Case study: mining NASA metadata

Monica Buczynski

October 12, 2020

Note: The purpose of this document is to showcase a sample of skills that I learned in *Text Mining with R: A Tidy Approach* by Julia Silge and David Robinson. Some scripts were taken from https://www.tidytextmining.com/s.html. The code for each exercise was studied carefully for understanding and then was retyped manually into R to maximize the learning experience; however, many of the scripts were altered for further analysis and presentation aesthetics. Additionally, I added my own code for further analysis and my own curiosity.

Skills that I focused on included:

- The tidy text format
- Sentiment analysis with tidy data
- Analyzing word and document frequency: tf-idf
- Relationships between words: n-grams and correlations
- Converting to and from non-tidy formats

8.1 How data is organized at NASA

Download the JSON file and take a look at the names of what is stored in the metadata.

```
metadata <- fromJSON("https://data.nasa.gov/data.json")
names(metadata$dataset)
```

```
##  [1] "accessLevel"                 "landingPage"
##  [3] "bureauCode"                  "issued"
##  [5] "@type"                       "modified"
##  [7] "references"                  "keyword"
##  [9] "contactPoint"                "publisher"
## [11] "identifier"                  "description"
## [13] "title"                       "programCode"
## [15] "distribution"                "accrualPeriodicity"
## [17] "theme"                       "temporal"
## [19] "spatial"                     "citation"
## [21] "data-presentation-form"      "release-place"
## [23] "series-name"                 "creator"
## [25] "graphic-preview-description" "graphic-preview-file"
## [27] "language"                    "editor"
## [29] "issue-identification"        "describedBy"
## [31] "describedByType"             "license"
## [33] "dataQuality"                 "rights"
```

Identify the type of data of the title, description, and keywords for each dataset - most useful for our analysis

```
class(metadata$dataset$title)
```

```
## [1] "character"
```

```
class(metadata$dataset$description)
```

```
## [1] "character"
```

```
class(metadata$dataset$keyword)
```

```
## [1] "list"
```

8.1.1 Wrangling and tidying the data

Set up separate tidy data frames for title, description, and keyword, keeping the dataset ids for each so that we can connect them later in the analysis if necessary.

```
nasa_title <- tibble(id = metadata$dataset$id,
                         title = metadata$dataset$title)
nasa_title
```

```
## # A tibble: 27,763 x 2
##    id                                    title
##    <chr>                                 <chr>
##  1 urn:nasa:pds:context_pds3:data_set:da~ ROSETTA-ORBITER EARTH RPCMAG 2 EAR2 R~
##  2 TECHPORT_9532                          Sealed Planetary Return Canister (SPR~
##  3 TECHPORT_9174                          Enhanced ORCA and CLARREO Depolarizer~
##  4 urn:nasa:pds:context_pds3:data_set:da~ NEAR EROS RADIO SCIENCE DATA SET - ER~
##  5 TECHPORT_5771                          A Constraint-Based Geospatial Data In~
##  6 TECHPORT_93196                         Goggle-Based Visual Field Device
##  7 TECHPORT_12939                         Highly Accurate Sensor for High-Purit~
##  8 urn:nasa:pds:context_pds3:data_set:da~ ASTEROID OCCULTATIONS V14.0
##  9 urn:nasa:pds:context_pds3:data_set:da~ VOYAGER 2 JUPITER MAGNETOMETER RESAMP~
## 10 C1236350976-GES_DISC                   POLDER/Parasol L2 Radiation Budget su~
## # ... with 27,753 more rows
```

Build the tidy data frame for the descriptions.

```
nasa_desc <- tibble(id = metadata$dataset$id,
                        desc = metadata$dataset$description)
nasa_desc
```

```
## # A tibble: 27,763 x 2
##    id                                    desc
##    <chr>                                 <chr>
##  1 urn:nasa:pds:context_pds3:data_set:da~ "This dataset contains EDITED RAW DAT~
##  2 TECHPORT_9532                          "Sample return missions have primary ~
##  3 TECHPORT_9174                          "Next generation Earth Science Satell~
##  4 urn:nasa:pds:context_pds3:data_set:da~ "The NEAR Eros Radio Science Data Set~
##  5 TECHPORT_5771                          "We propose to implement a constraint~
##  6 TECHPORT_93196                         "This proposed 2-yr project would: (1~
##  7 TECHPORT_12939                         "In this STTR effort, Los Gatos Resea~
##  8 urn:nasa:pds:context_pds3:data_set:da~ "This data set is intended to include~
##  9 urn:nasa:pds:context_pds3:data_set:da~ "This data set includes Voyager 2 Jup~
## 10 C1236350976-GES_DISC                   "This is the POLDER/Parasol Level-2 R~
## # ... with 27,753 more rows
```

Build the tidy data frame for the keywords. For this one, we need to use unnest() from tidyr, because they are in a list-column. This is a tidy data frame because we have one row for each keyword; this means we will have multiple rows for each dataset because a dataset can have more than one keyword.

```
nasa_keyword <- tibble(id = metadata$dataset$id,
                         keyword = metadata$dataset$keyword) %>%
  unnest(keyword)

nasa_keyword
```

```
## # A tibble: 124,739 x 2
##    id                                                    keyword
##    <chr>                                                 <chr>
```

```
##   1 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpc~ international rosetta ~
##   2 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpc~ earth
##   3 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpc~ unknown
##   4 TECHPORT_9532                                          jet propulsion laborat~
##   5 TECHPORT_9532                                          completed
##   6 TECHPORT_9174                                          completed
##   7 TECHPORT_9174                                          goddard space flight c~
##   8 urn:nasa:pds:context_pds3:data_set:data_set.near-a-r~ near earth asteroid re~
##   9 urn:nasa:pds:context_pds3:data_set:data_set.near-a-r~ eros
##  10 TECHPORT_5771                                          ames research center
## # ... with 124,729 more rows
```

Use tidytext's unnest_tokens() for the title and description fields so we can do the text analysis and remove stop words from the titles and descriptions.

```
nasa_title <- nasa_title %>%
  unnest_tokens(word, title) %>% anti_join(stop_words, by = "word")

nasa_desc <- nasa_desc %>%
  unnest_tokens(word, desc) %>%
  anti_join(stop_words, by ="word")

# View

nasa_title
```

```
## # A tibble: 240,598 x 2
##    id                                                              word
##    <chr>                                                           <chr>
##  1 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-~ rosetta
##  2 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-~ orbiter
##  3 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-~ earth
##  4 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-~ rpcmag
##  5 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-~ 2
##  6 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-~ ear2
##  7 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-~ raw
##  8 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-~ v3.0
##  9 TECHPORT_9532                                                   sealed
## 10 TECHPORT_9532                                                   planeta~
## # ... with 240,588 more rows
```

```
nasa_desc
```

```
## # A tibble: 2,641,345 x 2
##    id                                                               word
##    <chr>                                                            <chr>
##  1 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-v~ dataset
##  2 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-v~ edited
##  3 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-v~ raw
##  4 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-v~ data
##  5 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-v~ earth
##  6 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-v~ flyby
##  7 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-v~ ear2
##  8 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-v~ closest
##  9 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-v~ approa~
## 10 urn:nasa:pds:context_pds3:data_set:data_set.ro-e-rpcmag-2-ear2-raw-v~ ca
```

```
## # ... with 2,641,335 more rows
```

8.1.2 Some initial simple exploration

What are the most common words in the NASA dataset titles? We can use count() from dplyr

```
nasa_title %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 16,196 x 2
##    word       n
##    <chr>  <int>
##  1 phase   8661
##  2 data    3649
##  3 ii      2615
##  4 ges     2457
##  5 disc    2456
##  6 1       2097
##  7 level   1741
##  8 v1.0    1713
##  9 global  1660
## 10 2       1638
## # ... with 16,186 more rows
```

What about the descriptions?

```
nasa_desc %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 55,180 x 2
##    word        n
##    <chr>   <int>
##  1 data    47041
##  2 system  19288
##  3 phase   12073
##  4 2       11851
##  5 product 11270
##  6 nasa    10170
##  7 space   10166
##  8 based    9747
##  9 1        9607
## 10 surface  9350
## # ... with 55,170 more rows
```

Words like "data" and "global" are used very often in NASA titles and descriptions. We may want to remove digits and some "words" like "v1" from these data frames for many types of analyses; they are not too meaningful for most audiences.

```
my_stopwords <- tibble(word = c(as.character(1:10),
                                "v1", "v03", "l2", "l3", "l4", "v5.2.0",
                                    "v003", "v004", "v005", "v006", "v7"))

nasa_title <- nasa_title %>% anti_join(my_stopwords)

nasa_desc <- nasa_desc %>%
  anti_join(my_stopwords)
```

What are the most common keywords?

```r
nasa_keyword %>%
  group_by(keyword)%>%
  count(sort = TRUE)
```

```
## # A tibble: 6,912 x 2
## # Groups:   keyword [6,912]
##    keyword                         n
##    <chr>                       <int>
##  1 national geospatial data asset 10659
##  2 ngda                        10659
##  3 earth science               10131
##  4 completed                    9021
##  5 atmosphere                   4513
##  6 active                       2885
##  7 oceans                       2298
##  8 land surface                 2203
##  9 spectral/engineering         1679
## 10 goddard space flight center  1537
## # ... with 6,902 more rows
```

We likely want to change all of the keywords to either lower or upper case to get rid of duplicates like "OCEANS" and "Oceans".

```r
nasa_keyword <- nasa_keyword %>% mutate(keyword = toupper(keyword))
```

8.2 Word co-ocurrences and correlations

8.2.1 Networks of Description and Title Words

We can use pairwise_count() from the widyr package to count how many times each pair of words occurs together in a title or description field.

```
# title
title_word_pairs <- nasa_title %>%
  pairwise_count(word, id, sort = TRUE, upper = FALSE)
```

```
## Warning: `distinct_()` is deprecated as of dplyr 0.7.0.
## Please use `distinct()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.

## Warning: `tbl_df()` is deprecated as of dplyr 1.0.0.
## Please use `tibble::as_tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
title_word_pairs
```

```
## # A tibble: 313,040 x 3
##    item1 item2      n
##    <chr> <chr>  <dbl>
##  1 phase ii      2498
##  2 ges   disc    1441
##  3 phase system   948
##  4 phase space    637
##  5 phase based    602
##  6 ges   degree   601
##  7 disc  degree   601
##  8 phase low      480
##  9 phase power    441
## 10 ges   level    426
## # ... with 313,030 more rows
```

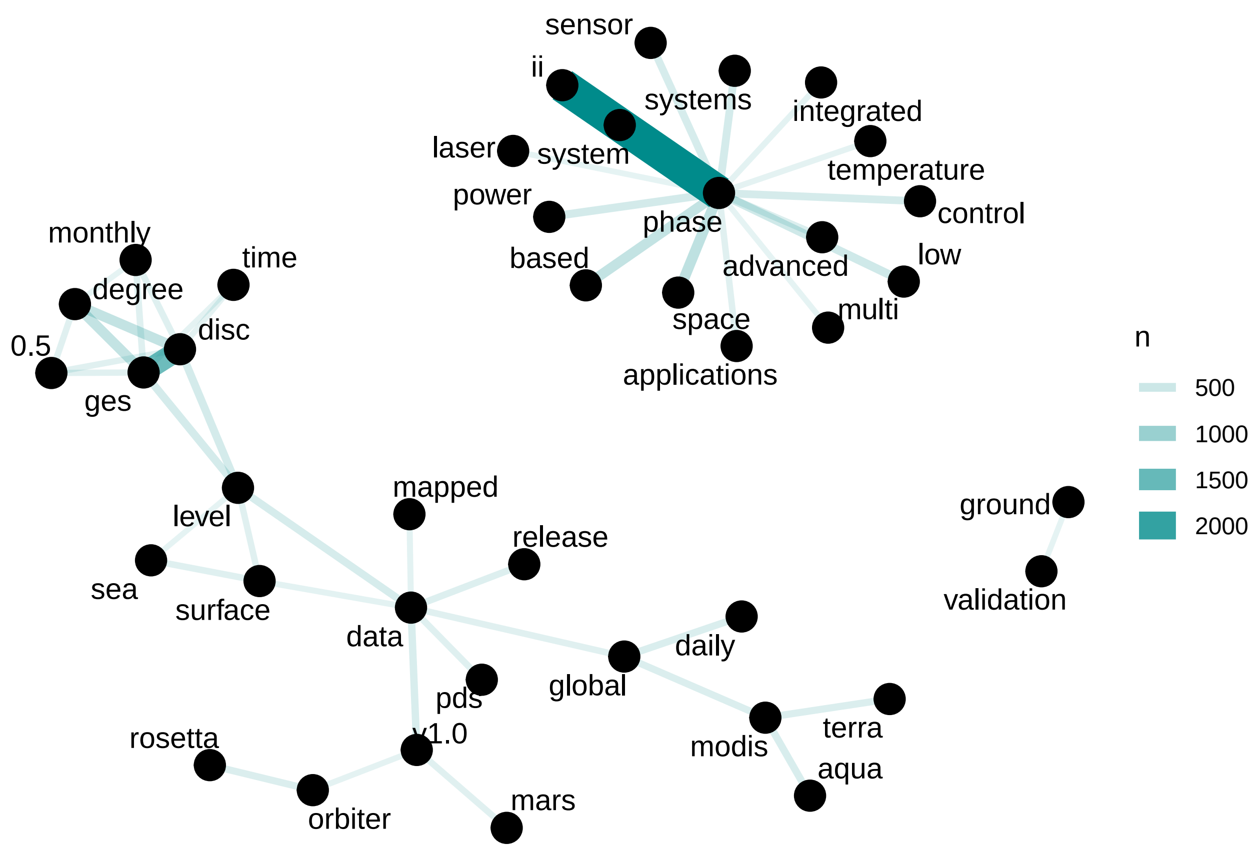
```
#description
desc_word_pairs <- nasa_desc %>%
  pairwise_count(word, id, sort = TRUE, upper = FALSE)

desc_word_pairs
```

```
## # A tibble: 19,963,312 x 3
##    item1 item2          n
##    <chr> <chr>      <dbl>
##  1 data  set         4519
##  2 data  system      3468
##  3 data  resolution  3194
##  4 data  time        3173
##  5 data  product     3160
##  6 data  nasa        3108
##  7 phase ii          3068
##  8 data  based       2955
##  9 data  level       2889
```

```
## 10 data  instrument  2877
## # ... with 19,963,302 more rows
```

Plot networks of these co-occurring words so we can see these relationships

```
set.seed(1234)
title_word_pairs %>%
  filter(n >= 250) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = n, edge_width = n), edge_colour = "cyan4") +
  geom_node_point(size = 5) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()
```



We do not see clear clustering structure in the network. We may want to use tf-idf as a metric to find characteristic words for each description field, instead of looking at counts of words.

```r
set.seed(1234)
desc_word_pairs %>%
  filter(n >= 1600) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = n, edge_width = n), edge_colour = "darkred") +
  geom_node_point(size = 5) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()
```

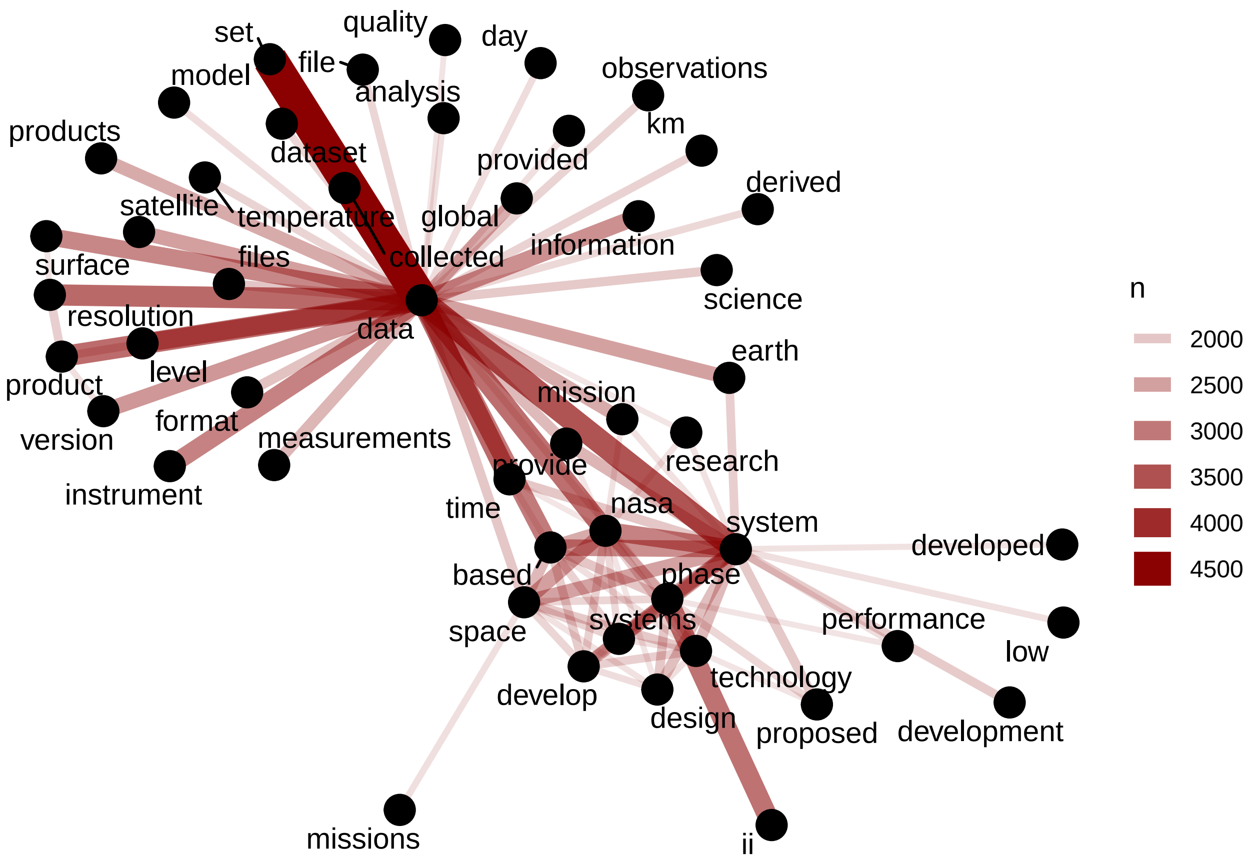


Figure 1: Word network in NASA dataset descriptions

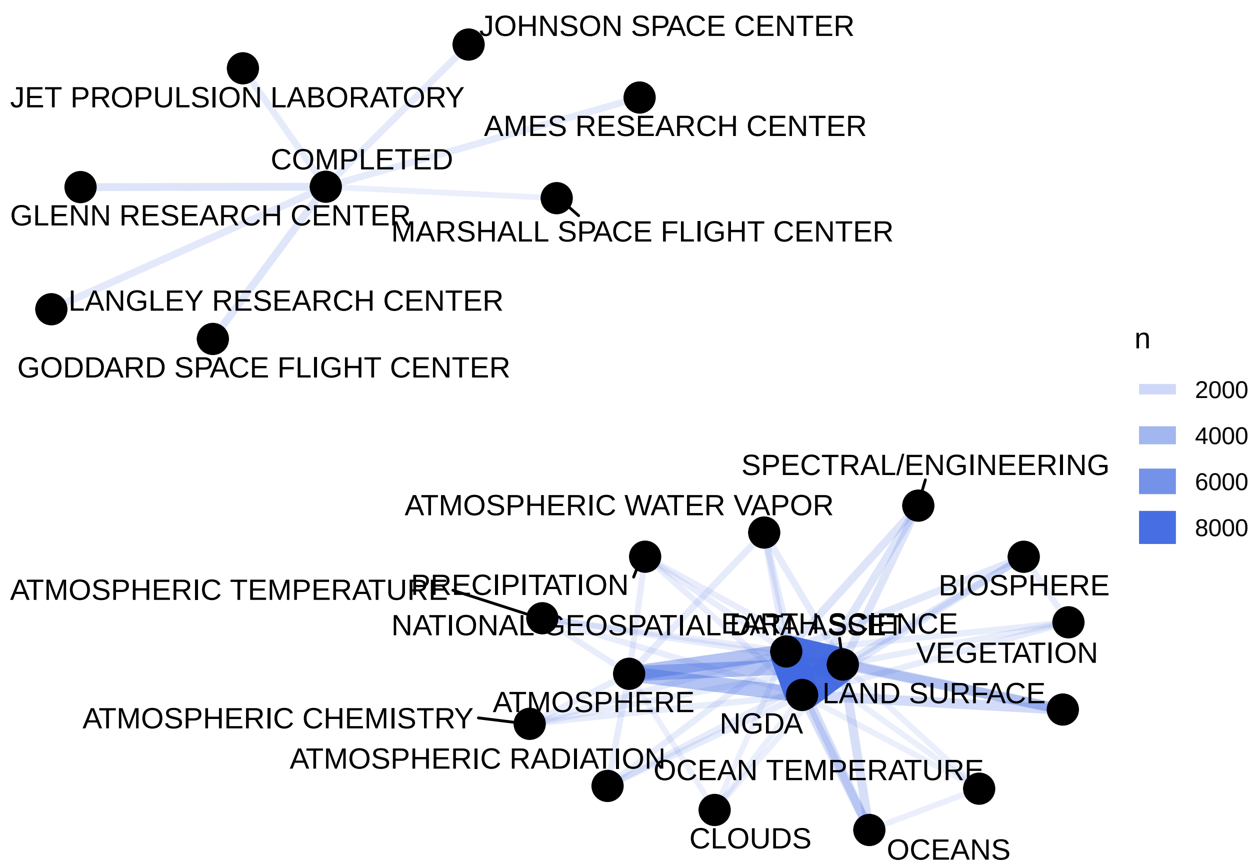
8.2.2 Networks of Keywords

Make a network of the keywords to see which keywords commonly occur together in the same datasets.

```
keyword_pairs <- nasa_keyword %>%  pairwise_count(keyword, id, sort = TRUE, upper = FALSE)

keyword_pairs
```

```
## # A tibble: 2,000,335 x 3
##    item1                        item2                            n
##    <chr>                        <chr>                        <dbl>
##  1 NATIONAL GEOSPATIAL DATA ASSET NGDA                         8288
##  2 EARTH SCIENCE                NATIONAL GEOSPATIAL DATA ASSET  7959
##  3 EARTH SCIENCE                NGDA                          7959
##  4 ATMOSPHERE                   EARTH SCIENCE                 3237
##  5 ATMOSPHERE                   NATIONAL GEOSPATIAL DATA ASSET  3227
##  6 ATMOSPHERE                   NGDA                          3227
##  7 EARTH SCIENCE                LAND SURFACE                  1906
##  8 NATIONAL GEOSPATIAL DATA ASSET LAND SURFACE                 1903
##  9 NGDA                         LAND SURFACE                  1903
## 10 EARTH SCIENCE                OCEANS                        1623
## # ... with 2,000,325 more rows
```

```r
set.seed(1234)
keyword_pairs %>%
  filter(n >= 700) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = n, edge_width = n), edge_colour = "royalblue") +
  geom_node_point(size = 5) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()
```



To examine the relationships among keywords in a different way, I find the correlation among the keywords. This looks for those keywords that are more likely to occur together than with other keywords for a dataset. When the correlation coefficient is equal to 1, these words always appear together.

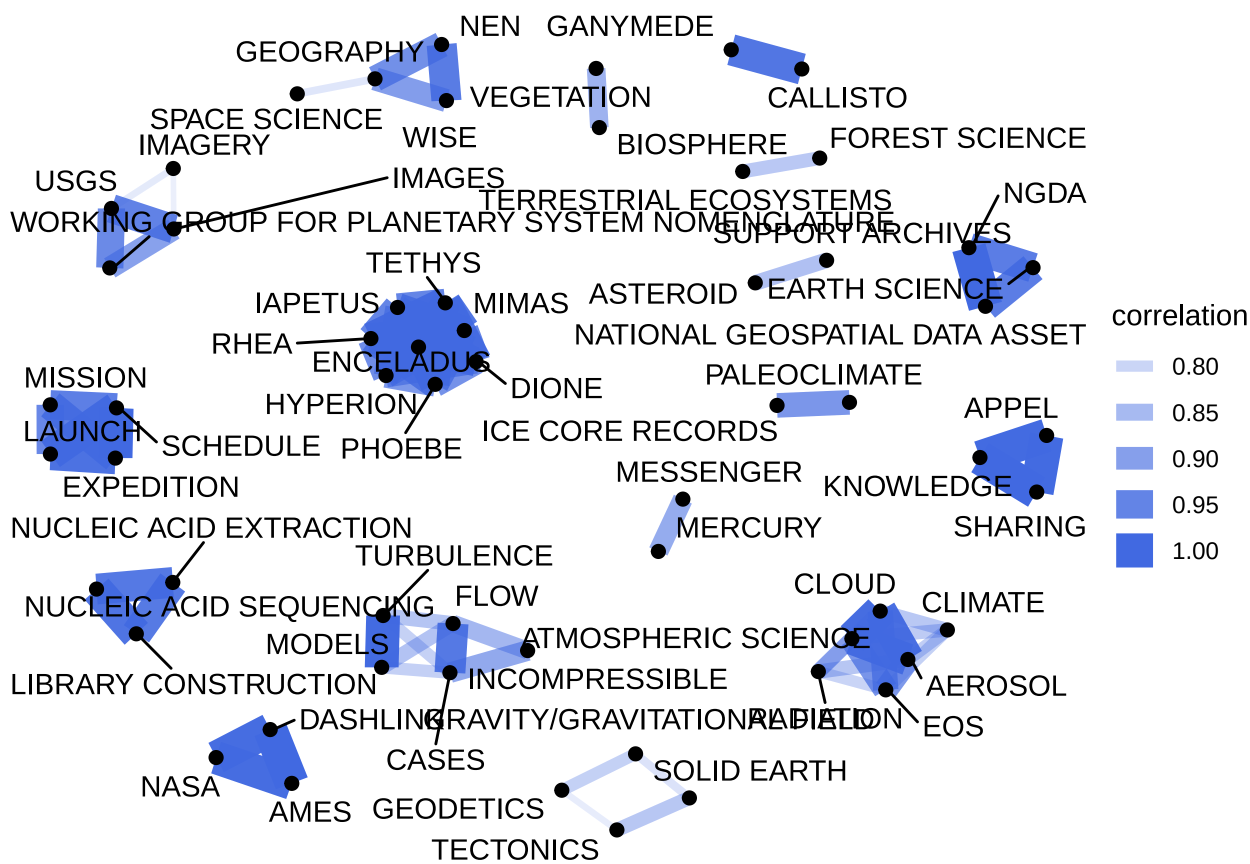
```r
keyword_cors <- nasa_keyword %>%
  group_by(keyword) %>%
  filter(n() >= 50) %>%
  pairwise_cor(keyword, id, sort = TRUE, upper = FALSE)

keyword_cors
```

```
## # A tibble: 15,753 x 3
##    item1                         item2      correlation
##    <chr>                         <chr>            <dbl>
##  1 AMES                          DASHLINK         1.
##  2 NATIONAL GEOSPATIAL DATA ASSET NGDA            1
##  3 SCHEDULE                      EXPEDITION       1
##  4 KNOWLEDGE                     SHARING          1
##  5 MODELS                        TURBULENCE       0.997
##  6 KNOWLEDGE                     APPEL            0.997
##  7 SHARING                       APPEL            0.997
##  8 ATMOSPHERIC SCIENCE           CLOUD            0.994
##  9 AMES                          NASA             0.991
## 10 NASA                          DASHLINK         0.991
## # ... with 15,743 more rows
```

Visualize the network of keyword correlations.

Note: This network appears much different than the co-occurrence network. The difference is that the co-occurrence network asks a question about which keyword pairs occur most often, and the correlation network asks a question about which keywords occur more often together than with other keywords.

```r
set.seed(1234)
keyword_cors %>%
  filter(correlation > .75) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation, edge_width = correlation), edge_colour = "royalblue") +
  geom_node_point(size = 2) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()
```



8.3 Calculating tf-idf for the description fields

Use tf-idf, the term frequency times inverse document frequency, to identify words that are especially important to a document within a collection of documents.

8.3.1 What is tf-idf for the description field words?

```
desc_tf_idf <- nasa_desc %>%
  count(id, word, sort = TRUE) %>%
  ungroup() %>%
  bind_tf_idf(word, id, n)
```

What are the highest tf-idf words in the NASA description fields?

Note: Notice we have run into an issue here; both n and term frequency are equal to 1 for these terms, meaning that these were description fields that only had a single word in them. If a description field only contains one word, the tf-idf algorithm will think that is a very important word.

```
desc_tf_idf %>%
  arrange(-tf_idf)
```

```
## # A tibble: 1,683,523 x 6
##    id                word                                  n    tf   idf tf_idf
##    <chr>             <chr>                             <int> <dbl> <dbl>  <dbl>
##  1 C1633360161-OB_~ bio_optics_chl_polarization            2     1 10.1    10.1
##  2 C1633360353-OB_~ gulfcarbon                             2     1 10.1    10.1
##  3 C1206487217-ASF  palsar_radiometric_terrain_correct~     1     1 10.1    10.1
##  4 C1206487504-ASF  palsar_radiometric_terrain_correct~     1     1 10.1    10.1
##  5 TECHPORT_33575   abcd                                   1     1 10.1    10.1
##  6 TECHPORT_94546   xxxx                                   1     1 10.1    10.1
##  7 TECHPORT_94119   aerosciences                           1     1  9.04   9.04
##  8 NASA-438         lgrs                                   1     1  8.06   8.06
##  9 NASA-446         lgrs                                   1     1  8.06   8.06
## 10 NASA-463         lgrs                                   1     1  8.06   8.06
## # ... with 1,683,513 more rows
```

8.3.2 Connecting description fields to keywords

Full join of the keyword data frame and the data frame of description words with tf-idf, and then find the highest tf-idf words for a given keyword.

```
desc_tf_idf <- full_join(desc_tf_idf, nasa_keyword, by = "id")
```

Plot some of the most important words, as measured by tf-idf, for a few example keywords used on NASA datasets.

```
desc_tf_idf %>%
  filter(!near(tf, 1)) %>%
  filter(keyword %in% c("SOLAR ACTIVITY", "CLOUDS",
                        "SEISMOLOGY", "ASTROPHYSICS",
                        "HUMAN HEALTH", "BUDGET")) %>%
  arrange(desc(tf_idf)) %>%
  group_by(keyword) %>%
  distinct(word, keyword, .keep_all = TRUE) %>%
  top_n(15, tf_idf) %>%
  ungroup() %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  ggplot(aes(word, tf_idf, fill = keyword)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~keyword, ncol = 3, scales = "free") +
  coord_flip() +
  labs(title = "Highest tf-idf words in NASA metadata description fields",
       caption = "NASA metadata from https://data.nasa.gov/data.json",
       x = NULL, y = "tf-idf")
```

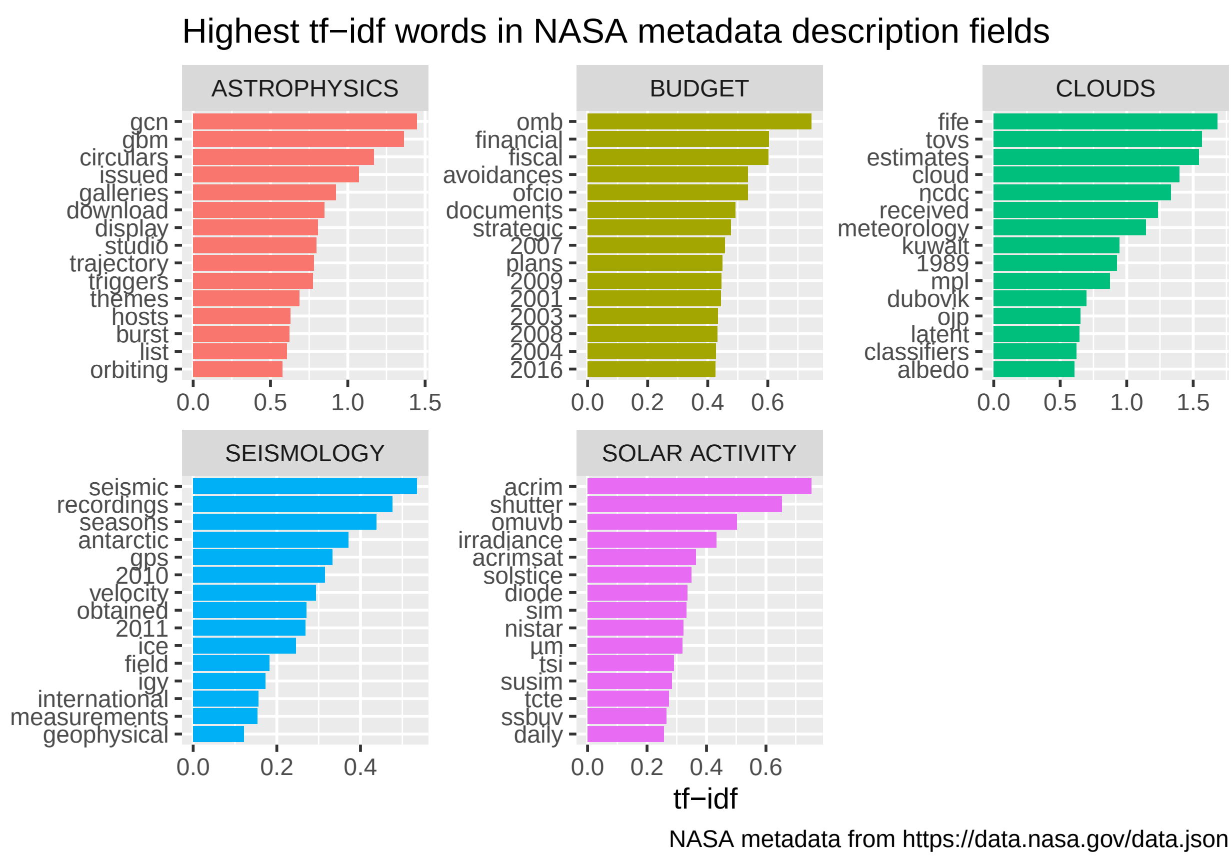


Figure 2: Distribution of tf-idf for words from datasets labeled with selected keywords

8.4 Topic modeling

8.4.1 Casting to a document-term matrix

Need to make a DocumentTermMatrixRows: correspond to documents (description texts in our case) and columns correspond to terms (i.e., words); it is a sparse matrix and the values are word counts.

Clean up the text a bit using stop words to remove some of the nonsense "words" leftover from HTML or other character encoding. Use bind_rows() to add our custom stop words to the list of default stop words from the tidytext package, and then use anti_join() to remove them all from our data frame.

```
my_stop_words <- bind_rows(stop_words,
                           tibble(word = c("nbsp", "amp", "gt", "lt",
                                           "timesnewromanpsmt", "font",
                                           "td", "li", "br", "tr", "quot",
                                           "st", "img", "src", "strong",
                                           "http", "file", "files",
                                           as.character(1:12)),
                                  lexicon = rep("custom", 30)))


word_counts <- nasa_desc %>%
  anti_join(my_stop_words) %>%
  count(id, word, sort = TRUE) %>%
  ungroup()

word_counts
```

```
## # A tibble: 1,675,101 x 3
##    id                     word      n
##    <chr>                  <chr> <int>
##  1 C1625703857-LAADS      93       84
##  2 TECHPORT_93269         em       60
##  3 C1237113343-GES_DISC   f11      44
##  4 C1432254058-GES_DISC   data     44
##  5 C1227323456-LARC_ASDC ceres    43
##  6 C5769450-LARC_ASDC     ceres    43
##  7 C1227323481-LARC_ASDC ceres    42
##  8 C1237113343-GES_DISC   f13      42
##  9 C1227323455-LARC_ASDC ceres    41
## 10 C1227323457-LARC_ASDC ceres    41
## # ... with 1,675,091 more rows
```

Cast a df with cast_tdm. Turns a "tidy" one-term-per-document-per-row data frame into a DocumentTermMatrix. 100% sparse –> almost all of the entries in this matrix are zero.

```
desc_dtm <- word_counts %>%
  cast_dtm(id, word, n)

desc_dtm
```

```
## <<DocumentTermMatrix (documents: 25267, terms: 55139)>>
## Non-/sparse entries: 1675101/1391522012
## Sparsity           : 100%
## Maximal term length: 166
## Weighting          : term frequency (tf)
```

Topic modeling section of this case was reviewed but not completed due to poor function of personal device. See https://www.tidytextmining.com/nasa.html#ready-for-topic-modeling.