



Anomaly Detection In R

▲ Content

01 | 정의

02 | Statistical Tests

03 | Model-Based

04 | 사례 연구 (논문 소개)



Define

이상값은 즉 다른 데이터 포인트와 크게 다른 데이터 포인트와 거리가 먼 값 또는 관측치

관찰은 실제로 이상치라고 부르기 전에 항상 동일한 현상에 대한 다른 관찰과 비교되어야 함.

ex) 실제로 키가 200cm (미국의 경우 6'7") 인 사람은
일반 인구에 비해 이상치로 간주 될 가능성이 높지만
농구 선수의 키를 측정한 경우 동일한 사람이 이상치로 간주되지 않을 수 있음

[Outliers detection in R](#)



Define

Anomaly Detection : ‘Anomaly’는 정상(normal)의 반대 개념이며 개념 정의를 위해서는 ‘Normal’에 대한 정의부터 내려야 한다. ‘정상’에 대한 개념은 각 분야 및 문제마다 다르게 정의될 수 있기 때문에 ‘이상’에 대한 개념 역시 다 다르게 정의될 수 있음
(Novelty Detection, Outlier Detection, …)

Anomalies are also referred to as outliers, novelties, noise, deviations and exception

Novelty Detection

Outlier Detection

Out-of-distribution Detection



Define

Anomaly Detection

Novelty Detection : Training Data 가 Outlier 에 의해서 오염되지 않은 상태에서 새로운 관측치로부터 이상치를 확인하는 것의 관심 있음, 일반적으로 처음 관측한 데이터 분포의 영역 안에 새로운 관측치가 들어간다면 동일한 집단이라고 볼 수 있음 ([Scikit-Learn](#))
(semi-supervised anomaly)

Outlier Detection

Out-of-distribution Detection

▲ Define

Anomaly Detection

Novelty Detection

Outlier Detection : 훈련 데이터 내에서 다른 관측치와 멀리 떨어진 관측치로 정의되는 이상치가 포함됩니다. 따라서 이상치 탐지 추정은 비정상적인 관찰을 무시하고 훈련 데이터가 가장 집중된 영역을 맞추려고 합니다. ([Scikit-Learn](#))
(unsupervised anomaly detection)

"이상치는 다른 메커니즘에 의해 생성되었다는 의심을 불러 일으키기 위해 다른 관찰과 너무 많이 벗어난 관찰입니다." [Hawkins 1980]:

Out-of-distribution Detection

▲ Define

Anomaly Detection

Novelty Detection

Outlier Detection

Out-of-distribution Detection : 데이터는 학습 데이터의 분포와는 다른 분포를 갖는 데이터를 의미한다. 분류 문제에서 out of distribution은 학습 데이터에 포함되지 않은 class를 가진 데이터를 의미한다. 예를 들어 CIFAR-10을 분류하는 모델 입장에서 SVHN 데이터는 out of distribution이라고 할 수 있다. Out of Distribution 데이터를 탐지하는 것은 딥러닝 모델의 안정성에 있어서 굉장히 중요한 문제이다. ([link](#))

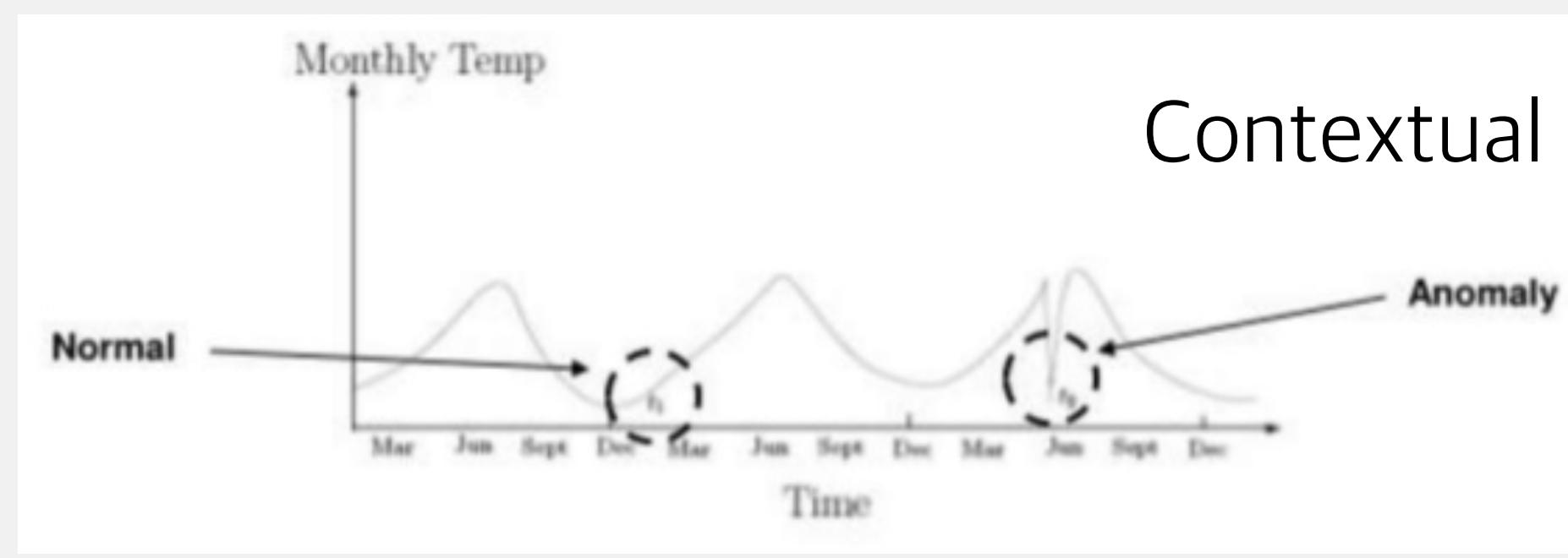
▲ 이상치의 종류

Anomaly detection

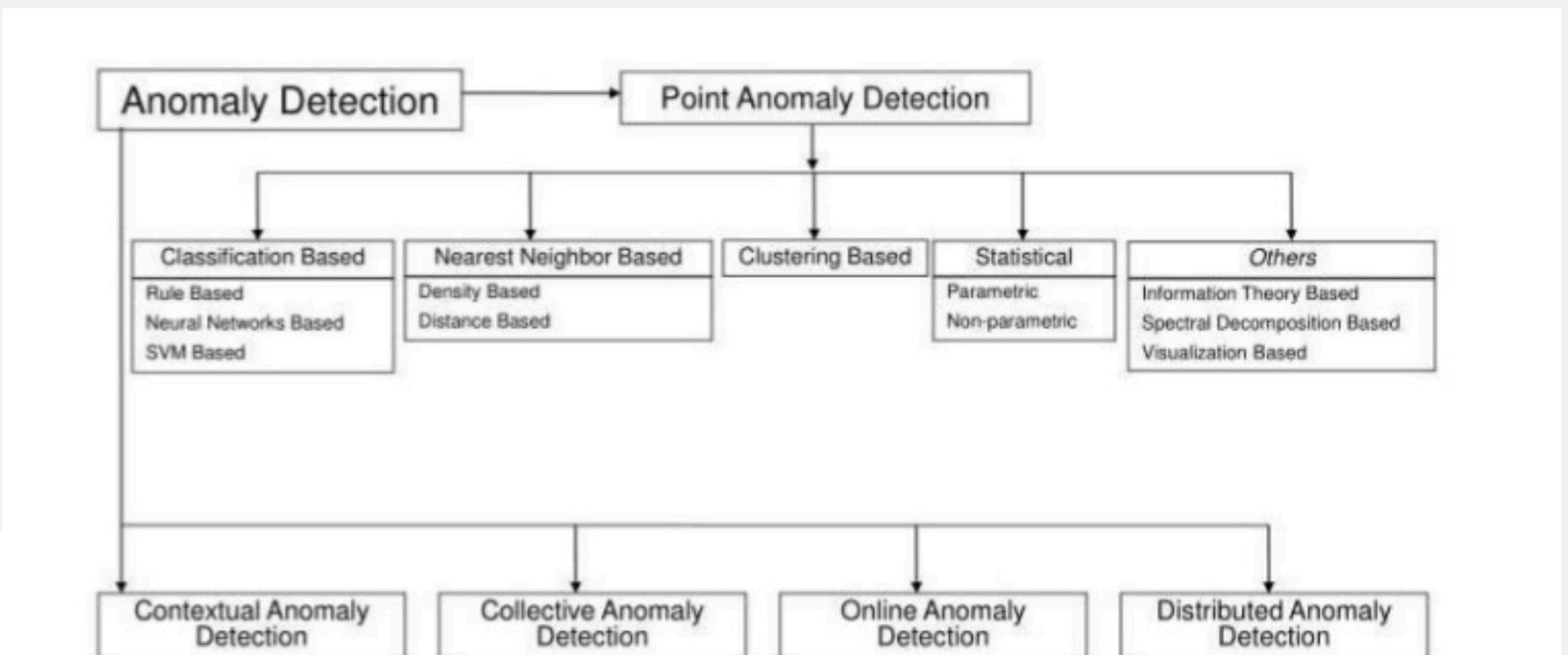
Point Anomalies : 데이터 셋의 뭉치에서 벗어난 값

Contextual Anomalies : 컨텍스트에 동떨어진 값, 컨텍스트의 정의가 필요

Collective Anomalies : 데이터 수집 문제로 발생한 이상값



Contextual



* Anomaly Detection – A Survey, Varun Chandola, Arindam Banerjee, and Vipin Kumar, To Appear in ACM Computing Surveys 2008.

▲ 이상치의 Task

1. 데이터에 대한 선형적 지식 없이 Outlier를 판별 (데이터가 정적이라는 판단)
현재 상태의 확률이 직전 상태의 확률과 같아지게 되는 평형 상태에 도달한 확률 분포
2. Label 데이터 필요한 경우
(지도 학습(Supervised classification)과 유사하며 Pre-Labelled Data 를 필요로 함)
3. Normality 만 모델링하거나 Abnormality를 극히 조금의 Case 에서 모델링
Pre-classified 된 Data 를 필요로 하지만 Normal이라고 마크된 데이터만 학습



사례

- Fraud detection
- 분실 카드
- 결제 이상탐지
- 계정 이상탐지
- 가격 이상탐지
- Detecting measurement errors
- “One person’s noise could be another person’s signal.”

▲ Method

- **Statistical Tests**

⇒ Given a certain kind of statistical distribution (e.g., Gaussian)

⇒ Compute the parameters assuming all data points have been generated by such a statistical distribution (e.g., mean and standard deviation)

- Depth-based Approaches
- Deviation-based Approaches
- Distance based Approaches
- Density-based Approaches
- High-dimensional Approaches

• <https://archive.siam.org/meetings/sdm10/tutorial3.pdf>

▲ Method

- Statistical Tests
- 아래의 자료를 요약
- <https://statsandr.com/blog/outliers-detection-in-r/>

▲ Method

- Statistical Tests

```
dat <- ggplot2::mpg  
summary(dat$hwy)
```

```
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.  
##    12.00   18.00  24.00  23.44  27.00  44.00
```

```
min(dat$hwy)
```

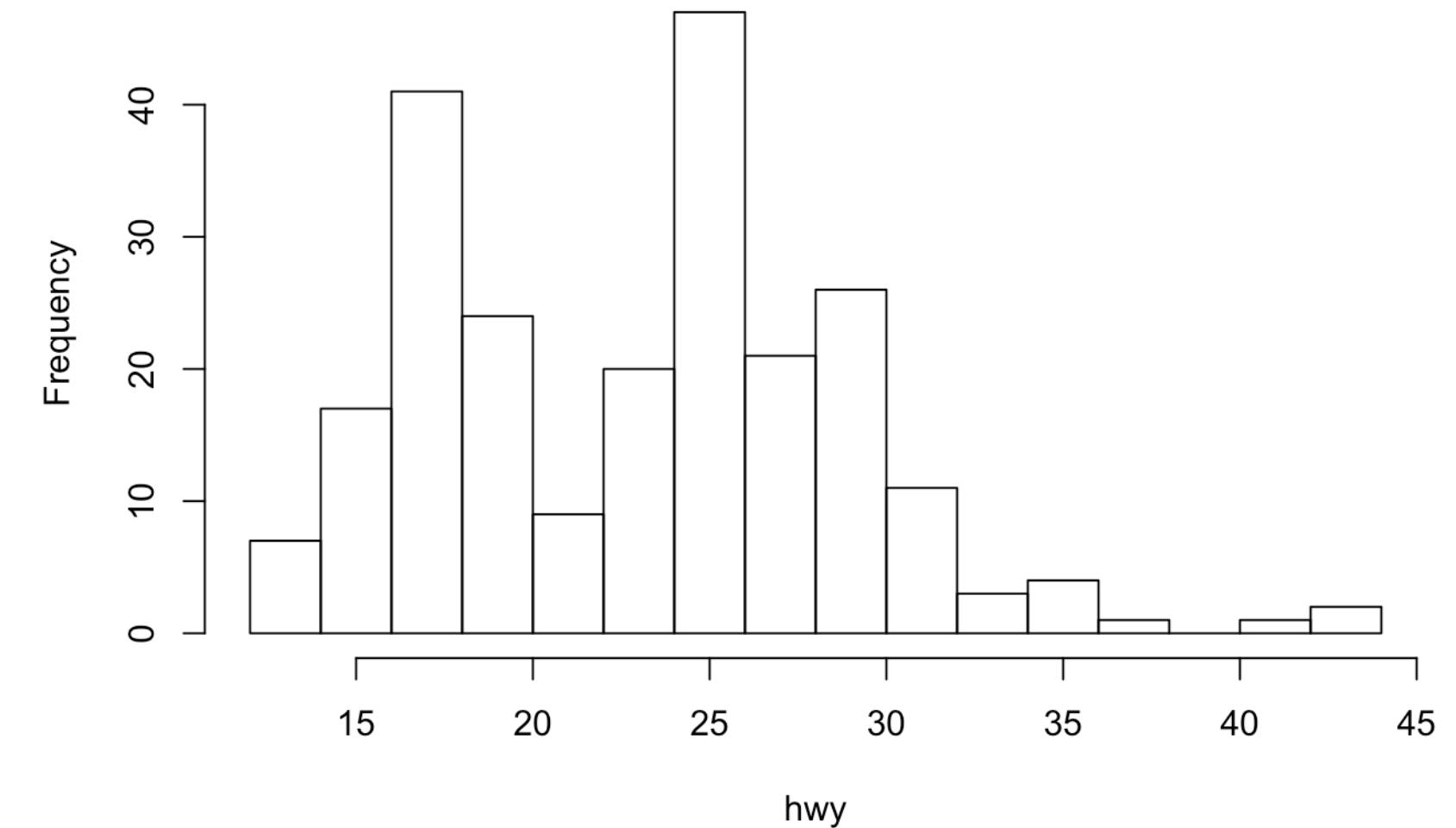
```
## [1] 12
```

```
max(dat$hwy)
```

```
## [1] 44
```

```
hist(dat$hwy,  
     xlab = "hwy",  
     main = "Histogram of hwy",  
     breaks = sqrt(nrow(dat))  
) # set number of bins
```

Histogram of hwy

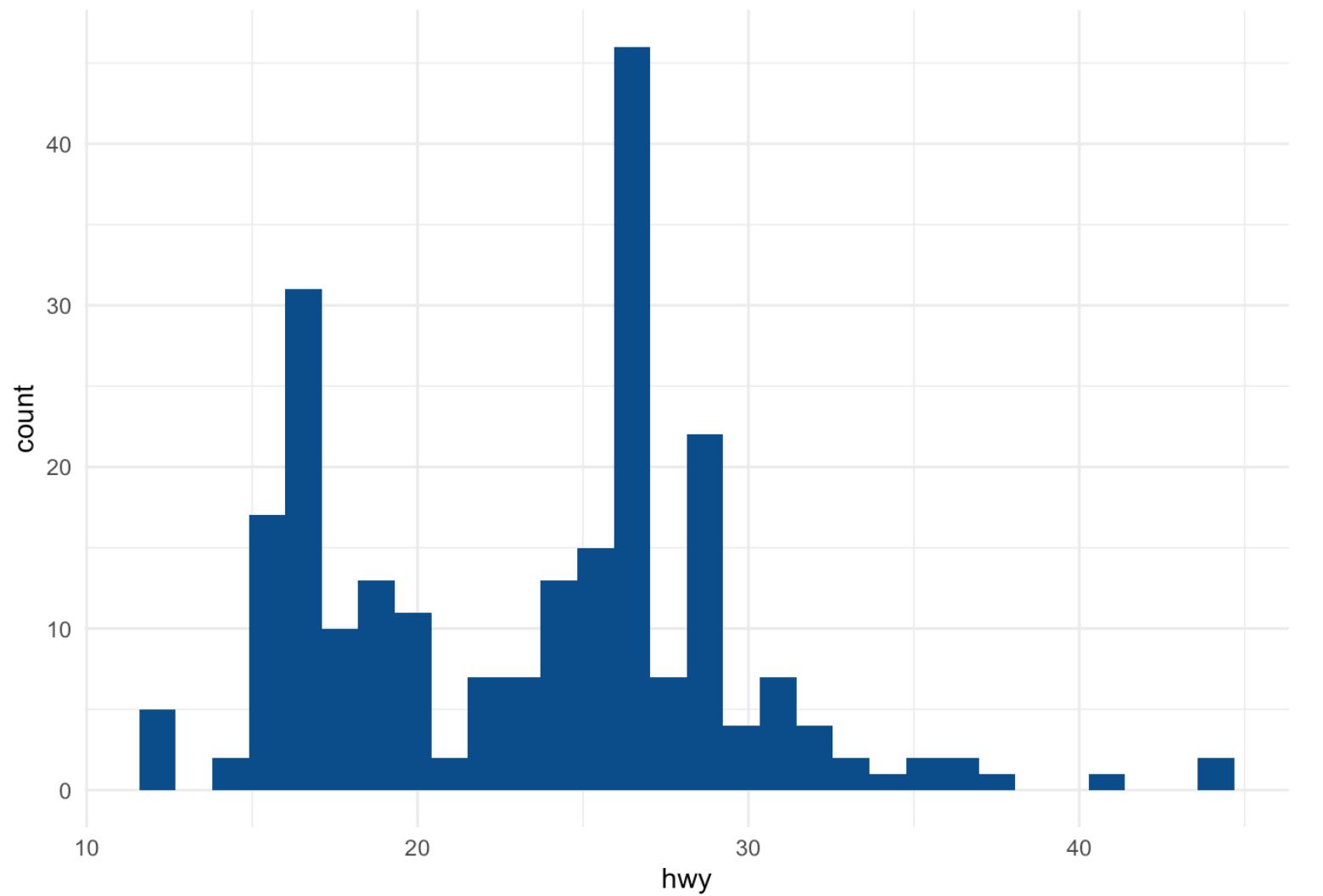


▲ Method

- Statistical Tests

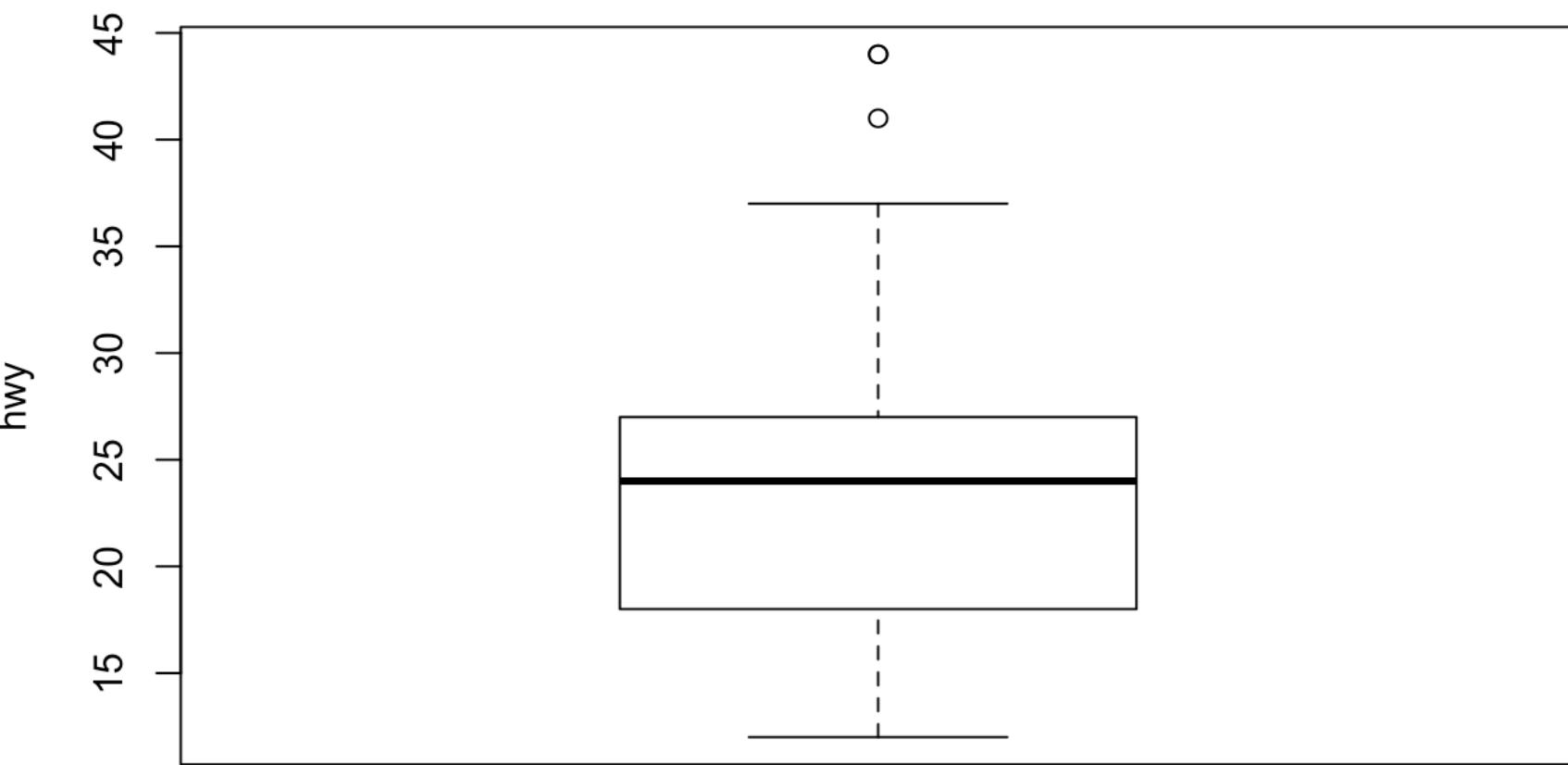
```
library(ggplot2)

ggplot(dat) +
  aes(x = hwy) +
  geom_histogram(bins = 30L, fill = "#0c4c8a") +
  theme_minimal()
```



Boxplot

```
boxplot(dat$hwy,
        ylab = "hwy"
      )
```



▲ Method

• Statistical Tests

boxplot.stats () \$out 함수 덕분에 IQR 기준에 따라 잠재적 인 이상 값의 값을 추출 할 수도 있습니다.

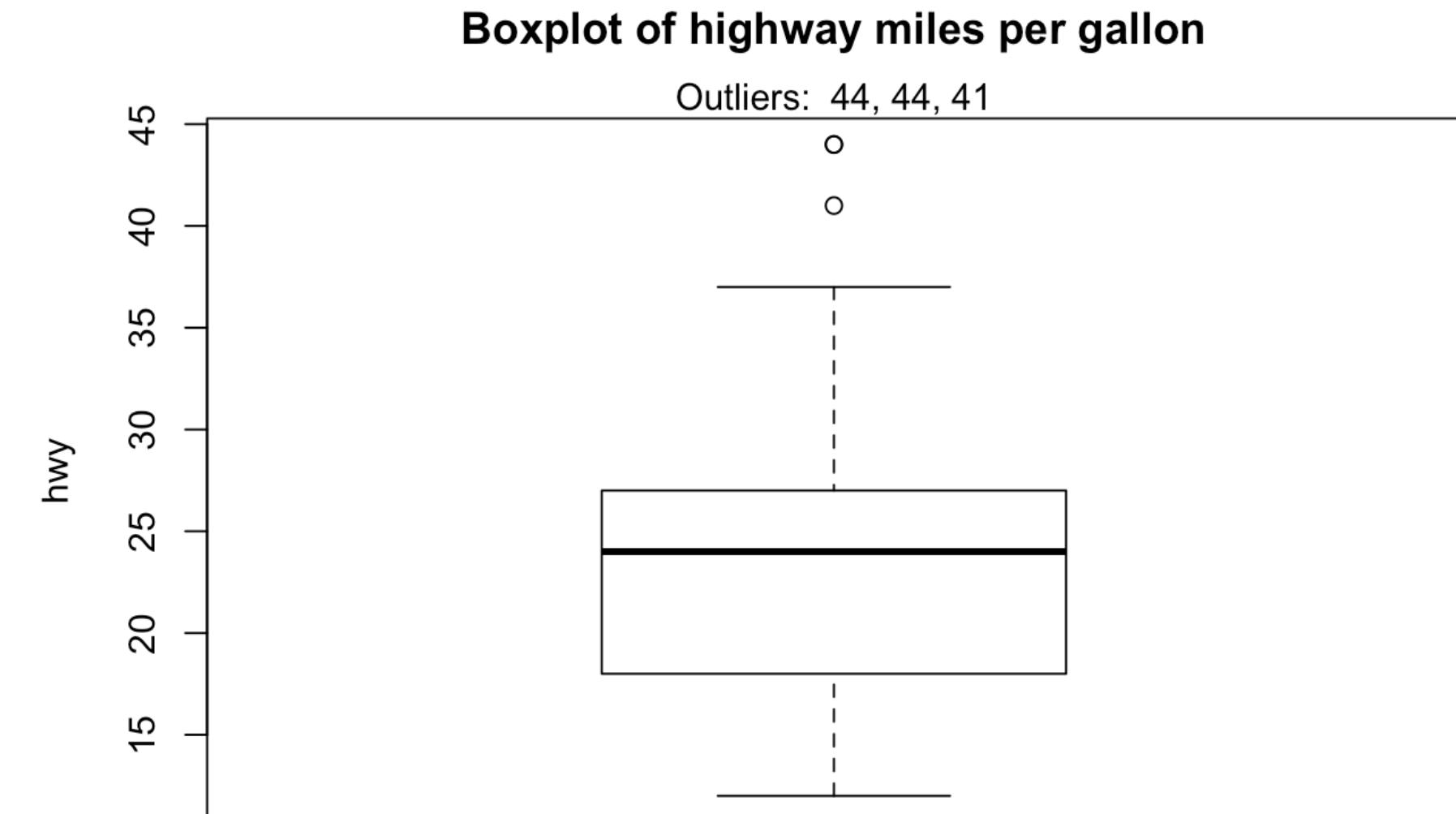
```
boxplot.stats(dat$hwy)$out  
  
## [1] 44 44 41
```

보시다시피, 실제로 잠재적 이상치로 간주되는 3 개의 점이 있습니다. 값이 44 인 관측치 2 개와 값 41 인 관측치 1 개입니다.

which () 함수 덕분에 다음 특이 치에 해당하는 행 번호를 추출 할 수 있습니다.

```
out <- boxplot.stats(dat$hwy)$out  
out_ind <- which(dat$hwy %in% c(out))  
out_ind  
  
## [1] 213 222 223  
  
dat[out_ind, ]  
  
## # A tibble: 3 x 11  
##   manufacturer model  displ  year cyl trans drv  cty  hwy fl class  
##   <chr>        <chr>  <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>  
## 1 volkswagen  jetta     1.9  1999     4 manual... f      33    44 d compact  
## 2 volkswagen  new be...  1.9  1999     4 manual... f      35    44 d subcom...  
## 3 volkswagen  new be...  1.9  1999     4 auto(l... f      29    41 d subcom...
```

```
boxplot(dat$hwy,  
       ylab = "hwy",  
       main = "Boxplot of highway miles per gallon"  
)  
mtext(paste("Outliers: ", paste(out, collapse = ", ")))
```



▲ Method

• Statistical Tests

Percentiles

- 이상값 탐지 방법은 백분위 수를 기반으로합니다. 백분위 수 방법을 사용하면 2.5 및 97.5 백분위 수에 의해 형성된 구간 밖에있는 모든 관측치가 잠재적 이상 값으로 간주됩니다. 1과 99 또는 5와 95 백분위 수와 같은 다른 백분위 수도 간격을 구성하는데 고려 될 수 있습니다.
- 하한 및 상한 백분위 수 값 (따라서 간격의 하한 및 상한)은 quantile () 함수로 계산할 수 있습니다.

```
lower_bound <- quantile(dat$hwy, 0.025)
lower_bound
```

```
## 2.5%
## 14
```

```
upper_bound <- quantile(dat$hwy, 0.975)
upper_bound
```

```
## 97.5%
## 35.175
```

- 이 방법에 따르면 14 미만 및 35.175 초과의 모든 관측치는 잠재적 인 이상 값으로 간주됩니다. 간격을 벗어난 관측치의 행 번호는 which () 함수를 사용하여 추출 할 수 있습니다.

```
outlier_ind <- which(dat$hwy < lower_bound | dat$hwy > upper_bound)
outlier_ind
```

```
## [1] 55 60 66 70 106 107 127 197 213 222 223
```

▲ Method

- Statistical Tests

```
lower_bound <- quantile(dat$hwy, 0.01)
upper_bound <- quantile(dat$hwy, 0.99)

outlier_ind <- which(dat$hwy < lower_bound | dat$hwy > upper_bound)

dat[outlier_ind, ]
```

```
## # A tibble: 3 x 11
##   manufacturer model    displ  year   cyl trans   drv     cty   hwy fl class
##   <chr>        <chr>    <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
## 1 volkswagen   jetta     1.9   1999     4 manual... f       33    44 d   compact
## 2 volkswagen   new be...  1.9   1999     4 manual... f       35    44 d   subcom...
## 3 volkswagen   new be...  1.9   1999     4 auto(l... f       29    41 d   subcom...
```

- 백분위 수를 1과 99로 설정하면 IQR 기준과 동일한 잠재적 이상 값이 제공됩니다.

▲ Method

• Statistical Tests

Hampel filter

- 햄펠 필터라고하는 또 다른 방법은 중앙값에 의해 형성된 구간 (I) 외부의 값에 3 개의 중앙값 절대 편차 (MAD)를 더하거나 빼는 것을 이상 값으로 간주하는 것입니다.
- 절대 편차 중앙값이며 데이터 중앙값에서 절대 편차의 중앙값으로 정의됩니다.
- 이 메서드의 경우 먼저 median () 및 mad () 함수 덕분에 간격 제한을 설정합니다.

```
lower_bound <- median(dat$hwy) - 3 * mad(dat$hwy)
lower_bound
```

```
## [1] 1.761
```

```
upper_bound <- median(dat$hwy) + 3 * mad(dat$hwy)
upper_bound
```

```
## [1] 46.239
```

- 이 방법에 따르면 1.761 미만 및 46.239 이상의 모든 관측치는 잠재적 이상 값으로 간주됩니다. 간격을 벗어난 관측치의 행 번호는 which () 함수를 사용하여 추출 할 수 있습니다.

```
outlier_ind <- which(dat$hwy < lower_bound | dat$hwy > upper_bound)
outlier_ind
```

```
## integer(0)
```

- Hampel 필터에 따르면 hwy 변수에 대한 잠재적 인 이상 치는 없습니다.

▲ Method (사례)

- Hampel Filter
- 지표 분석에서 적용하기가 간단하고 편리함 (상대방을 이해시키기가 쉬움)

- 1) 게임 경제 인플레이션 분석



- 게임 로그 기반으로, 경제 시스템을 분석해서 사업/기획이 경제 시스템을 판단하는 데 사용

- 2) 이커머스 상품 구매건수 분석



- 구매 Action 을 기반으로, 특정 시기에 많이 구매가 발생하는 상품들을 기반으로 데이터 분석

▲ Method (사례)

-

게임 경제 인플레이션 분석

- 인플레이션? 통화량이 팽창하여 화폐 가치가 폭락하며 물가가 계속적으로 등극하여 일반 대중의 실질적 소득이 감소되는 현상 (반대 현상 : 디플레이션)
- 화폐 생산과 화폐 소비의 균형이 맞아야 게임 상의 물가가 유지되는데, 화폐 생산이 소비보다 많아지게 되면 인플레이션 발생



재화의 종류를 늘려 각각의 소비처를 분산한 것은,
게임성에도 영향을 미친다는 의견이 있었다.

▲ Method (사례)

• Hampel Filter

[인플레이션 지표 프로세스]

- 1) 게임 DB 데이터를 이용해 일별 자원 보유량(통계량) 계산
- 2) 재화 보유량이 계산해둔 상/하한선을 연속적으로 넘을 경우 **Alerm**
- 3) 2번 규칙 벗어나더라도 보유량이 연속적인 RUN(+, -) 발생할 경우 **Alerm**
(들어오는 측정 값이 이전 14일 중앙값 보다 증가, 감소)
- 4) 증감 원인 판단하기 위해 획득/소진 패턴 확인
- 5) 담당 사업부에 메신저 또는 메일로 전달

[참고사항]

세모 : 이전 14날의 중앙값보다 증가,
동그라미 : 이전 14날의 중앙값보다 감소
(+-----+-----++ : 런 개념)

상/하한선 결정은 시계열 통계량 및 파라미터 필요 (n : 관찰기간, level : 수준)

runMedian(returns medians over a n-period moving window)

runMAD (runMAD: returns median/mean absolute deviations over a n-period moving window)



▲ Method (사례)

- Hampel Filter
- 이커머스 상품 구매건수 분석 (사례 자료는 비공개)

▲ Method

- Statistical Tests
- 아래는 정규분포를 따르는 경우 사용하는 방법들
- Grubbs's test : 정규분포를 따르는 데이터에서 하나의 이상치를 발견할 수 있는 검정 방법
- Dixon's test
- Rosner's test

- 가설 -

Grubbs's Test가 검정하고자 하는 가설은 다음과 같다.

H_0 : 데이터에 이상치가 하나도 없다.

H_a : 이상치가 하나 있다.

- 검정 방법 -

y_1, y_2, \dots, y_n 을 현재 관측 데이터라고 할 때 Grubbs's Test의 검정 통계량은 다음과 같다.

$$G = \frac{\max_{i=1,\dots,n} |y_i - \bar{y}|}{s}$$

여기서 \bar{y} 는 표본평균, s 는 표본 표준편차이다.

여러 개의 이상치가 있다면
검정통계량이 작아져서, 귀무 가설을
기각하지 못하게 됨

▲ Method

- Statistical Tests

- Grubbs's test :
정규분포를 따르는
데이터에서 하나의
이상치를 발견할 수 있는
검정 방법

R에서 Grubbs 테스트를 수행하려면 {outliers} 패키지의 grubbs.test () 함수를 사용합니다.

```
#install.packages("outliers")
```

```
library(outliers)
test <- grubbs.test(dat$hwy)
test
```

```
##
##  Grubbs test for one outlier
##
## data: dat$hwy
## G = 3.45274, U = 0.94862, p-value = 0.05555
## alternative hypothesis: highest value 44 is an outlier
```

- p- 값은 0.056입니다. 5 % 유의 수준에서 가장 높은 값 44가 특이 치가 아니라는 가설을 기각하지 않습니다.
- 기본적으로 테스트는 가장 높은 값에 대해 수행됩니다 (R 출력 : 대립 가설 : 가장 높은 값 44는 이상치에 표시됨). 가장 낮은 값에 대한 테스트를 수행하려면 grubbs.test () 함수에 반대 = TRUE 인수를 추가하면 됩니다.

```
test <- grubbs.test(dat$hwy, opposite = TRUE)
test
```

```
##
##  Grubbs test for one outlier
##
## data: dat$hwy
## G = 1.92122, U = 0.98409, p-value = 1
## alternative hypothesis: lowest value 12 is an outlier
```

- R 출력은 이제 테스트가 가장 낮은 값에 대해 수행됨을 나타냅니다 (대립 가설 참조 : 가장 낮은 값 12는 특이 치임).
- p- 값은 1입니다. 5 % 유의 수준에서 가장 낮은 값 12가 특이 치가 아니라는 가설을 기각하지 않습니다.

▲ Method

- Statistical Tests
- Dixon's test : 정규분포로부터 추출된 데이터의 이상치를 순서 통계량을 이용하는 검정하는 방법

- 가설 -

Dixon's Q-Test는 다음의 가설을 검정한다.

H_0 : 데이터에 이상치가 하나도 없다.

H_a : 데이터에 이상치가 1개 있다.

1 단계) 데이터를 오름차순으로 정렬한다. 정렬된 데이터를 $y_{(i)}, i = 1, \dots, n$ 라 하자. 즉,

$$y_{(1)} \leq y_{(2)} \leq \dots \leq y_{(n-1)} \leq y_{(n)}$$

이다.

2 단계) 검정 통계량을 구한다.

$$r = \frac{y_{(2)} - y_{(1)}}{y_{(n)} - y_{(1)}} \text{ or } \frac{y_{(n)} - y_{(n-1)}}{y_{(n)} - y_{(1)}}$$

3 단계) 유의 수준 α 와 데이터 개수 n 에 대하여 단측 검정이라면 기각값 $c_{\alpha,n}$ 을 구한다. 기각값은 테이블을 이용하여 찾을 수 있다(난 기각값을 계산하는 과정을 구현하였다).

4 단계) 검정 통계량 r 과 $c_{\alpha,n}$ 을 비교한다. $r > c_{\alpha,n}$ 이면 귀무가설 H_0 를 기각한다. 즉, $y_{(1)}, y_{(n)}$ 을 이상치라고 판단한다.

▲ Method

- Statistical Tests

- Dixon's test :

정규분포로부터 추출된 데이터의 이상치를 순서 통계량을 이용하는 검정하는 방법

- 작은 표본 크기
($n \leq 25$)에 유용된 검정

```
subdat <- dat[1:20, ]
test <- dixon.test(subdat$hwy)
test
```

```
##
## Dixon test for outliers
##
## data: subdat$hwy
## Q = 0.57143, p-value = 0.006508
## alternative hypothesis: lowest value 15 is an outlier
```

- 결과는 가장 낮은 값 15가 특이 치라는 것을 보여줍니다 (p- 값 = 0.007).
- 가장 높은 값을 테스트하려면 dixon.test () 함수에 반대 = TRUE 인수를 추가하면됩니다.

```
test <- dixon.test(subdat$hwy,
  opposite = TRUE
)
test
```

```
##
## Dixon test for outliers
##
## data: subdat$hwy
## Q = 0.25, p-value = 0.8582
## alternative hypothesis: highest value 31 is an outlier
```

- 결과는 가장 높은 값 31이 특이 치가 아님을 보여줍니다 (p- 값 = 0.858).
- 상자 그림과 비교하여 특이 치에 대한 통계 테스트 결과를 항상 확인하여 모든 잠재적 특이 치를 테스트했는지 확인하는 것이 좋습니다.

▲ Method

- Statistical Tests

- Rosner's test : 표본 크기가 20 이상일 경우 이상치가 N개(default, n=3)까지 예상될 때 이용하고 Grubb's 및 Dixon 테스트와 달리 한 번에 여러 특이치를 감지하는 데 사용

```
#install.packages('EnvStats')
library(EnvStats)
```

```
##
## Attaching package: 'EnvStats'
```

```
## The following objects are masked from 'package:stats':
## predict, predict.lm
```

```
## The following object is masked from 'package:base':
## print.default
```

```
test <- rosnerTest(dat$hwy,
  k = 3
)
test
```

```
##
## Results of Outlier Test
## -----
##
## Test Method: Rosner's Test for Outliers
##
## Hypothesized Distribution: Normal
##
## Data: dat$hwy
##
## Sample Size: 234
##
## Test Statistics:
##   R.1 = 13.722399
##   R.2 = 3.459098
##   R.3 = 3.559936
##
## Test Statistic Parameter: k = 3
##
## Alternative Hypothesis: Up to 3 observations are not
##                           from the same Distribution.
##
## Type I Error: 5%
##
## Number of Outliers Detected: 1
##
##   i  Mean.i    SD.i Value Obs.Num   R.i+1 lambda.i+1 Outlier
## 1 0 24.21795 13.684345  212     34 13.722399  3.652091  TRUE
## 2 1 23.41202  5.951835   44     213  3.459098  3.650836 FALSE
## 3 2 23.32328  5.808172   44     222  3.559936  3.649575 FALSE
```

```
test$all.stats
```

```
##   i  Mean.i    SD.i Value Obs.Num   R.i+1 lambda.i+1 Outlier
## 1 0 24.21795 13.684345  212     34 13.722399  3.652091  TRUE
## 2 1 23.41202  5.951835   44     213  3.459098  3.650836 FALSE
## 3 2 23.32328  5.808172   44     222  3.559936  3.649575 FALSE
```

▲ Method

- Model-based 기법
- Prophet (페이스북에서 제공하는 시계열 예측 라이브러리)
- 시계열 모형의 특징
 - $Y(t) = g(t)$ (성장) + $s(t)$ (계절성) + $h(t)$ (휴일/이벤트) + ε_t
(성장은 선형, 로지스틱, 계절성은 푸리에 급수를 이용해 근사치 조정, 휴일/이벤트는 특정 기간 예측조정하면서 Curve Fitting, Prophet 모형은 MAP/MCMC 기반으로 학습 가능)
 - 딱히 이해하지 않아도 직관적으로 컨트롤하는데 장점이 있기에 많은 데이터 분석가 사용
 - MCMC는 초기 시계열 데이터가 부족한 상황에서 시뮬레이션으로 Fitting 결과 제공

▲ Method (사례)

- Model-based 기법
- Prophet 을 이용한 이상치 탐지 사례
- 시계열 지표 분석 (시계열 데이터 갯수가 충분히 많은 경우)
- Prophet 사례 자료는 비공개

▲ Method

- Model-based 기법

```
1 auudata_all <- auudata_all %>% as.tibble() %>% mutate(  
2   gameID = gameID,  
3   ds = ymd(YYMMDD),  
4   y = auuScore) %>%  
5   select(gameID, ds, gamename, y)  
6  
7 auudata_d1 <- auudata_all_1 %>%  
8   nest(-gamename) %>%  
9   mutate(m = map(data, prophet)) %>%  
10  mutate(future = map(m, make_future_dataframe, period=12, freq='week')) %>%  
11  mutate(forecast = map2(m, future, predict))  
12  
13 auudata_d2 <- auudata_all_2 %>%  
14   nest(-gamename) %>%  
15   mutate(m = map(data, prophet, changepoint.prior.scale = 0.1, mcmc.samples=1000)) %>%  
16   mutate(future = map(m, make_future_dataframe, period=12, freq='week')) %>%  
17   mutate(forecast = map2(m, future, predict))
```

ds, y 형태로 formatting, 위에 있는 코드는 여러 개를 한꺼번에 계산할 때 map 함수를 이용

▲ Research

• 네이버 스마트스토어 가격 조작을 통한 어뷰징 대응 안내

■ 대상

- 상품 등록/수정시 비정상적으로 판매가를 변경하거나 과도한 할인을 적용하여 허위 거래를 조작하는 Case

■ 대응 방안

- 1) 과도한 가격 변동 내역 및 구매 정보를 분석해 비정상 거래를 탐지함
- 2) 대상 거래 건의 경우 구매가 발생하더라도 검색 랭킹에 반영되는 판매자수로 인정하지 않음

■ 반영일

- 10월 중 적용 예정 (신규 및 기존 등록된 상품도 해당됨)

Q. 비정상 거래란 무엇인가요?

A. 비정상 거래란, 검색 랭킹 상승을 위해 과도하게 할인된 가격으로 판매가 또는 할인가를 수정한 후 허위 거래를 생성하는 경우를 말합니다.

예를 들어 판매가 10,000원의 상품을 판매가 또는 할인가를 수정해 10원으로 변경한 후, 대량 구매하여 랭킹 및 구매평을 조작하는 케이스가 해당합니다.

정상적으로 판매 활동을 진행하는 일반 판매자분들은 대부분 제재 대상에 해당되지 않습니다.

▲ Research

- Anomaly Detection for an E-commerce Pricing System

Feature Set	Notation	# Features	Example
Raw Price	\mathcal{P}	17	Price, Cost
Baseline Price	\mathcal{A}	6	Price, Cost
Baseline Log Price	\mathcal{A}_L	5	$\log(\text{Price} / \text{Cost})$
Time Series	\mathcal{T}	2	AvgHistPrice
Transformed Price	\mathcal{P}_T	32	IMU
Log Transformed	\mathcal{P}_L	39	$\log(\text{Price} / \text{Cost})$
Hierarchy	\mathcal{H}	5	SubCategoryId
Binary	\mathcal{B}	9	IsPromo
Categorical	\mathcal{C}	3	PromotionType
Other Numerical	\mathcal{O}	3	Inventory

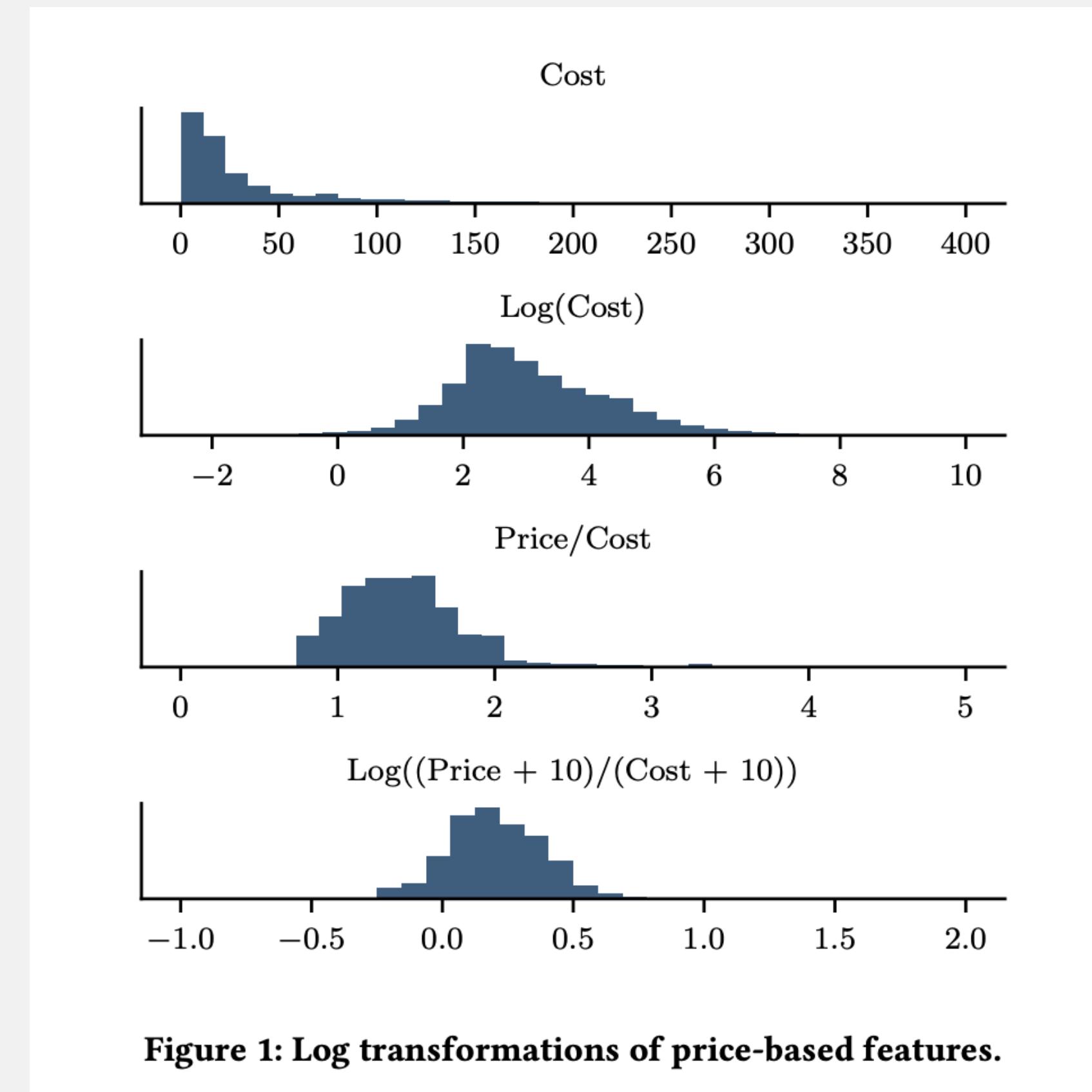


Figure 1: Log transformations of price-based features.

▲ Research

- Anomaly Detection for an E-commerce Pricing System
- Baseline : Gaussian Naive Bayes Baseline Model

Table 2: Models.

Approach	Type	# Features
GaussianNB	Unsupervised	5
Isolation Forest	Unsupervised	121
Autoencoder	Unsupervised	89
GBM	Supervised	121
RF	Supervised	121

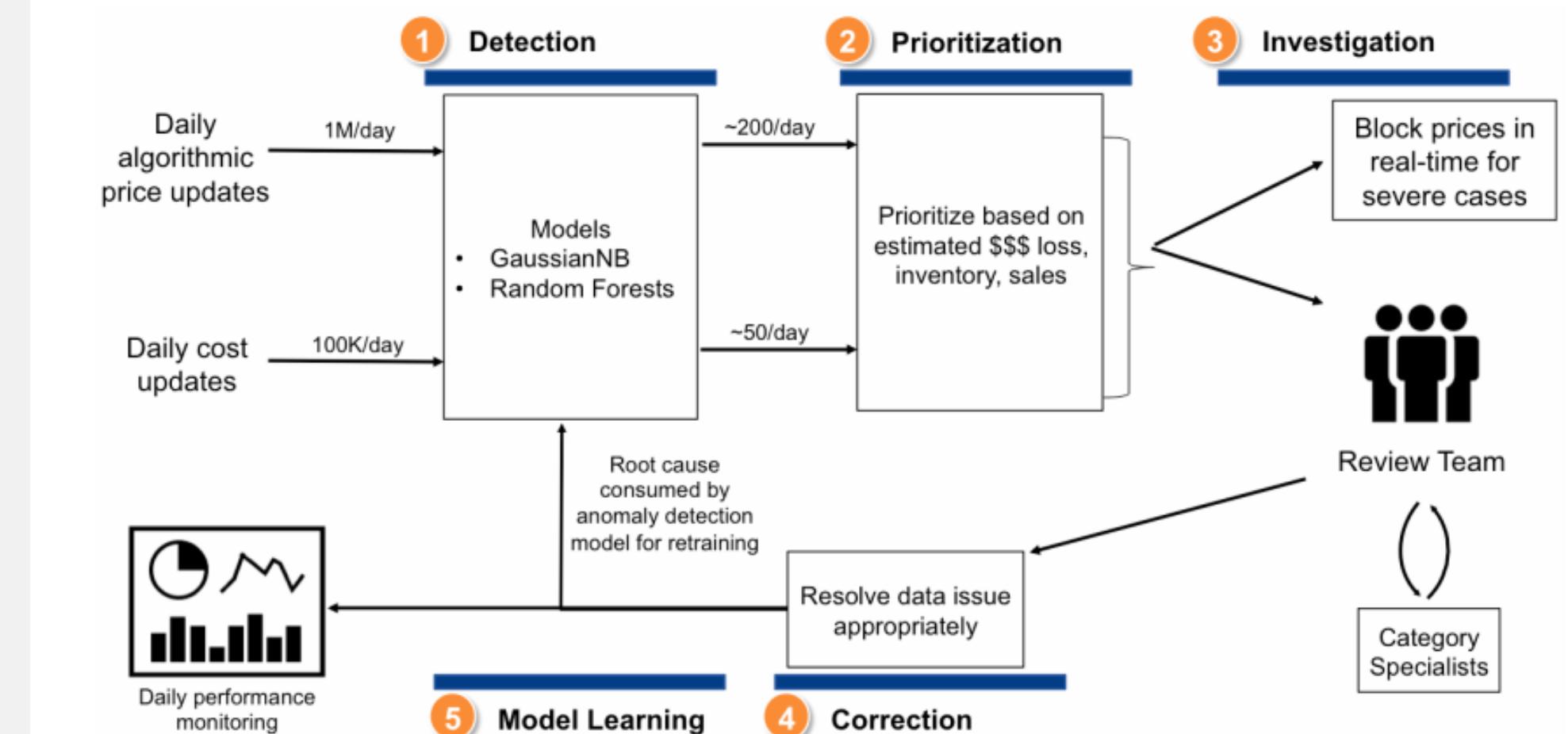
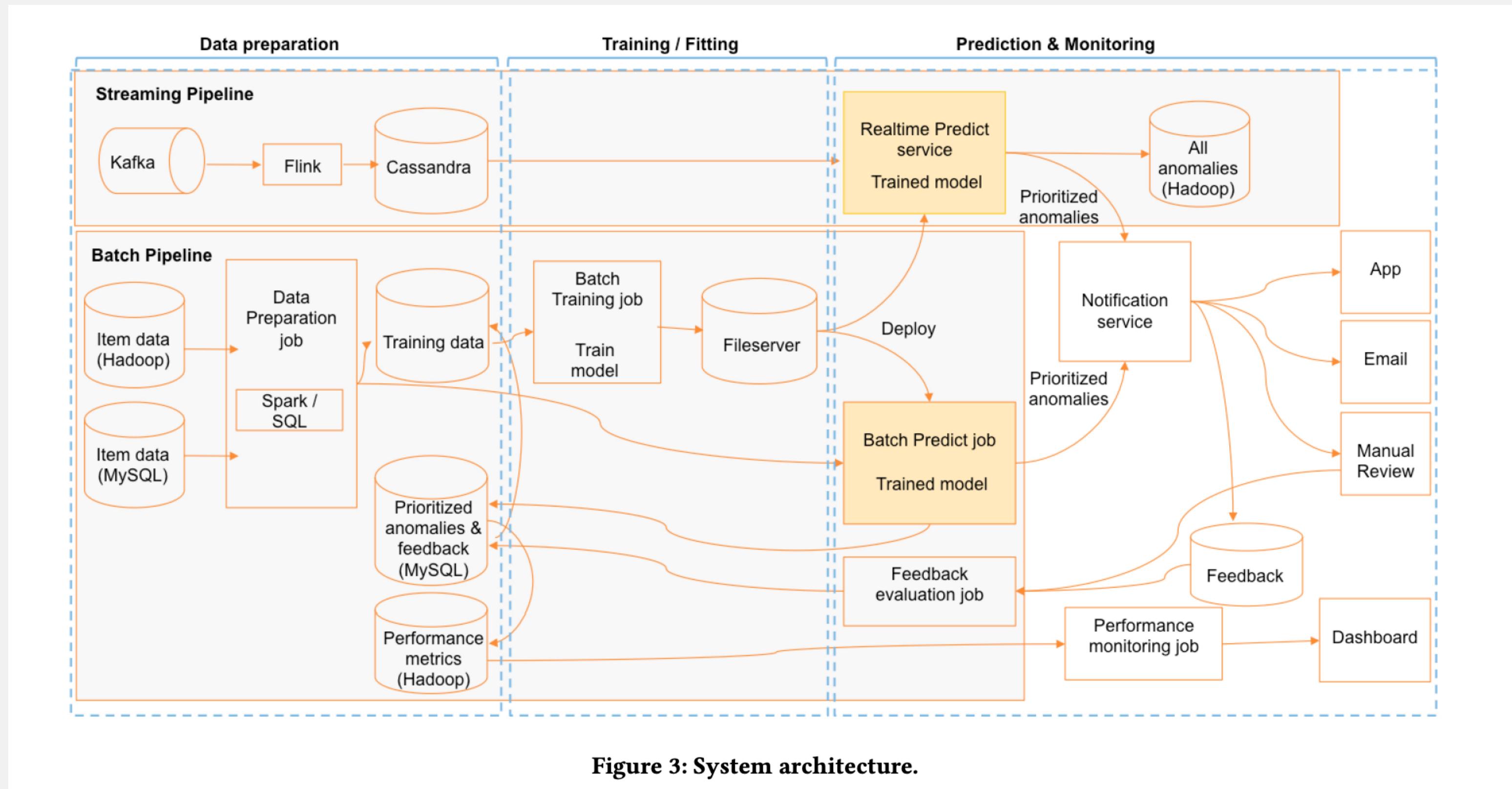


Figure 2: Overall system process.

▲ Research

- Anomaly Detection for an E-commerce Pricing System



▲ Research

- Anomaly Detection for an E-commerce Pricing System

Table 3: Dataset.

Class	Training Set	Test Set
Normal Instances	4,627,747	1,066,932
Anomaly Instances	1,069	1,068
Total	4,628,816	1,068,000

Normal 은 Catalog 정보 활용
Anomaly 는 운영팀

Table 4: Performance of GaussianNB models at different hierarchy levels. We use 5-fold cross validation with stratified splits. AUC refers to the area under the precision-recall curve.

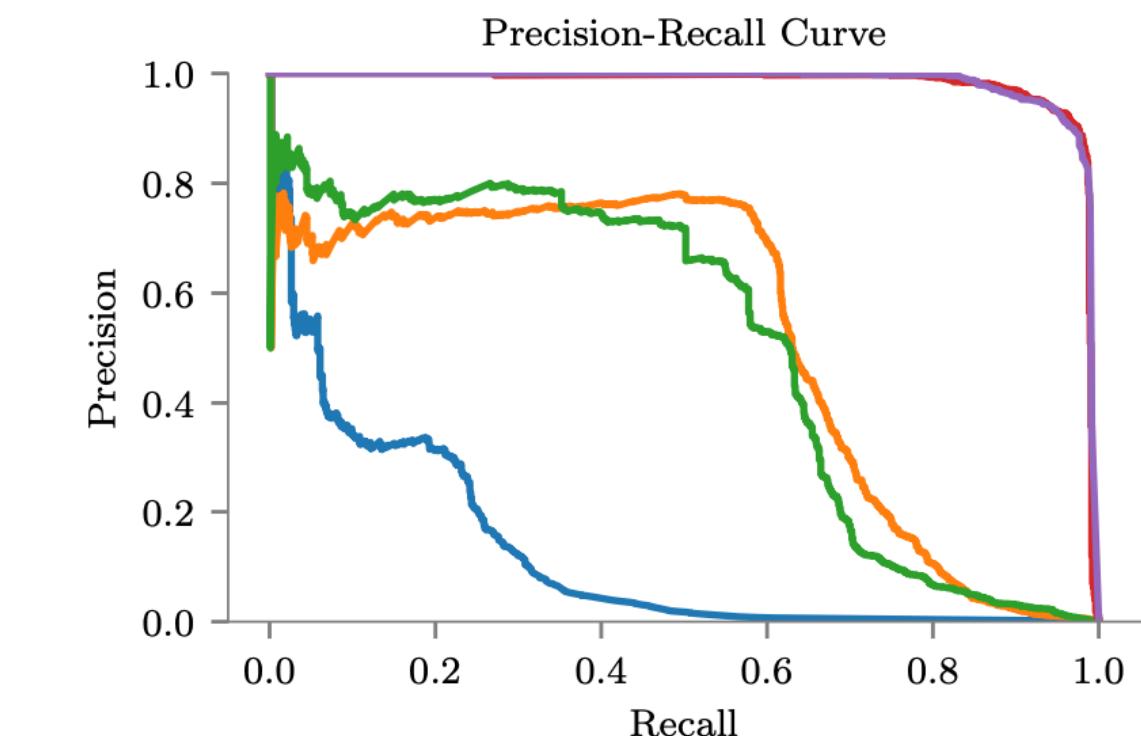
Hierarchy	# Models	Precision	Recall	F_1 Score	AUC
SubCat	2704	0.3343	0.2210	0.2661	0.1350
Cat	590	0.2827	0.2285	0.2527	0.1234
Dep	160	0.2894	0.2303	0.2565	0.1217
SuperDep	37	0.3051	0.2060	0.2459	0.1098
Div	7	0.2682	0.2247	0.2445	0.1000

▲ Research

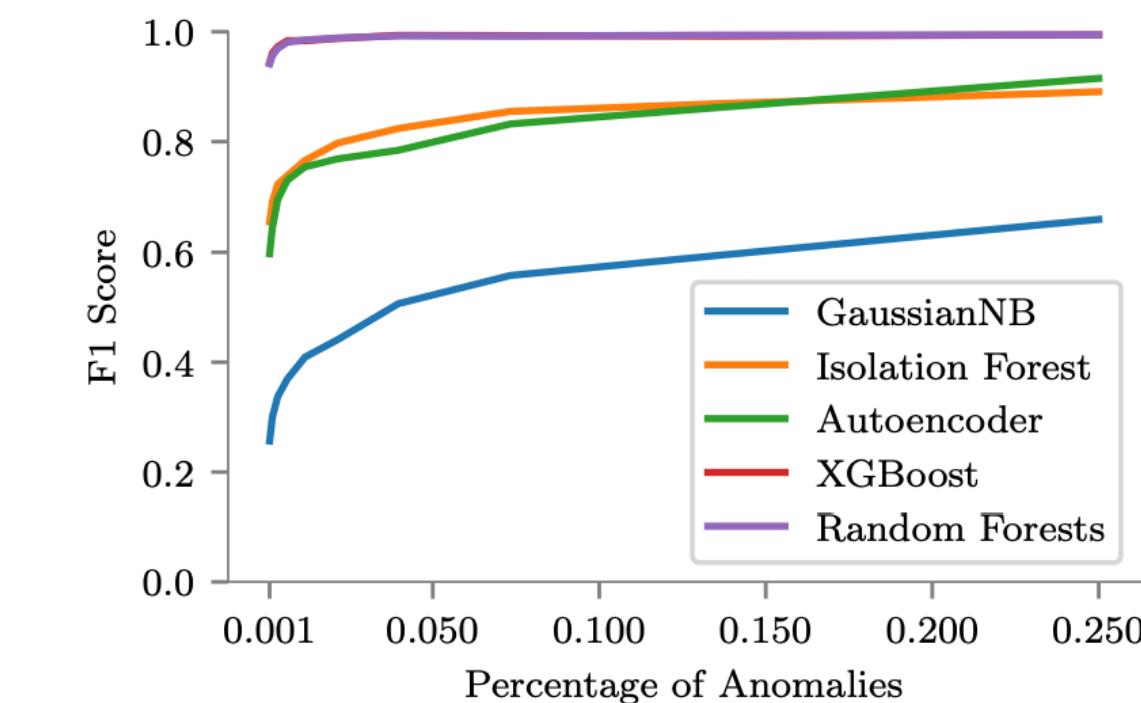
- Anomaly Detection for an E-commerce Pricing System

Table 5: Performance of various anomaly detection models.
AUC refers to the area under the precision-recall curve.

Approach	Precision	Recall	F_1 Score	AUC
GaussianNB	0.2894	0.2303	0.2565	0.1217
Isolation Forest	0.7555	0.5787	0.6554	0.5184
Autoencoder	0.6573	0.5478	0.5975	0.5008
GBM	0.9284	0.9597	0.9438	0.9810
RF	0.9402	0.9429	0.9416	0.9831



(a) Precision-Recall Curve



(b) Model Performance vs Percentage of Anomalies

▲ Research

- Anomaly Detection for an E-commerce Pricing System

Table 6: Training and prediction times of anomaly detection models. We randomly sampled 1000 items from the test set with 25% anomalies and reported the time from predicting them all at once (batch) and one-by-one (online). The prediction times are the average prediction time per item both for batch and online.

Approach	Train	Batch	Online
	time [s]	Prediction time [ms]	Prediction time [ms]
GaussianNB	451.487	0.021	0.091
Isolation Forest	396.229	0.078	29.902
Autoencoder	6853.187	0.026	0.934
GBM	3138.834	0.005	0.169
RF	3794.588	0.321	215.925

Table 7: Results from production launch. FP refers to the number of False Positives, i.e., number of predictions that were not actually anomalies.

	# Alerts	# Reviewed	# FP	Precision
Original	5,205	1,625	756	53.5%
Adjusted	5,205	1,625	386	76.2%

▲ Reference

[Outliers detection in R](#)

[Anomaly detection](#)

[Scikit-Learn](#)

<http://dsba.korea.ac.kr/seminar/?mod=document&uid=246>

[A survey of outlier detection](#)

<https://zephyrus1111.tistory.com/96>

[Anomaly Detection for an E-commerce Pricing System](#)