Programming in R Workshop

Dr. Sumukh Deshpande

2020-10-20

Table of contents

Load packages

Load `purrr`, `tidyverse` and `dplyr` packages.

```
library(purrr)
library(tidyverse)
library(conflicted)
library(dplyr)
```

Load the Eukaryotes dataset - only have to run this once to get the data

```
eukaryotes <- read_tsv(
  file = "ftp://ftp.ncbi.nlm.nih.gov/genomes/GENOME_REPORTS/eukaryotes.txt",
  na = c("", "na", "-")
)

# Reformat dataset headers
names_new <- names(eukaryotes) |>
  str_replace_all("[#%()]", "") |>
  str_replace_all("[ /]", "_") |>
  str_to_lower()
```

```r
eukaryotes <- eukaryotes |>
  set_names(names_new)

# Save tibble
write_tsv(eukaryotes, "eukaryotes.tsv")
```

Load the saved dataset

```r
eukaryotes <- read_tsv("https://raw.githubusercontent.com/swuyts/purrr_tutorial/master/data/e
```

```
Rows: 11508 Columns: 19
-- Column specification --------------------------------------------------------
Delimiter: "\t"
chr  (10): organism_name, bioproject_accession, group, subgroup, assembly_ac...
dbl   (7): taxid, bioproject_id, size_mb, gc, scaffolds, genes, proteins
date  (2): release_date, modify_date

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

How many different organisms are there in our dataset?

```r
# Put your answer here
```

Subset dataframe by selecting variables for the analysis:

Using `n_distinct` to each variable of eukaryotes_subset

Nested Tibbles

Split the eukaryotes dataset according to groups defined in the group variable:

Split the "eukaryotes_nested" into 5 smaller dataframes.

Combine nested tibbles and map

Count number of rows for each sub data frames

Create a new column using `mutate()`

How many different organisms are there per group ?

There are two different ways:

Apply the function to our nested data:

We can define the functions on the fly:

pmap and walk2 functions

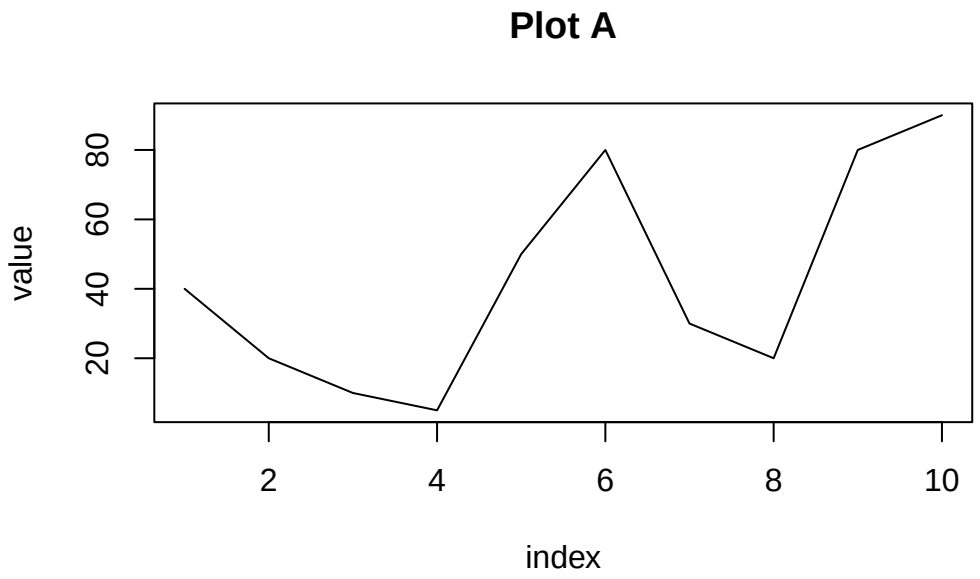
This about the following example for `pmap`, what will it do?:

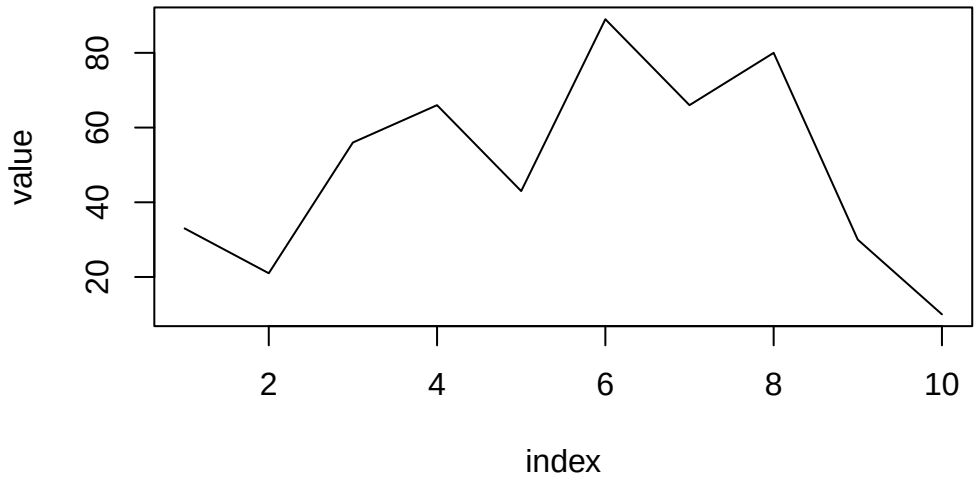
```
df <- data.frame(
  x = c("ATTTTACTGGGAGGGAA", "TATTTTTTAAAGGGCCC", "GCGCGCCCCAAATTATAGGC", "TGCCACATTTTATCCGCC
  pattern = c("A", "T", "G", "C"),
  replacement = c("a", "t", "g", "c"),
  stringsAsFactors = FALSE
  )

pmap(df, gsub)
```

```
[[1]]
[1] "aTTTTaCTGGGaGGGaa"

[[2]]
[1] "tAttttttAAAGGGCCC"

[[3]]
[1] "gCgCgCCCCAAATTATAggC"

[[4]]
[1] "TGccAcATTTTATccGcGcA"
```

Example for `walk2`:

```r
df1 <- data.frame(
  index = c(1,2,3,4,5,6,7,8,9,10,1,2,3,4,5,6,7,8,9,10),
  value = c(40,20,10,5,50,80,30,20,80,90,33,21,56,66,43,89,66,80,30,10),
  category = c("A","A","A","A","A","A","A","A","A","A","B","B","B","B","B","B","B","B","B","B")
)

df1 %>%
  split(.$category) %>%
  .[order(names(.))] %>%
  walk2(paste('Plot', names(.)),
        ~plot(value ~ index, data = .x, type = "l", main = .y))
```




Plot B
value
20
40
60
80
2
4
6
8
10
index