TidyVerse

Zachary Safir

4/10/2021

```
library(tidyverse)
library(knitr)
library(kableExtra)
```

Introduction

The tidyverse contains a collection of data science packages that work together in harmony to accomplish various goals. This vignette will demonstrate several ways to make full use of their combined capability.

The Data

For this demonstration, we will use a dataset that is included with dpylr itself. It contains data on the characters from the Starwars series. It contains various pieces of data that describe each character.

```
kable(head(starwars),format = "latex") %>%
    kable_styling(wraptable_width = "0pt")
```

```
starwars
```

```
## # A tibble: 87 x 14
##    name    height  mass hair_color  skin_color eye_color birth_year sex   gender
##    <chr>    <int> <dbl> <chr>       <chr>      <chr>          <dbl> <chr> <chr>
##  1 Luke S~    172    77 blond       fair       blue            19   male  mascu~
##  2 C-3PO      167    75 <NA>        gold       yellow         112   none  mascu~
##  3 R2-D2       96    32 <NA>        white, bl~ red             33   none  mascu~
##  4 Darth ~    202   136 none        white      yellow          41.9 male  mascu~
##  5 Leia O~    150    49 brown       light      brown           19   fema~ femin~
##  6 Owen L~    178   120 brown, grey light      blue            52   male  mascu~
##  7 Beru W~    165    75 brown       light      blue            47   fema~ femin~
##  8 R5-D4       97    32 <NA>        white, red red             NA   none  mascu~
##  9 Biggs ~    183    84 black       light      brown           24   male  mascu~
## 10 Obi-Wa~    182    77 auburn, wh~ fair       blue-gray       57   male  mascu~
## # ... with 77 more rows, and 5 more variables: homeworld <chr>, species <chr>,
## #   films <list>, vehicles <list>, starships <list>
```

Interestingly, some of the columns of data are full of lists. The column displayed below, shows which films a character appeared in.

```
head(starwars$films)
```

```
## [[1]]
## [1] "The Empire Strikes Back" "Revenge of the Sith"
## [3] "Return of the Jedi"      "A New Hope"
## [5] "The Force Awakens"
##
## [[2]]
## [1] "The Empire Strikes Back" "Attack of the Clones"
## [3] "The Phantom Menace"      "Revenge of the Sith"
## [5] "Return of the Jedi"      "A New Hope"
##
## [[3]]
## [1] "The Empire Strikes Back" "Attack of the Clones"
## [3] "The Phantom Menace"      "Revenge of the Sith"
## [5] "Return of the Jedi"      "A New Hope"
## [7] "The Force Awakens"
##
## [[4]]
## [1] "The Empire Strikes Back" "Revenge of the Sith"
## [3] "Return of the Jedi"      "A New Hope"
##
## [[5]]
## [1] "The Empire Strikes Back" "Revenge of the Sith"
## [3] "Return of the Jedi"      "A New Hope"
## [5] "The Force Awakens"
##
## [[6]]
## [1] "Attack of the Clones" "Revenge of the Sith"  "A New Hope"
```

The first thing to figure out is how to pick out only characcters that appear in certian films. In order to use filter from dpylr on a list, we need to use a purr function with it. As filter is expecting a logical value, we need to return something logical. Using map_lgl, we can accomplish this.

```
starwars %>%
filter(map_lgl(films,~ "Attack of the Clones" %in% .))
```

```
## # A tibble: 40 x 14
##    name    height  mass hair_color  skin_color eye_color birth_year sex   gender
##    <chr>    <int> <dbl> <chr>       <chr>      <chr>          <dbl> <chr> <chr>
##  1 C-3PO      167  75   <NA>        gold       yellow         112   none  mascu~
##  2 R2-D2       96  32   <NA>        white, bl~ red             33   none  mascu~
##  3 Owen L~    178 120   brown, grey light      blue            52   male  mascu~
##  4 Beru W~    165  75   brown       light      blue            47   fema~ femin~
##  5 Obi-Wa~    182  77   auburn, wh~ fair       blue-gray       57   male  mascu~
##  6 Anakin~    188  84   blond       fair       blue            41.9 male  mascu~
##  7 Yoda        66  17   white       green      brown          896   male  mascu~
##  8 Palpat~    170  75   grey        pale       yellow          82   male  mascu~
##  9 Boba F~    183  78.2 black       fair       brown           31.5 male  mascu~
## 10 Nute G~    191  90   none        mottled g~ red             NA   male  mascu~
## # ... with 30 more rows, and 5 more variables: homeworld <chr>, species <chr>,
## #   films <list>, vehicles <list>, starships <list>
```

In order to use filter on multiple values, we need to use the base R function "all".

```
starwars %>%
filter(map_lgl(films,~ all( c("Attack of the Clones","A New Hope") %in% .)))
```

```
## # A tibble: 5 x 14
##   name      height  mass hair_color   skin_color eye_color birth_year sex   gender
##   <chr>      <int> <dbl> <chr>        <chr>      <chr>          <dbl> <chr> <chr> 
## 1 C-3PO        167    75 <NA>         gold       yellow           112 none  mascu~
## 2 R2-D2         96    32 <NA>         white, bl~ red               33 none  mascu~
## 3 Owen La~     178   120 brown, grey light      blue              52 male  mascu~
## 4 Beru Wh~     165    75 brown        light      blue              47 fema~ femin~
## 5 Obi-Wan~     182    77 auburn, wh~ fair       blue-gray         57 male  mascu~
## # ... with 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

We can also use tidyr in order to flatten our lists full of data out. The resulting dataframe of this action is shown below.

```
starwars %>%
  select(name,films) %>%
  unnest(films)
```

```
## # A tibble: 173 x 2
##    name           films                  
##    <chr>          <chr>                  
##  1 Luke Skywalker The Empire Strikes Back
##  2 Luke Skywalker Revenge of the Sith    
##  3 Luke Skywalker Return of the Jedi     
##  4 Luke Skywalker A New Hope             
##  5 Luke Skywalker The Force Awakens      
##  6 C-3PO          The Empire Strikes Back
##  7 C-3PO          Attack of the Clones   
##  8 C-3PO          The Phantom Menace     
##  9 C-3PO          Revenge of the Sith    
## 10 C-3PO          Return of the Jedi     
## # ... with 163 more rows
```

With our data in a normal format, we can use the dpylr count function to discover which film is most common.

```
starwars %>%
  unnest(films) %>%
  count(films) %>%
  arrange(n)
```

```
## # A tibble: 7 x 2
##   films                       n
##   <chr>                   <int>
## 1 The Force Awakens          11
## 2 The Empire Strikes Back    16
## 3 A New Hope                 18
## 4 Return of the Jedi         20
## 5 Revenge of the Sith        34
## 6 The Phantom Menace         34
## 7 Attack of the Clones       40
```

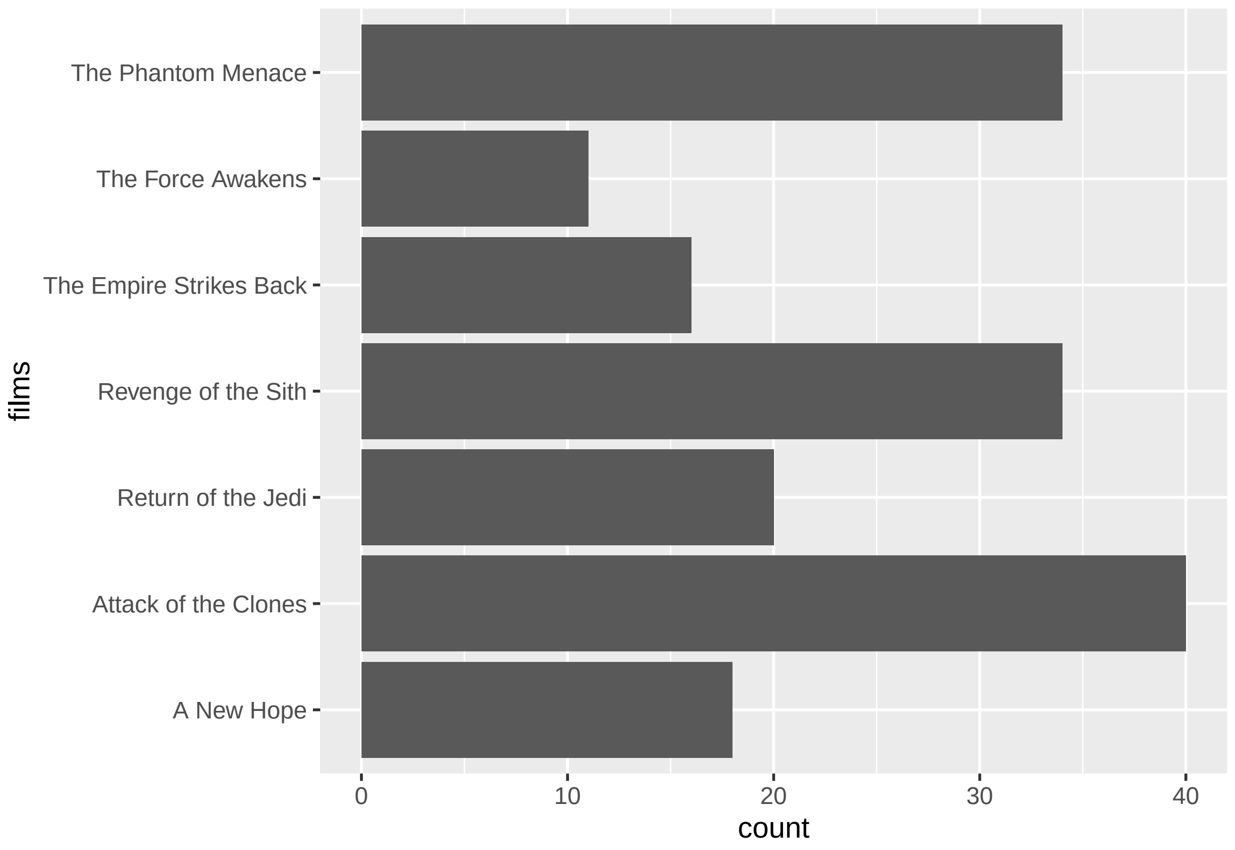
An other interesting function comes from forcats. In the previous example, we had a small number of a categories. However, quite often we will have A handful of common categories, and a whole bunch of other smaller groups. In such a case, we can use the forcats fct_lump to grab the most common categories, and lump the least most into a Other category.

```
starwars %>%
  filter(!is.na(homeworld)) %>%
  mutate(homeworld = fct_lump(homeworld, n = 3)) %>%
  count(homeworld) %>%
  arrange(n)
```

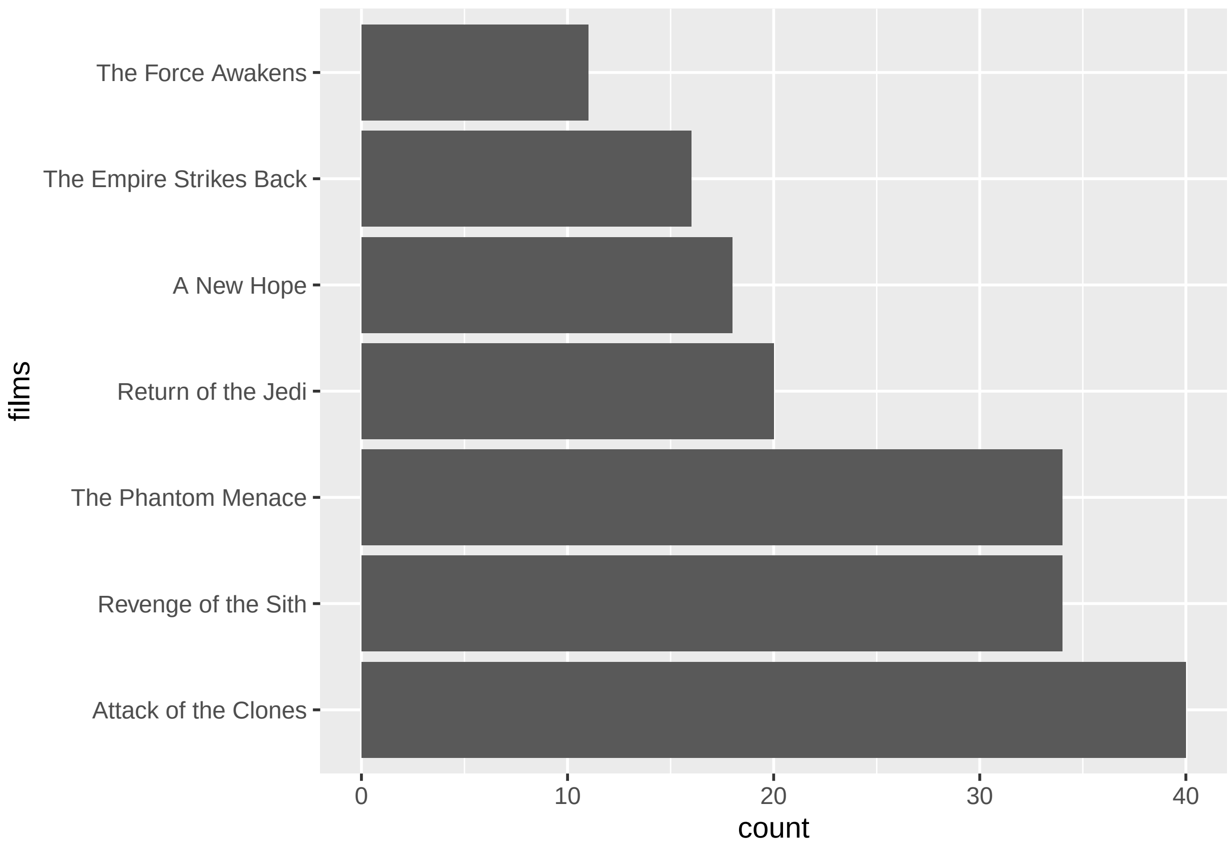
```
## # A tibble: 6 x 2
##   homeworld     n
##   <fct>     <int>
## 1 Alderaan      3
## 2 Coruscant     3
## 3 Kamino        3
## 4 Tatooine     10
## 5 Naboo        11
## 6 Other        47
```

Finally, we will demonstrate the fct_infreq function. In the first plot shown below, by default the plot is not ordered in any kind of way. However, by using fct_infreq in the second plot, we are able to reorder the values by their frequency in the data.

```
starwars %>%
  unnest(films) %>%
  ggplot(aes(films)) +
    geom_bar() +
    coord_flip()
```



```
starwars %>%
  unnest(films) %>%
  mutate(films = fct_infreq(films)) %>%
  ggplot(aes(films)) +
    geom_bar() +
    coord_flip()
```


The Force Awakens
The Empire Strikes Back
A New Hope
Return of the Jedi
The Phantom Menace
Revenge of the Sith
Attack of the Clones
films
0
10
20
30
40
count