

지능형 축산시설

AI, ML, Data Science,
and Animal Science

Youngjun Na

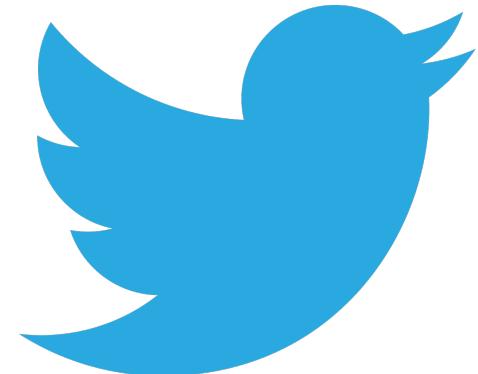
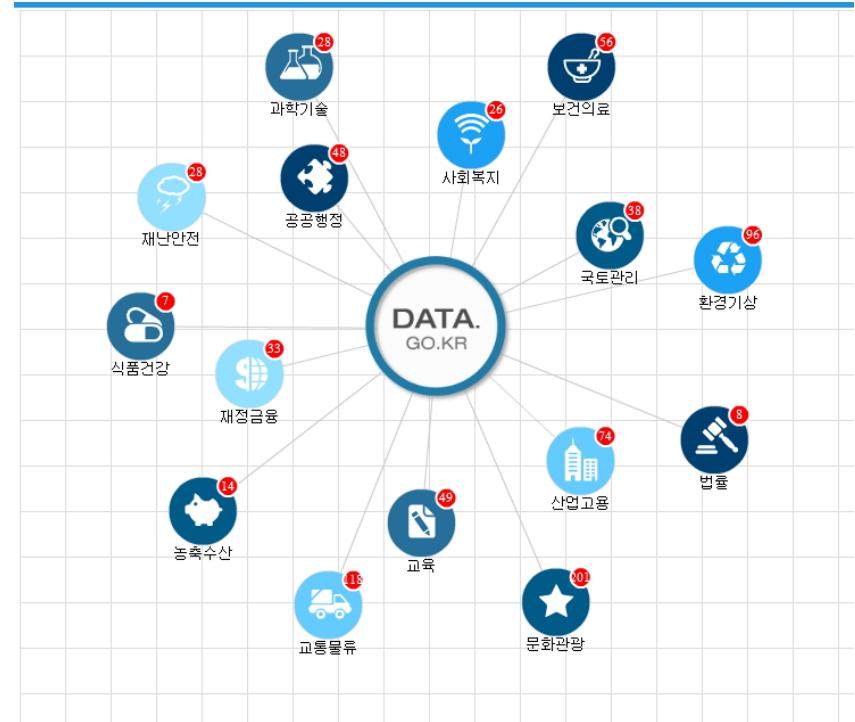
Github: <https://github.com/youngjunna>

Email: ruminoreticulum@gmail.com

데이터의 시대

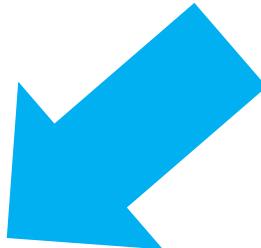
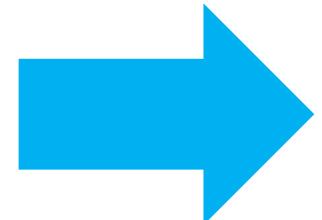
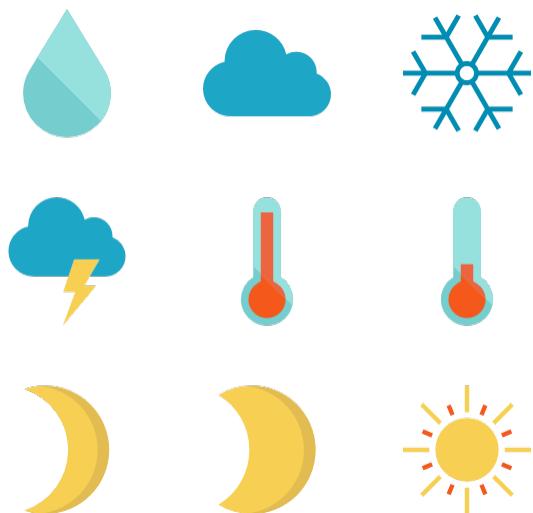
데이터 대폭발시대

- 전자상거래
- 비즈니스
- 소셜미디어
- 과학
- IoT
- 정부 등



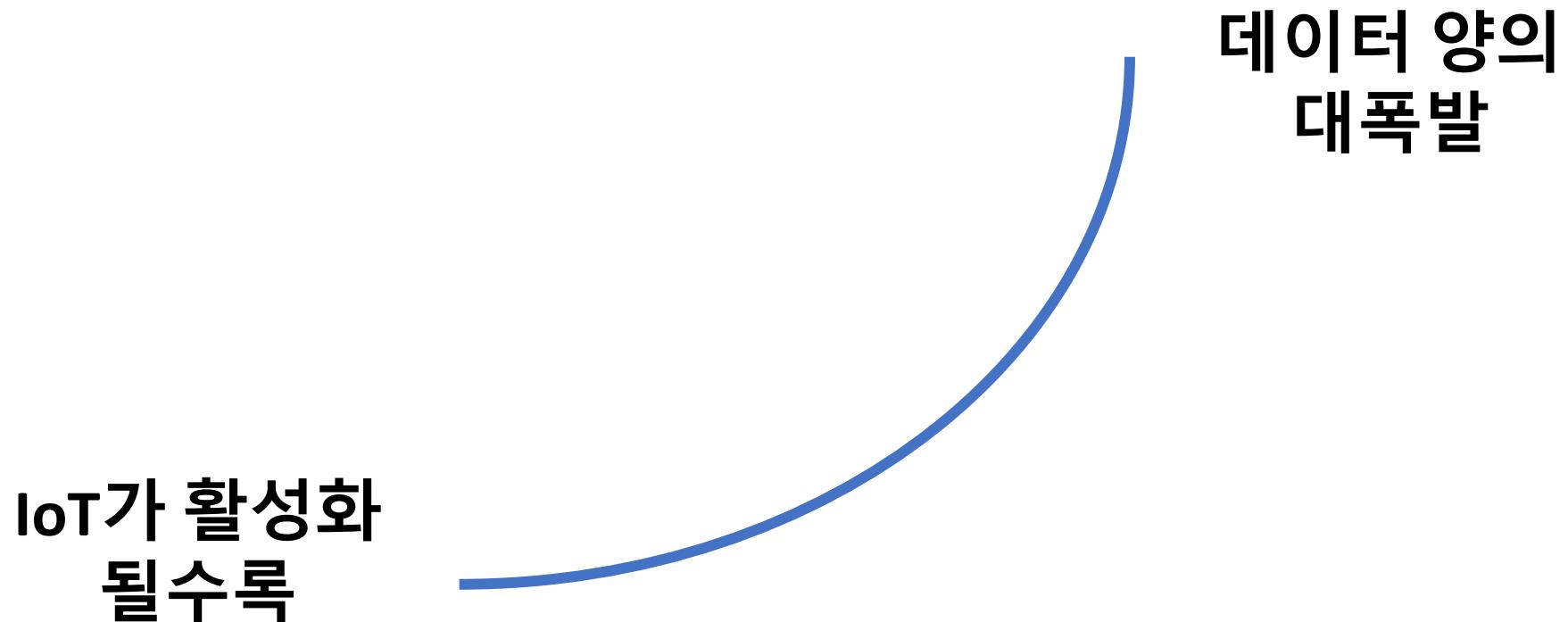
축산분야 IoT의 발달

- 축산분야에서 IoT의 발달
- IoT의 핵심 == 자동화된 많은 센서(sensor)



Data Science + Animal Science

- IoT (Internet of Things)의 발달로 인한 데이터 양의 대폭발



**대체 이 데이터로
무엇을 할 것인가?**

데이터혁명 시대

- 산업혁명 시대
 - = 물건 생산을 위해 육체노동을 기계로 자동화 하자
- 데이터혁명 시대
 - = 데이터 분석을 위해 정신노동을 기계로 자동화 하자
(물론 육체노동 자동화는 디폴트)

단순하고 반복적인 일들은 기계에게 맡기고

최대한 우리는

1. 어떻게 분석할 것인지 기획하고
2. 서비스의 질을 높일 것인지
3. 좀 더 본질적인 고민을 하자
4. 그리고 사랑하는 사람들과 더 많은 시간을 보내자

AI and Machine learning

AlphaGo

BBC News Sport Weather iPlayer TV Radio More Search

Find local news

Home UK World Business Politics Tech Science Health Education Entertainment & Arts More

Technology

Google achieves AI 'breakthrough' at Go

An artificial intelligence program developed by Google beats Europe's top player at the ancient Chinese game of Go, about a decade earlier than expected.

27 January 2016 | Technology

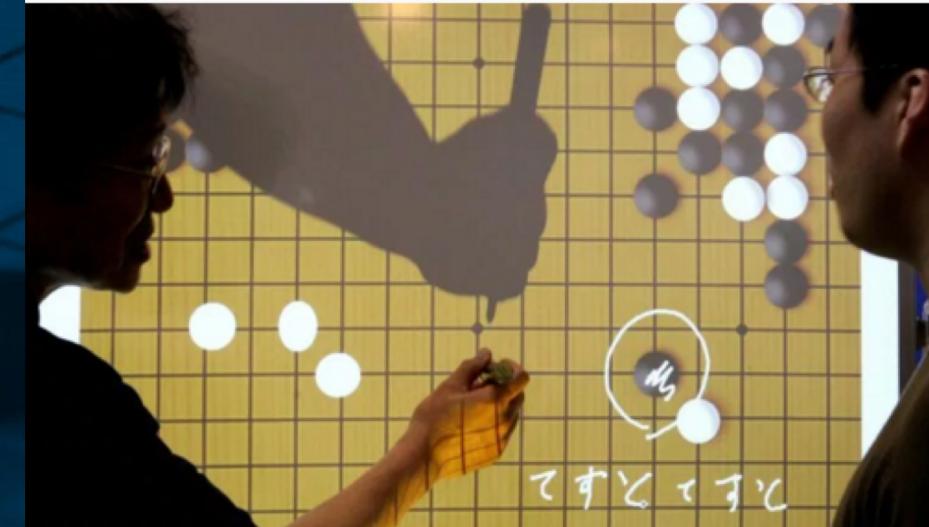
How did they do it?
What is the game Go?

Facebook trains AI to beat humans at Go



Google's AI just cracked the game that supposedly no computer could beat

By Mike Murphy | January 27, 2016

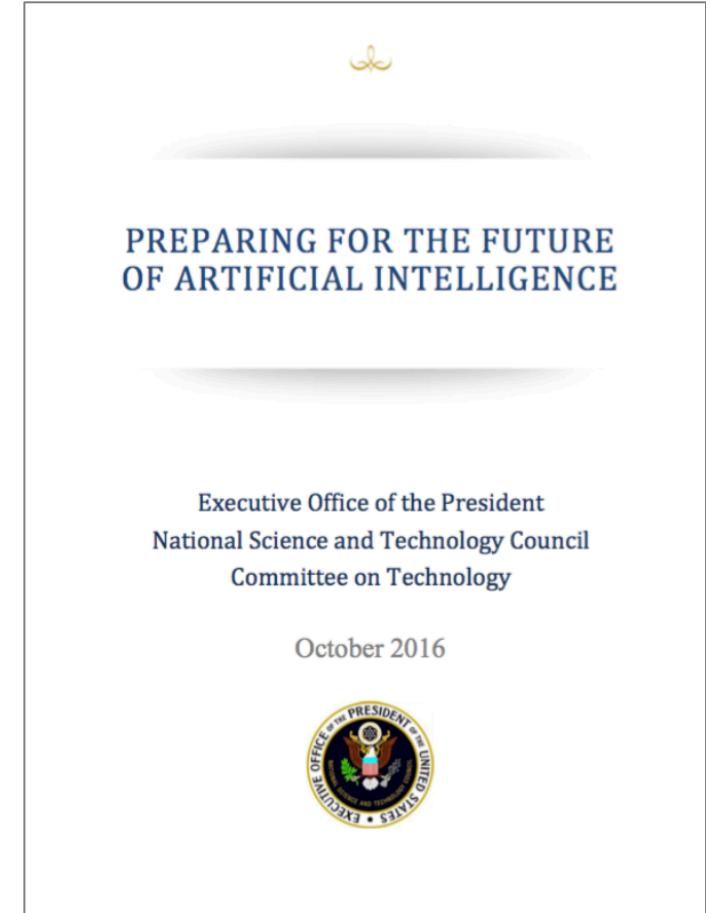


Going up. (Reuters/Kiyoshi Ota)

Computers have slowly started to encroach on activities we previously believed only the brilliantly sophisticated human brain could handle. IBM's Deep Blue supercomputer beat Grand Master Garry Kasparov at chess in 1997, and in 2011 IBM's Watson beat former human winners at the quiz game *Jeopardy*. But the ancient board game Go has long been one of the major goals of artificial intelligence research. It's understood to be one of the most difficult games for computers to handle due to the sheer number of possible moves a player can make at any given point. Until now, that is.

AI의 미래를 위한 준비(미국 백악관 보고서)

- 인공지능은 향후 20년간 인간의 지능을 뛰어넘을 수 없을 것으로 판단된다.
- 하지만 언젠가는 인간과 대등해 질 것이고 여러 분야에서 인간의 능력을 뛰어 넘을 것으로 예측된다



AI의 역사

- 용어 자체는 1956년에 최초로 만들어 졌지만 이에 대한 아이디어는 Alan Turing이 1950년 발행한 논문인 “Computing Machinery and Intelligence”에서 본격적으로 시작
- “기계는 생각을 할 수 있을까? ” 라는 질문을 제기

2010년 전후의 AI 발전 요인

1. 전자상거래, 비즈니스, 소셜미디어, 과학, 정부 등의 많은 데이터 생산
2. 이렇게 생성된 데이터는 획기적으로 향상된 기계학습 알고리즘의 재료로 쓰임
3. 강력해진 컴퓨터 능력을 활용

2010년 전후의 AI 발전 요인

1. 많은 데이터

= IoT 발달로 인한 데이터의 축적

2. 알고리즘

= 기존의 학습 알고리즘 + Deep learning

3. 컴퓨터 POWER

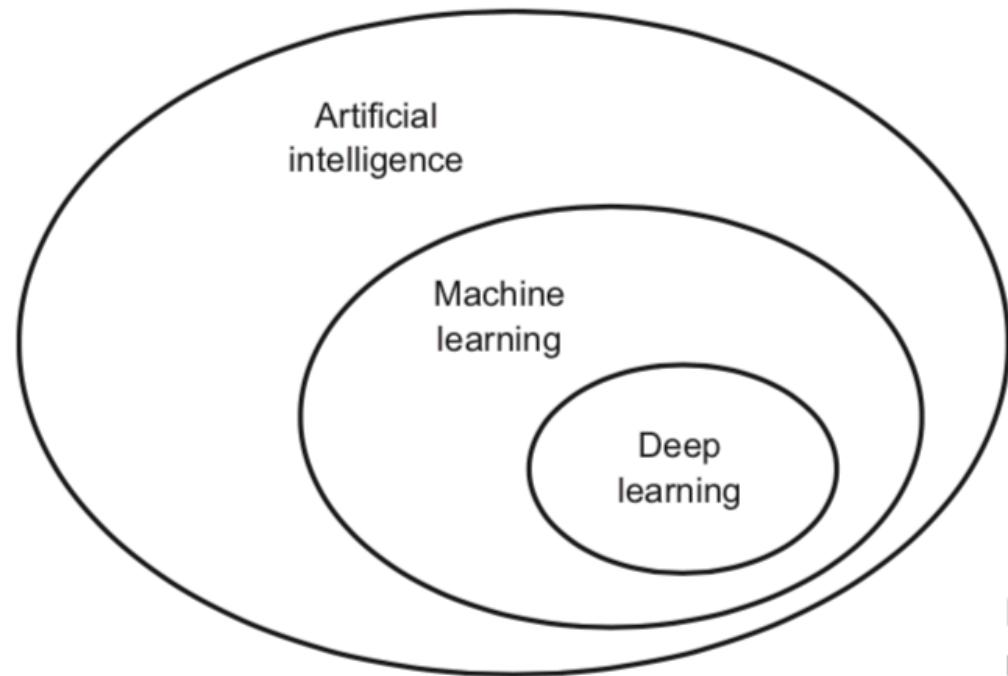
= 컴퓨터의 발달 + 클라우드 컴퓨터(AWS, Google Cloud Platform)

AI는 무엇인가?

- Stuart Russell and Peter Norvig의 분류
 1. 인간처럼 생각하는 시스템(인지적 구조)
 2. 인간처럼 행동하는 시스템(언어처리, 지식표현, 자동추론)
 3. 합리적으로 생각하는 시스템(논리풀이, 추론, 최적화)
 4. 합리적으로 행동하는 시스템(인식, 추리, 학습, 대화, 의사결정, 행동을 이뤄내는 로봇)

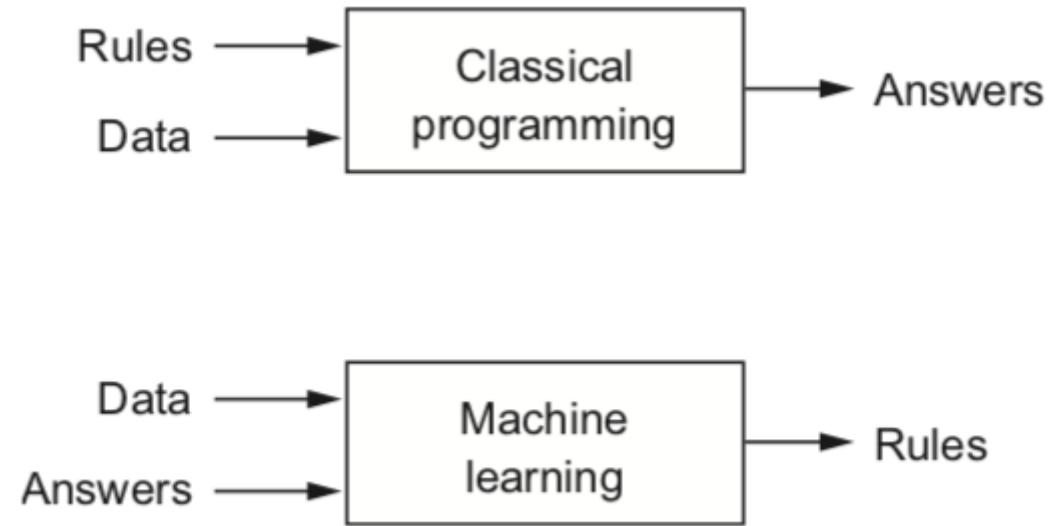
Machine learning (기계학습)

- “특정한 작업을 수행하는 방법을 기계가 스스로 학습할 수 있는가?”
- “프로그래머가 직접 데이터 처리 규칙을 작성하는 대신 컴퓨터가 데이터를 보고 이러한 규칙을 자동으로 학습할 수 있을까?”



A new programming paradigm

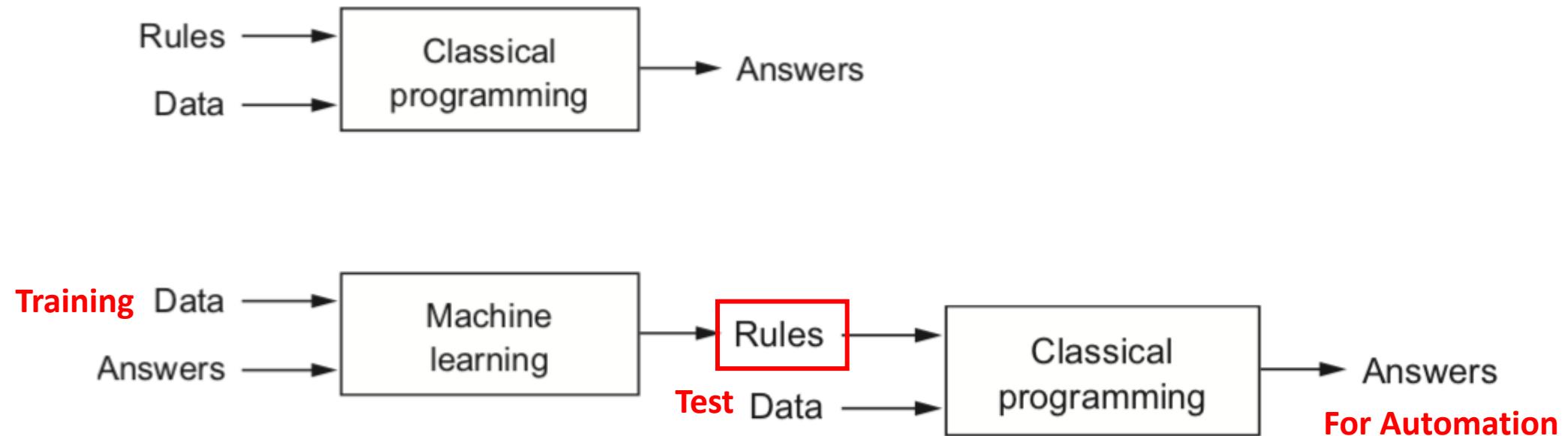
- 고전적인 인공지능 패러다임에서 인간은 규칙을 입력하고 이에 따라 처리할 데이터를 입력하고 **답변을 얻음**
- 머신러닝을 사용할 때는 규칙과 답변을 입력한 후 **규칙을 얻음**



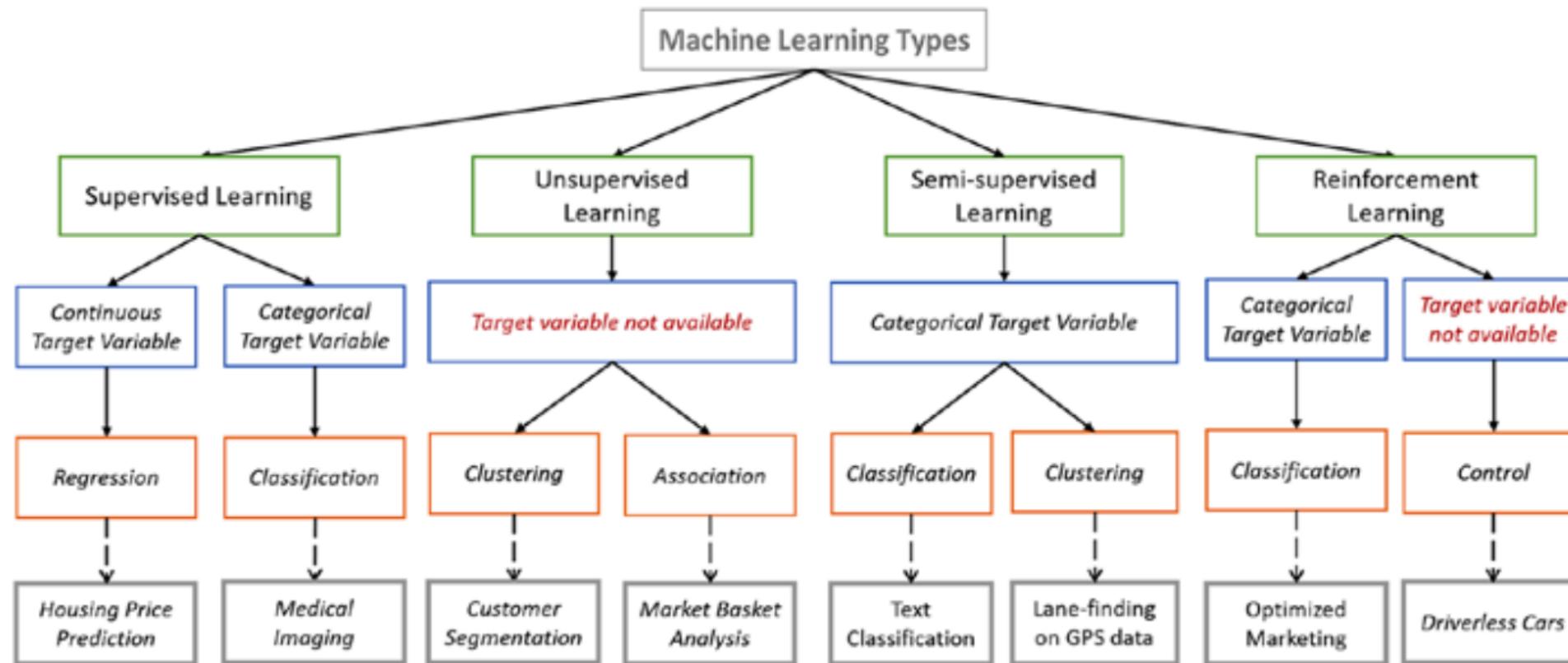
A new programming paradigm

- ML은 관련된 많은 사례를 접한 후
- 이 사례속에서 통계적 구조를 찾아내어
- 자동화 하는데 필요한 규칙을 제시

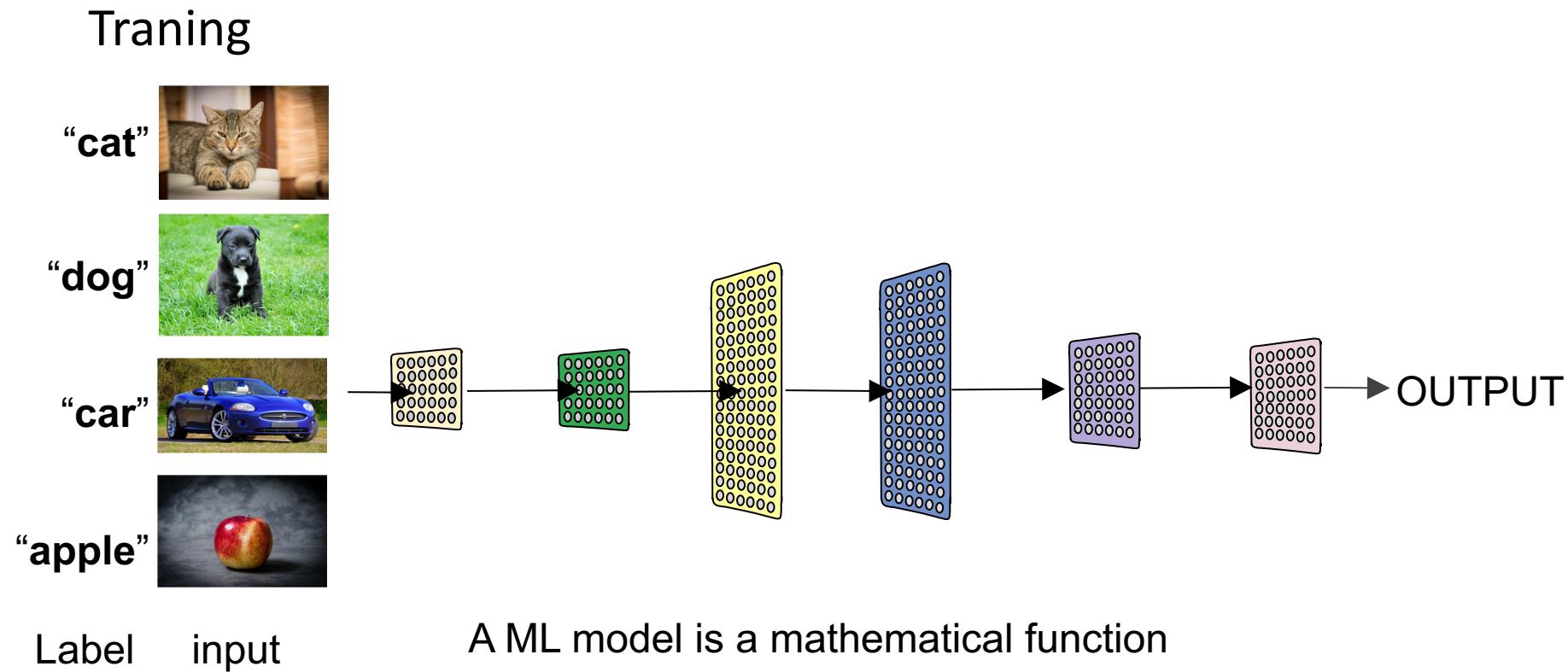
A new programming paradigm



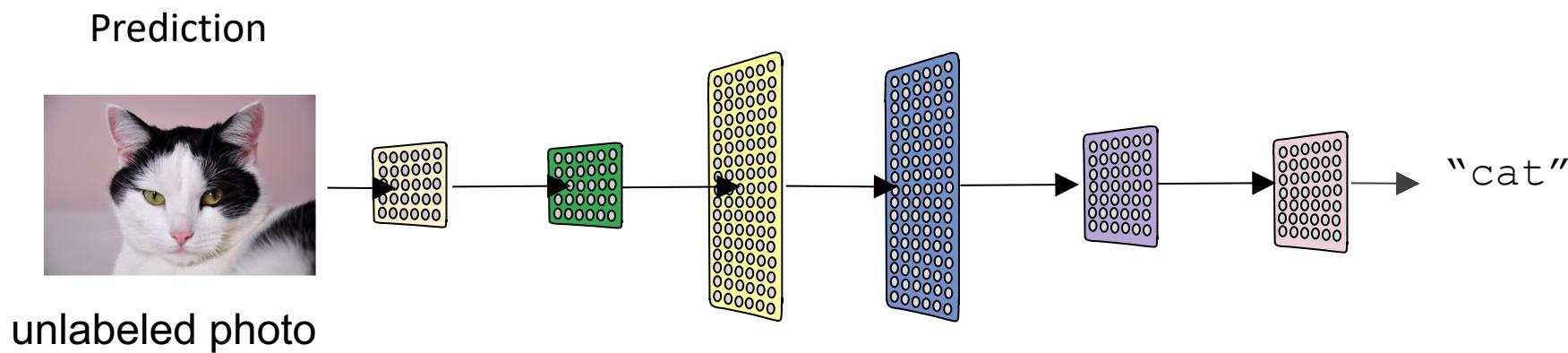
Types of Machine Learning



Supervised Learning

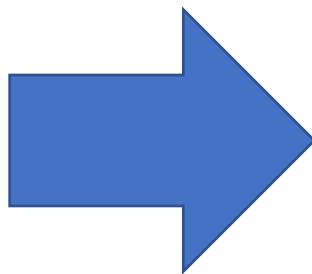
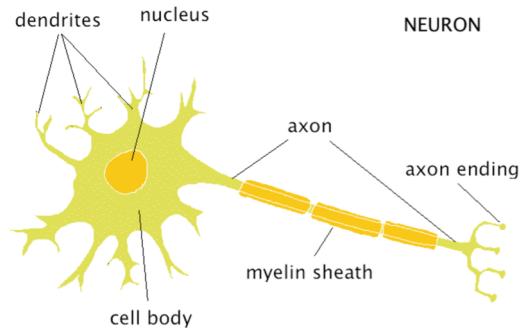


Supervised Learning

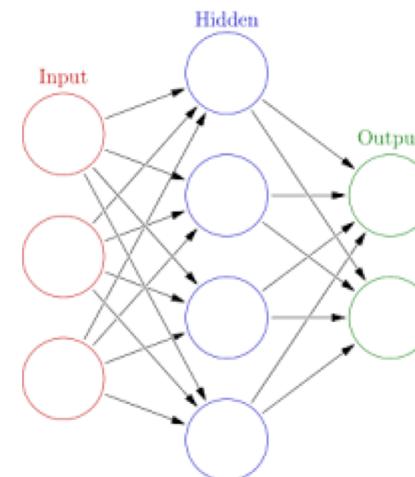


Deep learning

인간 두뇌의 뉴런 구조



뉴럴 네트워크



Baby learns from their experiences,
books, and parents.



AI learns from their own data, other models, and programmers.

Brain == Model

Senses == Sensors

Action



AI, AI, AI

AI가 활용될 수 있는 분야

- IT
- 제조업
- 의료
- 농업/축산업
- 데이터로부터 판단이 필요한 모든 분야

AI, ML의 한계점

- 학습한 범위 밖에서의 유연성이 거의 없고 상식도 없음
- 논리적인 다음 단계를 상식으로 추론할 수 없음
- 기계가 새로운 개념을 찾아낼 수 있을까?

<https://github.com/YoungjunNa/2019-intelligent-livestock-facilities>

물 들어올 때 노젓자! 방학특강 + 실습

"Animal Science with Data Science!"

17th: R Programming (6 hours)

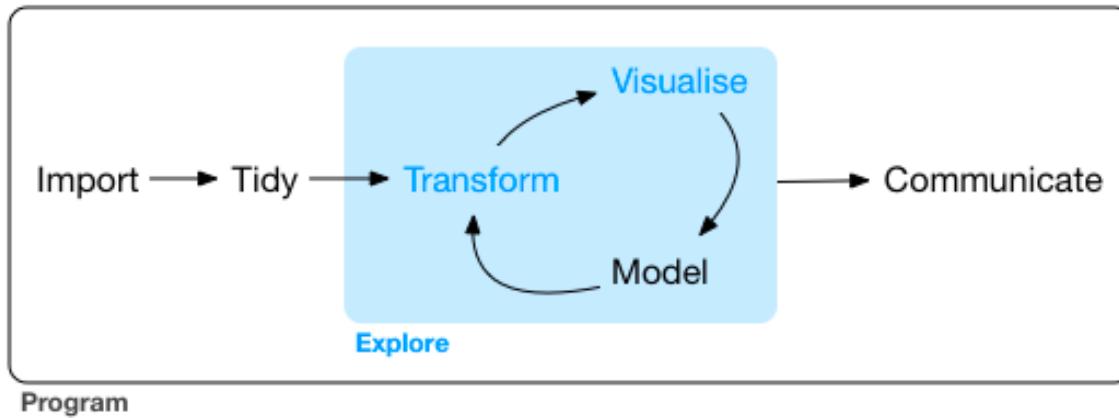
- 함수: `function`
- 조건부 실행: `if`
- 벡터: 원자벡터와 리스트
- `for` 루프
- 반복작업: `purrr`

18th: Modeling (6 hours)

- Modeling 기초
- Machine learning and AI
- 분류: 최근접 이웃 분류(kNN)
- 수치 데이터 예측: 회귀
- 신경망과 서포트 벡터 머신
- 클러스터링: k-평균 군집화
- Deep Learning with `keras`

데이터과학의 과정

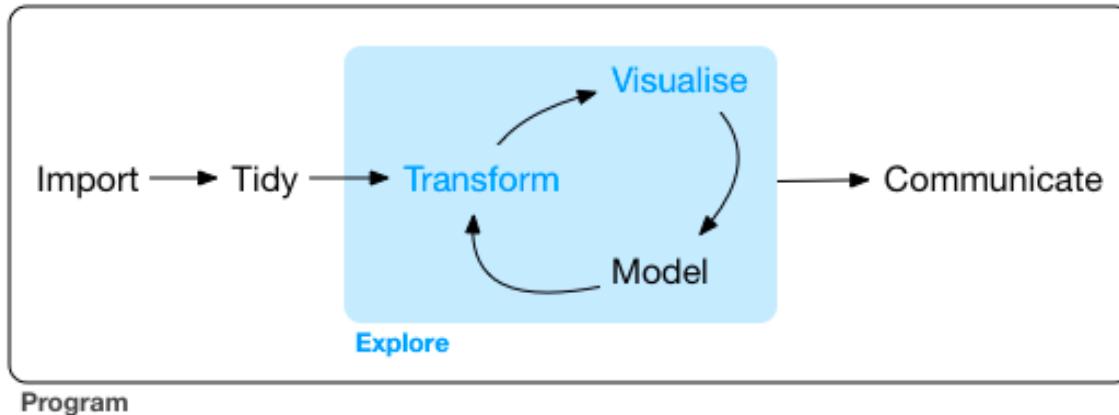
데이터 과학 프로젝트의 과정



1. Import (불러오기)

2. Tidy (정리하기): 일관된 형식으로 -> 데이터에 집중할 수 있게

데이터 과학 프로젝트의 과정



3. Transform (데이터 변형):

- 관측값의 범위를 좁히고(예: 여자만/남자만 보자!)
- 새로운 변수를 작성(예: 거리와 시간으로부터 속도를 계산)
- 요약통계 계산(빈도나 평균)

데이터 과학 프로젝트의 과정

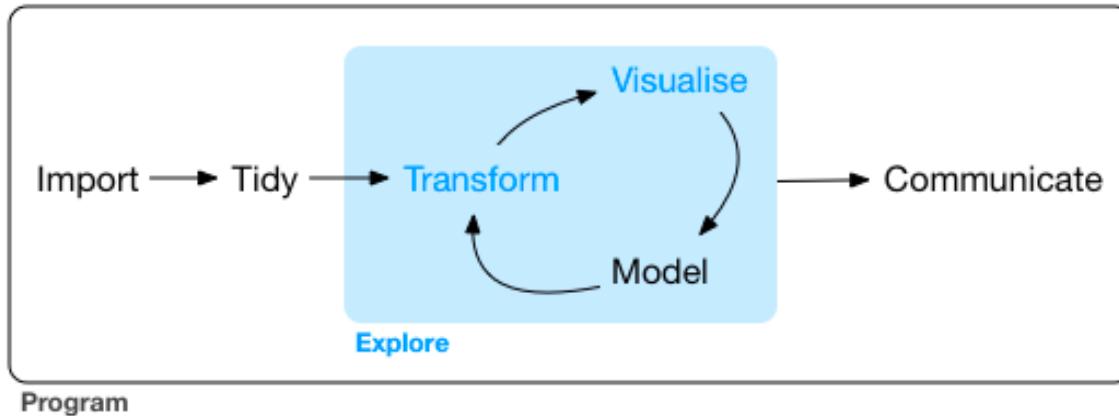
4. Visualization (시각화 하기):

- 지식을 생성하는 방법 -> 시각화와 모델링
- 시각화는 근본적으로 인간의 활동

5. Modeling (모델링)

- 문제가 충분히 정교하다면 모델을 사용하여 문제의 답을 확인
- 근본적으로 수학/전산도구
- 사람을 채용하는 것보다 컴퓨터를 사는게 더 저렴하다

데이터 과학 프로젝트의 과정



6. Communication (의사소통):

- 데이터과학의 마지막 단계
- 결과 내용을 다른 사람에게 효과적으로 전달

그리고 프로그래밍..

- 프로그래밍은 이 모든 것을 위해 사용되는 다용도 도구
- 전문 프로그래머가 될 필요는 없지만 프로그래밍은 반복작업을 자동화 하고 문제를 더 쉽게 해결할 수 있게 도와줄 수 있음

Python, Julia, C+, SAS, SPSS

- 모두 다 유용한 도구
- 일부는 유료(SAS, SPSS)
- 한번에 하나를 깊게 들어가는 것이 좋다
- 이왕이면 무료로

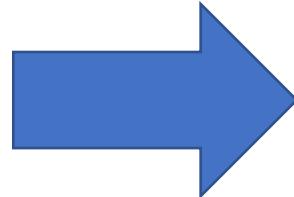
오픈소스 언어



: 바퀴를 다시 발명하지 마라

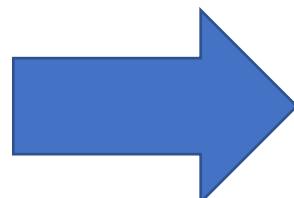
모든 것을 다 알고 시작할 수 없다

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$$



=average(A1:A4)

$$SD = \sqrt{\frac{\sum (x - \bar{x})^2}{N - 1}}$$



=stdev(A1:A4)

모든 것을 다 알고 시작할 수 없다

Gradient descent algorithm

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \left[\frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) \right]$$
$$\theta_1 := \theta_1 - \alpha \left[\frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)} \right]$$

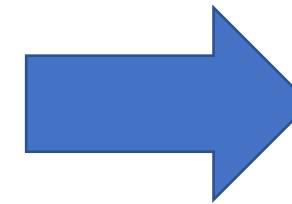
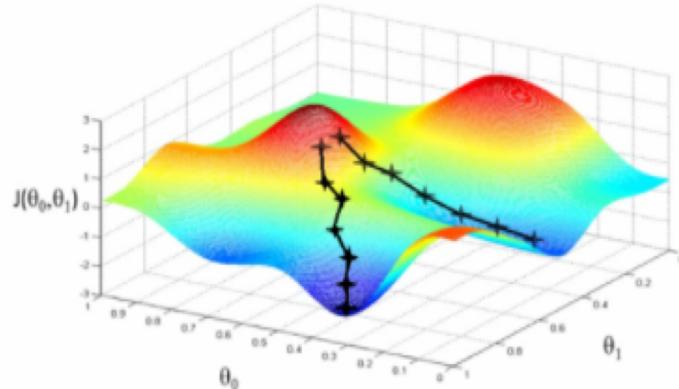
}

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

update
 θ_0 and θ_1
simultaneously

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$h_\theta(x) = \underline{\theta_0 + \theta_1 x}$$



`tf.train.GradientOptimizer()`

코드 한줄로 끝!

오픈소스의 지향점



1. 소스코드 공개
2. 무료로 이용
3. 더 많은 사람들이 사용
4. 버그발견, 기능추가, 문서보강, 새로운 아이디어

GNU 선언문(1985; 1993 개정)



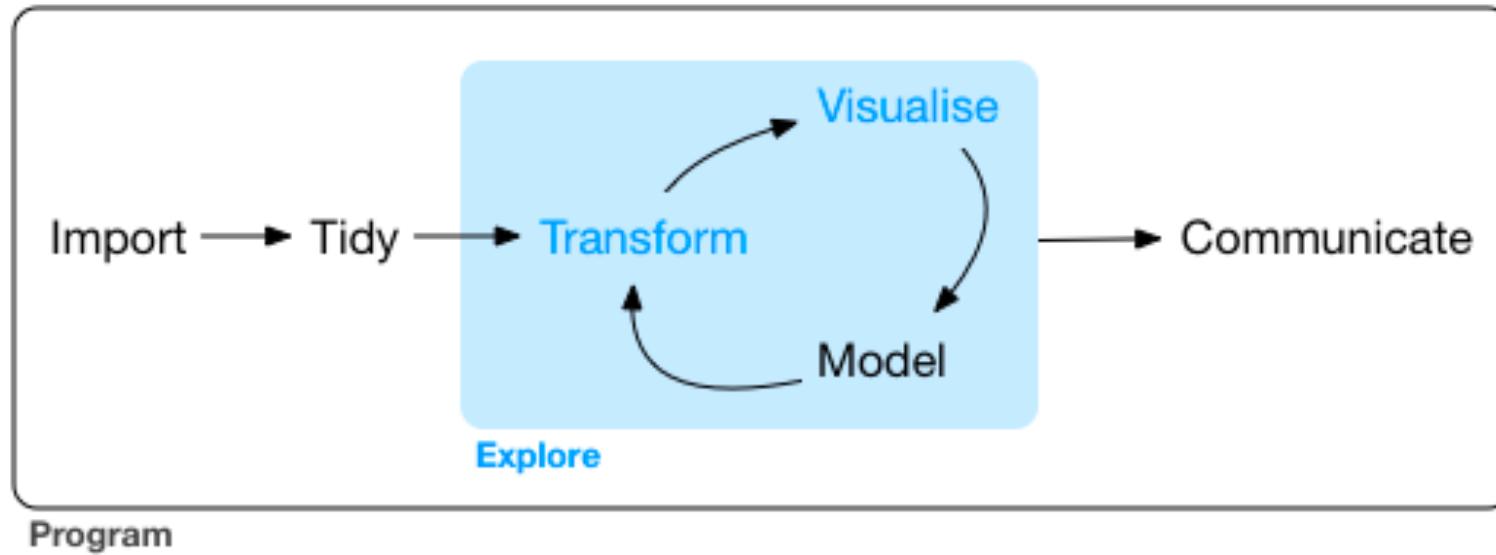
- <https://www.gnu.org/gnu manifesto.ko.html>
- Once GNU is written, everyone will be able to obtain good system software free, just like air.
- This means much more than just saving everyone the price of a Unix license. It means that much wasteful duplication of system programming effort will be avoided. This effort can go instead into advancing the state of the art.
- 일단 GNU가 만들어지면, 모든 사람들은 훌륭한 시스템 소프트웨어를 공기처럼 무료로 얻을 수 있게 될 것이다.
- 이것은 모든 사람이 단지 유닉스 사용에 대한 라이선스 비용을 절약할 수 있다는 것보다 훨씬 더 많은 의미를 갖는다. 이것은 시스템 프로그래밍에 소모되는 불필요한 노력의 증복을 피할 수 있게 된다는 것을 의미하며, 절약된 노력은 기술 수준을 향상시키는데 사용될 수 있을 것이다.

오픈소스 라이센스 R

- R 언어는 GPL (General Public License)를 채택
- 사용자들이 소프트웨어를 자유롭게 공유하고 수정할 수 있음



데이터 과학 프로젝트의 과정



1st REPORT

(1/9)

R & RStudio 설치하기

- <https://cran.r-project.org/>
- <https://www.rstudio.com/products/rstudio/download/>
- Global Option -> Code -> Saving -> Default text encoding -> UTF-8로 설정
- Global Option -> Appearance -> 예쁘게 설정해 보세요
- Console에 `Sys.info()` 입력 후 전체화면을 예쁘게 스크린샷
-> 슬랙 DM으로 제출