Foundations of Functional Programming with purrr_Problem solving with purrr

dizhen

5/19/2020

```r
library(repurrrsive)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(purrr)
```

Using purrr in your workflow

```r
# Load the data
data(gh_users)

# Check if data has names
names(gh_users)
```

```
## NULL
```

```r
# Map over name element of list
map(gh_users, ~.x[["name"]])
```

```
## [[1]]
## [1] "Gábor Csárdi"
##
## [[2]]
## [1] "Jennifer (Jenny) Bryan"
```

```
##
## [[3]]
## [1] "Jeff L."
##
## [[4]]
## [1] "Julia Silge"
##
## [[5]]
## [1] "Thomas J. Leeper"
##
## [[6]]
## [1] "Maëlle Salmon"
```

```r
# Name gh_users with the names of the users
gh_users_named <- gh_users %>%
    set_names(map_chr(gh_users, "name"))

# Check gh_repos structure
# str(gh_repos)

# Name gh_repos with the names of the repo owner
gh_repos_named <- gh_repos %>%
    map_chr(~ .[[1]]$owner$login) %>%
    set_names(gh_repos, .)
```

```r
# Determine who joined github first
map_chr(gh_users, ~.[['created_at']]) %>%
      set_names(map_chr(gh_users, "name")) %>%
    sort()
```

```
## Jennifer (Jenny) Bryan          Gábor Csárdi                Jeff L.
## "2011-02-03T22:37:41Z" "2011-03-09T17:29:25Z" "2012-03-24T18:16:43Z"
##       Thomas J. Leeper          Maëlle Salmon            Julia Silge
## "2013-02-07T21:07:00Z" "2014-08-05T08:10:04Z" "2015-05-19T02:51:23Z"
```

```r
# Determine user versus organization
map_lgl(gh_users, ~.[["type"]] == "User")
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE
```

```r
# Determine who has the most public repositories
map_int(gh_users, ~.[["public_repos"]]) %>%
    set_names(map_chr(gh_users, "name")) %>%
    sort()
```

```
##            Julia Silge          Maëlle Salmon           Gábor Csárdi
##                     26                     31                     52
##                Jeff L.       Thomas J. Leeper Jennifer (Jenny) Bryan
##                     67                     99                    168
```

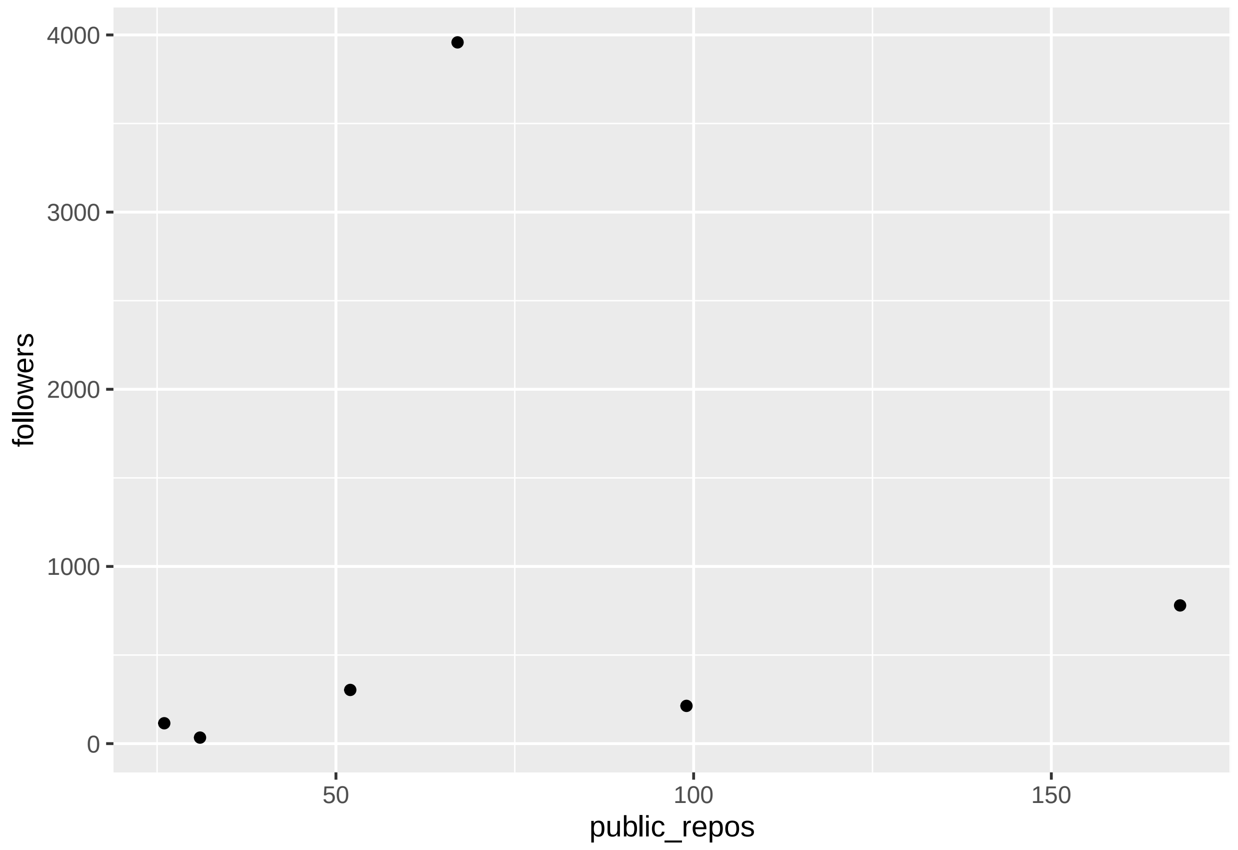
Even more complex problems

```r
# Map over gh_repos to generate numeric output
map(gh_repos,
    ~map_dbl(.x,
             ~.x[["size"]])) %>%
    # Grab the largest element
    map(~max(.x))
```

```
## [[1]]
## [1] 39461
##
## [[2]]
## [1] 96325
##
## [[3]]
## [1] 374812
##
## [[4]]
## [1] 24070
##
## [[5]]
## [1] 558176
##
## [[6]]
## [1] 76455
```

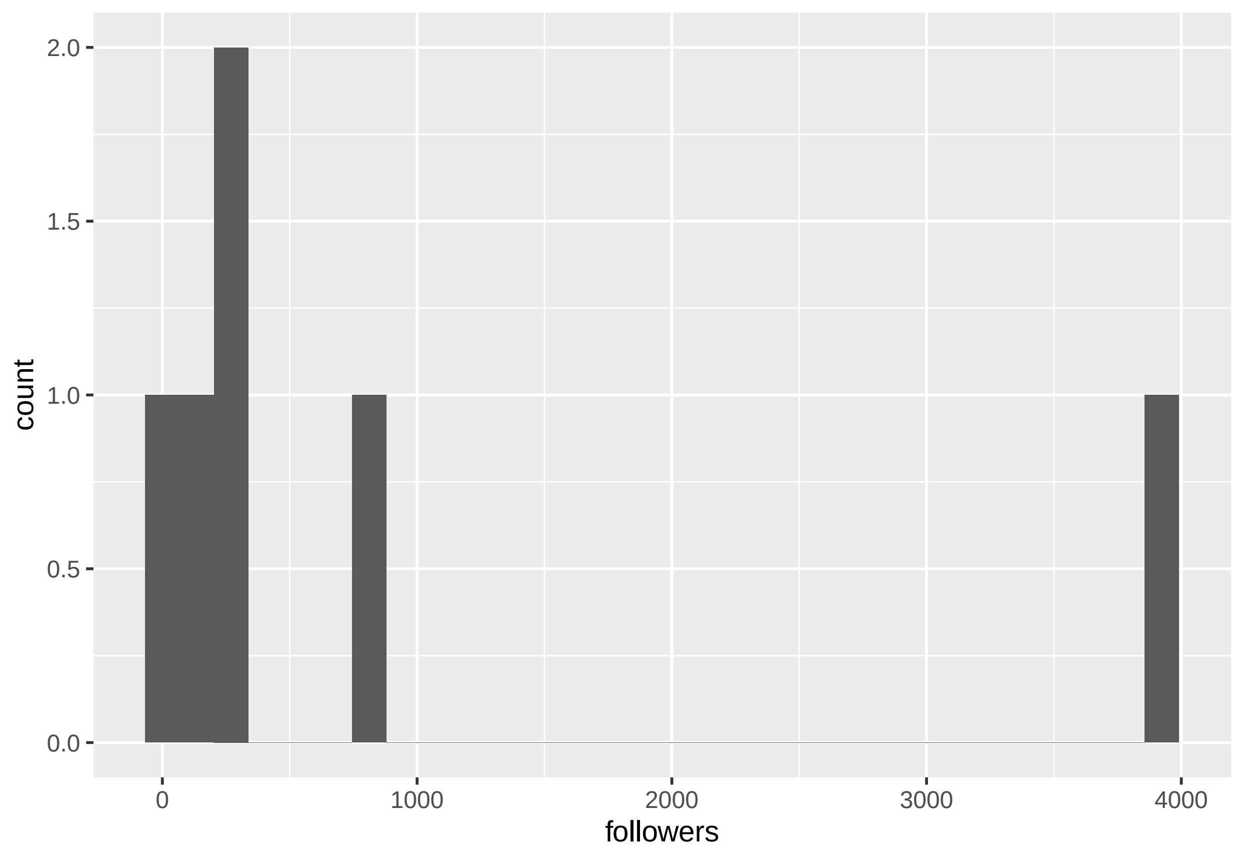
Graphs in purrr

```r
# method 1

fo <- map(gh_users, ~.x[["followers"]])
pu <- map(gh_users, ~.x[["public_repos"]])
gh_users_df <- data.frame(public_repos = unlist(pu), followers = unlist(fo))
rownames(gh_users_df) <- c(1:6)


# Scatter plot of public repos and followers
ggplot(data = gh_users_df,
       aes(x = public_repos, y = followers))+
    geom_point()
```

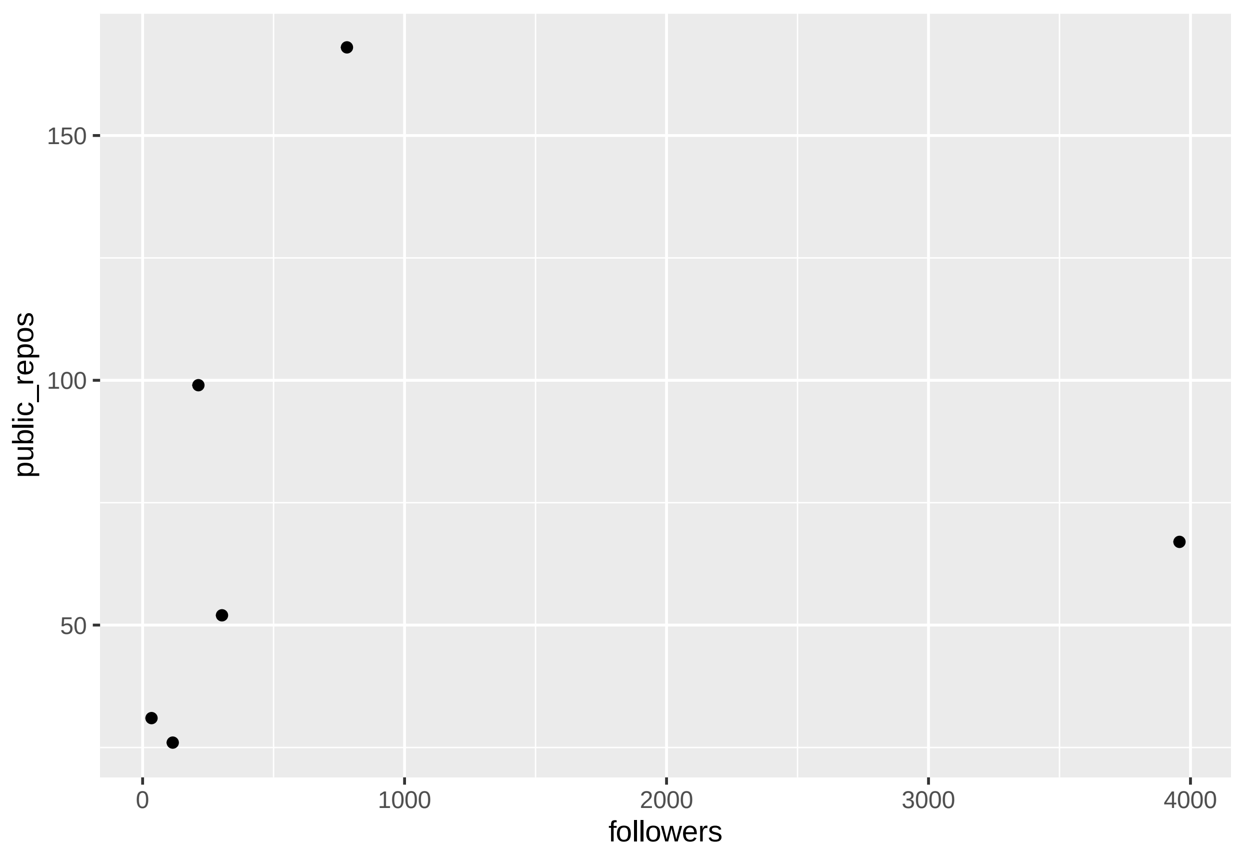


```r
# Histogram of followers
gh_users_df %>%
    ggplot(aes(x = followers))+
        geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# method 2
# Create a dataframe with four columns
map_df(gh_users, `[`,
       c("login", "name", "followers", "public_repos")) %>%
       # Plot followers by public_repos
    ggplot(.,
         aes(x = followers, y = public_repos)) +
        # Create scatter plots
        geom_point()
```



```
library(tidyr)

# Turn data into correct dataframe format
film_by_character <- tibble(filmtitle = map_chr(sw_films, "title")) %>%
    mutate(filmtitle, characters = map(sw_films, "characters")) %>%
    unnest()
```

```
## Warning: `cols` is now required.
## Please use `cols = c(characters)`
```

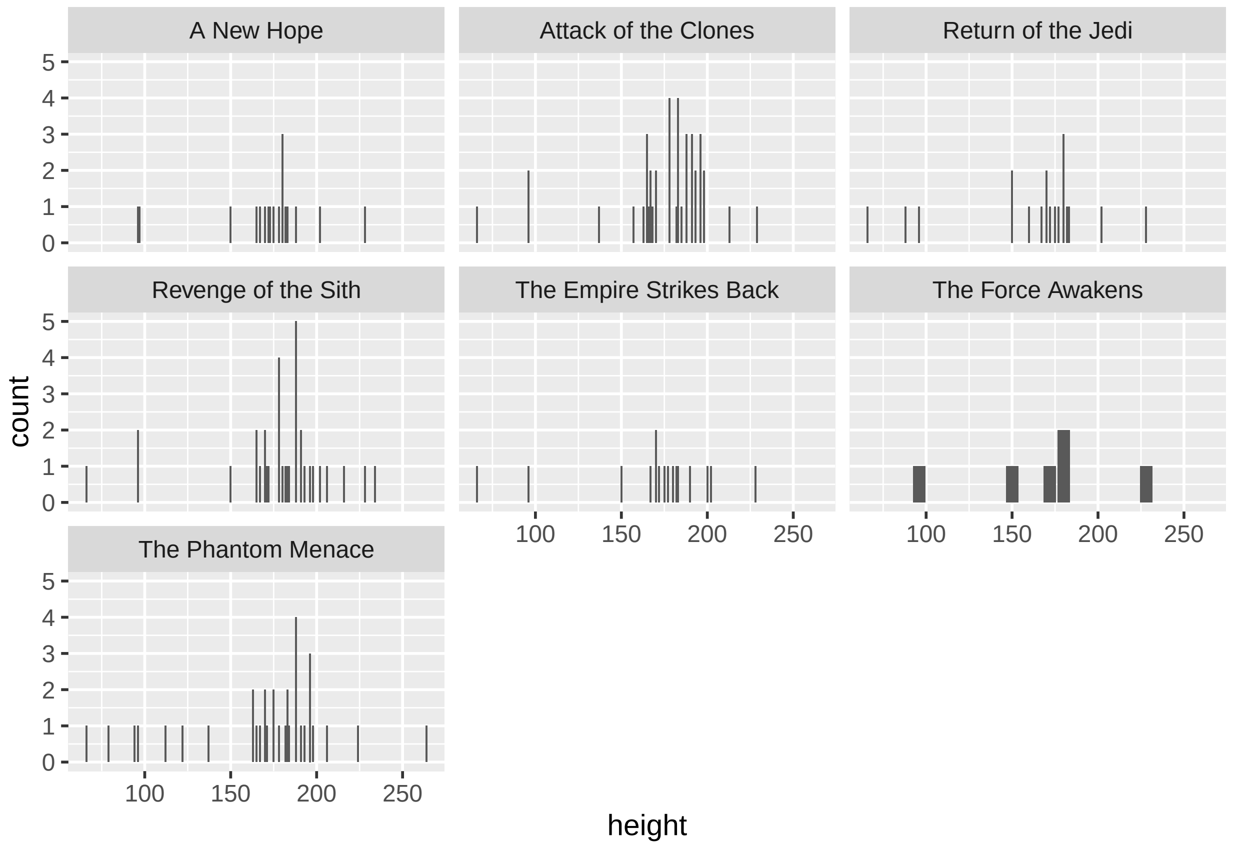
```
# Pull out elements from sw_people
sw_characters <- map_df(sw_people, `[`, c("height","mass","name","url"))

# Join our two new objects
character_data <- inner_join(film_by_character, sw_characters, by = c("characters" = "url")) %>%
    # Make sure the columns are numbers
    mutate(height = as.numeric(height), mass = as.numeric(mass))
```

```
## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion
```

```r
# Plot the heights, faceted by film title
ggplot(character_data, aes(x = height)) +
  geom_histogram(stat = "count") +
  facet_wrap(~ filmtitle)
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
## Warning: Removed 6 rows containing non-finite values (stat_count).
```


A New Hope
Attack of the Clones
Return of the Jedi
Revenge of the Sith
The Empire Strikes Back
The Force Awakens
The Phantom Menace
count
height