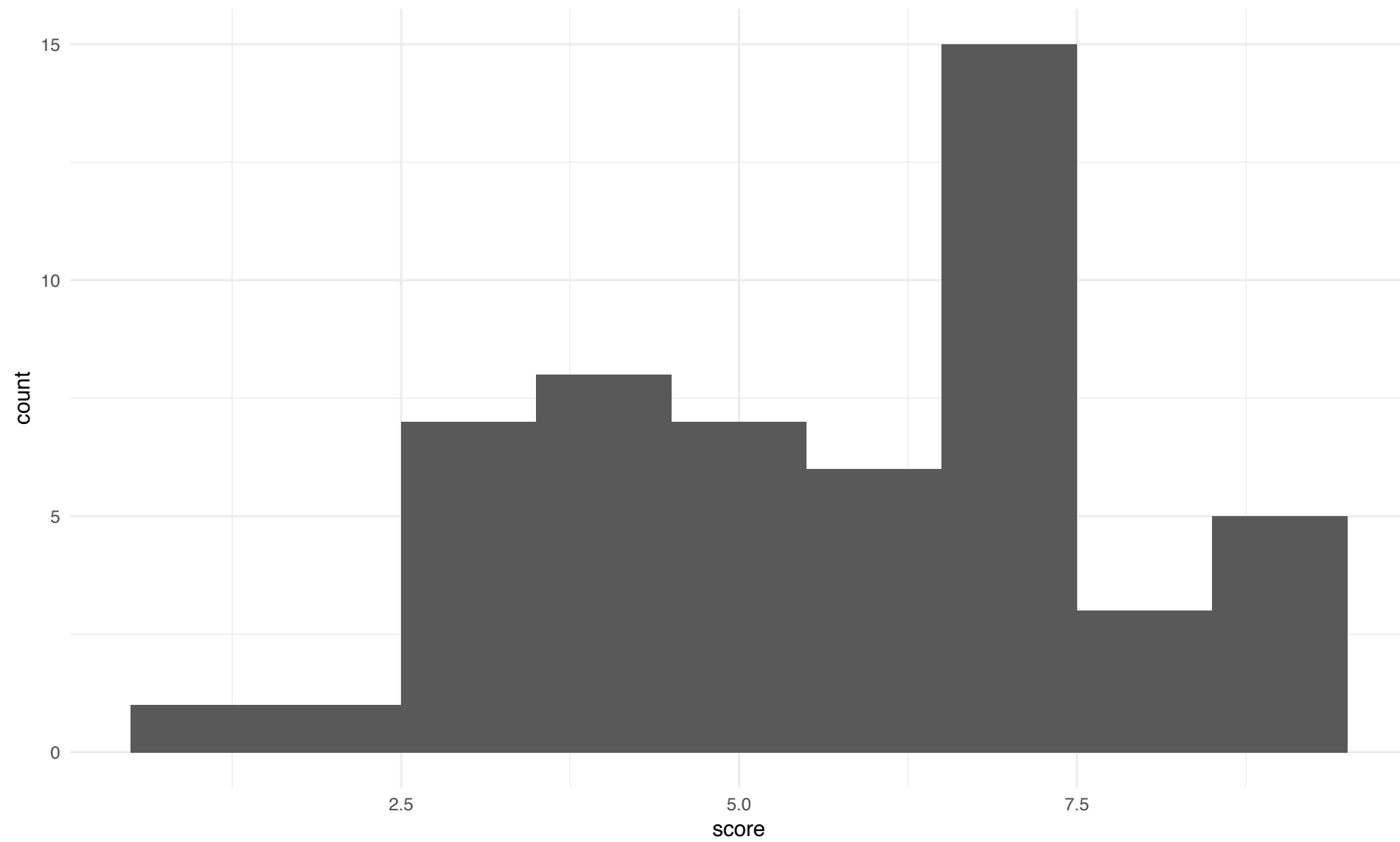
CAs

- Karen Zhou (zz2404@tc.columbia.edu)
- David Cody (dmc2233@tc.columbia.edu)

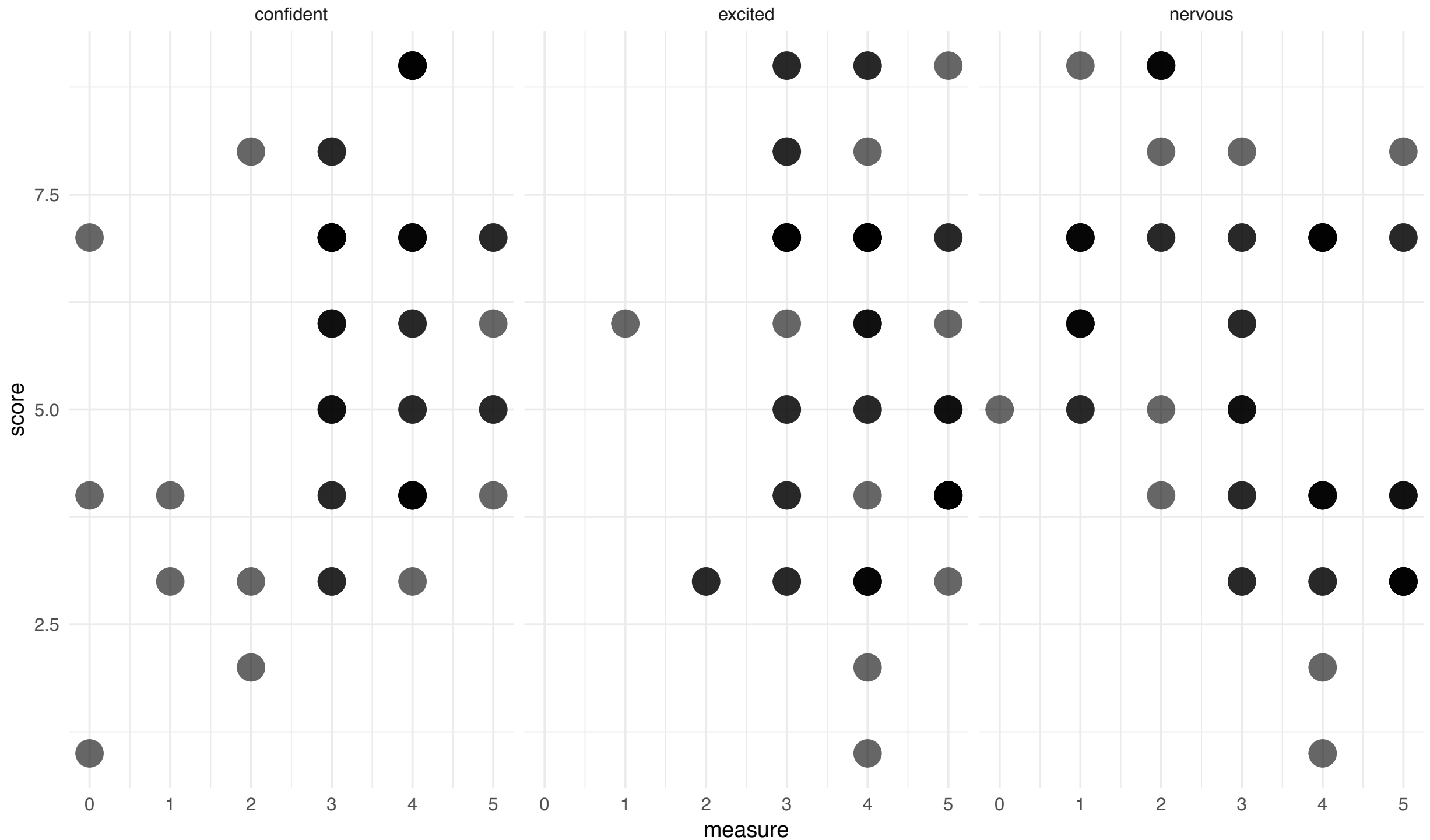
CAs

- Open RStudio
- Create a vector with Karen & Dave's name
- Randomly draw from that vector one name
- That is the person you should email

Checkin Test



Checkin Test



Checkin Test

Q1. What is the R code to print the column names of a dataframe?

`colnames` or `names`

- A lot of people using the `print()` function

72% correct

Checkin Test

Q2. What is the R code to print the first 10 rows if a dataframe?

```
head(DF, 10)
```

42% correct

Checkin Test

Q3. What is the R code to calculate the mean of the "score" column in the "MATH.TEST" dataframe?

```
mean ( MATH . TEST $ score )
```

- \$ is how we look inside an object

42% Correct

Checkin Test

Q4. What is the R code to convert the variable "count" from a character type variable to a numeric type variable?

```
as.numeric(DF$count)
```

55% Correct

Checkin Test

Q5. What is the R code to upload the .csv file "test.scores.csv" into R as a dataframe?

```
TS <- read.csv("test.scores.csv", header =  
TRUE)
```

32% Correct

Data Wrangling

Data Wrangling with dplyr and tidyr

Cheat Sheet



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:

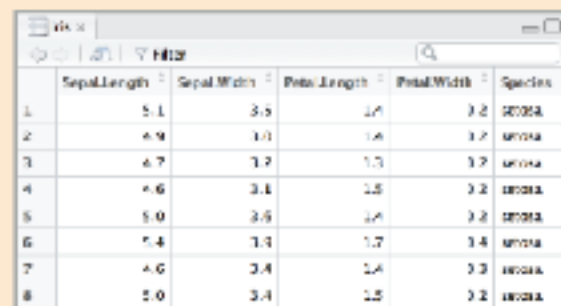
```
Source: local data frame [150 x 5]
   Sepal.Length Sepal.Width Petal.Length
1             5.1         3.5          1.4
2             4.9         3.0          1.4
3             4.7         3.2          1.3
4             4.6         3.1          1.5
5             5.0         3.6          1.4
..          ...
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).



	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	3.4	1.2	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.

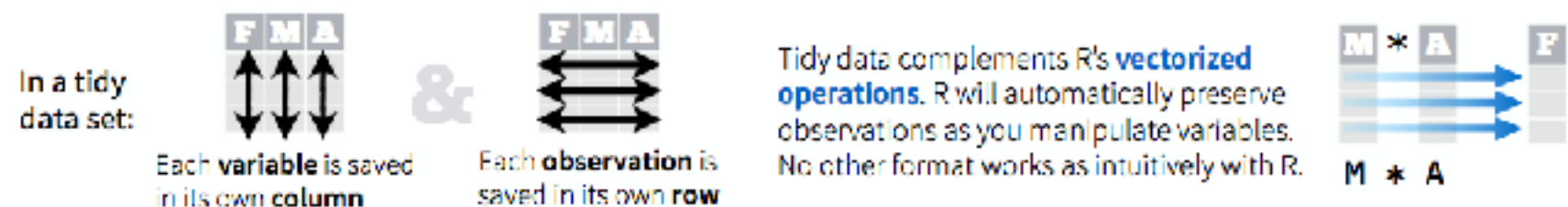
$x \%>\% f(y)$ is the same as $f(x, y)$

$y \%>\% f(x, ., z)$ is the same as $f(x, y, z)$

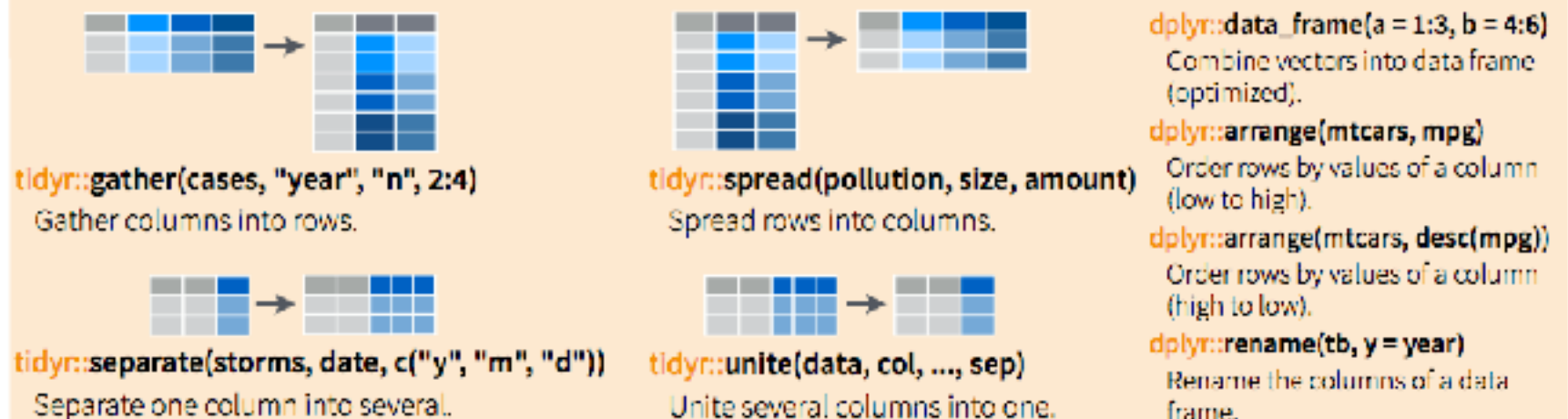
"Piping" with %>% makes code more readable, e.g.

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

Tidy Data - A foundation for wrangling in R



Reshaping Data - Change the layout of a data set



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).

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("x"))
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("x"))
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.

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

Tidy Data

(#notidynodessert)

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*

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

Reshape

- Similar to generating pivot tables
- Long format \longleftrightarrow 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

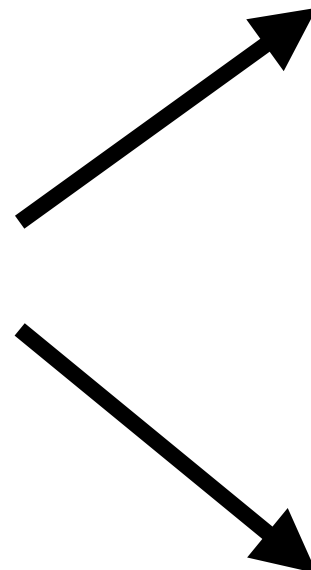


Student	Quiz	Date
Francis	10	2-1-16
Francis	10	2-10-16
Francis	11	2-20-16
Alex	14	2-1-16

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		mean
Francis	10	10	11	→	10.3
Alex	14	15	18		15.7
Kaji	11	17	14		14
Miriam	8	10	8		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



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:

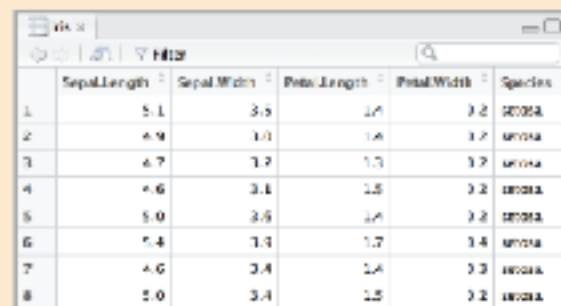
```
Source: local data frame [150 x 5]
   Sepal.Length Sepal.Width Petal.Length
1           5.1         3.5          1.4
2           4.9         3.0          1.4
3           4.7         3.2          1.3
4           4.6         3.1          1.5
5           5.0         3.6          1.4
..          ...          ...          ...
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).



	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	3.4	1.2	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.

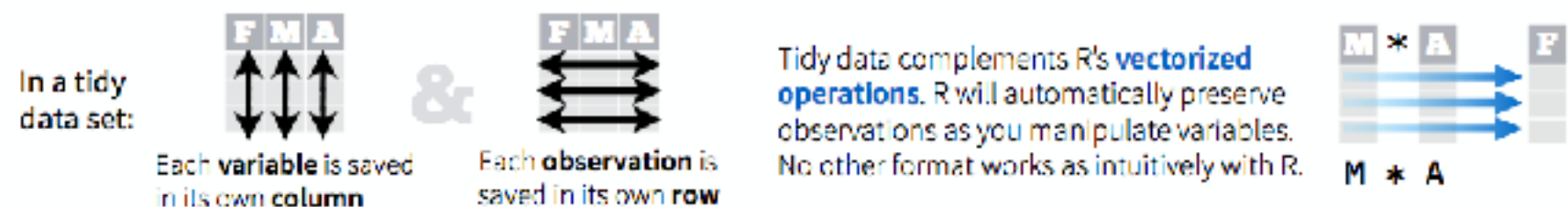
$x \%>\% f(y)$ is the same as $f(x, y)$

$y \%>\% f(x, ., z)$ is the same as $f(x, y, z)$

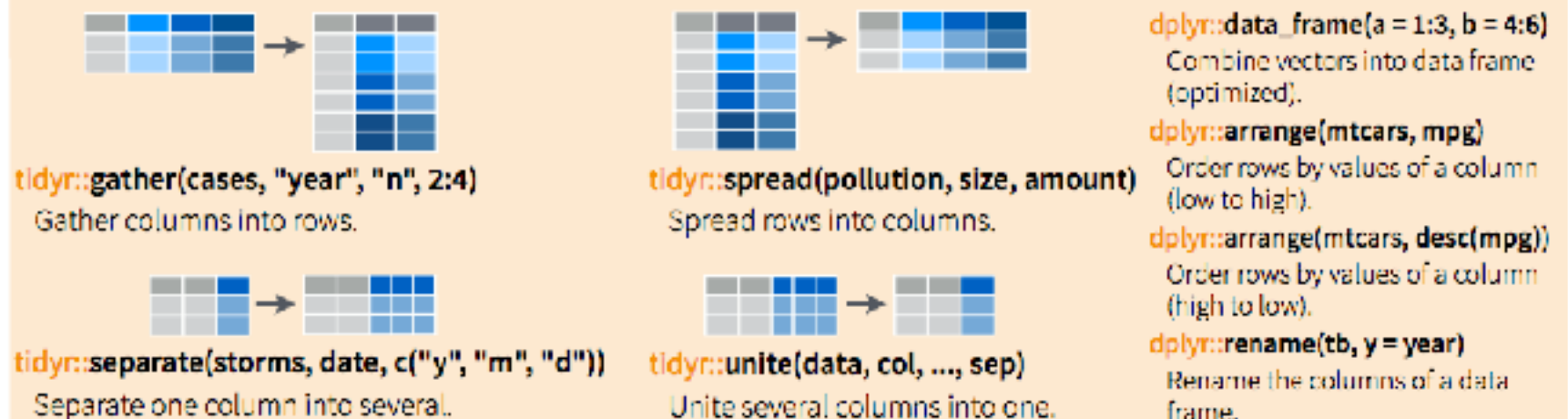
"Piping" with %>% makes code more readable, e.g.

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

Tidy Data - A foundation for wrangling in R



Reshaping Data - Change the layout of a data set



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).

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("x"))
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("x"))
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.

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

Reshape

- Similar to generating pivot tables
- Long format \longleftrightarrow 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::gather()

Specify:

- dataframe
- key: the variable that the reshape will be based on
- value: new variable names that will be generated for new data frame
- gather_cols: the variables 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

tidyr::spread()

Specify:

- dataframe
- key: the variable 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

Class Activity 1

[core-methods-in-edm/class-activity-1](#)