Data wrangling & manipulation in R

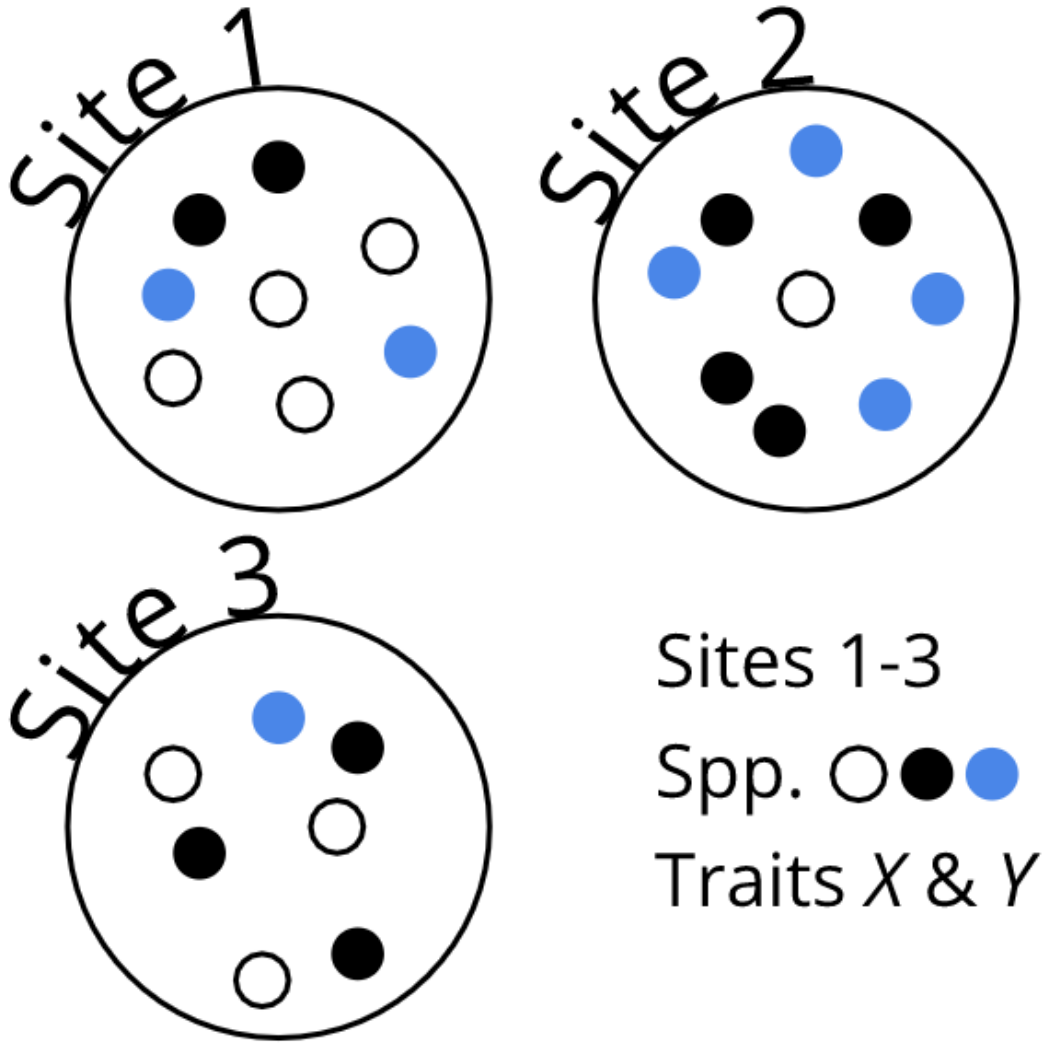
Dept. Biological Sciences Postgraduate Workshop

Ruan van Mazijk, MSc candidate

2019-05-17

Motivation

Motivation



An example data-collection scenario in biology

Site 1					Site 1					
Sp. 1		Sp. 2		Sp. 3		Sp. 1		Sp. 2		Sp. 3	
X	Y	X	Y	X	Y	X	Y	X	Y	X	Y

One way to lay out your collected data...

Site 1			Site 2		
Sp.	X	Y	Sp.	X	Y

Another way...

Site	Sp.	X	Y

The “best” way. Will make your life easiest in the long-term.

Workshop outline

- Embracing the rectangle

Workshop outline

- Embracing the rectangle
- **Making** your data rectangular

Workshop outline

- Embracing the rectangle
- **Making** your data rectangular
- Things to see & do in rectangle land

Workshop outline

- Embracing the rectangle
- **Making** your data rectangular
- Things to see & do in rectangle land
- `mutate()` & friends—How to extend your raw dataset

Workshop outline

- Embracing the rectangle
- **Making** your data rectangular
- Things to see & do in rectangle land
- `mutate()` & friends—How to extend your raw dataset
- ~~Complicated~~ Exotic problems

Embracing the rectangle

Embracing the rectangle

Long vs wide data

Remember this?

Site 1						Site 1					
Sp. 1		Sp. 2		Sp. 3		Sp. 1		Sp. 2		Sp. 3	
X	Y	X	Y	X	Y	X	Y	X	Y	X	Y

Embracing the rectangle

Long vs wide data

Remember this?

Site 1						Site 1					
Sp. 1		Sp. 2		Sp. 3		Sp. 1		Sp. 2		Sp. 3	
X	Y	X	Y	X	Y	X	Y	X	Y	X	Y

This is *wide-form* data. Let's move away from that…

Using the `iris` dataset built into R!

Wide-form data

Wide-form data

```
## $setosa
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1          5.1         3.5          1.4         0.2
## 2          4.9         3.0          1.4         0.2
##
## $versicolor
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1          7.0         3.2          4.7         1.4
## 2          6.4         3.2          4.5         1.5
##
## $virginica
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1          6.3         3.3          6.0         2.5
## 2          5.8         2.7          5.1         1.9
```

Classic long-form data

Classic long-form data

```
##    Species Sepal.Length Sepal.Width Petal.Length Petal.W
## 1    setosa          5.1         3.5          1.4       0.2
## 2    setosa          4.9         3.0          1.4       0.2
## 3    setosa          4.7         3.2          1.3       0.2
## 4 versicolor          7.0         3.2          4.7       1.4
## 5 versicolor          6.4         3.2          4.5       1.5
## 6 versicolor          6.9         3.1          4.9       1.5
## 7  virginica          6.3         3.3          6.0       2.5
## 8  virginica          5.8         2.7          5.1       1.9
## 9  virginica          7.1         3.0          5.9       2.1
```

We can get longer…

We can get longer...

```
##       Species        trait trait_value
## 1      setosa Sepal.Length         5.1
## 2  versicolor Sepal.Length         7.0
## 3   virginica Sepal.Length         6.3
## 4      setosa  Sepal.Width         3.5
## 5  versicolor  Sepal.Width         3.2
## 6   virginica  Sepal.Width         3.3
## 7      setosa Petal.Length         1.4
## 8  versicolor Petal.Length         4.7
## 9   virginica Petal.Length         6.0
## 10     setosa  Petal.Width         0.2
## 11 versicolor  Petal.Width         1.4
## 12  virginica  Petal.Width         2.5
```

The advantages of long data

- Machine-readable

The advantages of long data

- Machine-readable
- The standard for most software/R-functions (e.g. `lm()`, `plot()`, `ggplot()`)

The advantages of long data

- Machine-readable
- The standard for most software/R-functions
  (e.g. `lm()`, `plot()`, `ggplot()`)
- How most statistical methods treat data mathematically

The advantages of long data

- Machine-readable
- The standard for most software/R-functions
  (e.g. `lm()`, `plot()`, `ggplot()`)
- How most statistical methods treat data mathematically
- Easier to subset & wrangle further!

Making your data rectangular

Making your data rectangular

What are your options?

1. (Easiest to lay it out like that from the start…)
    - (Many tools (to follow) assume your data is nice & *tidy*)

Making your data rectangular

What are your options?

1. (Easiest to lay it out like that from the start…)
    - (Many tools (to follow) assume your data is nice & *tidy*)

2. *Careful* Excel work
    - Risky…

3. Use R!
    - Many tools *also* help in *tidying* data
    - Namely, the package `tidyr`

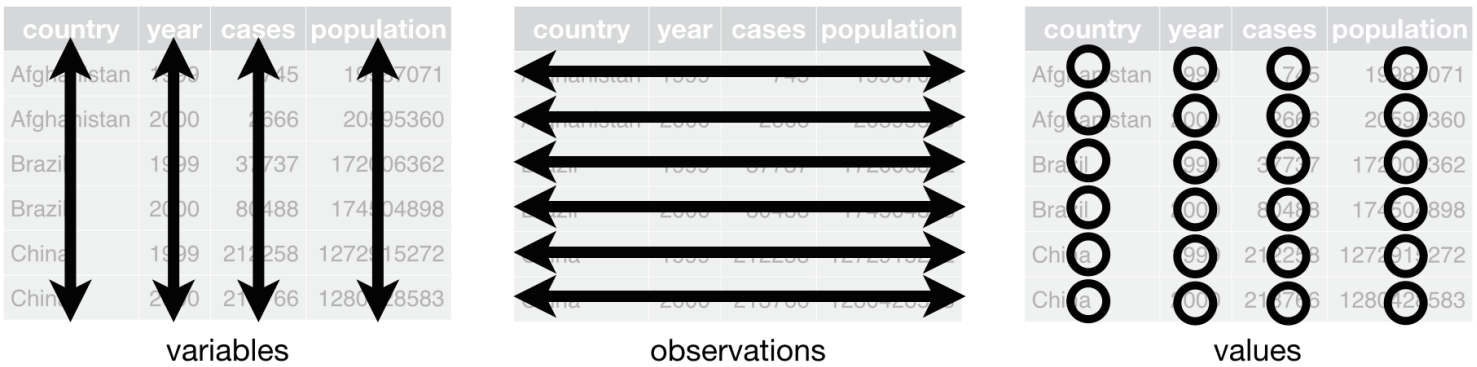
`tidyr::`

An R-package all about getting to *this*[1]:

[1] CC BY-NC-ND 3.0 Grolemund & Wickham 2017. *R for Data Science*

tidyr::

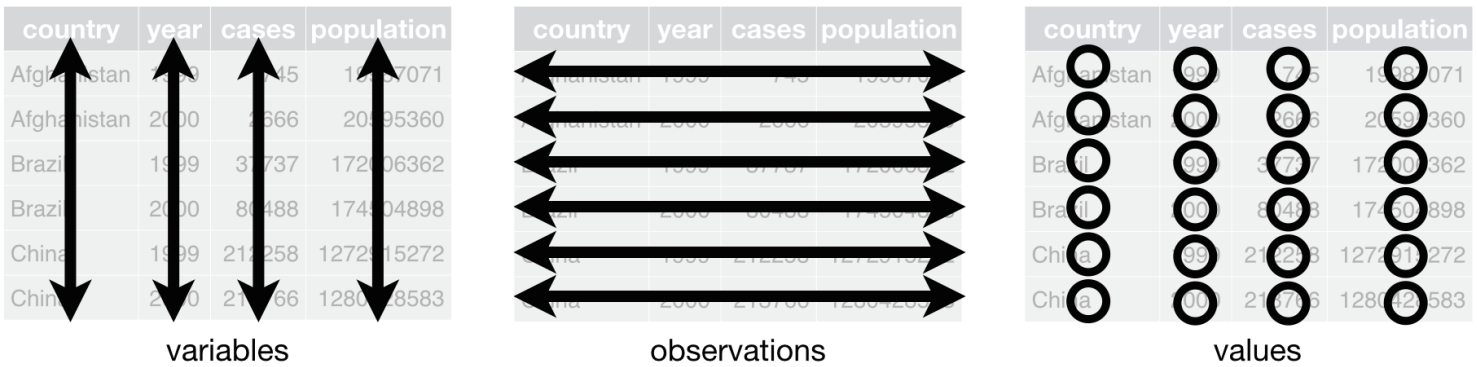
An R-package all about getting to *this*[1]:



[1] CC BY-NC-ND 3.0 Grolemund & Wickham 2017. *R for Data Science*

`tidyr::`

An R-package all about getting to *this*[1]:

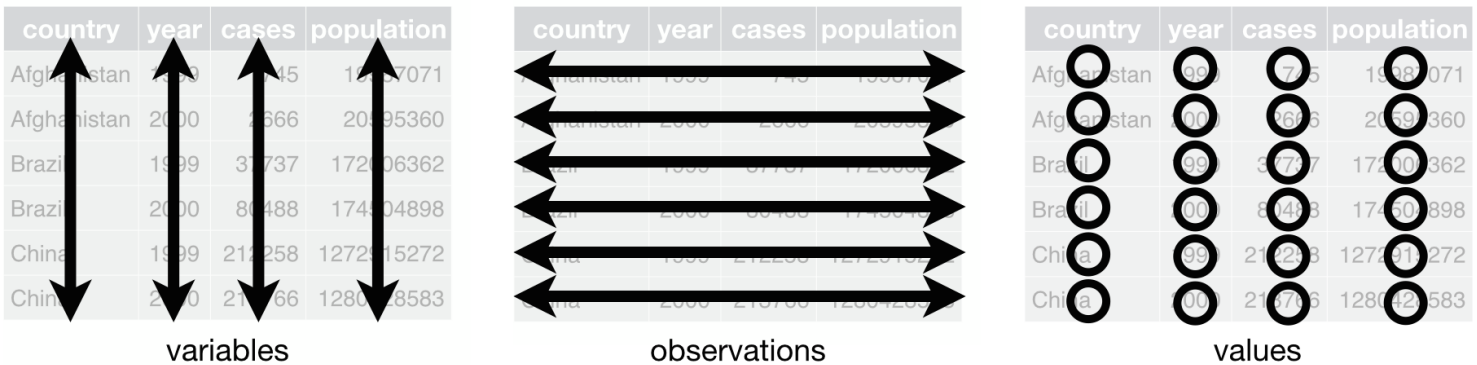


1. Each **variable** must have its own **column**.

[1] CC BY-NC-ND 3.0 Grolemund & Wickham 2017. *R for Data Science*

`tidyr::`

An R-package all about getting to *this*[1]:

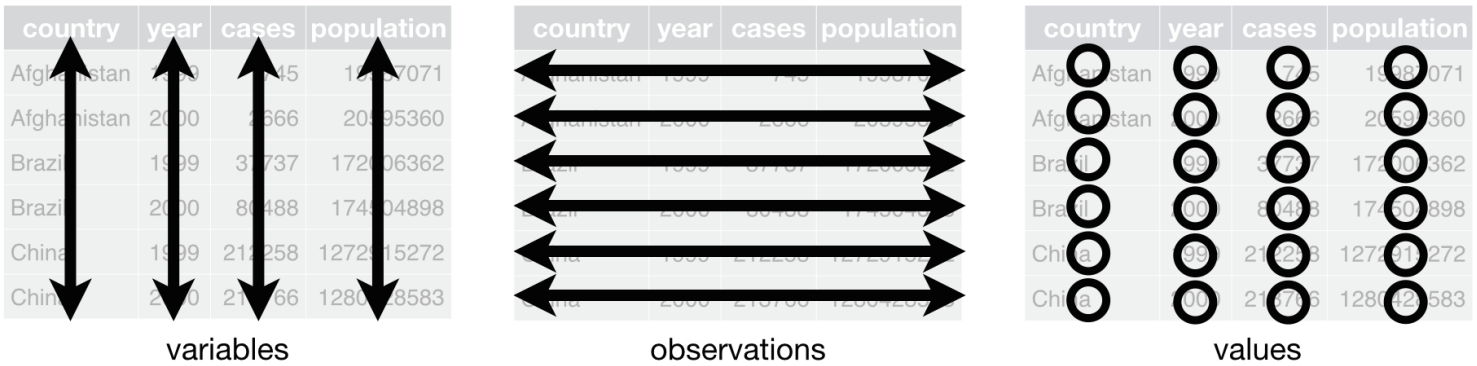


1. Each **variable** must have its own **column**.
2. Each **observation** must have its own **row**.

[1] CC BY-NC-ND 3.0 Grolemund & Wickham 2017. *R for Data Science*

`tidyr::`

An R-package all about getting to *this*[1]:



1. Each **variable** must have its own **column**.
2. Each **observation** must have its own **row**.
3. Each **value** must have its own **cell**.

[1] CC BY-NC-ND 3.0 Grolemund & Wickham 2017. *R for Data Science*

`tidyr::` cont.

Verbs to tidy your data

Untidy observations?

```
gather()    # if > 1 observation per row
spread()    # if observations live in > 1 row
```

`tidyr::` cont.

Verbs to tidy your data

Untidy observations?

```
gather()    # if > 1 observation per row
spread()    # if observations live in > 1 row
```

Untidy variables?

```
separate()  # if > 1 variable per column
unite()     # if variables live in > 1 column
```

Note the following when choosing `tidyr::` verbs

- Be clear on what your **observations** are
  - Like, what **unit** of your study counts as an observation
  - E.g. Leaf traits: plant leaf vs plant individual
  - E.g. Reproductive success: egg size vs clutch size
  - **This will depend on your study &/or data!**
- Variables are discrete, seperate ideas

Things to see & do in rectangle land

Using `dplyr::` to work with data

1. Subsetting etc.
2. Extending your dataset

`dplyr::`

Verbs to touch, slice-up, subset & look inside data

```
select()  # !!!
filter()  # !!!
group_by()
summarise()
arrange()
join()
mutate()
```

1. Subsetting etc.

```
select()  # !!!
filter()  # !!!
```

```
iris
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width   Sp
## 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
## 9           4.4         2.9          1.4         0.2    setosa
## 10          4.9         3.1          1.5         0.1    setosa
## 11          5.4         3.7          1.5         0.2    setosa
## 12          4.8         3.4          1.6         0.2    setosa
## 13          4.8         3.0          1.4         0.1    setosa
## 14          4.3         3.0          1.1         0.1    setosa
## 15          5.8         4.0          1.2         0.2    setosa
```

```
iris <- as_tibble(iris)
iris
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Speci
##          <dbl>       <dbl>        <dbl>       <dbl> <fct>
##  1          5.1         3.5          1.4         0.2 setosa
##  2          4.9         3            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           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           3.4          1.5         0.2 setosa
##  9          4.4         2.9          1.4         0.2 setosa
## 10          4.9         3.1          1.5         0.1 setosa
## # ... with 140 more rows
```

```
iris[[1]]
```

~~Complicated~~ Exotic problems