Project Report

Ye Zhang

10/01/2017

Introduction

The Public Library of Science (PLoS) is a nonprofit open access science, technology and medicine publisher, innovator and advocacy organization with a library of open access journals and other scientific literature under an open content license. This project is to perform an analysis of the statistical analyses in all published PLoS papers, so as to answer quenstions as below:

- What are the most common techniques?
- How do they vary by field?
- Are there any trends over the last 10-15 years?

Methods and Materials

Data: the dataset for this project should include all the published PLoS papers from its 7 journals, PLoS one, PLoS Biology, PLoS Medicine, PLoS Comutational Biology, PLoS Genetic, PLoS Neglected Tropical Diseases and PLos Pathogens. For each publication, there'are a list of information we need to download from the websites into our R program as the dataset:

- Article title
- Authors
- Article DOI
- PLoS journal
- Date of publication
- Materials and Methods part

Usually the statistical analysis techique utilized in a publication is described in the *Materials and Methods* section of the article,thus we should focus on extracting all the types of data analyses techniques mentioned in the *Materials and Methods* section of all the publications. One possibe way is to look for certain key words, such as "Hypothesis testing", "t test", "linear regression", "log linear regression", et al. With this method, it is important to establish a decent pool of key words before extraction, and some refereneces summarizing the statistical analyses methods online could be helpful, such as https://www.statisticallysignificantconsulting.com/Statistical-Tests.htm.

After extracting all the key words from articles, we can then start to answer the three questions listed at the beginning. With the dataset established through **Step 1** and **2**, it's possible to figure out the most commonly utilized analyses techniques, and coorelation between these techniques and the fields (the PLoS journal) and publication years. Take the key word "t test" as an example, we can figure out how many times the "t test" is mentioned over the years as well as in articles among 7 different fields.

```
## Loading required package: NLP

## Loading required package: rplos

## Loading required package: fulltext
```

Preliminary Data preparation

In order to establish a pool for the key words, first a list of full articles with the word "statistics" in "abstract" is searched using R package "rplos", which contains functions that can be used for PLoS article searching and information download. By indicating "statistic" in the "materials and methods" part, we can achieve result `outide_id` containing all the DOIs of all the full articles that we are interested in and then download the abstracts of these articles. Here I download abstracts of 500 articles with the word "statistics" in their abstracts. After tidying up this preliminary download data, I unnest the tokens using `word`, `bigram (two words combination)` and `trigram (three words combination)` respectively and calculated the frequency of these `word`, `bigram` and `trigram`. Then I can have a rough summary of the most frequent statistical methods mentioned in the 500 abstracts after going through these three data frames ordered with frequency.

```r
install.packages("tidytext",repos="http://cran.rstudio.com/")
library(tidytext)
library(dplyr)
library(tidyr)
library(stringr)

# out_id_all <- searchplos(q="materials_and_methods: statistics",
#                      fl="id", fq='doc_type: full', sort='publication_date desc')
# out_id_all$meta


out_id_all <- searchplos(q="abstract: statistics",
                    fl="id", fq='doc_type: full', sort='publication_date desc')
out_id_all$meta

# out_id <- searchplos(q="materials_and_methods: statistics",
#                      fl="id", fq='doc_type: full', sort='publication_date desc', limit = 500)

out_id <- searchplos(q="abstract: statistics",
                     fl="id", fq='doc_type: full', sort='publication_date desc', limit = 500)

# Abstract text xml given a DOI
out_fulltext <- plos_fulltext(doi=out_id$data$id[1])
data <- xmlParse(out_fulltext[[1]])
out_abstract_1 <- xpathSApply(data, "//abstract", xmlValue)
tidy_abstract_1 <- out_abstract_1 %>% str_replace_all("[[:punct:]]", " ") %>%
  str_replace_all("[[:digit:]]"," ") %>% tidy()

tidy_abstract_all <- tidy_abstract_1

for (i in 2:500) {
  out_ft <- plos_fulltext(doi=out_id$data$id[i])
  out_abs <- xpathSApply(xmlParse(out_ft[[1]]), "//abstract", xmlValue)
  tidy_abs <- out_abs %>% str_replace_all("[[:punct:]]", " ") %>% str_replace_all("[[:digit:]]"," ") %>%
  tidy_abstract_all <- rbind(tidy_abstract_all,tidy_abs)
}

save(tidy_abstract_all, file="tidy_500absRData")
```

```r
install.packages("tidytext",repos="http://cran.rstudio.com/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/5s/d6trnk_14_lb96_4zy9nc57c0000gn/T//Rtmp08B3vU/downloaded_packages
```

```r
library(tidytext)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:fulltext':
##
##     collect
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(stringr)

load("tidy_500absRData")

file_word <- tidy_abstract_all %>%
  unnest_tokens(word, x) %>%
  anti_join(stop_words) %>%
  group_by(word) %>%
  tally() %>%
  arrange(desc(n))
```

```
## Joining, by = "word"
```

```
file_bigram <- tidy_abstract_all %>%
  unnest_tokens(bigram, x, token="ngrams", n=2) %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  unite(bigram, word1, word2, sep = " ") %>%
  count(bigram, sort=TRUE) %>%
  arrange(desc(n))

file_trigram <- tidy_abstract_all %>%
  unnest_tokens(trigram, x, token="ngrams", n=3) %>%
  separate(trigram, c("word1", "word2", "word3"), sep = " ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word3 %in% stop_words$word) %>%
  unite(trigram, word1, word2, word3, sep = " ") %>%
  count(trigram, sort=TRUE) %>%
  arrange(desc(n))

head(file_word)
```

```
## # A tibble: 6 x 2
##          word     n
##         <chr> <int>
## 1       study   567
## 2    patients   521
## 3        data   460
## 4    analysis   439
```

```
## 5        risk   354
## 6 significant   343
```

```r
head(file_bigram, 10)
```

```
## # A tibble: 10 x 2
##                       bigram     n
##                        <chr> <int>
##  1 statistically significant   159
##  2              risk factors    58
##  3             meta analysis    56
##  4             breast cancer    44
##  5       logistic regression    44
##  6             mental health    41
##  7   significant differences    37
##  8      statistical analysis    37
##  9                    aor ci    36
## 10             public health    35
```

```r
head(file_trigram,10)
```

```
## # A tibble: 10 x 2
##                                  trigram     n
##                                    <chr> <int>
##  1                       body mass index    19
##  2 statistically significant differences    19
##  3  statistically significant difference    16
##  4                 confidence interval ci    15
##  5          logistic regression analysis    15
##  6                  cross sectional study    12
##  7          randomized controlled trials    11
##  8                       acetate pet mri    10
##  9                    children aged months    10
## 10                       clif sofa score    10
```

Data download for analysis

As stated above, create a decent pool for common statistical methods by looking for the most frequent `word`, `bigram` and `trigram` in 500 abstracts. Then download publication information including DOI, title, publication journal, and publication date, with these key words in "material and methods" part.

First, search for the number of publications with each key words in their "materials and methods".

```r
dic <- c("logistic regression", "meta analysis", "bootstrap", "ANOVA", "clustering", "bayesian", "t-tes
         "linear regression", "machine learning", "maximum likelihood", "neural network", "random fores
         "support vector machine", "MCMC")

# Using keywords in the pool and return "material and methods"
LogReg <- searchplos(q="materials_and_methods: logistic regression",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc')
MetaAnal <- searchplos(q="materials_and_methods: meta analysis",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc')
Bootstrap <- searchplos(q="materials_and_methods: bootstrap",
                     fl=c("id","title","journal","publication_date"),
```

```
                    fq='doc_type: full', sort='publication_date desc')
ANOVA <- searchplos(q="materials_and_methods: ANOVA",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc')
Cluster <- searchplos(q="materials_and_methods: clustering",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc')
Bayesian <- searchplos(q="materials_and_methods: bayesian",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc')
Ttest <- searchplos(q="materials_and_methods: t-test",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc')
LinReg <- searchplos(q="materials_and_methods: linear regression",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc')
MachLrn <- searchplos(q="materials_and_methods: machine learning",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc')
MaxL <- searchplos(q="materials_and_methods: maximum likelihood",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc')
NeuNet <- searchplos(q="materials_and_methods: neural network",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc')
RamFor <- searchplos(q="materials_and_methods: random forest",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc')
SVM <- searchplos(q="materials_and_methods: support vector machine",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc')
MCMC <- searchplos(q="materials_and_methods: MCMC",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc')

counts <- c(LogReg$meta$numFound, MetaAnal$meta$numFound, Bootstrap$meta$numFound, ANOVA$meta$numFound,
            Cluster$meta$numFound, Bayesian$meta$numFound, Ttest$meta$numFound, LinReg$meta$numFound,
            MachLrn$meta$numFound, MaxL$meta$numFound, NeuNet$meta$numFound, RamFor$meta$numFound,
            SVM$meta$numFound, MCMC$meta$numFound)
df <- data.frame(methods = dic, counts = counts)
```

```
LogReg_all <- searchplos(q="materials_and_methods: logistic regression",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc', limit=15920)
MetaAnal_all <- searchplos(q="materials_and_methods: meta analysis",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc', limit=8350)
Bootstrap_all <- searchplos(q="materials_and_methods: bootstrap",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc', limit = 10464)
ANOVA_all <- searchplos(q="materials_and_methods: ANOVA",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc', limit = 43599)
Cluster_all <- searchplos(q="materials_and_methods: clustering",
```

```r
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc', limit = 27202)
Bayesian_all <- searchplos(q="materials_and_methods: bayesian",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc',limit = 7514)
Ttest_all <- searchplos(q="materials_and_methods: t-test",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc', limit = 42138)
LinReg_all <- searchplos(q="materials_and_methods: linear regression",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc', limit = 26222)
MachLrn_all <- searchplos(q="materials_and_methods: machine learning",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc', limit = 2605)
MaxL_all <- searchplos(q="materials_and_methods: maximum likelihood",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc', limit = 15709)
NeuNet_all <- searchplos(q="materials_and_methods: neural network",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc', limit=2956)
RamFor_all <- searchplos(q="materials_and_methods: random forest",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc',limit = 8916)
SVM_all <- searchplos(q="materials_and_methods: support vector machine",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc', limit = 2068)
MCMC_all <- searchplos(q="materials_and_methods: MCMC",
                     fl=c("id","title","journal","publication_date"),
                    fq='doc_type: full', sort='publication_date desc', limit = 2800)
save(LogReg_all, MetaAnal_all, Bootstrap_all, ANOVA_all, Cluster_all, Bayesian_all, Ttest_all,
     LinReg_all, MachLrn_all, MaxL_all, NeuNet_all, RamFor_all, SVM_all, MCMC_all,
     file = "data.RData")
#Data <- rbind(LogReg_all,MetaAnal_all)
#write.csv(Data, "Data.csv")
```

Data cleaning

After downloading the "abstract"" and "materials and methods" from articles we are intereted in, we clean the data using R package "tidyr" and "tidytext", removing the stopwords and punctuations.

Results and Discussion

1. Compare the frequency of statistical methods by counting the number of articles using the key words

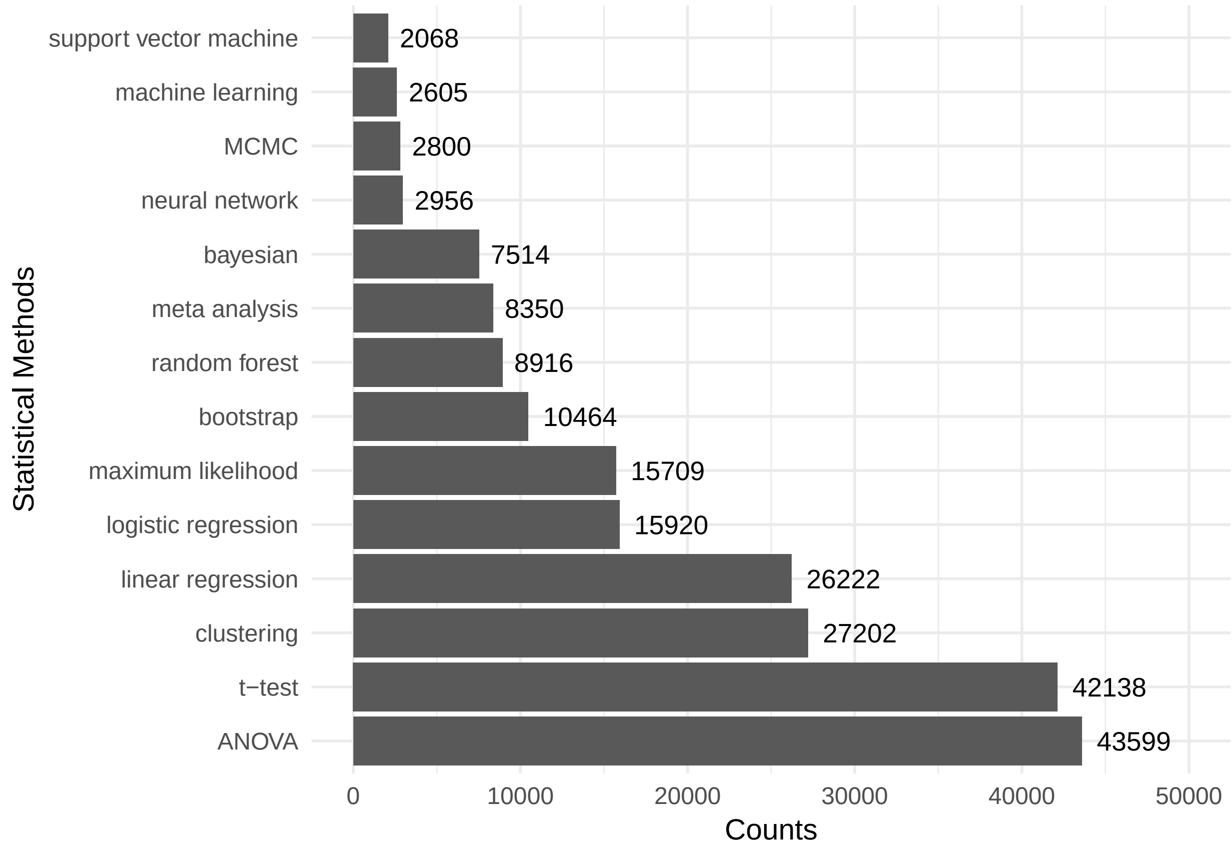
```r
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':
##
##     annotate
```

```r
# plosword(pool, vis = TRUE)

ggplot(data=df, aes(x=reorder(methods,-counts), y=counts)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label=counts), hjust=-0.2, size=3.5) +
  theme_minimal() +
  xlab("Statistical Methods") +
  ylab("Counts") +
  coord_flip(ylim=c(0,50000))
```



2. Calcuate the frequency of each method in each PLOS journal.

```r
load("data.RData")
library("stringr")
JournalName <- c("PLOS ONE", "PLOS Biology", "PLOS Medicine", "PLOS Computational Biology", "PLOS Geneti
                 "PLOS Neglected Tropical Diseases", "PLOS Pathogens")
Field <- c("General", "Biology", "Medicine", "Computayional Biology", "Genetics", "Neglected tropical di

J_Log <- sapply(JournalName, function (x) sum(str_count(LogReg_all$data$journal, x)))
J_Meta <- sapply(JournalName, function (x) sum(str_count(MetaAnal_all$data$journal, x)))
J_boot <- sapply(JournalName, function (x) sum(str_count(Bootstrap_all$data$journal, x)))
J_ANOVA <- sapply(JournalName, function (x) sum(str_count(ANOVA_all$data$journal, x)))
J_Cluster <- sapply(JournalName, function (x) sum(str_count(Cluster_all$data$journal, x)))
J_Bayes <- sapply(JournalName, function (x) sum(str_count(Bayesian_all$data$journal, x)))
J_ttest <- sapply(JournalName, function (x) sum(str_count(Ttest_all$data$journal, x)))
J_Lin <- sapply(JournalName, function (x) sum(str_count(LinReg_all$data$journal, x)))
J_Mach <- sapply(JournalName, function (x) sum(str_count(MachLrn_all$data$journal, x)))
```

```
J_MaxL <- sapply(JournalName, function (x) sum(str_count(MaxL_all$data$journal, x)))
J_Neu <- sapply(JournalName, function (x) sum(str_count(NeuNet_all$data$journal, x)))
J_Ram <- sapply(JournalName, function (x) sum(str_count(RamFor_all$data$journal, x)))
J_SVM <- sapply(JournalName, function (x) sum(str_count(SVM_all$data$journal, x)))
J_MCMC <- sapply(JournalName, function (x) sum(str_count(MCMC_all$data$journal, x)))

df_J <- data.frame(rbind(J_Log, J_Meta, J_boot, J_ANOVA, J_Cluster, J_Bayes, J_ttest, J_Lin, J_Mach, J_M
rownames(df_J) <- dic
colnames(df_J) <- Field
df <- cbind(df, df_J)
```

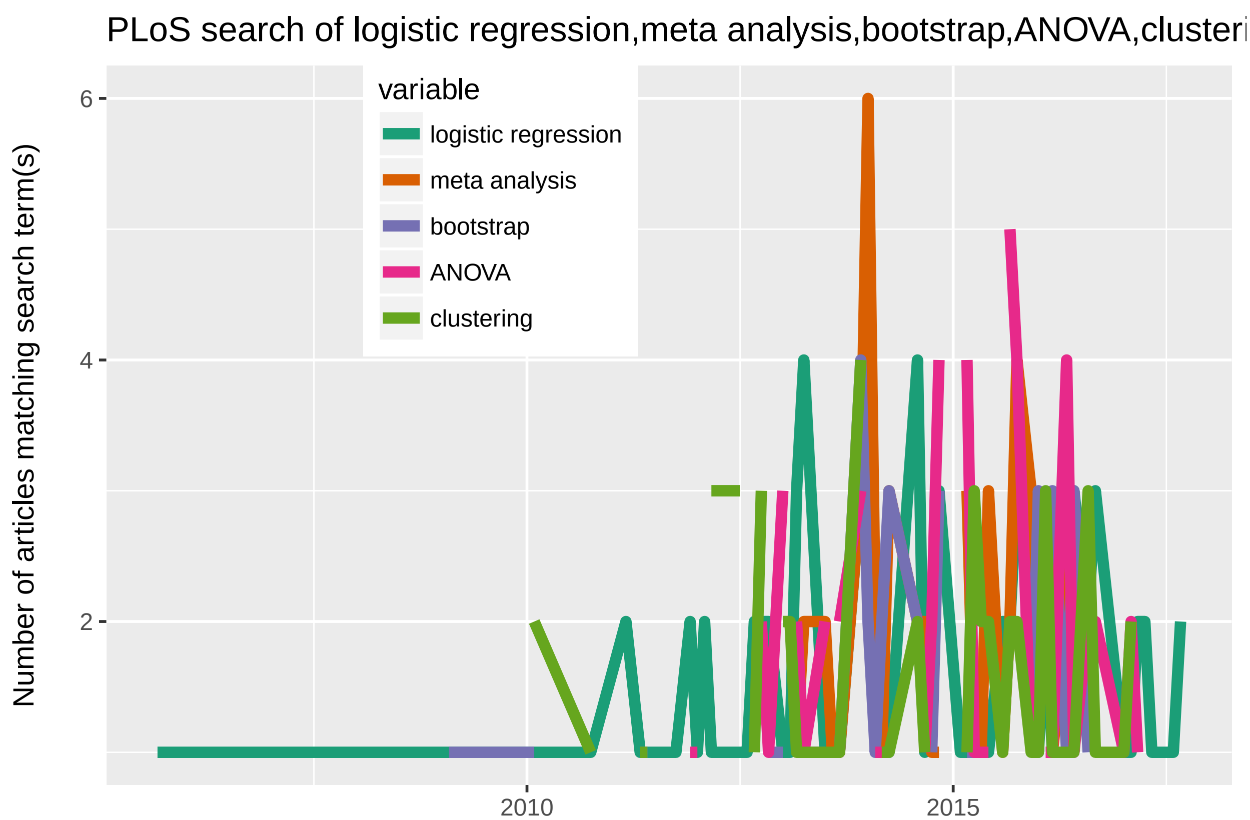
```
#install.packages("gridExtra")
#library(gridExtra)
#require(gridExtra)
library(grid)
require(grid)
mytheme <- gridExtra::ttheme_default(
  core = list(fg_params=list(cex = 0.5)),
  colhead = list(fg_params=list(cex = 0.5)),
  rowhead = list(fg_params=list(cex = 0.5)))
tb <- gridExtra::tableGrob(df_J, theme = mytheme)
grid.draw(tb)
```

	General	Biology	Medicine	Computayional Biology	Genetics	Neglected tropical diseases	Pathogen
logistic regression	6189	17	97	57	80	295	24
meta analysis	2608	23	58	47	126	83	41
bootstrap	3230	40	32	91	120	185	71
ANOVA	14283	99	24	60	278	282	376
clustering	8145	102	89	352	357	373	184
bayesian	2328	30	24	154	66	137	51
t−test	13592	31	23	79	445	297	434
linear regression	9137	79	122	244	250	306	138
machine learning	868	10	5	81	22	29	16
maximum likelihood	4823	79	40	270	191	270	101
neural network	704	31	0	184	12	7	4
random forest	3389	14	27	64	35	124	32
support vector machine	591	10	3	68	18	27	8
MCMC	895	8	4	53	22	71	22

3. Plot through time

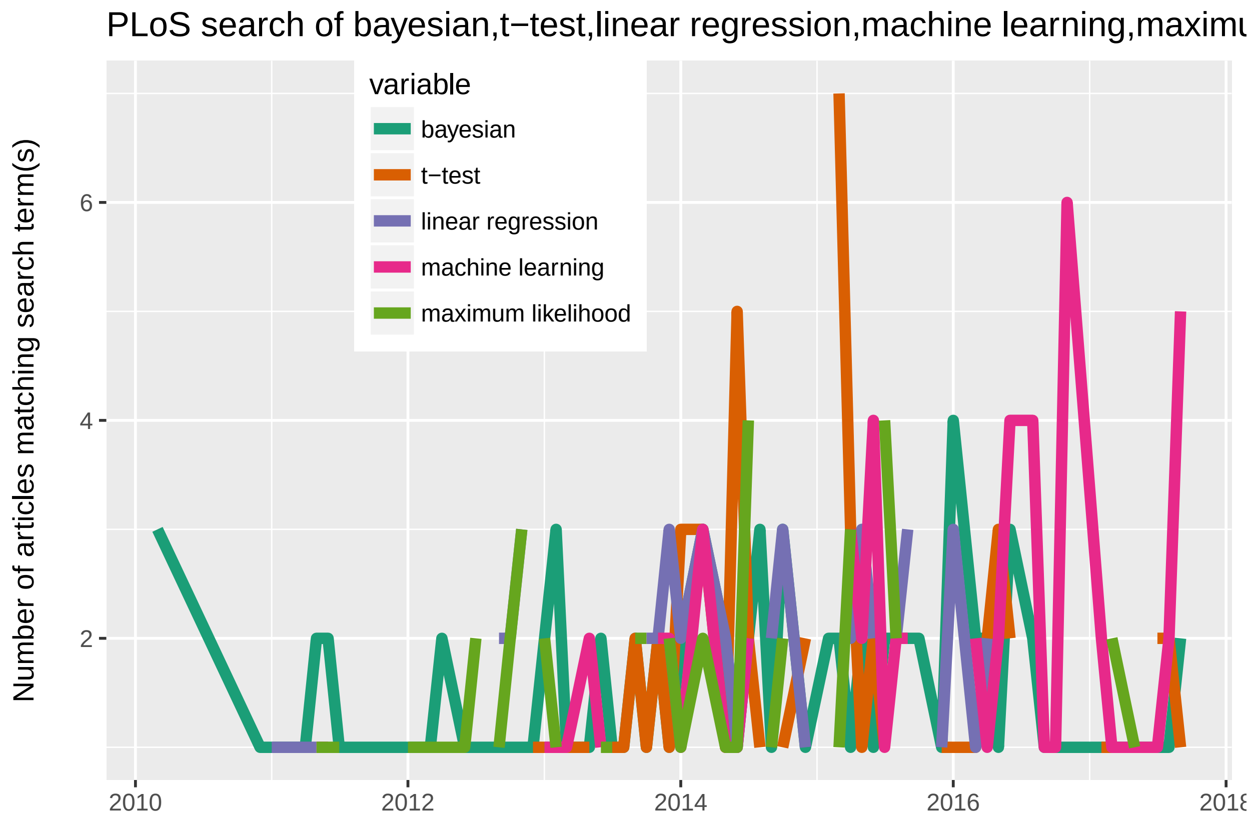
```
pool <- list("logistic regression", "meta analysis", "bootstrap", "ANOVA", "clustering", "bayesian", "t-
         "linear regression", "machine learning", "maximum likelihood", "neural network", "random forest
         "support vector machine", "MCMC")
plot_throughtime(terms = pool[1:5], limit = 100)
```

```
## Warning: Removed 31 rows containing missing values (geom_path).
```



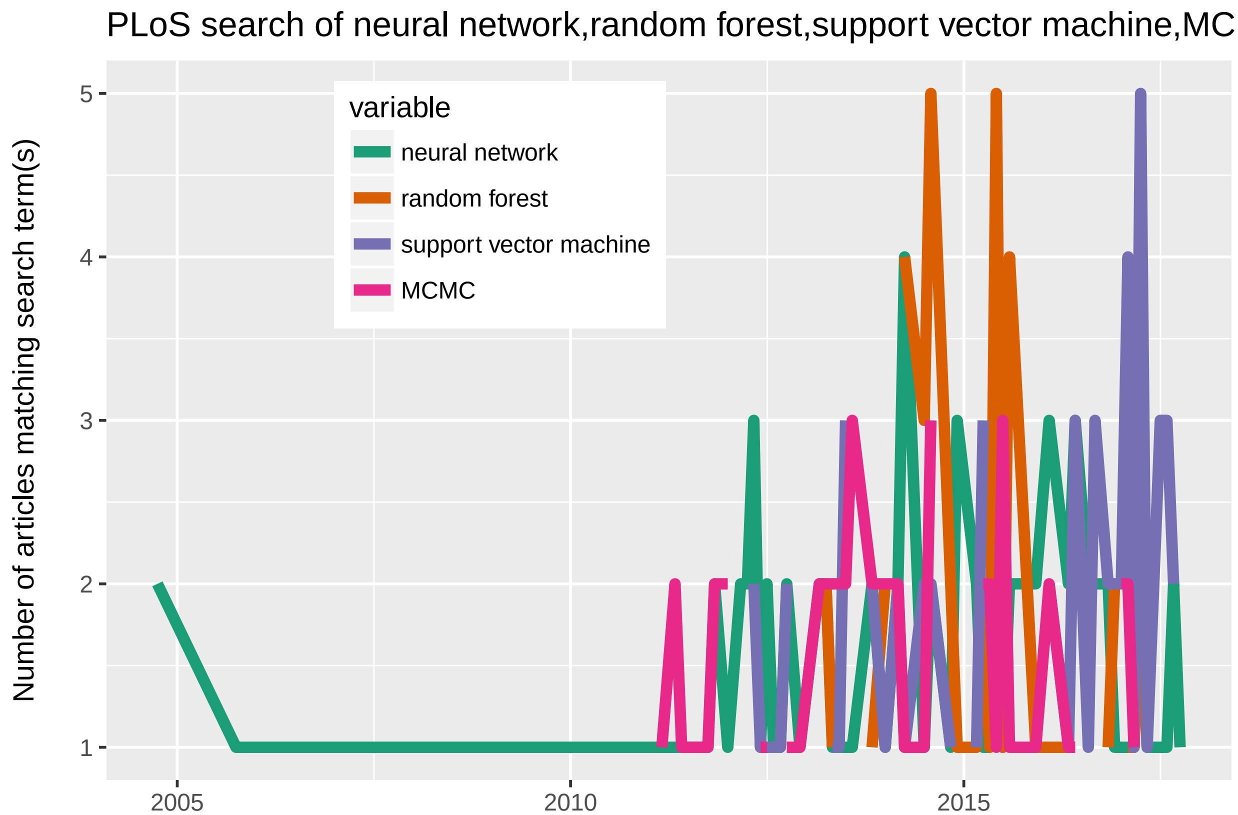

```
plot_throughtime(terms = pool[6:10], limit = 100)
```

```
## Warning: Removed 8 rows containing missing values (geom_path).
```




```
plot_throughtime(terms = pool[11:14], limit = 100)
```

```
## Warning: Removed 28 rows containing missing values (geom_path).
```




```
output <- highplos(q='linear regression', hl.fl = 'Materials and Methods')
```

```
## http://api.plos.org/search?wt=json&q=linear regression&start=0&hl=true&hl.fl=Materials and Methods
```

Reference

1. https://www.statisticallysignificantconsulting.com/Statistical-Tests.htm.