08. R 정형데이터 분석 04
나이브 베이지안 분류모델

성현곤

충북대학교 도시공학과
Dept.of URBAN ENGINEERING

목차

나이브 베이즈 모델

- 나이브 베이지안 모델
  - *a Supervised Machine Learning algorithm based on the Bayes Theorem*
  - 종속(예측) 변수가 2개 이상인 수준을 가진 명목(분류) 변수이면서 분류 예측 모델
  - 설명(특징) 변수가 요인으로 분류되는 경우 보다 더 적합한 예측 모델링 기법
- 사용 예시
  - 스팸 이메일 필터링과 같은 텍스트 분류
  - 컴퓨터 네트워크에서 침입이나 비정상행위 탐지
  - 일련의 관찰된 증상에 대한 의학적 질병 진단 등

- 왜 나이브(Naive)인가?
  - 데이터의 특징(설명) 변수들이 모두 동등하게 중요하고 독립적이라는 다소 비 현실적인 가정을 하고 있기 때문임
    - 예: 비가 오는 날의 예측에서 습도가 매우 중요하지만 다른 변수들과 동등하게 중요하다고 간주

나이브 베이즈 모델 개요

베이즈 정리(Bayes' Theorem)

- 베이즈 정리(Bayes' Theorem)
  - 프랑스의 수학자이자 천문학자인 피에르 시몽 라플라스(Laplace)가 완성
  - 어떤 사건의 결과가 동시에 함께 발생할 수 없다.
    - 상호배타적이고 포괄적인(mutually exclusive and exhaustive) 관계 가정
  - 사건 B가 일어난 경우, 사건 A도 일어날 조건부 확률 (P(A|B))
    - A의 확률은 B의 확률에 종속적이다.
  - P(A∩B)는 A와 B가 동시에 발생할 확률
  - P(B)는 B가 발생할 확률
  - 베이즈 정리는 P(A|B)의 최대 우도추정치는 B가 발생한 모든 경우에 A가 B와 함께 발생할 확률의 비율

우도(Likelihood) 사전확률

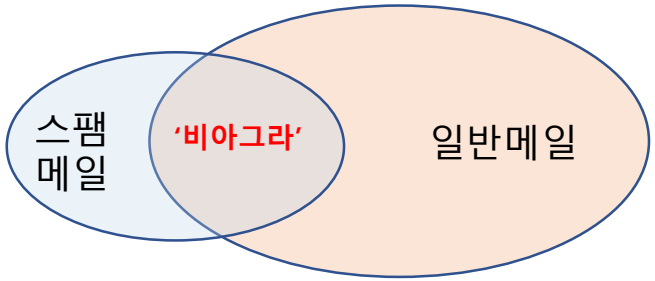
$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B \cap A)}{P(B)} = \frac{P(B|A) \cdot P(A)}{P(B)}$$

사후확률 (Posterior probability)

한계 우도 (marginal likelihood)

사후 확률은 우도 확률과 사전 확률의 곱에 비례

이메일에 '비아그라' 단어가 들어있을 경우, 스팸이멜로 분류될 확률



$$P(\text{스팸메일}|\text{비아그라}) = \frac{P(\text{비아그라}|\text{스팸메일}) \cdot P(\text{스팸})}{P(\text{비아그라})}$$

확률(probability), 우도(가능도, likelihood), 최대우도(maximum likelihood)
- 확률: 어떤 주어진 확률분포에서 관측값(또는 구간)이 얼마의 확률을 가지는가?
- 우도: 어떤 값이 관측되었을 때, 이 값이 어떤 확률분포에서 왔는 지에 대한 추정 확률(확률의 확률)
- 최대우도: 각 관측값에 대한 총 우도의 최대가 되게 하는 분포

나이브 베이즈 모델 개요

라플라스 근사와 추정량

- 라플라스 근사(Laplace Approximation)
  - 사후 분포를 근사하기 위한 방법 중의 하나
    - 사후 확률 분포가 애매하지만 그 분포와 가장 비슷한 가우시안 분포를 찾고 이를 대신 사용하는 방법
    - 어떤 확률 분포에 대한 근사 분포로서 가우시안 분포를 찾는 과정
  - 나이브 베이즈 공리와 라플라스 추정량
    - 나이브 베이즈 확률은 조건부 확률
      - 즉, 우도가 0%라면 해당 사후확률을 0으로 만들어, 다른 증거를 실질적으로 무효화하고 기각하게 됨
      - 이에 대한 해결책으로 '라플라스 추정량'을 사용
  - 라플라스 추정량
    - 데이터 학습과정에서의 빈도표의 각 합계에 작은 숫자를 더하여 각 클래스에 대하여 발생할 확률이 0이 되지 않도록 보장하는 방법
      - 사건 발생 확률이 매우 희소한 행렬(빈도표)에 어느 하나도 0이 되지 않도록 설정
    - 훈련 데이터셋이 아주 크다면 라플라스 추정량은 필요가 없음
    - R에서 기본값은 0임

우도(Likelihood) 사전확률

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B \cap A)}{P(B)} = \frac{P(B|A) \cdot P(A)}{P(B)}$$

사후확률 (Posterior probability)

한계 우도 (marginal likelihood)

사후 확률은 우도 확률과 사전 확률의 곱에 비례

나이브 베이즈 모델 장단점

- 나이브 베이즈 알고리즘의 장단점

장점	단점
- 간단하고 빠르고 매우 효율적이다. - 잡음과 누락 데이터를 잘 처리한다. - 훈련에는 상대적으로 적은 예시가 필요하지만, 대용량의 예시에도 매우 잘 작동한다. - 예측을 위한 추정 확률을 쉽게 얻을 수 있다.	- 모든 특징이 동등하게 중요하고 독립이라는 가정이 잘못된 경우가 자주 있다. - 수치 특징이 많은 데이터셋에는 이상적이지 않다. - 추정된 확률이 예측된 클래스보다 덜 신뢰할만하다.

출처: 브레트 란츠 (2017) R을 활용한 머신 러닝, 에이콘(p.151)

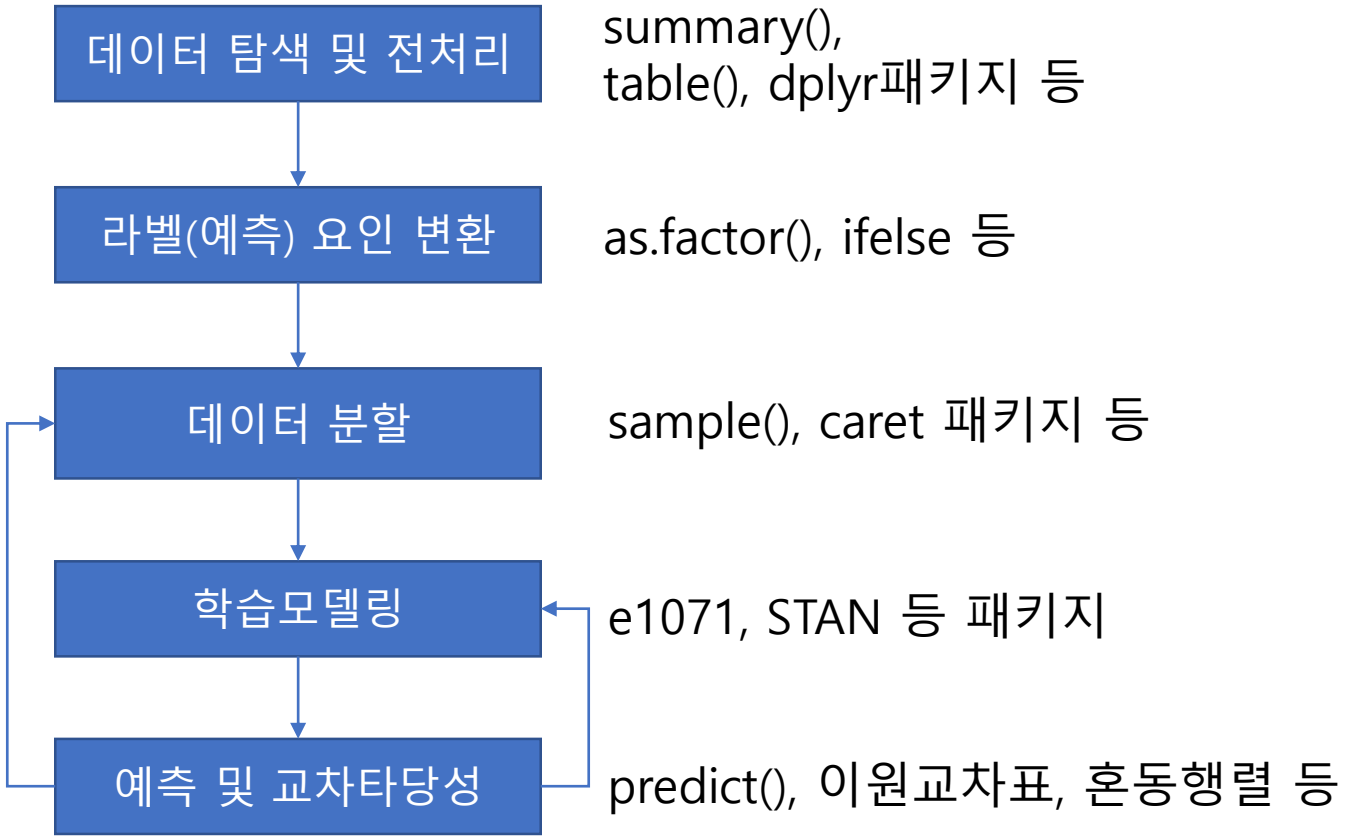
  - 집단(수준) 분류 예측모델에서 높은 효율성을 가지고 있으나, 인공지능과 같이 오류 발생이 극도로 적어야 하는 경우에는 적용하기 어려움

- 나이브 베이즈 알고리즘에서의 변수 변형
  - 독립(특징) 변수들이 연속형인 경우에는 구간으로 적절히 나누어 이산화(discrete)하여 실행할 수 있음
    - 이산화로 변환할 시에는 가지고 있는 연속변수의 속성에 대한 정보의 손실이 없도록 하여야 함
    - 데이터 분포에서 자연스러운 범주 또는 절단점(cutting points)을 찾기 위하여 탐색필요
    - 명확한 절단점이 없다면 분위값, 평균, 중위값 등을 활용하여 이산화

나이브 베이즈 모델 분석절차

- R 패키지
  - E1071,
  - klaR,
  - Naivebayes,
  - Bnclassify,
  - Caret 등

- 나이브 베이지안 분류예측 절차



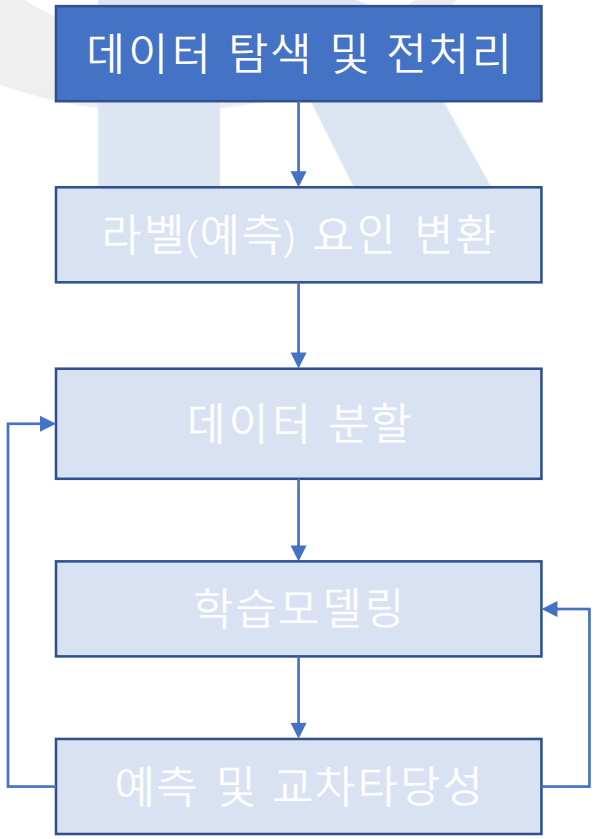
연습문제 01

- 아래와 같은 내용을 활용하여 나이브 베이즈 조건부 확률 공리로 스팸 이메일로 분류될 사후 확률을 구하라.
  - 내가 받은 총 100개의 이메일 중에서 스팸메일이 10개로 분류되었고, 그 스팸 이메일 중에서 '비아그라 ' 라는 단어가 발견되어진 이메일은 총 5개이다.

연습문제 02

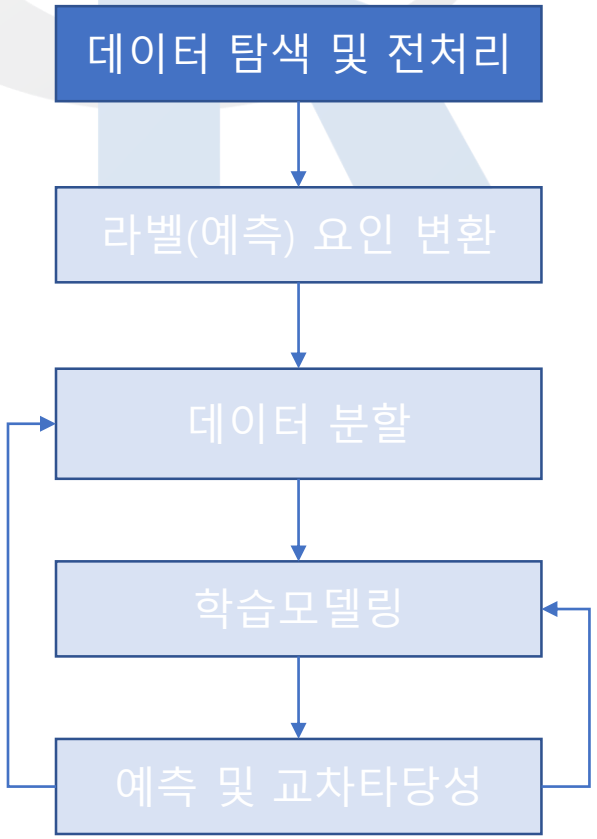
- 나이브 베이즈 분류 예측모델에서 라플라스 추정량을 사용하여야 하는 이유와 사용하지 않아도 되는 경우를 설명하시오.

실습 1: 자원봉사활동 참여 여부(e1071 패키지 활용)



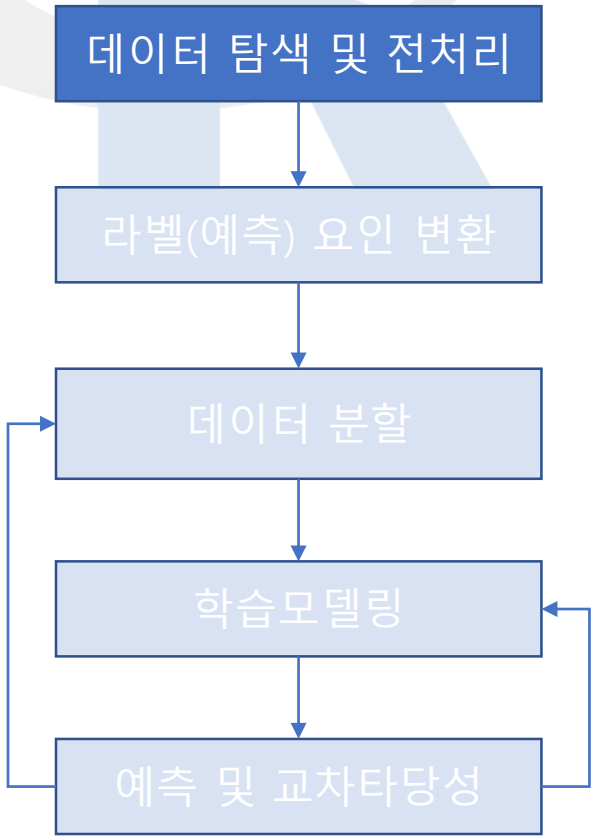

```
> # 베이지언 분류 모델을 지원해 주는 e1071, STAN같은 다양한 패키지
> library(e1071) # 미설치시 install.packages("e1071")
> library(caret) # 미설치시 install.packages("caret")
> library(ggplot2)
> library(dplyr)
> ## 실습 1: 지난 1년간 자원봉사활동 참여 여부(e1071 패키지 활용)
> setwd("K:\\기타\\2019년2학기\\수치해석\\실습데이터")
> list.files() # 현재 작업폴더 파일 확인
 [1] "(2018년 기준) 서울서베이 조사표_가구용.pdf"     "~$실습자료_서울서베이_가구원.xlsx"
 [3] "apt2.csv"                                      "apt3.csv"
 [5] "data_by_sigungu (1).csv"                       "data_by_sigungu_2018.xlsx"
 [7] "df.seoul.csv"                                  "df.seoul.worker.csv"

> df<-read.csv("df.seoul.worker.csv") # 데이터 불러오기
> str(df) # 데이터 구조 확인
'data.frame':   26107 obs. of  51 variables:
 $ X                  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ hh_id              : int  529 530 531 531 531 532 532 533 534 534
 $ hhm.id             : int  1 1 2 3 1 2 1 1 1 2 ...
 $ com.mode           : int  2 2 2 2 2 2 2 1 1 2 ...
 $ com.2modes         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ commuting.time     : int  30 20 20 20 20 40 30 20 20 30 ...
 $ commuting.area     : int  3 2 2 2 2 3 3 2 1 3 ...
 $ age                : int  71 69 39 35 67 60 65 56 50 54 ...
 $ gender             : int  1 1 1 2 2 2 1 1 1 2 ...
 $ edu.level          : int  2 2 3 3 2 1 2 2 2 2 ...
 $ job.type           : int  2 2 2 2 2 3 2 2 2 2 ...
 $ job.position       : int  4 4 3 5 1 1 1 3 1 1 ...
 $ income.level       : int  6 5 6 6 6 6 6 6 6 6 ...
 $ marriage.status    : int  1 1 1 1 5 1 1 3 1 1 ...
 $ voluntary.working  : int  2 2 2 2 2 2 2 2 2 2 ...
 $ hhm.no             : int  2 2 5 5 5 3 3 1 2 2 ...
 $ hh.income          : int  11 9 12 12 12 15 15 11 11 11 ...
 $ h.type             : int  2 2 2 2 2 2 2 2 2 2 ...
 $ h.own              : int  1 2 3 3 3 2 2 3 3 3 ...
 $ area.living        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ s.reside.year      : int  50 50 25 34 25 38 45 20 10 10 ...
 $ h.reside.year      : int  5 3 3 3 3 3 3 2 3 3 ...
```

실습 1: 자원봉사활동 참여 여부(e1071 패키지 활용)



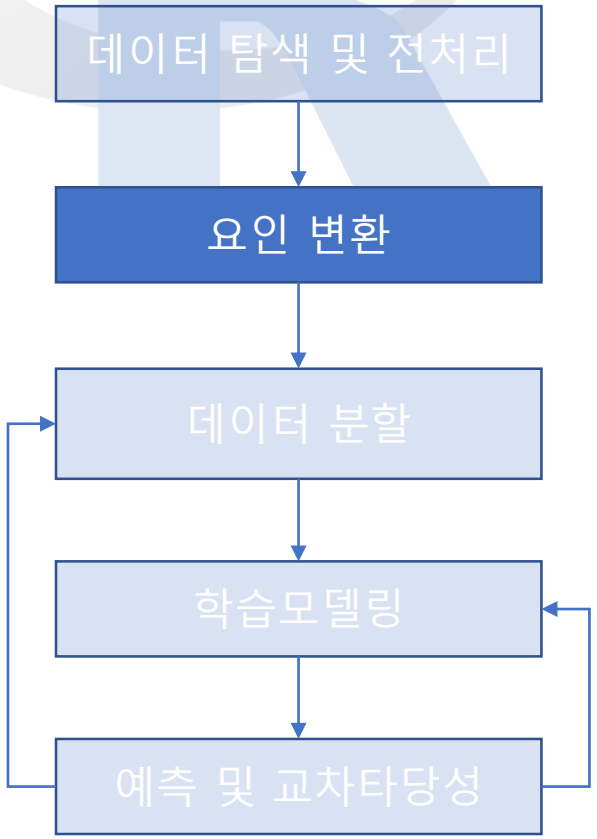

```
> attach(df) # df 객체 바로 접근하기 색인
> table(voluntary.working) # 지난 1년간 자원봉사 활동 참여: 1 = yes, 2 = no
voluntary.working
    1     2
 2509 23598
> table(com.2modes) # 통근수단: 1 = 승용차, 0 = 기타
com.2modes
    0     1
19021  7086
> table(gender) # 성별: 1=남성, 2=여성
gender
    1     2
15943 10164
> table(edu.level) # 교육수준: 1=중학교 이하, 2=고졸 이하, 3=대졸이하, 4=대학원 이상
edu.level
    1     2     3     4
 1075  8379 16259   394
> table(job.type) # 직업 유형: 1=관리전문직, 2=사무직, 3=생산판매직, 4=기타
job.type
    1     2     3     4
 2481 14303  9268    55
> table(job.position) # 고용형태: 1=상용근로, 2=임시일용근로, 3=고용없는자영업, 4=고용원 있는 자영업, 5=무급가족종사자
job.position
    1     2     3     4     5
18534  2817  2987  1463   304
> table(income.level) # 개인소득: 1 ' 100만원 미만', 2 '100-200만원 미만', 3 '200-300만원 미만', 4 '300-400만원 미만', 5 '400-500만원 미만', 6
00만원 이상'
income.level
    1     2     3     4     5     6
  121   919  2589  4512  5151 12815
> table(hhm.no) # 가구원수: 1인, 2인,....
hhm.no
   1    2    3    4    5    6    7    8
2482 6854 9105 6619  911  122    6    8
> table(marriage.status) # 혼인상태: 1=기혼, 2=미혼, 3=이혼/별거, 5=사별, 6=등거
marriage.status
    1     2     3     5     6
18550  5238  1357   952    10
```

실습 1: 자원봉사활동 참여 여부(e1071 패키지 활용)



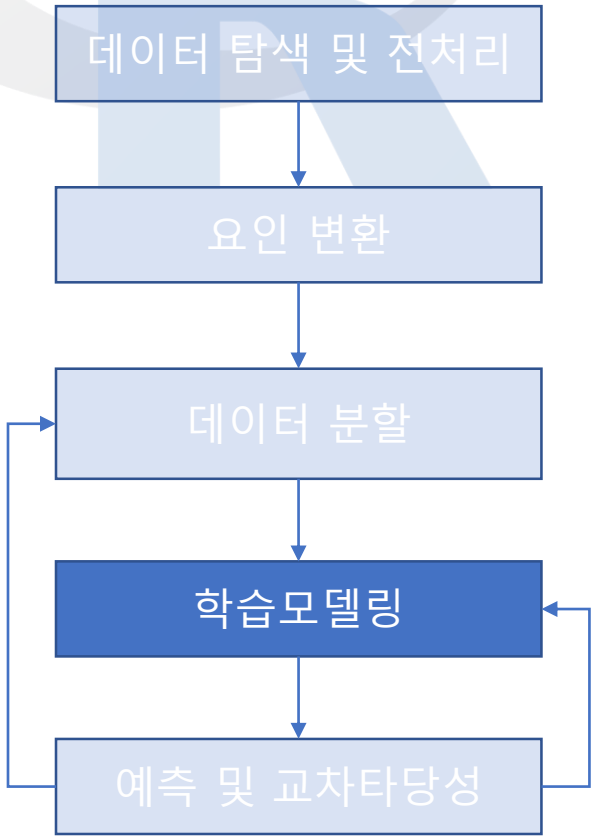

```
> detach() # 바로가기 색인 해제
> df <- df %>%
+   mutate(  # mutate() 함수로 숫자 변수를 요인으로 변환
+     voluntary.working = factor(voluntary.working), # 자원봉사 활동 참여 여부
+     gender = factor(gender), # 성별
+     com.2modes = factor(com.2modes), # 통근수단
+     edu.level = factor(edu.level), # 교육수준
+     job.type = factor(job.type), # 직업유형
+     job.position = factor(job.position), # 고용형태
+     income.level = factor(income.level), # 개인소득
+     hhm.no = factor(hhm.no), # 가구원수
+     marriage.status = factor(marriage.status) # 혼인상태
+   )
> df2 <- df %>% # select() 함수로 필요한 변수들만 추출하여 df2에 할당
+   select(voluntary.working, # 자원보상활동 변수
+          com.2modes:hh.income, # com.2modes 부터 hh.income까지 변수들
+          area.living) # 주거지역 대생활권
> str(df2)
'data.frame':   26107 obs. of  14 variables:
 $ voluntary.working: Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 2 ...
 $ com.2modes       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ commuting.time   : int  30 20 20 20 20 40 30 20 20 30 ...
 $ commuting.area   : int  3 2 2 2 2 3 3 2 1 3 ...
 $ age              : int  71 69 39 35 67 60 65 56 50 54 ...
 $ gender           : Factor w/ 2 levels "1","2": 1 1 1 2 2 2 1 1 1 2 ...
 $ edu.level        : Factor w/ 4 levels "1","2","3","4": 2 2 3 3 2 1 2 2 2 2 ...
 $ job.type         : Factor w/ 4 levels "1","2","3","4": 2 2 2 2 2 3 2 2 2 2 ...
 $ job.position     : Factor w/ 5 levels "1","2","3","4",..: 4 4 3 5 1 1 1 3 1 1 ...
 $ income.level     : Factor w/ 6 levels "1","2","3","4",..: 6 5 6 6 6 6 6 6 6 6 ...
 $ marriage.status  : Factor w/ 5 levels "1","2","3","5",..: 1 1 1 1 4 1 1 3 1 1 ...
 $ hhm.no           : Factor w/ 8 levels "1","2","3","4",..: 2 2 5 5 5 3 3 1 2 2 ...
 $ hh.income        : int  11 9 12 12 12 15 15 11 11 11 ...
 $ area.living      : int  1 1 1 1 1 1 1 1 1 1 ...
```

실습 1: 자원봉사활동 참여 여부(e1071 패키지 활용)



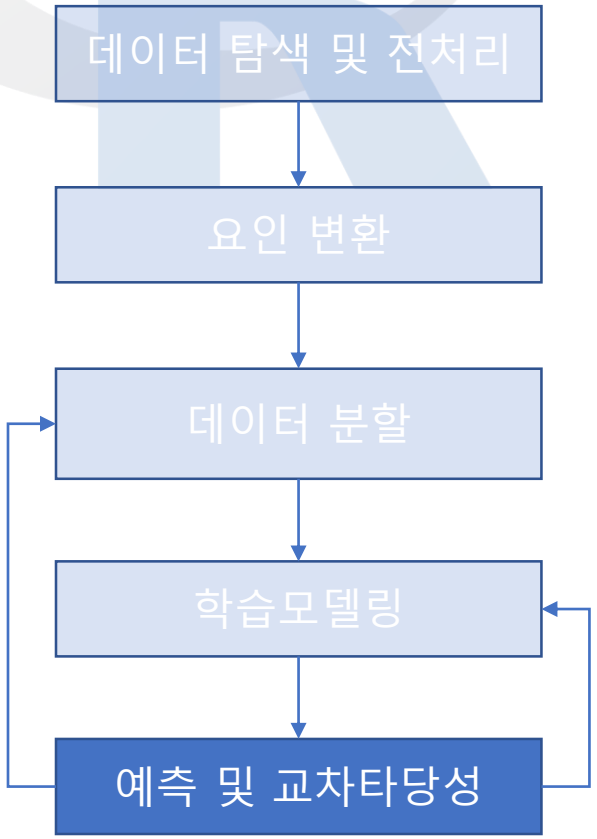

```
> detach() # 바로가기 색인 해제
> df <- df %>%
+   mutate(  # mutate() 함수로 숫자 변수를 요인으로 변환
+     voluntary.working = factor(voluntary.working), # 자원봉사 활동 참여 여부
+     gender = factor(gender), # 성별
+     com.2modes = factor(com.2modes), # 통근수단
+     edu.level = factor(edu.level), # 교육수준
+     job.type = factor(job.type), # 직업유형
+     job.position = factor(job.position), # 고용형태
+     income.level = factor(income.level), # 개인소득
+     hhm.no = factor(hhm.no), # 가구원수
+     marriage.status = factor(marriage.status) # 혼인상태
+   )
> df2 <- df %>% # select() 함수로 필요한 변수들만 추출하여 df2에 할당
+   select(voluntary.working, # 자원보상활동 변수
+          com.2modes:hh.income, # com.2modes 부터 hh.income까지 변수들
+          area.living) # 주거지역 대생활권
> str(df2)
'data.frame':   26107 obs. of  14 variables:
 $ voluntary.working: Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 2 ...
 $ com.2modes       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ commuting.time   : int  30 20 20 20 20 40 30 20 20 30 ...
 $ commuting.area   : int  3 2 2 2 2 3 3 2 1 3 ...
 $ age              : int  71 69 39 35 67 60 65 56 50 54 ...
 $ gender           : Factor w/ 2 levels "1","2": 1 1 1 2 2 2 1 1 1 2 ...
 $ edu.level        : Factor w/ 4 levels "1","2","3","4": 2 2 3 3 2 1 2 2 2 2 ...
 $ job.type         : Factor w/ 4 levels "1","2","3","4": 2 2 2 2 2 3 2 2 2 2 ...
 $ job.position     : Factor w/ 5 levels "1","2","3","4",..: 4 4 3 5 1 1 1 3 1 1 ...
 $ income.level     : Factor w/ 6 levels "1","2","3","4",..: 6 5 6 6 6 6 6 6 6 6 ...
 $ marriage.status  : Factor w/ 5 levels "1","2","3","5",..: 1 1 1 1 4 1 1 3 1 1 ...
 $ hhm.no           : Factor w/ 8 levels "1","2","3","4",..: 2 2 5 5 5 3 3 1 2 2 ...
 $ hh.income        : int  11 9 12 12 12 15 15 11 11 11 ...
 $ area.living      : int  1 1 1 1 1 1 1 1 1 1 ...
```

실습 1: 자원봉사활동 참여 여부(e1071 패키지 활용)



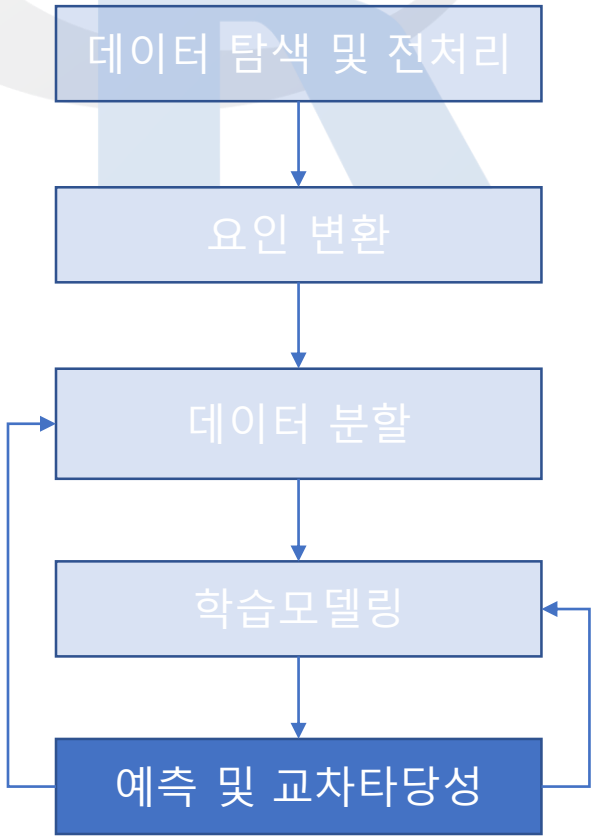

```
> ### 3. 데이터 분할
> set.seed(1234) # 난수 생성 규칙 설정(재현성 확보)
> ?createDataPartition() # 데이터 분할 함수(Data Splitting functions)
> yes.train <- createDataPartition(y=df2$voluntary.working, # 자원봉사활동 변수 기준
+                                  p=0.7, # 0.7의 비율로 무작의 추출
+                                  list=FALSE) # 리스타 아닌 행렬로
> class(yes.train) # 해당 객체 속성 확인
[1] "matrix"
> train <- df2[yes.train, ] # df2를 intrain에 해당하는 행들만 추출
> test <- df2[-yes.train, ] # df2를 intrain에 해당하지 않는 행들만 추출

> ### 4. 훈련데이터 학습모델링
> train <- df2[yes.train, ] # df2를 intrain에 해당하는 행들만 추출
> test <- df2[-yes.train, ] # df2를 intrain에 해당하지 않는 행들만 추출
> ?naiveBayes() # 나이브 베이즈 분류기(Naive Bayes Classifier) 함수: laplace=0 기본 값임
> m.nb <- naiveBayes(voluntary.working ~ # 종속변수
+                              ., # 이외 나머지 변수들은 특징(독립) 변수
+                            data = train) # 훈련데이터 셋으로 학습모델링
> m.nb # 실행결과 확인

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
         1          2
0.09613701 0.90386299

Conditional probabilities:
   com.2modes
Y           0         1
  1 0.7313603 0.2686397
  2 0.7286761 0.2713239
```

실습 1: 자원봉사활동 참여 여부(e1071 패키지 활용)

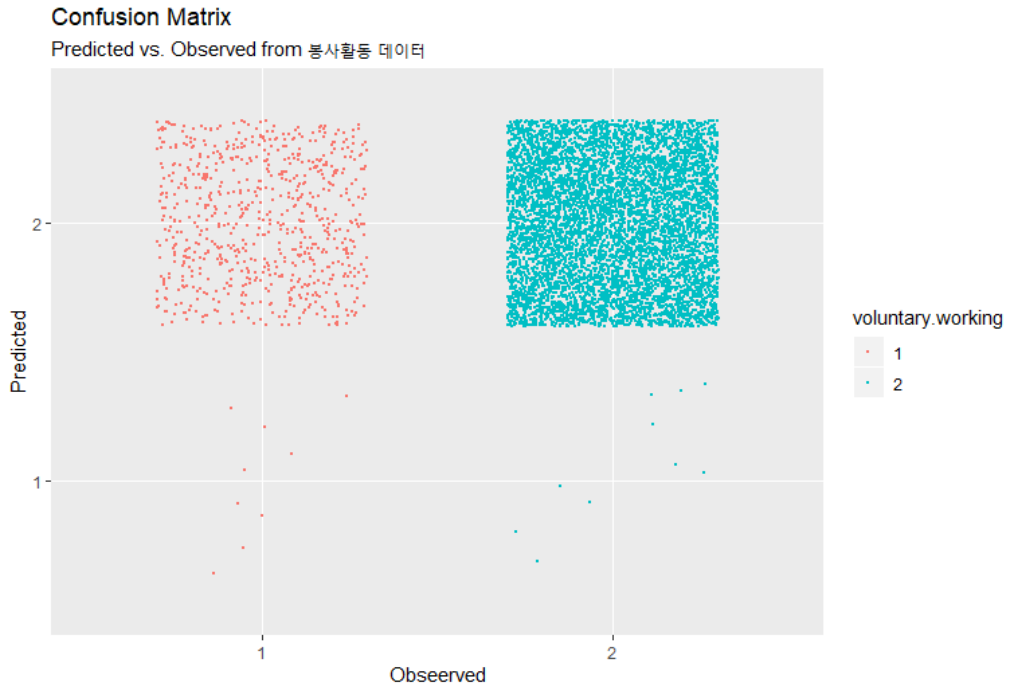



```
> ### 5. 예측 및 교차타당성
> nbpred <- predict(m.nb, # 훈련 학습모델 기반 예측
+                   test, # test 데이터셋 활용
+                   type='class') # 예측값을 확률이 아닌 분류로
> ?confusionMatrix() # 혼동행렬 생성 함수
> cm.01 <- confusionMatrix(nbpred, # nbpred 실행결과
+                          test$voluntary.working) # test 데이터셋의 voluntary.working
> cm.01 # 혼동행렬 실행결과 확인
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1    9   10
         2  743 7069

               Accuracy : 0.9038
                 95% CI : (0.8971, 0.9103)
    No Information Rate : 0.904
    P-Value [Acc > NIR] : 0.525

                  Kappa : 0.0187

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.011968
            Specificity : 0.998587
         Pos Pred Value : 0.473684
         Neg Pred Value : 0.904890
             Prevalence : 0.096029
         Detection Rate : 0.001149
   Detection Prevalence : 0.002426
      Balanced Accuracy : 0.505278

       'Positive' Class : 1
```

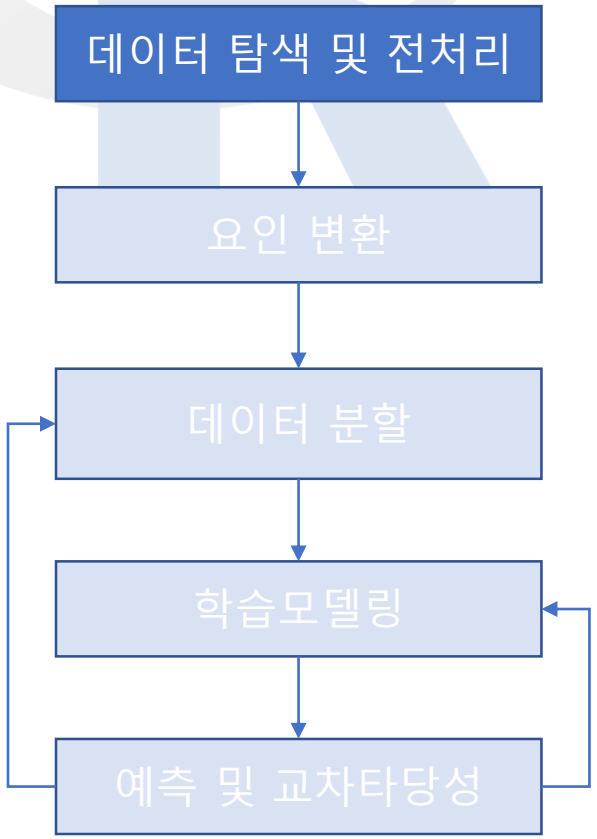
실습 1: 자원봉사활동 참여 여부(e1071 패키지 활용)




```
> ggplot(test, # 정답과 오답 분류표 그래프화
+        aes(voluntary.working, # 미학요인: x 축
+            nbpred, # y축
+            color = voluntary.working)) + # 색상
+   geom_jitter(width = 0.3, # jitter 그래프, 폭은 0.3배
+               height = 0.4, # 높이는 0.4배
+               size = 0.2) + # 점의 크기는 0.2배
+   labs(title = "Confusion Matrix", # 제목
+          subtitle = "Predicted vs. Observed from 봉사활동 데이터", # 하위 제목
+          y = "Predicted", # y축: 1=yes, 2=no
+          x = "Obseerved") # x 축: 1=yes, 2=no
+
```

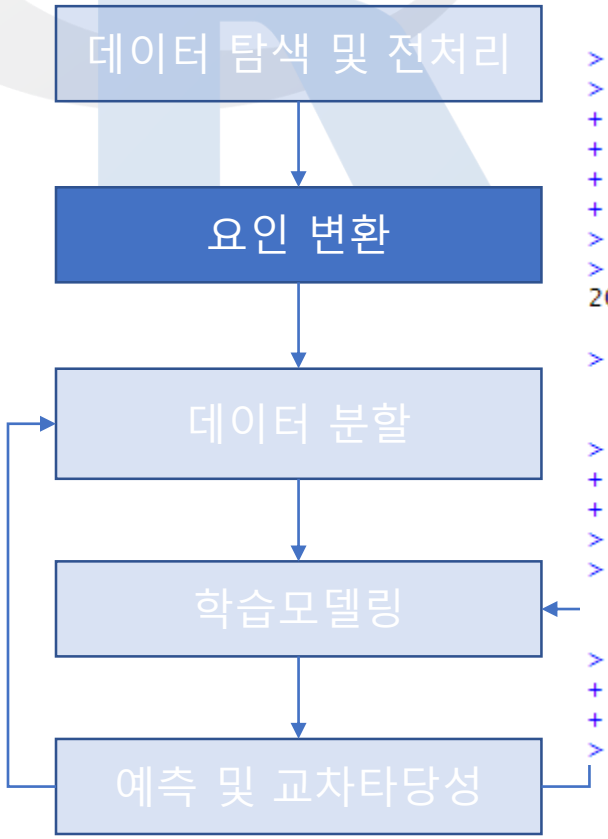


실습 2: 연속변수를 모두 요인으로 변환하여 모델링 (e1071 caret 패키지 둘 다 사용)



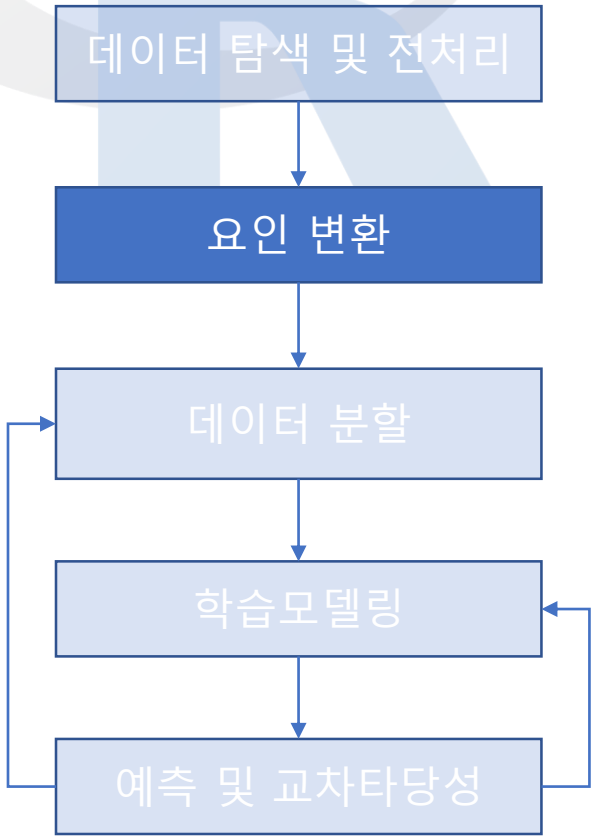

```
> ## 실습 2: 연속변수를 모두 요인으로 변환하여 모델링(e1071 caret 패키지 둘 다 사용)
> ### 1.데이터 탐색
> str(df2)
'data.frame':   26107 obs. of  17 variables:
 $ voluntary.working: Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 2 ...
 $ com.2modes       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ commuting.time   : int  30 20 20 20 20 40 30 20 20 30 ...
 $ commuting.area   : int  3 2 2 2 2 3 3 2 1 3 ...
 $ age              : int  71 69 39 35 67 60 65 56 50 54 ...
 $ gender           : Factor w/ 2 levels "1","2": 1 1 1 2 2 2 1 1 1 2 ...
 $ edu.level        : Factor w/ 4 levels "1","2","3","4": 2 2 3 3 2 1 2 2 2 2 ...
 $ job.type         : Factor w/ 4 levels "1","2","3","4": 2 2 2 2 2 3 2 2 2 2 ...
 $ job.position     : Factor w/ 5 levels "1","2","3","4",..: 4 4 3 5 1 1 1 3 1 1 ...
 $ income.level     : Factor w/ 6 levels "1","2","3","4",..: 6 5 6 6 6 6 6 6 6 6 ...
 $ marriage.status  : Factor w/ 5 levels "1","2","3","5",..: 1 1 1 1 4 1 1 3 1 1 ...
 $ hhm.no           : Factor w/ 8 levels "1","2","3","4",..: 2 2 5 5 5 3 3 1 2 2 ...
 $ hh.income        : int  11 9 12 12 12 15 15 11 11 11 ...
 $ area.living      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ age_group        : Factor w/ 5 levels "20s_less","30s",..: 5 5 2 2 5 5 5 4 4 4 ...
 $ com.time2        : Factor w/ 3 levels "less","median",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ hh.income2       : Factor w/ 3 levels "high","low","median": 3 3 3 3 3 1 1 3 3 3 ...
```

실습 2: 연속변수를 모두 요인으로 변환하여 모델링 (e1071 caret 패키지 둘 다 사용)



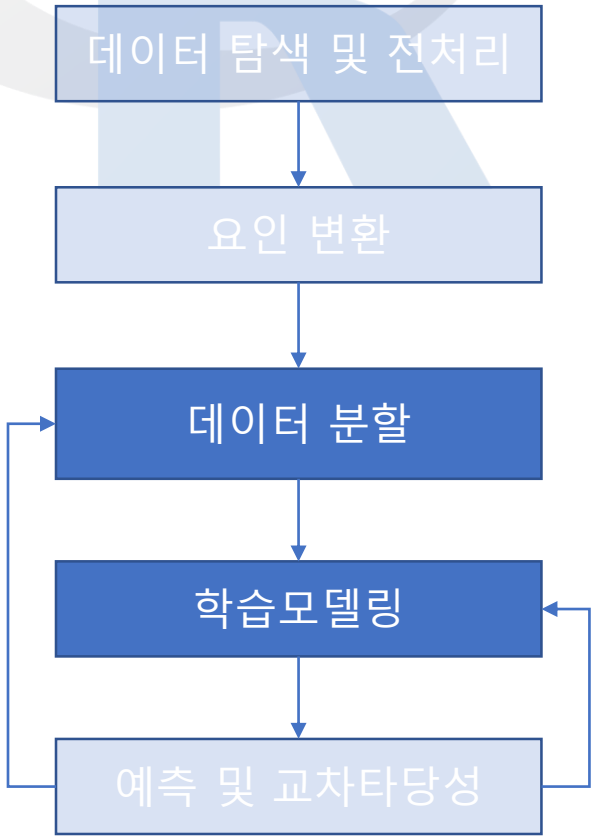

```
> ### 2. 요인 변환
> df2 <- df2 %>%
+   mutate(age_group = ifelse(age <=29, "20s_less", # age_group: 20대 또는 이하
+                             ifelse(age>=30 & age < 40, "30s", # 30대
+                                    ifelse(age>=40 & age < 50, "40s", # 40대
+                                           ifelse(age>=50 & age < 60, "50s", "60s_over"))))) # 50대, 60대 이상
> df2$age_group <- as.factor(df2$age_group) # 요인으로 변환
> summary(df2$age_group)
20s_less      30s      40s      50s 60s_over
    1989     6716     6334     5964     5104
> quantile(df2$commuting.time) # 통근시간 분위 통계량 확인
  0%  25%  50%  75% 100%
   1   20   30   40  240
> df2 <- df2 %>%  mutate(com.time2 =
+                          ifelse(commuting.time <20, "less", # 20분 미만
+                                 ifelse(commuting.time > 40, "more", "median"))) # 40분 초과, 기타 중간
> df2$com.time2 <- as.factor(df2$com.time2) # 요인으로 변환
> quantile(df2$hh.income) # 총 가구소득
  0%  25%  50%  75% 100%
   1    8   10   13   19
> df2 <- df2 %>%
+   mutate(hh.income2 = ifelse(hh.income <8, "low", # 8구간 미만
+                              ifelse(hh.income >13, "high", "median"))) # 13구간 초과
> df2$hh.income2 <- as.factor(df2$hh.income2) # 요인으로 변환
```

실습 2: 연속변수를 모두 요인으로 변환하여 모델링 (e1071 caret 패키지 둘 다 사용)



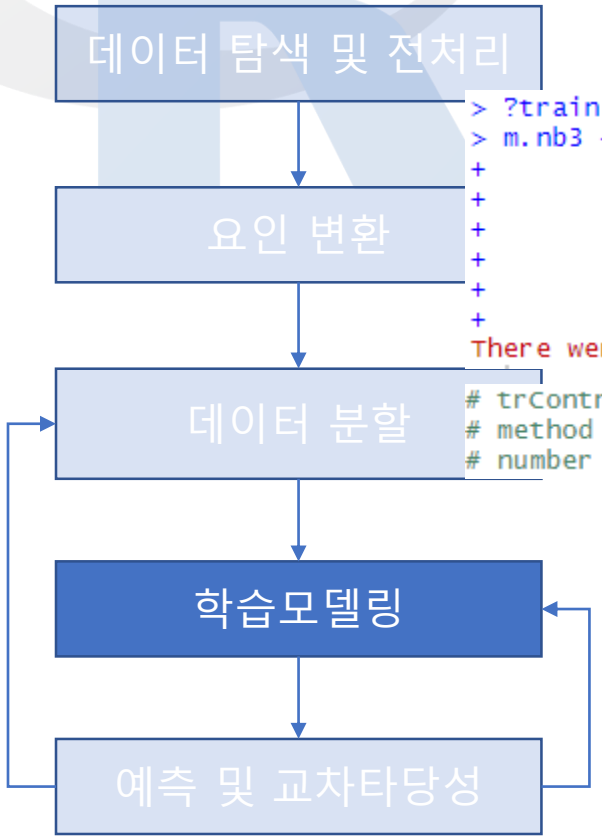

```
> df3 <- df2 %>%
+   select(-age, -commuting.time, -hh.income) # 3가지 연속변수 모두 제외하고 새 객체에 할당
> df3 <- df2 %>%
+   mutate(area.living = factor(area.living), # mutate()함수로 요인 변환
+          commuting.area = factor(commuting.area))
> df3 <- df2 %>%
+   select(-age, -commuting.time, -hh.income) # 3가지 연속변수 모두 제외하고 새 객체에 할당
> str(df3) # 데이터 구조로 모든 변수 요인 확인
'data.frame':   26107 obs. of  14 variables:
 $ voluntary.working: Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 2 ...
 $ com.2modes       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ commuting.area   : int  3 2 2 2 2 3 3 2 1 3 ...
 $ gender           : Factor w/ 2 levels "1","2": 1 1 1 2 2 2 1 1 1 2 ...
 $ edu.level        : Factor w/ 4 levels "1","2","3","4": 2 2 3 3 2 1 2 2 2 2 ...
 $ job.type         : Factor w/ 4 levels "1","2","3","4": 2 2 2 2 2 3 2 2 2 2 ...
 $ job.position     : Factor w/ 5 levels "1","2","3","4",..: 4 4 3 5 1 1 1 3 1 1 ...
 $ income.level     : Factor w/ 6 levels "1","2","3","4",..: 6 5 6 6 6 6 6 6 6 6 ...
 $ marriage.status  : Factor w/ 5 levels "1","2","3","5",..: 1 1 1 1 4 1 1 3 1 1 ...
 $ hhm.no           : Factor w/ 8 levels "1","2","3","4",..: 2 2 5 5 5 3 3 1 2 2 ...
 $ area.living      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ age_group        : Factor w/ 5 levels "20s_less","30s",..: 5 5 2 2 5 5 5 4 4 4 ...
 $ com.time2        : Factor w/ 3 levels "less","median",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ hh.income2       : Factor w/ 3 levels "high","low","median": 3 3 3 3 3 1 1 3 3 3 ...
```

실습 2: 연속변수를 모두 요인으로 변환하여 모델링 (e1071 caret 패키지 둘 다 사용)




```
> ### 3. 데이터 분할
> set.seed(1234) # 난수생성 규칙 설정(재현성)
> yes.train2 <- createDataPartition(y=df3$voluntary.working, # 분할 기준 변수
+                                   p=0.7, # 확률 0.7
+                                   list=FALSE) # 리스트가 아닌 행렬로
> train2 <-df3[yes.train2, ] # 훈련 데이터셋
> test2 <-df3[-yes.train2, ] # 시험 데이터셋
> ### 4. 나이브 베이즈 학습모델링
> m.nb2 <- naiveBayes(voluntary.working ~ # 예측 변수
+                       ., # 나머지는 모두 특징(설명)변수
+                     data = train2) # 사용할 데이터 객체
> m.nb2

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
         1          2
0.09613701 0.90386299

Conditional probabilities:
   com.2modes
Y           0         1
  1 0.7313603 0.2686397
  2 0.7286761 0.2713239

   commuting.area
Y       [,1]      [,2]
  1 2.276608 0.8876425
  2 2.378473 0.8434769

   gender
Y           1         2
  1 0.5167900 0.4832100
  2 0.6193474 0.3806526
```

실습 2: 연속변수를 모두 요인으로 변환하여 모델링 (e1071 caret 패키지 둘 다 사용)



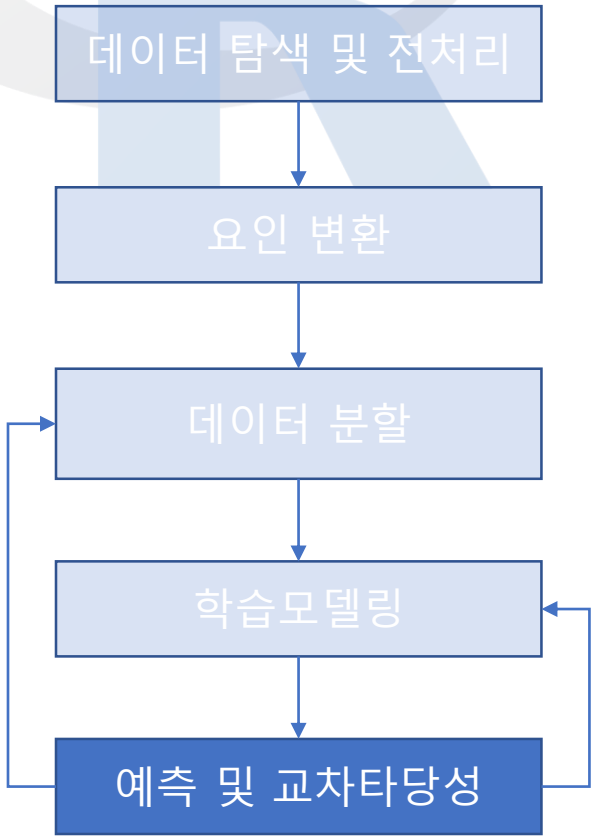

```
> ?train() # caret 패키지: Fit Predictive Models over Different Tuning Parameters
> m.nb3 <- train(train2[ , -1], # 첫열 제외 모든 열: 설명변수
+                train2[ , 1], # 첫열만: 종속변수
+                   method ='nb', # Naive Bayes: 사용가능 모델링(http://topepo.github.io/caret/train-models-by-tag.html)
+                   trControl=trainControl(method = "repeatedcv",
+                                          number = 5, # # 5-fold 데이터 분할
+                                          repeats = 5), # cross validation
+                   data = train2) # 사용할 데이터 객체: 전체 데이터
There were 50 or more warnings (use warnings() to see the first 50)

# trControl() # Control parameters for train
# method  = The resampling method: "boot", "boot632", "optimism_boot", "boot_all", "cv", "repeatedcv", "LOOCV", "LGOCV"
# number  = Either the number of folds or number of resampling iterations
```

실습 2: 연속변수를 모두 요인으로 변환하여 모델링 (e1071 caret 패키지 둘 다 사용)

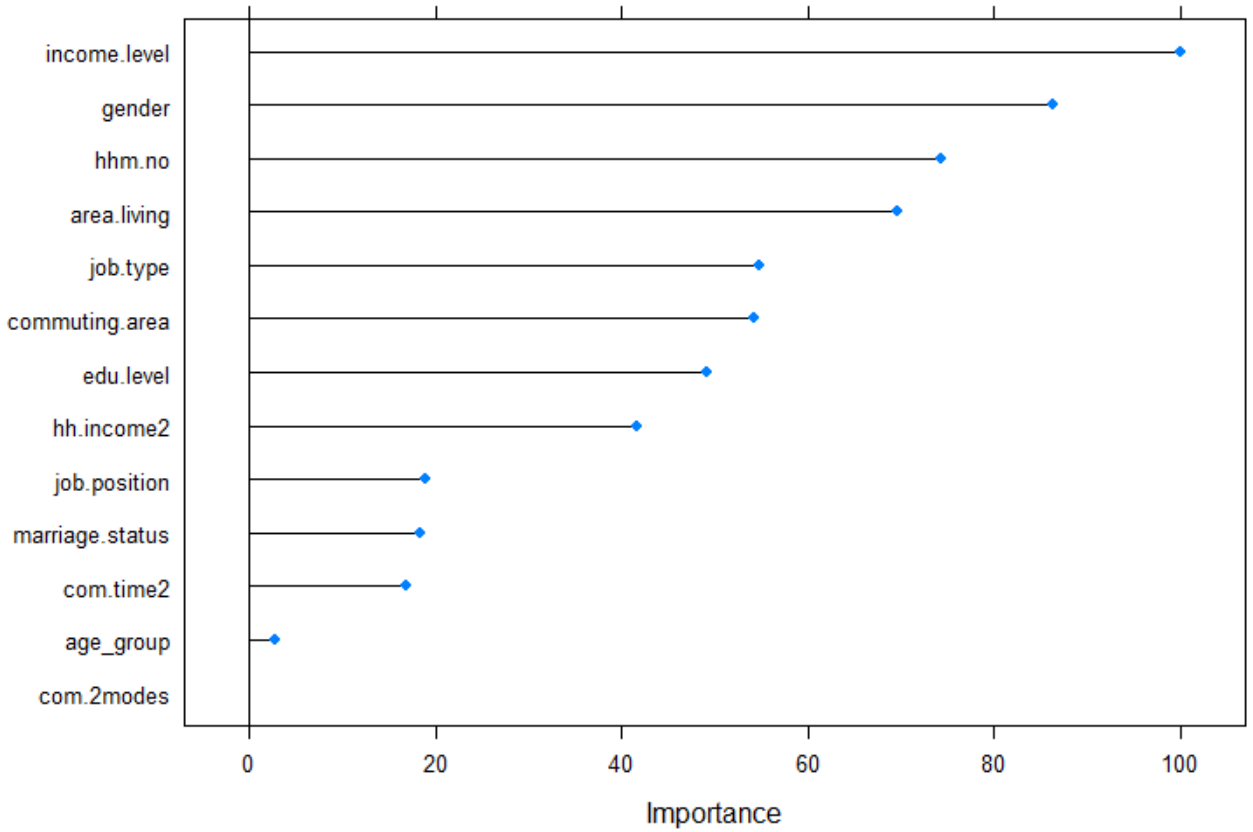
```
> ### 5. 예측 및 교차 타당성
> nbpred2 <- predict(m.nb2, # Predict testing set
+                    newdata = test2, # 시험 데이터셋 활용
+                    type = 'class') # 수준으로 예측
> cm.02 <- confusionMatrix(nbpred2, # e1071 패키지 활용 훈련 학습모델 결과 예측치
+                          test2[ , 1]) # 시험 데이터셋 실측치
> cm.02 # 혼동행렬 결과 보기
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1    8    5
         2  744 7074

               Accuracy : 0.9044
                 95% CI : (0.8976, 0.9108)
    No Information Rate : 0.904
    P-Value [Acc > NIR] : 0.4638

                  Kappa : 0.0177

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.010638
            Specificity : 0.999294
         Pos Pred Value : 0.615385
         Neg Pred Value : 0.904835
             Prevalence : 0.096029
         Detection Rate : 0.001022
   Detection Prevalence : 0.001660
      Balanced Accuracy : 0.504966

       'Positive' Class : 1
```

```
> nbpred3 <- predict(m.nb3, # caret 패키지 훈련 학습모델로 예측
+                    newdata = test2) # 시험 데이터셋
There were 50 or more warnings (use warnings() to see the first 50)
> cm.03 <- confusionMatrix(nbpred3, test2$voluntary.working)
> cm.03
Confusion Matrix and Statistics

          Reference
Prediction    1    2
         1   11    8
         2  741 7071

               Accuracy : 0.9044
                 95% CI : (0.8976, 0.9108)
    No Information Rate : 0.904
    P-Value [Acc > NIR] : 0.4638

                  Kappa : 0.0239

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.014628
            Specificity : 0.998870
         Pos Pred Value : 0.578947
         Neg Pred Value : 0.905146
             Prevalence : 0.096029
         Detection Rate : 0.001405
   Detection Prevalence : 0.002426
      Balanced Accuracy : 0.506749

       'Positive' Class : 1
```

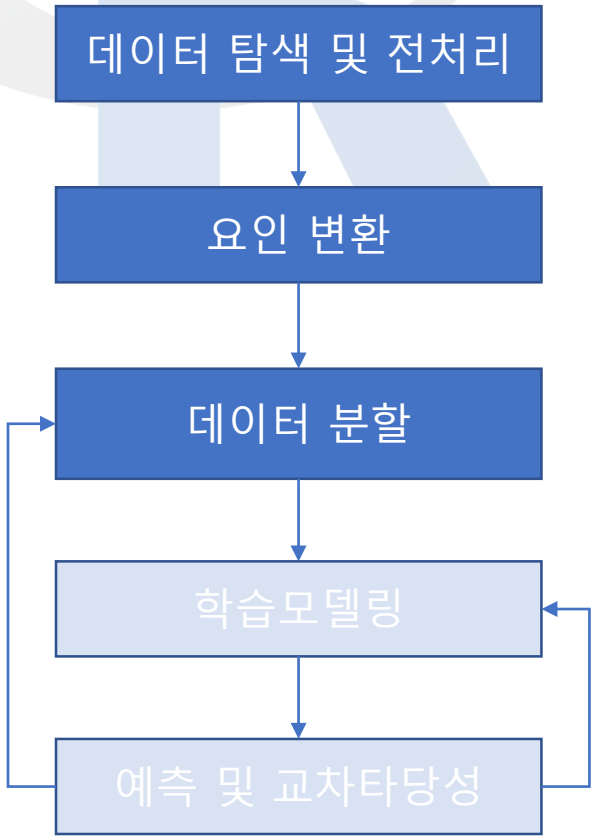
실습 2: 연속변수를 모두 요인으로 변환하여 모델링 (e1071 caret 패키지 둘 다 사용)




```
> #Plot variable performance
> ?varImp() # Calculation of variable importance for regression and classification models
> v.i <- varImp(m.nb3) # caret 패키지: "e1071' 패키지의 함수 결과로는 작동하지 않음
> v.i
ROC curve variable importance

                Importance
income.level       100.000
gender              86.250
hhm.no              74.379
area.living         69.563
job.type            54.788
commuting.area      54.165
edu.level           49.133
hh.income2          41.592
job.position        18.922
marriage.status     18.320
com.time2           16.931
age_group            2.919
com.2modes           0.000
> plot(v.i)
```

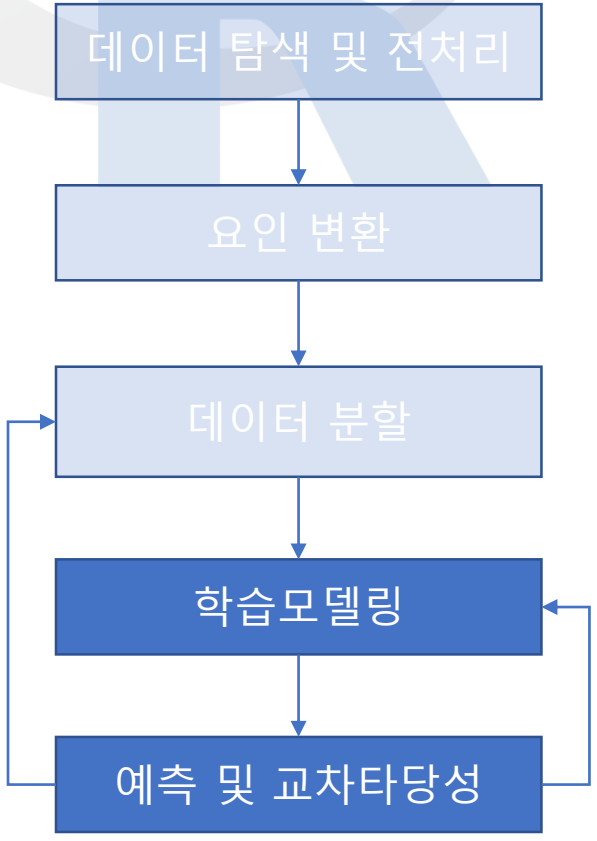


실습3: marriage.status(LaPlase 추정통계량 적용)



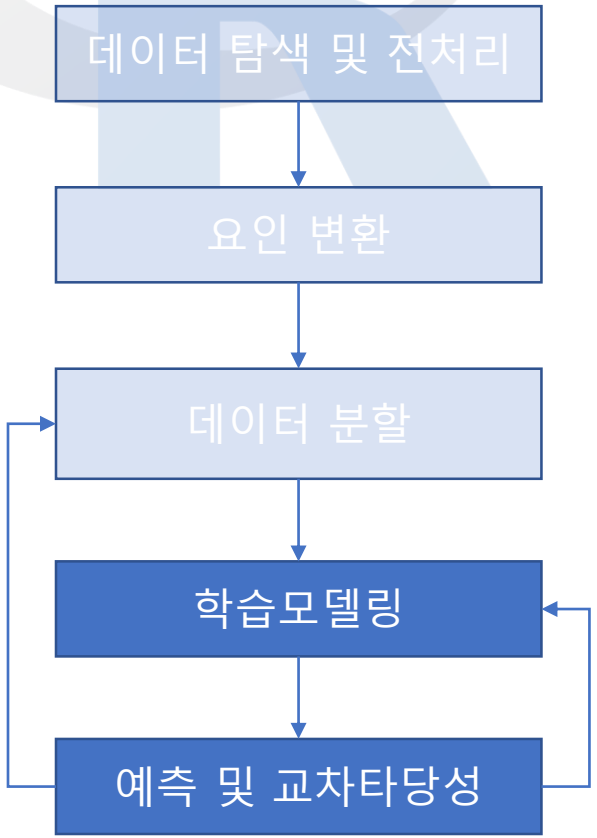

```
> # 실습3: marriage.status(LaPlase 추정통계량 적용)
> ### 1. 데이터 탐색 및 요인 변환
> table(df3$marriage.status) # 혼인상태: 1=기혼, 2=미혼, 3=이혼/별거, 5=사별, 6=등거

    1     2     3     5     6
18550  5238  1357   952    10
> ?recode() # dlpyr:: Recoding factors using recode
> df3$marriage.status <- recode(df3$marriage.status, "6" = "2") # 동거를 미혼과 합침
> table(df3$marriage.status) # 혼인상태: 1=기혼, 2=비혼, 3=이혼/별거, 5=사별

    1     2     3     5
18550  5248  1357   952
> ### 2. 데이터 분할
> set.seed(1234)
> yes.train4 <- createDataPartition(y=df3$marriage.status,
+                                   p=0.7,
+                                   list=FALSE)
> train4 <-df3[yes.train4, ] # 훈련용 데이터셋
> test4 <-df3[-yes.train4, ] # 시험용 데이터셋
```

실습3: marriage.status(LaPlase 추정통계량 적용)



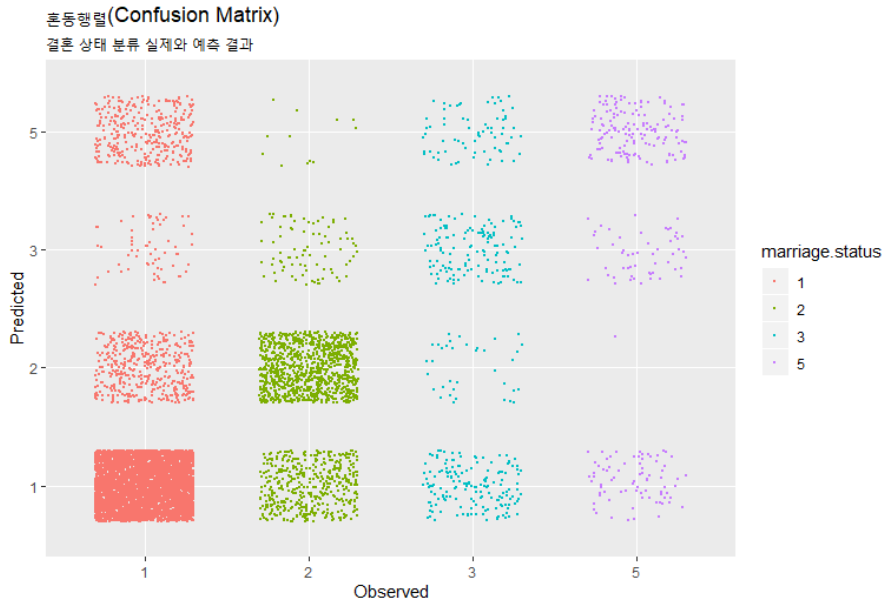

```
> ### 4. 예측 및 교차 타당성 검증
> nbpred4 <- predict(m.nb4, # 훈련 학습모델 기반 예측
+                    test4, # test 데이터셋 활용
+                    type='class') # 예측값을 확률이 아닌 분류로
> ?confusionMatrix() # 혼동행렬 생성 함수
> cm.04 <- confusionMatrix(nbpred4, # nbpred 실행결과
+                          test4$marriage.status) # test 데이터셋의 voluntary.working
> cm.04 # 혼동행렬 실행결과 확인
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    5
         1 4739  449  162   80
         2  424 1029   36    1
         3   67   83  137   54
         5  335   13   72  150

Overall Statistics

               Accuracy : 0.7732
                 95% CI : (0.7638, 0.7824)
    No Information Rate : 0.7106
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5113

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 5
Sensitivity            0.8516   0.6537  0.33661  0.52632
Specificity            0.6951   0.9263  0.97252  0.94434
Pos Pred Value         0.8727   0.6906  0.40176  0.26316
Neg Pred Value         0.6560   0.9141  0.96395  0.98141
Prevalence             0.7106   0.2010  0.05197  0.03639
Detection Rate         0.6052   0.1314  0.01749  0.01915
Detection Prevalence   0.6934   0.1903  0.04354  0.07279
Balanced Accuracy      0.7733   0.7900  0.65457  0.73533
```

실습3: marriage.status(LaPlace 추정량 계량 적용)




```
> ### 5. Lapace 추정량=0.1 모델링 및 예측
> m.nb4 <- naiveBayes(marriage.status ~ # 예측 변수
+                       ., # 모든 변수
+                       laplace = 0.1, # laplace = 0.1
+                       data = train4) # 사용할 데이터 객체
> nbpred4 <- predict(m.nb4,
+                    test4,
+                    type='class')
> confusionMatrix(nbpred4,
+                 test4$marriage.status)
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    5
         1 4740  449  162   80
         2  424 1030   36    1
         3   65   83  137   54
         5  336   12   72  150

Overall Statistics

               Accuracy : 0.7735
                 95% CI : (0.764, 0.7827)
    No Information Rate : 0.7106
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5118

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 5
Sensitivity            0.8518   0.6544  0.33661  0.52632
Specificity            0.6951   0.9263  0.97279  0.94434
Pos Pred Value         0.8728   0.6908  0.40413  0.26316
Neg Pred Value         0.6562   0.9142  0.96396  0.98141
Prevalence             0.7106   0.2010  0.05197  0.03639
Detection Rate         0.6053   0.1315  0.01749  0.01915
Detection Prevalence   0.6935   0.1904  0.04329  0.07279
Balanced Accuracy      0.7734   0.7904  0.65470  0.73533
```

```
> ggplot(test4, aes(marriage.status, nbpred4, color = marriage.status)) +
+   geom_jitter(width = 0.3, height = 0.3, size=0.3) +
+   labs(title="혼동행렬(Confusion Matrix)",
+        subtitle="결혼 상태 분류 실제와 예측 결과",
+        y="Predicted",
+        x="Observed") # 1=기혼, 2=비혼, 3=이혼/별거, 5=사별
```



연습문제 03

- 위의 결혼상태 훈련모델에서 laplace = 0.3으로 주고, laplace = 0.1과 0일 대의 결과와 비교하시오.

```
> m.nb6 <- naiveBayes(marriage.status ~ # 예측 변수
+                       ., # 모든 변수
+                     laplace = 0.3, # laplace = 0.3
+                     data = train4) # 사용할 데이터 객체
> nbpred6 <- predict(m.nb6,
+                    test4,
+                    type='class')
> cm.06 <- confusionMatrix(nbpred6,
+                          test4$marriage.status)
> cm.06 # laplace = 0.3
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    5
         1 4742  449  162   80
         2  424 1030   36    1
         3   65   82  137   54
         5  334   13   72  150

Overall Statistics

               Accuracy : 0.7737
                 95% CI : (0.7643, 0.7829)
    No Information Rate : 0.7106
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5121

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 5
Sensitivity            0.8521   0.6544  0.33661  0.52632
Specificity            0.6951   0.9263  0.97293  0.94447
Pos Pred Value         0.8728   0.6908  0.40533  0.26362
Neg Pred Value         0.6568   0.9142  0.96397  0.98141
Prevalence             0.7106   0.2010  0.05197  0.03639
Detection Rate         0.6055   0.1315  0.01749  0.01915
Detection Prevalence   0.6938   0.1904  0.04316  0.07266
Balanced Accuracy      0.7736   0.7904  0.65477  0.73539
```

요약

- 나이브 베이즈 모델 개요
  - 나이브 베이즈 모델
  - 베이즈 정리(Bayes' Theorem)
  - 라플라스 근사와 추정량
  - 나이브 베이즈 모델 장단점
  - 나이브 베이즈 모델 분석절차
- 실습 1: 지난 1년간 자원봉사활동 참여 여부(e1071 패키지 활용)
- 실습 2: 연속변수를 모두 요인으로 변환하여 모델링(e1071, caret 패키지 사용)
- 실습 3: 혼인상태(LaPlace 추정통계량 적용)

끝

- 질의와 토의(Question & Discussion)
  - 이번 강의 내용을 시청하고, 실행하면서 궁금한 점이나 어려운 점에 대하여 토의해봅시다.

- 다음 차 강의주제
  - **시계열 데이터 분석**