first look at the dataset

- library data
- automation

Load Library Package

“Use the Tidyverse, 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

```
crossref_data <- read_csv("https://raw.githubusercontent.com/LibraryCarpentry/lc-open-refine/gh-pages/da
    col_types = cols(Date = col_date(format = "%m/%d/%Y")))
```

Take a quick look at the data

```r
glimpse(crossref_data)
```

```
## Observations: 1,001
## Variables: 11
## $ Title     <chr> "The Fisher Thermodynamics of Quasi-Probabilities", ...
## $ Authors   <chr> "Flavia Pennini|Angelo Plastino", "Naveed Aslam|Pete...
## $ DOI       <chr> "10.3390/e17127853", "10.3390/agriculture5041172", "...
## $ URL       <chr> "https://doaj.org/article/b75e8d5cca3f46cbbd63e91be5...
## $ Date      <date> 2015-01-11, 2015-01-11, 2015-01-11, 2015-01-11, 201...
## $ Language  <chr> "English", "English", "English", "EN", "EN", "Englis...
## $ Subjects  <chr> "Fisher information|quasi-probabilities|complementar...
## $ ISSNs     <chr> "1099-4300", "2077-0472", "1422-0067", "2304-6740", ...
## $ Publisher <chr> "MDPI AG", "MDPI AG", "MDPI AG", "MDPI AG", "MDPI AG...
## $ Citation  <chr> "Entropy, Vol 17, Iss 12, Pp 7848-7858 (2015)", "Agr...
## $ Licence   <chr> "CC BY", "CC BY", "CC BY", "CC BY", "CC BY", "CC BY"...
```

```r
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

Skimr is a easy way to have a quick look at the variables in the data frame. In this case the data are mostly character string data. With numeric data skimr will produce a thumbnail histogram (sparkline)

```r
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
```

Faceting

Two methods to generate a quick table of the languages represented in the dataframe: `count()` and `forcats::fct_count`. Since these data are primarily character, it's helpful to learn about factor data and the forcats package. These two tables are the same. It looks like the data are published in English (spelled two different ways), FRench and Spanish.

```
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
```

```
fct_count(crossref_data$Language, sort = TRUE)
```

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

This time, facet on the governing license. All but six articles are covered by a createive commons 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. Sort in descending order.

```
crossref_data %>%
  count(Publisher, sort = TRUE)
```

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

```
## # 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
```

The above table is not very useful (unless tracking publishing teams that are always expressed identically.) Let's exploring some methods to generate a count of the pipe character separating each author in a single author field. The `stringr::str_count()` function is a great way to calculate the number of delimiters in each author field.

Note that counting a pipe character `|` requires using a Regular Expression, or regex. Anyone manipulating string characters with computers will be far more capable after spending some time learning about regular expressions. In this case the we're looking for a pipe character `|`. The special trick, here, in understanding regex is to know that a pipe character has special meaning. Therefore we have to escape, or make it know that we want the literal pipe character and not the special meaning pipe character. To escape a character in regex one uses a backslash `\`. But the weird part is that, in R, one has to escape the the escape character: `\\|` means look for a literal `|`.

Below we count the number of pipe characters in each row of the Author field. Using the `head` function we only display the first six values (rows) in the Author column.

```
str_count(crossref_data$Authors, "\\|") %>% head()
```

```
## [1] 1 1 2 3 2 3
```

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)
```

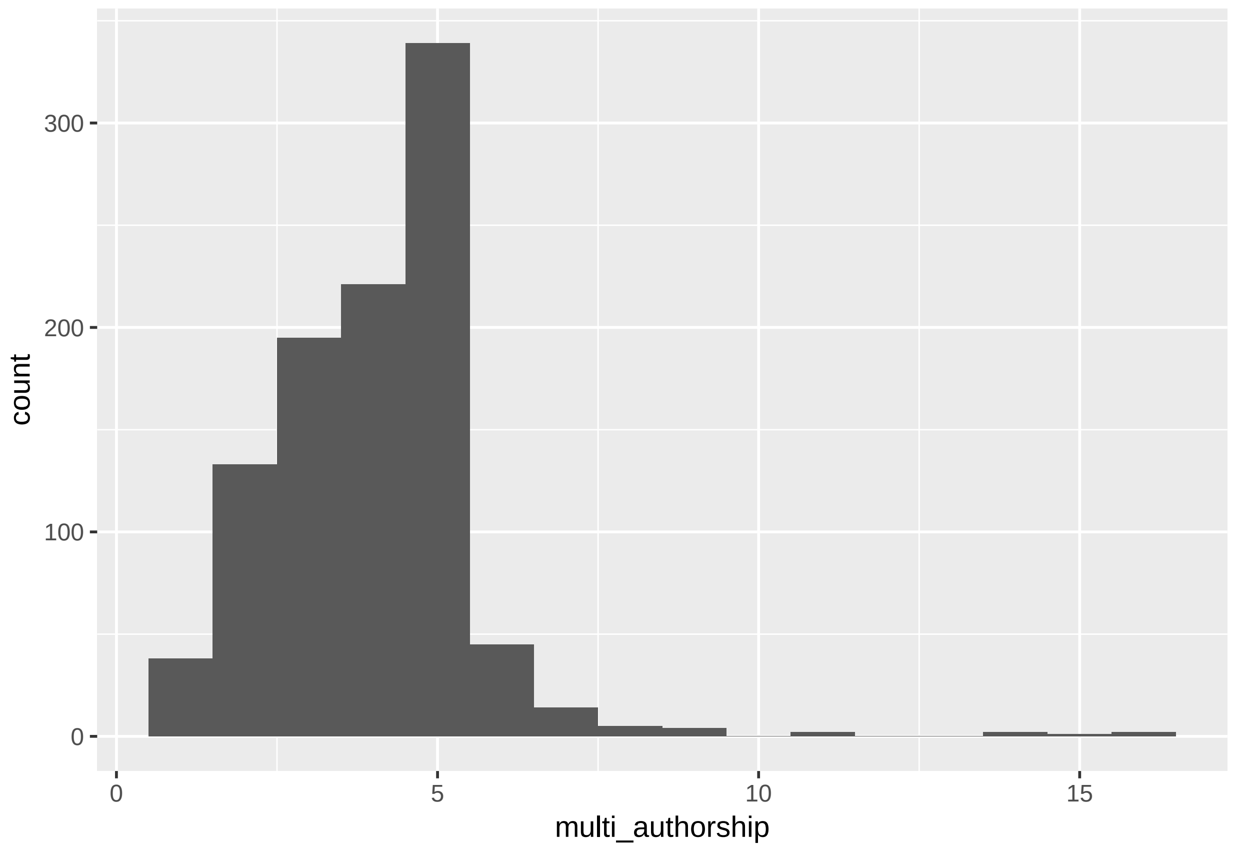
```
## # A tibble: 1,001 x 2
##    Authors                                                    multi_authorship
##    <chr>                                                                 <dbl>
##  1 Flavia Pennini|Angelo Plastino                                            2
##  2 Naveed Aslam|Peter C. 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

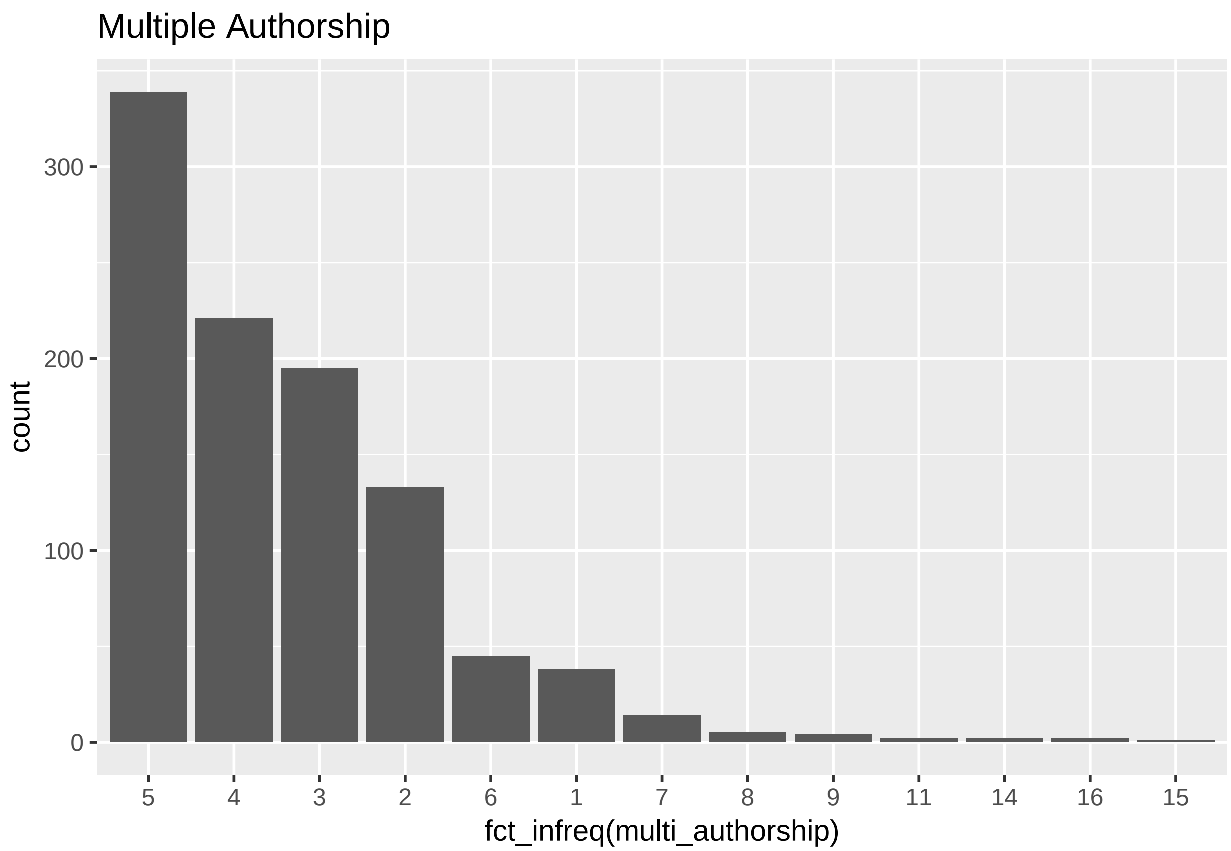
Generate a histogram distribution of the multiple authorship variable.

```
crossref_data %>%
  select(Authors) %>%
  mutate(multi_authorship = str_count(Authors, "\\|") + 1) %>%
  select(multi_authorship, Authors) %>%
  ggplot() +
  aes(multi_authorship) +
  geom_histogram(binwidth = 1)
```



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

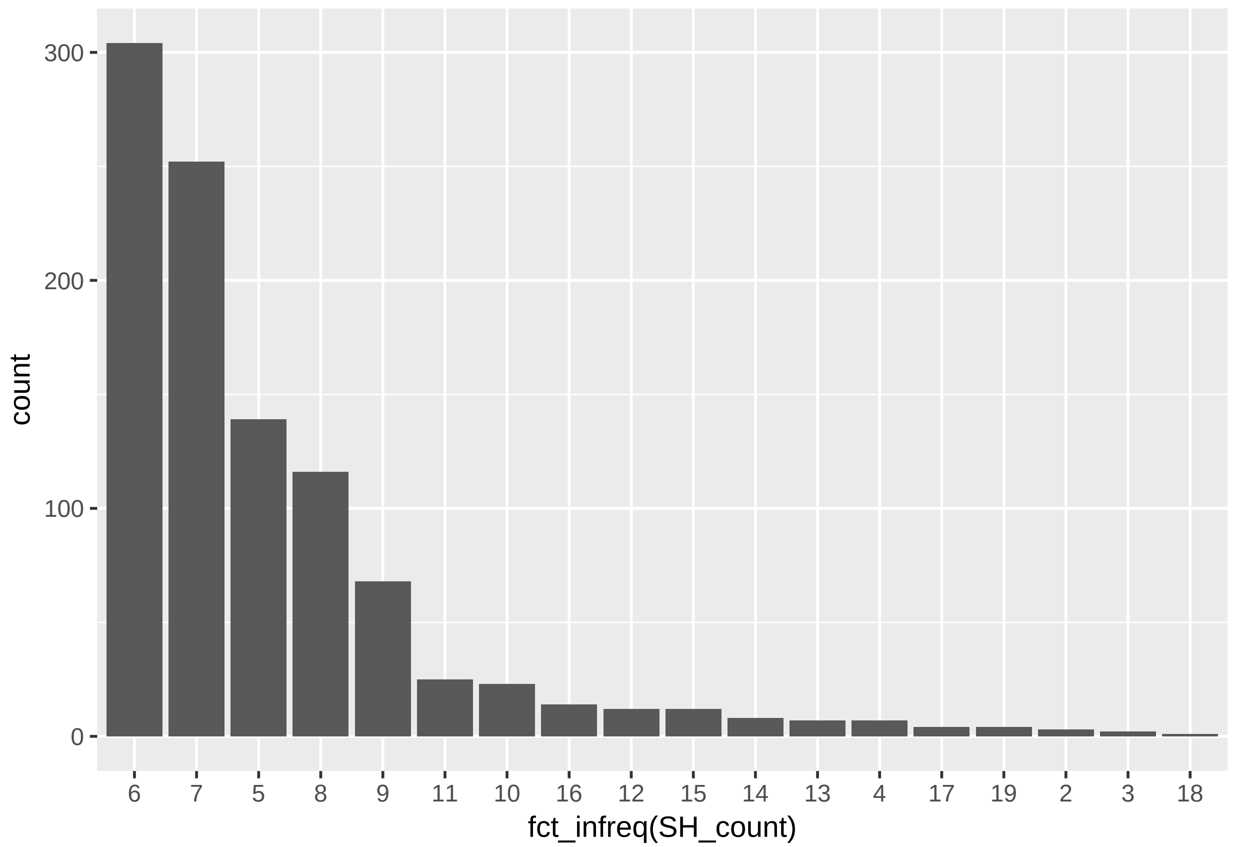
```
auth_count <- crossref_data %>%
  select(Authors) %>%
  mutate(multi_authorship = str_count(Authors, "\\|") + 1) %>%
  mutate(multi_authorship = as.character(multi_authorship)) %>%
  select(multi_authorship, Authors)

ggplot(auth_count) +
  aes(fct_infreq(multi_authorship)) +
  geom_bar() +
  ggtitle("Multiple Authorship")
```



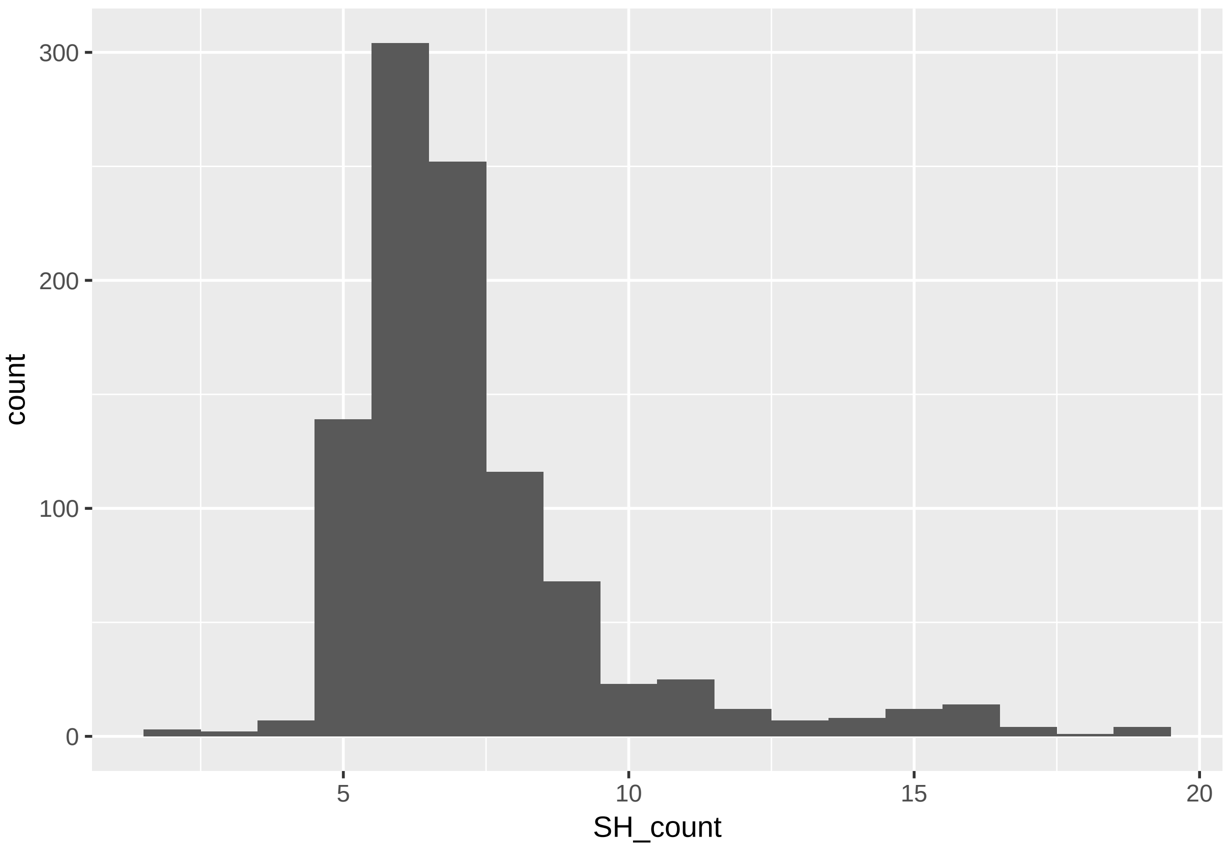
Explore Subject Headings

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

```
crossref_data %>%
  mutate(SH_count = str_count(Subjects, "\\|") + 1) %>%
  mutate(SH_count = as.character(SH_count)) %>%
  ggplot() +
  aes(fct_infreq(SH_count)) +
  geom_bar()
```



```
crossref_data %>%
  mutate(SH_count = str_count(Subjects, "\\|") + 1) %>%
  ggplot() +
  aes(SH_count) +
  geom_histogram(binwidth = 1)
```



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
```

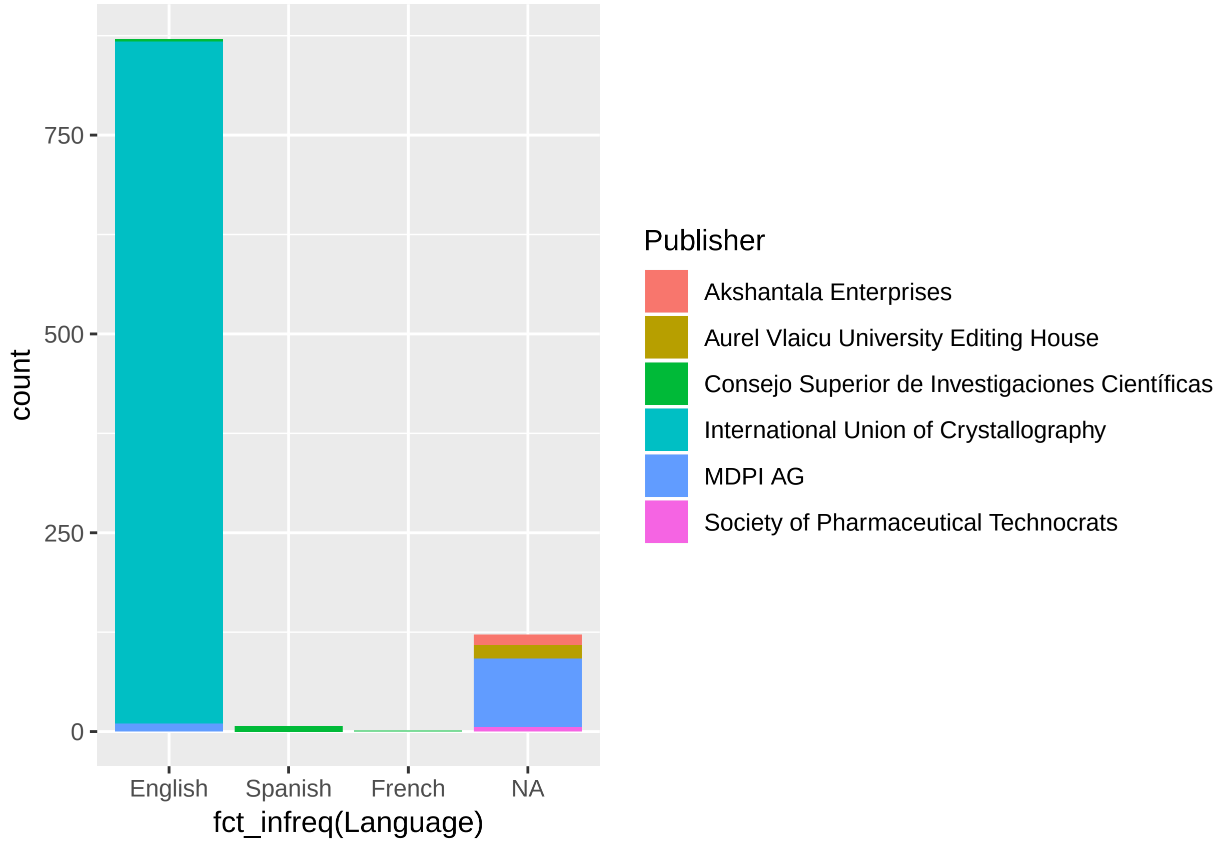
Since `EN` and `English` are synonymous, let's combine them into a single value. `case_when` is a great function for collapsing values.

```
crossref_data <- crossref_data %>%
  mutate(Language = case_when(
    Language == "EN" ~ "English",
    Language == "ES" ~ "Spanish",
    Language == "FR" ~ "French"
  ))
```

Visualize the Languages.

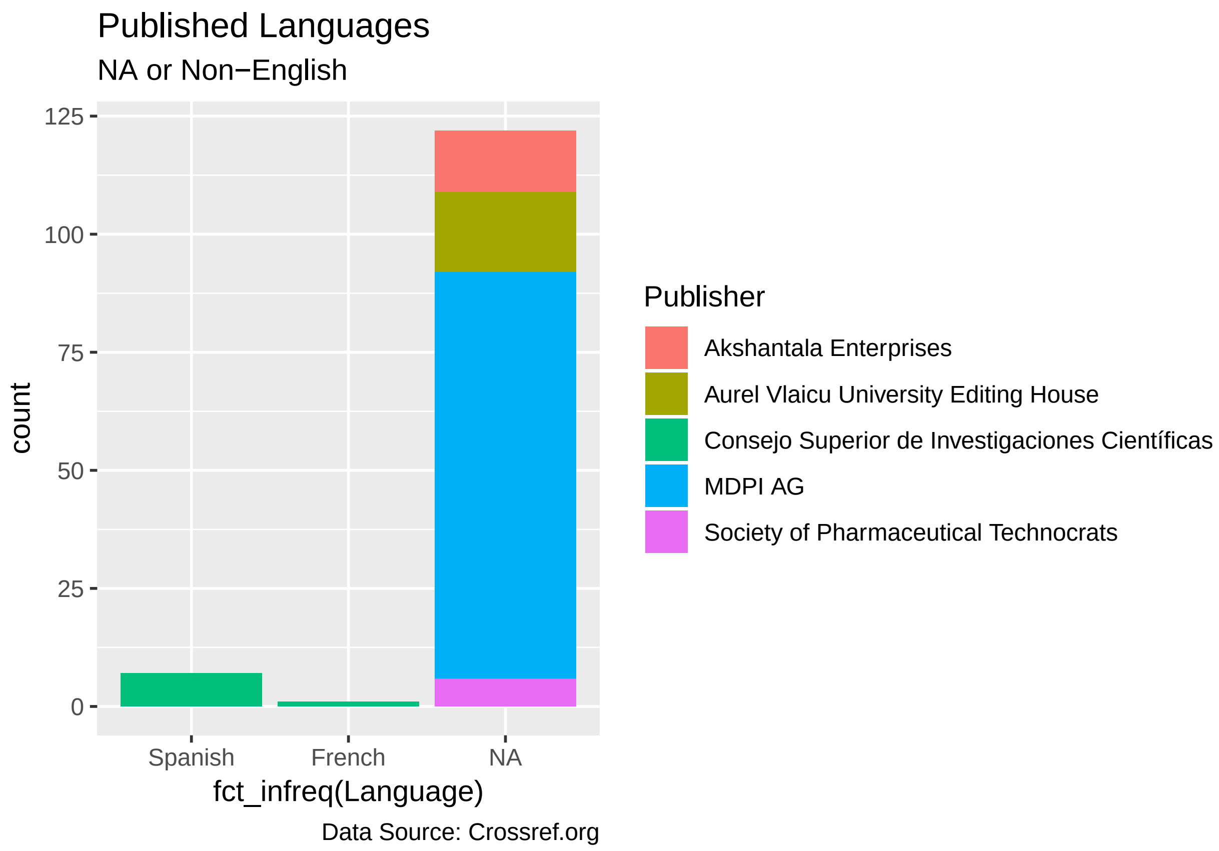
Stacked Bar graph shows frequency by Language. Each stack of a bar distinguishes the publishers. English Language is huge and somewhat over-powers the reset of the graph. Make a second graph (below) to drill down on the lesser represented languages.

```
crossref_data %>%
  ggplot() +
  aes(fct_infreq(Language), fill = Publisher) +
  geom_bar()
```



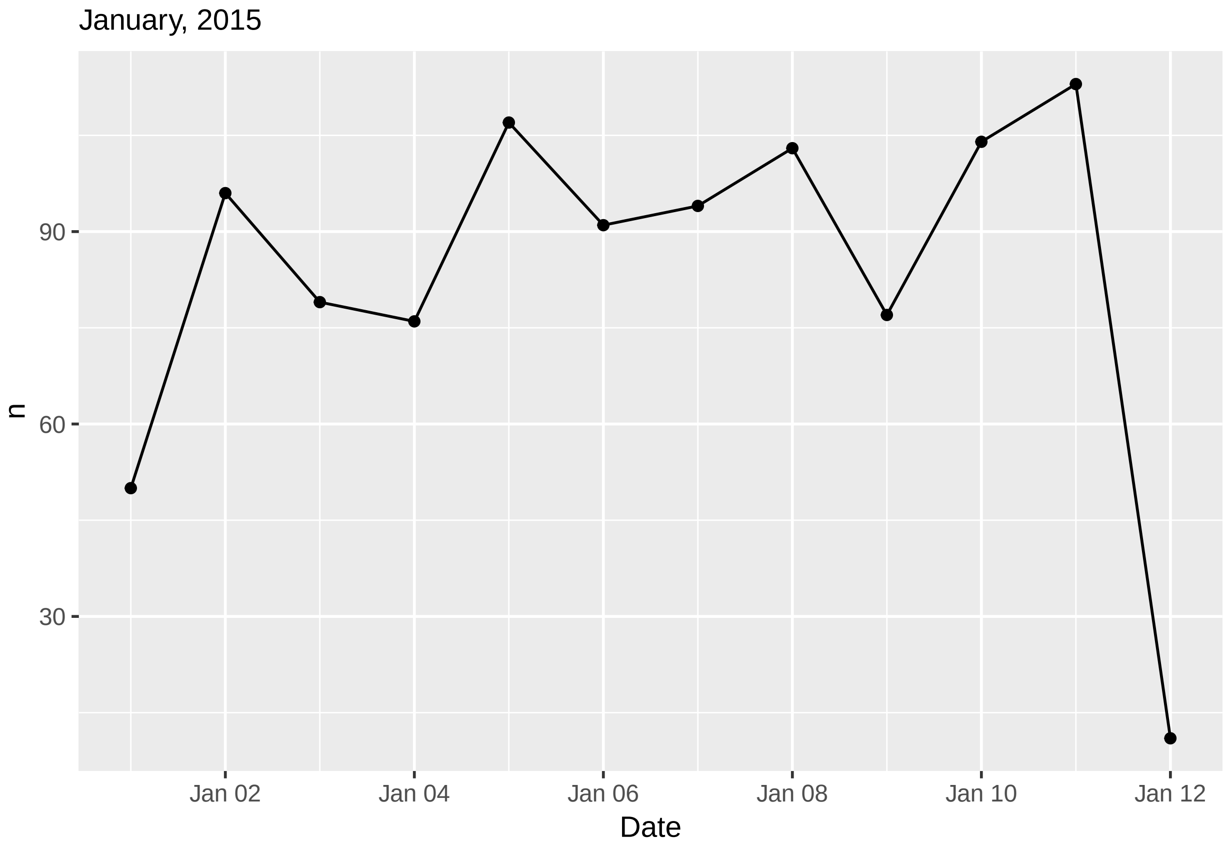
Filter the data to show only the "NA", "French", and "Spanish".

```
crossref_data %>%
  filter(is.na(Language) | Language == "French" | Language == "Spanish") %>%
  ggplot() +
  aes(fct_infreq(Language), fill = Publisher) +
  geom_bar() +
  labs(title = "Published Languages",
       subtitle = "NA or Non-English",
       caption = "Data Source: Crossref.org")
```



Time Series

```
crossref_data %>%
  count(Date) %>%
  ggplot(aes(Date, n)) +
  geom_point() +
  geom_line() +
  labs("Publishing Frequency by Day",
       subtitle = "January, 2015")
```


January, 2015
90
60
30
n
Jan 02
Jan 04
Jan 06
Jan 08
Jan 10
Jan 12
Date