텍스트 분석을 위한 R

https://mrchypark.github.io/textR

[pdf버전] [문의하기] [의견 및 오류 신고]

스타누르기는 컨텐츠 제작자를 춤추게 합니다.

박찬엽

2018년 09월 18일

텍스트 관련 R 패키지 설치 가이드

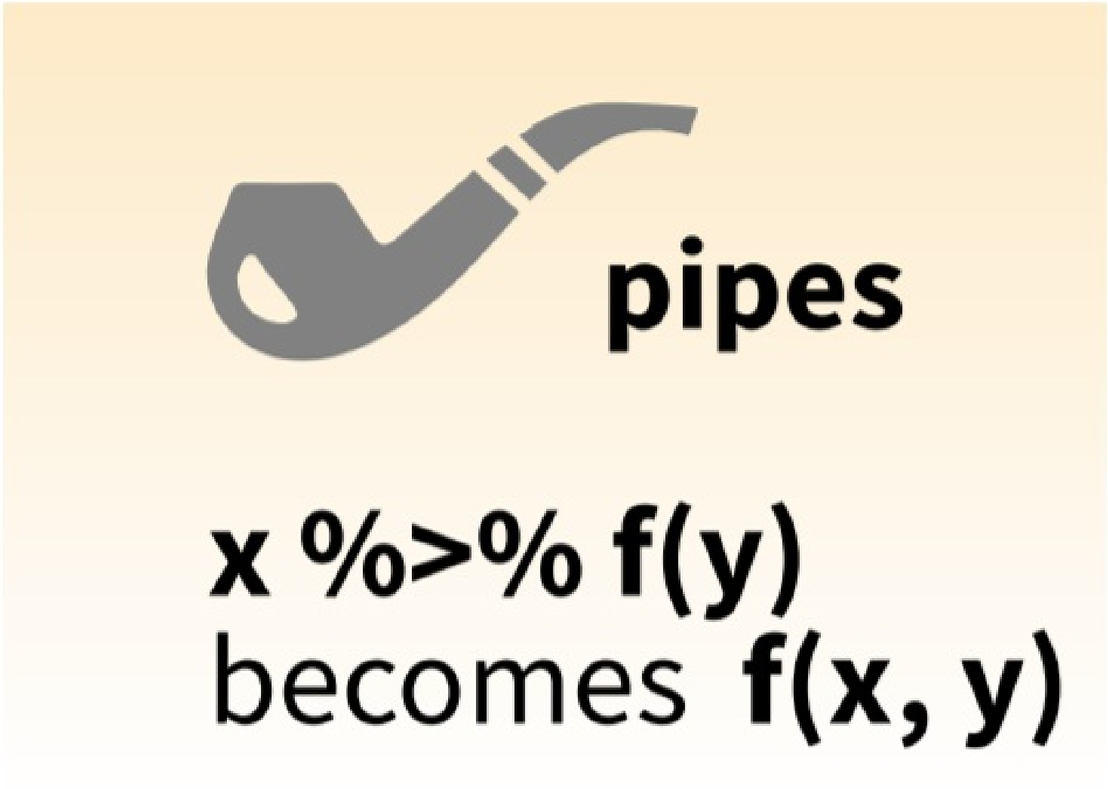
https://mrchypark.github.io/textR/installation

pdf 다운로드

* R 3.5.1 을 기준으로 작성하였습니다.

사전 지식 1

함수를 연결하는 파이프 연산자(%>%)



파이프 연산자(%>%)

함수를 중첩해서 사용할 일이 점점 빈번해 짐

```
plot(diff(log(sample(rnorm(10000,mean=10,sd=1),size=100,replace=FALSE))),col="red",type="l")
```

파이프 연산자(%>%)

함수를 중첩해서 사용할 일이 점점 빈번해 짐

```
plot(diff(log(sample(rnorm(10000,mean=10,sd=1),size=100,replace=FALSE))),col="red",type="l")
```

%>%를 사용하면

1. 생각의 순서대로 함수를 작성할 수 있음
2. 중간 변수 저장을 할 필요가 없음
3. 순서가 읽이 용이하여 기억하기 좋음

```
rnorm(10000,mean=10,sd=1) %>%
  sample(size=100,replace=FALSE) %>%
  log %>%
  diff %>%
  plot(col="red",type="l")
```

파이프 연산자(%>%)

flights 데이터에 파이프 연산자 사용예 1

```
flights %>%
  group_by(year,month,day) %>%
  summarise(delay=mean(dep_delay, na.rm = TRUE))
```

```
## # A tibble: 365 x 4
## # Groups:   year, month [?]
##     year month   day delay
##    <int> <int> <int> <dbl>
##  1  2013     1     1 11.5
##  2  2013     1     2 13.9
##  3  2013     1     3 11.0
##  4  2013     1     4  8.95
##  5  2013     1     5  5.73
##  6  2013     1     6  7.15
##  7  2013     1     7  5.42
##  8  2013     1     8  2.55
##  9  2013     1     9  2.28
## 10  2013     1    10  2.84
## # ... with 355 more rows
```

파이프 연산자(%>%)

group_by()는 filter()와도 함께 사용할 수 있음

```
popular_dests <- flights %>%
  group_by(dest) %>%
  filter(n() > 365)
popular_dests
```

```
## # A tibble: 332,577 x 19
## # Groups:   dest [77]
##      year month   day dep_time sched_dep_time dep_delay arr_time
##     <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 332,567 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

파이프 연산자(%>%)

사용할 데이터부터 순서대로 함수를 작성할 수 있는 장점

```
popular_dests %>%
  filter(arr_delay > 0) %>%
  mutate(prop_delay = arr_delay / sum(arr_delay)) %>%
  select(year:day, dest, arr_delay, prop_delay)
```

```
## # A tibble: 131,106 x 6
## # Groups:   dest [77]
##     year month   day dest  arr_delay prop_delay
##    <int> <int> <int> <chr>     <dbl>      <dbl>
##  1  2013     1     1 IAH          11  0.000111
##  2  2013     1     1 IAH          20  0.000201
##  3  2013     1     1 MIA          33  0.000235
##  4  2013     1     1 ORD          12  0.0000424
##  5  2013     1     1 FLL          19  0.0000938
##  6  2013     1     1 ORD           8  0.0000283
##  7  2013     1     1 LAX           7  0.0000344
##  8  2013     1     1 DFW          31  0.000282
##  9  2013     1     1 ATL          12  0.0000400
## 10  2013     1     1 DTW          16  0.000116
## # ... with 131,096 more rows
```

사전 지식 2

tidy data + universe

tidyverse 패키지는

1. RStudio가 개발, 관리하는 패키지
2. 공식 문서가 매우 잘 되어 있음
3. 사용자층이 두터워 영어로 검색하면 많은 질답을 찾을 수 있음
4. 커뮤니티 설명글도 매우 많음
5. 6개의 핵심 패키지 포함 23가지 패키지로 이루어진 메타 패키지
6. tidy data 라는 사상과 파이프 연산자로 대동단결
7. 사상에 영감을 받아 맞춰서 제작하는 개인 패키지가 많음 (ex> tidyquant, tidytext 등)

```
if (!requireNamespace("tidyverse")){
  install.packages("tidyverse")}
library(tidyverse)
```

tidy data 란

1. Hadley Wickham 2. 고감자님의 블로그 3. 헬로우데이터과학

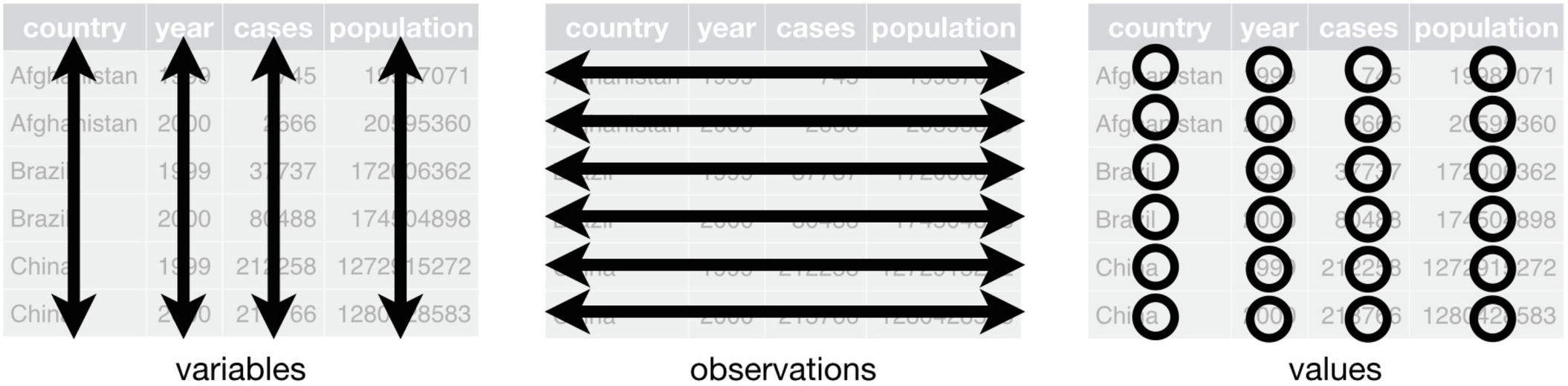
1.1 Each variable forms a column.
1.2 각 변수는 개별의 열(column)으로 존재한다.
1.3 각 열에는 개별 속성이 들어간다.

2.1 Each observation forms a row.
2.2 각 관측치는 행(row)를 구성한다.
2.3 각 행에는 개별 관찰 항목이 들어간다.

3.1 Each type of observational unit forms a table.
3.2 각 테이블은 단 하나의 관측기준에 의해서 조직된 데이터를 저장한다.
3.3 각 테이블에는 단일 유형의 데이터가 들어간다.

* 출처 : 금융데이터 분석을 위한 R 입문

tidy data 란



* 출처 : Garrett Grolemund의 Data Science with R 블로그

long form vs wide form

long form

1. 컴퓨터가 계산하기 좋은 모양
2. tidy data의 요건을 충족
3. tidyverse의 패키지 대부분의 입력 형태

wide form

1. 사람이 눈으로 보기 좋은 모양
2. 2개 변수에 대한 값만 확인 가능
3. dashboard 형이라고도 하며 조인 등 연산이 어려움

tidy text data 란

- a table with one-token-per-row
- 한 행(row)에 한 토큰(token)으로 테이블을 구성해야 한다.

tidy text data 란

- a table with one-token-per-row
- 한 행(row)에 한 토큰(token)으로 테이블을 구성해야 한다.

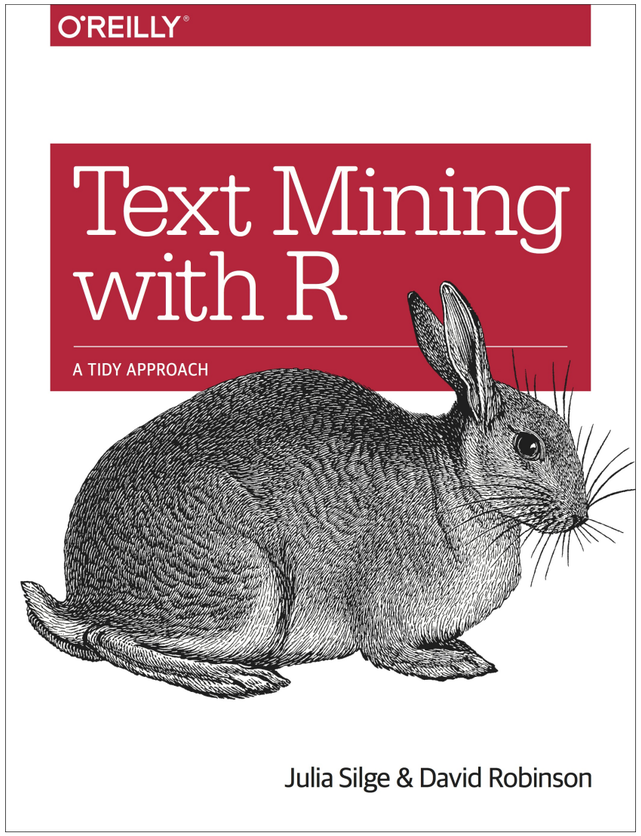
그럼 Token 이란?

글자 중 의미를 가진 단위를 총칭.
tokenization은 가지고 있는 텍스트 자원을 token 단위로 나누는 것을 뜻함.
ex> 자소(자음, 모음), 음소(글자), 형태소, 단어, n-gram 등

tidytext 패키지 소개

- 한 행(row)에 한 토큰(token)으로 테이블을 구성하기 위한 패키지
- 파이프 연산자를 지원
- 여러 가지 token과 tm 패키지와의 호환 기능을 제공
- 자세히 소개하는 온라인 사이트(영문)

```
if(!requireNamespace("tidytext")){
  install.packages("tidytext")
}
library(tidytext)
```



데이터 패키지 소개

presidentSpeechKr

대통령 기록 연구실의 대통령 연설문을 제공

```
if(!requireNamespace("presidentSpeechKr")){
  remotes::install_github("presidentSpeechKr")
}
library(presidentSpeechKr)
```

대통령 조건 확인

```
get_president()
```

```
##  [1] "이승만" "윤보선" "박정희" "최규하" "전두환" "노태우" "김영삼"
##  [8] "김대중" "노무현" "이명박"
```

대통령 조건 확인

```
get_president()
```

```
##  [1] "이승만" "윤보선" "박정희" "최규하" "전두환" "노태우" "김영삼"
##  [8] "김대중" "노무현" "이명박"
```

연설 분야 조건 확인

```
get_field()
```

```
##  [1] "국정전반"          "정치/사회"         "산업/경제"         "외교/통상"
##  [5] "국방"              "과학기술정보"      "교육"              "문화/체육/관광"
##  [9] "환경"              "기타"
```

연설 유형 확인

```
get_event()
```

```
##  [1] "취임사"       "신년사"       "국회연설"     "기념사"       "만찬사"
##  [6] "환영사"       "치사"         "성명/담화문" "라디오연설"   "기타"
```

연설 유형 확인

```
get_event()
```

```
##  [1] "취임사"       "신년사"       "국회연설"     "기념사"       "만찬사"
##  [6] "환영사"       "치사"         "성명/담화문" "라디오연설"   "기타"
```

연설 리스트 데이터

```
data(spidx)
str(spidx)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    6681 obs. of  6 variables:
##  $ president: chr  "이승만" "이승만" "이승만" "이승만" ...
##  $ field    : chr  "기타" "국정전반" "정치/사회" "국정전반" ...
##  $ event    : chr  "성명/담화문" "취임사" "성명/담화문" "기타" ...
##  $ title    : chr  "학생제군에게" "대통령 취임사(大統領就任辭) " "민족이 원하는 길을 따를 결심, 국무총리 인준 부결
##  $ date     : chr  "1948" "1948.07.24" "1948.07.29" "1948.08.09" ...
##  $ link     : chr  "http://www.pa.go.kr/research/contents/speech/index.jsp?spMode=view&catid=c_pa020
```

대통령 조건 연설 검색

```
library(dplyr)
spidx %>%
  filter(president == "윤보선")
```

```
## # A tibble: 3 x 6
##   president field  event  title                date   link
##   <chr>     <chr>  <chr>  <chr>                <chr>  <chr>
## 1 윤보선    국정전반~ 취임사 제2대 윤보선 대통령 취임사~ 1960.~ http://www.pa.go.kr/res~
## 2 윤보선    기타   기타   "윤보선 대통령 부산연설 \~ 1960.~ http://www.pa.go.kr/res~
## 3 윤보선    기타   기타   "윤보선 대통령 대구연설 \~ 1960.~ http://www.pa.go.kr/res~
```

연설문 텍스트 가져오기

```
tar <-
  spidx %>%
  filter(president == "윤보선") %>%
  select(link) %>%
  top_n(1)
```

```
## Selecting by link
```

```
get_speech(tar)
```

```
## # A tibble: 1 x 9
##   title    date   president place field event source paragraph content
##   <chr>    <chr>  <chr>     <chr> <chr> <chr> <chr>       <int> <chr>
## 1 "윤보선 대통~ 1960.~ 윤보선    지역  기타  기타  ""              1 "영남지방의 한해 ~
```

연습문제

1. presidentSpeechKr 패키지에서 검색할 수 있는 대통령은 총 몇명인가요?
2. **윤보선** 대통령과 **박정희** 대통령은 각각 몇 개의 연설문이 있나요?
3. `nchar()` 함수는 글자수를 세주는 함수입니다. **최규하** 대통령의 취임사는 총 몇 글자 인가요?

단어 단위로 잘라보자!

unnest_tokens() 함수

기본값인 단어 단위(특수문자 제거, 띄어쓰기 기준) token으로 동작.

```
get_speech(tar) %>%
  select(president, content) %>%
  unnest_tokens(word, content)
```

```
## # A tibble: 100 x 2
##    president word
##    <chr>     <chr>
##  1 윤보선    영남지방의
##  2 윤보선    한해
##  3 윤보선    상황을
##  4 윤보선    시찰차
##  5 윤보선    십오일
##  6 윤보선    상오
##  7 윤보선    구시
##  8 윤보선    삼십분
##  9 윤보선    특별기편으로
## 10 윤보선    대구에
## # ... with 90 more rows
```

unnest_tokens() 함수 설명

텍스트 데이터를 token 단위로 풀어내는 함수

```
unnest_tokens(
  tbl = 텍스트 데이터,          # 다루고자 하는 텍스트 데이터 객체
  output = 결과열의 이름,       # token화의 결과가 작성될 열의 이름
  input = 목표 텍스트 열,       # 텍스트 데이터 객체 내의 텍스트 열
  token = "word",              # 기본값(띄어쓰기 단위)이 있어 생략 가능
  ...                          # 기타 옵션들
  )
```

```r
# 연설문 중 1개를 가져와서
get_speech(tar) %>%
  # 대통령 컬럼과 연설문 컬럼만 선택한 후
  select(president, content) %>%
  # 연설문 컬럼을 word 단위로 쪼갠 결과물을 word라는 컬럼으로 출력
  unnest_tokens(word, content)
```

```
## # A tibble: 100 x 2
##    president word
##    <chr>     <chr>
##  1 윤보선    영남지방의
##  2 윤보선    한해
##  3 윤보선    상황을
##  4 윤보선    시찰차
##  5 윤보선    십오일
##  6 윤보선    상오
##  7 윤보선    구시
##  8 윤보선    삼십분
##  9 윤보선    특별기편으로
## 10 윤보선    대구에
## # ... with 90 more rows
```

띄어쓰기 단위로 나눴을 때 문제점

하다가 몇 가지 단어가 되는지

Korean verb '하다' Conjugated

regular verb

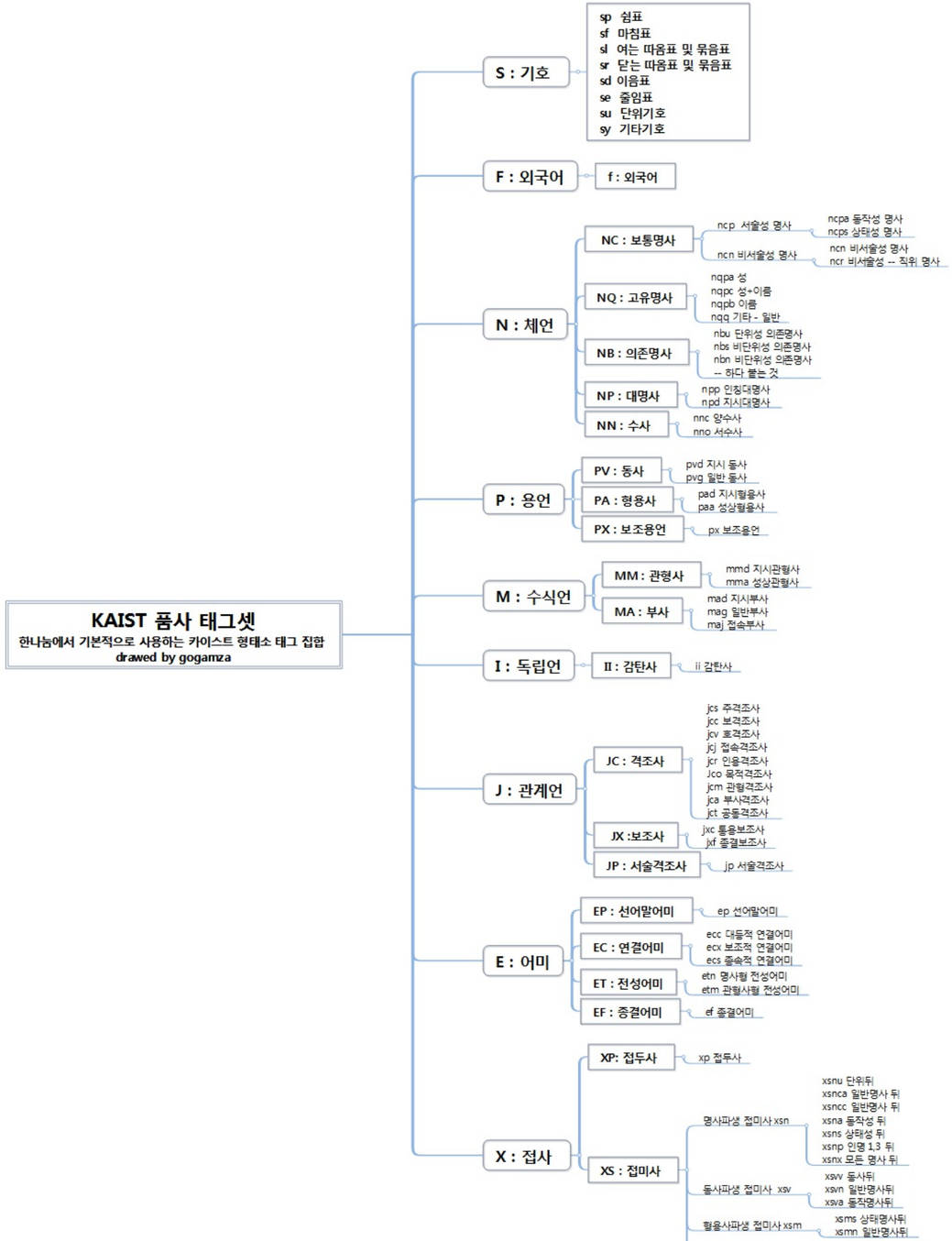
Form		Conjugation
base	하	ha
base2	하	ha
base3	하	ha
declarative present informal low	해	hae
declarative present informal high	해요	hae-yo
declarative present formal low	한다	han-da
declarative present formal high	합니다	hab-ni-da
past base	했	haess
declarative past informal low	했어	haess-eo
declarative past informal high	했어요	haess-eo-yo
declarative past formal low	했다	haess-da
declarative past formal high	했습니다	haess-seub-ni-da
future base	할	hal
declarative future informal low	할 거야	hal geo-ya
declarative future informal high	할 거예요	hal geo-ye-yo
declarative future formal low	할 거다	hal geo-da
declarative future formal high	할 겁니다	hal geob-ni-da
declarative future conditional informal low	하겠어	ha-gess-eo
declarative future conditional informal high	하겠어요	ha-gess-eo-yo
declarative future conditional formal low	하겠다	ha-gess-da
declarative future conditional formal high	하겠습니다	ha-gess-seub-ni-da
inquisitive present informal low	해?	hae?
inquisitive present informal high	해요?	hae-yo?
inquisitive present formal low	하니?	ha-ni?
inquisitive present formal high	합니까?	hab-ni-gga?
inquisitive past informal low	했어?	haess-eo?
inquisitive past informal high	했어요?	haess-eo-yo?
inquisitive past formal low	했니?	haess-ni?
inquisitive past formal high	했습니까?	haess-seub-ni-gga?
imperative present informal low	해	hae
imperative present informal high	하세요	ha-se-yo
imperative present formal low	해라	hae-ra
imperative present formal high	하십시오	ha-sib-si-o
propositive present informal low	해	hae
propositive present informal high	해요	hae-yo
propositive present formal low	하자	ha-ja
propositive present formal high	합시다	hab-si-da
connective if	하면	ha-myeon
connective and	하고	ha-go
nominal ing	함	ham

한글의 특징 형태소

형태소란 : 의미를 가지는 최소 단위

> 철수가 밥을 먹었다.

```
## $철수가
## [1] "철수/ncpa+가/jcc" "철수/ncpa+가/jcs"
##
## $밥을
## [1] "밥/ncn+을/jco"  "밥/ncpa+을/jco" "밥/ncps+을/jco"
##
## $먹었다
## [1] "먹/pvg+었/ep+다/ef"
##
## $.
## [1] "./sf" "./sy"
```



크게 보기

여러 체계의 형태소 품사

R의 대표적인 형태소 분석기

RcppMeCab

1. 일본어 형태소 분석기인 mecab 기반
2. C++ 로 작성하여 속도가 매우 빠름
3. 일본어, 중국어 등도 사용 가능
4. 형태소 분석 함수를 제공
5. 띄어쓰기에 덜 민감함

KoNLP

1. 가장 유명한 형태소 분석기
2. java로 작성된 한나눔 분석기 기반
3. 우리샘, NiaDIC 등 자체 사전
4. 텍스트 분석을 위한 기능들을 제공
5. 친절한 설명서

RcppMeCab 설치 확인

RcppMeCab 실행

```
> library(RcppMeCab)
> pos("롯데마트가 판매하고 있는 흑마늘 양념 치킨이 논란이 되고 있다.")
```

```
## $`롯데마트가 판매하고 있는 흑마늘 양념 치킨이 논란이 되고 있다.`
##  [1] "롯데마트/NNP" "가/JKS"       "판매/NNG"     "하/XSV"
##  [5] "고/EC"        "있/VX"        "는/ETM"       "흑/NNG"
##  [9] "마늘/NNG"     "양념/NNG"     "치킨/NNG"     "이/JKS"
## [13] "논란/NNG"     "이/JKS"       "되/VV"        "고/EC"
## [17] "있/VX"        "다/EF"        "./SF"
```

KoNLP 설치 확인

KoNLP 실행

```
library(KoNLP)
SimplePos09("롯데마트가 판매하고 있는 흑마늘 양념 치킨이 논란이 되고 있다.")
```

```
## $롯데마트가
## [1] "롯데마트가/N"
##
## $판매하고
## [1] "판매/N+하고/J"
##
## $있는
## [1] "있/P+는/E"
##
## $흑마늘
## [1] "흑마늘/N"
##
## $양념
## [1] "양념/N"
##
## $치킨이
## [1] "치킨/N+이/J"
##
## $논란이
## [1] "논란/N+이/J"
##
## $되고
```

연습문제

1. **김영삼** 대통령의 첫 국무회의 연설문을 띄어쓰기 단위로 자르면 총 몇 단어인가요?
2. **노태우** 대통령의 취임사를 `RcppMeCab` 패키지로 형태소 분석한 결과를 출력하세요.
3. **김대중** 대통령의 취임사를 `KoNLP` 패키지의 `SimplePos09()` 함수로 형태소 분석한 결과를 출력하세요.

띄어쓰기 문제

MeCab은 띄어쓰기가 없는 일본어 기반의 분석기이므로 띄어쓰기가 잘 안되어 있는 상태에 영향을 덜받음.
KoNLP(한나눔)는 띄어쓰기가 중요한 판단 정보로 활용되어 띄어쓰기가 이상할 경우 성능에 영향을 많이 받음.

```
> library(RcppMeCab)
> pos("롯데마트가판매하고있는흑마늘양념치킨이논란이도
```

```
## $`롯데마트가판매하고있는흑마늘양념치킨이논란이되고있다.`
##  [1] "롯데마트/NNP" "가/JKS"       "판매/NNG"
##  [4] "하/XSV"       "고/EC"        "있/VX"
##  [7] "는/ETM"       "흑마/NNG"     "늘/MAG"
## [10] "양념치킨/NNP" "이/JKS"       "논란/NNG"
## [13] "이/JKS"       "되/VV"        "고/EC"
## [16] "있/VX"        "다/EF"        "./SF"
```

```
library(KoNLP)
SimplePos09("롯데마트가판매하고있는흑마늘양념치킨이
```

```
## $롯데마트가판매하고있는흑마늘양념치킨이논란이되고있다
## [1] "롯데마트가판매하고있는흑마늘양념치킨이논란이되고있다/N
##
## $.
## [1] "./S"
```

KoSpacing 패키지

딥러닝 한글 띄어쓰기 패키지.

```
library(KoSpacing)
spacing("김형호영화시장분석가는'1987'의네이버영화정보네티즌10점평에서언급된단어들을지난해12월27일부터올해1월10
```

```
## [1] "김형호 영화시장 분석가는 '1987'의 네이버 영화 정보 네티즌 10점 평에서 언급된 단어들을 지난해 12월 27일부터 올해
```

띄어쓰기 적용후 결과

```
library(KoSpacing)
library(KoNLP)
set_env()
```

```
## loaded KoSpacing model!
```

```
text <- "롯데마트가판매하고있는흑마늘양념치킨이논란이되고있다."
text <- spacing(text)
SimplePos09(text)
```

```
## $롯데마트가
## [1] "롯데마트가/N"
##
## $판매하고
## [1] "판매/N+하고/J"
##
## $있는
## [1] "있/P+는/E"
##
## $흑마늘
## [1] "흑마늘/N"
##
## $양념치킨이
## [1] "양념치킨/N+이/J"
```