first look at the dataset

* library data
* automation

## Load Library Package

“Use the [Tidyverse](https://www.tidyverse.org/), Luke” – O-W.Kenobi

library(tidyverse)

## Registered S3 methods overwritten by 'ggplot2':  
## method from   
## [.quosures rlang  
## c.quosures rlang  
## print.quosures rlang

## -- Attaching packages -------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.1 v purrr 0.3.2  
## v tibble 2.1.1 v dplyr 0.8.1  
## v tidyr 0.8.3 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ----------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(skimr)

##   
## Attaching package: 'skimr'

## The following object is masked from 'package:stats':  
##   
## filter

## Get Data

Crossref data used from the **Setup** to the [LC OpenRefine Workshop](https://librarycarpentry.org/lc-open-refine/setup.html)

crossref\_data <- read\_csv("https://raw.githubusercontent.com/LibraryCarpentry/lc-open-refine/gh-pages/data/doaj-article-sample.csv",   
 col\_types = cols(Date = col\_date(format = "%m/%d/%Y")))

crossref\_data

## # A tibble: 1,001 x 11  
## Title Authors DOI URL Date Language Subjects ISSNs Publisher  
## <chr> <chr> <chr> <chr> <date> <chr> <chr> <chr> <chr>   
## 1 The ~ Flavia~ 10.3~ http~ 2015-01-11 English Fisher ~ 1099~ MDPI AG   
## 2 Afla~ Naveed~ 10.3~ http~ 2015-01-11 English aflatox~ 2077~ MDPI AG   
## 3 Meta~ Rafael~ 10.3~ http~ 2015-01-11 English PKS|NRP~ 1422~ MDPI AG   
## 4 Synt~ Fabriz~ 10.3~ http~ 2015-01-11 EN lanthan~ 2304~ MDPI AG   
## 5 Perf~ Magali~ 10.3~ http~ 2015-01-11 EN snow mo~ 2306~ MDPI AG   
## 6 Dihy~ Xiaoxi~ 10.3~ http~ 2015-01-11 English Malus c~ 1420~ MDPI AG   
## 7 Ioni~ Anton ~ 10.3~ http~ 2015-01-11 English ionic l~ 2073~ MDPI AG   
## 8 Char~ Weihon~ 10.3~ http~ 2015-01-11 English Coryneb~ 1422~ MDPI AG   
## 9 Quat~ Tosiak~ 10.3~ http~ 2015-01-11 English infinit~ 2073~ MDPI AG   
## 10 Imag~ Christ~ 10.3~ http~ 2015-01-11 <NA> hepatoc~ 2075~ MDPI AG   
## # ... with 991 more rows, and 2 more variables: Citation <chr>,  
## # Licence <chr>

## skimr

skim(crossref\_data)

## Skim summary statistics  
## n obs: 1001   
## n variables: 11   
##   
## -- Variable type:character ----------------------------------  
## variable missing complete n min max empty n\_unique  
## Authors 0 1001 1001 7 291 0 883  
## Citation 0 1001 1001 39 104 0 1000  
## DOI 23 978 1001 16 29 0 977  
## ISSNs 0 1001 1001 9 19 0 51  
## Language 15 986 1001 2 7 0 4  
## Licence 6 995 1001 5 11 0 3  
## Publisher 0 1001 1001 7 47 0 6  
## Subjects 0 1001 1001 17 337 0 988  
## Title 0 1001 1001 18 318 0 1000  
## URL 0 1001 1001 57 57 0 1000  
##   
## -- Variable type:Date ---------------------------------------  
## variable missing complete n min max median n\_unique  
## Date 0 1001 1001 2015-01-01 2015-01-12 2015-01-07 12

## Facetting

Generate a quick table of the languages represented in the dataframe. Looks like English (spelled two different ways), FRench and ?Spanish? (represented by ES).

crossref\_data %>%   
 count(Language)

## # A tibble: 5 x 2  
## Language n  
## <chr> <int>  
## 1 <NA> 15  
## 2 EN 871  
## 3 English 107  
## 4 ES 7  
## 5 FR 1

This time, facet on the governing license

crossref\_data %>%   
 count(Licence)

## # A tibble: 4 x 2  
## Licence n  
## <chr> <int>  
## 1 <NA> 6  
## 2 CC BY 954  
## 3 CC BY-NC 11  
## 4 CC BY-NC-ND 30

Facet on the publisher

crossref\_data %>%   
 count(Publisher)

## # A tibble: 6 x 2  
## Publisher n  
## <chr> <int>  
## 1 Akshantala Enterprises 13  
## 2 Aurel Vlaicu University Editing House 17  
## 3 Consejo Superior de Investigaciones Científicas 11  
## 4 International Union of Crystallography 858  
## 5 MDPI AG 96  
## 6 Society of Pharmaceutical Technocrats 6

Facet by authors, and sort by the most prolific. This field appears to be a multi-valued field that is pipe | separated. How do we count and visualize how many articles have multiple authors?

crossref\_data %>%   
 count(Authors) %>%   
 arrange(-n)

## # A tibble: 883 x 2  
## Authors n  
## <chr> <int>  
## 1 Yoshinobu Ishikawa 7  
## 2 Gihaeng Kang|Jineun Kim|Hyunjin Park|Tae Ho Kim 6  
## 3 M. P. Savithri|M. Suresh|R. Raghunathan|R. Raja|A. SubbiahPandi 6  
## 4 Gamal A. El-Hiti|Keith Smith|Amany S. Hegazy|Saud A. Alanazi|Bens~ 5  
## 5 Gihaeng Kang|Jineun Kim|Eunjin Kwon|Tae Ho Kim 5  
## 6 Hea-Chung Joo|Ki-Min Park|Uk Lee 5  
## 7 Dohyun Moon|Jong-Ha Choi 4  
## 8 M. S. Krishnamurthy|Noor Shahina Begum 4  
## 9 Rajamani Raja|Subramani Kandhasamy|Paramasivam T. Perumal|A. Subb~ 4  
## 10 Augusto Rivera|Jicli José Rojas|Jaime Ríos-Motta|Michael Bolte 3  
## # ... with 873 more rows

Exploring some methods to generate a cound of the pipe delimeter. stringr::str\_count() appears to be a great way to calculate this.

dim(as\_tibble(str\_split(crossref\_data$Authors[5], "\\|", simplify = TRUE)))[2]

## Warning: `as\_tibble.matrix()` requires a matrix with column names or a `.name\_repair` argument. Using compatibility `.name\_repair`.  
## This warning is displayed once per session.

## [1] 3

str\_count(crossref\_data$Authors[1:5], "\\|")

## [1] 1 1 2 3 2

## Transform Data

Use dplyr::mutate to generate a new field that calculates how many authors each observation contains.

crossref\_data %>%   
 select(Authors) %>%   
 mutate(multi\_authorship = str\_count(Authors, "\\|") + 1) %>%   
 select(Authors, multi\_authorship)

## # A tibble: 1,001 x 2  
## Authors multi\_authorship  
## <chr> <dbl>  
## 1 Flavia Pennini|Angelo Plastino 2  
## 2 Naveed Aslam|Peter C. Wynn 2  
## 3 Rafael R. C. Cuadrat|Juliano C. Cury|Alberto M. R. Dáv~ 3  
## 4 Fabrizio Ortu|Hao Zhu|Marie-Emmanuelle Boulon|David P.~ 4  
## 5 Magali Troin|Richard Arsenault|François Brissette 3  
## 6 Xiaoxiao Qin|Yun Feng Xing|Zhiqin Zhou|Yuncong Yao 4  
## 7 Anton Axelsson|Linda Ta|Henrik Sundén 3  
## 8 Weihong Min|Huiying Li|Hongmei Li|Chunlei Liu|Jingshen~ 5  
## 9 Tosiaki Kori|Yuto Imai 2  
## 10 Christina Schraml|Sascha Kaufmann|Hansjoerg Rempp|Rola~ 7  
## # ... with 991 more rows

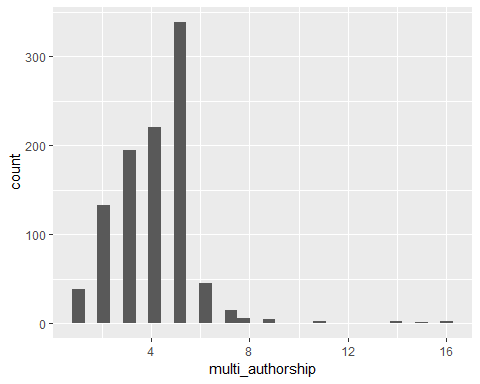
## Visualize

### Authors

Generate a histogram distribution of the multiple authorship variable.

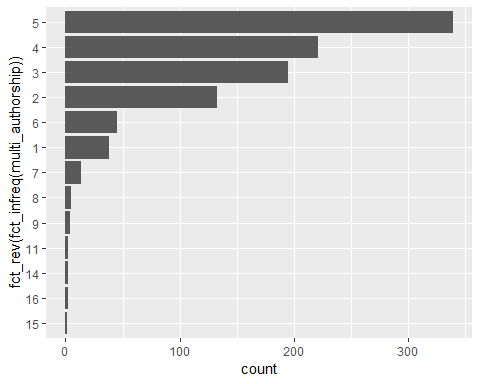
crossref\_data %>%   
 select(Authors) %>%   
 mutate(multi\_authorship = str\_count(Authors, "\\|") + 1) %>%   
 select(multi\_authorship, Authors) %>%   
 ggplot() +  
 aes(multi\_authorship) +  
 geom\_histogram()

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



This time generate as a bargraph and sort by the most frequent representation. Articles with five authors is the most frequent representation in the dataset.

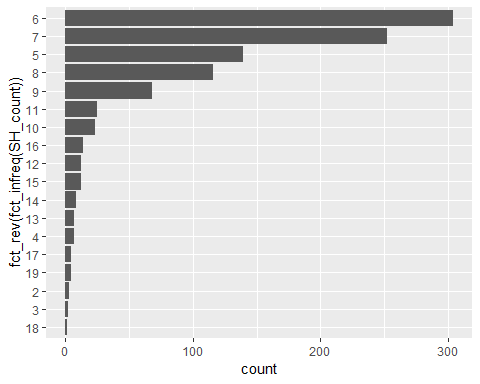
auth\_count <- crossref\_data %>%   
 select(Authors) %>%   
 mutate(multi\_authorship = str\_count(Authors, "\\|") + 1) %>%   
 mutate(multi\_authorship = as.character(multi\_authorship)) %>%   
 select(multi\_authorship, Authors)  
  
ggplot(auth\_count) +  
 aes(fct\_rev(fct\_infreq(multi\_authorship))) +  
 geom\_bar() +  
 coord\_flip()



### Explore Subject Headings

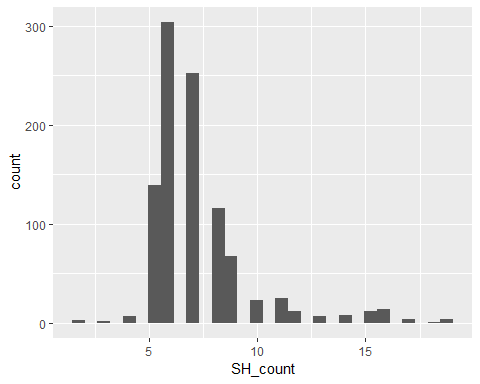
Visualize the frequency of multiple subject headings, just as with authors (A bargraph and a histogram)

crossref\_data %>%   
 mutate(SH\_count = str\_count(Subjects, "\\|") + 1) %>%   
 mutate(SH\_count = as.character(SH\_count)) %>%   
 ggplot() +  
 aes(fct\_rev(fct\_infreq(SH\_count))) +  
 geom\_bar() +  
 coord\_flip()



crossref\_data %>%   
 mutate(SH\_count = str\_count(Subjects, "\\|") + 1) %>%   
 ggplot() +  
 aes(SH\_count) +  
 geom\_histogram()

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



## Data Transformations

Using dplyr, mutate a new variable and transform the data so that ‘EN’ and ‘English’ are the same. Transform ‘ES’ to “Spanish”, and ‘FR’ to “French”.

dplyr::case\_when() is one specialized way to perform an if\_else transformation.

crossref\_data %>%   
 count(Language)

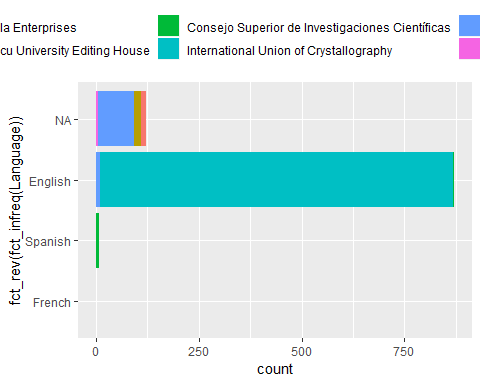
## # A tibble: 5 x 2  
## Language n  
## <chr> <int>  
## 1 <NA> 15  
## 2 EN 871  
## 3 English 107  
## 4 ES 7  
## 5 FR 1

crossref\_data <- crossref\_data %>%   
 mutate(Language = case\_when(  
 Language == "EN" ~ "English",  
 Language == "ES" ~ "Spanish",  
 Language == "FR" ~ "French"  
 ))

### Visualize the Languages.

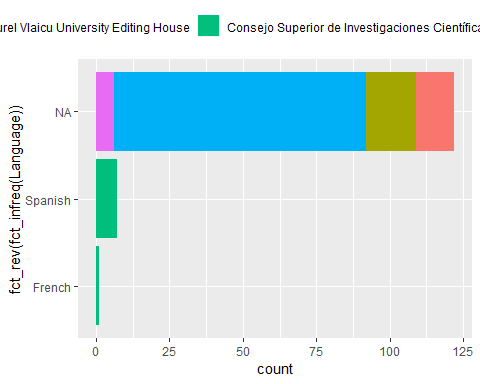
Stacked Bargraph shows frequency by Language. Each stack of a bar distinguishes the publishers. English Language is huge and somewhat over-powers the reset of the graph. Make a second graph (below) to drill down on the lesser represented languages.

crossref\_data %>%   
 ggplot() +  
 aes(fct\_rev(fct\_infreq(Language)), fill = Publisher) +  
 geom\_bar() +  
 coord\_flip() +  
 theme(legend.position="top")



Filter the data to show only the “NA”, “French”, and “Spanish”.

crossref\_data %>%   
 filter(is.na(Language) | Language == "French" | Language == "Spanish") %>%   
 ggplot() +  
 aes(fct\_rev(fct\_infreq(Language)), fill = Publisher) +  
 geom\_bar() +  
 coord\_flip() +  
 theme(legend.position="top")



## Time Series

crossref\_data %>%   
 count(Date) %>%   
 ggplot(aes(Date, n)) +  
 geom\_point() +  
 geom\_line() +  
 ggtitle("Publishing Date Frequency", subtitle = "One Week in January, 2015")

