

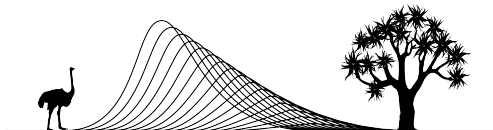
An introduction to the R Tidyverse for effective data wrangling and analysis

Dominic Henry

Centre for Statistics in Ecology, Environment and Conservation UCT

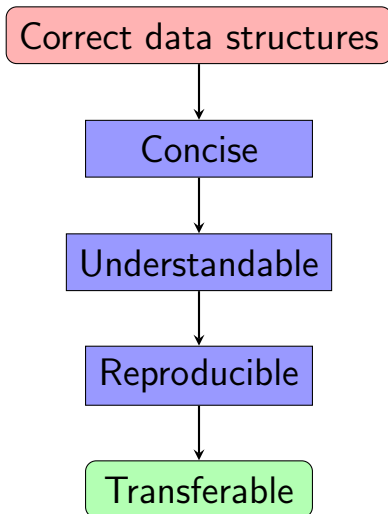
dominichenry@gmail.com

29 March 2018

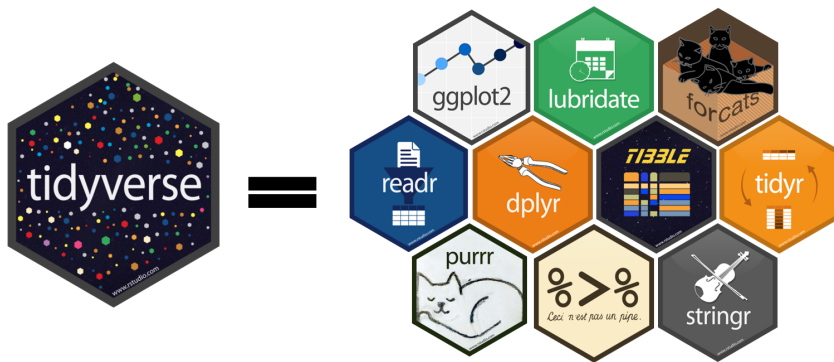


SEEC - Statistics in Ecology, Environment and Conservation

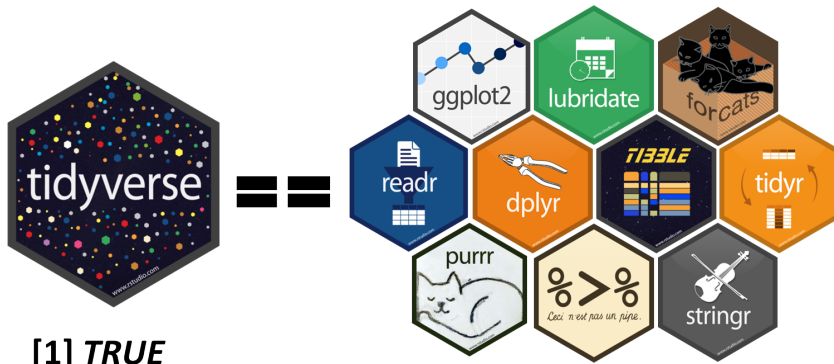
Effective data analysis workflow



What is the tidyverse?



What is the tidyverse?



- Design philosophy
- Grammar
- Data structures and representations

- Design philosophy
- Grammar
- Data structures and representations

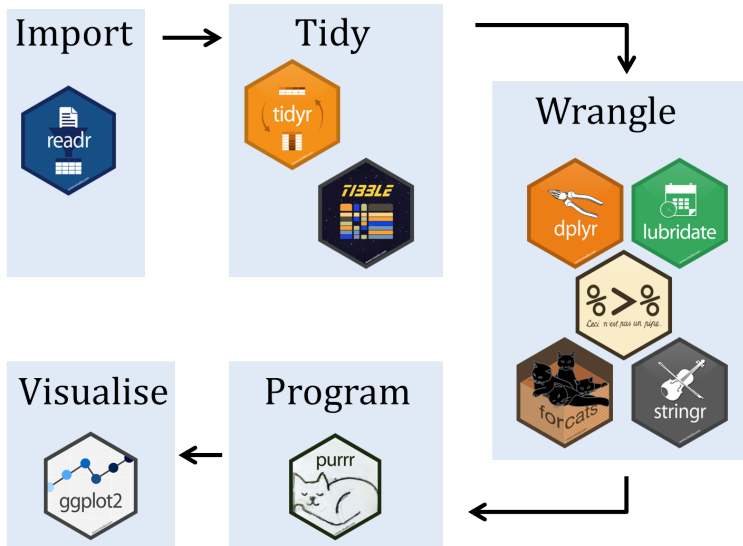
Hadley Wickham



Install and load

```
> library(tidyverse)
-- Attaching packages ----- tidyverse 1.2.1 --
v ggplot2 2.2.1      v purrr   0.2.4
v tibble  1.4.2      v dplyr   0.7.4
v tidyr   0.8.0      v stringr 1.3.0
v readr   1.1.1      v forcats 0.3.0
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
> |
```

Workflow




```
read_csv()
```



`read_csv()`



Behaviour:

- ⇒ Discards row names
- ⇒ Retains non-conventional column names
- ⇒ Ability to detect dates and times
- ⇒ Factors be damned! (characters remain characters)



`read_csv()`

Behaviour:

- ⇒ Discards row names
- ⇒ Retains non-conventional column names
- ⇒ Ability to detect dates and times
- ⇒ Factors be damned! (characters remain characters)

Pros:

- ⇒ Fast (progress bar)
- ⇒ Sneak peek into column types
- ⇒ Creates a tibble object

`tbl_df`



Data frame with modern features

`tbl_df`



Data frame with modern features

Improvements over `data.frame` objects:

- ⇒ Aesthetics and ease of reading
- ⇒ Column type information
- ⇒ Fit to console
- ⇒ Store lists as columns!



```
# A tibble: 476 x 10
  Site Protection Year Month Air_temp Wind_speed abundance richness Latitude Longitude
  <chr> <chr>      <int> <int>   <dbl>   <dbl>   <int>   <int>   <dbl>   <dbl>
1 KZN1~ FP      2012     4    19.0     0.      25      4    -28.3    32.4
2 KZN1~ FP      2012     4    20.0     3.00     9      5    -28.2    32.5
3 KZN1~ FP      2012     4    27.0     5.00    101     7    -28.2    32.5
4 KZN1~ FP      2012     4    29.0     9.00     1      1    -28.2    32.5
5 KZN1~ FP      2012     4    28.2     7.40     6      5    -28.1    32.5
6 KZN1~ FP      2012     4    26.7     3.70    55      4    -28.4    32.4
7 KZN1~ FP      2012     4    23.1     8.50    10      2    -28.0    32.4
8 KZN1~ FP      2012     4    24.3    13.2    51      6    -28.0    32.4
9 KZN1~ FP      2012     4    25.4     3.70     5      2    -28.0    32.4
10 KZN1~ FP      2012     4    27.4     7.40    28      2    -27.9    32.4
# ... with 466 more rows
> |
```

Customise with `global options()` settings

Data structure (tidy data)

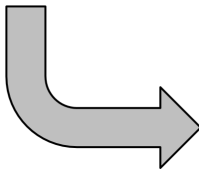
Species	Jan	Feb	Mar
Pied Kingfisher	5	2	7
African Jacana	20	0	23
Cape Teal	0	9	55



Data structure (tidy data)



Species	Jan	Feb	Mar
Pied Kingfisher	5	2	7
African Jacana	20	0	23
Cape Teal	0	9	55

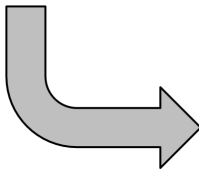


Species	Month	Count
Pied Kingfisher	Jan	5
Pied Kingfisher	Feb	2
Pied Kingfisher	March	7
African Jacana	Jan	20
African Jacana	Feb	0
African Jacana	Mar	23
Cape Teal	Jan	0
Cape Teal	Feb	9
Cape Teal	Mar	55

Data structure (tidy data)



Species	Jan	Feb	Mar
Pied Kingfisher	5	2	7
African Jacana	20	0	23
Cape Teal	0	9	55

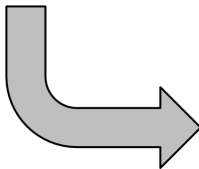


Species	Month	Count
Pied Kingfisher	Jan	5
Pied Kingfisher	Feb	2
Pied Kingfisher	March	7
African Jacana	Jan	20
African Jacana	Feb	0
African Jacana	Mar	23
Cape Teal	Jan	0
Cape Teal	Feb	9
Cape Teal	Mar	55

Data structure (tidy data)



Species	Jan	Feb	Mar
Pied Kingfisher	5	2	7
African Jacana	20	0	23
Cape Teal	0	9	55



`gather()` or `spread()`

Species	Month	Count
Pied Kingfisher	Jan	5
Pied Kingfisher	Feb	2
Pied Kingfisher	March	7
African Jacana	Jan	20
African Jacana	Feb	0
African Jacana	Mar	23
Cape Teal	Jan	0
Cape Teal	Feb	9
Cape Teal	Mar	55



Typical tasks:

- ⇒ Explore structure
- ⇒ Validate observations



Typical tasks:

- ⇒ Explore structure
- ⇒ Validate observations
- ⇒ Create variables
- ⇒ Select observations



Typical tasks:

- ⇒ Explore structure
- ⇒ Validate observations
- ⇒ Create variables
- ⇒ Select observations
- ⇒ Summarise data
- ⇒ Prepare input for models & visualisations

Wrangle



The pipe operator



The pipe operator



```
data %>% f1() %>% f2() %>% f3()
```


The pipe operator



```
data %>% f1() %>% f2() %>% f3()
```

vs.

```
f3(f2(f1(data)))
```



`select()`: choose variables (cols) by name



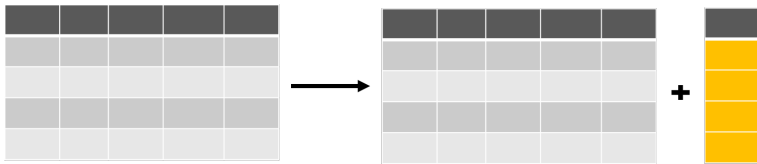


`filter()`: filter observations (rows) based on their value



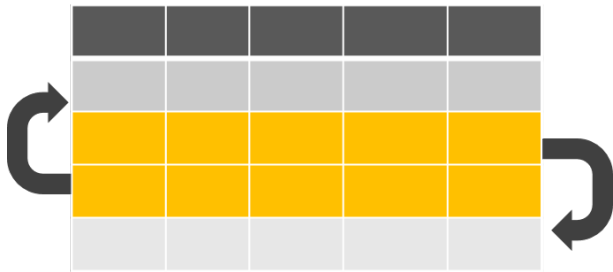


`mutate()`: create new variables from existing ones



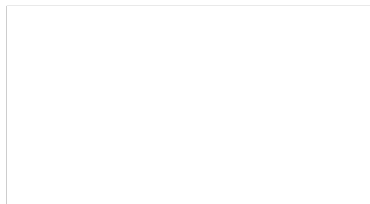
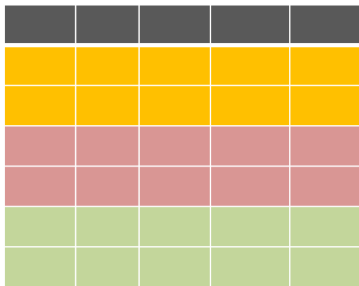


arrange(): change the order of observations





`group_by()`: select a factor by which to group observations



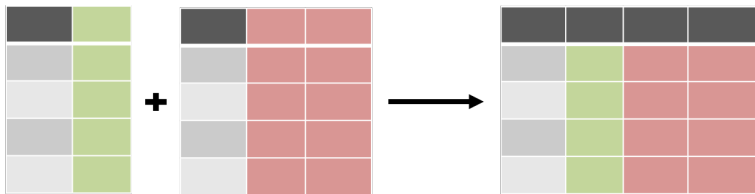


`summarise()`: reduce observations into single value





`*_join()`: combine data frames (`*left`, `right`, `inner`, `full`)



Option 1

```
data1 <- select(data, ...)  
data2 <- filter(data1, ...)  
data3 <- mutate(data2, ...)
```





Option 1

```
data1 <- select(data, ...)  
data2 <- filter(data1, ...)  
data3 <- mutate(data2, ...)
```

Option 2

```
data %>% select(...) %>% filter(...) %>% mutate(...)
```



Advantages

⇒ Improved understanding - reads like a sentence



Advantages

- ⇒ Improved understanding - reads like a sentence
- ⇒ Remove unnecessary intermediate steps



Advantages

- ⇒ Improved understanding - reads like a sentence
- ⇒ Remove unnecessary intermediate steps
- ⇒ Reduce creative effort (naming things sensibly is hard!)



Advantages

- ⇒ Improved understanding - reads like a sentence
- ⇒ Remove unnecessary intermediate steps
- ⇒ Reduce creative effort (naming things sensibly is hard!)
- ⇒ Focus on the final desired output

`dmy_hms()`



Base R is confusing and frustrating

`dmy_hms()`



Base R is confusing and frustrating

Quite simply, lubridate makes it easy to import and perform date time operations!



What do `grep()`, `grepl()`, `regexpr()`, `gsub()`, `gregexpr()` do?



What do `grep()`, `grepl()`, `regexpr()`, `gsub()`, `gregexpr()` do?

How about `str_detect()`, `str_count()`, `str_which()`, `str_locate()`,
`str_extract()`, `str_subset()`, `str_replace()`?

`fct_reorder()`



Have you ever tried to change the factor order on a ggplot?

`fct_reorder()`



Have you ever tried to change the factor order on a ggplot?

Many, many useful functions: <http://forcats.tidyverse.org/reference/index.html>

```
map(.x, .f, ...)
```








- The core of purrr is a set of functions for manipulating vectors

```
map(.x, .f, ...)
```




- The core of purrr is a set of functions for manipulating vectors
- Piping still a big part of the functionality
- Alternative to `apply()` family of functions

 Questions Developer Jobs Tags Users

13 4    


Why use purrr::map instead of lapply?

Ask Question


 64

Is there any reason why I should use

```
map(<list-like-object>, function(x) <do stuff>)
```




instead of

 35

```
lapply(<list-like-object>, function(x) <do stuff>)
```


the output should be the same and the benchmarks I made seem to show that `lapply` is slightly faster (it should be as `map` needs to evaluate all the non-standard-evaluation input).

So is there any reason why for such simple cases I should actually consider switching to `purrr::map`? I am not asking here about one's likes or dislikes about the syntax, other functionalities provided by `purrr` etc., but strictly about comparison of `purrr::map` with `lapply` assuming using the standard evaluation, i.e. `map(<list-like-object>, function(x) <do stuff>)`. Is there any advantage that `purrr::map` has in terms of performance, exception handling etc.? The comments below suggest that it does not, but maybe someone could elaborate a little bit more?

 `purrr`

[share](#) [edit](#)

asked Jul 14 '17 at 10:45


 **Tim**
2,450 ● 1 ● 13 ● 27

asked 7 months ago

viewed 8,962 times

active 17 days ago

BLOG

 [Evaluating Options for Amazon's HQ2 Using Stack Overflow Data](#)

HOT META POSTS

11

[How to handle edits to your question that alter or expand on the information...](#)

3

[What to do \(if anything\) about a strike-through edit to an answer?](#)

7

[Opaque Links in Questions](#)

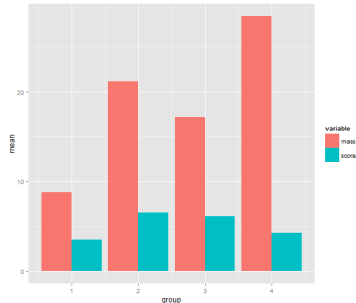
Looking for a job?

[Javascript Developer](#)

<https://stackoverflow.com/questions/45101045/why-use-purrrmap-instead-of-lapply>

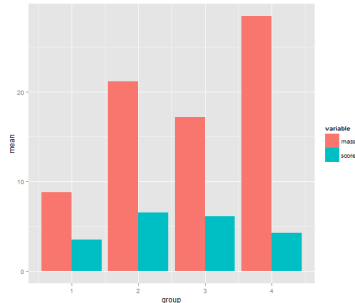


- Reduce number of objects by using dplyr + ggplot via pipe



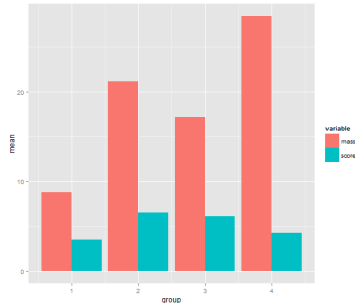


- Reduce number of objects by using dplyr + ggplot via pipe
- Flexible in the preliminary data exploration stage





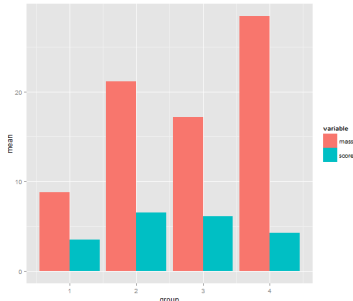
- Reduce number of objects by using dplyr + ggplot via pipe
- Flexible in the preliminary data exploration stage
- Compliment to summary tables





- Reduce number of objects by using dplyr + ggplot via pipe
- Flexible in the preliminary data exploration stage
- Compliment to summary tables

```
data %>% select() %>% filter() %>%  
group_by() %>% summarise() %>%  
ggplot(aes(x, y, fill)) +  
geom_bar()
```



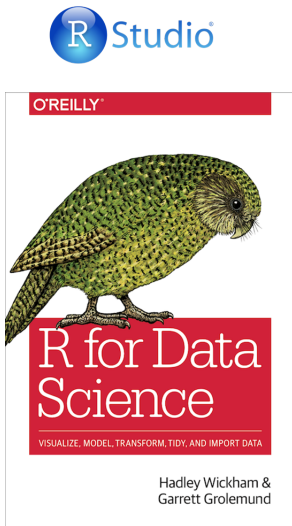
Practical demonstration

1. Point count data
2. Site by species data

Switching from Base R

Base R command	Tidyverse Command	What it does and why you should use the tidyverse version	Comment
read.csv()	read_csv()	reads in a csv file, but its much faster, shows progress bar for large files, can automatically parse data types	also see read_delim(), read_tsv() and readxl::read_xlsx()
sort(), order()	arrange()	sort column(n) within a data frame	see also order_by()
mtcars\$mpg = ...	mutate()	modify a column	see also transmute() which drops existing

<http://www.significantdigits.org/2017/10/switching-from-base-r-to-tidyverse/>



www.tidyverse.org

@hadleywickham

@dataandme

#rstats

 **stackoverflow**

 **Studio** Community

SEEC Stats Toolbox Schedule 2018



Date	Topic	Speaker
29th March	Data wrangling with the R tidyverse	Dominic Henry
26th April	Generalised Linear Mixed Models	Mzabalazo Ngwenya
31st May	Occupancy models	Res Altwegg
26th July	Species distribution models	Vernon Visser
30th August	Handling spatial data	Jasper Slingsby
27th September	Generalised Additive Models	Birgit Erni
25th October	Diagnostics for data exploration and presenting results of regression-type analyses	Greg Distiller
29th November	An introduction into Bayesian models	Allan Clark