Linear Regression Models
P8111

Lecture 02

Jeff Goldsmith

January 21, 2016

THE DEPARTMENT OF BIOSTATISTICS
Columbia University MAILMAN SCHOOL OF PUBLIC HEALTH

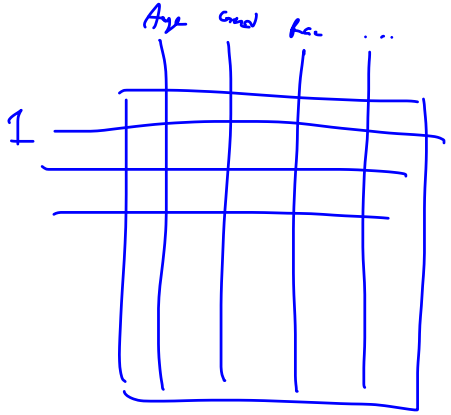
Today's Lecture

- `dplyr`

Data organization and manipulation

- You're going to spend a lot of time doing this
- Being better will make your life easier and happier

Data frames



- Most common way of storing dataset in R
- 2D array consisting of named vectors
- Can include variables of many types (numeric, logical, factor, string)
- Used consistently, data frames make analysis easier

Data frames

```
data = data.frame(
  seq = 1:10,
  let = letters[1:10],
  bool = 1:10 < 5
)
```

Data frames: exploration

```
> dim(data)
[1] 10  3
> head(data)
Source: local data frame [6 x 3]

     seq    let  bool
   (int) (fctr) (lgl)
1      1      a  TRUE
2      2      b  TRUE
3      3      c  TRUE
4      4      d  TRUE
5      5      e FALSE
6      6      f FALSE
```

Data frames: exploration

```
> summary(data)
      seq          let        bool
 Min.   : 1.00   a      :1   Mode :logical
 1st Qu.: 3.25   b      :1   FALSE:6
 Median : 5.50   c      :1   TRUE :4
 Mean   : 5.50   d      :1   NA's :0
 3rd Qu.: 7.75   e      :1
 Max.   :10.00   f      :1
                 (Other):4
```

Data frames: exploration

```
data$seq
summary(data$seq)
```

tbl_df: an upgrade to data frames

```
> data = tbl_df(data)
> data
Source: local data frame [10 x 3]

     seq    let  bool
   (int) (fctr) (lgl)
1      1      a  TRUE
2      2      b  TRUE
3      3      c  TRUE
4      4      d  TRUE
5      5      e FALSE
6      6      f FALSE
7      7      g FALSE
8      8      h FALSE
9      9      i FALSE
10    10      j FALSE
```

tbl_df: an upgrade to data frames

```
> glimpse(data)
Observations: 10
Variables: 3
$ seq  (int) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
$ let  (fctr) a, b, c, d, e, f, g, h, i, j
$ bool (lgl) TRUE, TRUE, TRUE, TRUE, FALSE, FALSE, FALSE
```

dplyr

- `dplyr` is a new(ish) package for the management and manipulation of data frames built by Hadley Wickham and Romain Francois
- The package contains several functions focused on the most common tasks – these functions each do one thing, and do it extremely well
- Functions are designed in a uniformly sensible way: the first argument is a dataframe, and the output is a dataframe

dplyr

- Think of `dplyr`'s functions as verbs: they're actions you want to take on the data
- Arguments to functions clarify the action to take
- Verbs include `filter()`, `arrange()`, `select()`, `rename()`, `mutate()`, `summarize()`, `sample_n()`
- You should absolutely become fluent in these actions

`dplyr` : Two Other Things

- Grouping (`group_by()`) can make some tasks infinitely easier
- The pipe operator (`%>%`) will change your life

Live coding

Some final thoughts

- You don't know how good you have it
- You can (and should) limit the amount of subsets you save to your workspace
- Many R functions (like `lm`) have `data` and `subset` options, allowing you to pass datasets and trim, or to have these as the last step in a pipe

Cheat Sheet

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:

F M A — Each **variable** is saved in its own **column**

&

F M A — 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.

M * A F

M * A

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

Today's big ideas

- Intro to `dplyr`
- Intro to coding

- Introduction to dplyr (on CRAN)
- Data Wrangling Cheat Sheet
- `swirl` (Getting and Cleaning Data)
- STAT 545 "Basic care and feeding... ", "dplyr: ..."
- Exploratory Data Analysis with R (Managing Data)
- R Programming for Data Science (Ch 13)