통계적모델링과머신러닝실습 HW2

2018313461 정하늘

Getting ready

```
setwd("C:/Users/gksmf/OneDrive/바탕 화면/7학기/HW2")

library(mice)
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
library(rms)
```

```
## Loading required package: Hmisc
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

```
library(finalfit)
library(e1071)
```

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:Hmisc':
##
##     impute
```

```
library(gam)
```

```
## Loading required package: splines
```

```
## Loading required package: foreach
```

```
## Loaded gam 1.20
```

```
library(bestNormalize)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:Hmisc':
##
##     src, summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Q1

```
tr1 = read.csv('train.csv')
te1 = read.csv('test.csv')
```

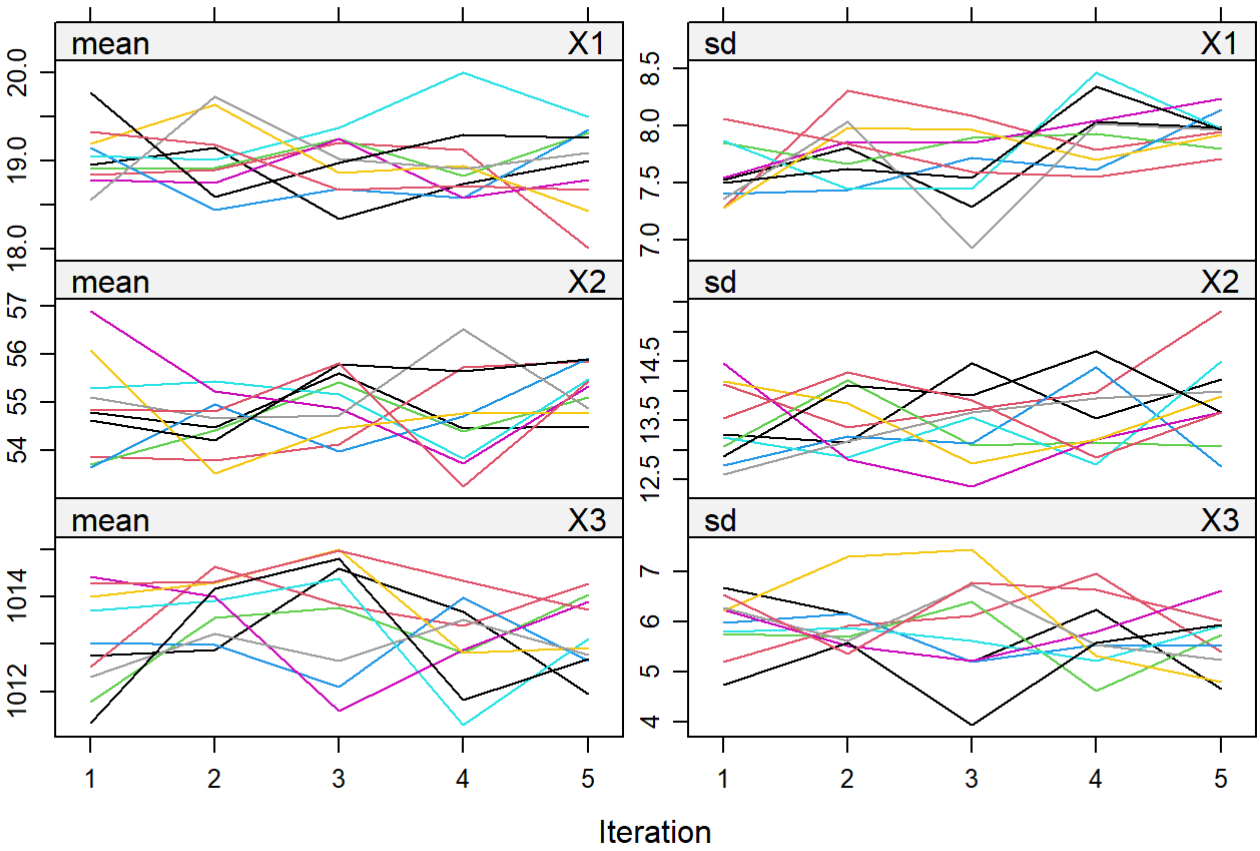
1. Data pre-processing

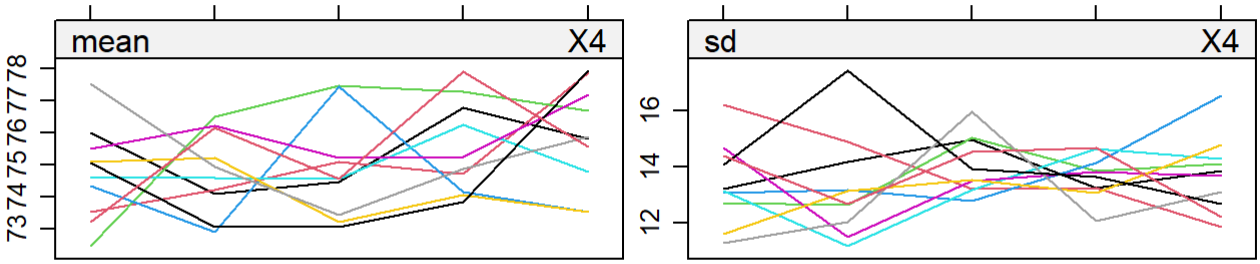
1) Missing values

변수는 모두 연속형이며, Y를 제외한 X1 ~ X4는 모두 결측값이 존재한다. 또한 결측값들이 MICE를 사용하기에 적합한 NA로 코딩되어 있다.

MICE

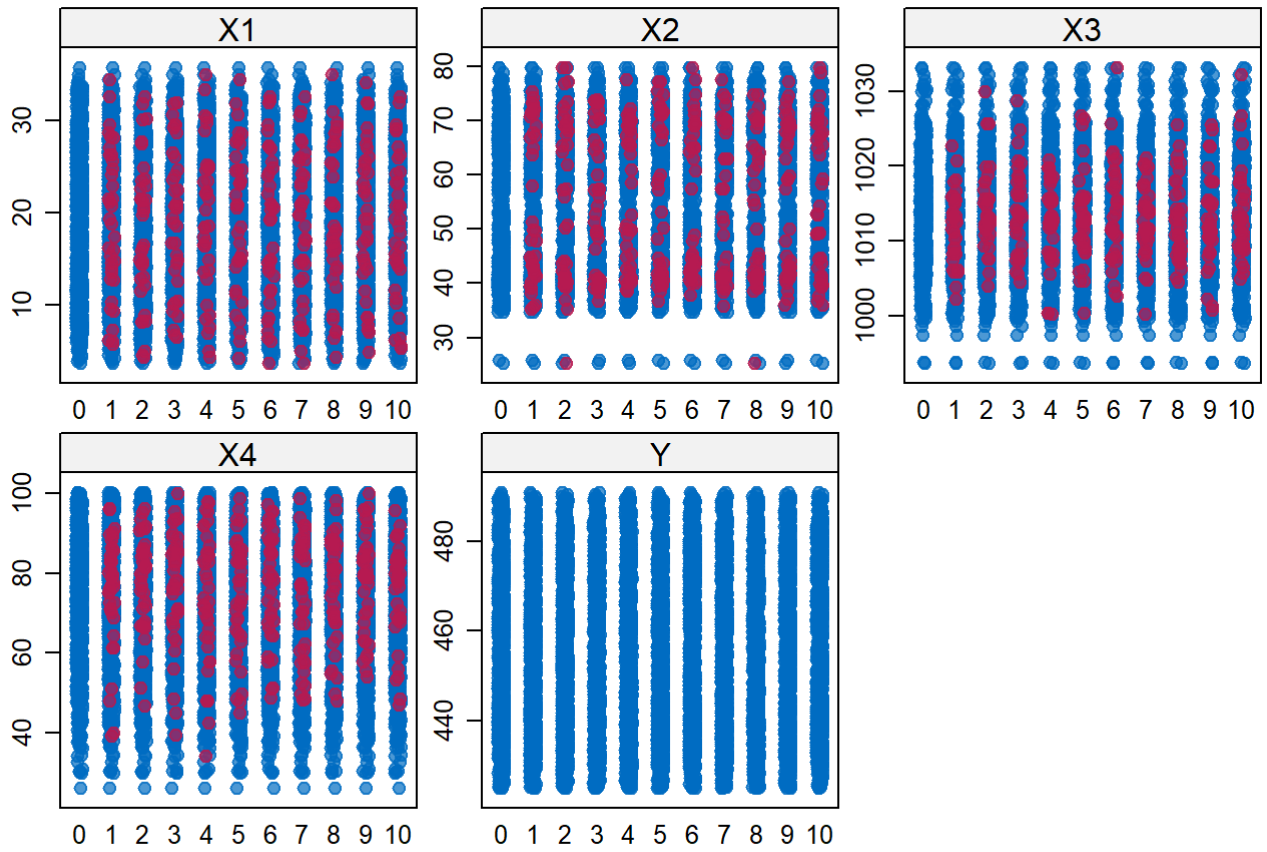
```
set.seed(0) # 일관된 분석을 위해 임의로 설정

imp = mice(tr1, m = 10, method = c('pmm','pmm','pmm','pmm',''), print = F)

plot(imp, c('X1', 'X2', 'X3', 'X4'))
```



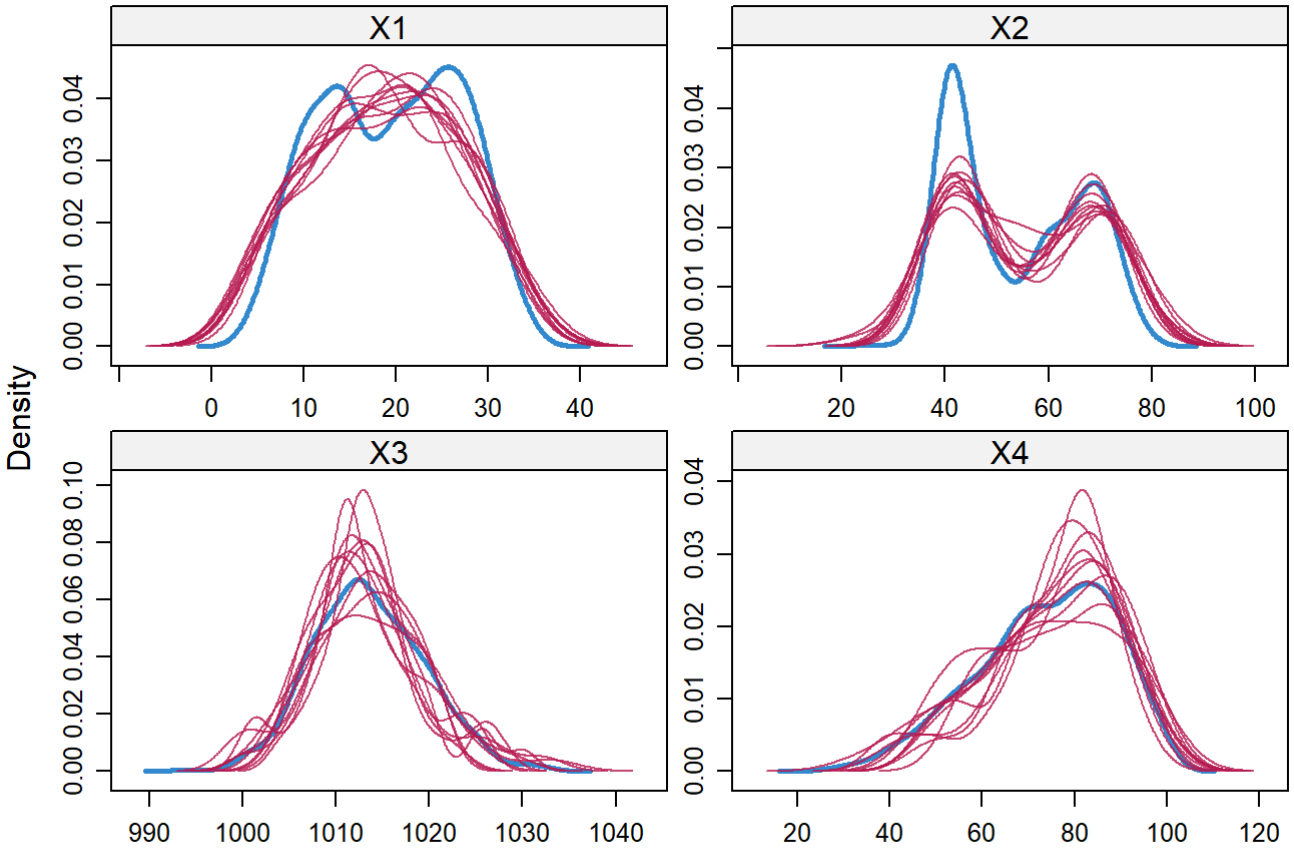


10개의 imputed set을 생성한 결과 5개의 변수가 모두 잘 얽혀 있는 것을 확인할 수 있다.

```
stripplot(imp, pch = 20, cex = 1.2)
```




```
densityplot(imp, scales = list(relation = 'free'), layout = c(2,2))
```

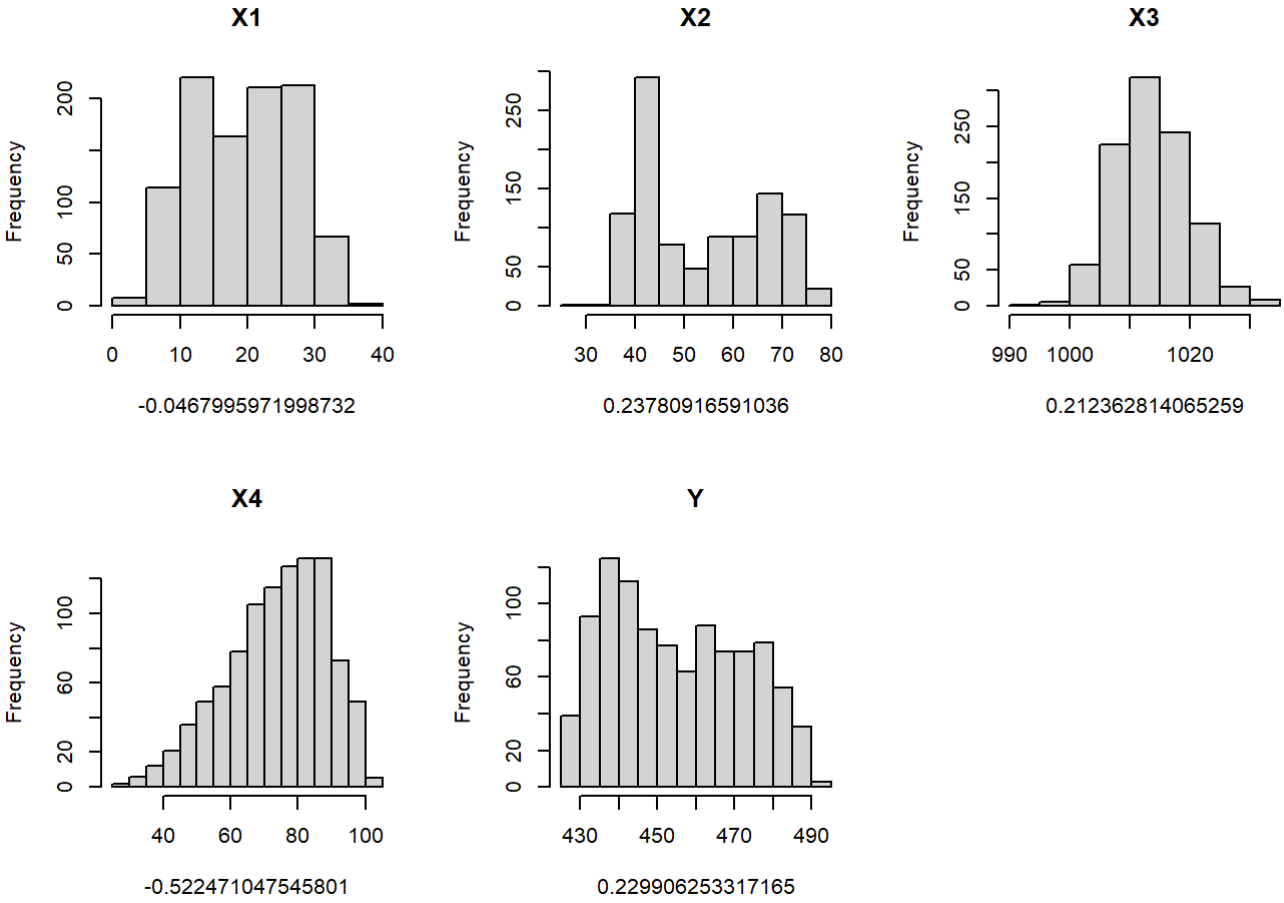


Imputed data가 observed data에 특이값 혹은 위화감 없이 잘 녹아들었다.

2) Transformation

Transformation이 필요한 변수가 있는지 확인하기 위해 첫 번째 imputed set을 사용하였다.

```
imp.dat0 = complete(imp, 1)

par(mfrow = c(2,3))
for (j in c('X1', 'X2', 'X3', 'X4', 'Y'))
{
  hist(imp.dat0[,j], main = j, xlab = skewness(imp.dat0[,j]))
}
```



한쪽으로 심하게 치우친 그래프가 없고, 모든 변수의 왜도가 0에 가까우므로 transformation이 불필요하다.

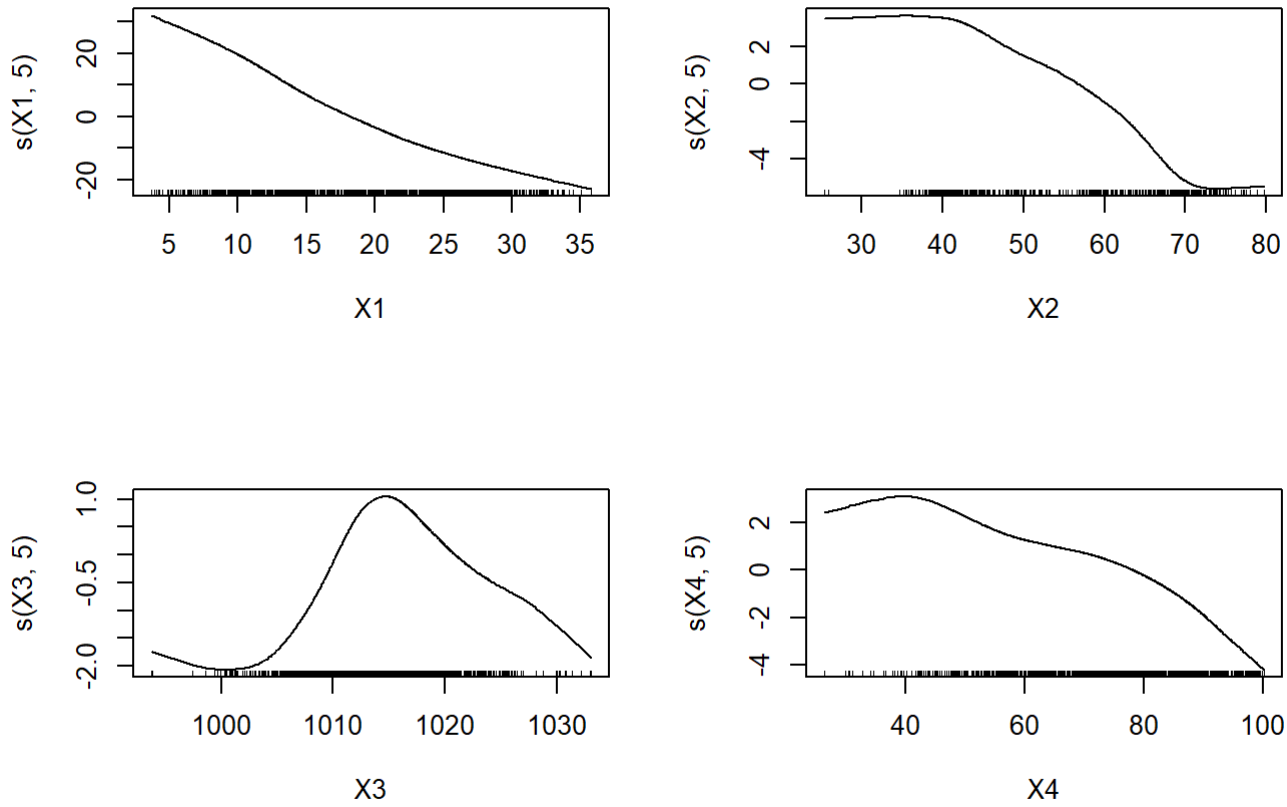
2. Modelling

계속해서 첫 번째 imputed set을 사용하였다.

GAM

모든 변수를 자유도가 5인 smoothing spline으로 적합시켜 X변수와 Y변수 사이의 관계를 관찰해보았다.

```
fit1.0 = gam(Y ~ s(X1, 5) + s(X2, 5) + s(X3, 5) + s(X4, 5), data = imp.dat0)
par(mfrow = c(2,2))
plot(fit1.0)
```

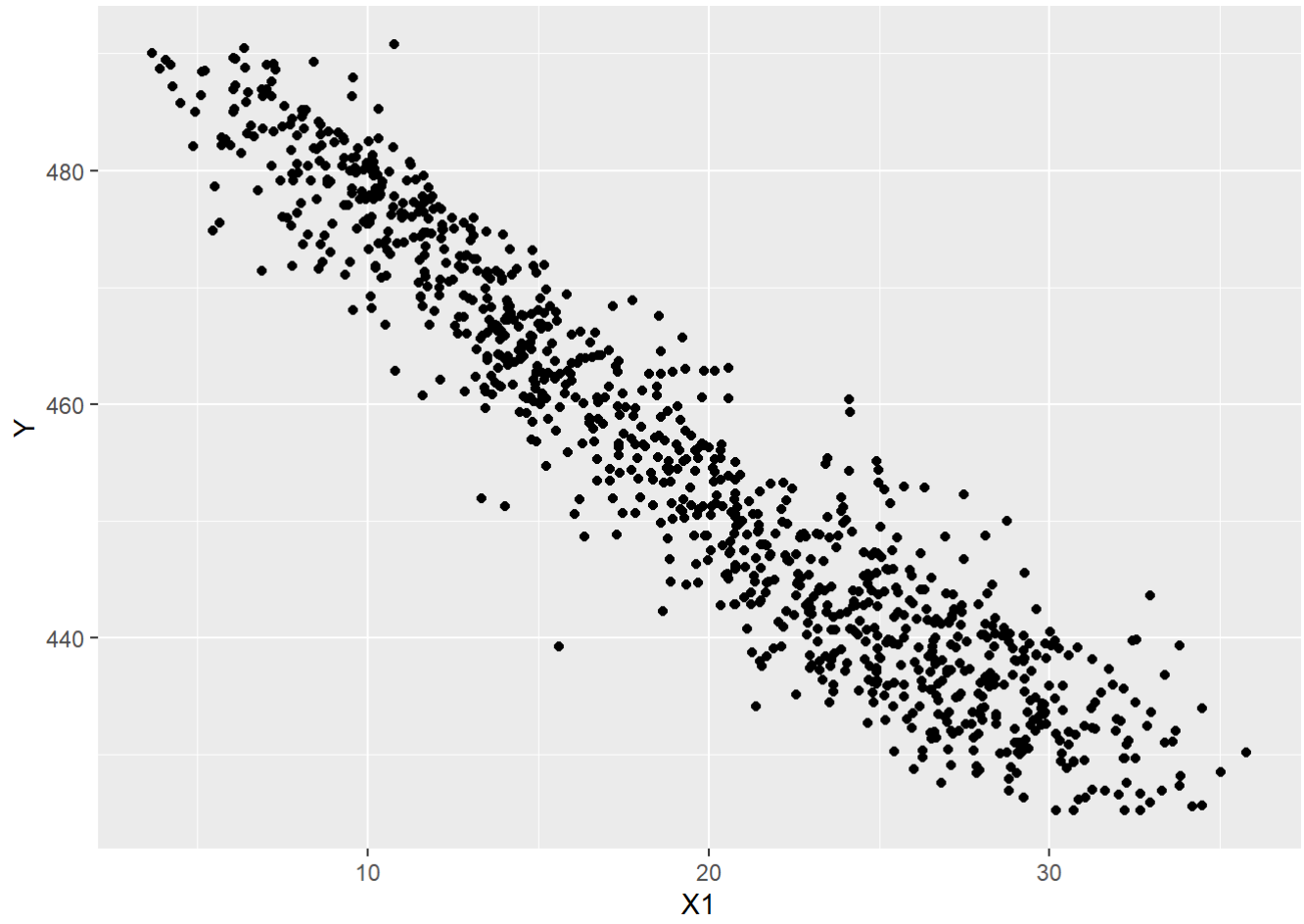


X1은 선형, X2, X4는 선형 혹은 지수함수, X3는 이차 혹은 삼차함수의 형태로 보인다. 또한 summary(fit1.0)을 확인해본 결과 모든 변수가 유의하게 나타났다.

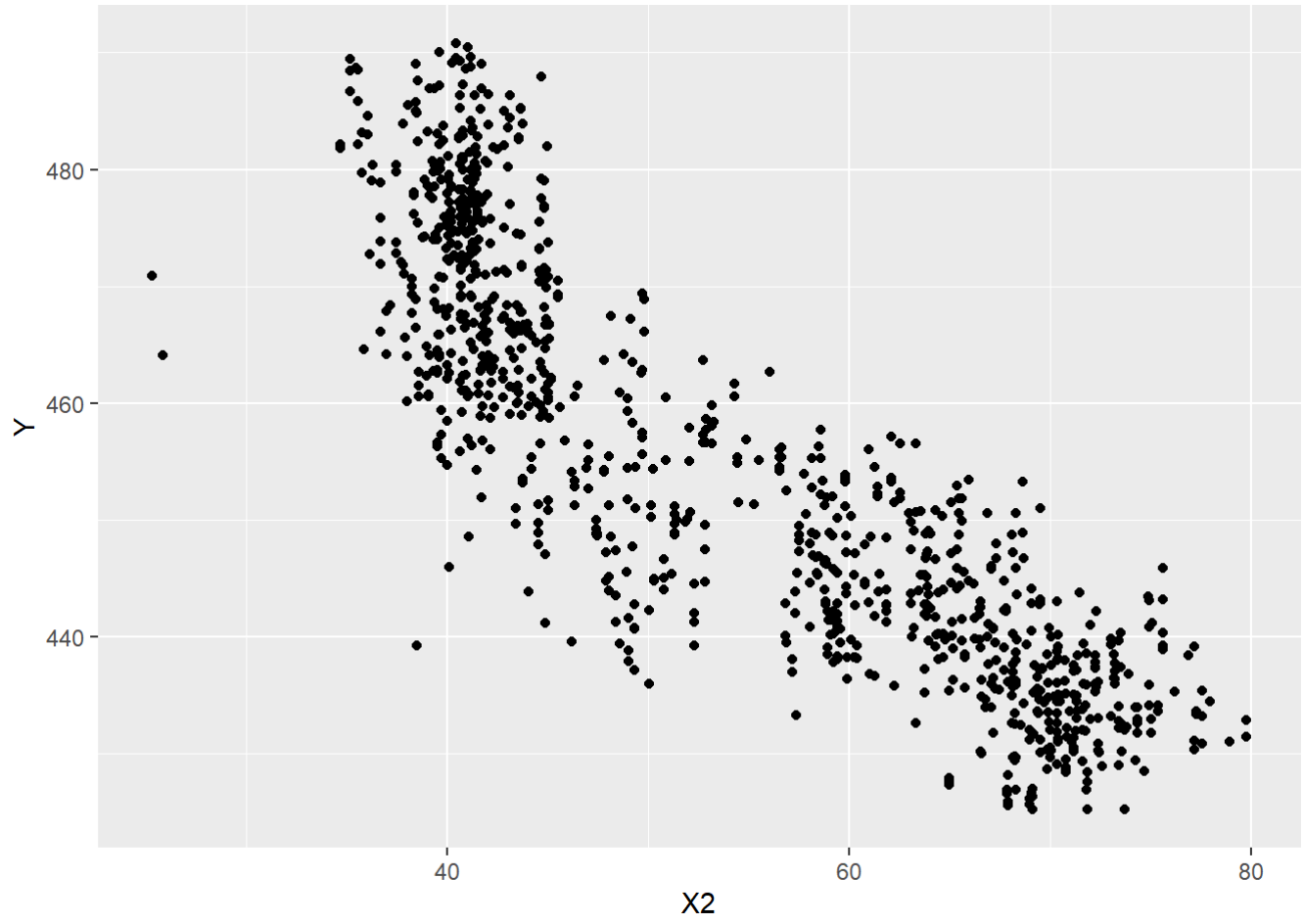
ggplot

산점도를 그려, GAM에서 관찰한 형태가 적합한지 이차적으로 확인하였다.

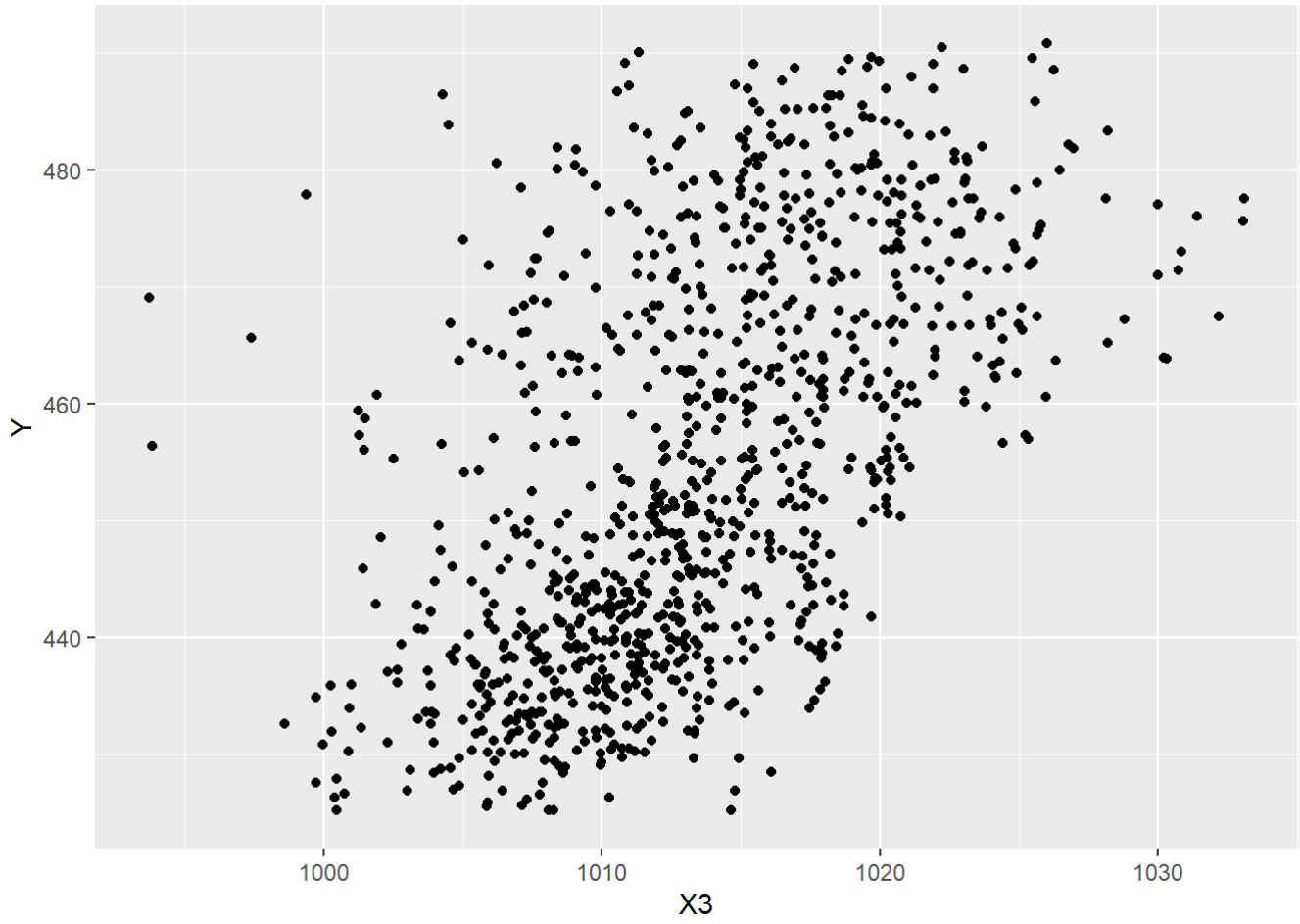
```
imp.dat0 %>% ggplot(aes(x = X1, y = Y)) + geom_point()
```



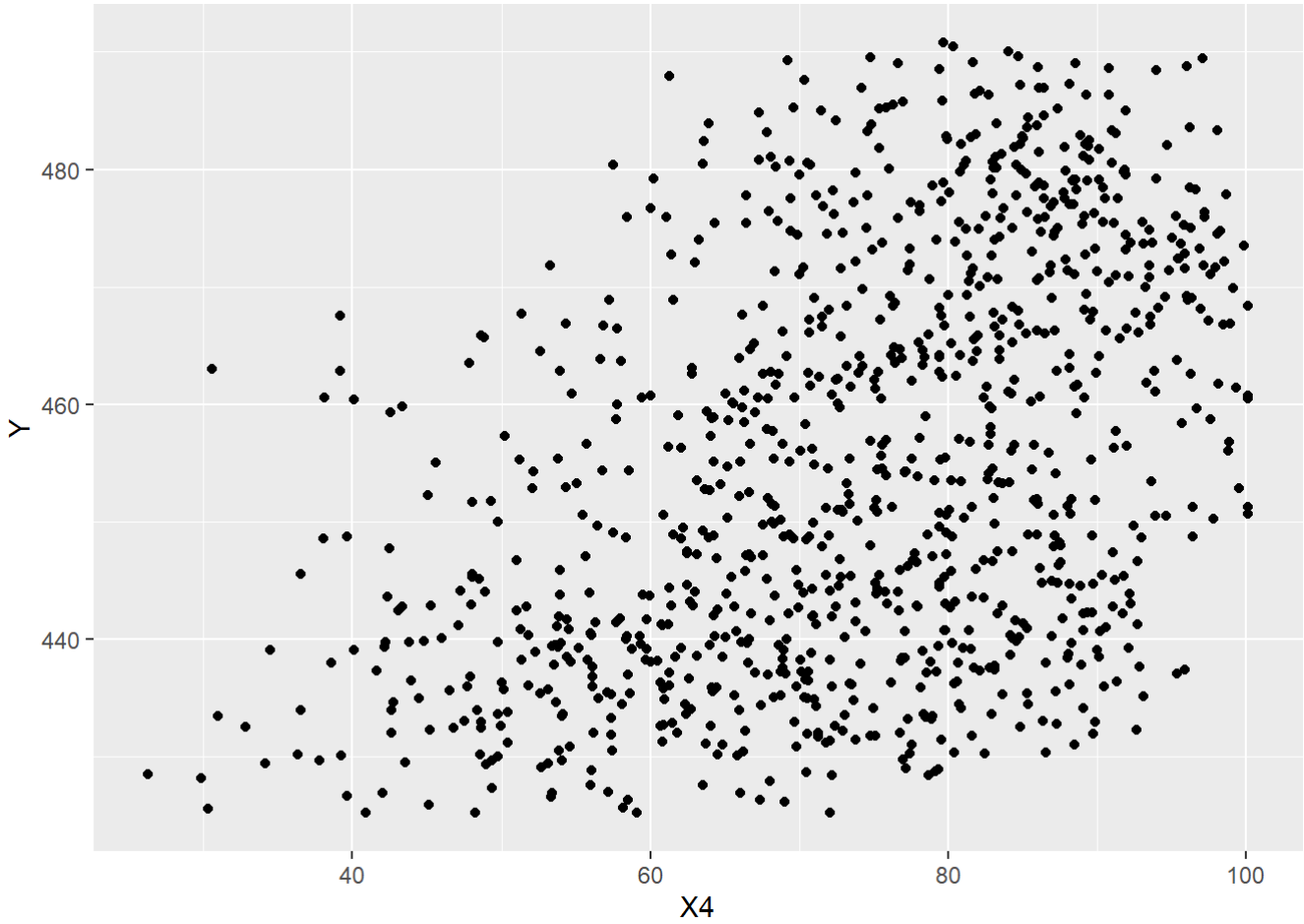

```
imp.dat0 %>% ggplot(aes(x = X2, y = Y)) + geom_point()
```



```
imp.dat0 %>% ggplot(aes(x = X3, y = Y)) + geom_point()
```




```
imp.dat0 %>% ggplot(aes(x = X4, y = Y)) + geom_point()
```



X1은 산점도에서도 Y와 명확한 선형관계가 나타나 있고, X2 또한 지수적으로 감소하는 형태다. 반면에 X3, X4는 Y와의 관계가 불명확하다.

따라서 위의 fit1.0과 산점도에서 공통적으로 드러나는 X1, X2의 형태를 명시한 채, X3과 X4는 어떤 형태로 적합시킬지 anova test를 시행하였다.

```
fit1.1 = gam(Y ~ X1 + log(X2) + s(X3, 5) + s(X4, 5), data = imp.dat0)
fit1.2 = gam(Y ~ X1 + log(X2) + I(X3^2) + X3 + s(X4, 5), data = imp.dat0)
anova(fit1.1, fit1.2) # fit1.1
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ X1 + log(X2) + s(X3, 5) + s(X4, 5)
## Model 2: Y ~ X1 + log(X2) + I(X3^2) + X3 + s(X4, 5)
##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
## 1       987      19964
## 2       990      20116 -2.9997  -152.02  0.05714 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

P값이 유의하므로 X3은 자유도가 5인 smoothing spline을 사용하였다.

```
fit1.3 = gam(Y ~ X1 + log(X2) + s(X3, 5) + X4, data = imp.dat0)
fit1.4 = gam(Y ~ X1 + log(X2) + s(X3, 5) + log(X4), data = imp.dat0)
anova(fit1.1, fit1.3, fit1.4) # fit1.3
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ X1 + log(X2) + s(X3, 5) + s(X4, 5)
## Model 2: Y ~ X1 + log(X2) + s(X3, 5) + X4
## Model 3: Y ~ X1 + log(X2) + s(X3, 5) + log(X4)
##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
## 1       987      19964
## 2       991      20090 -4.0001 -126.912   0.1796
## 3       991      20119  0.0000  -28.819
```

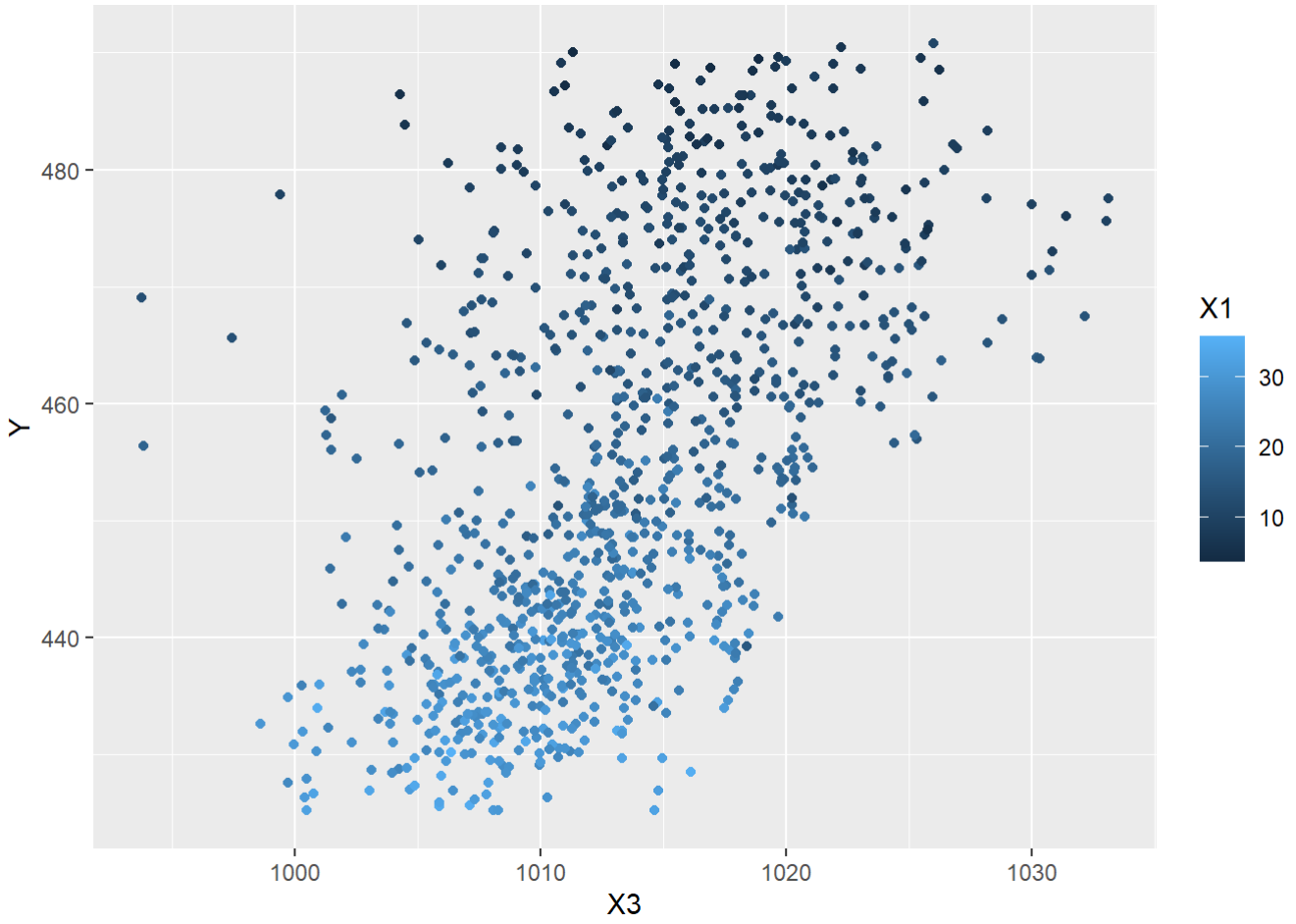
P값이 유의하지 않으므로 X4는 선형을 사용하였다.

Interaction

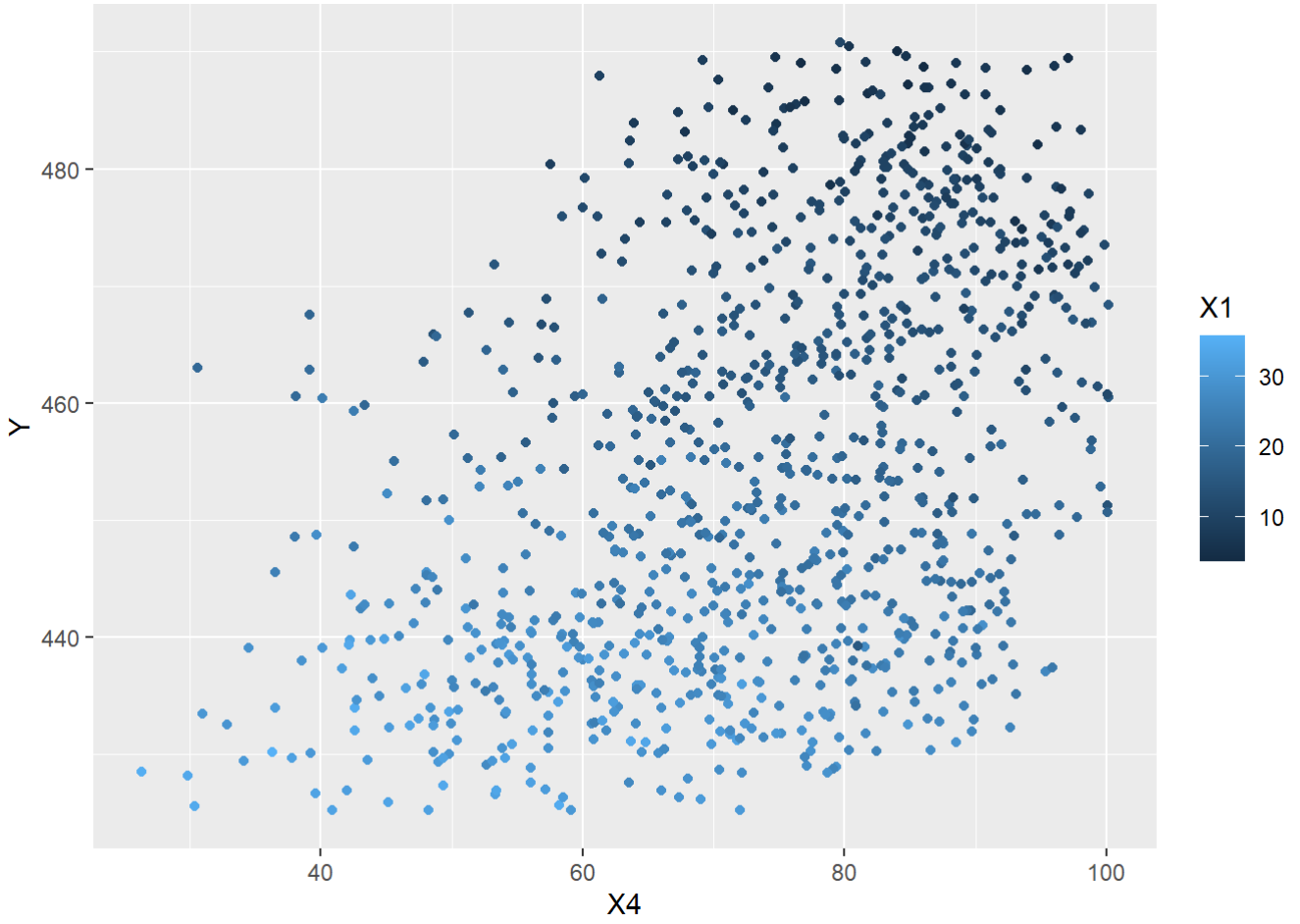
Interaction은 필자의 주관을 개입하여 다소 임의적으로 설정하였다.

압력(X3)과 습도(X4)가 온도(X1)에 영향을 받는다는 배경지식을 이용하여 해당 interaction을 고려해보았다.

```
imp.dat0 %>% ggplot(aes(x = X3, y = Y, col = X1)) + geom_point()
```




```
imp.dat0 %>% ggplot(aes(x = X4, y = Y, col = X1)) + geom_point()
```



두 관계 모두에서 명확한 경향성이 확인되므로, 기존 모델(fit1.3)에 X1과 X3, X1과 X4 interaction을 추가하였다.

```
fit1.5 = gam(Y ~ X1 + log(X2) + s(X3, 5) + X4 + X1*X3 + X1*X4, data = imp.dat0)
AIC(fit1.5)
```

```
## [1] 5820.235
```

```
AIC(fit1.3)
```

```
## [1] 5858.117
```

summary(fit1.5)를 확인해본 결과 모든 변수가 유의하게 나타나며, interaction을 추가한 fit1.5의 AIC 값이 더 작으므로, 최종 모델로 fit1.5를 채택하였다.

3. Prediction

최종 모델에는 앞서 생성한 10개의 imputed set을 모두 사용하였다.

```
M = imp$m
imp.dat = vector(mode = 'list', length = M)
for (m in 1:M) imp.dat[[m]] = complete(imp, m)

p.model = function(dat) gam(Y ~ X1 + log(X2) + s(X3, 5) + X4 + X1*X3 + X1*X4, data = dat)

fit.imp = lapply(imp.dat, p.model)
```

```
Yhat = lapply(fit.imp, predict, newdata = te1)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
Yhat = matrix(unlist(Yhat), nrow(te1), M)
Yhat = apply(Yhat, 1, mean)
Y = te1$Y

# Test MSE
testMSE = mean((Y - Yhat)^2)
testMSE
```

```
## [1] 18.90676
```

이 모델의 test MSE 값은 18.90676이다.

Q2

```
tr2_0 = read.csv('pm25_tr.csv')
te2_0 = read.csv('pm25_te.csv')

data2 = rbind(tr2_0, te2_0)
# 1 ~ 1944 : train
# 1945 ~ 2064 : test
```

수월한 가공을 위해 train set과 test set을 한시적으로 병합하였다. 가공은 train set(tr2_0)을 관찰한 후 data2를 가공하는 방식으로 진행하였다.

1. Data pre-processing

1) Looking through

(1) year, month, day, hour

시간과 관련된 네 변수는 factor로 변환하였다. year 변수는 데이터셋을 통틀어 같은 값을 가지므로 사용하지 않기로 하였다.

```
data2$year = as.factor(data2$year)
data2$month = as.factor(data2$month)
data2$day = as.factor(data2$day)
data2$hour = as.factor(data2$hour)
```

(2) cbwd, lws

누적 바람 속도(lws)는 바람 방향(cbwd)에 의존하므로 일시적 바람 속도를 나타내는 변수인 tws(temporary wind speed)를 추가하였다.

```
n = dim(data2)[1]
cbwd = data2$cbwd; lws = data2$lws
tws = rep(0, n)
for (i in 2:n) tws[i] = ifelse(cbwd[i] == cbwd[i-1], lws[i] - lws[i-1], lws[i])
tws[1] = lws[1]
data2 = data2 %>% mutate(tws = tws)
```

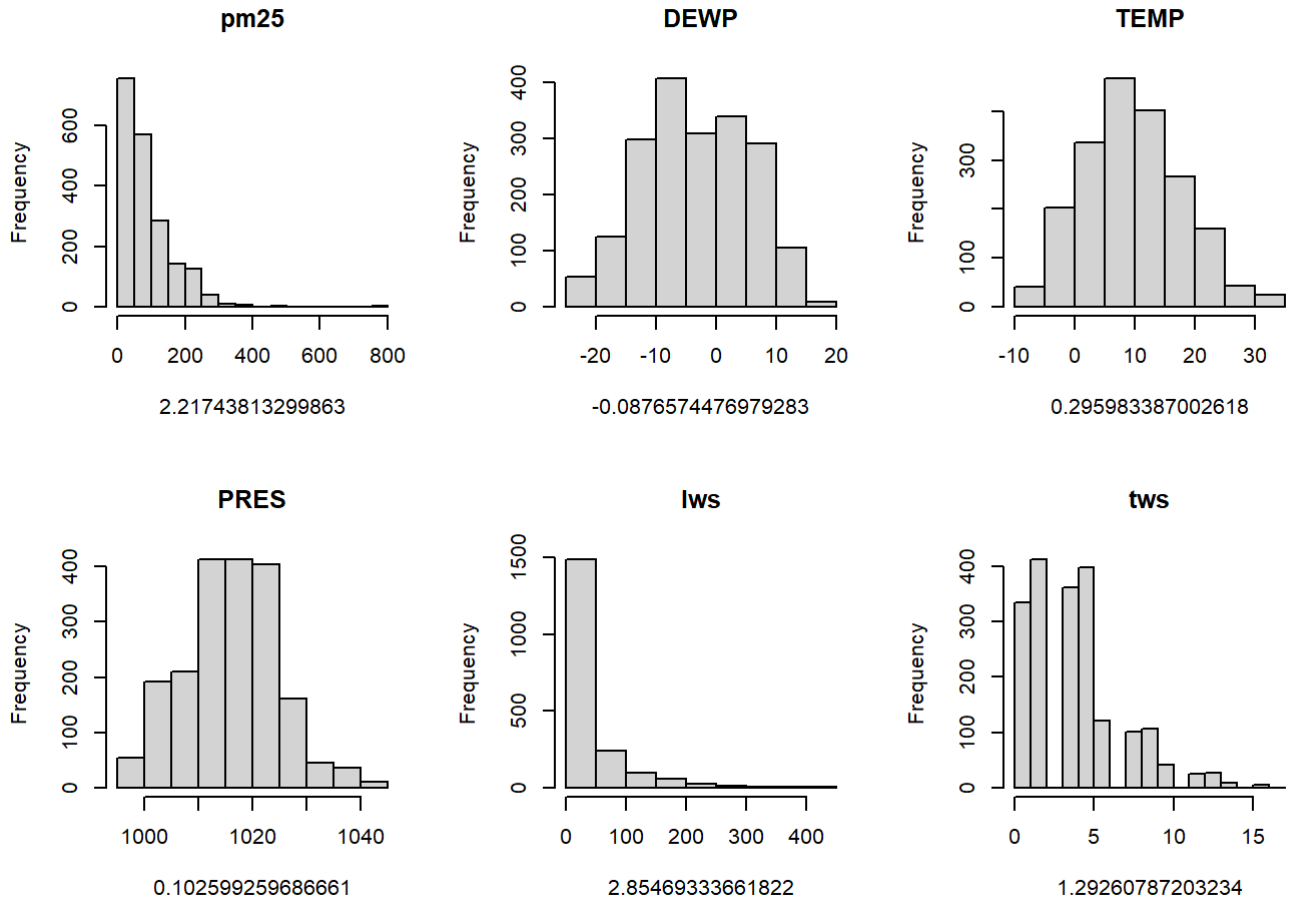
전반적인 가공을 마쳤으므로 다시 train set과 test set으로 분리하였다.

```
tr2 = data2[1:1944,]
te2 = data2[1945:2064,]
```

2) Transformation

Transformation이 필요한 연속형 변수가 있는지 확인하였다.

```
tr2.num = tr2 %>% select_if(is.numeric)
par(mfrow = c(2,3))
for (j in names(tr2.num))
{
  hist(tr2.num[,j], main = j, xlab = skewness(tr2.num[,j]))
}
```



pm25, lws, tws 변수의 그래프가 한쪽으로 치우쳐 있으며, 왜도 역시 0과 가깝지 않다. 다만 lws 변수는 누적 수치라는 점과, tws 변수는 lws 변수로부터 만들어졌다는 점을 감안해 transformation을 보류하였다. Y변수인 pm25 변수는 지수적으로 감소하고 있어, 로그를 취하여 normal하게 만드는 것이 X변수들과의 관계를 확인하는 데에 용이할 것이므로, 로그를 취한 채 모델링을 진행하였다.

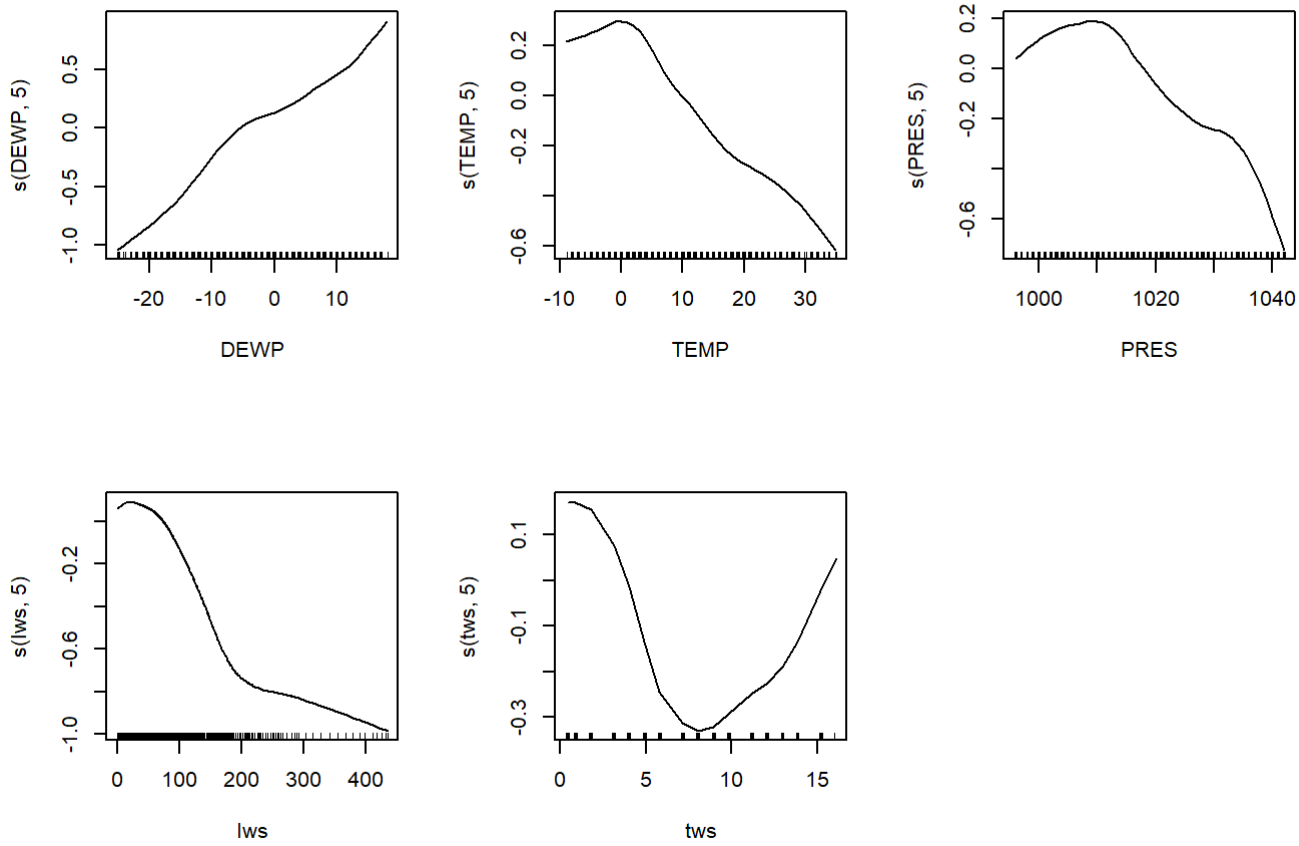
2. Modelling

(1) 연속형 변수

GAM

모든 연속형 변수를 자유도가 5인 smoothing spline으로 적합시켜 X변수와 Y변수 사이의 관계를 관찰해보았다.

```
fit2.00 = gam(log(pm25) ~ s(DEWP,5) + s(TEMP,5) + s(PRES,5) + s(lws,5) + s(tws,5), data = tr2)
par(mfrow = c(2,3))
plot(fit2.00)
```

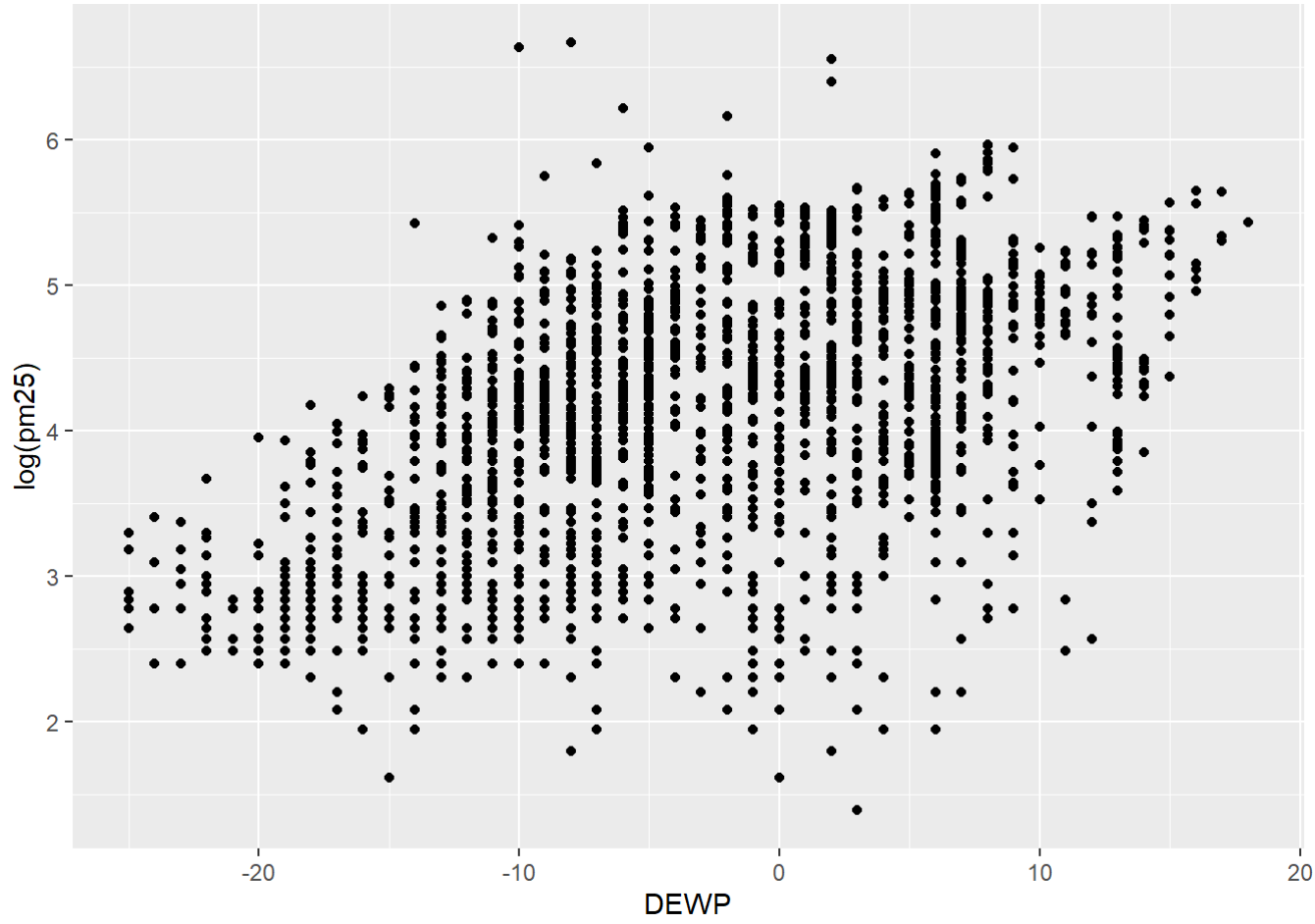


DEWP은 선형, tws는 이차함수의 형태로 보이며, 나머지 변수들은 형태가 다소 불명확하다.

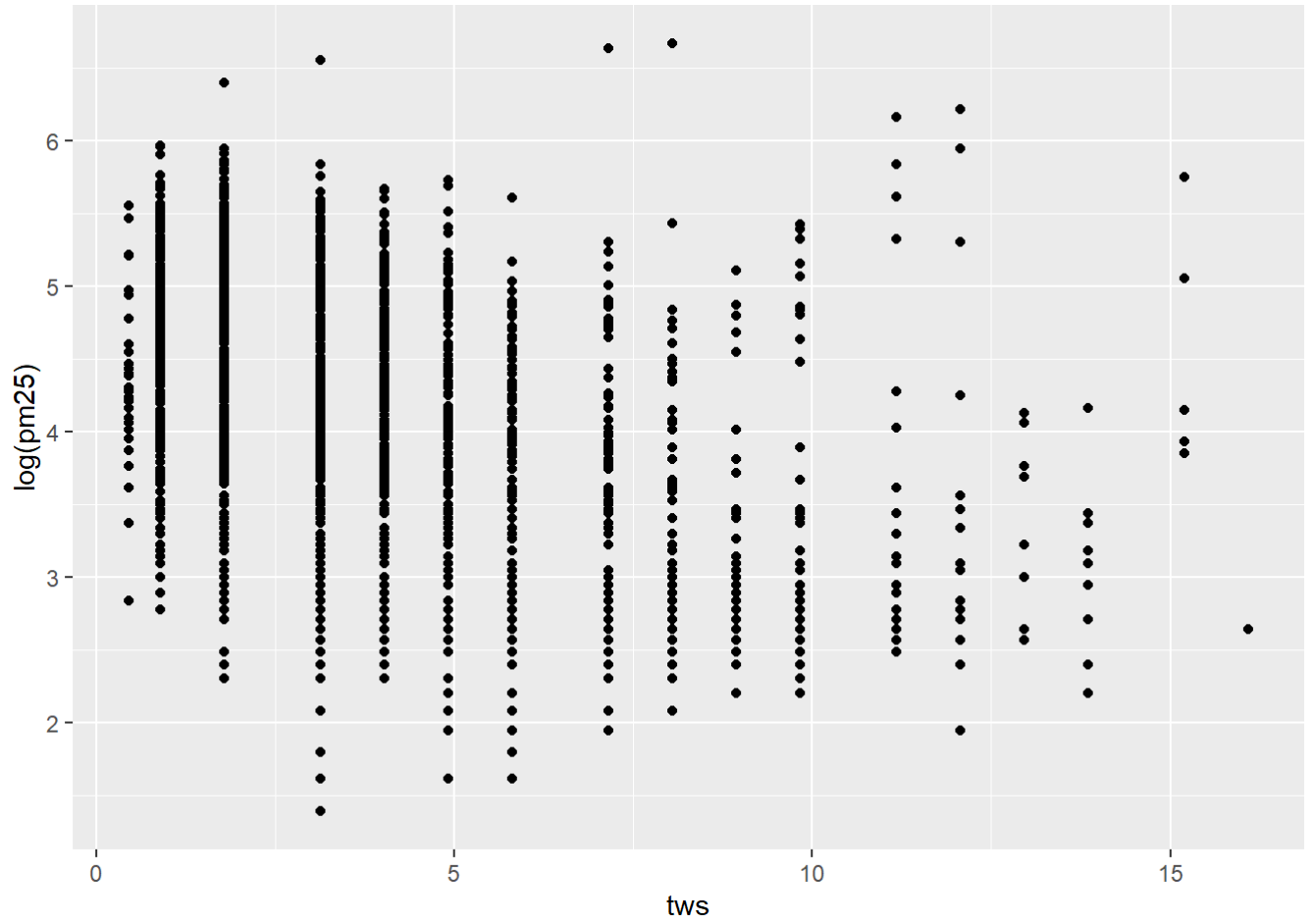
ggplot

산점도를 그려, 위의 GAM에서 관찰한 형태가 적합한지 이차적으로 확인하였다.

```
tr2 %>% ggplot(aes(x = DEWP, y = log(pm25))) + geom_point()
```




```
tr2 %>% ggplot(aes(x = tws, y = log(pm25))) + geom_point()
```



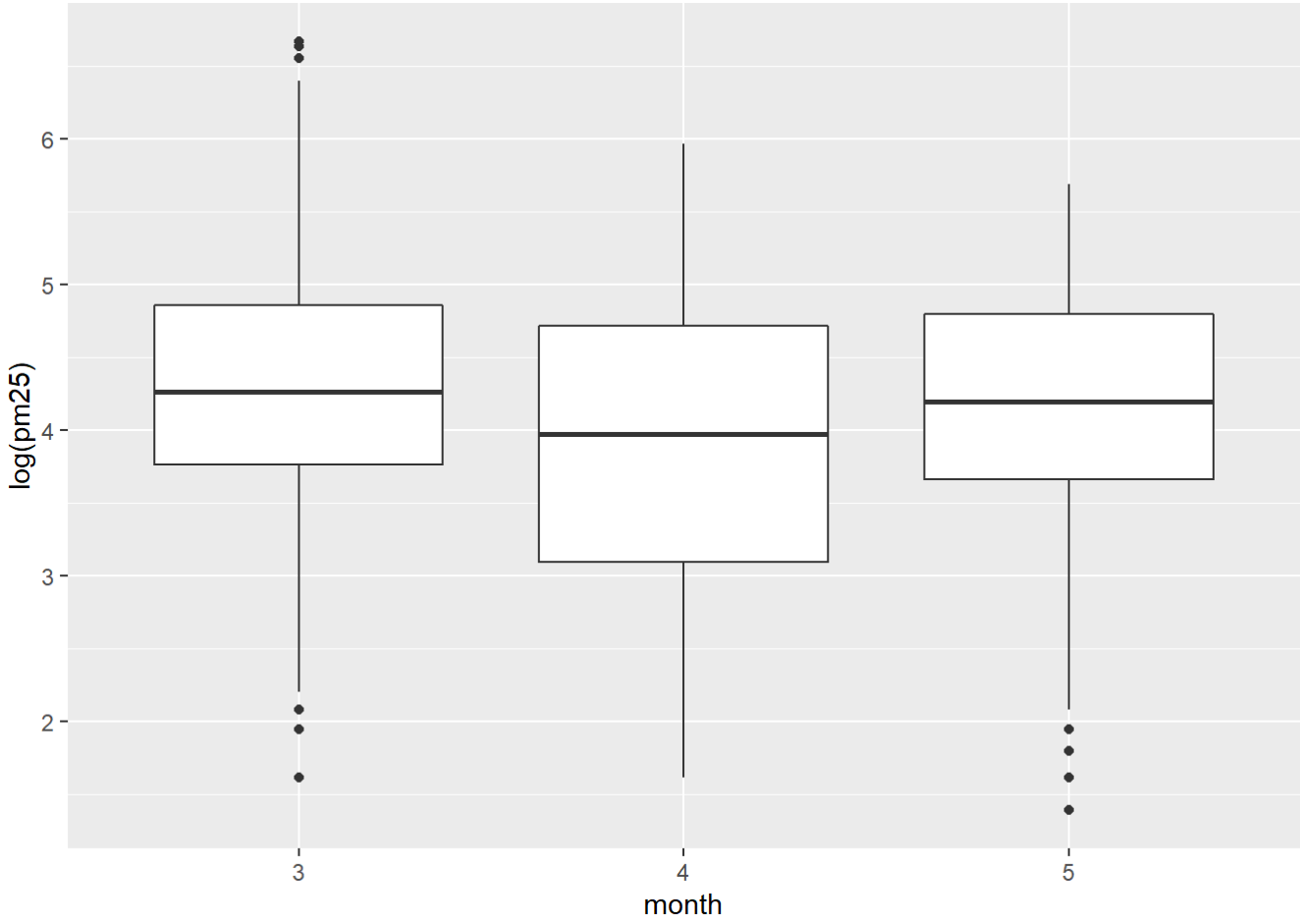
DEWP와 tws는 각각 선형, 이차함수의 경향이 존재하지만 명확하진 않다. 그 외의 변수들은 형태가 더욱 불명확하며, 생략하였다.

위의 fit2.00과 산점도를 근거로 하여 DEWP와 tws를 어떤 형태로 적합시킬지 anova test를 시행하였으며, 그 외의 연속형 변수는 모두 자유도 5의 smoothing spline으로, 아래 범주형 변수를 포함하여 적합시켰다. 결과는 범주형 변수를 다룬 이후에 첨부하였다.

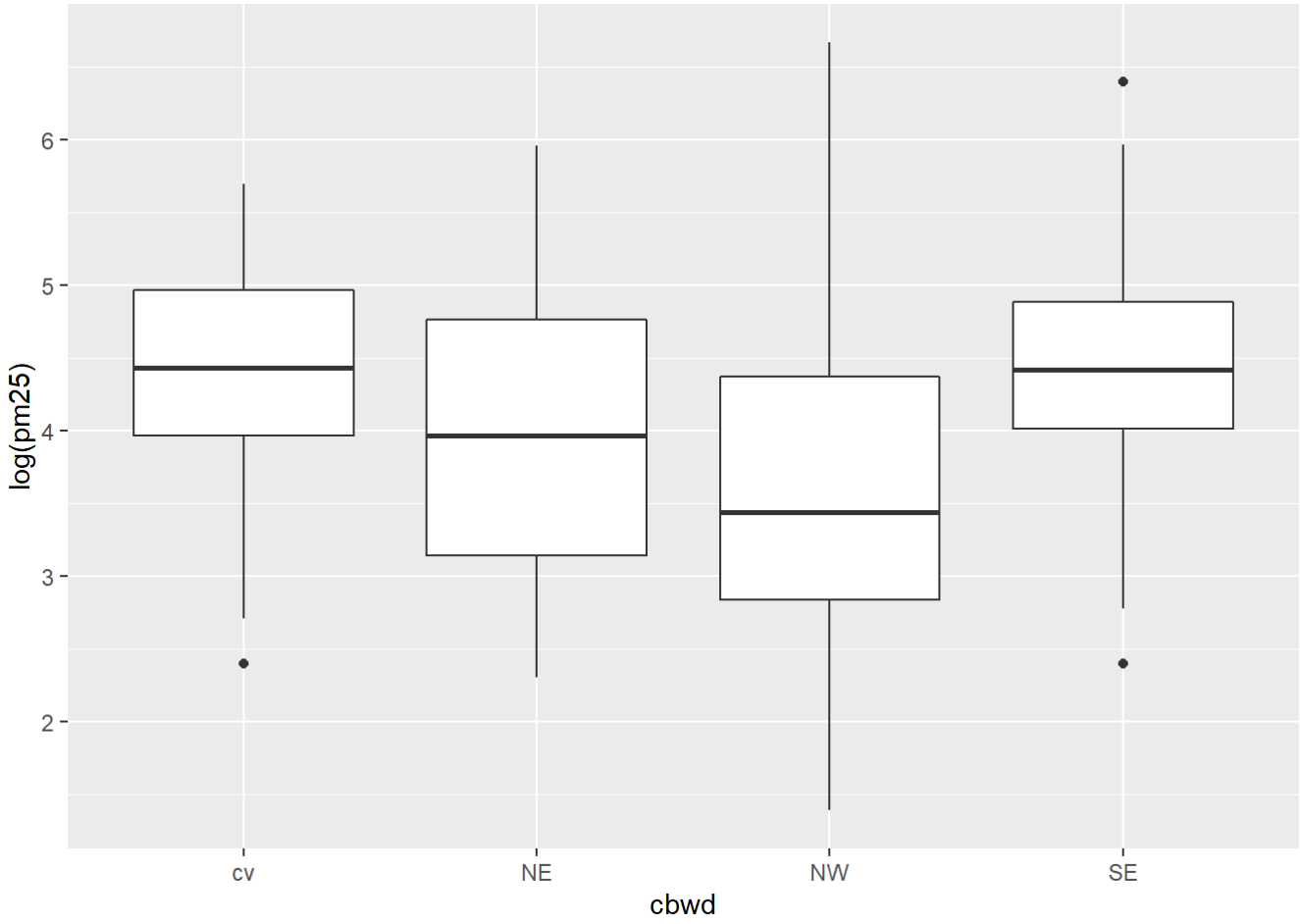
(2) 범주형 변수

시간을 나타내는 변수인 month, day, hour 변수 중에서 day, hour를 모델에 사용하려면 month 변수와 결합하여 일별 개념으로 나타내야 하는데, 그렇게 하면 하나의 일자가 하나의 factor, 즉, 범주가 되므로 month 변수만 사용하였다.

```
tr2 %>% ggplot(aes(x = month, y = log(pm25))) + geom_boxplot()
```




```
tr2 %>% ggplot(aes(x = cbwd, y = log(pm25))) + geom_boxplot()
```



통계적으로 유의한지는 아직 알 수 없지만, 범주별로 차이가 존재함을 확인하였다.

```
fit2.0 = gam(log(pm25) ~ month + cbwd + s(DEWP,5) + s(TEMP,5) + s(PRES,5) + s(Iws,5) + s(tws,5
), data = tr2)
fit2.1 = gam(log(pm25) ~ month + cbwd + DEWP + s(TEMP,5) + s(PRES,5) + s(Iws,5) + I(tws^2) + tw
s, data = tr2)
fit2.2 = gam(log(pm25) ~ month + cbwd + DEWP + s(TEMP,5) + s(PRES,5) + s(Iws,5) + s(tws,5), dat
a = tr2)
fit2.3 = gam(log(pm25) ~ month + cbwd + s(DEWP,5) + s(TEMP,5) + s(PRES,5) + s(Iws,5) + I(tws^2)
+ tws, data = tr2)
anova(fit2.0, fit2.1, fit2.2, fit2.3) # fit2.0
```

```
## Analysis of Deviance Table
##
## Model 1: log(pm25) ~ month + cbwd + s(DEWP, 5) + s(TEMP, 5) + s(PRES,
##     5) + s(Iws, 5) + s(tws, 5)
## Model 2: log(pm25) ~ month + cbwd + DEWP + s(TEMP, 5) + s(PRES, 5) + s(Iws,
##     5) + I(tws^2) + tws
## Model 3: log(pm25) ~ month + cbwd + DEWP + s(TEMP, 5) + s(PRES, 5) + s(Iws,
##     5) + s(tws, 5)
## Model 4: log(pm25) ~ month + cbwd + s(DEWP, 5) + s(TEMP, 5) + s(PRES,
##     5) + s(Iws, 5) + I(tws^2) + tws
##   Resid. Df Resid. Dev       Df Deviance  Pr(>Chi)
## 1      1913     681.39
## 2      1920     690.55 -7.00016  -9.1574 0.0005676 ***
## 3      1917     687.60  3.00026   2.9481 0.0406356 *
## 4      1916     684.29  0.99964   3.3050 0.0023166 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Anova test 결과가 유의하므로 연속형 변수는 모두 자유도가 5인 smoothing spline을 채택하였다. 또한 summary(fit2.0)를 확인해본 결과 모든 변수가 유의하게 나타났다.

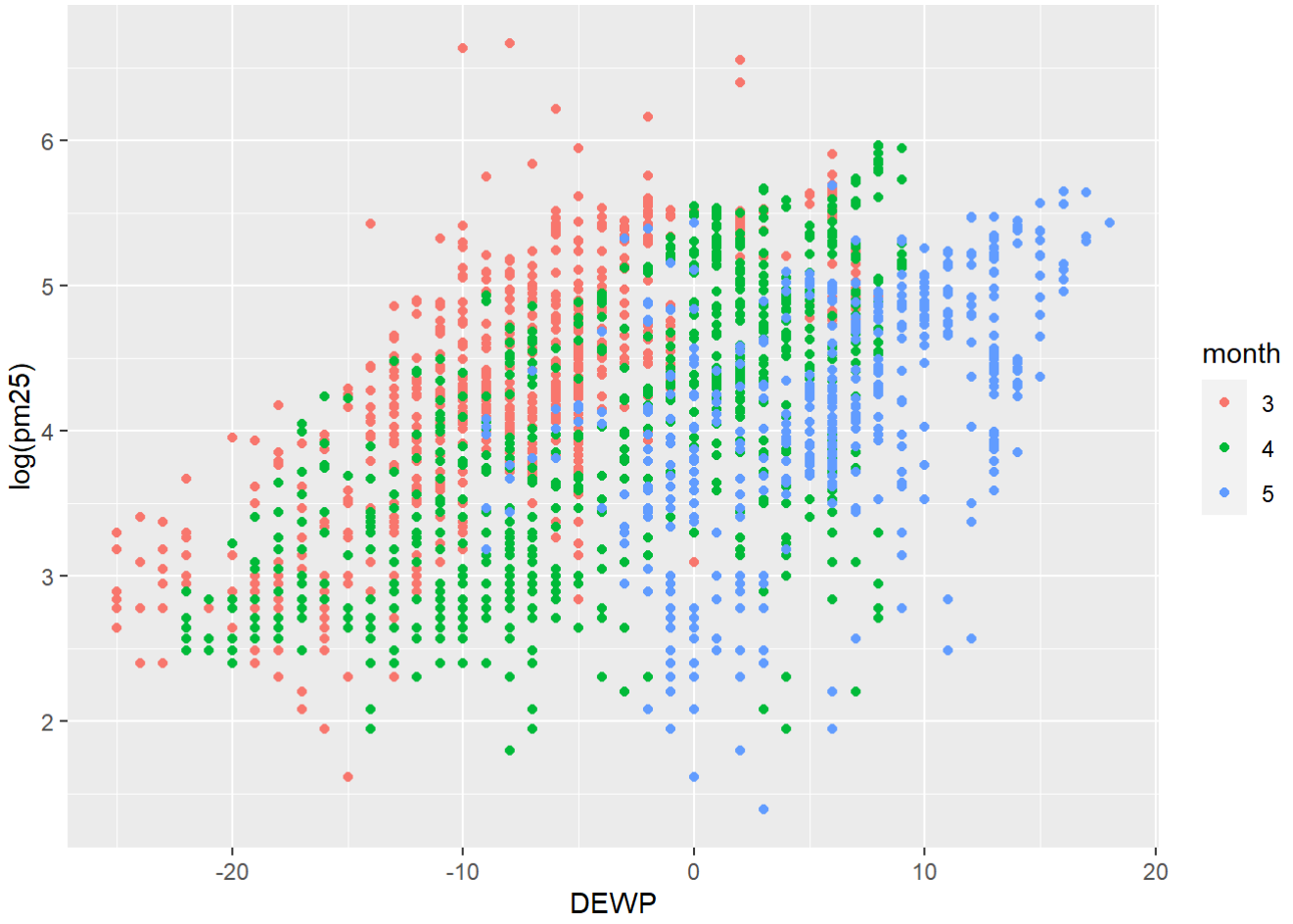
Interaction

Interaction은 필자의 주관을 개입하여 다소 임의적으로 설정하였다.

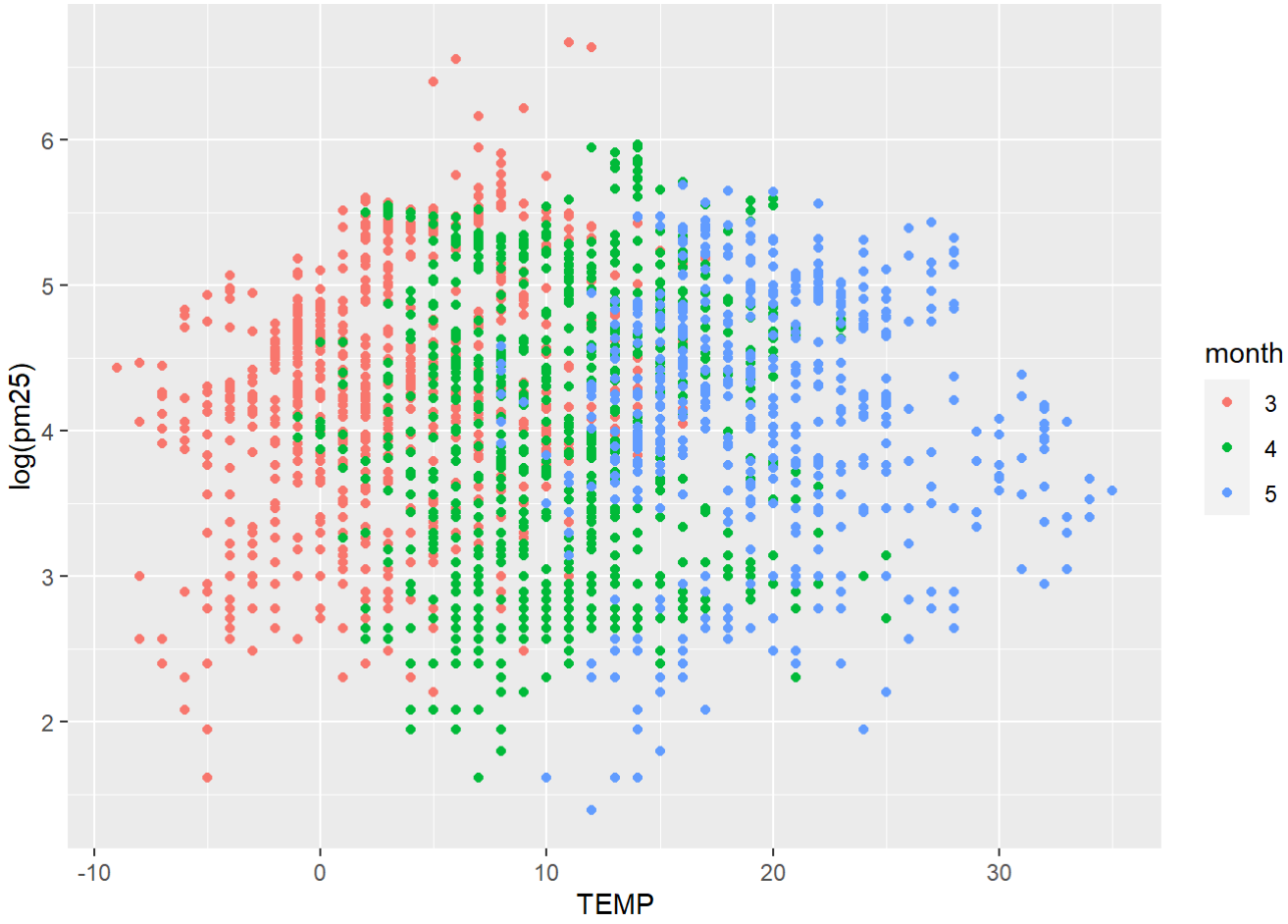
(1) month와 다른 변수들

다수의 변수들이 시간을 나타내는 month 변수에 영향을 받을 것이라고 생각하여 해당 interaction을 고려해보았다.

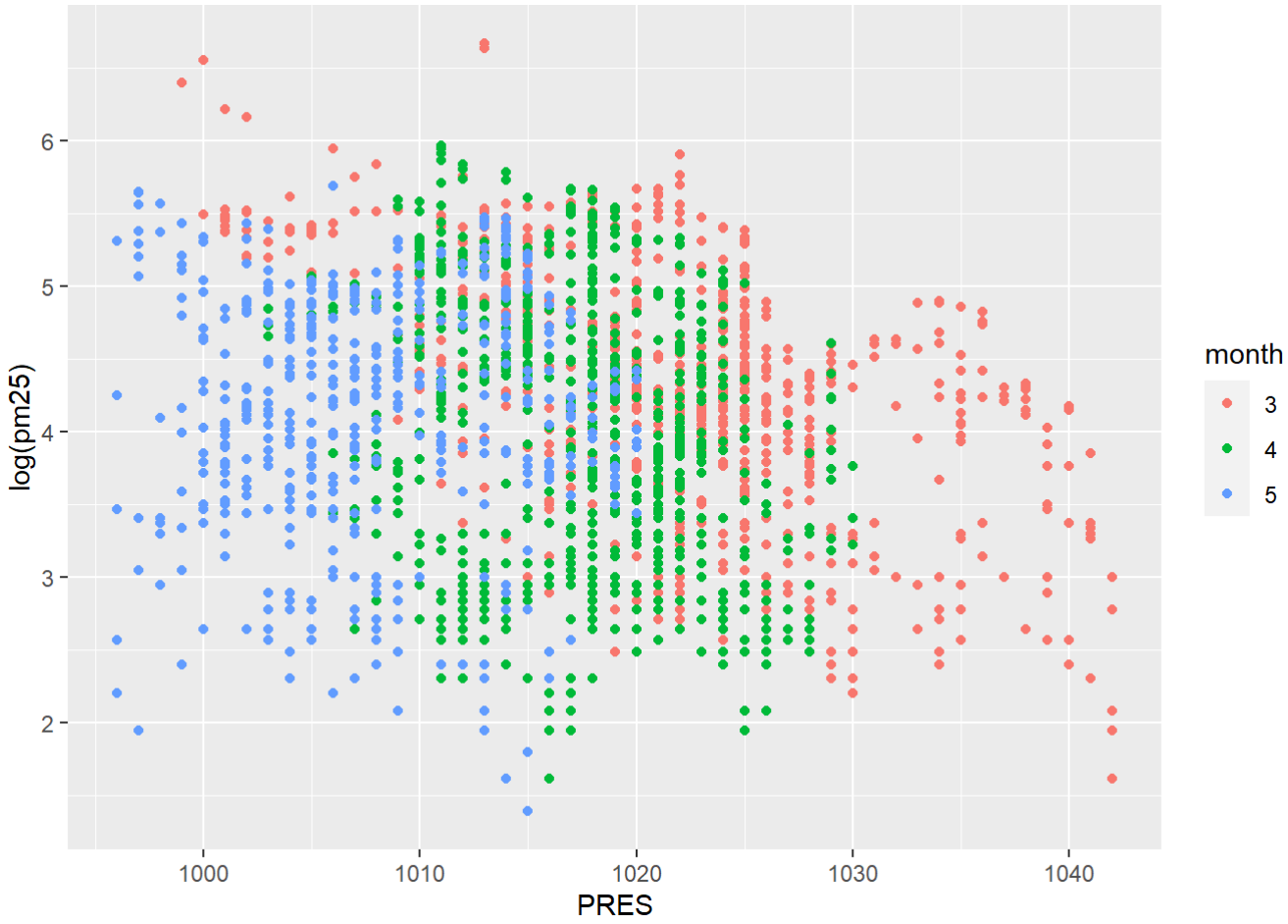
```
tr2 %>% ggplot(aes(x = DEWP, y = log(pm25), col = month)) + geom_point()
```




```
tr2 %>% ggplot(aes(x = TEMP, y = log(pm25), col = month)) + geom_point()
```



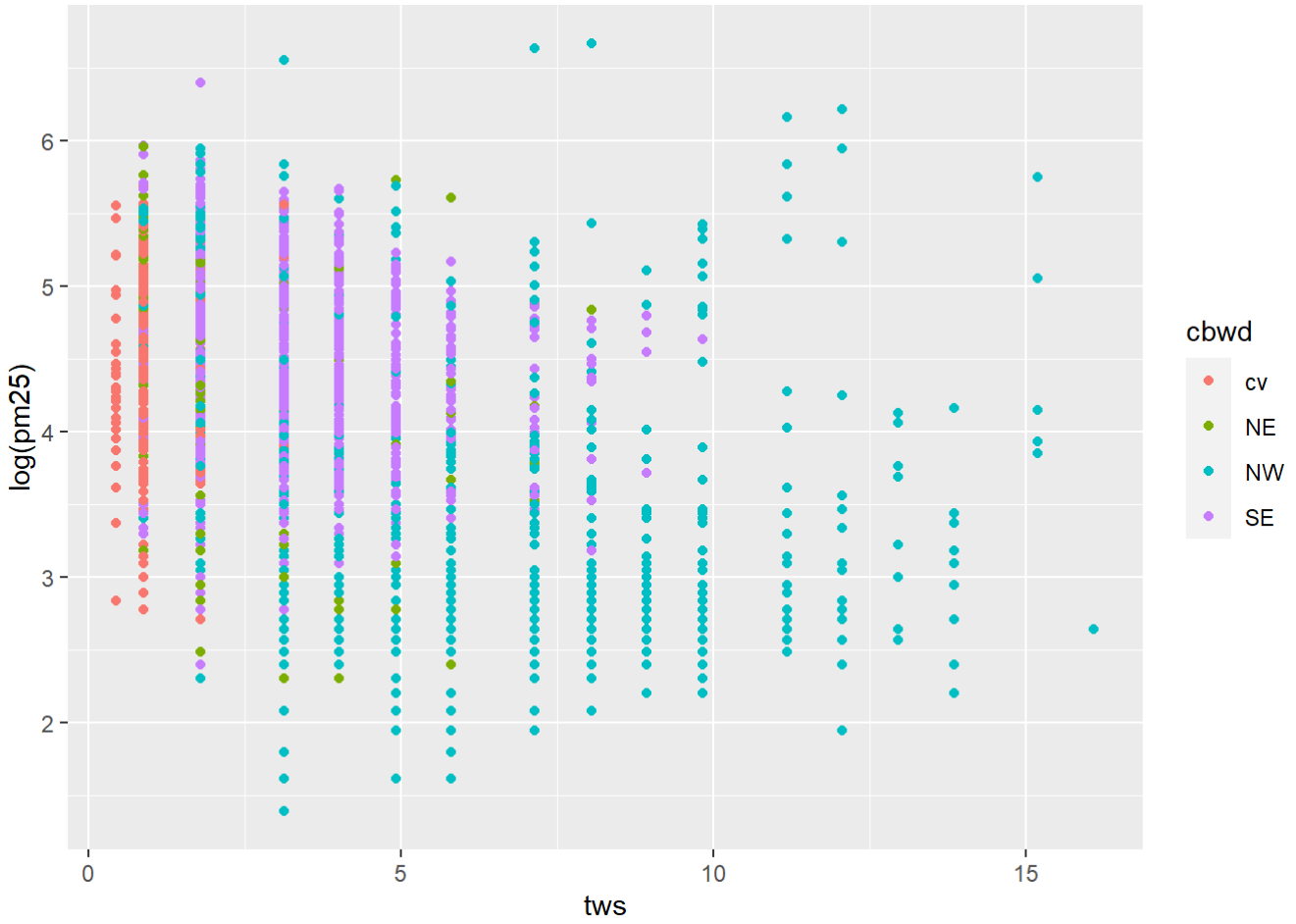

```
tr2 %>% ggplot(aes(x = PRES, y = log(pm25), col = month)) + geom_point()
```



세 변수 DEWP, TEMP, PRES가 month와의 경향성이 가장 명확하게 나타났다.

(2) 바람 관련 변수들

```
tr2 %>% ggplot(aes(x = tws, y = log(pm25), col = cbwd)) + geom_point()
```



바람과 관련된 변수들인 lws, tws, cbwd 중 tws와 cbwd의 경향성이 가장 명확하게 나타났다.

위로부터 기존 모델(fit2.0)에 DEWP와 month, TEMP와 month, PRES와 month, tws와 cbwd의 interaction을 추가하였다.

```
fit2.4 = gam(log(pm25) ~ month + cbwd +s(DEWP,5) + s(TEMP,5) + s(PRES,5) + s(lws,5) + s(tws,5)
 +
               DEWP*month + TEMP*month + PRES*month + tws*cbwd, data = tr2)
AIC(fit2.4)
```

```
## [1] 3445.87
```

```
AIC(fit2.0)
```

```
## [1] 3542.806
```

summary(fit2.4)를 확인해본 결과 모든 변수가 유의하게 나타나며, interaction을 추가한 fit2.4의 AIC 값이 더 작으므로, 최종 모델로 fit2.4를 채택하였다.

3. Prediction

```
Y2 = te2$pm25
Yhat2 = exp(predict(fit2.4, te2))
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
# Test MSE
testMSE2 = mean((Y2 - Yhat2)^2)
testMSE2
```

```
## [1] 1466.533
```

이 모델의 Test MSE 값은 1466.533이다.