

2019-10-16

# computational Statistics

*HW#4*

192STG11 우나영

# computational Statistics

## HW#4

### 1. Problem

EM 알고리즘은 관측되지 않는 잠재변수(latent variable)에 의존하는 확률 모델에서 반복적인 computation 을 통해 MLE 를 구하는 방법이다. 즉 데이터  $X=x$  가 주어 졌을 때 대응되는 잠재변수  $Z$  를 이용해 매개변수인  $\theta$ 의 MLE 를 구한다. 특히 EM 은 incomplete 데이터의 MLE 를 계산하는 데 유용하다. 하지만 Incomplete 데이터의 경우 full loglikelihood 를 계산하는 것이 수리적으로 불가능하다. 따라서 잠재변수의 값을 추정하여 loglikelihood 를 구할 수밖에 없다. EM 은 잠재변수의 값을 추정하기 위해 아래와 같은 conditional expectation of loglikelihood 를 사용한다. 이를 loglikelihood 의 Expectation 을 구하는 E-step 이라 부른다.

$$Q(\theta, \theta_{old}) = E(l(\theta|Y)|x, \theta_{new})$$

$$\text{where } \begin{cases} X: \text{Incomplete data (observed)} \\ Y: \text{Missing data (unobserved)} \end{cases}$$

이렇게 예측된 잠재변수의 기대 값을 이용해 conditional expectation of loglikelihood 를 최대화하는 매개변수  $\theta$ 를 구한다. 이를 기대 값을 Maximization 시키는 변수를 구하는 M-step 이라 부른다. EM 은 이러한 E-step 과 M-step 을 반복적으로 수행하여 MLE 를 구한다. EM 알고리즘을 정리하면 아래와 같다.

|   |
|---|
| Initiate parameter $\theta_{old}$   |
| E – step : compute $Q(\theta, \theta_{old})$  |
| M – step: $\theta_{new} = \underset{\theta}{\operatorname{argmax}} Q(\theta, \theta_{old})$                 |
| Iterate E-step and M-step until $\theta$ converge which means $ \theta_{old} - \theta_{new}  < \varepsilon$ |

한편 clustering 방법론은 크게 underlying 분포의 존재를 가정하는 것과 분포 가정없이 objective function 을 최적화하는 방법으로 나뉜다. Gaussian Mixture Model(GMM)은 분포 가정을 하는 clustering 방법으로 데이터가 K 개의 정규분포로부터 생성되었다고 가정한다. GMM 에는 두 가지의 모수가 존재한다. 첫번째는 K 개의 클러스터의 크기를 나타내는 Weight 값이고 두 번째는 K 개의 정규분포 각각의 모수인 평균과 분산이다. GMM 에서 Weight 를 잠재변수로 설정하면 EM 을 통해 나머지 모수를 구할 수 있다. 이번 과제를 통해 가장 간단한 1 dimensional 2 cluster GMM 을 EM 을 통해 구현해 볼 것이다. 데이터가  $X_1, \dots, X_n \sim \pi\phi_1 + (1 - \pi)\phi_2$  를 따른다고 가정한다. 이때  $\pi$ 는 첫번째 클러스터의

weight,  $\phi_1$ 는  $N(\mu_1, \sigma_1^2)$ 의 확률 분포 함수, 그리고  $\phi_2$ 는  $N(\mu_2, \sigma_2^2)$ 의 확률 분포 함수이다. 잠재변수는  $Y_i = \begin{cases} 1 & X_i \text{ from } \phi_1 \\ 0 & X_i \text{ from } \phi_2 \end{cases}$  으로 cluster membership 에 대한 dummy variable 이다. 하지만 실제로 관측 불가능하기 때문에 EM 알고리즘을 이용하여  $Y_i$ 를 비롯한 나머지 매개변수인  $\theta = (\pi, \mu_1, \mu_2, \sigma_1, \sigma_2)$ 를 구해보자.

$K^{th}$  E - step

$$\begin{aligned} Y_i &= E(Y_i | \theta^{(k-1)}, X_i) = p(Y_i = 1 | \theta^{(k-1)}, X_i) \\ &= \frac{p(Y_i=1)p(X_i|Y_i=1, \theta^{(k-1)})}{p(Y_i=1)p(X_i|Y_i=1, \theta^{(k-1)}) + p(Y_i=0)p(X_i|Y_i=0, \theta^{(k-1)})} \\ &= \frac{\pi^{(k-1)}\phi_1(X_i; \mu_1^{(k-1)}, \sigma_1^{(k-1)})}{\pi^{(k-1)}\phi_1(X_i; \mu_1^{(k-1)}, \sigma_1^{(k-1)}) + (1 - \pi^{(k-1)})\phi_2(X_i; \mu_2^{(k-1)}, \sigma_2^{(k-1)})} \end{aligned}$$

$K^{th}$  M - step

$$\begin{aligned} \widehat{\pi^{(k)}} &= \frac{\sum Y_i^{(k)}}{n}, \quad \widehat{\mu_1^{(k)}} = \frac{\sum X_i Y_i^{(k)}}{\sum Y_i^{(k)}}, \quad \widehat{\mu_2^{(k)}} = \frac{\sum X_i (1 - Y_i^{(k)})}{\sum (1 - Y_i^{(k)})}, \\ \widehat{\sigma_1^{(k)}} &= \sqrt{\frac{\sum Y_i^{(k)} (X_i - \widehat{\mu_1^{(k)}})^2}{\sum Y_i^{(k)}}}, \quad \widehat{\sigma_2^{(k)}} = \sqrt{\frac{\sum (1 - Y_i^{(k)}) (X_i - \widehat{\mu_2^{(k)}})^2}{\sum (1 - Y_i^{(k)})}} \end{aligned}$$

Iterate until  $|l(\theta^{(k)}) - l(\theta^{(k-1)})| < \varepsilon$  where  $\varepsilon = 10^{-6}$

mygmm 은 위와 같은 식을 따라 작성되었고 Result 에서 다양한 예제를 통해 mygmm 의 수렴 양상을 살펴볼 수 있다.

또한 EM 알고리즘은 incomplete data 의 MLE 를 구하는 과정에서 unobserved data 를 추정하기 때문에 missing data 를 impute 할 수 있다는 장점이 있다. Result 에서 incomplete 한 이변량 정규분포의 imputation 을 실현해볼 것이다. 이에 앞서 이변량 정규 분포의 EM 알고리즘에 대해 알아보자. 우선 주어진 데이터는 다음과 같다.

|              |               |              |
|--------------|---------------|--------------|
| Obs: 1 ... m | m+1 .... m+m1 | m+m1+1 ... n |
| w1           | missing       | observed     |
| w2           | observed      | missing      |

$$W = (w_1, w_2) \sim N(\mu, \Sigma)$$

$$\text{where } \mu = (\mu_1, \mu_2), \quad \Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix}$$

$$\phi(w, \theta) = (2\pi)^{-1} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(w - \mu)^t \Sigma^{-1}(w - \mu)\right)$$

$$\theta = (\mu_1, \mu_2, \sigma_{11}, \sigma_{12}, \sigma_{22})$$

정규분포는 exponential family 중 하나로 충분 통계량(Sufficient Statistics)으로 간편하게 MLE 를 구할 수 있다. 하지만 missing data 에서는 결측치로 인해 충분 통계량 값을 구할 수 없다. 그러므로 EM 알고리즘에서 E-step 에서 conditional expectation of loglikelihood 를 이용해 결측치를 impute 한 후 M-step 에서 충분 통계량을 구해 MLE 를 구해  $\theta$  를 업데이트한다. 이 과정을 수렴할 때까지 반복하면 결측치를 impute 한 값과  $\theta$  의 MLE 값을 동시에 구할 수 있다. 알고리즘의 수식은 아래와 같다.

$$\text{Sufficient Statistics } T = (T_1, T_2, T_{11}, T_{22}, T_{12})$$

$$\text{where } T_i = \sum_{j=1}^n w_{ij}, i = 1, 2, T_{hi} = \sum_{j=1}^n w_{hj} w_{ij}, h, i = 1, 2$$

$K^{th}$  E - step

$$Q(\theta, \theta^{(k-1)}) = E_{\theta^{(k-1)}}[l_c(\theta) | Y]$$

$$\begin{cases} E_{\theta^{(k-1)}}[w_{1j} | w_{2j}] \\ E_{\theta^{(k-1)}}[w_{1j}^2 | w_{2j}] \end{cases} \quad j = m + 1, \dots, m + m_1$$

$$\begin{cases} E_{\theta^{(k-1)}}[w_{2j} | w_{1j}] \\ E_{\theta^{(k-1)}}[w_{2j}^2 | w_{1j}] \end{cases} \quad j = m + m_1 + 1, \dots, n$$

Calculate using conditional distribution of  $W_1$  given  $W_2 = w_2$

$$1) E[w_1 | w_2] = \mu_1 + \sigma_{12} \sigma_{22}^{-1} (w_2 - \mu_2)$$

$$2) Var[w_1 | w_2] = \sigma_{11} (1 - \rho^2)$$

$$\rightarrow E_{\theta^{(k-1)}}[w_{1j} | w_{2j}] = \mu_1^{(k-1)} + \sigma_{12}^{(k-1)} \sigma_{22}^{(k-1)-1} (w_{2j} - \mu_2^{(k-1)}) = w_{1j}^{(k-1)}$$

$$\rightarrow E_{\theta^{(k-1)}}[w_{1j}^2 | w_{2j}] = Var[w_1 | w_2] + [E[w_1 | w_2]]^2 = \sigma_{11}^{(k-1)} (1 - \rho^{(k-1)2}) + (w_{1j}^{(k-1)})^2$$

$K^{th}$  M - step

$$\widehat{\mu}_i^{MLE} = \frac{T_i}{n}, i = 1, 2, \quad \widehat{\sigma}_{hi}^{MLE} = \frac{T_{hi} - \frac{1}{n} T_h T_i}{n}, h, i = 1, 2, \quad \widehat{\sigma}_{ii}^{MLE} = \frac{T_{ii} - \frac{1}{n} T_i^2}{n}, i = 1, 2$$

다음으로는 Least Squares with missing response variable 데이터를 Healy-westmacott procedure 을 사용해 imputation 해 볼 것이다. 일반적으로 반응변수(response variable)가 결측치인 경우 해당 관측치를 제거하는 것이 보편적이지만 데이터에 따라서 모델링을 위해 반드시 imputation 이 필요한 경우가 있다. Healy-westmacott procedure 는 EM step 으로 결측치를 추정한다. 우선 초기값을 대입해 모델링 한다. E-step 에서 모델링으로 생성된  $\hat{y}$  로 결측치를 impute 한 후 M-step 에서 다시 모델링하여  $\hat{y}$  을 업데이트한다. 수렴할 때까지 E-step 과 M-step 을 반복한다. 본 과제에서는 UCI repository



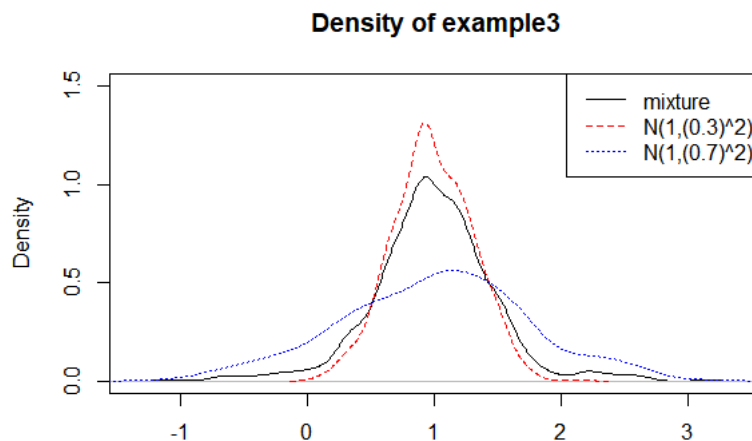


|                            |   |
|----------------------------|---|
| Cluster membership(n=100)  | [1] 11121112212111211222221121111211111122112211212<br>[50] 1111221121212111121221211222211221222222122221111   |
| Cluster membership(n=1000) | [1] 121222211211111211211121111212121111121112111211<br>[52] 11121122122121112222111222212212121112121212221<br>[103] 1222121211221211111211211211212122212111222112<br>[154] 212212112221111212221212212111222122122211212122<br>[205] 221222111211122221221212111212211212112122221221<br>[256] 1111211122211121111211222122112112221122122211111<br>[307] 2112211221122112211122111211212121221112121112211<br>[358] 2122212122211122122211112211212122121211221121<br>[409] 111212121212121112221221212221122121212112112111<br>[460] 12112111212212221112222122112112112112221112222<br>[511] 11211111122212112111112212112112111212121122221<br>[562] 11112221211211121212211122121122122112111122221<br>[613] 1112222121121221121121212211221121221112122211212<br>[664] 22221112211221212122121122211112112222221222122<br>[715] 21122222212122222222211212222212221122222222222<br>[766] 1222222222222222222222222222222212122121222222122222<br>[817] 22222122122222212221222222222222222222221221211212<br>[868] 222221222222222222222222222222222222222122211222112<br>[919] 2222211212222222222222212222222221222222212221121222<br>[970] 222221122222222222222222222221222221 |

두 번째 예제는  $N(1, (0.5)^2)$ 와  $N(1.5, (0.5)^2)$ 의 mixture를 clustering 한 것이다.  $N(1, (0.5)^2)$ 는 cluster 1 이고  $N(1.5, (0.5)^2)$ 는 cluster 2 로 정하였다.  $n$  이 커질수록 MLE 의 consistency 성질에 따라 추정치들이 모수에 근사 한다는 것을 알 수 있다. 하지만 추정된  $\pi$  값의 경우 오차가 크다. 이는 두 분포의 mean 이 가까워 공유하는 면적이 넓기 때문에 clustering 을 정확하게 하기 어렵다는 점을 시사한다.

Example 3)  $\mu_1$  과  $\mu_2$  이 동일한 두 분포의 mixture

example 3



| mixture 구조             | cluster  | 1               | 2               |
|------------------------|----------|-----------------|-----------------|
|                        | $\pi$    | 0.7             | 0.3             |
|                        | $dist^n$ | $N(1, (0.3)^2)$ | $N(1, (0.7)^2)$ |
| $n$                    | 100      | 1000            |                 |
| $\hat{\pi}_1^{mle}$    | 0.6686   | 0.2350          |                 |
| $\hat{\pi}_2^{mle}$    | 0.3314   | 0.7650          |                 |
| $\hat{\mu}_1^{mle}$    | 1.0031   | 0.9732          |                 |
| $\hat{\mu}_2^{mle}$    | 1.0914   | 1.0001          |                 |
| $\hat{\sigma}_1^{mle}$ | 0.4578   | 0.7944          |                 |
| $\hat{\sigma}_2^{mle}$ | 0.1821   | 0.3269          |                 |

$$\text{where } \mu = (\mu_1, \mu_2), \quad \Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix}$$





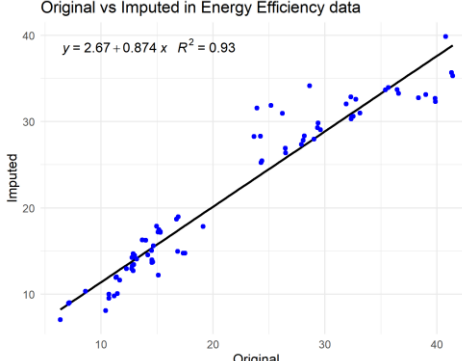
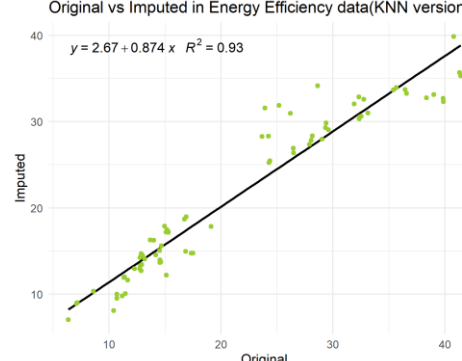
결측치를 impute 하고 M-step 에서 충분통계량을 이용하여 모수들의 MLE 를 구하는 과정을 반복하였다. 그 결과 모수의 MLE 값과 결측치는 위와 같다. Stopping rule 은 max iteration 1000 번,  $|l(\theta^{(k)}) - l(\theta^{(k-1)})| < \varepsilon$  where  $\varepsilon = 10^{-6}$ 을 사용하였다.

# impute using Healy-westmacott procedure

ex1. Energy efficiency

데이터 설명

|    |                      |            |                           |
|----|----------------------|------------|---------------------------|
| X1 | Relative Compactness | X5         | Overall Height            |
| X2 | Surface Area         | X6         | Orientation               |
| X3 | Wall Area            | X7         | Glazing Area              |
| X4 | Roof Area            | X8         | Glazing Area Distribution |
| Y  | Heating Load         | n=768, p=8 |                           |

|               |  |         |        |   |         |
|---------------|--|---------|--------|---|---------|
| initial value | $\bar{y}$  |         |        | knn imputation  |         |
| graph         | Original vs Imputed in Energy Efficiency data                                      |         |        | Original vs Imputed in Energy Efficiency data(KNN version)                          |         |
|               |  |         |        |  |         |
| iteration     | 6  |         |        | 4   |         |
| comparison    | initial  | imputed | true y | initial   | imputed |
| 1             | 22.39  | 33.67   | 35.4   | 35.03   | 33.67   |
| 2             |  | 10.08   | 11.45  | 11.20   | 10.08   |
| 3             |  | 12.98   | 12.27  | 12.34   | 12.98   |
| 4             |  | 14.69   | 12.86  | 12.96   | 14.69   |
| 5             |  | 29.04   | 29.60  | 28.96   | 29.04   |
| 6             |  | 10.02   | 10.70  | 10.68   | 10.02   |
| 7             |  | 12.21   | 15.12  | 15.33   | 12.21   |
| 8             |  | 32.31   | 39.86  | 39.42   | 32.31   |
| 9             |  | 39.83   | 40.78  | 39.58   | 39.83   |
| 10            |  | 12.02   | 11.38  | 11.52   | 12.02   |
| 11            |  | 31.86   | 25.17  | 25.59   | 31.86   |
| 12            |  | 14.75   | 17.35  | 16.98   | 14.75   |
| 13            |  | 34.14   | 28.64  | 29.51   | 34.14   |
| 14            |  | 33.93   | 35.67  | 34.48   | 33.93   |
| 15            |  | 9.02    | 7.18   | 10.98   | 9.02    |
| 16            |  | 26.37   | 26.48  | 26.36   | 26.37   |
| 17            |  | 32.86   | 32.31  | 31.47   | 32.86   |
| 18            |  | 30.99   | 33.13  | 30.73   | 30.99   |
| 19            |  | 33.72   | 36.45  | 34.59   | 33.72   |
| 20            |  | 15.63   | 14.66  | 14.61   | 15.63   |
| 21            |  | 13.98   | 14.54  | 14.14   | 13.98   |
| 22            |  | 14.09   | 13.18  | 12.99   | 14.09   |
| 23            |  | 17.88   | 14.96  | 15.11   | 17.88   |
| 24            |  | 13.74   | 14.61  | 14.47   | 13.74   |
| 25            |  | 14.51   | 13.01  | 12.95   | 14.51   |

|    |  |       |       |       |       |
|----|--|-------|-------|-------|-------|
| 26 |  | 28.27 | 23.67 | 24.77 | 28.27 |
| 27 |  | 32.70 | 39.83 | 38.81 | 32.70 |
| 28 |  | 30.61 | 32.53 | 32.38 | 30.61 |
| 29 |  | 11.64 | 11.67 | 12.18 | 11.64 |
| 30 |  | 13.69 | 14.55 | 14.43 | 13.69 |
| 31 |  | 30.32 | 32.33 | 32.48 | 30.32 |
| 32 |  | 33.28 | 36.57 | 35.78 | 33.28 |
| 33 |  | 8.11  | 10.42 | 10.45 | 8.11  |
| 34 |  | 8.94  | 7.10  | 9.53  | 8.94  |
| 35 |  | 32.57 | 32.75 | 31.38 | 32.57 |
| 36 |  | 32.03 | 31.89 | 32.23 | 32.03 |
| 37 |  | 14.27 | 12.77 | 12.96 | 14.27 |
| 38 |  | 27.98 | 29.02 | 29.06 | 27.98 |
| 39 |  | 27.83 | 28.05 | 29.07 | 27.83 |
| 40 |  | 14.96 | 16.84 | 16.81 | 14.96 |
| 41 |  | 8.96  | 7.10  | 7.77  | 8.96  |
| 42 |  | 28.33 | 28.15 | 28.55 | 28.33 |
| 43 |  | 30.55 | 32.40 | 32.39 | 30.55 |
| 44 |  | 11.94 | 11.34 | 11.52 | 11.94 |
| 45 |  | 18.71 | 16.74 | 17.04 | 18.71 |
| 46 |  | 16.25 | 13.99 | 14.47 | 16.25 |
| 47 |  | 35.69 | 41.30 | 41.84 | 35.69 |
| 48 |  | 10.35 | 8.60  | 11.77 | 10.35 |
| 49 |  | 32.75 | 38.35 | 38.94 | 32.75 |
| 50 |  | 31.57 | 23.93 | 32.26 | 31.57 |
| 51 |  | 18.98 | 16.90 | 16.71 | 18.98 |
| 52 |  | 25.45 | 24.38 | 24.49 | 25.45 |
| 53 |  | 12.96 | 12.73 | 12.62 | 12.96 |
| 54 |  | 33.14 | 39.01 | 38.75 | 33.14 |
| 55 |  | 27.35 | 27.90 | 29.38 | 27.35 |
| 56 |  | 14.55 | 14.17 | 14.25 | 14.55 |
| 57 |  | 29.85 | 29.39 | 28.73 | 29.85 |
| 58 |  | 17.18 | 15.30 | 15.07 | 17.18 |
| 59 |  | 17.21 | 15.09 | 15.16 | 17.21 |
| 60 |  | 13.43 | 12.91 | 12.66 | 13.43 |
| 61 |  | 16.28 | 13.68 | 14.47 | 16.28 |
| 62 |  | 9.55  | 10.70 | 10.95 | 9.55  |
| 63 |  | 29.30 | 29.34 | 29.43 | 29.30 |
| 64 |  | 28.30 | 24.23 | 25.01 | 28.30 |
| 65 |  | 9.79  | 11.16 | 11.23 | 9.79  |
| 66 |  | 7.05  | 6.37  | 6.82  | 7.05  |
| 67 |  | 12.72 | 12.88 | 12.98 | 12.72 |
| 68 |  | 26.92 | 26.45 | 26.39 | 26.92 |
| 69 |  | 17.84 | 19.12 | 17.80 | 17.84 |
| 70 |  | 30.97 | 26.19 | 27.68 | 30.97 |
| 71 |  | 35.30 | 41.40 | 39.97 | 35.30 |
| 72 |  | 15.08 | 14.51 | 14.56 | 15.08 |
| 73 |  | 17.47 | 15.16 | 15.29 | 17.47 |
| 74 |  | 13.29 | 12.80 | 13.00 | 13.29 |
| 75 |  | 17.23 | 15.29 | 15.26 | 17.23 |
| 76 |  | 25.24 | 24.28 | 24.49 | 25.24 |
| 77 |  | 14.78 | 17.50 | 15.41 | 14.78 |

---

```

Call:
lm(formula = Y ~ ., data = df)

Residuals:
    Min       1Q   Median       3Q      Max
-9.894 -1.179  0.000  1.190  7.718

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  83.965744  18.156330   4.625 4.41e-06 ***
x1          -64.624675   9.815194  -6.584 8.53e-11 ***
x2           -0.087315   0.016288  -5.361 1.10e-07 ***
x3            0.061644   0.006342   9.721 < 2e-16 ***
x4              NA         NA         NA     NA
x5            4.125913   0.322412  12.797 < 2e-16 ***
x6           -0.026842   0.090340  -0.297  0.76646
x7           19.559102   0.776469  25.190 < 2e-16 ***
x8            0.208340   0.066695   3.124  0.00185 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.799 on 760 degrees of freedom
Multiple R-squared:  0.9223,    Adjusted R-squared:  0.9216
F-statistic: 1289 on 7 and 760 DF,  p-value: < 2.2e-16

```

---

$\bar{y}$ 와 KNN 을 initial value 로 지정했을 때 Healy-westmacott procedure imputed 값이 거의 동일한 값으로 수렴했다. 따라서 Healy-westmacott procedure 는 Initial value 에 sensitive 하지 않음을 알 수 있다. 위 데이터의 경우 KNN 이  $\bar{y}$ 가 initial value 일 때보다 iteration 횟수가 더 적기 때문에 더 빨리 수렴하는 것을 알 수 있다. 위의 lm output 은 Healy-westmacott procedure 의 impute 값으로 lm 을 fitting 시킨 결과이다.  $R^2$ 가 0.92 로 굉장히 linear model 에 잘 fitting 되었음을 알 수 있다. 또한 위의 그래프는 x 축을  $y^{true}$ 로 y 축을  $y^{imputed}$ 로 그린 것이다. 그래프의 점들이 직선에 모여 있으며  $R^2$ 가 0.93 라는 것으로 보아  $y^{imputed}$ 가  $y^{true}$ 에 근사한다는 점을 알 수 있다. 따라서 initial value 에 관계 없이 모델이 같은 경우 impute 값이 같고 그 impute 값이 모델에 잘 피팅 될 때  $y^{true}$ 와  $y^{imputed}$ 값이 근사한다는 점으로 보아 Healy-westmacott procedure 은 model 에 dependent 함을 결론 지을 수 있다. 한편 Healy-westmacott procedure 로 impute 한 것과 KNN 으로 impute 한 것의 성능을 비교하기 위해  $y^{true}$ 에서  $y^{imputed}$ 와 KNN initial 의 RMSE 를 구해보았다. 아래 결과 값을 비교해 보았을 때  $y^{imputed}$ 가 KNN initial 보다  $y^{true}$ 에 가까운 것을 알 수 있다.

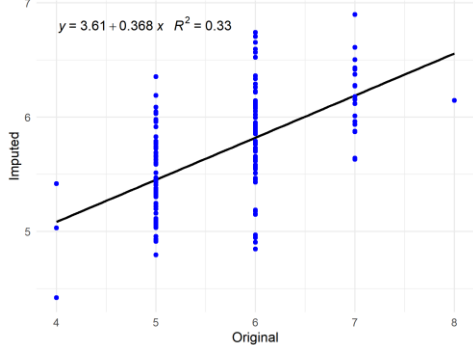
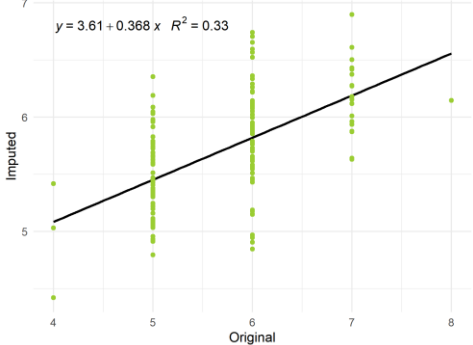
$$\frac{1}{n} \sum (y^{true} - y^{knn\ initial})^2 = 0.1554$$

$$\frac{1}{n} \sum (y^{true} - y^{imputed})^2 = 0.0481$$

## ex2. Wine Quality

## 데이터 설명

|    |                                  |              |                      |
|----|----------------------------------|--------------|----------------------|
| X1 | Fixed Acidity                    | X7           | Total Sulfur Dioxide |
| X2 | Volatile Acidity                 | X8           | Density              |
| X3 | Citric Acid                      | X9           | pH                   |
| X4 | Residual Sugar                   | X10          | sulphates            |
| X5 | Chlorides                        | X11          | alcohol              |
| X6 | Free Sulfur Dioxide              | n=1599, p=11 |                      |
| Y  | Quality (score between 0 and 10) |              |                      |

| initial value | $\bar{y}$   |         |        | knn imputation   |         |
|---------------|---|---------|--------|--|---------|
| graph         | Original vs Imputed in Wine Quality data  |         |        | Original vs Imputed in Wine Quality data(KNN version)                              |         |
|               |  |         |        |  |         |
| iteration     | 5   |         |        | 4  |         |
| comparison    | initial   | imputed | true y | initial  | imputed |
| 1             |   | 6.61    | 7      | 7  | 6.61    |
| 2             |   | 5.08    | 5      | 5  | 5.08    |
| 3             |   | 5.87    | 7      | 5  | 5.87    |
| 4             |   | 6.50    | 7      | 7  | 6.50    |
| 5             |   | 5.71    | 6      | 5  | 5.71    |
| 6             |   | 5.98    | 5      | 6  | 5.98    |
| 7             |   | 5.25    | 5      | 6  | 5.25    |
| 8             |   | 6.23    | 6      | 6  | 6.23    |
| 9             |   | 5.56    | 6      | 5  | 5.56    |
| 10            |   | 5.15    | 6      | 5  | 5.15    |
| 11            |   | 5.62    | 6      | 6  | 5.62    |
| 12            |   | 5.92    | 6      | 6  | 5.92    |
| 13            |   | 5.86    | 6      | 6  | 5.86    |
| 14            |   | 5.60    | 5      | 6  | 5.60    |
| 15            |   | 5.62    | 6      | 5  | 5.62    |
| 16            |   | 5.60    | 5      | 6  | 5.60    |
| 17            |   | 5.38    | 5      | 5  | 5.38    |
| 18            |   | 5.89    | 6      | 6  | 5.89    |
| 19            |   | 6.65    | 6      | 7  | 6.65    |
| 20            |   | 6.27    | 7      | 7  | 6.27    |
| 21            |   | 5.89    | 6      | 6  | 5.89    |
| 22            |   | 6.42    | 7      | 7  | 6.42    |
| 23            |   | 5.04    | 5      | 5  | 5.04    |
| 24            |   | 5.43    | 5      | 5  | 5.43    |
| 25            |   | 5.51    | 5      | 6  | 5.51    |
| 26            |   | 5.51    | 6      | 5  | 5.51    |
| 27            |   | 6.29    | 6      | 7  | 6.29    |
| 28            |   | 5.08    | 5      | 6  | 5.08    |
| 29            |   | 4.91    | 6      | 5  | 4.91    |
| 30            |   | 5.59    | 5      | 5  | 5.59    |
| 31            |   | 5.96    | 5      | 6  | 5.96    |
| 32            |   | 5.77    | 5      | 6  | 5.77    |
| 33            |   | 5.45    | 6      | 6  | 5.45    |
| 34            |   | 6.05    | 5      | 5  | 6.05    |

|    |      |      |   |   |      |
|----|------|------|---|---|------|
| 35 |      | 5.76 | 6 | 6 | 5.76 |
| 36 |      | 5.70 | 6 | 6 | 5.70 |
| 37 |      | 5.35 | 5 | 5 | 5.35 |
| 38 |      | 5.74 | 5 | 6 | 5.74 |
| 39 |      | 5.74 | 5 | 5 | 5.74 |
| 40 |      | 5.30 | 5 | 5 | 5.30 |
| 41 |      | 6.11 | 6 | 6 | 6.11 |
| 42 |      | 5.04 | 5 | 5 | 5.04 |
| 43 |      | 6.22 | 6 | 6 | 6.22 |
| 44 |      | 6.90 | 7 | 7 | 6.90 |
| 45 |      | 4.42 | 4 | 5 | 4.42 |
| 46 |      | 5.89 | 6 | 6 | 5.89 |
| 47 |      | 5.66 | 6 | 5 | 5.66 |
| 48 |      | 5.69 | 5 | 5 | 5.69 |
| 49 |      | 6.09 | 5 | 6 | 6.09 |
| 50 |      | 5.43 | 5 | 5 | 5.43 |
| 51 |      | 5.16 | 5 | 5 | 5.16 |
| 52 |      | 6.43 | 7 | 6 | 6.43 |
| 53 |      | 5.79 | 6 | 6 | 5.79 |
| 54 |      | 5.41 | 5 | 5 | 5.41 |
| 55 |      | 4.96 | 5 | 5 | 4.96 |
| 56 |      | 5.07 | 5 | 5 | 5.07 |
| 57 |      | 6.15 | 8 | 7 | 6.15 |
| 58 |      | 5.65 | 6 | 5 | 5.65 |
| 59 |      | 5.63 | 5 | 6 | 5.63 |
| 60 |      | 6.59 | 6 | 7 | 6.59 |
| 61 |      | 4.85 | 6 | 6 | 4.85 |
| 62 |      | 5.86 | 6 | 6 | 5.86 |
| 63 |      | 5.76 | 6 | 6 | 5.76 |
| 64 |      | 5.47 | 6 | 5 | 5.47 |
| 65 |      | 5.19 | 6 | 5 | 5.19 |
| 66 |      | 4.97 | 6 | 6 | 4.97 |
| 67 |      | 6.01 | 7 | 6 | 6.01 |
| 68 |      | 5.05 | 5 | 5 | 5.05 |
| 69 |      | 6.19 | 5 | 5 | 6.19 |
| 70 |      | 5.33 | 5 | 5 | 5.33 |
| 71 |      | 5.46 | 6 | 5 | 5.46 |
| 72 |      | 4.93 | 5 | 5 | 4.93 |
| 73 |      | 5.79 | 5 | 6 | 5.79 |
| 74 |      | 5.73 | 6 | 6 | 5.73 |
| 75 |      | 6.02 | 6 | 6 | 6.02 |
| 76 |      | 5.93 | 6 | 6 | 5.93 |
| 77 |      | 5.77 | 5 | 6 | 5.77 |
| 78 |      | 6.26 | 6 | 6 | 6.26 |
| 79 |      | 5.86 | 6 | 6 | 5.86 |
| 80 |      | 5.37 | 5 | 5 | 5.37 |
| 81 |      | 6.05 | 6 | 6 | 6.05 |
| 82 | 5.63 | 5.43 | 5 | 5 | 5.43 |
| 83 |      | 6.28 | 7 | 6 | 6.28 |
| 84 |      | 5.73 | 5 | 6 | 5.73 |
| 85 |      | 6.14 | 6 | 6 | 6.14 |
| 86 |      | 6.09 | 6 | 6 | 6.09 |
| 87 |      | 5.69 | 5 | 6 | 5.69 |
| 88 |      | 6.74 | 6 | 6 | 6.74 |
| 89 |      | 5.90 | 6 | 5 | 5.90 |
| 90 |      | 4.91 | 5 | 5 | 4.91 |
| 91 |      | 5.88 | 6 | 6 | 5.88 |
| 92 |      | 5.55 | 6 | 5 | 5.55 |
| 93 |      | 5.78 | 6 | 5 | 5.78 |
| 94 |      | 5.95 | 6 | 6 | 5.95 |
| 95 |      | 5.65 | 5 | 5 | 5.65 |
| 96 |      | 6.14 | 6 | 6 | 6.14 |
| 97 |      | 5.64 | 7 | 5 | 5.64 |

|     |  |      |   |   |      |
|-----|--|------|---|---|------|
| 98  |  | 5.92 | 5 | 5 | 5.92 |
| 99  |  | 5.44 | 5 | 5 | 5.44 |
| 100 |  | 5.46 | 5 | 5 | 5.46 |
| 101 |  | 4.91 | 5 | 5 | 4.91 |
| 102 |  | 5.18 | 6 | 6 | 5.18 |
| 103 |  | 5.69 | 5 | 5 | 5.69 |
| 104 |  | 5.15 | 6 | 5 | 5.15 |
| 105 |  | 6.10 | 6 | 6 | 6.10 |
| 106 |  | 5.11 | 5 | 5 | 5.11 |
| 107 |  | 5.32 | 5 | 6 | 5.32 |
| 108 |  | 5.99 | 6 | 6 | 5.99 |
| 109 |  | 4.79 | 5 | 5 | 4.79 |
| 110 |  | 5.73 | 6 | 6 | 5.73 |
| 111 |  | 5.42 | 4 | 5 | 5.42 |
| 112 |  | 6.38 | 7 | 6 | 6.38 |
| 113 |  | 5.63 | 7 | 6 | 5.63 |
| 114 |  | 5.03 | 5 | 5 | 5.03 |
| 115 |  | 6.15 | 7 | 6 | 6.15 |
| 116 |  | 5.58 | 6 | 6 | 5.58 |
| 117 |  | 5.73 | 6 | 6 | 5.73 |
| 118 |  | 6.03 | 5 | 6 | 6.03 |
| 119 |  | 5.33 | 5 | 5 | 5.33 |
| 120 |  | 5.95 | 6 | 6 | 5.95 |
| 121 |  | 5.47 | 5 | 6 | 5.47 |
| 122 |  | 6.12 | 7 | 6 | 6.12 |
| 123 |  | 6.13 | 6 | 6 | 6.13 |
| 124 |  | 6.00 | 6 | 6 | 6.00 |
| 125 |  | 5.43 | 6 | 5 | 5.43 |
| 126 |  | 5.10 | 5 | 5 | 5.10 |
| 127 |  | 5.64 | 6 | 6 | 5.64 |
| 128 |  | 5.55 | 6 | 6 | 5.55 |
| 129 |  | 5.58 | 6 | 6 | 5.58 |
| 130 |  | 5.04 | 5 | 5 | 5.04 |
| 131 |  | 6.52 | 6 | 6 | 6.52 |
| 132 |  | 5.07 | 5 | 5 | 5.07 |
| 133 |  | 5.20 | 5 | 5 | 5.20 |
| 134 |  | 5.34 | 5 | 5 | 5.34 |
| 135 |  | 4.95 | 6 | 5 | 4.95 |
| 136 |  | 5.91 | 6 | 5 | 5.91 |
| 137 |  | 5.36 | 5 | 5 | 5.36 |
| 138 |  | 6.34 | 6 | 6 | 6.34 |
| 139 |  | 5.93 | 7 | 6 | 5.93 |
| 140 |  | 6.35 | 5 | 6 | 6.35 |
| 141 |  | 6.08 | 6 | 6 | 6.08 |
| 142 |  | 6.14 | 6 | 6 | 6.14 |
| 143 |  | 5.73 | 6 | 6 | 5.73 |
| 144 |  | 5.44 | 5 | 5 | 5.44 |
| 145 |  | 5.88 | 7 | 6 | 5.88 |
| 146 |  | 5.96 | 7 | 6 | 5.96 |
| 147 |  | 5.22 | 5 | 5 | 5.22 |
| 148 |  | 6.18 | 7 | 7 | 6.18 |
| 149 |  | 6.71 | 6 | 7 | 6.71 |
| 150 |  | 5.66 | 5 | 5 | 5.66 |
| 151 |  | 6.57 | 6 | 6 | 6.57 |
| 152 |  | 5.03 | 4 | 5 | 5.03 |
| 153 |  | 5.83 | 5 | 6 | 5.83 |
| 154 |  | 5.22 | 5 | 5 | 5.22 |
| 155 |  | 5.90 | 6 | 6 | 5.90 |
| 156 |  | 6.13 | 6 | 6 | 6.13 |
| 157 |  | 5.60 | 5 | 5 | 5.60 |
| 158 |  | 5.73 | 6 | 5 | 5.73 |
| 159 |  | 6.08 | 6 | 6 | 6.08 |
| 160 |  | 6.36 | 6 | 6 | 6.36 |

---

```

Call:
lm(formula = Y ~ ., data = df)

Residuals:
    Min       1Q   Median       3Q      Max
-2.69724 -0.33135 -0.00003  0.38933  2.03790

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.484e+01  2.029e+01   0.732   0.4646
x1           2.329e-02  2.484e-02   0.938   0.3485
x2          -1.136e+00  1.159e-01  -9.801 < 2e-16 ***
x3          -2.194e-01  1.409e-01  -1.557   0.1196
x4           8.535e-03  1.436e-02   0.594   0.5524
x5          -1.889e+00  4.014e-01  -4.707  2.73e-06 ***
x6           4.613e-03  2.078e-03   2.220   0.0266 *
x7          -3.232e-03  6.976e-04  -4.634  3.88e-06 ***
x8          -1.063e+01  2.071e+01  -0.513   0.6077
x9          -4.486e-01  1.834e-01  -2.446   0.0146 *
x10           8.909e-01  1.094e-01   8.140  7.89e-16 ***
x11           2.858e-01  2.535e-02  11.274 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6203 on 1587 degrees of freedom
Multiple R-squared:  0.3859,    Adjusted R-squared:  0.3817
F-statistic: 90.67 on 11 and 1587 DF,  p-value: < 2.2e-16

```

---

$\bar{y}$ 와 KNN 을 initial value 로 지정했을 때 Healy-westmacott procedure imputed 값이 거의 동일한 값으로 수렴했다. 위 데이터의 경우 KNN 이  $\bar{y}$ 가 initial value 일 때보다 iteration 횟수가 더 적기 때문에 더 빨리 수렴하는 것을 알 수 있다. 위의 lm output 에 따르면  $R^2$ 가 0.38 로 linear model 에 부적합한 것으로 보인다. 또한 위의 그래프를 통해 위의 데이터의 Y 값이 0~10 사이의 integer value 인데 반해 impute 값은 실수(Real Value)이기 때문에  $R^2$ 가 0.33 으로 그래프의 점들이 직선을 따르지 않는다는 것을 알 수 있다. 한편 Healy-westmacott procedure 로 impute 한 것과 KNN 으로 impute 한 것의 성능을 비교하기 위해  $y^{true}$ 에서  $y^{imputed}$ 와 KNN initial 의 RMSE 를 구해보았다. 아래 결과 값을 비교해 보았을 때  $y^{imputed}$ 가 KNN initial 보다  $y^{true}$ 에 가까운 것을 알 수 있다.

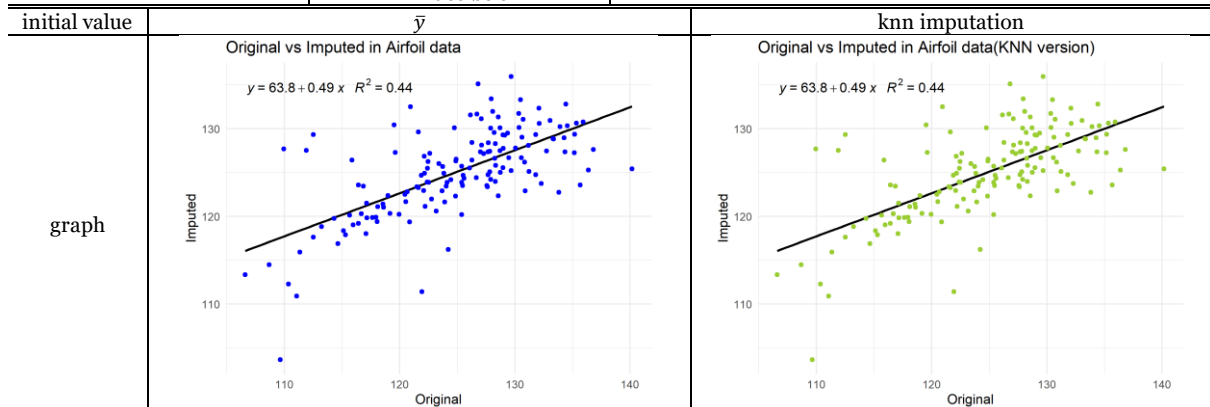
$$\frac{1}{n} \sum (y^{true} - y^{knn\ initial})^2 = 0.0438$$

$$\frac{1}{n} \sum (y^{true} - y^{imputed})^2 = 0.0184$$

## ex3. Airfoil Self-noise

## 데이터 설명

|    |   |              |  |
|----|---|--------------|--|
| X1 | Frequency, in Hertz                     | X4           | Free-stream velocity in meters per second      |
| X2 | Angle of attack in degrees              | X5           | Suction side displacement thickness, in meters |
| X3 | Chord length in meters                  | n= 1503, p=5 |  |
| Y  | Scaled sound pressure level in decibels |              |  |



| iteration  | 6       |         |        | 6       |         |
|------------|---------|---------|--------|---------|---------|
| comparison | initial | imputed | true y | initial | imputed |
| 1          |         | 116.21  | 124.21 | 125.90  | 116.21  |
| 2          |         | 122.33  | 128.56 | 125.79  | 122.33  |
| 3          |         | 123.52  | 125.47 | 128.47  | 123.52  |
| 4          |         | 120.23  | 119.94 | 123.47  | 120.23  |
| 5          |         | 131.65  | 126.71 | 127.84  | 131.65  |
| 6          |         | 118.36  | 115.14 | 120.52  | 118.36  |
| 7          |         | 122.49  | 120.47 | 122.91  | 122.49  |
| 8          |         | 126.41  | 115.86 | 124.93  | 126.41  |
| 9          |         | 121.11  | 118.09 | 120.62  | 121.11  |
| 10         |         | 130.07  | 128.71 | 131.47  | 130.07  |
| 11         |         | 130.60  | 132.11 | 132.13  | 130.60  |
| 12         |         | 117.62  | 112.52 | 121.13  | 117.62  |
| 13         |         | 124.22  | 127.90 | 125.15  | 124.22  |
| 14         |         | 121.66  | 120.54 | 132.43  | 121.66  |
| 15         |         | 135.92  | 129.68 | 131.52  | 135.92  |
| 16         |         | 126.17  | 130.84 | 128.45  | 126.17  |
| 17         |         | 130.43  | 119.51 | 127.85  | 130.43  |
| 18         |         | 127.15  | 127.28 | 128.45  | 127.15  |
| 19         |         | 124.33  | 125.58 | 128.25  | 124.33  |
| 20         |         | 133.29  | 130.50 | 133.59  | 133.29  |
| 21         |         | 132.33  | 132.09 | 131.81  | 132.33  |
| 22         |         | 128.84  | 133.31 | 130.43  | 128.84  |
| 23         |         | 119.38  | 118.05 | 123.47  | 119.38  |
| 24         |         | 123.92  | 122.39 | 127.79  | 123.92  |
| 25         |         | 117.90  | 115.30 | 119.09  | 117.90  |
| 26         |         | 123.57  | 116.41 | 121.69  | 123.57  |
| 27         |         | 121.13  | 122.20 | 122.38  | 121.13  |
| 28         |         | 116.89  | 114.63 | 115.75  | 116.89  |
| 29         |         | 123.59  | 135.67 | 124.68  | 123.59  |
| 30         |         | 127.78  | 130.05 | 128.48  | 127.78  |
| 31         |         | 129.49  | 129.37 | 127.76  | 129.49  |
| 32         |         | 125.70  | 125.39 | 127.71  | 125.70  |
| 33         |         | 119.87  | 117.65 | 117.31  | 119.87  |
| 34         |         | 120.31  | 116.68 | 117.09  | 120.31  |
| 35         |         | 120.60  | 123.18 | 128.44  | 120.60  |



|    |        |        |        |        |        |
|----|--------|--------|--------|--------|--------|
| 36 |        | 122.92 | 123.69 | 123.79 | 122.92 |
| 37 |        | 123.36 | 121.50 | 124.40 | 123.36 |
| 38 |        | 115.93 | 111.35 | 115.45 | 115.93 |
| 39 |        | 129.28 | 128.96 | 127.58 | 129.28 |
| 40 |        | 121.62 | 123.92 | 124.93 | 121.62 |
| 41 |        | 125.11 | 131.22 | 126.37 | 125.11 |
| 42 |        | 123.44 | 124.05 | 123.79 | 123.44 |
| 43 |        | 119.75 | 114.31 | 121.80 | 119.75 |
| 44 |        | 131.55 | 126.15 | 129.97 | 131.55 |
| 45 |        | 125.40 | 140.16 | 129.26 | 125.40 |
| 46 |        | 126.51 | 124.90 | 129.01 | 126.51 |
| 47 |        | 112.30 | 110.36 | 117.71 | 112.30 |
| 48 |        | 123.43 | 116.85 | 122.23 | 123.43 |
| 49 |        | 132.49 | 120.95 | 131.89 | 132.49 |
| 50 |        | 123.74 | 132.30 | 124.68 | 123.74 |
| 51 |        | 123.88 | 124.11 | 126.52 | 123.88 |
| 52 |        | 127.40 | 127.83 | 126.51 | 127.40 |
| 53 |        | 127.35 | 134.34 | 131.43 | 127.35 |
| 54 |        | 129.63 | 121.62 | 127.66 | 129.63 |
| 55 |        | 123.87 | 125.40 | 132.43 | 123.87 |
| 56 |        | 130.31 | 134.57 | 132.75 | 130.31 |
| 57 |        | 111.41 | 121.93 | 125.39 | 111.41 |
| 58 |        | 131.11 | 127.10 | 130.43 | 131.11 |
| 59 |        | 123.88 | 122.53 | 125.45 | 123.88 |
| 60 |        | 133.40 | 127.95 | 129.98 | 133.40 |
| 61 |        | 124.75 | 127.68 | 125.60 | 124.75 |
| 62 |        | 128.40 | 126.27 | 130.58 | 128.40 |
| 63 |        | 128.13 | 127.13 | 129.30 | 128.13 |
| 64 |        | 123.98 | 128.31 | 125.94 | 123.98 |
| 65 |        | 128.96 | 134.27 | 128.06 | 128.96 |
| 66 |        | 127.67 | 109.95 | 124.52 | 127.67 |
| 67 |        | 127.33 | 127.01 | 126.69 | 127.33 |
| 68 |        | 132.80 | 134.43 | 133.74 | 132.80 |
| 69 |        | 125.47 | 122.43 | 123.86 | 125.47 |
| 70 |        | 126.30 | 124.88 | 130.39 | 126.30 |
| 71 |        | 129.32 | 131.81 | 127.76 | 129.32 |
| 72 |        | 126.02 | 123.42 | 126.24 | 126.02 |
| 73 |        | 126.69 | 130.56 | 126.42 | 126.69 |
| 74 |        | 127.28 | 119.62 | 124.52 | 127.28 |
| 75 |        | 119.83 | 117.15 | 122.31 | 119.83 |
| 76 |        | 130.04 | 127.90 | 129.23 | 130.04 |
| 77 |        | 126.24 | 122.47 | 131.17 | 126.24 |
| 78 |        | 129.36 | 135.19 | 130.43 | 129.36 |
| 79 |        | 127.74 | 128.99 | 130.36 | 127.74 |
| 80 |        | 125.80 | 128.34 | 123.86 | 125.80 |
| 81 |        | 126.41 | 126.27 | 126.23 | 126.41 |
| 82 | 124.84 | 127.63 | 136.83 | 133.84 | 127.63 |
| 83 |        | 130.22 | 133.92 | 131.88 | 130.22 |
| 84 |        | 125.55 | 128.98 | 132.43 | 125.55 |
| 85 |        | 103.65 | 109.64 | 113.10 | 103.65 |
| 86 |        | 130.93 | 133.06 | 132.23 | 130.93 |
| 87 |        | 121.40 | 118.56 | 124.68 | 121.40 |
| 88 |        | 129.01 | 130.63 | 131.45 | 129.01 |
| 89 |        | 125.07 | 127.81 | 126.94 | 125.07 |
| 90 |        | 113.36 | 106.60 | 112.72 | 113.36 |
| 91 |        | 131.33 | 128.56 | 129.97 | 131.33 |
| 92 |        | 120.32 | 119.17 | 123.15 | 120.32 |
| 93 |        | 123.27 | 121.63 | 123.71 | 123.27 |
| 94 |        | 127.52 | 111.91 | 124.52 | 127.52 |
| 95 |        | 119.18 | 116.42 | 119.53 | 119.18 |
| 96 |        | 131.97 | 128.08 | 131.89 | 131.97 |
| 97 |        | 129.24 | 129.12 | 127.66 | 129.24 |
| 98 |        | 119.89 | 117.96 | 121.86 | 119.89 |

|     |  |        |        |        |        |
|-----|--|--------|--------|--------|--------|
| 99  |  | 120.12 | 115.66 | 125.89 | 120.12 |
| 100 |  | 124.66 | 121.89 | 125.70 | 124.66 |
| 101 |  | 114.47 | 108.69 | 115.79 | 114.47 |
| 102 |  | 130.61 | 135.35 | 133.79 | 130.61 |
| 103 |  | 119.36 | 120.86 | 124.33 | 119.36 |
| 104 |  | 125.01 | 128.74 | 130.84 | 125.01 |
| 105 |  | 128.80 | 133.35 | 130.58 | 128.80 |
| 106 |  | 128.29 | 128.30 | 130.58 | 128.29 |
| 107 |  | 128.10 | 131.19 | 132.71 | 128.10 |
| 108 |  | 127.24 | 135.16 | 130.66 | 127.24 |
| 109 |  | 110.90 | 111.08 | 115.14 | 110.90 |
| 110 |  | 127.36 | 127.69 | 125.72 | 127.36 |
| 111 |  | 126.67 | 129.47 | 126.15 | 126.67 |
| 112 |  | 124.92 | 123.84 | 129.74 | 124.92 |
| 113 |  | 127.45 | 132.76 | 127.38 | 127.45 |
| 114 |  | 125.67 | 123.74 | 130.17 | 125.67 |
| 115 |  | 118.82 | 113.23 | 119.07 | 118.82 |
| 116 |  | 123.49 | 127.58 | 124.61 | 123.49 |
| 117 |  | 122.82 | 120.65 | 126.26 | 122.82 |
| 118 |  | 122.93 | 122.09 | 127.13 | 122.93 |
| 119 |  | 131.07 | 130.72 | 129.97 | 131.07 |
| 120 |  | 131.71 | 130.33 | 131.89 | 131.71 |
| 121 |  | 128.39 | 127.70 | 130.57 | 128.39 |
| 122 |  | 126.61 | 128.25 | 130.88 | 126.61 |
| 123 |  | 124.16 | 124.45 | 125.52 | 124.16 |
| 124 |  | 130.76 | 135.94 | 132.38 | 130.76 |
| 125 |  | 127.17 | 122.61 | 131.56 | 127.17 |
| 126 |  | 129.54 | 128.22 | 127.58 | 129.54 |
| 127 |  | 120.20 | 125.38 | 128.90 | 120.20 |
| 128 |  | 127.48 | 128.71 | 126.64 | 127.48 |
| 129 |  | 121.96 | 122.78 | 124.56 | 121.96 |
| 130 |  | 122.74 | 133.81 | 123.68 | 122.74 |
| 131 |  | 122.89 | 130.90 | 123.68 | 122.89 |
| 132 |  | 125.22 | 129.69 | 126.37 | 125.22 |
| 133 |  | 125.28 | 136.38 | 129.69 | 125.28 |
| 134 |  | 124.42 | 126.65 | 129.26 | 124.42 |
| 135 |  | 122.29 | 124.84 | 125.66 | 122.29 |
| 136 |  | 118.04 | 117.09 | 119.81 | 118.04 |
| 137 |  | 123.39 | 127.63 | 128.47 | 123.39 |
| 138 |  | 130.11 | 130.37 | 125.20 | 130.11 |
| 139 |  | 119.04 | 115.97 | 120.38 | 119.04 |
| 140 |  | 124.68 | 125.52 | 127.14 | 124.68 |
| 141 |  | 129.30 | 112.51 | 123.44 | 129.30 |
| 142 |  | 130.09 | 124.76 | 129.75 | 130.09 |
| 143 |  | 135.09 | 126.81 | 131.25 | 135.09 |
| 144 |  | 124.74 | 131.82 | 124.93 | 124.74 |
| 145 |  | 125.51 | 126.09 | 126.98 | 125.51 |
| 146 |  | 126.84 | 122.15 | 125.50 | 126.84 |
| 147 |  | 122.36 | 118.99 | 127.08 | 122.36 |
| 148 |  | 121.51 | 117.13 | 125.46 | 121.51 |
| 149 |  | 124.90 | 122.17 | 130.36 | 124.90 |
| 150 |  | 121.03 | 118.62 | 125.79 | 121.03 |

---

```

Call:
lm(formula = Y ~ ., data = df)

Residuals:
    Min       1Q   Median       3Q      Max
-17.111  -2.563   0.000   2.646  15.793

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.329e+02  5.130e-01  258.98  <2e-16 ***
x1           -1.298e-03  3.966e-05  -32.74  <2e-16 ***
x2           -3.942e-01  3.663e-02  -10.76  <2e-16 ***
x3           -3.530e+01  1.536e+00  -22.99  <2e-16 ***
x4            9.880e-02  7.659e-03   12.90  <2e-16 ***
x5           -1.606e+02  1.414e+01  -11.36  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.529 on 1497 degrees of freedom
Multiple R-squared:  0.551,    Adjusted R-squared:  0.5495
F-statistic: 367.4 on 5 and 1497 DF,  p-value: < 2.2e-16

```

---

$\bar{y}$ 와 KNN 을 initial value 로 지정했을 때 Healy-westmacott procedure imputed 값이 거의 동일한 값으로 수렴했다. 위의 lm output 에 따르면  $R^2$ 가 0.55 이며 위의 그래프는 x 축을  $y^{true}$ 로 y 축을  $y^{imputed}$ 로 그렸을 때 lm 의  $R^2$ 가 0.44 인 것을 보여준다. 한편 Healy-westmacott procedure 로 impute 한 것과 KNN 으로 impute 한 것의 성능을 비교하기 위해  $y^{true}$ 에서  $y^{imputed}$ 와 KNN initial 의 RMSE 를 구해보았다. 아래 결과 값을 비교해 보았을 때  $y^{imputed}$ 가 KNN initial 보다  $y^{true}$ 에 가까운 것을 알 수 있다.

$$\frac{1}{n} \sum (y^{true} - y^{knn\ initial})^2 = 1.7811$$

$$\frac{1}{n} \sum (y^{true} - y^{imputed})^2 = 0.1937$$

### 3. Discussion

---

1 dimensional 2 clusters GMM 을 mygmm 함수로 구현해 봄으로써 EM 알고리즘을 깊이 이해할 수 있었다. mygmm 이 가장 성능이 좋을 때는  $\mu_1$ 과  $\mu_2$ 의 거리가 멀어 mixture 의 density 가 양봉을 이룰 때였다. 반면  $\mu_1$ 과  $\mu_2$ 의 거리가 가까워 두 분포가 overlap 을 될 때는 mixture 로부터 cluster 하는 성능이 떨어졌다.  $\mu_1$ 과  $\mu_2$ 이 같을 때는 cluster 들의 식별가능성이 사라졌다. 또한 n 의 크기가 커질수록 MLE 의 consistency 성질에 따라 추정치들이 모수로 근사한다는 것을 알 수 있었다. 1 dimensional 2 clusters 에서 p dimensional k clusters 으로 확장하고 싶었지만 코딩의 미숙함으로 인해 구현할 수 없어 아쉬움이 남는다. 하지만 다행히 R 의 mclust 패키지를 이용한다면 직접 코딩하지 않고 간편하게 p dimensional k clusters 를 시행할 수 있다.

한편 EM 은 incomplete 데이터의 unobserved data 의 conditional expectation 을 구해 추정한다는 점에서 결측치를 impute 할 수 있다는 장점을 갖고 있다. Result 의 incomplete 한 이변량 정규분포 예제에 EM 을 이용해 결측치를 impute 하고 나머지 모수들의 MLE 값을 구해볼 수 있었다. 이러한 EM 의 imputation 은 특히 numeric 데이터의 결측치를 해결하기 위한 방법 중 하나로 널리 활용되고 있다.

Healy-westmacott procedure 는 반응 변수(response variable)을 impute 시키는 일종의 EM 알고리즘이다. Healy-westmacott procedure 은 initial value 와 model 에 dependent 한 것으로 알려져 있다. 본 과제에서는 initial value 를  $\bar{y}$ 와 KNN 로 두 가지의 경우를 고려했으며 model 은 linear model 을 사용하였다. 그 결과 impute 값은 initial value 에 관계 없이 동일한 값으로 수렴했다. 또한 데이터가 lm 모델에 피팅이 잘 될수록 Impute 값이 true value 에 근사했다. 이를 통해 Healy-westmacott procedure 은 model 에 dependent 하기 때문에 impute 하기 위해 데이터에 적합한 모델을 사용해야 한다는 것을 알았다. 한편 대표적인 imputation 방법 중 하나인 KNN 과 Healy-westmacott procedure 의 impute 값을 RMSE measure 를 이용하여 비교했을 때 후자의 RMSE 값이 더 작아 true value 에 근사 한다는 것을 알 수 있었다.

## 4. Appendix

```
rm(list=ls())
library(tidyverse); library(mclust)

# hwl make your own gmm function
mygmm <- function(x){

  # initial parameter
  x <- as.vector(x)
  pi1 <- pi2 <- 0.5
  mu <- sample(x,2)
  mu1 <- mu[which.min(mu)]; mu2 <- mu[which.max(mu)] # Assumed that mu1 < mu2
  sigma1 <- sigma2 <- sd(x)
  n <- length(x)
  old.loglik <- sum(log(pi1)+log(dnorm(x,mu1,sigma1)) + log(pi2)+log(dnorm(x,mu2,sigma2)))

  err <- 1
  thr <- 10^(-6)
  maxiter <- 100
  niter <- 0

  while ( niter <= maxiter && err>thr ){
    # E -step( calculate yi & unobserved conditional expectation )
    y <- (dnorm(x=x, mean=mu1,
sd=sigma1)*pi1) / (pi1*dnorm(x,mu1,sigma1)+pi2*dnorm(x,mu2,sigma2))

    # M -step
    # update parameter( maximization )
    pi1 <- sum(y)/n; pi2 <- 1-pi1
    mu1 <- sum(x*y)/sum(y); mu2 <- sum(x*(1-y))/sum(1-y);
    sigma1 <- sqrt(sum(y*(x-mu1)^2)/sum(y)); sigma2 <- sqrt(sum((1-y)*(x-mu2)^2)/sum(1-y));
    new.loglik <- sum(log(pi1)+log(dnorm(x,mu1,sigma1)) + log(pi2)+log(dnorm(x,mu2,sigma2)))
```

```

    # update niter and err
    niter <- niter + 1
    err <- abs(new.loglik - old.loglik)
    old.loglik <- new.loglik
  }

  mem <- ifelse(y>0.5, 1, 2) # 1 이 mu1 을 가진 cluster
  theta <- data.frame(pi=c(pi1, pi2), mu=c(mu1, mu2), sigma=c(sigma1,sigma2))
  rownames(theta) <- c("cluster1", "cluster2")

  return(list(cluster=mem, parameter=theta))
}

# ex1) mean 차이가 큰 분포
set.seed(1)
comp1.vals <- data.frame(comp="A", vals=rnorm(70, mean=1, sd=0.5))
comp2.vals <- data.frame(comp="B", vals=rnorm(30, mean=3, sd=0.5))
vals.df <- rbind(comp1.vals, comp2.vals)
ex1.s <- mygmm(vals.df[,2])

comp1.vals <- data.frame(comp="A", vals=rnorm(700, mean=1, sd=0.5))
comp2.vals <- data.frame(comp="B", vals=rnorm(300, mean=3, sd=0.5))
vals.df <- rbind(comp1.vals, comp2.vals)
ex1.b <- mygmm(vals.df[,2])

plot(density(vals.df[,2]),main="Density of example1",ylim=c(0,1))
lines(density(comp1.vals[,2]),lty=2,col='red')
lines(density(comp2.vals[,2]),lty=3,col='blue')
legend("topright",legend=c("mixture", "N(1, (0.5)^2)", "N(3, (0.5)^2)"),
      col=c("black", "red", "blue"),lty=c(1,2,3))

# ex2) mean 0이 overlapping
set.seed(1)
comp1.vals <- data.frame(comp="A", vals=rnorm(70, mean=1, sd=0.5))
comp2.vals <- data.frame(comp="B", vals=rnorm(30, mean=1.5, sd=0.5))
vals.df <- rbind(comp1.vals, comp2.vals)
ex1.s <- mygmm(vals.df[,2])

comp1.vals <- data.frame(comp="A", vals=rnorm(700, mean=1, sd=0.5))
comp2.vals <- data.frame(comp="B", vals=rnorm(300, mean=1.5, sd=0.5))
vals.df <- rbind(comp1.vals, comp2.vals)
ex1.b <- mygmm(vals.df[,2])

plot(density(vals.df[,2]),main="Density of example2",ylim=c(0,1))
lines(density(comp1.vals[,2]),lty=2,col='red')
lines(density(comp2.vals[,2]),lty=3,col='blue')
legend("topright",legend=c("mixture", "N(1, (0.5)^2)", "N(1.5, (0.5)^2)"),
      col=c("black", "red", "blue"),lty=c(1,2,3))

# ex3) mean 이 같은 분포
set.seed(1)
comp1.vals <- data.frame(comp="A", vals=rnorm(70, mean=1, sd=0.3))
comp2.vals <- data.frame(comp="B", vals=rnorm(30, mean=1, sd=0.7))
vals.df <- rbind(comp1.vals, comp2.vals)
ex1.s <- mygmm(vals.df[,2])

```

```

comp1.vals <- data.frame(comp="A", vals=rnorm(700, mean=1, sd=0.3))
comp2.vals <- data.frame(comp="B", vals=rnorm(300, mean=1, sd=0.7))
vals.df <- rbind(comp1.vals, comp2.vals)
ex1.b <- mygmm(vals.df[,2])

plot(density(vals.df[,2]),main="Density of example3",ylim=c(0,1.5))
lines(density(comp1.vals[,2]),lty=2,col='red')
lines(density(comp2.vals[,2]),lty=3,col='blue')
legend("topright",legend=c("mixture", "N(1, (0.3)^2)", "N(1, (0.7)^2)"),
      col=c("black", "red", "blue"),lty=c(1,2,3))

# hw2 Bivariate Normal distribution's imputation using EM
# hw2 Bivariate Normal distribution's imputation using EM
tb <- data.frame(w1=c(8,11,16,18,6,4,20,25,9,13),
                 w2=c(10,14,16,15,20,4,18,22,NA,NA))

myftn <- function(df){

  # Initial values
  n <- nrow(df)
  mu1 <- mean(df$w1)
  sig11 <- var(df$w1)*((n-1)/n)
  mu2 <- mean(df$w2,na.rm=T)
  sig22 <- var(df$w2, na.rm=T)*((8-1)/8)
  sig12 <- 1
  cor <- sig12/(sqrt(sig11)*sqrt(sig22))
  niter <- 0; err <- 1
  maxiter <- 1000
  old.likeli <- 0

  while( niter <= maxiter && err > 10^(-10)){

    # E-step
    e19 <- mu2 + (sig12/sig11)*(df$w1[9]-mu1)
    e29 <- sig22*(1-cor^2)+ e19^2

    e110 <- mu2 + (sig12/sig11)*(df$w1[10]-mu1)
    e210 <- sig22*(1-cor^2)+ e110^2

    w2 <- c(df$w2[1:8], e19, e110)

    t1 <- sum(df$w1)
    t2 <- sum(w2)
    t11 <- sum((df$w1)^2)
    t22 <- sum(w2^2)
    t12 <- sum(df$w1*w2)

    # M-step
    mu1 <- t1/n; mu2 <- t2/n
    sig11 <- (t11-n*mu1^2)/n
    sig22 <- (t22-n*mu2^2)/n
    sig12 <- (t12-((t1*t2)/n))/n
    cor <- sig12/(sqrt(sig11)*sqrt(sig22))
    xi <- sig11*sig22-sig12^2

    new.likeli <- (-n*log(2*pi))-(1/2)*n*log(xi)-(1/2)*(xi)^(-1)*(sig22*t11+sig11*t22-
2*sig12*t12-(1/2)*(t1*(mu1*sig22-mu2*sig12)+t2*(mu2*sig11-mu1*sig12))
+n*((mu1^2)*sig22+(mu2^2)*sig11-2*mu1*mu2*sig12))

```

```

    # update niter & err
    niter <- niter + 1
    err <- abs(old.likeli-new.likeli)
    old.likeli <- new.likeli
  }

  mu <- matrix(c(mu1, mu2)); rownames(mu) <- c("mu1","mu2")
  sig <- matrix(c(sig11, sig12, sig12, sig22),ncol=2,nrow=2); rownames(sig) <- c("w1","w2");
  colnames(sig) <- c("w1","w2")
  return(list(mu=mu, sig=sig, w2=w2))
}

myftn(tb)

#hw3 impute using Healy-westmacott procedure
library(ggpmisc); library(class)
setwd('C:/Users/dnskd/Desktop/19-2/계특/과제/hw4')

# 3 example from UCI repository
en <- read.csv('energy_efficiency.csv')
wine <- read.csv('winequality-red.csv',sep=";")
air <- read.csv('airfoil_self_noise.csv')
colnames(wine) <- c(paste0("X",1:11),"Y")
colnames(air)[6] <- "Y"

# initial value type: ybar
myimp <- function(df,thr,ind){

  # missing data
  missing <- sample(nrow(df), round(nrow(df)*0.1, 0) )
  true.y <- df[missing, ind]
  df[missing, ind] <- NA

  # initial value
  ybar <- mean(df[,ind], na.rm=T)
  init <- ybar
  df[missing, ind] <- ybar
  fit <- lm(Y ~. , data=df)
  yhat <- fit$fitted.values[missing]
  y.old <- yhat

  maxiter <- 1000; niter <- 0; err <- 1

  while(niter <= maxiter && err >= thr ){

    # E-step
    df[missing, ind] <- yhat

    # M-step
    fit <- lm(Y ~. , data=df)
    yhat <- fit$fitted.values[missing]
    y.new <- yhat

    # update niter and err
    niter <- niter + 1
    err <- sum((y.old-y.new)^2)
    y.old <- y.new
  }
}

```

```

    return(list(missing=missing, imputed=y.new, true.y=true.y, init=init,
fit=summary(fit),iteration=niter))

}

# ex1. Energy Efficiency
set.seed(1)
result1 <- myimp(en, 10^(-6), 9)
df <- data.frame(y=result1$imputed, x=result1$true.y)
my.formula <- y ~ x
ggplot(data=df, aes(x = x, y = y)) +
  geom_smooth(method = "lm", se=FALSE, color="black", formula = my.formula) +
  stat_poly_eq(formula = my.formula,
               aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),
               parse = TRUE) +geom_point(col='blue')+ggtitle("Original vs Imputed in Energy
Efficiency data")+xlab("Original")+ylab("Imputed")+theme_minimal()
ggsave('energy.png')

# ex2. Wine Quality
set.seed(1)
result2 <- myimp(wine, 10^(-6), 12)
df <- data.frame(y=result2$imputed, x=result2$true.y)
my.formula <- y ~ x
ggplot(data=df, aes(x = x, y = y)) +
  geom_smooth(method = "lm", se=FALSE, color="black", formula = my.formula) +
  stat_poly_eq(formula = my.formula,
               aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),
               parse = TRUE) +geom_point(col='blue')+ggtitle("Original vs Imputed in Wine
Quality data")+xlab("Original")+ylab("Imputed")+theme_minimal()
ggsave('wine.png')

# ex3. Airfoil
set.seed(1)
result3 <- myimp(air, 10^(-6), 6)
df <- data.frame(y=result3$imputed, x=result3$true.y)
my.formula <- y ~ x
ggplot(data=df, aes(x = x, y = y)) +
  geom_smooth(method = "lm", se=FALSE, color="black", formula = my.formula) +
  stat_poly_eq(formula = my.formula,
               aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),
               parse = TRUE) +geom_point(col='blue')+ggtitle("Original vs Imputed in Airfoil
data")+xlab("Original")+ylab("Imputed")+theme_minimal()
ggsave('air.png')

# initial value using KNN
library(VIM)

myimp2 <- function(df, ind){

  # missing data
  missing <- sample(nrow(df),round(nrow(df)*0.1,0))
  true.y <- df[missing, ind]
  df[missing, ind] <- NA

  # initial value
  knn.imp <- knn(df, variable=c("Y"), k=10)
  knn.imp <- knn.imp[, -(ind+1)]
  init <- knn.imp[missing, ind]
  fit <- lm(Y~., knn.imp)
  yhat <- fit$fitted.values[missing]

```



```

y.old <- yhat
err <- 1; niter <- 0; maxiter <- 1000; thr <- 10^(-6)

while(niter <= maxiter && err >= thr){

  # E-step
  knn.imp[missing, ind] <- yhat

  # M-step
  fit <- lm(Y~., knn.imp)
  yhat <- fit$fitted.values[missing]
  y.new <- yhat

  # update niter and err
  niter <- niter + 1
  err <- sum((y.old - y.new)^2)
  y.old <- y.new
}

return(list(missing=missing, imputed=y.new, true.y=true.y, init=init,
fit=summary(fit), iteration=niter))

}

# ex1. Energy Efficiency
set.seed(1)
output1 <- myimp2(en, 9)
df <- data.frame(y=output1$imputed, x=output1$true.y)
my.formula <- y ~ x
ggplot(data=df, aes(x = x, y = y)) +
  geom_smooth(method = "lm", se=FALSE, color="black", formula = my.formula) +
  stat_poly_eq(formula = my.formula,
    aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),
    parse = TRUE) +geom_point(col='olivedrab3')+ggtitle("Original vs Imputed in
Energy Efficiency data (KNN version)")+xlab("Original")+ylab("Imputed")+theme_minimal()
ggsave('energyknn.png')

# ex2. Wine Quality
set.seed(1)
output2 <- myimp2(wine, 12)
df <- data.frame(y=output2$imputed, x=output2$true.y)
my.formula <- y ~ x
ggplot(data=df, aes(x = x, y = y)) +
  geom_smooth(method = "lm", se=FALSE, color="black", formula = my.formula) +
  stat_poly_eq(formula = my.formula,
    aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),
    parse = TRUE) +geom_point(col='olivedrab3')+ggtitle("Original vs Imputed in
Wine Quality data (KNN version)")+xlab("Original")+ylab("Imputed")+theme_minimal()
ggsave('wineknn.png')

# ex3. Airfoil
set.seed(1)
output3 <- myimp2(air, 6)
df <- data.frame(y=output3$imputed, x=output3$true.y)
my.formula <- y ~ x
ggplot(data=df, aes(x = x, y = y)) +
  geom_smooth(method = "lm", se=FALSE, color="black", formula = my.formula) +
  stat_poly_eq(formula = my.formula,
    aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),
    parse = TRUE) +geom_point(col='olivedrab3')+ggtitle("Original vs Imputed in
Airfoil data (KNN version)")+xlab("Original")+ylab("Imputed")+theme_minimal()

```

```
ggsave('airknn.png')

# RMSE 비교
sqrt(mean(output1$true.y-output1$init)^2)
sqrt(mean(output1$true.y-output1$imputed)^2)

sqrt(mean(output2$true.y-output2$init)^2)
sqrt(mean(output2$true.y-output2$imputed)^2)

sqrt(mean(output3$true.y-output3$init)^2)
sqrt(mean(output3$true.y-output3$imputed)^2)
```