Exam 1 review

purrr::map

map: apply a function to each element of a vector or list

output: list (map_dbl, map_lgl, ... give different output types)

for example:

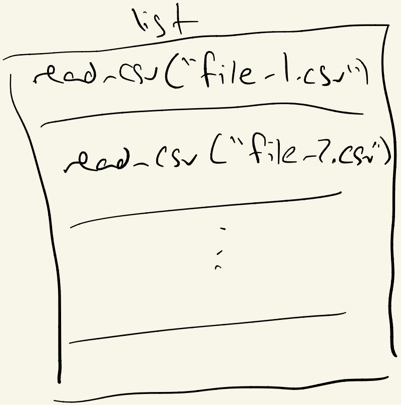
csv_paths <- vector of file paths ("file_1.csv", "file_2.csv", ...)

input_data <- map(csv_paths, read_csv) ⟶

list of dataframes

vector or list to apply function to

function to apply



list

read_csv("file-1.csv")

read_csv("file-2.csv")

⋮

```
input_data <- map(csv_paths , read_csv)
```

for loop alternative:

```
input_data <- list()
```

← empty list to fill in

```
for(i in 1:length(csv_paths)) {
```

or :

```
for(i in seq_along(csv_paths))
```

```
    input_data[[i]] <- read_csv(csv_paths[i])
```

i^{th} file path

read in i^{th} file path

Store the data frame in i^{th} entry of input_data list

```
}
```

Functions

Q: when do we need multiple arguments?

- when the user needs to specify multiple pieces of information

ex: correlation : relationship between two variables

```
correlation <- function(x, y)
```

variable 1

variable 2

ex: l_p norm : 1) vecter to take norm of

2) value for p

```
lp_norm <- function(x, p)
```

Q: when do I want an anonymous function?

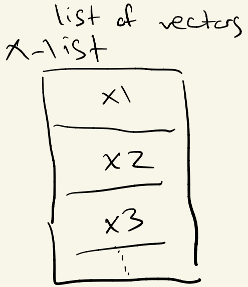
A: when at least 2 conditions are satisfied:

① The function is short (so we can write on one line)

② I only want it in one or 2 places in my code (functions which get used repeatedly should be named)

ex: if we want a special case of a general function (e.g., fixing some of the arguments)

lp-norm <- function(x, p) {...}   (calculates $\|x\|_p$)



want: $\|x_1\|_2$

$\|x_2\|_2$

$\|x_3\|_2$

$\vdots$

~~map(x-list, lp-norm)~~

fails b/c the function to map can only take 1 argument

map(x-list, function(x) lp-norm(x, 2))