Data Wrangling R with Answers

Prof. Eric A. Suess

Some of the code from Chapter 4, Section 1.

In this chapter dplyr is introduced. We will be using dplyr all year.

The main idea of data wrangling with dplyr are the 5 verbs.

**select()** # take a subset of columns

**filter()** # take a subset of rows

**mutate()** # add or modify existing columns

**arrange()** # sort the rows

**summarize()** # aggregate the data across rows

The dplyr package is part of the tidyverse. We will install and load the tidyverse.

library(mdsr)
library(tidyverse)

# Star Wars dataset

data("starwars")
glimpse(starwars)

## Observations: 87
## Variables: 13
## $ name <chr> "Luke Skywalker", "C-3PO", "R2-D2", "Darth Vader", "L…
## $ height <int> 172, 167, 96, 202, 150, 178, 165, 97, 183, 182, 188, …
## $ mass <dbl> 77.0, 75.0, 32.0, 136.0, 49.0, 120.0, 75.0, 32.0, 84.…
## $ hair\_color <chr> "blond", NA, NA, "none", "brown", "brown, grey", "bro…
## $ skin\_color <chr> "fair", "gold", "white, blue", "white", "light", "lig…
## $ eye\_color <chr> "blue", "yellow", "red", "yellow", "brown", "blue", "…
## $ birth\_year <dbl> 19.0, 112.0, 33.0, 41.9, 19.0, 52.0, 47.0, NA, 24.0, …
## $ gender <chr> "male", NA, NA, "male", "female", "male", "female", N…
## $ homeworld <chr> "Tatooine", "Tatooine", "Naboo", "Tatooine", "Alderaa…
## $ species <chr> "Human", "Droid", "Droid", "Human", "Human", "Human",…
## $ films <list> [<"Revenge of the Sith", "Return of the Jedi", "The …
## $ vehicles <list> [<"Snowspeeder", "Imperial Speeder Bike">, <>, <>, <…
## $ starships <list> [<"X-wing", "Imperial shuttle">, <>, <>, "TIE Advanc…

# select()

starwars %>% select(name, species)

## # A tibble: 87 x 2
## name species
## <chr> <chr>
## 1 Luke Skywalker Human
## 2 C-3PO Droid
## 3 R2-D2 Droid
## 4 Darth Vader Human
## 5 Leia Organa Human
## 6 Owen Lars Human
## 7 Beru Whitesun lars Human
## 8 R5-D4 Droid
## 9 Biggs Darklighter Human
## 10 Obi-Wan Kenobi Human
## # … with 77 more rows

# filter()

starwars %>%
 filter(species == "Droid")

## # A tibble: 5 x 13
## name height mass hair\_color skin\_color eye\_color birth\_year gender
## <chr> <int> <dbl> <chr> <chr> <chr> <dbl> <chr>
## 1 C-3PO 167 75 <NA> gold yellow 112 <NA>
## 2 R2-D2 96 32 <NA> white, bl… red 33 <NA>
## 3 R5-D4 97 32 <NA> white, red red NA <NA>
## 4 IG-88 200 140 none metal red 15 none
## 5 BB8 NA NA none none black NA none
## # … with 5 more variables: homeworld <chr>, species <chr>, films <list>,
## # vehicles <list>, starships <list>

# select()

starwars %>%
 select(name, ends\_with("color"))

## # A tibble: 87 x 4
## name hair\_color skin\_color eye\_color
## <chr> <chr> <chr> <chr>
## 1 Luke Skywalker blond fair blue
## 2 C-3PO <NA> gold yellow
## 3 R2-D2 <NA> white, blue red
## 4 Darth Vader none white yellow
## 5 Leia Organa brown light brown
## 6 Owen Lars brown, grey light blue
## 7 Beru Whitesun lars brown light blue
## 8 R5-D4 <NA> white, red red
## 9 Biggs Darklighter black light brown
## 10 Obi-Wan Kenobi auburn, white fair blue-gray
## # … with 77 more rows

# mutate()

starwars %>%
 mutate(name, bmi = mass / ((height / 100) ^ 2)) %>%
 select(name:mass, bmi)

## # A tibble: 87 x 4
## name height mass bmi
## <chr> <int> <dbl> <dbl>
## 1 Luke Skywalker 172 77 26.0
## 2 C-3PO 167 75 26.9
## 3 R2-D2 96 32 34.7
## 4 Darth Vader 202 136 33.3
## 5 Leia Organa 150 49 21.8
## 6 Owen Lars 178 120 37.9
## 7 Beru Whitesun lars 165 75 27.5
## 8 R5-D4 97 32 34.0
## 9 Biggs Darklighter 183 84 25.1
## 10 Obi-Wan Kenobi 182 77 23.2
## # … with 77 more rows

# arrange()

starwars %>%
 arrange(desc(mass))

## # A tibble: 87 x 13
## name height mass hair\_color skin\_color eye\_color birth\_year gender
## <chr> <int> <dbl> <chr> <chr> <chr> <dbl> <chr>
## 1 Jabb… 175 1358 <NA> green-tan… orange 600 herma…
## 2 Grie… 216 159 none brown, wh… green, y… NA male
## 3 IG-88 200 140 none metal red 15 none
## 4 Dart… 202 136 none white yellow 41.9 male
## 5 Tarf… 234 136 brown brown blue NA male
## 6 Owen… 178 120 brown, gr… light blue 52 male
## 7 Bossk 190 113 none green red 53 male
## 8 Chew… 228 112 brown unknown blue 200 male
## 9 Jek … 180 110 brown fair blue NA male
## 10 Dext… 198 102 none brown yellow NA male
## # … with 77 more rows, and 5 more variables: homeworld <chr>,
## # species <chr>, films <list>, vehicles <list>, starships <list>

# summarize()

starwars %>%
 group\_by(species) %>%
 summarise(
 n = n(),
 mass = mean(mass, na.rm = TRUE)
 ) %>%
 filter(n > 1)

## # A tibble: 9 x 3
## species n mass
## <chr> <int> <dbl>
## 1 Droid 5 69.8
## 2 Gungan 3 74
## 3 Human 35 82.8
## 4 Kaminoan 2 88
## 5 Mirialan 2 53.1
## 6 Twi'lek 2 55
## 7 Wookiee 2 124
## 8 Zabrak 2 80
## 9 <NA> 5 48

# Questions

Develop the R code to answer the following questions.

1. How many films are in the dataset?

starwars %>% select(films) %>%
 unlist() %>%
 unique()

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

1. Are there more Droids or humans in the Star Wars movies? There are 5 Droids and 35 Humans. So more Humans.

starwars %>% select(species) %>%
 filter(species=="Droid" | species=="Human") %>%
 group\_by(species) %>%
 summarize(n=n())

## # A tibble: 2 x 2
## species n
## <chr> <int>
## 1 Droid 5
## 2 Human 35

1. Which of the Star Wars movies was Luke Skywalker in?

starwars %>% filter(name=="Luke Skywalker") %>%
 select(films) %>%
 unlist()

## films1 films2
## "Revenge of the Sith" "Return of the Jedi"
## films3 films4
## "The Empire Strikes Back" "A New Hope"
## films5
## "The Force Awakens"

1. Pose a question and answer it by wrangling the starwars dataset.

What was the distribution of hights? What was the distribution of hights by species?

starwars %>% ggplot(aes(x=height)) +
 geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 6 rows containing non-finite values (stat\_bin).



starwars %>% ggplot(aes(x=height, color=gender)) +
 geom\_histogram(aes(y=..density..))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 6 rows containing non-finite values (stat\_bin).



starwars %>% ggplot(aes(x=height, color=gender)) +
 geom\_density(aes(y=..density..))

## Warning: Removed 6 rows containing non-finite values (stat\_density).

## Warning: Groups with fewer than two data points have been dropped.

## Warning: Groups with fewer than two data points have been dropped.



# Presidential examples

Try out the code in Chapter 4 Section 1 using the presidential data set.

presidential

## # A tibble: 11 x 4
## name start end party
## <chr> <date> <date> <chr>
## 1 Eisenhower 1953-01-20 1961-01-20 Republican
## 2 Kennedy 1961-01-20 1963-11-22 Democratic
## 3 Johnson 1963-11-22 1969-01-20 Democratic
## 4 Nixon 1969-01-20 1974-08-09 Republican
## 5 Ford 1974-08-09 1977-01-20 Republican
## 6 Carter 1977-01-20 1981-01-20 Democratic
## 7 Reagan 1981-01-20 1989-01-20 Republican
## 8 Bush 1989-01-20 1993-01-20 Republican
## 9 Clinton 1993-01-20 2001-01-20 Democratic
## 10 Bush 2001-01-20 2009-01-20 Republican
## 11 Obama 2009-01-20 2017-01-20 Democratic

## Star Wars API and R package

More Star Wars stuff you might find interesting.

* Check out the [Star Wars](https://www.starwars.com/) website.
* Check out the Star Wars API [sawpi](https://swapi.co/).
* And check out the R package [rwars](https://github.com/Ironholds/rwars).

## rwars package

This is a package that connects to the [sawpi](https://swapi.co/) to pull data from the API.

If the package does not install from CRAN you can isntall it from github.

library(devtools)
install\_github("ironholds/rwars")

library(rwars)

planet\_schema <- get\_planet\_schema()
names(planet\_schema)

## [1] "properties" "$schema" "type" "required" "description"
## [6] "title"

## rwars package

Get an individual starship - an X-wing.

Hopefully it won’t time out and will actually bring the data back.

x\_wing <- get\_starship(12)
x\_wing

## $name
## [1] "X-wing"
##
## $model
## [1] "T-65 X-wing"
##
## $manufacturer
## [1] "Incom Corporation"
##
## $cost\_in\_credits
## [1] "149999"
##
## $length
## [1] "12.5"
##
## $max\_atmosphering\_speed
## [1] "1050"
##
## $crew
## [1] "1"
##
## $passengers
## [1] "0"
##
## $cargo\_capacity
## [1] "110"
##
## $consumables
## [1] "1 week"
##
## $hyperdrive\_rating
## [1] "1.0"
##
## $MGLT
## [1] "100"
##
## $starship\_class
## [1] "Starfighter"
##
## $pilots
## $pilots[[1]]
## [1] "https://swapi.co/api/people/1/"
##
## $pilots[[2]]
## [1] "https://swapi.co/api/people/9/"
##
## $pilots[[3]]
## [1] "https://swapi.co/api/people/18/"
##
## $pilots[[4]]
## [1] "https://swapi.co/api/people/19/"
##
##
## $films
## $films[[1]]
## [1] "https://swapi.co/api/films/2/"
##
## $films[[2]]
## [1] "https://swapi.co/api/films/3/"
##
## $films[[3]]
## [1] "https://swapi.co/api/films/1/"
##
##
## $created
## [1] "2014-12-12T11:19:05.340000Z"
##
## $edited
## [1] "2014-12-22T17:35:44.491233Z"
##
## $url
## [1] "https://swapi.co/api/starships/12/"

## Alternative API that can be accessed via an R package

The [compstatr](https://cran.r-project.org/web/packages/compstatr/vignettes/compstatr.html) R package gives direct access to the St. Louis Metropolitan Police Department’s [website](https://www.slmpd.org/Crimereports.shtml).

library(compstatr)

cs\_last\_update()

## [1] "August 2019"

i <- cs\_create\_index()

aug19 <- cs\_get\_data(year = 2019, month = "August", index = i)
aug19

## # A tibble: 4,624 x 20
## complaint coded\_month date\_occur flag\_crime flag\_unfounded
## <chr> <chr> <chr> <chr> <chr>
## 1 19-039099 2019-08 01/01/190… Y <NA>
## 2 19-039141 2019-08 01/01/201… Y <NA>
## 3 19-037923 2019-08 01/01/201… Y <NA>
## 4 19-040019 2019-08 01/01/201… Y <NA>
## 5 19-039212 2019-08 01/01/201… Y <NA>
## 6 19-037912 2019-08 01/01/201… Y <NA>
## 7 19-035473 2019-08 01/15/201… <NA> <NA>
## 8 19-005861 2019-08 02/07/201… <NA> <NA>
## 9 19-038425 2019-08 02/11/201… Y <NA>
## 10 19-033762 2019-08 02/12/201… Y <NA>
## # … with 4,614 more rows, and 15 more variables:
## # flag\_administrative <chr>, count <chr>, flag\_cleanup <chr>,
## # crime <chr>, district <chr>, description <chr>, ileads\_address <chr>,
## # ileads\_street <chr>, neighborhood <chr>, location\_name <chr>,
## # location\_comment <chr>, cad\_address <chr>, cad\_street <chr>,
## # x\_coord <chr>, y\_coord <chr>

The [ukpolice](https://github.com/evanodell/ukpolice) R package to download data from UK Police public data API.

library(ukpolice)
library(ggplot2)
library(dplyr)

tv\_ss <- ukc\_stop\_search\_force("thames-valley", date = "2018-12")

tv\_ss2 <- tv\_ss %>%
 filter(!is.na(officer\_defined\_ethnicity) & outcome != "" ) %>%
 group\_by(officer\_defined\_ethnicity, outcome) %>%
 summarise(n = n()) %>%
 mutate(perc = n/sum(n))

p1 <- ggplot(tv\_ss2, aes(x = outcome, y = perc,
 group = outcome, fill = outcome)) +
 geom\_col(position = "dodge") +
 scale\_y\_continuous(labels = scales::percent,
 breaks = seq(0.25, 0.8, by = 0.25)) +
 scale\_x\_discrete(labels = scales::wrap\_format(15)) +
 theme(legend.position = "none", axis.text.x = element\_text(size = 8)) +
 labs(x = "Outcome",
 y = "Percentage of stop and searches resulting in outcome",
 title = "Stop and Search Outcomes by Police-Reported Ethnicity",
 subtitle = "Thames Valley Police Department, December 2018",
 caption = "(c) Evan Odell | 2019 | CC-BY-SA") +
 facet\_wrap(~officer\_defined\_ethnicity)

p1



Alternatively you could use the other [ukpolice](https://github.com/njtierney/ukpolice) R package that is available through github.

And here is a nice blog post about crime in SF [Using R for Crime Analysis](https://wetlands.io/maps/Crime-Analysis-Using-R.html.)