R Review

INFO 370

Learning Objectives

Review the fundamentals of R for data science:

- Wrangling 2D data structures with **dplyr**
- Using **tidyr** to reshape data
- Plotting with **ggplot2**

Perform EDA in R and compare the process to using Python

*Reminder: reference the INFO 201 course book for more information

Wrangling Data

DPLYR

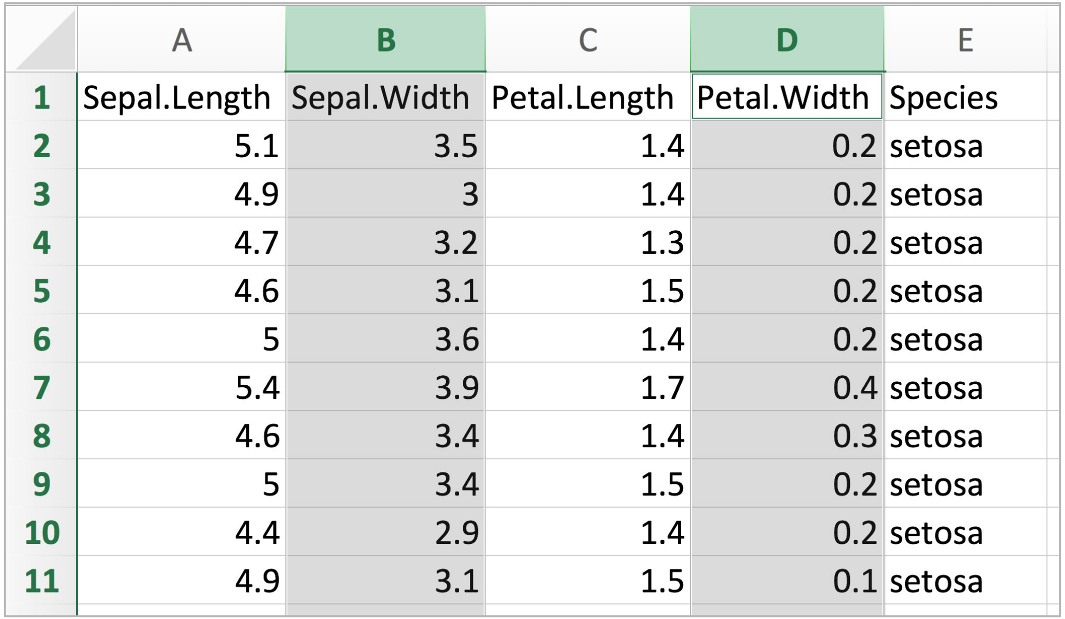
"A grammar for data manipulation"

Provides verbs for common tasks

Make your code easier to write and read

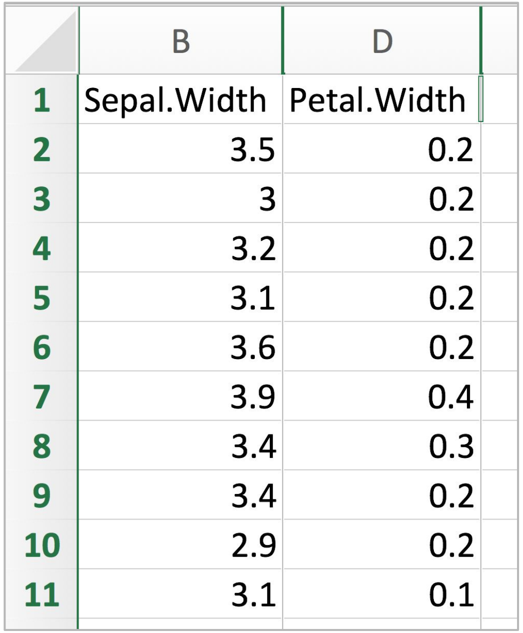
Written by Hadley Wickham

iris



	A	B	C	D	E
1	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5	3.6	1.4	0.2	setosa
7	5.4	3.9	1.7	0.4	setosa
8	4.6	3.4	1.4	0.3	setosa
9	5	3.4	1.5	0.2	setosa
10	4.4	2.9	1.4	0.2	setosa
11	4.9	3.1	1.5	0.1	setosa

widths

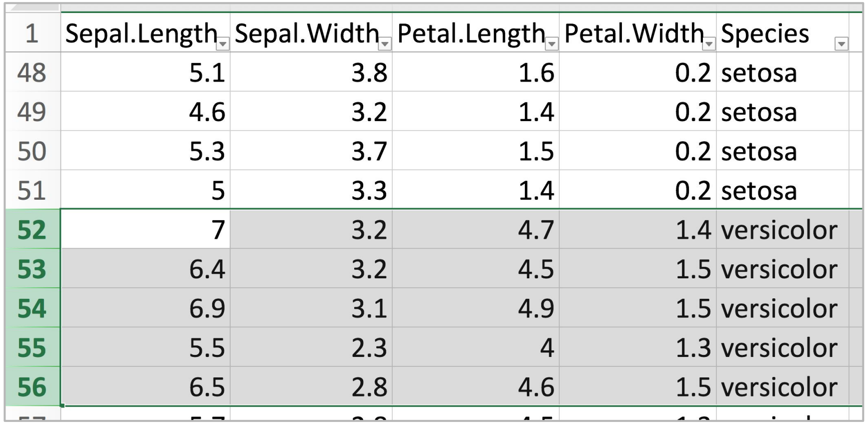


	B	D
1	Sepal.Width	Petal.Width
2	3.5	0.2
3	3	0.2
4	3.2	0.2
5	3.1	0.2
6	3.6	0.2
7	3.9	0.4
8	3.4	0.3
9	3.4	0.2
10	2.9	0.2
11	3.1	0.1

```r
widths <- select(iris, Sepal.Width, Petal.Width)
```

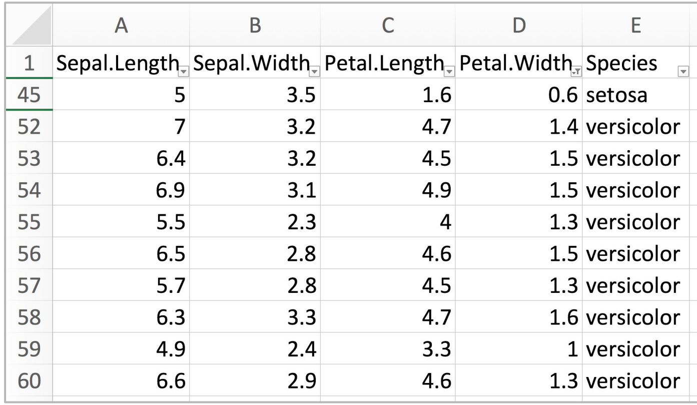
Select

iris



1	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
48	5.1	3.8	1.6	0.2	setosa
49	4.6	3.2	1.4	0.2	setosa
50	5.3	3.7	1.5	0.2	setosa
51	5	3.3	1.4	0.2	setosa
52	7	3.2	4.7	1.4	versicolor
53	6.4	3.2	4.5	1.5	versicolor
54	6.9	3.1	4.9	1.5	versicolor
55	5.5	2.3	4	1.3	versicolor
56	6.5	2.8	4.6	1.5	versicolor

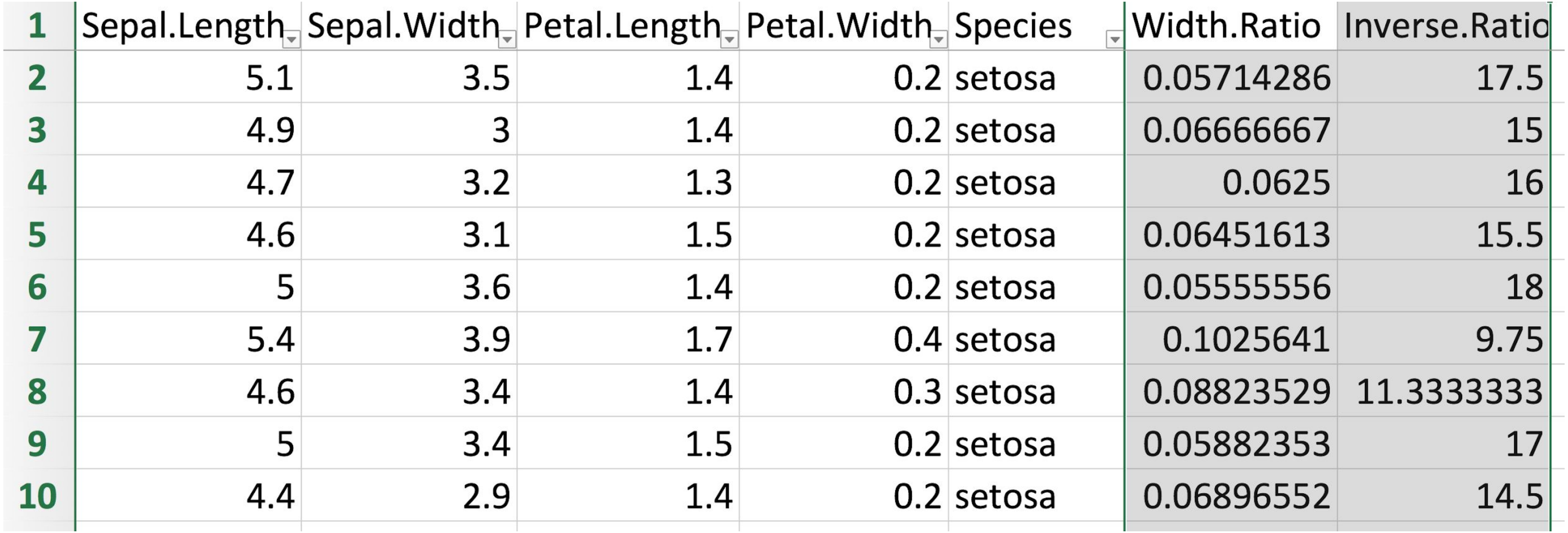
large.widths



	A	B	C	D	E
1	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
45	5	3.5	1.6	0.6	setosa
52	7	3.2	4.7	1.4	versicolor
53	6.4	3.2	4.5	1.5	versicolor
54	6.9	3.1	4.9	1.5	versicolor
55	5.5	2.3	4	1.3	versicolor
56	6.5	2.8	4.6	1.5	versicolor
57	5.7	2.8	4.5	1.3	versicolor
58	6.3	3.3	4.7	1.6	versicolor
59	4.9	2.4	3.3	1	versicolor
60	6.6	2.9	4.6	1.3	versicolor

```
large.widths <- filter(iris, Sepal.Width > .6)
```

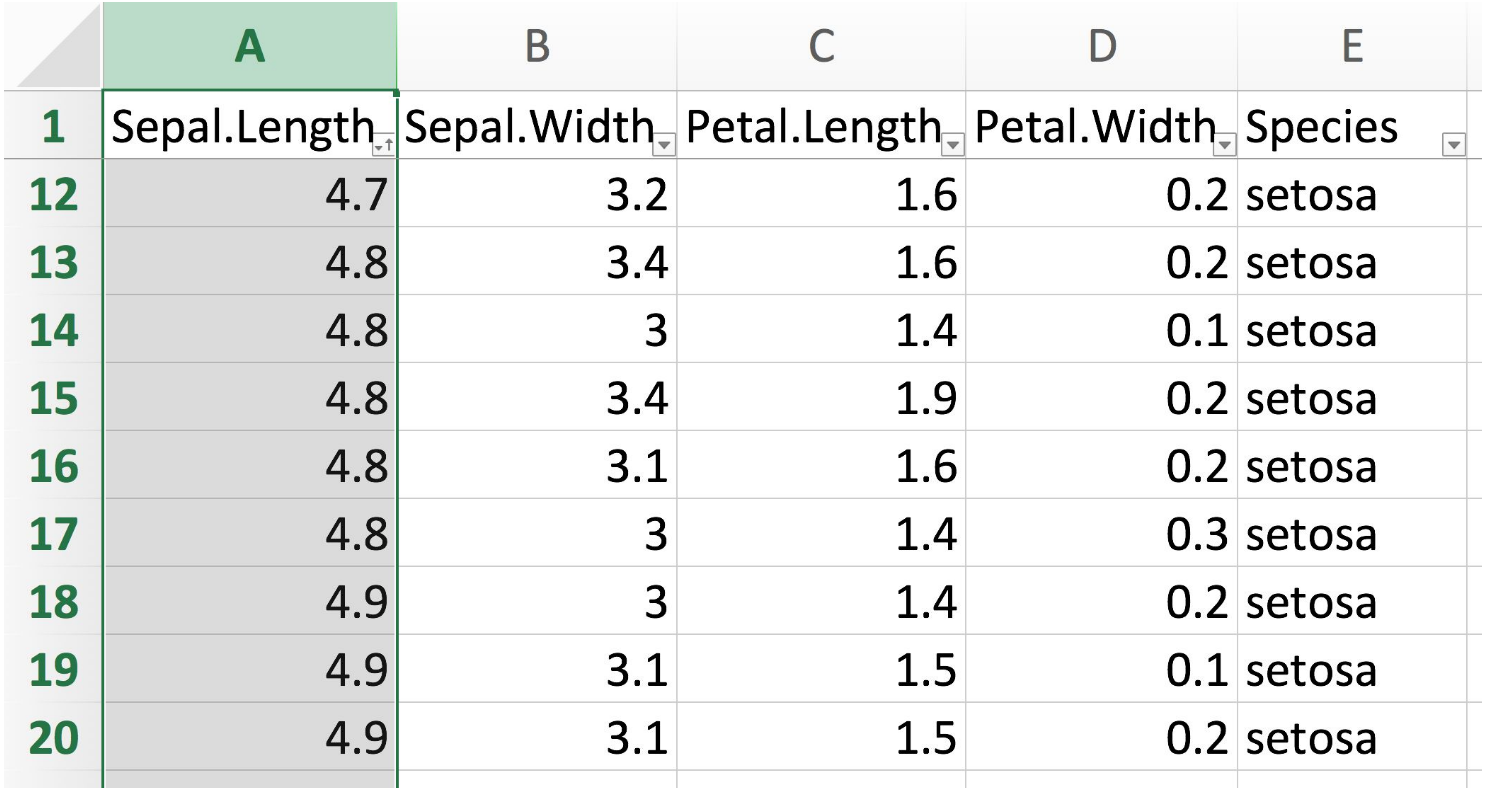
Filter



1	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species	Width.Ratio	Inverse.Ratio
2	5.1	3.5	1.4	0.2	setosa	0.05714286	17.5
3	4.9	3	1.4	0.2	setosa	0.06666667	15
4	4.7	3.2	1.3	0.2	setosa	0.0625	16
5	4.6	3.1	1.5	0.2	setosa	0.06451613	15.5
6	5	3.6	1.4	0.2	setosa	0.05555556	18
7	5.4	3.9	1.7	0.4	setosa	0.1025641	9.75
8	4.6	3.4	1.4	0.3	setosa	0.08823529	11.3333333
9	5	3.4	1.5	0.2	setosa	0.05882353	17
10	4.4	2.9	1.4	0.2	setosa	0.06896552	14.5

```
new.df <- mutate(iris,
                 Width.Ratio = Petal.Width/Sepal.Width,
                 Inverse.Ratio = 1/Width.Ratio)
```

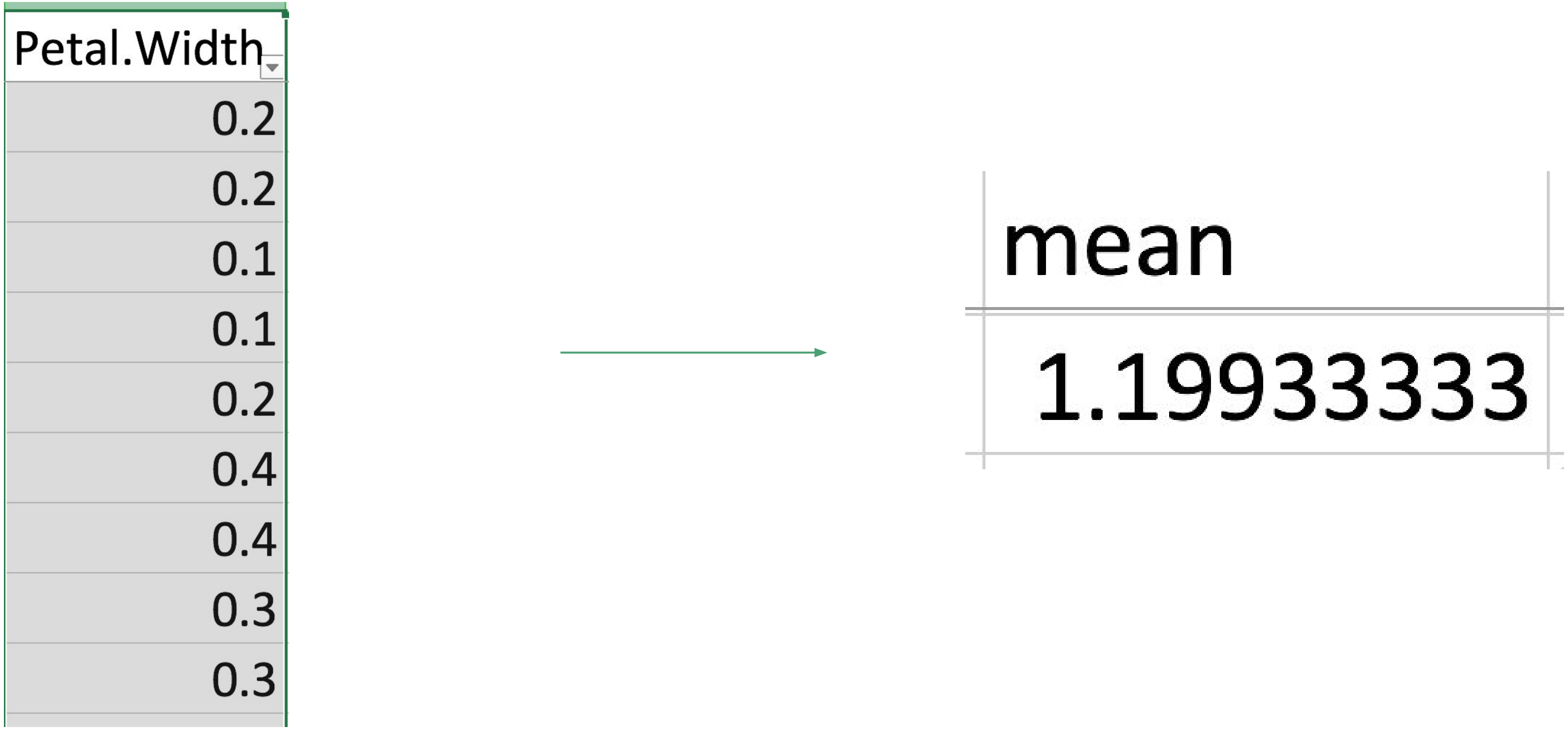
Mutate



	A	B	C	D	E
1	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
12	4.7	3.2	1.6	0.2	setosa
13	4.8	3.4	1.6	0.2	setosa
14	4.8	3	1.4	0.1	setosa
15	4.8	3.4	1.9	0.2	setosa
16	4.8	3.1	1.6	0.2	setosa
17	4.8	3	1.4	0.3	setosa
18	4.9	3	1.4	0.2	setosa
19	4.9	3.1	1.5	0.1	setosa
20	4.9	3.1	1.5	0.2	setosa

```
sorted <- arrange(iris, Sepal.Length)
```

Arrange

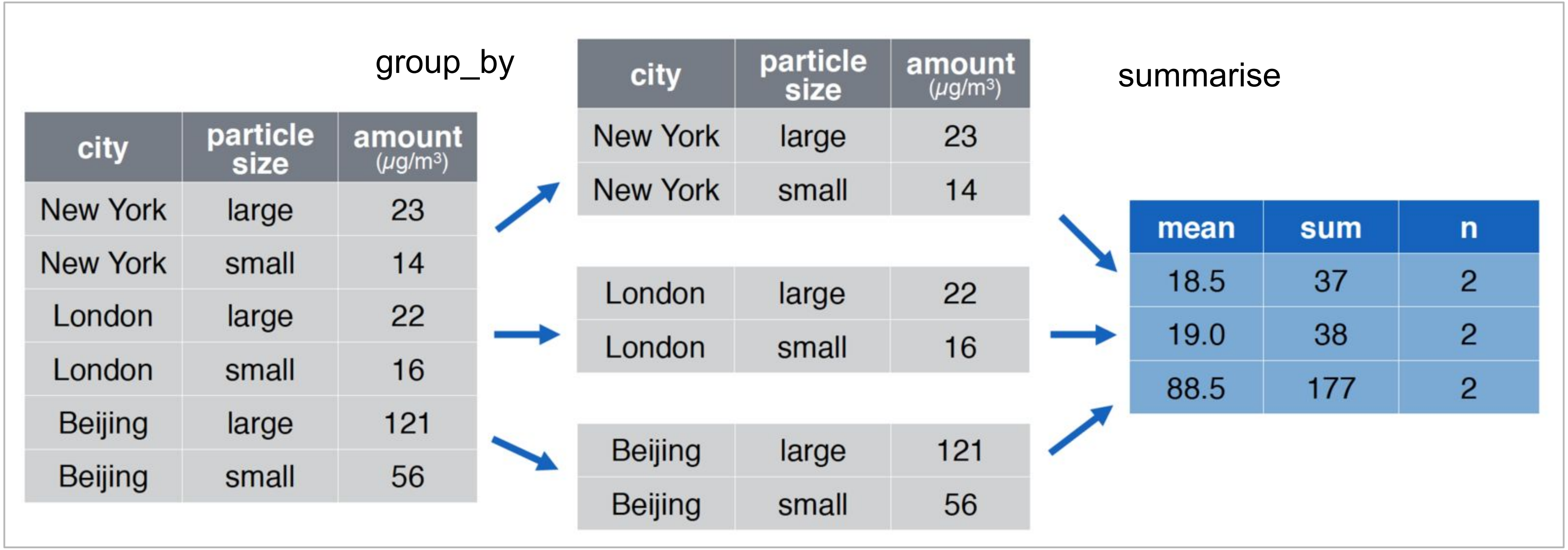


Petal.Width
0.2
0.2
0.1
0.1
0.2
0.4
0.4
0.3
0.3

mean
1.19933333

```
mean.width <- summarise(iris, mean = mean(Petal.Width))
```

Summarize



group_by

city	particle size	amount (μg/m^3)
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	particle size	amount (μg/m^3)
New York	large	23
New York	small	14

London	large	22
London	small	16

Beijing	large	121
Beijing	small	56

summarise

mean	sum	n
18.5	37	2
19.0	38	2
88.5	177	2

```
# Group the pollution data.frame by city for comparison
pollution <- group_by(pollution, city) %>%
             summarise(mean = mean(amount), sum = sum(amount), n = n())
```

Group-by, then Summarize

The Pipe Operator

Takes the ***result from one function*** and passes it in as the ***first argument*** to the next function

Part of the DPLYR package

Written in R as %>% (use the shortcut)

This will completely simplify your code

```r
# Begin your piped operation: filter down to only four cylinder cars
best.car.name <- mutate(mtcars, car.name = row.names(mtcars)) %>%
                       filter(cyl == 4) %>%
                       filter(mpg == max(mpg)) %>%
                       select(car.name)
```

Pipe Operator!

Reshaping Data

Tidy Data

The goal of tidyr is to help you create tidy data. Tidy data is data where:

- Each variable is in a column.
- Each observation is a row.
- Each value is a cell.

Quote from documentation

*You might need to take on a **different shape** for creating graphics

Reshaping Data

Two common reshaping operations in the tidyr package

- **gather()** takes multiple columns, and gathers them into key-value pairs: it makes “wide” data longer.
- **spread()** takes two columns (key & value) and spreads in to multiple columns, it makes “long” data wider.

```r
# Make a data.frame
library(tidyr)
library(ggplot2)
students <- data.frame(
  names=c('Mason', 'Tabi', 'Bryce'),
  math_exam1 = c(91, 82, 93),
  math_exam2 = c(88, 79, 77),
  spanish_exam1 = c(79, 88, 92),
  spanish_exam2 = c(99, 92, 92)
)

# Make long data (by student-exam)
students.exam.long <- gather(students, exam, score, -names)

# Make a group of histograms, one for each exam (facet by exam)
ggplot(students.exam.long) + geom_bar(mapping=aes(score)) + facet_wrap(~exam)

# Make wide data (by exam)
spread(students.exam.long, names, score)
```

New Key

Columns
to gather

New value

Gather and spread

Graphing Data

Grammar of Graphics

Same principles of using a grammar of data manipulation

Create a **consistent vocabulary** for the tasks we perform:

- **Data** to be shown in the plot
- **Geom**etric objects we wish to display
- **Aes**thetic mappings between our data values and their visual encodings
- **Stat**istical transformations to be performed on the data
- **Scales** of values to be applied to our aesthetics
- **Coord**inate system to organize our geometries
- **Facets** (groups) of our data to show in different plots (small multiples)

Basic use

Use the **ggplot()** function to draw a plot, then describe elements via the grammar

The **aes** function describes *which aesthetics* (x position, color, etc.) should be driven by *which data*

Data to plot

Add a geometry

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

Geometry to add (circles)

Describe aesthetic mapping from data space to a visual space

Fork and clone this repo for in-class code-alongs! (ungraded)

class/r-review

notebook-set-2

Upcoming...

Notebook set 2 due **Monday night**

Reading 2 (probability and statistics) due **next Tuesday** before class

- Late submissions not accepted!

Next week: Developing metrics + basic statistical tests