Augmented Feature Matrix Approach

Marshall Honaker

12/3/2020

First, let's import the necessary packages:

```
library(fields)
library(RandomFields)
library(mvtnorm)
library(ggplot2)
library(matrixcalc)
library(gstat)
library(GpGp)
```

Dr. Sang's recommendations:

> The model is Y=quantile regression (X_beta + X_basis*beta_spline) We can then use the current framework, but now with (p+# of basis) estimating equations, and the working weight matrix is calculated using X=(covariates, spatial basis covariates).

Sample code provided by Dr. Sang:

```
library(FRK)
n <- 500
coords <- cbind(runif(n), runif(n))
G <- auto_basis(manifold = plane(), data = coords, regular = 0, max_basis=20,
           nres = 2, type = "bisquare")
```

```
## ...Automatically choosing number of functions...
```

```
## Loading required namespace: INLA
```

```
## this is the design matrix corresponding to 20 basis functions
X_basis <- eval_basis(G, coords)
# matrix(X_basis). you can convert sparse basis design matrix to regular matrix if needed
dim(X_basis)
```

```
## [1] 500  22
```

We can use this to generate the matrix of spatial basis covariates that we will append to the original feature matrix.

Next we will write a method to obtain the desired augmented feature matrix. We will scale each spatial basis covariate by its L^2 norm.

```
augmented_feature_mat <- function(y, X, locs){
  G <- auto_basis(manifold = plane(), data = locs, regular = 0, max_basis=20,
                  nres = 2, type = "bisquare")
  spatial_basis <- as.matrix(eval_basis(G, locs))
  for(i in 1:ncol(spatial_basis)){
    spatial_basis[,i] <- spatial_basis[,i]/sqrt(sum(spatial_basis[,i]^2))
```

```
  }
  return(as.matrix(cbind(X, spatial_basis))[,-1]) ## See item (1) below
}
```

Now, let's move on to the more rudimentary functions of our

```
get_score<- function(y, X, tau, beta){  ## See item (2) below
  n <- nrow(X)
  temp <- t(X[1,,drop=FALSE])*ifelse(y[1] - X[1,,drop=FALSE]%*%as.matrix(beta) < 0, tau  - 1, tau)[1,1]
  for(i in 2:n){
    temp <- temp + t(X[i,,drop=FALSE])*ifelse(y[i] - X[i,,drop=FALSE]%*%as.matrix(beta) < 0, tau  - 1,
  }
  return(temp)
}

weight <- function(X, tau){  ## See item (4) below
  n <- nrow(X)
  temp <- X[1,]%*%t(X[1,])
  for(i in 2:n){
    temp <- temp + X[i,]%*%t(X[i,])
  }
  coef <- n/(tau*(1 - tau))
  return(coef*solve(temp))
}

get_likelihood <- function(y, X, tau, beta, locs, C){ ## See item (5) below
  score <- get_score(y, X, tau, beta)
  coef <- -1/(2*length(y))
  weight_X <- weight(X, tau)
  kernel <- exp(coef*(t(score)%*%weight_X%*%score)[1,1])
  return(C*kernel)
}
```

A few comments on the methods above:

1. The `augmented_feature_mat()` method was returning a feature matrix with the column of 1s for the intercept. `R` will include this automatically, so I modified the `return()` statement so the output would not include this column.
2. The `get_score()` function outputs a p length vector whose entries correspond to the features in the feature matrix. The last element in this vector (corresponding to the additional columns we get by appending `X_basis` to out original feature matrix) is much larger than the others, so we should keep an eye on this.
3. I deleted the `get_spatial_covar_mat()` function because we no longer need it.
4. The paper specifies that the weight matrix W must be positive definite. Even though we followed the definition for W given in the paper to a tee, our function doesn't return a positive definite matrix if you consider the eighth decimal place of each entry. Since the eighth decimal isn't especially significant in the grand scheme of things, I added the `round(., 7)` method to make sure that the method returns a weight matrix that satisfies the positive definiteness requirement.
5. I'm not sure how to come up with good values of C in the likelihood function. I've been using $C = 1$ so far just so the code will run and play nice. I need to read through the paper some more to figure out how they did it.

Next, let's look at the methods for the Importance Sampling algorithm for a single quantile.

```
get_updated_params <- function(y, X, tau, beta, Sigma, locs, draw){
  n <- length(y)
```

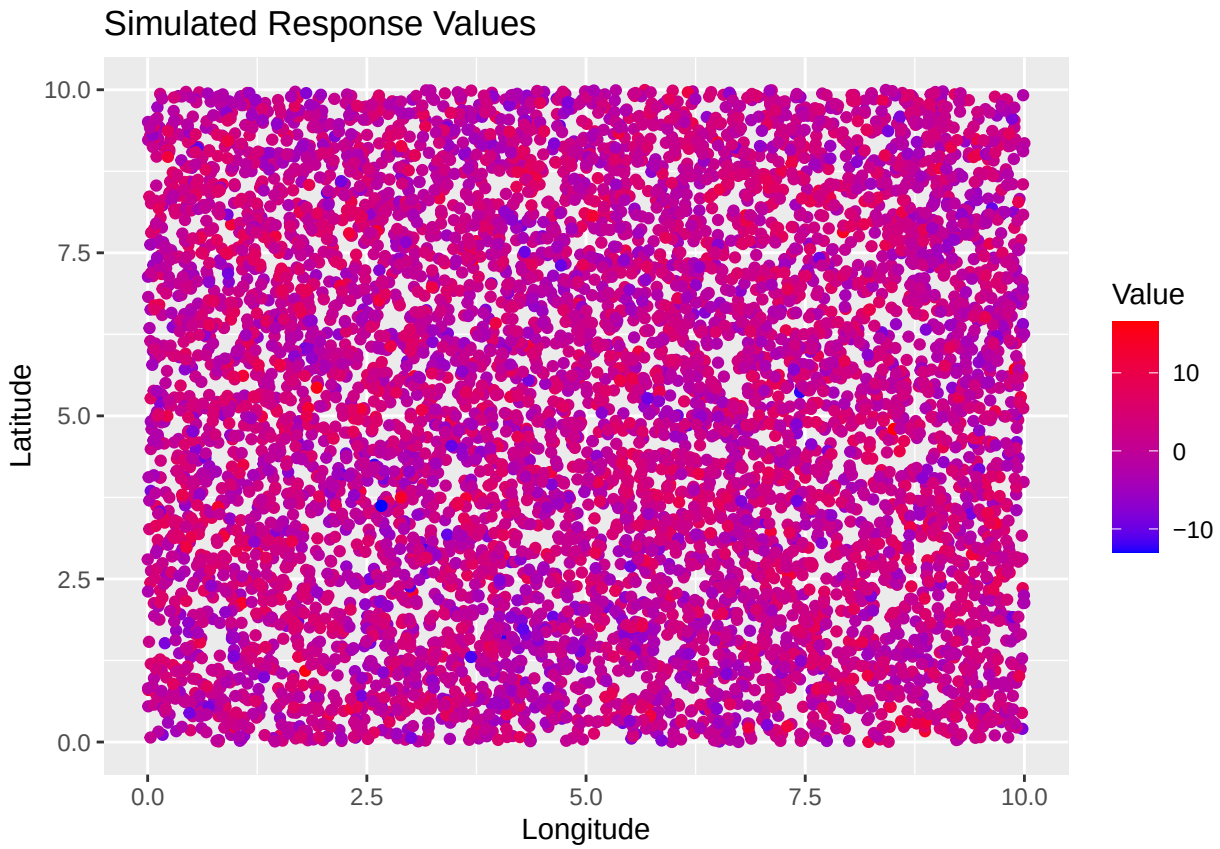
```
  p <- ncol(X)

  w <- get_likelihood(y, X, tau, beta, locs, 1)*(1/(2*n)^p)/dmvnorm(draw, beta, Sigma)

  mu_hat <- apply(w*draw, 2, sum)/sum(w)

  S <- matrix(nrow = p, ncol = p)
  for(i in 1:p){
    for(j in 1:p){
      S[i,j] <- sum(w*draw[,i]*draw[,j])/sum(w) - mu_hat[i]*mu_hat[j]
    }
  }
  temp <- list(mu_hat, S)
  names(temp) <- c("mu", "S")
  return(temp)
}

adaIS_singleQuantile <- function(y, X, tau, C, locs, M, num_reps){
  p <- ncol(X) ## Step 1: Initialize starting values for IS algorithm
  n <- length(y) ## Step 1: Initialize starting values for IS algorithm
  beta <- coef(lm(y ~ X))[-1] ## Step 1: Initialize starting values for IS algorithm
  S <- cov(X) ## Step 1: Initialize starting values for IS algorithm

  a_1 <- coef(lm(y ~ X))[-1]
  a_0 <- quantile(y - X%*%a_1, tau)
  a <- c(a_0, a_1)
  X <- cbind(rep(1, n), X)
  temp <- X[1,]%*%t(X[1,])
  for(i in 2:n){
    temp <- temp + X[i,]%*%t(X[i,])
  }
  S_0 <- 1*tau*(1 - tau)*solve((1/n)*temp)

  params <- list(a, S_0)
  names(params) <- c("mu", "S")
  draw <- rmvnorm(M, params$mu, params$S)

  # params <- list(beta, S)  ## Put our initial parameter estimates/starting values into a list called
  # names(params) <- c("mu", "S")

  for(i in 2:num_reps){ ## Step 4: Repeat steps 2 and 3 until we achieve the desired effective sample s
    draw <- rmvnorm(M, params$mu, params$S)  ## Step 2: Simulate M values from the proposal distributio
    params <- get_updated_params(y, X, tau, params$mu, params$S, locs, draw) ## Step 3: Update the para
  }
  return(params)
}
```

Next, let's simulate some data so we can test what we have so far.

First, let's generate and plot the simulated data.

```
n <- 10000
locs <- as.matrix(cbind(runif(n, 0, 10), runif(n, 0, 10)))
sigma_sq <- 2
phi <- .4
```

```r
nu <- .5
tau_sq <- .5

w <- fast_Gp_sim(c(sigma_sq, phi, nu, 0), "matern_isotropic", locs, 30)

X <- as.matrix(cbind(rep(1, n), rnorm(n), rnorm(n)))
beta <- as.matrix(c(1, 2, 3))
p <- length(beta)
y <- rnorm(n, X%*%beta + w, sqrt(tau_sq))

training_indices <- sample(1:n, round(.75*n), replace = FALSE)

y_train <- y[training_indices]
X_train <- X[training_indices,]
locs_train <- locs[training_indices,]

y_test <- y[-training_indices]
X_test <- X[-training_indices,]
locs_test <- locs[-training_indices,]

training_data <- as.data.frame(cbind(y_train, X_train[,-1], locs_train))
names(training_data) <- c("y_train", "X1_train", "X2_train", "long_train", "lat_train")

ggplot() + geom_point(data = training_data, aes(x = long_train, y = lat_train, col = y_train)) +
  scale_colour_gradient(low = "blue", high = "red") +
  labs(title = "Simulated Response Values",
       x = "Longitude",
       y = "Latitude",
       col = "Value")
```



Next, let's test out our more basic methods.

```
tau <- .5

X_augmented <- augmented_feature_mat(y_train, X_train, locs_train)
```

```
## ...Automatically choosing number of functions...
```

```
X_augmented[1:10,]
```

```
##              [,1]       [,2] [,3]         [,4]        [,5]         [,6]
##  [1,]  0.03695320  1.3053999    0 0.0000000000 0.000000000 0.0000000000
##  [2,] -1.20781641 -1.0745926    0 0.0000000000 0.000000000 0.0000000000
##  [3,] -0.58063394 -0.1738436    0 0.0006267942 0.038925195 0.0796342477
##  [4,] -0.31306584 -1.2366242    0 0.0000000000 0.000000000 0.0000000000
##  [5,] -1.64215952 -0.1908433    0 0.0010955075 0.003467456 0.0000000000
##  [6,]  0.81086489 -0.5539762    0 0.0000000000 0.000000000 0.0000000000
##  [7,] -0.78874202  0.1131865    0 0.0000000000 0.000000000 0.0000000000
##  [8,] -0.05250944  0.8628573    0 0.0001619284 0.026075981 0.0455971567
##  [9,]  1.45817131  0.2257365    0 0.0000000000 0.000000000 0.0000000000
## [10,] -0.15358254  1.1384114    0 0.0000000000 0.006439801 0.0007375459
##              [,7] [,8]        [,9]       [,10]        [,11] [,12]       [,13]
##  [1,] 0.000000e+00    0 0.000000000 0.00000000 0.0000000000     0 0.022500167
##  [2,] 0.000000e+00    0 0.000000000 0.00000000 0.0000000000     0 0.017006094
##  [3,] 1.916277e-02    0 0.000000000 0.00000000 0.0000000000     0 0.000000000
##  [4,] 0.000000e+00    0 0.000000000 0.00000000 0.0000000000     0 0.018856512
##  [5,] 9.742431e-05    0 0.000000000 0.00000000 0.0000000000     0 0.018673661
##  [6,] 0.000000e+00    0 0.006793340 0.02085830 0.0004101751     0 0.009486886
```

```
##  [7,] 0.000000e+00    0 0.000852363 0.01793082 0.0009163113     0 0.011765798
##  [8,] 3.515540e-02    0 0.000000000 0.00000000 0.0000000000     0 0.002132370
##  [9,] 0.000000e+00    0 0.000000000 0.00000000 0.0000000000     0 0.016961801
## [10,] 2.360512e-02    0 0.000000000 0.00000000 0.0000000000     0 0.011641170
##              [,14]        [,15]       [,16]       [,17]        [,18]
##  [1,] 0.0001458333 6.130049e-03 0.009758591 0.010419769 7.072718e-03
##  [2,] 0.0000000000 1.990760e-02 0.000000000 0.000000000 0.000000e+00
##  [3,] 0.0000000000 0.000000e+00 0.000000000 0.007583421 0.000000e+00
##  [4,] 0.0000000000 2.111466e-02 0.000000000 0.001765673 0.000000e+00
##  [5,] 0.0000000000 5.993618e-03 0.006545042 0.022700024 0.000000e+00
##  [6,] 0.0000000000 1.372844e-02 0.000000000 0.000000000 0.000000e+00
##  [7,] 0.0000000000 1.233497e-02 0.000000000 0.000000000 4.263043e-05
##  [8,] 0.0000000000 9.625808e-05 0.000000000 0.015683361 0.000000e+00
##  [9,] 0.0146484819 0.000000e+00 0.021709765 0.004165516 2.226880e-02
## [10,] 0.0000000000 8.980511e-03 0.000000000 0.018433516 0.000000e+00
##             [,19]        [,20]      [,21]       [,22]       [,23]      [,24]
##  [1,] 0.004717796 0.0000000000 0.00000000 0.000000000 0.009585746 0.00000000
##  [2,] 0.015733486 0.0006647737 0.00488449 0.000000000 0.006791228 0.00000000
##  [3,] 0.000000000 0.0000000000 0.00000000 0.000000000 0.003168521 0.03125516
##  [4,] 0.006871162 0.0041438397 0.00000000 0.000000000 0.015673232 0.00000000
##  [5,] 0.000000000 0.0000000000 0.00000000 0.000000000 0.020516654 0.01067047
##  [6,] 0.021968622 0.0000000000 0.03465123 0.000000000 0.000000000 0.00000000
##  [7,] 0.025989966 0.0000000000 0.02417640 0.000000000 0.000000000 0.00000000
##  [8,] 0.000000000 0.0019941032 0.00000000 0.000000000 0.016786094 0.03719566
##  [9,] 0.001194497 0.0000000000 0.00000000 0.001636349 0.000000000 0.00000000
## [10,] 0.000000000 0.0089043764 0.00000000 0.000000000 0.028887698 0.02468159
##              [,25]
##  [1,] 0.0000000000
##  [2,] 0.0000000000
##  [3,] 0.0000000000
##  [4,] 0.0000000000
##  [5,] 0.0000000000
##  [6,] 0.0004864137
##  [7,] 0.0028729498
##  [8,] 0.0000000000
##  [9,] 0.0007717118
## [10,] 0.0000000000
```

```
beta <- coef(lm(y_train ~ X_augmented))[-1]
score <- get_score(y_train, X_augmented, tau, beta)
score
```

```
##            [,1]
##  [1,]  12.93964
##  [2,] -58.92972
##  [3,]  11.41769
##  [4,]  17.48384
##  [5,]  17.08167
##  [6,]  10.74156
##  [7,]  15.54802
##  [8,]  13.15793
##  [9,]  12.88510
## [10,]  15.09248
## [11,]  14.11452
## [12,]  13.31164
```

```
## [13,]  34.16047
## [14,]  19.87559
## [15,]  25.09839
## [16,]  24.48092
## [17,]  24.45314
## [18,]  21.86595
## [19,]  23.04089
## [20,]  17.39634
## [21,]  15.20366
## [22,]  15.85234
## [23,]  24.78885
## [24,]  19.15024
## [25,]  19.09134
```

```
weight_mat <- weight(X_augmented, tau)
weight_mat
```

```
##                [,1]          [,2]          [,3]          [,4]          [,5]
##  [1,]  4.039160e+00   0.009202206 -3.453324e+01 -9.141757e+01  9.964012e+01
##  [2,]  9.202206e-03   4.001593166  2.648576e+01  1.555910e+01 -7.011270e+01
##  [3,] -3.453324e+01  26.485763725  6.335128e+05  6.944713e+05 -5.278169e+05
##  [4,] -9.141757e+01  15.559103117  6.944713e+05  2.525244e+06 -1.956909e+06
##  [5,]  9.964012e+01 -70.112700911 -5.278169e+05 -1.956909e+06  2.463317e+06
##  [6,] -1.557656e+01  38.076261179  2.017333e+05  3.988936e+05 -6.188783e+05
##  [7,]  3.563984e+01 -19.114743617 -2.836201e+05 -1.143232e+06  1.247789e+06
##  [8,] -1.067305e+01 -51.016379575  9.401254e+04  2.358744e+05  4.865257e+05
##  [9,]  2.008161e+01  15.426210443 -1.244605e+04 -1.885028e+05 -1.556948e+05
## [10,]  2.440684e+00  13.029719858  1.927776e+04 -1.867506e+05  6.059262e+04
## [11,]  1.843669e+01  15.165949221 -6.249028e+04 -1.979742e+05 -3.121226e+04
## [12,]  2.082305e+01  15.472277389  2.313987e+05 -1.674719e+05  9.319178e+04
## [13,] -6.803726e+01 -46.733746102 -8.436132e+05  1.791890e+06 -1.153816e+06
## [14,] -3.303640e+01 -17.808361848 -6.532876e+05  5.657527e+05 -5.105461e+05
## [15,]  2.728093e+01  45.926216675  2.182558e+05 -1.145108e+06  5.778968e+05
## [16,] -7.099960e+00  -9.494600086  9.914162e+05  3.806710e+05  3.031149e+05
## [17,]  6.401631e+01  88.937389301 -4.317994e+05 -2.459753e+06  8.885382e+05
## [18,]  5.400538e+01  51.583943669  3.802990e+05 -1.475071e+06  6.737675e+05
## [19,]  3.911320e+01  23.592806911  4.006694e+05 -7.351271e+05  3.669840e+05
## [20,] -1.516344e+01  52.114255129 -3.080161e+05  6.343687e+05 -1.168501e+06
## [21,] -1.849251e+01 -24.467703566 -8.979287e+04  5.376365e+05 -1.357681e+05
## [22,]  8.852436e+01 -21.125966082 -1.118238e+06 -2.143106e+06  1.360336e+06
## [23,]  2.205645e+01 -52.092120676  6.897490e+05 -8.222453e+04  8.358676e+05
## [24,] -1.141394e+02   4.982550653  3.488437e+05  2.840909e+06 -2.713915e+06
## [25,] -6.646073e+01 -66.175571167 -3.086476e+05  1.089693e+06 -3.043516e+05
##                [,6]          [,7]          [,8]          [,9]         [,10]
##  [1,]     -15.57656  3.563984e+01 -1.067305e+01      20.08161  2.440684e+00
##  [2,]      38.07626 -1.911474e+01 -5.101638e+01      15.42621  1.302972e+01
##  [3,]  201733.25339 -2.836201e+05  9.401254e+04  -12446.05433  1.927776e+04
##  [4,]  398893.63669 -1.143232e+06  2.358744e+05 -188502.78198 -1.867506e+05
##  [5,] -618878.34561  1.247789e+06  4.865257e+05 -155694.81576  6.059262e+04
##  [6,]  405712.56262 -3.377209e+05 -2.166219e+05  170489.21283 -8.810801e+04
##  [7,] -337720.89007  1.150390e+06  4.508470e+05 -299751.55168  9.568460e+04
##  [8,] -216621.92992  4.508470e+05  1.287538e+06 -877962.25374 -2.905255e+05
##  [9,]  170489.21283 -2.997516e+05 -8.779623e+05  814399.99854  2.500336e+05
## [10,]  -88108.00818  9.568460e+04 -2.905255e+05  250033.62461  5.212553e+05
## [11,]  153527.07575  2.400072e+04 -5.454709e+04    9753.64683 -3.152856e+05
```

```
## [12,]   93228.28567 -8.473634e+04 -2.029474e+05  303875.16001  1.462768e+05
## [13,] -463191.01328 -6.026599e+05  1.060733e+05 -451296.75652  3.607477e+04
## [14,] -100201.60048 -1.770289e+05 -6.678939e+04 -142828.88323 -1.490954e+05
## [15,]  -60943.53980  7.260792e+05 -1.125695e+05   -8200.89571  3.591885e+05
## [16,]   61497.82645 -2.432133e+05  5.540050e+05  -77218.11314 -3.074215e+04
## [17,]  221915.02681  1.035549e+06 -8.916607e+05  529759.19735  1.837816e+05
## [18,]  321663.39000  5.279021e+05 -3.648056e+05  404285.51299  1.043662e+05
## [19,]  406284.69913 -3.196586e+04 -3.149098e+05  548552.67098 -2.074058e+05
## [20,]  224766.76989 -9.794789e+05 -1.152192e+06  643835.25029  3.656696e+04
## [21,]  -72707.48416 -8.224231e+04  6.036674e+05 -579937.02256 -4.163296e+05
## [22,] -345940.28901  9.227093e+05 -4.520623e+05  196085.66828  1.405152e+05
## [23,]  291506.04450  8.477842e+03  7.714440e+05  -32450.45184 -4.066761e+05
## [24,]   70661.21561 -1.799339e+06 -3.220261e+05  -91315.11898  7.065252e+04
## [25,] -474008.85455 -7.286022e+04  6.888229e+05 -780220.46134  2.208158e+04
##               [,11]         [,12]         [,13]         [,14]         [,15]
##  [1,]      18.43669  2.082305e+01 -6.803726e+01 -3.303640e+01  2.728093e+01
##  [2,]      15.16595  1.547228e+01 -4.673375e+01 -1.780836e+01  4.592622e+01
##  [3,]  -62490.27500  2.313987e+05 -8.436132e+05 -6.532876e+05  2.182558e+05
##  [4,] -197974.21141 -1.674719e+05  1.791890e+06  5.657527e+05 -1.145108e+06
##  [5,]  -31212.25756  9.319178e+04 -1.153816e+06 -5.105461e+05  5.778968e+05
##  [6,]  153527.07575  9.322829e+04 -4.631910e+05 -1.002016e+05 -6.094354e+04
##  [7,]   24000.71889 -8.473634e+04 -6.026599e+05 -1.770289e+05  7.260792e+05
##  [8,]  -54547.08929 -2.029474e+05  1.060733e+05 -6.678939e+04 -1.125695e+05
##  [9,]    9753.64683  3.038752e+05 -4.512968e+05 -1.428289e+05 -8.200896e+03
## [10,] -315285.64949  1.462768e+05  3.607477e+04 -1.490954e+05  3.591885e+05
## [11,]  546037.96071 -2.772149e+03 -5.573896e+05 -5.436909e+04  1.913432e+04
## [12,]   -2772.14920  5.170484e+05 -1.351631e+06 -7.405877e+05  3.605976e+05
## [13,] -557389.58213 -1.351631e+06  8.641583e+06  3.715461e+06 -3.143380e+06
## [14,]  -54369.09492 -7.405877e+05  3.715461e+06  2.115617e+06 -1.412662e+06
## [15,]   19134.32256  3.605976e+05 -3.143380e+06 -1.412662e+06  2.172567e+06
## [16,] -355168.68891  7.903933e+05 -3.745249e+06 -2.201621e+06  1.044319e+06
## [17,]  683380.96695  3.141501e+05 -3.371470e+06 -8.573373e+05  1.938872e+06
## [18,]  659863.81556  7.957272e+05 -5.020351e+06 -2.229290e+06  2.033995e+06
## [19,]  480065.91500  8.704675e+05 -4.383112e+06 -1.737689e+06  1.068033e+06
## [20,]   42843.11622 -2.256407e+05  2.652817e+06  1.316875e+06 -1.324250e+06
## [21,]   61297.83427 -4.174738e+05  1.457163e+06  6.519523e+05 -7.972191e+05
## [22,]  356621.19339 -2.404429e+05  4.358003e+05  6.170821e+05  3.092644e+05
## [23,]  219862.64837  7.512577e+05 -4.318819e+06 -2.018485e+06  5.732802e+05
## [24,] -537701.37450 -4.646264e+05  4.440700e+06  1.530709e+06 -1.714831e+06
## [25,] -825751.13035 -1.022003e+06  4.407745e+06  1.791517e+06 -1.255817e+06
##               [,16]         [,17]         [,18]         [,19]         [,20]
##  [1,] -7.099960e+00  6.401631e+01  5.400538e+01  3.911320e+01 -1.516344e+01
##  [2,] -9.494600e+00  8.893739e+01  5.158394e+01  2.359281e+01  5.211426e+01
##  [3,]  9.914162e+05 -4.317994e+05  3.802990e+05  4.006694e+05 -3.080161e+05
##  [4,]  3.806710e+05 -2.459753e+06 -1.475071e+06 -7.351271e+05  6.343687e+05
##  [5,]  3.031149e+05  8.885382e+05  6.737675e+05  3.669840e+05 -1.168501e+06
##  [6,]  6.149783e+04  2.219150e+05  3.216634e+05  4.062847e+05  2.247668e+05
##  [7,] -2.432133e+05  1.035549e+06  5.279021e+05 -3.196586e+04 -9.794789e+05
##  [8,]  5.540050e+05 -8.916607e+05 -3.648056e+05 -3.149098e+05 -1.152192e+06
##  [9,] -7.721811e+04  5.297592e+05  4.042855e+05  5.485527e+05  6.438353e+05
## [10,] -3.074215e+04  1.837816e+05  1.043662e+05 -2.074058e+05  3.656696e+04
## [11,] -3.551687e+05  6.833810e+05  6.598638e+05  4.800659e+05  4.284312e+04
## [12,]  7.903933e+05  3.141501e+05  7.957272e+05  8.704675e+05 -2.256407e+05
## [13,] -3.745249e+06 -3.371470e+06 -5.020351e+06 -4.383112e+06  2.652817e+06
```

```
## [14,] -2.201621e+06 -8.573373e+05 -2.229290e+06 -1.737689e+06  1.316875e+06
## [15,]  1.044319e+06  1.938872e+06  2.033995e+06  1.068033e+06 -1.324250e+06
## [16,]  3.889665e+06 -9.501311e+05  1.178815e+06  1.814797e+06 -1.701681e+06
## [17,] -9.501311e+05  4.462001e+06  2.958186e+06  1.652118e+06 -3.881872e+05
## [18,]  1.178815e+06  2.958186e+06  3.762722e+06  2.508507e+06 -1.288946e+06
## [19,]  1.814797e+06  1.652118e+06  2.508507e+06  2.772167e+06 -8.632237e+05
## [20,] -1.701681e+06 -3.881872e+05 -1.288946e+06 -8.632237e+05  2.274559e+06
## [21,] -3.892468e+05 -1.006025e+06 -1.011030e+06 -7.989081e+05  1.002377e+05
## [22,] -2.215525e+06  2.273490e+06  6.095184e+05 -2.813939e+05  3.696823e+05
## [23,]  3.001574e+06 -9.032447e+04  1.950649e+06  2.448895e+06 -1.871600e+06
## [24,] -5.934808e+05 -3.410190e+06 -3.004086e+06 -2.147092e+06  2.010512e+06
## [25,] -9.367067e+05 -2.554893e+06 -3.286615e+06 -2.825979e+06  5.585155e+05
##               [,21]         [,22]         [,23]         [,24]         [,25]
##  [1,] -1.849251e+01  8.852436e+01  2.205645e+01 -1.141394e+02 -6.646073e+01
##  [2,] -2.446770e+01 -2.112597e+01 -5.209212e+01  4.982551e+00 -6.617557e+01
##  [3,] -8.979287e+04 -1.118238e+06  6.897490e+05  3.488437e+05 -3.086476e+05
##  [4,]  5.376365e+05 -2.143106e+06 -8.222453e+04  2.840909e+06  1.089693e+06
##  [5,] -1.357681e+05  1.360336e+06  8.358676e+05 -2.713915e+06 -3.043516e+05
##  [6,] -7.270748e+04 -3.459403e+05  2.915060e+05  7.066122e+04 -4.740089e+05
##  [7,] -8.224231e+04  9.227093e+05  8.477842e+03 -1.799339e+06 -7.286022e+04
##  [8,]  6.036674e+05 -4.520623e+05  7.714440e+05 -3.220261e+05  6.888229e+05
##  [9,] -5.799370e+05  1.960857e+05 -3.245045e+04 -9.131512e+04 -7.802205e+05
## [10,] -4.163296e+05  1.405152e+05 -4.066761e+05  7.065252e+04  2.208158e+04
## [11,]  6.129783e+04  3.566212e+05  2.198626e+05 -5.377014e+05 -8.257511e+05
## [12,] -4.174738e+05 -2.404429e+05  7.512577e+05 -4.646264e+05 -1.022003e+06
## [13,]  1.457163e+06  4.358003e+05 -4.318819e+06  4.440700e+06  4.407745e+06
## [14,]  6.519523e+05  6.170821e+05 -2.018485e+06  1.530709e+06  1.791517e+06
## [15,] -7.972191e+05  3.092644e+05  5.732802e+05 -1.714831e+06 -1.255817e+06
## [16,] -3.892468e+05 -2.215525e+06  3.001574e+06 -5.934808e+05 -9.367067e+05
## [17,] -1.006025e+06  2.273490e+06 -9.032447e+04 -3.410190e+06 -2.554893e+06
## [18,] -1.011030e+06  6.095184e+05  1.950649e+06 -3.004086e+06 -3.286615e+06
## [19,] -7.989081e+05 -2.813939e+05  2.448895e+06 -2.147092e+06 -2.825979e+06
## [20,]  1.002377e+05  3.696823e+05 -1.871600e+06  2.010512e+06  5.585155e+05
## [21,]  8.095225e+05 -2.109875e+05 -2.166576e+05  6.835154e+05  1.052371e+06
## [22,] -2.109875e+05  2.892125e+06 -1.317239e+06 -1.821300e+06 -4.552216e+05
## [23,] -2.166576e+05 -1.317239e+06  3.967067e+06 -2.228076e+06 -1.923406e+06
## [24,]  6.835154e+05 -1.821300e+06 -2.228076e+06  5.428918e+06  2.456164e+06
## [25,]  1.052371e+06 -4.552216e+05 -1.923406e+06  2.456164e+06  3.752210e+06
```

```
#is.positive.definite(weight_mat)

lik <- get_likelihood(y_train, X_augmented, tau, beta, locs_train, 1)
lik
```

```
## [1] 0
```

As can be seen above, the code for the more basic methods runs. However, we still run into the same issues we discussed earlier.

Now, let's move on to the Importance Sampling methods.

```
Sigma <- cov(X_augmented)
draw <- rmvnorm(1, beta, Sigma)
pars <- get_updated_params(y_train, X_augmented, tau, beta, Sigma, locs_train, draw)
pars
```

```
## $mu
```

```
##  X_augmented1  X_augmented2  X_augmented3  X_augmented4  X_augmented5
##           NaN           NaN           NaN           NaN           NaN
##  X_augmented6  X_augmented7  X_augmented8  X_augmented9 X_augmented10
##           NaN           NaN           NaN           NaN           NaN
## X_augmented11 X_augmented12 X_augmented13 X_augmented14 X_augmented15
##           NaN           NaN           NaN           NaN           NaN
## X_augmented16 X_augmented17 X_augmented18 X_augmented19 X_augmented20
##           NaN           NaN           NaN           NaN           NaN
## X_augmented21 X_augmented22 X_augmented23 X_augmented24 X_augmented25
##           NaN           NaN           NaN           NaN           NaN
##
## $S
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
##  [2,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
##  [3,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
##  [4,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
##  [5,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
##  [6,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
##  [7,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
##  [8,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
##  [9,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [10,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [11,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [12,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [13,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [14,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [15,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [16,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [17,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [18,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [19,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [20,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [21,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [22,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [23,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [24,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
## [25,]  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN   NaN   NaN   NaN   NaN
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
##  [1,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
##  [2,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
##  [3,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
##  [4,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
##  [5,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
##  [6,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
##  [7,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
##  [8,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
##  [9,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [10,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [11,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [12,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [13,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [14,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [15,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
```

```
## [16,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [17,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [18,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [19,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [20,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [21,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [22,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [23,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [24,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
## [25,]   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
```

```
## post_dist_params <- adaIS_singleQuantile(y_train, X_augmented, tau, 1, locs_train, 5000, 10)
## get_updated_params() isn't working, so we can't expect adaIS_singleQuantile() to work either
```

Because this is a Bayesian regression model, we will need to derive the posterior predictive distribution to make predictions at the test locations using the test data. Once the Importance Sampling methods work and give us proper values for the parameters of the posterior distribution, we can make the desired predictions to evaluate the predictive performance of our model.