IT 취업 트렌드 현황 분석

2조 데이터 레스토랑

성민제, 배석란, 임세진, 임소빈, 손예정

목차

[프로젝트 개요 및 목표 3](#_Toc103260306)

[크롤링 데이터 전처리 8](#_Toc103260307)

[키워드 시각화 10](#_Toc103260308)

[설문조사 분석 13](#_Toc103260309)

[연관분석 17](#_Toc103260310)

[연관 분석의 기본 개념 17](#_Toc103260311)

[연관 분석 해석 22](#_Toc103260312)

[LDA(Latent Dirichlet Allocation) 과정 23](#_Toc103260313)

[LDA 실시 23](#_Toc103260314)

[tf-idf 보기 23](#_Toc103260315)

[해석 27](#_Toc103260316)

[LDA 실시 27](#_Toc103260317)

[LDA해석 30](#_Toc103260318)

## 프로젝트 개요 및 목표

- 경기도 클라우드, 빅데이터 기반 인공지능 양성자 과정 2기 교육생 모두 각자 해당 분야의 프로그래밍 언어, 프레임워크, 기술, 포트폴리오 등 자신이 원하는 분야로 취업하기 위해 많은 노력을 하고있음 특히 IT분야는 다른 분야의 취업시장과 다르게, 프로그래밍 실력, 포트폴리오 등을 기업들이 주 평가사항으로 다루고 있음. IT직무마다 다른 프로그래밍 자격사항을 요구하고 있으므로 진행하고 있는 구직활동이 실제로 기업들이 원하는 스펙, 활동인지 분석해보고 더 효과적인 취업준비를 위한 분석을 실시함.

프로젝트 목표

- 탐색적 분석을 통해 수집된 설문조사, 구인공고 크롤링, 산업통계 자료를 분석

- 인터뷰 자료를 분석하여 취업 시 필요 요건들 확인

- 구인공고 크롤링 자료를 활용하여 언어모델링 실시

- 구인자 맞춤 프로그래밍 언어 추천

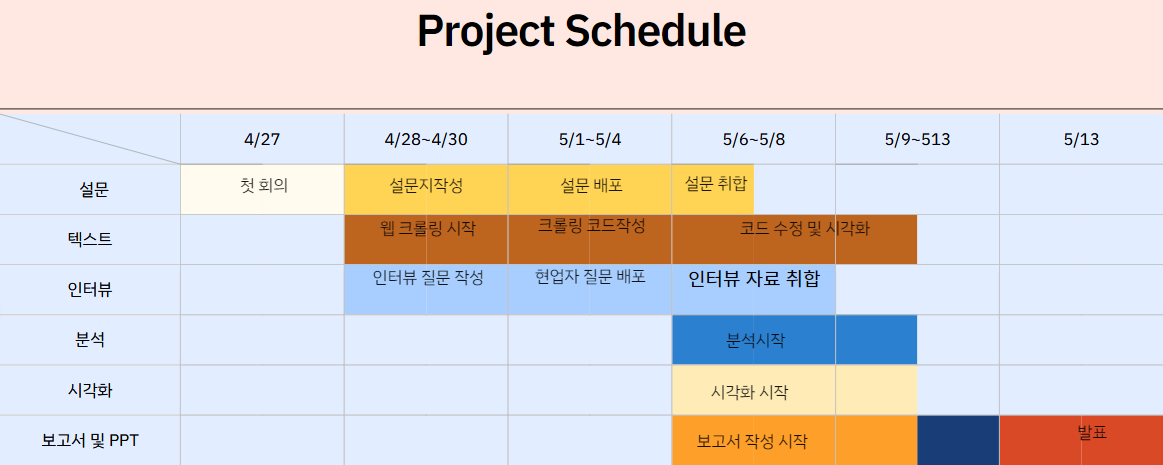
기간

- 2022.05.09. ~ 2022.05.13. (총 7일)

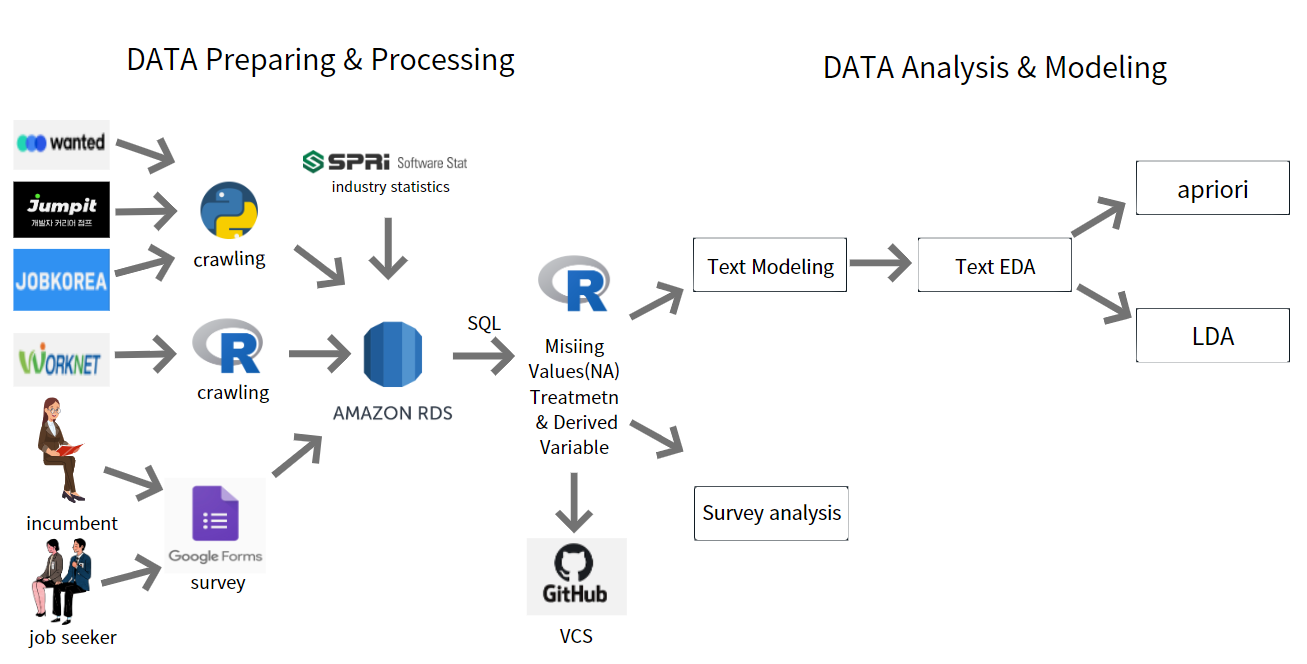
팀원 업무분담



프로젝트 활용



데이터 개요 (활용데이터)

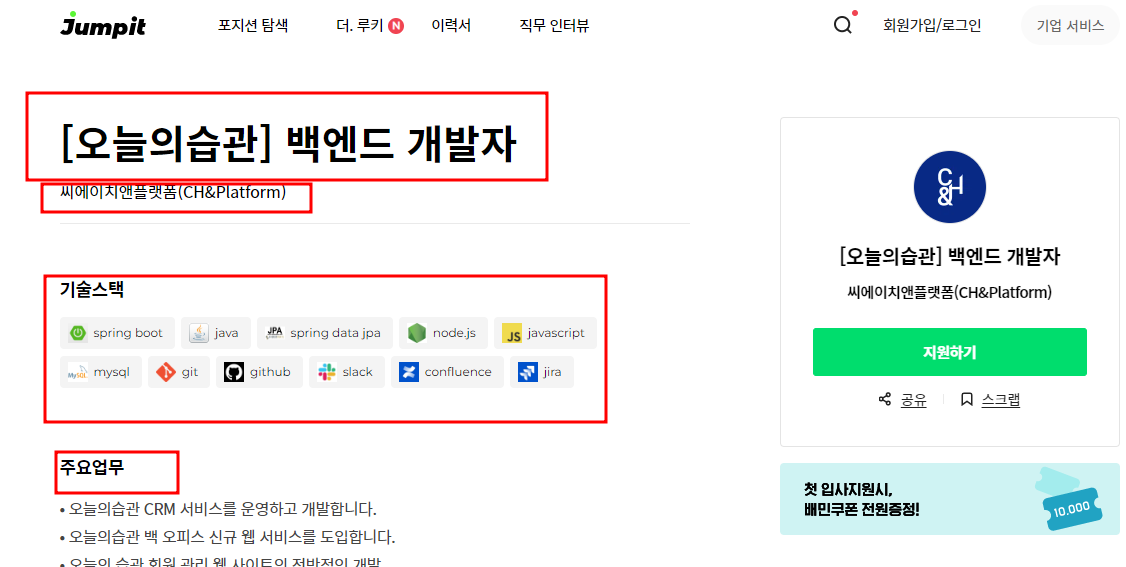


1. 자료소개

1) 산업별 통계

(1) SWSTAT(소프트웨어통계포털)을 이용하여 소프트웨어 산업의 시장 매출, 인력 등의 현황통계를 활용하여 전체적인 산업분석을 실시함.

2) 구인공고 데이터



(1) 데이터는 구인공고 사이트(점핏, 잡코리아, 워크넷, 원티드)의 IT관련 구인공고를 참고하여 크롤링을 실시함. 각 사이트 별로 API를 활용해서 자료를 얻을 수 없어, 직접 크롤링을 하여 해당 사이트의 구인공고 자료를 받았으며, 2022년 5월 2일 ~ 2022년 5월 6일까지 올라온 구인공고를 사용함. 회사명, 구인공고제목 기준 중복데이터를 제거하였으며 15588건의 데이터를 찾을 수 있었음 해당 데이터는 AWS RDBMS에 적재하여 데이터 분석시 AWS RDMBS에서 SQL로 해당 데이터를 불러와서 활용하였음.

|  |  |  |  |
| --- | --- | --- | --- |
| **사이트** | **데이터 개수** | **활용 언어 및 프레임워크** | **기간** |
| 점핏 | 3406 | R, rvst, stringr | 2022년 5월 2일 ~ 5월 6일 |
| 워크넷 | 66 |
| 잡코리아 | 11153 | Python, BeautifulSoup Selenium, pandas |
| 원티드 | 1755 |
| 총 합계 | 16380 | - |
| 중복데이터 | 792 | - |
| 중복제거 총 합계 | 15588 | - |

3) 설문조사 데이터

(1) 구직자 설문조사 데이터

- 구글플랫폼 설문지를 제작하고 링크를 활용하여 인터넷 설문을 실시하였음, IT 교육에 참여하는 교육생, IT취업준비 관련 카카오톡 단톡방 등을 활용하여 해당데이터들을 수집하고, AWS RDMBS에 적재하고 데이터 분석 시 AWS RDMBS에서 SQL로 해당 데이터를 불러와서 활용하였음.

(2) 현직자 설문조사 데이터

- 한국품질재단에서 실시한 현직자 설문조사 자료를 협조 받아 분석에 사용하였음

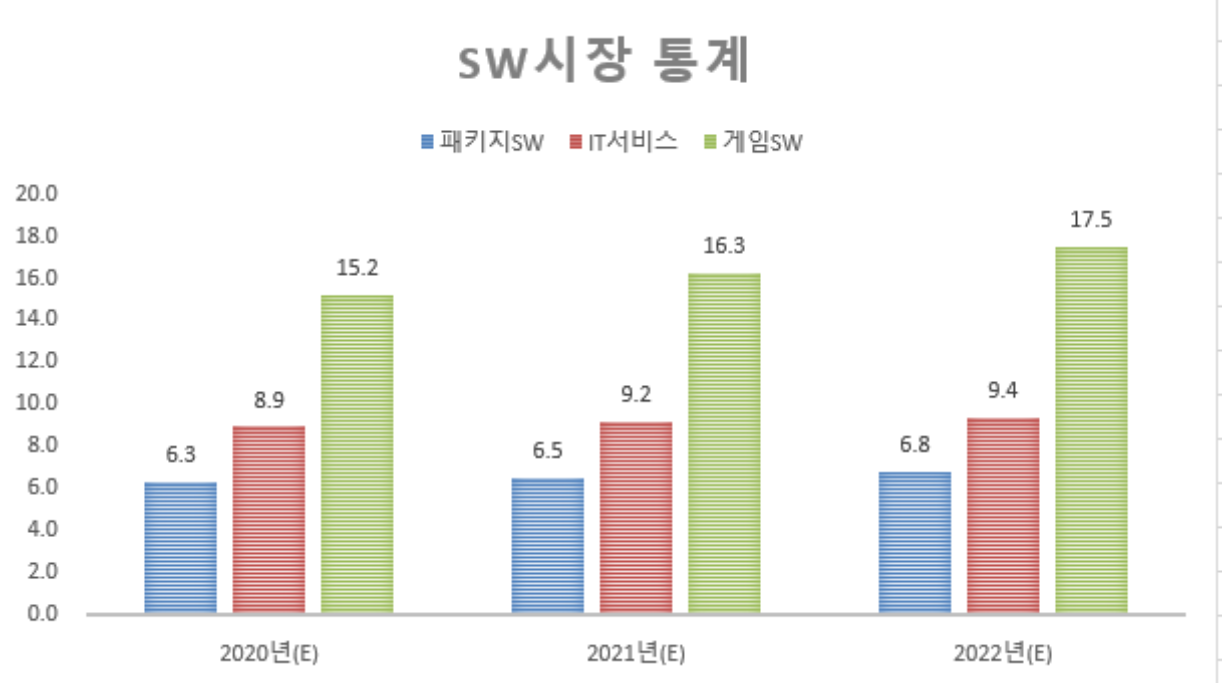
|  |  |  |
| --- | --- | --- |
| **항목** | **현직자** | **구직자** |
| 기간 | 2월 ~ 3월 (1개월) | 5월 2일 ~ 5월 6일 (1주일) |
| 조사대상 | 현직자 | IT취준생 |
| 조사방법 | 인공지능협회 소속된 기업들 현업자 대상 인터넷 조사 실시 | IT 교육 참여하는 교육생, IT 취업준비 카카오톡 단톡방 활용 |
| 데이터 수 | 87 | 70 |

4) 인터뷰

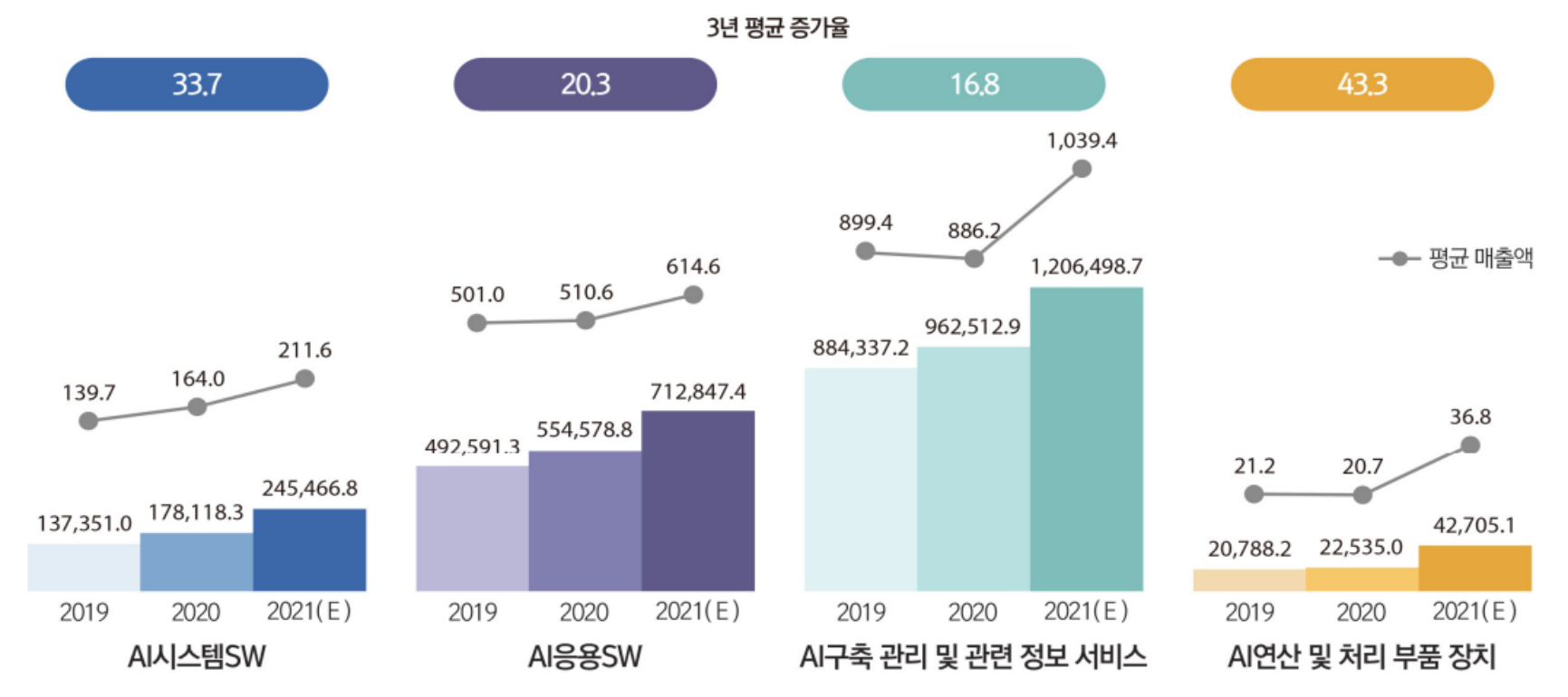
- 앱개발자, 데이터사이언티스트, 프론트개발자, 백엔드개발자, 풀스택개발자 5명의 현직자를 대상으로 서면으로 인터뷰를 실시하였음.

## 소프트웨어 시장 현황

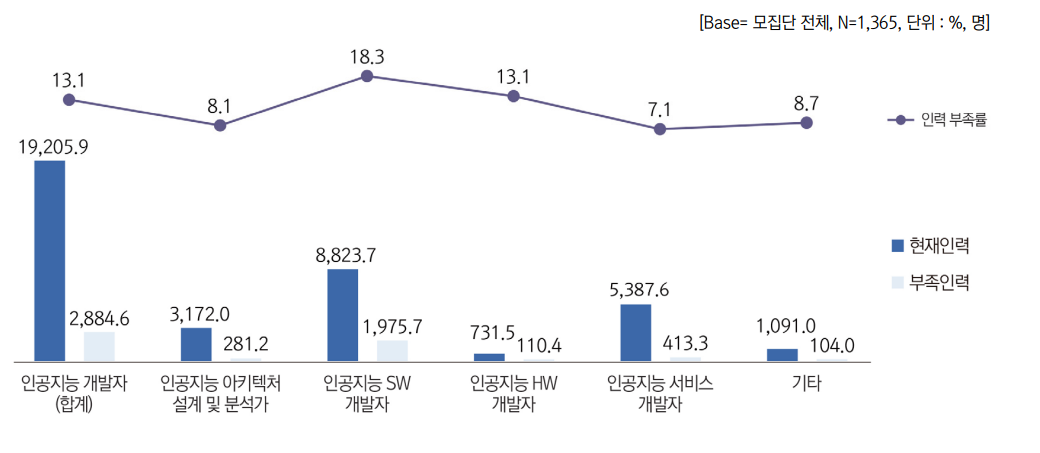
## 



소프트웨어 정책 연구소의 자료에 따르면 상위 그래프와 같이 각 패키지, IT, 게임 소프트웨어 서비스 시장 규모는 2020년 기준 약 30.4조 원에 달하는 것으로 나타났음. 패키지 소프트웨어의 시장 규모는 약 6.3조 원, IT서비스 부문의 규모는 약 9.2조 원, 게임 소프트웨어 서비스 시장 규모는 15.2조 원으로 집계되었음. 또한, 계속 성장할 것으로 보여짐.



2021년 인공지능산업 실태 조사에 따르면 AI시스템, AI응용, AI구축 관리 및 관련 정보 소프트웨어 및 AI연산 및 처리 부품 장치 부문에서 3년 연속 증가한 것으로 나타났음. 세부적으로는 AI연산 및 처리 부품 장치 부문 시장이 가장 컸음.



직업별 인공지능 채용 예정 및 부족 인력 현황에서는 전체 부문에서 인공지능 개발자가 약 13%의 부족률을 보임. 전체적으로 인력 부족이 발생하고 있는 것으로 나타남.

## 크롤링 데이터 전처리

다음과 같은 원데이터에서

## New names:  
## \* `` -> `...1`

## # A tibble: 6 x 3  
## coname title career  
## <chr> <chr> <chr>   
## 1 "[<a href=\"/company/221\">아이지에이웍스</a>]" [<h1>머신러닝 엔지~ [<dd>~  
## 2 "[<a href=\"/company/588\">미식의시대</a>]" [<h1>신한은행 배달~ [<dd>~

CSS 태그가 함께 붙은 크롤링 데이터를 정제함.

## # A tibble: 6 x 3  
## coname title career   
## <chr> <chr> <chr>   
## 1 아이지에이웍스 머신러닝 엔지니어 (디파이너리) 신입 - ~  
## 2 미식의시대 신한은행 배달앱 '땡겨요' 개인화 큐레이션 개발 총괄 5 - 10년

파싱이 완료된 데이터를 통합한 후, 변수 전처리를 시행함.

(원데이터)

## # A tibble: 6 x 5  
## coname keyword qualif prefer career  
## <chr> <chr> <chr> <chr> <chr>   
## 1 포디리플레이코리아 opencv, c, c++, tensorflow, pytorch,~ "Open~ "비전~ 3 - 1~  
## 2 카비 deeplearning, machinelearning, c, c+~ "관련~ "영상~ 3 - 1~

회사 위치 컬럼을 시/구 단위로 나눈 후, 서울을 서울특별시로 정규화하였음.

## # A tibble: 6 x 2  
## coname loc   
## <chr> <chr>   
## 1 포디리플레이코리아 서울특별시 강남구  
## 2 카비 서울특별시 금천구

경력 컬럼을 정제 후, 경력 무관은 0으로, 신입은 1로, 경력직은 2로 할당하여 새롭게 저장해 줌.

## # A tibble: 6 x 3  
## coname career career2  
## <chr> <chr> <dbl>  
## 1 포디리플레이코리아 3-10 2  
## 2 카비 3-10 2

자격요건/우대조건 컬럼을 정제해줌.

## # A tibble: 6 x 1  
## qualif   
## <chr>   
## 1 "OpenCV, C/C++, Python\r\r\n유관업무 3년 이상 경력"   
## 2 "관련 업무 경력 3년 이상\r\r\n컴퓨터공학 또는 영상처리 관련학과 석사(대학원) ~

먼저 소문자로 변환 후, 제어 문자를 삭제함.  
마크업 언어를 제거해주고, 특수문자 및 숫자를 삭제함.  
그리고 딥러닝/dl 과 같이 뜻이 동일한 단어들을 영어로 변환해 통일함.

## # A tibble: 6 x 1  
## qualif   
## <chr>   
## 1 opencv, c c++, python 유관업무 년 이상 경력   
## 2 관련 업무 경력 년 이상 컴퓨터공학 또는 영상처리 관련학과 석사, 대학원 이상~

정제된 자격요건/우대조건 컬럼을 토큰화 해줌.

## Warning in readLines("C:/R/R/stack.txt", encoding = "UTF-8"): 'C:/R/R/  
## stack.txt'에서 불완전한 마지막 행이 발견되었습니다

## 375826 words dictionary was built.

## # A tibble: 6 x 1  
## qualif2   
## <chr>   
## 1 opencv, c, c++, python, 유관, 업무, 년, 이상, 경, 력   
## 2 관련, 업무, 경력, 년, 이상, 컴퓨터공학, 영상처리, 관련학과, 석사, 대학원, 학~

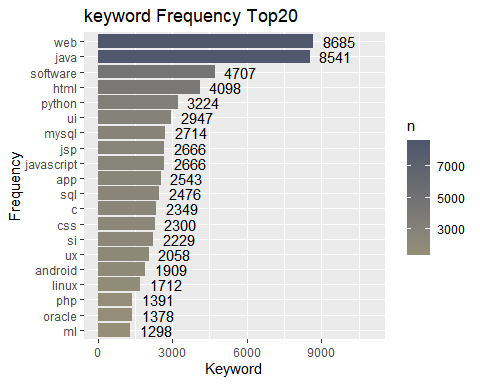
토큰화된 단어 중 핵심어들을 keyword2에 저장함.

## Warning in ex$keyword[is.na(ex$keyword)] <- gsub("[가-힣]", "", ex$qualif2):  
## number of items to replace is not a multiple of replacement length

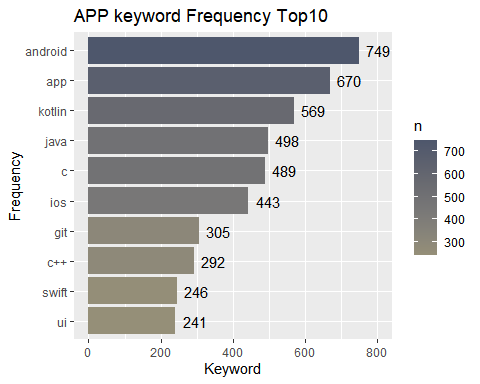
## # A tibble: 6 x 1  
## keyword2   
## <chr>   
## 1 "opencv, c, c++, tensorflow, pytorch, python, ar, deeplearning"   
## 2 "deeplearning, machinelearning, c, c++, python"

## 키워드 시각화

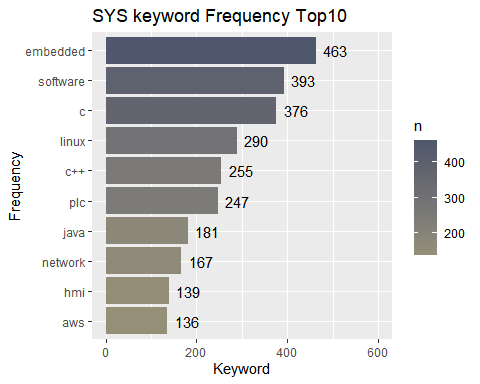
토큰화한 키워드를 이용하여 어떤 키워드가 어떤 직군의 문서에서 자주 나왔는지를 언어 및 직무 스택을 중심으로 살펴봄. Count 함수를 이용하여 단어의 등장 빈도를 세주었고, 이를 막대 그래프를 이용해 시각화하였음.



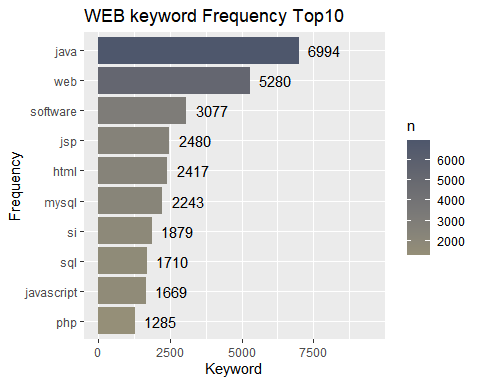
전체 키워드 시각화 결과에서는 web과 java가 약 9000번으로 다른 키워드에 비해 월등히 자주 등장한 것으로 나타남. 이는 웹프로그래머 공고가 다른 직군에 비해 많기 때문으로 웹프로그래머와 관련이 있는 HTML, JSP, JavaScript, PHP 등에 대한 키워드 역시 자주 나타남. 이어서 Python, MySQL, SQL, Oracle, 머신러닝 등 데이터 분석 및 데이터베이스에 대한 키워드가 자주 나타났고, C, App, Android가 이어서 자주 등장하였음.



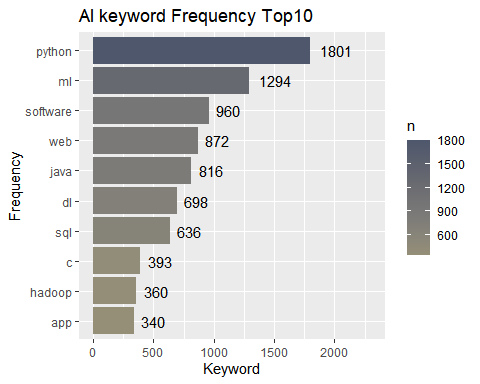
어플리케이션 개발 직군에서는 Android, Kotlin 등 안드로이드 개발 관련 키워드가 가장 많이 나타났고, 이어서 Java, C 등 개발 언어가 뒤를 이었음. Ios, Swift 등 Ios 어플리케이션 개발에 대한 키워드는 각각 6번째와, 9번째로 자주 등장한 것으로 나타남. 한국 어플리케이션 개발은 안드로이드 개발 직군에 대한 수요가 더 많은 것으로 나타남.



시스템, DBA 관련 직군에서는 임베디드 키워드가 가장 자주 나타났고, 그 뒤로 C, C++, Java 등 개발 언어에 대한 키워드가 자주 나타났음. Linux가 다른 직군에 비해 높은 순위를 차지한 것을 보아 Linux 개발 환경을 많이 사용하는 것으로 추측해볼 수 있음. 이어서 PLC, HMI 등 직무 관련 키워드가 많이 등장하였음.



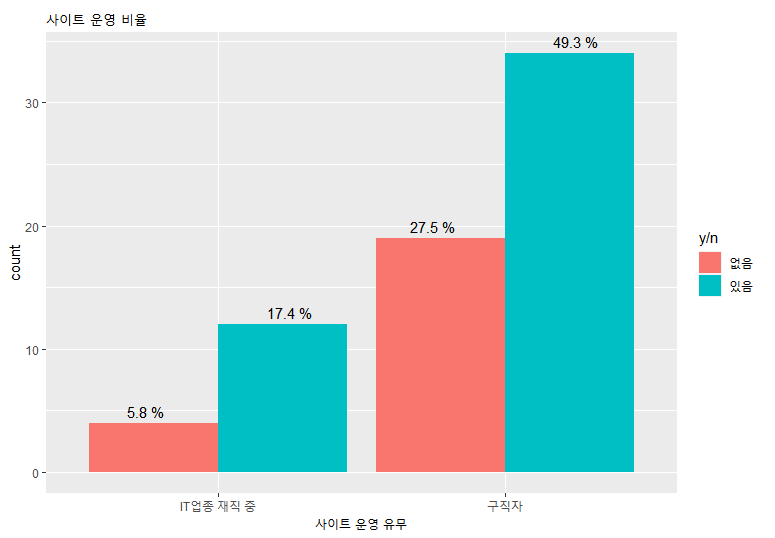
웹 개발 직군에서는 Java가 가장 자주 등장하였고, JSP, JavaScript 등 관련 키워드 역시 자주 등장함. Web, HTML, PHP 등 웹 개발 키워드 역시 자주 등장하였음. 이밖에 MySQL, SI, SQL 등 백엔드 개발 관련 키워드가 각각 6번째, 7번째 8번째를 차지하였음.



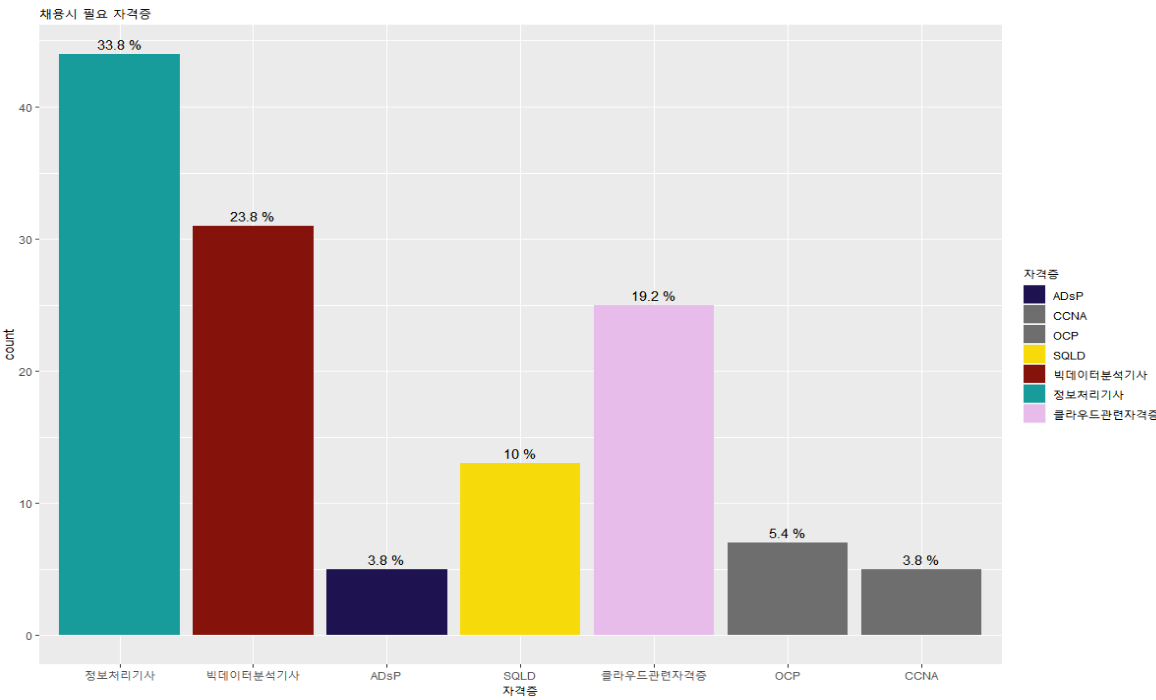
AI 및 데이터 관련 직군에서는 Python이 가장 자주 나타났고, 머신러닝과 딥러닝 키워드가 상당히 자주 나타난 것으로 보아 기계학습 관련 직무에 수요가 큰 것으로 보임. 이어서 Java 및 C언어가 자주 나타났고, SQL, Hadoop 등 데이터베이스 관련 키워드도 자주 나왔음.

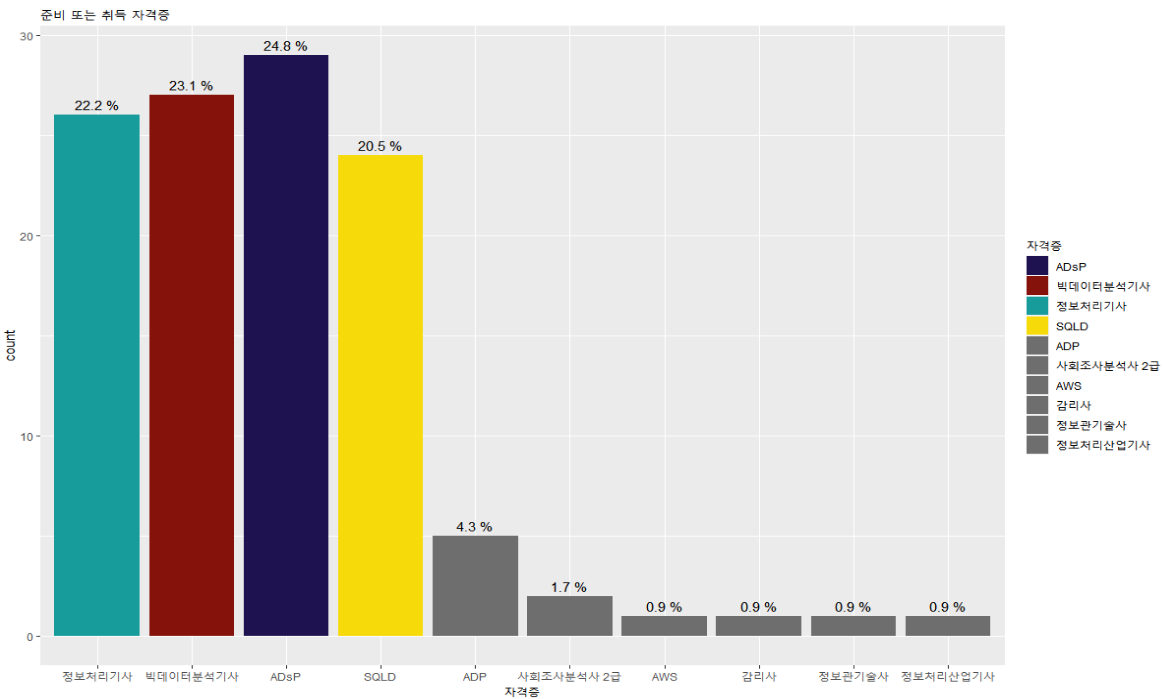
## 설문조사 분석

구직자 설문조사 했고, 현직자 중 설문조사가 완료된 파일을 받아 AWS에 적재 후 R로 불러와서 분석함.



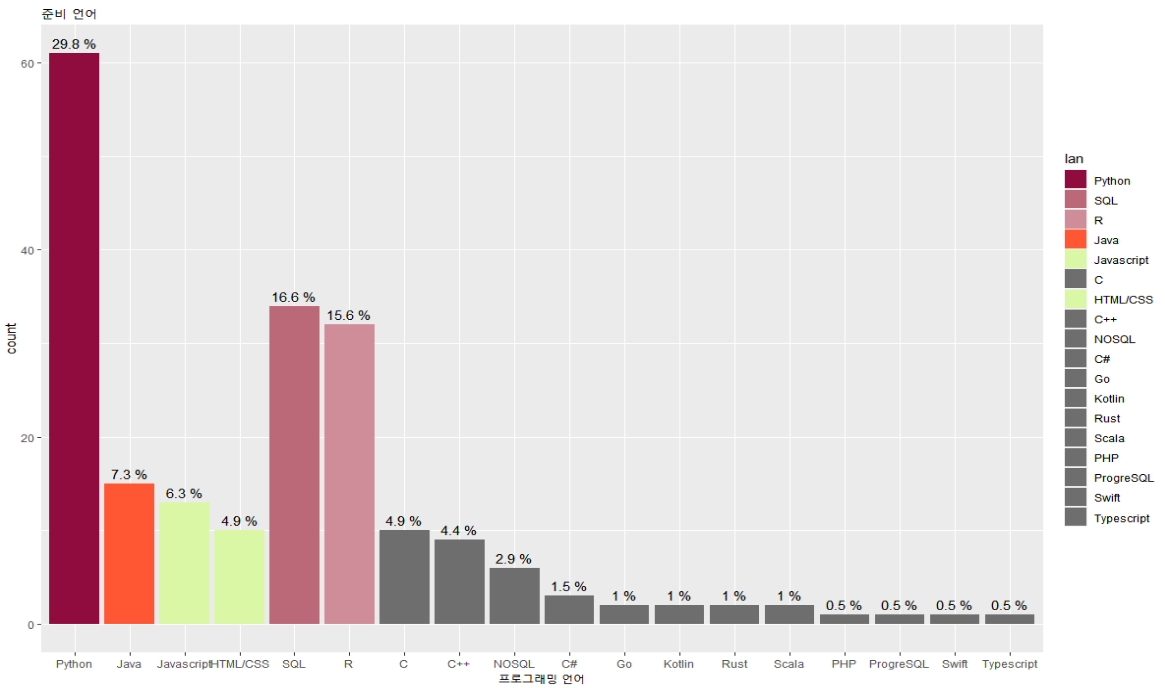
구직자 설문조사에서 사이트(notion, github, blog)를 운영하는 비율을 비교해 보았다. 많은 비율의 구직자들이 사이트를 운영하고 있음.

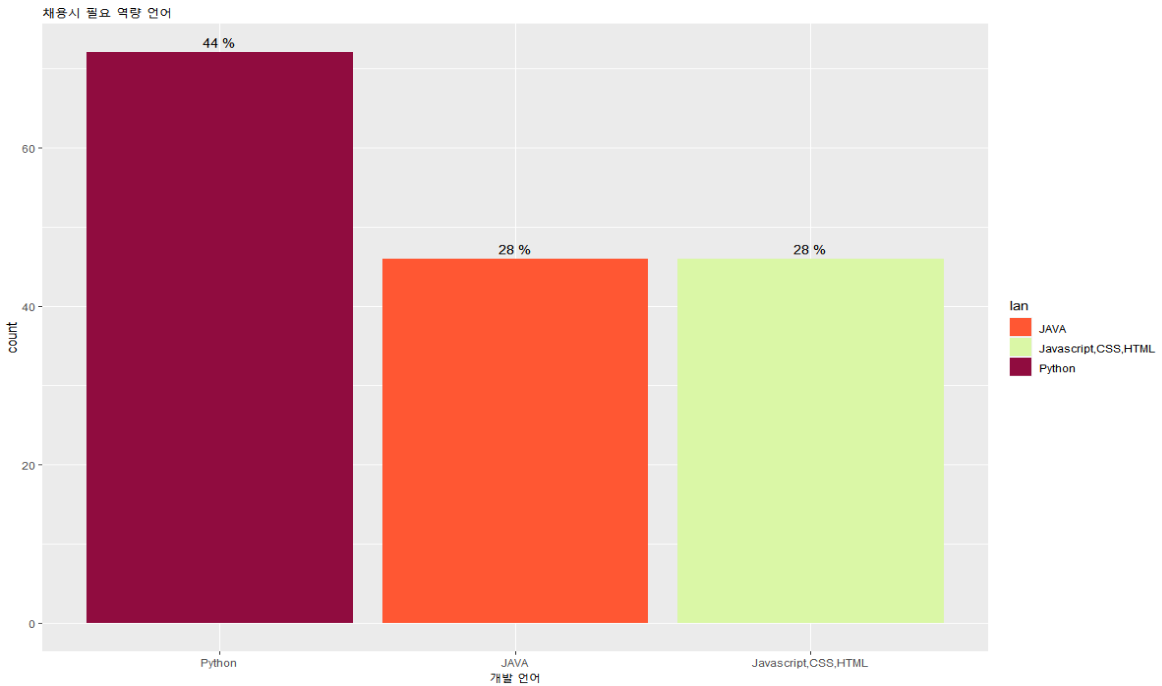




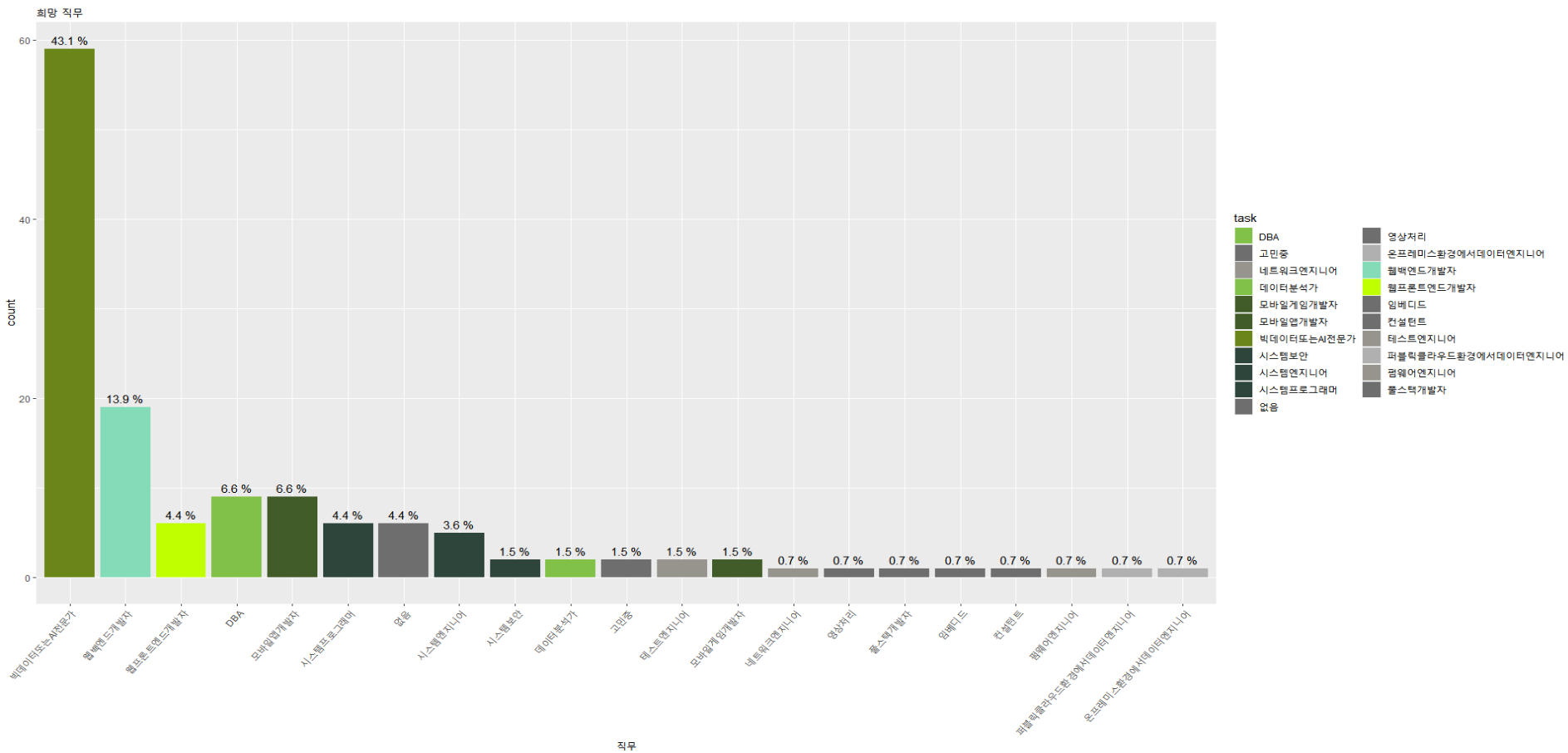
구직자 설문조사와 현직자 설문조사에서 자격증 비교 분석 함.

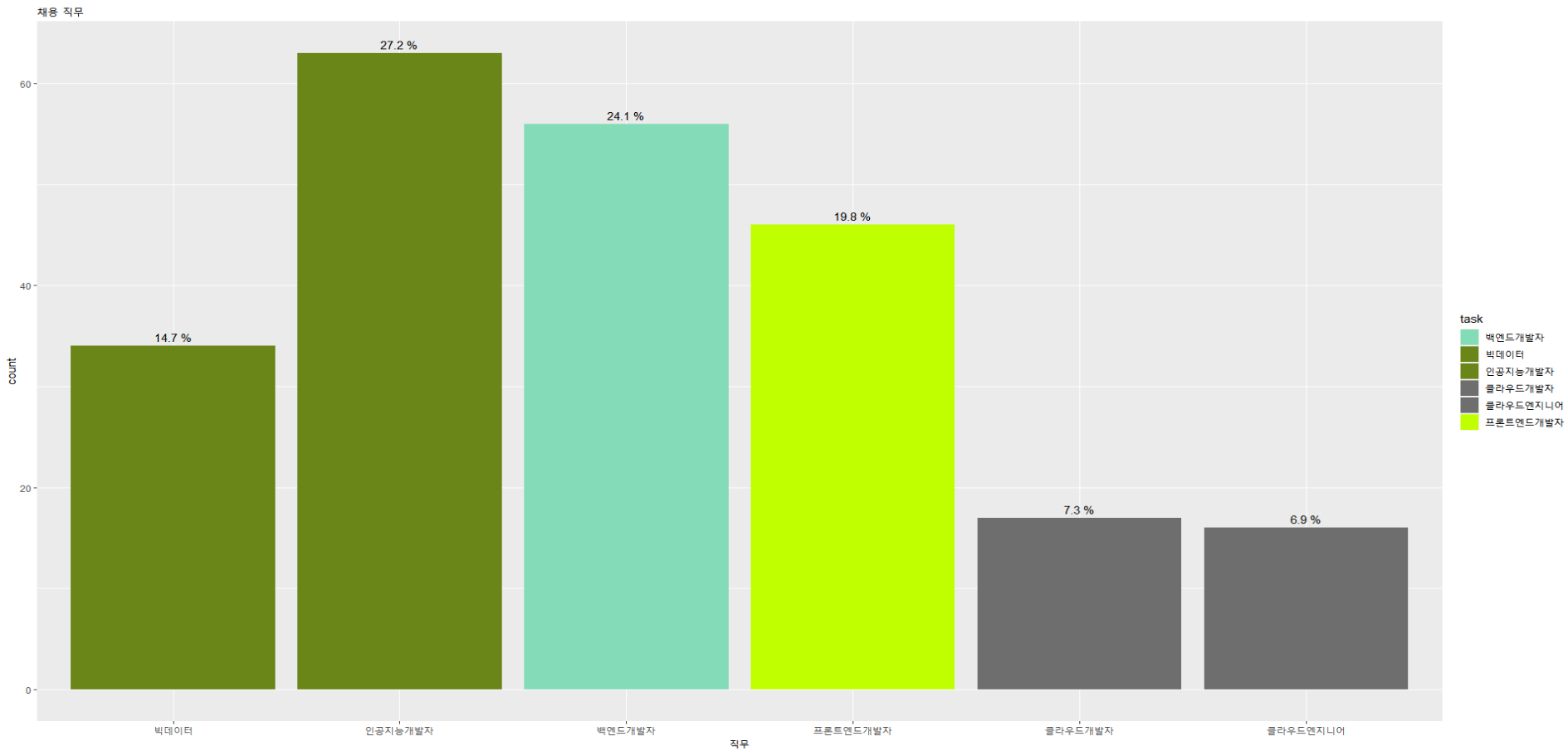
대체적으로 구직자가 준비하는 자격증과 현직자가 필요하다고 생각하는 자격증이 큰차이가 없음, ADsP와 SQLD가 구직자가 준비하는 것에 비해 현직자가 필요하다고 생각하는 비율이 낮았고, 현직자가 필요하다고 생각하는 자격증에 클라우드 관련 자격증의 비율이 높았음. 구직자 설문조사가 인공지능과 빅데이터 관련해서 준비하는 분들에 집중적으로 이루어져서 나온 결과라고 보여짐.





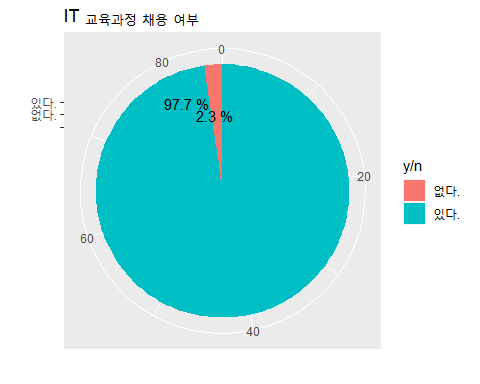
구직자 설문조사와 현직자 설문조사에서 개발 언어를 비교 분석하였음.

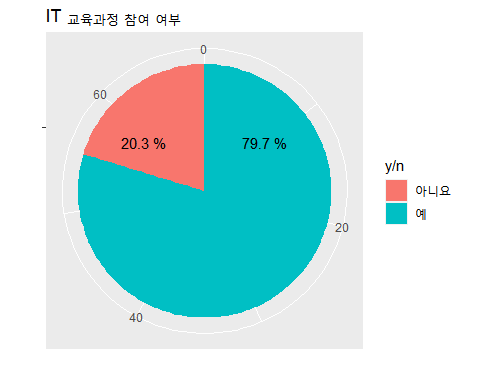




대체적으로 구직자가 준비하는 언어와 현직자가 필요하다고 생각하는 언어가 비슷했음. 구직자가 준비하는 언어중에 SQL과 R이 많았는데 이것도 구직자 설문조사가 인공지능과 빅데이터 관련해서 준비하는 분들에 집중적으로 이루어져서 나온 결과라고 보여짐.

구직자 설문조사와 현직자 설문조사에서 희망 직무와 채용직무를 비교 분석 함.



구직자 설문조사와 현직자 설문조사에서 IT캠프 참여 여부와 채용여부를 비교 분석함. 대체적으로 구직자는 IT캠프를 참여하고 있었고, 현직자는 IT캠프를 참여한 인재를 채용의사가 97.7%로 있는것으로 보여짐.

## 연관분석

연관분석은 군집분석에 의해서 그룹핑된 cluster를 대상으로 해당 그룹에 대한 특성을 분석하는 방법으로 장바구니 분석이라고 한다. 유사한 개체들을 클러스터로 그룹화하여 각 집단의 특성을 파악함. A라는 제품을 구매할 때, B라는 제품도 함께 구매하는 규칙의 패턴을 구하고자 하는 것이 연관규칙 분석의 목적임. 해당 프로젝트에서는 A라는 언어 또는 프레임워크가 B라는 언어 또는 프레임워크가 같이 요청되는 구인공고 간의 기술 또는 스택의 규칙을 발견하기 위해 사용하였음.

### 연관 분석의 기본 개념

* 지지도 전체 거래 중 항목 A와 항목 B를 동시에 포함하는 거래의 비율로 정의함.
* 신뢰도 항목 A를 포함한 거래 중에서 항목 A와 항목 B가 같이 포함될 확률, 연관성의 정도를 파악함.
* 향상도 A를 주어지지 않았을 때의 품목 B의 확률에 비해 A가 주어졌을 때의 품목 B의 확률의 증가 비율, 연관규칙 A=>B는 품목 A와 품목 B의 구매가 서로 관련이 없는 경우에 향상도가 1이 됨.
* 연관규칙 절차 최소 지지도보다 큰 집합만을 대상으로 높은 지지도를 갖는 품목 집합을 찾는 것이며, 처음에는 5%로 잡고 규칙이 충분히 도출되는 지를 보고 다양하게 조절하여 시도한다. 처음부터 너무 낮은 최소 지지도를 선정하는 것은 많은 리소스가 소모되어 불필요함.
* 사용 알고리즘 1세대 알고리즘은 Apriori 나 2세대인 FP Tree, 3세대 FPV를 이용해 연관분석 실시가 가능함 해당 프로젝트에서는 4가지 분류를 통해 나타난 직무별 프로그래밍 언어 또는 스택을 간단한 원리로 쉽게 의미를 파악할 수 있는 apriori알고리즘을 사용하여 연관성을 갖는 패턴을 을 찾아보도록 함.

직군별로 데이터를 분류

“앱”관련 직무의 연관분석을 실시함.

head(final2\_app$keyword2)

## [1] "swift, objective-c, rest api, ios"   
## [2] "django, python, java, docker, ios, c, c++, c#, swift, git"  
## [3] "swift, objective-c, ios"   
## [4] "ios, rxswift, rest api, git, jira, zeplin, confluence"   
## [5] "ios, swift, rxswift, rest api"   
## [6] "ios, swift, jira, objective-c, swiftui, git"

#변수에서 명사단위로 추출  
lword <- Map(extractNoun, final2\_app$keyword2)  
  
# 비어있는 라인 제거  
lword <- unique(lword)

워드 트랜젝션을 확인함.

wordtran <- as(lword, "transactions")

## Warning in asMethod(object): removing duplicated items in transactions

head(wordtran,5)

## transactions in sparse format with  
## 5 transactions (rows) and  
## 1411 items (columns)

transrules <- apriori(wordtran, parameter = list(support=0.1, conf=0.01))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.01 0.1 1 none FALSE TRUE 5 0.1 1  
## maxlen target ext  
## 10 rules TRUE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 139   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[1411 item(s), 1394 transaction(s)] done [0.00s].  
## sorting and recoding items ... [18 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 done [0.00s].  
## writing ... [46 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

연관 단어 시각화를 위해서 자료 구조 변경.

# 연관어 시각화를 위한 igrpah 패키지  
library(igraph)

relueg <- graph.edgelist(rulemat[c(19:40),], directed = F)  
relueg

## IGRAPH c7f851a UN-- 11 22 --   
## + attr: name (v/c)  
## + edges from c7f851a (vertex names):  
## [1] {os} --{android} {os} --{android} {c++} --{c}   
## [4] {c++} --{c} {swift} --{ios} {swift} --{ios}   
## [7] {android}--{ui} {android}--{ui} {android}--{git}   
## [10] {android}--{git} {android}--{ios} {android}--{ios}   
## [13] {app} --{kotlin} {app} --{kotlin} {android}--{app}   
## [16] {android}--{app} {kotlin} --{java} {kotlin} --{java}   
## [19] {android}--{java} {android}--{java} {android}--{kotlin}   
## [22] {android}--{kotlin}

plot.igraph(relueg)

‘DBA+시스템’ 연관성분석 시각화

|  |
| --- |
| C:\Users\Admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\시스템+DBA.PNG |

“앱” 연관성 분석 시각화

|  |
| --- |
|  |

"빅데이터·AI(인공지능)" 연관성 분석 시각화

|  |
| --- |
|  |

“웹 프로그래밍” 연관성 분석 시각화

|  |
| --- |
|  |

### 연관 분석 해석

* “DBA +시스템” 직군의 경우 C, C++을 중심으로EMBEDED, LINUX, C# 이 함께 연관되어 단어가 나타남을 볼 수 있다. 시스템 직군으로 취업을 원할 경우, C, C++ 등을 중심으로 LINUX, C# 다루는 능력이 도움될 것임.
* “앱” 관련 직군의 경우, Android를 중심으로 kotlin, ios, java, app, os, ui 등의 키워드가 함께 나타남을 알 수 있다. 언어로는 kotilin, java, swift 등이 나타났음.
* "빅데이터·AI(인공지능)" 직군의 경우, python을 중심으로, ml, dl, sql, r, ai 등의 키워드가 나타남을 알 수 있다. 데이터 직군으로 취업을 원할 경우, python을 중심으로 머신러닝 활용 경험과, 데이터 핸들링 능력을 갖추는 것이 필요할 것임.
* “웹 프로그래밍” 직군의 경우, java와 web을 중심으로 si, jsp, mysql, html 등의 단어가 같이 추출 되었음을 알 수 있음.

## LDA(Latent Dirichlet Allocation) 과정

자연어 처리에서 잠재 디리클레 할당(Latent Dirichlet allocation, LDA)은 주어진 문서에 대하여 각 문서에 어떤 주제들이 존재하는지를 서술하는 대한 확률적 토픽 모델 기법이며 토픽들이 혼합으로 구성되어 있으며 이 생성되는 과정을 역추적 하는 방식임.

* Latent: 잠재적인, Dirichlet: 수학자 이름, Allocation: 할당

문서들은 토픽들의 혼합으로 구성되어 있고, 토픽들은 확률 분포에 기반하여 단어들을 생성한다고 가정함.

## LDA 실시

keyword열에서 정제한 데이터에서 프로그래밍 언어와 프레임 워크만 추출하여 LDA 과정을 실시함.

### tf-idf 보기

LDA이전에 tf-idf를 통해서 구인공고별 특정 단어의 중요성을 파악해본다. TF-IDF(Term Frequency - Inverse Document Frequency)는 정보 검색과 텍스트 마이닝에서 이용하는 가중치로, 여러 문서로 이루어진 문서군이 있을 때 어떤 단어가 특정 문서 내에서 얼마나 중요한 것인지를 나타내는 통계적 수치임. 문서의 핵심어를 추출하거나, 검색 엔진에서 검색 결과의 순위를 결정하거나, 문서들 사이의 비슷한 정도를 구하는 등의 용도로 사용할 수 있음. ’키워드의 빈도’에 ’문서 빈도의 역수’를 곱하여, 주로 ’문서의 핵심어를 추출할 때 사용되는 가중치’이다.여기저기에서 등장하는 키워드는 아니면서도 의미적으로 중요한 키워드를 뽑기 위해 TF-IDF는 활용된다. ’키워드의 빈도’에 ’문서 빈도의 역수’를 곱하여 모든 문서에 자주 나타난 키워드에는 페널티(낮은 가중치)를 주는 것임.

출처: <https://data-traveler.tistory.com/33> [클로이의 데이터 여행기]

#tf-idf 행렬  
ti\_app<-ti %>% filter(job=="앱")  
# 앱 관련 구인공고의 tf-idf 행렬  
kable(head(ti\_app,10))

| job | word | n | tf | idf | tf\_idf |
| --- | --- | --- | --- | --- | --- |
| 앱 | coutine | 73 | 0.0044264 | 1.3862944 | 0.0061363 |
| 앱 | jetpack | 53 | 0.0032137 | 1.3862944 | 0.0044551 |
| 앱 | rxjava | 93 | 0.0056391 | 0.6931472 | 0.0039087 |
| 앱 | ,android | 44 | 0.0026680 | 1.3862944 | 0.0036986 |
| 앱 | android os | 211 | 0.0127941 | 0.2876821 | 0.0036806 |
| 앱 | mvp | 84 | 0.0050934 | 0.6931472 | 0.0035305 |
| 앱 | ㆍ | 67 | 0.0040626 | 0.6931472 | 0.0028160 |
| 앱 | mvvm | 161 | 0.0097623 | 0.2876821 | 0.0028084 |
| 앱 | native | 154 | 0.0093379 | 0.2876821 | 0.0026863 |
| 앱 | custom | 28 | 0.0016978 | 1.3862944 | 0.0023536 |

ti\_web<-ti %>% filter(job=="웹프로그래머")  
# 웹프로그래머 관련 구인공고의 tf-idf 행렬  
kable(head(ti\_web,10))

| job | word | n | tf | idf | tf\_idf |
| --- | --- | --- | --- | --- | --- |
| 웹프로그래머 | nexacro | 102 | 0.0016086 | 1.3862944 | 0.0022300 |
| 웹프로그래머 | miplatform | 199 | 0.0031383 | 0.6931472 | 0.0021753 |
| 웹프로그래머 | servlet | 146 | 0.0023025 | 0.6931472 | 0.0015960 |
| 웹프로그래머 | websquare | 59 | 0.0009305 | 1.3862944 | 0.0012899 |
| 웹프로그래머 | mhtml | 98 | 0.0015455 | 0.6931472 | 0.0010713 |
| 웹프로그래머 | dba | 229 | 0.0036114 | 0.2876821 | 0.0010389 |
| 웹프로그래머 | css3 | 225 | 0.0035483 | 0.2876821 | 0.0010208 |
| 웹프로그래머 | apm | 80 | 0.0012616 | 0.6931472 | 0.0008745 |
| 웹프로그래머 | es6 | 80 | 0.0012616 | 0.6931472 | 0.0008745 |
| 웹프로그래머 | crm | 172 | 0.0027125 | 0.2876821 | 0.0007803 |

ti\_data<-ti %>% filter(job=="빅데이터·AI(인공지능)")  
# 빅데이터·AI(인공지능) 관련 구인공고의 tf-idf 행렬  
kable(head(ti\_data,10))

| job | word | n | tf | idf | tf\_idf |
| --- | --- | --- | --- | --- | --- |
| 빅데이터·AI(인공지능) | ml | 1233 | 0.0409607 | 0.2876821 | 0.0117837 |
| 빅데이터·AI(인공지능) | dl | 597 | 0.0198326 | 0.2876821 | 0.0057055 |
| 빅데이터·AI(인공지능) | pytch | 104 | 0.0034549 | 1.3862944 | 0.0047895 |
| 빅데이터·AI(인공지능) | nlp | 180 | 0.0059797 | 0.6931472 | 0.0041448 |
| 빅데이터·AI(인공지능) | data | 140 | 0.0046509 | 0.6931472 | 0.0032237 |
| 빅데이터·AI(인공지능) | r, | 60 | 0.0019932 | 1.3862944 | 0.0027632 |
| 빅데이터·AI(인공지능) | tensflow | 118 | 0.0039200 | 0.6931472 | 0.0027171 |
| 빅데이터·AI(인공지능) | ai, | 114 | 0.0037871 | 0.6931472 | 0.0026250 |
| 빅데이터·AI(인공지능) | ml, | 49 | 0.0016278 | 1.3862944 | 0.0022566 |
| 빅데이터·AI(인공지능) | dl, | 97 | 0.0032224 | 0.6931472 | 0.0022336 |

# DBA+시스템 관련 구인공고의 tf-idf 행렬  
ti\_sys<-ti %>% filter(job=="DBA+시스템")  
# 2d, 3d, rpg 등의 keyword는 기존 공고 결합과정 잘못분류되어 제거하여 분석함  
kable(head(ti\_sys,10))

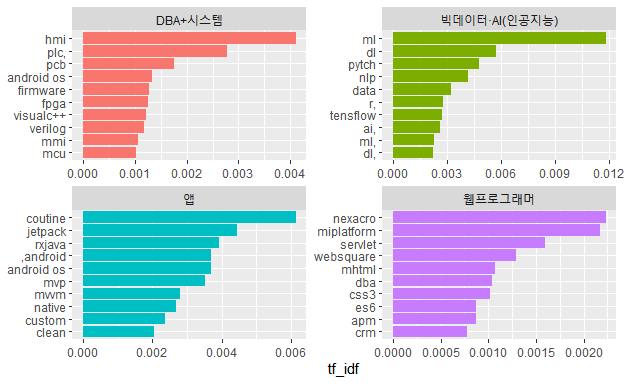
| job | word | n | tf | idf | tf\_idf |
| --- | --- | --- | --- | --- | --- |
| DBA+시스템 | flash | 319 | 0.0337209 | 1.3862944 | 0.0467471 |
| DBA+시스템 | 3d | 445 | 0.0470402 | 0.2876821 | 0.0135326 |
| DBA+시스템 | 2d | 386 | 0.0408034 | 0.2876821 | 0.0117384 |
| DBA+시스템 | rpg | 321 | 0.0339323 | 0.2876821 | 0.0097617 |
| DBA+시스템 | hmi | 135 | 0.0142706 | 0.2876821 | 0.0041054 |
| DBA+시스템 | plc, | 38 | 0.0040169 | 0.6931472 | 0.0027843 |
| DBA+시스템 | pcb | 58 | 0.0061311 | 0.2876821 | 0.0017638 |
| DBA+시스템 | android os | 44 | 0.0046512 | 0.2876821 | 0.0013381 |
| DBA+시스템 | firmware | 42 | 0.0044397 | 0.2876821 | 0.0012772 |
| DBA+시스템 | fpga | 41 | 0.0043340 | 0.2876821 | 0.0012468 |

table(data$job)

##   
## DBA+시스템 빅데이터·AI(인공지능) 앱   
## 1670 2648 1492   
## 웹프로그래머   
## 8419

top2<-ti2 %>% group\_by(job) %>% slice\_max(tf\_idf,n=10,with\_ties = F)

# 각 분야별 단어빈도에 대한 사항을 시각화  
ggplot(top2,aes(x=reorder\_within(word,tf\_idf,job),y=tf\_idf,fill=job))+  
 geom\_col(show.legend=F)+coord\_flip()+  
 facet\_wrap(~job,scales="free",ncol=2)+  
 scale\_x\_reordered()+labs(x=NULL)



## 해석

해당 그래프를 활용하여 문서 주제별 자주 업급되는 keyword를 확인할 수 있었음.

* 먼저, DBA + 시스템의 경우, hmi, plc, pcb, android os 등의 단어가 다른 주제에 비해서 많이 언급된 단어로 나타남.
* 데이터·AI(인공지능) 관련 구인공고의 경우, 다른 주제에 비해 ml, dl, pytch, nlp, data 등의 단어가 많이 명시 되었다. python, sql의 경우에는 다른 주제에도 많이 나타나는 단어이기 때문에 수치가 높지 않아 보임.
* 앱의 경우, coutine, jetpack, rxjava, android의 단어가 많이 등장함.
* 웹 프로그래밍의 경우 nexacro, miplatform, servlet, websquare 등의 단어가 분포함을 알 수 있었음.
* 프로그래밍 언어의 경우, 다른 문서 주제에서도 찾아 볼 수 있는 키워드이기 때문에 tf-idf행렬에서 제외된 것으로 보임.

## LDA 실시

final2<- read\_excel("final\_final2.xlsx")  
b<- final2 %>% filter(!is.na(job))  
keyword <- iconv(b$keyword2, "utf-8","UTF-8")

전처리 끝난 데이터 바탕으로 형태소 분석을 한 데이터를 corpus로 만들어준 다음, TDM 형태로 만들어 줌,

# TDM  
options(mc.cores=1)  
keycorpus <- VCorpus(VectorSource(keyword))  
tdm <- TermDocumentMatrix(keycorpus,  
 control = list(  
 removePunctuation= T,  
 removeNumbers = T))

nTerms는 추출한 단어 수, nDocs는 추출문서의 수를 보여줌, head()로 어떤 단어들이 추출되었는지 확인함.

nTerms(tdm)

## [1] 2735

nDocs(tdm)

## [1] 14229

head(Terms(tdm))

## [1] "‘you" "“hello" "aaai" "aac" "aar" "aarrr"

TDM 형식을 DTM 형식으로 바꿔줌.

dtm <- as.DocumentTermMatrix(tdm)  
rowTotals <- apply(dtm, 1, sum)  
dtm.new <- dtm[rowTotals >0, ]  
dtm<- dtm.new  
dtm

## <<DocumentTermMatrix (documents: 13597, terms: 2735)>>  
## Non-/sparse entries: 82485/37105310  
## Sparsity : 100%  
## Maximal term length: 21  
## Weighting : term frequency (tf)

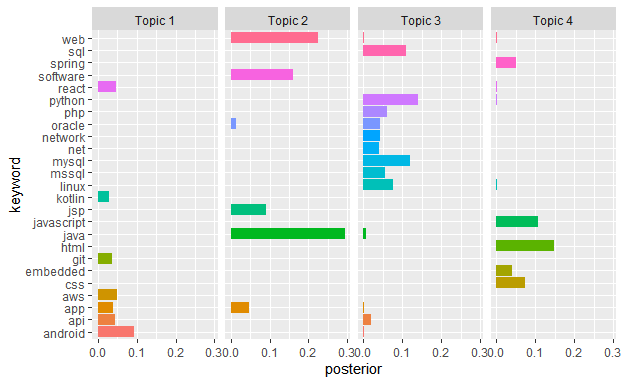
topicmodels의 LDA함수를 사용하여 텍스트를 분류, K는 나눌 집단 수의며 4개의 집단으로 나눠 모델을 제작함

lda <- LDA(dtm, k=4, control = list(seed=12345))  
term <- terms(lda,20)  
  
term

## Topic 1 Topic 2 Topic 3 Topic 4   
## [1,] "android" "java" "python" "html"   
## [2,] "aws" "web" "mysql" "javascript"  
## [3,] "react" "software" "sql" "css"   
## [4,] "api" "jsp" "linux" "spring"   
## [5,] "app" "app" "php" "embedded"   
## [6,] "git" "ajax" "mssql" "framework"   
## [7,] "kotlin" "asp" "oracle" "vuejs"   
## [8,] "nodejs" "erp" "network" "jquery"   
## [9,] "rest" "oracle" "net" "hadoop"   
## [10,] "ios" "rpg" "bigdata" "springboot"  
## [11,] "typescript" "flash" "database" "plc"   
## [12,] "docker" "was" "windows" "spark"   
## [13,] "native" "crm" "api" "backend"   
## [14,] "swift" "miplatform" "aspnet" "frontend"   
## [15,] "github" "servlet" "dba" "kafka"   
## [16,] "kubernetes" "xml" "tensorflow" "hmi"   
## [17,] "jira" "scm" "unix" "data"   
## [18,] "nosql" "application" "application" "apache"   
## [19,] "gcp" "proc" "nlp" "webui"   
## [20,] "amazon" "nexacro" "mariadb" "visualc"

그래프로 확인하면 다음과 같음.

x <- posterior(lda)$terms  
y <- data.frame(t(x[,apply(x,2,max)>0.03]))  
z <- data.frame(type=paste("Topic",1),  
 keyword=rownames(y), posterior=y[,1])  
  
  
for (i in 1:4) {  
 z <- rbind(z, data.frame(type=paste("Topic",i),  
 keyword=rownames(y), posterior=y[,i]))  
}  
  
  
ggplot(z, aes(keyword,posterior, fill=as.factor(keyword))) +  
 geom\_bar(position = "dodge", stat = "identity") +  
 coord\_flip()+  
 facet\_wrap(~type, nrow=1) +  
 theme(legend.position = "none")



## LDA해석

해당 그래프를 바탕으로 단어 모음 별 토픽을 분석함.

* Topic 1는 react, kotiln, git aws, app, api, android의 단어 등을 보아 “앱 개발자” 주제라고 유추해 볼 수 있음.
* Topic 2는 java, web, software, jsp 등의 단어가 나타남을 보이고 이는 “웹 개발자” 주제라고 유추해 볼 수 있음.
* Topic 3는 python, sql, mysql, mssql, linux 등의 단어가 나타남을 보이는데 이는 “빅데이터 또는 AI 관련” 직군 주제라고 유추해 볼 수 있음.
* TOpic 4는 html, javascript, css, spirng 등의 내용을 보아 “웹(프론트엔드, 백엔드)” 관련 주제라고 유추해 볼 수 있음.
* 기존에 범주로 나누었던 DBA + 시스템 개발자의 경우, 주제가 명확히 구분되기 힘든 직군 이며, 그 수가 적어 비지도 학습으로 분류되지는 않은 것으로 보임.

## 결론

산업 통계에 따르면 개발자, 그 중에서도 AI 및 데이터 개발자에 대한 수요가 있는 것으로 나타남.

설문조사에 따르면 채용 의사가 있는 기업들 중 약 41.9%의 기업에서 AI 및 빅데이터 직군의 개발자를 채용할 계획이 있는 것으로 나타남. 이 밖에 웹 개발자, 클라우드 개발자에 대한 수요가 있었음. 중요하게 보는 언어로는 Python과 Java 및 HTML, CSS 등이 있었음.

키워드 빈도 분석 결과, 전체 키워드에서는 Web, Java 키워드가 눈에 띄게 많이 나타났음. 이는 웹 개발자의 구인 공고가 다른 직군에 비해 월등히 많이 나왔기 때문으로 보임. 이 밖에 주요 키워드로는, Python, Mysql, JSP 등이 있었음. 직무 분류 하에서는 시스템 및 DBA 직군에서 Embedded, C 및 Linux 키워드가 많이 나왔고, 앱 개발 직군에서는 Android, Kotlin, Java 키워드가 많이 관찰되었음. 데이터 및 AI 개발 직군에서는 Python과 머신러닝, Java 등의 키워드가 자주 나왔음. 마지막으로 웹 개발 직군에서는 Java와 JSP, HTML 등의 키워드가 자주 관찰되었음.

토픽 모델링의 경우, LDA를 이용했을 때, AI 및 데이터 분석 직군, 앱 개발, 웹 개발 둘로 문서 주제가 나뉜 것으로 추측해 볼 수 있음. Apriori를 이용한 연관 분석 결과, DBA 및 시스템에서는 Embedded와 C를 중심으로 Linux, C++ 등의 키워드가 나타났고, 웹 개발자는 Web과 Java를 중심으로 HTML, SI, SQL 등의 키워드가 나타났음. 데이터 및 AI 개발 직군에서는 Python을 중심으로 딥러닝, 인공지능, R 및 SQL 등의 키워드가 관찰됐음. 앱 개발자에서는 안드로이드 키워드가 중심이 되어 Kotlin, Java가 나타났고, IOS와 Swift가 함께 나타나는 경향을 보였음. TF-IDF 행렬에서는 특성상, 해당 문서에서 고유하게 자주 나오는 키워드가 상위 키워드로 관찰되었음. 이 때문에 개발 언어보다 해당 직무의 고유 스택이 핵심 키워드로 관찰되는 경향이 있었음. 시스템 및 DBA에서는 HMI, PLC로 나타났고, 데이터 및 AI 부문에서는 머신러닝, 딥러닝, Pytorch 등이 관찰되었음. 앱 개발 직무에서는 Jetpack, RxJava 등의 키워드가 관찰되었고, 웹 개발 직무에서는 Neaxccro, Miplatform, Servlet 등의 키워드가 핵심 키워드로 선정되었음.

결론적으로, 아직 개발자에 대한 취업 시장이 레드 오션이라 볼 수는 없으며, 만일 AI 및 빅데이터 분석 직무에 취업을 희망하고 있다면 개발 언어로는 Python을, 주요 스택으로는 머신러닝 및 딥러닝 스택을 개발하는 것이 좋을 것으로 보임.