purrr for biostatisticians

⚔

with examples

Daniel D. Sjoberg

Memorial Sloan Kettering Cancer Center
Department of Epidemiology and Biostatistics

February, 19 2019

purrr
www.rstudio.com

let's get started

purrr package

purrr enhances R’s functional programming toolkit (read: the apply family of functions) by providing a complete and consistent set of tools for working with functions and vectors

best place to start is the family of functions which allow you to replace many for loops* with code that is more succinct and easier to read

functions transform their input by applying a function to each element and returning a vector the same length as the input

[*] And much much more

base::apply vs purrr::map

base apply is to ______________ as
purrr map is to ______________ ?

base::apply vs purrr::map

base apply is to as
purrr map is to

WHOOPI
No Sex.
No Booze.
No Men.
No Way.
SISTER ACT

WHOOPI
SISTER ACT 2:
BACK IN THE HABIT

www.impawards.com

base::apply

- first argument to is the data; the first argument to is the function
- no consistent way to pass additional arguments; most use , uses , and some require you to create a new anonymous function
- output from is not consistent

base::apply

- first argument to is the data; the first argument to is the function
- no consistent way to pass additional arguments; most use , uses , and some require you to create a new anonymous function
- output from is not consistent

- , , and use to suppress names in output; does not have this argument

purrr::map

- the family has greater consistency among functions
- , , and inputs are the same and allow for flexible input
- consistent methods for passing additional arguments

purrr::map

- the family has greater consistency among functions
- , , and inputs are the same and allow for flexible input
- consistent methods for passing additional arguments

- the output from the map family of functions is predictable and easily modified

use cases

1. subgroup analyses
2. sensitivity analyses
3. reading all files in a folder
4. bootstrap analyses
5. other purrr package functions

usage

usage

usage

pass a function name to

additional function arguments can be passed as well

usage

pass a function name to

additional function arguments can be passed as well

create a new function with

usage

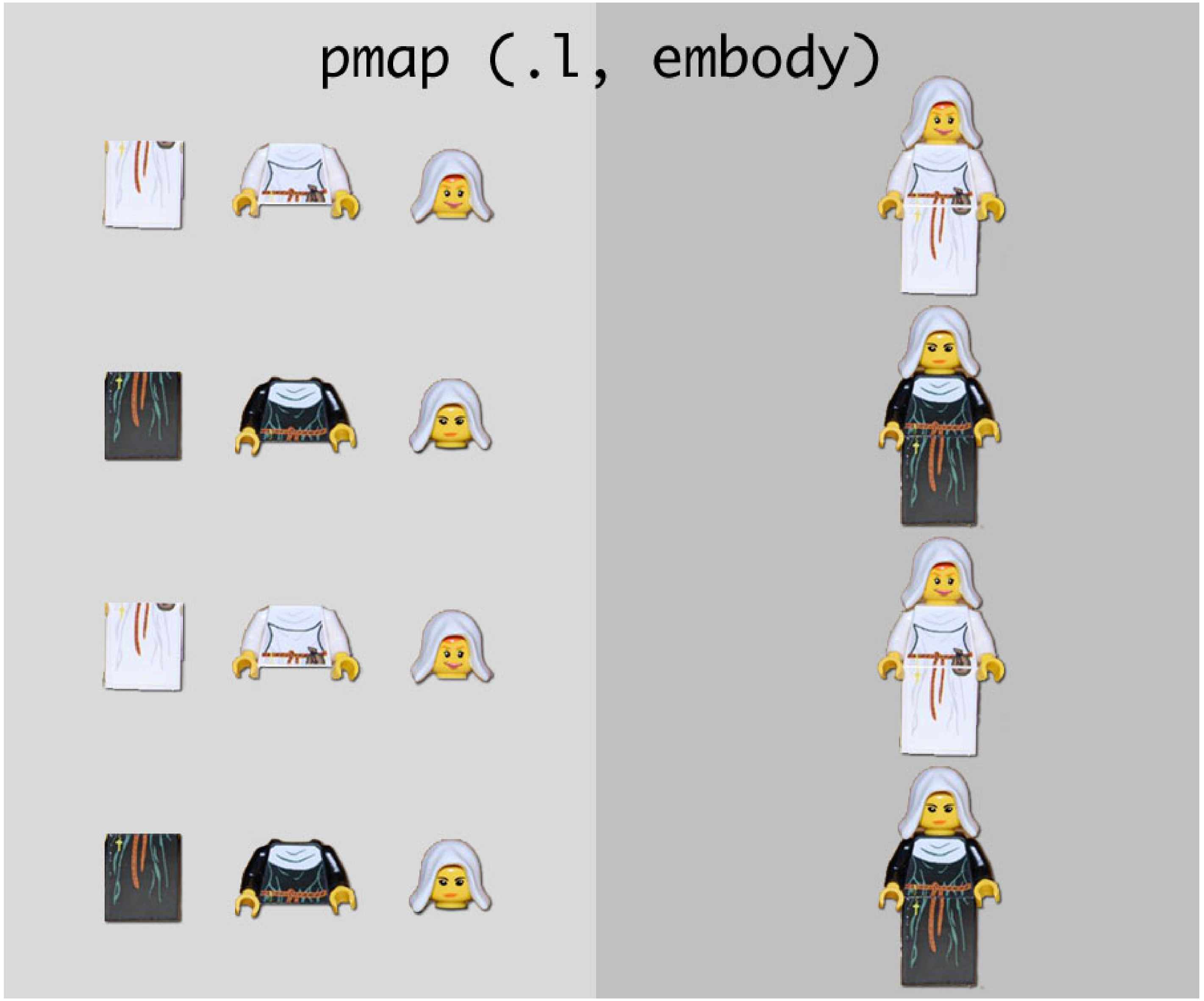
pass a function name to

additional function arguments can be passed as well

create a new function with

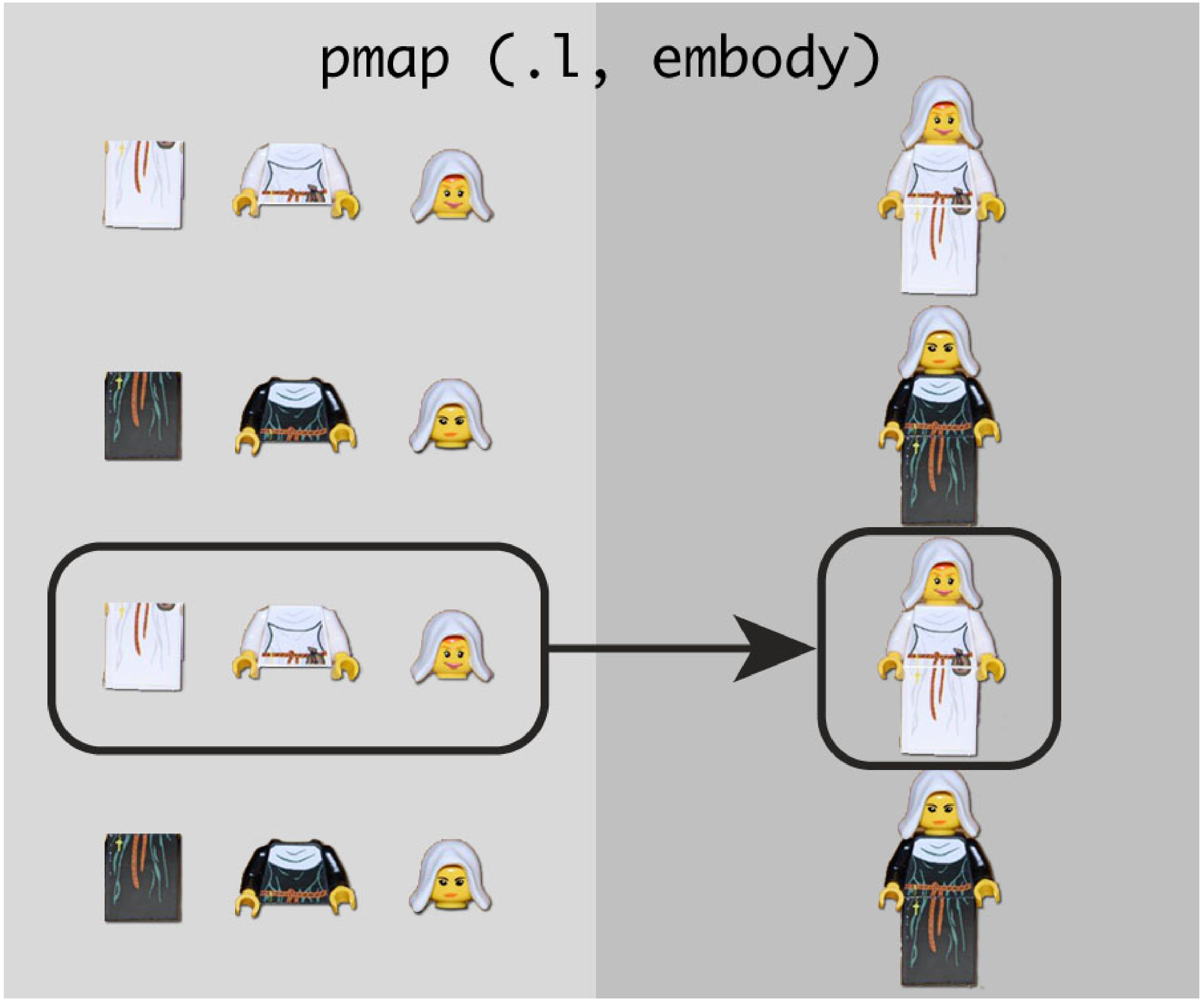
use the " " shortcut to create a function ( )

usage



```
pmap (.l, embody)
```

LEGO analogy by Jenny Bryan



LEGO analogy by Jenny Bryan

trial dataset

		N = 107	N = 93
Age, yrs	192	47 (39, 58)	46 (36, 54)
Marker Level, ng/mL	192	0.61 (0.22, 1.20)	0.72 (0.22, 1.63)
T Stage	200		
T1		25 (23%)	26 (28%)
T2		26 (24%)	23 (25%)
T3		29 (27%)	13 (14%)
T4		27 (25%)	31 (33%)
Grade	200		
I		38 (36%)	29 (31%)
II		34 (32%)	24 (26%)
III		35 (33%)	40 (43%)
Tumor Response	191	52 (51%)	30 (33%)

use cases

1.
2. sensitivity analyses
3. read all files in a folder
4. bootstrap analyses
5. other purrr package functions

subgroup analysis

tidyr::nest + purrr::map

subgroup analysis

tidyr::nest + purrr::map

tibbles

- share the same structure as data frames
- possible to have a list column
- this means you can put any object in a tibble!
- keep related objects together in a row, no matter object complexity

subgroup analysis

- tidyr::nest + dplyr::mutate + purrr::map

subgroup analysis

tidyr::nest + dplyr::mutate + purrr::map

subgroup analysis

tidyr::nest + dplyr::mutate +

output types

the default output of is a list

we can coerce the output type with

map()	list
map_dbl()	double
map_int()	integer
map_lgl()	logical
map_dfr()	tibble (bind_rows)
map_dfc()	tibble (bind_cols)

when using the functions, runs as it typically would with the added step of coercing the output at the end

tip: make sure your code works with before adding

use cases

1. subgroup analyses
2.
3. read all files in a folder
4. bootstrap analyses
5. other purrr package functions

sensitivity analyses

run your analysis among

- all patients ( )
- excluding low grade patients ( )

TRUE
grade != 'I'

sensitivity analyses

run your analysis among

- all patients ( )
- excluding low grade patients ( )

sensitivity analyses

sensitivity analyses

- you can also save figures in a tibble

use cases

1. subgroup analyses
2. sensitivity analyses
3.
4. bootstrap analyses
5. other purrr package functions

read files

store vector of the files you want to import

read files

- store vector of the files you want to import

- use   to read the files
- returns a list where each element is a tibble

read files

- append each of the data sets with the function
- after files have been imported, will create one final tibble

read files

- append each of the data sets with the function
- after files have been imported, will create one final tibble

- include an identifier with a piped

use cases

1. subgroup analyses
2. sensitivity analyses
3. read all files in a folder
4.
5. other purrr package functions

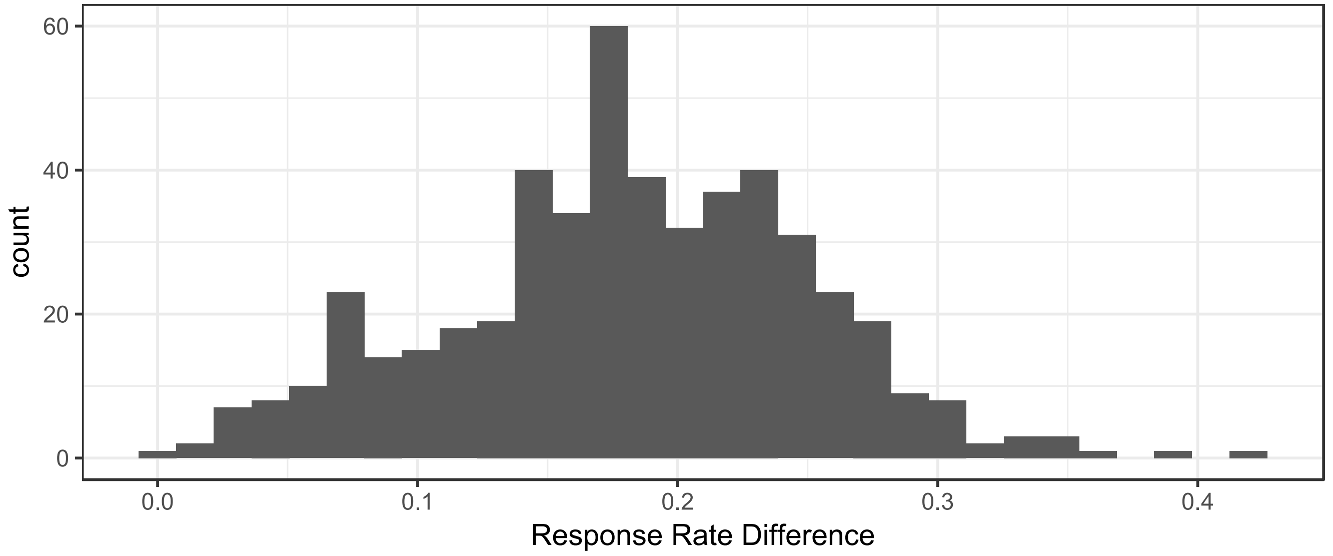
bootstrap analyses

use bootstrap re-sampling to estimate the difference in response rate by treatment

we'll use 500 re-sampled data sets to estimate the standard deviation of the response rate difference

assuming normality of the response rate difference, construct a 95% confidence interval for the difference

bootstrap analyses



bootstrap analyses

bootstrap confidence interval 18% (95% CI 4.6%, 32%)

bootstrap analyses

bootstrap confidence interval 18% (95% CI 4.6%, 32%)

Wald confidence interval 18% (95% CI 4.1%, 31%)

use cases

1. subgroup analyses
2. sensitivity analyses
3. read all files in a folder
4. bootstrap analyses
5.

other purrr functions

-
-
-
- and
- and
-
- , , and
-

Yes, Lauren Hill was in !

unlike and its variants which always return a fixed object type (list for , integer vector for , etc), the family always returns the same type as the input object is short hand for if has names, or if it does not is short hand for if has names, or if it does not recursively traverses nested vectors and map a function at a certain depth

- is equivalent to .
- is equivalent to
- is equivalent to

and

keep or discard elements of a list or vector pluck is similar to " " and selects a single element from a list or vector

use position or name to select item

pluck is easier to read when used with the pipe ( )

like without the factors…finally!

check out , ,

they are similar, but return lists rather than a tibble

, , and

these functions wrap functions

instead of generating side effects through printed output, messages, warnings, and errors, they return enhanced output

wrapped function returns a list with components and

wrapped function instead returns a list with components , , and

wrapped function uses a default value (otherwise) whenever an error occurs

similar to and

- negates a predicate function
- a predicate function returns and , e.g.
- returns a , not the same as operator
- good for piping

done! questions?

slides available at danieldsjoberg.com/purrr-for-biostatisticians

source code available at github.com/ddsjoberg/purrr-for-biostatisticians

www.flaticon.com