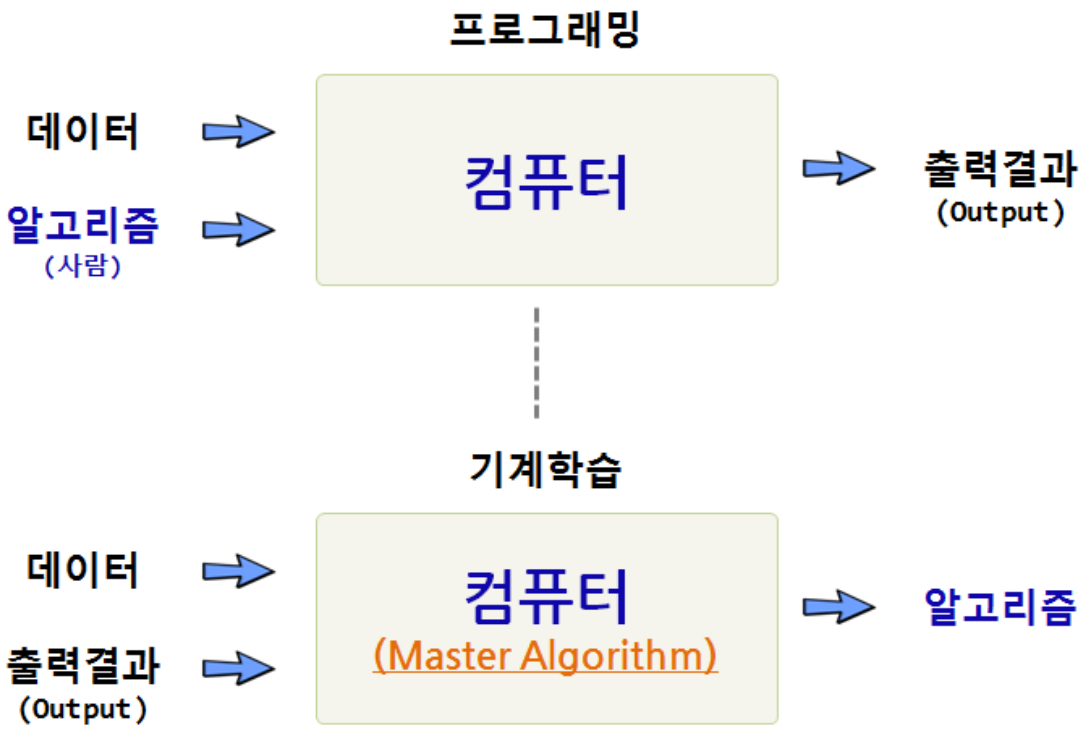
Tidyverse와 기계학습(ML)

데이터뽀개기 2018 - Hello Kaggler!

이광춘
(페북 그룹:Tidyverse Korea)

2018/10/07

Tidyverse 기계학습 들어가며

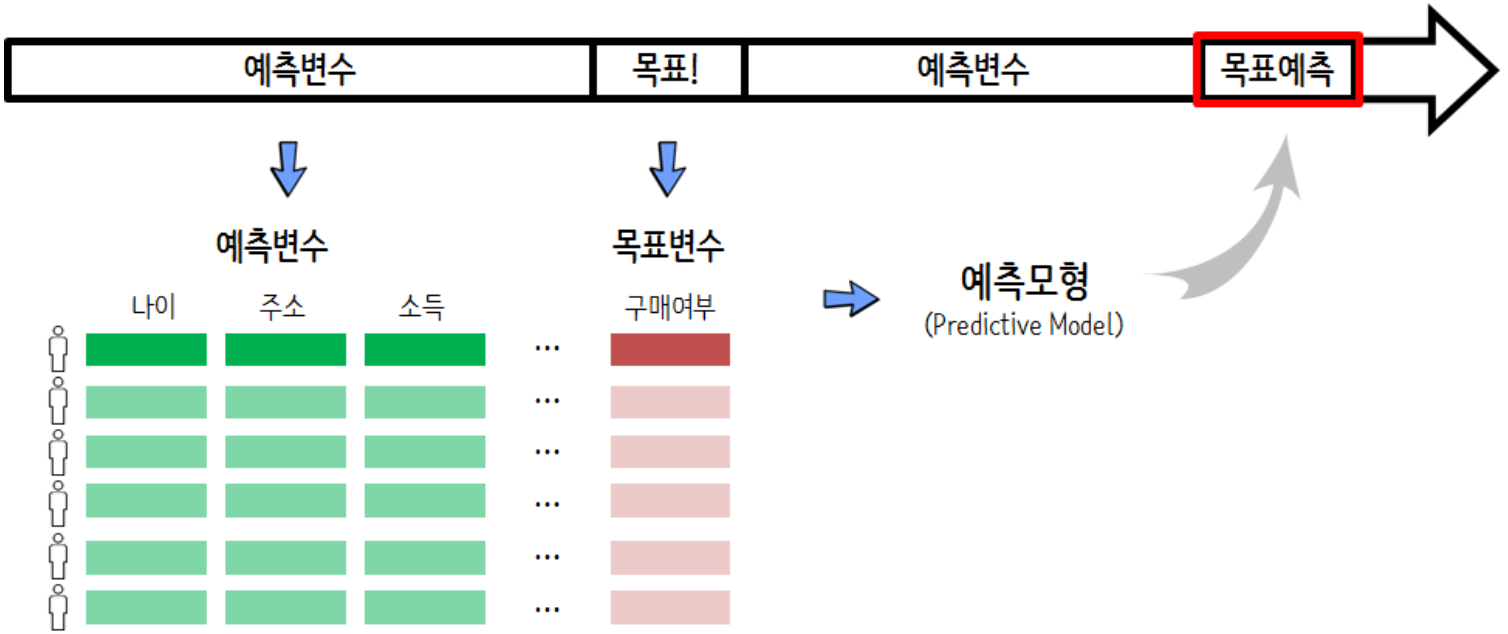

프로그래밍
데이터
알고리즘
(사람)
컴퓨터
출력결과
(Output)
기계학습
데이터
출력결과
(Output)
컴퓨터
(Master Algorithm)
알고리즘

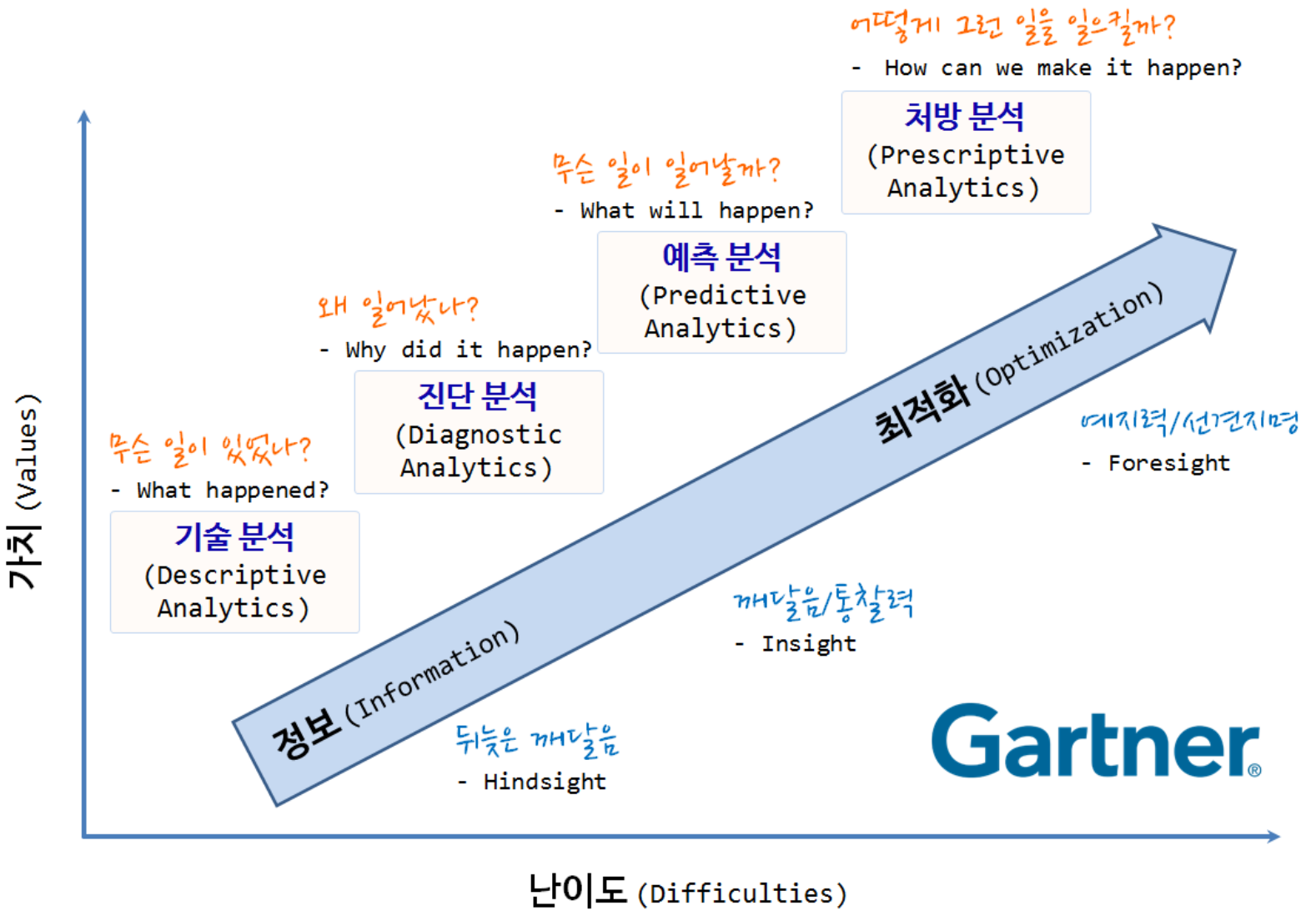
예측모형 데이터프레임

데이터 랭글링(data wrangling)의 목적은 예측모형을 위한 데이터프레임 생성:
Analytic Basetable

예측모형 데이터프레임 작업흐름

축적된 데이터를 통해서 예측변수와 목표변수를 활용하여 예측모형을 개발하여 예측변수를 입력값으로 하여 목표예측을 수행.



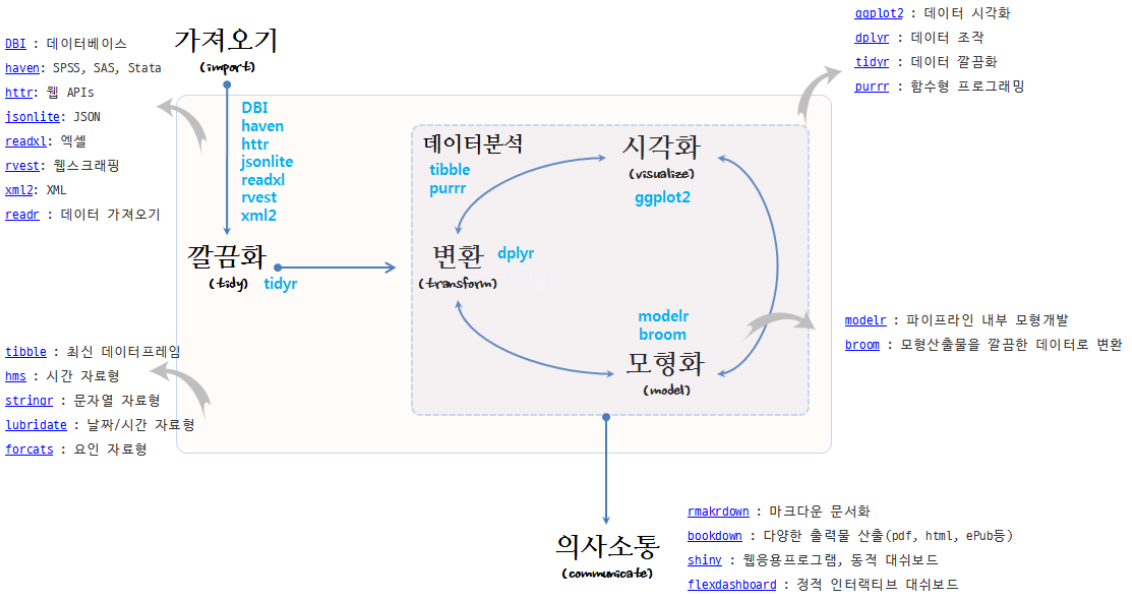

어떻게 그런 일을 일으킬까?
- How can we make it happen?
처방 분석
(Prescriptive Analytics)
무슨 일이 일어날까?
- What will happen?
예측 분석
(Predictive Analytics)
왜 일어났나?
- Why did it happen?
진단 분석
(Diagnostic Analytics)
무슨 일이 있었나?
- What happened?
기술 분석
(Descriptive Analytics)
최적화 (Optimization)
예지력/선견지명
- Foresight
깨달음/통찰력
- Insight
뒤늦은 깨달음
- Hindsight
정보 (Information)
Gartner
가치 (Values)
난이도 (Difficulties)

자동화된 기계학습

수작업 예측모형
(Manual Predictive Model)

자동화된 예측모형
(Automated Predictive Model)

xwMOOC 모형: 기계학습 - gapminer + rsample + purrr



tidyverse **2015년 버전**

- xwMOOC 데이터 과학 -tidyverse 데이터 과학 기본체계
- Tidyverse

tidyverse 성명서(manifesto)

엉망진창인 R 도구상자(messyverse)와 비교하여 깔끔한 R 도구상자(tidyverse)는 깔끔한(tidy) API에 다음과 같은 4가지 원칙을 제시한다.

- 기존 자료구조를 재사용: Reuse existing data structures.
- 파이프 연산자로 간단한 함수를 조합: Compose simple functions with the pipe.
- 함수형 프로그래밍을 적극 사용: Embrace functional programming.
- 기계가 아닌 인간을 위한 설계: Design for humans.

- xwMOOC 데이터 과학 -tidyverse 데이터 과학 기본체계
- Hadley Wickham(2017-11-13), "The tidy tools manifesto"

기계학습 데이터셋

gapminder

Hans Rosling's 200 Countries, 200 Years, 4 Minutes - The Joy of Stats - BBC Four

```
# 0. 환경설정 -----
library(tidyverse)
library(dslabs)
```

회귀모형 - purrr + trelliscopejs

- xwMOOC 모형: 회귀모형 - purrr + trelliscopejs

gapminder 데이터셋

dslabs 팩키지 내장 gapminder 데이터셋 10% 표본 추출 → 데이터 랭글링

gapminder 정제작업

결측값(NA)을 바탕으로 패턴을 추출하고, 1960 ~ 2011년까지 데이터를 예측모형 데이터로 활용함.

```
# 1. 데이터 정제 -----
gapminder_lc <- gapminder %>%
   group_by(country) %>%
   nest()

gapminder_lc_df <- gapminder_lc %>%
   mutate(na_cnt = map_int(data, ~ sum(is.na(.x)))) %>%
   filter(na_cnt == 8) %>%
   unnest(data) %>%
   filter(year <= 2011) %>%
   select(-na_cnt) %>%
   select(-continent, -region)
```

훈련/검증/시험 데이터셋 (코드)

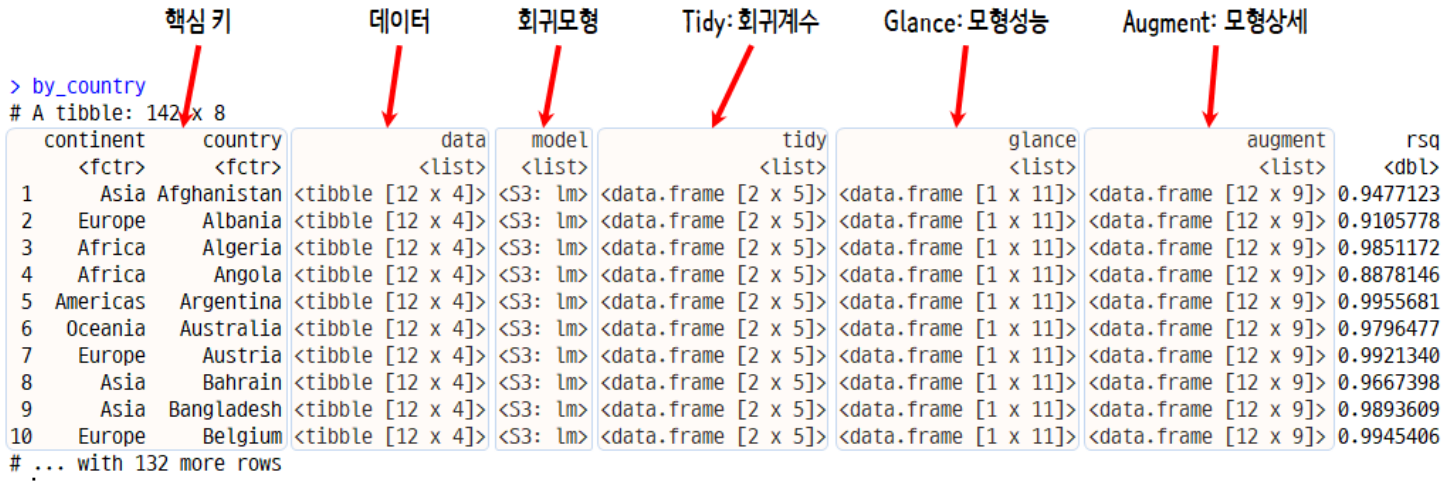
```
library(rsample)

## 훈련/시험 데이터 분할
gapminder_split <- initial_split(gapminder_lc_df, prop = 0.70)

train_df <- training(gapminder_split)
test_df  <- testing(gapminder_split)

## 훈련 데이터를 검증(Cross Validation) 데이터 분할
gapminder_cv_split <- vfold_cv(train_df, v = 5)

cv_df <- gapminder_cv_split %>%
           mutate(train    = map(splits, ~training(.x)),
                  validate = map(splits, ~testing(.x)))

cv_df
```

훈련/검증/시험 데이터셋

```
#  5-fold cross-validation
# A tibble: 5 x 4
  splits       id    train                validate
  <list>       <chr> <list>               <list>
1 <S3: rsplit> Fold1 <tibble [2,242 x 7]> <tibble [561 x 7]>
2 <S3: rsplit> Fold2 <tibble [2,242 x 7]> <tibble [561 x 7]>
3 <S3: rsplit> Fold3 <tibble [2,242 x 7]> <tibble [561 x 7]>
4 <S3: rsplit> Fold4 <tibble [2,243 x 7]> <tibble [560 x 7]>
5 <S3: rsplit> Fold5 <tibble [2,243 x 7]> <tibble [560 x 7]>
```

Tidyverse 기계학습

many models




```
> by_country
# A tibble: 142 x 8
   continent     country              data     model              tidy            glance           augment       rsq
      <fctr>      <fctr>            <list>    <list>            <list>            <list>            <list>     <dbl>
 1      Asia Afghanistan <tibble [12 x 4]> <S3: lm> <data.frame [2 x 5]> <data.frame [1 x 11]> <data.frame [12 x 9]> 0.9477123
 2    Europe     Albania <tibble [12 x 4]> <S3: lm> <data.frame [2 x 5]> <data.frame [1 x 11]> <data.frame [12 x 9]> 0.9105778
 3    Africa     Algeria <tibble [12 x 4]> <S3: lm> <data.frame [2 x 5]> <data.frame [1 x 11]> <data.frame [12 x 9]> 0.9851172
 4    Africa      Angola <tibble [12 x 4]> <S3: lm> <data.frame [2 x 5]> <data.frame [1 x 11]> <data.frame [12 x 9]> 0.8878146
 5  Americas   Argentina <tibble [12 x 4]> <S3: lm> <data.frame [2 x 5]> <data.frame [1 x 11]> <data.frame [12 x 9]> 0.9955681
 6   Oceania   Australia <tibble [12 x 4]> <S3: lm> <data.frame [2 x 5]> <data.frame [1 x 11]> <data.frame [12 x 9]> 0.9796477
 7    Europe     Austria <tibble [12 x 4]> <S3: lm> <data.frame [2 x 5]> <data.frame [1 x 11]> <data.frame [12 x 9]> 0.9921340
 8      Asia     Bahrain <tibble [12 x 4]> <S3: lm> <data.frame [2 x 5]> <data.frame [1 x 11]> <data.frame [12 x 9]> 0.9667398
 9      Asia  Bangladesh <tibble [12 x 4]> <S3: lm> <data.frame [2 x 5]> <data.frame [1 x 11]> <data.frame [12 x 9]> 0.9893609
10    Europe     Belgium <tibble [12 x 4]> <S3: lm> <data.frame [2 x 5]> <data.frame [1 x 11]> <data.frame [12 x 9]> 0.9945406
# ... with 132 more rows
```

회귀모형 적합: 맛보기 (코드)

```
library(broom); library(Metrics)

# 회귀모형 적합
model_cv_df <- cv_df %>%
  mutate(lm_model  = map(train,  ~lm(life_expectancy ~ ., data=.x)))

# 회귀모형 성능 평가
model_cv_df <- model_cv_df %>%
    mutate(valid_actual = map(validate, ~.x$life_expectancy),
           valid_pred   = map2(lm_model, validate, ~predict(.x, .y))) %>%
    mutate(valid_mae    = map2_dbl(valid_actual, valid_pred,
                                   ~mae(actual = .x, predicted = .y)),
           valid_rmse    = map2_dbl(valid_actual, valid_pred,
                                    ~rmse(actual = .x, predicted = .y)))

model_cv_df$valid_mae
# model_cv_df$valid_rmse

mean(model_cv_df$valid_mae)
```

회귀모형 적합: 맛보기

```
       1        2        3        4        5 
1.519359 1.570984 1.509646 1.548877 1.543030 
```

```
[1] 1.538379
```

예측모형 아키텍처 (코드)

```
library(broom); library(e1071);library(ranger);library(extrafont); loadfonts()
# 회귀모형 적합
model_cv_df <- model_cv_df %>%
    mutate(lm_model  = map(train,  ~lm(life_expectancy ~ ., data=.x)),
           rf_model  = map(train,  ~ranger(life_expectancy ~ ., data=.x)),
           svm_model = map(train,  ~svm(life_expectancy ~ ., data=.x,  probability
# 회귀모형 성능평가
model_cv_df <- model_cv_df %>%
    mutate(valid_actual = map(validate, ~.x$life_expectancy),
           valid_lm_pred   = map2(lm_model, validate, ~predict(.x, .y)),
           valid_rf_pred   = map2(rf_model, validate, ~predict(.x, .y)$predictions
           valid_svm_pred  = map2(svm_model, validate, ~predict(.x, .y))) %>%
    mutate(valid_lm_mae    = map2_dbl(valid_actual, valid_lm_pred,
                                      ~mae(actual = .x, predicted = .y)),
           valid_rf_mae    = map2_dbl(valid_actual, valid_rf_pred,
                                      ~mae(actual = .x, predicted = .y)),
           valid_svm_mae   = map2_dbl(valid_actual, valid_svm_pred,
                                      ~mae(actual = .x, predicted = .y)))
```

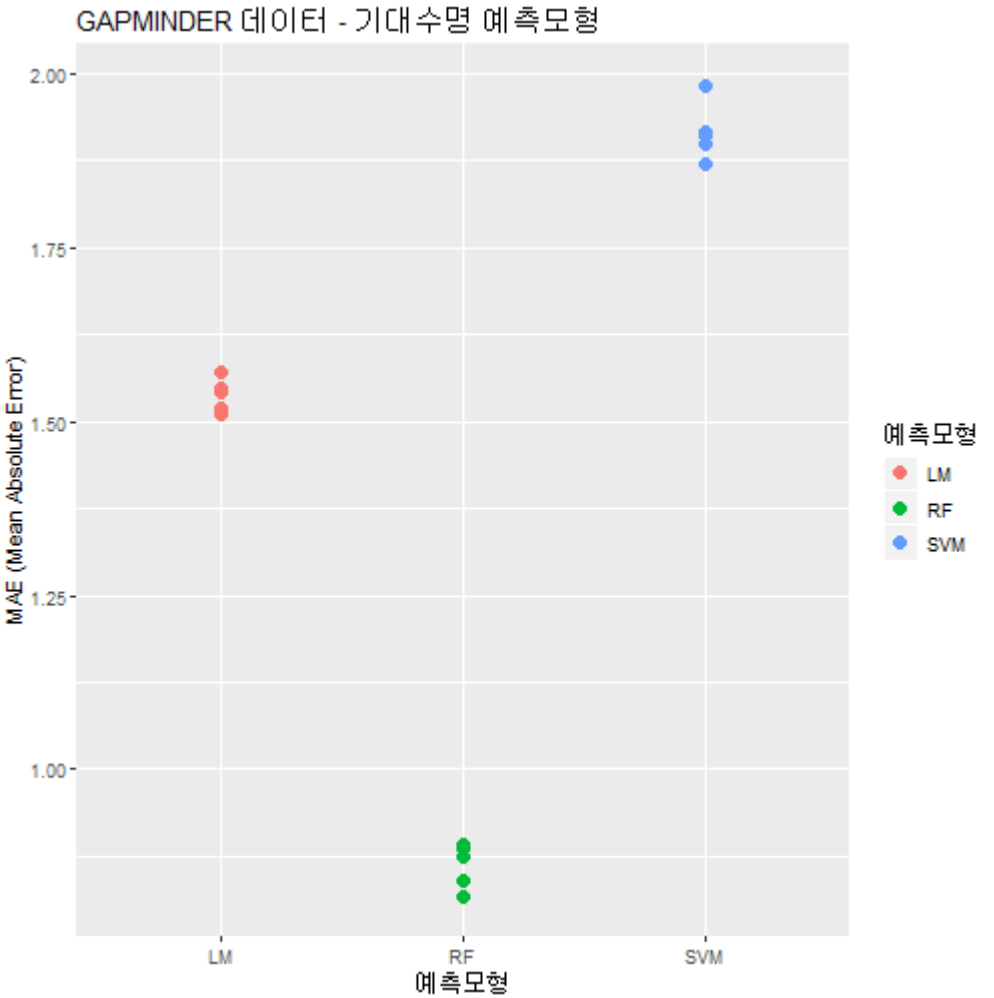
예측모형 아키텍처

```
#  5-fold cross-validation
# A tibble: 5 x 17
  splits  id    train  validate lm_model valid_actual valid_pred valid_mae
  <list>  <chr> <list> <list>   <list>   <list>       <list>          <dbl>
1 <S3: r~ Fold1 <tibb~ <tibble~ <S3: lm> <dbl [561]>  <dbl [561~      1.52
2 <S3: r~ Fold2 <tibb~ <tibble~ <S3: lm> <dbl [561]>  <dbl [561~      1.57
3 <S3: r~ Fold3 <tibb~ <tibble~ <S3: lm> <dbl [561]>  <dbl [561~      1.51
4 <S3: r~ Fold4 <tibb~ <tibble~ <S3: lm> <dbl [560]>  <dbl [560~      1.55
5 <S3: r~ Fold5 <tibb~ <tibble~ <S3: lm> <dbl [560]>  <dbl [560~      1.54
# ... with 9 more variables: valid_rmse <dbl>, rf_model <list>,
#   svm_model <list>, valid_lm_pred <list>, valid_rf_pred <list>,
#   valid_svm_pred <list>, valid_lm_mae <dbl>, valid_rf_mae <dbl>,
#   valid_svm_mae <dbl>
```

예측모형 아키텍처 성능 비교 (코드)

```
model_df <- data.frame(LM = model_cv_df$valid_lm_mae,
                       RF = model_cv_df$valid_rf_mae,
                       SVM = model_cv_df$valid_svm_mae)

model_df %>% gather(model, MAE) %>%
    ggplot(aes(x=model, y=MAE, color=model)) +
       geom_point(size=3) +
       labs(x="예측모형", y="MAE (Mean Absolute Error)",
            color="예측모형",
            title="GAPMINDER 데이터 - 기대수명 예측모형")
```

예측모형 아키텍처 성능 비교



예측모형 튜닝 (코드)

```
# Random Forest 모형적합
model_cv_df <- model_cv_df %>%
  crossing(mtry = c(2,ceiling(sqrt(ncol(gapminder_lc_df)-2)),5),
           num.trees = c(500, 1000))  %>%
  mutate(rf_tune_model  = pmap(list(train, mtry, num.trees),
                               ~ranger(life_expectancy ~ ., data=.x, mtry=.y)))

# RandomForest 성능평가
model_cv_df <- model_cv_df %>%
    mutate(valid_actual = map(validate, ~.x$life_expectancy),
           valid_rf_tune_pred   = map2(rf_tune_model, validate,
                                       ~predict(.x, .y)$predictions)) %>%
    mutate(valid_rf_tune_mae    = map2_dbl(valid_actual, valid_rf_tune_pred,
                                           ~mae(actual = .x, predicted = .y)))

model_cv_df %>%
    group_by(mtry, num.trees) %>%
    summarise(mean_mae = mean(valid_rf_tune_mae))
```

예측모형 튜닝

```
# A tibble: 6 x 3
# Groups:   mtry [?]
   mtry num.trees mean_mae
  <dbl>     <dbl>    <dbl>
1     2       500    0.862
2     2      1000    0.868
3     3       500    0.868
4     3      1000    0.871
5     5       500    0.879
6     5      1000    0.876
```

예측 성능: 시험데이터 (코드)

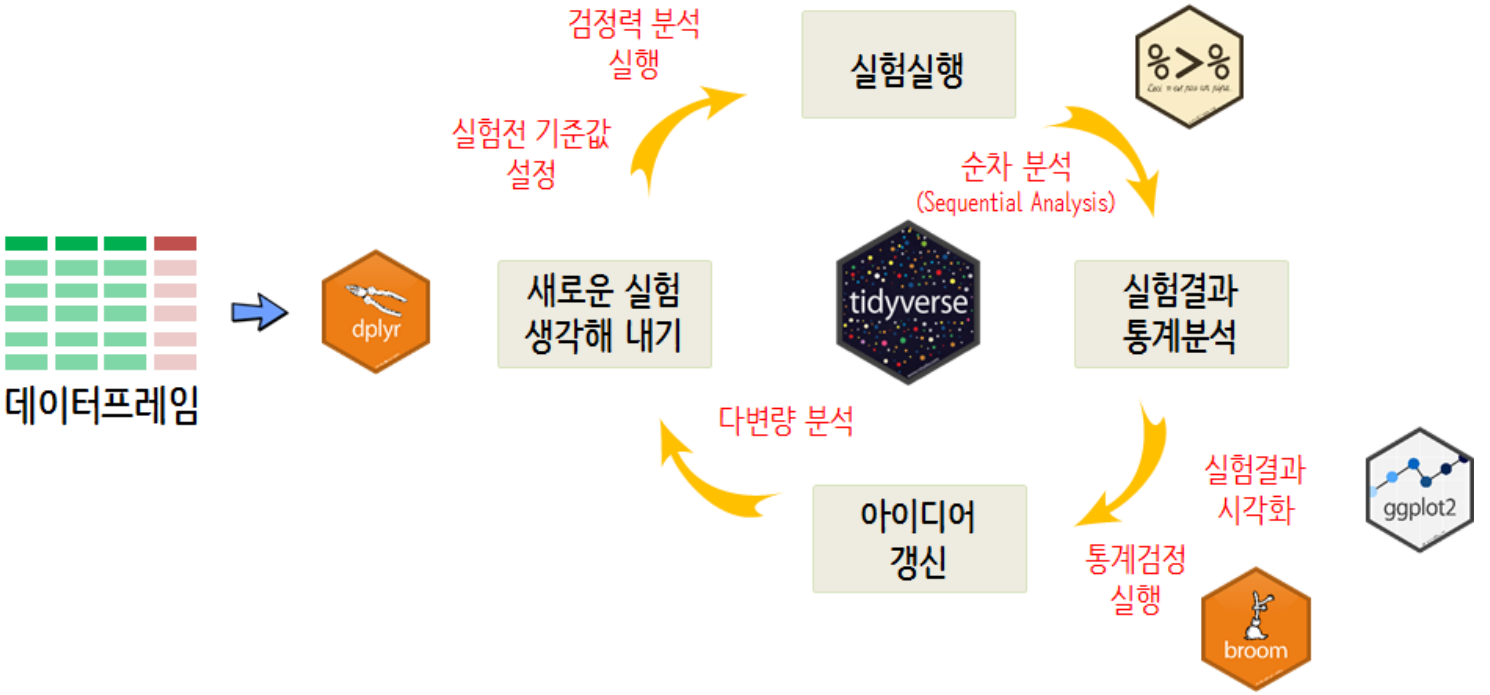
```
gapminder_pm <- ranger(life_expectancy ~ ., data = train_df,
                       mtry = 3, num.trees = 500)

test_df$pred <- predict(gapminder_pm, test_df)$predictions

test_df %>%
    mutate(absolute_err = abs(life_expectancy-pred)) %>%
    summarise(mae = mean(absolute_err))
```

예측 성능: 시험데이터

```
# A tibble: 1 x 1
    mae
  <dbl>
1 0.766
```

열심히 한 캐글 그리고 ...

Tidyverse와 A/B 검정



- 데이터 과학 - tidyverse 데이터 과학 기본체계
  - Tidyverse와 함께 하는 A/B 테스팅

Tidyverse Korea 페북 그룹

https://www.facebook.com/groups/tidyverse/

혹시... 시간이 남으면...

직사각형 모형데이터를 넘어

- 지리정보: xwMOOC 모형 - 나무모형과 지리정보 만남 - 택시
- 자연어: xwMOOC 자연어 처리 - 텍스트 SMS 스팸분류 - Random Forest
- 이미지: xwMOOC 딥러닝 - R 케라스(keras), 패션 MNIST
- 추천: 스파크 + MovieLens 데이터
- 소리
- ...