R 기초 데이터 분석

한국고용정보원
민종열 대리

한국고용정보원
Korea Employment Information Service

PROFILE

민종열 대리

통계학과 졸업

한국고용정보원 빅데이터분석TF팀 재직 중

빅데이터 기획 및 분석 업무 담당

원내 빅데이터 분석 프로젝트 진행

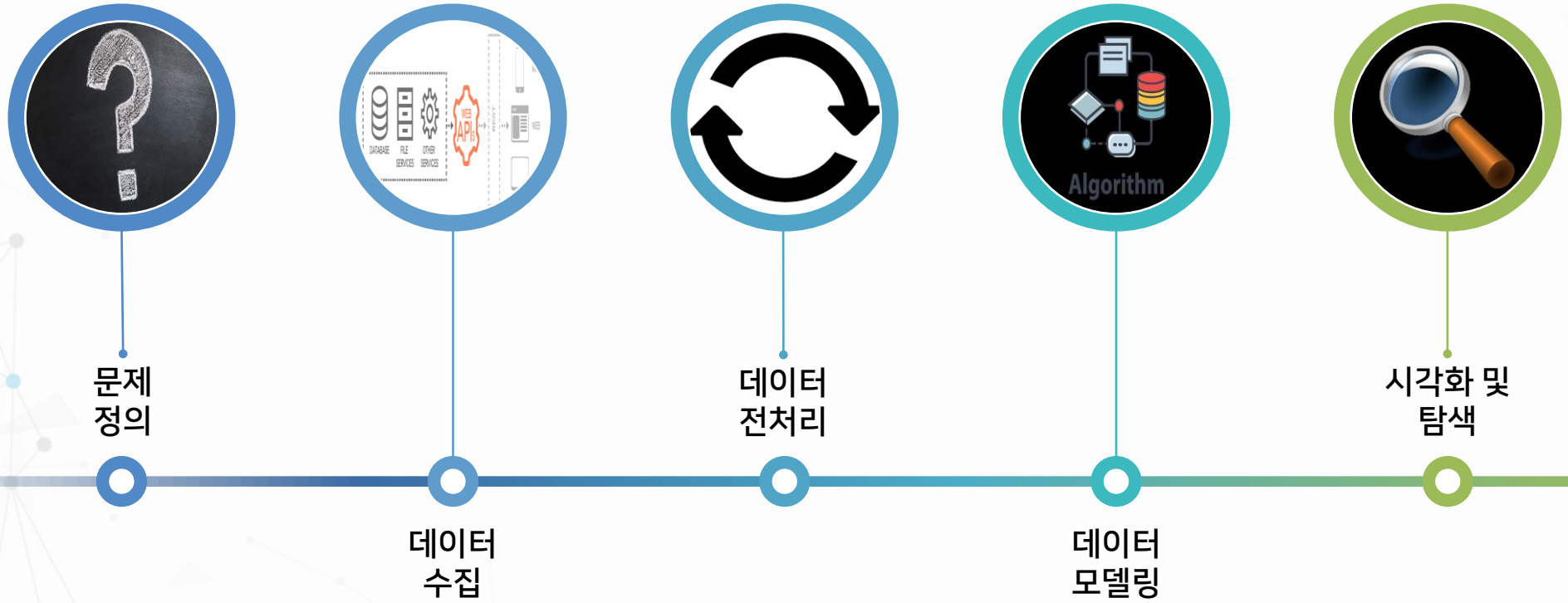
CONTENTS

Chapter

I

데이터 전처리

01 실제 빅데이터 분석 업무 절차



01 활용할 데이터

타이타닉 데이터

§ 타이타닉에 탑승한 승객들의 생존 여부에 관련된 데이터

§ 분석 관련 데이터 중 유명한 예제

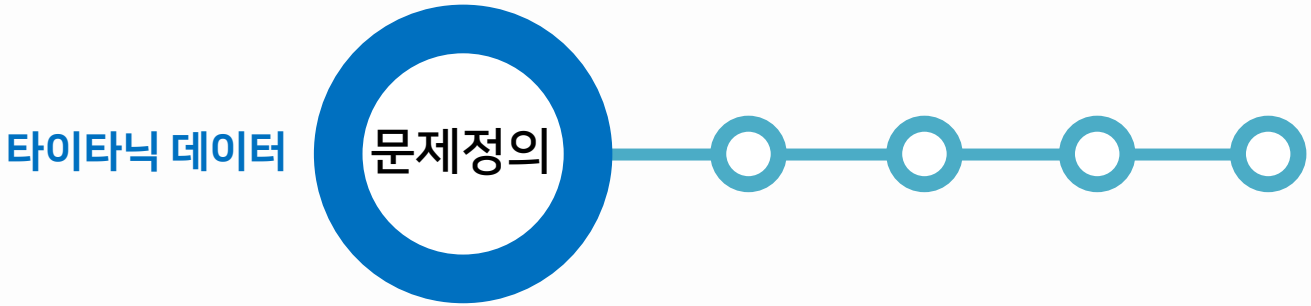
01 타이타닉 데이터

컬럼	컬럼명	데이터타입
PassengerId	승객번호	Int
Survived	생존여부	Int
Pclass	좌석등급	Int
Name	나이	Chr
Sex	성별	Chr
Age	나이	Num
Sibsp	동승한 형제/배우자수	Int
Parch	동승한 부모/자식수	Int
Ticket	티켓 번호	Chr
Fare	승객 요금	Num
Cabin	방 호수	Chr
Embarked	탑승지	Chr

타이타닉 데이터

- 데이터 다운로드
- 코드)
- Install.packages('titanic')
- library(titanic)
- titanic_train # 타이타닉 학습용 데이
- str(titanic_train) # 데이터 구조 확인
- summary(titanic_train) # 데이터 분포 확인

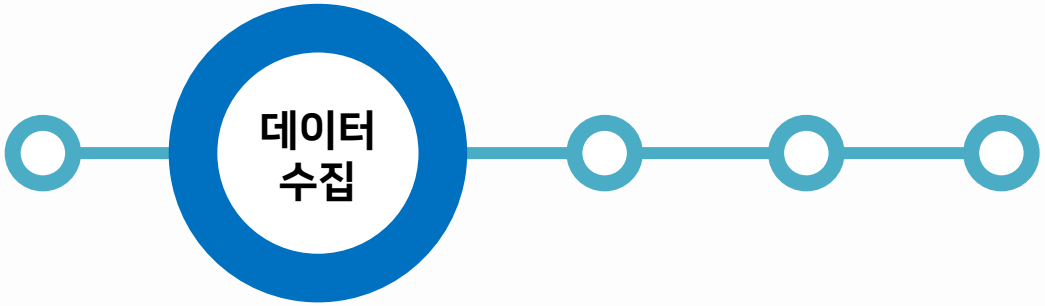
01 문제 정의



문제 정의	어떤 특성을 가지고 있는 사람들이 타이타닉 사고에서 생존율이 높을까?
목표 변수	Survived : 생존 여부

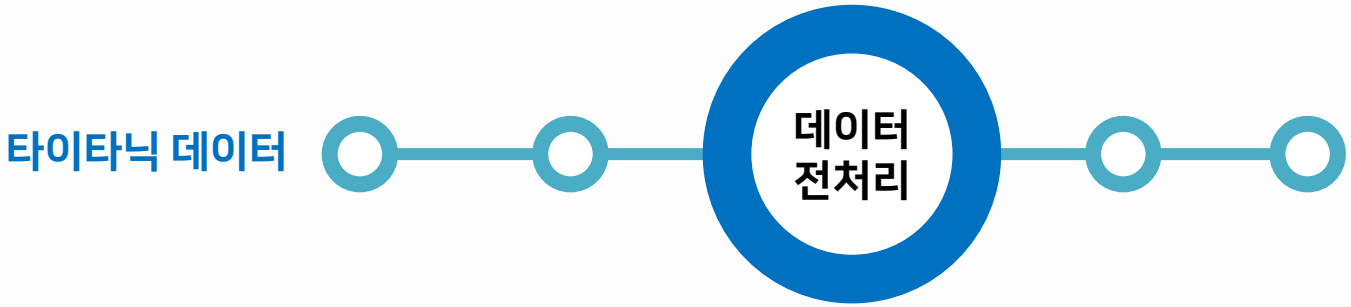
01 제목을 입력하세요

타이타닉 데이터



패키지를 이용	R과 Python 패키지에는 해당 패키지에 어울리는 데이터가 존재
크롤링	블로그나 SNS에 있는 데이터를 Python 코딩을 통해 수집
노가다	일일이 데이터를 입력
공개 데이터 활용	공공데이터 포털 및 구글 등에 공개된 데이터 포털에서 데이터 수집
구매	원하는 데이터를 가지고 있는 기업에 구매

01 내용을 입력하세요



데이터 타입 변경	데이터 타입이 chr -> factor, chr -> num, num -> chr 등 목적에 맞는 타입으로 변경
결측치 제거	비어있는 데이터를 확인하고 적절한 값을 채워준다 (ex : NA, NULL)
이상치 제거	이상한 데이터를 확인하고 제거 및 변경 해준다 (ex : 연령이 -1세 -> 0세 or 평균나이로 변경)
데이터 구간화	연령 데이터를 5세 단위 등으로 구간화 해준다 (why? : 알고리즘의 성능과 속도를 높이기 위해!)
안쓰는 컬럼 제거	알고리즘에서 사용할 수 없는 컬럼 제거

01 데이터 타입 변경

```
str(titanic_train)

'data.frame':   891 obs. of  12 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florenc
 $ Sex        : chr  "male" "female" "female" "female" ...
 $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
 $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/02. 3101282" "113803" ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Cabin      : chr  "" "C85" "" "C123" ...
 $ Embarked   : chr  "S" "C" "S" "S" ...
```

바꿔야할 데이터 타입

- 생존 여부(Survived) : Factor
- 좌석 등급(Pclass) : Factor
- 성별(Sex) : Factor
- 탑승지(Embarked) : Factor

- 타입 변경 R 코드
- Num으로 변경 : as.numeric()
- Chr으로 변경 : as.character()
- Factor로 변경 : as.factor()

01 데이터 타입 변경

```
'data.frame':   891 obs. of  12 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
 $ Pclass     : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
 $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Fl
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
 $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Cabin      : chr  "" "C85" "" "C123" ...
 $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

코드 실행 결과

- titanic_train$Survived = as.factor(titanic_train$Survived)
- titanic_train$Pclass = as.factor(titanic_train$Pclass)
- titanic_train$Sex = as.factor(titanic_train$Sex)
- titanic_train$Embarked = as.factor(titanic_train$Embarked)

- str(titanic_train)

01 결측치 제거

```
  PassengerId     Survived  Pclass      Name              Sex
 Min.   :  1.0   0:549     1:216   Length:891         female:314
 1st Qu.:223.5   1:342     2:184   Class :character   male  :577
 Median :446.0             3:491   Mode  :character
 Mean   :446.0
 3rd Qu.:668.5
 Max.   :891.0

      Age            SibSp           Parch           Ticket
 Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
 1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character
 Median :28.00   Median :0.000   Median :0.0000   Mode  :character
 Mean   :29.70   Mean   :0.523   Mean   :0.3816
 3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
 Max.   :80.00   Max.   :8.000   Max.   :6.0000
 NA's   :177
      Fare            Cabin             Embarked
 Min.   :  0.00   Length:891          :  2
 1st Qu.:  7.91   Class :character    C:168
 Median : 14.45   Mode  :character    Q: 77
 Mean   : 32.20                       S:644
 3rd Qu.: 31.00
 Max.   :512.33
```

결측치 확인

- summary(titanic_train)

- 결측치는 보통 NA or NULL로 표현
- Summary를 통해 확인해봤을 때 Age 컬럼에 177개의 NA값이 존재하는 것을 확인할 수 있음

- 결측치 확인 코드 : is.na()

01 결측치 처리

```
[32] print(mean(titanic_train_no$Age))
     print(mean(titanic_train_mean$Age))
     print(mean(titanic_train_sx$Age))
```

```
[1] 29.69912
[1] 29.69912
[1] 29.73603
```

결측치 처리

- 결측치 처리에는 여러가지 방법이 있음
- 1. 제거 : 평균 나이 : 29.6
- 2. 평균값으로 대체 : 평균 나이 : 29.6
- 3. 특정 조건의 평균값으로 대체
- → 남자, 여자 구분해서 나이 산정
- → 평균 나이 : 29.7

- 편의상 결측치가 있는 데이터는 사용X

01 결측치 처리

- 결측치 처리에는 여러가지 방법이 있음
- 1. 제거
- 2. 평균값으로 대체
- 3. 특정 조건의 평균값으로 대체
- 4. Age를 예측할 수 있는 알고리즘을 활용해서 Age를 예측 후 대체

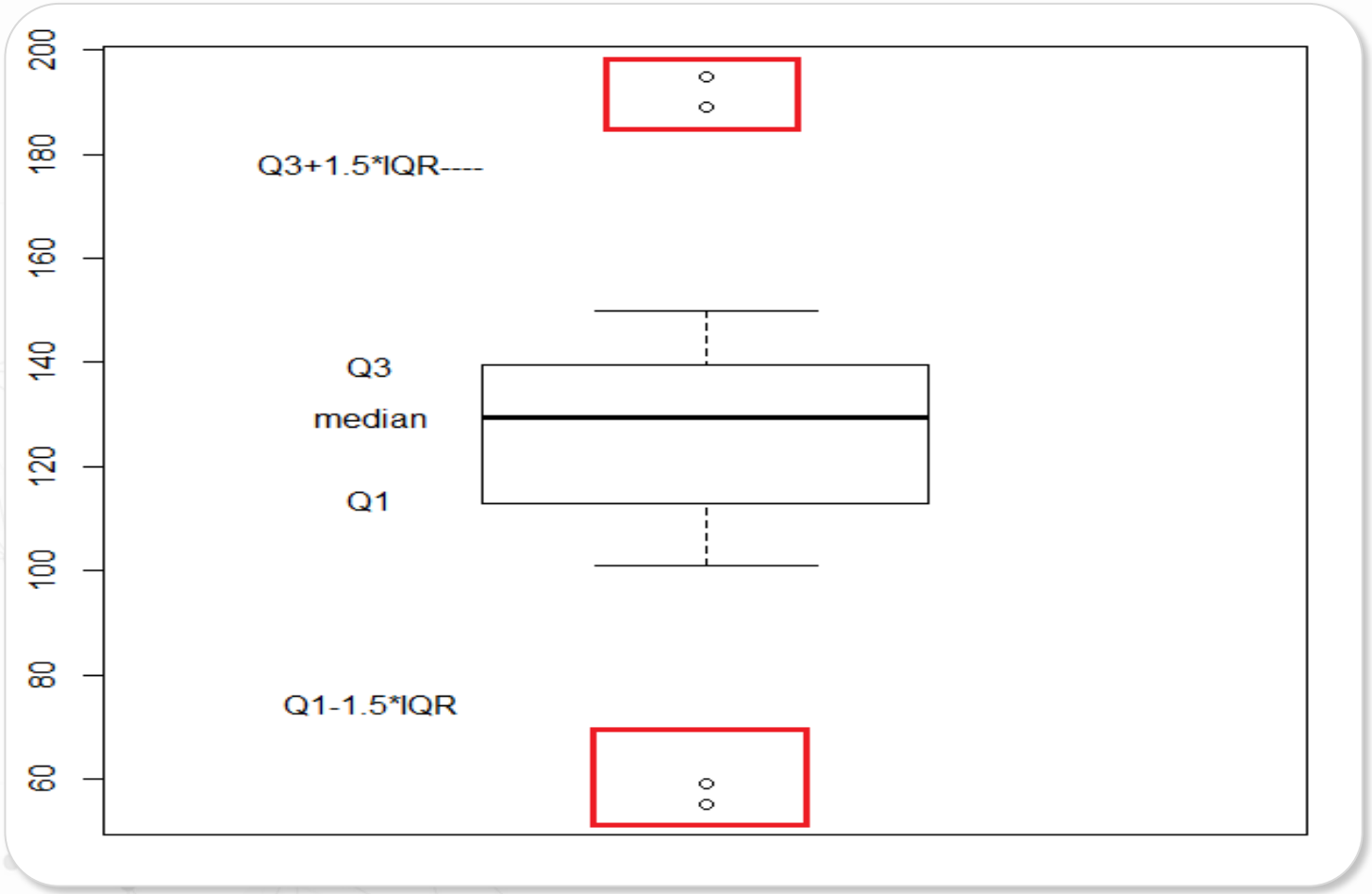
- 코드)
- 1.
- titanic_train_no = na.omit(titanic_train)
- 2.
- titanic_train_mean = titanic_train # titanic_train에 직접 변경을 하게 되면 titanic_train을 추후 활용할 수 없으니
- titanic_train_mean$Age[is.na(titanic_train_mean$Age)] = mean(titanic_train_mean$Age,na.rm = T)

01 결측치 처리

- 결측치 처리에는 여러가지 방법이 있음
- 1. 제거
- 2. 평균값으로 대체
- 3. 특정 조건의 평균값으로 대체
- 4. Age를 예측할 수 있는 알고리즘을 활용해서 Age를 예측 후 대체

- 코드)
- 3.
- titanic_train_sx = titanic_train
- titanic_train_sx$Age[is.na(titanic_train_sx$Age)&titanic_train_sx$Sex == 'female'] = mean(titanic_train_sx$Age[titanic_train_sx$Sex == 'female'], na.rm = T)
- titanic_train_sx$Age[is.na(titanic_train_sx$Age)&titanic_train_sx$Sex == 'male'] = mean(titanic_train_sx$Age[titanic_train_sx$Sex == 'male'], na.rm = T)
- 4. 알고리즘을 통한 Age 예측은 추후 심화과정에서 진행

01 이상치



이상치

§ 이상치란 이상한 데이터!

§ 예 : 다들 연봉이 3000만원인데 혼자 3억인 사람이 있다. -> 이상치일까?

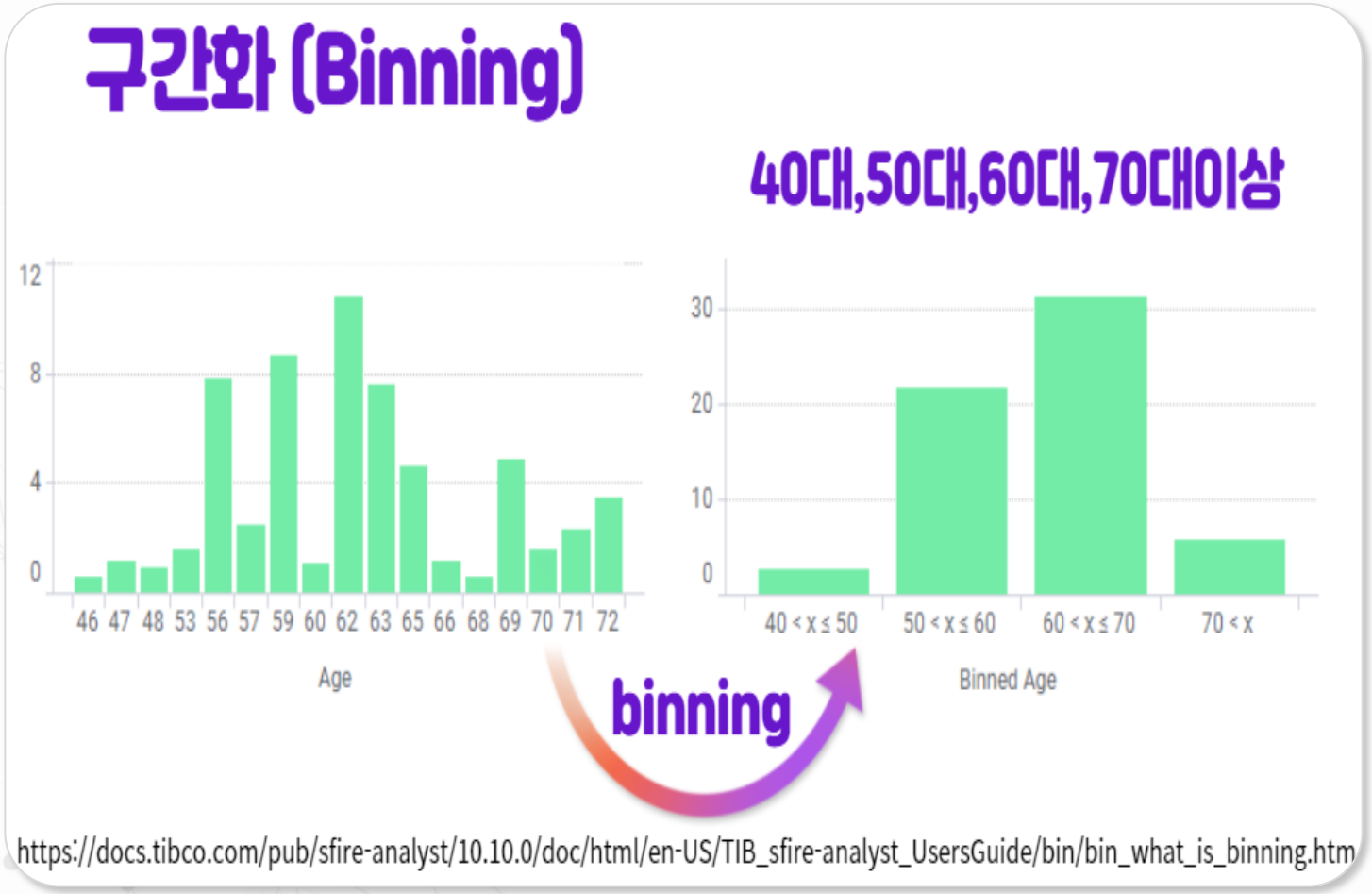
§ → 3억이 사장님이라면?

§ → 결국 이상치를 판단할 수 있는 것은 해당 데이터에 얼마나 많은 지식을 가지고 있어야 판단 가능

§ 혹은 통계적인 방법을 통해 이상치를 판단

→ 해당 내용은 통계 내용이 포함되기에 심화 학습에서 진행

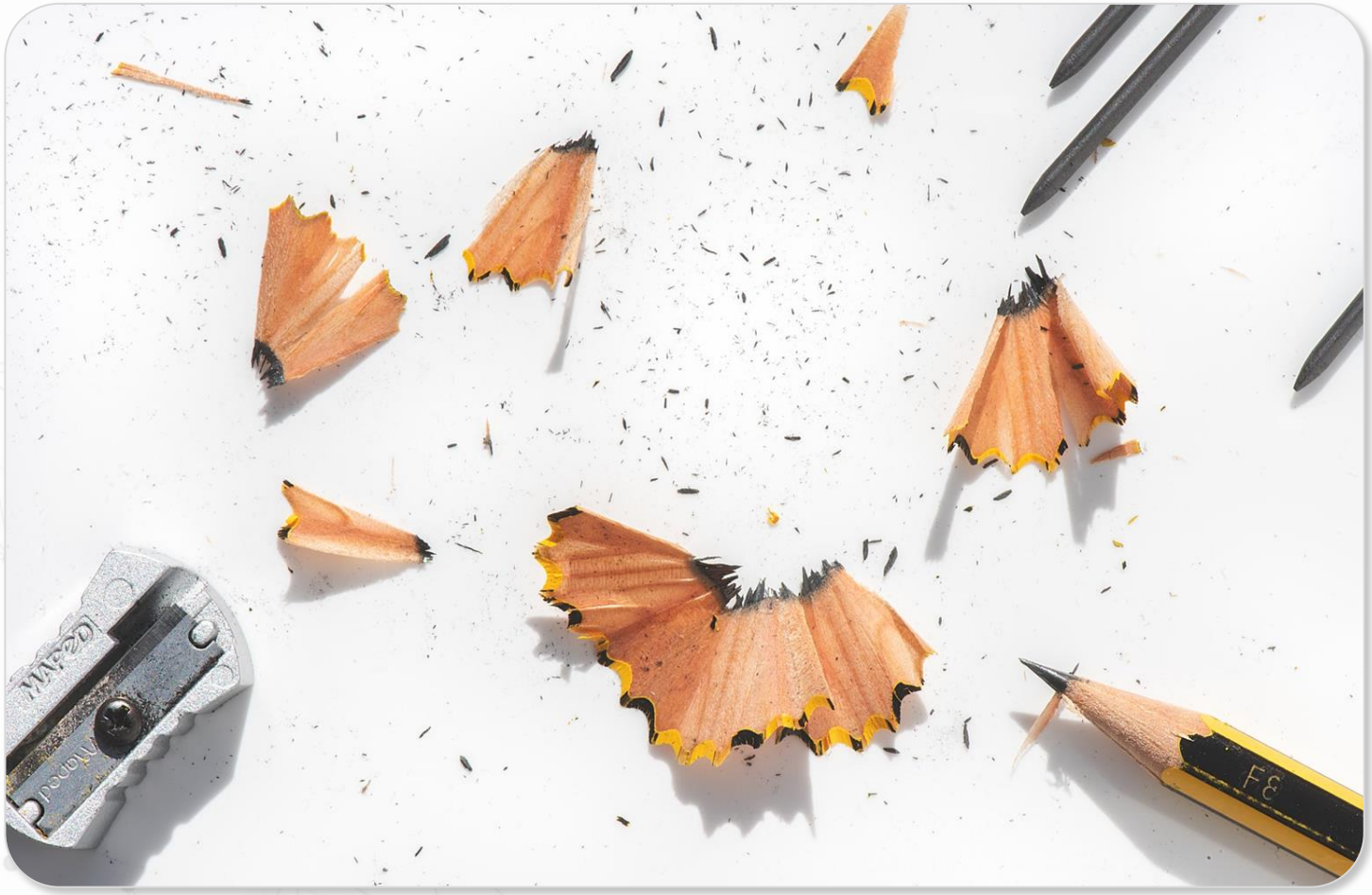
01 데이터 구간화



데이터 구간화

- 데이터 구간화는 모델링의 속도 및 결과를 향상시키기 위해 많이 고려되는 전처리 기법
- 현재 타이타닉 데이터는 데이터가 크지 않기에 데이터 구간화는 진행하지 않는다.

01 안쓰는 컬럼 제거



컬럼 제거

§ 보통 개인 ID, 이름과 같은 컬럼은 모델에 활용할 수 없다.

§ WHY?

→ 개인의 속성이기 때문

§ 개인의 속성을 알고리즘에 넣었을 때 생기는 문제

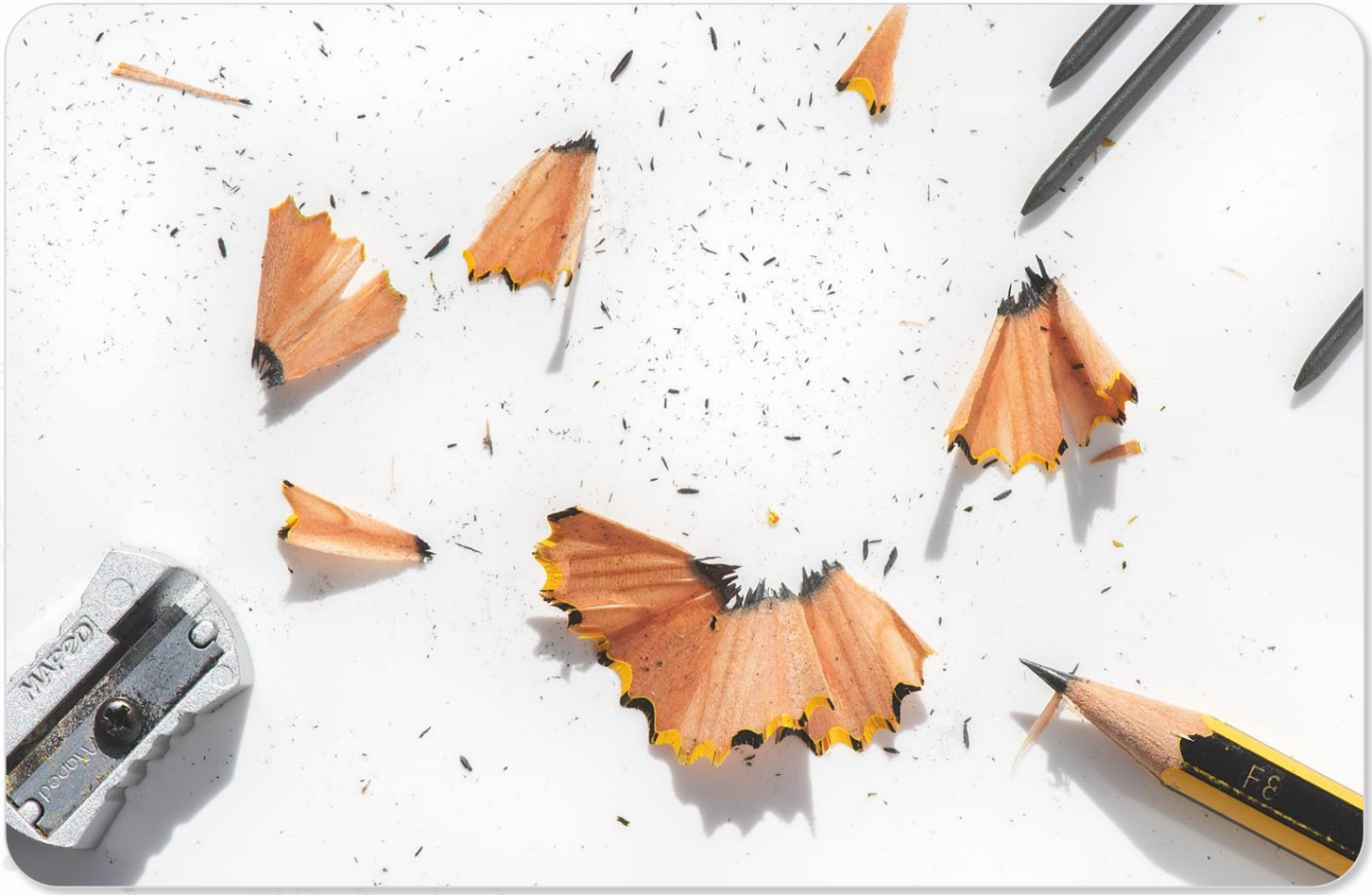
→ 민종열이라는 이름을 가지고 있는 사람은 타이타닉에서 살아남을 확률이 높다

→ 이게 과연 사용가능한 알고리즘인가?

§ 해석에 문제가 생김

§ 즉 컬럼 제거!

01 안쓰는 컬럼 제거



타이타닉 데이터에서는?

- 승객번호 (PassengerID) 제거
- 승객명 (Name) 제거
- 티켓번호(Ticket) 제거
- 방 호수 (Cabin) 제거

- 만약 방 호수에 대한 자세한 정보가 있었다면? Ex : 방에 선체에서의 위치, 방에 층수
- → 해당 방에 관련된 데이터도 사용 가능

01 안쓰는 컬럼 제거

§ 컬럼 제거 코드

§ 1. 열을 지정해서 제거

§ ※ 현재 승객번호(PassengerId), 성명(Name), 티켓번호(Ticket), 방이름(Cabin)은 1,4,9,11에 있음

§ 코드) titanic_train_no[,-c(1,4,9,11)]

§ 2. 컬럼명을 활용해 컬럼을 제거

§ 코드) subset(titanic_train_no, select = -c(PassengerId,Name,Ticket,Cabin))

§ 이후 새로운 데이터프레임 생성

§ train = subset(titanic_train_no, select = -c(PassengerId,Name,Ticket,Cabin))

01 실습

- Q1) 타이타닉 데이터를 불러오기
- Q2) 타이타닉 데이터의 구조 및 통계치를 볼 수 있는 코드를 입력하라 (hint : s**, su*****)
- Q3) 타이타닉 데이터의 생존 여부(Survived), 좌석 등급(Pclass), 성별(Sex), 탑승지(Embarked)를 요인(factor) 형태로 변경하라
- Q4) summary()를 활용하여 결측치가 있는 컬럼을 확인하라
- Q5) 결측치가 있는 데이터를 제외하고 새로운 데이터프레임을 생성하라
- Q6) 승객번호(PassengerId), 성명(Name), 티켓번호(Ticket), 방이름(Cabin)을 제거한 Train 데이터를 생성하라

01 실습

- Q1) 타이타닉 데이터를 불러오기
- A1) Install.packages('titanic')
- library(titanic)
- titanic_train # 타이타닉 학습용 데이터
- titanic_test # 타이타닉 평가용 데이터 (생존여부 X)

- Q2) 타이타닉 데이터의 구조 및 통계치를 볼 수 있는 코드를 입력하라 (hint : s**, su*****)
- A2) str(titanic_train) # 데이터 구조 확인
- summary(titanic_train) # 데이터 분포 확인

- Q3) 타이타닉 데이터의 생존 여부(Survived), 좌석 등급(Pclass), 성별(Sex), 탑승지(Embarked)를 요인(factor) 형태로 변경하라
- A3)
- titanic_train$Survived = as.factor(titanic_train$Survived)
- titanic_train$Pclass = as.factor(titanic_train$Pclass)
- titanic_train$Sex = as.factor(titanic_train$Sex)
- titanic_train$Embarked = as.factor(titanic_train$Embarked)

01 실습

§ Q4) summary()를 활용하여 결측치가 있는 컬럼을 확인하라

§ A4) summary(titanic_train)

§ Q5) 결측치가 있는 데이터를 제외하고 새로운 데이터프레임을 생성하라

§ A6) titanic_train_no = na.omit(titanic_train)

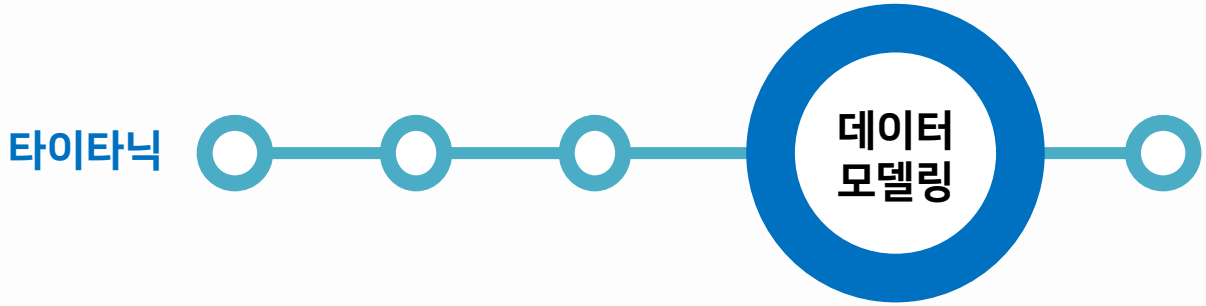
§ Q6) 승객번호(PassengerId), 성명(Name), 티켓번호(Ticket), 방이름(Cabin)을 제거한 Train 데이터를 생성하라

§ A6) train = subset(titanic_train_no, select = -c(PassengerId,Name,Ticket,Cabin))

Chapter

II 탐색적 데이터 분석

02 데이터 모델링



탐색적 데이터 분석	탐색적 데이터 분석을 통해 다른 변수와 목표 변수간의 관계를 미리 확인
데이터 모델링	데이터 형태에 따라 알맞은 모델은 선택
모델 튜닝	하이퍼 파라미터 조정 등의 모델 성능을 높이기 위한 다양한 시도를 진행

02 탐색적 데이터 분석이란?

탐색적 데이터 분석(EDA)

§ 알고리즘을 생성하기 전 데이터를 파악하기 위한 분석

§ 탐색적 데이터 분석으로도 충분히 좋은 분석 결과를 가져갈 수 있다.

§ 탐색적 자료 분석 : 상관관계, 시각화, 대시보드 -> 대부분 시각화가 주가 된다고 생각

§ ※ 다양한 시각화 기법을 활용한다면 인사이틀 추출할 수 있음

02 배달 데이터를 이용한 탐색적 자료 분석

탐색적 데이터 분석(EDA)

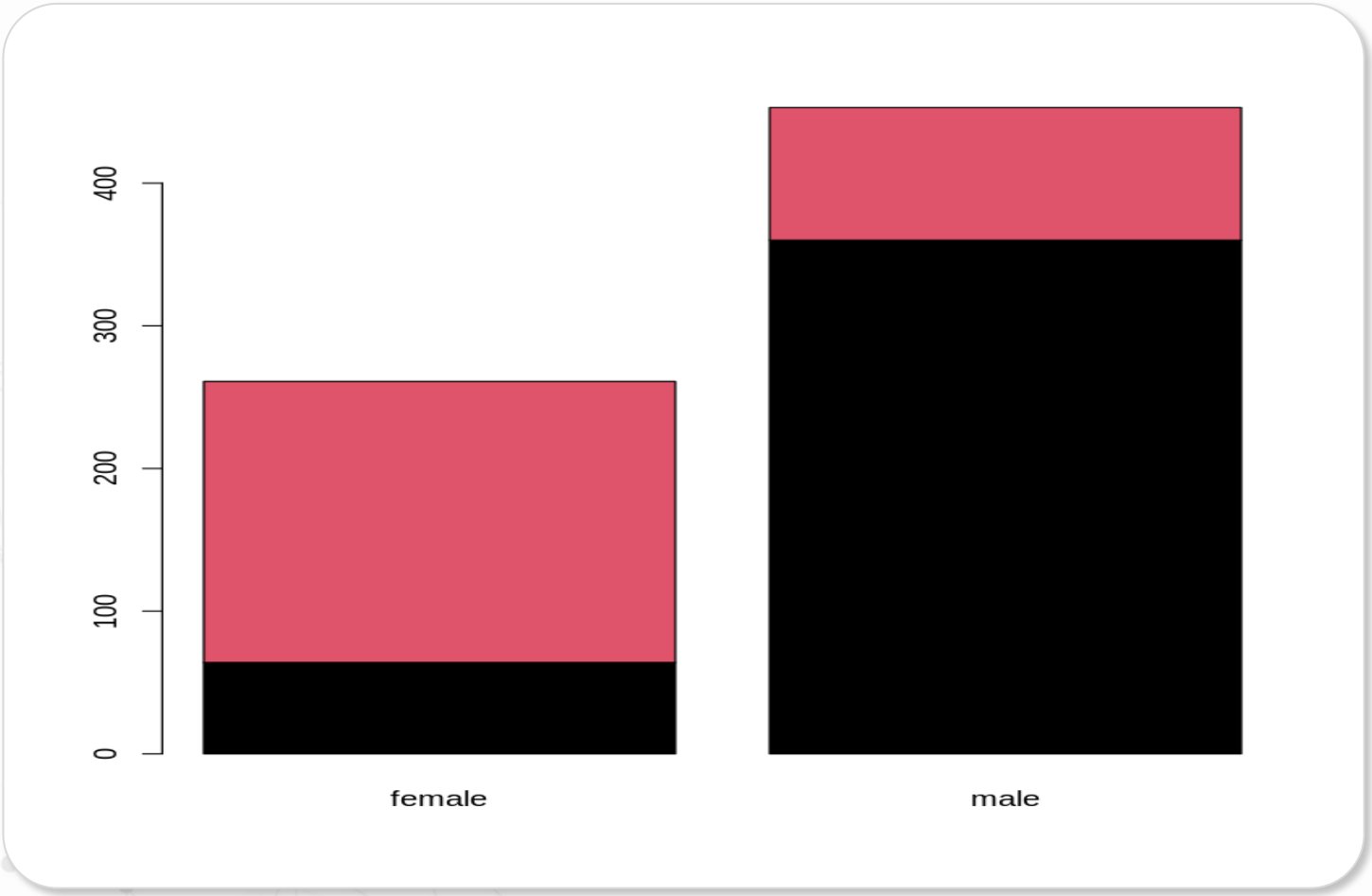
- 1년 중 배달음식 주문이 적은 시기는 설날과 추석이다
- 1년 중 배달음식 주문이 가장 많은 시기는 12월 24일이다.
- 3. 점심에는 중국음식을 저녁에는 치킨을 가장 많이 주문한다.
- 4. 강서구 주민들이 배달음식을 가장 많이 주문한다.

02 배달 데이터를 이용한 탐색적 자료 분석

탐색적 데이터 분석(EDA)

- 1년 중 배달음식 주문이 적은 시기는 설날과 추석이다
- 1년 중 배달음식 주문이 가장 많은 시기는 12월 24일이다.
- 3. 점심에는 중국음식을 저녁에는 치킨을 가장 많이 주문한다.
- 4. 강서구 주민들이 배달음식을 가장 많이 주문한다.

02 성별에 따른 생존여부



성별에 따른 생존여부

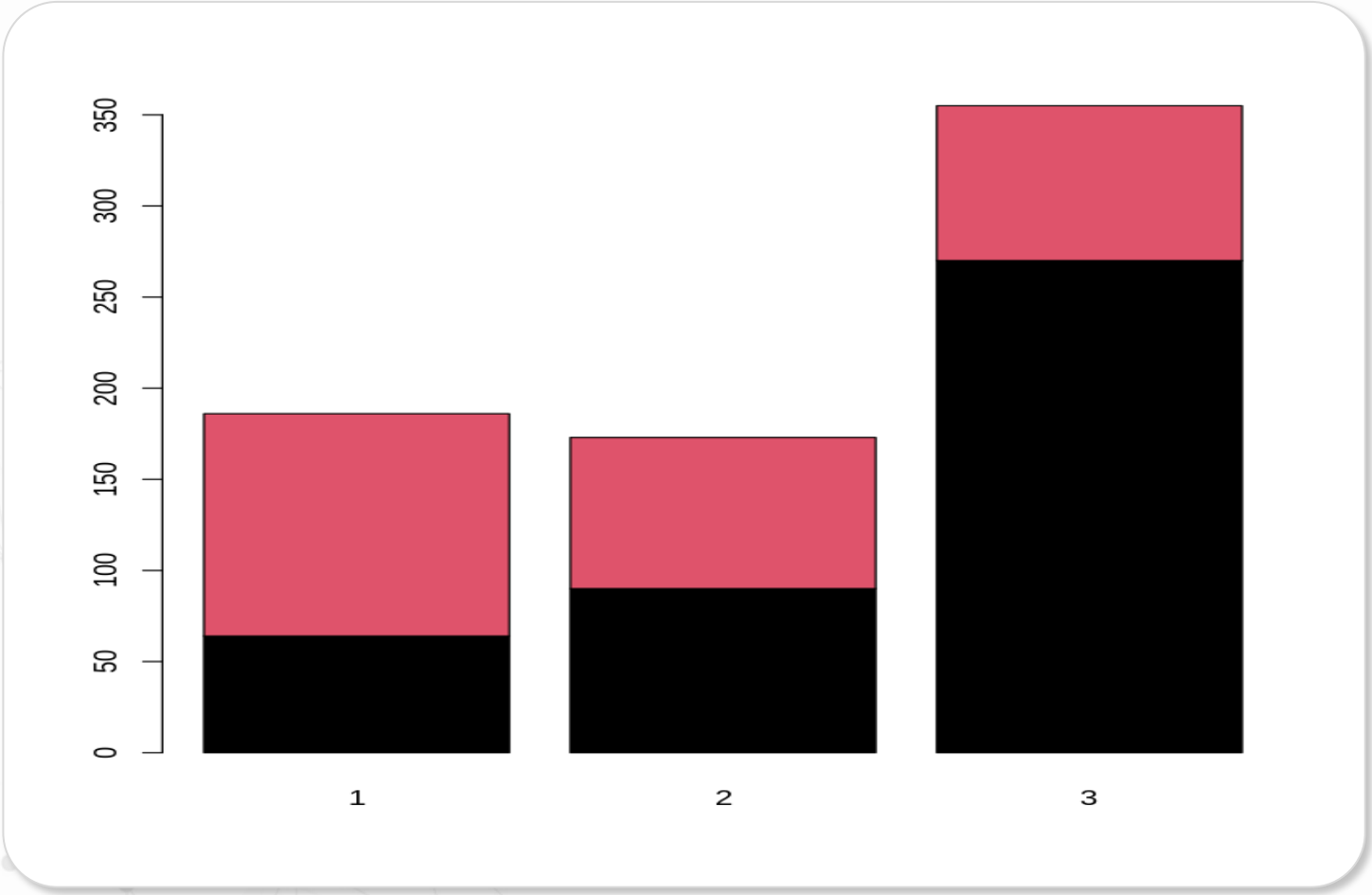
§ 핑크 : 생존, 검은색 : 죽음

§ 육안으로 확인해도 여성이 남성에 비해 생존율이 좋음

§ 코드)

§ barplot(height = table(train$Survived,train$Sex), col = unique(train$Survived))

02 좌석에 따른 생존여부

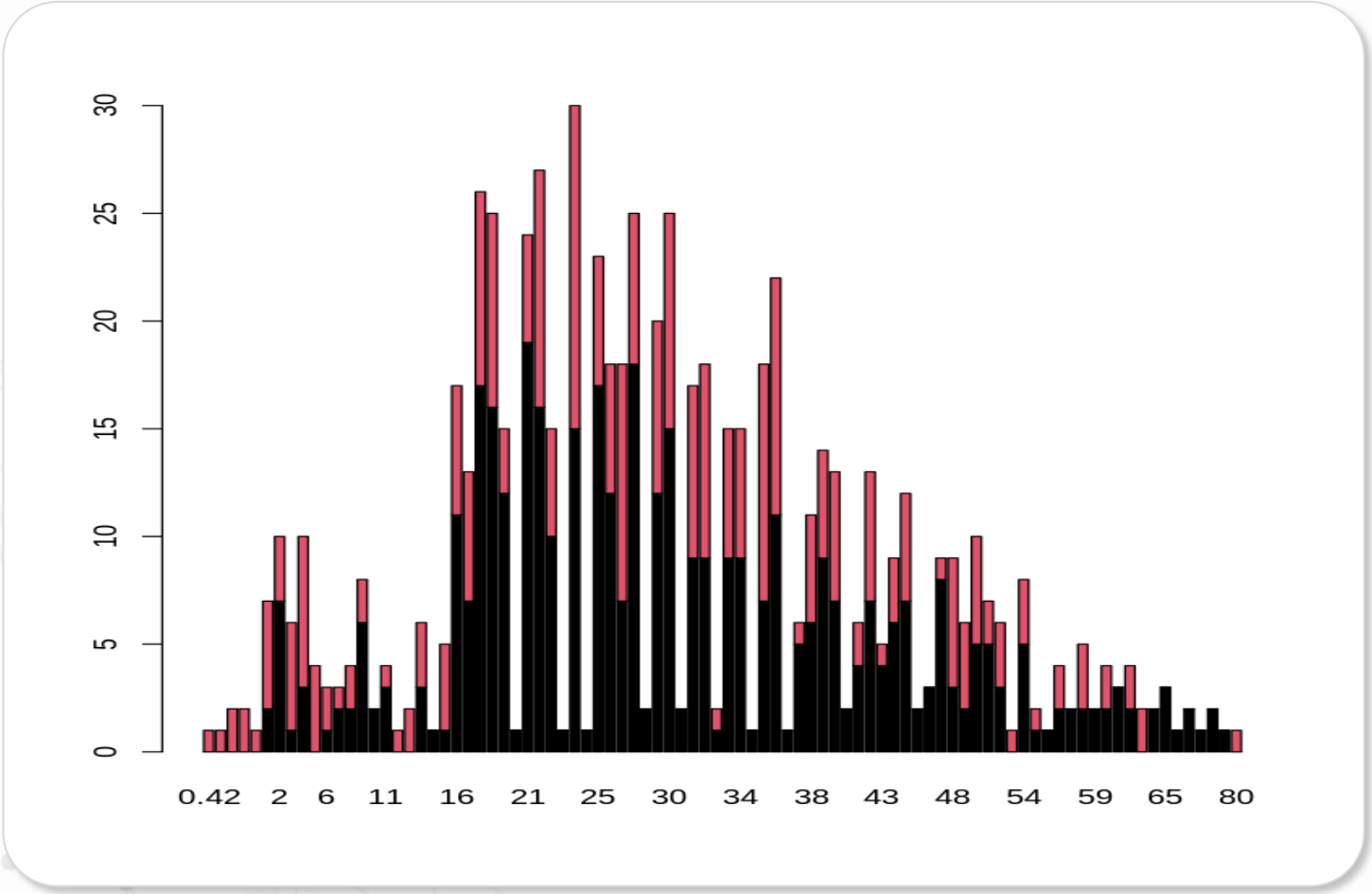


좌석에 따른 생존여부

- 핑크 : 생존, 검은색 : 죽음
- 3등석의 승객들의 생존율이 좋지 못함

- 코드)
- barplot(height = table(train$Survived,train$Pclass), col = unique(train$Survived))

02 나이에 따른 생존여부

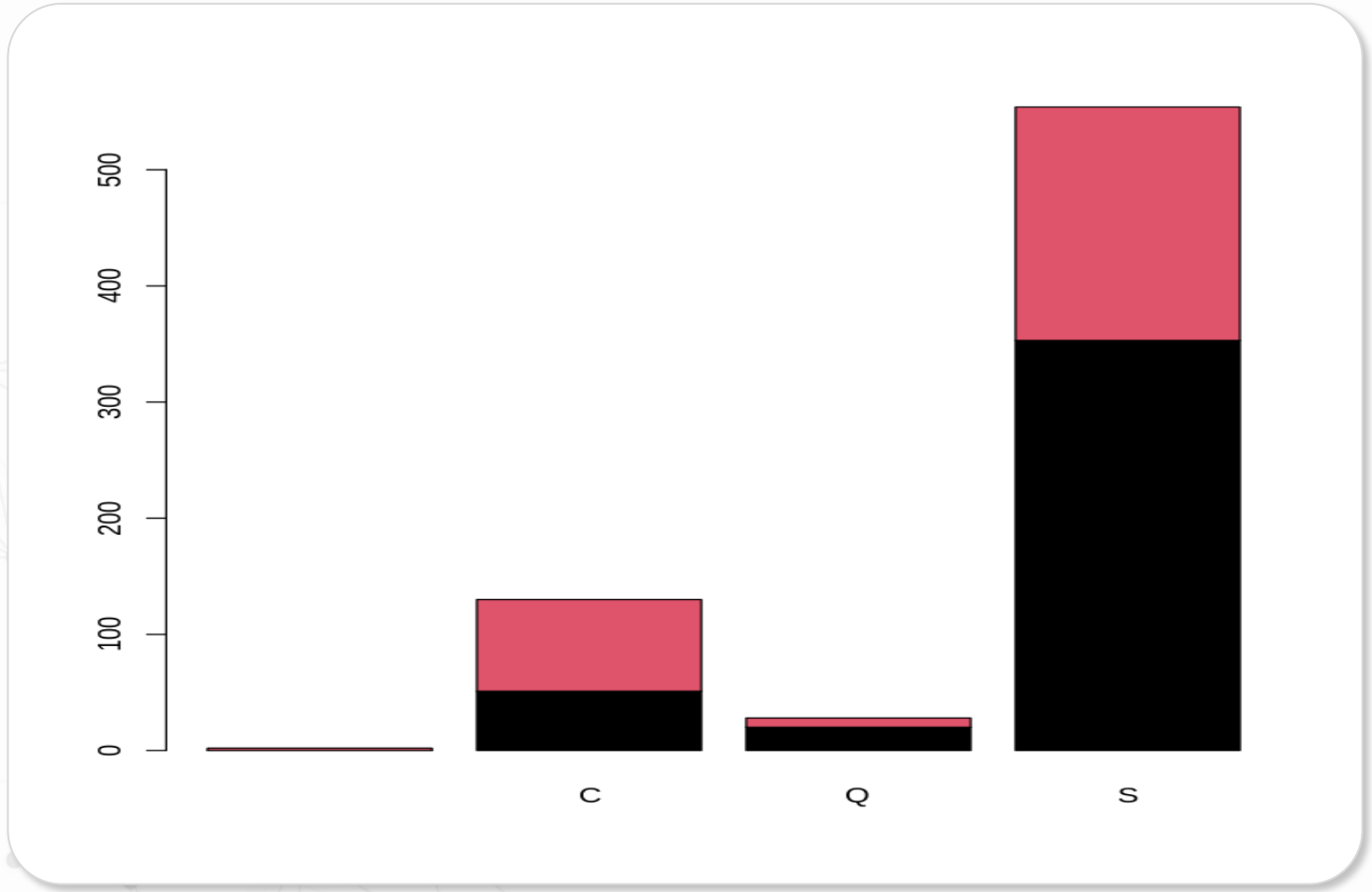


나이에 따른 생존여부

- 핑크 : 생존, 검은색 : 죽음
- 나이에 따른 생존율은 나이가 숫자형태로 존재하기에 확인하기가 어려움
- 나이를 구간화해야만 의미를 파악할 수 있을 것으로 보임

- 코드)
- barplot(height = table(train$Survived,train$Age), col = unique(train$Survived))

02 출발지에 따른 생존여부



출발지에 따른 생존여부

- 핑크 : 생존, 검은색 : 죽음
- 비율로 확인을 해야 봐야함
- 비율은 육안상 비슷해 보임

- 코드)
- barplot(height = table(train$Survived,train$Embarked), col = unique(train$Survived))

- 탐색적 자료 분석은 여기까지!

02 실습

- Q1) 성별에 따른 생존여부 차이를 시각화하라
- Q2) 좌석에 따른 생존여부 차이를 시각화하라
- Q3) 나이에 따른 생존여부 차이를 시각화하라
- Q4) 출발지에 따른 생존여부 차이를 시각화하라

02 실습

§ Q1) 성별에 따른 생존여부 차이를 시각화하라

§ A1) barplot(height = table(train$Survived,train$Sex), col = unique(train$Survived))

§ Q2) 좌석에 따른 생존여부 차이를 시각화하라

§ A2) barplot(height = table(train$Survived,train$Pclass), col = unique(train$Survived))

§ Q3) 나이에 따른 생존여부 차이를 시각화하라

§ A3) barplot(height = table(train$Survived,train$Age), col = unique(train$Survived))

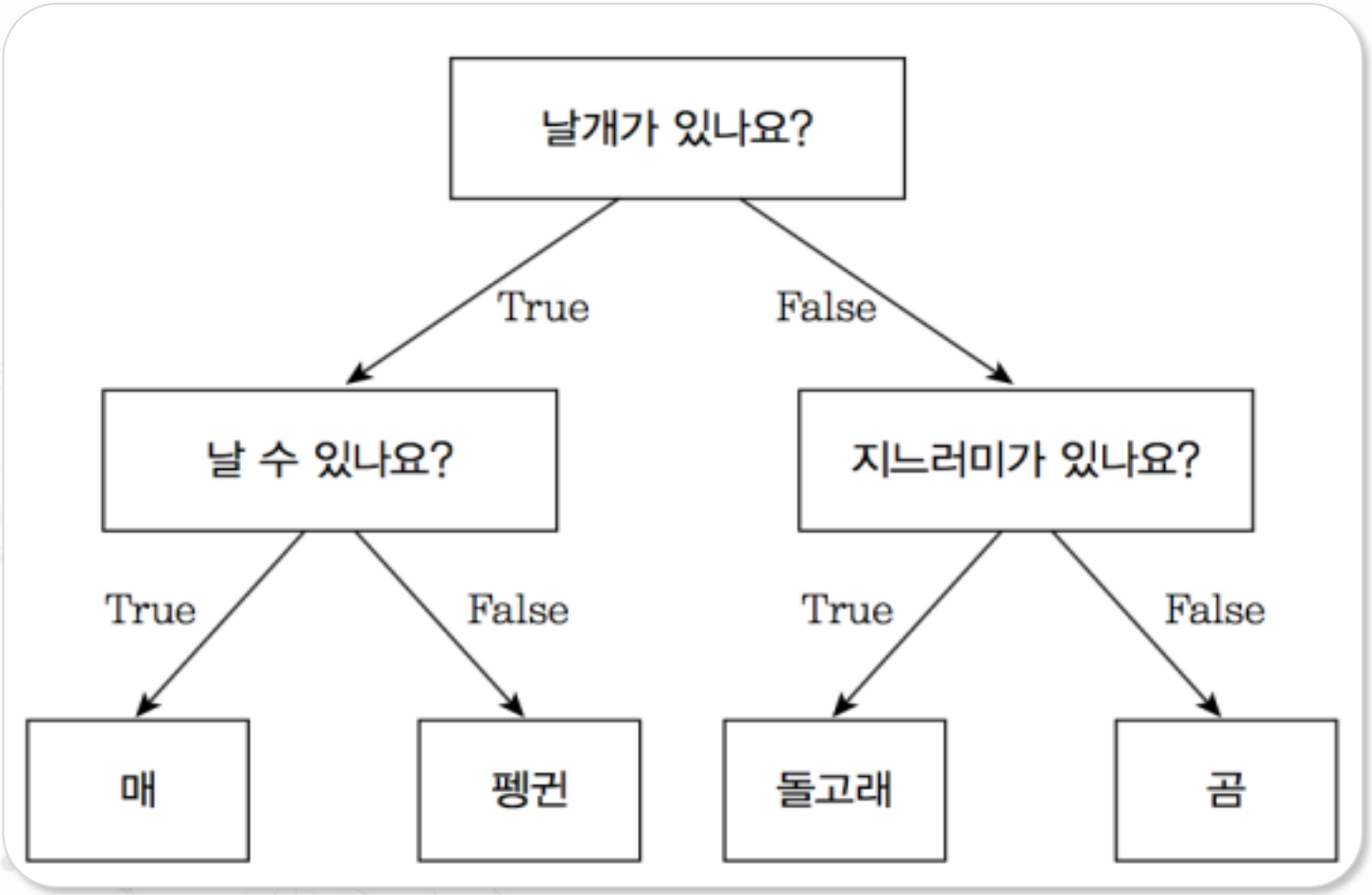
§ Q4) 출발지에 따른 생존여부 차이를 시각화하라

§ A4) barplot(height = table(train$Survived,train$Embarked), col = unique(train$Survived))

Chapter

III 데이터 모델링

03 우리가 사용할 모델



의사결정나무

§ 생존여부를 가장 많이 분리하는 컬럼을 기준으로 가지를 생성

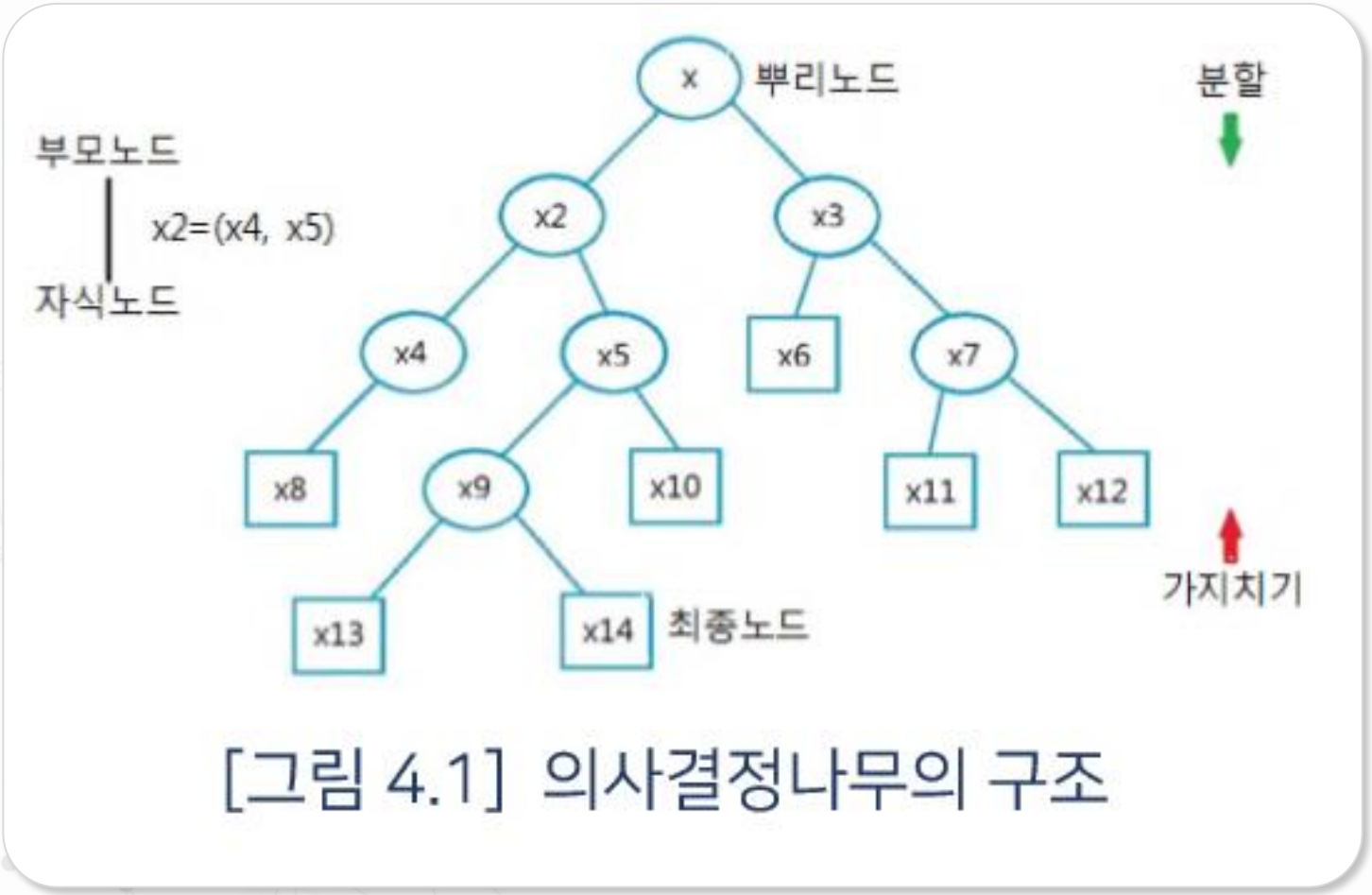
§ 예시) 동물 분류 의사결정나무

§ 1. 날개는 매, 펭귄, 돌고래, 곰을 가장 명확하게 나누는 기준

§ 2. 날 수 있음의 여부는 매, 펭귄을 나누는 가장 확실하게 나는 기준

§ 3. 지느러미는 곰과 돌고래를 나누는 기준

03 우리가 사용할 모델



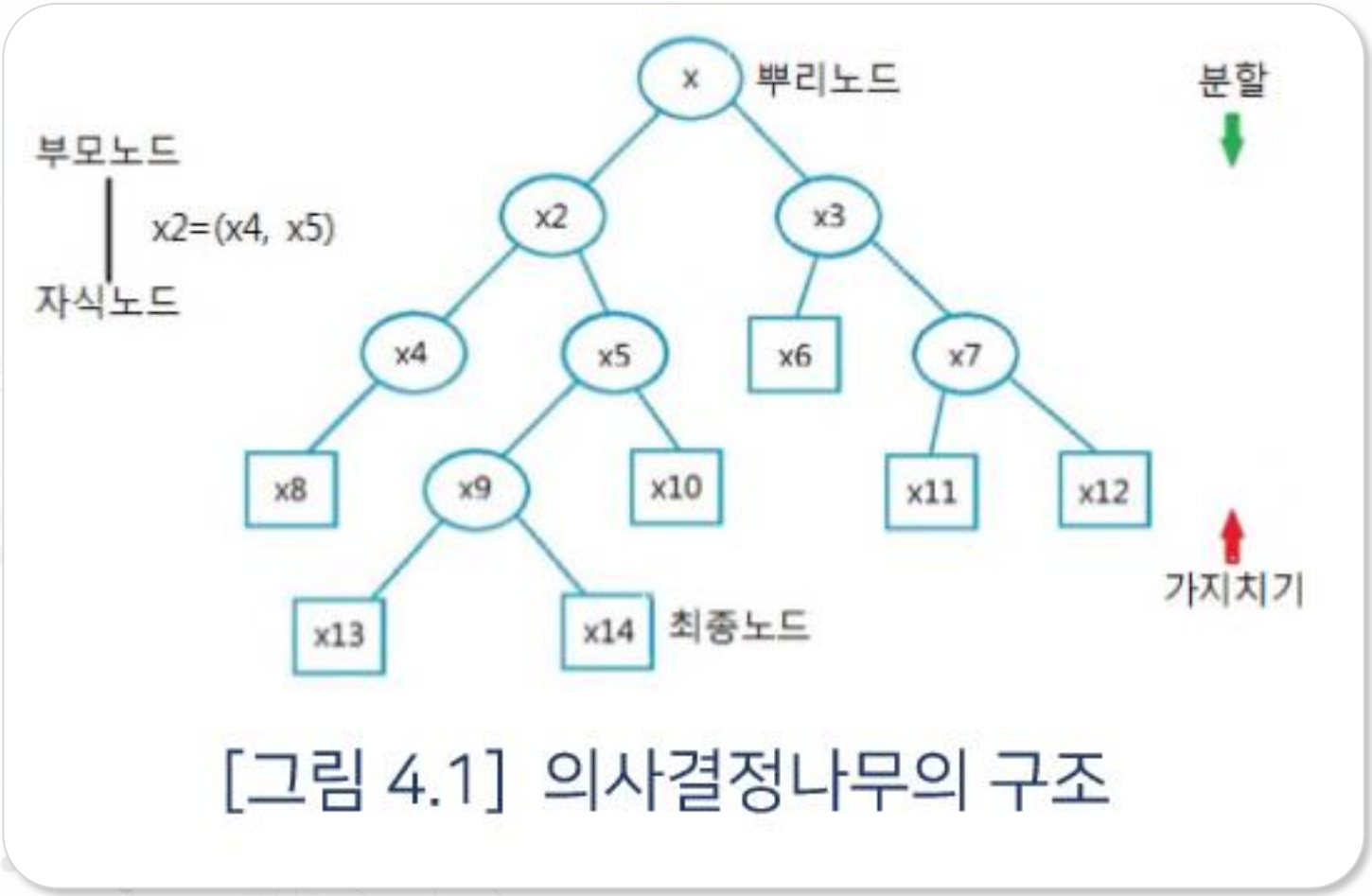
[그림 4.1] 의사결정나무의 구조

의사결정나무

- 코드)
- install.packages('rpart')
- library(rpart)

- dt = rpart(Survived ~ ., data = train)
- ※ 의사결정나무를 만드는 코드
- ※ dt에 의사결정나무 모델이 생성됨

03 모델 튜닝



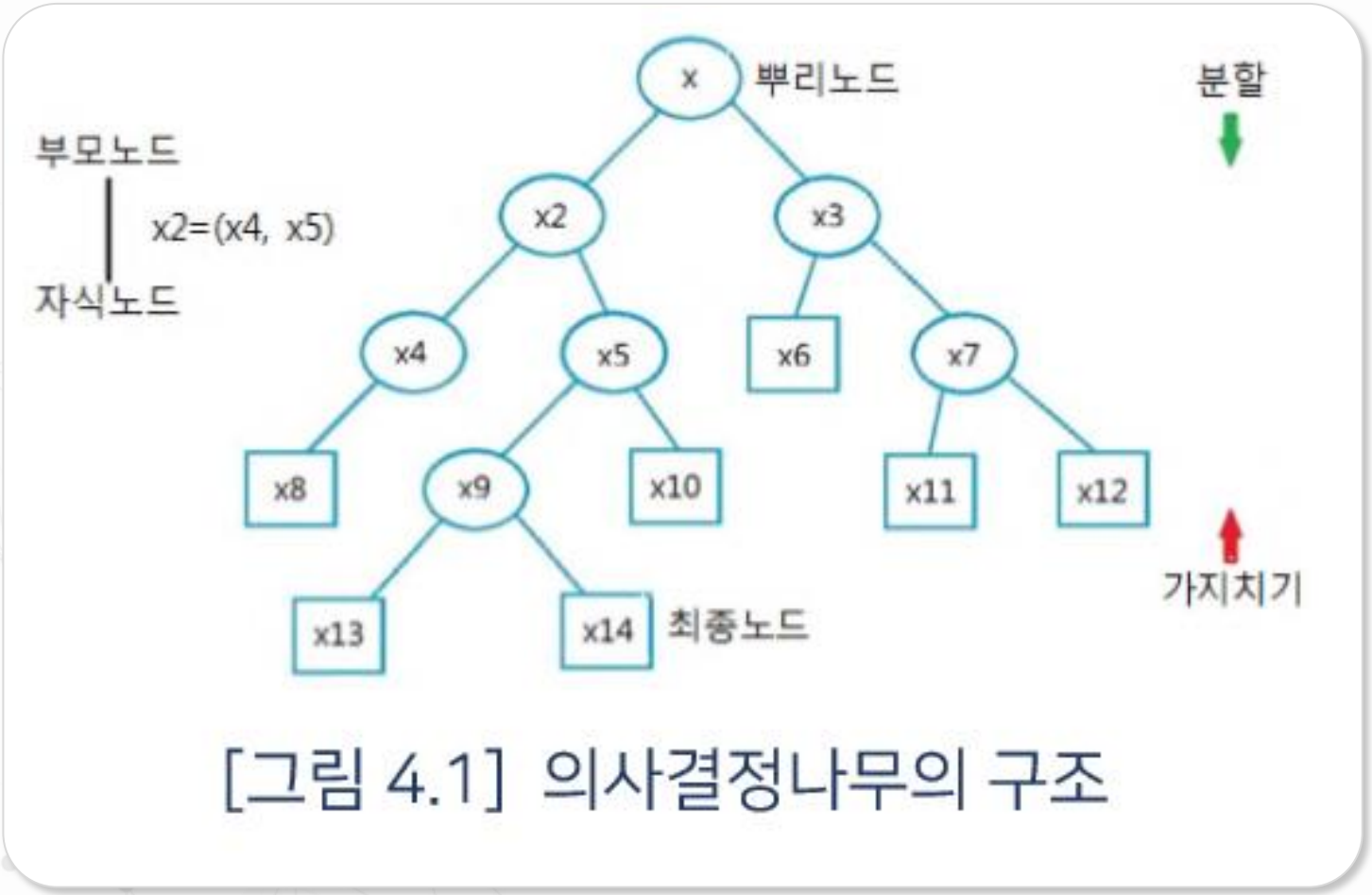
[그림 4.1] 의사결정나무의 구조

모델 튜닝

§ 튜닝을 위해 rpart 함수를 확인하는 코드

§ help(rpart)

```
rpart                    package:rpart
R Documentation

_□R_□e_□c_□u_□r_□s_□i_□v_□e
_□P_□a_□r_□t_□i_□t_□i_□o_□n_□i_□n_□g
_□a_□n_□d
_□R_□e_□g_□r_□e_□s_□s_□i_□o_□n
_□T_□r_□e_□e_□s

_□D_□e_□s_□c_□r_□i_□p_□t_□i_□o_□n:

     Fit a  'rpart'  model

_□U_□s_□a_□g_□e:

     rpart(formula, data, weights,
subset, na.action = na.rpart, method,
           model = FALSE, x = FALSE, y
= TRUE, parms, control, cost, ...)
```

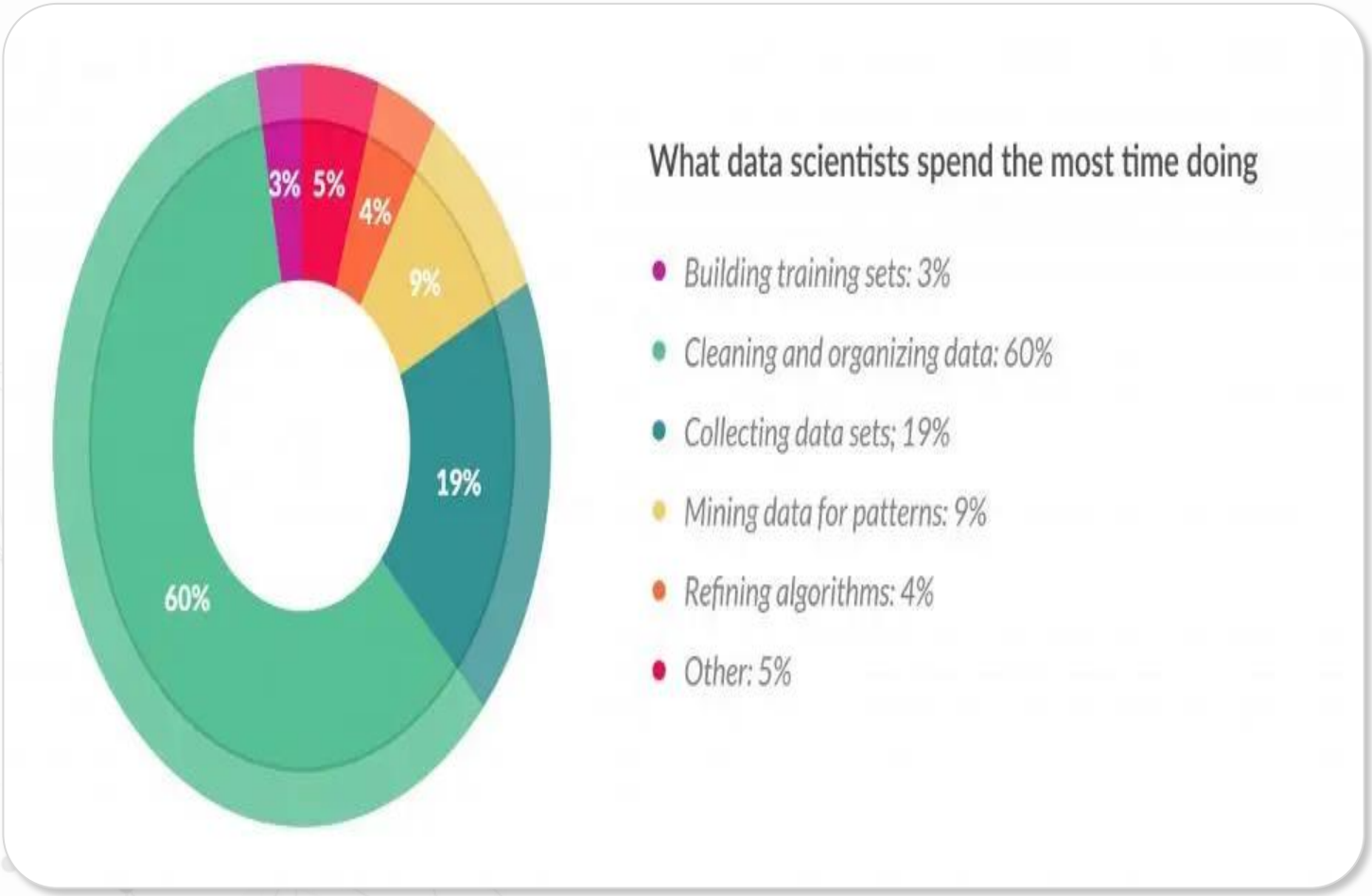


[그림 4.1] 의사결정나무의 구조

모델 튜닝

§ 모델 튜닝의 경우 해당 모델을 잘 알고 있어야 하기에 해당 내용은 추후 심화학습을 진행시 얘기하도록 하겠음

03 왜 이렇게 모델링은 짧을까?



포브스 선정

§ R & Python의 경우 모델 관련된 코드는 다른 통계학자 및 공학자들이 개발을 해 놓았다.

§ 그렇기에 우리가 모델을 구현할 필요 없이 해당 패키지를 사용하면 된다.

§ 분석가가 가장 시간을 많이들이는 단계

§ 1. 데이터 전처리 (60%)

§ 2. 데이터 수집 (19%)

§ 3. 학습용 데이터셋 생성 및 모델링(7%)

§ 4. 해석 및 기타 (8%)

03 실습

§ Q1) 의사결정나무 모델을 생성하라

03 실습

§ Q1) 의사결정나무 모델을 생성하라

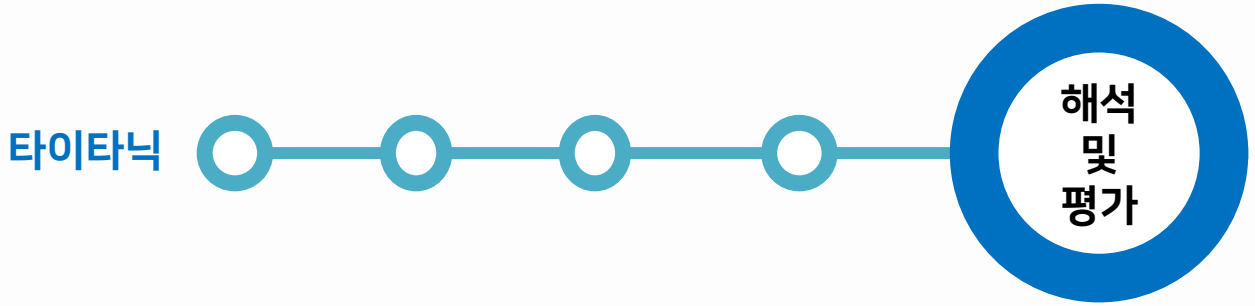
§ A1)

```
install.packages('rpart')
library(rpart)

dt = rpart(Survived ~ ., data=train)
```

Chapter

Ⅳ 평가 및 해석

04 해석 및 평가



모델 평가	모델의 성능을 평가
모델 해석	모델을 해석
중요 변수 확인	생존여부에 가장 영향을 많이 주는 중요 변수를 확인

04 평가 지표에 대해 알아보자! (Confusion Matrix)

		실제	
		Y (1)	N (0)
예측	Y (1)	True Positive (TP)	False Positive (FP)
	N (0)	False Negative (FN)	True Negative (TN)

구성 성분

- True Positive(TP)
- 실제 Positive & 예측 Positive
- False Positive(FP)
- 실제 Positive & 예측 Negative
- True Negative(TN)
- 실제 Negative & 예측 Negative
- False Negative(FN)
- 실제 Negative & 예측 Positive

04 평가 지표에 대해 알아보자! (Confusion Matrix)

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

$$specificity = \frac{TN}{TN + FP}$$

평가지표

- 특이도(Specificity) : 실제 Negative 중 Negative로 예측한 비율
- 민감도(Sensitivity | Recall) : 실제 Positive 중 Positive로 예측한 비율
- 정밀도(precision) : Positive로 예측한 것 중 실제 Postive인 비율
- 정확도(accuracy) : 예측한 것 중 실제로 맞춘 비율

04 모델의 평가지표

pred \ real	0	1
0	401	90
1	23	200

모델의 평가지표

- 코드)
- pred = predict(tf, type = 'class')
- → 의사결정나무를 활용해 예측 진행
- real = train$Survived
- → 실제 정답

- table(pred = pred, real = real)
- → TN, FN, TP, FP 생성
- → 해당 쿼리를 사용해 직접 계산을 해도 되지만 우리에게는 caret 패키지가 있다!

04 모델의 평가지표

```
Confusion Matrix and Statistics

          real
pred         0   1
    0      401  90
    1       23 200

                   Accuracy : 0.8417
                     95% CI : (0.8129, 0.8678)
       No Information Rate : 0.5938
       P-Value [Acc > NIR] : < 2.2e-16

                      Kappa : 0.6595

    Mcnemar's Test P-Value : 5.341e-10

                Sensitivity : 0.6897
                Specificity : 0.9458
             Pos Pred Value : 0.8969
             Neg Pred Value : 0.8167
                 Prevalence : 0.4062
             Detection Rate : 0.2801
       Detection Prevalence : 0.3123
          Balanced Accuracy : 0.8177

           'Positive' Class : 1
```

caret 패키지 사용

- § 코드)
- § Install.packages('caret')
- § → caret 패키지 다운로드
- § library(caret)
- § → 패키지 사용

- § confusionMatrix(table(pred = pred, real = real),positive = '1')
- § → confusion Matrix 생성

04 모델의 평가지표

```
Confusion Matrix and Statistics

          real
pred         0    1
    0      401   90
    1       23  200

                    Accuracy : 0.8417
                      95% CI : (0.8129, 0.8678)
       No Information Rate : 0.5938
       P-Value [Acc > NIR] : < 2.2e-16

                     Kappa : 0.6595

 Mcnemar's Test P-Value : 5.341e-10

                Sensitivity : 0.6897
                Specificity : 0.9458
             Pos Pred Value : 0.8969
             Neg Pred Value : 0.8167
                 Prevalence : 0.4062
             Detection Rate : 0.2801
       Detection Prevalence : 0.3123
          Balanced Accuracy : 0.8177

           'Positive' Class : 1
```

caret 패키지 사용

- § 특이도(Specificity) : 실제 Negative 중 Negative로 예측한 비율
- § → 0.9458
- § 민감도(Sensitivity | Recall) : 실제 Positive 중 Positive로 예측한 비율
- § → 0.6897
- § 정밀도(precision) : Positive로 예측한 것 중 실제 Postive인 비율
- § → 0.8167
- § 정확도(accuracy) : 예측한 것 중 실제로 맞춘 비율
- § → 0.8417

04 모델의 평가지표

```
Confusion Matrix and Statistics

          real
pred         0   1
    0      401  90
    1       23 200

                   Accuracy : 0.8417
                     95% CI : (0.8129, 0.8678)
    No Information Rate : 0.5938
    P-Value [Acc > NIR] : < 2.2e-16

                      Kappa : 0.6595

 Mcnemar's Test P-Value : 5.341e-10

                Sensitivity : 0.6897
                Specificity : 0.9458
             Pos Pred Value : 0.8969
             Neg Pred Value : 0.8167
                 Prevalence : 0.4062
             Detection Rate : 0.2801
       Detection Prevalence : 0.3123
          Balanced Accuracy : 0.8177

           'Positive' Class : 1
```

결과 확인

§ 모델이 80% 이상의 정확도를 가지고 있기에 모델이 나쁘지 않다고 자체적으로 판단을 내리겠음

§ 몇 %의 정확도를 가지고 있어야 좋은 모델일까?

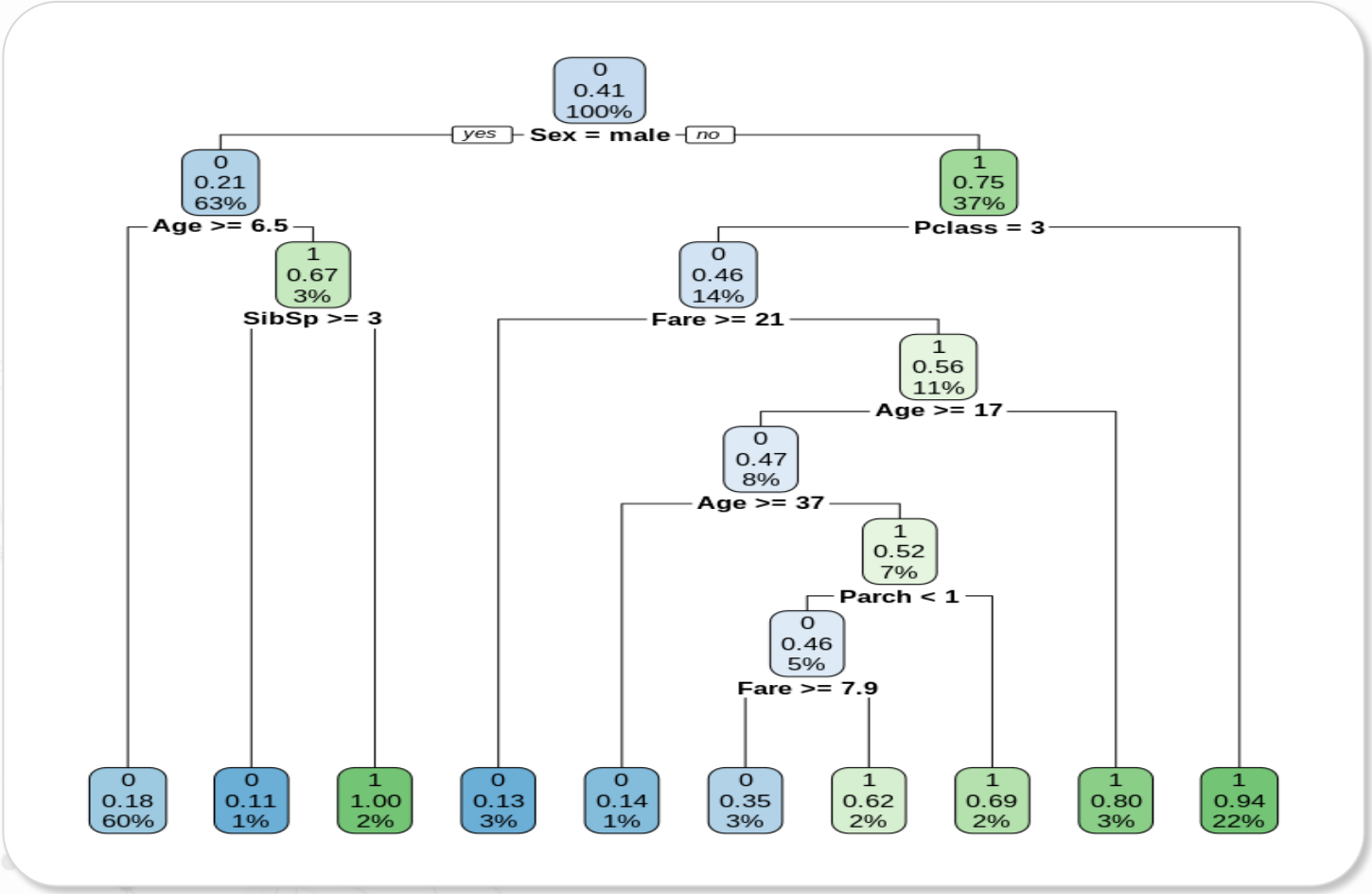
§ → 도메인(영역)에 따라 다르다

§ → 도메인 지식을 갖춰야 한다.

§ → 다른 모델과의 비교를 통해 가장 좋은 모델을 선택

§ → 제 경험상 0.7 이상의 정확도를 가지고 있으면 나쁘지 않은 모델

04 해석

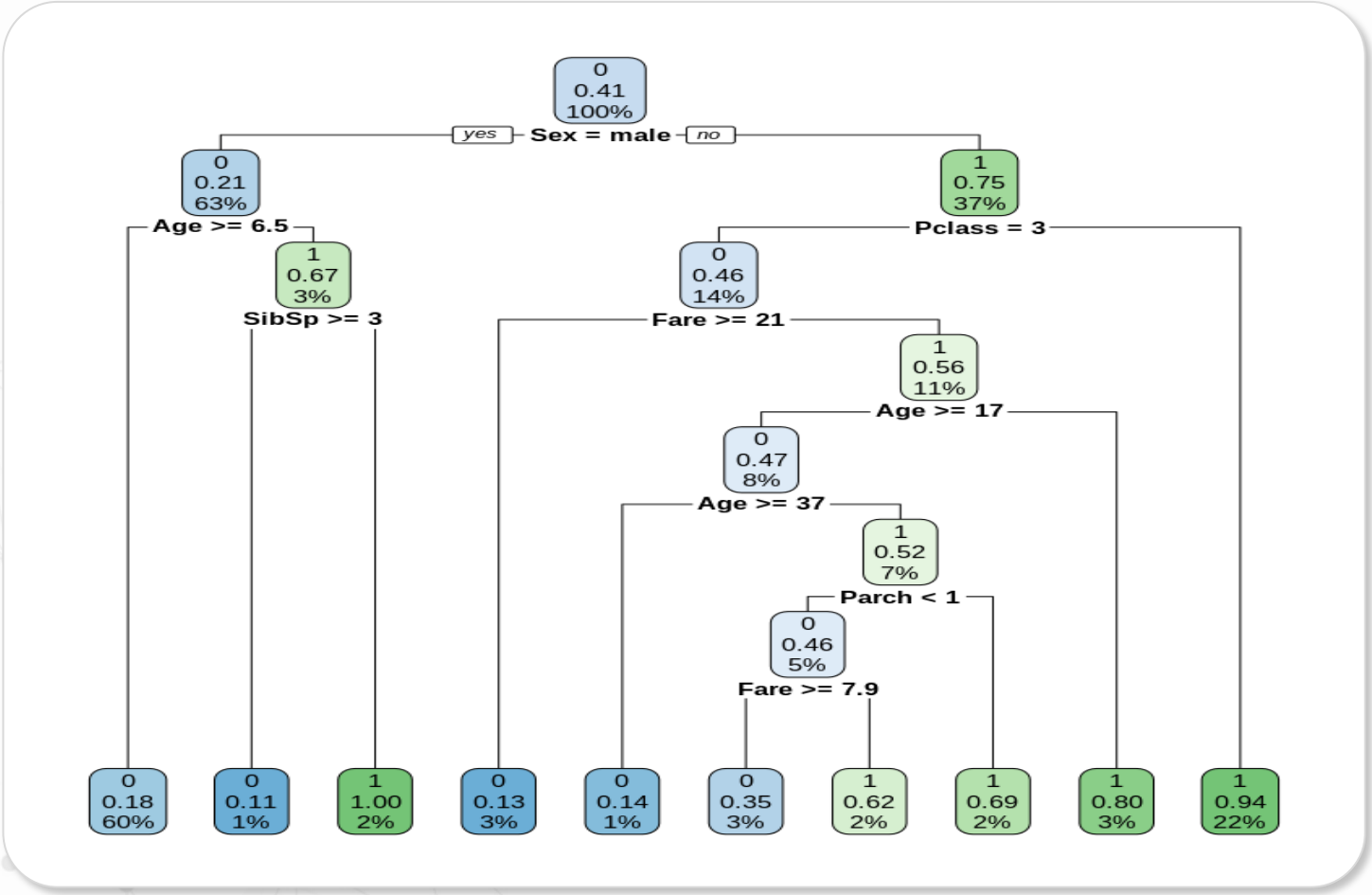


의사결정나무 해석

§ 코드)

```
install.packages('rpart.plot')
library('rpart.plot')
rpart.plot(dt)
```

04 해석



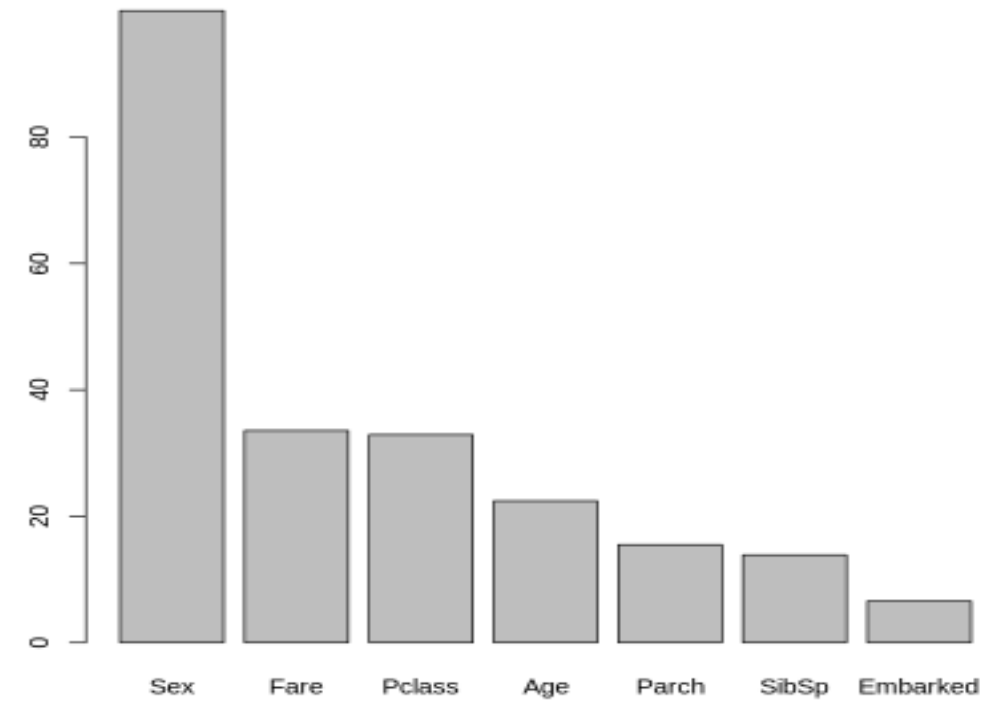
의사결정나무 해석

- 왼쪽으로 가는 것으로 해석 진행!

- 1번 가지 : 성별이 남성이면 죽는 것으로 판단
- 1 -> 2번 가지 : 나이가 7세 이상이면 죽을 것으로 판단
- 2 -> 3번 가지 : 나이가 6세 이하더라도 동승한 형제, 배우자수가 3명 이상이면 죽을 것으로 판단

04 해석

```
      Sex      Fare    Pclass       Age     Parch     SibSp  Embarked
99.998174 33.541304 32.896238 22.420936 15.500505 13.878592  6.573743
```



변수 중요도

- 생존여부를 판단하는데 가장 중요한 컬럼

- 1. 성별
- 2. 티켓가격
- 3. 좌석등급
- 4. 나이
- 5. 동승 부모/자식 수
- 6. 동승 형제/배우자 수
- 7. 탑승지

코드)
print(dt$variable.importance)
barplot(dt$variable.importance)

04 실습

- Q1) 모델을 활용해 결과(Survived)를 예측해라 (predict())
- Q2) 모델의 평가지표를 생성해라 (table(), confusionMatrix())
- Q2) 의사결정나무 plot을 생성해라
- Q3) 변수중요도를 확인해라

04 실습

- Q1) 모델을 활용해 결과(Survived)를 예측해라 (predict())
- pred = predict(dt, type = 'class')

- Q2) 모델의 평가지표를 생성해라 (table(), confusionMatrix())
- A2)
- real = train$Survived
- table(pred = pred, real = real)
- install.packages('caret')
- library(caret)
- confusionMatrix(table(pred = pred, real = real),positive = '1')

- Q3) 의사결정나무 plot을 생성해라
- A3)
- install.packages('rpart.plot')
- library('rpart.plot')
- rpart.plot(tf)

00 실습

- Q4) 변수중요도를 확인해라
- A4)
- print(tf$variable.importance)
- barplot(tf$variable.importance)

R을 마치며

궁금한게 있으시면

Name : 민종열 대리

Email : wpdntm3001@naver.com

Hp : 010-5439-5931

수고하셨습니다.

한국고용정보원
Korea Employment Information Service