Imputations for Walmart Data

Sanghoon Park

March 19th Kaggle Study Presentation

github.com/statisticsplaybook/kaggle-study

Missing data

Values that are not available
that would be meaningful if not observed.

Type of Missing Data

Missing Completely at Random (MCAR)

Missing at Random (MAR)

Missing Not at Random (MNAR)

Expectation

Missing Data를 "적절하게" 채워넣으면,
더 좋은 추정치를 얻을 수 있을 것.

But How?

Missing Completely at Random (MCAR)

Missing not related to X or Y

Missingness unrelated to values of variables, either missing or observed

If completely random, results unbiased

However, missing data rarely MCAR

Missing Completely at Random (MCAR)

Missing not related to Y, but may be related to X

Missingness unrelated to values of variables, either missing or observed

Cause of missing data unrelated to missing values

But may be related to observed values of other variables

Missing Not at Random (MNAR)

Missing related to Y

Relationship between missingness and values

Also, missingness due to unobserved predictors

if non-random, must be modeled or else risks bias (non-ignorable)

Approaches

Single-imputation methods

- Substitution
- Carrying-forward
- Interpolation

Multiple-imputation methods

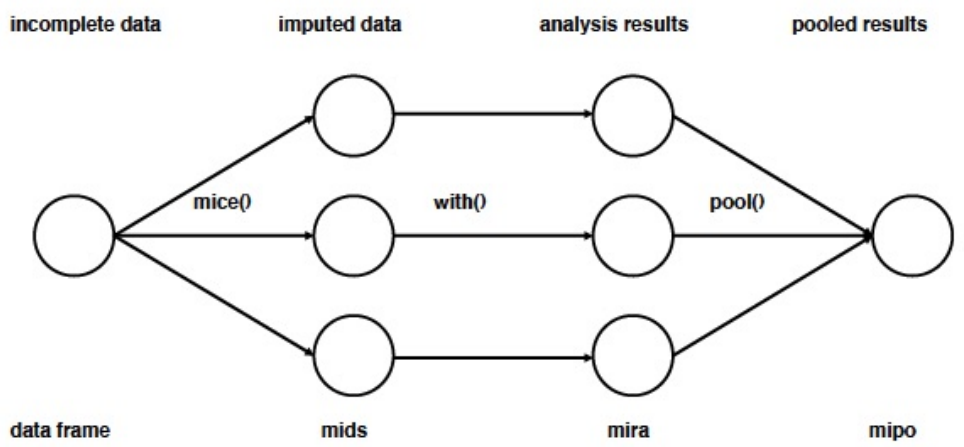


Figure 1: Main steps used in multiple imputation.

Single Imputation Methods

Goal:

Produce complete dataset by filling in missing values

Substitution

e.g. mean, median, mode substitution

But mean substitute based on assumption of normal distribution

Linear interpolation

Use function to extrapolate missings

Follows expected trends

Model Based Methods

Regression | Multiple imputation | k-nearest neighbors

Regression

Uses results from regression to fill in missing values

K-Nearest Neighbors

Mean of the k values coming from the k most similar complete observations

Multiple imputation

Uses the distribution of the observed data to estimate plausible values

Random components incorporated to reflect uncertainty

Individually analyze multiple datasets $\longrightarrow$ Estimates combined

Score Comparisons

`Park-benchmark-comparison-mar-19.R`

Building benchmark model:

```
# Load packages
library(tidyverse)
library(tidymodels)
library(magrittr)
library(knitr)
```

Score Comparisons

Import dataset as `train` and `test`

```
train <- read_csv(
  here("data/walmart/train.csv.zip"))
test <- read_csv(
  here("data/walmart/test.csv.zip"))
```

Combine two datasets

```
all_data <- bind_rows(train, test) %>%
    janitor::clean_names()
```

Benchmark model: lm with store

전처리를 위한 recipe 설정과 X 추출

```
bench_recipe <- all_data %>%
    recipe(weekly_sales ~ store) %>%
    step_mutate(
        store = as.factor(store)) %>%
    prep()
# Produce X manipulated by recipe
benchmark <- juice(bench_recipe)
```

Fitting the model

Train과 Test 분리

```
index <-
  seq_len(nrow(train))
train_bench <-
  benchmark[index,]
test_bench <-
  benchmark[-index,]
```

lm 엔진을 이용하여 fit.

```
lm_benchmark_fit <-
    linear_reg() %>%
  set_engine("lm") %>%
    fit(weekly_sales ~ store,
        data = train_bench)
```

Fitting the model

전처리한 데이터로 `lm` 엔진을 통해 계산한 계수들을 이용

`test` 데이터의 예측변수로 `NA`인 `weekly_sales` 추정

```
result_bench <- predict(lm_benchmark_fit,
                        test_bench)
```

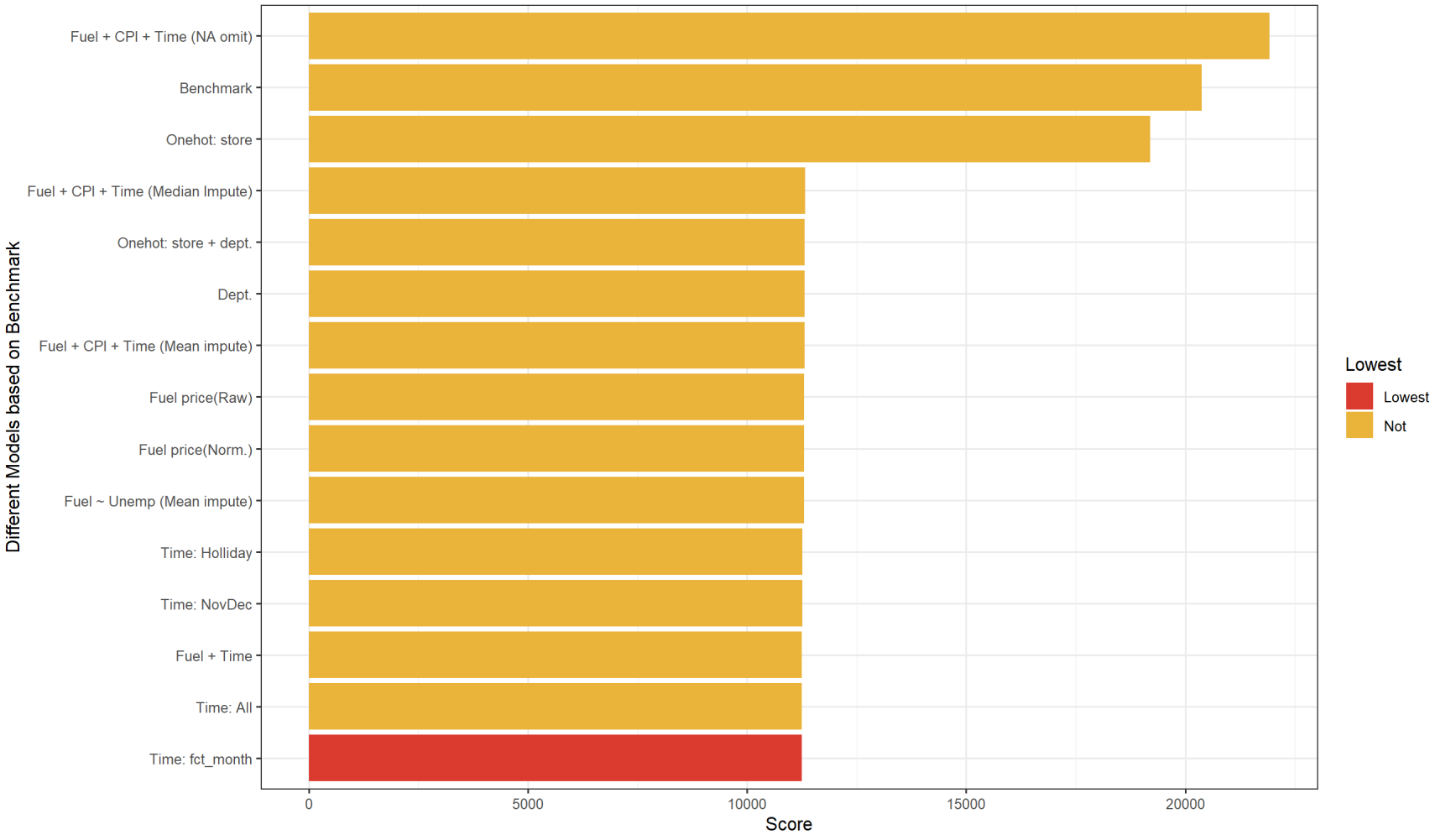
Submit to the Kaggle

1. Kaggle에서 받아온 제출용 공양식을 R로 불러온다.
2. 공란인 test의 Weekly_Sales를 추정한 모델의 weekly_sales로 대체한다.
3. 다시 엑셀 파일로 내보내 제출한다.

Submit to the Kaggle

```
submission_bench <-
  read_csv("디렉토리/sampleSubmission.csv")
submission_bench$Weekly_Sales <- result_bench$.pred
write.csv(submission_bench,
          row.names = F,
          "디렉토리/benchmark_model.csv")
```

Model fits


Fuel + CPI + Time (NA omit)
Benchmark
Onehot: store
Fuel + CPI + Time (Median Impute)
Onehot: store + dept.
Dept.
Fuel + CPI + Time (Mean impute)
Fuel price(Raw)
Fuel price(Norm.)
Fuel ~ Unemp (Mean impute)
Time: Holliday
Time: NovDec
Fuel + Time
Time: All
Time: fct_month
Different Models based on Benchmark
0
5000
10000
15000
20000
Score
Lowest
Lowest
Not

Let's play with Walmart Data