R Programming For Natural Resource Professionals

Lecture 8

Tidy iterations: purrr

Discussion of assigned topics

Debugging: Macayla, Becca, Everett, Dan, Alicia

Tidyverse lifecycles: Ben, Dakota, Eric, Keenan

Additional RMarkdown features: Jordan, Andrew, AnaSara, Jeremy

<u>Prep to lead class for 5-7 minutes to teach your peers about the assigned topic.</u>

Potential discussion points:
1) Define it.
2) Explain it in an R context.
3) Explain it in a natural resources context.
4) Develop questions to ask the class.

Can email me a slide or two if you’d like (jhomola@uwsp.edu)

Learning objectives for this week

1. Understand the purpose of purrr
2. Understand the structure of purrr call
3. Perform basic operations using purrr

What is purr?

- Tidyverse’s functional programming tool kit
- Designed to replace most for loops

Key purrr functions

```
purrr::map(x, f)
```

Purpose: Apply the same function to multiple datasets (like a for loop).

- For each element of x, do f

Returns: A *list* of results

Key purrr functions

```
purrr::map(x, f)
```

x: input list or vector
f: the function to apply to the vector

purrr shorthand

Because purrr is part of the tidyverse, it accepts pipes.

```
dat %>%
  map(funct)
```

purrr shorthand

```
function(x){
   x + 10
}
```

=

```
~{.x+10}
```

Key purrr functions

```
purrr::map_*(…)
```

Purpose: Apply the same function to multiple datasets (like a for loop).

Returns: A datatype of your choosing (e.g., `map_dbl` or `map_df`)

Key purrr functions

```
purrr::map_*(…)
```

…: Note that map_* will pass along additional arguments

Ex: `dat %>% map_df(mean, na.rm = TRUE, trim = 0.5)`

Key purrr functions

```
purrr::map_at(…)
purrr::map_if(…)
```

Perform a map function at certain variables (map_at)
or based on a certain test (map_if)