Loading and saving data
(a refresher)

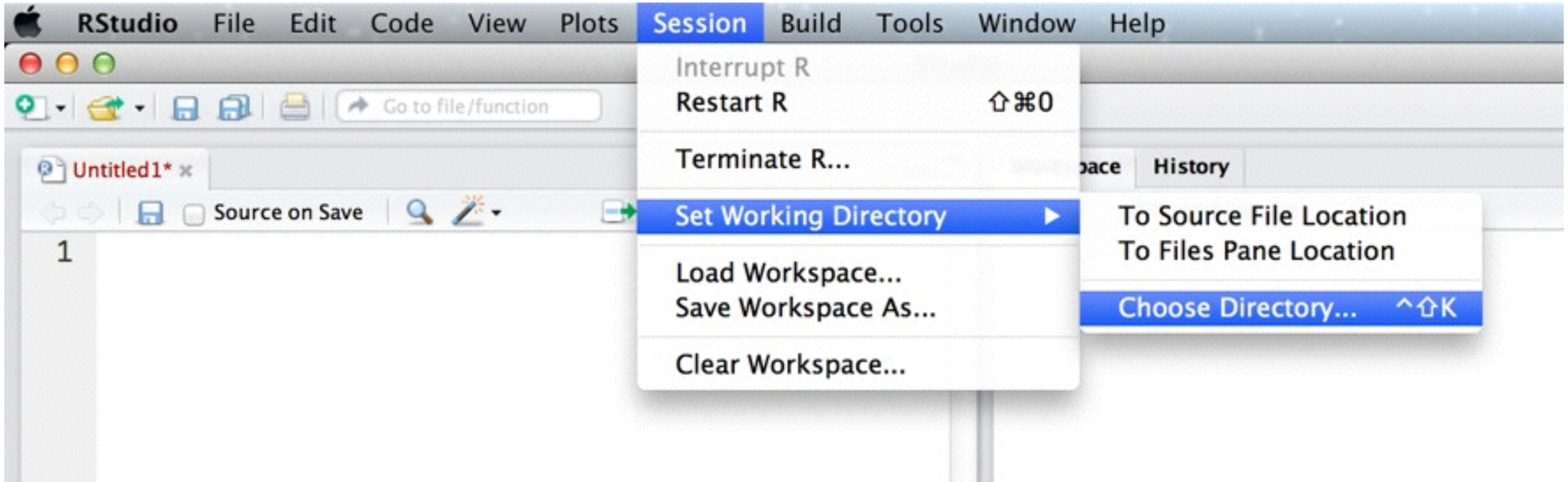
Working directory

When you start R, it associates itself with a folder (i.e, directory) on your computer.

- This folder is known as your "**working directory**"
- When you save files, R will save them here
- When you load files, R will look for them here
- To see which folder is your working directory, run `getwd()`

To change the working directory

In the toolbar, go to
Session>Set Working Directory>Choose Directory...



Loading data

Load csv files with **read.csv()**

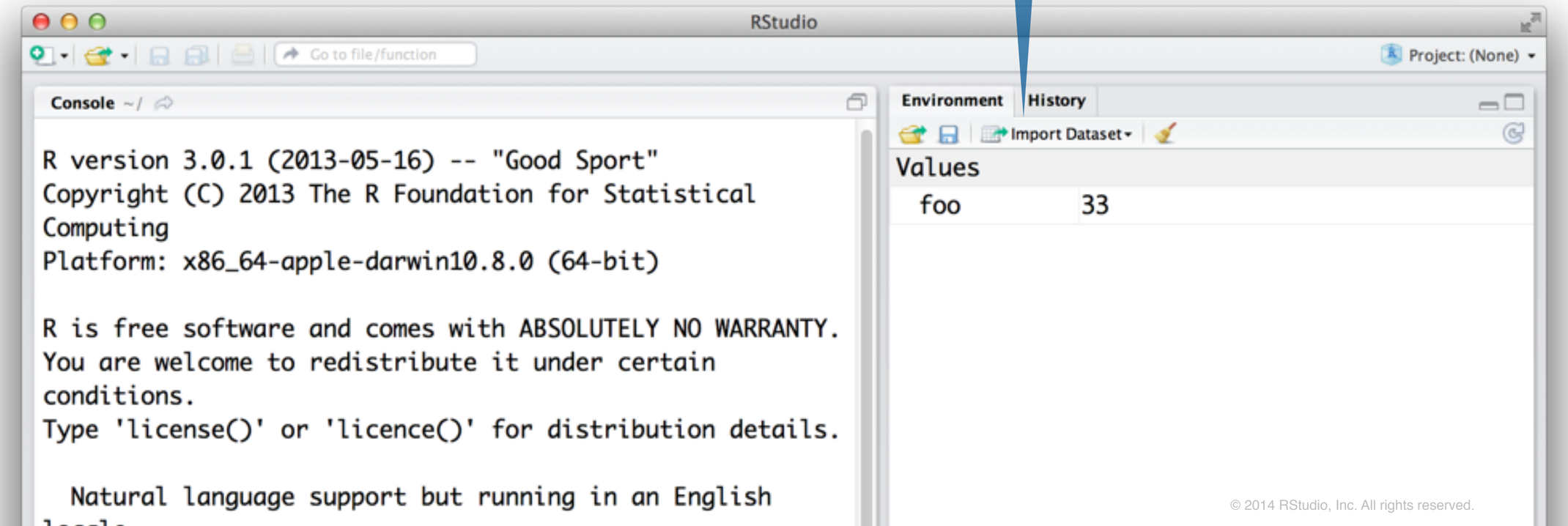
```
read.csv("data/my-data.csv", stringsAsFactors = FALSE)
```

A file path from your working directory to the csv file

To prevent headaches

Import Dataset

Click to select and read in a file



Saving data

Save data frames with csv files with **write.csv()**

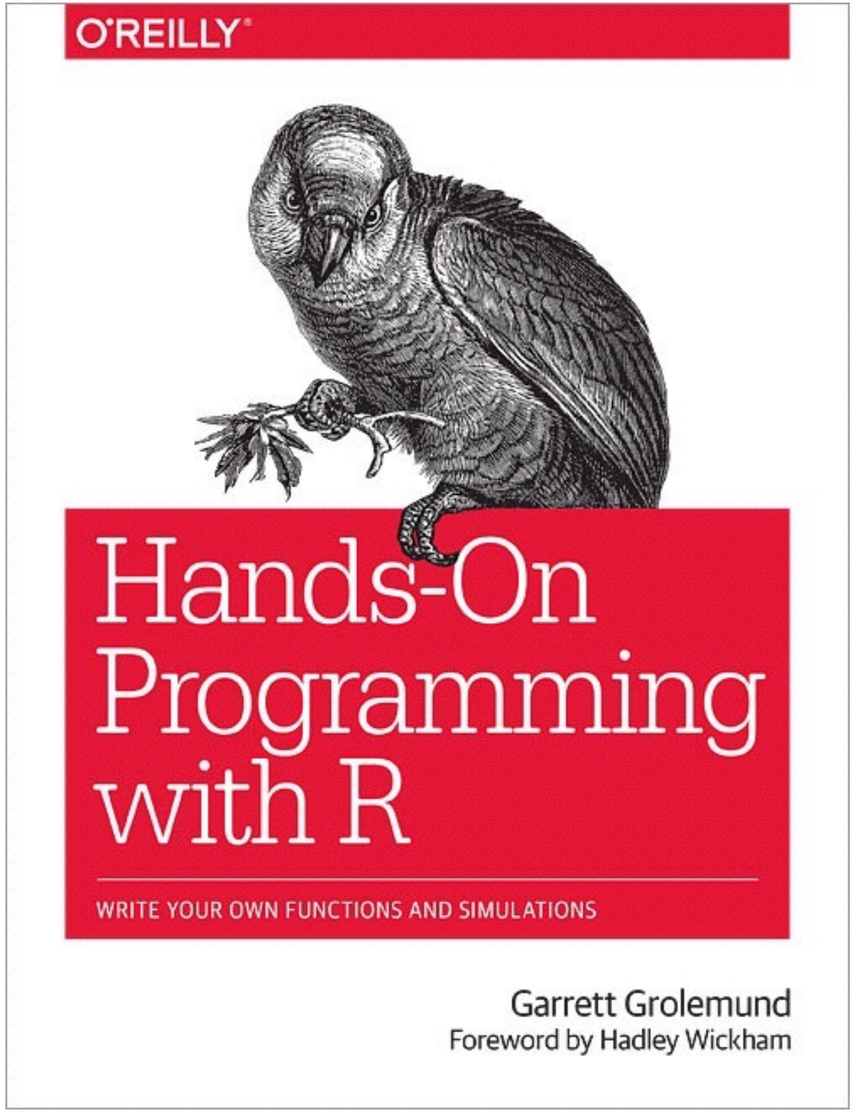
```
write.csv(df, file = "data/my-data.csv", row.names = FALSE)
```

Data frame to save

Path to file you wish to create

To prevent headaches

How to learn more

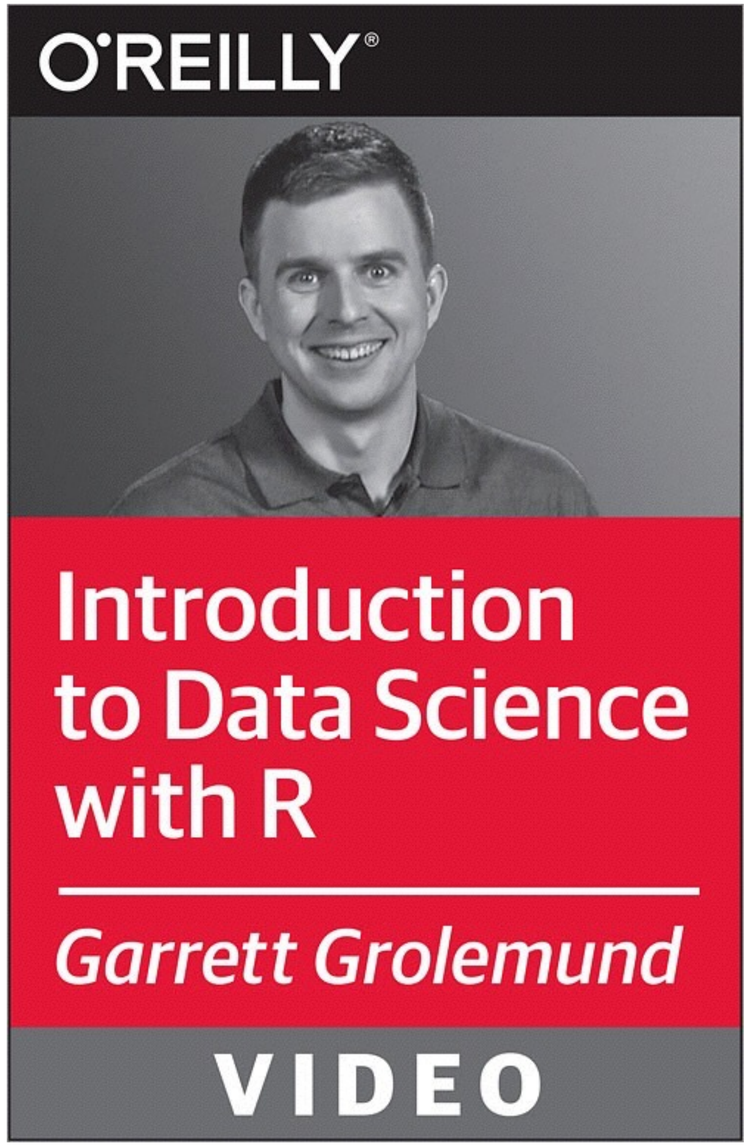


The R language

Three hands-on projects that teach the breadth of the R language.

- Storing, saving, and retrieving data.
- Writing programs and simulations.
- Vectorized programming.

bit.ly/hands-on-programming-with-R



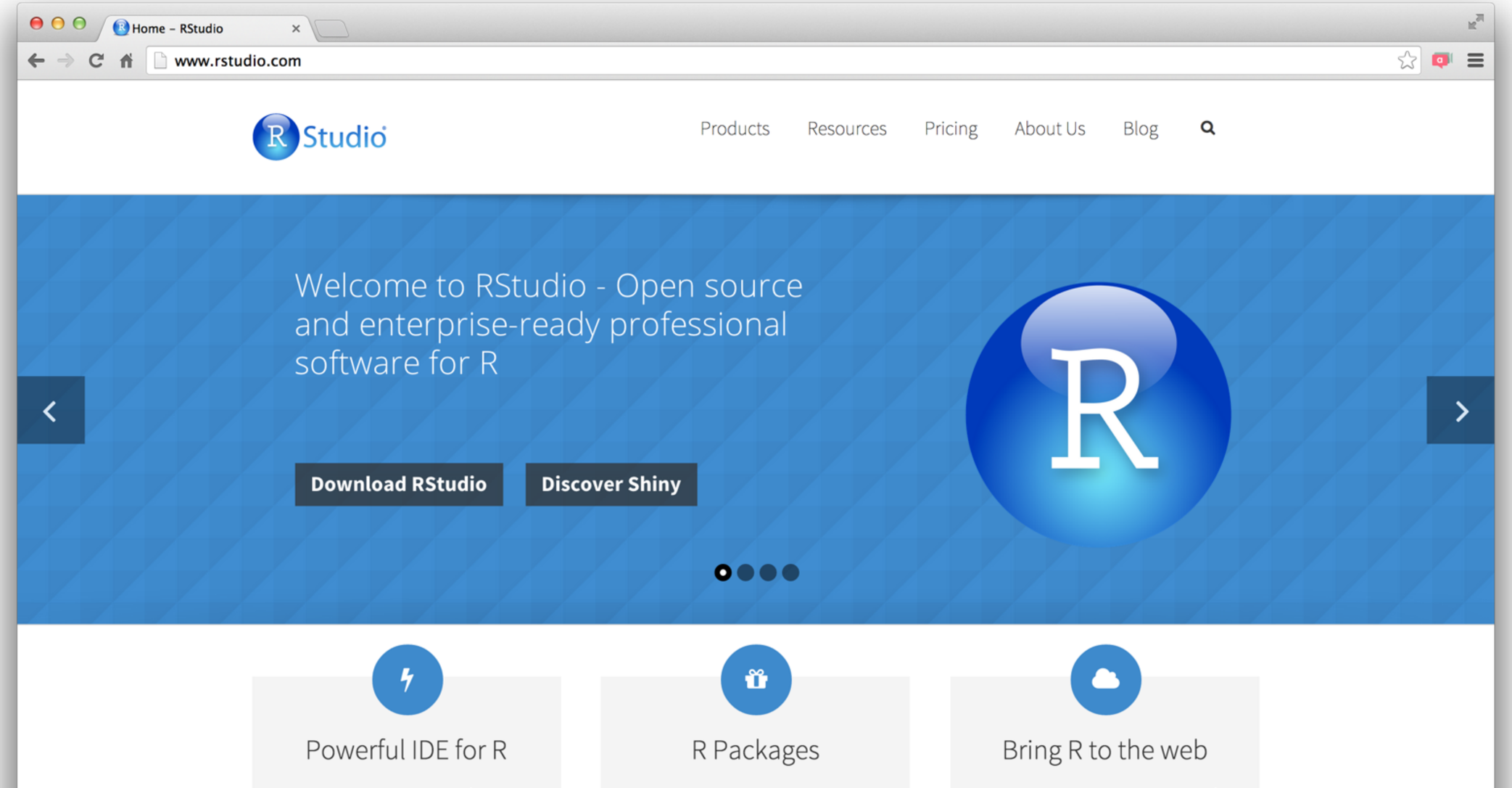
Data Science with R

R's tools for data science. Reshape2, dplyr, and ggplot2 packages.

- Tidy data
- Data visualization and customizing graphics
- Statistical modeling with R

bit.ly/intro-to-data-science-with-R

www.rstudio.com



Data Wrangling
with dplyr and tidyr
Cheat Sheet

RStudio

Tidy Data - A foundation for wrangling in R

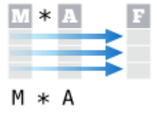
In a tidy data set:

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.



Syntax - Helpful conventions for wrangling

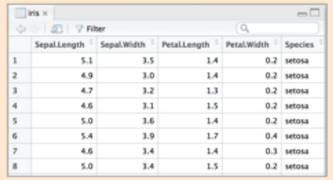
dplyr::**tbl_df(iris)**
Converts data to tbl class. tbl's are easier to examine than data frames. R displays only the data that fits onscreen:

```
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	3.9	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)
```

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.

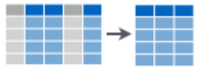
dplyr::**slice(iris, 10:15)**
Select rows by position.

dplyr::**top_n(storms, 2, date)**
Select and order top n entries (by grouped 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.

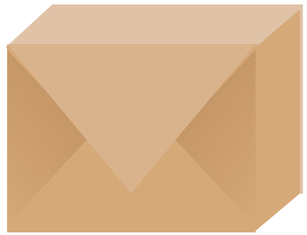
devtools::install_github("rstudio/EDAWR") for data sets

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

http://www.rstudio.com/resources/cheatsheets/

Final Project

nycflights13



A package of data sets about airline travel in 2013 in New York City.

```
library(nycflights13)
?airlines
?airports
?flights
?planes
?weather
```

flights

(336,776 x 16)

On-time data for all flights that departed NYC (i.e. JFK, LGA or EWR) in 2013.

year	month	day	dep_time	dep_delay	arr_time	arr_delay	carrier	tailnum	flight	origin	dest	air_time	distance	hour	minute
2013	1	1	517	2	830	11	UA	N14228	1545	EWR	IAH	227	1400	5	17
2013	1	1	533	4	850	20	UA	N24211	1714	LGA	IAH	227	1416	5	33
2013	1	1	542	2	923	33	AA	N619AA	1141	JFK	MIA	160	1089	5	42
2013	1	1	544	-1	1004	-18	B6	N804JB	725	JFK	BQN	183	1576	5	44
2013	1	1	554	-6	812	-25	DL	N668DN	461	LGA	ATL	116	762	5	54
2013	1	1	554	-4	740	12	UA	N39463	1696	EWR	ORD	150	1065	5	54

flights

(336,776 x 16)

On-time data for all flights that departed NYC (i.e. JFK, LGA or EWR) in 2013.

year	month	day	dep_time	dep_delay	arr_time	arr_delay	carrier	tailnum	flight	origin	dest	air_time	distance	hour	minute
2013	1	1	517	2	830	11	UA	N14228	1545	EWR	IAH	227	1400	5	17
2013	1	1	533	4	850	20	UA	N24211	1714	LGA	IAH	227	1416	5	33
2013	1	1	542	2	923	33	AA	N619AA	1141	JFK	MIA	160	1089	5	42
2013	1	1	544	-1	1004	-18	B6	N804JB	725	JFK	BQN	183	1576	5	44
2013	1	1	554	-6	812	-25	DL	N668DN	461	LGA	ATL	116	762	5	54
2013	1	1	554	-4	740	12	UA	N39463	1696	EWR	ORD	150	1065	5	54

airlines

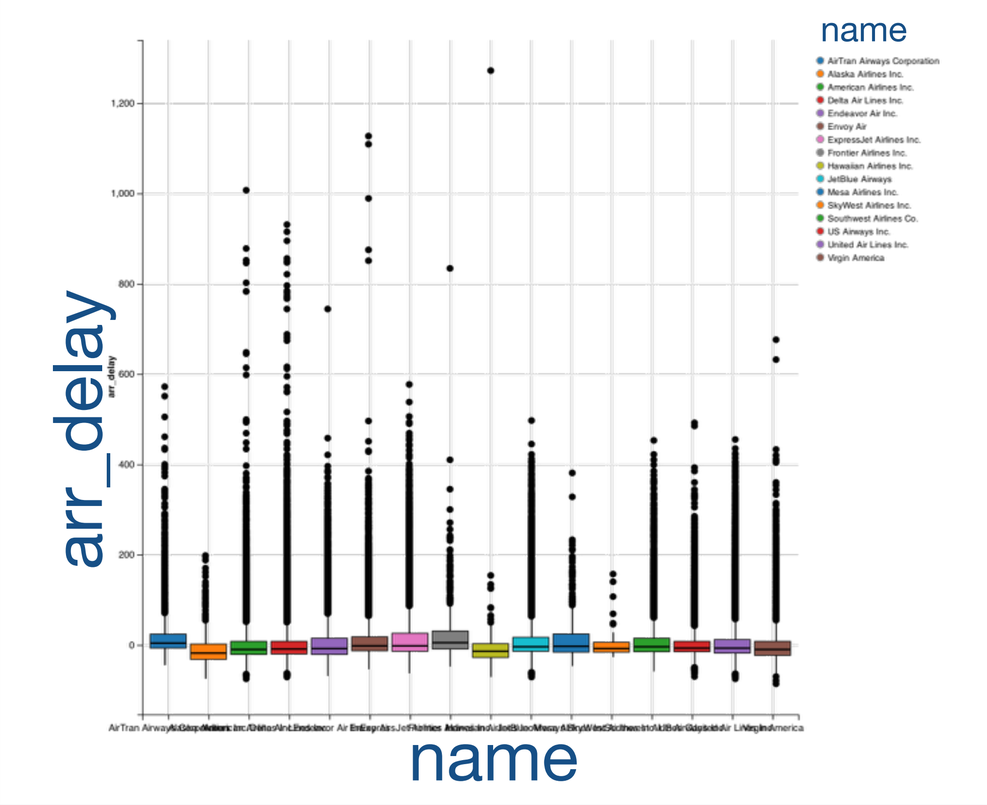
(16 x 2)

Airline names by carrier code

carrier	name
9E	Endeavor Air Inc.
AA	American Airlines Inc.
AS	Alaska Airlines Inc.
B6	JetBlue Airways
DL	Delta Air Lines Inc.
EV	ExpressJet Airlines Inc.

Final project

Use the flights and airlines datasets to recreate the plot on the left and the table on the right.



(16 x 2)

name	median (arrival delay)
Frontier Airlines Inc.	6
AirTran Airways Corporation	5
Envoy Air	-1
ExpressJet Airlines Inc.	-1
Mesa Airlines Inc.	-2
JetBlue Airways	-3
Southwest Airlines Co.	-3
United Air Lines Inc.	-6
US Airways Inc.	-6
Endeavor Air Inc.	-7

Plot

1 Join

2 Filter (NA's)

3 Visualize

Table

1 Join

2 Filter (NA's)

3 Group

4 Summarise

5 Arrange

Plot

```
flights %>%
  left_join(airlines, by = "carrier") %>%
  filter(!is.na(arr_delay)) %>%
  ggvis(x = ~name, y = ~arr_delay, fill = ~name) %>%
  layer_boxplots()
```

Table

```
flights %>%
  left_join(airlines, by = "carrier") %>%
  filter(!is.na(arr_delay)) %>%
  group_by(name) %>%
  summarise(median = median(arr_delay)) %>%
  arrange(desc(median))
```