함수추정 및 실습 HW3

김보창

```
set.seed(123)
```

실행할때마다 동일한 결과가 나오도록 set.seed를 통해 시드를 설정해준다.

Q1

1-(a)

```
mykrig1<-function(nd, ne, xx, yy, ex, bw)
{
# (1)
  full <- matrix(0, nd, nd)
  full[lower.tri(full)] <- dist(xx)
  dist1 <- full + t(full)
  cc <- exp( - dist1^2/bw^2)
# (2)
  dvector <- function(exs, xx, bw)
  {
    dv <- exp( - (exs - xx)^2/bw^2)
    return(dv)
  }
  dmat <- apply(matrix(ex, nrow = 1), 2, dvector, xx = xx,
    bw = bw)
# (3)
  ccr <- solve(cc)
  ccs <- sum(apply(ccr, 2, sum))
  chi <- (-2 * (1 - apply((ccr %*% dmat), 2, sum)))/ccs
  chimat <- matrix(rep(chi, times = nd), ncol = ne,
    byrow = T)
```

```
# (4)
  ww <- ccr %*% (dmat - 0.5 * chimat)
  ey <- t(ww) %*% yy
# (5)
  return(ey)
}
```

먼저, 교재의 4장 (E) 부분에 해당하는 코드를 이용하여 ordinary kriging을 구해주는 함수를 구현하였다.

여기서 kriging에 사용하는 theoretical correlogram은

gaussian correlogram으로, 다음과 같다.

$Correlo(r) = \exp(-(\frac{r}{\delta})^2)$

위 함수는 데이터인 xx, yy와 데이터의 개수인 nd, 추정할 데이터의 위치인 ex와 그 개수인 ne, 그리고 위의 theoretical correlogram에서 사용할 δ를 받아 ex의 위치에서 ordinary kriging된 결과를 리턴해주는 함수이다.

cross validation 값의 정의는 아래와 같다.

$$CV[\hat{m}(x)] = \frac{\sum_{k=1}^{n}\left(Y_k - \hat{m}^{-k}\left(X_k\right)\right)^2}{n}$$

문제에서 원하는것은, 이 cross validation을 구할때 양 끝점을 제외하는 경우는 사용하지 않을것이므로, 위의 정의를 다시 아래와 같이 바꿔서 사용할 것이다.

$$CV[\hat{m}(x)] = \frac{\sum_{k=2}^{n-1}\left(Y_k - \hat{m}^{-k}\left(X_k\right)\right)^2}{n}, X_1 \leq X_2 \ldots \leq X_n$$

이제, 위 정의대로 cross validation을 구해주는 함수를 짜자.

cross validation을 구할때, 양 끝점을 제외해야하므로 X들이 sorting된 상태여야하고, 따라서 sorting을 해주었다.

아래 함수는 xdata와 ydata, 그리고 bandwidth들을 받아서 각 bandwidth에 해당하는 CV값들을 리턴해준다.

```
cv_ord_krig<-function(xdata, ydata, bws)
{
Processing math: 100%  <- length(ydata)
```

```
    ydata <- ydata[order(xdata)]
    xdata <- xdata[order(xdata)]
    #sort by ascending order. order of ydata, xdata is important!!


    get_krig_cv <- function(bw, x1, y1)
    {
        cv <- 0
        nd <- length(y1)

        if(nd < 3) return(NULL) # we can't get cv

        for(i in seq(from = 2, to = nd-1))
        {
            xdata <- x1[-i]
            ydata <- y1[-i]
            remain_xdata <- x1[i]
            remain_ydata <- y1[i]
            estim_y <- mykrig1(nd-1, 1, xdata, ydata, as.vector(remain_xdata), bw)
            cv <- cv + sum((remain_ydata - estim_y)**2)
        }

        cv <- cv / (nd - 2)
        return(cv)
    }


    cvlst <- lapply(as.list(bws),get_krig_cv, x1 = xdata, y1 = ydata)
    cvlst <- unlist(cvlst)
    return(cvlst)
}
```

아래함수는 bandwidth와 cv array를 받아서 그래프를 출력해주는 함수다.

```
plot_cv <- function(bandwidth_array, cv_array, description = "")
```

Processing math: 100%

```
    par(mfrow = c(1, 1))
    plot(bandwidth_array, cv_array, type = "n",
     xlab = "bandwidth", ylab = "CV", main = description)
    points(bandwidth_array, cv_array, pch = 1, cex = 0.5)
    #lines(smtparam_array, cv_array, lwd = 1)
    pcvmin <- seq(along = cv_array)[cv_array == min(cv_array, na.rm = TRUE)]
    spancv <- bandwidth_array[pcvmin]
    cvmin <- cv_array[pcvmin]
    points(spancv, cvmin, cex = 1, pch = 15, col = "red")
}
```

plot graph함수는 xdata, ydata, smtparam을 받아 xdata의 위치에서 추정된 값을 리턴하는 estimate_func을 받아서, 원래 데이터를 점으로, estimate된 결과를 파란 선으로 그려준다.

```
plot_graph <- function(estimate_func, xdata, ydata, smtparam, description = "")
{
    par(mfrow=c(1,1))
    estimated_y <- estimate_func(xdata, ydata, smtparam)
    plot(xdata, ydata , xlab = "X", ylab = "Y", type = "p", main = description, sub = sprintf("using smtparam : %
f", smtparam))
    lines(xdata, estimated_y, col = "blue")
}
```

위 함수를 사용하기 위해 다음과 같이 estimate 함수를 정의한다.

```
estimate_ord_krig <- function(xdata, ydata, bw)
{
    nd <- length(ydata)
    return(mykrig1(nd, nd, xdata, ydata, xdata, bw))
}
```

마지막으로, xdata와 ydata, bandwidth 목록을 받아서 이중 가장 CV값이 낮은 bandwidth를 사용한 kriging 그래프를 출력해주는 함수를 다음과 같이 구현하였다.

```
full_test_kriging <- function(xdata, ydata, smtparam_array, description = "")
```

```
    cv_array <- cv_ord_krig(xdata, ydata, smtparam_array)
    plot_cv(smtparam_array, cv_array, description)

    smt_idx <- which(cv_array == min(cv_array, na.rm = TRUE))
    if(length(smt_idx) > 1)
    {
        smt_idx <- smt_idx[1]
    }
    best_smt <- smtparam_array[smt_idx]


    plot_graph(estimate_ord_krig, xdata, ydata, best_smt, description = description)
}
```

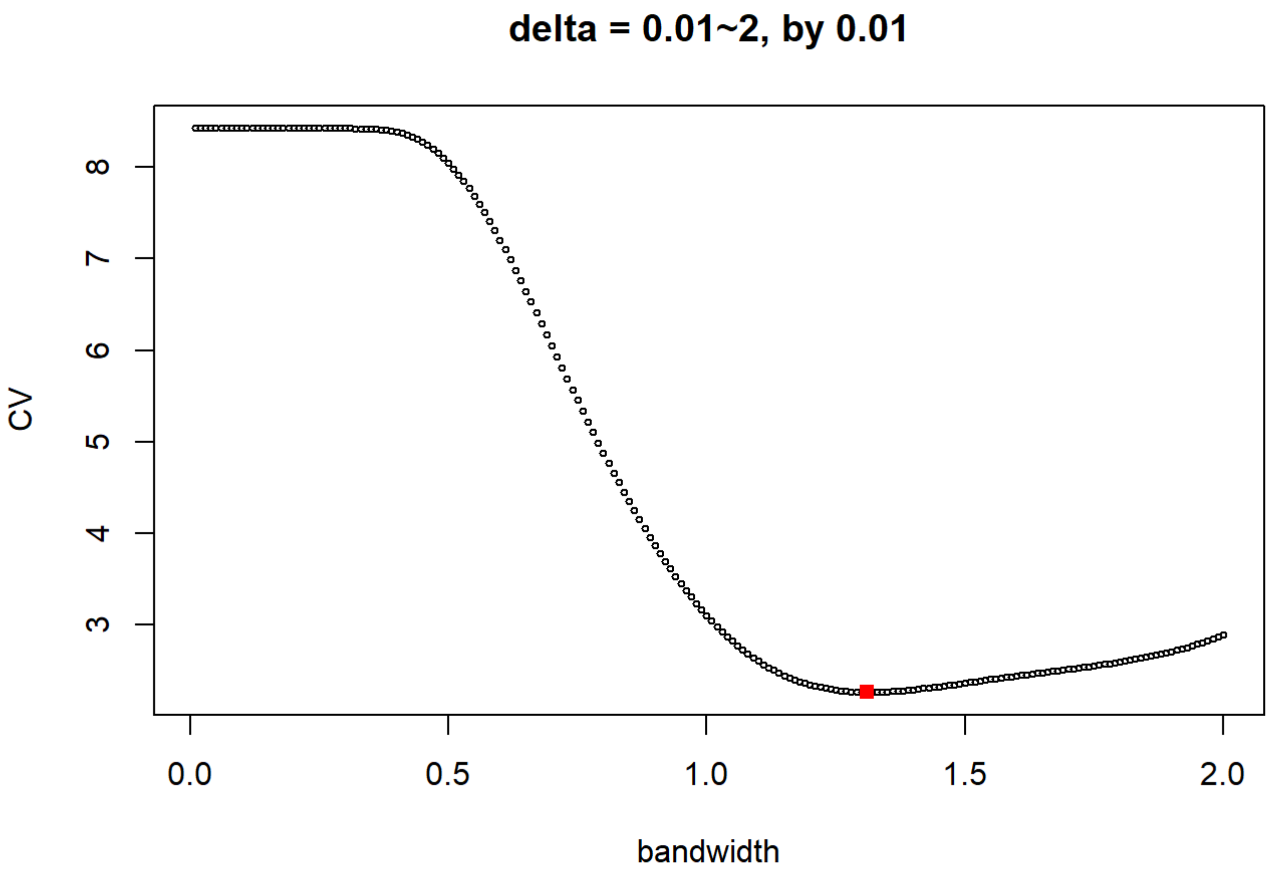
1-(b)

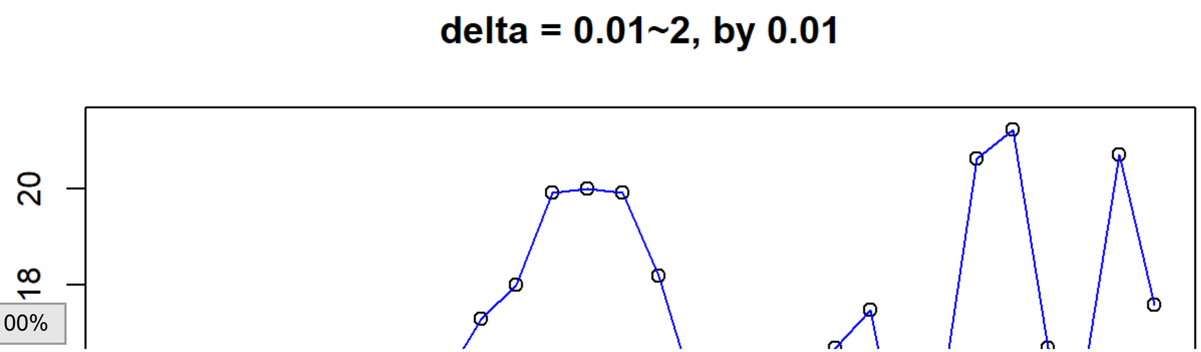
problem 2.3의 chapter 2의 데이터는 eqispaced data로, x값으로 1,2…의 값을 가지게 된다.

해당 데이터를 생성하고, 위 함수를 이용하여 가장 CV값을 낮게 만드는 δ를 찾자.

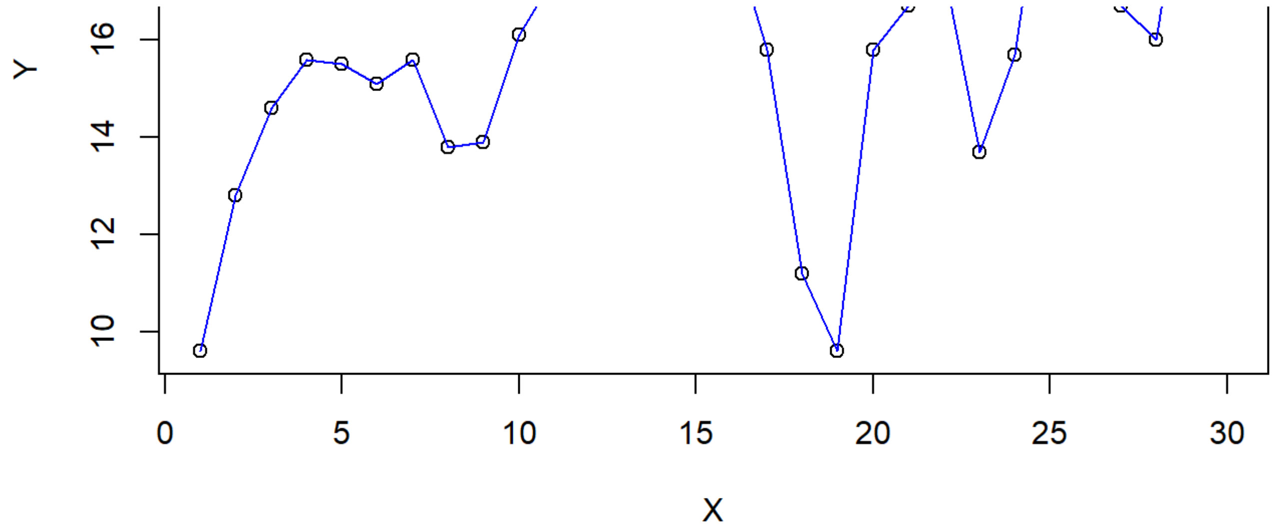
delta의 값으로는 0.01~2까지, 0.01 간격의 값을 주었다.

```
ydata <- c(9.6, 12.8, 14.6, 15.6 ,15.5 ,15.1, 15.6, 13.8, 13.9, 16.1, 17.3, 18, 19.9, 20, 19.9, 18.2, 15.8, 11.2,
 9.6, 15.8, 16.7, 17.5, 13.7, 15.7, 20.6, 21.2, 16.7, 16, 20.7, 17.6)
xdata <- seq_len(length(ydata))


full_test_kriging(xdata, ydata, seq(0.01,2,by = 0.01), description = "delta = 0.01~2, by 0.01")
```





Processing math: 100%



using smtparam : 1.310000

위에서 알 수 있듯이, $\delta = 1.31$에서 CV가 가장 작고, 이 값을 이용하여 ordinary criging을 한 결과로 출력된 그래프는 위와 같다.

Q2

2-(a)

모델이 다음과 같은 simple kriging에서의 Hat matrix를 구하고, Hat matrix의 성분들을 plot해주는 함수를 만들자.

모델은 다음과 같다.

$$y_i = e_i = e_i^c + e_i^u$$

.

$$E[e_i^c] = 0, E[e_i^u] = 0, Cov(e_i^c, e_j^c) = [C']_{ij}, Cov(e_0^c, e_i^c) = d_i^{'}, Cov(e_0^c, e_0^u) = 0, Cov(e_i^c, e_i^u) = 0$$

$$Cov(e_i^u, e_j^c) = 0, Cov(e_i^u, e_j^u) = 0(i \neq j), Cov(e_i^u, e_i^u) = \sigma^2, Cov(e_0^u, e_0^u) = \sigma^2$$

Processing math: 100% $\hat{e}_0^c$)를 구하기 위해, $\hat{y}_0 = \sum_{i=1}^{n} \alpha_i y_i$ 와 같은 linear predicter라 하면, 이때의 α는 다음과 같이 구해진다.

$\alpha = (\alpha 1, \alpha 2, ... \alpha_n)^t$라 할때,

아래 식을 최소화 하는 α를 구하면 되고,

$$Var_{simple, smoothing} = E((e_0^c - \sum_{i=1}^{n} \alpha_i (e_i^c + e_i^u))^2) = E[(e_0^c)^2] - 2\sum_{i=1}^{n} \alpha_i d_i' + \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i [C']_{ij} \alpha_j + \sigma^2 \sum_{i=1}^{n} \alpha_i$$

위를 α에 대해 미분했을때, 0이되는 α가 최소화 하는 값이므로, 이러한 $\alpha = (C' + \sigma^2 I_n)^{-1} d_0'$ 이므로, $(d_0' = (d_1', d_2' ... d_n')^t, d_i' = Cov(e_0^c, e_i^c))$ 가 되고,

따라서 Hat matrix H를 구하기 위해서는, $\hat{y}_i$를 구해야 하는데,

$\hat{y}_i$에 해당하는 $d_i' = (Cov(e_i^c, e_1^c), ..., Cov(e_i^c, e_n^c))^t$ 에서, 이는 C'의 ith row와 같으므로, d_i'를 ith row로 가지는 n x n matrix를 D'이라 하면,

$\hat{y} = (C' + \sigma^2 I_n)^{-1} D' y$ 에서, $D' = C'$이므로,

따라서 $\hat{y} = (I_n + \sigma^2 D'^{-1})^{-1} y$ 이므로,

hat matrix $H = (I_n + \sigma^2 C'^{-1})^{-1}$ 가 된다.

따라서 이를 구하자.

여기서, C'의 값은 정확히 알 수 없는 값이므로, theoretical correlogram을 사용하도록 하겠다.

gaussian correlogram을 사용한다. $Correlo(r) = \exp(-(\frac{r}{\delta})^2)$

이제 이를 구해주는 함수를 짜자. 다음 함수는 x data, bandwidth, sigma square를 받아서 Hat matrix의 각 성분들을 구해서 리턴해준다.

```
plot_hatmat <- function(hat_mat)
{
   y <- matrix(unlist(hat_mat), nrow = dim(hat_mat)[1], ncol = dim(hat_mat)[2])

   persp(y, zlim = c(-0.3, 1), xlab = "Hat_i", ylab = "Hat_j",  zlab = "Hatmat_value", lab = c(3,3,3), theta = -3
0, phi = 20, ticktype = "detailed")
}
```

Processing math: 100%

```r
simple_krig_hat<-function(xx, bw, sig2)
{
  nd <- length(xx)
# (1) get C' matrix
  full <- matrix(0, nd, nd)
  full[lower.tri(full)] <- dist(xx)
  dist1 <- full + t(full)
  cc <- exp( - dist1^2/bw^2)
# (2) get (I + \sigma^2 C'^-1)^-1
  cc_inv <- solve(cc)
  I_plus_cc_inv <- diag(nd) + sig2 * cc_inv
  hat_mat <- solve(I_plus_cc_inv)

  plot_hatmat(hat_mat)

  return(hat_mat)
}
```

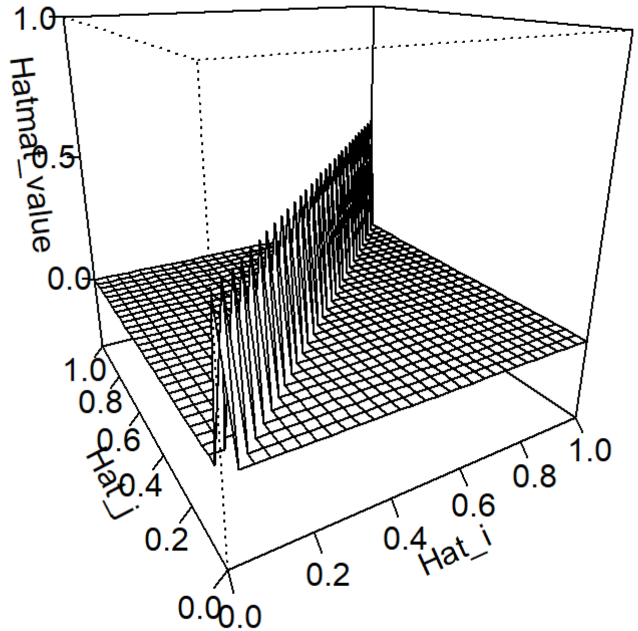
이제, 1번에서 사용했던 데이터를 사용해서 hat matrix의 값과 그래프를 그려보면 다음과 같다.

$sigma^2 = 1$로 가정하였다.

bandwidth(δ)를 0.3 , 1, 3으로 주었을때의 hat matrix는 다음과 같다.

```r
ydata <- c(9.6, 12.8, 14.6, 15.6 ,15.5 ,15.1, 15.6, 13.8, 13.9, 16.1, 17.3, 18, 19.9, 20, 19.9, 18.2, 15.8, 11.2,
 9.6, 15.8, 16.7, 17.5, 13.7, 15.7, 20.6, 21.2, 16.7, 16, 20.7, 17.6)
xdata <- seq_len(length(ydata))

simple_krig_hat(xdata, 0.3, 1)
```

Processing math: 100%




```
##                [,1]          [,2]          [,3]          [,4]
## [1,]   5.000000e-01  3.736335e-06 -2.792039e-11  2.086399e-16
## [2,]   3.736335e-06  5.000000e-01  3.736335e-06 -2.792039e-11
## [3,]  -2.792039e-11  3.736335e-06  5.000000e-01  3.736335e-06
## [4,]   2.086399e-16 -2.792039e-11  3.736335e-06  5.000000e-01
## [5,]  -1.559097e-21  2.086399e-16 -2.792039e-11  3.736335e-06
## [6,]   1.165061e-26 -1.559097e-21  2.086399e-16 -2.792039e-11
## [7,]  -8.706118e-32  1.165061e-26 -1.559097e-21  2.086399e-16
## [8,]   6.505794e-37 -8.706118e-32  1.165061e-26 -1.559097e-21
```

```
##  [9,]  -4.861565e-42   6.505794e-37  -8.706118e-32   1.165061e-26
## [10,]   3.632887e-47  -4.861565e-42   6.505794e-37  -8.706118e-32
## [11,]  -2.714736e-52   3.632887e-47  -4.861565e-42   6.505794e-37
## [12,]   2.028632e-57  -2.714736e-52   3.632887e-47  -4.861565e-42
## [13,]  -1.515930e-62   2.028632e-57  -2.714736e-52   3.632887e-47
## [14,]   1.132804e-67  -1.515930e-62   2.028632e-57  -2.714736e-52
## [15,]  -8.465072e-73   1.132804e-67  -1.515930e-62   2.028632e-57
## [16,]   6.325668e-78  -8.465072e-73   1.132804e-67  -1.515930e-62
## [17,]  -4.726963e-83   6.325668e-78  -8.465072e-73   1.132804e-67
## [18,]   3.532303e-88  -4.726963e-83   6.325668e-78  -8.465072e-73
## [19,]  -2.639573e-93   3.532303e-88  -4.726963e-83   6.325668e-78
## [20,]   1.972466e-98  -2.639573e-93   3.532303e-88  -4.726963e-83
## [21,] -1.473958e-103   1.972466e-98  -2.639573e-93   3.532303e-88
## [22,]  1.101440e-108 -1.473958e-103   1.972466e-98  -2.639573e-93
## [23,] -8.230699e-114  1.101440e-108 -1.473958e-103   1.972466e-98
## [24,]  6.150529e-119 -8.230699e-114  1.101440e-108 -1.473958e-103
## [25,] -4.596087e-124  6.150529e-119 -8.230699e-114  1.101440e-108
## [26,]  3.434504e-129 -4.596087e-124  6.150529e-119 -8.230699e-114
## [27,] -2.566491e-134  3.434504e-129 -4.596087e-124  6.150529e-119
## [28,]  1.917854e-139 -2.566491e-134  3.434504e-129 -4.596087e-124
## [29,] -1.433149e-144  1.917854e-139 -2.566491e-134  3.434504e-129
## [30,]  1.070945e-149 -1.433149e-144  1.917854e-139 -2.566491e-134
##                 [,5]           [,6]           [,7]           [,8]
##  [1,]  -1.559097e-21   1.165061e-26  -8.706118e-32   6.505794e-37
##  [2,]   2.086399e-16  -1.559097e-21   1.165061e-26  -8.706118e-32
##  [3,]  -2.792039e-11   2.086399e-16  -1.559097e-21   1.165061e-26
##  [4,]   3.736335e-06  -2.792039e-11   2.086399e-16  -1.559097e-21
##  [5,]   5.000000e-01   3.736335e-06  -2.792039e-11   2.086399e-16
##  [6,]   3.736335e-06   5.000000e-01   3.736335e-06  -2.792039e-11
##  [7,]  -2.792039e-11   3.736335e-06   5.000000e-01   3.736335e-06
##  [8,]   2.086399e-16  -2.792039e-11   3.736335e-06   5.000000e-01
##  [9,]  -1.559097e-21   2.086399e-16  -2.792039e-11   3.736335e-06
## [10,]   1.165061e-26  -1.559097e-21   2.086399e-16  -2.792039e-11
## [11,]  -8.706118e-32   1.165061e-26  -1.559097e-21   2.086399e-16
## [12,]   6.505794e-37  -8.706118e-32   1.165061e-26  -1.559097e-21
## [13,]  -4.861565e-42   6.505794e-37  -8.706118e-32   1.165061e-26
## [14,]   3.632887e-47  -4.861565e-42   6.505794e-37  -8.706118e-32
```

```
## [15,]  -2.714736e-52   3.632887e-47  -4.861565e-42   6.505794e-37
## [16,]   2.028632e-57  -2.714736e-52   3.632887e-47  -4.861565e-42
## [17,]  -1.515930e-62   2.028632e-57  -2.714736e-52   3.632887e-47
## [18,]   1.132804e-67  -1.515930e-62   2.028632e-57  -2.714736e-52
## [19,]  -8.465072e-73   1.132804e-67  -1.515930e-62   2.028632e-57
## [20,]   6.325668e-78  -8.465072e-73   1.132804e-67  -1.515930e-62
## [21,]  -4.726963e-83   6.325668e-78  -8.465072e-73   1.132804e-67
## [22,]   3.532303e-88  -4.726963e-83   6.325668e-78  -8.465072e-73
## [23,]  -2.639573e-93   3.532303e-88  -4.726963e-83   6.325668e-78
## [24,]   1.972466e-98  -2.639573e-93   3.532303e-88  -4.726963e-83
## [25,] -1.473958e-103   1.972466e-98  -2.639573e-93   3.532303e-88
## [26,]  1.101440e-108 -1.473958e-103   1.972466e-98  -2.639573e-93
## [27,] -8.230699e-114  1.101440e-108 -1.473958e-103   1.972466e-98
## [28,]  6.150529e-119 -8.230699e-114  1.101440e-108 -1.473958e-103
## [29,] -4.596087e-124  6.150529e-119 -8.230699e-114  1.101440e-108
## [30,]  3.434504e-129 -4.596087e-124  6.150529e-119 -8.230699e-114
##                 [,9]          [,10]          [,11]          [,12]
##  [1,]  -4.861565e-42   3.632887e-47 -2.714736e-52  2.028632e-57
##  [2,]   6.505794e-37  -4.861565e-42  3.632887e-47 -2.714736e-52
##  [3,]  -8.706118e-32   6.505794e-37 -4.861565e-42  3.632887e-47
##  [4,]   1.165061e-26  -8.706118e-32  6.505794e-37 -4.861565e-42
##  [5,]  -1.559097e-21   1.165061e-26 -8.706118e-32  6.505794e-37
##  [6,]   2.086399e-16  -1.559097e-21  1.165061e-26 -8.706118e-32
##  [7,]  -2.792039e-11   2.086399e-16 -1.559097e-21  1.165061e-26
##  [8,]   3.736335e-06  -2.792039e-11  2.086399e-16 -1.559097e-21
##  [9,]   5.000000e-01   3.736335e-06 -2.792039e-11  2.086399e-16
## [10,]   3.736335e-06   5.000000e-01  3.736335e-06 -2.792039e-11
## [11,]  -2.792039e-11   3.736335e-06  5.000000e-01  3.736335e-06
## [12,]   2.086399e-16  -2.792039e-11  3.736335e-06  5.000000e-01
## [13,]  -1.559097e-21   2.086399e-16 -2.792039e-11  3.736335e-06
## [14,]   1.165061e-26  -1.559097e-21  2.086399e-16 -2.792039e-11
## [15,]  -8.706118e-32   1.165061e-26 -1.559097e-21  2.086399e-16
## [16,]   6.505794e-37  -8.706118e-32  1.165061e-26 -1.559097e-21
## [17,]  -4.861565e-42   6.505794e-37 -8.706118e-32  1.165061e-26
## [18,]   3.632887e-47  -4.861565e-42  6.505794e-37 -8.706118e-32
## [19,]  -2.714736e-52   3.632887e-47 -4.861565e-42  6.505794e-37
## [20,]   2.028632e-57  -2.714736e-52  3.632887e-47 -4.861565e-42
```

Processing math: 100%

```
## [21,]  -1.515930e-62   2.028632e-57 -2.714736e-52  3.632887e-47
## [22,]   1.132804e-67  -1.515930e-62  2.028632e-57 -2.714736e-52
## [23,]  -8.465072e-73   1.132804e-67 -1.515930e-62  2.028632e-57
## [24,]   6.325668e-78  -8.465072e-73  1.132804e-67 -1.515930e-62
## [25,]  -4.726963e-83   6.325668e-78 -8.465072e-73  1.132804e-67
## [26,]   3.532303e-88  -4.726963e-83  6.325668e-78 -8.465072e-73
## [27,]  -2.639573e-93   3.532303e-88 -4.726963e-83  6.325668e-78
## [28,]   1.972466e-98  -2.639573e-93  3.532303e-88 -4.726963e-83
## [29,] -1.473958e-103   1.972466e-98 -2.639573e-93  3.532303e-88
## [30,]  1.101440e-108 -1.473958e-103  1.972466e-98 -2.639573e-93
##               [,13]         [,14]         [,15]         [,16]
##  [1,] -1.515930e-62  1.132804e-67 -8.465072e-73  6.325668e-78
##  [2,]  2.028632e-57 -1.515930e-62  1.132804e-67 -8.465072e-73
##  [3,] -2.714736e-52  2.028632e-57 -1.515930e-62  1.132804e-67
##  [4,]  3.632887e-47 -2.714736e-52  2.028632e-57 -1.515930e-62
##  [5,] -4.861565e-42  3.632887e-47 -2.714736e-52  2.028632e-57
##  [6,]  6.505794e-37 -4.861565e-42  3.632887e-47 -2.714736e-52
##  [7,] -8.706118e-32  6.505794e-37 -4.861565e-42  3.632887e-47
##  [8,]  1.165061e-26 -8.706118e-32  6.505794e-37 -4.861565e-42
##  [9,] -1.559097e-21  1.165061e-26 -8.706118e-32  6.505794e-37
## [10,]  2.086399e-16 -1.559097e-21  1.165061e-26 -8.706118e-32
## [11,] -2.792039e-11  2.086399e-16 -1.559097e-21  1.165061e-26
## [12,]  3.736335e-06 -2.792039e-11  2.086399e-16 -1.559097e-21
## [13,]  5.000000e-01  3.736335e-06 -2.792039e-11  2.086399e-16
## [14,]  3.736335e-06  5.000000e-01  3.736335e-06 -2.792039e-11
## [15,] -2.792039e-11  3.736335e-06  5.000000e-01  3.736335e-06
## [16,]  2.086399e-16 -2.792039e-11  3.736335e-06  5.000000e-01
## [17,] -1.559097e-21  2.086399e-16 -2.792039e-11  3.736335e-06
## [18,]  1.165061e-26 -1.559097e-21  2.086399e-16 -2.792039e-11
## [19,] -8.706118e-32  1.165061e-26 -1.559097e-21  2.086399e-16
## [20,]  6.505794e-37 -8.706118e-32  1.165061e-26 -1.559097e-21
## [21,] -4.861565e-42  6.505794e-37 -8.706118e-32  1.165061e-26
## [22,]  3.632887e-47 -4.861565e-42  6.505794e-37 -8.706118e-32
## [23,] -2.714736e-52  3.632887e-47 -4.861565e-42  6.505794e-37
## [24,]  2.028632e-57 -2.714736e-52  3.632887e-47 -4.861565e-42
## [25,] -1.515930e-62  2.028632e-57 -2.714736e-52  3.632887e-47
## [26,]  1.132804e-67 -1.515930e-62  2.028632e-57 -2.714736e-52
```

Processing math: 100%

```
## [27,] -8.465072e-73  1.132804e-67 -1.515930e-62  2.028632e-57
## [28,]  6.325668e-78 -8.465072e-73  1.132804e-67 -1.515930e-62
## [29,] -4.726963e-83  6.325668e-78 -8.465072e-73  1.132804e-67
## [30,]  3.532303e-88 -4.726963e-83  6.325668e-78 -8.465072e-73
##               [,17]         [,18]         [,19]         [,20]
##  [1,] -4.726963e-83  3.532303e-88 -2.639573e-93  1.972466e-98
##  [2,]  6.325668e-78 -4.726963e-83  3.532303e-88 -2.639573e-93
##  [3,] -8.465072e-73  6.325668e-78 -4.726963e-83  3.532303e-88
##  [4,]  1.132804e-67 -8.465072e-73  6.325668e-78 -4.726963e-83
##  [5,] -1.515930e-62  1.132804e-67 -8.465072e-73  6.325668e-78
##  [6,]  2.028632e-57 -1.515930e-62  1.132804e-67 -8.465072e-73
##  [7,] -2.714736e-52  2.028632e-57 -1.515930e-62  1.132804e-67
##  [8,]  3.632887e-47 -2.714736e-52  2.028632e-57 -1.515930e-62
##  [9,] -4.861565e-42  3.632887e-47 -2.714736e-52  2.028632e-57
## [10,]  6.505794e-37 -4.861565e-42  3.632887e-47 -2.714736e-52
## [11,] -8.706118e-32  6.505794e-37 -4.861565e-42  3.632887e-47
## [12,]  1.165061e-26 -8.706118e-32  6.505794e-37 -4.861565e-42
## [13,] -1.559097e-21  1.165061e-26 -8.706118e-32  6.505794e-37
## [14,]  2.086399e-16 -1.559097e-21  1.165061e-26 -8.706118e-32
## [15,] -2.792039e-11  2.086399e-16 -1.559097e-21  1.165061e-26
## [16,]  3.736335e-06 -2.792039e-11  2.086399e-16 -1.559097e-21
## [17,]  5.000000e-01  3.736335e-06 -2.792039e-11  2.086399e-16
## [18,]  3.736335e-06  5.000000e-01  3.736335e-06 -2.792039e-11
## [19,] -2.792039e-11  3.736335e-06  5.000000e-01  3.736335e-06
## [20,]  2.086399e-16 -2.792039e-11  3.736335e-06  5.000000e-01
## [21,] -1.559097e-21  2.086399e-16 -2.792039e-11  3.736335e-06
## [22,]  1.165061e-26 -1.559097e-21  2.086399e-16 -2.792039e-11
## [23,] -8.706118e-32  1.165061e-26 -1.559097e-21  2.086399e-16
## [24,]  6.505794e-37 -8.706118e-32  1.165061e-26 -1.559097e-21
## [25,] -4.861565e-42  6.505794e-37 -8.706118e-32  1.165061e-26
## [26,]  3.632887e-47 -4.861565e-42  6.505794e-37 -8.706118e-32
## [27,] -2.714736e-52  3.632887e-47 -4.861565e-42  6.505794e-37
## [28,]  2.028632e-57 -2.714736e-52  3.632887e-47 -4.861565e-42
## [29,] -1.515930e-62  2.028632e-57 -2.714736e-52  3.632887e-47
## [30,]  1.132804e-67 -1.515930e-62  2.028632e-57 -2.714736e-52
##                [,21]          [,22]          [,23]          [,24]
##  [1,] -1.473958e-103  1.101440e-108 -8.230699e-114  6.150529e-119
```

```
##  [2,]   1.972466e-98 -1.473958e-103  1.101440e-108 -8.230699e-114
##  [3,]  -2.639573e-93   1.972466e-98 -1.473958e-103  1.101440e-108
##  [4,]   3.532303e-88  -2.639573e-93   1.972466e-98 -1.473958e-103
##  [5,]  -4.726963e-83   3.532303e-88  -2.639573e-93   1.972466e-98
##  [6,]   6.325668e-78  -4.726963e-83   3.532303e-88  -2.639573e-93
##  [7,]  -8.465072e-73   6.325668e-78  -4.726963e-83   3.532303e-88
##  [8,]   1.132804e-67  -8.465072e-73   6.325668e-78  -4.726963e-83
##  [9,]  -1.515930e-62   1.132804e-67  -8.465072e-73   6.325668e-78
## [10,]   2.028632e-57  -1.515930e-62   1.132804e-67  -8.465072e-73
## [11,]  -2.714736e-52   2.028632e-57  -1.515930e-62   1.132804e-67
## [12,]   3.632887e-47  -2.714736e-52   2.028632e-57  -1.515930e-62
## [13,]  -4.861565e-42   3.632887e-47  -2.714736e-52   2.028632e-57
## [14,]   6.505794e-37  -4.861565e-42   3.632887e-47  -2.714736e-52
## [15,]  -8.706118e-32   6.505794e-37  -4.861565e-42   3.632887e-47
## [16,]   1.165061e-26  -8.706118e-32   6.505794e-37  -4.861565e-42
## [17,]  -1.559097e-21   1.165061e-26  -8.706118e-32   6.505794e-37
## [18,]   2.086399e-16  -1.559097e-21   1.165061e-26  -8.706118e-32
## [19,]  -2.792039e-11   2.086399e-16  -1.559097e-21   1.165061e-26
## [20,]   3.736335e-06  -2.792039e-11   2.086399e-16  -1.559097e-21
## [21,]   5.000000e-01   3.736335e-06  -2.792039e-11   2.086399e-16
## [22,]   3.736335e-06   5.000000e-01   3.736335e-06  -2.792039e-11
## [23,]  -2.792039e-11   3.736335e-06   5.000000e-01   3.736335e-06
## [24,]   2.086399e-16  -2.792039e-11   3.736335e-06   5.000000e-01
## [25,]  -1.559097e-21   2.086399e-16  -2.792039e-11   3.736335e-06
## [26,]   1.165061e-26  -1.559097e-21   2.086399e-16  -2.792039e-11
## [27,]  -8.706118e-32   1.165061e-26  -1.559097e-21   2.086399e-16
## [28,]   6.505794e-37  -8.706118e-32   1.165061e-26  -1.559097e-21
## [29,]  -4.861565e-42   6.505794e-37  -8.706118e-32   1.165061e-26
## [30,]   3.632887e-47  -4.861565e-42   6.505794e-37  -8.706118e-32
##                [,25]          [,26]          [,27]          [,28]
##  [1,] -4.596087e-124  3.434504e-129 -2.566491e-134  1.917854e-139
##  [2,]  6.150529e-119 -4.596087e-124  3.434504e-129 -2.566491e-134
##  [3,] -8.230699e-114  6.150529e-119 -4.596087e-124  3.434504e-129
##  [4,]  1.101440e-108 -8.230699e-114  6.150529e-119 -4.596087e-124
##  [5,] -1.473958e-103  1.101440e-108 -8.230699e-114  6.150529e-119
##  [6,]   1.972466e-98 -1.473958e-103  1.101440e-108 -8.230699e-114
##  [7,]  -2.639573e-93   1.972466e-98 -1.473958e-103  1.101440e-108
```

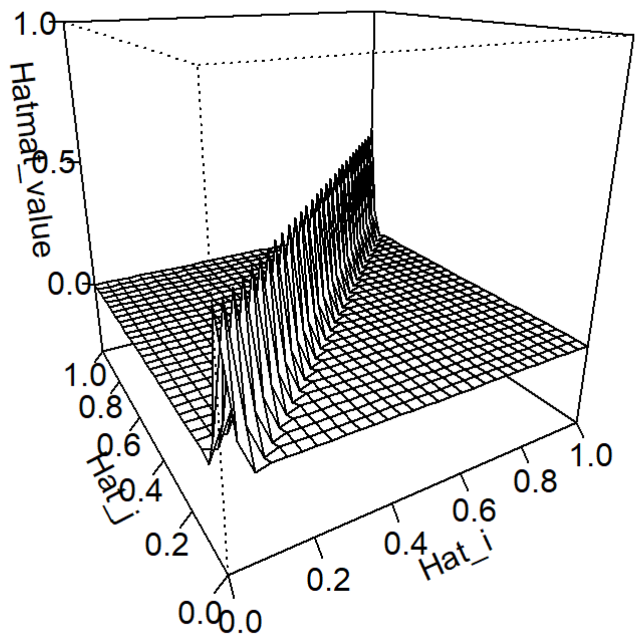
```
##  [8,]   3.532303e-88  -2.639573e-93   1.972466e-98 -1.473958e-103
##  [9,]  -4.726963e-83   3.532303e-88  -2.639573e-93   1.972466e-98
## [10,]   6.325668e-78  -4.726963e-83   3.532303e-88  -2.639573e-93
## [11,]  -8.465072e-73   6.325668e-78  -4.726963e-83   3.532303e-88
## [12,]   1.132804e-67  -8.465072e-73   6.325668e-78  -4.726963e-83
## [13,]  -1.515930e-62   1.132804e-67  -8.465072e-73   6.325668e-78
## [14,]   2.028632e-57  -1.515930e-62   1.132804e-67  -8.465072e-73
## [15,]  -2.714736e-52   2.028632e-57  -1.515930e-62   1.132804e-67
## [16,]   3.632887e-47  -2.714736e-52   2.028632e-57  -1.515930e-62
## [17,]  -4.861565e-42   3.632887e-47  -2.714736e-52   2.028632e-57
## [18,]   6.505794e-37  -4.861565e-42   3.632887e-47  -2.714736e-52
## [19,]  -8.706118e-32   6.505794e-37  -4.861565e-42   3.632887e-47
## [20,]   1.165061e-26  -8.706118e-32   6.505794e-37  -4.861565e-42
## [21,]  -1.559097e-21   1.165061e-26  -8.706118e-32   6.505794e-37
## [22,]   2.086399e-16  -1.559097e-21   1.165061e-26  -8.706118e-32
## [23,]  -2.792039e-11   2.086399e-16  -1.559097e-21   1.165061e-26
## [24,]   3.736335e-06  -2.792039e-11   2.086399e-16  -1.559097e-21
## [25,]   5.000000e-01   3.736335e-06  -2.792039e-11   2.086399e-16
## [26,]   3.736335e-06   5.000000e-01   3.736335e-06  -2.792039e-11
## [27,]  -2.792039e-11   3.736335e-06   5.000000e-01   3.736335e-06
## [28,]   2.086399e-16  -2.792039e-11   3.736335e-06   5.000000e-01
## [29,]  -1.559097e-21   2.086399e-16  -2.792039e-11   3.736335e-06
## [30,]   1.165061e-26  -1.559097e-21   2.086399e-16  -2.792039e-11
##                [,29]          [,30]
##  [1,] -1.433149e-144  1.070945e-149
##  [2,]  1.917854e-139 -1.433149e-144
##  [3,] -2.566491e-134  1.917854e-139
##  [4,]  3.434504e-129 -2.566491e-134
##  [5,] -4.596087e-124  3.434504e-129
##  [6,]  6.150529e-119 -4.596087e-124
##  [7,] -8.230699e-114  6.150529e-119
##  [8,]  1.101440e-108 -8.230699e-114
##  [9,] -1.473958e-103  1.101440e-108
## [10,]   1.972466e-98 -1.473958e-103
## [11,]  -2.639573e-93   1.972466e-98
## [12,]   3.532303e-88  -2.639573e-93
## [13,]  -4.726963e-83   3.532303e-88
```

Processing math: 100%

```
## [14,]   6.325668e-78  -4.726963e-83
## [15,]  -8.465072e-73   6.325668e-78
## [16,]   1.132804e-67  -8.465072e-73
## [17,]  -1.515930e-62   1.132804e-67
## [18,]   2.028632e-57  -1.515930e-62
## [19,]  -2.714736e-52   2.028632e-57
## [20,]   3.632887e-47  -2.714736e-52
## [21,]  -4.861565e-42   3.632887e-47
## [22,]   6.505794e-37  -4.861565e-42
## [23,]  -8.706118e-32   6.505794e-37
## [24,]   1.165061e-26  -8.706118e-32
## [25,]  -1.559097e-21   1.165061e-26
## [26,]   2.086399e-16  -1.559097e-21
## [27,]  -2.792039e-11   2.086399e-16
## [28,]   3.736335e-06  -2.792039e-11
## [29,]   5.000000e-01   3.736335e-06
## [30,]   3.736335e-06   5.000000e-01
```

```
simple_krig_hat(xdata, 1, 1)
```

Processing math: 100%




```
##               [,1]          [,2]          [,3]          [,4]
##  [1,]  4.821477e-01  9.772852e-02 -1.353774e-02  1.662161e-03
##  [2,]  9.772852e-02  4.637045e-01  1.002834e-01 -1.385142e-02
##  [3,] -1.353774e-02  1.002834e-01  4.633506e-01  1.003268e-01
##  [4,]  1.662161e-03 -1.385142e-02  1.003268e-01  4.633453e-01
##  [5,] -1.916561e-04  1.698330e-03 -1.385643e-02  1.003274e-01
##  [6,]  2.128326e-05 -1.956727e-04  1.698886e-03 -1.385650e-02
##  [7,] -2.306464e-06  2.171853e-05 -1.957330e-04  1.698894e-03
##  [8,]  2.458122e-07 -2.352854e-06  2.172496e-05 -1.957338e-04
```

Processing math: 100%

```
##  [9,] -2.588967e-08  2.506981e-07 -2.353530e-06  2.172504e-05
## [10,]  2.703494e-09 -2.639987e-08  2.507688e-07 -2.353539e-06
## [11,] -2.805268e-10  2.756435e-09 -2.640721e-08  2.507697e-07
## [12,]  2.897120e-11 -2.859943e-10  2.757192e-09 -2.640730e-08
## [13,] -2.981295e-12  2.953383e-11 -2.860722e-10  2.757202e-09
## [14,]  3.059578e-13 -3.039035e-12  2.954183e-11 -2.860732e-10
## [15,] -3.133384e-14  3.118711e-13 -3.039855e-12  2.954193e-11
## [16,]  3.203841e-15 -3.193847e-14  3.119548e-13 -3.039865e-12
## [17,] -3.271844e-16  3.265587e-15 -3.194702e-14  3.119559e-13
## [18,]  3.338109e-17 -3.334841e-16  3.266459e-15 -3.194713e-14
## [19,] -3.403203e-18  3.402334e-17 -3.335731e-16  3.266470e-15
## [20,]  3.467587e-19 -3.468642e-18  3.403240e-17 -3.335742e-16
## [21,] -3.531619e-20  3.534228e-19 -3.469565e-18  3.403252e-17
## [22,]  3.595580e-21 -3.599472e-20  3.535172e-19 -3.469577e-18
## [23,] -3.659796e-22  3.664608e-21 -3.600439e-20  3.535179e-19
## [24,]  3.723506e-23 -3.730051e-22  3.665589e-21 -3.600453e-20
## [25,] -3.790305e-24  3.795212e-23 -3.731048e-22  3.665666e-21
## [26,]  3.854320e-25 -3.859483e-24  3.796931e-23 -3.731002e-22
## [27,] -3.908524e-26  3.916830e-25 -3.866923e-24  3.795854e-23
## [28,]  3.908505e-27 -4.038942e-26  3.920009e-25 -3.867387e-24
## [29,] -3.934459e-28  4.105115e-27 -4.048987e-26  3.914411e-25
## [30,]  5.421848e-29 -3.681362e-28  3.948056e-27 -3.940976e-26
##                [,5]          [,6]          [,7]          [,8]
##  [1,] -1.916561e-04  2.128326e-05 -2.306464e-06  2.458122e-07
##  [2,]  1.698330e-03 -1.956727e-04  2.171853e-05 -2.352854e-06
##  [3,] -1.385643e-02  1.698886e-03 -1.957330e-04  2.172496e-05
##  [4,]  1.003274e-01 -1.385650e-02  1.698894e-03 -1.957338e-04
##  [5,]  4.633452e-01  1.003274e-01 -1.385650e-02  1.698894e-03
##  [6,]  1.003274e-01  4.633452e-01  1.003274e-01 -1.385650e-02
##  [7,] -1.385650e-02  1.003274e-01  4.633452e-01  1.003274e-01
##  [8,]  1.698894e-03 -1.385650e-02  1.003274e-01  4.633452e-01
##  [9,] -1.957338e-04  1.698894e-03 -1.385650e-02  1.003274e-01
## [10,]  2.172504e-05 -1.957338e-04  1.698894e-03 -1.385650e-02
## [11,] -2.353539e-06  2.172504e-05 -1.957338e-04  1.698894e-03
## [12,]  2.507697e-07 -2.353539e-06  2.172504e-05 -1.957338e-04
## [13,] -2.640730e-08  2.507697e-07 -2.353539e-06  2.172504e-05
## [14,]  2.757202e-09 -2.640730e-08  2.507697e-07 -2.353539e-06
```

Processing math: 100%

```
## [15,] -2.860732e-10  2.757202e-09 -2.640730e-08  2.507697e-07
## [16,]  2.954193e-11 -2.860732e-10  2.757202e-09 -2.640730e-08
## [17,] -3.039865e-12  2.954193e-11 -2.860732e-10  2.757202e-09
## [18,]  3.119559e-13 -3.039865e-12  2.954193e-11 -2.860732e-10
## [19,] -3.194713e-14  3.119559e-13 -3.039865e-12  2.954193e-11
## [20,]  3.266471e-15 -3.194713e-14  3.119559e-13 -3.039865e-12
## [21,] -3.335742e-16  3.266471e-15 -3.194713e-14  3.119559e-13
## [22,]  3.403252e-17 -3.335742e-16  3.266471e-15 -3.194713e-14
## [23,] -3.469577e-18  3.403252e-17 -3.335742e-16  3.266471e-15
## [24,]  3.535184e-19 -3.469577e-18  3.403252e-17 -3.335742e-16
## [25,] -3.600435e-20  3.535185e-19 -3.469576e-18  3.403252e-17
## [26,]  3.665620e-21 -3.600425e-20  3.535190e-19 -3.469576e-18
## [27,] -3.731287e-22  3.665599e-21 -3.600445e-20  3.535183e-19
## [28,]  3.795421e-23 -3.731202e-22  3.665550e-21 -3.600418e-20
## [29,] -3.864659e-24  3.794775e-23 -3.729987e-22  3.664676e-21
## [30,]  3.859646e-25 -3.787999e-24  3.723456e-23 -3.659750e-22
##                [,9]          [,10]         [,11]         [,12]
##  [1,] -2.588967e-08  2.703494e-09 -2.805268e-10  2.897120e-11
##  [2,]  2.506981e-07 -2.639987e-08  2.756435e-09 -2.859943e-10
##  [3,] -2.353530e-06  2.507688e-07 -2.640721e-08  2.757192e-09
##  [4,]  2.172504e-05 -2.353539e-06  2.507697e-07 -2.640730e-08
##  [5,] -1.957338e-04  2.172504e-05 -2.353539e-06  2.507697e-07
##  [6,]  1.698894e-03 -1.957338e-04  2.172504e-05 -2.353539e-06
##  [7,] -1.385650e-02  1.698894e-03 -1.957338e-04  2.172504e-05
##  [8,]  1.003274e-01 -1.385650e-02  1.698894e-03 -1.957338e-04
##  [9,]  4.633452e-01  1.003274e-01 -1.385650e-02  1.698894e-03
## [10,]  1.003274e-01  4.633452e-01  1.003274e-01 -1.385650e-02
## [11,] -1.385650e-02  1.003274e-01  4.633452e-01  1.003274e-01
## [12,]  1.698894e-03 -1.385650e-02  1.003274e-01  4.633452e-01
## [13,] -1.957338e-04  1.698894e-03 -1.385650e-02  1.003274e-01
## [14,]  2.172504e-05 -1.957338e-04  1.698894e-03 -1.385650e-02
## [15,] -2.353539e-06  2.172504e-05 -1.957338e-04  1.698894e-03
## [16,]  2.507697e-07 -2.353539e-06  2.172504e-05 -1.957338e-04
## [17,] -2.640730e-08  2.507697e-07 -2.353539e-06  2.172504e-05
## [18,]  2.757202e-09 -2.640730e-08  2.507697e-07 -2.353539e-06
## [19,] -2.860732e-10  2.757202e-09 -2.640730e-08  2.507697e-07
## [20,]  2.954193e-11 -2.860732e-10  2.757202e-09 -2.640730e-08
```

```
## [21,] -3.039865e-12  2.954193e-11 -2.860732e-10  2.757202e-09
## [22,]  3.119559e-13 -3.039865e-12  2.954193e-11 -2.860732e-10
## [23,] -3.194713e-14  3.119559e-13 -3.039865e-12  2.954193e-11
## [24,]  3.266471e-15 -3.194713e-14  3.119559e-13 -3.039865e-12
## [25,] -3.335742e-16  3.266471e-15 -3.194713e-14  3.119559e-13
## [26,]  3.403252e-17 -3.335742e-16  3.266470e-15 -3.194713e-14
## [27,] -3.469578e-18  3.403251e-17 -3.335742e-16  3.266470e-15
## [28,]  3.535165e-19 -3.469567e-18  3.403240e-17 -3.335731e-16
## [29,] -3.599465e-20  3.534236e-19 -3.468643e-18  3.402333e-17
## [30,]  3.595651e-21 -3.531586e-20  3.467584e-19 -3.403202e-18
##               [,13]         [,14]         [,15]         [,16]
##  [1,] -2.981295e-12  3.059578e-13 -3.133384e-14  3.203841e-15
##  [2,]  2.953383e-11 -3.039035e-12  3.118711e-13 -3.193847e-14
##  [3,] -2.860722e-10  2.954183e-11 -3.039855e-12  3.119548e-13
##  [4,]  2.757202e-09 -2.860732e-10  2.954193e-11 -3.039865e-12
##  [5,] -2.640730e-08  2.757202e-09 -2.860732e-10  2.954193e-11
##  [6,]  2.507697e-07 -2.640730e-08  2.757202e-09 -2.860732e-10
##  [7,] -2.353539e-06  2.507697e-07 -2.640730e-08  2.757202e-09
##  [8,]  2.172504e-05 -2.353539e-06  2.507697e-07 -2.640730e-08
##  [9,] -1.957338e-04  2.172504e-05 -2.353539e-06  2.507697e-07
## [10,]  1.698894e-03 -1.957338e-04  2.172504e-05 -2.353539e-06
## [11,] -1.385650e-02  1.698894e-03 -1.957338e-04  2.172504e-05
## [12,]  1.003274e-01 -1.385650e-02  1.698894e-03 -1.957338e-04
## [13,]  4.633452e-01  1.003274e-01 -1.385650e-02  1.698894e-03
## [14,]  1.003274e-01  4.633452e-01  1.003274e-01 -1.385650e-02
## [15,] -1.385650e-02  1.003274e-01  4.633452e-01  1.003274e-01
## [16,]  1.698894e-03 -1.385650e-02  1.003274e-01  4.633452e-01
## [17,] -1.957338e-04  1.698894e-03 -1.385650e-02  1.003274e-01
## [18,]  2.172504e-05 -1.957338e-04  1.698894e-03 -1.385650e-02
## [19,] -2.353539e-06  2.172504e-05 -1.957338e-04  1.698894e-03
## [20,]  2.507697e-07 -2.353539e-06  2.172504e-05 -1.957338e-04
## [21,] -2.640730e-08  2.507697e-07 -2.353539e-06  2.172504e-05
## [22,]  2.757202e-09 -2.640730e-08  2.507697e-07 -2.353539e-06
## [23,] -2.860732e-10  2.757202e-09 -2.640730e-08  2.507697e-07
## [24,]  2.954193e-11 -2.860732e-10  2.757202e-09 -2.640730e-08
## [25,] -3.039865e-12  2.954193e-11 -2.860732e-10  2.757202e-09
## [26,]  3.119559e-13 -3.039865e-12  2.954193e-11 -2.860732e-10
```

Processing math: 100%

```
## [27,] -3.194713e-14  3.119559e-13 -3.039865e-12  2.954193e-11
## [28,]  3.266459e-15 -3.194702e-14  3.119548e-13 -3.039855e-12
## [29,] -3.334841e-16  3.265587e-15 -3.193847e-14  3.118711e-13
## [30,]  3.338109e-17 -3.271845e-16  3.203841e-15 -3.133384e-14
##               [,17]         [,18]         [,19]         [,20]
##  [1,] -3.271845e-16  3.338109e-17 -3.403203e-18  3.467580e-19
##  [2,]  3.265587e-15 -3.334841e-16  3.402333e-17 -3.468645e-18
##  [3,] -3.194702e-14  3.266459e-15 -3.335731e-16  3.403240e-17
##  [4,]  3.119559e-13 -3.194713e-14  3.266470e-15 -3.335742e-16
##  [5,] -3.039865e-12  3.119559e-13 -3.194713e-14  3.266471e-15
##  [6,]  2.954193e-11 -3.039865e-12  3.119559e-13 -3.194713e-14
##  [7,] -2.860732e-10  2.954193e-11 -3.039865e-12  3.119559e-13
##  [8,]  2.757202e-09 -2.860732e-10  2.954193e-11 -3.039865e-12
##  [9,] -2.640730e-08  2.757202e-09 -2.860732e-10  2.954193e-11
## [10,]  2.507697e-07 -2.640730e-08  2.757202e-09 -2.860732e-10
## [11,] -2.353539e-06  2.507697e-07 -2.640730e-08  2.757202e-09
## [12,]  2.172504e-05 -2.353539e-06  2.507697e-07 -2.640730e-08
## [13,] -1.957338e-04  2.172504e-05 -2.353539e-06  2.507697e-07
## [14,]  1.698894e-03 -1.957338e-04  2.172504e-05 -2.353539e-06
## [15,] -1.385650e-02  1.698894e-03 -1.957338e-04  2.172504e-05
## [16,]  1.003274e-01 -1.385650e-02  1.698894e-03 -1.957338e-04
## [17,]  4.633452e-01  1.003274e-01 -1.385650e-02  1.698894e-03
## [18,]  1.003274e-01  4.633452e-01  1.003274e-01 -1.385650e-02
## [19,] -1.385650e-02  1.003274e-01  4.633452e-01  1.003274e-01
## [20,]  1.698894e-03 -1.385650e-02  1.003274e-01  4.633452e-01
## [21,] -1.957338e-04  1.698894e-03 -1.385650e-02  1.003274e-01
## [22,]  2.172504e-05 -1.957338e-04  1.698894e-03 -1.385650e-02
## [23,] -2.353539e-06  2.172504e-05 -1.957338e-04  1.698894e-03
## [24,]  2.507697e-07 -2.353539e-06  2.172504e-05 -1.957338e-04
## [25,] -2.640730e-08  2.507697e-07 -2.353539e-06  2.172504e-05
## [26,]  2.757202e-09 -2.640730e-08  2.507697e-07 -2.353539e-06
## [27,] -2.860732e-10  2.757202e-09 -2.640730e-08  2.507697e-07
## [28,]  2.954183e-11 -2.860722e-10  2.757192e-09 -2.640721e-08
## [29,] -3.039035e-12  2.953383e-11 -2.859943e-10  2.756435e-09
## [30,]  3.059578e-13 -2.981295e-12  2.897120e-11 -2.805268e-10
##               [,21]         [,22]         [,23]         [,24]
##  [1,] -3.531597e-20  3.595570e-21 -3.659671e-22  3.724459e-23
```
Processing math: 100%

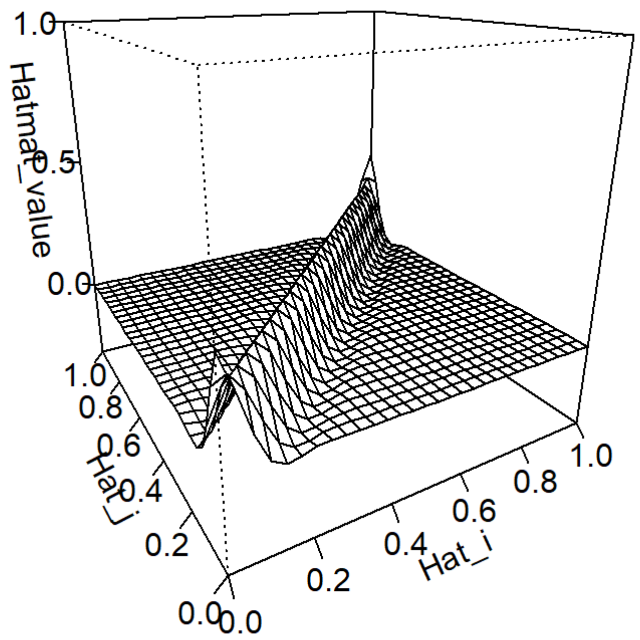
```
##  [2,]  3.534229e-19 -3.599460e-20  3.664594e-21 -3.730041e-22
##  [3,] -3.469567e-18  3.535169e-19 -3.600435e-20  3.665520e-21
##  [4,]  3.403252e-17 -3.469578e-18  3.535178e-19 -3.600428e-20
##  [5,] -3.335742e-16  3.403252e-17 -3.469579e-18  3.535181e-19
##  [6,]  3.266471e-15 -3.335742e-16  3.403252e-17 -3.469578e-18
##  [7,] -3.194713e-14  3.266471e-15 -3.335742e-16  3.403252e-17
##  [8,]  3.119559e-13 -3.194713e-14  3.266471e-15 -3.335742e-16
##  [9,] -3.039865e-12  3.119559e-13 -3.194713e-14  3.266471e-15
## [10,]  2.954193e-11 -3.039865e-12  3.119559e-13 -3.194713e-14
## [11,] -2.860732e-10  2.954193e-11 -3.039865e-12  3.119559e-13
## [12,]  2.757202e-09 -2.860732e-10  2.954193e-11 -3.039865e-12
## [13,] -2.640730e-08  2.757202e-09 -2.860732e-10  2.954193e-11
## [14,]  2.507697e-07 -2.640730e-08  2.757202e-09 -2.860732e-10
## [15,] -2.353539e-06  2.507697e-07 -2.640730e-08  2.757202e-09
## [16,]  2.172504e-05 -2.353539e-06  2.507697e-07 -2.640730e-08
## [17,] -1.957338e-04  2.172504e-05 -2.353539e-06  2.507697e-07
## [18,]  1.698894e-03 -1.957338e-04  2.172504e-05 -2.353539e-06
## [19,] -1.385650e-02  1.698894e-03 -1.957338e-04  2.172504e-05
## [20,]  1.003274e-01 -1.385650e-02  1.698894e-03 -1.957338e-04
## [21,]  4.633452e-01  1.003274e-01 -1.385650e-02  1.698894e-03
## [22,]  1.003274e-01  4.633452e-01  1.003274e-01 -1.385650e-02
## [23,] -1.385650e-02  1.003274e-01  4.633452e-01  1.003274e-01
## [24,]  1.698894e-03 -1.385650e-02  1.003274e-01  4.633452e-01
## [25,] -1.957338e-04  1.698894e-03 -1.385650e-02  1.003274e-01
## [26,]  2.172504e-05 -1.957338e-04  1.698894e-03 -1.385650e-02
## [27,] -2.353539e-06  2.172504e-05 -1.957338e-04  1.698894e-03
## [28,]  2.507688e-07 -2.353530e-06  2.172496e-05 -1.957330e-04
## [29,] -2.639987e-08  2.506981e-07 -2.352854e-06  2.171853e-05
## [30,]  2.703494e-09 -2.588967e-08  2.458122e-07 -2.306464e-06
##               [,25]         [,26]         [,27]         [,28]
##  [1,] -3.789800e-24  3.855164e-25 -3.954893e-26  4.033906e-27
##  [2,]  3.796157e-23 -3.860595e-24  3.927905e-25 -3.999339e-26
##  [3,] -3.731182e-22  3.796931e-23 -3.865072e-24  3.914298e-25
##  [4,]  3.665648e-21 -3.731055e-22  3.796408e-23 -3.862360e-24
##  [5,] -3.600430e-20  3.665595e-21 -3.731166e-22  3.796193e-23
##  [6,]  3.535185e-19 -3.600435e-20  3.665543e-21 -3.730998e-22
##  [7,] -3.469577e-18  3.535182e-19 -3.600435e-20  3.665684e-21
```

```
##  [8,]  3.403252e-17 -3.469577e-18  3.535180e-19 -3.600426e-20
##  [9,] -3.335742e-16  3.403252e-17 -3.469578e-18  3.535161e-19
## [10,]  3.266471e-15 -3.335742e-16  3.403251e-17 -3.469567e-18
## [11,] -3.194713e-14  3.266470e-15 -3.335742e-16  3.403240e-17
## [12,]  3.119559e-13 -3.194713e-14  3.266470e-15 -3.335731e-16
## [13,] -3.039865e-12  3.119559e-13 -3.194713e-14  3.266459e-15
## [14,]  2.954193e-11 -3.039865e-12  3.119559e-13 -3.194702e-14
## [15,] -2.860732e-10  2.954193e-11 -3.039865e-12  3.119548e-13
## [16,]  2.757202e-09 -2.860732e-10  2.954193e-11 -3.039855e-12
## [17,] -2.640730e-08  2.757202e-09 -2.860732e-10  2.954183e-11
## [18,]  2.507697e-07 -2.640730e-08  2.757202e-09 -2.860722e-10
## [19,] -2.353539e-06  2.507697e-07 -2.640730e-08  2.757192e-09
## [20,]  2.172504e-05 -2.353539e-06  2.507697e-07 -2.640721e-08
## [21,] -1.957338e-04  2.172504e-05 -2.353539e-06  2.507688e-07
## [22,]  1.698894e-03 -1.957338e-04  2.172504e-05 -2.353530e-06
## [23,] -1.385650e-02  1.698894e-03 -1.957338e-04  2.172496e-05
## [24,]  1.003274e-01 -1.385650e-02  1.698894e-03 -1.957330e-04
## [25,]  4.633452e-01  1.003274e-01 -1.385650e-02  1.698886e-03
## [26,]  1.003274e-01  4.633452e-01  1.003274e-01 -1.385643e-02
## [27,] -1.385650e-02  1.003274e-01  4.633453e-01  1.003268e-01
## [28,]  1.698886e-03 -1.385643e-02  1.003268e-01  4.633506e-01
## [29,] -1.956727e-04  1.698330e-03 -1.385142e-02  1.002834e-01
## [30,]  2.128326e-05 -1.916561e-04  1.662161e-03 -1.353774e-02
##               [,29]         [,30]
##  [1,] -4.122617e-28  5.574595e-29
##  [2,]  4.027576e-27 -4.462071e-28
##  [3,] -4.032500e-26  4.006018e-27
##  [4,]  3.927435e-25 -3.921118e-26
##  [5,] -3.866061e-24  3.843735e-25
##  [6,]  3.795754e-23 -3.789157e-24
##  [7,] -3.729684e-22  3.724087e-23
##  [8,]  3.664659e-21 -3.659665e-22
##  [9,] -3.599484e-20  3.595569e-21
## [10,]  3.534234e-19 -3.531591e-20
## [11,] -3.468644e-18  3.467581e-19
## [12,]  3.402333e-17 -3.403204e-18
## [13,] -3.334841e-16  3.338109e-17
```

```
## [14,]  3.265587e-15 -3.271845e-16
## [15,] -3.193847e-14  3.203841e-15
## [16,]  3.118711e-13 -3.133384e-14
## [17,] -3.039035e-12  3.059578e-13
## [18,]  2.953383e-11 -2.981295e-12
## [19,] -2.859943e-10  2.897120e-11
## [20,]  2.756435e-09 -2.805268e-10
## [21,] -2.639987e-08  2.703494e-09
## [22,]  2.506981e-07 -2.588967e-08
## [23,] -2.352854e-06  2.458122e-07
## [24,]  2.171853e-05 -2.306464e-06
## [25,] -1.956727e-04  2.128326e-05
## [26,]  1.698330e-03 -1.916561e-04
## [27,] -1.385142e-02  1.662161e-03
## [28,]  1.002834e-01 -1.353774e-02
## [29,]  4.637045e-01  9.772852e-02
## [30,]  9.772852e-02  4.821477e-01
```

```
simple_krig_hat(xdata, 3, 1)
```

Processing math: 100%




```
##                [,1]          [,2]          [,3]          [,4]
##  [1,]  3.590121e-01  2.435908e-01  1.023238e-01  7.707093e-03
##  [2,]  2.435908e-01  2.664417e-01  2.047053e-01  9.939493e-02
##  [3,]  1.023238e-01  2.047053e-01  2.501073e-01  2.034750e-01
##  [4,]  7.707093e-03  9.939493e-02  2.034750e-01  2.500147e-01
##  [5,] -2.137015e-02  1.582826e-02  1.028063e-01  2.037319e-01
##  [6,] -1.332261e-02 -1.630724e-02  1.795501e-02  1.029665e-01
##  [7,] -9.485690e-04 -1.296213e-02 -1.615581e-02  1.796642e-02
##  [8,]  3.210811e-03 -2.168754e-03 -1.347469e-02 -1.619442e-02
```

Processing math: 100%

```
##  [9,]  1.864771e-03  2.502153e-03 -2.466436e-03 -1.349711e-02
## [10,]  4.197047e-05  1.848821e-03  2.495453e-03 -2.466941e-03
## [11,] -4.909702e-04  2.285510e-04  1.927197e-03  2.501356e-03
## [12,] -2.544340e-04 -3.942792e-04  2.691674e-04  1.930256e-03
## [13,]  8.544274e-06 -2.576811e-04 -3.956431e-04  2.690647e-04
## [14,]  7.389107e-05 -1.953610e-05 -2.694766e-04 -3.965316e-04
## [15,]  3.424060e-05  6.087882e-05 -2.500208e-05 -2.698883e-04
## [16,] -3.279194e-06  3.548677e-05  6.140228e-05 -2.496266e-05
## [17,] -1.099940e-05  9.008374e-07  3.724265e-05  6.153455e-05
## [18,] -4.548281e-06 -9.270940e-06  1.626909e-06  3.729736e-05
## [19,]  7.599483e-07 -4.837077e-06 -9.392249e-06  1.617775e-06
## [20,]  1.621741e-06  1.436492e-07 -5.095961e-06 -9.411751e-06
## [21,]  5.948807e-07  1.395673e-06  4.868821e-08 -5.103113e-06
## [22,] -1.495391e-07  6.517087e-07  1.419544e-06  5.048748e-08
## [23,] -2.369356e-07 -5.949977e-08  6.895179e-07  1.422379e-06
## [24,] -7.635999e-08 -2.079199e-07 -4.733436e-08  6.904107e-07
## [25,]  2.707140e-08 -8.664705e-08 -2.122351e-07 -4.765380e-08
## [26,]  3.429155e-08  1.405284e-08 -9.202257e-08 -2.125497e-07
## [27,]  9.527272e-09  3.069249e-08  1.269058e-08 -9.198086e-08
## [28,] -4.666124e-09  1.129283e-08  3.138044e-08  1.269076e-08
## [29,] -4.749893e-09 -2.964435e-09  1.129282e-08  3.069263e-08
## [30,] -6.477792e-10 -4.749934e-09 -4.666158e-09  9.527335e-09
##               [,5]          [,6]          [,7]          [,8]
##  [1,] -2.137015e-02 -1.332261e-02 -9.485690e-04  3.210811e-03
##  [2,]  1.582826e-02 -1.630724e-02 -1.296213e-02 -2.168754e-03
##  [3,]  1.028063e-01  1.795501e-02 -1.615581e-02 -1.347469e-02
##  [4,]  2.037319e-01  1.029665e-01  1.796642e-02 -1.619442e-02
##  [5,]  2.493022e-01  2.032878e-01  1.029349e-01  1.807346e-02
##  [6,]  2.032878e-01  2.490253e-01  2.032681e-01  1.030016e-01
##  [7,]  1.029349e-01  2.032681e-01  2.490239e-01  2.032728e-01
##  [8,]  1.807346e-02  1.030016e-01  2.032728e-01  2.490078e-01
##  [9,] -1.613225e-02  1.811222e-02  1.030044e-01  2.032635e-01
## [10,] -1.349571e-02 -1.613137e-02  1.811228e-02  1.030042e-01
## [11,] -2.483309e-03 -1.350591e-02 -1.613210e-02  1.811474e-02
## [12,]  2.492873e-03 -2.488598e-03 -1.350629e-02 -1.613083e-02
## [13,]  1.930541e-03  2.493051e-03 -2.488585e-03 -1.350633e-02
## [14,]  2.715281e-04  1.932077e-03  2.493160e-03 -2.488955e-03
```

Processing math: 100%

```
## [15,] -3.953900e-04  2.722398e-04  1.932128e-03  2.492989e-03
## [16,] -2.699976e-04 -3.954581e-04  2.722351e-04  1.932144e-03
## [17,] -2.532935e-05 -2.702262e-04 -3.954744e-04  2.722901e-04
## [18,]  6.138292e-05 -2.542390e-05 -2.702330e-04 -3.954518e-04
## [19,]  3.732269e-05  6.139870e-05 -2.542279e-05 -2.702368e-04
## [20,]  1.671832e-06  3.735638e-05  6.140112e-05 -2.543083e-05
## [21,] -9.391924e-06  1.684182e-06  3.735727e-05  6.139822e-05
## [22,] -5.108092e-06 -9.395022e-06  1.683963e-06  3.735799e-05
## [23,]  4.261551e-08 -5.112921e-06 -9.395307e-06  1.684905e-06
## [24,]  1.419877e-06  4.119426e-08 -5.112905e-06 -9.395312e-06
## [25,]  6.912986e-07  1.420389e-06  4.119563e-08 -5.112924e-06
## [26,] -4.670682e-08  6.913010e-07  1.419880e-06  4.261457e-08
## [27,] -2.125487e-07 -4.765103e-08  6.904139e-07  1.422380e-06
## [28,] -9.202181e-08 -2.122334e-07 -4.733224e-08  6.895194e-07
## [29,]  1.405317e-08 -8.664669e-08 -2.079195e-07 -5.949924e-08
## [30,]  3.429160e-08  2.707111e-08 -7.636058e-08 -2.369360e-07
##                [,9]         [,10]         [,11]         [,12]
##  [1,]  1.864771e-03  4.197047e-05 -4.909702e-04 -2.544340e-04
##  [2,]  2.502153e-03  1.848821e-03  2.285509e-04 -3.942793e-04
##  [3,] -2.466436e-03  2.495453e-03  1.927197e-03  2.691674e-04
##  [4,] -1.349711e-02 -2.466941e-03  2.501356e-03  1.930257e-03
##  [5,] -1.613225e-02 -1.349571e-02 -2.483309e-03  2.492873e-03
##  [6,]  1.811222e-02 -1.613137e-02 -1.350591e-02 -2.488598e-03
##  [7,]  1.030044e-01  1.811228e-02 -1.613210e-02 -1.350629e-02
##  [8,]  2.032635e-01  1.030042e-01  1.811474e-02 -1.613083e-02
##  [9,]  2.490024e-01  2.032633e-01  1.030056e-01  1.811548e-02
## [10,]  2.032633e-01  2.490024e-01  2.032634e-01  1.030056e-01
## [11,]  1.030056e-01  2.032634e-01  2.490020e-01  2.032632e-01
## [12,]  1.811548e-02  1.030056e-01  2.032632e-01  2.490019e-01
## [13,] -1.613085e-02  1.811548e-02  1.030056e-01  2.032632e-01
## [14,] -1.350655e-02 -1.613086e-02  1.811554e-02  1.030057e-01
## [15,] -2.489055e-03 -1.350655e-02 -1.613083e-02  1.811555e-02
## [16,]  2.492998e-03 -2.489055e-03 -1.350655e-02 -1.613083e-02
## [17,]  1.932176e-03  2.492999e-03 -2.489063e-03 -1.350656e-02
## [18,]  2.723032e-04  1.932176e-03  2.492996e-03 -2.489065e-03
## [19,] -3.954539e-04  2.723032e-04  1.932177e-03  2.492996e-03
## [20,] -2.702414e-04 -3.954540e-04  2.723042e-04  1.932177e-03
```

```
## [21,] -2.543239e-05 -2.702414e-04 -3.954540e-04  2.723032e-04
## [22,]  6.139860e-05 -2.543242e-05 -2.702414e-04 -3.954539e-04
## [23,]  3.735797e-05  6.139817e-05 -2.543085e-05 -2.702368e-04
## [24,]  1.683941e-06  3.735723e-05  6.140110e-05 -2.542276e-05
## [25,] -9.395033e-06  1.684169e-06  3.735637e-05  6.139870e-05
## [26,] -5.108093e-06 -9.391920e-06  1.671835e-06  3.732268e-05
## [27,]  5.048860e-08 -5.103108e-06 -9.411748e-06  1.617766e-06
## [28,]  1.419545e-06  4.868865e-08 -5.095962e-06 -9.392254e-06
## [29,]  6.517087e-07  1.395672e-06  1.436476e-07 -4.837078e-06
## [30,] -1.495394e-07  5.948803e-07  1.621741e-06  7.599492e-07
##               [,13]         [,14]         [,15]         [,16]
##  [1,]  8.544282e-06  7.389106e-05  3.424058e-05 -3.279213e-06
##  [2,] -2.576812e-04 -1.953624e-05  6.087871e-05  3.548673e-05
##  [3,] -3.956432e-04 -2.694767e-04 -2.500214e-05  6.140228e-05
##  [4,]  2.690650e-04 -3.965313e-04 -2.698881e-04 -2.496259e-05
##  [5,]  1.930541e-03  2.715282e-04 -3.953900e-04 -2.699976e-04
##  [6,]  2.493051e-03  1.932077e-03  2.722396e-04 -3.954584e-04
##  [7,] -2.488585e-03  2.493160e-03  1.932128e-03  2.722349e-04
##  [8,] -1.350633e-02 -2.488955e-03  2.492989e-03  1.932144e-03
##  [9,] -1.613085e-02 -1.350655e-02 -2.489055e-03  2.492999e-03
## [10,]  1.811548e-02 -1.613086e-02 -1.350655e-02 -2.489055e-03
## [11,]  1.030056e-01  1.811554e-02 -1.613083e-02 -1.350655e-02
## [12,]  2.032632e-01  1.030057e-01  1.811555e-02 -1.613083e-02
## [13,]  2.490019e-01  2.032632e-01  1.030057e-01  1.811555e-02
## [14,]  2.032632e-01  2.490019e-01  2.032632e-01  1.030057e-01
## [15,]  1.030057e-01  2.032632e-01  2.490019e-01  2.032632e-01
## [16,]  1.811555e-02  1.030057e-01  2.032632e-01  2.490019e-01
## [17,] -1.613083e-02  1.811555e-02  1.030057e-01  2.032632e-01
## [18,] -1.350656e-02 -1.613083e-02  1.811555e-02  1.030057e-01
## [19,] -2.489065e-03 -1.350656e-02 -1.613083e-02  1.811555e-02
## [20,]  2.492996e-03 -2.489063e-03 -1.350655e-02 -1.613083e-02
## [21,]  1.932176e-03  2.492999e-03 -2.489055e-03 -1.350655e-02
## [22,]  2.723033e-04  1.932176e-03  2.492999e-03 -2.489055e-03
## [23,] -3.954516e-04  2.722901e-04  1.932144e-03  2.492989e-03
## [24,] -2.702330e-04 -3.954744e-04  2.722350e-04  1.932128e-03
## [25,] -2.542391e-05 -2.702263e-04 -3.954582e-04  2.722398e-04
## [26,]  6.138289e-05 -2.532940e-05 -2.699977e-04 -3.953900e-04
```

```
## [27,]  3.729733e-05  6.153453e-05 -2.496267e-05 -2.698883e-04
## [28,]  1.626903e-06  3.724265e-05  6.140229e-05 -2.500209e-05
## [29,] -9.270938e-06  9.008436e-07  3.548677e-05  6.087882e-05
## [30,] -4.548278e-06 -1.099939e-05 -3.279192e-06  3.424060e-05
##               [,17]         [,18]         [,19]         [,20]
##  [1,] -1.099940e-05 -4.548262e-06  7.599830e-07  1.621800e-06
##  [2,]  9.008720e-07 -9.270887e-06 -4.837035e-06  1.436842e-07
##  [3,]  3.724273e-05  1.627029e-06 -9.392117e-06 -5.095854e-06
##  [4,]  6.153445e-05  3.729718e-05  1.617689e-06 -9.411660e-06
##  [5,] -2.532933e-05  6.138297e-05  3.732275e-05  1.671877e-06
##  [6,] -2.702263e-04 -2.542393e-05  6.139866e-05  3.735629e-05
##  [7,] -3.954746e-04 -2.702331e-04 -2.542291e-05  6.140102e-05
##  [8,]  2.722904e-04 -3.954514e-04 -2.702366e-04 -2.543088e-05
##  [9,]  1.932176e-03  2.723034e-04 -3.954538e-04 -2.702413e-04
## [10,]  2.492999e-03  1.932176e-03  2.723030e-04 -3.954540e-04
## [11,] -2.489063e-03  2.492996e-03  1.932177e-03  2.723042e-04
## [12,] -1.350656e-02 -2.489064e-03  2.492996e-03  1.932177e-03
## [13,] -1.613083e-02 -1.350656e-02 -2.489065e-03  2.492996e-03
## [14,]  1.811556e-02 -1.613083e-02 -1.350656e-02 -2.489063e-03
## [15,]  1.030057e-01  1.811555e-02 -1.613083e-02 -1.350655e-02
## [16,]  2.032632e-01  1.030057e-01  1.811555e-02 -1.613083e-02
## [17,]  2.490019e-01  2.032632e-01  1.030057e-01  1.811554e-02
## [18,]  2.032632e-01  2.490019e-01  2.032632e-01  1.030056e-01
## [19,]  1.030057e-01  2.032632e-01  2.490019e-01  2.032632e-01
## [20,]  1.811554e-02  1.030056e-01  2.032632e-01  2.490020e-01
## [21,] -1.613086e-02  1.811548e-02  1.030056e-01  2.032634e-01
## [22,] -1.350655e-02 -1.613085e-02  1.811548e-02  1.030056e-01
## [23,] -2.488955e-03 -1.350633e-02 -1.613083e-02  1.811474e-02
## [24,]  2.493160e-03 -2.488585e-03 -1.350629e-02 -1.613210e-02
## [25,]  1.932077e-03  2.493051e-03 -2.488598e-03 -1.350591e-02
## [26,]  2.715281e-04  1.930541e-03  2.492873e-03 -2.483309e-03
## [27,] -3.965316e-04  2.690647e-04  1.930256e-03  2.501356e-03
## [28,] -2.694766e-04 -3.956431e-04  2.691674e-04  1.927197e-03
## [29,] -1.953611e-05 -2.576810e-04 -3.942792e-04  2.285510e-04
## [30,]  7.389106e-05  8.544273e-06 -2.544340e-04 -4.909702e-04
##               [,21]         [,22]         [,23]         [,24]
##  [1,]  5.949230e-07 -1.495092e-07 -2.369235e-07 -7.634986e-08
```

Processing math: 100%

```
## [2,]  1.395693e-06  6.517260e-07 -5.949728e-08 -2.079291e-07
## [3,]  4.874437e-08  1.419559e-06  6.895035e-07 -4.736053e-08
## [4,] -5.102912e-06  5.068507e-08  1.422503e-06  6.904643e-07
## [5,] -9.391917e-06 -5.108114e-06  4.258446e-08  1.419852e-06
## [6,]  1.684077e-06 -9.395088e-06 -5.112937e-06  4.120073e-08
## [7,]  3.735723e-05  1.683980e-06 -9.395272e-06 -5.112887e-06
## [8,]  6.139805e-05  3.735782e-05  1.684791e-06 -9.395374e-06
## [9,] -2.543235e-05  6.139862e-05  3.735797e-05  1.683928e-06
## [10,] -2.702413e-04 -2.543226e-05  6.139833e-05  3.735734e-05
## [11,] -3.954540e-04 -2.702415e-04 -2.543092e-05  6.140106e-05
## [12,]  2.723032e-04 -3.954540e-04 -2.702369e-04 -2.542286e-05
## [13,]  1.932176e-03  2.723033e-04 -3.954517e-04 -2.702330e-04
## [14,]  2.492999e-03  1.932176e-03  2.722900e-04 -3.954745e-04
## [15,] -2.489055e-03  2.492998e-03  1.932144e-03  2.722350e-04
## [16,] -1.350655e-02 -2.489055e-03  2.492989e-03  1.932128e-03
## [17,] -1.613086e-02 -1.350655e-02 -2.488955e-03  2.493160e-03
## [18,]  1.811548e-02 -1.613085e-02 -1.350633e-02 -2.488585e-03
## [19,]  1.030056e-01  1.811548e-02 -1.613083e-02 -1.350629e-02
## [20,]  2.032634e-01  1.030056e-01  1.811474e-02 -1.613210e-02
## [21,]  2.490024e-01  2.032633e-01  1.030042e-01  1.811228e-02
## [22,]  2.032633e-01  2.490024e-01  2.032635e-01  1.030044e-01
## [23,]  1.030042e-01  2.032635e-01  2.490078e-01  2.032728e-01
## [24,]  1.811228e-02  1.030044e-01  2.032728e-01  2.490239e-01
## [25,] -1.613137e-02  1.811222e-02  1.030016e-01  2.032681e-01
## [26,] -1.349571e-02 -1.613225e-02  1.807346e-02  1.029349e-01
## [27,] -2.466941e-03 -1.349711e-02 -1.619442e-02  1.796642e-02
## [28,]  2.495453e-03 -2.466436e-03 -1.347469e-02 -1.615581e-02
## [29,]  1.848821e-03  2.502153e-03 -2.168754e-03 -1.296213e-02
## [30,]  4.197046e-05  1.864771e-03  3.210811e-03 -9.485690e-04
##               [,25]         [,26]         [,27]         [,28]
## [1,]  2.707809e-08  3.429676e-08  9.529331e-09 -4.666460e-09
## [2,] -8.666065e-08  1.404449e-08  3.068943e-08  1.129240e-08
## [3,] -2.122580e-07 -9.203614e-08  1.268371e-08  3.137689e-08
## [4,] -4.763129e-08 -2.125290e-07 -9.195515e-08  1.271323e-08
## [5,]  6.912820e-07 -4.671762e-08 -2.125547e-07 -9.202502e-08
## [6,]  1.420395e-06  6.912971e-07 -4.765624e-08 -2.122346e-07
## [7,]  4.120045e-08  1.419882e-06  6.904193e-07 -4.732469e-08
```

Processing math: 100%

```
##  [8,] -5.112949e-06  4.260387e-08  1.422366e-06  6.894986e-07
##  [9,] -9.395050e-06 -5.108103e-06  5.048852e-08  1.419549e-06
## [10,]  1.684219e-06 -9.391907e-06 -5.103098e-06  4.870685e-08
## [11,]  3.735635e-05  1.671813e-06 -9.411762e-06 -5.095967e-06
## [12,]  6.139864e-05  3.732265e-05  1.617753e-06 -9.392261e-06
## [13,] -2.542391e-05  6.138291e-05  3.729736e-05  1.626916e-06
## [14,] -2.702263e-04 -2.532935e-05  6.153458e-05  3.724268e-05
## [15,] -3.954581e-04 -2.699976e-04 -2.496264e-05  6.140230e-05
## [16,]  2.722399e-04 -3.953900e-04 -2.698883e-04 -2.500209e-05
## [17,]  1.932077e-03  2.715282e-04 -3.965316e-04 -2.694766e-04
## [18,]  2.493051e-03  1.930541e-03  2.690647e-04 -3.956431e-04
## [19,] -2.488598e-03  2.492873e-03  1.930256e-03  2.691674e-04
## [20,] -1.350591e-02 -2.483310e-03  2.501356e-03  1.927197e-03
## [21,] -1.613137e-02 -1.349571e-02 -2.466941e-03  2.495453e-03
## [22,]  1.811222e-02 -1.613225e-02 -1.349711e-02 -2.466436e-03
## [23,]  1.030016e-01  1.807346e-02 -1.619442e-02 -1.347469e-02
## [24,]  2.032681e-01  1.029349e-01  1.796642e-02 -1.615581e-02
## [25,]  2.490253e-01  2.032878e-01  1.029665e-01  1.795501e-02
## [26,]  2.032878e-01  2.493022e-01  2.037319e-01  1.028063e-01
## [27,]  1.029665e-01  2.037319e-01  2.500147e-01  2.034750e-01
## [28,]  1.795501e-02  1.028063e-01  2.034750e-01  2.501073e-01
## [29,] -1.630724e-02  1.582826e-02  9.939493e-02  2.047053e-01
## [30,] -1.332261e-02 -2.137015e-02  7.707093e-03  1.023238e-01
##               [,29]         [,30]
##  [1,] -4.750135e-09 -6.479938e-10
##  [2,] -2.964440e-09 -4.750248e-09
##  [3,]  1.129058e-08 -4.667089e-09
##  [4,]  3.070419e-08  9.528730e-09
##  [5,]  1.405172e-08  3.429135e-08
##  [6,] -8.664438e-08  2.707364e-08
##  [7,] -2.079135e-07 -7.635804e-08
##  [8,] -5.951764e-08 -2.369452e-07
##  [9,]  6.517115e-07 -1.495392e-07
## [10,]  1.395689e-06  5.948886e-07
## [11,]  1.436473e-07  1.621742e-06
## [12,] -4.837083e-06  7.599469e-07
## [13,] -9.270934e-06 -4.548280e-06
```

Processing math: 100%

```
## [14,]  9.008494e-07 -1.099940e-05
## [15,]  3.548677e-05 -3.279194e-06
## [16,]  6.087882e-05  3.424060e-05
## [17,] -1.953611e-05  7.389107e-05
## [18,] -2.576811e-04  8.544268e-06
## [19,] -3.942792e-04 -2.544340e-04
## [20,]  2.285509e-04 -4.909702e-04
## [21,]  1.848821e-03  4.197046e-05
## [22,]  2.502153e-03  1.864771e-03
## [23,] -2.168754e-03  3.210811e-03
## [24,] -1.296213e-02 -9.485690e-04
## [25,] -1.630724e-02 -1.332261e-02
## [26,]  1.582826e-02 -2.137015e-02
## [27,]  9.939493e-02  7.707093e-03
## [28,]  2.047053e-01  1.023238e-01
## [29,]  2.664417e-01  2.435908e-01
## [30,]  2.435908e-01  3.590121e-01
```

bandwidth가 커질수록, 데이터에서 먼 위치의 값들도 고려하게 되므로, diagonal entry의 hat matrix value는 줄어들고, diagonal entry에서 상대적으로 먼 위치에서의 값이 증가함을 알 수 있다.

Q3

universal kriging을 이용해서 Q1과 같은 과정을 반복하자.

다만, polynomial equation의 degree도 같이 optimized되어야 하므로, 이러한 과정을 진행해줄 수 있도록 위에서 구현한 함수를 살짝 바꿀것이다.

먼저,

```
mykrig2<-function(nd, ne, xx, yy, ex, bw, np)
{
# (1)
  full <- matrix(0, nd, nd)
  full[lower.tri(full)] <- dist(xx)
  dist1 <- full + t(full)
  cc <- exp( - dist1^2/bw^2)
```

```r
  ll <- t(chol(cc))
  llr <- solve(ll)
# (3)
  powerf <- function(jj, x1)
  {
    pw <- x1^jj
    return(pw)
  }
  gg <- apply(matrix(c(0:(np - 1)), nrow = 1), 2, powerf,
    x1 = xx)
  aa <- qr(llr %*% gg)
  qq <- qr.Q(aa)
  rr <- qr.R(aa)
# (4)
  bb <- t(qq) %*% llr %*% yy
  beta1 <- solve(rr[1:np, ], bb[1:np])
# (5)
  dvector <- function(exs, xx, bw)
  {
    dv <- exp( - (exs - xx)^2/bw^2)
    return(dv)
  }
  dmat <- apply(matrix(ex, nrow = 1), 2, dvector, xx = xx,
    bw = bw)
# (6)
  ccr <- solve(cc)
  almat <- ccr %*% dmat
# (7)
  mm <- as.vector(beta1 %*% t(gg))
  exmat <- t(apply(matrix(c(0:(np - 1)), nrow = 1), 2,
    powerf, x1 = ex))
  ey <- beta1 %*% exmat + as.vector(t(almat) %*% (yy - mm))
# (8)
  return(ey)
}
```


[illegible] 4장 (G) 부분에 해당하는 코드를 이용하여 polynomial 함수를 이용한 universal kriging을 구해주는 함수를 구현하였다.

즉, 다음과 같은 모델을 사용한다.

$y = \sum_{j=0}^{p-1}\beta_j g_j(x) + e$, 여기서 $g_j(x) = x^j$가 된다.

여기서 kriging에 사용하는 theoretical correlogram은

gaussian correlogram으로, 다음과 같다.

$Correlo(r) = \exp(-(\frac{r}{\delta})^2)$

위 함수는 데이터인 xx, yy와 데이터의 개수인 nd, 추정할 데이터의 위치인 ex와 그 개수인 ne, 그리고 위의 theoretical correlogram에서 사용할 δ인 bw와 polynomial의 차수인 np (정확히는 np-1 차수의 polynomial을 사용함)을 받아 ex의 위치에서 함수로 polynomial function을 사용한, universal kriging된 결과를 리턴해주는 함수이다.

이제, 위 함수를 사용하여 Q1과 같이, polynomial과 bandwidth를 바꿔가면서 CV를 가장 작게만드는 차수와 bandwidth를 찾고, kriging된 결과를 출력해보자.

먼저, bandwidth들과 polynomial들의 차수를 받아서 각 값에 해당하는 CV값들을 계산해주는 함수를 만든다.

```
cv_univ_pol_krig<-function(xdata, ydata, bws, pols)
{
    nd <- length(ydata)

    ydata <- ydata[order(xdata)]
    xdata <- xdata[order(xdata)]
    #sort by ascending order. order of ydata, xdata is important!!


    get_univ_pol_krig_cv <- function(bw, pol, x1, y1)
    {
        cv <- 0
        nd <- length(y1)

        if(nd < 3) return(NULL) # we can't get cv

        for(i in seq(from = 2, to = nd-1))
        {
            xdata <- x1[-i]
            ydata <- y1[-i]
```

```
            remain_xdata <- x1[i]
            remain_ydata <- y1[i]
            estim_y <- mykrig2(nd-1, 1, xdata, ydata, as.vector(remain_xdata), bw, pol)
            cv <- cv + sum((remain_ydata - estim_y)**2)
        }

        cv <- cv / (nd - 2)
        return(cv)
    }

    cvmat <- matrix(nrow = length(bws), ncol = length(pols))
    for(i in seq_len(length(pols)))
    {
       cvlst <- lapply(as.list(bws),get_univ_pol_krig_cv, pol = pols[i], x1 = xdata, y1 = ydata)
       cvlst <- unlist(cvlst)
       cvmat[,i] <- cvlst
    }

    return(cvmat)
}
```

아래함수는 bandwidth, polynomial 들과 cv matrix를 받아서 그래프를 출력해주는 함수다.

```
plot_2d_cv <- function(bandwidth_array, pol_array, cv_mat, description = "")
{
    par(mfrow = c(1, 1))
    persp(cv_mat, xlab = "bandwidth", ylab = "poly_degree", zlab = "cv", theta = -30, phi = 20, lab = c(3,3,3))

}
```

plot univ krig graph함수는 xdata, ydata, bw, pol을 받아 xdata의 위치에서, 원래 데이터를 점으로, estimate된 결과를 파란 선으로 그려준다.

```
plot_univ_krig_graph <- function(xdata, ydata, bw, pol, description = "")
{
    par(mfrow=c(1,1))
    estimated_y <- mykrig2(length(xdata), length(xdata), xdata, ydata, xdata, bw, pol)
```

```
    plot(xdata, ydata , xlab = "X", ylab = "Y", type = "p", main = description, sub = sprintf("using bw : %f, pol = %d", bw, pol))
    lines(xdata, estimated_y, col = "blue")
}
```

마지막으로, xdata와 ydata, bandwidth, polynomial degree 목록을 받아서 이중 가장 CV값이 낮은 parameter를 사용한 kriging 그래프를 출력해주는 함수를 다음과 같이 구현하였다.

```
full_test_univ_kriging <- function(xdata, ydata, smtparam_array, pol_array, description = "")
{
    cv_mat <- cv_univ_pol_krig(xdata, ydata, smtparam_array, pol_array)
    plot_2d_cv(smtparam_array, pol_array, cv_mat, description)

    smt_idx <- which(cv_mat == min(cv_mat, na.rm = TRUE),arr.ind = TRUE)

    if(length(smt_idx) > 1)
    {
        smt_idx <- smt_idx[1,]
    }

    bw_idx <- smt_idx[1]
    pol_idx <- smt_idx[2]

    best_bw <- smtparam_array[bw_idx]
    best_pol <- pol_array[pol_idx]

    plot_univ_krig_graph(xdata, ydata, best_bw, best_pol, description = description)
}
```

problem 2.3의 chapter 2의 데이터는 eqispaced data로, x값으로 1,2…의 값을 가지게 된다.

해당 데이터를 생성하고, 위 함수를 이용하여 가장 CV값을 낮게 만드는 δ를 찾자.

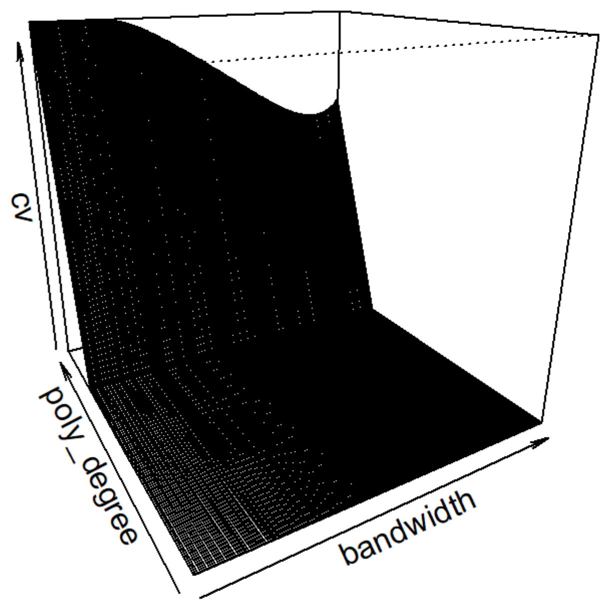
delta의 값으로는 0.01~2까지, 0.01 간격의 값을 주었고 polynomial은 0차~4차까지를 사용하였다.

```
ydata <- c(9.6, 12.8, 14.6, 15.6 ,15.5 ,15.1, 15.6, 13.8, 13.9, 16.1, 17.3, 18, 19.9, 20, 19.9, 18.2, 15.8, 11.2, [illegible], 15.8, 16.7, 17.5, 13.7, 15.7, 20.6, 21.2, 16.7, 16, 20.7, 17.6)
```

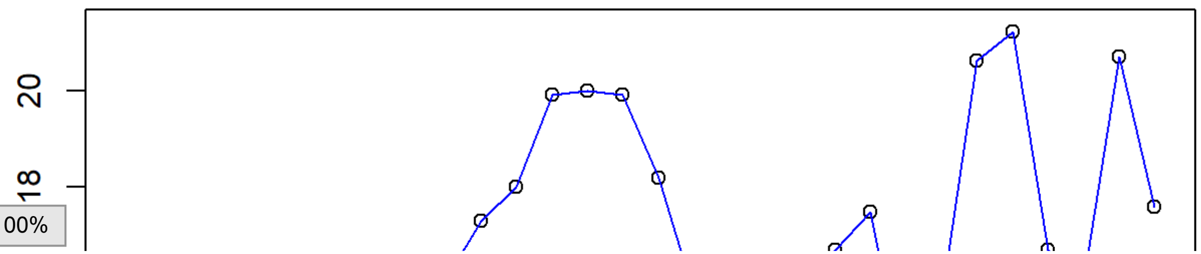
Processing math: 100%

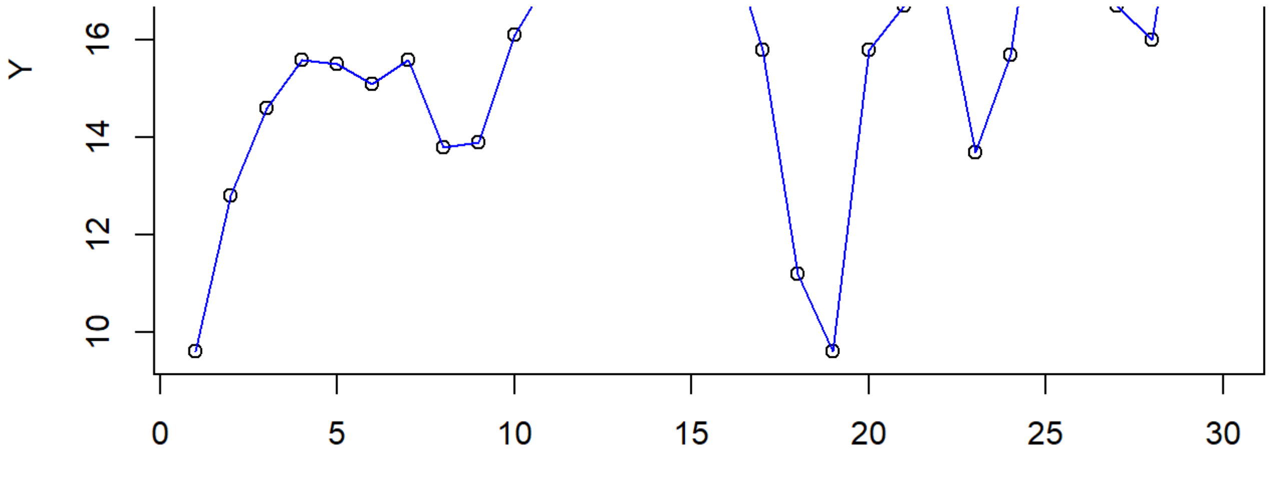
```
xdata <- seq_len(length(ydata))


full_test_univ_kriging(xdata, ydata, seq(0.01,2,by = 0.01),seq(1,5) ,description = "delta = 0.01~2, by 0.01, pol
 = 0~4")
```

Processing math: 100%



delta = 0.01~2, by 0.01, pol = 0~4





pol-1이 degree 이므로, 결과로 알수 있듯이, degree가 0차, bandwidth가 1.31일때 가장 CV값이 낮음을 알 수 있다.

즉, ordinary kriging이 가장 효율적이고, 이때의 bandwidth가 1.31이 되어야함을 알 수 있었다.

Q4

4-(a)

$$X = \begin{pmatrix} 1 & X_{11} & X_{11}^2 & \dots & X_{11}^p & X_{12} & X_{12}^2 & \dots & X_{12}^q \\ 1 & X_{21} & X_{21}^2 & \dots & X_{21}^p & X_{22} & X_{22}^2 & \dots & X_{22}^q \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & X_{n1}^2 & \dots & X_{n1}^p & X_{n2} & X_{n2}^2 & \dots & X_{n2}^q \end{pmatrix}$$

$$\beta = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_p \\ d_1 \\ \vdots \\ d_q \end{pmatrix}$$

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

라 하면, (4.116)의 모델에서 y는 다음과 같이 쓸 수 있다.

$y = X\beta + e$

이때, e_i는 서로 독립이므로, $e \sim (0, \sigma^2 I_n)$이 성립한다. 즉, 평균이 0이고, 분산이 단위행렬의 실수배가 된다.

따라서, normal equation을 구하기 위해

$E_{p4} = ||y - X||^2 $을 최소화하는 β를 구하면 되는데, 이는 우리가 linear regression에서 많이 접해왔던 형태이므로, 이를 β로 미분한 값이 0이되는 β에서 최솟값을 가짐을 알고,

$||y - X\beta||^2 = (y - X\beta)^t(y - X\beta) = y^t y - 2\beta^t X^t y - \beta^t X^t X\beta$에서,

위 식을 β로 미분해서 0이되는 β는

$\frac{\partial E_{p4}}{\partial \beta} = -2X^t y + 2X^t X\beta = 0$ 에서, normal equation은 다음과 같은 형태이다. $X^t X\beta = X^t y$

추가적으로, X^tX가 non-singular matrix이면 $\hat{\beta} = (X^tX)^{-1}X^ty$임도 알고있다.

4-(b)

위 normal equation을 이용해서, regression coefficient인 $\hat{\beta}$를 구하는 함수를 짜고,

나아가 해당하는 hat matrix까지 구해주는 함수를 짜자.

hat matrix의 경우, $\hat{y} = X\hat{\beta} = X(X^tX)^{-1}X^ty$에서,

$H = X(X^tX)^{-1}X^t$임을 안다.

poly함수의 인자로 raw = TRUE를 주면, 받은 데이터를 degree만큼 제곱해서 확장한 matrix를 리턴해준다.

예를들어 poly(c(1,2,3), deg = 4, raw = TRUE)를 실행하면,

첫번째 column은 1,2,3 , 두번째 column은 1,4,9, 세번째 column은 1,8,27, 네번째 column은 1,16,81인 matrix를 리턴해주므로

다음함수는 이를 이용하여 design matrix를 만들어서 리턴해준다.

4-c에서도 사용할 수 있도록, 관련 기능을 추가하여 구현하였다. (r_0(.) =)

```
get_design_mat_two_predictor <- function(x1data, x2data, deg1, deg2, raw = FALSE)
{
    if(raw == FALSE)
    {
        X <- matrix(rep(1/sqrt(length(x1data)), length(x1data)), nrow = length(x1data), ncol = 1)
        d1 <- poly(x1data, degree = deg1, raw = FALSE)
        d2 <- poly(x2data, degree = deg2, raw = FALSE)
        X <- cbind(X,d1,d2)
        return(X)
    }
    else
    {
        X <- matrix(rep(1, length(x1data)), nrow = length(x1data), ncol = 1)
        d1 <- poly(x1data, degree = deg1, raw = TRUE)
        d2 <- poly(x2data, degree = deg2, raw = TRUE)
        X <- cbind(X,d1,d2)
        return(X)
```

Processing math: 100%

```
    }
}
```

다음 함수는 x1data와 x2data, ydata를 받고, 각 x1,x2의 차수를 받아

위와 같은 normal equation을 이용해 $\hat{beta}$를 구해준다.

```
additive_reg_coef_raw <- function(x1data, x2data, ydata, deg1, deg2)
{
    # first, make design matrix
    X <- get_design_mat_two_predictor(x1data, x2data, deg1, deg2, raw = TRUE)

    # next, get beta = (X'X)^-1X'y

    y <- matrix(ydata, nrow = length(ydata), ncol = 1)

    beta = solve(t(X) %*% X, t(X) %*% y)
    # solve(A,b) return A^-1 b.

    return(beta)
}
```

또한, hat matrix역시 다음과 같이 구할 수 있다. hat matrix를 구할때는 ydata는 필요하지 않다.

```
additive_hat_mat_raw <- function(x1data, x2data, deg1, deg2)
{
    # first, make design matrix
    X <- get_design_mat_two_predictor(x1data, x2data, deg1, deg2, raw = TRUE)

    # next, get H = X(X'X)^-1X'

    hat = X %*% solve(t(X) %*% X) %*% t(X)
    # solve(A) return A^-1.

    return(hat)
}
```

이렇게 구현한 함수를 다음과 같이 테스트 해볼 수 있다.

$Y_i = sin(0.2\pi X_{i1}) + 0.05X_{i2}^2 - 0.4X_{i2} + 3 + e_i$, $e_i \sim N(0, 0.1^2)$이고, e_i들의 분포가 서로 독립인 데이터를 이용해서,

$Y_i = c_0 + \sum_{i=1}^{p} c_i X_{i1}^i + \sum_{i=1}^{q} d_i X_{i2}^i$

p = 4, q = 2

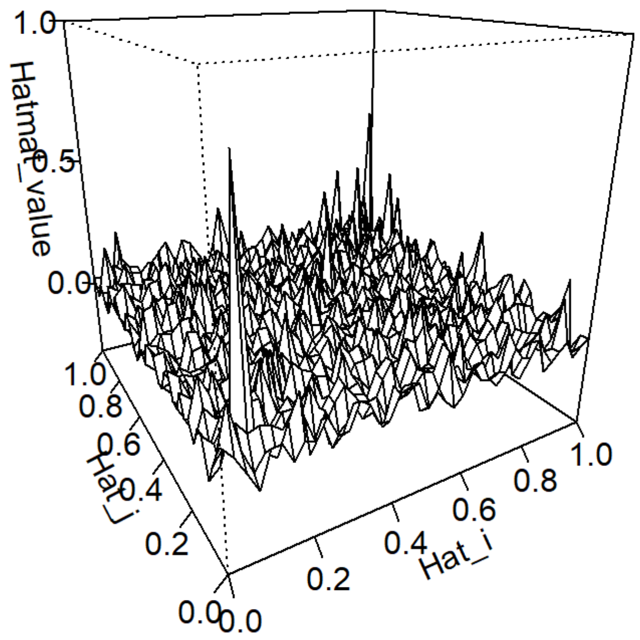
와 같은 모델을 이용하여 추정한 coefficient들과 hat matrix는 다음과 같다.

```
# (1)
nd <- 40
# (2)
set.seed(100)
xx1 <- runif(nd, min =0, max = 10)
xx2 <- runif(nd, min =0, max = 10)
yy <- sin(0.2 * pi * xx1) + xx2^2 * 0.05 - xx2 * 0.4 +
  3 + rnorm(nd, mean = 0, sd = 0.1)
# (3)
deg1 <- 4
deg2 <- 2
reg_coef_raw <- additive_reg_coef_raw(xx1, xx2, yy, deg1, deg2)
reg_hat_raw <- additive_hat_mat_raw(xx1, xx2, deg1, deg2)

reg_coef_raw
```

```
##             [,1]
##    2.3389918834
## 1  1.4373069398
## 2 -0.3894209717
## 3  0.0243278717
## 4  0.0001102478
## 1 -0.3545429132
## 2  0.0466388309
```

Processing math: 100%

```
:mat(reg_hat_raw)
```



4-(c)

$$
\begin{pmatrix}
r_0(X_{11}) & r_1(X_{11}) & r_2(X_{11}) & \cdots & r_p(X_{11}) & s_1(X_{12}) & \cdots & s_q(X_{12}) \\
r_0(X_{21}) & r_1(X_{21}) & r_2(X_{21}) & \cdots & r_p(X_{21}) & s_1(X_{22}) & \cdots & s_q(X_{22}) \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
r_0(X_{n1}) & r_1(X_{n1}) & r_2(X_{n1}) & \cdots & r_p(X_{n1}) & s_1(X_{n2}) & \cdots & s_q(X_{n2})
\end{pmatrix}
$$

위와 같은 design matrix를 사용해야 한다. 이때, $r_0(.) = \frac{1}{\sqrt{n}}$, $< r_i, r_j >= \delta_{ij}$, $< s_i, s_j >= \delta_{ij}$, $\delta_{ij} = I(i = j)$ 가 성립한다. 즉, r_i들과 s_i들은 orthogoanl polynomial이다.

이때, poly 함수를 이용하면, 각각 $r_j(X_{i1})$와 $s_j(X_{i2})$에 해당하는, orthogonal polynomial들의 값들을 생성해주므로, 위에서 구현한 get_design_mat_two_predictor에 raw 인자를 FALSE로 주면, 위와 같은 design matrix를 받아올 수 있다.

따라서, 위 design matrix를 사용하여 각각의 coefficient와 hat matrix를 리턴하는 함수를 짜면,

```
additive_reg_coef <- function(x1data, x2data, ydata, deg1, deg2)
{
    # first, make design matrix
    X <- get_design_mat_two_predictor(x1data, x2data, deg1, deg2, raw = FALSE)

    # next, get beta = (X'X)^-1X'y

    y <- matrix(ydata, nrow = length(ydata), ncol = 1)

    beta = solve(t(X) %*% X, t(X) %*% y)
    # solve(A,b) return A^-1 b.

    return(beta)
}
```

```
additive_hat_mat <- function(x1data, x2data, deg1, deg2)
{
    # first, make design matrix
    X <- get_design_mat_two_predictor(x1data, x2data, deg1, deg2, raw = FALSE)
```

Processing math: 100%

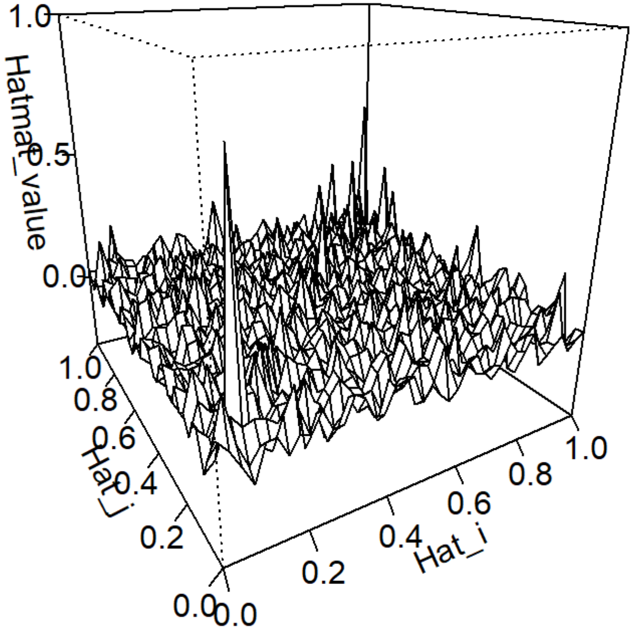
```
    # next, get H = X(X'X)^-1X'

    hat = X %*% solve(t(X) %*% X) %*% t(X)
    # solve(A) return A^-1.

    return(hat)
}
```

```
# (1)
nd <- 40
# (2)
set.seed(100)
xx1 <- runif(nd, min =0, max = 10)
xx2 <- runif(nd, min =0, max = 10)
yy <- sin(0.2 * pi * xx1) + xx2^2 * 0.05 - xx2 * 0.4 +
  3 + rnorm(nd, mean = 0, sd = 0.1)
# (3)
deg1 <- 4
deg2 <- 2
reg_coef <- additive_reg_coef(xx1, xx2, yy, deg1, deg2)
reg_hat <- additive_hat_mat(xx1, xx2, deg1, deg2)

reg_coef
```

```
##           [,1]
##    16.26000948
## 1 -3.80061611
## 2  0.76799668
## 3  2.59191361
## 4  0.02656535
## 1  2.26023827
## 2  1.62748066
```

Processing math: 100%

```
plot_hatmat(reg_hat)
```



각각의 coefficient와 hat matrix를 plot한 결과는 위와 같다.

4-(d)

simulation data를 이용하여 계샀나 두 경우의 hat matrix가 같음을 보이자.

위에서 구한 두 hat matrix의 차이를 보이면 된다.

```
max(abs(reg_hat - reg_hat_raw))
```

```
## [1] 2.649991e-12
```

위와 같이, 두 hat matrix의 성분들의 차이의 절댓값중 가장 큰값이 $2.65 * 10^{-12}$로, 매우 작은 값임을 알 수 있다.

따라서 두 matrix의 각 성분들의 차이는 거의 없다고 봐도 될 정도임을 보였고,

즉, 두 hat matrix가 같음을 보였다.

Q5

4장 (M) 부분의 함수는 다음과 같다. (eval = FALSE로 실행되지 않는 코드임)

```
##(M) Derivation of a regression equation with the form
## of ACE by solving an eigenvalue problem using iterative calculation
aceit1<-function(nd, it, npx1, npx2, npy, xx1, xx2, yy)
{
# (1)
  yyst <- yy
# (2)
  powerf <- function(jj, x1)
  {
    pw <- x1^jj
    return(pw)
  }
  xxm1 <- apply(matrix(c(1:npx1), nrow = 1),
    2, powerf, x1 = xx1)
  xxm2 <- apply(matrix(c(1:npx2), nrow = 1),
    2, powerf, x1 = xx2)
  xxmat <- cbind(xxm1, xxm2)
  xxmean <- apply(xxmat, 2, mean)
  xxmat <- sweep(xxmat, 2, xxmean)
  yymat <- apply(matrix(c(1:npy), nrow = 1),
    2, powerf, x1 = yy)
  yymean <- apply(yymat, 2, mean)
  yymat <- sweep(yymat, 2, yymean)
  hatx <- xxmat %*% solve (crossprod (xxmat)) %*% t(xxmat)
```

```
  haty <- yymat %*% solve(crossprod(yymat)) %*% t(yymat)
  hatyx <- haty %*% hatx
# (3)
  for(ii in 1 :it) {
    yyst <- hatyx %*% yyst
    yyst <- (sqrt(nd) * yyst)/sqrt(sum(yyst^2))
  }
# (4)
  return(yyst)
}
```

위 함수를 적당히 고쳐서, smoothing spline을 각 variable의 smoother로 사용하도록 고쳐보자.

ACE에서 가정하는 모형은 $\eta(Y_i) = m_1(X_{i1}) + m_2(X_{i2}) + \epsilon_i$ 이므로,

이 모형을 fit하기 위해 addtive model을 이용해서, ACE 알고리즘을 실행하면,

1. 표준화된 Y_i들을 $Y_i^* = \frac{Y_i - \bar{Y}_i}{sd(Y_i)}$라 하고,
2. 데이터 $\{(X_{i1}, X_{i2}, Y_i^*)\}$을 사용하여 $m_1(X_{i1}) + m_2(X_{i2})$을 y_i^* 로, $\hat{Y}_i^*$ 을 추정한다.
3. 그 후, 데이터 $\{(Y_i, \hat{Y}_i^*)\}$을 사용하여 $\eta(Y_i)$를 $\hat{y}_i^*$ 로, $\tilde{Y}_i^*$ 를 추정하고,
4. 이 $\tilde{Y}_i^*$ 을 다시 표준화해서, Y_i^* 을 다시 정의한다.

그 후 2~4를 충분히 반복해서, 최종 추정값으로 $Y_i^{*(k)}$ 를 사용하게 된다.

이때, 2번을 하는 과정은 Y^*에 H_X를 곱하는것과 같고,

3번을 하는 과정은 $H_X y^*$에 H_Y를 곱하는것과 같으므로,

결국 2~3번 과정은 $H_Y H_X y^*$을 적용하는것과 같다.

Processing math: 100%

또한, 여기서 standard deviation을 다음과 같이 쓸수 있으므로, $sd(\{\tilde{Y}_i\}) = \frac{||\tilde{y}^*||}{\sqrt{n}}$

위 코드를 통해 ACE 알고리즘이 돌아갈 수 있는것이다.

이 과정을 위 함수를 수정해서 적용해보자.

hatmatrix를 사용해야 하므로, smoothing spline의 hat matrix를 구해야 한다.

Y_i측면에서 구하는것은 predictor가 1개이므로 구하기 쉽지만, X_{i1} , X_{i2}인 case에서 smoothing spline의 hat matrix를 구하려면, 약간의 조작이 필요하다.

이때, $\hat{y} = Hy$에서 $y = e_i$, e_i는 ith member가 1, 그 외의 성분이 모두 0 인 n x 1 vector일때, 결과로 나오는것이 H의 ith column이므로, 이를 이용하여 hat matrix를 구할것이다.

아래 함수는 xdata1, xdata2, lambx1, lambx2를 받아 $m_1(x_1) + m_2(x_2)$에서, 각각의 $m(x)$가 smoothing spline일때, 이에 해당하는 hat matrix를 리턴해준다.

이때 $m_1(x)$의 smoothing parameter는 lambx1, $m_2(x)$의 smoothing parameter는 lambx2이다.

```
hat_smoothing_spline_two <- function(xdata1, xdata2, lambx1, lambx2)
{
    nd <- length(xdata1)
    I_n <- diag(nd)
    hatmat <- matrix(nrow = nd, ncol = nd)

    for(i in 1:nd) {
        ydata <- I_n[, i]

        fit.sp1 <- smooth.spline(xdata1, ydata, lambda = lambx1, all.knots = TRUE)
        fit.sp2 <- smooth.spline(xdata2, ydata, lambda = lambx2, all.knots = TRUE)

        pred1 <- predict(fit.sp1, xdata1)
        pred2 <- predict(fit.sp2, xdata2)

        estim_y <- pred1$y + pred2$y
        hatmat[,i] <- estim_y
    }
    return(hatmat)
}
```

마찬가지로, 아래 함수는 ydata, lambda를 받아 $\eta(y)$에서, $\eta(y)$가 smoothing spline일때, 이에 해당하는 hat matrix를 리턴해준다.

```
hat_smoothing_spline_one <- function(xdata, lamb)
{
    nd <- length(xdata)
    I_n <- diag(nd)
    hatmat <- matrix(nrow = nd, ncol = nd)

    for(i in 1:nd) {
        ydata <- I_n[, i]

        fit.spl <- smooth.spline(xdata, ydata, lambda = lamb, all.knots = TRUE)

        pred <- predict(fit.spl, xdata)

        estim_y <- pred$y
        hatmat[,i] <- estim_y
    }
    return(hatmat)
}
```

따라서, 위와 같이 hat matrix를 만들 수 있으므로,

이제 맨 처음 도입했던 함수를 적절히 고치자.

아래 함수는 data의 개수 nd, iteration 회수 it, data에 해당하는 xx1, xx2, yy를 받고,

x1, x2, y에 해당되는 labmda를 받아, iteration 회수만큼 ACE 알고리즘을 돌린 결과를 리턴해준다.

```
##(M) Derivation of a regression equation with the form
## of ACE by solving an eigenvalue problem using iterative calculation
aceit1_smooth<-function(nd, it, xx1, xx2, yy, lambx1, lambx2, lamby)
{
# (1)
  yyst <- (yy - mean(yy))/sqrt(sum((yy - mean(yy))^2)/nd) # need to calculate this!

<- hat_smoothing_spline_two(xx1, xx2, lambx1, lambx2)
```

```
  haty <- hat_smoothing_spline_one(yy, lamby)
  hatyx <- haty %*% hatx
# (3)
  for(ii in 1:it) {
    yyst <- hatyx %*% yyst
    yyst <- (sqrt(nd) * yyst)/sqrt(sum(yyst^2))
  }
# (4)
  return(yyst)
}
```

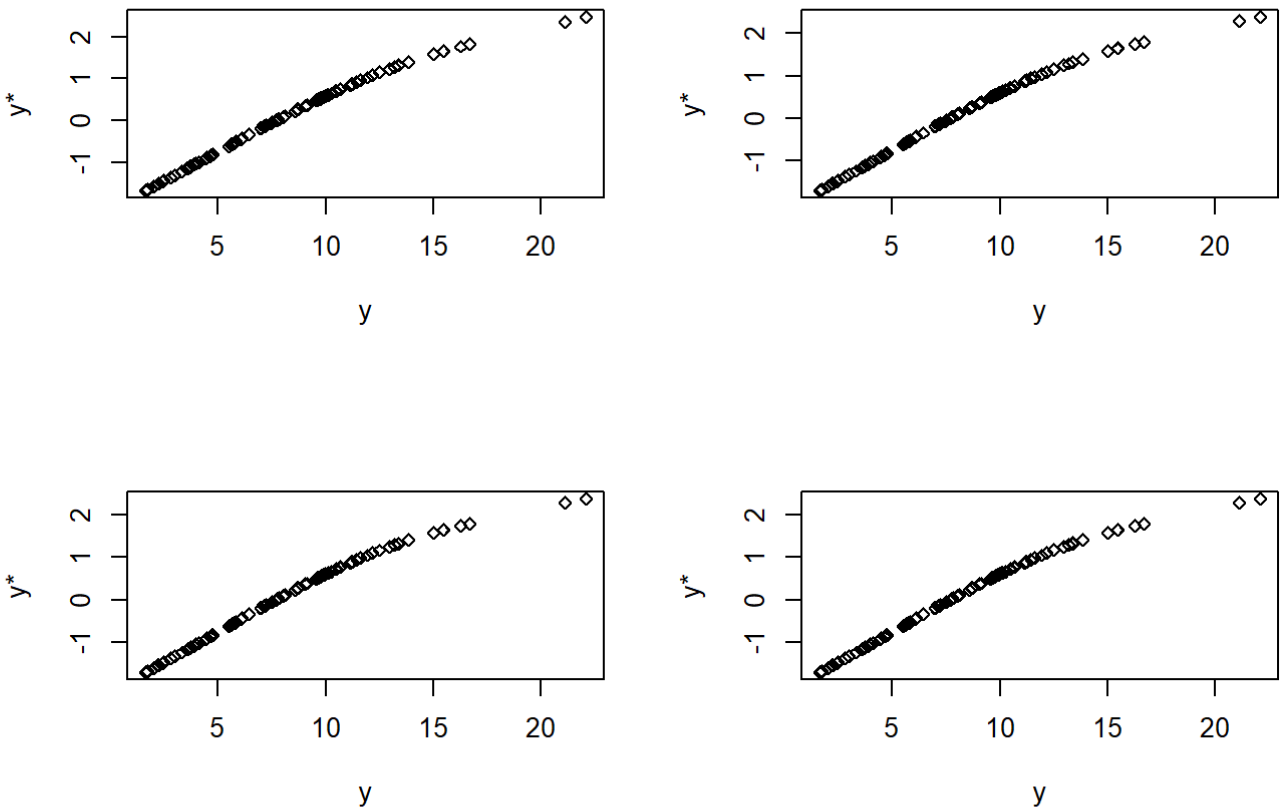
위와 같이 고칠 수 있다.

이제, 이렇게 만들어진 object를 사용해서 test를 해보자.

책에서 aceit1 부분을 test하는 코드를 약간 바꿔 사용하였다.

각 lambda로는 모두 0.01을 사용하였다.

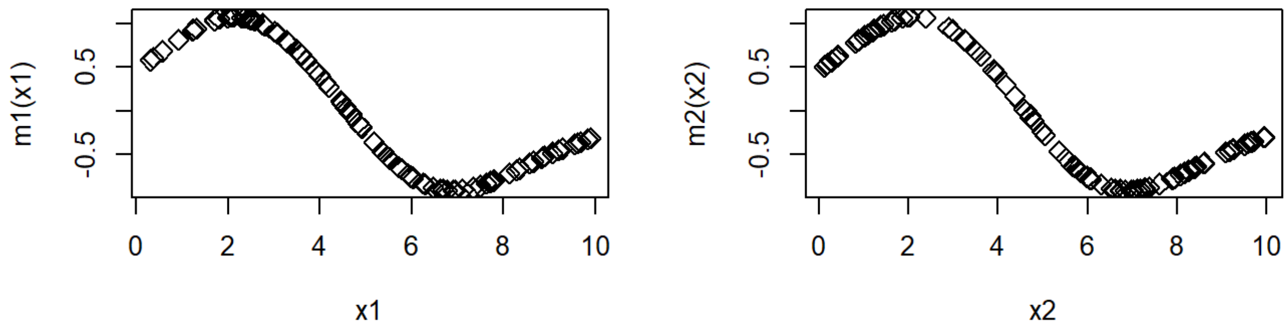
```
nd <- 100
set.seed(100)
xx1 <- runif(nd, min =0, max = 10)
xx2 <- runif(nd, min =0, max = 10)
yy <- (sin(0.2 * pi * xx1) + xx2^2 * 0.05 - xx2 * 0.4
  + 3 + rnorm(nd, mean =0, sd = 0.1))^2
# (2)
lambx1 <- 0.01
lambx2 <- 0.01
lamby <- 0.01
# (3)
par(mfrow = c(2, 2))
# (4)
for(it in 1:4) {
  yyst <- aceit1_smooth(nd, it, xx1, xx2, yy, lambx1, lambx2, lamby)
  plot(yy, yyst, type = "n", xlab = "y", ylab = "y*")
  points(yy, yyst, cex = 0.7, pch =5)
  }
```




```
fit.sp1 <- smooth.spline(xx1, yyst, lambda = lambx1, all.knots = TRUE)
fit.sp2 <- smooth.spline(xx1, yyst, lambda = lambx2, all.knots = TRUE)

fit.tmd1 <- predict(fit.sp1, xx1)
fit.tmd2 <- predict(fit.sp2, xx2)
eyd1 <- fit.tmd1$y
eyd2 <- fit.tmd2$y
yhat <- eyd1 + eyd2
plot(xx1, eyd1, type = "n", xlab = "x1", ylab = "m1(x1)")
```

Processing math: 100%

```
points(xx1, eyd1, cex = 1.2, pch =5)
plot(xx2, eyd2, type = "n", xlab = "x2", ylab = "m2(x2)")
points(xx2, eyd2, cex = 1.2, pch =5)
```



iterate 횟수에 따른 y의 추정값과, iterate가 4번 된 결과로 나온 $m_1(x_1), m_2(x_2)$의 결과가 출력되었다.

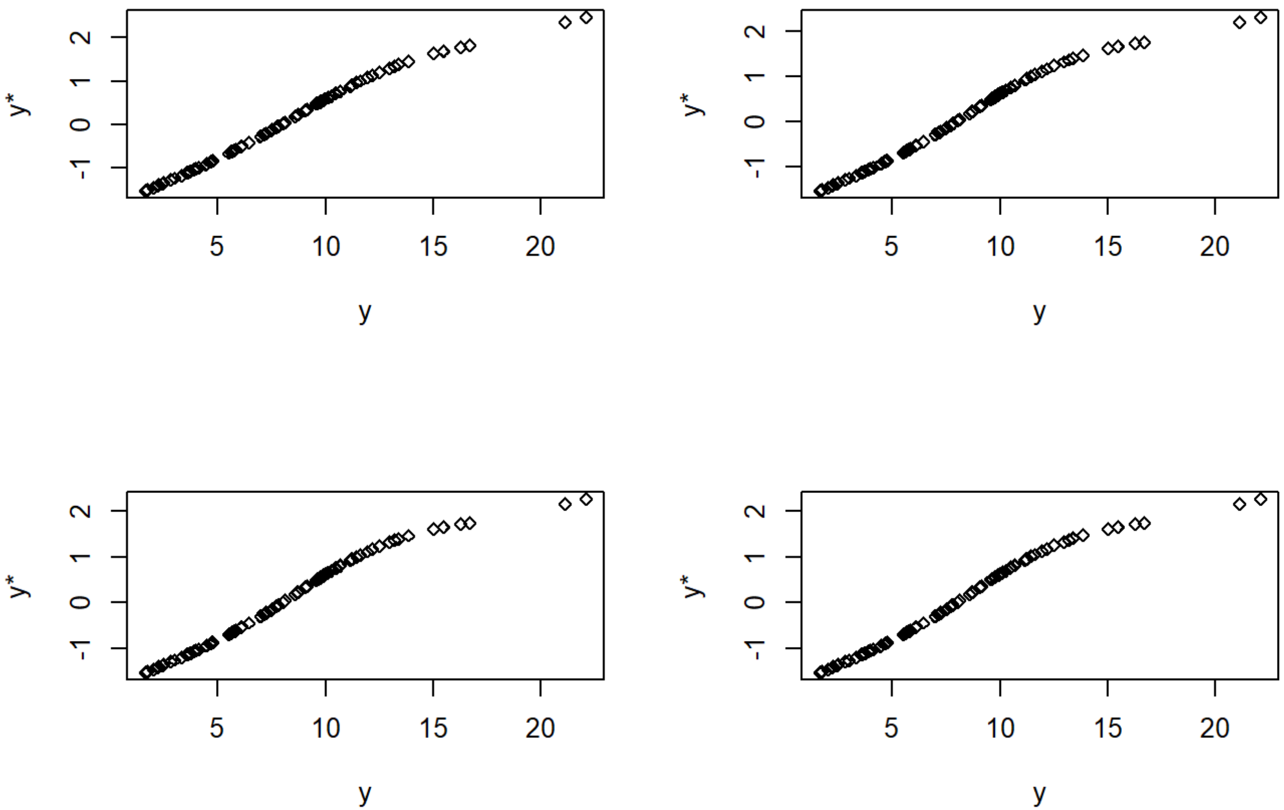
lambda가 같아서 m1과 m2가 똑같은 형태가 나오는데,

만약, lambda를 바꾼다면 다음과 같이 fit되는것을 볼 수 있다.

 a를 0.001로, x2의 lambda를 0.03, y의 lambda를 0.003으로 주었다.

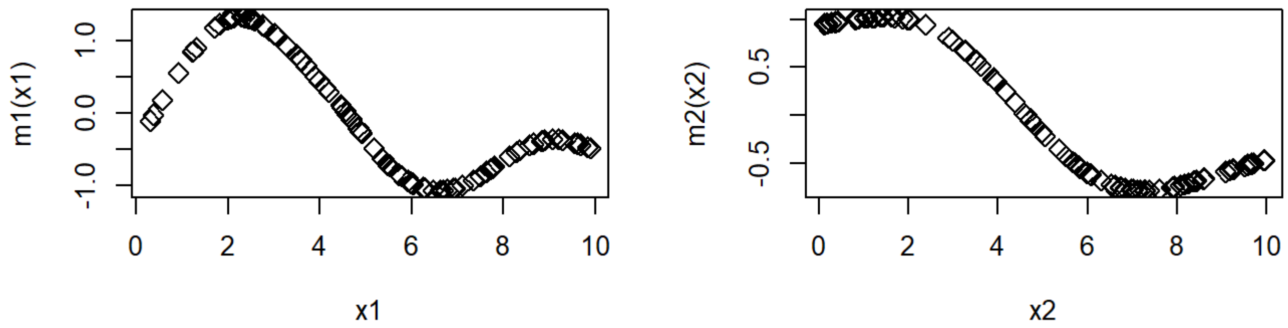
```r
nd <- 100
set.seed(100)
xx1 <- runif(nd, min =0, max = 10)
xx2 <- runif(nd, min =0, max = 10)
yy <- (sin(0.2 * pi * xx1) + xx2^2 * 0.05 - xx2 * 0.4
  + 3 + rnorm(nd, mean =0, sd = 0.1))^2
# (2)
lambx1 <- 0.001
lambx2 <- 0.03
lamby <- 0.003
# (3)
par(mfrow = c(2, 2))
# (4)
for(it in 1:4) {
  yyst <- aceit1_smooth(nd, it, xx1, xx2, yy, lambx1, lambx2, lamby)
  plot(yy, yyst, type = "n", xlab = "y", ylab = "y*")
  points(yy, yyst, cex = 0.7, pch =5)
  }
```

Processing math: 100%




```
fit.sp1 <- smooth.spline(xx1, yyst, lambda = lambx1, all.knots = TRUE)
fit.sp2 <- smooth.spline(xx1, yyst, lambda = lambx2, all.knots = TRUE)

fit.tmd1 <- predict(fit.sp1, xx1)
fit.tmd2 <- predict(fit.sp2, xx2)
eyd1 <- fit.tmd1$y
eyd2 <- fit.tmd2$y
yhat <- eyd1 + eyd2
plot(xx1, eyd1, type = "n", xlab = "x1", ylab = "m1(x1)")
```

```
points(xx1, eyd1, cex = 1.2, pch =5)
plot(xx2, eyd2, type = "n", xlab = "x2", ylab = "m2(x2)")
points(xx2, eyd2, cex = 1.2, pch =5)
```



두 m의 형태가 서로 다름을 볼 수 있다.