**II**

**1. 인공지능 개발 환경**

1. **개발 언어 소개**
   1. R

R 은 오픈소스 프로그래밍 언어로 통계 계산과 그래픽 활용에 특화됐다. 벨 연구소의 존 챔버스, 릭 배커, 그리고 앨런 윌크스가 개발한 통계언어인 S 에 기반해, 뉴질랜드 오클랜드 대학의 로버트 젠틀맨과 로스 이하카가 지금의 R 을 개발했다. 현재는 R 코어 팀이 개발하고 있으며, 다양한 통계 기법과 수치 해석 기법을 지원한다. R 에는 사용자가 제작한 패키지 (추가적인 기능을 하는 함수)를 CRAN (the Comprehensive R Archive Network)으로부터 받을 수 있다. R은 선형 및 비선형 모델링, 통계 분석, 시계열 분석, 분류, 클러스터링 등 다양한 통계 및 그래픽 기술에 강점이 있다.

1. **개발 환경 소개**

2.5. R - Rstudio

Rstudio는 R 사용을 위한 통합 개발 환경 (Integrated Development Environment, IDE)로, 어도비 콜드퓨전의 개발자인 JJ Allaire가 개발했다. Rstudio는 일반 데스크톱 어플리케이션과 웹 브라우저 방식으로 사용할 수 있다. 일반적인 IDE와 마찬가지로 디버깅, 컴파일, 배포 등 개발을 위한 기능을 통합적으로 제공한다.

2.6. Rstuio 사용하기

- Rstudio는 <https://www.rstudio.com/>을 통해 다운받을 수 있다. 일반적인 데스크톱 어플리케이션의 경우는 계정 생성이 필요없고 무료로 다운받을 수 있다.

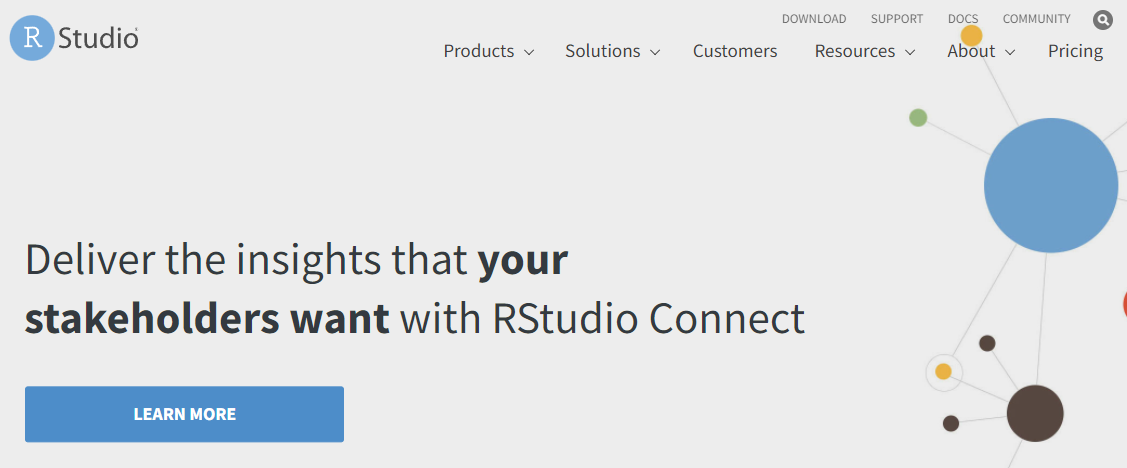


그림 1. Rstudio.com의 프론트 페이지

- Product tap의 데스크톱 어플리케이션을 선택한다. 일반 Open source edition을 다운로드 받고 설치하면 사용할 수 있다.

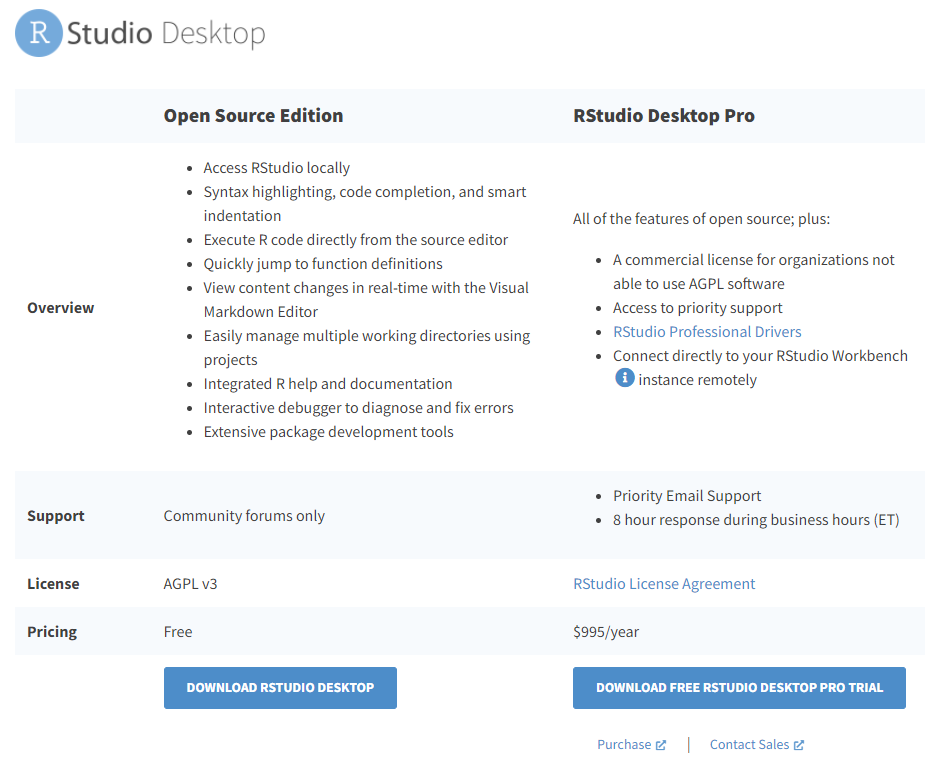


그림 2. Rstudio 어플리케이션 다운로드 페이지

- Rstudio를 실행하면 아래와 같은 화면을 볼 수 있다. File 탭의 <New File>을 보면 코드 스크립트를 작성할 수 있는 <R script> 버튼이 있다. 우측 상단 Environmental 화면에서는 <Import Datasets> 버튼을 통해 데이터를 환경에 불러올 수 있다.

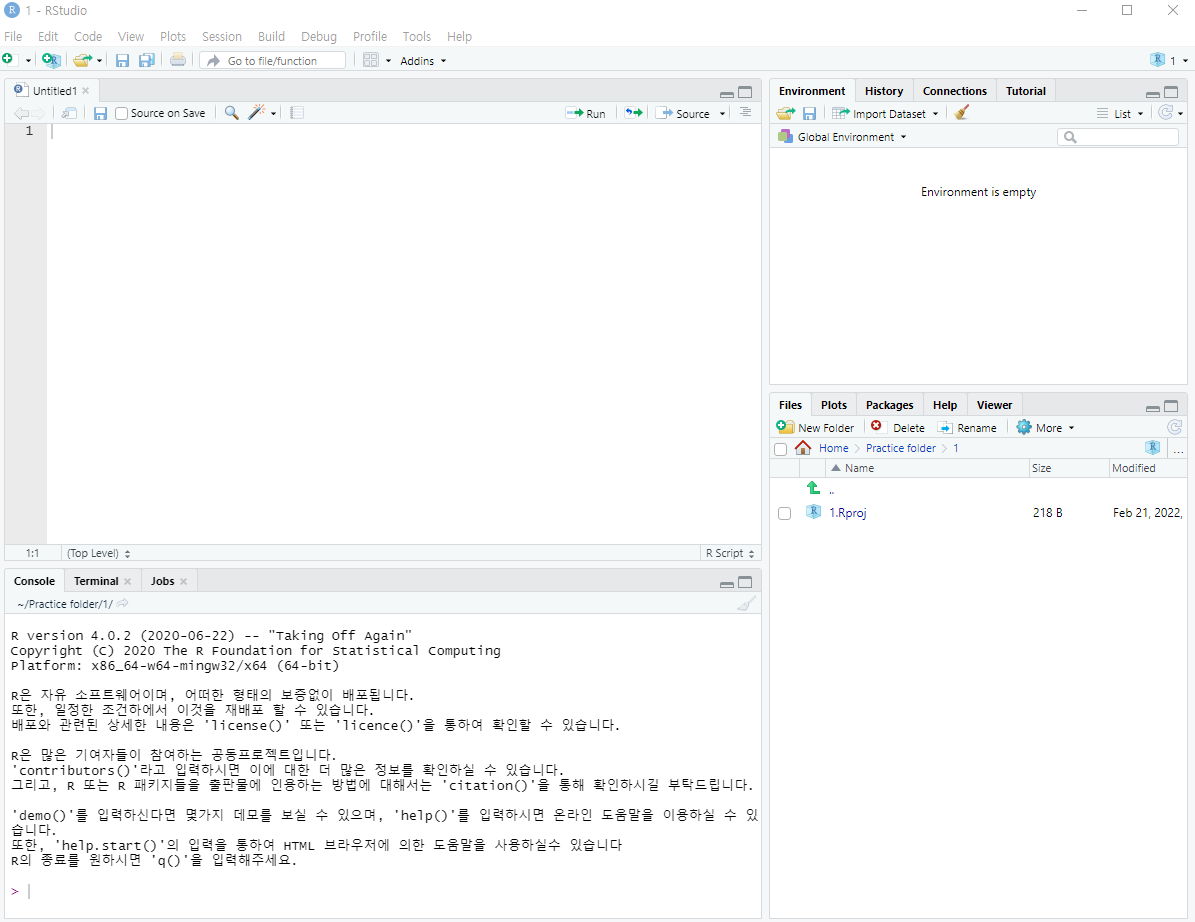


그림 3. Rstudio의 실행 화면

1. **R 기초**

3.1. 기초 문법

R은 파이썬과 마찬가지로 숫자형, 문자형 등의 자료 형식이 존재한다. “str()” 함수를 통해 다음과같이 데이터 형식을 파악할 수 있다.

|  |
| --- |
| > a = 1  > str (a)  num 2  > b = 2  > str (b)  num 2  > c = ‘k’  > str(c)  chr “k” |

일반적인 사칙연산(+, -, \*, /)은 파이썬과 같은 연산자를 사용한다. 하지만, 나눗셈의 나머지 값은 %% 연산자를 사용한다.

|  |
| --- |
| > a = 7  > b = 2  > a + b  9  > a \* b  14  > a / b  3.5  > a \*\* b  49  > b %% a  2  > a %/% b  3 |

* 1. 조건문(if), 반복문(while, for)

조건문과 반복문은 파이썬과 비슷한 구조를 갖고 있다. 다만, 조건문의 조건 부분과 수행부분이 더 명확히 나눠진다.

|  |
| --- |
| > James = 77  > Jessica = 54  > if (James > Jessica) {  print(‘James is heavier than Jessica’)  } else if (James == Jessica) {  print(‘James is equal to Jessica’)  } else {  print(‘James is lighter than Jessica’) }  [1] “James is heavier than Jessica” |

R에서는 갱신된 정보를 print하기위해 “sprint()” 함수를 사용하며, 해당 정보를 그때마다 저장 (ex. 아래 script의 ‘sys’)하여 “print()” 함수로 불러온다.

|  |
| --- |
| > EXP = 0  > while (EXP < 10){  EXP = EXP +1  sys = sprintf('경험치가 %d입니다.', EXP)  print(sys)  if (EXP == 10){  print('레벨업!')  }}  [1] 경험치가 1입니다.  [1] 경험치가 2입니다.  …  [1] 경험치가 8입니다.  [1] 경험치가 10입니다.  [1] 레벨업! |

for 반복문을 사용하여 파이썬과 마찬가지로 소수를 판별할 수 있지만, 특정 구간의 수를 표현할 때는 “:” 연산자를 사용한다.

|  |
| --- |
| > n = 9  > for (i in 2:8){  if (n %% i == 0){  print('not prime')  } else {  print('prime')  }  }  [1] “prime”  [1] “not prime”  [1] “prime”  [1] “prime”  [1] “prime”  [1] “prime”  [1] “prime” |

‘