

Functionals - Tutorial

Thiyanga Talagala

Required packages

```
library(tidyverse)
library(purrr)
```

Dataset

```
library(palmerpenguins)
data(penguins, package = 'palmerpenguins')
head(penguins)
```

```
# A tibble: 6 x 7
  species island bill_length_mm bill_depth_mm flipper_length_~ body_mass_g sex
  <fct>   <fct>         <dbl>         <dbl>         <int>         <int> <fct>
1 Adelie Torge~           39.1           18.7           181           3750 male
2 Adelie Torge~           39.5           17.4           186           3800 fema~
3 Adelie Torge~           40.3            18            195           3250 fema~
4 Adelie Torge~           NA             NA             NA             NA <NA>
5 Adelie Torge~           36.7           19.3           193           3450 fema~
6 Adelie Torge~           39.3           20.6           190           3650 male
```

Extract numeric columns only

```
penguins.numeric <- penguins %>% select(-c(species, island, sex))
```

Help

Explore the following command before attempt the question.

```
# Create a blank vector of size 2
output1 <- vector("double", 2)
output1
```

```
[1] 0 0
```

```
# Create a blank list of size 3
output2 <- vector("list", 3)
output2
```

```
[[1]]  
NULL
```

```
[[2]]  
NULL
```

```
[[3]]  
NULL
```

```
# find number of unique values is a vector  
a <- c(1, 1, 2, 3, 4, 5)  
n_distinct(a)
```

```
[1] 5
```

Useful map function

map and map_dbl

Questions

1. Write code using for loop to compute the mean of every column in `penguins.numeric`.

```
output <- vector("double", ncol(penguins.numeric)) # create a blank vector to store the results  
names(output) <- names(penguins.numeric)  
for (i in names(penguins.numeric)) {  
  output[i] <- mean(penguins.numeric[[i]], na.rm = TRUE)  
}  
output
```

bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
43.92193	17.15117	200.91520	4201.75439

2. Write code that uses one of the map functions to compute the mean of every column in `penguins.numeric`.

```
map_dbl(penguins.numeric, mean, na.rm=TRUE)
```

bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
43.92193	17.15117	200.91520	4201.75439

3. Write for loop to generate 10 random normals for each of $\mu = 10, 20, 30, 40, 50$.

```
n <- 10  
# values of the mean  
mu <- c(10, 20, 30, 40, 50)  
normals <- vector("list", length(mu))  
for (i in 1:length(normals)) {  
  normals[[i]] <- rnorm(n, mean = mu[i])  
}  
normals
```

```

[[1]]
[1] 11.110990 10.123880 9.858165 9.201265 10.824761 11.006102 8.991420
[8] 10.100771 10.437101 10.899154

[[2]]
[1] 22.14850 20.15527 20.54766 18.75422 21.58836 19.35676 20.59928 20.94260
[9] 18.87523 22.32676

[[3]]
[1] 28.83001 31.16702 28.96329 29.78391 29.10366 29.04028 30.03889 30.67883
[9] 30.86325 30.19973

[[4]]
[1] 40.89037 40.03090 40.17976 39.14751 39.20272 39.25134 39.76430 38.15361
[9] 40.73687 39.84855

[[5]]
[1] 50.62698 47.23871 50.96095 51.00715 50.40918 50.49221 50.68494 47.72974
[9] 49.33729 49.92795

```

4. Write code that uses one of the map functions to generate 10 random normals for each of $\mu = 10, 20, 30, 40, 50$.

```
map(c(10, 20, 30, 40, 50), ~rnorm(n = 10, mean = .))
```

```

[[1]]
[1] 10.324349 9.038665 10.853518 12.342677 10.616903 10.945412 10.878176
[8] 9.330330 10.962272 9.263849

[[2]]
[1] 21.29511 18.44714 20.20994 19.36594 21.76230 20.34217 20.27508 19.55660
[9] 19.33704 21.53497

[[3]]
[1] 29.22145 29.70930 31.70053 30.34683 31.39334 30.72510 30.12831 30.17591
[9] 29.56117 31.07849

[[4]]
[1] 39.89485 39.38908 40.54801 40.40953 37.23302 38.76237 40.84379 39.09273
[9] 41.12099 38.94538

[[5]]
[1] 49.59778 50.60236 50.08120 51.10960 52.21051 50.14250 51.03622 48.85161
[9] 49.62164 50.08567

```

5. Write code that used for loop to compute the number of unique values in each column of the `penguins.numeric` dataset.

```

penguins.numeric_uniq <- vector("double", ncol(penguins.numeric))
names(penguins.numeric_uniq) <- names(penguins.numeric)
for (i in names(penguins.numeric_uniq)) {
  penguins.numeric_uniq[i] <- n_distinct(penguins.numeric[[i]])
}
penguins.numeric_uniq

```

```
bill_length_mm    bill_depth_mm flipper_length_mm    body_mass_g
      165              81              56              95
```

6. Write code that uses one of the map functions to compute the number of unique values in each column of the `penguins.numeric` dataset.

```
map_dbl(penguins.numeric, n_distinct)
```

```
bill_length_mm    bill_depth_mm flipper_length_mm    body_mass_g
      165              81              56              95
```

This tutorial is based on R4DS.